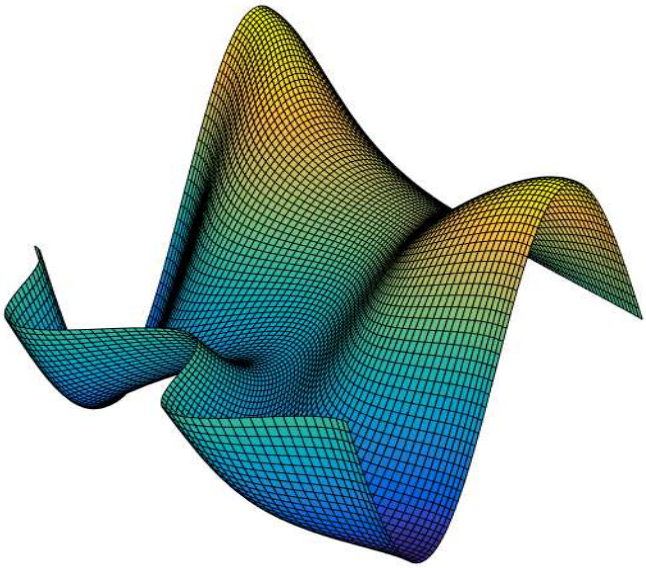
TEXTOS DE MATEMÁTICA
EDITORA INSTITUTO DE MATEMÁTICA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

INTRODUÇÃO AO MÉTODO DE ELEMENTOS FINITOS

— MAURO A. RINCON —
e
— I-SHIH LIU —



Introdução ao Método de Elementos Finitos

Computação e Análise em Equações Diferenciais Parciais

Mauro A. Rincon
I-Shih Liu

Universidade Federal do Rio de Janeiro
Instituto de Matemática

2020

Prefácio

O presente livro tem sua origem em 1997, quando visávamos obter soluções numéricas de equações diferenciais parciais do tipo elíptico, sobretudo aquelas relacionadas aos modelos matemáticos da Teoria da Elasticidade Linear. As questões matemáticas sobre existência, unicidade e regularidade de solução foram tratadas incialmente e, na sequência, para as simulações numéricas, optamos pelo Método de Elementos Finitos. Como os programas computacionais em linguagem Fortran inicialmente utilizados demostraram não ser suficientemente didáticos aos nossos propósitos, decidimos desenvolver nossos próprios programas computacionais utilizando a linguagem C. Eles são simples e de fácil entendimento.

O livro está organizado da seguinte forma:

No Capítulo 1 são abordados os seguintes tópicos da Física Matemática: a Equação da Condução de Calor e a Teoria da Elasticidade Linear. São esses os modelos que são tratados com o método de elementos finitos nos próximos capítulos do livro. Para a melhor compreensão da dedução dos modelos, são consideradas as leis de conservação, as equações constitutivas lineares e o conceito de pequenas deformações. Também são feitas comparações entre alguns métodos numéricos, tendo como objetivo motivar o estudo do método de elementos finitos. No final deste capítulo, apresentamos o método de aproximação por diferenças finitas, que será utilizado nas equações de evolução tratadas nos capítulos 6 e 7.

O Capítulo 2 trata de um problema modelo estacionário unidimensional com condições na fronteira. Alguns exemplos numéricos com vários tipos de valores de fronteira são apresentados e comparados com as respectivas soluções exatas. Os erros nas normas de $L^2(\Omega)$ e $H^1(\Omega)$ são mostrados em cada um dos exemplos dados.

No Capítulo 3 introduzimos as usuais funções base de ordem superior, mais precisamente, a função base quadrática, a base spline cúbica e a função base de Hermite. Estimativas de erros em espaços de Sobolev são obtidas para o problema elíptico modelo.

No Capítulo 4 apresentamos o problema estacionário bidimensional. São estudadas as formulações forte e fraca, a existência e a unicidade de solução. Também aqui consideramos vários tipos de valores de fronteira, com as soluções numéricas compa-

radas com as respectivas soluções exatas. Para resolver o sistema linear associado ao método de elementos finitos, utilizamos o algoritmo de Crout. Concluímos o capítulo mostrando os gráficos das soluções e os erros nas normas $L^2(\Omega)$ e $H^1(\Omega)$.

O Capítulo 5 é dedicado ao problema modelo de elasticidade linear no caso bidimensional. Os mesmos tópicos do Capítulo 4 são abordados, considerando-se, no entanto, a solução vetorial do problema. Simulações numéricas, erros e gráficos são apresentados.

No Capítulo 6 introduzimos alguns dos métodos numéricos mais conhecidos da literatura e os algoritmos para resolver o problema parabólico modelo: `Equação do calor`. São apresentados alguns exemplos numéricos e comparações entre os diversos métodos numéricos, assim como gráficos e tabelas de erros.

No Capítulo 7 abordamos alguns dos mais conhecidos métodos numéricos da literatura e seus algoritmos para resolver o problema hiperbólico modelo: `Equação da onda`. E, como no Capítulo 6, são apresentados exemplos, comparações entre os diversos métodos numéricos, gráficos e tabelas de erros.

Os Capítulos 8, 9 e 10 são complementares, mais indicados a alunos que desejem se aprofundar na Análise Numérica e Matemática. Assim, não são necessários num primeiro curso cujo foco seja os métodos numéricos desenvolvidos nos capítulos anteriores.

O Apêndice A contém os programas computacionais utilizados na obtenção das soluções numéricas dos modelos estacionários tratados no presente texto. As variáveis, funções e subrotinas dos programas são referidos no texto e no índice remissivo usando-se a fonte *typewriter*. Por exemplo: `Nel, Phi, Solver` etc.

Devido à sua forma simples e didática, esperamos que este livro possa cumprir o objetivo desejado: o de um livro texto para um primeiro curso sobre o Método de Elementos Finitos dedicado a alunos de iniciação científica e mestrado interessados em análise numérica de equações diferenciais parciais.

Queremos expressar nossos agradecimentos a todos os alunos e professores que nos enviaram correções e sugestões e, em particular, aos alunos e colegas da área de Algoritmos e Métodos Numéricos do Programa de Pós-Graduação em Informática (PPGI).

Receberemos com prazer críticas e sugestões que venham a contribuir para o aperfeiçoamento deste livro.

I-S. Liu
M. A. Rincon

Sumário

CAPÍTULO 1

Introdução

Embora os modelos da Física Matemática para a equação de condução do calor e da elasticidade linear sejam, na sua forma mais geral, respectivamente, do tipo parabólico e hiperbólico, ambas se reduzem a problemas do tipo elíptico em regimes estacionários. Sendo assim, vamos inicialmente considerar as equações do tipo elíptico para desenvolver o método de elementos finitos.

Vale observar que o problema da condução do calor em um dado corpo é formulado por uma equação escalar, enquanto que nas deformações elásticas de um corpo, o modelo envolve um sistema de equações para as coordenadas do vetor deslocamento.

1.1 Condução do Calor

Consideremos um corpo rígido ocupando uma região $V \subset \mathbb{R}^3$. Sejam ϱ e e respectivamente a densidade de massa e a densidade da energia (interna) do corpo. Seja $\Omega \subset V$ uma região fixa arbitrária com fronteira suave $\partial\Omega$.

1.1.1 Equação da Energia

A variação da energia total em Ω é geralmente atribuída ao fluxo da energia q através da fronteira e ao suprimento de energia r dentro da região devido a fontes externas. Esta relação pode ser representada por

$$\frac{d}{dt}\int_\Omega \varrho e\, dx = \int_{\partial\Omega} q\, d\Gamma + \int_\Omega \varrho r\, dx. \tag{1.1}$$

Seja $\boldsymbol{n}$ o vetor normal unitário externo na fronteira e $\boldsymbol{h}$ o vetor fluxo do calor. Então o fluxo de energia q entrando no corpo pode ser expressado como

$$q = -\boldsymbol{h}\cdot\boldsymbol{n}. \tag{1.2}$$

Sendo Ω uma região fixa com fronteira suficientmente suave, obtemos de (1.1), (1.2) e do Teorema da Divergência,

$$\int_\Omega \Big\{ \varrho \frac{\partial e}{\partial t} + \operatorname{div} \boldsymbol{h} - \varrho r \Big\} dx = 0. \tag{1.3}$$

Note que se o corpo é rígido, e portanto indeformável, a densidade ϱ não varia com o tempo. Assim, para obter a relação acima na sua forma local utilizamos o seguinte teorema:

Teorema. *Se $f \in C(V, \mathbb{R})$ e*

$$\int_\Omega f \, dx = 0, \quad \forall\, \Omega \subset V,$$

então $f(\boldsymbol{x}) = 0$ para todo $\boldsymbol{x} \in V$.

Como a relação (1.3) é válida para qualquer $\Omega \subset V$, o teorema acima nos garante que

$$\varrho \frac{\partial e}{\partial t} + \operatorname{div} \boldsymbol{h} - \varrho r = 0 \quad \text{em } V. \tag{1.4}$$

Esta equação é conhecida como equação da energia.

1.1.2 Equações Constitutivas

Para problemas de condução de calor, a quantidade física mais importante é a temperatura, que não aparece explicitamente na equação da energia. Com efeito, precisamos das equações constitutivas que relacionam a energia e o fluxo do calor para a temperatura do corpo $u(\boldsymbol{x}, t)$, de uma maneira dependente do material. As relações lineares, muito usadas em aplicações práticas, são dadas por

$$e(\boldsymbol{x}, t) = c(\boldsymbol{x})\, u(\boldsymbol{x}, t),$$

e a lei de Fourier para a condução do calor,

$$\boldsymbol{h}(\boldsymbol{x}, t) = -Q(\boldsymbol{x}) \nabla u(\boldsymbol{x}, t), \quad \text{ou} \quad h_i = -\sum_j Q_{ij} \frac{\partial u}{\partial x_j}, \tag{1.5}$$

onde c é o calor específico e Q o tensor de condutividade térmica. Com base na termodinâmica, podemos admitir que $c > 0$ e a matriz Q é simétrica e definida positiva, *i.e.*, para qualquer vetor não nulo $\boldsymbol{v}$,

$$\boldsymbol{v} \cdot Q\boldsymbol{v} > 0.$$

Por outro lado, a energia suplementar r não é uma quantidade constitutiva; ela pode representar a energia irrradiada pelo meio ambiente, ou uma fonte de energia no interior do corpo. Se $\widehat{u}$ denota a temperatura do meio ambiente, podemos escrever

$$r(\boldsymbol{x}, t) = -\beta(\boldsymbol{x})(u(\boldsymbol{x}, t) - \widehat{u}) + \gamma(\boldsymbol{x}, t).$$

O primeiro termo do lado direito é a lei de Newton para a irrradiação, e o segundo termo representa a fonte de energia. O coeficiente β é um parâmetro do material e é uma quantidade não negativa, visto que se a temperatura do meio ambiente é menor do que a do corpo, há perda de energia do corpo.

Agora podemos reescrever a equação de energia (1.4) como uma equação diferencial para a temperatura,

$$c\,\frac{\partial u}{\partial t} - \sum_{i,j} \frac{\partial}{\partial x_i}\Big(Q_{ij}\frac{\partial u}{\partial x_j}\Big) + \beta\, u = f, \tag{1.6}$$

onde $f = \beta\widehat{u} + \gamma$. Essa é uma equação diferencial do tipo parabólica, pois $c > 0$ e Q é definida positiva.

1.1.3 Tensor de Condutividade Térmica

O tensor de condutividade térmica Q é uma quantidade material, i.e., depende das propriedades características do material. Dizemos que o material é *isotrópico* se seu comportamento não é alterado sob qualquer mudança de orientação do estado de referência do corpo. No caso da condução do calor, a isotropia é caracterizada pela condição de que o tensor de condutividade térmica é invariante sob qualquer rotação, ou em termos matemáticos,

$$RQR^T = Q \tag{1.7}$$

qualquer que seja a matriz ortogonal R. Uma matriz que satisfaz a condição (1.7) é denominada matriz isotrópica.

É fácil mostrar que uma matriz isotrópica pode ser escrita na forma simples, $Q = \alpha I$, ou em termos de componentes,

$$Q_{ij} = \alpha\,\delta_{ij},$$

onde I representa a matriz identidade e δ_{ij} é o *delta de Kronecker*, definido como

$$\delta_{ij} = \begin{cases} 1 & \text{para } i = j, \\ 0 & \text{para } i \neq j. \end{cases}$$

Para a demonstração, consideremos por simplicidade o caso bidimensional. Seja R a rotação do ângulo θ dada por

$$R = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta \\ \operatorname{sen}\theta & \cos\theta \end{bmatrix}.$$

Então, de (1.7) obtemos

$$\begin{bmatrix} \cos\theta & -\operatorname{sen}\theta \\ \operatorname{sen}\theta & \cos\theta \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} \cos\theta & \operatorname{sen}\theta \\ -\operatorname{sen}\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$$

para qualquer ângulo θ. Em particular, se tomarmos $\theta = \pi/2$, segue de imediato que

$$Q_{11} = Q_{22} = \alpha, \quad Q_{12} = -Q_{21}.$$

Se Q é tensor de condutividade térmica, o parâmetro α é denominado *coeficiente de condutividade térmica*. Nesse caso, temos da relação de simetria $Q_{ij} = Q_{ji} = 0$, e como a matriz é definida positiva, o coeficiente de condutividade térmica é uma quantidade positiva, $\alpha > 0$.

Um corpo é chamado *homogêneo* se as propriedades do material que o compõe são independentes da posição $\boldsymbol{x}$ do estado de referência.

Para um corpo homogêneo, ambos α e β são constantes que dependem do tipo de material e segue de (1.6) a conhecida equação do calor:

$$c\frac{\partial u}{\partial t} - \alpha\,\Delta u + \beta\,u = f,$$

onde c, α e β são constantes positivas e Δ é o operador de Laplace.

1.1.4 Condições de Fronteira

O problema estacionário para a equação do calor é governado pela seguinte equação diferencial parcial do tipo elíptico,

$$-\sum_{i,j} \frac{\partial}{\partial x_i}\Big(Q_{ij}\frac{\partial u}{\partial x_j}\Big) + \beta\,u = f, \quad \boldsymbol{x} \in V, \tag{1.8}$$

onde V é a região fixa ocupada pelo corpo, o tensor de condutividade térmica $Q_{ij}(\boldsymbol{x})$ é uma matriz definida positiva e $\beta(\boldsymbol{x}) > 0$. Se o corpo é isotrópico, então $Q_{ij}(\boldsymbol{x}) = \alpha(\boldsymbol{x})\,\delta_{ij}$ e a equação (1.8) toma a forma

$$-\sum_{i} \frac{\partial}{\partial x_i}\Big(\alpha\frac{\partial u}{\partial x_i}\Big) + \beta u = f. \tag{1.9}$$

Se além disso, o corpo é homogêneo, então a equação se reduz a

$$-\alpha\sum_{i} \frac{\partial^2 u}{\partial x_i^2} + \beta\,u = f, \tag{1.10}$$

onde α e β são constantes positivas.

É comum nas aplicações considerar prescritos na fronteira ∂V combinações da temperatura u com o fluxo de calor q. Dentre essas condições, podemos citar dois tipos especiais de condições para todo $\boldsymbol{x} \in \partial V$:

(1) $\quad u(\boldsymbol{x}) = u_0(\boldsymbol{x}),$

(2) $\quad -\boldsymbol{h}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) = u_1(\boldsymbol{x}),$

onde $u_0(\boldsymbol{x})$ e $u_1(\boldsymbol{x})$ são funções prescritas e o fluxo de calor q é dado por (1.2). Um problema com condições de fronteira do primeiro tipo é usualmente chamado de *problema de Dirichlet*, enquanto problemas com condições de fronteira do segundo tipo são chamados de *problemas de Neumann*.

Vale observar que, além desses casos especiais, há situações físicas onde seja necessário impor a condição de Dirichlet sobre uma dada parte da fronteira $\Gamma_1 \subset \partial V$ e uma condição de Newmann na parte complementar $\Gamma_2 = \partial V \setminus \Gamma_1$.

Usando a lei de Fourier (1.5), a condição de fronteira do tipo Neumann pode ser expressada em componentes, i.e.,

$$\sum_{i,j} Q_{ij} \frac{\partial u}{\partial x_j} n_i = u_1, \tag{1.11}$$

ou para um corpo homogêneo, em termos da derivada normal para a temperatura na fronteira,

$$\sum_{i} \frac{\partial u}{\partial x_i} n_i = u_1.$$

1.2 Elasticidade Linear

Consideremos agora um corpo deformável. Seja $B \subset \mathbb{R}^3$ a região ocupada pelo corpo no seu estado de referência e $\xi : B \times \mathbb{R} \to \mathbb{R}^3$ uma aplicação bijetora e regular,

$$\boldsymbol{x} = \xi(\boldsymbol{X}, t),$$

chamada de movimento do corpo. O movimento é uma deformação do estado de referência que depende do tempo. O gradiente da deformação F, a velocidade $\dot{\boldsymbol{x}}$ e a aceleração $\ddot{\boldsymbol{x}}$ são definidas como

$$F = \nabla_X \xi, \qquad \dot{\boldsymbol{x}} = \frac{\partial \xi}{\partial t}, \qquad \ddot{\boldsymbol{x}} = \frac{\partial^2 \xi}{\partial t^2},$$

respectivamente. Sejam (X_1, X_2, X_3) e (x_1, x_2, x_3) respectivamente as coordenadas cartesianas dos pontos da região ocupada pelo corpo no estado de referência e no estado atual. Assim, podemos expressar o gradiente da deformação F em termos de suas componentes, ou mais precisamente, por

$$F_{ij} = \frac{\partial x_i}{\partial X_j}.$$

O gradiente da deformação F é uma transformação linear de B em $B_t = \xi(B, t)$ e como estamos supondo F não singular, podemos assumir que $\det F(\boldsymbol{X}) > 0$ para todo $\boldsymbol{x}$. O gradiente da deformação F fornece uma aproximação linear da deformação ξ, que de modo geral é uma função não linear. Portanto, se considerarmos um pequeno segmento linear $d\boldsymbol{X}$ no estado de referência, temos no estado deformado

$$d\boldsymbol{x} = \xi(\boldsymbol{X} + d\boldsymbol{X}) - \xi(\boldsymbol{X}) = F(\boldsymbol{X})d\boldsymbol{X} + o(2), \tag{1.12}$$

onde $o(2)$ denota os termos de ordem superiores em $|d\boldsymbol{X}|$. Desta forma, o gradiente da deformação é considerado como uma medida de deformação local.

1.2.1 Pequena Deformação e Rotação Infinitesimal

Denotemos por

$$\boldsymbol{u}(\boldsymbol{X}, t) = \xi(\boldsymbol{X}, t) - \boldsymbol{X} \tag{1.13}$$

o vetor deslocamento do material no ponto $\boldsymbol{X} \in B$ e no tempo t. Assim, denotando por H o gradiente do vetor deslocamento, temos

$$F = I + H,$$

onde I é o tensor identidade, sendo $H = 0$ quando não há deformação. Assim, na teoria linear para pequenas deformações, assume-se que a norma do gradiente do deslocamento é uma pequena quantidade, que simbolicamente denotamos por $||H|| \ll 1$.

Introduzimos o tensor de deformação linear E e o tensor rotacional infinitesimal W como as partes simétrica e anti-simétrica do gradiente do deslocamento, *i.e.*,

$$E = \frac{1}{2}(H + H^T), \qquad W = \frac{1}{2}(H - H^T). \tag{1.14}$$

Para interpretar fisicamente estas definições, observamos inicalmente que

$$F^T F = (1 + H)^T(1 + H) = 1 + (H + H^T) + o(2) = 1 + 2E + o(2).$$

Sejam $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$ a base canônica do $\mathbb{R}^3$, $d\boldsymbol{X}_1$, $d\boldsymbol{X}_2$ dois pequenos segmentos lineares no estado de referência e $d\boldsymbol{x}_1$, $d\boldsymbol{x}_2$ os segmentos correspondentes no estado deformado. Por (1.12) e desprezando os termos de ordem superior de $||H||$, segue que

$$d\boldsymbol{x}_1 \cdot d\boldsymbol{x}_2 = F^T F d\boldsymbol{X}_1 \cdot d\boldsymbol{X}_2 = d\boldsymbol{X}_1 \cdot d\boldsymbol{X}_2 + 2E d\boldsymbol{X}_1 \cdot d\boldsymbol{X}_2. \tag{1.15}$$

Tomando $d\boldsymbol{X}_1 = d\boldsymbol{X}_2 = d\boldsymbol{X} = s_0\boldsymbol{e}_1$ e $|d\boldsymbol{x}| = s$, e sabendo que $E\boldsymbol{e}_i \cdot \boldsymbol{e}_j$ é a componente E_{ij}, obtemos

$$E_{11} = \frac{s^2 - s_0^2}{2s_0^2} = \frac{(s - s_0)(s + s_0)}{2s_0} \approx \frac{s - s_0}{s_0}.$$

Em outras palavras, E_{11} mede o alongamento relativo do segmento na direção $\boldsymbol{e}_1$ em relação ao comprimento original, também conhecido como deformação Lagrangeana. Observemos que as demais componentes da diagonal de E têm a mesma interpretação.

Para interpretar o significado das componentes fora da diagonal, consideremos $d\boldsymbol{X}_1 = s_0\boldsymbol{e}_1$ e $d\boldsymbol{X}_2 = s_0\boldsymbol{e}_2$, e seja θ o ângulo entre $\boldsymbol{d}x_1$ e $\boldsymbol{d}x_2$. Então de (1.15), temos

$$s^2 \cos\theta = 2s_0^2 E_{12},$$

que implica

$$E_{12} \approx \frac{1}{2}\sin\gamma \approx \frac{\gamma}{2},$$

onde $\gamma = 90^\circ - \theta$ é um ângulo pequeno para pequenas deformações. Portanto, a componente E_{12} mede a mudança de ângulo entre as direções dX_1 e dX_2.

Uma outra interpretação simples é sobre a variação do volume. Seja dV um pequeno volume do elemento formado por $(d\boldsymbol{X}_1, d\boldsymbol{X}_2, d\boldsymbol{X}_3)$. Então de (1.12), o volume dv do elemento no estado deformado é dado por

$$\begin{aligned} dv &= d\boldsymbol{x}_1 \cdot (d\boldsymbol{x}_2 \times d\boldsymbol{x}_3) = F d\boldsymbol{X}_1 \cdot (F d\boldsymbol{X}_2 \times F d\boldsymbol{X}_3) \\ &= (\det F) d\boldsymbol{X}_1 \cdot (d\boldsymbol{X}_2 \times d\boldsymbol{X}_3) = (\det F) dV. \end{aligned}$$

Lembrando que $F = I + H$, temos $\det F = 1 + \operatorname{tr} H + o(2)$, onde o traço de H é a soma das componentes da diagonal de H. Além disso, temos

$$\operatorname{tr} H = \frac{dv - dV}{dV}.$$

Por definição,

$$\operatorname{tr} H = \operatorname{div} \boldsymbol{u} = \operatorname{tr} E = E_{11} + E_{22} + E_{33},$$

e portanto, o traço de E, chamado de dilatação, mede a variação do volume com relação ao estado de referência.

Em termos do sistema de coordenadas, temos de (1.13) o vetor deslocamento,

$$u_i(X_j, t) = x_i(X_j, t) - X_i,$$

o vetor velocidade e o vetor aceleração,

$$\dot{x}_i = \frac{\partial u_i}{\partial t}, \qquad \ddot{x}_i = \frac{\partial^2 u_i}{\partial t^2}.$$

e o gradiente do deslocamento H,

$$H_{ij} = \frac{\partial u_i}{\partial X_j} = \sum_k \frac{\partial u_i}{\partial x_k}\frac{\partial x_k}{\partial X_j} = \sum_k \frac{\partial u_i}{\partial x_k}\Big(\delta_{kj} + \frac{\partial u_k}{\partial X_j}\Big) = \frac{\partial u_i}{\partial x_j} + o(2).$$

Na teoria linear, os termos de ordem superior são insignificantes, o que implica não ser necessário distinguir as coordenadas dos estados de referência e atual, isto é, não é necessário introduzir o estado de referência na teoria linear. Portanto, podemos escrever o gradiente do deslocamento como o gradiente com respeito ao estado atual, de modo que, de (1.14), o tensor de deformação e a rotação infinitesimal podem ser escritos como

$$E_{ij} = \frac{1}{2}\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big), \qquad W_{ij} = \frac{1}{2}\Big(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\Big). \tag{1.16}$$

1.2.2 Equação do Movimento

Seja $V \subset \mathbb{R}^3$ a região ocupada pelo corpo no estado atual. Segunda a Lei de Newton, podemos escrever a equação do movimento de uma parte arbitrária $\Omega \subset V$ do corpo na seguinte forma integral,

$$\int_\Omega \varrho \ddot{\boldsymbol{x}}\, dx = \int_{\partial\Omega} \boldsymbol{t}\, d\Gamma + \int_\Omega \boldsymbol{f}\, dx, \tag{1.17}$$

onde o lado direito representa as forças agindo sobre o corpo Ω. Existem dois tipos de forças; a primeira chamada de força do corpo $\boldsymbol{f}$ que é devido a forças externas, tal como a gravitação. A segunda, denominada tração de superfície $\boldsymbol{t}$, é a força agindo sobre a superfície do corpo $\partial\Omega$. A relação (1.17) é também conhecida como a conservação do momento linear. Observe que para um ponto no interior do corpo $\boldsymbol{x} \in \partial\Omega \subset V$, a tração $\boldsymbol{t}$ é a força agindo sobre a parte Ω pela parte restante do corpo $V\backslash\Omega$ através da superfície $\partial\Omega$ no ponto $\boldsymbol{x}$. Para as superfícies de duas partes diferentes com um ponto em comum $\boldsymbol{x}$, os valores da tração $\boldsymbol{t}$ no $\boldsymbol{x}$ são geralmente diferentes nas respectivas superfícies. A ideia clássica para simplificar a dependência da tração em $\partial\Omega$, conhecida como a *hipótese de Cauchy*, garante que se duas superfícies tem a mesma normal em $\boldsymbol{x}$ então os valores da tração são iguais em $\boldsymbol{x}$.

A consequência principal da hipótese de Cauchy, também conhecida como o *Teorema de Cauchy*, assegura que a tração $\boldsymbol{t}$ é função linear do vetor normal $\boldsymbol{n}$, *i.e.*,

$$\boldsymbol{t} = \sigma \boldsymbol{n} \qquad \text{or} \qquad t_i = \sum_{i,j} \sigma_{ij} n_j, \tag{1.18}$$

onde $\boldsymbol{n}$ é a normal unitária externa de $\partial\Omega$. O tensor σ é denominado *tensor de tensão*.

Usando o teorema da divergência, a equação integral (1.17), válida para qualquer $\Omega \subset V$, implica a seguinte equação de movimento,

$$\varrho \ddot{\boldsymbol{x}} - \operatorname{div} \sigma = \boldsymbol{f}, \tag{1.19}$$

ou na forma componente,

$$\varrho \frac{\partial^2 u_i}{\partial t^2} - \sum_j \frac{\partial \sigma_{ij}}{\partial x_j} = f_i.$$

Uma outra consequência importante referente ao tensor de tensão segue da conservação do momento angular, o qual assegura a simetria do tensor de tensão,

$$\sigma^T = \sigma \qquad \text{ou} \qquad \sigma_{ij} = \sigma_{ji}. \tag{1.20}$$

De fato, a conservação do momento angular é dada por

$$\int_\Omega \boldsymbol{r} \times \varrho \ddot{\boldsymbol{x}} \, dx = \int_{\partial\Omega} \boldsymbol{r} \times \sigma \boldsymbol{n} \, d\Gamma + \int_\Omega \boldsymbol{r} \times \boldsymbol{f} \, dx, \tag{1.21}$$

onde $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{x}_0$ é o vetor posição relativo a algum ponto de referência $\boldsymbol{x}_0 \in \mathbb{R}^3$. Em primeiro lugar, para provar a implicação de (1.20), calculamos o produto interno de (1.21) com um vetor constante $\boldsymbol{a}$ e aplicamos o teorema da divergência para obter

$$\boldsymbol{a} \cdot \boldsymbol{r} \times \varrho \ddot{\boldsymbol{x}} - \operatorname{div}(\sigma^T(\boldsymbol{a} \times \boldsymbol{r})) = \boldsymbol{a} \cdot \boldsymbol{r} \times \boldsymbol{f}, \tag{1.22}$$

onde usamos a identidade,

$$\boldsymbol{a} \cdot \boldsymbol{r} \times \sigma \boldsymbol{n} = \boldsymbol{a} \times \boldsymbol{r} \cdot \sigma \boldsymbol{n} = (\sigma^T(\boldsymbol{a} \times \boldsymbol{r})) \cdot \boldsymbol{n}.$$

Por outro lado, de (1.19) segue que

$$\boldsymbol{a} \cdot \boldsymbol{r} \times \varrho \ddot{\boldsymbol{x}} - \boldsymbol{a} \cdot \boldsymbol{r} \times \operatorname{div} \sigma = \boldsymbol{a} \cdot \boldsymbol{r} \times \boldsymbol{f}. \tag{1.23}$$

Comparando (1.22) e (1.23), obtemos a seguinte relação,

$$\operatorname{div}(\sigma^T(\boldsymbol{a} \times \boldsymbol{r})) = \boldsymbol{a} \cdot \boldsymbol{r} \times \operatorname{div} \sigma, \tag{1.24}$$

de onde se obtém, após alguns cálculos, a simetria do tensor de tensão, $\sigma^T = \sigma$. Esses cálculos estão feitos na Sec. 1.3.

Para uma interpretação física do tensor de tensão, consideremos o seguinte argumento: suponhamos que em um dado ponto da superfície $\partial\Omega$ tenhamos $\boldsymbol{n} = \boldsymbol{e}_1$. Então, a componente da tração $\boldsymbol{t}$ normal à superfície é dada por

$$\boldsymbol{t} \cdot \boldsymbol{e}_1 = \boldsymbol{e}_1 \cdot \sigma \boldsymbol{e}_1 = \sigma_{11},$$

e a componente tangencial na direção $\boldsymbol{e}_2$ é dada por

$$\boldsymbol{t} \cdot \boldsymbol{e}_2 = \boldsymbol{e}_2 \cdot \sigma \boldsymbol{e}_1 = \sigma_{21}.$$

Assim, σ_{11} e σ_{21} são respectivamente as forças normal e tangencial por unidade de área no tal ponto da superfície com normal na direção do eixo x_1. As demais componentes de σ_{ij} têm significados semelhantes.

1.2.3 Lei de Hooke

A equação constitutiva para materiais elásticos sob pequenas deformações pode ser expressada como uma relação linear entre a tensão e a deformação:

$$\sigma(\boldsymbol{x}, t) = C(\boldsymbol{x})E(\boldsymbol{x}, t) \qquad \text{ou} \qquad \sigma_{ij} = \sum_{k,l} C_{ijkl} E_{kl}, \tag{1.25}$$

onde C é o tensor de elasticidade de quarta ordem. Sendo simétricos os tensores de tensão e de deformação, o tensor elasticidade satisfaz a seguinte propriedade de simetria:

$$C_{ijkl} = C_{jikl} = C_{ijlk}. \tag{1.26}$$

Além disso, por argumentos da termodinâmica, prova-se que existe uma função de energia potencial $W(E)$ tal que

$$\sigma = \frac{\partial W}{\partial E}.$$

Como consequência, o tensor elasticidade é dado por

$$C_{ijkl} = \frac{\partial^2 W}{\partial E_{ij} \partial E_{kl}},$$

que implica em uma simetria adicional,

$$C_{ijkl} = C_{klij}. \tag{1.27}$$

A relação linear (1.25) é chamada de lei de Hooke para materiais elásticos. Na sua generalidade, o tensor de elasticidade envolve muitas constantes de material. De fato,

considerando-se as simetrias (1.26) e (1.27) e materiais não isotrópicos de um modo geral, existem 21 constantes. Este número é bastante reduzido se o material possui alguma simetria de orientação, tal como a simetria dos sólidos cristalinos. De maneira similar à apresentada na Sec. 1.1.3, para corpos isotrópicos, o tensor de elasticidade deve ser invariante sob qualquer orientação, e como consequência, pode-se provar que o número de constantes do material se reduz a somente duas, a saber,

$$C_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}). \tag{1.28}$$

As duas constantes do material λ e μ são chamadas de constantes de Lamé, e a lei de Hooke (1.25) pode então ser escrita na seguinte forma,

$$\sigma_{ij} = \lambda(\operatorname{tr} E)\delta_{ij} + 2\mu E_{ij}, \tag{1.29}$$

onde $\operatorname{tr} E = \operatorname{div} \boldsymbol{u} = E_{11} + E_{22} + E_{33}$.

Por considerações da termodinâmica, é usual admitir-se que o tensor de elasticidade é definido positivo, *i.e.*,

$$\sum_{i,j,k,l} C_{ijkl}S_{ij}S_{kl} > 0, \tag{1.30}$$

qualquer que seja a matriz não singular e simétrica S_{ij}. Em particular, para corpos isotrópicos, tem-se que

$$\mu > 0, \qquad 3\lambda + 2\mu > 0.$$

1.2.4 Problemas Elastostáticos

Para problemas estáticos da elasticidade linear, segue de (1.19) que a equação de equilíbrio para o vetor deslocamento $\boldsymbol{u}(\boldsymbol{x})$ é um sistema de equações diferenciais parciais do tipo elíptico em V,

$$-\sum_{j,\,k,\,l} \frac{\partial}{\partial x_j}\Big(C_{ijkl}\frac{\partial u_k}{\partial x_l}\Big) = f_i \qquad i = 1, 2, 3. \tag{1.31}$$

O tensor de elasticidade C_{ijkl} em (1.30) é um tensor definido positivo de quarta ordem, com as propriedades de simetria dados por (1.26) e (1.27). A força externa $\boldsymbol{f}$ é uma função dada. A propriedade de simetria $C_{ijkl} = C_{ijlk}$ foi usada para obter a equação (1.31). De fato, de (1.25) e (1.16) temos

$$\begin{aligned}
\sigma_{ij} &= \sum_{k,l} C_{ijkl}E_{kl} = \frac{1}{2}\sum_{k,l} C_{ijkl}\Big(\frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k}\Big) \\
&= \frac{1}{2}\sum_{k,l}\Big(C_{ijkl}\frac{\partial u_k}{\partial x_l} + C_{ijkl}\frac{\partial u_l}{\partial x_k}\Big) = \frac{1}{2}\sum_{k,l}\Big(C_{ijkl}\frac{\partial u_k}{\partial x_l} + C_{ijlk}\frac{\partial u_k}{\partial x_l}\Big) \\
&= \frac{1}{2}\sum_{k,l}\Big(C_{ijkl}\frac{\partial u_k}{\partial x_l} + C_{ijkl}\frac{\partial u_k}{\partial x_l}\Big) = \sum_{k,l} C_{ijkl}\frac{\partial u_k}{\partial x_l}.
\end{aligned}$$

Se o corpo é isotrópico, segue de (1.29) a equação de equilíbrio,

$$-\sum_k \frac{\partial}{\partial x_i}\Big(\lambda \frac{\partial u_k}{\partial x_k}\Big) - \sum_k \frac{\partial}{\partial x_k}\Big(\mu(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i})\Big) = f_i. \tag{1.32}$$

Além disso, se o corpo é homogêneo, então λ e μ são constantes e a equação se reduz a

$$-(\lambda+\mu)\,\mathrm{grad}(\mathrm{div}\,\boldsymbol{u}) - \mu\Delta\boldsymbol{u} = \boldsymbol{f}.$$

É usual admitir-se prescritos ou o vetor deslocamento $\boldsymbol{u}$, ou a tração $\boldsymbol{t}$ sobre a fronteira ∂V. Mais especificamente, temos dois tipos de condições de fronteira:

$$u_i(\boldsymbol{x}) = q_i(\boldsymbol{x}) \quad \text{ou} \quad \sum_j \sigma_{ij}(\boldsymbol{x}) n_j(\boldsymbol{x}) = p_i(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \partial V,$$

onde $q_i(\boldsymbol{x})$ e $p_i(\boldsymbol{x})$, $i = 1, 2, 3$, são funções prescritas. $i = 1, 2, 3$.

A primeira condição é chamada de *Condição do tipo Dirichlet*, enquanto que a segunda é denominada *Condição do tipo Neumann*. A condição de fronteira do tipo Neumann pode ser escrita na forma

$$\sum_{j,k,l} C_{ijkl} \frac{\partial u_k}{\partial x_l} n_j = p_i. \tag{1.33}$$

1.3 Convenção de Somatório

A convenção de somatório é frequentemente utilizada para simplificar as expressões envolvendo somas com respeito a índices repetidos:

Convenção de somatório. *Na expressão de uma soma, quando aparece um par de índices, i.e., o índice é repetido exatamente uma vez, convenciona-se que os termos referentes a esse índice estão sendo somados, sendo então desnecessário explicitar o símbolo de somatório.*

O domínio do índice fica subentendido no contexto. No uso desta convenção, em nenhum lugar da expressão o índice pode ser repetido mais de uma vez, caso contrário há possibilidades de erro. No caso do somatório em que um índice aparece somente um única vez ou mais de duas vezes, o símbolo de somatório tem que ser explicitamente indicado para não haver confusão.

Com esta convenção de somatório, podemos representar os dois problemas principais deste capítulo como segue:

1. *Problema estacionário da condução do calor*, (1.8) e (1.11):

$$\begin{cases} -\dfrac{\partial}{\partial x_i}\Big(Q_{ij}\dfrac{\partial u}{\partial x_j}\Big) + \beta\, u = f & \text{em} \quad V, \\ u = u_0 \quad \text{ou} \quad Q_{ij}\dfrac{\partial u}{\partial x_j}\, n_i = u_1 & \text{sobre} \quad \partial V. \end{cases}$$

2. *Problemas elastostáticos*, (1.31) e (1.33):

$$\begin{cases} -\dfrac{\partial}{\partial x_j}\Big(C_{ijkl}\dfrac{\partial u_k}{\partial x_l}\Big) = f_i & \text{em} \quad V, \\ u_i = q_i \quad \text{ou} \quad C_{ijkl}\dfrac{\partial u_k}{\partial x_l} n_j = p_i & \text{sobre} \quad \partial V. \end{cases}$$

Para dar mais um exemplo do uso da convenção de somatório, provaremos a simetria do tensor de tensão da relação (1.24). Primeiro, introduziremos o símbolo de permutação:

$$\varepsilon_{ijk} = \begin{cases} 1 & \text{se } (i,j,k) \text{ é uma permutação par de (1,2,3),} \\ -1 & \text{se } (i,j,k) \text{ é uma permutação ímpar de (1,2,3),} \\ 0 & \text{para os demais casos.} \end{cases}$$

Com este símbolo, o produto vetorial $\boldsymbol{v} \times \boldsymbol{u}$ pode ser escrito na forma componente como

$$(\boldsymbol{v} \times \boldsymbol{u})_i = \varepsilon_{ijk} v_j u_k.$$

Agora podemos escrever a relação (1.24) como

$$\frac{\partial}{\partial x_i}(\sigma_{ji}(\varepsilon_{jkl} a_k r_l)) = a_j \varepsilon_{jkl} r_k \frac{\partial \sigma_{li}}{\partial x_i}. \tag{1.34}$$

Relembramos que a_i é um vetor constante e r_i é o vetor posição de x_i, e portanto

$$\frac{\partial r_l}{\partial x_i} = \delta_{il}.$$

Fazendo a derivada do lado esquerdo de (1.34), obtemos

$$\frac{\partial}{\partial x_i}(\sigma_{ji}(\varepsilon_{jkl} a_k r_l)) = \frac{\partial \sigma_{ji}}{\partial x_i}\varepsilon_{jkl} a_k r_l + \sigma_{ji}\varepsilon_{jkl} a_k \delta_{il}. \tag{1.35}$$

Fazendo a mudança de índices no somatório, o primeiro termo do lado direito pode ser reescrito como

$$\frac{\partial \sigma_{ji}}{\partial x_i}\varepsilon_{jkl} a_k r_l = \frac{\partial \sigma_{li}}{\partial x_i}\varepsilon_{ljk} a_j r_k,$$

que é igual ao termo do lado direito de (1.34), porque $\varepsilon_{ljk} = \varepsilon_{jkl}$. Portanto, o segundo termo do lado direito de (1.35) é zero,

$$\sigma_{ji}\varepsilon_{jkl}a_k\delta_{il} = \sigma_{ji}\varepsilon_{jki}a_k = 0.$$

Como a_k é arbitrário, temos

$$\sigma_{ji}\varepsilon_{jki} = 0.$$

Este é um sistema de três equações para $k = 1, 2, 3$, dado por

$$\sigma_{32} - \sigma_{23} = 0, \qquad \sigma_{13} - \sigma_{31} = 0, \qquad \sigma_{21} - \sigma_{12} = 0.$$

Em outras palavras, o tensor de tensão é simétrico.

1.4 Métodos Numéricos

Considere o problema modelo de determinar uma função $u = u(x)$ que satisfaça a seguinte equação diferencial com as condições de contorno abaixo:

$$\begin{cases} u''(x) = f(x, u(x), u'(x)) & \forall\, x \in (0,1), \\ u(0) = u(1) = 0, \end{cases}$$

onde estamos denotando por u' a derivada de u. Suponhamos que f seja uma função regular e que o problema admita uma única solução. A solução aproximada do problema pode ser obtida por duas classes de métodos numéricos: o método das diferenças finitas e o método das projeções. A ideia básica do método das diferenças finitas é transformar o problema de resolver uma equação diferencial num problema de resolver um sistema de equações algébricas, substituindo as derivadas por diferenças finitas. Por outro lado, o método das projeções consiste em obter uma aproximação da solução da equação diferencial, usando uma combinação linear finita de funções conhecidas, usualmente denominadas *funções bases*. Conceitualmente, se considerarmos que a solução do problema pertence a algum espaço de funções de dimensão infinita, então a solução aproximada é obtida num subespaço de dimensão finita, gerado pelas funções bases. A projeção da solução sobre o subespaço de dimensão finita é a solução aproximada. O método de elementos finitos é baseado no método das projeções.

Para exemplificar as ideias gerais do método das projeções com um exemplo simples, consideremos o seguinte problema de segunda ordem:

$$\begin{cases} -u''(x) + u(x) = f(x) & \forall\, x \in (0,1), \\ u(0) = u(1) = 0. \end{cases} \tag{1.36}$$

onde $f = f(x)$ é uma função regular. Suponhamos que a solução aproximada do problema (1.36) seja dada por

$$u_m(x) = \sum_{j=1}^{m} C_j \varphi_j(x), \tag{1.37}$$

onde as funções bases φ_j satisfazem as condições de fronteira:

$$\varphi_j(0) = \varphi_j(1) = 0, \qquad j = 1, \cdots m. \tag{1.38}$$

Nestas condições, a solução aproximada $u_m(x)$, dada por (1.37), satisfaz a condição de fronteira e o problema se reduz a se determinar os coeficientes C_j em (1.37). Existem várias possíveis aproximações e aqui daremos somente as duas mais conhecidas: método de colocação e método de Galerkin.

1.4.1 Método da Colocação

Sejam $x_1, \cdots, x_m$ os pontos da malha da discretização do intervalo $[0, 1]$, denominados nós ou pontos nodais. Queremos que a solução aproximada satisfaça a equação diferencial nesses m pontos. Portanto, substituindo $u_m(x)$ definida em (1.37) na equação diferencial, obtemos

$$-\sum_{j=1}^{m} C_j \varphi_j''(x_i) + \sum_{j=1}^{m} C_j \varphi_j(x_i) = f(x_i), \qquad i = 1, \cdots, m,$$

onde estamos admitindo que as funções bases são de classe C^2, *i.e.*, duas vezes continuamente diferenciáveis. Podemos isolar os coeficientes C_j para obter o seguinte sistema linear:

$$\sum_{j=1}^{m} C_j \left(-\varphi_j''(x_i) + \varphi_j(x_i)\right) = f(x_i), \qquad i = 1, \cdots, m. \tag{1.39}$$

Assim, denotando

$$a_{ij} = -\varphi_j''(x_i) + \varphi_j(x_i), \tag{1.40}$$

obtemos o sistema linear $AC = F$, onde $A = (a_{ij})$ é uma matriz quadrada de ordem m, $C = (C_1, \cdots, C_m)^T$ é o vetor incógnita do sistema e $F = (f(x_1), \cdots, f(x_m))^T$ é um vetor conhecido, denominado *vetor força nodal*. Uma vez determinandos os coeficientes C_j do sistema linear, a solução aproximada da equação diferencial (1.36) é obtida através de (1.37). A matriz A depende somente das funções bases consideradas e, de um modo geral, é uma matriz cheia e não simétrica. O uso de funções bases com suporte pequeno localizado nos pontos nodais pode simplificar a forma da matriz A e facilitar a resolução do sistema linear.

1.4.2 Método de Galerkin

O método de Galerkin é baseado no conceito de ortogonalidade de funções. Dadas duas funções integráveis u e v definidas em [0,1], dizemos que são ortogonais se

$$(u : v) := \int_0^1 u(x)v(x)\,dx = 0. \tag{1.41}$$

Voltando ao problema modelo (1.36), consideremos a função resíduo $r(x)$ definida por

$$r(x) = -u''(x) + u(x) - f(x) \qquad \forall\, x \in (0,1). \tag{1.42}$$

É fácil ver que se $u(x)$ é a solução exata do problema (1.36), então $r(x) = 0$ e a função resíduo é ortogonal a qualquer função. No entanto, para a solução aproximada $u(x) = u_m(x)$ definida em (1.37), não podemos esperar que $r(x)$ seja identicamente nula, pois $u_m(x)$ é apenas uma combinação linear das funções bases. O método de Galerkin consiste em determinar $u_m(x)$ de tal forma a preservar a propriedade de ortogonalidade da solução exata, isto é, que a função resíduo seja ortogonal a todas as funções bases $\{\varphi_1, \varphi_2, \cdots, \varphi_m\}$. Mais precisamente,

$$\int_0^1 [-u_m''(x) + u_m(x) - f(x)]\varphi_i(x)dx = 0, \qquad i = 1, \cdots, m.$$

Substituindo (1.37) na integral acima, obtemos

$$\sum_{j=1}^{m} C_j \int_0^1 [-\varphi_j''(x) + \varphi_j(x)]\varphi_i(x)dx = \int_0^1 f(x)\varphi_i(x)dx, \qquad i = 1, \cdots, m. \tag{1.43}$$

Integrando por partes o primeiro termo, obtemos

$$-\int_0^1 \varphi_j''(x)\varphi_i(x)dx = -\varphi_j'(x)\varphi_i(x)\Big|_0^1 + \int_0^1 \varphi_j'(x)\varphi_i'(x)dx = \int_0^1 \varphi_j'(x)\varphi_i'(x)dx,$$

pois $\varphi_i(0) = \varphi_i(1) = 0$. Definindo

$$\begin{aligned} a_{ij} &= \int_0^1 \varphi_j'(x)\varphi_i'(x)dx + \int_0^1 \varphi_j(x)\varphi_i(x)dx, \\ f_i &= \int_0^1 f(x)\varphi_i(x)dx, \end{aligned} \tag{1.44}$$

obtemos de (1.43) um sistema linear $AC = F$ com incógnitas $C = (C_1, \cdots, C_m)^T$, onde $F = (f, \cdots, f_m)^T$ e $A = (a_{ij})$ é uma matriz $m \times m$.

É imediato de (1.44) que é suficiente supor $\varphi_j(x)$ e suas derivadas $\varphi_j'(x)$ funções quadrado-integráveis, *i.e.*, pertencentes ao espaço $L^2(0,1)$, para que os coeficientes a_{ij} estejam bem definidos. Portanto, as classes de funções disponíveis para funções bases no método de Galerking são mais amplas do que aquelas possíveis para o método da colocação. De fato, funções contínuas e seccionalmente lineares, por exemplo, satisfazem as condições para formar bases no método de Galerking, mas não são necessariamente duas vezes diferenciáveis.

Além disso, note que a matriz A definida por (1.44) é simétrica, o que facilita a resolução do sistema linear $AC = F$, e portanto, apresenta outra vantagem do método de Galerkin sobre o método da colocação.

A matriz A depende somente da escolha de funções bases. Uma escolha clássica que resulta em uma matriz diagonal (o caso ideal), é o da escolha das autofunções do operador diferencial do problema (1.36), *i.e.*, $D = -d^2/dx^2 + I$,

$$\varphi_j(x) = \operatorname{sen}(j\pi x), \quad j = 1, \cdots, m. \tag{1.45}$$

Neste caso, como as funções bases são mutuamente ortogonais e satisfazem a condição

$$\int_0^1 \operatorname{sen}(i\pi x)\operatorname{sen}(j\pi x)\,dx = \int_0^1 \cos(i\pi x)\cos(j\pi x)\,dx = \begin{cases} \frac{1}{2} & \text{se} \quad i = j, \\ 0 & \text{se} \quad i \neq j, \end{cases}$$

Substituindo em (1.44), obtemos um sistema linear $AC = F$, onde a matriz A é diagonal, ou seja

$$\begin{aligned} a_{jj} &= \frac{1}{2}\Big(1 + (j\pi)^2\Big), \\ f_j &= \int_0^1 f(x)\operatorname{sen}(j\pi x)\,dx, \end{aligned} \qquad j = 1, \ldots, m. \tag{1.46}$$

Portanto, os valores de C_j são dados por

$$C_j = \frac{2f_j}{1 + (j\pi)^2}$$

e a solução aproximada do problema (1.36), usando a base de autofunções, é dada por

$$u_m(x) = \sum_{j=1}^{m} \frac{2f_j}{1 + (j\pi)^2}\operatorname{sen}(j\pi x).$$

Observemos que esta solução corresponde à série truncada da solução clássica obtida pelo *método de séries de Fourier*,

$$u(x) = \sum_{j=1}^{\infty} \frac{\widehat{f}_j}{1 + (j\pi)^2}\operatorname{sen}(j\pi x),$$

onde $\widehat{f}_j$, $j \in \mathbb{N}$, são os coeficientes de Fourier da função $f(x)$,

$$\widehat{f}_j = 2\int_0^1 f(x)\,\mathrm{sen}(j\pi x)\,dx.$$

De (1.46) temos $\widehat{f}_j = 2f_j$.

Evidentemente, as autofunções definem uma base ideal para o cálculo da solução, mas infelizmente, em situação mais gerais, as autofunções do operador diferencial associado ao problema não são explicitamente definidas. Porém, como veremos adiante, podemos escolher uma base "local" tal que a matriz dos coeficientes seja uma matriz banda, o que reduz bastante o número de operações para a resolução do sistema linear.

1.4.3 Elementos Finitos

Em cada um dos métodos numéricos de projeção, o problema computacional central é resolver o sistema algébrico (linear ou não linear). Assim, é desejável que a matriz dos coeficientes tenha algumas propriedades que permitam facilidade de resolução (menor número de operações) e seja bem condicionada. A matriz dos coeficientes A depende fundamentalmente das funções bases $\{\varphi_1(x), \cdots, \varphi_m(x)\}$ que geram o subespaço onde estamos procurando a solução aproximada $u_m(x)$.

A ideia fundamental do método de elementos finitos é introduzir funções bases com suporte pequeno, localizado nos pontos nodais dos elementos. Para exemplificar, consideremos o problema unidimensional (1.36) e $\{x_0 = 0 < x_1, \cdots, x_m < x_{m+1} = 1\}$ uma discretização uniforme do intervalo $[0, 1]$. Cada elemento $[x_{j+1} - x_j]$ é um intervalo de comprimento $h = 1/(m+1)$. Seja $\{\varphi_1(x), \cdots, \varphi_m(x)\}$ uma base definida por

$$\varphi_j(x) = \begin{cases} \dfrac{1}{h}(x - x_{j-1}) & \forall\, x \in [x_{j-1}, x_j], \\ \dfrac{1}{h}(x_{j+1} - x) & \forall\, x \in [x_j, x_{j+1}], \\ 0 & \forall\, x \notin [x_{j-1}, x_{j+1}], \end{cases} \qquad j = 1, \ldots, m. \tag{1.47}$$

Neste caso, o suporte da função $\varphi_j(x)$ é o intervalo $[x_{j-1}, x_{j+1}]$ que contém o ponto nodal x_j. Logo, a matriz A obtida pelo método de Galerkin é uma matriz tridiagonal $m \times m$, sendo o número de operações para a resolução do sistema proporcional a m.

Note que a função base $\varphi_j(x)$ definida em (1.47) é contínua e seccionalmente linear, sendo portanto uma função de $L^2(0, 1)$. Mas ela não pode ser usada como função base para o problema (1.36) utilizando o método de colocação, pois ela não é de classe C^2. Entretanto, funções mais regulares constituídas de polinômios (funções spline) são muito usadas como funções bases nos métodos de colocação e de Galerkin.

Em geral, nenhum dos dois métodos discutidos aqui tem uma grande vantagem sobre o outro e isto depende de cada problema em particular. No que se refere aos problemas elíticos, consideraremos exclusivamente elementos finitos, utilizando o Método de Galerkin.

1.5 Problemas Variacionais Abstratos

Para uma melhor compreensão dos conceitos usados nos próximos capítulos, introduziremos algumas definições e teoremas fundamentais para a prova da existência e unicidade de solução para uma ampla classe de problemas variacionais. Para isso, denotemos V e H dois espaços de Hilbert com produto interno e norma representados, respectivamente, por $((\cdot:\cdot))$, $\|\cdot\|$ e $(\cdot:\cdot)$, $|\cdot|$.

Definição 1. *Dizemos que uma função $a(\cdot,\cdot) : V \times V \to \mathbb{R}$ é uma* forma bilinear em *V se ela é linear em cada uma das suas componentes, isto é, para cada $u \in V$, as aplicações $v \mapsto a(u,v)$ e $v \mapsto a(v,u)$ são lineares em V.*

Definição 2. *A forma bilinear $a(\cdot,\cdot)$ é contínua em V se existe uma constante $C_1 > 0$ tal que*

$$|a(u,v)| \leq C_1\|u\|\|v\|, \qquad \forall u, v \in V.$$

Definição 3. *A forma bilinear $a(\cdot,\cdot)$ é coerciva em V, se existe uma constante $C_2 > 0$ tal que*

$$a(v,v) \geq C_2\|v\|^2, \quad \forall v \in V.$$

Definição 4. *A forma bilinear $a(\cdot,\cdot)$ é simétrica em V se*

$$a(u,v) = a(v,u) \qquad \forall u, v \in V.$$

Definição 5. *Uma função $f : V \to \mathbb{R}$ linear é contínua se existe uma constante $C_3 > 0$ tal que*

$$|\langle f, v\rangle| \leq C_3\|v\|, \qquad \forall v \in V.$$

Nesta caso, diz-se que f é um elemento do dual de V, representado por V' ou V^*.

1.5.1 Formulação Variacional Abstrata

Queremos determinar uma função $u \in V$ tal que

$$a(u,v) = \langle f, v\rangle, \qquad \forall v \in V. \tag{1.48}$$

O resultado do teorema que segue se aplica aos problemas considerados nos próximos capítulos.

Teorema (Lax-Milgram). *Seja $a(\cdot,\cdot)$ uma forma bilinear, contínua e coerciva. Então, se f é uma forma linear e contínua em V, o problema variacional abstrato (1.48) possui uma única solução $u \in V$. Além disso, a aplicação $f \mapsto u$ é contínua de V' em V.*

A demonstração do teorema de Lax-Milgram pode ser encontrado em ([2, 13, 14]).

Uma interessante forma de se caracterizar a solução do problema variacional abstrato pode ser feita através da minimização do funcional $E : V \to \mathbb{R}$, denominado funcional energia, dado por

$$E(v) = \frac{1}{2}\, a(v,v) - \langle f, v\rangle, \quad \forall v \in V. \tag{1.49}$$

Para que os problemas (1.48) e (1.49) sejam equivalentes é necessário que a forma bilinear $a(\cdot,\cdot)$ seja simétrica, como podemos ver na seguinte proposição:

Proposição. *Seja $a(\cdot,\cdot) : V \times V \to \mathbb{R}$ uma forma bilinear, contínua, coerciva, simétrica e $f : V \to \mathbb{R}$ uma forma linear e contínua em V. Então, $u \in V$ é a solução do problema (1.48) se, e somente se, u minimiza o funcional E em (1.49).*

Demonstração: Seja u a solução do problema (1.48). Então,

$$a(u, u-v) = \langle f, u-v\rangle, \quad \forall v \in V.$$

Sob as hipóteses, temos

$$a(u,u-v) \le \langle f, u-v\rangle + \tfrac{1}{2}\, a(u-v,u-v) \Longleftrightarrow$$

$$a(u,u) - a(u,v) \le \langle f, u\rangle - \langle f, v\rangle + \tfrac{1}{2}\, a(u,u) - a(u,v) + \tfrac{1}{2}\, a(v,v), \quad \forall v \in V,$$

de onde se conclui que $E(u) \le E(v)$ qualquer que seja $v \in V$ e portanto a solução u minimiza o funcional energia. Para mostrar que esta solução é única, considere uma outra solução $\tilde{u} \in V$ de energia mínima. Então, por definição de mínimo, temos

$$E(u) \le E\left(\frac{u+\tilde{u}}{2}\right) \quad \text{e} \quad E(\tilde{u}) \le E\left(\frac{u+\tilde{u}}{2}\right).$$

Da definição de E, obtemos desenvolvendo cada um dos termos,

$$\frac{1}{2}\, a(u,u) + \frac{1}{2}\, a(\tilde{u},\tilde{u}) \le \frac{1}{4}\{a(u,u) + a(\tilde{u},\tilde{u}) + a(u,\tilde{u}) + a(\tilde{u},u)\}$$

Logo $\frac{1}{4}\, a(u-\tilde{u}, u-\tilde{u}) \le 0$. Sendo $a(\cdot,\cdot)$ coerciva, conclui-se que $u = \tilde{u}$.

Reciprocamente, seja u mínimo do funcional (1.49). Então, vale a seguinte desigualdade:

$$\frac{E(u+\lambda v)-E(u)}{\lambda} \geq 0, \quad \forall v \in V, \quad \forall \lambda \in \mathbb{R},$$

de onde se obtém

$$\lim_{\lambda \to 0^+} \frac{E(u+\lambda v)-E(u)}{\lambda} = a(u,v) - \langle f, v \rangle$$

e consequentemente, $a(u,v) - \langle f, v \rangle \geq 0$ para todo $v \in V$. Como a desigualdade é válida para todo v podemos substituir v por $-v$ para obter a igualdade

$$a(u,v) - \langle f, v \rangle = 0, \quad \forall v \in V.$$

Portanto u é solução do problema (1.48). □

1.5.2 Espaços Funcionais

Para aplicar os resultados abstratos aos problemas específicos que trateremos no texto, faz-se necessária a introdução de noções básicas da teoria das distribuições e dos espaços de Sobolev.

• **Espaços das Funções Testes:**

Seja Ω um conjunto aberto e limitado do $\mathbb{R}^n$ com fronteira suave. Denotamos por $C^\infty(\Omega)$ o espaço vetorial das funções reais definidas em Ω infinitamente continuamente diferenciáveis. Dada uma função u definida em Ω, denomina-se suporte de u ao fecho em Ω do conjunto dos pontos de Ω onde a função u é diferente de zero. Por $C_0^\infty(\Omega)$ estamos denotando o subespaço do espaço $C^\infty(\Omega)$ com suporte compacto contido em Ω.

• **Convergência em** $C_0^\infty(\Omega)$

Definição 6. *Diz que uma sequência $(\varphi_\nu)_{\nu \in \mathbb{N}}$ pertencente ao espaço $C_0^\infty(\Omega)$ converge para zero quando forem satisfeitas as seguintes condições:*

1. *Todas as funções φ_ν da sequência possuem suportes contidos em um mesmo compacto $K \subset \Omega$.*

2. *Os elementos da sequência $(\varphi_\nu)_{\nu \in \mathbb{N}}$ e as suas derivadas de todas as ordens convergem uniformemente para zero em K, ou seja, $D^j \varphi_\nu \to 0$ uniformemente em K, qualquer que seja o multi-índice $j = (j_1, j_2, \cdots, j_n)$, $j_k \in \mathbb{N}$.*

Lembremos que, por definição,

$$D^j = \frac{\partial^{j_1+\cdots+j_n}}{\partial x_1^{j_1} \cdots \partial x_n^{j_n}}$$

O espaço vetorial $C_0^\infty(\Omega)$ munido da noção de convergência acima é denominado *espaço das funções testes*, representado por $\mathcal{D}(\Omega)$. Com essa noção de convergência, podemos introduzir o que se entende por uma distribuição.

• Distribuições

Definição 7. *Denomina-se distribuição sobre Ω toda forma linear T contínua em $\mathcal{D}(\Omega)$. Mais precisamente, uma distribuição T é um funcional $T : \mathcal{D}(\Omega) \to \mathbb{R}$ satisfazendo as seguintes condições:*

1. $T(\alpha\psi + \beta\varphi) = \alpha T(\psi) + \beta T(\varphi) \qquad \alpha, \beta \in \mathbb{R}; \qquad \psi, \varphi \in \mathcal{D}(\Omega).$
2. *T é contínua em $\mathcal{D}(\Omega)$, ou seja, se (φ_ν) converge para zero em $\mathcal{D}(\Omega)$, então $T(\varphi_\nu)$ converge para zero em $\mathbb{R}$*

O espaço das distribuições com a noção de convergência é denotado por $\mathcal{D}'(\Omega)$.

É fácil mostrar que uma função $u \in L^1_{loc}(\Omega)$, dita localmente integrável, define univocamente uma distribuição T_u pela seguinte relação:

$$T_u(\varphi) = \int_\Omega u(x)\varphi(x)dx.$$

Assim, podemos identificar de forma única a distribuição T_u com a função $u \in L^1_{loc}(\Omega)$, de modo que, com esta identificação, tem-se $L^1_{loc}(\Omega) \subset \mathcal{D}'(\Omega)$.

Por razões que ficam evidentes na Análise Funcional, é usual escrever $T(\varphi) = \langle T, \varphi\rangle$ quando $T \in \mathcal{D}'(\Omega)$, e quando não houver ambiguidade,

$$\langle u, \varphi\rangle = \int_\Omega u(x)\varphi(x)dx \qquad \forall\varphi \in \mathcal{D}(\Omega). \tag{1.50}$$

Para $T \in \mathcal{D}'(\Omega)$, $\alpha = (\alpha_1, \cdots, \alpha_n) \in \mathbb{N}^n$ e $|\alpha| = \alpha_1 + \cdots + \alpha_n$, definimos a derivada de ordem α de T no sentido das distribuições do seguinte modo:

$$\left\langle \frac{\partial^\alpha T}{\partial x^\alpha}, \varphi \right\rangle = (-1)^\alpha \left\langle T, \frac{\partial^\alpha \varphi}{\partial x^\alpha} \right\rangle, \qquad \forall\varphi \in \mathcal{D}(\Omega). \tag{1.51}$$

Observe que pela definição (1.51) toda distribuição é infinitamente derivável, visto que $\varphi \in C_0^\infty(\Omega)$.

• Espaços de Sobolev

Com a noção de distribuição e suas derivadas, podemos definir os espaços de Sobolev. Para isso, lembremos que se Ω é um aberto do $\mathbb{R}^n$, o espaço $L^2(\Omega)$ definido por

$$L^2(\Omega) = \left\{ u : \Omega \to \mathbb{R};\ \int_\Omega |u(x)|^2\, dx < \infty \right\}$$

é um espaço de Hilbert quando munido do produto interno e norma definidos respectivamente por

$$(u:v) = \int_\Omega u(x)v(x)\,dx \quad \text{e} \quad \|u\|_0^2 = \int_\Omega |u(x)|^2\,dx.$$

Vimos anteriormente que toda função $u \in L^1_{loc}(\Omega)$ define univocamente uma distribuição em Ω. Assim, o resultado também é válido para funções $u \in L^2(\Omega)$, já que $L^2(\Omega) \subset L^1_{loc}(\Omega)$. Assim, as função de $L^2(\Omega)$ podem ser consideradas distribuições sobre Ω e, nesse sentido, possuem derivadas de todas as ordens.

É importante observar que as derivadas no sentido das distribuições das funções $u \in L^2(\Omega)$ não são necessariamente funções $L^1_{loc}(\Omega)$.

Com as observações acima, podemos definir os espaços de Sobolev. $H^m(\Omega)$, $m \in \mathbb{N}$.

$$H^m(\Omega) := \Big\{u : \Omega \to \mathbb{R};\ u \in L^2(\Omega),\ \frac{\partial^\alpha u}{\partial x_1^{\alpha_1}\cdots\partial x_n^{\alpha_n}} \in L^2(\Omega)\Big\},$$

onde $\alpha = \alpha_1 + \cdots + \alpha_n$, $0 \le \alpha \le m$, $\alpha_i \in \mathbb{N}$.

O produto interno em $H^m(\Omega)$ é dado por

$$(u:v)_{H^m} = \sum_{\alpha=0}^{m} \int_\Omega \frac{\partial^\alpha u}{\partial x_1^{\alpha_1}\cdots\partial x_n^{\alpha_n}} \frac{\partial^\alpha v}{\partial x_1^{\alpha_1}\cdots\partial x_n^{\alpha_n}}\,dx \qquad \forall u, v \in H^m(\Omega). \tag{1.52}$$

Prova-se que o espaço de Sobolev $H^m(\Omega)$ munido do produto interno definido em (1.52) é, para todo $m \in$ N, um espaço de Hilbert. Por convenção, estabelemos que $H^0(\Omega) = L^2(\Omega)$ e para $m = 1$, temos

$$H^1(\Omega) = \Big\{u \in L^2(\Omega);\ \frac{\partial u}{\partial x_i} \in L^2(\Omega), i = 1, \ldots, n\Big\}$$

com o produto interno e a norma definidos por

$$(u:v)_1 := \int_\Omega uv\,dx + \int_\Omega \nabla u \cdot \nabla v\,dx \quad \text{e} \quad \|u\|_1^2 := \int_\Omega |u|^2\,dx + \int_\Omega |\nabla u|^2\,dx.$$

Um subespaço importante do espaço $H^1(\Omega)$ é o espaço $H^1_0(\Omega)$, definido pelas funções $u \in H^1(\Omega)$ que se anulam na fronteira Γ de Ω, ou seja,

$$H^1_0(\Omega) = \Big\{u \in H^1(\Omega);\ u = 0 \text{ sobre } \Gamma\Big\}.$$

Algumas propriedades importantes do espaço $H^1_0(\Omega)$ usadas no texto são dadas abaixo, cuja demonstração pode ser encontrada em ([2, 13, 14]).

1. **Dual**. Sendo $H_0^1(\Omega)$ um espaço de Hilbert, denota-se por $H^{-1}(\Omega)$ o seu espaço dual, isto é, o espaço dos operadores lineares e contínuos definidos em $H_0^1(\Omega)$. Desse modo, caracteriza-se $H^{-1}(\Omega)$ como sendo o espaço das distribuições sobre Ω da forma:
$$T = v_0 + \frac{\partial v_1}{\partial x_1} + \cdots + \frac{\partial v_n}{\partial x_n},$$
onde $v_0, v_1, \cdots, v_n$ são funções do espaço $L^2(\Omega)$

2. **Densidade**. O espaço das funções testes $\mathcal{D}(\Omega)$ é denso em $H_0^1(\Omega)$ na norma do $H^1(\Omega)$, ou seja, para qualquer função $u \in H_0^1(\Omega)$, existe uma sucessão de funções $(u_\nu)_{\nu\in\mathbb{N}}$ em $\mathcal{D}(\Omega)$ convergente para u em $H^1(\Omega)$.

   Obs: Identificando o espaço $L^2(\Omega)$ com seu dual $(L^2(\Omega))'$, podemos estabelecer as seguintes inclusões:
$$\mathcal{D}(\Omega) \subset H_0^1(\Omega) \subset L^2(\Omega) \approx (L^2(\Omega))' \subset H^{-1}(\Omega) \subset \mathcal{D}'(\Omega)$$

3. **Equivalência de normas**. No espaço $H_0^1(\Omega)$, as normas $\|u\|_1$ e $\|\nabla u\|_0$ são equivalentes, ou seja, existem constantes positivas C_1 e C_2 tais que
$$C_1\|u\|_1 \leq \|\nabla u\|_0 \leq C_2\|u\|_1,$$
onde estamos denotando por $\|\cdot\|_1$ e $\|\cdot\|_0$ as normas do $H^1(\Omega)$ e $L^2(\Omega)$, respectivaqmente. A primeira desigualdade é conhecida como desigualdade de Poincaré-Friedricks e a segunda é uma consequência imediata da definição de norma.

4. **Densidade em** $H^m(\Omega)$. O espaço das funções contínuas de ordem k, $C^k(\overline{\Omega})$, é densos em $H^m(\Omega)$ $m = 0, 1\cdots$; onde $k \geq m$. Esta importante propriedade implica que qualquer função do $H^m(\Omega)$ e suas derivadas pode ser aproximada por funções do espaço das funções contínuas $C^k(\overline{\Omega})$. Neste caso, as derivadas no sentido das distribuições coincidem com aquelas no sentido clássico.

5. **Teorema do Traço**. A aplicação linear e contínua, $\gamma_0 : H^1(\Omega) \to L^2(\Gamma)$, denomina-se traço sobre Γ. Desta forma para toda função $u \in H^1(\Omega)$, o traço de $\gamma_0 u = u|_\Gamma \in L^2(\Gamma)$. Da continuidade da aplicação traço, obtém-se
$$\|\gamma_0 u\|_{L^2(\Gamma)} \leq \|u\|_1.$$

Tendo em vista os problemas de evolução que trateremos adiante, os seguintes conceitos e resultados matemáticos merecem menção:

Definição 8. *Seja Y um espaço de Banach real. Diz-se que a função $u : [0,T] \to Y$ é fracamente contínua em Y se, para todo $y' \in Y'$, a função $t \in [0,T] \mapsto \langle u(t), y' \rangle$ é função contínua. O espaço das funções fracamente contínuas em Y é denotado por $C_s([0,T];Y)$.*

Sejam X e Y espaços de Banach reais, sendo X um espaço reflexivo com injeção contínua de X em Y.

Lema 1. *Se $u \in L^\infty(0,T;X) \bigcap C_s([0,T];Y)$ então $u \in C_s([0,T];X)$.*

Lema 2. *Seja $W(0,T) = \left\{ u \in L^P(0,T;X); \dfrac{du}{dt} \in L^P(0,T;Y) \right\}$.*
Se $u \in W(0,T)$ então $u \in C[[0,T];Y]$.

Teorema de Aubin-Lions. *Sejam B_0, B e B_1 espaços de Banach com imersão compacta entre B_0 e B e imersão contínua entre B e B_1. Se uma sucessão $\{u_m\}_{m\in\mathbb{N}}$ satisfaz as condições*

$$\{u_m\}_{m\in\mathbb{N}} \text{ limitada em } L^P(0,T;B_0) \quad e \quad \{u'_m\}_{m\in\mathbb{N}} \text{ limitada em } L^P(0,T;B_1),$$

existe uma subsucessão $u_{m_k k\in\mathbb{N}}$ de u_m que converge forte em $L^P(0,T;B)$.

Um exemplo comum de espaços satisfazendo as imersões acima são: $B_0 = H_0^1(\Omega)$, $B = L^2(\Omega)$ e $B_1 = H^{-1}(\Omega)$ ou $B_1 = L^2(\Omega)$, com $\Omega \subset \mathbb{R}^n$ domínio limitado.

Desigualdade de Gronwall. *Sejam φ e ψ funções reais e integráveis em $[a,b]$, tais que $\varphi(t) \geq 0$ e $\psi(t) \geq 0$, $\forall t \in [a,b]$. Se α e β são não negativos e*

$$\varphi(t) \leq \alpha + \beta \int_a^t \varphi(s)\psi(s)\, ds,$$

então $\varphi(t) \leq \alpha \exp\left(\beta \int_a^t \psi(s)\, ds\right)$.

Em particular, se $\psi(t) = 1$ então, $\varphi(t) \leq \alpha e^{\beta(t-a)}$, $\quad \forall t \in [a,b]$.

1.6 Aproximação por Diferenças Finitas

Para se resolver uma equação diferencial de evolução (como as do tipo parabólico ou hiperbólico por exemplo) pelo método de elementos finitos, é usual por razões de estabilidade que a derivada no tempo seja aproximada pelo método das diferenças finitas. Dessa forma, para concluir esse capítulo, apresentaremos algumas considerações sobre a aproximação da derivada por diferenças finitas e sobre a ordem de convergência.

Seja f uma função suficientemente regular. Pelo Teorema de Taylor podemos expandir a função $f(x)$ na vizinhança de $x \in (a,b)$. Assim, se $\delta > 0$ é tal que $(x-\delta, x+\delta) \subset (a,b)$, tem-se para h suficientemente pequeno,

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{iv}(x) + \cdots \tag{1.53}$$

e de forma análoga,

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{iv}(x) - \cdots. \tag{1.54}$$

Somando os termos (1.53) e (1.54), obtemos

$$f(x-h) - 2f(x) + f(x+h) = h^2 f''(x) + \mathcal{O}(h^4), \tag{1.55}$$

onde $\mathcal{O}(h^4)$ denota todos os termos de potência maior ou igual a quatro em h. Admitindo que esses termos sejam pequenos quando comparados com potências inferiores de h ($h \ll 1$), podemos descartá-los para obter a seguinte aproximação para a derivada de segunda ordem da função,

$$f''(x) \simeq \frac{1}{h^2}\Big(f(x+h) - 2f(x) + f(x-h)\Big), \tag{1.56}$$

com o erro da aproximação de ordem $\mathcal{O}(h^2)$. A aproximação (1.56) é conhecida por diferença central.

Por outro lado, podemos obter uma aproximação pela diferença central para a primeira derivada de função $f(x)$, fazendo a diferença entre os termos (1.53) e (1.54), ou seja

$$f(x+h) - f(x-h) = 2hf'(x) + \mathcal{O}(h^3),$$

e dessa forma temos a seguinte aproximação por diferença central para a primeira derivada

$$f'(x) \simeq \frac{1}{2h}\Big(f(x+h) - f(x-h)\Big), \tag{1.57}$$

também com erro de aproximação de ordem $\mathcal{O}(h^2)$.

Desprezando os termos com potência dois ou superior de h em (1.53) e (1.54), temos também as seguintes aproximações para a primeira derivada:

$$f'(x) \simeq \frac{1}{h}\Big(f(x+h) - f(x)\Big) \qquad \text{(Diferença progressiva ou adiantada)}, \tag{1.58}$$

$$f'(x) \simeq \frac{1}{h}\Big(f(x) - f(x-h)\Big) \qquad \text{(Diferença regressiva ou atrasada)}, \tag{1.59}$$

sendo ambos os erros das aproximações de ordem $\mathcal{O}(h)$.

Notação:

Seja f uma função nas variáveis independentes $x \in [a,b]$ e $t \in [0,T]$ e consideremos a seguinte discretização uniforme: $a = x_0 < x_1 < \cdots x_M = b$ e $0 = t_0 < t_1 < \cdots t_N = T$, onde $h = x_{i+1} - x_i$ e $\Delta t = t_{n+1} - t_n$, são denominados passos. Assim $h = (b-a)/M$ e $\Delta t = T/N$ e cada elemento discreto pode ser obtido por,

$$\begin{array}{lll} x_i = x_0 + ih, & \text{para} & i = 1, 2, \cdots M \\ t_n = t_0 + n\Delta t = n\Delta t, & \text{para} & n = 1, 2, \cdots N. \end{array}$$

Vamos denotar a função f nos pontos discretos (x_i, t_n) da seguinte forma:

$$f(x_i, t_n) = f(x_0 + ih, n\Delta t) = f_i^n.$$

Com essa notação a diferença central (1.56) é dada por

$$\left(\frac{\partial^2 f(x,t)}{\partial x^2}\right)_{i,n} \simeq \frac{1}{h^2}\Big(f_{i+1}^n - 2f_i^n + f_{i-1}^n\Big), \qquad \text{(Diferença central "espaço")} \quad (1.60)$$

com erro de ordem $\mathcal{O}(h^2)$ e similarmente,

$$\left(\frac{\partial^2 f(x,t)}{\partial t^2}\right)_{i,n} \simeq \frac{1}{(\Delta t)^2}\Big(f_i^{n+1} - 2f_i^n + f_i^{n-1}\Big), \qquad \text{(Diferença central "tempo")} \quad (1.61)$$

com erro de ordem $\mathcal{O}(\Delta t^2)$. Para a diferença progressiva (regressiva) temos

$$\begin{array}{ll} \left(\dfrac{\partial f(x,t)}{\partial x}\right)_{i,n} \simeq \dfrac{1}{h}\Big(f_{i+1}^n - f_i^n\Big), & \text{(Diferença progressiva)} \\ \left(\dfrac{\partial f(x,t)}{\partial x}\right)_{i,n} \simeq \dfrac{1}{h}\Big(f_i^n - f_{i-1}^n\Big), & \text{(Diferença regressiva)} \\ \left(\dfrac{\partial f(x,t)}{\partial t}\right)_{i,n} \simeq \dfrac{1}{\Delta t}\Big(f_i^{n+1} - f_i^n\Big), & \text{(Diferença progressiva "tempo")} \\ \left(\dfrac{\partial f(x,t)}{\partial t}\right)_{i,n} \simeq \dfrac{1}{\Delta t}\Big(f_i^n - f_i^{n-1}\Big), & \text{(Diferença regressiva "tempo")} \end{array} \quad (1.62)$$

com erro de ordem $\mathcal{O}(h)$ e $\mathcal{O}(\Delta t)$. Por abuso de notação, usaremos o símbolo $=$ em lugar de $\simeq$.

CAPÍTULO 2

Problema Estacionário Unidimensional

O problema modelo que estudaremos a seguir é um dos mais típicos problemas elípticos e tem várias aplicações físicas. São estudadas as formulações forte e fraca do problema com diversas condições de fronteira e sua influência no sistema linear, consequência da aplicação do método de elementos finitos. Resultados numéricos são mostrados juntamente com os respectivos erros. No Apêndice são apresentadas passo-a-passo as etapas do programa computacional

2.1 Formulação do Problema

O problema modelo que abordaremos nesse capítulo é o de determinar uma função $u \in H_0^1(0,1)$ tal que

$$\begin{cases} -\alpha u''(x) + \beta u(x) = f(x) \ , \quad \forall\, x \in (0,1) \\ u(0) = u(1) = 0, \end{cases} \tag{2.1}$$

onde $\alpha > 0$ e $\beta \geq 0$ são constantes e f é uma função regular.

Dentre as diversas interpretações que a solução u do problema (2.1) acima representa, citemos a posição de equilíbrio de uma corda flexivel que está presa pelas extremidades, sendo f a força distribuída que atua sobre ela.

Além das condições de fronteira acima, outros tipos também serão considerados.

2.1.1 Formulação Variacional

O Método de elementos finitos não é aplicável diretamente ao problema (2.1). Assim, faz-se necessário expressar o problema numa forma mais conveniente para que seja possível aplicar o método de Galerkin.

Consideremos o espaço das funções teste em $(0,1)$,

$$\mathcal{D}(0,1) = \{v \in C_0^\infty(0,1);\ v(0) = v(1) = 0\}.$$

Multiplicando a equação (2.1) por $v \in \mathcal{D}(0,1)$ e integrando, obtemos

$$\int_0^1 -\alpha u'' v\ dx + \int_0^1 \beta u v\ dx = \int_0^1 f\, v\ dx \quad \forall\, v \in \mathcal{D}(0,1). \tag{2.2}$$

Assim, segue da integração por partes,

$$-\alpha \int_0^1 u'' v\ dx = -\alpha (u'v)\Big|_0^1 + \alpha \int_0^1 u'v'\ dx = \alpha \int_0^1 u'v'\, dx,$$

pois $v(0) = v(1) = 0$. Substituindo em (2.2), segue que

$$\alpha \int_0^1 u'v'\, dx + \beta \int_0^1 uv\ dx = \int_0^1 f\, v\, dx \quad \forall\, v \in \mathcal{D}(0,1). \tag{2.3}$$

Como $\mathcal{D}(0,1)$ é denso em $V = H_0^1(0,1)$, a igualdade (2.3) também é válida para todo $v \in V$. Assim, definindo

$$a(u,v) = \alpha \int_0^1 u'v'\, dx + \beta \int_0^1 uv\, dx, \tag{2.4}$$

e

$$(f : v) = \int_0^1 f\, v\, dx\ , \tag{2.5}$$

então (2.3) é equivalente a determinar $u \in H_0^1(0,1)$ satisfazendo:

$$a(u,v) = (f : v)\ ,\ \forall\, v \in V\ . \tag{2.6}$$

Vamos agora utilizar o Teorema de Lax-Milgram para mostrar que o problema variacional (2.6) tem uma única solução. Com efeito

1. $a(\cdot,\cdot)$ é bilinear e simétrica. Sejam u, v e w funções pertencentes ao espaço de Hilbert $V = H_0^1(0,1)$. Usando a definição (2.4), é imediato verificar que

$$a(u+w,v) = a(u,v)+a(w,v), \quad a(u,v+w) = a(u,v)+a(u,w), \quad a(u,v) = a(v,u).$$

2. $a(\cdot,\cdot)$ é contínua. De fato, usando a desigualdade de Schwartz, obtemos:

$$\begin{aligned}|a(u,v)| &\leq \alpha \int_0^1 |u'v'|\,dx + \beta \int_0^1 |uv|\,dx \\ &\leq \alpha \left(\int_0^1 |u'|^2\,dx\right)^{1/2} \left(\int_0^1 |v'|^2\,dx\right)^{1/2} \\ &\quad + \beta \left(\int_0^1 |u|^2\,dx\right)^{1/2} \left(\int_0^1 |v|^2\,dx\right)^{1/2} \\ &\leq \alpha \|u'\|_0 \|v'\|_0 + \beta \|u\|_0 \|v\|_0 \\ &\leq \delta_1 \|u\|_1 \|v\|_1,\end{aligned}$$

onde, na última desigualdade, usamos a equivalência das normas em $V = H_0^1$ e $\delta_1 = \max\{\alpha, \beta\}$. Logo a forma $a(\cdot,\cdot)$ é contínua em V.

3. $a(\cdot,\cdot)$ é coerciva. De fato, temos

$$a(u,u) \geq \alpha \int_0^1 |u'|^2\,dx + \beta \int_0^1 u^2\,dx \geq \delta_2 \|u\|_1^2, \quad \forall u \in V,$$

com $\delta_2 = \min\{\alpha, \beta\}$ se $\beta > 0$ e $\delta_2 = C_1\alpha$ se $\beta = 0$, onde C_1 é a constante devida à equivalência entre a noras $\|u\|_1$ e $\|u'\|_0$.

Vale observar que, embora a hipótese $\beta \geq 0$ nos permita mostrar de forma simples que a forma bilinear $a(\cdot,\cdot)$ é coerciva em $H_0^1(0,1)$, essa condição pode ser relaxada para a constante β "não muito negativas". De fato, pode-se mostrar neste caso a forma $a(\cdot,\cdot)$ é coerciva desde que $\beta \geq -\pi^2\alpha$.

Por outro lado, considerando $f \in L^2(0,1)$ em (2.5), vemos que a forma linear $\langle f, v\rangle = \int_0^1 f\,v\,dx$ é limitada em V, pois

$$|\langle f, v\rangle| \leq \int_0^1 |fv|\,dx \leq \|f\|_0 \|v\|_1 \qquad \forall v \in V.$$

Utilizando o Teorema de Lax-Milgram ([2]) obtemos a existência e unicidade da solução $u \in V = H_0^1(0,1)$. Além disso, utilizando o Teorema de regularidade elíptica (ver [2, 13, 14]), mostra-se que para $f \in L^2(0,1)$, tem-se $u \in V = H_0^1(0,1) \cap H^2(0,1)$ e, além disso,

$$\|u\|_2 \leq c\|f\|_0, \tag{2.7}$$

onde c é uma constante que somente depende de $0,1 = (0,1)$.

Pode-se mostrar que nestas condições de regularidade, os problemas (2.1) e (2.6) são equivalentes.

2.1.2 Método de Galerkin

O método de Galerkin consiste em aproximar o espaço das soluções por um subespaço de dimensão finita. Para isso, escolhemos um subespaço V_m gerado por um conjuntos linearmente independente de m elementos do espaço de Hilbert $H_0^1(\Omega)$, ou seja,

$$V_m = [\varphi_1, \varphi_2, ..., \varphi_m] \tag{2.8}$$

onde $\{\varphi_i,\, i \in \mathbb{N}\}$ é uma base hilbertiana de $H_0^1(\Omega)$. Assim, podemos buscar uma solução aproximada $u^h \in V_m$ do problema (2.6) no subespaço V_m.

• Problema Aproximado

Definimos a solução aproximada $u_h \in V_m$ do problema (2.6) por

$$a(u_h, v) = (f : v)\,, \qquad \forall\, v \in V_m. \tag{2.9}$$

Sendo $u_h \in V_m$, temos necessariamente

$$u_h(x) = \sum_{j=1}^{m} C_j \varphi_j(x)\,, \qquad \varphi_j \in V_m, \tag{2.10}$$

de modo que, substituindo u_h dado por (2.10) em (2.9) tem-se

$$a\left(\sum_{j=1}^{m} C_j \varphi_j(x) : v\right) = (f : v)\,, \qquad \forall\, v \in V_m.$$

Como $v \in V_m$, podemos tomar $v = \varphi_i$. Logo,

$$\sum_{j=1}^{m} a(\varphi_i, \varphi_j) C_j = (\varphi_i :\, f)\,, \qquad \text{para } i = 1, ..., m. \tag{2.11}$$

Denotando por $K = [K_{ij}]$ a matriz de ordem $m \times m$ (denominada matriz rigidez global ou matriz global), $F = [F_i]$ o vetor de ordem $m \times 1$ (denominado vetor força global) e $C = (C_1, ..., C_m)^T$ o vetor incógnita, obtemos o sistema linear

$$KC = F, \tag{2.12}$$

onde

$$K_{ij} = a(\varphi_i, \varphi_j) \quad \text{e} \quad F_i = (\varphi_i : f), \quad 1 \leq i,\, j \leq m.$$

De (2.4) e (2.5) a matriz rigidez e o vetor força são dados respectivamente por

$$K_{ij} = \int_0^1 \left(\alpha \frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx} + \beta \varphi_i \varphi_j\right) dx \tag{2.13}$$

$$F_i = \int_0^1 f \varphi_i \, dx \tag{2.14}$$

• **Propriedades da Matriz Rigidez**

(i) K é simétrica. De fato:

$$K_{ij} = \int_0^1 \left(\alpha \frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx} + \beta\varphi_i\varphi_j \right) dx = \int_0^1 \left(\alpha \frac{d\varphi_j}{dx}\frac{d\varphi_i}{dx} + \beta\varphi_j\varphi_i \right) dx = K_{ji} \tag{2.15}$$

(ii) K é definida positiva. De fato, seja $\boldsymbol{d} = (d_1, ..., d_m)^T$. Então,

$$\begin{aligned} \boldsymbol{d}^T K \boldsymbol{d} &= \sum_{i,j=1}^{m} d_i K_{ij} d_j = \sum_{i,j=1}^{m} d_i a(\varphi_i, \varphi_j) d_j \\ &= a\Big(\sum_{i=1}^{m} d_i\varphi_i, \sum_{j=1}^{m} d_j\varphi_j \Big) = a(v, v) > 0, \end{aligned} \tag{2.16}$$

pois

$$a(v, v) = \alpha \int_0^1 \left(\frac{dv}{dx} \right)^2 dx + \beta \int_0^1 v^2 dx \geq \delta_2 \|v\|^2_{H_0^1(\Omega)} > 0$$

para $\alpha, \beta > 0$, onde $\delta_2 = \min\{\alpha, \beta\}$. Por outro lado $\boldsymbol{d}^T K \boldsymbol{d} = 0 \Leftrightarrow a(v, v) = 0$. Assim $\|v\|^2_{H_0^1(\Omega)} = 0$. Como as funções φ_j são linearemente independentes, temos $d_j = 0$ para todo $j = 1, \ldots, m$.

2.2 Função de Interpolação

Para o cálculo da matriz global K e do vetor força F, precisamos definir explicitamente as funções φ_i que compõem a base do subespaço V_m. A escolha de φ_i é essencial para a otimização do sistema linear. No caso que estamos tratando, $\Omega = (0.1)$, de modo que podemos escolher as funções φ_i como sendo funções trigonométricas ou polinomiais, para as quais as condições de fronteira sejam satisfeitas.

O objetivo principal na escolha de φ_i é fazer com que a matriz K_{ij} não seja uma matriz cheia, isto é, seja uma matriz com muitos elementos nulos e, de preferência, que seus elementos obedeçam a uma certa ordem. Uma matriz nessas condições é denominada matriz esparsa e o sistema linear resultante é, em geral, bem condicionado.

Para a solução do problema (2.1), onde as condições de contorno são do tipo Dirichlet homogênas, consideremos as funções de interpolação linear por partes, satisfazendo as seguinte condição:

$$\varphi_i(x_j) = \begin{cases} 1, & \text{se } i = j, \\ 0, & \text{se } i \neq j, \end{cases} \tag{2.17}$$

onde $x_j \in [0,1]$ é denominado ponto nodal ou simplesmente nó. Os nós são pontos discretos do intervalo $[0,1]$ distribuídos de forma equidistante ou não. Tomando m divisões em $[0,1]$, definimos o passo

$$h_i = x_{i+1} - x_i, \qquad i = 1, ..., m. \tag{2.18}$$

No caso dos nós serem equidistantes, $h_i = h = 1/m$.

Em cada nó x_i, definimos a função φ_i linear por partes satisfazendo a condição (2.17), ou seja, φ_i para $i = 1, ..., m$ é definida por

$$\varphi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}} = \dfrac{x - x_{i-1}}{h_{i-1}}\,, & \forall\, x \in [x_{i-1}, x_i], \\ \dfrac{x - x_{i+1}}{x_i - x_{i+1}} = \dfrac{x_{i+1} - x}{h_i}\,, & \forall\, x \in [x_i, x_{i+1}], \\ 0\,, & \forall\, x \notin [x_{i-1}, x_{i+1}]. \end{cases} \tag{2.19}$$

Geometricamente, as funções φ_i podem ser representadas por

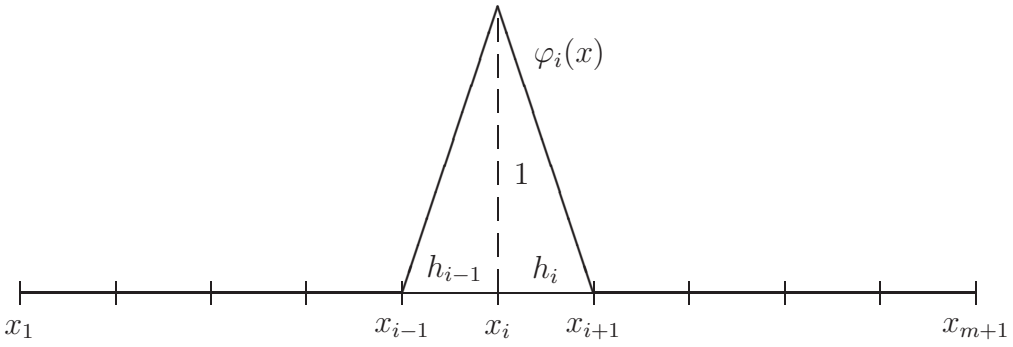


Figura 2.1: Função base

De (2.19) podemos calcular a derivada de $\varphi_i(x)$, obtendo-se:

$$\frac{d\varphi_i}{dx}(x) = \begin{cases} \dfrac{1}{h_{i-1}} & \forall\, x \in (x_{i-1}, x_i), \\ -\dfrac{1}{h_i} & \forall\, x \in (x_i, x_{i+1}), \\ 0 & \forall\, x \notin (x_{i-1}, x_{i+1}). \end{cases} \tag{2.20}$$

A derivada da função $\varphi_i(x)$ é descontínua no ponto x_i, mas isto não afeta o cálculo da matriz K_{ij}, pois $\varphi_i'(x)$ é contínua (constante) por partes.

É claro que o espaço V_m gerado pelas funções φ_i é um subespaço do espaço $V = H^1(0,1)$, pois $\varphi_i \in C^0[0,1] \subset L^2(0,1)$ e $\varphi_i' \in L^2(0,1)$.

• A Matriz Rigidez

Utilizando a função φ_i, vemos que a matriz K_{ij} é uma matriz tridiagonal. De fato, por definição

$$K_{ij} = \int_0^1 \left(\alpha \frac{d\varphi_i}{dx} \frac{d\varphi_j}{dx} + \beta \varphi_i \varphi_j \right) dx$$

e como $\varphi_i \varphi_j \equiv 0$ se $|i-j| \geq 2$, os termos da matriz K_{ij} não necessariamente nulos são $K_{i-1,i}$, $K_{i,i}$ e $K_{i,i+1}$, ou seja, a matriz tem a seguinte forma;

$$K = \begin{bmatrix} * & * & 0 & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 & 0 \\ 0 & * & * & * & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & * & * & * & 0 \\ 0 & 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & * & * \end{bmatrix} \tag{2.21}$$

Como a matriz K é simétrica, somente são necessários os cálculos de $K_{i,i}$ e $K_{i,i+1}$ dados por

$$\begin{aligned} K_{ii} &= \alpha \int_0^1 \left(\frac{d\varphi_i}{dx} \right)^2 dx + \beta \int_0^1 \varphi_i^2 \, dx = \alpha \int_{x_{i-1}}^{x_{i+1}} \left(\frac{d\varphi_i}{dx} \right)^2 dx + \beta \int_{x_{i-1}}^{x_{i+1}} \varphi_i^2 dx \\ &= \alpha \left[\int_{x_{i-1}}^{x_i} \left(\frac{d\varphi_i}{dx} \right)^2 dx + \int_{x_i}^{x_{i+1}} \left(\frac{d\varphi_i}{dx} \right)^2 dx \right] + \beta \left[\int_{x_{i-1}}^{x_i} \varphi_i^2 \, dx + \int_{x_i}^{x_{i+1}} \varphi_i^2 \, dx \right]. \end{aligned}$$

Calculando cada uma das integrais. obtemos

$$K_{ii} = \alpha \left(\frac{1}{h_{i-1}} + \frac{1}{h_i} \right) + \frac{\beta}{3} \left(h_{i-1} + h_i \right), \qquad i = 2, 3, ..., m. \tag{2.22}$$

Para o elemento $K_{i,i+1}$, temos:

$$\begin{aligned} K_{i,i+1} &= \alpha \int_{x_i}^{x_{i+1}} \frac{d\varphi_i}{dx} \frac{d\varphi_{i+1}}{dx} \, dx + \beta \int_{x_i}^{x_{i+1}} \varphi_i \varphi_{i+1} \, dx \\ &= \alpha \int_{x_i}^{x_{i+1}} \left(-\frac{1}{h_i^2} \right) dx + \beta \int_{x_i}^{x_{i+1}} \left(\frac{x_{i+1} - x}{h_i} \right) \left(\frac{x - x_i}{h_i} \right) dx = -\frac{\alpha}{h_i} + \frac{\beta h_i}{6}. \end{aligned}$$

Logo,

$$K_{i,i+1} = K_{i+1,i} = -\frac{\alpha}{h_i} + \frac{\beta h_i}{6}. \tag{2.23}$$

Devido às condições de fronteira, $K_{1,1} = K_{m+1,m+1} = 0$. Assim, a matriz dos coeficientes tem ordem $(m-1) \times (m-1)$ e é dada por;

$$K = \begin{bmatrix} a_1 & b_1 & 0 & 0 & \cdots & 0 & 0 \\ b_1 & a_2 & b_2 & 0 & \cdots & 0 & 0 \\ 0 & b_2 & a_3 & b_3 & \cdots & 0 & 0 \\ 0 & 0 & b_3 & \ddots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & b_{m-3} & 0 \\ 0 & 0 & 0 & 0 & \cdots & a_{m-2} & b_{m-2} \\ 0 & 0 & 0 & 0 & \ldots & b_{m-2} & a_{m-1} \end{bmatrix} \tag{2.24}$$

onde

$$a_{i-1} = K_{ii} = \alpha \left(\frac{1}{h_i} + \frac{1}{h_{i-1}} \right) + \frac{\beta}{3}(h_i + h_{i-1}), \qquad i = 2, 3, \ldots, m \tag{2.25}$$

e

$$b_{i-1} = K_{i,i+1} = \left(-\frac{\alpha}{h_i} + \frac{\beta h_i}{6} \right), \qquad i = 2, 3, \ldots, m. \tag{2.26}$$

Se a malha é uniforme, $h = h_i$, de modo que

$$a = \frac{2\alpha}{h} + \frac{2\beta h}{3} \quad \text{e} \quad b = -\frac{\alpha}{h} + \frac{\beta h}{6}. \tag{2.27}$$

• O Vetor Força

Através da função de interpolação φ_i dada por (2.19), o vetor força F definido em (2.14) pode ser calculado diretamente por

$$\begin{aligned} F_i &= \int_0^1 f\varphi_i \, dx = \int_{x_{i-1}}^{x_i} f\varphi_i \, dx + \int_{x_i}^{x_{i+1}} f\varphi_i \, dx \\ &= \frac{1}{h_{i-1}} \int_{x_{i-1}}^{x_i} f(x)(x - x_{i-1}) \, dx + \frac{1}{h_i} \int_{x_i}^{x_{i+1}} f(x)(x_{i+1} - x) \, dx, \end{aligned} \tag{2.28}$$

para $i = 2, 3, \ldots, m$. Por exemplo, tomando $f(x) = x$ para todo $x \in [0, 1]$, as integrais acima podem ser calculadas explicitamente.

$$F_i = \frac{h_{i-1}}{6}(3x_{i-1} + 2h_{i-1}) + \frac{h_i}{6}(3x_i + h_i), \quad i = 2, 3, \ldots, m. \tag{2.29}$$

Para a malha uniforme, temos

$$F_i = hx_i\,, \qquad i = 2, 3, \ldots, m. \tag{2.30}$$

No caso em que seja necessário calcular as integrais em (2.28) numericamente, podemos calcular a força F através da interpolação de f, usando as funções base φ_i como interpoladores. Com efeito, a função f pode ser interpolada pela função φ_i da seguinte forma:

$$f(x) = \sum_{i=1}^{m+1} \varphi_i(x) f_i, \tag{2.31}$$

onde $f_i = f(x_i)$. Portanto, obtemos

$$F_i = \int_0^1 f\varphi_i\, dx = \int_0^1 \left(\sum_{j=1}^{m+1} (\varphi_i\varphi_j) f_j \right) dx = \sum_{j=1}^{m+1} \int_{x_{i-1}}^{x_{i+1}} (\varphi_i\varphi_j) f_j\, dx.$$

Usando a definição de φ_i e considerando o suporte compacto, resulta que

$$\begin{aligned} F_i &= \sum_{j=i-1}^{i+1} \int_{x_{i-1}}^{x_{i+1}} (\varphi_i\varphi_j) f_j\, dx \\ &= f_{i-1} \int_{x_{i-1}}^{x_i} (\varphi_i\varphi_{i-1})\, dx + f_i \int_{x_{i-1}}^{x_{i+1}} (\varphi_i)^2\, dx + f_{i+1} \int_{x_i}^{x_{i+1}} (\varphi_i\varphi_{i+1})\, dx. \end{aligned} \tag{2.32}$$

Fazendo os cálculos das integrais, obtemos

$$\int_{x_{i-1}}^{x_i} (\varphi_i\varphi_{i-1})\, dx = \frac{h_{i-1}}{6}, \quad \int_{x_{i-1}}^{x_{i+1}} (\varphi_i)^2\, dx = \frac{1}{3}(h_{i-1} + h_i), \quad \int_{x_i}^{x_{i+1}} (\varphi_i\varphi_{i+1})\, dx = \frac{h_i}{6}.$$

Substituindo em (2.32) obtemos:

$$F_i = f_{i-1}\frac{h_{i-1}}{6} + f_i\frac{h_i + h_{i-1}}{3} + f_{i+1}\frac{h_i}{6}, \tag{2.33}$$

que coincide com a força calculada anteriormente, quando tomamos $f(x) = x$. Se a malha é uniforme, então,

$$F_i = \frac{h}{6}\{f_{i-1} + 4f_i + f_{i+1}\}. \tag{2.34}$$

Agora, utilizando a matriz rigidez K dada por (2.24) e a força F dada por (2.33), obtemos o sistema linear $KC = F$, onde $C = (C_2, \ldots, C_m)^T$ é o vetor incógnita. As coordenadas C_i do vetor C são, exatamente, o valor da solução aproximada $u_h = u_h(x_i)$, para todo $i = 2, \ldots, m$. Com efeito, fazendo $x = x_i$ e substituindo em (2.10) temos:

$$u_h\,(x_i) = \sum_{j=2}^{m} C_j\varphi_j(x_i) = C_i. \tag{2.35}$$

2.3 Sistema Linear

Para resolver o sistema linear, existem vários métodos diretos ou iterativos. Em razão da matriz K ser tridiagonal, utilizaremos apropriadamente o Método de Eliminação de Gauss. Um outro método adequado para a matriz acima é o Método de Cholesky (usando a propriedade da matriz ser definida positiva e simétrica).

A matriz K e o vetor força F são dados por

$$K = \begin{bmatrix} a_1 & b_1 & 0 & 0 & \cdots & 0 & 0 \\ b_1 & a_2 & b_2 & 0 & \cdots & 0 & 0 \\ 0 & b_2 & a_3 & b_3 & \cdots & 0 & 0 \\ 0 & 0 & b_3 & \ddots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & b_{m-3} & 0 \\ 0 & 0 & 0 & 0 & \cdots & a_{m-2} & b_{m-2} \\ 0 & 0 & 0 & 0 & ... & b_{m-2} & a_{m-1} \end{bmatrix} \quad \text{e} \quad F = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ \vdots \\ F_{m-2} \\ F_{m-1} \end{bmatrix}$$

Definimos:

$$d_1 = a_1, \quad G_1 = F_1, \quad m_1 = \frac{b_1}{d_1},$$

e para $i = 2, 3, \ldots m-1$ sucessivamente,

$$d_i = a_i - m_{i-1} b_{i-1}, \quad G_i = F_i - m_{i-1} G_{i-1}, \quad m_i = \frac{b_i}{d_i}, \tag{2.36}$$

onde $d_i = d_{ii}$, é o elemento da diagonal principal. No final do procedimento obtemos a matriz triangular superior:

$$\widehat{K} = \begin{bmatrix} d_1 & b_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & d_2 & b_2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & d_3 & b_3 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \ddots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & b_{m-3} & 0 \\ 0 & 0 & 0 & 0 & \cdots & d_{m-2} & b_{m-2} \\ 0 & 0 & 0 & 0 & ... & 0 & d_{m-1} \end{bmatrix}$$

e o vetor força

$$\widehat{F} = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_{m-1} \end{bmatrix}$$

A solução do sistema equivalente $\widehat{K}C = \widehat{F}$, obtida por retro-substituição, é dada por

$$C_{m-1} = \frac{G_{m-1}}{d_{m-1}}.$$

Para $i = m-2, m-3, \ldots, 1$, temos:

$$C_i = \frac{1}{d_i}(G_i - b_i C_{i+1}). \tag{2.37}$$

De (2.35), a solução aproximada $u_h(x_i) = C_i$.

2.4 Matriz Local e Força Local

Introduziremos os conceitos de matriz local e força local para o problema modelo (2.6). A matriz local e a força local contribuirão para a formação da matriz global (rigidez) K e o vetor força F .

Este procedimento não tem vantagem sobre o anterior no caso unidimensional. Para os problemas bidimensionais ou tridimensionais, esta formulação local é significativamente mais simples. Assim, é razoável introduzir este conceito no caso unidimensional para facilitar a compreensão nos casos de dimensõe superiores.

Considere $\Omega = (0,1)$ e uma discretização não necessariamente uniforme dada por

$$x_{i+1} = x_i + h_i\ , \qquad i = 1, 2, \ldots, m,$$

onde $x_1 = 0$ e $x_{m+1} = 1$.

Para cada intervalo $[x_i, x_{i+1}]$, considere um elemento e, denominado elemento finito e as coordenadas locais $[x_1^e, x_2^e] = [x_i, x_{i+1}]$. Geometricamente os m elementos podem ser representados por

Para cada intervalo local $[x_1^e, x_2^e]$ do elemento e, definimos a função de interpolação local dada por

$$\varphi_a^e(x) = \begin{cases} \varphi_1^e = \dfrac{x_2^e - x}{h_e}, & \forall\ x \in [x_1^e, x_2^e], \\ \varphi_2^e = \dfrac{x - x_1^e}{h_e}, & \forall\ x \in [x_1^e, x_2^e], \\ 0, & \forall\ x \notin [x_1^e, x_2^e], \end{cases} \tag{2.38}$$

onde $h_e = x_2^e - x_1^e$.

A função de interpolação φ_i definida em (2.19) é a junção das funções de interpolação local φ_2^{e-1} e φ_1^e, ou seja

$$\varphi_i(x) = \begin{cases} \varphi_2^{e-1}, & \forall\ x \in [x_1^{e-1}, x_2^{e-1}] = [x_{i-1}, x_i], \\ \varphi_1^e\ , & \forall\ x \in [x_1^e, x_2^e] = [x_i, x_{i+1}], \\ 0 & \forall\ x \notin [x_{i-1}, x_{i+1}]. \end{cases}$$

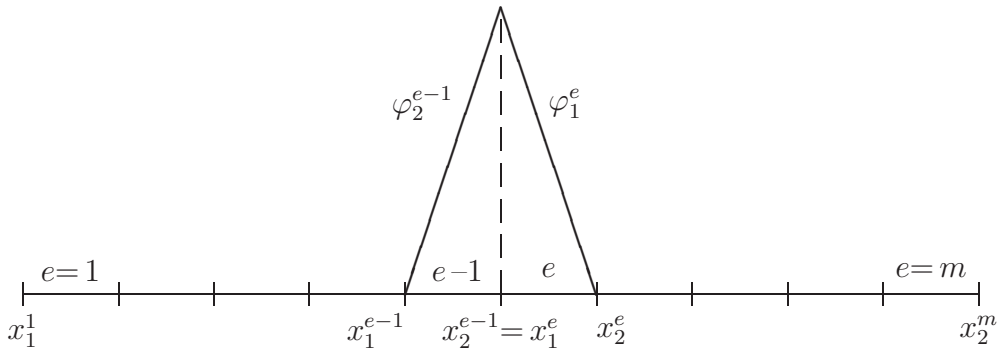


Figura 2.2: Função base local

A matriz global K e o vetor força F são definidos respectivamente por

$$K_{ij} = \int_0^1 \left(\alpha \frac{d\varphi_i}{dx} \frac{d\varphi_j}{dx} + \beta \varphi_i \varphi_j \right) dx \quad \text{e} \quad F_i = \int_0^1 f\ \varphi_i \, dx,$$

para $1 \leq i, j \leq m+1$.

Restringindo a matriz K e a força F a cada elemento finito e, temos:

$$K_{ab}^e = \int_{x_1^e}^{x_2^e} \left(\alpha \frac{d\varphi_a^e}{dx} \frac{d\varphi_b^e}{dx} + \beta \varphi_a^e \varphi_b^e \right) dx, \qquad 1 \leq a, b \leq 2, \tag{2.39}$$

e

$$F_a^e = \int_{x_1^e}^{x_2^e} f \varphi_a^e \, dx, \qquad 1 \leq a \leq 2, \tag{2.40}$$

que denominamos de matriz local e força local, respectivamente.

Para os $(m+1)$ nós da discretização de $\Omega = (0,1)$ temos m elementos. Logo:

$$K = \sum_{e=1}^{m} K^e \qquad \text{e} \qquad F = \sum_{e=1}^{m} F^e. \tag{2.41}$$

No intervalo $[x_1^e, x_2^e]$ as únicas funções de interpolação não nulas são as funções φ_1^e e φ_2^e, definidas em (2.38). Assim, na matriz K_{ab}^e, os únicos elementos não necessariamente nulos são os elementos K_{11}^e, K_{12}^e, K_{21}^e e K_{22}^e que pertencem a e-ésima e $(e+1)$-ésima linha e coluna, respectivamente. Temos, então,

$$K^e = \begin{array}{c} \begin{array}{cccc} & e & e+1 & \end{array} \\ \left[\begin{array}{cccc} 0 & \downarrow & \downarrow & 0 \\ & K^e_{11} & K^e_{12} & \longleftarrow \\ & K^e_{21} & K^e_{22} & \longleftarrow \\ 0 & & & 0 \end{array}\right] \end{array} \begin{array}{l} \\ e \\ e+1 \\ \\ \end{array}$$

Os elementos $K^e_{11}, K^e_{12}, K^e_{21}$ e K^e_{22} podem ser representados por $K^e_{ee}, K^e_{e,e+1}, K^e_{e+1,e}$ e $K^e_{e+1,e+1}$, respectivamente. Analogamente, para a força F^e, os únicos elementos não necessariamente nulos são F^e_1 e F^e_2. Assim,

$$F^e = [0, \ldots, 0, \underbrace{F^e_1, F^e_2}, 0, \ldots, 0]^t,$$

onde $F^e_1 = F^e_e$ e $F^e_2 = F^e_{e+1}$.

Para evitar o problema de armazenamento de matriz e uma quantidade grande de operações entre elementos nulos, considere a submatriz K^e de ordem 2×2 e F^e de ordem 2×1 formadas respectivamente pelos coeficientes e coordenadas não nulos, ou seja,

$$K^e = \begin{bmatrix} K^e_{11} & K^e_{12} \\ K^e_{21} & K^e_{22} \end{bmatrix} \qquad \text{e} \qquad F^e = \begin{bmatrix} F^e_1 \\ F^e_2 \end{bmatrix} \tag{2.42}$$

A matriz local K^e é uma matriz padrão dos elementos e somente serão diferentes se $h_e = x^e_2 - x^e_1$ for diferente para cada e.

Um passo importante é a alocação dos elementos da matriz local K^e para a matriz rigidez K.

2.5 Matriz Global e Força Global

Considere a matriz local e a força local definidas, respectivamente, em (2.39) e (2.40) e a função $\varphi^e_a(x)$ definida por (2.38), com $[x^e_1, x^e_2] = [x_e, x_{e+1}]$. Para calcular o coeficiente K_{ee} da matriz global K, basta fazer $i = j = e$. Assim,

$$\begin{aligned} K_{ee} &= \int_0^1 \left(\alpha \frac{d\varphi_e}{dx}\frac{d\varphi_e}{dx} + \beta\varphi_e\varphi_e\right) dx \\ &= int_{x_{e-1}}^{x_e} \left(\alpha \frac{d\varphi_e}{dx}\frac{d\varphi_e}{dx} + \beta\varphi_e\varphi_e\right) dx + \int_{x_e}^{x_{e+1}} \left(\alpha \frac{d\varphi_e}{dx}\frac{d\varphi_e}{dx} + \beta\varphi_e\varphi_e\right) dx. \end{aligned}$$

Em termos da contribuição local, obtemos

$$K_{ee} = \int_{x_1^{e-1}}^{x_2^{e-1}} \left(\alpha \frac{d\varphi_2^{e-1}}{dx} \frac{d\varphi_2^{e-1}}{dx} + \beta \varphi_2^{e-1} \varphi_2^{e-1} \right) dx + \int_{x_1^e}^{x_2^e} \left(\alpha \frac{d\varphi_1^e}{dx} \frac{d\varphi_1^e}{dx} + \beta \varphi_1^e \varphi_1^e \right) dx$$
$$= K_{22}^{e-1} + K_{11}^e.$$

Logo, o coeficiente K_{ee} da matriz global K recebe contribuição dos elementos finitos $(e-1)$ e do elemento e através da relação

$$K_{ee} = K_{22}^{e-1} + K_{11}^e, \qquad e = 2, 3, \ldots, m, \tag{2.43}$$

onde K_{22}^{e-1} e K_{11}^e são os coeficientes da matriz local (de ordem 2×2) K^{e-1} e K^e.

De forma análoga, o coeficiente $K_{e,e+1} = K_{e+1,e}$ da matriz global K é dado por

$$K_{e,e+1} = \int_0^1 \left(\alpha \frac{d\varphi_e}{dx} \frac{d\varphi_{e+1}}{dx} + \beta \varphi_e \varphi_{e+1} \right) dx = \int_{x_e}^{x_{e+1}} \left(\alpha \frac{d\varphi_e}{dx} \frac{d\varphi_{e+1}}{dx} + \beta \varphi_e \varphi_{e+1} \right) dx$$
$$= \int_{x_1^e}^{x_2^e} \left(\alpha \frac{d\varphi_1^e}{dx} \frac{d\varphi_2^e}{dx} + \beta \varphi_1^e \varphi_2^e \right) dx = K_{12}^e.$$

Assim, o coeficiente global $K_{e,e+1}$ recebe apenas a contribuição local do coeficiente K_{12}^e da matriz local do elemento e.

De forma geral, a contribuição local para os coeficientes da matriz global é dada pelo seguinte algoritmo:

Para $e = 2, 3, \ldots, m$, temos:

$$\begin{cases} K_{ee} = K_{22}^{e-1} + K_{11}^e, \\ K_{e,e+1} = K_{12}^e, \end{cases} \tag{2.44}$$

Para os coeficientes K_{11}, $K_{m+1,m+1}$ relativo aos nós extremos, temos

$$K_{11} = \int_0^1 \left(\alpha \frac{d\varphi_1}{dx} \frac{d\varphi_1}{dx} + \beta \varphi_1 \varphi_1 \right) dx = \int_{x_1^1}^{x_2^1} \left(\alpha \frac{d\varphi_1}{dx} \frac{d\varphi_1}{dx} + \beta \varphi_1 \varphi_1 \right) dx$$
$$= \int_{x_1^1}^{x_2^1} \left(\alpha \frac{d\varphi_1^1}{dx} \frac{d\varphi_1^1}{dx} + \beta \varphi_1^1 \varphi_1^1 \right) dx = K_{11}^1, \tag{2.45}$$

pois,

$$\varphi_1(x) = \begin{cases} \varphi_1^1(x) = \dfrac{x_2^1 - x}{h_1}, & \forall x \in [x_1^1, x_2^1], \\ 0, & \forall x \notin [x_1^1, x_2^1]. \end{cases}$$

De forma análoga, temos:

$$\varphi_{m+1}(x) = \begin{cases} 0, & \forall x \notin [x_1^m, x_2^m], \\ \varphi_2^m(x) = \dfrac{x - x_1^m}{h_m}, & \forall x \in [x_1^m, x_2^m] = [x_m, x_{m+1}], \end{cases}$$

de modo que,

$$K_{m+1,m+1} = \int_{x_1^m}^{x_2^m} \left(\alpha \frac{d\varphi_2^m}{dx} \frac{d\varphi_2^m}{dx} + \beta \varphi_2^m \varphi_2^m \right) dx = K_{22}^m. \tag{2.46}$$

De (2.44), (2.45) e (2.46) obtemos a matriz global K dada pelo algoritmo:

$$\begin{array}{ll} K_{11} = K_{11}^1, & \\ K_{ee} = K_{22}^{e-1} + K_{11}^e, & e = 2, 3, \ldots, m, \\ K_{e,e+1} = K_{e+1,e} = K_{12}^e, & e = 1, 2, 3, \ldots, m, \\ K_{m+1,m+1} = K_{22}^m. & \end{array} \tag{2.47}$$

Observe que no caso das condições de fronteria do tipo Dirichlet homogêneas que tratamos acima, os coeficientes K_{11} e $K_{m+1,m+1}0$ são nulos. No entanto, convém mantê-los explícitos no algoritmo acima, tendo em vista a solução de problemas com outros tipos de condições de fronteira.

No que se refere ao vetor força, segue da definição de φ_a^e em (2.38), com $[x_e, x_{e+1}] = [x_1^e, x_2^e]$,

$$\begin{aligned} F_e &= \int_0^1 f(x)\varphi_a^e(x)\, dx = \int_{x_{e-1}}^{x_{e+1}} f(x)\varphi_a^e(x)\, dx \\ &= \int_{x_1^{e-1}}^{x_2^{e-1}} f(x)\varphi_2^{e-1}(x)\, dx + \int_{x_1^e}^{x_2^e} f(x)\varphi_1^e(x)\, dx = F_2^{e-1} + F_1^e. \end{aligned} \tag{2.48}$$

Para F_1 e F_{m+1}, temos

$$F_1 = \int_0^1 f(x)\varphi_1(x)\, dx = \int_{x_1^1}^{x_2^1} f(x)\varphi_1^1(x)\, dx = F_1^1, \tag{2.49}$$

$$F_{m+1} = \int_0^1 f(x)\varphi_{m+1}(x)\, dx = \int_{x_1^m}^{x_2^m} f(x)\varphi_2^m(x)\, dx = F_2^m. \tag{2.50}$$

De (2.48), (2.49) e (2.50) temos,

$$\begin{array}{ll} F_1 = F_1^1, & \\ F_e = F_2^{e-1} + F_1^e, & e = 2.3, \ldots, m, \\ F_{m+1} = F_2^m. & \end{array} \tag{2.51}$$

Para determinar explicitamente a matriz global K e o vetor força F, precisamos calcular as matrizes locais K^e e a força local F^e.

• Matriz Global

Cada matriz local K^e é dada por

$$K^e = \begin{bmatrix} K^e_{11} & K^e_{12} \\ K^e_{21} & K^e_{22} \end{bmatrix},$$

onde

$$K^e_{11} = \int_{x^e_1}^{x^e_2} \left(\alpha \frac{d\varphi^e_1}{dx}\frac{d\varphi^e_1}{dx} + \beta \varphi^e_1 \varphi^e_1 \right) dx = K^e_{22}$$

e

$$K^e_{21} = K^e_{12} = \int_{x^e_1}^{x^e_2} \left(\alpha \frac{d\varphi^e_1}{dx}\frac{d\varphi^e_2}{dx} + \beta \varphi^e_1 \varphi^e_2 \right) dx.$$

De (2.38), tem-se que

$$\varphi^e_1 = \frac{x^e_2 - x}{h_e}, \qquad \varphi^e_2 = \frac{x - x^e_1}{h_e}.$$

Logo,

$$\frac{d\varphi^e_1}{dx} = -\frac{1}{h_e}, \qquad \frac{d\varphi^e_2}{dx} = \frac{1}{h_e}.$$

Substituindo e fazendo os cálculos obtém-se:

$$K^e = \begin{bmatrix} \dfrac{\alpha}{h_e} + \dfrac{\beta h_e}{3} & \dfrac{-\alpha}{h_e} + \dfrac{\beta h_e}{6} \\ \dfrac{-\alpha}{h_e} + \dfrac{\beta h_e}{6} & \dfrac{\alpha}{h_e} + \dfrac{\beta h_e}{3} \end{bmatrix}. \tag{2.52}$$

Usando o algoritmo (2.47), obtemos a matriz global que é tridiagonal e simétrica:

$$K = \begin{bmatrix} K_{11} & K_{12} & 0 & \cdots & 0 & 0 \\ K_{21} & K_{22} & K_{23} & \cdots & 0 & 0 \\ 0 & K_{32} & K_{33} & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \ddots & K_{m,m} & K_{m,m+1} \\ 0 & 0 & 0 & 0 & K_{m+1,m} & K_{m+1,m+1} \end{bmatrix} \tag{2.53}$$

É claro que a matriz global K calculada em (2.24) e (2.53) é a mesma, exceto que em (2.24) foram consideradas as condições de fronteira e, portanto, todos os elementos da primeira e última fila são zeros.

A matriz local (2.52) é válida para todo elemento $e = 1, 2, \ldots, m$. Logo, para obtenção da matriz global K é suficiente o uso do algoritmo (2.47). No próximo capítulo mostraremos que, para o caso bidimensional, a sistemática é a mesma e assim adotaremos este procedimento, pois facilita a obtenção da matriz global do sistema linear $KC = F$ e também do vetor força F.

• Vetor Força

Do algoritmo (2.51) tem-se que $F_1 = F_1^1$ e $F_{m+1} = F_2^m$; e de (2.48), $F_e = F_2^{e-1} + F_1^e$, para $e = 2, 3, \ldots, m$.

Para obter F_e explicitamente, calculemos F_2^{e-1} e F_1^e. Usando a interpolação local da função $f(x)$ dada por

$$f(x) = \sum_{a=1}^{2} f_a^e \ \varphi_a^e(x)\, dx,$$

co, $f(x_a^e) = f_a^e$, podemos escrever

$$F_1^e = \int_{x_1^e}^{x_2^e} f(x)\varphi_1^e(x)\, dx = f_1^e \int_{x_1^e}^{x_2^e} (\varphi_1^e(x)\varphi_1^e(x))\, dx + f_2^e \int_{x_1^e}^{x_2^e} (\varphi_2^e(x)\, \varphi_1^e(x))\, dx, \quad (2.54)$$

onde $f_1^e = f(x_1^e) = f(x_e)$ e $f_2^e = f(x_2^e) = f(x_{e+1})$. Agora, de (2.38) temos,

$$\int_{x_1^e}^{x_2^e} (\varphi_1^e(x))^2\, dx = \frac{h_e}{3} \quad (2.55)$$

e

$$\int_{x_1^e}^{x_2^e} (\varphi_2^e(x)\varphi_1^e(x))\, dx = \frac{h_e}{6}. \quad (2.56)$$

Substituindo em (2.54) conclui-se que:

$$F_1^e = \frac{h_e}{3} f_1^e + \frac{h_e}{6} f_2^e. \quad (2.57)$$

De forma análoga, temos,

$$\begin{aligned} F_2^e &= \int_{x_1^e}^{x_2^e} f(x)\varphi_2^e(x)\, dx = \int_{x_1^e}^{x_2^e} (\varphi_1^e(x)\varphi_2^e(x)) f_1^e(x)\, dx + \int_{x_1^e}^{x_2^e} (\varphi_2^e(x))^2 f_2^e(x)\, dx \\ &= \frac{h_e}{6} f_1^e + \frac{h_e}{3} f_2^e \end{aligned} \quad . \quad (2.58)$$

Portanto, para $e = 2, 3, \ldots, m$, a força F_e é dada por

$$F_e = F_2^{e-1} + F_1^e = \frac{h_{e-1}}{6} \left(f_1^{e-1} + 2 f_2^{e-1} \right) + \frac{h_e}{6} \left(2 f_1^e + f_2^e \right). \quad (2.59)$$

De (2.49) e (2.50), F_1 e F_{m+1} são calculados por

$$\begin{aligned} F_1 &= \int_{x_1^1}^{x_2^1} f(x)\varphi_1^1(x)\,dx = f_1^1 \int_{x_1^1}^{x_2^1} (\varphi_1^1(x))^2\,dx + f_2^1 \int_{x_1^1}^{x_2^1} (\varphi_2^1(x)\varphi_1^1(x))\,dx \\ &= \frac{h_1}{3} f_1^1 + \frac{h_1}{6} f_2^1, \end{aligned}$$

isto é,

$$F_1 = F_1^1 = \frac{h_1}{6}(2f_1^1 + f_2^1) \tag{2.60}$$

e

$$\begin{aligned} F_{m+1} &= \int_{x_1^m}^{x_2^m} f(x)\varphi_2^m(x)\,dx = f_1^m \int_{x_1^m}^{x_2^m} (\varphi_1^m(x)\varphi_2^m(x))\,dx + f_2^m \int_{x_1^m}^{x_2^m} (\varphi_2^m(x)\,)^2\,dx \\ &= \frac{h_m}{6} f_1^m + \frac{h_m}{3} f_2^m, \end{aligned}$$

ou seja,

$$F_{m+1} = F_2^m = \frac{h_m}{6}(f_1^m + 2f_2^m). \tag{2.61}$$

Portanto, o vetor global F é dado por

$$F = (F_1, \ldots, F_{m+1})^T \tag{2.62}$$

Observe que a força F acima coincide com (2.33), exceto pela primeira e última linha que são as condições de fronteira. Em particular, se a malha é uniforme, tem-se:

$$\begin{aligned} F_1 &= \frac{h}{6}\left(2f_1^1 + f_2^1\right), \\ F_e &= \frac{h}{6}(f_1^{e-1} + 4f_1^e + f_2^e) \quad \text{para } e = 2, 3, \ldots, m, \\ F_{m+1} &= \frac{h}{6}(f_1^m + 2f_2^m). \end{aligned}$$

2.6 Integração Numérica

A solução aproximada através do Método de Galerkin é dada por

$$u_h(x) = \sum_{i=1}^{m+1} C_i\varphi_i(x),$$

onde $\varphi_i(x)$ é uma função de interpolação linear. Assim, como definida acima, a solução aproximada $u_h = u_h(x)$ do problema

$$a(u_m, v) = (f : v), \qquad \forall v \in V_m$$

também é um polinômio de grau um por partes.

Em geral, podemos melhorar a solução aproximada aumentando o grau do polinômio interpolador. Por exemplo, se cada elemento finito e for interpolado por splines cúbicos, a solução aproximada $u_h(x)$ será um polinômio cúbico por partes e a solução numérica pode ser uma melhor aproximação da solução exata $u = u(x)$, se comparada com a obtida por interpolação linear. Nesste caso, os cálculos da matriz local K_{ij}^e e F_i^e são mais trabalhosos, havendo então a necessidade de introduzir um método de integração numérica, como por exemplo, o método da quadratura Gaussiana, também conhecida como Método de Gauss-Legendre, com dois pontos interiores ξ_1 e ξ_2, cujo valor é exato para polinômios de grau $\nu \leq 3$. Para maiores detalhes sobre quadratura Gaussiana, veja Burden e Faires [3].

A quadratura Gaussiana no caso unidimensional é dada por

$$\int_{-1}^{1} g(\xi)\, d\xi = \sum_{i=1}^{N} g(\xi_i) w_i,$$

onde N é o número de pontos de integração, ξ_i é a coordenada e w_i é o peso associado a ξ_i. Quando $N = 2$, $\xi_1 = \dfrac{-\sqrt{3}}{3} = -\xi_2$ e $w_1 = w_2 = 1$. Nestas condições o erro da integração é dado por

$$E_G = \frac{1}{135} \frac{d^4 g(\xi)}{d\xi^4}$$

e a integral é calculada por

$$\int_{-1}^{1} g(\xi)\, d\xi = g\left(\frac{-\sqrt{3}}{3}\right) + g\left(\frac{\sqrt{3}}{3}\right). \tag{2.63}$$

Sendo o intervalo de integração da função g o intervalo fechado $[-1, 1]$, para calcular a matriz local e a força local do elemento e cujas coordenadas são dadas por $[x_1^e, x_2^e]$, é necessário fazer a seguinte transformação isoparamétrica:

$$\xi : [x_1^e, x_2^e] \to [-1, 1],$$

$$x \longmapsto \xi(x) = \frac{2x - x_1^e - x_2^e}{h_e},$$

onde $h_e = x_2^e - x_1^e$. A função inversa ξ^{-1} de ξ é dada por

$$x^e : [-1, 1] \to [x_1^e, x_2^e],$$

$$\xi \longmapsto x^e(\xi) = \frac{1}{2}(x_1^e + x_2^e + h_e\xi). \tag{2.64}$$

Como $\dfrac{dx^e}{d\xi} = \dfrac{h_e}{2}$, podemos, então, calcular a matriz local e a força local. Com efeito, sabemos de (2.39) que

$$K_{ab}^e = \int_{x_1^e}^{x_2^e} \left(\alpha \frac{d\varphi_a^e}{dx}\frac{d\varphi_b^e}{dx} + \beta\varphi_a^e\varphi_b^e \right) dx.$$

Pela Regra da Cadeia,

$$\frac{d\varphi_a^e}{dx} = \frac{d\varphi_a^e}{d\xi}\frac{d\xi}{dx} = \frac{d\varphi_a^e}{d\xi}\frac{2}{h_e},$$

de modo que

$$K_{ab}^e = \int_{-1}^{1} \left(\frac{4\alpha}{h_e^2}\frac{d\varphi_a^e}{d\xi}\frac{d\varphi_b^e}{d\xi} + \beta\varphi_a\varphi_b \right) \frac{h_e}{2} d\xi = \int_{-1}^{1} g(\xi)\, d\xi = g\left(\frac{-\sqrt{3}}{3}\right) + g\left(\frac{\sqrt{3}}{3}\right), \tag{2.65}$$

onde,

$$g(\xi) = \left(\frac{4\alpha}{h_e^2}\frac{d\varphi_a^e}{d\xi}\frac{d\varphi_b^e}{d\xi} + \beta\varphi_a^e\varphi_a^e \right) \frac{h_e}{2}\,.$$

Para explicitar a função g, precisamos definir a função de interpolação $\varphi_i(\xi)$ no intervalo $[-1, 1]$. Por exemplo, usando polinômio de grau um, temos

$$\varphi_a^e(\xi) = \begin{cases} \varphi_1^e(\xi) &= \frac{1}{2}(1-\xi), & \forall \xi \in [-1,1], \\ \varphi_2^e(\xi) &= \frac{1}{2}(1+\xi), & \forall \xi \in [-1,1], \\ 0\,, & & \forall \xi \notin [-1,1]. \end{cases} \tag{2.66}$$

A função $\varphi_a^e(\xi)$ é equivalente a função $\varphi_a^e(x)$ definida em (2.38) para o intervalo $[x_1^e, x_2^e]$. Utilizando a função (2.66) em (2.65), obtém-se a matriz local K^e dada por (2.52).

Para problemas unidimensionais que utilizam funções de interpolação linear, a transformação isoparamétrica não é vantajosa, tendo em vista a facilidade dos cálculos dos coeficientes da matriz local K^e. Se a função $\varphi_a^e(\xi)$ for um spline cúbico, os cálculos se tornam mais trabalhosos e sua aplicação, juntamente com a quadratura Gaussiana, facilita os cálculos de K^e. A aplicação da transformação isoparamétrica é muito útil nos problemas de dimensão 2 ou 3.

De forma análoga, podemos calcular F_e definida em (2.48) diretamente usando a transformação isoparamétrica e a quadratura Gaussiana. Com efeito,

$$\begin{aligned} F_1^e &= \int_{x_1^e}^{x_2^e} f(x)\varphi_1^e(x)\,dx = \int_{-1}^{1} f\left(\frac{1}{2}(x_1^e + x_2^e + h_e\xi)\right))\varphi_1^e(\xi)\frac{h_e}{2}\,d\xi = \int_{-1}^{1} g_1(\xi)\,d\xi \\ &= g_1\left(\frac{-\sqrt{3}}{3}\right) + g_1\left(\frac{\sqrt{3}}{3}\right), \end{aligned} \tag{2.67}$$

onde usamos a aplicação $x^e(\xi)$ definida em (2.64) e a função $g_1(\xi)$ dada por

$$g_1(\xi) = f\left(\frac{x_1^e + x_2^e + h_e\xi}{2}\right)\varphi_1^e(\xi)\frac{h_e}{2}, \tag{2.68}$$

sendo $\varphi_1^e(\xi)$ a função de interpolação do intervalo $[-1, 1]$.

Se tomarmos, em particular, a função de interpolação linear (2.66), obtemos

$$g_1(\xi) = \frac{h_e}{4} f\left(\frac{x_1^e + x_2^e + h_e\xi}{2}\right)(1 - \xi).$$

Analogamente, para F_2^e,

$$F_2^e = \int_{-1}^{1} g_2(\xi)\,d\xi = g_2\left(\frac{-\sqrt{3}}{3}\right) + g_2\left(\frac{\sqrt{3}}{3}\right), \tag{2.69}$$

onde

$$g_2(\xi) = f\left(\frac{x_1^e + x_2^e + h_e\xi}{2}\right)\varphi_2^e(\xi)\frac{h_e}{2}, \tag{2.70}$$

ou, se tomarmos (2.66),

$$g_2(\xi) = \frac{h_e}{4} f\left(\frac{x_1^e + x_2^e + h_e\xi}{2}\right)(1 + \xi).$$

Usando o algoritmo (2.51), obtém-se a força global F.

Para facilitar o cálculo de $g_1(\xi)$ e $g_2(\xi)$, podemos interpolar f utilizando as funções de interpolação $\varphi_a(\xi)$, $a = 1, 2$. De fato,

$$f(x^e(\xi)) = f(\xi) = \sum_{a=1}^{2} f_a^e\ \varphi_a(\xi), \tag{2.71}$$

onde $f_a^e = f(\xi_a)$. Logo,

$$F_a^e = \int_{x_1^e}^{x_2^e} f(x)\varphi_a^e(x)\,dx = \int_{-1}^{1} f\ (\xi)\frac{\partial x^e}{\partial \xi}\varphi_a(x^e(\xi))\,d\xi = \int_{-1}^{1}\left(\sum_{b=1}^{2}\varphi_a(\xi)\varphi_b(\xi)\frac{\partial x^e}{\partial \xi}f_b^e\right)d\xi. \tag{2.72}$$

Em particular, para a interpolação linear, temos

$$\varphi_1(\xi) = \frac{1}{2}(1-\xi) \qquad \text{e} \qquad \varphi_2(\xi) = \frac{1}{2}(1+\xi).$$

Como $\dfrac{dx^e}{d\xi} = \dfrac{h_e}{2}$, segue que

$$F_a^e = \frac{h_e}{2}\left(\int_{-1}^{1} \varphi_a(\xi)\varphi_1(\xi)f_1^e + \varphi_a(\xi)\varphi_2(\xi)f_2^e\right) d\xi. \tag{2.73}$$

Fazendo os cálculos, obtém-se

$$F^e = \frac{h_e}{6}\begin{bmatrix} 2f_1^e + f_2^e \\ f_1^e + 2f_2^e \end{bmatrix},$$

que é a mesma força calculada em (2.58) e (2.59). A diferença consiste no uso da transformação isoparamétrica.

2.7 Condições de Fronteira

O problema que tratamos até aqui envolve condições de fronteira do tipo Dirichlet homogêneas, o que possibilitou considerar a matriz global K e o vetor força F, respectivamente de ordem $(m+1)\times(m+1)$ e $(m+1)\times 1$, obtidos através da matriz local K^e e da força local F^e. Vamos agora considerar quatro exemplos com tipos diferentes de condições de fronteira e analisar sua influência na matriz K e no vetor F.

Em todos os exemplos a seguir, subdividimos o intervalo $[0,1]$ em m partes, obtendo $(m+1)$ nós, onde $x_1 = 0$, $x_{m+1} = 1$ e $h_i = x_{i+1} - x_i$. Para cada nó x_i, definimos a função base φ_i como em (2.17) e, para cada sub-intervalo $[x_i, x_{i+1}] = [x_e, x_{e+1}] = [x_1^e, x_2^e]$, $e = 1, 2, \cdots, m$, definimos um elemento finito e e a função de interpolação local $\varphi_a^e(x)$ como em (2.38).

• **Exemplo 1.** (Condição não homogêneas do tipo Dirichlet) Neste caso, o problema é o seguinte:

$$\begin{cases} -\alpha u'' + \beta u = f, \\ u(0) = p, \quad u(1) = q. \end{cases}$$

Para este caso a solução u é conhecida nos extermos em $x_1 = 0$ e $x_{m+1} = 1$ e a discretização fornece $m-1$ incógnitas (nós) para os quais queremos determinar a solução aproximada. Lembrando que

$$u_h(x) = \sum_{i=1}^{m+1} C_i\varphi_i(x),$$

temos, para $x = x_1 = 0$, $p = u_h(x_1) = C_1\varphi_1(x_1) = C_1 0$. Logo, $C_1 = p$. Analogamente, para $x = x_{m+1} = 1$, obtemos $C_{m+1} = q$. Portanto, no sistema linear $KC = F$ de ordem $m+1$, as constantes C_1 e C_{m+1} são conhecidas e assim temos $m-1$ incógnitas, a saber, $C_2, \ldots, C_m$.

Como a matriz global K tem ordem $(m+1)\times(m+1)$ e o vetor força F tem $(m+1)$ componentes, devemos considerar o sistema linear de tal forma a assegurar que $C_1 = p$ e $C_{m+1} = q$. De fato, a matriz global K é dada por (2.53) e o vetor força F é dado por (2.62). Assim, a primeira linha do sistema

$$\begin{bmatrix} K_{11} & K_{12} & \cdots & 0 & 0 \\ K_{21} & K_{22} & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & K_{mm} & K_{m,m+1} \\ 0 & 0 & 0 & K_{m+1,m} & K_{m+1,m+1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \\ C_{m+1} \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \\ F_{m+1} \end{bmatrix}$$

é $K_{11}C_1 + K_{12}C_2 = F_1$, de modo que se $K_{11} = 1$, $K_{12} = 0$ e $F_1 = p$, obtemos $C_1 = p$, que é a condição imposta.

Para a segunda linha com $F_1 = p$, temos

$$K_{21}C_1 + K_{22}C_2 + K_{23}C_3 = F_2 \iff K_{22}C_2 + K_{23}C_3 = F_2 - K_{21}p := \widetilde{F}_2.$$

De forma análoga, para a última linha, é suficiente escolhemos $K_{m+1,m+1} = 1$, $K_{m+1,m} = 0$ e $F_{m+1} = q$, de modo a obter $C_{m+1} = q$.

Para a penúltima linha temos

$$K_{m,m-1}C_{m-1} + K_{m,m}C_m = F_m - qK_{m,m+1} := \widetilde{F}_m.$$

Portanto, com as definições acima, o sistema linear $KC = F$ de ordem $(m+1)\times(m+1)$ que reproduz as condições de fronteira impostas no problema toma a seguinte forma:

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & K_{22} & K_{23} & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & K_{m,m-1} & K_{m,m} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ \vdots \\ \vdots \\ C_m \\ C_{m+1} \end{bmatrix} = \begin{bmatrix} p \\ \widetilde{F}_2 \\ F_3 \\ \vdots \\ \vdots \\ \widetilde{F}_m \\ q \end{bmatrix} = \begin{bmatrix} p \\ F_2 - pK_{21} \\ F_3 \\ \vdots \\ \vdots \\ F_m - qK_{m,m+1} \\ q \end{bmatrix} \tag{2.74}$$

O sistema linear pode, então, ser resolvido pelo algoritmo dado anteriormente, cuja solução é

$$C = (C_1, C_2, \ldots, C_m, C_{m+1}) = (p, C_2, \ldots, C_m, q).$$

Assim, a solução aproximada nos nós x_i é dada por

$$u_h(x_i) = C_i\,, \qquad i = 1, 2, \ldots, m, m+1$$

Em particular se $p = q = 0$, o vetor força é exatamente o mesmo dado em (2.33). Além disso, se excluirmos a primeira e a última filas de K, obteremos uma submatriz que é exatamente a matriz dada por (2.24).

Uma forma mais geral de obter diretamente o sistema linear que reproduz as condições de fronteira do problema 1 pode ser feita "transferindo-se" adequadamente as condições de contorno ao termo fonte f. De fato, tendo em vista a linearidade da equação diferencial, podemos obter a solução do problemas com as condições de fronteira não homogêneas a partir do problema com condições homogêneas, somando-se ao termo de fonte f uma função adequada.

No caso específico que estamos tratando, o problema variacional é

$$a(u, v) = (f, v),\ \forall v \in H^1(0,1), \quad u(0) = p,\ u(1) = q. \tag{2.75}$$

Consideremos funções $u_p, u_q \in H^1(0,1)$ satifazendo as condições

$$u_p(0) = p,\ u_p(1) = 0 \quad \text{e} \quad u_q(0) = 0,\ u_q(1) = q$$

e o problema variacional homogêneo

$$a(w, v) = (f, v) - a(u_p, v) - a(u_q, v), \quad \forall v \in H_0^1(0,1). \tag{2.76}$$

Uma vez resolvido o problema (2.76), temos a solução de (2.75) definindo-se $u = w + u_p + u_q$.

O problema discreto associado ao problema (2.76) toma a forma

$$a(w_h, v) = (f, v) - a(u_{h,p}, v) - a(u_{h,q}, v), \quad \forall v \in V_m. \tag{2.77}$$

Tomando $w_h(x) = \sum_{j=2}^{m} C_j \varphi_j(x)$ e $v = \varphi_i$ e substituindo na igualdade acima, temos:

$$\sum_{j=2}^{m} C_j a(\varphi_j, \varphi_i) = (f, \varphi_i) - a(u_{h,p}, \varphi_i) - a(u_{h,q}, \varphi_i)$$

para $i = 2, \cdots, m$. Usando como interpoladores as funções base $\varphi_j(x)$, podemos interpolar f, $u_{h,p}$ e $u_{h,q}$ da seguinte forma:

$$\begin{aligned} f(x) &= \sum_{j=1}^{m+1} \varphi_j(x) f_j, \quad \forall x \in \Omega, \\ u_{h,p}(x) &= p\varphi_1(x), \quad \forall x \in \partial\Omega, \\ u_{h,q}(x) &= q\varphi_{m+1}(x), \quad \forall x \in \partial\Omega. \end{aligned}$$

Assim, definindo

$$K_{ij} = a(\varphi_i, \varphi_j) \tag{2.78}$$

e

$$\begin{aligned} F_i &= (f, \varphi_i) - u_{h,p}K_{1i} - u_{h,q}K_{m+1,i} \\ &= \sum_{j=1}^{m+1} f_j \int_{x_{i-1}}^{x_{i+1}} (\varphi_j \varphi_i)\, dx - u_{h,p}K_{1i} - u_{h,q}K_{m+1,i}, \end{aligned} \tag{2.79}$$

obtemos o sistema linear $KC = F$ de ordem $(m-1) \times (m-1)$.

Utilizando a matriz local e a força local temos

$$K = \sum_{e=1}^{m} K^e \quad \text{e} \quad F = \sum_{e=1}^{m} F^e,$$

onde

$$K_{ab}^e = \int_{x_1^e}^{x_2^e} \left(\alpha \frac{d\varphi_a}{dx}\frac{d\varphi_b}{dx} + \beta \varphi_a \varphi_b \right) dx, \qquad 1 \le a, b \le 2$$

$$F_a^e = \sum_{b=1}^{2} f_b^e \int_{x_1^e}^{x_2^e} \varphi_a^e(x)\varphi_b^e(x)\, dx - pK_{1a}^e - qK_{m+1\ a}^e = f_a^e - pK_{1a}^e - qK_{m+1\ a}^e \tag{2.80}$$

Observe que a matriz global K é a mesma matriz dada por (2.53). Calculemos agora a força F, observando as igualdades (2.48), (2.49), (2.50), o algoritmo (2.51) e usando o fato da matriz K ser tridiagonal,

$$\begin{aligned} F_2 &= f_2^1 + f_1^2 - pK_{12}, \\ F_i &= f_i^e, \quad i = 3, 4, \ldots, m-1, \\ F_m &= f_2^{m-1} + f_1^m - qK_{m+1\ m}. \end{aligned} \tag{2.81}$$

Dessa forma, tem-se

$$\begin{bmatrix} K_{22} & K_{23} & \cdots & 0 & 0 \\ K_{32} & K_{33} & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & K_{m-1\ m-1} & K_{m-1\ m} \\ 0 & 0 & 0 & K_{m\ m-1} & K_{m\ m} \end{bmatrix} \begin{bmatrix} C_2 \\ C_3 \\ \vdots \\ C_{m-1} \\ C_m \end{bmatrix} = \begin{bmatrix} F_2 \\ F_3 \\ \vdots \\ F_{m-1} \\ F_m \end{bmatrix}. \tag{2.82}$$

Resolvendo este sistema linear, obtemos a solução $C = (C_1, \ldots, C_{m+1})$, onde pela condição de fronteira $C_1 = p$ e $C_{m+1} = q$.

O sistema (2.82) é equivalente ao sistema (2.74) quando impomos linhas e colunas extras relativas à condição de $C_1 = p$ e $C_{m+1} = q$.

O procedimento acima será usado no próximo capítulo para o problema bidimensional para valores de fronteira.

• **Exemplo 2.** (Condição não homogêneas do tipo Neumann) Neste caso, consideraremos o seguite problema:

$$\begin{cases} -\alpha u'' + \beta u = f, \\ u'(0) = p, \quad u'(1) = q. \end{cases}$$

Neste exemplo, a derivada da solução é conhecida nos nós $x_1 = 0$ e $x_{m+1} = 1$, ou seja,

$$\frac{du_h}{dx}(x_1) = p \quad \text{e} \quad \frac{du_h}{dx}(x_{m+1}) = q.$$

Assim, a solução $u_h = u_h(x_i)$ é desconhecida nos $(m+1)$ nós x_i, $i = 1, \ldots, m+1$.

Para este tipo de fronteira, uma nova formulação variacional precisa ser feita. Multiplicando a equação diferencial por v e integrando em $(0,1)$, temos

$$\int_0^1 -\alpha u'' v \, dx + \int_0^1 \beta uv \, dx = \int_0^1 fv \, dx \,, \quad \forall v \in H^1(\Omega).$$

Integrando por partes a primeira integral, tem-se

$$\begin{aligned} \int_0^1 -\alpha u'' v \, dx &= -\alpha \left(\frac{du}{dx}(1)v(1) - \frac{du}{dx}(0)v(0) \right) + \alpha \int_0^1 \frac{du}{dx}\frac{dv}{dx} \, dx \\ &= -\alpha q v(1) + \alpha p v(0) + \alpha \int_0^1 \frac{du}{dx}\frac{dv}{dx} \, dx \end{aligned}$$

Assim,

$$\alpha \int_0^1 \frac{du}{dx}\frac{dv}{dx} \, dx + \beta \int_0^1 uv \, dx = \int_0^1 fv \, dx \, + \alpha(qv(1) - pv(0)).$$

Definindo

$$a(u,v) = \int_0^1 \left(\alpha \frac{du}{dx}\frac{dv}{dx} \, + \beta uv \right) dx \tag{2.83}$$

e

$$(f,v) = \int_0^1 f(x)v(x) \, dx \tag{2.84}$$

o problema variacional é o de se determinar $u \in H^1(0,1)$ solução de

$$a(u,v) = (f,v) + \alpha q v(1) - \alpha p v(0) \,, \quad \forall v \in H^1(\Omega) \tag{2.85}$$

e o problema variacional aproximado é o de se determinar $u_h \in V_m$, tal que

$$a(u_h, v) = (f,v) + \alpha q v(1) - \alpha p v(0),$$

onde $V_m = [\varphi_1, \ldots, \varphi_m]$ é um subespaço de $H^1(\Omega)$ gerado por m vetores linearmente independentes.

Assim, para $u_h \in V_m$,

$$u_h(x) = \sum_{j=1}^{m+1} C_j \varphi_j(x).$$

Substituindo em (2.85) e tomando $v = \varphi_i$,

$$\sum_{j=1}^{m+1} C_j a(\varphi_i, \varphi_j) = (f, \varphi_i) + \alpha q \varphi_i(1) - \alpha p \varphi_i(0).$$

Definindo

$$K_{ij} = a(\varphi_i, \varphi_j) \tag{2.86}$$

e

$$F_i = (f, \varphi_i) + \alpha q \varphi_i(1) - \alpha p \varphi_i(0) \tag{2.87}$$

para $i = 1, \ldots, m+1$, obtém-se um sistema linear da forma $KC = F$. A matriz global K é a mesma dada por (2.53). A única mudança no sistema linear é a força F.

- **Cálculo da Força**

Como anteriormente, considere a força local F_a^e, $a = 1, 2$, definida por

$$F_a^e = \int_{x_1^e}^{x_2^e} f(x)\varphi_a^e(x)\,dx - \alpha p \varphi_a^e(0) + \alpha q \varphi_a^e(1) = f_a^e - \alpha p \varphi_a^e(0) + \alpha q \varphi_a^e(1)$$

1. Para $e = 1$, temos $F_1^1 = f_1^1 - \alpha p$ e $F_2^1 = f_2^1$, pois $\varphi_1^1(x_1) = \varphi_1^1(0) = 1$, $\varphi_2^1(x_1) = \varphi_2^1(0) = 0$ e $\varphi_a^1(x_{m+1}) = \varphi_a^1(1) = 0$;
2. Para $e = 2, \ldots, m-1$, temos $F_1^1 = f_1^1 - \alpha p$ e $F_2^1 = f_2^1$, pois $\varphi_a^e(0) = \varphi_a^e(1) = 0$;
3. Para $e = m$, temos $F_1^m = f_1^m$ e $F_2^m = f_2^m + \alpha q$, pois $\varphi_2^m(1) = \varphi_2(x_{m+1}) = \varphi_{m+1}(x_{m+1}) = 1$ e $\varphi_1^m(1) = \varphi_m(x_{m+1}) = 0$.

Do algoritmo (2.51), conclui-se que

$$F_1 = F_1^1, \quad F_e = F_2^{e-1} + F_1^e, \ (e = 2, \ldots, m) \quad \text{e} \quad F_{m+1} = F_2^m.$$

De (2.57) e (2.58), obtém-se, para $e = 2, \ldots, m$,

$$F_1^e = \frac{h_e}{3} f_1^e + \frac{h_e}{6} f_2^e \quad \text{e} \quad F_2^e = \frac{h_e}{6} f_1^e + \frac{h_e}{3} f_2^e;$$

e nos extremos, tem-se F_1^1 e F_2^m calculados por

$$F_1 = F_1^1 = \int_{x_1^1}^{x_2^1} f(x)\varphi_1^1(x)\,dx - \alpha p = \frac{h_1}{6}(2f_1^1 + f_2^1) - \alpha p,$$
$$F_{m+1} = F_2^m = \int_{x_1^m}^{x_2^m} f(x)\varphi_2^m(x)\,dx + \alpha q = \frac{h_m}{6}(f_1^m + 2f_2^m) + \alpha q.$$

Para $e = 2, \ldots, m$,

$$F_e = \frac{h_{e-1}}{6}(f_1^{e-1} + 2f_2^{e-1}) + \frac{h_e}{6}(2f_1^e + f_2^2),$$

onde $f_1^e = f_2^{e-1} = f(x_e)$ e $f_2^e = f_1^{e+1} = f(x_{e+1})$. Temos assim o vetor Força $F = (F_1, \ldots, F_{m+1})$, cujos coeficientes são os calculados acima.

Utilizando a matriz global K dada em (2.53) e o vetor força dado acima, obtém-se, então, o sistema linear de ordem $m+1$, cuja solução $C = (C_1, \ldots C_{m+1})$ é a solução aproximada $u_h(x_i)$ nos nós x_i do problema 2.

Comparando os vetores força dos exemplos 1 e 2, nota-se que os únicos componentes diferentes são F_1, F_2, F_m e F_{m+1}.

• **Exemplo 3.** (Condição não homogêneas do tipo mista) Consideraremos o problema

$$\begin{cases} -\alpha u'' + \beta u = f, \\ u'(0) = p, \quad u(1) = q. \end{cases}$$

Neste caso a derivada da solução é conhecida no ponto $x_1 = 0$ e a solução é conhecida no ponto $x_{m+1} = 1$. Assim, as condições de fronteira deste exemplo são mistas e podem ser resolvidas usando os exemplos 1 e 2. Agora, temos m incógnitas, $C_1 \ldots C_m$, onde $C_i = u_h(x_i)$, $i = 1, \ldots, m$.

Utilizando os procedimentos anteriores podemos definir a matriz K_{ij} e o vetor força F_i respectivamente por

$$K_{ij} = a(\varphi_i, \varphi_j) = \int_0^1 \left(\alpha \frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx} + \beta\varphi_i\varphi_j\right) dx,$$
$$F_i = \int_0^1 f\varphi_i\,dx - \alpha p\varphi_i(0).$$

Para $1 \le i, j \le m$, a matriz K_{ij} e o vetor F_j são os mesmos do exemplo 2.

Para garantir que a solução $u_h(x)$ satisfaça a condição de fronteira $u_h(x_{m+1}) = q$,

façamos como no exemplo 1,

$$\begin{aligned} &K_{m+1,m+1} = 1 \\ &K_{m+1,m} = 0, \\ &F_m = F_m - qK_{m,m+1}, \\ &F_{m+1} = q. \end{aligned}$$

Dessa forma, obtemos

$$\begin{bmatrix} K_{11} & K_{12} & \cdots & 0 & 0 \\ K_{21} & K_{22} & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & K_{mm} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \\ C_{m+1} \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m - qK_{m,m+1} \\ q \end{bmatrix} \tag{2.88}$$

onde $K_{m,m+1} = -\dfrac{\alpha}{h_m} + \dfrac{\beta h_m}{6}$ é dado por (2.52) e F_m é dado por (2.59).

• **Exemplo 4.** (Condição não homogêneas do tipo mista) Consideraremos agora o seguite problema:

$$\begin{cases} -\alpha u'' + \beta u = f, \\ u(0) = p, \quad u'(1) = q. \end{cases}$$

como no caso anterior, p e q são números reais dados.

Assim, temos $u(x_1) = u(0) = p$ e $\dfrac{du}{dx}(x_{m+1}) = \dfrac{du}{dx}(1) = q$. O procedimento é análogo ao anterior, resultando num sistema linear da forma

$$\begin{bmatrix} K_{11} & K_{12} & \cdots & 0 & 0 \\ K_{21} & K_{22} & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & K_{mm} & K_{m,m+1} \\ 0 & 0 & \cdots & K_{m+1,m} & K_{m+1,m+1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \\ C_{m+1} \end{bmatrix} = \begin{bmatrix} p \\ F_2 - pK_{12} \\ \vdots \\ F_m \\ F_{m+1} \end{bmatrix} \tag{2.89}$$

onde todos os elementos K_{ij}, com $2 \leq i, j \leq m+1$ são iguais aos do exemplo 2, bem como o vetor F_j, $j = 2, \ldots, m+1$. Além disso, por (2.52), $K_{12} = \dfrac{-\alpha}{h_1} + \dfrac{\beta h_1}{6}$.

2.8 Programa Computacional

Descrevemos abaixo as subrotinas em linguagem C do programa computacional para o cálculo da solução aproximada dos problemas a seguir.

1. `InputData`

   Os dados de entrada são os números reais positivos α e β e o número de elementos $\texttt{Nel} = m$. O número de nós é dado por $\texttt{Nos} = \texttt{Nel} + 1 = m + 1$.

2. `InputCondFront`

   As condições de fronteira nos nós extremos do intervalo considerado são classificados por tipos:

$$\begin{cases} \text{tipo 1} & : \text{ Dirichlet;} \\ \text{tipo 2} & : \text{ Neumann.} \end{cases}$$

   Quando o nó x_1 é do tipo 1, significa que a solução é conhecida neste nó (condição de Dirichlet), ou seja,

$$u(x_1) = p\,, \qquad p \in \mathbb{R}.$$

   Se o nó x_1 é do tipo 2, é a derivada da solução u que é conhecida (condição de Newmann), ou seja,

$$\frac{du}{dx}(x_1) = p\,, \qquad p \in \mathbb{R}.$$

   O tipo de fronteira, como mostrado nos exemplos, tem fundamental importância na montagem do sistema linear.

3. `CoordGlobal`

   Considere o intervalo $[a, b]$ e a discretização uniforme ou não do intervalo. Os pontos discretos $X[i]$, $i = 1, 2, \cdots, \texttt{Nos}$, do intervalo são as coordenadas globais. Para a malha uniforme é suficiente introduzir os extremos do intervalo e o número de elementos (`Nel`) da discretização, pois os nós globais são calculados automaticamente pelo programa. Para a malha não uniforme, todos os nós globais deverão ser introduzidos um a um.

4. `CoordLocal` (x_1^e, x_2^e)

   Para o elemento $e = 1, 2, \ldots, m$, definem-se as coordenadas locais (x_1^e, x_2^e), para o cálculo da matriz local K^e, da seguinte forma:

$$(x_1^e, x_2^e) = (x_e, x_{e+1}) \quad \text{ e } \quad h_e = x_{e+1} - x_e.$$

   Se a discretização é uniforme, $h_e = h$.

5. `NoLG`(e, a)

   A subrotina `NoLG`(e, a) identifica os nós locais $a = 1, 2$ do elemento e com os respectivos nós globais A da discretização:

$$A = \texttt{NoLG}(e, a) = \begin{cases} e, & \text{se } a = 1; \\ e + 1, & \text{se } a = 2. \end{cases}$$

   Com isso, se estabelece uma relação entre os nós globais A e os nós locais x_1^e e x_2^e. Por exemplo:

$a \backslash e$	1	2	3	$\cdots$	m
1	1	2	3	$\cdots$	m
2	2	3	4	$\cdots$	$m+1$

   Assim, o elemento e que tem coordenadas locais (x_1^e, x_2^e), pertence ao intervalo $[x_e, x_{e+1}]$, onde x_e e x_{e+1} são nós globais.

6. Transformada isoparamétrica `Xi`(e, ξ)

   Foram dadas duas opções para o cálculo da força F. Na primeira, o cálculo da integral é feita diretamente e na segunda, usa-se a quadratura Gaussiana para o qual é necessário a transformação isoparamétrica, que é a opção utilizada neste programa. Então, definimos em (2.64) a função

$$x^e : [-1, 1] \longrightarrow [x_e, x_{e+1}],$$

   onde

$$x^e(\xi) = \frac{1}{2}(x_1^e + x_2^e + h_e\xi) = x_1^e + \frac{h_e}{2}(\xi + 1),$$

   e a função `Xi`(e, ξ) é a aplicação $x^e(\xi)$.

7. `MatrizRigidez`

   Para cada elemento e, a matriz local K^e é dada por

$$K^e = \begin{bmatrix} K_{11}^e & K_{12}^e \\ K_{21}^e & K_{22}^e \end{bmatrix},$$

   onde os coeficientes da matriz estão definidos em (2.52). Visto que `NoLG`$(e, 1) = e$ e `NoLG`$(e, 2) = e + 1$, usamos o seguinte para estabalecer a matriz local referente a cada elemento e:

$$\begin{bmatrix} K_{11} = K_{11}^1 \\ K_{e,e} = K_{22}^{e-1} + K_{11}^e \\ K_{e,e+1} = K_{12}^e\,, \quad e = 2, 3, \ldots, m \\ K_{m+1,m+1} = K_{22}^m \end{bmatrix}$$

Este procedimento está feito no final da subrotina `MatrizRigidez` com uma modificação no armazenamento da matriz. Como a matriz rigidez (global) é tridiagonal, então, ao invés de armazenar todos os elementos, são armazenados somente os elementos das diagonais reordenadas em forma de 3 colunas definidas pelos elementos $K_{e,e-1}$, $K_{e,e}$ e $K_{e,e+1}$ que são armazenados como 1^0, 2^0 e 3^0 colunas, respectivamente, ou seja,

$$\left[\begin{array}{l} K_{e+1,1} = K_{e,e-1} \\ K_{e,2} = K_{e,e} \\ K_{e,3} = K_{e,e+1} \end{array}\right.$$

8. `VetorForca`

Para calcular a força local $f(x)$ utilizando a quadratura Gaussiana, definimos as funções `g1` e `g2` como em (2.68) e (2.70), ou seja,

$$g_1(\xi) = f\left(\frac{1}{2}(x_1^e + \frac{h_e}{2}(\xi+1))\right)\varphi_1^e(\xi)\frac{h_e}{2},$$

$$g_2(\xi) = f\left(\frac{1}{2}(x_1^e + \frac{h_e}{2}(\xi+1))\right)\varphi_2^e(\xi)\frac{h_e}{2}.$$

Logo,

$$F_1^e = \int_{-1}^{1} g_1(\xi)\, d\xi = g_1\left(-\frac{\sqrt{3}}{3}\right) + g_1\left(\frac{\sqrt{3}}{3}\right),$$

$$F_2^e = \int_{-1}^{1} g_2(\xi)\, d\xi = g_2\left(-\frac{\sqrt{3}}{3}\right) + g_2\left(\frac{\sqrt{3}}{3}\right),$$

são as forças locais. Para calcular o vetor força F, usa-se o algoritmo:

$$\begin{array}{l} F_1 = F_1^1, \\ F_e = F_2^{e-1} + F_1^e\,, \quad e = 2, 3, \ldots, m, \\ F_{m+1} = F_2^m. \end{array}$$

Note que $\texttt{NoLG}(e,1) = e$ e $\texttt{NoLG}(e,2) = e+1$. Assim temos

$$\begin{array}{rcl} F_e & \longleftarrow & F_e + F_1^e \\ F_{e+1} & \longleftarrow & F_2^e \end{array}$$

definidos no final da subrotina `VetorForca`.

9. `CondFront`(p,q)

Nos exemplos 1, 2, 3 e 4 estudamos as mudanças no sistema linear em função das condições de fronteira. Nesta subrotina é possível resolver qualquer um dos problemas modelos dados anteriormente.

(a) Se o nó x_1 é do tipo 1, então, $u(x_1) = p$ é conhecido. Neste caso, temos na ordem:

$$K_{11} = 1\,; \quad K_{12} = 0\,; \quad F_1 = p\,; \quad F_2 \leftarrow F_2 - K_{21}p \quad \text{e} \quad K_{21} = 0.$$

(b) Se o nó x_{m+1} é do tipo 1, então, $u(x_{m+1}) = q$. Neste caso:

$$K_{m+1,m+1} = 1;\ K_{m+1,m} = 0;\ F_{m+1} = q;\ F_m \leftarrow F_m - K_{m,m+1}q;\ K_{m,m+1} = 0.$$

Observe que os coeficientes K_{21} e $K_{m,m+1}$ são anulados somente depois do cálculo de F_2 e F_m.

(c) Se o nó x_1 é do tipo 2, então $u'(x_1) = p$. Assim,

$$F_1 \longleftarrow F_1 - \alpha p.$$

(d) Se o nó x_{m+1} é do tipo 2, então, $u'(x_{m+1}) = q$. Logo,

$$F_{m+1} \longleftarrow F_{m+1} + \alpha q.$$

A combinação dos quatro tipos são os problemas 1, 2, 3 e 4.

10. `LinearSolver`

Para qualquer um dos dos tipos de fronteira considerados, o sistema linear $KC = F$ é de ordem $m+1$. Para resolver o sistema linear, basta usar o algoritmo dado em (2.36) e (2.37), adaptando os coeficientes da matriz para a forma coluna dada acima.

11. `Norma`

Para o cálculo de erro na norma $L^2(\Omega)$ e na norma $H^1(\Omega)$, o programa utiliza as fórmulas definidas em 2.94 e 2.95.

12. `OutputData`

Imprime a matriz rigidez e o vetor força.

13. `SolExata`

As soluções exatas (2.90), (2.91), (2.92) e (2.93) dos problemas 1, 2, 3, 4 são definidas nesta subrotina.

14. `OutputResultado`

Imprime a solução aproximada nos nós x_i.

2.9 Exemplos

Nesta seção apresentamos exemplos para o problema modelo com diferentes tipos de fronteira. As estimativas de erro de soluções numéricas em geral serão dadas no próximo capítulo. Porém, se conhecemos a solução exata do problema, podemos calcular diretamente os errors nas normas de $H^1(0,1)$ e $L^2(0,1)$ entre a solução numérica e exata.

Exemplos com solução exata

Para obter uma solução exata, consideremos um caso particular com $f(x) = x$. Nestas condições, a solução exata dos Exemplos 1, 2, 3 e 4, da Seção 2.7, são dadas, respectivamente por,

$$u_1(x) = \frac{x}{\beta} + \frac{1}{e^{\eta} - e^{-\eta}} \left\{ (q - pe^{-\eta} - \frac{1}{\beta})e^{\eta x} + (pe^{\eta} - q + \frac{1}{\beta})e^{-\eta x} \right\}, \quad (2.90)$$

$$u_2(x) = \frac{x}{\beta} + (p - \frac{1}{\beta})\frac{e^{\eta x}}{\eta} + \left\{ (q - \frac{1}{\beta}) + (\frac{1}{\beta} - p)e^{\eta} \right\} \frac{e^{\eta x} + e^{-\eta x}}{\eta(e^{\eta} - e^{-\eta})}, \quad (2.91)$$

$$u_3(x) = \frac{x}{\beta} + (p - \frac{1}{\beta})\frac{e^{\eta(x-1)} - e^{\eta(1-x)}}{\eta(e^{\eta} + e^{-\eta})} + \eta(q - \frac{1}{\beta})\frac{e^{\eta x} + e^{-\eta x}}{\eta(e^{\eta} + e^{-\eta})}, \quad (2.92)$$

$$u_4(x) = \frac{x}{\beta} + p\eta\frac{e^{\eta(x-1)} + e^{\eta(1-x)}}{\eta(e^{\eta} + e - \eta)} + (q - \frac{1}{\beta})\frac{e^{\eta x} - e^{-\eta x}}{\eta(e^{\eta} + e^{-\eta})}, \quad (2.93)$$

onde $\eta^2 = \beta/\alpha$.

O conhecimento da solução exata possibilitará o cálculo do erro definido por

$$E(x) = u_h(x) - u_{ex}(x).$$

As normas em $L^2(\Omega)$ e $H^1(\Omega)$ do erro E são definidas respectivamente por

$$\|E\|_0 = \Big\{ \int_\Omega |E(x)|^2 dx \Big\}^{1/2} \text{ e } \|E\|_1 = \Big\{ \int_\Omega \Big(|E(x)|^2 + \Big|\frac{dE}{dx}(x)\Big|^2 \Big) dx \Big\}^{1/2}.$$

Para o caso discreto com a malha uniforme $h = h_i = x_{i+1} - x_i$, temos as normas

$$\|E\|_0 = \Big\{ h \sum_{i=1}^{m} E_i^2 \Big\}^{1/2}, \quad (2.94)$$

$$\|E\|_1 = \Big\{ h \sum_{i=1}^{m} E_i^2 \Big\}^{1/2} + \Big\{ \frac{1}{h} \sum_{i=1}^{m} (E_{i+1} - E_i)^2 \Big\}^{1/2}, \quad (2.95)$$

onde denotamos $E_i = E(x_i)$.

As normas discretas acima são simples aplicações do método dos retângulos. Ainda considerando a malha uniforme $h = h_i = x_{i+1} - x_i$, podemos também calcular as normas discretas usando o método dos trapézios. Por exemplo, a normas discretas em $L^2(\Omega)$ e $H^1(\Omega)$ tomam a forma:

$$\begin{aligned} \|E\|_0 &= \Big\{\frac{h}{2}\Big(E_1^2 + 2\sum_{i=2}^{m} E_i^2 + E_{m+1}^2\Big)\Big\}^{1/2}, \\ \|E\|_1 &= \|E\|_0 + \Big\{\frac{1}{2h}\Big((E_2 - E_1)^2 + 2\sum_{i=2}^{m-1}(E_{i+1} - E_i)^2 + (E_{m+1} - E_m)^2\Big)\Big\}^{1/2}. \end{aligned} \tag{2.96}$$

Solução Numérica

Exemplo 1 *Problema de Dirichlet*

$$\begin{cases} -\alpha u'' + \beta u = f, \\ u(0) = p, \quad u(1) = q. \end{cases} \tag{2.97}$$

(**1**.) Considere $\alpha = \beta = 1$, $p = q = 0$ e $f(x) = x$. Nestas condições, a solução exata, representada pela Figura (2.3) é dada por

$$u(x) = x + \frac{1}{e - e^{-1}}(e^{-x} - e^{x}), \quad x \in [0, 1].$$

Usando a malha uniforme com $h = 1/20$, temos 20 elementos e 21 nós. A solução aproximada u_h e a solução exata u nos nós são dadas na Tabela(2.1)

Os erros absolutos E_a e relativos E_r nas normas $L^2(0,1)$ e $H_0^1(0,1)$ obtidos são

$$\begin{aligned} &\|E_a\|_0 = 7.7673 \times 10^{-6}, \quad \|E_a\|_1 = 2.6648 \times 10^{-5}, \\ &\|E_r\|_0 = 0.0186\%, \qquad\quad \|E_r\|_1 = 0.0187\%. \end{aligned}$$

Para efeito de comparação, para mostrar que o erro depende da malha, seguem abaixo os erros absolutos e relativos para 5 e 10 elementos, respectivamente,

$$\begin{cases} \|E_a\|_0 = 1.2064 \times 10^{-4}, \quad \|E_a\|_1 = 4.2306 \times 10^{-4}, \\ \|E_r\|_0 = 0.3045\%, \qquad\quad \|E_r\|_1 = 0.3057\%. \end{cases}$$

$$\begin{cases} \|E_a\|_0 = 3.131 \times 10^{-5}, \quad \|E_a\|_1 = 1.0779 \times 10^{-4}, \\ \|E_r\|_0 = 0.0758\%, \qquad\quad \|E_r\|_1 = 0.0762\%, \end{cases}$$

No	Aproximada	Exata
1	+0.00000000	+0.00000000
2	+0.00743773	+0.00743638
3	+0.01476900	+0.01476628
4	+0.02188711	+0.02188311
5	+0.02868480	+0.02867955
6	+0.03505403	+0.03504759
7	+0.04088567	+0.04087818
8	+0.04606926	+0.04606078
9	+0.05049271	+0.05048336
10	+0.05404203	+0.05403205
11	+0.05660105	+0.05659056
12	+0.05805111	+0.05804033
13	+0.05827080	+0.05825994
14	+0.05713560	+0.05712492
15	+0.05451763	+0.05450734
16	+0.05028528	+0.05027583
17	+0.04430292	+0.04429452
18	+0.03643054	+0.03642360
19	+0.02652339	+0.02651833
20	+0.01443165	+0.01442886
21	+0.00000000	+0.00000000

Tabela 2.1: Solução Nodal

Note que o erro $\|E_a\|_0$ na norma L^2 para $h = 1/20$, $h = 1/10$ e $h = 1/5$ satisfaz a relação $E(h) \to 0$ quando $h \to 0$, como podemos verificar na Figura (2.4), onde estamos negligenciando os erros de arredondamento. Provaremos no próximo capítulo, na Seção Análise de Erro, que o erro na norma L^2 é de ordem $\mathcal{O}(h^2)$, ou seja, é da forma $E(h) = ch^2$, onde a constante c nesse exemplo é $c = 0.03023$. Para a norma H^1, mostraremos que o erro absoluto é de ordem $\mathcal{O}(h)$, embora possa ser mostrado que a ordem de convergência é da forma $\mathcal{O}(h^{(1+\xi)})$, com $0 \leq \xi < 1$.

(**2**.) Considere as mesmas condições anteriores, com $\alpha = 0.001$. Neste caso, os erros absolutos e relativos para 20 elementos são dados por

$$\|E_a\|_0 = 8.367 \times 10^{-3}, \quad \|E_a\|_1 = 2.305 \times 10^{-1},$$

$$\|E_r\|_0 = 1.5838\%, \qquad \|E_r\|_1 = 6.5318\%.$$

Note que o erro é pior nesse exemplo quando comparado com o anterior, mostrando uma dependência do parâmetro α, que como veremos nos próximos capítulos, o erro absoluto é inversamente proporcional a α.

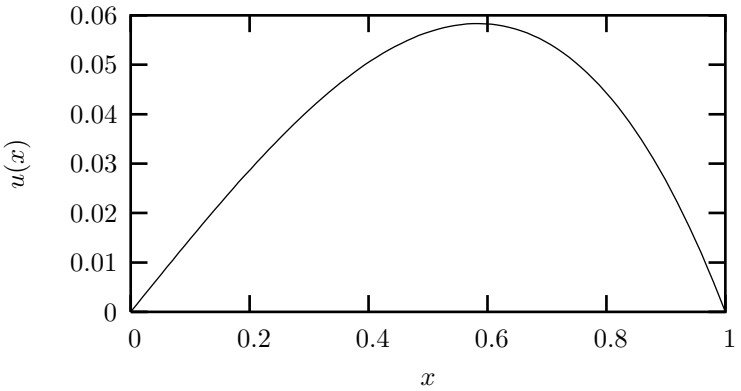


Figura 2.3: Solução exata

(**3**.) Considere $\alpha = \beta = 1$, $p = 0$, $q = 1$ e $f(x) = x$. Neste caso, a solução exata e a solução aproximada são iguais e dadas por $u(x) = u_h(x) = x$. Os erros relativos obtidos para 5 elementos são nulos, i.e., $\|E_r\|_0 = 0$ e $\|E_r\|_1 = 0$, mas devido aos erros de arredondamento, foram encontrados os seguintes erros absolutos e relativos para 20 elementos:

$$\|E_a\|_0 = 1.452 \times 10^{-6}, \quad \|E_r\|_1 = 5.075 \times 10^{-6},$$

$$\|E_r\|_0 = 0.0003\%, \qquad \|E_r\|_1 = 0.0004\%.$$

(**4**.) Considere $\alpha = \beta = 1$, $p = 1$, $q = 0$ e $f(x) = x$. Então a solução exata é dada por

$$u(x) = x + \frac{1}{e - e^{-1}}((e+1)e^{-x} - (e^{-1}+1)e^{x})$$

e os erros absolutos e relativos para 20 elementos são dados por

$$\|E_a\|_0 = 2.577 \times 10^{-6}, \quad \|E_a\|_1 = 1.555 \times 10^{-5},$$

$$\|E_r\|_0 = 0.0005\%, \qquad \|E_r\|_1 = 0.0013\%.$$

Exemplo 2 *Problema de Neumann*

$$\begin{cases} -\alpha u'' + \beta u = f \\ u'(0) = p, \quad u'(1) = q \end{cases} \tag{2.98}$$

(**1**.) Considere $\alpha = \beta = 1$, $p = q = 0$ e $f(x) = x$. Neste caso, a solução exata, representada pela Figura (2.5), é dada por

$$u(x) = x - e^{x} + \frac{1}{e - e^{-1}}((e-1)(e^{-x} + e^{x})).$$

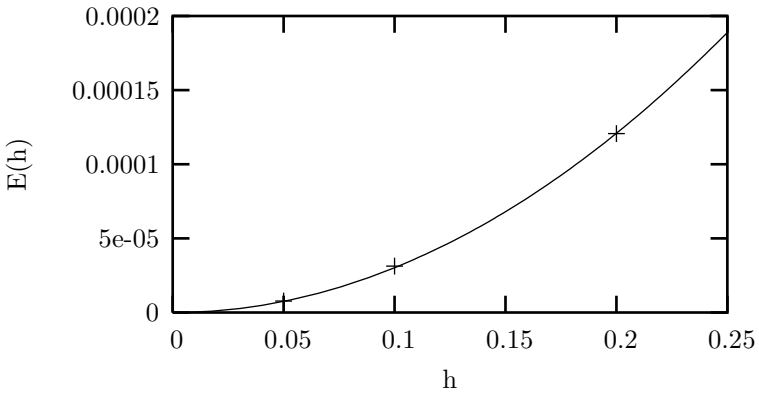


Figura 2.4: Erro absoluto na norma L^2

Usando a malha uniforme com $h = 1/20$, temos 20 elementos e 21 nós. A solução aproximada u_h e a solução exata u nos nós são dadas na Tabela (2.2).

Os erros absolutos e relativos obtidos para 20 elementos são

$$\|E_a\|_0 = 1.4875 \times 10^{-5}, \quad \|E_a\|_1 = 2.025 \times 10^{-5},$$

$$\|E_r\|_0 = 0.0030\%, \qquad \|E_r\|_1 = 0.0040\%.$$

(**2**.) Tomando $\alpha = \beta = 1$, $p = q = 1$ e $f(x) = x$, a solução exata e a solução aproximada são iguais e dadas por $u(x) = u_h(x) = x$. Para 5 elementos obtemos os seguintes erros relativos, $\|E_r\|_0 = 0.0001\%$ e $\|E_r\|_1 = 0.0001\%$.

Novamente, podemos verificar que, devido aos erros de arredondamento, quando aumentamos o número de elementos para 20, o erro relativo também cresce: $\|E_r\|_0 = 0.0016\%$ e $\|E_r\|_1 = 0.0031\%$.

(**3**.) Tomando $\alpha = \beta = 1$, $p = 1$, $q = 0$ e $f(x) = x$, então a solução exata é

$$u(x) = x - \frac{1}{e - e^{-1}}(e^{-x} + e^{x})$$

e os erros absolutos e relativos para 20 elementos são dados por

$$\|E_a\|_0 = 2.2002 \times 10^{-4}, \quad \|E_a\|_1 = 2.2020 \times 10^{-4},$$

$$\|E_r\|_0 = 0.0274\%, \qquad \|E_r\|_1 = 0.0419\%.$$

Exemplo 3 *Problema misto*

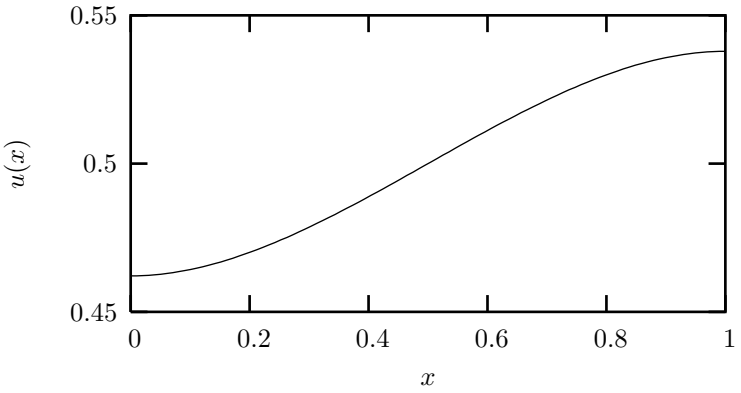


Figura 2.5: Solução exata

(**1**.) Considere as condições de fronteira dadas por $u'(0) = 0,\ u(1) = 1$, com $\alpha = \beta = 1$ e $f(x) = x$. Então a solução exata, representada pela Figura (2.6), é dada por

$$u(x) = x + \frac{1}{e + e^{-1}}(e\, e^{-x} - e^{-1}e^{x}).$$

A solução aproximada u_h e a solução exata u nos nós são dadas na Tabela(2.3):

Os erros absolutos e relativos para 20 elementos obtidos são

$$\|E_a\|_0 = 2.9346 \times 10^{-5}, \quad \|E_a\|_1 = 5.47 \times 10^{-5},$$
$$\|E_r\|_0 = 0.0034\%, \qquad \|E_r\|_1 = 0.0061\%.$$

(**2**.) Considere as condições de fronteira dadas por $u(0) = 1$ e $u'(1) = 0$, com $\alpha = \beta = 1$ e $f(x) = x$. Assim, a solução exata, representada pela Figura (2.7), é dada por

$$u(x) = x + \frac{1}{e + e^{-1}}((e+1)e^{-x} + (e^{-1} - 1)e^{x}).$$

A solução aproximada u_h e a solução exata u nos nós são dadas pela Tabela (2.4)
Os erros relativos obtidos são

$$\|E_a\|_0 = 2.2557 \times 10^{-5}, \quad \|E_a\|_1 = 4.4658 \times 10^{-5},$$
$$\|E_r\|_0 = 0.0025\%, \qquad \|E_r\|_1 = 0.0048\%.$$

No	Aproximada	Exata
1	+0.46209684	+0.46211720
2	+0.46265385	+0.46267408
3	+0.46424291	+0.46426281
4	+0.46674299	+0.46676257
5	+0.47003525	+0.47005433
6	+0.47400284	+0.47402132
7	+0.47853073	+0.47854859
8	+0.48350516	+0.48352227
9	+0.48881352	+0.48882976
10	+0.49434412	+0.49435946
11	+0.49998564	+0.50000000
12	+0.50562716	+0.50564057
13	+0.51115775	+0.51117027
14	+0.51646620	+0.51647776
15	+0.52144074	+0.52145141
16	+0.52596867	+0.52597868
17	+0.52993637	+0.52994567
18	+0.53322870	+0.53323734
19	+0.53572881	+0.53573722
20	+0.53731793	+0.53732580
21	+0.53787494	+0.53788280

Tabela 2.2: Solução Nodal

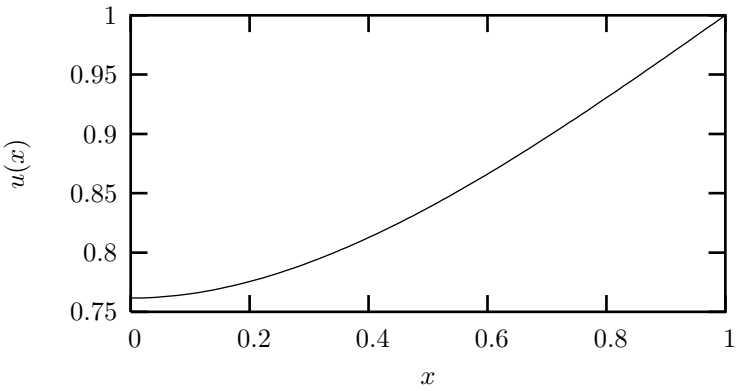


Figura 2.6: Solução exata

No	Aproximada	Exata
1	+0.76155293	+0.76159418
2	+0.76248443	+0.76252550
3	+0.76519787	+0.76523858
4	+0.76957500	+0.76961505
5	+0.77550173	+0.77554089
6	+0.78286773	+0.78290582
7	+0.79156655	+0.79160345
8	+0.80149484	+0.80153024
9	+0.81255239	+0.81258607
10	+0.82464194	+0.82467365
11	+0.83766848	+0.83769804
12	+0.85153961	+0.85156691
13	+0.86616498	+0.86618978
14	+0.88145620	+0.88147837
15	+0.89732635	+0.89734566
16	+0.91369003	+0.91370648
17	+0.93046325	+0.93047667
18	+0.94756287	+0.94757313
19	+0.96490663	+0.96491349
20	+0.98241276	+0.98241621
21	+1.00000000	+1.00000000

Tabela 2.3: Solução Nodal

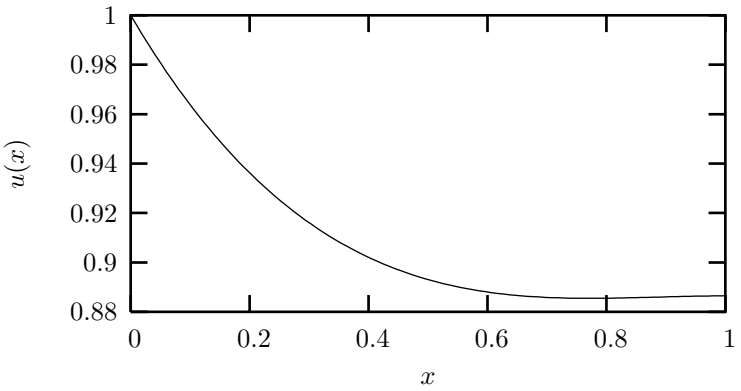


Figura 2.7: Solução exata

No	Aproximada	Exata
1	+1.00000000	+1.00000000
2	+0.98073423	+0.98073846
3	+0.96379620	+0.96380430
4	+0.94901866	+0.94903004
5	+0.93623948	+0.93625379
6	+0.92530161	+0.92531860
7	+0.91605282	+0.91607201
8	+0.90834486	+0.90836591
9	+0.90203339	+0.90205604
10	+0.89697766	+0.89700150
11	+0.89303994	+0.89306480
12	+0.89008522	+0.89011097
13	+0.88798118	+0.88800752
14	+0.88659751	+0.88662428
15	+0.88580561	+0.88583273
16	+0.88547844	+0.88550586
17	+0.88549024	+0.88551772
18	+0.88571596	+0.88574356
19	+0.88603115	+0.88605875
20	+0.88631147	+0.88633901
21	+0.88643253	+0.88646013

Tabela 2.4: Solução Nodal

2.10 Exercícios

1. Execute o programa `PEU.cpp` com diferentes valores de α e verifique que, quando diminuímos o valor de α, o erro aumenta nas normas L^2 e H^1.

2. Define-se malha geométrica no intervalo $[0,1]$ a malha cujos pontos nodais são da seguinte forma
$$\begin{cases} x_i = \lambda^{m-i}, & i = 1, \cdots, m \\ x_0 = 0, \end{cases} \tag{2.99}$$
onde $0 < \lambda < 1$ é uma constante positiva. Nessas condiçãos, dizemos que a malha tem razão α.

   Usando a malha geométrica com $\lambda = 4/5$, altere o programa `PEU.cpp` de forma a determinar a solução aproximada do Exemplo 2 do Capítulo 2, com 20 elementos. Compare com a solução aproximada obtida usando a malha uniforme. Em que malha obtemos melhor solução?

3. Considere o seguinte problema, chamado de Problema Sturm-Liouville:
$$\begin{cases} -\Big(p(x)u'(x)\Big)' + q(x)u(x) = f, & \forall x \in (0,1) \\ u(0) = 0, \qquad u'(1) = 0, \end{cases} \tag{2.100}$$
onde u' denota $\dfrac{du}{dx}$, $p(x)$ e $q(x)$ são funções que representam características físicas. Assumiremos que $p, q \in C^1(0,1)$ e, além disso, que existem constantes positivas α_1 , α_2 , β_1 , β_2 e δ tais que:

   (i) $\alpha_1 \geq p(x) \geq \alpha_2 > 0, \qquad \beta_1 \geq q(x) \geq \beta_2 \geq 0, \qquad \forall x \in [0,1]$;

   (ii) $|p'(x)| \leq \delta, \qquad \forall x \in [0,1]$.

   Considere o espaço $V = \{v \in H^1(0,1); v(0) = 0\}$ e $f \in L^2(0,1)$.

   (a) Mostre que a formulação variacional do problema é dada por:
$$\int_0^1 (pu'v' + quv)dx = \int_0^1 fv\, dx, \qquad \forall v \in V.$$

   (b) Usando como base as funções lineares por partes φ_i, determine a matriz global e o vetor força global.

   (c) Mostre que a matriz global é definida positiva.

(d) Usando as funções locais lineares por partes φ_a^e, determine a matriz local e a força local para cada elemento e.

Sugestão: Considere as funções p e q constantes sobre cada elemento e. Por exemplo, no intervalo $[x_1^e, x_2^e]$, tome $p(x) = (p(x_1^e) + p(x_2^e))/2$. Esta aproximação facilita no cálculo da matriz local.

(e) Usando o programa `PEU.cpp`, determine a solução numérica aproximada no intervalo $[0, 1]$ nas seguintes condições:

i. $p(x) = 1 + x, \quad q(x) = 2 \quad \text{e} \quad f(x) = 2x(4 - x)$;

ii. $p(x) = 1 + \dfrac{1}{2}x^2, \quad q(x) = \dfrac{1 + 3x}{2 - x} \quad \text{e} \quad f(x) = 6x^2 - x + 2.$

(f) Sabendo que $u(x) = x(2 - x)$ é a solução exata do problema com as funções do item anterior, determine o erro nas normas de $L^2(0, 1)$ e $H^1(0, 1)$.

CAPÍTULO 3

Função Base e Estimativa de Erro

No capítulo anterior, as soluções aproximadas nos vários exemplos considerados foram geradas pelas funções base φ_i, onde cada φ_i é uma funções linear por partes. Neste capítulo serão tratados outros tipos de função base: os polinômios por partes de grau 2 e de grau 3, assim como os procedimentos para obtenção da solução aproximada, isto é, a matriz força local e os respectivos algoritmos para obtenção da matriz e força global. Introduzimos também estimativas de erro para espaços de Sobolev e a relação entre os erros e o grau do polinômio gerador no espaço V_m das soluções aproximadas. Vale observar desde já que o uso de funções base de ordem superior é indispensável para alguns problemas, tais como o da viga elástica, que abordaremos ainda neste capítulo.

3.1 Função Base de Ordem Superior

Definimos em (2.19) as funções $\varphi_i(x)$ lineares por partes e com elas formamos a base geradora do subespaço $V_m = [\varphi_1, \varphi_2, \ldots, \varphi_m]$, i.e.,

$$V_m = \left\{ v_h \in V;\ v_h|_{I_i} = v_h^i \in P_1(I_i) \right\},$$

onde $P_1(I_i)$ é o conjunto dos polinômios do primeiro grau em I_i e V um subespaço de $H^1(0,1)$ (dependendo das condições de contorno do problema)

É natural a questão de se saber se, aumentando o grau do polinômio interpolador, melhoramos a ordem de convergência da soluções aproximadas. Para analisar essa questão, vamos primeiramente introduzir exemplos de bases interpoladoras de graus $k = 2$ e $k = 3$.

Para cada $i = 1, \ldots, m$ e $k \in \mathbb{N}$, seja $P_k(I_i)$ o conjunto dos polinômios de grau menor ou igual a k definidos no intevalo I_i e consideremos o espaço de elementos finitos V_m^k definido por

$$V_m^k = \left\{ v_h \in V;\ v_h^i \in P_k(I_i) \right\},$$

onde v_h^i denota a restrição de v_h ao intervalo (elemento) $I_i = [x_{i-1}, x_i]$. Observe que $V \subset C([0,1])$, de modo que as funções de V_m^k são contínuas, polinomiais por partes e satisfazem as mesmas condições de fronteira de V.

No que segue apresentaremos os subespaços V_m^k gerados por bases clássicas de polinômios de grau $k = 2$ e $k = 3$, definidos em cada elemento I_i.

3.1.1 Base Quadrática

Considere o subespaço

$$V_m^2 = \left\{ v_h \in V;\ v_h|_{I_i} = v_h^i \in P_2(I_i) \right\},$$

onde $P_2(I_i)$ é o conjunto dos polinômios de grau 2 definidos em cada elemento finito I_i.

Para gerar cada um dos polinômios interpoladores de grau 2 por partes, é preciso introduzir três pontos, de modo que o intervalo $\Omega = (0,1)$ possa ser discretizado por um número par de pontos. Mais precisamente, para cada intervalo $I_i = [x_{2i}\ ,\ x_{2(i+1)}]$, introduzimos um ponto intermediário denotado por $x_{2i+1} \in I_i$, gerando dessa forma os subintervalos $[x_{2i}\ , x_{2i+1}]$ e $[x_{2i+1}\ , x_{2(i+1)}]$ para cada $i = 0, 1, \cdots n-1$, com $x_0 = 0$ e $x_{2m} = 1$. Definimos então o passo de cada intervalo por $h_i = x_{2(i+1)} - x_{2i}$. Assim, temos o subespaço $V_m^2 = [\varphi_0, \varphi_1, \cdots \varphi_{2m}]$, onde as funções são definidas das seguinte forma:

1. para índices pares,

$$\begin{aligned} &\varphi_{2i}(x_{2i}) = 1, \quad \varphi_{2i}(x_{2i+1}) = \varphi_{2i}(x_{2i+2}) = 0, && \text{em } I_i, \\ &\varphi_{2i}(x_{2i-1}) = \varphi_{2i}(x_{2i-2}) = 0 && \text{em } I_{i-1}. \end{aligned}$$

2. para índices ímpares,

$$\varphi_{2i+1}(x_{2i+1}) = 1, \quad \varphi_{2i+1}(x_{2i}) = \varphi_{2i+1}(x_{2i+2}) = 0, \quad \text{em } I_i.$$

Portanto, podemos defini-las explicitamente por

$$\varphi_{2i}(x) = \begin{cases} -\dfrac{1}{h_i}(x - x_{2i-2}) + \dfrac{2}{h_i^2}(x - x_{2i-2})^2, & \forall x \in [x_{2i-2}, x_{2i}], \\ 1 - \dfrac{3}{2h_i}(x - x_{2i}) + \dfrac{1}{2h_i^2}(x - x_{2i})^2, & \forall x \in [x_{2i}, x_{2i+2}], \\ 0, & \forall x \notin [x_{2i-2}, x_{2i+2}]. \end{cases} \tag{3.1}$$

$$\varphi_{2i+1}(x) = \begin{cases} -\dfrac{4}{h_i^2}(x - x_{2i})(x - x_{2i+2}), & \forall x \in [x_{2i}, x_{2i+2}], \\ 0, & \forall x \notin [x_{2i}, x_{2i+2}]. \end{cases} \tag{3.2}$$

Nos extremos $x_0 = 0$ e $x_{2m} = 1$ temos respectivamente,

$$\varphi_0 = \begin{cases} 1 - \dfrac{3}{2h_i}\, x + \dfrac{1}{2h_i^2}\, x^2, & \forall\, x \in [x_0, x_2], \\ 0, & \forall\, x \notin [x_0, x_2] \end{cases},$$

$$\varphi_{2n} = \begin{cases} 1 + \dfrac{3}{h_i}(x-1) + \dfrac{2}{h_i^2}(x-1)^2, & \forall\, x \in [x_{2m-2}, x_{2m}], \\ 0, & \forall\, x \notin [x_{2m-2}, x_{2m}]. \end{cases}$$

Note que no nó x_{2i}, temos $(\varphi_{2i} : \varphi_{2i+j})_0 = 0$ se $|j| \geq 3$, de modo que a matriz global K é uma matriz pentagonal.

- **Matriz Rigidez Local**

Consideremos a matriz cujos coeficientes K_{ij} são

$$K_{ij} := \int_0^1 \left(\alpha \frac{d\varphi_i}{dx} \frac{d\varphi_j}{dx} + \beta \varphi_i \varphi_j \right) dx.$$

Repetindo os argumentos do capítulo anterior, cada intervalo $I_i = [x_{2i}\ , x_{2i+2}]$ (com $x_0 = 0$ e $x_{2m} = 1$) está associado um elemento e, de tal forma que $I_i = [x_1^e, x_2^e]$. Assim, em cada elemento e tem-se definidas as seguintes funções de interpolação local $\varphi_a^e(x)$ em $[x_1^e, x_2^e]$, (veja Fig. 3.1)

$$\begin{cases} \varphi_1^e(x) = 1 - \dfrac{3}{h_e}(x - x_1^e) + \dfrac{2}{h_e^2}(x - x_1^e)^2, \\ \varphi_2^e(x) = -\dfrac{4}{h_e^2}(x - x_1^e)(x - x_2^e), \\ \varphi_3^e(x) = -\dfrac{1}{h_e}(x - x_1^e) + \dfrac{2}{h_e^2}(x - x_1^e)^2, \end{cases} \tag{3.3}$$

onde $h_e = x_2^e - x_1^e$.

Assim, cada matriz local K_{ab}^e $(1 \leq a, b\, e 3)$ e Força Local F_a^e são definidas respectivamente por

$$\begin{aligned} K_{ab}^e &= \int_{x_1^e}^{x_2^e} \left(\alpha \frac{d\varphi_a^e}{dx} \frac{d\varphi_b^e}{dx} + \beta \varphi_a^e \varphi_b^e \right) dx \\ F_a^e &= \int_{x_1^e}^{x_2^e} f \varphi_a^e \, dx. \end{aligned} \tag{3.4}$$

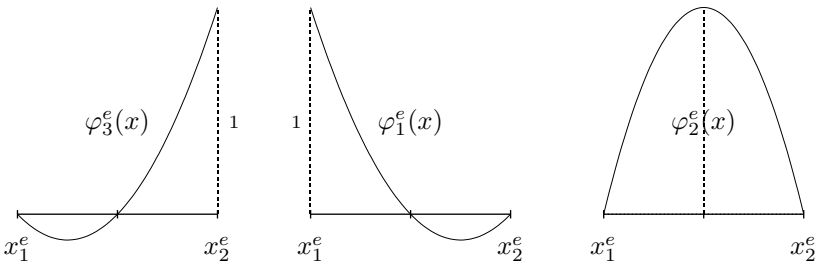


Figura 3.1: Função base local: quadrática

Os termos da matriz local (3.4) são dados explicitamente por

$$k_{11}^e = k_{33}^e = \alpha\frac{7}{3h_e} + \beta\frac{2h_e}{15}, \qquad k_{12}^e = k_{21}^e = k_{23}^e = k_{32}^e = -\alpha\frac{8}{3h_e} + \beta\frac{h_e}{15},$$
$$k_{22}^e = \alpha\frac{16}{3h_e} + \beta\frac{8h_e}{15}, \qquad k_{13}^e = k_{31}^e = \alpha\frac{1}{3h_e} - \beta\frac{h_e}{30}.$$

De forma geral, sem considerar os valores de fronteira, a contribuição local para os coeficientes da matriz global K de ordem $(2m+1)\times(2m+1)$ é dada pelo seguinte algoritmo:

$$K_{00} = k_{11}^1, \quad K_{01} = k_{12}^1, \quad K_{02} = k_{13}^1,$$
$$K_{11} = k_{22}^1, \quad K_{12} = k_{23}^1, \quad K_{2m\,2m} = k_{33}^{m-1}.$$

$$\begin{cases} K_{(2e-1),(2e+1)} = K_{(2e+1),(2e-1)} = 0, \\ K_{(2e),(2e)} = k_{33}^e + k_{11}^{e+1}, \\ K_{(2e),(2e+1)} = k_{12}^e, \\ K_{(2e),(2e+2)} = k_{13}^e, \\ K_{(2e+1),(2e+1)} = k_{22}^e, \\ K_{(2e+1),(2e+2)} = k_{23}^e, \end{cases} \qquad e = 1, 2, 3, \ldots, m-1. \tag{3.5}$$

A contribuição do vetor força local f_a^e para o vetor força global F de ordem $(2m+1)$ é dado pelo algoritmo:

$$F_{2e} = f_3^{e-1} + f_1^e, \qquad F_{2e+1} = f_2^e, \quad e = 1, 2, 3, \ldots, m-1,$$

$$F_0 = f_1^1, \qquad F_1 = f_2^1,$$
$$F_{2m-1} = f_2^{m-1}, \qquad F_{2m} = f_3^{m-1}.$$

Os coeficientes referentes à fronteira do problema aproximado dependem do tipo de condição de fronteira do problema contínuo, ou seja, para cada tipo Dirichlet, Neumann ou misto, a inserção segue o mesmo procedimento da interpolação linear.

3.1.2 Base Cúbica

Um spline cúbico é uma função interpoladora, polinomial por partes, de grau 3 em cada intervalo $I_i = [x_i, x_{i+1}]$ e duas vezes diferenciável em cada nó x_i. Mais precisamente, dada uma função $g(x)$, o spline cúbico $C(x)$ permite interpolar $g(x)$, sendo duas vezes diferenciável em cada nó x_i. Assim, em cada intervalo I_i, a função $C(x)$ tem a forma

$$C(x) = C_i(x) = a_{i3}x^3 + a_{i2}x^2 + a_{i1}x^1 + a_{i0}, \quad x \in I_i, \quad i = 1, 2 \ldots, m-1.$$

Para que $C(x)$ satisfaça as propriedades de interpolação e seja duas vezes diferenciável, é necessário que os coeficientes a_{ij} de $C(x)$ sejam adequadamente definidos, isto é, satisfaçam as seguintes condições:

1. $C(x_i) = g(x_i), \qquad i = 1, 2 \ldots m,$
2. $C_{i-1}(x_i) = C_i(x_i), \quad C'_{i-1}(x_i) = C'_i(x_i), \quad C''_{i-1}(x_i) = C''_i(x_i), \quad i = 2, 3, \ldots m-1.$

Os coeficientes $a_{i3},\ a_{i2},\ a_{i1},\ a_{i0}$ a serem determinados totalizam $4(m-1)$ incógnitas e o número de condições requeridas é: $m+3(m-2) = 4m-6$. Assim, faz-se necessário impor duas condições a mais para que $C(x)$ seja determinado de forma única.

Para exemplificar, embora essa não seja a melhor escola, podemos considerar as seguintes condições adicionais:

$$C''(x_1) = C''(x_m) = 0. \tag{3.6}$$

Nesse caso o spline é denominado *spline cúbico natural*. Não é a boa escolha porque possui a desvantagem de não satifazer a condição

$$C'_j(x_i) = C_j(x_i) = 0, \quad \text{se} \quad |i-j| \geq 2.$$

O ideal é construir um spline cúbico $C_i(x)$ que se anula fora do intervalo $[x_{i-2}, x_{i+2}]$ e dessa forma gerar uma matriz global heptagonal, ou seja, uma matriz cujos coeficientes satisfaçam

$$K_{ij} = \int_0^1 \left(\alpha \frac{dC_i}{dx}\frac{dC_j}{dx} + \beta C_i C_j \right) dx = 0 \quad \text{se} \quad |i-j| > 3.$$

Para isso, consideremos inicialmente a função par $\theta : \mathbb{R} \to \mathbb{R}$ da seguinte forma

$$\theta(x) = \begin{cases} \theta_1(x), & \forall\, x \in [0,1], \\ \theta_2(x), & \forall\, x \in [1,2], \\ \theta_1(-x), & \forall\, x \in [-1,0], \\ \theta_2(-x), & \forall\, x \in [-2,-1], \\ 0, & \forall\, x \notin [-2,2], \end{cases}$$

onde

$$\theta_1(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3,$$
$$\theta_2(x) = b_0 + b_1 x + b_2 x^2 + b_3 x^3,$$

devem satisfazer as seguintes propriedades:

$$\begin{cases} \theta_1(0) = 1, & \theta_1'(0) = 0, & \\ \theta_1(1) = \theta_2(1), & \theta_1'(1) = \theta_2'(1), & \theta_1''(1) = \theta_2''(1), \\ \theta_2(2) = 0, & \theta_2'(2) = 0, & \theta_2''(2) = 0. \end{cases}$$

Com estas oito condições podemos determinar os coeficientes a_i e b_i a partir de um sistema linear, cuja solução nos fornece as seguintes funções $\theta_1(x)$ e $\theta_2(x)$:

$$\begin{cases} \theta_1(x) = 1 - \dfrac{3}{2}x^2 + \dfrac{3}{4}x^3, \\ \theta_2(x) = 2 - 3x + \dfrac{3}{2}x^2 - \dfrac{1}{4}x^3. \end{cases} \tag{3.7}$$

Consideremos agora a discretização uniforme do intervalo $[a,b]$ na forma $a = x_1 < x_2 < cdots < x_{m+1} = b$, com $h = x_{i+1} - x_i$. Então, para cada nó x_i, com $i = 1, 2, \cdots, m+1$, definimos o elemento de função base spline cúbico, conhecido como *B-spline*,

$$B_i(x) = \theta\left(\frac{x - x_i}{h}\right).$$

Nessas condições a função $B_i(x)$ pode ser reescrita explicitamente na seguinte forma:

$$B_i(x) = \begin{cases} \dfrac{1}{4h^3}(x - x_{i-2})^3, & \forall\, x \in [x_{i-2}, x_{i-1}], \\ \dfrac{1}{4} + \dfrac{3}{4h}(x - x_{i-1}) + \dfrac{3}{4h^2}(x - x_{i-1})^2 - \dfrac{3}{4h^3}(x - x_{i-1})^3, & \forall\, x \in [x_{i-1}, x_i], \\ \dfrac{1}{4} + \dfrac{3}{4h}(x_{i+1} - x) + \dfrac{3}{4h^2}(x_{i+1} - x)^2 - \dfrac{3}{4h^3}(x_{i+1} - x)^3, & \forall\, x \in [x_i, x_{i+1}], \\ \dfrac{1}{4h^3}(x_{i+2} - x)^3, & \forall\, x \in [x_{i+1}, x_{i+2},] \\ 0, & \forall\, x \notin [x_{i-2}, x_{i+2}]. \end{cases}$$

Pode-se mostrar que $B_i(x)$ é de fato uma base para os splines cúbicos, ou seja, toda spline cúbica pode ser escrita como combinação linear das B-splines.

Note que para as funções $B_1,\ B_2,\ B_m$ e B_{m+1}, precisamos introduzir os pontos nodais auxiliares: $x_{-1},\ x_0$, e $x_{m+2},\ x_{m+3}$, que dependem dos valores de fronteira.

Consideremos como anteriormente o intervalo $\Omega = (0,1)$ com $x_1 = 0$ e $x_{m+1} = 1$ e os valores de fronteira do tipo Dirichlet homogêneas, isto é, $u(x_1) = u(x_{m+1}) = 0$. As funções $B_3, \cdots, B_{m-1}$ se anulam nas fronteiras de cada intervalo que as define, mas as funções B_1, B_2, B_m e B_{m+1} não se anulam. Com esse objetivo, definimos

$$\begin{cases} \varphi_i(x) = B_i(x), \quad i = 3, \ldots, m-1, \\ \varphi_1(x) = B_1(x) - 4B_0(x), \\ \varphi_2(x) = B_2(x) - B_0(x), \\ \varphi_m(x) = B_m(x) - B_{m+2}(x), \\ \varphi_{m+1}(x) = B_{m+1}(x) - 4B_{m+2}(x), \end{cases} \tag{3.8}$$

onde $B_0(x)$ e $B_{m+2}(x)$ são os B-splines nos pontos $x_0 = x_1 - h$ e $x_{m+2} = x_{m+1} + h$, respectivamente.

Temos então

$$\varphi_i(x_i) = 1, \quad i = 2, 3, \ldots, m, \qquad \varphi_i(x_{i+1}) = 1/4, \quad i = 1, 2, \ldots, m-1,$$
$$\varphi_i(x_{i-1}) = 1/4, \quad i = 3, 4, \ldots, m+1,$$
$$\varphi_1(x_1) = \varphi_2(x_1) = \varphi_m(x_{m+1}) = \varphi_{m+1}(x_{m+1}) = 0.$$

Portanto satisfazem os valores de fronteira. Dessa forma a solução aproximada é um polinômio cúbico tal que $u_h(x_1) = u_h(x_{m+1}) = 0$, onde V_m é o subespaço gerado por $V_m = [\varphi_2, \varphi_3, \ldots, \varphi_m]$, dada por

$$u_h(x) = \sum_{j=2}^{m} d_j \varphi_j(x)\,, \qquad \varphi_j \in V_m.$$

Note que as funções φ_1 e φ_{m+1} são linearmente dependentes das funções $\varphi_i,\ i = 2, 3, \ldots, m$.

Note também que pela definição do B-spline, não temos a propriedade de interpolação, ou seja, $u_h(x_i) \neq d_i$, entretanto a solução aproximada nos nós x_i pode ser

obtida pela relação,

$$\left|\begin{array}{l} u_h(x_2) = d_2 + \frac{1}{4}d_3, \\ u_h(x_i) = \frac{1}{4}d_{i-1} + d_i + \frac{1}{4}d_{i+1} \quad \text{para} \quad i = 3, 4, \ldots, m-1, \\ u_h(x_m) = \frac{1}{4}d_{m-1} + d_m. \end{array}\right.$$

que pode ser representada na forma matricial $A\boldsymbol{d} = \boldsymbol{u}_h$, onde $\boldsymbol{d} = [(d_2, d_3, \ldots, d_m]^T$, $\boldsymbol{u}_h = [u_h(x_2), u_h(x_3), \ldots u_h(x_m)]^T$ e A é uma matriz quadrada de ordem $(m-1)\times(m-1)$, tridiagonal, ou seja,

$$A = \begin{pmatrix} 1 & 1/4 & 0 & \cdots & 0 & 0 \\ 1/4 & 1 & 1/4 & \cdots & 0 & 0 \\ 0 & 1/4 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \vdots & \vdots & \vdots & 0 \\ 0 & 0 & \cdots & 1/4 & 1 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1 \end{pmatrix} \tag{3.9}$$

• **Matriz Local**

Com os mesmos argumentos anteriores, temos para cada elemento e no intervalo $[x_1^e, x_2^e] = [x_i, x_{i+1}]$, com $h = x_2^e - x_1^e$, as seguintes funções de interpolação local (ver Fig. 3.2),

$$\left\{\begin{array}{l} \varphi_1^e(x) = \dfrac{1}{4} + \dfrac{3}{4h}(x_2^e - x) + \dfrac{3}{4h^2}(x_2^e - x)^2 - \dfrac{3}{4h^3}(x_2^e - x)^3, \\ \varphi_2^e(x) = \dfrac{1}{4h^3}(x_2^e - x)^3, \\ \varphi_3^e(x) = \dfrac{1}{4} + \dfrac{3}{4h}(x - x_1^e) + \dfrac{3}{4h^2}(x - x_1^e)^2 - \dfrac{3}{4h^3}(x - x_1^e)^3, \\ \varphi_4^e(x) = \dfrac{1}{4h^3}(x - x_1^e)^3. \end{array}\right. \tag{3.10}$$

Para cada elemento e, os elementos da matriz local K^e (de ordem 4×4) e da Força Local f^e (de ordem 4×1) são dadas respectivamente por

$$\begin{array}{ll} k_{ab}^e = \displaystyle\int_{x_1^e}^{x_2^e} \left(\alpha \frac{d\varphi_a^e}{dx}\frac{d\varphi_b^e}{dx} + \beta\varphi_a^e\varphi_b^e\right) dx, & 1 \le a, b \le 4, \\ f_a^e = \displaystyle\int_{x_1^e}^{x_2^e} f\varphi_a^e dx, & 1 \le a \le 4. \end{array} \tag{3.11}$$

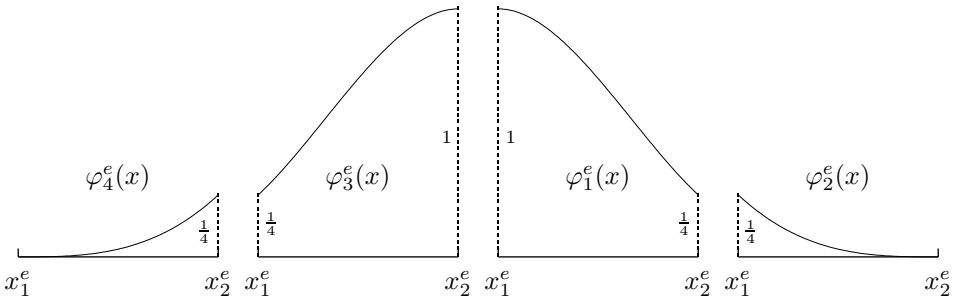


Figura 3.2: Função base local: spline cúbico

Explicitamente, os termos da matriz local (3.11) são dados por

$$
\begin{aligned}
k_{11}^e = k_{33}^e &= \alpha\frac{51}{80h} + \beta\frac{297h}{560}, & k_{12}^e = k_{21}^e = k_{34}^e = k_{43}^e &= \alpha\frac{21}{160h} + \beta\frac{129h}{2240},\\
k_{22}^e = k_{44}^e &= \alpha\frac{9}{80h} + \beta\frac{h}{112}, & k_{13}^e = k_{31}^e &= \alpha\frac{-87}{160h} + \beta\frac{933h}{2240},\\
k_{24}^e = k_{42}^e &= \alpha\frac{-3}{160h} + \beta\frac{h}{2240}, & k_{14}^e = k_{41}^e = k_{23}^e = k_{32}^e &= \alpha\frac{-9}{40h} + \beta\frac{3h}{112}.
\end{aligned}
$$

A contribuição local da matriz k_{ab}^e para a matriz global K, de ordem $(m+1)\times(m+1)$, desconsiderando a simetria da matriz (3.11), pode ser obtida pelo seguinte algoritmo:

$$
\left\{\begin{array}{lcll}
K_{ee} & = & k_{44}^{e-2} + k_{33}^{e-1} + k_{11}^e + k_{22}^{e+1}, & e = 3,4,\ldots,m-1,\\
K_{e(e+1)} & = & k_{34}^{e-1} + k_{13}^e + k_{12}^{e+1}, & e = 3,4\ldots,m-2,\\
K_{(e+1)e} & = & k_{43}^{e-1} + k_{31}^e + k_{21}^{e+1}, & e = 3,4\ldots,m-2,\\
K_{e(e+2)} & = & k_{14}^e + k_{23}^{e+1}, & e = 2,3,\ldots,m-2,\\
K_{(e+2)e} & = & k_{41}^e + k_{32}^{e+1}, & e = 2,3,\ldots,m-2,\\
K_{e(e+3)} & = & k_{24}^{e+1}, & e = 1,2,\ldots,m-2,\\
K_{(e+3)e} & = & k_{42}^{e+1}, & e = 1,2,\ldots,m-2.
\end{array}\right. \tag{3.12}
$$

Tendo em conta as condições (3.8) e para satisfazer a condição de fronteira nula, o algoritmo associado aos elementos da matriz global K referentes a essas restrições é

dado respectivamente por

$$\begin{cases} K_{11} = k_{11}^1 + 16k_{22}^1 - 8k_{12}^1 + k_{22}^2, & K_{22} = k_{33}^1 + k_{11}^2 + k_{22}^3 - 2k_{32}^1 + k_{22}^1 \\ K_{12} = k_{13}^1 + k_{21}^2 - k_{12}^1 - 4(k_{23}^1 - k_{22}^1), & K_{21} = k_{31}^1 + k_{12}^2 - k_{21}^1 - 4(k_{32}^1 - k_{22}^1), \\ K_{13} = k_{14}^1 + k_{23}^2 - 4k_{24}^1, & K_{31} = k_{41}^1 + k_{32}^2 - 4k_{42}^1, \\ K_{23} = k_{34}^1 + k_{13}^2 + k_{21}^3 - k_{24}^1, & K_{32} = k_{43}^1 + k_{31}^2 + k_{12}^3 - k_{42}^1, \end{cases}$$

$$\begin{cases} K_{m-1,m+1} = k_{14}^{m-1} + k_{23}^m - 4k_{24}^m, & K_{m+1,m-1} = k_{41}^{m-1} + k_{32}^m - 4k_{42}^m, \\ K_{m-1,m} = k_{34}^{m-2} + k_{13}^{m-1} + k_{21}^m - k_{24}^m, & K_{m,m-1} = k_{43}^{m-2} + k_{31}^{m-1} + k_{12}^m - k_{42}^m, \\ K_{m,m} = k_{44}^{m-2} + k_{33}^{m-1} + k_{11}^m - 2k_{14}^m + k_{44}^m, & \\ K_{m,m+1} = k_{34}^{m-1} + k_{13}^m - 4(k_{14}^m - k_{44}^m) - k_{43}^m, & \\ K_{m+1,m} = k_{43}^{m-1} + k_{31}^m - 4(k_{41}^m - k_{44}^m) - k_{34}^m & \\ K_{m+1,\,m+1} = k_{44}^{m-1} + k_{33}^m + 16k_{44}^m - 8k_{34}^m. & \end{cases}$$

A contribuição do vetor força local f_a^e para o vetor força global $\boldsymbol{F}$ de ordem n é dado pelo algoritmo:

$$F_e = f_4^{e-2} + f_3^{e-1} + \quad \text{para} \quad e = 3, 4, ldots, m-1,$$

com os valores nos extremos dados por

$$\begin{array}{ll} F_2 = f_3^1 + f_1^2 + f_2^3, & F_1 = f_1^1 + f_2^2, \\ F_m = f_4^{m-2} + f_3^{m-1} + f_1^m, & F_{m+1} = f_4^{m-1} + f_3^m. \end{array}$$

Assim, o sistema linear $Kbfld = \boldsymbol{F}$ tem $m+1$ incógnitas $\boldsymbol{d} = (d_1, d_2, \cdots d_{m+1})$. A forma de inserção dos valores de fronteira é semelhante ao procedimento que obtivemos para os polinômios lineares por partes (ver Seção 2.7). Dessa forma, se as condições de fronteira são do tipo Dirichlet, Neumann ou misto, teremos sistemas quadrados de ordem $m-1$, $m+1$ e m, respectivamente. Vale observar que o algoritmo (3.12) foi deduzido considerando a simetria da matriz local.

3.1.3 Base de Hermite

Nesta seção vamos introduzir funções base convenientes para determinar a solução aproximada do problema da viga elástica, formulada por Bernoulli-Euler.

Para $\Omega = (0,1)$, a formulação forte do problema da viga elástica com extremos engastados pode ser dada formalmente como um problema de valores de fronteira para uma equação diferencial de quarta ordem. Mais precisamente, dada $f : (0,1) \to \mathbb{R}$, o problema consiste em se determinar a função $u : [0,1] \to \mathbb{R}$ tal que

$$\begin{cases} \dfrac{d^4u}{dx^4} = f \qquad \text{em} \quad (0,1), \\ u(0) = u(1) = u'(0) = u'(1) = 0. \end{cases} \tag{3.13}$$

Podemos repetir o mesmo procedimento aplicado ao problema (2.1), ou seja, multiplicar os dois lados da equação em (3.13) por uma função teste e integrar por partes duas vezes. Assim, obtemos a seguinte formulação variacional

$$a(u,v) = (f:v)\,,\ \forall\, v \in V, \tag{3.14}$$

onde

$$a(u,v) = \int_0^1 u''(x)v''(x)\,dx \quad \text{e} \quad (f:v) = \int_0^1 f(x)v(x)\,dx. \tag{3.15}$$

Note que para obter a formulação (3.14) com $f \in L^2(0,1)$, é necessário que u, u' e u'' pertençam ao espaço $L^2(0,1)$, o que significa que $u \in H^2(0,1)$. Além disso, em vista das condições de fronteira homogêneas, o espaço adequado é $H_0^2(0,1)$.

Seguindo o mesmo raciocínio, queremos determinar a solução aproximada da forma:

$$u_h(x) = \sum_{j=1}^{m} d_j\varphi_j(x), \quad \varphi_j \in V_m, \tag{3.16}$$

onde $V_m = [\varphi_1, \varphi_2, \cdots, \varphi_m]$ e $\varphi_j(0) = \varphi_j(1) = \varphi_j'(0) = \varphi_j'(1) = 0$. Vimos também que determinar a solução aproximada consiste em resolver o sistema linear $K\boldsymbol{d} = \boldsymbol{F}$, onde os coeficientes da matriz global K são dados por

$$K_{ij} = \int_0^1 \left(\frac{d^2\varphi_i}{dx^2}\,\frac{d^2\varphi_j}{dx^2}\right) dx. \tag{3.17}$$

Assim, as funções testes precisam pertencer a $H_0^2(0,1)$.

Obviamente, as funções base de polinômios de grau 1 ou grau 2 por partes não são adequadas para o problema da viga, porque a segunda derivada não pertence ao espaço $L^2(0,1)$. Portanto, para este problema, os splines cúbicos são adequados, porque pertencem ao espaço $C^2(0,1)$, fazendo sentido a integral (3.17), desde que sejam modificados nos extremos $x = 0$ e $x = 1$, de tal forma que a solução aproximada definida em (3.16) satisfaça os valores de fronteira $u_h(0) = u_h'(0) = u_h(1) = u_h'(1) = 0$.

Um outro tipo de função teste de interpolação cúbica é obtido pelos polinômios de Hermite, que descrevemos a seguir. Note que, sendo $H_0^2(0,1)$ o espaço da soluções admissíveis, qualquer subsespaço $V_m \subset H_0^2(0,1)$ deve ser pelo menos de classe $C^1(0,1)$, ou seja, que a função base φ_i e sua primeira derivada φ_i' seja contínua em $[0,1]$. Os polinômios de Hermite satisfazem essa condição de continuidade e são portanto muito convenientes para o problema da viga (3.13).

Consideremos a discretização do intervalo $[a,b]$ na forma $a = x_0 < x_1 < \cdots < x_m = b$, onde, para cada nó x_i, definimos dois elementos de função base, $\phi_i(x)$ e $\psi_i(x)$, satisfazendo

$$\begin{cases} \phi_i(x_i) = 1, & \phi_i(x_{i+1}) = 0, & \phi_i(x_{i-1}) = 0, \\ \phi_i'(x_i) = 0, & \phi_i'(x_{i+1}) = 0, & \phi_i'(x_{i-1}) = 0, \end{cases} \tag{3.18}$$

$$\begin{cases} \psi_i(x_i) = 0, & \psi_i(x_{i+1}) = 0, & \psi_i(x_{i-1}) = 0, \\ \psi_i'(x_i) = 1, & \psi_i'(x_{i+1}) = 0, & \psi_i'(x_{i-1}) = 0, \end{cases} \tag{3.19}$$

e se anulam para $x \notin [x_{i-1}, x_{i+1}]$. Assim, no intervalo $[x_i, x_{i+1}]$, consideramos $\phi_i(x)$ e $\psi_i(x)$ como polinômios cúbicos $a_0 + a_1x + a_2x^2 + a_3x^3$ e podemos determinar os quatro coeficientes com as quatro condições dadas em (3.18) e (3.19) respectivamente para $\phi_i(x)$ e $\psi_i(x)$. Também podemos fazer o mesmo no intervalo $[x_{i-1}, x_i]$. Considerando a malha uniforme com $h = x_{i+1} - x_i$, obtemos polinômios de Hermite, que são funções $C^1(a, b)$ por parte, na seguinte forma:

$$\phi_i(x) = \begin{cases} \left(\left|\dfrac{x - x_i}{h}\right| - 1\right)^2 \left(2\left|\dfrac{x - x_i}{h}\right| + 1\right), & \forall\, x \in [x_{i-1}, x_{i+1}], \\ 0, & \forall\, x \notin [x_{i-1}, x_{i+1}], \end{cases} \tag{3.20}$$

$$\psi_i(x) = \begin{cases} (x - x_i)\left(\left|\dfrac{x - x_i}{h}\right| - 1\right)^2, & \forall\, x \in [x_{i-1}, x_{i+1}], \\ 0, & \forall\, x \notin [x_{i-1}, x_{i+1}]. \end{cases} \tag{3.21}$$

Observe que, com relação à função base de Hermite, a solução aproximada $u_h(x)$ pode ser escrita na forma,

$$u_h(x) = \sum_{i=0}^{m} \Big(c_i\, \phi_i(x) + d_i\, \psi_i(x) \Big). \tag{3.22}$$

De (3.20) e (3.21), tem-se que $u_h(x_i) = c_i$ e $u_h'(x_i) = d_i$.

Note que temos $2(m+1)$ incógnitas $\boldsymbol{c} = (c_0, c_1, \ldots, c_m)$ e $\boldsymbol{d} = (d_0, d_1, \ldots, d_m)$ para se determinar a solução aproximada $u_h(x)$ e também a sua derivada du_h/dx.

Para gerar as $2(m+1)$ equações correspondentes, podemos fazer o seguinte procedimento: considere o problema aproximado de (3.14), com a formulação (3.15), tomando u_h na forma (3.22)

$$\int_0^1 \left(\sum_{i=0}^{m} c_i \phi_i''(x) + d_i \psi_i''(x) \right) v''(x)\, dx = \int_0^1 f(x) v(x)\, dx, \qquad \forall v \in V_m.$$

Tomando $v = \phi_j(x)$ e depois $v = \psi_j(x)$ na identidade acima, obtemos dois sistemas com $(m+1)$ incógnitas cada, dados por

$$\int_0^1 \left(\sum_{i=0}^{m} c_i \phi_i''(x) \phi_j''(x) \right) dx + \int_0^1 \left(\sum_{i=0}^{m} d_i \psi_i''(x) \phi_j''(x) \right) dx = \int_0^1 f(x) \phi_j(x)\, dx,$$

$$\int_0^1 \left(\sum_{i=0}^{m} c_i \phi_i''(x) \psi_j''(x) \right) dx + \int_0^1 \left(\sum_{i=0}^{m} d_i \psi_i''(x) \psi_j''(x) \right) dx = \int_0^1 f(x) \psi_j(x)\, dx,$$

para cada $j = 0, 1, \ldots, m$.

Definindo as matrizes K e L com coeficientes dados respectivamente por

$$K_{ij} = \int_0^1 \phi_i''(x)\phi_j''(x)\,dx \quad \text{e} \quad L_{ij} = \int_0^1 \psi_i''(x)\phi_j''(x)\,dx,$$

podemos escrever os dois sistemas na forma

$$\left\{ \begin{array}{rcl} K\boldsymbol{c} + L^T\boldsymbol{d} & = & \boldsymbol{F}_1 \\ L\boldsymbol{c} + M\boldsymbol{d} & = & \boldsymbol{F}_2 \end{array} \right. \tag{3.23}$$

Esses dois sistemas são dependentes e podem ser escritos na forma de matriz bloco:

$$\begin{pmatrix} K & L^T \\ L & M \end{pmatrix} \begin{pmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \end{pmatrix}. \tag{3.24}$$

Temos portanto que resolver o sistema linear $\widehat{\mathbf{K}}\widehat{\mathbf{C}} = \widehat{\mathbf{F}}$ de ordem $2(m+1)$, notando que as matrizes K, L e M são tridiagonais e $\widehat{\mathbf{K}}$ é simétrica, com $[\boldsymbol{c}, \boldsymbol{d}]^T$ e $[\boldsymbol{F}_1, \boldsymbol{F}_2]^T$ vetores de $m+1$ componentes.

Como vimos em (2.31), as forças podem ser interpoladas, usando como base a função ϕ_i na forma

$$f(x) = \sum_{i=0}^{m} f_i\phi_i(x). \tag{3.25}$$

Assim, podemos escrever, como em (2.32),

$$\begin{aligned} F_{1j} &= \int_0^1 f(x)\phi_j(x)\,dx = \int_0^1 \sum_{i=0}^{m} f_i\phi_i(x)\phi_j(x)\,dx \\ &= f_{i-1}\int_{x_{i-1}}^{x_i} \phi_i\phi_{i-1}\,dx + f_i\int_{x_{i-1}}^{x_{i+1}} (\phi_i)^2\,dx + f_{i+1}\int_{x_i}^{x_{i+1}} \phi_i\phi_{i+1}\,dx, \end{aligned}$$

$$\begin{aligned} F_{2j} &= \int_0^1 f(x)\psi_j(x)\,dx = \int_0^1 \sum_{i=0}^{m} f_i\phi_i(x)\psi_j(x)\,dx \\ &= f_{i-1}\int_{x_{i-1}}^{x_i} \phi_i\psi_{i-1}\,dx + f_i\int_{x_{i-1}}^{x_{i+1}} \phi_i\psi_i\,dx + f_{i+1}\int_{x_i}^{x_{i+1}} \phi_i\psi_{i+1}\,dx. \end{aligned}$$

Note que também podemos interpolar a função f usando como função base a função ψ_i, ou seja, $f(x) = \sum_{i=0}^{m} f_i\psi_i'(x)$, e calcular $\mathcal{F}_1$ e $\mathcal{F}_2$ de forma análoga.

No que conerne as condições de fronteira, o sistema linear $\widehat{\mathbf{K}}\widehat{\mathbf{C}} = \widehat{\mathbf{F}}$ tem infinitas soluções, pois a matriz $\widehat{\mathbf{K}}$ é singular. Para obter uma única solução $\widehat{\mathbf{C}}$, necessário se

faz impor os valores de fronteira dados no problema contínuo. Em particular, para o exemplo (3.13), devemos ter as seguintes condições para a solução aproximada: $u_h(x_0) = u_h(0) = u_h(x_m) = u_h(1) = 0$ e $u'_h(x_0) = u'_h(0) = u'_h(x_m) = u'_h(1) = 0$, ou mais precisamente,

$$
\begin{aligned}
c_0 := u_h(0) &= \sum_{i=0}^{m} \Big(c_i\, \phi_i(0) + d_i\, \psi_i(0) \Big) = 0, \\
c_m := u_h(1) &= \sum_{i=0}^{m} \Big(c_i\, \phi_i(1) + d_i\, \psi_i(1) \Big) = 0, \\
d_0 := u'_h(0) &= \sum_{i=0}^{m} \Big(c_i\, \phi'_i(0) + d_i\, \psi'_i(0) \Big) = 0, \\
d_m := u'_h(1) &= \sum_{i=0}^{m} \Big(c_i\, \phi'_i(1) + d_i\, \psi'_i(1) \Big) = 0.
\end{aligned}
$$

Isso quer dizer que a matriz bloco $\widehat{\mathbf{K}}$ do sistema linear deve ser modificada de forma a garantir que a solução seja da forma, $\widehat{\mathbf{C}} = [0, c_1, c_2, \ldots, c_{m-1}, 0, 0, d_1, d_2, \ldots, d_{m-1}, 0]^T$), o que significa que o sistema passa a ter quatro incógnitas a menos, i.e., o sistema é de ordem $2(m-1)$. Para condições de fronteira não homogênea ou outros tipos de valor de fronteira, o procedimento é similar ao da Seção 2.7.

Em resumo, com a base de Hermite, os valores da função u_h e sua primeira derivada u'_h no nó x_i coincidem com os coeficiente de interpolação das funções base ϕ_i e ψ_i, respectivamente, ou seja $u_h(x_i) = c_i$ e $u'_h(x_i) = d_i$.

• **Matriz Local**

Com os mesmos argumentos anteriores, definimos a função base local no intervalo $[x_1^e, x_2^e]$ por (ver Fig. 3.3)

$$
\begin{cases}
\varphi_1^e(x) = \Big(\dfrac{x - x_1^e}{h} - 1 \Big)^2 \Big(\dfrac{2(x - x_1^e)}{h} + 1 \Big), \\
\varphi_2^e(x) = \Big(\dfrac{x_2^e - x}{h} - 1 \Big)^2 \Big(\dfrac{2(x_2^e - x)}{h} + 1 \Big), \\
\psi_1^e(x) = \Big(x - x_1^e \Big) \Big(\dfrac{x - x_1^e}{h} - 1 \Big)^2, \\
\psi_2^e(x) = \Big(x - x_2^e \Big) \Big(\dfrac{x_2^e - x}{h} - 1 \Big)^2.
\end{cases}
\tag{3.26}
$$

Assim, em cada elemento e, a matriz local K_{ab}^e de ordem 4×4 e o vetor força local

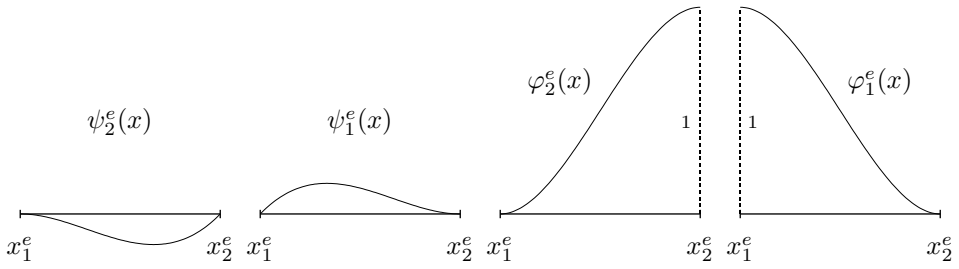


Figura 3.3: Função base local: Hermite

f_a^e de ordem 4×1 podem ser calculados por

$$K_{ab}^e = \int_{x_1^e}^{x_2^e} \left(\frac{d^2\varphi_a^e}{dx^2} \, \frac{d^2\varphi_b^e}{dx^2} \right) dx, \qquad 1 \leq a, b \leq 4,$$

$$f_a^e = \int_{x_1^e}^{x_2^e} f\varphi_a^e dx, \qquad 1 \leq a \leq 4.$$

As quantidades globais K_{ij} e f_i podem ser obtidas da maneira semelhante somando-se as contribuições locais de todos elementos nos nós correspondentes.

3.2 Análise de Erro do Problema Estacionário

O estudo da convergência das soluções aproximadas u_h obtidas pelo método de elementos finitos para a solução exata u do problema contínuo se faz a partir de estimativas de erro, cuja ordem de convergência depende do grau do polinômio interpolador de u_h no interior de cada elemento finito e, assim como do método de aproximação utilizado.

Entretanto, antes de abordarmos as estimativas de erros referentes especificament ao método de elemetos finitos, vejamos um resultado de caráter geral.

Sejam V um espaço de Hilbert real com norma $\|\cdot\|_V$ induzida pelo produto interno $(\cdot : \cdot)_V$, $f : V \to \mathbb{R}$ um funcional linear e contínuo em V e uma forma bilinear $a : V \times V \to \mathbb{R}$.

Consideremos então o seguinte problema variacional abstrato: *determinar* $u \in V$ *tal que*

$$a(u, v) = \langle f, v \rangle \qquad \forall\, v \in V, \tag{3.27}$$

onde $\langle \cdot, \cdot \rangle$ denota o produto de dualidade entre v e v'. Nesse contexto geral, o método de Galerkin consiste em se estabelecer um subespaço de dimensão m de V, V_m, e o problema aproximado: *encontrar* $u_h \in V_m$ *tal que*

$$a(u_h, v_h) = \langle f, v_h \rangle \qquad \forall\, v_h \in V_m, \tag{3.28}$$

onde o parâmetro h depende de m de tal forma que $h \to 0$ quando $m \to +\infty$.

A convergência a que nos referimos ocorre se $\lim_{h\to 0} \|u - u_h\|_V = 0$.

No que segue, suponhamos que a forma bilinear $a(\cdot, \cdot)$ é contínua e coerciva em V, i.e., existem constantes positivas δ_1 e δ_2 tais que, $\forall u, v \in V$,

$$\begin{cases} |a(u,v)| \le \delta_1 \|u\|_V \|v\|_V, \\ a(v,v) \ge \delta_2 \|v\|_V^2, \end{cases} \qquad \forall u, v \in V. \tag{3.29}$$

Nessas condições temos o seguinte resultado geral:

Teorema 3.1. (Lema de Céa) *Seja* $e := u - u_h$ *o erro absoluto de aproximação. Então, sob as hipóteses (3.29), temos*

$$a(e, v_h) = a(u - u_h, v_h) = 0 \quad e \quad \|u - u_h\|_V \le \delta \|u - v_h\|_V,$$

para todo $v_h \in V_m$, *onde* $\delta = \delta_1/\delta_2$.

Observação: A primeira condição significa que o erro é ortogonal ao subespaço V_m, com respeito a $a(\cdot, \cdot)$, ou ainda que u_h é a projeção ortogonal de u no subespaço V_m, com respeito a $a(\cdot, \cdot)$. A segunda condição significa que a solução u_h obtida pelo método de Galerkin é a melhor aproximação para u com respeito à norma $\|\cdot\|_V$ e, consequentemente, uma condição suficiente para a convergência é a existência de uma família V_m de subespaços do espaço V tal que, para cada $u \in V$, $\lim_{h\to 0} \inf_{v_h \in V_m} \|u - v_h\|_V = 0$.

Demonstração: Como $V_m \subset V$, a condição

$$a(u, v_h) = \langle f, v_h \rangle, \quad \forall\, v_h \in V_m. \tag{3.30}$$

é imediata. Assim, subtraindo (3.30) de (3.28) e usando a bilinearidade de $a(\cdot, \cdot)$ obtemos,

$$a(u - u_h, v_h) = a(e, v_h) = 0 \qquad \forall\, v_h \in V_m. \tag{3.31}$$

Por outro lado, como $a(\cdot, \cdot)$ é coerciva e bilinear temos

$$\begin{aligned} \delta_2 \|u - u_h\|_V^2 &\le a(u - u_h, u - v_h + v_h - u_h) \\ &\le a(u - u_h, u - v_h) + a(u - u_h, v_h - u_h) \\ &= a(u - u_h, u - v_h), \end{aligned}$$

onde usamos a propriedade de ortogonalidade (3.31). Da continuidade (3.29) obtemos

$$\|u - u_h\|_V \leq \delta \|u - v_h\|_V, \qquad \forall\ v_h \in V_m. \tag{3.32}$$

onde $\delta = \delta_1/\delta_2 > 0$. □

Corolário 3.1. *Se além das hipóteses em (3.29), a forma bilinear $a(\cdot,\cdot)$ é simétrica em V, i.e., $a(u,v) = a(v,u)$ para todo $u, v \in V$, então*

$$\|u - u_h\|_V \leq \sqrt{\delta}\|u - v_h\|_V \qquad \forall\, v_h \in V_m. \tag{3.33}$$

Demonstração: Considerando a simetria de $a(\cdot,\cdot)$, temos

$$\begin{aligned} a(u - v_h, u - v_h) &= a(u - u_h + u_h - v_h, u - u_h + u_h - v_h) \\ &= a(u - u_h, u - u_h) + 2a(u - u_h, u_h - v_h) + a(u_h - v_h, u_h - v_h). \end{aligned}$$

Da coercividade (3.29) e ortogonalidade (3.31) obtém-se

$$a(u - u_h, u - u_h) \leq a(u - v_h, u - v_h) \qquad \forall\, v_h \in V_m.$$

Da continuidade e coercividade (3.29),

$$\delta_2 \|u - u_h\|_V^2 \leq \delta_1 \|u - v_h\|_V^2,$$

o que é equivalente a

$$\|u - u_h\|_V \leq \sqrt{\delta}\|u - v_h\|_V.$$

Logo, quando a forma bilinear $a(\cdot,\cdot)$ é simétrica, temos uma melhor aproximação co relação à obtida por (3.32). □

Veja que o Lema de Céa mostra essencialmente que o erro $\|u - u_h\|_V$ é a distância $d(u, V_m) = \inf_{v_h \in V_m} \|u - v_h\|_V$ entre a função $u \in V$ e o subespaço $V_m \subset V$.

Corolário 3.2. *Nas mesmas condições do Corolário 3.1, temos*

$$\|u_h\|_V \leq \sqrt{\delta}\|u\|_V. \tag{3.34}$$

Demonstração:

$$\begin{aligned} a(u,u) &= a(u - u_h + u_h, u - u_h + u_h) = a(e + u_h, e + u_h) \\ &= a(u_h, u_h) + a(e,e) + 2a(e, u_h) = a(u_h, u_h) + a(e,e). \end{aligned}$$

Como $a(e, u_h) = 0$, conclui-se que $0 < a(e,e) = a(u,u) - a(u_h, u_h)$, e portanto $a(u_h, u_h) \leq a(u,u)$. Usando a continuidade e coercividade (3.29) obtemos

$$\|u_h\|_V \leq \sqrt{\delta}\|u\|_V. \quad \square$$

Os resultados anteriores nos permitem afirmar que, de todas as funções $v_h \in V_m$, a solução aproximada $u_h \in V_m$ do problema dado pelo método de Galerkin é a que melhor se aproxima da função exata $u \in V$ na norma V. Este resultado ainda é insuficiente para quantificar esta proximidade, que obviamente depende da escolha do subespaço V_m.

A seguir, usando os resultados anteriores, determinaremos estimativas para a convergência de u_h quando $h \to 0$, em termos de normas de espaços de Sobolev, ou seja, determinaremos estimativas em função de h para $\|u - u_h\|_1$ e $\|u - u_h\|_0$. Para isso, como veremos a seguir, faz-se necessário inicialmente obter estimativas de erro para a função de interpolação.

3.2.1 Erro de Interpolação

Consideremos o problema variacional

$$a(u, v) = \langle f, v \rangle, \quad \forall\, v \in V = H_0^1(0, 1), \tag{3.35}$$

onde $f \in V = H^{-1}(0, 1)$ e

$$a(u, v) = \alpha \int_0^1 u'(x)v'(x)\, dx + \beta \int_0^1 u(x)v(x)\, dx. \tag{3.36}$$

Observação: Se um polinômio qualquer $\widetilde{u}(x)$ de grau k interpola a solução exata u nos pontos nodais $\{x_i\}$, i.e, $u(x_i) = \widetilde{u}(x_i)$ em cada intervalo $I_i = [x_i, x_{i+1}]$, $i = 0, 1, \ldots m$, dizemos que $\widetilde{u}$ é um "interporlante". Por exemplo, no caso do problema (3.35), consideramos o subespaço V_m gerado pelas funções lineares por partes (2.19) e, tendo como base a função $\varphi_i(x)$, definimos

$$\widetilde{u}(x) = \sum_{i=1}^{m} u_i \varphi_i(x), \tag{3.37}$$

onde $u_i = u(x_i)$ é o valor da solução exata nos pontos nodais. Sendo $\widetilde{u}$ um polinômio de grau $k = 1$ em cada elemento finito I_i, $\widetilde{u}$ é um interpolante.

Devemos chamar a atenção para que não se confunda o interpolante $\widetilde{u}$ com a solução aproximada u_h, obtida variacionalmente pelo método de Galerkin.

No resultado seguinte, estabelecemos uma estimativa de erro de interpolação na norma de $H^1(0, 1)$ entre a solução exata do problema (3.35) e o interpolante definido por (3.37).

Teorema 3.2. *Sejam $f \in L^2(0,1)$, $u \in H_0^1(0,1) \cap H^2(0,1)$ a solução exata do problema (3.35) e $\widetilde{u}$ o interpolante definido por (3.37). Então,*

$$\|u - \widetilde{u}\|_1 = \|u - \widetilde{u}\|_0 + \|u' - \widetilde{u}'\|_0 \leq \frac{h}{\pi}\left(1 + \frac{h^2}{\pi^2}\right)^{\frac{1}{2}} \|u\|_2, \tag{3.38}$$

Demonstração: No que segue, denotamos $\widetilde{u}_i$ a restrição de $\widetilde{u}$ em cada intervalo I_i e $\|\cdot\|_{0,I_i}$ a norma de $L^2(I_i)$, isto é,

$$\|v\|_{0,I_i}^2 = \int_{x_i}^{x_{i+1}} |v(s)|^2 \, ds$$

Com isso, podemos escrever

$$\|u - \widetilde{u}\|_1^2 = \sum_{i=1}^{m} \left(\|u - \widetilde{u}_i\|_{0,I_i}^2 + \|u' - \widetilde{u}_i'\|_{0,I_i}^2\right)$$

Seja $w = u - \widetilde{u}$ e $w_i = w|_{I_i}$. Sendo $\widetilde{u}_i$ um interpolante de u em I_i, temos que $w_i(x_i) = w_i(x_{i+1}) = 0$. Desta forma, podemos considerar a expansão w_i em série de Fourier de senos, i.e,

$$w_i(x) = \sum_{m=1}^{\infty} c_{mi} \ \text{sen}\, \frac{m\pi(x - x_i)}{h}, \quad \forall\, c_{mi} \in \mathbb{R}$$

Fazendo a troca de variável $y = x - x_i$, obtemos que

$$\|w_i\|_{0,I_i}^2 = \int_{x_i}^{x_{i+1}} w_i(x)^2 \, dx = \int_0^h w_i(y)^2 \, dy = \frac{h}{2} \sum_{m=1}^{\infty} (c_{mi})^2.$$

A última igualdade é conhecida como identidade de Parseval, que é uma consequência da ortogonalidade das funções trigonométricas da série, i.e.,

$$\int_0^h \cos\frac{m\pi y}{h} \ \cos\frac{n\pi y}{h} \, dy = \int_0^h \text{sen}\, \frac{m\pi y}{h} \ \text{sen}\, \frac{n\pi y}{h} \, dy = \frac{h}{2}\delta_{mn}, \quad \forall m, n \in \mathbb{N}.$$

Temos também que

$$w_i'(x) = w_i'(y) = \sum_{m=1}^{\infty} c_{mi} \frac{m\pi}{h} \ \cos\frac{m\pi y}{h}, \quad \forall c_{mi} \in \mathbb{R}.$$

Logo, usando novamente a ortogonalidade anterior obtemos,

$$\|w_i'\|_{0,I_i}^2 = \int_{x_i}^{x_{i+1}} w_i'(x)^2 \, dx = \frac{h}{2} \sum_{m=1}^{\infty} (c_{m_i})^2 \ \frac{m^2\pi^2}{h^2}$$

e de forma análoga,

$$\|w_i''\|_{0,I_i}^2 = \int_{x_i}^{x_{i+1}} w_i''(x)^2\, dx = \frac{h}{2}\sum_{m=1}^{\infty}(c_{m_i})^2\, \frac{m^4\pi^4}{h^4}.$$

Note que, sendo a função interpolante $\widetilde{u}_i$ linear em cada intervalo $I_i = [x_i, x_{i+1}]$, temos $\widetilde{u}_i'' = 0$ e então $w_i''(x) = u''(x)$, para todo $x \in I_i$. Logo, como $m \geq 1$, obtém-se

$$\|w_i'\|_{0,I_i}^2 \leq \sum_{m=1}^{\infty}(c_{m_i})^2\, \frac{m^4\pi^2}{h^2} = \frac{h^2}{\pi^2}\|u''\|_{0,I_i}^2, \quad i = 1, 2, \ldots, m.$$

Fazendo o somatório em i de 1 a m, obtemos

$$\sum_{i=1}^{m}\|w_i'\|_0^2 = \|u' - \widetilde{u}'\|_0^2 \leq \frac{h^2}{\pi^2}\|u''\|_0^2.$$

Extraindo a raiz quadrada, obtemos a estimativa de erro para a derivada w_i' na norma $L^2(0,1)$,

$$\|w'\|_0 = \|u' - \widetilde{u}'\|_0 \leq \frac{h}{\pi}\|u''\|_0. \tag{3.39}$$

Diz-se neste caso que o erro é de ordem h e denota-se $\mathcal{O}(h)$.

De forma análoga, temos

$$\|w_i\|_{0,I_i}^2 = \frac{h}{2}\sum_{m=1}^{\infty}(c_{mi})^2 \leq \frac{h}{2}\sum_{m=1}^{\infty}(c_{mi})^2 m^4 = \frac{h^4}{\pi^4}\|u''\|_{0,I_i}^2, \qquad i = 1, 2, \ldots, m.$$

Logo, somando em i de 1 até m, obtemos

$$\sum_{i=1}^{m}\|w_i\|_0^2 = \|u - \widetilde{u}\|_0^2 \leq \frac{h^4}{\pi^4}\|u''\|_0^2.$$

Extraindo a raiz quadrada, obtemos a estimativa de erro para w_i na norma $L^2(0,1)$,

$$\|w\|_0 = \|u - \widetilde{u}\|_0 \leq \frac{h^2}{\pi^2}\|u''\|_0. \tag{3.40}$$

Diz-se neste caso que o erro de interpolação é de ordem h^2 e denota-se $\mathcal{O}(h^2)$.

Das desigualdades (3.39) e (3.40), conclui-se a estimativa de erro na norma $H^1(0,1)$,

$$\|w\|_1^2 = \|u - \widetilde{u}\|_1^2 = \|u - \widetilde{u}\|_0^2 + \|u' - \widetilde{u}'\|_0^2 \leq \left(\frac{h^4}{\pi^4} + \frac{h^2}{\pi^2}\right)\|u''\|_0^2,$$

ou seja,

$$\|w\|_1 = \|u - \widetilde{u}\|_1 = \|u - \widetilde{u}\|_0 + \|u' - \widetilde{u}'\|_0 \leq \frac{h}{\pi}\left(1 + \frac{h^2}{\pi^2}\right)^{\frac{1}{2}} \|u\|_2, \tag{3.41}$$

onde $\|\cdot\|_2$ denota a norma de $H^2(0,1)$ e por definição, $\|u\|_2 \geq \|u''\|_0$. □

Note que a estimativa de erro em $H^1(0,1)$ é de ordem $\mathcal{O}(h)$ e somente não é ordem $\mathcal{O}(h^2)$ em razão da estimativa da derivada (3.39).

A seguir demonstramoss as estimativas de erros entre a solução exata u e a solução aproximada u_h obtida pelo método de Galerkin.

3.2.2 Erro na Norma $H^1(\Omega)$

Vimos no Teorema 3.1 que, de um modo geral, a solução u_h, projeção ortogonal de u com respeito a forma $a(\cdot,\cdot)$ no espaço V_m, satisfaz a desigualdade (3.32) dada por

$$\|u - u_h\|_V \leq \delta \|u - v_h\|_V, \qquad \forall\ v_h \in V_m,$$

onde $\delta = \delta_1/\delta_2$, com $\delta_1 = \max\{\alpha, \beta\}$, $\delta_2 = \min\{\alpha, \beta\}$ e $V = H_0^1(0,1)$. Tomando em particular $v_h = \widetilde{u} \in V_m$, obtemos a estimativa de erro na norma $H^1(0,1)$,

$$\|u - u_h\|_1 \leq \delta \|u - \widetilde{u}\|_1 \leq \frac{\delta h}{\pi}\left(1 + \frac{h^2}{\pi^2}\right)^{\frac{1}{2}} \|u\|_2, \tag{3.42}$$

com erro de ordem $\mathcal{O}(h)$. Por outro lado, usando (2.7), obtemos a seguinte estimativa em termos da função f, dada por

$$\|u - u_h\|_1 \leq \delta \|u - \widetilde{u}\|_1 \leq \frac{c\delta h}{\pi}\left(1 + \frac{h^2}{\pi^2}\right)^{\frac{1}{2}} \|f\|_0, \tag{3.43}$$

com erro de ordem $\mathcal{O}(h)$.

No caso específico do problema (3.35), sendo a forma bilinear $a(\cdot,\cdot)$ simétrica, segue de (3.33) a mesma estimativa, substituindo-se δ por $\sqrt{\delta}$.

3.2.3 Erro na Norma $L^2(\Omega)$

A desigualdade (3.40) mostra que a estimativa do erro de interpolação na norma $L^2(0,1)$ é de ordem $\mathcal{O}(h^2)$ com a dependência da norma de u'' em $L^2(0,1)$. Já a desigualdade (3.43) fornece uma estimativa de ordem $\mathcal{O}(h)$ na norma de $H^1(0,1)$ e somente a imersão contínua de $H^1(0,1)$ em $L^2(0,1)$ não melhora em nada as estimativas

já obidas. Entretanto, em razão da estimativa (3.40), é razoável acreditar ser quadrática a ordem de convergência $\|u-u_h\|_0$ sem a dependência de $\|u''\|_0$. De fato, podemos provar isso usando um argumento devido a Nitsche, como veremos a seguir.

Teorema 3.3. *Sejam $f \in L^2(0,1)$, $u \in H_0^1(0,1) \cap H^2(0,1)$ a solução do problema (3.35) e $u_h \in V_m$ a solução aproximada. Então, u_h converge para a solução u na norma $L^2(0,1)$ com ordem de convergêucia quadrática em relação a h. Mais precisamente,*

$$\|u-u_h\|_0 \leq \widehat{C}h^2\|f\|_0, \tag{3.44}$$

onde $\widehat{C}$ é uma constante positiva independente de h e f.

Demonstração: Pela desigualdade (3.43) (para todo $h \leq \pi$),

$$\|u-u_h\|_1 \leq C_0 h\|f\|_0, \tag{3.45}$$

onde $C_0 = c\sqrt{2\delta}/\pi > 0$ independe de h.

Consideremos w e w_h respectivamente soluções dos seguintes problemas variacionais:

$$\begin{aligned} a(w,v) &= (u-u_h,v), \quad \forall v \in V, \\ a(w_h,v_h) &= (u-u_h,v), \quad \forall v \in V_m, \end{aligned} \tag{3.46}$$

de modo que, novamente da desigualdade (3.43), temos

$$\|w-w_h\|_1 \leq C_0 h\|u-u_h\|_0. \tag{3.47}$$

Substituindo v por v_h na primeira identidade de (3.46) e subtraindo da segunda, obtemos $a(w-w_h,v_h)=0$, para todo $v_h \in V_m$. Em particular, $a(w-w_h,u_h)=0$. Assim, temos da primeira identidade de (3.46) com $v = u-u_h$ e de (3.47),

$$\begin{aligned} \|u-u_h\|_0^2 = a(w-w_h,u-u_h) &\leq \|w-w_h\|_1\|u-u_h\|_1 \\ &\leq C_0^2 h^2\|u-u_h\|_0\|f\|_0. \end{aligned}$$

Portanto, para $\widehat{C} = C_0^2$, temos $\|u-u_h\|_0 \leq \widehat{C}h^2\|f\|_0$, como queríamos provar. □

Note que ainda podemos ter problemas de convergência devido a erros numéricos quando α é muito pequeno. De fato, considerando em particular $\alpha = \varepsilon > 0$ e $\beta = 1$, então de (3.43) temos:

$$\|u-u_h\|_1 \leq \frac{c_1}{\varepsilon}h\|f\|_0,$$

onde $c_1 \geq 0$ é uma constante positiva. Neste caso o método somente terá uma boa aproximação se $h \ll \varepsilon$, o que muitas vezes é inviável em situações práticas.

Como consequência imediata dos rsultado anteriores, temos a convergência da solução aproximada u_h para a solução exata u quando $h \to 0$. Mais precisamente,

Corolário 3.3. *Se $f \in L^2(0,1)$ então a solução aproximada u_h do problema (2.9) converge para a solução $u \in H^2(0,1) \cap H_0^1(0,1)$ do problema (2.6), i.e,*

$$\lim_{h\to 0}\|u-u_h\|_1 = 0 \quad e \quad \lim_{h\to 0}\|u-u_h\|_0 = 0.$$

3.2.4 Erro na Norma $H^m(\Omega)$

Vimos nos Teoremas 3.2 e 3.3 que as estimativas de erro nas normas do $H^1(0,1)$ e $L^2(0,1)$ são respectivamente de ordem $\mathcal{O}(h)$ e $\mathcal{O}(h^2)$. Veremos a seguir que a ordem de convergência pode ser melhor se a solução exata for mais regular, i.e., se pertencer a um espaço mais regular que o espaço $H_0^1(0,1) \cap H^2(0,1)$. Nesta seção enunciaremos alguns resultados gerais de estimativa de erro em normas de espaço de Sobolev, que dependem da regularidade da solução exata e do grau do polinômio interpolador, cujas demonstrações podem ser encontradas em [5, 16, 24].

Teorema 3.4. *Se $u \in H^m(0,1)$, $m \geq 0$, então existe uma função interpolante $\widetilde{u} \in V_m^k$, tal que para todo s, $0 \leq s \leq m$,*

$$\|u - \widetilde{u}\|_s \leq ch^{\alpha}\|u\|_m, \tag{3.48}$$

onde $\alpha = \min\{k+1-s, m-s\}$, k é o grau do polinômio interpolador e c é uma constante independente de u e de h.

Corolário 3.4. (Estimativa Ótima) *Se $u \in H^{k+1}(0,1)$ e $\widetilde{u} \in V_m^k$, então*

$$\|u - \widetilde{u}\|_m \leq ch^{k+1-m}\|u\|_{k+1}, \tag{3.49}$$

onde $k \geq 1$ é o grau do polinômio interpolador da função base que gera o subespaço vetorial de dimensão finita e $m \leq k$.

A consequência imediata do Corolário 3.4 é que a estimativa depende da regularidade da solução. Isso significa, por exemplo, que se a solução u do problema pertence ao espaço $H^2(0,1)$, as melhores estimativas a serem obtida são as de ordem $\mathcal{O}(h)$ e $\mathcal{O}(h^2)$ respectivamente nas normas do $H^1(\Omega)$ e $L^2(\Omega)$, independente do grau k do polinômio interpolador.

Pode-se mostrar que é válido o seguinte resultado de regularidade para o problema (2.1) (ver [2]): *se $f \in H^m(0,1)$, com $m \in \mathbb{N}$, $m \geq 1$, a solução $u \in H^{m+2}(0,1)$ e vale a desiguladade $\|u\|_{m+2} \leq \|f\|_m$.*

Assim, vemos que a estimativa depende do polinômio interpolador P_k, ou seja, para os polinômios lineares por partes ($k = 1$), quadráticos ($k = 2$) e cúbicos ($k = 3$), obtém-se, respectivamente, as estimativas em $L^2(0,1)$ e $H^1(0,1)$:

$$\|u - \widetilde{u}\|_0 \leq ch^2\|u\|_2, \quad \|u - \widetilde{u}\|_0 \leq ch^3\|u\|_3, \quad \|u - \widetilde{u}\|_0 \leq ch^4\|u\|_4,$$

$$\|u - \widetilde{u}\|_1 \leq ch^1\|u\|_2, \quad \|u - \widetilde{u}\|_1 \leq ch^2\|u\|_3, \quad \|u - \widetilde{u}\|_1 \leq ch^3\|u\|_4.$$

Para a conclusão da análise do erro precisamos estabelecer uma relação entre o erro $\|u - u_h\|_m$ e $\|u - \widetilde{u}\|_m$, onde u é a solução exata, u_h é a solução aproximada e $\widetilde{u}$ é

a função interpolante. Para isso, é suficiente tomar $\widetilde{u} = v_h \in V_m$ no Teorema (3.1), obtendo-se:

$$\|u - u_h\|_m \leq \delta \|u - \widetilde{u}\|_m. \tag{3.50}$$

Portanto, todas as estimativas anteriores são válidas para estimar o erro entre a solução exata e a solução aproximada obtida pelo método de Galerkin, ou seja, de (3.49) e (3.50) obtemos:

$$\|e\|_m = \|u - u_h\|_m \leq \delta \|u - \widetilde{u}\|_m \leq C h^{k+1-m} \|u\|_{k+1}, \tag{3.51}$$

onde $C = c\delta$.

3.3 Erro Numérico

Os resultados de convergência obtidos são as estimativas de erro entre a solução exata e a solução numérica aproximada. Como em geral a solução exata não é conhecida, vamos intoduzir nesta seção um dos procedimentos usuais que permitem estimar a "exatidão" dos resultados numéricos, fazendo um paralelo com aos resultados teóricos de estimativa de erro.

Como em geral não conhecemos a solução exata u, tomemos em seu lugar a solução numérica u_N calculada, com N suficientemente grande (ou equivalentemente, h suficientemente pequeno), onde N é o número de nós da discretização.

Procedemos em seguida nas seguintes etapas:

(1) construímos uma sucessão de soluções numéricas $\{u_i\}$ associadas a diversos tamanhos de malha h_i, para os quais calculamos os respectivos erros $\|\hat{E}_i\|_m = \|u_i - u_N\|_m$, para $i = 1, 2, \cdots (N-1)$.

(2) observamos a estimativa (3.51) associada a solução u_N

$$\|\hat{E}_i\|_m = \|u_i - u_N\|_m \leq c_1 h^{k+1-m} \|u_N\|_{k+1} \approx c_2 h_i^p, \quad \text{para} \quad i = 1, 2, \cdots, (N-1), \tag{3.52}$$

onde $p = (k + 1 - m)$ é a taxa de convergência, sendo $\|u_N\|_{k+1}$ é um número real conhecido para algum k.

Podemos então escrever

$$\alpha_i = \frac{\|\hat{E}_i\|_m}{\|\hat{E}_{i+1}\|_m} = \left(\frac{h_i}{h_{i+1}}\right)^p, \qquad \forall i = 1, 2, \cdots, \tag{3.53}$$

onde $h_{i+1} < h_i$.

Através da relação (3.53), dizemos que p é a **taxa de convergência numérica**. Na teoria o valor de p é constante, mas na prática obtemos valores aproximados para p,

dependentes da sucessão i, que pode ser comparado com o resultado teórico estimado. O número p pode ser explicitamente calculado por

$$p = \frac{\ln \alpha_i}{\ln\left(\frac{h_i}{h_{i+1}}\right)}, \qquad \text{para} \quad i = 1, 2, \cdots. \tag{3.54}$$

• **Exemplo** (Construção da sucessão da solução numérica)

Considere as soluções numéricas aproximadas $u_1, u_2, \cdots u_{N/2}$, obtidas usando o passo $h_i = (5 \times 2^{i+1})^{-1}$, ou seja, $h_i = 2h_{i+1}$, $i = 0, 1, \cdots (N-2)/2$. Considere $\hat{E}_i$ o erro associado a uma norma, comparando as soluções u_i com u_N. Para essa sucessão temos, por exemplo:

$$p = \frac{\ln \alpha_i}{\ln\left(\frac{h_i}{h_{i+1}}\right)} = \frac{\ln \alpha_i}{\ln 2}, \quad \text{para} \quad i = 0, 1, \cdots, (N-2)/2 \tag{3.55}$$

3.4 Exercícios

1. Determine os elementos da matriz local do problema (3.13), usando os polinômios de Hermite como função base e obtenha um algoritmo para a determinação da matriz global e da força global.

2. Usando o exercício anterior, dê uma estimativa de erro nas normas $L^2(0,1)$ e $H^1(0,1)$.

3. Na interpolação linear, vimos que a solução aproximada $u_h(x_i)$ era exatamente o valor de d_i, solução do sistema sistema linear $K\boldsymbol{d} = \boldsymbol{F}$. Considerando o polinômio interpolador quadrático, qual é a relação entre $u_h(x_i)$ e a componente d_i do vetor $\boldsymbol{d}$ do sistema linear $K\boldsymbol{d} = \boldsymbol{F}$?

4. Mostre que o conjunto de funções B-splines $\{\varphi_1, \varphi_2, \cdots, \varphi_m, \varphi_{m+1}\}$ definidas em (3.8) formam um conjunto linearmente dependente.

   Sugestão: Mostre que o sistema $A\boldsymbol{d} = \boldsymbol{0}$ tem infinitas soluções, quando consideradas as duas funções no conjunto.

CAPÍTULO 4

Problema Estacionário Bidimensional

Neste capítulo vamos tratar problemas de contorno referentes a equações elípticas lineares no caso bidimensional. Como motivação, vamos nos referir aos problemas como modelos de transferência de calor em regime estacionário numa placa condutora. São apresentadas as formulações forte e fraca, assim como estudadas as condições para existência e unicidade de solução em alguns tipos de condição de fronteira. Apresentamos também um programa utilizando o método de elementos finitos para a solução numérica da equação e analisamos a influência das condições de fronteira no sistema linear associado à discretização. O método de Crout é o utilizado para resolver o sistema linear, com exemplos numéricos e ilustração gráfica apresentados, assim como os erros da solução aproximada, utilizando diversos tamanhos de malha. As etapas da elaboração do programa em linguagem C estão no final do capítulo e o código no apêndice deste texto.

4.1 Formulação do Problema

Seja $\Omega \subset \mathbb{R}^2$ um conjunto aberto com fronteira suave Γ tal que

$$\Gamma = \overline{\Gamma_q \cup \Gamma_p}, \quad \Gamma_q \cap \Gamma_p = \emptyset.$$

Os pontos de Ω são denotados por x e o vetor normal unitário exterior a Γ é denotado por n.

Denotamos por $q_i = q_i(x)$ e $u = u(x)$ respectivamente a i-ésima componente do fluxo de calor e a temperatura no pont x. A fonte externa de calor por unidade de volume é denotada por $f = f(x)$ e admitimos que o fluxo de calor é dado pela lei de Fourier, i.e.,

$$q_i = -Q_{ij}\frac{\partial u}{\partial x_j},$$

onde Q_{ij} é uma matriz simétrica definida positiva em todo $x \in \Omega$, denominada *condutividade térmica*. Quando o corpo é homogêneo, a matriz Q_{ij} é constante (independe de $x \in \Omega$). No caso isotrópico $Q_{ij}(x) = Q(x)\delta_{ij}$.

Nas condições acima e usando a convenção de somatório, a questão que se coloca é a do seguinte problema: *dadas as funções* $f : \Omega \to \mathbb{R}$, $q : \Gamma_q \to \mathbb{R}$ *e* $p : \Gamma_p \to \mathbb{R}$, *determinar* $u : \overline{\Omega} \to \mathbb{R}$ *tal que*

$$\begin{cases} -\dfrac{\partial}{\partial x_i}\Big(Q_{ij}\dfrac{\partial u}{\partial x_j}\Big) = f, & em \ \ \Omega, \\ u = q, & em \ \ \Gamma_q, \\ -q_i n_i = Q_{ij}\dfrac{\partial u}{\partial x_j} n_i = p, & em \ \ \Gamma_p. \end{cases} \tag{4.1}$$

Referimo-nos ao problema (4.1) como a *formulação forte* do problema do calor, onde se deseja que a solução $u(x)$ tenha regularidade suficiente para que as derivadas na formulação acima sejam entendidas no sentido clássico, como por exemplo se $u \in C^2(\overline{\Omega})$, ou mais geralmente, no sendido $L^2(\Omega)$.

Quando o corpo é isotrópico e homogêneo, a equação em (4.1) toma a forma

$$-\Delta u = f, \tag{4.2}$$

que é a bem conhecida equação de Poisson. Se $\Gamma = \Gamma_q$, a condição de fronteira é do tipo Dirichlet; se $\Gamma = \Gamma_p$, a condição de fronteira é do tipo Neumann e se $\Gamma = \overline{\Gamma_p \cup \Gamma_q}$ com $\Gamma_p \neq \emptyset$ e $\Gamma_q \neq \emptyset$, a condição de fronteira é do tipo misto.

4.1.1 Formulação Fraca

Com as funções f, p e q acima, consideremos o espaço

$$V = \{v \in H^1(\Omega);\ v = 0 \text{ em } \Gamma_q\}$$

e o subespaço afim $H = \{v \in H^1(\Omega);\ u = q \text{ em } \Gamma_q\}$. Definimos a forma bilinear a e a forma linear F,

$$a : V \times V \to \mathbb{R} \quad \text{e} \quad F : V \to \mathbb{R},$$

respectivamente por

$$a(u,v) = \int_\Omega Q_{ij}\frac{\partial u}{\partial x_j}\frac{\partial v}{\partial xi}\,dx \quad \text{e} \quad F(v) = \int_\Omega fv\,dx + \int_{\Gamma_p} pv\,dx.$$

A formulação fraca do problema (4.1) é a seguinte: *determinar* $u \in H$ *tal que*

$$a(u,v) = F(v), \quad \forall v \in V. \tag{4.3}$$

No que segue mostramos que as duas formulações são equivalentes sob condições de regularidade, isto é, vamos admitir a regularidade das funções que justifiquem os cálculos formais abaixo.

Primeiramente mostremos que (4.1) $\Rightarrow$ (4.3). De fato, multiplicando a primeira equação de (4.1) por $v \in H^1(\Omega)$ e integrando em Ω, segue do Teorema da Divergência,

$$\int_\Omega Q_{ij}\frac{\partial u}{\partial x_j}\frac{\partial v}{\partial x_i}\,dx - \int_\Gamma Q_{ij}\frac{\partial u}{\partial x_j}n_i v\,d\Gamma = \int_\Omega fv\,dx. \tag{4.4}$$

Em particular, para todo $v \in V$, a identidade acima se reduz a

$$\int_\Omega Q_{ij}\frac{\partial u}{\partial x_j}\frac{\partial v}{\partial x_i}\,dx - \int_{\Gamma_p} Q_{ij}\frac{\partial u}{\partial x_j}n_i v\,d\Gamma = \int_\Omega fv\,dx$$

e em vista da condição de fronteira em Γ_p, temos

$$\int_\Omega Q_{ij}\frac{\partial u}{\partial x_j}\frac{\partial v}{\partial x_i}\,dx - \int_{\Gamma_p} pv\,d\Gamma = \int_\Omega fv\,dx, \quad \forall v \in V,$$

o que corresponde à formulação fraca (4.3).

Reciprocamente, para mostrar que (4.3) $\Rightarrow$ (4.1), seja u solução regular do problema (4.3). Então, pelo Teorema da Divergência, temos

$$-\int_\Omega \frac{\partial}{\partial x_i}\left(Q_{ij}\frac{\partial u}{\partial x_j}\right)v\,dx + \int_\Gamma Q_{ij}\frac{\partial u}{\partial x_j}n_i v\,d\Gamma - \int_\Omega fv\,dx - \int_{\Gamma_p} pv\,d\Gamma = 0, \quad \forall v \in V. \tag{4.5}$$

Em particular, para todo $v = 0$ na fronteira Γ, obtemos

$$-\int_\Omega \frac{\partial}{\partial x_i}\left(Q_{ij}\frac{\partial u}{\partial x_j}\right)v\,dx - \int_\Omega fv\,dx = 0,$$

de modo que u satisfaz necessariamente a equação

$$-\frac{\partial}{\partial x_i}\left(Q_{ij}\frac{\partial u}{\partial x_j}\right) - f = 0 \quad \text{em} \quad \Omega$$

e a identidade (4.5) se reduz a

$$\int_{\Gamma_p}\left(Q_{ij}\frac{\partial u}{\partial x_j}n_i - p\right)v\,d\Gamma = 0, \quad \forall v \in V.$$

Assim, temos necessariamente a condição

$$Q_{ij}\frac{\partial u}{\partial x_j}n_i = p \quad \text{em} \quad \text{em } \Gamma_p.$$

Observação: Os cálculos formais acima se justificam para $u \in H$ e $v \in V$ pela teoria dos Espaços do Sobolev, onde as integrais de fronteira acima são definidas em $H^{1/2}(\Gamma_q)$ e $H^{-1/2}(\Gamma_p)$ pelo Teorema do Traço.

4.1.2 Existência e Unicidade de Solução

Mostraremos a existência e a unicidade da solução do problema (4.3), quando Q_{ij}, f, p e q são funções regulares. Considere para todo $v \in V$,

$$\begin{aligned} a(v,u) &= \int_\Omega Q_{ij}\frac{\partial u}{\partial x_j}\frac{\partial v}{\partial x_i}\,dx, \\ (v:f) &= \int_\Omega fv\,dx, \\ (v:p)_\Gamma &= \int_{\Gamma_p} pv\,d\Gamma, \end{aligned} \tag{4.6}$$

de modo que a formulação fraca possa ser expressa na forma $a(v,u) = (v:f)+(v:p)_\Gamma$, $\forall v \in V$, onde, por hipótese, a matriz condutividade Q_{ij} é simétrica e definida positiva.

Considere a norma do subespaço V de $H^1(\Omega)$ dada por

$$\|v\|_V^2 = \int_\Omega |\nabla v(x)|^2\,dx.$$

Pode-se mostrar que se a medida superficial de Γ_q é positiva, as normas $\|\,.\,\|_V$ e $\|\,.\,\|_{H^1(\Omega)}$ são equivalentes no espaço V. Com isso, pode-se mostrar que a forma bilinear $a : V \times V \longrightarrow R$ satisfaz as condições de Lax-Milgram, isto é, para todo $u, v \in V$,

i) $a(v,u) = a(u,v)$, pois Q_{ij} é simétrica.

ii) $|a(v,u)| = |((v,u))_V| \leq C\|v\|_V\|u\|_V$.

iii) $|a(v,v)| = \displaystyle\int_\Omega \frac{\partial v}{\partial x_i}Q_{ij}\frac{\partial v}{\partial x_j}\,dx \geq C\int_\Omega |\nabla v(x)|^2\,dx = C\|v\|_V^2,$

Por outro lado, a aplicação linear $v \in V \longmapsto (v:f)+(v:p)_\Gamma \in \mathbb{R}$ é contínua para $p \in L^2(\Gamma_p)$ e $f \in L^2(\Omega)$.

Observação: Pela Teoria dos Espaço de Sobolev, as hipóteses que permitem garantir a validade das condições de Lax-Milgran são: $Q_{ij} \in L^\infty(\Omega)$, $f \in H^{-1}(\Omega)$ e $p \in H^{-1/2}(\Gamma_p)$. O leitor interessado pode consultar a bibliografia sobre Problemas Elíticos e Espaços de Sobolev.

4.1.3 Formulação de Galerkin

Seja V_h um subespaço de V de dimensão finita e consideremos $H_h = V_h + \{\widetilde{q}\}$, onde $\widetilde{q} \in V$ tal que $\widetilde{q}(x) = q(x)$ para todo $x \in \Gamma_q$. A existência de $\widetilde{q}$ é garantida pelo fato

de que o operador de traço $\gamma : H^1(\Omega) \to H^{1/2}(\Gamma_q)$ é sobrejetor. Assim, se $u_h \in H_h$, a função w_h definida por

$$w_h(x) = u_h(x) - \widetilde{q}(x), \tag{4.7}$$

pertence ao subespaço V_h, de modo que $v_h(x) = 0$ para $x \in \Gamma_q$.

Da mesma forma, definimos a função $\widetilde{p} \in H^1(\Omega)$ tal que $\widetilde{p}(x) = p(x)$ para $x \in \Gamma_p$. Nessas condições, podemos considerar o problema variacional aproximado associado ao problema (4.3), a saber,

$$a(u_h, v_h) = (f : v_h) + (p : v_h)_\Gamma, \quad \forall v_h \in V_h.$$

Por (4.7) temos,

$$a(w_h, v_h) = (f : v_h) + (p : v_h)_{\Gamma_p} - a(\widetilde{q}, v_h), \quad \forall v_h \in V_h. \tag{4.8}$$

Podemos então formular o seguinte problema:

Problema aproximado. *Dadas as funções f, p e q, queremos determinar a função $w_h \in V_h$ solução da formulação* (4.8).

Seja $\{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ uma base do subespaço V_h. Dessa forma, todo elemento $w_h \in V_h$ pode ser representado por

$$w_h = \sum_{j=1}^{m} C_j \varphi_j. \tag{4.9}$$

Substituindo em (4.8), tem-se

$$a\left(\sum_{j=1}^{m} C_j \varphi_j, v_h\right) = (f : v_h) + (p : v_h)_{\Gamma_p} - a(\widetilde{q}, v_h), \quad \forall v_h \in V_h.$$

A igualdade permanece válida tomando em particular $v_h = \varphi_i$, ou seja,

$$a\left(\sum_{j=1}^{m} C_j \varphi_j, \varphi_i\right) = (f : \varphi_i) + (\varphi_i, p)_{\Gamma - p} - a(\widetilde{q} : \varphi_i,).$$

Da linearidade da forma $a(\cdot, \cdot)$, segue que

$$\sum_{j=1}^{m} C_j a(\varphi_i, \varphi_j) = (f : \varphi_i) + (p : \varphi_i,)_{\Gamma_p} - a(\widetilde{q}, \varphi_i).$$

Denotando

$$K_{ij} := a(\varphi_i, \varphi_j), \quad 1 \leq i, j \leq m, \tag{4.10}$$

e

$$F_i = (f :, \varphi_i) + (p : \varphi_i)_{\Gamma_p} - a(\widetilde{q}, \varphi_i), \tag{4.11}$$

a formulação (4.8) pode ser escrita na forma matricial, $KC = F$.

A solução do sistema linear de ordem $m \times m$ permite calcular a solução numérica $u_h = w_h + \widetilde{q} \in H_h$ do problema aproximado. Assim, se m é suficientemenete grande, espera-se obter uma boa aproximação da solução $u \in H$ do problema contínuo. Por outro lado, se K é uma matriz cheia, sua ordem aumenta quadraticamente com relação a m, o que nos obriga a considerar o Método de Elementos Finitos que, como sabemos, a escolha conveniente da base $\{\varphi_1, \ldots, \varphi_m\}$ de V_h, fornece uma matriz de banda K de fácil resolução e que cresce linearmente com relação a m.

Entretanto, nas aproximações por elementos finitos, as funções $v_h \in V_h$ são, em geral, funções polinomiais por partes, de modo que as restrições sobre a fronteira podem ser incompatíveis com as condições impostas em H_h. Assim, devemos considerar um domínio aproximado Ω_h e redefinir o problema variacional (4.8), substituindo os espaços V_h e H_h de funções definidas em Ω por aquelas definidas sobre Ω_h.

4.2 Discretização do Domínio

De um modo geral, dado um domínio $\Omega \subset \mathbb{R}^n$, considera-se um domínio aproximadao Ω_h que possa ser dividido em subregiões Ω_e, de tal forma a satisfazer as condições:

$$\Omega_h = \left(\bigcup_{e=1}^{\texttt{Nel}} \overline{\Omega}_e \right)^{\circ} \qquad \text{e} \qquad \Omega_e \cap \Omega_k = \emptyset, \quad \text{se} \quad e \neq k,$$

onde `Nel` é o número total de elementos. Na partição do domínio definimos os nós globais A, $A = 1, 2, \ldots,$ `Nno`, onde `Nno` é o número total de nós da malha.

As subregiões Ω_e no caso bidimensional são geralmente triângulos ou retângulos, consistindo de 3 ou 4 nós locais para cada elemento finito Ω_e.

• Construção da Malha

Para ilustrar a geração da malha, consideremos o caso particular em que o domínio Ω é o retângulo $(a, b) \times (c, d)$. Os elementos finitos Ω_e que serão representados por e, também serão retângulos. O procedimento para a obtenção de elementos triangulares é feito de forma semelhante. Para obter os elementos retangulares, basta considerar partições de $[a, b]$ e de $[c, d]$, e fazer o produto cartesiano dos subintervalos. Há diversos tipos de malhas possíveis, tais como a malha geométrica, malha radical e em particular

a malha uniforme. Por simplicidade, para gerar a malha uniforme, consideremos o seguinte procedimento:

Define-se $h = (b-a)/\texttt{Nelx}$ e $k = (d-c)/\texttt{Nely}$, onde `Nelx` e `Nely` são os números de elementos nas direções x e y, respectivamente. Logo,

$$\begin{aligned} x_i &= x_0 + ih, \quad i = 1, 2, \ldots, \texttt{Nelx}, \quad \text{com} \quad x_0 = a, \\ y_j &= y_0 + jk, \quad j = 1, 2, \ldots, \texttt{Nely}, \quad \text{com} \quad y_0 = c \end{aligned}$$

Assim,

$$[a,b] \times [c,d] = \bigcup_{i=0}^{\texttt{Nelx}} [x_i, x_{i+1}] \times \bigcup_{j=0}^{\texttt{Nely}} [y_j, y_{j+1}]$$

A subrotina `DataInput` faz a geração da malha uniforme, onde em particular, tomamos $a = c = 0$ e $b = d = 1$. Dessa forma, $h = 1/\texttt{Nelx}$ e $k = 1/\texttt{Nely}$.

Para gerar malha não uniforme dentro da subrotina `DataInput`, basta entrar com as coordenadas x e y manualmente. A subrotina `DataInput` gera as posições (i, j) das coordenadas $(\texttt{x}[i], \texttt{y}[j])$, onde $i = 0, 1, \ldots, \texttt{Nelx}$ e $j = 0, 1, \ldots, \texttt{Nely}$.

• Nó Global $\mapsto$ Posição: Subrotina `NoPos`

O próximo passo é identificar o nó global A com a sua posição (i, j), obedecendo a enumeração sucessiva horizontal. A posição i do nó é o resto da divisão de $(A-1)$ por $(\texttt{Nelx}+1)$ e a posição j é o quociente da divisão de $(A-1)$ por $(\texttt{Nelx}+1)$, que serão representados na linguagem C por

$$(i, j) = \texttt{NoPos}(A),$$

onde

$$\begin{aligned} i &= (A-1)\%(\texttt{Nelx}+1), \\ j &= (A-1)/(\texttt{Nelx}+1). \end{aligned}$$

Por exemplo, considere a malha dada na Fig 4.1: Temos portanto 25 nós globais, 16 elementos (representados por um círculo) e $\texttt{Nelx} = \texttt{Nely} = 4$. Por exemplo, o nó global $A = 18$ é representado pelo par ordenado $(i, j) = (2, 3)$.

• Posição $\mapsto$ Nó Global: Subrotina `PosNo`

Esta subrotina faz o processo inverso da subrotina `NoPos`, isto é, dada a posição identifica-se o nó global A,

$$A = \texttt{PosNo}(i, j),$$

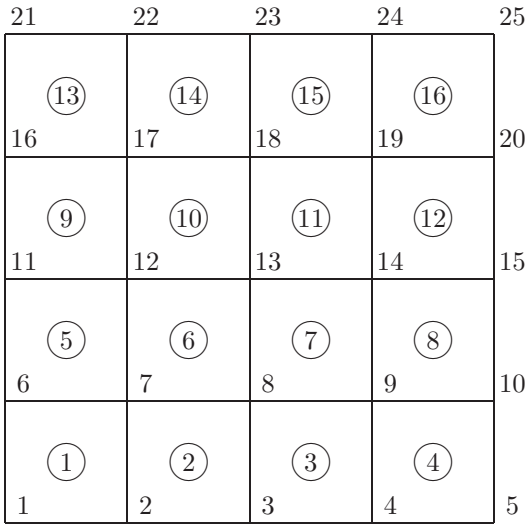


Figura 4.1: Malha de elementos retangulares

através da relação:

$$A = j(\texttt{Nelx} + 1) + i + 1.$$

Assim, na posição (1,4) temos o nó $A = 22$, considerando a malha anterior.

• **Elemento $\mapsto$ Posição: Subrotina** `ElmPos`

Para cada elemento e, temos 4 nós globais. Para identificá-los, é suficiente conhecer a posição de apenas um desses nós, que adotaremos neste caso como sendo o menor nó global (veremos adiante que este nó global corresponde ao nó local $a = 1$) do elemento e, dado pelo seguinte procedimento: dado um elemento e, a sua posição (i, j) é obtida pelo resto e o quociente da divisão. Assim, $(i, j) = \texttt{ElmPos}(e)$, onde $i = (e-1)\%\texttt{Nelx}$ e $j = (e-1)/\texttt{Nelx}$.

Por exemplo, considere o elemento $e = 8$ da malha anterior. Então, $(i, j) = (3, 1)$. Observando a tabela, verifica-se que na posição $(3, 1)$ temos o nó global $A = 9$ que é o representante do elemento $e = 8$. De forma análoga, temos $e = 11 \Rightarrow (i, j) = (2, 2)$, que é a posição do nó $A = 13$.

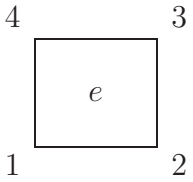
• **Nó Local $\mapsto$ Nó Global: Subrotina** `NoLG`

A subrotina `NoLG` identifica os nós locais $a = 1, 2, 3, 4$ de cada elemento (retângulo) com os nós globais A da malha,

$$A = \texttt{NoLG}(a, e)$$

da seguinte forma: dado um elemento e, o primeiro passo é identificar sua posição através da subrotina anterior, $(i, j) = \texttt{ElmPos}(e)$.

Sabemos, então, que na posição (i, j) existe um nó global A que representa o elemento e. O nó global A será então o primeiro nó local $a = 1$ do elemento e. Assim, $a = 1 \Leftrightarrow (i, j)$. Para $a = 2$, temos o nó global $(A + 1)$ que é identificado pela posição $(i + 1, j)$. Para $a = 3$, temos a posição $(i + 1, j + 1)$. Para $a = 4$, temos a posição $(i, j + 1)$. Geometricamente, temos



Com a posição identificada, o nó A pode ser encontrado pela função `PosNo`.

Considere a Fig. 4.1 e o elemento e=10. Então,

$$\texttt{NoLG}(1, 10) = 12, \quad \texttt{NoLG}(2, 10) = 13,$$
$$\texttt{NoLG}(3, 10) = 18, \quad \texttt{NoLG}(4, 10) = 17.$$

A identificação é extremamente importante pois permite elaborar cálculos locais em cada elemento e e depois transportá-los para os nós globais, onde realmente a solução é obtida.

Observação: No programa em linguagem C, em virtude de se iniciar a numeração com zero, os nós locais são numerados por $a = 0, 1, 2, 3$.

• **Número de Equações do Sistema:** `EqNo`

Alguns nós globais podem ter seus valores prescritos, ou seja, a solução $u_A = u(A)$ pode ser conhecida devido às condições de fronteira. Assim, para estes nós não é necessário gerar equações no sistema. Desta forma, a subrotina `EqNo` identifica o nó global A com a sua correspondente equação `eqn[`A`]` no sistema e o número total de equações `Neq` do sistema. Assim, temos

$$I = \texttt{eqn}[A],$$

onde $I = 1, \ldots,$ `Neq`. Por conveniência, para os nós onde os valores são prescritos, tomamos $I = 0$. É claro que o número de equações é menor ou igual ao número de nós globais, `Neq` $\leq$ `Nno`.

Por exemplo, suponhamos que a solução $u_A = u(A)$ seja conhecida nos nós globais $A = \{2, 3, 8, 12, 22, 24\}$ da Fig. 4.1. Então, tem-se que

A	1	2	3	4	5	6	7	8	9	10	11	12	13
`eqn`$[A]$	1	0	0	2	3	4	5	0	6	7	8	0	9
A	14	15	16	17	18	19	20	21	22	23	24	25	
`eqn`$[A]$	10	11	12	13	14	15	16	17	0	18	0	19	

Logo, temos 19 nós globais onde a solução não é conhecida e assim o número de equações `Neq` $= 19$. Podemos observar que o nó global A não corresponde, em geral, a A-ésima equação, como acontece no caso unidimensional. Se, em particular, no lugar da solução ser prescrita nos nós acima, tivermos a derivada da solução prescrita nestes nós, então `Neq` $=$ `Nno`, pois neste caso a solução é desconhecida em todos os 25 nós A da malha.

• **Valores de Fronteira:** `Fronteira,CondFront`

Para cada nó global A, introduzimos as condições de fronteira do tipo Dirichlet e do tipo Neumann. Nas condições de fronteira do tipo Dirichlet a solução $u_A = u(A)$ é prescrita no nó, enquanto na condição de fronteira de Neumann a derivada normal de $u(x)$ no nó A é prescrita. Inicialmente precisamos identificar quais são os nós da fronteira do domínio. Para isso introduzimos a subrotina (`Fronteira`).

Considere por exemplo, o retângulo $[a,\ b] \times [c,\ d]$ sendo o domínio com as fronteiras definidas por:

$$\begin{aligned}
\Gamma_1 &= \{(x, c) \in \Gamma;\ a \leq x \leq b\},\\
\Gamma_2 &= \{(b, y) \in \Gamma;\ c \leq y \leq d\},\\
\Gamma_3 &= \{(x, d) \in \Gamma;\ a \leq x \leq b\},\\
\Gamma_4 &= \{(a, y) \in \Gamma;\ c \leq y \leq d\}.
\end{aligned}$$

Então precisamos identificar os nós pertencentes a cada uma das fronteiras Γ_i, independentemente se são do tipo Neumann ou Dirichlet. A subrotina (`Fronteira`) tem essa função, além de quantificar os nós de cada fronteira pela expressão (`Nbn[i]`) "Número de nós da fronteira $[i]$". Assim, por exemplo, para a malha da Fig. 4.1, temos os seguintes nós globais em cada fronteira:

$$\begin{aligned}
\Gamma_1 &= \{1, 2, 3, 4, 5\} \qquad \texttt{Nbn}[1] = 5,\\
\Gamma_2 &= \{5, 10, 15, 20, 25\} \qquad \texttt{Nbn}[2] = 5,\\
\Gamma_3 &= \{21, 22, 23, 24, 25\} \qquad \texttt{Nbn}[3] = 5,\\
\Gamma_4 &= \{1, 6, 11, 16, 21\} \qquad \texttt{Nbn}[4] = 5.
\end{aligned}$$

O próximo passo na subrotina (`CondFront`), o operador deve definir as funções e as condições de fronteira do problema a ser implementado. O programa permite que sejam definidas condições de fronteira de Dirichlet e Neumann em cada uma das fronteiras Γ_i, mediante duas opções:

(Dirichlet): `typ` $(A) = 1$, a solução é conhecida.

(Neumann): `Bv`, a derivada da solução é conhecida.

A partir da introdução dos dados da fronteira a subrotina (`CondFront`) associa todos os nós de cada fronteira Γ_i aos seus valores e às formas de contribuição de cada tipo na obtenção da solução numérica, como veremos nas seções seguintes.

4.3 Construção do sistema linear

Seja N o conjunto de nós da malha e N_q o conjunto dos nós do tipo 1, isto é, os nós para os quais a solução é conhecida. Então o conjunto $N - N_q$ representa os nós para os quais a solução deverá ser determinada.

Seja $A \in N$ um nó global. Definimos em A uma função interpolante linear φ_A satisfazendo a condição:

$$\varphi_A(B) = \begin{cases} 1, & \text{se } A = B, \\ 0, & \text{se } A \neq B, \end{cases} \qquad \forall\, B \in N.$$

Para $A \in N - N_q$ o conjunto das funções lineares φ_A geram um espaço vetorial linear por partes V_h que é um subespaço do espaço V. Assim, toda função $w_h \in V_h$ pode ser escrita por

$$w_h(x) = \sum_{B \in N - N_q} C_B \varphi_B(x). \tag{4.12}$$

De forma análoga, para a obtenção do sistema linear tem-se

$$\sum_{B \in N - N_q} a(\varphi_A, \varphi_B) C_B = (\varphi_A : f) + (\varphi_A : p)_\Gamma - a(\varphi_A, \widetilde{q}), \quad \forall A \in N - N_q. \tag{4.13}$$

Variando os nós globais A em $N - N_q$ tem-se um sistema linear com `Neq` equações. Para definir a matriz dos coeficientes é necessário estabelecer uma identificação entre o nó A e a sua equação, através da subrotina `EqNo`. Seja

$$I = \texttt{eqn}[A] \quad e \quad J = \texttt{eqn}[B],$$

onde $1 \leq I, J \leq$ `Neq`. Então o sistema linear é definido por

$$\sum_{J=1}^{\texttt{Neq}} K_{IJ} C_J = F_I,$$

onde

$$K_{IJ} = a(\varphi_A, \varphi_B), \tag{4.14}$$

$$F_I = (\varphi_A : f) + (\varphi_A : p)_\Gamma - a(\varphi_A, \widetilde{q}). \tag{4.15}$$

A matriz K é denominada matriz rigidez global do sistema linear.

4.3.1 Interpolação dos Dados Iniciais

Para a equação do calor, a força f é definida em todo $x \in \Omega$ e as funções q e p são definidas em Γ_q e Γ_p respectivamente. Para a resolução do problema pelo método de elementos finitos, somente são necessários os valores de f, p e q nos nós A. Assim, é prático usar a interpolação, tendo como polinômios interpoladores os polinômios φ_A, base do subespaço V_h. Logo,

$$f_h(x) = \sum_{A \in N} \varphi_A(x) f_A, \quad q_h(x) = \sum_{A \in N_q} \varphi_A(x) q_A, \quad p_h(x) = \sum_{A \in Np} \varphi_A(x) p_A,$$

onde $f_A = f(x_A)$, $q_A = q(x_A)$ e $p_A = p(x_A)$. Usando a definição de $\varphi_A(x)$, temos então,

$$f_h(x_A) = f_A, \quad q_h(x_A) = q_A \quad \text{e} \quad p_h(x_A) = p_A.$$

Substituindo em (4.15) obtém-se

$$F_I = \sum_{B \in N} (\varphi_A : \varphi_B) f_B + \sum_{B \in Np} (\varphi_A : \varphi_B) p_B - \sum_{B \in N_q} a(\varphi_A : \varphi_B) q_B.$$

Usando a definição dada em (4.6) e substituindo em (4.15) obtém-se

$$F_I = \sum_{B \in N} \int_\Omega \varphi_A \varphi_B f_B \, dx + \sum_{B \in Np} \int_{\Gamma_p} \varphi_A \varphi_B p_B \, d\Gamma_p - \sum_{B \in N_q} K_{AB} q_B, \tag{4.16}$$

onde $I = \texttt{eqn}[A]$, para $1 \leq I \leq \texttt{Neq}$.

4.3.2 Propriedades da Matriz Rigidez

Analogamente, usando (4.6) e (4.14) temos

$$K_{IJ} = a(\varphi_A, \varphi_B) = \int_\Omega (\nabla \varphi_A)^T Q (\nabla \varphi_B) d\Omega, \tag{4.17}$$

onde $I = \texttt{eqn}[A]$ e $J = \texttt{eqn}[B]$. A matriz K_{IJ} tem seguintes propriedades:

Teorema 4.1. *A matriz K definida em (4.17) é simétrica e definida positiva.*

Demonstração: Temos $K_{IJ} = a(\varphi_A, \varphi_B) = a(\varphi_B, \varphi_A) = K_{JI}$, onde usamos o fato da matriz condutividade Q ser simétrica. Portanto K é simétrica. Assim, se $C = (C_1, C_2, ..., C_{\texttt{Neq}})$, então

$$C^T K C = \sum_{I,J=1}^{\texttt{Neq}} C_I K_{IJ} C_J = \sum_{A,B \in N - N_q} a(\widehat{C}_A \varphi_A, \widehat{C}_B \varphi_B)$$
$$= a(w_h, w_h) = \int_\Omega Q_{ij} \frac{\partial w_h}{\partial x_i} \frac{\partial w_h}{\partial x_j}\, dx \geq 0,$$

onde estamos usando a hipótese da matriz Q ser definida positiva e denotamos

$$w_h = \sum_{A,B \in N - N_q} \widehat{C}_A \varphi_A.$$

Por outro lado, se $C^T K C = 0$, segue que $Q_{ij}(\partial w_h/\partial x_i)(\partial w_h/\partial x_j) = 0$. Então a igualdade é verdadeira se w_h é uma constante. Mas se $\Gamma_q \neq \emptyset$ então $w_h \in V_h$ e portanto $w_h = 0$ em Γ_q. Assim $w_h = 0$, $\forall x \in \Omega$ e $C = 0$, o que implica que a matriz K é definida positiva. Note que, para problema de Neumann, temos $\Gamma_q = \emptyset$ e neste caso w_h é uma constante arbitrária e consequentemente a matriz K é somente semi-definida positiva.

4.3.3 Matriz Rigidez Local e Força Local

As fórmulas (4.17) e (4.16) para obtenção da matriz Rigidez K e do vetor Força F em todo o domínio Ω não são as mais convenientes. Um procedimento mais apropriado é definir as funções de interpolação φ_A para cada elemento finito e, onde φ_A é um polinômio interpolador em Ω_e e vale zero em $\Omega - \Omega_e$, onde Ω_e é o domínio do elemento e. Dessa forma, a função φ_A é denominada função de interpolação local e é denotada por φ_A^e. Usando a discretização do domínio Ω dado anteriormente, introduzimos a matriz local K^e e a força local F^e definidas para cada elemento finito e por

$$K_{IJ}^e = a(\varphi_A^e, \varphi_B^e) = \int_{\Omega_e} (\nabla \varphi_A^e)^T Q (\nabla \varphi_B^e)\, d\Omega, \tag{4.18}$$

$$F_I^e = \sum_{B \in N} \int_{\Omega_e} \varphi_A^e \varphi_B^e f_B\, d\Omega + \sum_{B \in Np} \int_{\Gamma_p^e} \varphi_A^e \varphi_B^e p_B\, d\Gamma - \sum_{B \in N_q} K_{AB}^e q_B, \tag{4.19}$$

onde $I = \texttt{eqn}[A]$, $J = \texttt{eqn}[B]$ e $1 \leq I, J \leq \texttt{Neq}$. Logo a matriz global K e o vetor força F são obtidos respectivamente por

$$K = \sum_{e=1}^{\texttt{Nel}} K^e \quad \text{e} \quad F = \sum_{e=1}^{\texttt{Nel}} F^e.$$

A matriz K^e_{IJ} e o vetor Força F^e_I definidos em (4.18) e (4.19) têm ordem $\texttt{Neq} \times \texttt{Neq}$ e $\texttt{Neq} \times 1$. Mas a função de interpolação φ^e_A tem suporte compacto em Ω_e e assim a matriz e o vetor força são nulos para todos os nós $B \notin \Omega_e$. Considerando em Ω_e, quatro nós globais, a matriz K^e_{IJ} tem somente uma submatriz de ordem 4×4 cujos elementos não são necessariamente nulos e o vetor força F^e_I tem somente quatro coordenadas não necessariamente nulas. Considere, por exemplo um elemento e dado por onde

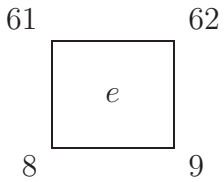


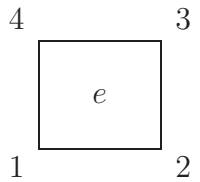
Figura 4.2: Um elemento e

$A = \{8, 9, 61, 62\}$ são os nós globais. Suponhamos que a estes nós correspondam as equações $\texttt{eqn}(A) = \{5, 6, 47, 48\}$, onde esta correspondência é dada pela subrotina `EqNo`. Então os elementos da matriz e do vetor força $K^e_{5,5}$, $K^e_{5,6}$, $K^e_{5,47}$, $K^e_{5,48}$; $K^e_{6,6}$, $K^e_{6,47}$, $K^e_{6,48}$; $K^e_{47,47}$, $K^e_{47,48}$, $K^e_{48,48}$ e F^e_5, F^e_6, F^e_{47}, F^e_{48} são os únicos elementos que não são necessariamente nulos. Para todos os nós $B \notin \Omega_e = \{8, 9, 61, 62\}$ a função de interpolação $\varphi^e_A \equiv 0$ e, portanto, os coeficientes da matriz K^e e as coordenadas de F^e correspondentes são nulas. Se, por exemplo, a malha tem uma quantidade de nós A que corresponde a 1000 equações, então, cada matriz K^e_{IJ} tem ordem 1000×1000, onde para cada elemento e somente $4 \times 4 = 16$ coeficientes da matriz e 4 coordenadas do vetor força não são necessariamente nulos.

A desvantagem deste procedimento está no armazenamento das matrizes K^e e do vetor força F^e e também do número desnecessário de operações entre zeros. Para o exemplo acima, cada matriz K^e tem $(1000 \times 1000) - (4 \times 4) = 999984$ coeficientes nulos. Esta é uma das razões para se introduzir os nós locais.

4.3.4 Nós Locais

Considere Ω_e um retângulo como dado abaixo:



onde $a = 1, 2, 3, 4$ são os nós locais do elemento e. Localmente define-se as funções de interpolação local dadas por

$$\varphi_a^e(\boldsymbol{x}_b^e) = \begin{cases} 1, & \text{se } a = b, \\ 0, & \text{se } a \neq b, \end{cases}$$

onde $\boldsymbol{x}_b^e$ é a posição do nó local do elemento e. Para cada elemento e definimos a matriz local $K^e = K_{ab}^e$ e o vetor força local F_a^e por

$$K_{ab}^e = a(\varphi_a^e, \varphi_b^e) = \int_{\Omega_e} (\nabla\varphi_a^e)^t Q(\nabla\varphi_b^e)\, d\Omega, \quad 1 \leq a, b \leq 4 \tag{4.20}$$

$$F_a^e = \sum_{b=1}^{4} \int_{\Omega_e} \varphi_a^e \varphi_b^e\, f_b^e\; d\Omega + \sum_{b=1}^{4} \int_{\Gamma_p^e} \varphi_a^e \varphi_b^e\, p_b^e\, d\Gamma - \sum_{b=1}^{4} K_{ab}^e q_b^e. \tag{4.21}$$

Seja $B = \texttt{NoLG}(b,e)$. Então a primeira, segunda e terceira integrais são válidas para $B \in N - N_q$, $B \in Np$ e $B \in N_q$, respectivamente. Efetivamente, a introdução do nó local permite obter uma matriz local K_{ab}^e de ordem (4×4) que é uma submatriz da matriz K_{IJ}^e formada pelos elementos não necessariamente nulos de K_{IJ}^e. De forma análoga, o vetor F_a^e é formado pelas quatro coordenadas não necessariamente nulas de F_I^e. Estabelecendo uma relação entre os nós locais a e os nós globais A, determina-se a contribuição de cada elemento e na obtenção da matriz global K e do vetor força F. Vimos anteriormente que a relação entre nós locais e nós globais é dada pela subrotina `NoLG` definida por

$$A = \texttt{NoLG}(a, e).$$

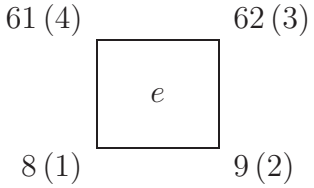
Além disso, é necessário relacionar o nó global A e a correspondente equação do sistema. Isto é feito pela subrotina `EqNo`, definida por

$$I = \texttt{eqn}[A], \quad 1 \leq I \leq \texttt{Neq}.$$

A composição entre as duas relações nos permite relacionar o nó local a com o número da equação I correspondente, através de

$$I = \texttt{eqn}[\texttt{NoLG}(a, e)] = \texttt{eqn}[A].$$

Considere o exemplo dado pela Fig. 4.2, com a introdução dos nós locais no sentido anti-horário.



Através de (4.20) e (4.21) calcula-se a matriz local K^e_{ab} e F^e_a dadas genericamente por

$$K^e_{ab} = \begin{bmatrix} K^e_{11} & K^e_{12} & K^e_{13} & K^e_{14} \\ & K^e_{22} & K^e_{23} & K^e_{24} \\ & & K^e_{33} & K^e_{34} \\ & & & K^e_{44} \end{bmatrix} \quad \text{e} \quad F^e_a = \begin{bmatrix} F^e_1 \\ F^e_2 \\ F^e_3 \\ F^e_4 \end{bmatrix}$$

K_{ab} é simétrica. Tem-se que

$$\begin{aligned} &\texttt{eqn}[\texttt{NoLG}(1,e)] = \texttt{eqn}[8] = 5, \\ &\texttt{eqn}[\texttt{NoLG}(2,e)] = \texttt{eqn}[9] = 6, \\ &\texttt{eqn}[\texttt{NoLG}(3,e)] = \texttt{eqn}[62] = 48, \\ &\texttt{eqn}[\texttt{NoLG}(4,e)] = \texttt{eqn}[61] = 47. \end{aligned}$$

Portanto, a contribuição do elemento e, na montagem da matriz global K e do vetor força são dados por

$$\begin{array}{lcllcl} K_{55} & \leftarrow & K_{55} & + & K^e_{11}, & \\ K_{56} & \leftarrow & K_{56} & + & K^e_{12}, & \\ K_{5\,48} & \leftarrow & K_{5\,48} & + & K^e_{13}, & \\ K_{5\,47} & \leftarrow & K_{5\,47} & + & K^e_{14}, & \qquad F_5 \leftarrow F_5 + F^e_1, \\ K_{66} & \leftarrow & K_{66} & + & K^e_{22}, & \qquad F_6 \leftarrow F_6 + F^e_2, \\ K_{6\,48} & \leftarrow & K_{6\,48} & + & K^e_{23}, & \qquad F_{48} \leftarrow F_{48} + F^e_3, \\ K_{6\,47} & \leftarrow & K_{6\,47} & + & K^e_{24}, & \qquad F_{47} \leftarrow F_{47} + F^e_4. \\ K_{48\,48} & \leftarrow & K_{48\,48} & + & K^e_{33}, & \\ K_{48\,47} & \leftarrow & K_{48\,47} & + & K^e_{34}, & \\ K_{47\,47} & \leftarrow & K_{47\,47} & + & K^e_{44}, & \end{array}$$

Variando $e = 1, 2, ..., \texttt{Nel}$, a matriz global K e o vetor força F do sistema linear são obtidos por

$$K = \sum_{e=1}^{\texttt{Nel}} K^e, \qquad F = \sum_{e=1}^{\texttt{Nel}} F^e,$$

onde K^e e F^e estão definidos em (4.20) e (4.21).

Para determinar a matriz local K^e e o vetor força local F^e, introduziremos a seguir a função de interpolação e sua derivada e também um método numérico para o cálculo da integral.

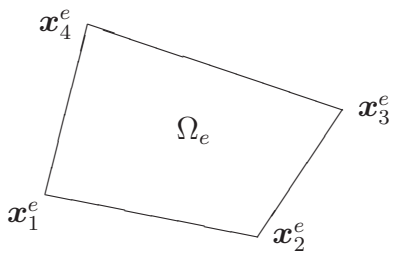


Figura 4.3: Um quadrilátero

4.3.5 Função de Interpolação

Consideremos um quadrilátero $\Omega_e \subset \Omega$ representado como abaixo:

Para cada Ω_e definiremos as funções teste ou função de interpolação, de tal forma que:

i) φ_a^e é de classe C^1 no interior de cada Ω^e,

ii) φ_a^e é contínua no interior de cada elemento Γ^e da fronteira,

onde $a = 1, 2, 3, 4$ são os nós de cada quadrilátero e $e = 1, 2, \ldots,$ `Nel` são os elementos da malha.

A enumeração dos vértices (nós) é ordenada no sentido anti-horário. Para cada lado do quadrilátero definiremos uma função de interpolação linear φ_a^e satisfazendo as condições i) e ii) e além disso, vamos impor as condições de interpolação:

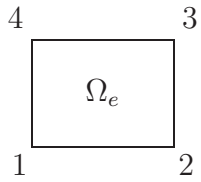
$$\varphi_a^e(\boldsymbol{x}_b^e) = \begin{cases} 1 & \text{se } a = b, \\ 0 & \text{se } a \neq b, \end{cases} \tag{4.22}$$

onde $1 \leq a, b \leq 4$ e $\boldsymbol{x}_b^e = (x_b^e, y_b^e)$.

Uma função bilinear em Ω_e é dada por

$$\varphi_a^e(x, y) = C_1 + C_2 x + C_3 y + C_4 xy; \quad a = 1, 2, 3, 4,$$

onde os C_i's deverão ser determinados de tal forma a satisfazer a condição de interpolação. Em particular, considere Ω_e um retângulo com as seguintes coordenadas locais:



Queremos determinar φ_1^e, φ_2^e, φ_3^e, φ_4^e, interpoladores lineares satisfazendo (4.22).

Usando os polinômios de Lagrange, obtém-se

$$\varphi_1^e(x,y) = \frac{(x-x_2^e)(y-y_2^e)}{(x_1^e-x_2^e)(y_1^e-y_2^e)},$$

$$\varphi_2^e(x,y) = \frac{(x-x_1^e)(y-y_2^e)}{(x_2^e-x_1^e)(y_1^e-y_2^e)},$$

$$\varphi_3^e(x,y) = \frac{(x-x_1^e)(y-y_1^e)}{(x_2^e-x_1^e)(y_2^e-y_1^e)},$$

$$\varphi_4^e(x,y) = \frac{(x-x_2^e)(y-y_1^e)}{(x_1^e-x_2^e)(y_2^e-y_1^e)},$$

onde $\boldsymbol{x}_1^e = (x_1^e, y_1^e)$, $\boldsymbol{x}_2^e = (x_2^e, y_1^e)$, $\boldsymbol{x}_3^e = (x_2^e, y_2^e)$, $\boldsymbol{x}_4^e = (x_1^e, y_2^e)$. A função de interpolação φ_a^e pode ser representada gráficamente por

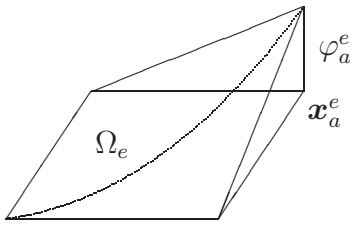


Figura 4.4: Função de interpolação

• Transformação Isoparamétrica

Em particular, quando o domínio é o quadrado unitário $[-1,1]\times[-1,1]$, a função de interpolação φ_a tem a forma simples, ou seja, denotando

$$\boldsymbol{\xi}_1 = (-1,-1), \quad \boldsymbol{\xi}_2 = (1,-1), \quad \boldsymbol{\xi}_3 = (1,1), \quad \boldsymbol{\xi}_4 = (-1,1).$$

temos

$$\varphi_1(\xi,\eta) = \frac{1}{4}(1-\xi)(1-\eta), \tag{4.23}$$

$$\varphi_2(\xi,\eta) = \frac{1}{4}(1+\xi)(1-\eta), \tag{4.24}$$

$$\varphi_3(\xi,\eta) = \frac{1}{4}(1+\xi)(1+\eta), \tag{4.25}$$

$$\varphi_4(\xi,\eta) = \frac{1}{4}(1-\xi)(1+\eta), \tag{4.26}$$

para $(\xi, \eta) \in [-1, 1] \times [-1, 1]$. Devido a forma simples do polinômio interpolador $\varphi_a(\xi, \eta)$, $a = 1, 2, 3, 4$, no quadrado unitário, e principalmente por usar elementos finitos uniformes, é conveniente fazer uma parametrização entre os elementos Ω_e e o quadrado unitário, $[-1, 1] \times [-1, 1]$, denotado por Ω_b.

Considere a aplicação:

$$(\xi, \eta) \in \Omega_b \mapsto (x, y) \in \Omega_e$$

definida por

$$x(\xi, \eta) = \sum_{a=1}^{4} \varphi_a(\xi, \eta) x_a^e, \tag{4.27}$$

$$y(\xi, \eta) = \sum_{a=1}^{4} \varphi_a(\xi, \eta) y_a^e. \tag{4.28}$$

As funções (4.27) e (4.28) são biunívocas entre o quadrilátero Ω_e e o quadrado unitário Ω_b. De fato, definindo

$$\begin{array}{ll} (\xi_1, \eta_1) = (-1, -1) & (\xi_2, \eta_2) = (1, -1) \\ (\xi_3, \eta_3) = (1, 1) & (\xi_4, \eta_4) = (-1, 1) \end{array} \tag{4.29}$$

então,

$$x(\xi_b, \eta_b) = \sum_{a=1}^{4} \varphi_a(\xi_b, \eta_b) x_a^e = \begin{cases} x_a^e, & se \quad a = b, \\ 0, & se \quad a \neq b. \end{cases}$$

$$y(\xi_b, \eta_b) = \sum_{a=1}^{4} \varphi_a(\xi_b, \eta_b) y_a^e = \begin{cases} y_a^e, & se \quad a = b, \\ 0, & se \quad a \neq b. \end{cases}$$

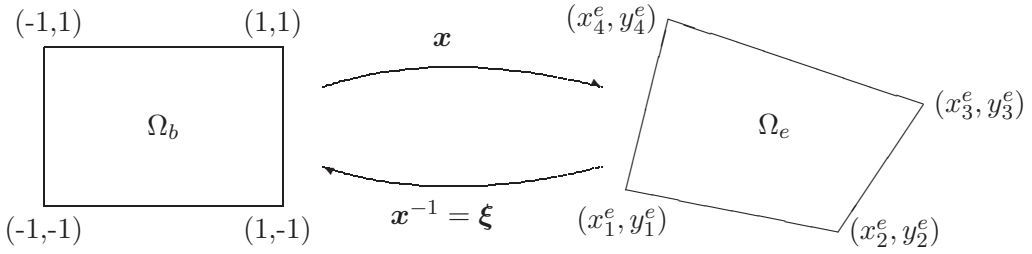


Figura 4.5: Transformação isoparamétrica

Na Fig. 4.5, denotamos $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi}) = (x, y)$ e $\boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{x}) = (\xi, \eta)$.

Com a notação vetorial, as aplicações (4.27) e (4.28) podem ser dadas por

$$\boldsymbol{x}(\boldsymbol{\xi}) = \sum_{a=1}^{4} \varphi_a(\boldsymbol{\xi}) \boldsymbol{x}_a^e. \tag{4.30}$$

Para verificar a existência da função

$$\boldsymbol{\xi} = \boldsymbol{x}^{-1} : \Omega_e \to \Omega_b,$$

usaremos o Teorema da Função Inversa. A função φ_a é diferenciável e portanto $\boldsymbol{x}(\boldsymbol{\xi})$ é diferenciável. Assim podemos calcular o Jacobiano da transformação isoparamétrica, (a positividade é devido ao sentido anti-horário do mapeamento) dado por

$$J = \det \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} > 0. \tag{4.31}$$

Como $\boldsymbol{x} : \Omega_b \to \Omega_e$ é bijetora e de classe C^1 (pois φ_a tem classe C^1) e o Jacobiano é positivo, o Teorema da Função Inversa garante a existência da função inversa $\boldsymbol{x}^{-1} = \boldsymbol{\xi} : \Omega_e \to \Omega_b$, com $\boldsymbol{x}^{-1}$ de classe C^1. Assim, temos um mapeamento entre os elementos finitos Ω_b e Ω_e.

A função de interpolação φ_a pode ser representada em Ω_b na seguinte forma compacta,

$$\varphi_a(\xi, \eta) = \frac{1}{4}(1 + \xi_a \xi)(1 + \eta_a \eta), \quad a = 1, 2, 3, 4, \tag{4.32}$$

com (ξ_a, η_a) definida em (4.29). Além disso, o gradiente da função φ_a é dado por

$$\nabla \varphi_a(\xi, \eta) = \frac{1}{4}\Big(\xi_a(1 + \eta_a \eta),\ \eta_a(1 + \xi_a \xi)\Big), \quad a = 1, 2, 3, 4. \tag{4.33}$$

As funções φ_a e $\nabla \varphi_a$ estão definidas nas subrotinas `Phi` e `DPhi` do programa.

Para o cálculo do Jacobiano (4.31) são usadas as funções $\boldsymbol{x}(\boldsymbol{\xi})$ em (4.30) e $\nabla \varphi_a(\xi)$ em (4.33) da seguinte forma:

$$\left(\frac{\partial x}{\partial \xi},\ \frac{\partial x}{\partial \eta}\right) = \sum_{a=1}^{4} x_a^e \left(\frac{\partial \varphi_a}{\partial \xi}(\xi, \eta),\ \frac{\partial \varphi_a}{\partial \eta}(\xi, \eta)\right) = \sum_{a=1}^{4} x_a^e \nabla \varphi_a(\xi, \eta), \tag{4.34}$$

$$\left(\frac{\partial y}{\partial \xi},\ \frac{\partial y}{\partial \eta}\right) = \sum_{a=1}^{4} y_a^e \left(\frac{\partial \varphi_a}{\partial \xi}(\xi, \eta),\ \frac{\partial \varphi_a}{\partial \eta}(\xi, \eta)\right) = \sum_{a=1}^{4} y_a^e \nabla \varphi_a(\xi, \eta), \tag{4.35}$$

Utilizando $\nabla\varphi_a(\xi,\eta)$ podemos calcular o Jacobiano da transformação entre o quadrado unitário Ω_b e o quadrilátero Ω_e. Quando Ω_e é um retângulo, o Jacobiano pode ser simplesmente calculado por

$$J = \frac{1}{4}(x_2^e - x_1^e)(y_2^e - y_1^e) = \frac{1}{4}dx^e dy^e. \tag{4.36}$$

Se os retângulos são uniformes, o Jacobiano é constante para todo elemento, sendo portanto desnecessário o cálculo para cada elemento.

Como estamos trabalhando com o retângulo Ω_e, não necessariamente uniforme, o cálculo do Jacobiano será feito por (4.36) na subrotina `Jacob`.

• Gradiente da Função de Interpolação

Para o cálculo da matriz local K^e e da força local F^e, precisamos calcular $\nabla\varphi_a(x,y)$ no quadrado unitário Ω_b, usando a transformação isoparamétrica. Temos que

$$\frac{\partial\varphi_a}{\partial x} = \frac{\partial\varphi_a}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial\varphi_a}{\partial\eta}\frac{\partial\eta}{\partial x},$$

$$\frac{\partial\varphi_a}{\partial y} = \frac{\partial\varphi_a}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial\varphi_a}{\partial\eta}\frac{\partial\eta}{\partial y}.$$

Na forma matricial temos

$$\begin{bmatrix} \dfrac{\partial\varphi_a}{\partial x} \\ \dfrac{\partial\varphi_a}{\partial y} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial\xi}{\partial x} & \dfrac{\partial\eta}{\partial x} \\ \dfrac{\partial\xi}{\partial y} & \dfrac{\partial\eta}{\partial y} \end{bmatrix} \begin{bmatrix} \dfrac{\partial\varphi_a}{\partial\xi} \\ \dfrac{\partial\varphi_a}{\partial\eta} \end{bmatrix}. \tag{4.37}$$

Em geral, não são conhecidos explicitamente as expressões para $\xi = \xi(x,y)$ e $\eta = \eta(x,y)$ e assim não podemos calcular diretamente os coeficientes da primeira matriz do lado direito de (4.37), mas eles podem ser calculados usando a matriz Hessiana $\boldsymbol{x}_\xi$ definida por

$$\boldsymbol{x}_\xi = \begin{bmatrix} \dfrac{\partial x}{\partial\xi} & \dfrac{\partial x}{\partial\eta} \\ \dfrac{\partial y}{\partial\xi} & \dfrac{\partial y}{\partial\eta} \end{bmatrix}. \tag{4.38}$$

Da definição de $\boldsymbol{x}(\xi,\eta)$ dada por (4.27) e (4.28), calcula-se a matriz $\boldsymbol{x}_\xi$. Então, a inversa de $\boldsymbol{x}_\xi$ é dada por

$$(\boldsymbol{x}_\xi)^{-1} = \boldsymbol{\xi}_x = \begin{bmatrix} \dfrac{\partial\xi}{\partial x} & \dfrac{\partial\xi}{\partial y} \\ \dfrac{\partial\eta}{\partial x} & \dfrac{\partial\eta}{\partial y} \end{bmatrix} = \frac{1}{J}\begin{bmatrix} \dfrac{\partial y}{\partial\eta} & -\dfrac{\partial x}{\partial\eta} \\ -\dfrac{\partial y}{\partial\xi} & \dfrac{\partial x}{\partial\xi} \end{bmatrix} = \frac{1}{J}\,\bar{\boldsymbol{x}}_\xi, \tag{4.39}$$

onde $J = J(\xi,\eta) = \det(\boldsymbol{x}_\xi)$ e $\bar{\boldsymbol{x}}_\xi$ é a matriz adjunta de $\boldsymbol{x}_\xi$.

Substituindo em (4.37) temos

$$\begin{bmatrix} \dfrac{\partial\varphi_a}{\partial x} \\ \dfrac{\partial\varphi_a}{\partial y} \end{bmatrix} = \frac{1}{J} \begin{bmatrix} \dfrac{\partial y}{\partial\eta} & -\dfrac{\partial y}{\partial\xi} \\ -\dfrac{\partial x}{\partial\eta} & \dfrac{\partial x}{\partial\xi} \end{bmatrix} \begin{bmatrix} \dfrac{\partial\varphi_a}{\partial\xi} \\ \dfrac{\partial\varphi_a}{\partial\eta} \end{bmatrix}. \tag{4.40}$$

A matriz $(\boldsymbol{x}_\xi)^{-1}$ calculada em (4.39) para quadriláteros pode ser simplificada se calculada para retângulos, pois as funções $\xi(x,y)$ e $\eta(x,y)$ neste caso são dadas explicitamente por

$$\xi(x,y) = \frac{1}{dx_e}((x - x_1^e) + (x - x_2^e)), \tag{4.41}$$

$$\eta(x,y) = \frac{1}{dy_e}((y - y_1^e) + (y - y_2^e)). \tag{4.42}$$

É fácil notar que (ξ,η) é um mapeamento entre o retângulo Ω_e e o quadrado unitário Ω_b. Assim temos

$$(\boldsymbol{x}_\xi)^{-1} = \begin{bmatrix} \dfrac{\partial\xi}{\partial x} & \dfrac{\partial\xi}{\partial y} \\ \dfrac{\partial\eta}{\partial x} & \dfrac{\partial\eta}{\partial y} \end{bmatrix} = \begin{bmatrix} \dfrac{2}{dx_e} & 0 \\ 0 & \dfrac{2}{dy_e} \end{bmatrix}$$

Portanto,

$$\Big(\frac{\partial\varphi_a}{\partial x}, \frac{\partial\varphi_a}{\partial y}\Big) = \Big(\frac{\partial\varphi_a}{\partial\xi}\frac{2}{dx_e}, \frac{\partial\varphi_a}{\partial\eta}\frac{2}{dy_e}\Big). \tag{4.43}$$

4.3.6 Quadratura Gaussiana

Para calcular explicitamente a matriz rigidez e o vetor força, introduzimos o Método da Quadratura Gaussiana para o cálculo da integral numérica. A quadratura Gaussiana é a mais apropriada neste contexto, pois são usados poucos pontos de integração com boa precisão. Em elementos finitos, como as funções de interpolação $\varphi_a^e(x)$ são polinômios de baixo grau, a quadratura é a mais conveniente.

Seja $g : \Omega_b \subset I\!R^2 \to I\!R$ uma função integrável, então,

$$\int_{\Omega_b} g\, d\Omega = \int_{-1}^{1}\int_{-1}^{1} g(\xi,\eta)\, d\xi d\eta. \tag{4.44}$$

Para o número de pontos interiores igual a dois nas direções ξ e η, os pontos são

$$(\xi_1,\eta_1) = (\frac{-\sqrt{3}}{3}, \frac{-\sqrt{3}}{3}), \quad (\xi_2,\eta_2) = (\frac{\sqrt{3}}{3}, \frac{-\sqrt{3}}{3}),$$
$$(\xi_3,\eta_3) = (\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}), \qquad (\xi_4,\eta_4) = (\frac{-\sqrt{3}}{3}, \frac{\sqrt{3}}{3}),$$

com pesos

$$w_1 = w_2 = w_3 = w_4 = 1.$$

Logo, temos

$$\int_{\Omega_e} g(\xi,\eta)\, d\Omega = \sum_{k=1}^{4} g(\xi_k, \eta_k). \tag{4.45}$$

4.3.7 Cálculo da Matriz Local

Para o cálculo da matriz local K^e precisamos também introduzir a matriz condutividade Q, que por hipótese é simétrica e definida positiva. Temos que $Q = Q(x,y)$, $\forall (x,y) \in \Omega \subset I\!R^2$ e os coeficientes da matriz podem ser dados por

$$Q = \begin{bmatrix} f_1(x,y) & f_2(x,y) \\ f_2(x,y) & f_3(x,y) \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12} & Q_{22} \end{bmatrix}$$

onde as funções f_i são definidas de forma a satisfazer as hipóteses sobre Q. Quando o material é homogêneo as funções f_i são constantes. Se o material é isotrópico então $f_2(x,y) = 0$ e $f_1(x,y) = f_3(x,y)$.

A matriz local (4.20) pode ser escrita como

$$K^e_{ab} = \int_{\Omega_e} (\nabla \varphi_a(x,y))^T Q(x,y)(\nabla \varphi_b)\, d\Omega = \int_{\Omega_b} B_a^T Q(\xi,\eta) B_b |J(\xi,\eta)|\, d\Omega_b, \tag{4.46}$$

onde por B_a estamos denotando o lado direito de (4.40), ou

$$B_a = \left[\boldsymbol{\xi}_x\right]_{2\times 2} \cdot \begin{bmatrix} \dfrac{\partial \varphi_a}{\partial \xi} \\ \dfrac{\partial \varphi_a}{\partial \eta} \end{bmatrix}_{2\times 4}$$

que é uma matriz de ordem 2×4.

Para cada Ω_e é necessário o cálculo da integral (4.46) que envolve o cálculo do Jacobiano $J(\xi,\eta)$ e a matriz adjunta $\bar{\boldsymbol{x}}_\xi$. Assim, para quadriláteros, a matriz local K^e_{ab} pode ser obtida utilizando um método de integração numérica.

Como vimos, se Ω_e é um retângulo, o o Jacobiano é dado por (4.36) e assim, temos a seguinte matriz local:

$$K^e_{ab} = J \int_{\Omega_b} (B_a^T Q B_b)\, d\Omega_b. \tag{4.47}$$

Para o cálculo da matriz local K^e_{ab} dada por (4.47) é suficiente definir

$$g(\xi,\eta) = B_a^T(\xi,\eta) Q(\xi,\eta) B_b(\xi,\eta)$$

e usar a quadratura Gaussiana (4.45).

Um outro procedimento para calcular a matriz local K^e_{ab} é dado na forma componente, como descreveremos a seguir.

A matriz K^e_{ab} definida por (4.20) pode ser escrita por

$$K^e_{ab} = \int_{\Omega_e} (\nabla\varphi_a)^T Q(\nabla\varphi_b) d\Omega = \sum_{k,l=1}^{2} \int_{\Omega_e} Q_{kl} \frac{\partial\varphi_a}{\partial x_k} \frac{\partial\varphi_b}{\partial x_l}\, d\Omega. \tag{4.48}$$

Como

$$\frac{\partial\varphi_a}{\partial x_k} = \frac{\partial\varphi_a}{\partial\xi_i}\frac{\partial\xi_i}{\partial x_k},$$

tem-se

$$K^e_{ab} = \sum_{i,j,k,l=1}^{2} \int_{\Omega_b} Q_{kl} \frac{\partial\varphi_a}{\partial\xi_i} \frac{\partial\xi_i}{\partial x_k} \frac{\partial\varphi_b}{\partial\xi_j} \frac{\partial\xi_j}{\partial x_l} |J|\, d\Omega_b. \tag{4.49}$$

Usando (4.43), obtém-se

$$K^e_{ab} = \sum_{i,j=1}^{2} \frac{4|J|}{dx^e_i dx^e_j} \int_{\Omega_b} Q_{ij} \frac{\partial\varphi_a}{\partial\xi_i} \frac{\partial\varphi_b}{\partial\xi_j}\, d\Omega_b. \tag{4.50}$$

Consideremos, em particular, que a matriz condutividade seja constante em Ω_e, (esta suposição é razoável para Ω_e pequeno) ou seja, dado um nó global $A = (x_A, y_A) \in \Omega_e$, vamos supor que

$$Q(x, y) = Q(x_A, y_A) = Q^e(A), \quad \forall x, y \in \Omega_e,$$

onde o nó global A escolhido está associado ao primeiro nó local a do elemento e, isto é,

$$A = \texttt{NoLG}(1, e).$$

Substituindo em (4.50) obtém-se

$$K^e_{ab} = \sum_{i,j=1}^{2} \frac{4|J|}{dx^e_i dx^e_j} Q^e_{ij}(A) \int_{\Omega_b} \frac{\partial\varphi_a}{\partial\xi_i} \frac{\partial\varphi_b}{\partial\xi_j}\, d\Omega_b.$$

Definindo a matriz

$$Q_{abij} = \int_{\Omega_b} \frac{\partial\varphi_a}{\partial\xi_i} \frac{\partial\varphi_b}{\partial\xi_j}\, d\Omega_b = \int_{-1}^{1}\int_{-1}^{1} \frac{\partial\varphi_a}{\partial\xi_i} \frac{\partial\varphi_b}{\partial\xi_j}\, d\xi d\eta. \tag{4.51}$$

que é independente do elemento e, temos

$$K^e_{ab} = 4|J| \sum_{i,j=1}^{2} \frac{1}{dx^e_i dx^e_j} Q^e_{ij}(A) Q_{abij}, \tag{4.52}$$

onde $1 \leq a, b \leq 4$.

Como $1 \leq i, j \leq 2$, temos as matrizes de ordem 4 Q_{ab11}, Q_{ab12}, Q_{ab21} e Q_{ab22} definidas por (4.51), que podem ser calculadas diretamente ou usando a quadratura Gaussiana.

Usando a quadratura Gaussiana, definimos

$$g_{abij} = \frac{\partial \varphi_a}{\partial \xi_i} \frac{\partial \varphi_b}{\partial \xi_j}.$$

Assim a integral (4.51) pode se calculada por

$$Q_{abij} = \sum_{l=1}^{4} g_{abij}(\xi_l, \eta_l). \tag{4.53}$$

Estas matrizes podem ser calculadas diretamente e são dadas por

$$Q_{ab11} = \frac{1}{6} \begin{bmatrix} 2 & -2 & -1 & 1 \\ -2 & 2 & 1 & -1 \\ -1 & 1 & 2 & -2 \\ 1 & -1 & -2 & 2 \end{bmatrix},$$

$$Q_{ab12} = \frac{1}{4} \begin{bmatrix} 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \end{bmatrix},$$

$$Q_{ab21} = \frac{1}{4} \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \end{bmatrix},$$

$$Q_{ab22} = \frac{1}{6} \begin{bmatrix} 2 & 1 & -1 & -2 \\ 1 & 2 & -2 & -1 \\ -1 & -2 & 2 & 1 \\ -2 & -1 & 1 & 2 \end{bmatrix}.$$

O cálculo de Q_{abij} está contido na subrotina `PhiMatriz` e o da matriz local está contido na subrotina `LocalSystem`.

Para o cálculo da matriz local K^e_{ab} quando Ω_e é um retângulo, temos as fórmulas (4.47) e (4.52).

Em (4.47) temos de calcular 4 integrais para cada elemento e, enquanto que em (4.52), somente são necessários o cálculo de 4 integrais.

Assim, por exemplo, se o número total de elementos de uma malha `Nel` = 1000, então temos que calcular 4000 integrais usando a formula (4.47). Além disso, a matriz condutividade é variável, enquanto que pela formulação (4.52) a matriz é assumida constante por elemento.

4.3.8 Cálculo da Força Local

A força local definida por (4.21) pode ser escrita por

$$F^e_a = f^e_a + p^e_a - q^e_a, \quad a = 1, 2, 3, 4, \tag{4.54}$$

onde

$$f^e_a = \sum_{b=1}^{4} \int_{\Omega_e} \varphi^e_a \varphi^e_b f^e_b \, d\Omega_e, \tag{4.55}$$

$$p^e_a = \sum_{b=1}^{4} \int_{\Gamma_p} \varphi^e_a \varphi^e_b p^e_b \, d\Gamma_e, \tag{4.56}$$

$$q^e_a = \sum_{b=1}^{4} K^e_{ab} q^e_b. \tag{4.57}$$

As igualdades (4.55), (4.56) e (4.57) são válidas para todo nó local $b \in \Omega_e$ tal que o correspondente nó global B, dado por $B = \texttt{NoLG}(b, e)$ satisfaça as respectivas condições: $B \in N$, $B \in N_p$ e $B \in N_q$.

• **Cálculo de f^e_a**

Consideremos Ω_e um retângulo. De (4.41) e (4.42) as funções $\xi(x, y)$ e $\eta(x, y)$ são aplicações isoparamétricas entre Ω_e e o quadrado unitário Ω_b.

Dessa forma tem-se

$$f^e_a = \sum_{b=1}^{4} f^e_b \int_{\Omega_e} \varphi^e_a \varphi^e_b \, d\Omega_e = \sum_{b=1}^{4} f^e_b \int_{\Omega_b} \varphi^e_a \varphi^e_b J \, d\Omega_b.$$

Definindo

$$Q_{ab} = \int_{\Omega_b} \varphi^e_a \varphi^e_b \, d\Omega_b, \tag{4.58}$$

então,

$$f_a^e = J \sum_{b=1}^{4} f_b^e Q_{ab}, \quad a = 1, 2, 3, 4. \tag{4.59}$$

Usando as definições de $\varphi_a(\xi, \eta)$ dada por (4.32), a matriz Q_{ab} pode ser calculada diretamente ou usando-se quadratura Gaussiana, isto é,

$$Q_{ab} = \frac{1}{9} \begin{bmatrix} 4 & 2 & 1 & 2 \\ 2 & 4 & 2 & 1 \\ 1 & 2 & 4 & 2 \\ 2 & 1 & 2 & 4 \end{bmatrix}. \tag{4.60}$$

O cálculo de Q_{ab} está contido na subrotina `PhiMatriz`.

Para as funções de interpolação linear φ_a, o cálculo de (4.58) pelo método de quadratura Gaussiana com dois pontos interiores em cada direção, é esato. Para efeito de programação, é preferível utilizar a quadratura Gaussiana para evitar a inserção dos coeficientes da matriz, mesmo que a matriz seja constante para todo elemento e como em (4.58).

4.3.9 Cálculo dos Valores de Fronteira

Os vetores p_a^e e q_a^e definidos anteriormente são as condições de fronteira do tipo Neumann e Dirichlet, respectivamente, os quais influenciam na força local F_a^e para os nós globais A que pertencem à fronteira Γ_p ou Γ_q, onde $\Gamma_p \cap \Gamma_q = \emptyset$.

Para ilustrar a contribuição da fronteira no cálculo da força local, considere a malha dada anteriormente na Fig. 4.1.

Seja Γ_q e Γ_p o conjunto de nós globais A onde a solução e a derivada da solução são prescritas em A, dados respectivamente por

$$\Gamma_q = \{1, 3, 5, 22, 24\}, \quad \Gamma_p = \{6, 10, 16, 20\}.$$

Considere $A = \texttt{NoLG}(a, e)$. Então temos

1) Os elementos $e = 6, 7, 10, 11$, não possuem nenhum nó global $A \in \Gamma_p$ ou $A \in \Gamma_q$. Assim
$$F_a^e = f_a^e, \quad a = 1, 2, 3, 4.$$
pois, $p_a^e = q_a^e = 0$.

2) Os elementos $e = 1, 4, 13, 16$, têm nós globais em Γ_q e Γ_p. Para estes elementos:
$$F_a^e = f_a^e + p_a^e - q_a^e.$$

3) Os elementos $e = 2, 3, 14, 15$, somente têm nós globais em Γ_q, logo,

$$p_a^e = 0, \quad F_a^e = f_a^e - q_a^e.$$

4) Os elementos $e = 5, 8, 9, 12$, somente têm nós globais em Γ_p, logo

$$q_a^e = 0, \quad F_a^e = f_a^e + p_a^e.$$

Os nós locais a referidos acima satisfazem a condição

$$A = \texttt{NoLG}(a, e), \quad \text{onde } A \in \Gamma_p \text{ ou } A \in \Gamma_q.$$

Por exemplo para o elemento $e = 4$ temos

$$\begin{aligned} 5 &= \texttt{NoLG}(2,4) \in \Gamma_q, \\ 10 &= \texttt{NoLG}(3,4) \in \Gamma_p, \\ 4 &= \texttt{NoLG}(1,4) \notin \Gamma_q \cup \Gamma_p, \\ 9 &= \texttt{NoLG}(4,4) \notin \Gamma_q \cup \Gamma_p, \end{aligned}$$

onde $A = 4, 5, 10, 9$ são os nós globais do elemento $e = 4$ e $a = 1, 2, 3, 4$ são os nós locais.

Assim, para o elemento $e = 4$, tem-se

$$\begin{aligned} F_1^e &= f_1^e, \\ F_2^e &= f_2^e - q_2^e, \\ F_3^e &= f_3^e + p_3^e, \\ F_4^e &= f_4^e. \end{aligned}$$

Os cálculos de q_a^e e p_a^e são dados a seguir.

• **Cálculo de q_a^e**

Por definição, temos

$$q_a^e = \sum_{b=1}^{4} K_{ab}^e q_b^e, \quad a = 1, 2, 3, 4.$$

Como vimos, somente são calculados as coordenadas $(q_1^e, q_2^e, q_3^e, q_4^e)$ que satisfazem a condição de nó global $A \in \Gamma_q$, onde $A = \texttt{NoLG}(a, e)$. Por definição de interpolação $q_b^e = q(x_b^e) = q(B)$, onde B é o nó global correspondente a (b, e). Assim, se $B \in \Gamma_q$, a solução do problema $u(x)$ é conhecida neste nó. Logo,

$$u(B) = q(B) = q_b^e.$$

Se $B \notin \Gamma_q$, então $q_b^e = 0$ e o cálculo de q_a^e é feito através do seguinte procedimento: definimos na subrotina `CondFront` se os nós $B \in \Gamma_q$ são do tipo $= 1$, isto é, $\texttt{typ}(B) = 1$. Se existe pelo menos um nó $B = \texttt{NoLG}(b, e)$ do tipo $\texttt{typ}(B) = 1$, então,

$$q_a^e = \sum_{b=1}^{4} K_{ab}^e q_b^e.$$

Caso contrário, $q_a^e = 0$, $a = 1, 2, 3, 4$. Este procedimento está contido na subrotina `LocalSystem`.

• **Cálculo de p_a^e**

Por definição,

$$p_a^e = \sum_{b=1}^{4} \int_{\Gamma_p} \varphi_a^e \varphi_b^e p_b^e \, d\Gamma,$$

onde $p_b^e = p(x_b^e) = p(B)$. Destacaos os dois tipos de nó $B \in \Gamma_p$: os nós com fluxo na direção x são, que são aqueles pertencentes às fronteiras $\Gamma_1 = \Gamma_{p1}$ e $\Gamma_3 = \Gamma_{p3}$; e os nós com fluxo na direção y, que são os nós da fronteira pertencentes a $\Gamma_2 = \Gamma_{p2}$ e $\Gamma_4 = \Gamma_{p4}$. Geometricamente mostramos a fronteira Γ_p na Fig. 4.6, onde Γ_{p1}, Γ_{p2}, Γ_{p3}, Γ_{p4} são

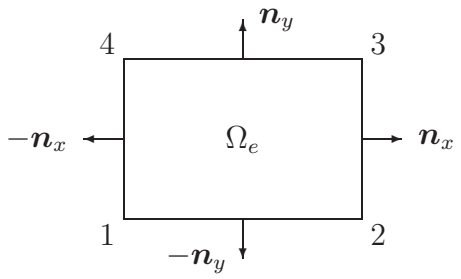


Figura 4.6: Normais externas

representados respectivamente pelo segmentos $\overline{12}$, $\overline{23}$, $\overline{34}$, $\overline{41}$ definidas pelos nós locais .

Assim, definimos as fronteiras horizontais e verticais por $\Gamma_{p1} \cup \Gamma_{p3}$ e $\Gamma_{p2} \cup \Gamma_{p4}$, para as quais temos conhecidos respectivamente o fluxo normal na direção y e o fluxo normal direção x.

Definimos a matriz M_{ab} por

$$M_{ab} = \int_{\Gamma} \varphi_a^e \varphi_b^e \, d\Gamma, \tag{4.61}$$

de modo que

$$p_a^e = \sum_{b=1}^{4} M_{ab} p_b^e. \tag{4.62}$$

Para determinar o vetor p_a^e é suficiente determinar a matriz M_{ab} em cada uma das fronteiras Γ_{pi}. Para calcular a matriz M_{ab} em Γ_{p1} *e* Γ_{p3}, basta observar que a função de interpolação linear só depende de x. Assim,

1) Em Γ_{p1} as funções $\varphi_a(x,y) = \varphi_a(x, y_1^e)$ são dadas por

$$\varphi_1(x, y_1^e) = \frac{x - x_2^e}{x_1^e - x_2^e}, \quad \varphi_2(x, y_1^e) = \frac{x - x_1^e}{x_2^e - x_1^e}, \quad \varphi_3 = (x, y_1^e) = \varphi_4(x, y_1^e) = 0.$$

2) Em Γ_{p3} temos $\varphi_a(x,y) = \varphi_a(x, y_2^e)$ dadas por

$$\varphi_1(x, y_2^e) = \varphi_2(x, y_2^e) = 0, \quad \varphi_3(x, y_2^e) = \frac{x - x_1^e}{x_2^e - x_1^e}, \quad \varphi_4(x, y_2^e) = \frac{x - x_2^e}{x_1^e - x_2^e}.$$

As funções de interpolação acima são obtidas das funções de interpolação (4.23) a (4.26) definidas sobre retângulos. Usando a definição de M_{ab}, conclui-se que os coeficientes não necessariamente nulos em Γ_{p1} são M_{11}, M_{12}, M_{21}, M_{22} e em Γ_{p3} são M_{33}, M_{34}, M_{43}, M_{44}, onde

$$\begin{aligned} M_{11} &= \int_{x_1^e}^{x_2^e} \varphi_1(x)\varphi_1(x)\, dx = \frac{dx^e}{3} = M_{22}, \\ M_{12} &= \int_{x_1^e}^{x_2^e} \varphi_1(x)\varphi_2(x)\, dx = \frac{dx^e}{6} = M_{21}. \end{aligned}$$

Assim, a matriz M_{ab} para a fronteira Γ_{p1} é dada por

$$M_{ab} = \frac{dx^e}{6} \begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Analogamente, para a fronteira Γ_{p3}, tem-se

$$M_{33} = M_{44} = \frac{dx^e}{3}, \quad M_{34} = M_{43} = \frac{dx^e}{6}.$$

Assim, em Γ_{p3} a matriz é

$$M_{ab} = \frac{dx^e}{6} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}.$$

O cálculo da matriz M_{ab} para as fronteiras Γ_{p2} e Γ_{p4} é análogo ao caso anterior, exceto que as funções de interpolação são constantes em relação a x. Logo, para a fronteira Γ_{p2}, $\varphi_a(x,y) = \varphi_a(x_2^e, y)$ e em Γ_{p4}, $\varphi_a(x,y) = \varphi_a(x_1^e, y)$.

Fazendo os cálculos, obtém-se a matriz M_{ab} em Γ_{p2} dada por

$$M_{ab} = \frac{dy^e}{6}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

e em Γ_{p4} a matriz é dada por

$$M_{ab} = \frac{dy^e}{6}\begin{bmatrix} 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}.$$

As matrizes M_{ab} definidas em Γ_{p1} e Γ_{p3} e as definidas em Γ_{p2} e Γ_{p4} podem ser respectivamente compactadas na forma:

$$M_{ab} = \frac{dx^e}{6}\begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}, \tag{4.63}$$

$$M_{ab} = \frac{dy^e}{6}\begin{bmatrix} 2 & 0 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}. \tag{4.64}$$

Considere o nó global A associado ao nó local a do elemento e pela relação $A = \texttt{NoLG}(a, e)$.

Se o fluxo normal está definido na direção x, então o nó $A \in \Gamma_{p2} \cup \Gamma_{p4}$ e a matriz M_{ab} é definida por (4.64). Se o fluxo normal está definido na direção y, então o nó $A \in \Gamma_{p1} \cap \Gamma_{p3}$ e a matriz M_{ab} está definida por (4.63).

Consideremos um elemento e da malha para o qual os nós locais $a = 2, 3$ estão associados a nós globais A_2, A_3, onde $A_2, A_3 \in \Gamma_{p2}$. Por definição,

$$p_a^e = \sum_{b=1}^{4} M_{ab} p_b^e$$

Para o elemento considerado, $p_b^e = 0$ se $b = 1, 4$. Logo,

$$\begin{aligned} p_1^e &\leftarrow 0, \\ p_2^e &\leftarrow p_2^e M_{22} + p_3^e M_{23} = \frac{dy^e}{6}(2p_2^e + p_3^e), \\ p_3^e &\leftarrow p_2^e M_{32} + p_3^e M_{33} = \frac{dy^e}{6}(p_2^e + 2p_3^e), \\ p_4^e &\leftarrow 0. \end{aligned}$$

Se $\Gamma_p \cap \Gamma_q = \phi$, a força local F_a^e recebe a contribuição de Γ_p nas seguintes coordenadas:

$$\begin{aligned} F_2^e &\leftarrow f_2^e + p_2^e, \\ F_3^e &\leftarrow f_3^e + p_3^e. \end{aligned}$$

As coordenadas F_1^e e F_4^e não recebem contribuição da fronteira Γ_p do elemento e.

Analogamente, suponhamos que para os nós locais $a = 1, 4$, tenhamos os nós globais associados $A_1, A_4 \in \Gamma_{p4}$. Nesse caso,

$$\begin{aligned} p_1^e &\leftarrow p_1^e M_{11} + p_4^e M_{14} = \frac{dy^e}{6}(2p_1^e + p_4^e), \\ p_2^e &\leftarrow 0, \\ p_3^e &\leftarrow 0, \\ p_4^e &\leftarrow p_1^e M_{41} + p_4^e M_{44} = \frac{dy^e}{6}(p_1^e + 2p_4^e) \end{aligned}$$

e a contribuição do elemento e para a força local é dada por

$$\begin{aligned} F_1^e &\leftarrow f_1^e + p_1^e, \\ F_4^e &\leftarrow f_4^e + p_4^e. \end{aligned}$$

Consideremos agora um exemplo no qual o fluxo é prescrito nos nós locais $a = 3, 4$, associados aos nós globais $A_3, A_4 \in \Gamma_{p3}$. Procedendo de forma análoga, temos

$$\begin{aligned} p_1^e &\leftarrow 0, \\ p_2^e &\leftarrow 0, \\ p_3^e &\leftarrow p_3^e M_{33} + p_4^e M_{34} = \frac{dx^e}{6}(2p_3^e + p_4^e), \\ p_4^e &\leftarrow p_3^e M_{43} + p_4^e M_{44} = \frac{dx^e}{6}(p_3^e + 2p_4^e). \end{aligned}$$

Assim, para os dois nós globais prescritos A_3 e A_4, temos as seguintes contribuições para a força local

$$\begin{aligned} F_3^e &\leftarrow f_3^e + p_3^e, \\ F_4^e &\leftarrow f_4^e + p_4^e. \end{aligned}$$

De forma análoga, considerando os nós locais $a = 1, 2$ associados aos nós globais $A_1, A_2 \in \Gamma_{p1}$, tem-se

$$\begin{aligned} p_1^e &\leftarrow p_1^e M_{11} + p_2^e M_{12} = \frac{dx^e}{6}(2p_1^e + p_2^e), \\ p_2^e &\leftarrow p_1^e M_{21} + p_2^e M_{22} = \frac{dx^e}{6}(p_1^e + 2p_2^e), \\ p_3^e &\leftarrow 0, \\ p_4^e &\leftarrow 0x \end{aligned}$$

de modo que ,

$$\begin{aligned} F_1^e &\leftarrow f_1^e + p_1^e, \\ F_2^e &\leftarrow f_2^e + p_2^e. \end{aligned}$$

Os valores $p_a^e = p(x_a^e, y_a^e) = p(A)$ são prescritos no problema. Quando calculamos no exemplo anterior o valor de p_a^e, consideramos dois nós globais para os quais o fluxo é prescrito. Mas é claro que a quantidade de nós prescritos em cada elemento pode ser diferente e portanto teríamos contribuição em mais ou menos coordenadas na força local F_a^e.

• **Contribuição da Fronteira de Neumann:** `TractionBoundary`

Vimos que a contribuição dos valores de fronteira do tipo de Neumann são definidos pelas matrizes (4.63) e (4.64) quando o fluxo está definido na direção x e y. A partir da identificação de todos os nós de fronteira feita pela subroutina `CondFront`, é necessário então identificar a direção do fluxo para inserir os valores referentes as matrizes correspondentes, ou seja, para calcular o valor de p_a^e, em $\Gamma_1,\ \Gamma_2,\ \Gamma_3,\ \Gamma_4$.0 Esse procedimento é feito pela subroutina `TractionBoundary`, que obviamente so se aplica aos valores de fronteira do tipo Neumann.

• **Cálculo da Matriz M_{ab} com parametrização**

Para o cálculo da matriz M_{ab} definida em (4.61), usamos o fato de que os elementos são retângulos. Mais geralmente, se os elementos da malha são quadriláteros, um dos procedimentos para calcular os elementos da matriz é pela parametrização. Para isso, considere o segmento de reta $(tx_b + (1-t)x_a)$ e a seguinte função de interpolação:

$$\varphi_a\Big(tx_b + (1-t)x_a\Big) = 1 - t.$$

Então,

$$\varphi_a(x) = \begin{cases} 1 & \text{se} \quad x = x_a, \\ 0 & \text{se} \quad x = x_b. \end{cases}$$

Logo,

$$
\begin{aligned}
M_{ab} &= \int_\Gamma \varphi_a^e \varphi_b^e \, d\Gamma = \int_0^1 \varphi_a(tx_b + (1-t)x_a)\ \varphi_b(tx_b + (1-t)x_a)|\Gamma_{ab}|\, dt \\
&= \int_0^1 (1-t)t|\Gamma_{ab}|\, dt = \frac{|\Gamma_{ab}|}{6}, \\
M_{aa} = M_{bb} &= \int_\Gamma \varphi_b^e \varphi_b^e \, d\Gamma = \int_0^1 \varphi_b(tx_b + (1-t)x_a)\ \varphi_b(tx_b + (1-t)x_a)|\Gamma_{ab}|\, dt \\
&= \int_0^1 t^2|\Gamma_{ab}|\, dt = \frac{|\Gamma_{ab}|}{3}.
\end{aligned}
$$

onde $d\Gamma = |\Gamma_{ab}|\, dt$, sendo $|\Gamma_{ab}|$ o comprimento do segmento de reta entre os nós a e b.

Em particular, no caso do retângulo, $|\Gamma_{ab}| = dx^e$ ou $|\Gamma_{ab}| = dy^e$, se o segmento de reta é horizontal ou vertical, respectivamente.

4.4 Construção da Matriz Global e Força Global

A matriz local $K^e = [K_{ab}^e]$ e o vetor força local $F^e = [F_a^e]$, para $e = 1, 2, \ldots, \texttt{Nel}$ e $1 \le a, b \le 4$, são utilizados na obtenção da matriz global e do vetor força global através da relação

$$K = \sum_{e=1}^{\texttt{Nel}} K^e, \quad F = \sum_{e=1}^{\texttt{Nel}} F^e.$$

A forma de alocação dos coeficientes da matriz local e força local para a matriz global K e o vetor força global F é uma das etapas fundamentais do método de elementos finitos.

Como vimos, a matriz local K^e é uma matriz quadrada de ordem 4 (para quadriláteros lineares) e F^e é uma matriz de ordem 4×1. A matriz global K é quadrada com ordem `Neq` e F uma matriz (vetor) com ordem `Neq` $\times 1$, onde `Neq` é o número de equações do sistema linear.

A maneira pela qual as matrizes locais e as forças locais contribuem para a matriz K e o vetor F pode ser descritos nas seguintes etapas:

I) Identificação do nó local (e, a) com o nó global A, usando a subrotina $A =$ `NoLG`(e, a);

II) Identificação do nó global A com `eqn`$[A]$, o número da equação correspondente no sistema através da subrotina `EqNo`.

Como já observamos, o número de equações `Neq` do sistema linear é menor ou igual ao número de nós (`Nno`) A da malha, como consequência das condições de fronteira.

Consideremos um exemplo para ilustrar a montagem da matriz global K e o vetor força global F através da matriz local e força local. Considere o exemplo dado pela Fig. 4.1, onde assumimos os conjuntos de vetores de fronteira:

$$\Gamma_q = \{1, 3, 5, 22, 24\},$$
$$\Gamma_p = \{6, 10, 16, 20\}.$$

Seja $A = \texttt{NoLG}(a, e)$, $A = 1, 2, \ldots, 25$, $a = 1, 2, 3, 4$, $e = 1, 2, \ldots, 16$, então

$$K = \sum_{e=1}^{16} K^e, \quad F = \sum_{e=1}^{16} F^e.$$

Na primeira etapa estabeleceremos uma identificação entre nós locais do elemento e com os nós globais, dados pela tabela

$a\backslash e$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	2	3	4	6	7	8	9	11	12	13	14	16	17	18	19
2	2	3	4	5	7	8	9	10	12	13	14	15	17	18	19	20
3	7	8	9	10	12	13	14	15	17	18	19	20	22	23	24	25
4	6	7	8	9	11	12	13	14	16	17	18	19	21	22	23	24

Na segunda etapa estabeleceremos uma identificação entre os nós globais e o número de equações dados pela tabela

A	1	2	3	4	5	6	7	8	9	10	11	12	13
`eqn`$[A]$	0	1	0	2	0	3	4	5	6	7	8	9	10
A	14	15	16	17	18	19	20	21	22	23	24	25	
`eqn`$[A]$	11	12	13	14	15	16	17	18	0	19	0	20	

Assim o número de equações do sistema linear `Neq` $= 20$, com o número total de nós `Nno` $= 25$, e o número de nós prescritos é igual a 5. Neste caso a matriz global K tem ordem 20×20 e os coeficientes são calculados através do seguinte procedimento, onde somente são listados os coeficientes da banda superior da matriz simétrica que recebem contribuição do nó.

1) Para o nó $A = 1$ o valor já está prescrito e assim este nó não gera a equação;

2) Para o nó $A = 2$, temos a primeira equação do sistema e portanto precisamos calcular a primeira linha do sistema: $K_{11}, K_{12}, \ldots, K_{1,20}$. Mas o nó $A = 2$ pertence

aos elementos $e = 1$ e $e = 2$, mais precisamente $2 = \texttt{NoLG}(1,2) = \texttt{NoLG}(2,1)$. Assim,

$$\begin{aligned} K_{11} &\leftarrow K^1_{22} + K^2_{11}, \\ K_{13} &\leftarrow K^1_{24}, \\ K_{14} &\leftarrow K^1_{23} + K^2_{14}, \\ K_{15} &\leftarrow K^2_{13}. \end{aligned}$$

3) O nó $A = 3$ é prescrito e portanto semelhante ao nó $A = 1$.

4) Para o nó $A = 4$ temos a segunda linha do sistema,

$$\begin{aligned} K_{22} &\leftarrow K^3_{22} + K^4_{11}, \\ K_{25} &\leftarrow K^3_{24}, \\ K_{26} &\leftarrow K^3_{23} + K^4_{14}, \\ K_{27} &\leftarrow K^4_{13}. \end{aligned}$$

5) O nó $A = 5$ é semelhante ao nó $A = 3$.

6) Para o nó $A = 6$ temos a terceira linha do sistema dado por

$$\begin{aligned} K_{33} &\leftarrow K^1_{44} + K^5_{11}, \\ K_{34} &\leftarrow K^1_{34} + K^5_{12}, \\ K_{38} &\leftarrow K^5_{14}, \\ K_{39} &\leftarrow K^5_{13}. \end{aligned}$$

7) Para o nó $A = 7$ temos a 4^a linha do sistema dado por

$$\begin{aligned} K_{44} &\leftarrow K^5_{22} + K^6_{11} + K^1_{33} + K^2_{44}, \\ K_{45} &\leftarrow K^6_{12} + K^2_{34}, \\ K_{48} &\leftarrow K^5_{2,4}, \\ K_{49} &\leftarrow K^5_{23} + K^6_{14}, \\ K_{49} &\leftarrow K^5_{23} + K^6_{14}, \\ K_{4\,10} &\leftarrow K^6_{13}. \end{aligned}$$

8) Para o nó $A = 8$ temos a quinta linha do sistema dada por

$$\begin{aligned} K_{55} &\leftarrow K^2_{33} + K^3_{44} + K^6_{22} + K^7_{11}, \\ K_{56} &\leftarrow K^3_{34} + K^7_{12}, \\ K_{59} &\leftarrow K^6_{24}, \\ K_{5\,10} &\leftarrow K^6_{23} + K^7_{14}, \\ K_{5\,11} &\leftarrow K^7_{13}. \end{aligned}$$

$$
\begin{bmatrix}
* & 0 & * & * & * & 0 & 0 & & & & & & & & & \\
 & * & 0 & 0 & * & * & * & 0 & & & & & & & & \\
 & & * & * & 0 & 0 & 0 & * & * & & & & & & & \\
 & & & * & * & 0 & 0 & * & * & * & & & & & & \\
 & & & & * & * & 0 & 0 & * & * & * & & & & & \\
 & & & & & * & * & 0 & 0 & * & * & * & & & & \\
 & & & & & & * & 0 & 0 & 0 & * & * & 0 & & & \\
 & & & & & & & * & * & 0 & 0 & 0 & * & * & & \\
 & & & & & & & & * & * & 0 & 0 & * & * & * & \\
 & & & & & & & & & * & * & 0 & 0 & * & * & \\
 & & & & & & & & & & * & * & 0 & 0 & * & \\
 & & & & & & & & & & & * & * & 0 & 0 & \\
 & & & & & & & & & & & & * & * & 0 & \\
 & & & & & & & & & & & & & * & * & \\
 & & & & & & & & & & & & & & * & \\
 & & & & & & & & & & & & & & & *
\end{bmatrix}
$$

Figura 4.7: Matriz global K

Assim, sucessivamente, a matriz global K do exemplo pode ser representada na forma da Fig. 4.7, onde os coeficientes com símbolos * são não necessariamente nulos e apenas estão simbolizando os elementos da forma: K_{ij} com $i \geq j$.

De forma análoga, o vetor força F_i é dado por

$$
\begin{aligned}
F_1 &\leftarrow F_2^1 + F_1^2, \\
F_2 &\leftarrow F_2^3 + F_1^4, \\
F_3 &\leftarrow F_4^1 + F_1^5, \\
F_4 &\leftarrow F_3^1 + F_4^2 + F_2^5 + F_1^6, \\
F_5 &\leftarrow F_3^2 + F_4^3 + F_2^6 + F_1^7, \\
F_6 &\leftarrow F_3^3 + F_4^4 + F_2^7 + F_1^8, \\
F_7 &\leftarrow F_3^4 + F_2^8.
\end{aligned}
$$

Assim, sucessivamente, calcula-se $F = [F_1, F_2, \cdots, F_{20}]^T$.

Sendo K_{ab}^e e F_a^e conhecidos, obtemos o sistema linear. O procedimento descrito na subrotina `GlobalSystem` faz a montagem do sistema global em ordem de elementos

pelas contribuições de cada nó local ao seu correspondente número da equação do sistema, `eqn`[`NoLG`(a, e)].

4.5 Resolução do Sistema Linear

Para a resolução de um sistema linear da forma $KC = F$, existem duas classes diferentes de metodologias: Métodos Diretos e os Métodos Iterativos.

Dentre os métodos diretos, os mais conhecidos são:

– Método de eliminação de Gauss.

– Decomposição LU.

– Método de Cholesky.

Dentre os métodos iterativos, os mais conhecidos são:

– Método de Gauss-Jacobi.

– Método de Gauss-Seidel.

– Método do gradiente e gradiente conjugado.

Uma desvantagem dos métodos diretos é a necessidade do armazenamento de matrizes. Porém, têm a vantagem de se obter a solução após um número finito de operações, enquanto que os métodos iterativos não precisam de armazenamento, mas têm a restrição sobre o raio espectral da matriz, que deve ser menor que um para a convergência do processo.

A matriz K, obtida pelo método de elementos finitos, tem uma estrutura especial; é uma matriz do tipo banda que permite uma compactação, minimizando assim o problema de armazenamento. Por essa razão, o sistema linear pode ser resolvido por um método direto.

O método de eliminação de Gauss e da decomposição LU podem ser usados para qualquer matriz K não singular e o número de operações é de ordem $\mathcal{O}(n^3)$. Quando a matriz é do tipo banda, o número de operações diminui significativamente. Por exemplo, se a matriz é tridiagonal, o número de operações é de ordem $\mathcal{O}(n)$.

O método de Cholesky somente é aplicável para matriz simétrica e positiva definida. O número de operações é também de ordem $\mathcal{O}(n^3)$, mas é aproximadamente metade do número de operações do método de eliminação de Gauss.

No problema da equação do calor, a matriz condutividade Q_{ij} é simétrica e positiva definida, o que implica a matriz K também ser positiva definida.

Assumindo que a matriz Q_{ij} seja somente simétrica, a matriz K do sistema linear é uma matriz banda e simétrica, para o qual o método de Cholesky não se aplica. Nesse

caso, a alternativa é usar uma variante da decomposição LU, dita decomposição U^TDU, que utiliza o algoritmo de Crout e que é uma consequência do seguinte resultado.

Teorema 4.2. *Se A é uma matriz simétrica, existe uma matriz triangular superior U com diagonal principal unitária e uma matriz diagonal D tais que $A = U^TDU$.*

Observações:

(1) Se A é positiva definida, os elementos da diagonal D_{ii} são positivos. Além disso,

$$\det(A) = \det(U^T)\det(D)\det(U) = \det(D),$$

o que possibilita a verificação da singularidade da matriz A.

(2) Se A é positiva definida, a decomposição de Cholesky coincide com a decomposição de Crout, bastando tomar $L = D^{1/2}U$. De fato,

$$A = U^TDU = U^TD^{1/2}D^{1/2}U = (D^{1/2}U)^T(D^{1/2}U) = L^TL.$$

(3) Uma matriz A não singular pode ser decomposta na forma $A = \widehat{L}\widehat{U}$, onde $\widehat{L}$ é uma matriz triangular inferior com diagonal unitária e $\widehat{U}$ uma matriz triangular superior. Além disso, se A é simétrica,

$$LU = A = A^T = (\widehat{L}\widehat{U})^T = \widehat{U}^T\widehat{L}^T = U^TDU$$

isto é, a decomposição de Crout é um caso particular da decomposição $\widehat{L}\widehat{U}$ no caso simétrico, onde

$$\widehat{U} = U, \quad \widehat{L}^T = DU.$$

(4) Na decomposição de Crout, somente é necessário o armazenamento da matriz triangular superior U, que pode ser compactada em consequência da característica da matriz A. Como a matriz D é uma matriz diagonal e as diagonais da matriz U são unitárias, pode-se armazenar na diagonal de U os elementos da matriz D.

(5) Após a decomposição de Crout, o sistema linear $Ax = b$ pode ser resolvido por $Ax = (U^TDU)(x) = (U^TD)y = U^Tz = b$, onde $Ux = y$, $Dy = z$ e $U^Tz = b$.

• **Algoritmo de Crout**

A matriz simétrica A pode ser fatorada por

$$A_{ij} = \sum_{k=1}^{j} U_{ki}D_{kk}U_{kj}, \ 1 \leq i \leq j,$$

onde $U_{ii} = 1$ e $U_{ij} = 0$, se $i > j$. Como a matriz U é triangular superior e $U_{ii} = 1$, para calcular DU é suficiente calcular os elementos U_{ij} para os quais $i < j$.

Para exemplificar, consideremos a matriz A de ordem 4. O algoritmo, que é diferente do Método de Eliminação de Gauss, consiste em fixar as colunas j's e variar as linhas i's até a diagonal. Mais precisamente,

Para $j = 1$ e $i = 1$,

$A_{11} = U_{11}D_{11}U_{11} = D_{11}$.

Para $j = 2$ e $i = 1, 2$,

$A_{12} = U_{11}D_{11}U_{12} + U_{21}D_{22}U_{22} = D_{11}U_{12}$,

$A_{22} = U_{12}D_{11}U_{12} + U_{22}D_{22}U_{22} = U_{12}D_{11}U_{12} + D_{22}$.

Para $j = 3$ e $i = 1, 2, 3$,

$A_{13} = D_{11}U_{13}$,

$A_{23} = U_{12}D_{11}U_{13} + D_{22}U_{23}$,

$A_{33} = U_{13}D_{11}U_{13} + U_{23}D_{22}U_{23} + D_{33}$.

Para $j = 4$ e $i = 1, 2, 3, 4$,

$A_{14} = D_{11}U_{14}$,

$A_{24} = U_{12}D_{11}U_{14} + D_{22}U_{24}$,

$A_{34} = U_{13}D_{11}U_{14} + U_{23}D_{22}U_{24} + D_{33}U_{34}$,

$A_{44} = U_{14}D_{11}U_{14} + U_{24}D_{22}U_{24} + U_{34}D_{33}U_{34} + D_{44}$.

Logo, as matrizes D e U podem ser obtidas pelas relações acima, onde

$$\begin{aligned}
D_{11} &= A_{11}, \\
U_{12} &= A_{12}/D_{11}, \\
D_{22} &= A_{22} - D_{11}U_{12}^2, \\
U_{13} &= A_{13}/D_{11}, \\
U_{23} &= (A_{23} - U_{12}D_{11}U_{13})/D_{22}, \\
D_{33} &= A_{33} - D_{11}U_{13}^2 - D_{22}U_{23}^2, \\
U_{14} &= A_{14}/D_{11}, \\
U_{24} &= (A_{24} - U_{12}D_{11}U_{14})/D_{22}, \\
U_{34} &= (A_{34} - U_{13}D_{11}U_{14} - U_{23}D_{22}U_{24})/D_{33}, \\
U_{44} &= (A_{44} - D_{11}U_{14}^2 - D_{22}U_{24}^2 - D_{33}U_{34}^2)/D_{44},
\end{aligned}$$

usando o fato de $U_{ij} = 0$ para $i > j$ e $U_{ii} = 1$.

Para facilitar a programação computacional, introduz-se uma matriz auxiliar defi-

nida por $L_{ji} = D_{ii}U_{ij}$. Assim, as igualdades acima podem ser escritas por

$$\begin{aligned}
D_{11} &= A_{11}, \\
L_{21} &= A_{12}, \\
U_{12} &= L_{21}/D_{11}, \\
D_{22} &= A_{22} - L_{21}U_{12}, \\
L_{31} &= A_{13}, \\
L_{32} &= A_{23} - U_{12}L_{31}, \\
U_{13} &= L_{31}/D_{11}, \\
U_{23} &= L_{32}/D_{22}, \\
D_{33} &= A_{33} - L_{31}U_{13} - L_{32}U_{23}, \\
L_{41} &= A_{14}, \\
L_{42} &= A_{24} - U_{12}L_{41}, \\
L_{43} &= A_{34} - U_{13}L_{41} - U_{23}L_{42}, \\
U_{14} &= L_{41}/D_{11}, \\
U_{24} &= L_{42}/D_{22}, \\
U_{34} &= L_{43}/D_{33}, \\
D_{44} &= A_{44} - L_{41}U_{14} - L_{42}U_{24} - L_{43}U_{34}.
\end{aligned}$$

O algoritmo acima é conhecido como *algoritmo de Crout* ou algoritmo $U^T DU$. Podemos escrevê-lo na forma forma geral,

$$\begin{aligned}
&\text{para } j = 1, 2, \ldots, n \\
&\quad \text{para } i = 1, 2, \ldots, j-1 \\
&\qquad L_{ji} = A_{ij} - \sum_{k=1}^{i-1} U_{ki}L_{jk} \\
&\qquad U_{ij} = L_{ji}/D_{ii} \\
&\qquad D_{jj} = A_{jj} - \sum_{i=1}^{j-1} L_{ji}U_{ij}
\end{aligned}$$

Consideremos um exemplo do uso do algoritmo: Dada a matriz

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

usando a fatorização acima obtemos

$$U = \begin{bmatrix} 1 & -1 & 0 & 0 \\ & 1 & -1 & 0 \\ & & 1 & -1 \\ & & & 1 \end{bmatrix} \; e \; D = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Assim, verifica-se que $A = U^T D U$ e sendo os elementos da diagonal de U unitários, podemos acoplar a matriz diagonal D na matriz U, isto é,

$$U_{ii} \leftarrow D_{ii}, \; i = 1, 2, \ldots$$

Dessa forma, não é necessário o armazenamento da matriz D. Além disso, para evitar a proliferação de armazenamento de matrizes, a matriz U pode ser substituída pela matriz A e a matriz auxiliar L não precisa ser armazenada. Assim, o algoritmo de Crout pode ser reescrito por

$$\begin{array}{l} \text{Para } j = 2, 3, \ldots, n \\ \quad \text{para } i = 2, 3, \ldots, j-1 \\ \qquad A_{ij} \leftarrow A_{ij} - \displaystyle\sum_{k=1}^{i-1} A_{ki} A_{kj} \\ \quad \text{para } i = 1, 2, \ldots, j-1 \\ \qquad T \leftarrow A_{ij} \\ \qquad A_{ij} \leftarrow T / A_{ii} \\ \qquad A_{jj} \leftarrow A_{jj} - T A_{ij} \end{array}$$

No final do procedimento obtemos uma matriz triangular superior A_{ij}, onde os elementos da diagonal A_{ii} são os elementos da matriz diagonal D.

Para a matriz Global K obtida pelo método de elementos finitos, algumas modificações são feitas no algoritmo de Crout a fim de otimizá-lo, pois ela é uma matriz banda, como veremos adiante.

Consideremos o algoritmo de Crout para o sistema linear $Ax = b$, dado por

$$Ax = (U^T D U)x = b.$$

Então, determinar a solução $x = (x_1, x_2, \ldots, x_n)$ é equivalente a resolver sequencialmente os três sistemas lineares

$$U^T z = b, \quad Dy = z, \quad Ux = y.$$

Etapa 1. O sistema $U^T z = b$ pode ser escrito por

$$\sum_{i=1}^{n} U_{ij} z_i = b_j.$$

Como a matriz U^T é triangular inferior, a solução $z = (z_1, z_2, \ldots, z_n)$ é dada por

$$\begin{aligned} &z_1 = b_1 \\ &\text{para } j = 2, 3, \ldots, n \\ &\quad z_j = b_j - \sum_{i=1}^{j-1} U_{ij} z_i \end{aligned}$$

Podemos eliminar a variável z no algoritmo acima através de

$$\begin{aligned} &\text{para } j = 2, 3, \ldots, n \\ &\quad b_j = b_j - \sum_{i=1}^{j-1} A_{ij} b_i \end{aligned}$$

Etapa 2. No sistema $Dy = z$, a matriz D é uma matriz diagonal,

$$D_{jj} y_j = z_j.$$

Portanto, fazemos

$$\begin{aligned} &\text{para } j = 1, 2, \ldots, n \\ &\quad b_j \leftarrow \frac{b_j}{A_{jj}}. \end{aligned}$$

Se $A_{jj} = 0$, a matriz original A é singular e consequentemente o sistema linear não tem solução única.

Etapa 3. Sistema $Ux = y$ pode ser escrito por

$$\sum_{j=1}^{n} U_{ij} x_j = y_i$$

Como a matriz U é triangular superior, o sistema pode ser resolvido por retrosubstituição, dado por

$$\begin{aligned} &x_n = y_n \\ &\text{para } i = n-1, n-2, \ldots, 1 \\ &\quad x_i = y_i - \sum_{j=i+1}^{n} U_{ij} x_j \end{aligned}$$

ou na forma equivalente

$$\begin{aligned} &\text{Para } j = n, n-1, \ldots, 2 \\ &\quad \text{para } i = 1, 2, \ldots, j-1 \\ &\quad\quad b_i \leftarrow b_i - A_{ij} b_j \end{aligned}$$

Assim, a solução x do sistema linear $Ax = b$ é armazenada no vetor b_j e a solução é obtida sem a necessidade das variáveis U, z, y e x.

A fatorização de Crout envolve aproximadamente m^2n operações, onde m é a metade da banda, sendo a banda definida como a soma das colunas dividido pelo número de equações. Assim $1 \ll m \ll n$. A resolução do sistema linear, após a fatorização, envolve $2mn$ operações.

4.5.1 Sistema Linear Global

Interessa-nos resolver o sistema linear $KC = F$ obtido pelo método de elementos finitos com sa aplicação do algoritmo de Crout. A matriz K tem a propriedade de ser esparsa, e com isso podemos fazer algumas modificações no algoritmo de Crout para evitar cálculos desnecessários com zeros. A estrutura da matriz K, quando utilizamos quadriláteros lineares, pode ser representada na forma mostrada na pag. 135, onde as diagonais representam elementos da matriz possivelmente diferentes de zero e estes elementos são isométricos em relação à diagonal principal.

Quando se usa o Algoritmo de Crout, a região triangular q não é alterada para a obtenção da matriz triangular superior. Sendo assim, os elementos nulos de K_{ij} pertencentes à região q não precisam ser computados e nem armazenados. Os demais elementos sofrerão modificações, inclusive os elementos nulos da região p.

Pela necessidade de cálculos na região triangular q, introduzimos a variável `band` (banda), definida por $\texttt{band} = \texttt{Nelx} + 3$, onde `Nelx` é o número de elementos (quadriláteros) na direção x. No final do procedimento obtemos a matriz triangular superior com R diagonais que são os elementos armazenados para se obter a solução do sistema linear. Este procedimento é encontrado na primeira parte da subrotina `Solver`.

A partir da fatorização da matriz K, o processo de obtenção da solução é simples, bastando usar o algoritmo anteriormente descrito. Os elementos $K_{ij} \in R$ são reordenados em colunas, assim a matriz é de ordem $\texttt{Neq} \times R$.

Resolvendo o sistema linear $KC = F$, obtém-se o vetor solução $C = (C_1, \cdots, C_{\texttt{Neq}})$. Como a solução aproximada é dada por

$$u_h(x) = w_h(x) + q^h(x) = \begin{cases} w_h(x) & \text{se } x \in \Omega/\Gamma_q, \\ q^h(x) & \text{se } x \in \Gamma_q \end{cases}$$

e a associação entre o nó global A e a sua correspondente equação I no sistema é dada por

$$I = \texttt{eqn}(A), \quad 1 \leq I \leq \texttt{Neq},$$

tem-se

$$u_h(A) = \begin{cases} C_I & \text{se } A \in N/N_q, \\ q_A & \text{se } A \in N_q \end{cases}$$

e a solução aproximada $u_h(x)$ é dada por

$$u_h(x) = \sum_{A=1}^{\texttt{Nno}} u_h(A)\varphi_A(x).$$

4.5.2 Erro da Solução Numérica

Para o cálculo numérico do erro nas normas dos espaços L^2 e H^1 ou H_0^1, precisamos da solução exata, conhecida a priori em alguns casos particulares, assim como faz necessário o cálculo do gradiente de funções da forma

$$v(x) = \sum_{A=1}^{\texttt{Nno}} d_A\varphi_A(x), \quad \text{isto é,} \quad \nabla v(x) = \sum_{A=1}^{\texttt{Nno}} d_A\nabla\varphi_A(x).$$

Esse gradiente é calculado pela subrotina `Dphi`.

Denotemos por $\overline{u}$ e u as soluções exata e aproximada, e respectivamente e por $\nabla\overline{u}$ e ∇u seus gradientes. Definimos o erro E na norma L^2 e na seminorma H^1 por

$$\|E\|_0 = \left(\int_\Omega |u(x) - \overline{u}(x)|^2\, dx\right)^{1/2},$$
$$\|E\|_1 = \left(\int_\Omega |\nabla u(x) - \nabla\overline{u}(x)|^2\, dx\right)^{1/2}$$

O cálculo das normas pode então ser efetuado por um método de integração numérica, como por exemplo, a quadratura Gaussiana.

• **Norma L^2 e Seminorma H^1**

A interpolação da função $v(x)$ representada em termos de elementos e coordenadas locais, tem a forma

$$v(x) = \sum_{e=1}^{\texttt{Nel}} \sum_{a=1}^{4} d_a^e\, \varphi_a^e(x).$$

Para o cálculo da norma do L^2, consideremos

$$\|v\|_0^2 = \int_\Omega v(x)^2 dx = \int_\Omega \sum_{e=1}^{\texttt{Nel}} \sum_{a,b=1}^{4} (d_a^e d_b^e)\varphi_a^e(x)\varphi_b^e(x)\, dx$$
$$= \sum_{e=1}^{\texttt{Nel}} \sum_{a,b=1}^{4} \left(d_a^e d_b^e \int_{\Omega_e} \varphi_a^e(x)\varphi_b^e(x)\, dx\right).$$

Sendo

$$\int_{\Omega_e} \varphi_a^e(x)\varphi_b^e(x)\,dx = \int_{\Omega_b} \varphi_a^e(\xi)\varphi_b^e(\xi) J\,d\Omega_b,$$

onde J é o Jacobiano da transformação isoparamétrica entre Ω_e e Ω_b, obtemos

$$\|v\|_0^2 = \sum_{a,b=1}^{4} Q_{ab} \sum_{e=1}^{\texttt{Nel}} J\, d_a^e d_b^e, \tag{4.65}$$

onde os coeficientes Q_{ab} da matriz local foram calculados usando quadratura Gaussiana (veja (4.58)).

Por outro lado, a seminorma é dada por

$$\|v\|_1^2 = \int_\Omega |\nabla v(x)|^2 dx = \sum_{e=1}^{\texttt{Nel}} \sum_{a,b=1}^{4} d_a^e d_b^e \left(\sum_{i=1}^{2} \int_{\Omega_e} \nabla_i \varphi_a^e(x) \nabla_i \varphi_b^e(x)\,dx \right).$$

Usando a transformação isoparamétrica entre Ω_e e Ω_b, temos

$$\int_{\Omega_e} \nabla_i \varphi_a^e \nabla_i \varphi_b^e\,dx = \sum_{i=1}^{2} \frac{4J}{(\Delta x_i^e)^2} \int_{\Omega_b} \frac{\partial \varphi_a}{\partial \xi_i} \frac{\partial \varphi_b}{\partial \xi_i}\,d\xi d\eta = \sum_{i=1}^{2} \frac{4J}{(\Delta x_i^e)^2} Q_{abii},$$

onde Q_{abii} é a matriz definida em (4.51) para $i = j$ e $\Delta x_i^e = x_i^e - x_{i-1}^e$.

Assim, temos

$$\|v\|_1^2 = \sum_{i=1}^{2} \sum_{a,b=1}^{4} \sum_{e=1}^{\texttt{Nel}} d_a^e d_b^e \frac{4J}{(\Delta x_i^e)^2} Q_{abii}. \tag{4.66}$$

O cálculo das normas é obtido pela subrotina `Norma`.

4.5.3 Entrada e Saída de Dados

A entrada de dados consiste somente do número de divisão no eixo-x e no eixo-y para a geração da malha e das condições de fronteira.

O programa admite malhas uniformes e não uniformes (geométrica, radical). Em qualquer caso, deve ser dado o número de divisões do intervalo no eixo-x : (`Nelx`) e no eixo-y (`Nely`). Para a malha uniforme é suficiente entrar com o extremo inferior do intervalo do eixo-x e do eixo-y que o programa calcula todos as outras coordenadas uniformemente.

Por exemplo: Consideremos o intervalo $[a, b]$ no eixo x e $[c, d]$ no eixo y. Definimos,

$$h = \frac{b-a}{\texttt{Nelx}}, \quad k = \frac{d-c}{\texttt{Nely}}, \quad x_0 = a, \quad y_0 = c.$$

Então,

$$x_i = x_0 + ih, \quad i = 1, 2, \ldots, \texttt{Nelx},$$
$$y_j = y_0 + jk, \quad j = 1, 2, \ldots, \texttt{Nely}.$$

Obseervemos que, no caso da malha não ser uniforme, devemos entrar com todos os dados das coordenadas x_i e y_j.

Todos os nós A da fronteira da malha são $\texttt{typ}[A] = 1$ (Dirichlet) ou $\texttt{typ}[A] \neq 1$ (Neumann), e a forma de contribuição na força é bastante diferente na subroutina `CondFront`.

No que se refere à saída de dados e a critério do operador, podem ser inseridos dados para a verificação da correção do programa. Este procedimento é aconselhável em várias etapas do programa. No nosso caso, as saídas de dados são:

1. Elementos:

   Todos os elementos com sua conectividade, ou seja, os elementos $e = 1, 2, \ldots, \texttt{Nel}$. com suas coordenadas locais.

2. Condição de Contorno:

   Todos os nós globais são classificados como: nó, tipo, valor.

3. Matriz Rigidez:

   O programa imprime a matriz rigidez K na forma compacta (em colunas) como descrito previamente.

4. Vetor Força:

   O programa imprime o vetor força F, com a contribuição dos valores de fronteira, se houver.

5. Vetor Solução:

   Após o vetor força são impressas as soluções aproximadas do problema.

6. Erro da solução numérica:

   São calculados os erros nas normas L^2 e H^1 ou H_0^1.

4.6 Exemplos Numéricos

Consideremos agora um exemplo numérico para o problema do calor:

$$\begin{cases} \dfrac{\partial}{\partial x_i}\Big(-Q_{ij}\dfrac{\partial u}{\partial x_j}\Big) = f, & \text{em } \Omega, \\ u = q, & \text{em } \Gamma_q, \\ -q_i n_i = Q_{ij}\dfrac{\partial u}{\partial x_j} n_i = p, & \text{em } \Gamma_p, \end{cases}$$

onde a solução exata é conhecida e a solução numérica aproximada é obtida pelo método de elementos finitos. Para o mesmo problema diferentes tipos de fronteira serão testados. Além disso, são calculados e apresentadas os erros nas normas $L^2(\Omega)$ e $H^1(\Omega)$ e a solução gráfica.

Para obtenção do exemplo numérico assumimos

1) $\Omega = [0,1] \times [0,1]$.

2) A matriz simétrica da condutividade será definida por

$$[Q_{ij}] = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

3) A força $f = f(x,y)$ é dada por

$$f(x,y) = 2\pi^2\Big(2\,\mathrm{sen}(\pi x)\cos(\pi y) + \cos(\pi x)\,\mathrm{sen}(\pi y)\Big).$$

Sob estas condições, verifica-se que a função $u = u(x,y)$ dada por

$$u(x,y) = \mathrm{sen}(\pi x)\cos(\pi y)$$

é uma solução exata da equação do calor, independente do tipo de fronteira que definiremos nos três exemplos numéricos que se seguem.

Consideremos as fronteiras denotadas por

$$\begin{aligned} \Gamma_1 &= \{(x,0) \in \partial\Omega;\ 0 \le x \le 1\}, \\ \Gamma_2 &= \{(1,y) \in \partial\Omega;\ 0 \le y \le 1\}, \\ \Gamma_3 &= \{(x,1) \in \partial\Omega;\ 0 \le x \le 1\}, \\ \Gamma_4 &= \{(0,y) \in \partial\Omega;\ 0 \le y \le 1\}. \end{aligned}$$

Exemplo 1. *ronteira de Dirichlet:* Neste caso $\Gamma_p = \emptyset$ e

$$\Gamma_q = \partial\Omega = \bigcup_{i=1}^{4} \Gamma_{q_i}, \quad \Gamma_{q_i} = \Gamma_i,$$

com

$$q(x) = \begin{cases} \operatorname{sen} \pi x & \text{em } \Gamma_{q_1}, \\ 0 & \text{em } \Gamma_{q_2}, \\ -\operatorname{sen} \pi x & \text{em } \Gamma_{q_3}, \\ 0 & \text{em } \Gamma_{q_4}. \end{cases}$$

Exemplo 2. *Fronteira de Neumann:* Neste caso $\Gamma_q = \emptyset$ e

$$\Gamma_p = \partial\Omega = \bigcup_{i=1}^{4} \Gamma_{p_i}, \quad \Gamma_{p_i} = \Gamma_i.$$

A função $p(x)$ em Γ_p é dada por

$$p = Q_{ij} \frac{\partial u}{\partial x_j} n_i.$$

Lembrando que nesse caso o vetor normal unitário n na fronteira é dado por $0, -1)$, $(1, 0)$, $(0, 1)$, $(-1, 0)$ respectivamente em $\Gamma_{p_1}, \Gamma_{p_2}, \Gamma_{p_3}$ e Γ_{p_4}, temos as seguintes condições de fronteira

$$p(x) = \begin{cases} -\pi \cos \pi x & \text{em } \Gamma_{p_1} = \Gamma_1, \\ -2\pi \cos \pi y & \text{em } \Gamma_{p_2} = \Gamma_2, \\ -\pi \cos \pi x & \text{em } \Gamma_{p_3} = \Gamma_3, \\ -2\pi \cos \pi y & \text{em } \Gamma_{p_4} = \Gamma_4. \end{cases}$$

Exemplo 3. *Fronteira mista:* Consideremos a fronteira do tipo Neumann em Γ_1 e Γ_3, dada por

$$p(x) = \begin{cases} -\pi \cos \pi x & \text{em } \Gamma_{p_1} = \Gamma_1, \\ -\pi \cos \pi x & \text{em } \Gamma_{p_3} = \Gamma_3, \end{cases}$$

e em Γ_2 e Γ_4 do tipo Dirichlet, dada por

$$q(x) = \begin{cases} 0 & \text{em } \Gamma_{q_2} = \Gamma_2, \\ 0 & \text{em } \Gamma_{q_4} = \Gamma_4. \end{cases}$$

Neste caso $\Gamma_p \neq \emptyset$ e $\Gamma_q \neq \emptyset$.

Observação: O problema do calor com condições de fronteira do tipo Neumann, como no Exemplo 2, não tem solução única. É fácil ver que se $u = u(x, y)$ é uma solução, então $v(x, y) = u(x, y) + C$ também é soluç˜ ao, qualquer que seja a constante C.

Para que se tenha a unicidade, é necessário impor alguma restrição àa soluções, por exemplo, restringir o conjunto das possíveis soluções ao subespaço

$$V = \{u \in H^1(\Omega);\ \int_\Omega u(x)\,dx = 0\}.$$

Isto significa que se $v(x, y)$ é uma solução qualquer, existe uma constante C, tal que, $u(x, y) = v(x, y) + C$ é uma solução desejada em V. De fato,

Para a solução numérica, consegue-se a unicidade impondo um dado valor em algum ponto da malha, como por exemplo,

$$u(\bar{x}, \bar{y}) = \alpha, \quad (\bar{x}, \bar{y}) \in \partial\Omega, \quad \alpha \text{ fixo.}$$

Por ser mais simples, este é o adotado no programa computacional.

Vale observar ainda a não unicidade da solução do problema discreto com condição de fronteira de Neumann, também pode ser verificada no do sistema linear.

• Etapas da Solução Numérica

Para melhor compreensão do texto, calculamos a solução numérica nos 3 exemplos dados, desenvolvendo todas as etapas na seguinte ordem:

1. Geração e enumeração da malha;
2. Condições de fronteira;
3. Matriz local e força local;
4. Matriz global e força global;
5. Resolução do sistema linear;
6. Estimativa de erro.

Exemplo 1

1. Geração e enumeração da malha:

Consideremos $\Omega = [0, 1] \times [0, 1]$ e a discretização uniforme em ambos os eixos, com passo $h = 1/\texttt{Nelx}$ e $k = 1/\texttt{Nely}$. Logo, o número de elementos e na direção x e direção y é igual a 4, isto é, $\texttt{Nelx} = \texttt{Nely} = 4$. Consequentemente, a malha tem 5 nós no eixo x e 5 nós no eixo y e portanto, o número total de nós $\texttt{Nno} = 25$ e o número total de elementos $\texttt{Nel} = 16$, os quais são enumerados em ordem crescente da esquerda para a direita. Geometricamente, temos a malha mostrada na Fig. 4.1.

2. Condição de fronteira:

Analisemos inicialmente a condição de fronteira de Dirichlet. Os valores de fronteira são inseridos pelo operador na subrotina `CondFront`, da seguinte forma: as fronteiras Γ_1, Γ_2, Γ_3 e Γ_4 são representados respectivamente dentro da subrotina por $bdy[1]$, $bdy[2]$, $bdy[3]$ e $bdy[4]$, os valores da fronteira $q(x)$ são definidos por

$$q(x) = \begin{cases} \operatorname{sen} \pi x, & \text{em } \Gamma_{q_1} = \Gamma_1, \\ 0, & \text{em } \Gamma_{q_2} = \Gamma_2, \\ -\operatorname{sen} \pi x, & \text{em } \Gamma_{q_3} = \Gamma_3, \\ 0, & \text{em } \Gamma_{q_4} = \Gamma_4. \end{cases}$$

são inseridos da seguinte forma:

$$\begin{aligned} \Gamma_1 &= \begin{cases} \texttt{n} = \texttt{bdy}[1][\texttt{i}], \quad p = \texttt{NoPos}(n), \\ \texttt{typ}[\texttt{n}] = 1, \ \texttt{u}[\texttt{n}] = sin(\pi * x[p.v[0]]); \end{cases} \\ \Gamma_2 &= \begin{cases} \texttt{n} = \texttt{bdy}[2][\texttt{i}], \quad p = \texttt{NoPos}(n), \\ \texttt{typ}[\texttt{n}] = 1, \ \texttt{u}[\texttt{n}] = 0; \end{cases} \\ \Gamma_3 &= \begin{cases} \texttt{n} = \texttt{bdy}[3][\texttt{i}], \quad p = \texttt{NoPos}(n), \\ \texttt{typ}[\texttt{n}] = 1, \ \texttt{u}[\texttt{n}] = sin(\pi * x[p.v[0]]); \end{cases} \\ \Gamma_4 &= \begin{cases} \texttt{n} = \texttt{bdy}[4][\texttt{i}], \quad p = \texttt{NoPos}(n), \\ \texttt{typ}[\texttt{n}] = 1, \ \texttt{u}[\texttt{n}] = 0. \end{cases} \end{aligned}$$

Como todos os nós da fronteira são prescritos a solução `u[n]` é conhecida nesses nós. O índice $[i]$ da função `bdy` percorre todos os nós de cada fronteira, identificando-os com os nós globais.

Em Γ_{q_1}, temos os seguintes nós globais $A = \{1, 2, 3, 4, 5\}$. Logo, os respectivos valores são:

$$q_A(x) = \{0;\ \sqrt{2}/2;\ 1;\ \sqrt{2}/2;\ 0\}$$

Em Γ_{q_2}, temos os nós globais $A = \{10, 15, 20\} \Rightarrow q_A(x) \equiv 0$.
Em Γ_{q_3}, temos os nós globais $A = \{21, 22, 23, 24, 25\}$, com os valores:

$$q_A(x) = \{0;\ -\sqrt{2}/2;\ -1;\ -\sqrt{2}/2;\ 0\}$$

Em Γ_{q_4}, temos os nós globais $A = \{6, 11, 16\} \Rightarrow q_A(x) \equiv 0$.

Temos, portanto, 16 nós para o qual o valor da solução é conhecido. Assim, precisamos determinar a solução numérica em 9 nós a saber:

$$\{7, 8, 9, 12, 13, 14, 17, 18, 19\}.$$

Os valores e o tipo de fronteira são introduzidos como dados na subrotina `CondFront`, onde relembramos que os nós A da fronteira do tipo $\texttt{typ}[A] = 1$ são os nós cujos valores são prescritos e a subrotina $\texttt{NoPos}(A)$ tem a função de identificar o nó A com a posição da malha para que seja possível o cálculo da função $q(x) = q(x, y)$ definida na fronteira Γ_q. Em particular, no Exemplo 1, $q(x) = \pm sen(\pi x)$. A subrotina `PosNo` tem a função inversa. Identificando todos os nós da malha cujo valor de fronteira é prescrito, o próximo passo é estabelecer o número de incógnitas (equações) do sistema linear, pois não necessariamente o nó A corresponde a A-ésima equação. A subrotina `EqNo` tem esta função. Dessa forma, para o Exemplo 1, as incógnitas são os nós $A = \{7, 8, 9, 12, 13, 14, 17, 18, 19\}$, que correspondem às equações $\texttt{eqn}[A] = 1, 2, \ldots, 9$ da matriz rigidez do sistema linear, sendo que, para os outros nós, a solução é conhecida. Assim, $Neq = 9$ e a matriz rigidez $[K]_{9\times 9}$.

3. Matriz local e força local:

Para cada elemento $e = 1, 2, 3, .., 16$, é calculada a matriz local K^e_{ab} dada por (4.52). Como estamos usando uma malha uniforme, o Jacobiano $\dot{j}$ é constante e, além disso, $\Delta x_k = \Delta x_l = 1/4$.

Fazendo o cálculo, obtém-se para cada elemento a matriz local,

$$K^e_{ab} = \frac{1}{6}\begin{bmatrix} 11 & -2 & -7 & -2 \\ -2 & 5 & -2 & -1 \\ -7 & -2 & 11 & -2 \\ -2 & -1 & -2 & 5 \end{bmatrix}$$

A força local F^e_a é definida em (4.54) por

$$F^e_a = f^e_a + p^e_a - q^e_a,$$

onde f^e_a, p^e_a e q^e_a são definidos, respectivamente, por (4.55), (4.56) e (4.57).

No Exemplo 1, $\Gamma_p = \phi$, logo, $p^e_a = 0$. Assim,

$$F^e_a = f^e_a - q^e_a.$$

Os elementos $e = \{6, 7, 10, 11\}$ não possuem nós na fronteira. Logo, para esses elementos

$$F^e_a = f^e_a.$$

Para todos os outros elementos restantes o vetor q^e_a contribui para a força local F^e_a, $a = 1, 2, 3, 4$. Para $e = 5$ e $e = 6$ o vetor força local são dados por

$$F^5_a = \begin{bmatrix} 0.292508 \\ 0.336815 \\ 0.292508 \\ 0.285411 \end{bmatrix}, \qquad F^6_a = \begin{bmatrix} 0.351010 \\ 0.320898 \\ 0.196797 \\ 0.248201 \end{bmatrix}.$$

4. Matriz global e força global:

A matriz Global K e o vetor Força F são dados por

$$K = \sum_{e=1}^{\texttt{Nel}} K^e, \qquad F = \sum_{e=1}^{\texttt{Nel}} F^e.$$

Utilizando apropriadamente a contribuição simétrica das matrizes locais e das forças locais, obtém-se para o Exemplo 1, a matriz rigidez K simétrica dada por

$$K = \frac{1}{6}\begin{bmatrix} 32 & -4 & 0 & -4 & -7 & 0 & 0 & 0 & 0 \\ & 32 & -4 & -1 & -4 & -7 & 0 & 0 & 0 \\ & & 32 & 0 & -1 & -4 & 0 & 0 & 0 \\ & & & 32 & -4 & 0 & -4 & -7 & 0 \\ & & & & 32 & -4 & -1 & -4 & -7 \\ & & & & & 32 & 0 & -1 & -4 \\ & & & & & & 32 & -4 & 0 \\ & & & & & & & 32 & -4 \\ & & & & & & & & 32 \end{bmatrix},$$

e o vetor força F dada por

$$F = \begin{bmatrix} +2.14492 \\ +3.03014 \\ +2.14035 \\ +0.71033 \\ +0.00000 \\ -0.71033 \\ -2.14035 \\ -3.03014 \\ -2.14492 \end{bmatrix}.$$

5. Resolução do sistema linear:

Resolvendo o sistema linear $Kd = F$, obtém-se a solução numérica aproximada

$$d = \begin{bmatrix} +0.48886 \\ +0.68935 \\ +0.48696 \\ +0.00417 \\ +0.00000 \\ -0.00417 \\ -0.48696 \\ -0.68935 \\ -0.48886 \end{bmatrix}$$

que corresponde, respectivamente, a solução u nos nós

$$A = \{7, 8, 9, 12, 13, 14, 17, 18, 19\}.$$

6. Cálculo do Erro:

A solução exata do problema é dada por

$$u(x, y) = sen\pi x \cdot \cos \pi y, \quad 0 \leq x \leq 1 \quad \text{e} \quad 0 \leq y \leq 1.$$

Comparando com a solução numérica aproximada $u_h(x, y)$ na norma do L^2 e H^1, obtemos

$$\|E\|_{L^2(\Omega)} = \|u - u_h\|_{L^2} = 6.853220 \times 10^{-3}$$
$$\|E\|_{H^1} = \|u - u_h\|_{H^1} = 5.253452 \times 10^{-2}$$

Para o mesmo problema, usando `Nelx` = `Nely` = $8 \Rightarrow h = k = 1/8$ e `Nelx` = `Nely` = $16 \Rightarrow h = k = 1/16$, que correspondem respectivamente a $\{Nno = 81;\ Nel = 64;\ Neq = 49\}$ e $\{Nno = 289;\ Nel = 256;\ Neq = 225\}$ obtém-se os erros:

$$\|E\|_{L^2} = 2.185674 \times 10^{-3}, \quad \|E\|_{H^1} = 1.601428 \times 10^{-2},$$
$$\|E\|_{L^2} = 5.785226 \times 10^{-4}, \quad \|E\|_{H^1} = 4.192837 \times 10^{-3}.$$

Exemplo 2

1. Condição de fronteira:

Considere a malha para `Nelx` = `Nely` = 4, como anteriormente. Para este exemplo $\Gamma_p = \partial\Omega$ e $\Gamma_q = \phi$. Mostramos que se $\Gamma_q = \phi$ então o problema não tem solução única. Assim fixamos um valor para a solução na fronteira, por exemplo,

$$u(0, 0) = 0$$

Em termos computacionais, esta condição significa que o nó $A = 1$, no qual está associado o par ordenado $(0, 0)$, é do tipo $\texttt{typ}[1] = 1$ e que $q(1) = 0$.

Qualquer um dos 25 nós poderia ser fixado. O número total de nós da malha é $\texttt{Nno} = 25$. Em Γ_p a derivada da solução (fluxo) é prescrita, e portanto, devemos determinar a solução numérica nos 24 nós, dado que, para o nó $A = 1$ a solução é prescrita.

Para a malha considerada e sob as condições de fronteira, temos os seguintes nós na fronteira:

$$\begin{aligned}
\Gamma_{p_1} &= \{2, 3, 4, 5\}, \\
\Gamma_{p_2} &= \{10, 15, 20\}, \\
\Gamma_{p_3} &= \{21, 22, 23, 24, 25\}, \\
\Gamma_{p_4} &= \{6, 11, 16\}.
\end{aligned}$$

Sabemos que que

$$\begin{aligned}
p &= -\pi \cos(\pi x), \quad \text{em} \quad \Gamma_{p_1} \cup \Gamma_{p_3} = \Gamma_1 \cup \Gamma_3, \\
p &= -2\pi \cos(\pi y), \quad \text{em} \quad \Gamma_{p_2} \cup \Gamma_{p_4} = \Gamma_2 \cup \Gamma_4.
\end{aligned}$$

Na subrotina `CondFront` basta introduzir as funções em Γ_1, Γ_2, Γ_3 e Γ_4, representadas no programa por $\texttt{bdy}[1][i]$, $\texttt{bdy}[2][i]$, $\texttt{bdy}[3][i]$ e $\texttt{bdy}[4][i]$, respectivamente. Para esse exemplo, temos

$$\begin{aligned}
\Gamma_1: &\quad n = \texttt{bdy}[1][i], \quad p = \texttt{NoPos}(n), \quad \texttt{Bv}[1][i] = -\pi * \cos(\pi * x[p.v[0]]), \\
\Gamma_2: &\quad n = \texttt{bdy}[2][i], \quad p = \texttt{NoPos}(n), \quad \texttt{Bv}[2][i] = -2\pi * \cos(\pi * y[p.v[1]]), \\
\Gamma_3: &\quad n = \texttt{bdy}[3][i], \quad p = \texttt{NoPos}(n), \quad \texttt{Bv}[3][i] = -\pi * \cos(\pi * x[p.v[0]]), \\
\Gamma_4: &\quad n = \texttt{bdy}[4][i], \quad p = \texttt{NoPos}(n), \quad \texttt{Bv}[4][i] = -2\pi * \cos(\pi * y[p.v[1]]).
\end{aligned}$$

Quando a função `Bv` está recebendo valores, significa que existe alguma fronteira Γ_i do tipo Neumann.

Para garantir a unicidade, após a identificação de todos os nós da fronteira e seus valores, deve ser fixado pelo menos um ponto. Assim por exemplo, se queremos fixar, entre as infinitas soluções, aquela cujo valor no nó global $A = 1$ seja zero $(u(A) = 0)$, devemos inserir na subrotina `CondFront` do programa, após Γ_4, como segue

$$\texttt{typ}[1] = 1, \quad \texttt{u}[1] = 0.$$

2. Matriz local e Força local

A matriz local é constante para todos os elementos e é a mesma do Exemplo 1. Para a força local tem-se

$$F_a^e = f_a^e + p_a^e - q_a^e.$$

Em particular, para os elementos $e = 6, 7, 8, 10, 11, 12$

$$F_a^e = f_a^e,$$

desde que não exista nó global em Γ_p ou Γ_q. Para o elemento $e = 1$, os nós globais $A = 2$ e 6 pertencem a Γ_p e o nó $A = 1$ pertence a Γ_q.

Observe que pela definição da força F_a^e, e desde que $A = 1 \in \Gamma_q$, tem-se

$$F_a^1 = f_a^1 - q_a^1, \quad a = 1, 2, 3, 4.$$

onde neste caso:

$$q_a^1 = \sum_{b=1}^{4} K_{ab}^1 q_b^1 = K_{a1}^1 q_1^1 = 0, \quad a = 1, 2, 3, 4,$$

pois $q_1^1 = 0$, consequência de $u = 0$, correspondente ao nó $A = 1$ prescrito. Para os outros nós $A = 2$ e 6 em Γ_p,

$$F_a^1 = f_a^1 + p_a^1, \quad a = 2, 4$$

e para o nó global $A = 7$, tem-se

$$F_3^1 = f_3^1.$$

As forças locais F_a^e, para $e = 1, 11, 16$, são dadas abaixo, respectivamente:

$$F_a^1 = \begin{bmatrix} 0.196797 \\ 0.135778 \\ 0.351010 \\ -0.122039 \end{bmatrix}, \quad F_a^{11} = \begin{bmatrix} -0.196797 \\ -0.248201 \\ -0.351010 \\ -0.320898 \end{bmatrix}, \quad F_a^{16} = \begin{bmatrix} -0.351010 \\ 0.383838 \\ 0.511922 \\ -0.135778 \end{bmatrix}.$$

3. Estimativa de Erro

A solução exata do problema é a mesma do Exemplo 1. Os erros, neste caso, são dados a seguir:

$$\|E\|_{L^2} = 9.047955 \times 10^{-2},$$
$$\|E\|_{H^1} = 1.797705 \times 10^{-1}.$$

Para esse exemplo temos $Nno = 25$, $Nel = 16$ e $Neq = 24$.

Exemplo 3

1. Geração e enumeração da malha

Consideremos a mesma malha dos exemplos anteriores, com `Nelx` = `Nely` = 4.

2. Condição de fronteira

Neste caso em Γ_p (Neumann) a função definida na fronteira Γ_1 e Γ_3 é dada por

$$p(x) = \begin{cases} -\pi\cos(\pi x) & \text{em } \Gamma_1, \\ -\pi\cos(\pi x) & \text{em } \Gamma_3 \end{cases}$$

e em Γ_2 e Γ_4 (Dirichlet) temos os valores prescritos.

$$q(x) = \begin{cases} 0 & \text{em } \Gamma_2, \\ 0 & \text{em } \Gamma_4 \end{cases}$$

Os valores de fronteira são inseridos pelo operador na subrotina `CondFront`, da seguinte forma: As fronteiras Γ_1, Γ_2, Γ_3 e Γ_4 são representados respectivamente dentro da subrotina por $bdy[1]$, $bdy[2]$, $bdy[3]$ e $bdy[4]$ e dessa forma as fronteiras de Neumann Γ_1 e Γ_3 e as fronteiras do tipo Dirichlet Γ_2 e Γ_4, são inseridos no programa por:

$$\begin{aligned}
&\Gamma_1 = \texttt{n} = \texttt{bdy[1][i]}, \quad p = \texttt{NoPos}(n), \quad \texttt{Bv}[1][i] = -\pi * cos(\pi * x[p.v[0]]); \\
&\Gamma_3 = \texttt{n} = \texttt{bdy[3][i]}, \quad p = \texttt{NoPos}(n), \quad \texttt{Bv}[3][i] = -\pi * cos(\pi * x[p.v[0]]); \\
&\Gamma_2 = \texttt{n} = \texttt{bdy[2][i]}, \quad p = \texttt{NoPos}(n), \quad \texttt{typ}[n] = 1;\ \texttt{u}[n] = 0; \\
&\Gamma_4 = \texttt{n} = \texttt{bdy[4][i]}, \quad p = \texttt{NoPos}(n), \quad \texttt{typ}[n] = 1;\ \texttt{u}[n] = 0.
\end{aligned}$$

Considerando a malha do exemplo, temos os seguintes nós globais em $\Gamma_p = \Gamma_1 \cup \Gamma_3$ e $\Gamma_q = \Gamma_2 \cup \Gamma_4$,

$$\begin{aligned}
\Gamma_p &= \{2, 3, 4, 22, 23, 24\}, \\
\Gamma_q &= \{5, 10, 15, 20, 25, 1, 6, 11, 16, 21, \}.
\end{aligned}$$

O procedimento é análogo aos anteriores. O número de incógnitas nestas condições é de $(25 - 10) = 15$ e portanto a matriz rigidez é de ordem 15×15.

3. Matriz local e Força local

Como nos exemplo anteriores a matriz local não se altera com a introdução de valores de fronteira diferentes, no entanto as forças locais definidas por

$$F_a^e = f_a^e + p_a^e - q_a^e$$

são dependentes dos valores de fronteira. Em particular:

- para os elementos $e = 6, 7, 10, 11$: $F_a^e = f_a^e$,
- para os elementos $e = 2, 3, 14, 15$: $F_a^e = f_a^e + p_a^e$,
- para os elementos $e = 5, 9, 8, 12$: $F_a^e = f_a^e - q_a^e$,
- Para os elementos $e = 1, 4, 13, 16$: $F_a^e = f_a^e + p_a^e - q_a^e$.

Para $e = 1, 11, 16$ temos respectivamente:

$$F_a^1 = \begin{bmatrix} -0.157562 \\ 0.004878 \\ 0.351010 \\ 0.248201 \end{bmatrix}, \quad F_a^{11} = \begin{bmatrix} -0.196797 \\ -0.248201 \\ -0.351010 \\ -0.320898 \end{bmatrix}, \quad F_a^{16} = \begin{bmatrix} -0.351010 \\ -0.248201 \\ 0.1575627 \\ -0.004878 \end{bmatrix}.$$

4. Estimativa de Erro

A solução exata do problema é a mesma do Exemplo 1 e os erros são:

$$\|E\|_{L^2} = 2.658667 \times 10^{-2},$$
$$\|E\|_{H^1} = 1.372426 \times 10^{-1}.$$

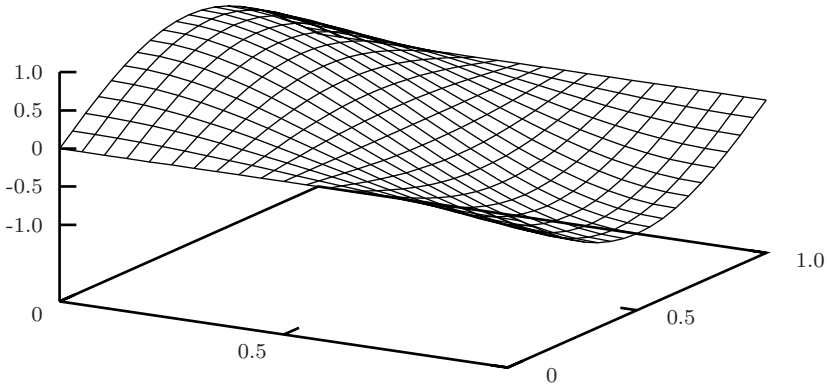
Para esse exemplo temos $\{Nno = 25,\ Nel = 16$ e $Neq = 15\}$. Note que $Neq = Nno - \Gamma_q$.

4.7 Resultados Numéricos

Para os três exemplos, são mostrados a seguir os gráficos das soluções numéricas e os erros em várias malhas diferentes.

Exemplo 1: Problema de Dirichlet

Solução numérica $u_h(x, y)$



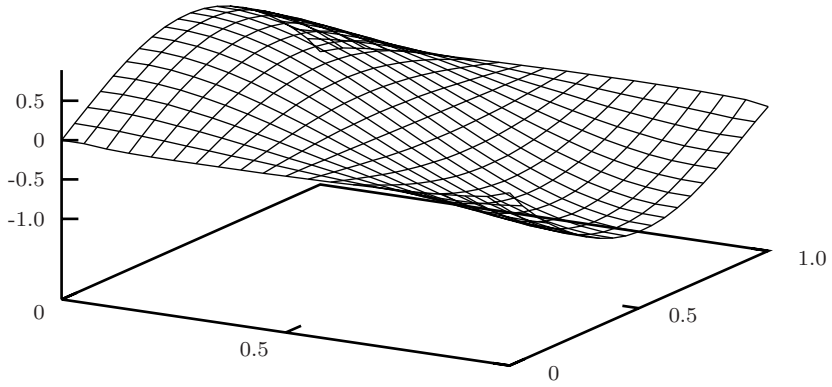
Erros Numéricos

Malha	$\| E \|_0$	$\| E \|_1$
10×10	1.44×10^{-3}	1.05×10^{-2}
20×20	3.73×10^{-4}	2.70×10^{-3}
20×25	3.08×10^{-4}	2.22×10^{-3}
25×20	3.06×10^{-4}	2.22×10^{-3}
30×30	1.69×10^{-4}	1.21×10^{-3}
40×40	9.59×10^{-5}	6.92×10^{-4}

Para a malha 40×40 tem-se $\{Nno = 1681;\ Nel = 1600;\ Neq = 1521\}$

Exemplo 2: Problema de Neumann

Solução numérica $u_h(x, y)$



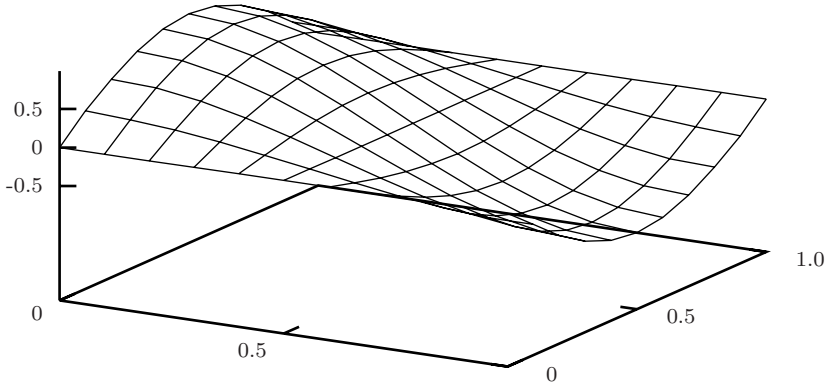
Erros numéricos

Malha	$\| E \|_0$	$\| E \|_1$
10×10	1.56×10^{-2}	3.17×10^{-2}
20×20	3.92×10^{-3}	8.04×10^{-3}
20×25	3.02×10^{-3}	6.47×10^{-3}
25×20	3.45×10^{-3}	6.87×10^{-3}
30×30	1.72×10^{-3}	3.59×10^{-3}
40×40	9.0×10^{-4}	2.01×10^{-3}

Para a malha 40×40 tem-se $\{Nno = 1681;\ Nel = 1600;\ Neq = 1680\}$

Exemplo 3: Problema misto

Solução numérica $u_h(x, y)$



Erros numéricos

Malha	$\| E \|_0$	$\| E \|_1$
10×10	5.04×10^{-3}	2.68×10^{-2}
20×20	1.29×10^{-3}	6.77×10^{-3}
20×25	1.04×10^{-3}	5.26×10^{-3}
25×20	1.09×10^{-3}	5.89×10^{-3}
30×30	5.81×10^{-4}	3.04×10^{-3}
40×40	3.31×10^{-4}	1.73×10^{-3}

Para a malha 40×40 tem-se $\{Nno = 1681;\ Nel = 1600;\ Neq = 1599\}$

4.8 Exercícios

Na literatura, mais usual discretizar o domínio Ω por elementos triangulares Ω_e, visto que sua geometria se ajusta melhor aos domínios mais gerais do que os elementos retangulares. Considere então a formulação variacional (4.8) do problema I e o programa computacional `Ncalor07.c`.

1. Faça a discretização uniforme do domínio retangular $[a, b] \times [c, d]$ por elementos triangulares Ω_e, refazendo as seguintes subrotinas:

   {`NoPos, PosNo, ElmPos, NoLG, EqNo, CondFront`}.

   Sugestão: Enumere os elementos triangulares dentro do domínio de tal forma que os nós globais sejam consecutivos e crescente nas diagonais.

2. Determine o mapeamento (transformação isoparamétrica) entre os domínios Ω_e e Ω_b, onde Ω_b tem como vértices $\xi_1 = (0, 0)$, $\xi_2 = (1, 0)$ e $\xi_3 = (0, 1)$.

3. Calcule o gradiente da função de interpolação.

4. Determine a matriz rigidez local e a força local, obedecendo os mesmos parâmetros deste capítulo.

5. Construa um algoritmo para obtenção da matriz global e força global.

6. Construa um algoritmo para resolver o sistema linear, considerando a esparsidade da matriz.

7. Refaça o Exemplo $1, 2$ e 3 obtendo a solução numérica aproximada, com os mesmos h e k da discretização do eixo x e eixo y, respectivamente. Observe que o número de elementos duplicará. Compare com a solução exata dada. Que elemento Ω_b é melhor?

CAPÍTULO 5

Problema de Elasticidade Linear - Caso Bidimensional

O método de elementos finitos será aplicado neste capítulo ao caso bidimensional do problema modelo de elasticidade linear. Neste caso, temos uma solução vetorial, de modo que algumas mudanças têm que ser introduzidas em relação ao que foi desenvolvido no capítulo anterior. De forma análoga, mostraremos as várias etapas da elaboração do programa computacional, exemplos numéricos, erros e solução gráfica. O programa computacional utilizado encontra-se no apêndice deste texto.

5.1 Formulação do Problema

Consideremos $\Omega \subset \mathbb{R}^2$ um conjunto aberto e limitado com fronteira Γ suave e os índices $1 \leq i, j, k, \ell \leq 2$. Denotemos por:

$$\begin{array}{lcl} \sigma & : & \text{tensor de tensão de Cauchy } \sigma = (\sigma_{ij}) : \Omega \to \mathbb{R}^2 \times \mathbb{R}^2; \\ \boldsymbol{u} & : & \text{vetor deslocamento } \boldsymbol{u} = (u_i) : \Omega \to \mathbb{R}^2; \\ \boldsymbol{f} & : & \text{força prescrita por unidade de volume } \boldsymbol{f} = (f_i) : \Omega \to \mathbb{R}^2; \\ \epsilon & : & \text{tensor de deformação infinitesimal } \epsilon = (\epsilon_{ij}) : \Omega \to \mathbb{R}^2 \times \mathbb{R}^2, \end{array}$$

onde

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{5.1}$$

Pela lei de Hooke, o tensor de deformação infinitesimal ϵ_{ij} e o tensor de tensão σ_{ij} estão relacionados por

$$\sigma_{ij} = C_{ijk\ell}\, \epsilon_{k\ell}, \tag{5.2}$$

onde $C_{ijk\ell}$ é denominado coeficiente de elasticidade definido para todo $x \in \Omega$. Se o corpo é homogêneo, $C_{ijk\ell}$ é constante. Note que usamos a convenção de somatório, isto é, índices repetidos representam um somatório das respectivas grandezas indexadas.

Sobre o tensor de elasticidade, admitimos as seguintes hipóteses:

1. o tensor elasticidade C_{ijkl} satisfaz a condição:

$$C_{ijk\ell} = C_{jik\ell} = C_{ij\ell k} = C_{k\ell ij}; \tag{5.3}$$

2. existem constantes positivas C e $\overline{C}$ tais que, para qualquer matriz simétrica S_{ij}, vale a desigualdade

$$CS_{ij}S_{ij} \leq C_{ijk\ell}S_{ij}S_{k\ell} \leq \overline{C}S_{ij}S_{ij}. \tag{5.4}$$

A hipótese (5.3) é devida à existência da função energia potencial e da simetria dos tensores de tensão e de deformação, enquanto que a hipótese (5.4) estabelece que o tensor elasticidade é limitado e estritamente positivo definido.

No que segue, admitimos que a fronteira Γ de Ω satisfaz às seguintes decomposições:

$$\Gamma = \Gamma_{q1} \cup \Gamma_{p1} = \Gamma_{q2} \cup \Gamma_{p2},$$

de tal forma que, sobre a fronteira Γ_{qi}, a componente i do deslocamento é prescrita; e sobre a fronteira Γ_{pi}, a componente i da tração é prescrita.

Observação: No caso em que as decomposições de Γ são tais que $\Gamma_{pi} \cap \Gamma_{qi} = \emptyset$ (veja Figura **??**), temos mais regularidade na solução do problema. Entretanto, para a formulação fraca do problema contínuo e do problema aproximado, não é necessário impor tal restrição.

Nessas condições, temos a seguinte formulação para o problema elastostático:

(I) Formulação forte. *Dadas as funções* $f_i : \Omega \to \mathbb{R}$, $q_i : \Gamma_{qi} \to \mathbb{R}$ *e* $p_i : \Gamma_{pi} \to \mathbb{R}$, *determinar* $u_i : \overline{\Omega} \to \mathbb{R}$ *tal que*

$$\begin{cases} -\dfrac{\partial \sigma_{ij}}{\partial x_j} = f_i & \text{em } \Omega, \\ u_i = q_i & \text{em } \Gamma_{qi}, \\ \sigma_{ij} n_j = p_i & \text{em } \Gamma_{pi}, \end{cases} \tag{5.5}$$

onde q_i *é a componente* i *do deslocamento prescrito e* p_i *é a componente* i *da tração prescrita na fronteira.*

Se $\cup_{i=1,2}\Gamma_{p_i} = \emptyset$, a condição de fronteira é dita condição do tipo Dirichlet, ou condição de deslocamento na fronteira. Se $\cup_{i=1,2}\Gamma_{q_i} = \emptyset$, a condição de fronteira é dita do tipo Neumann ou condição de tração na fronteira. Se $\Gamma_{p_i} \neq \emptyset$ e $\Gamma_{q_i} \neq \emptyset$, a condição de fronteira é denominada de tipo misto.

Para a descrição da formulação fraca do problema (I), consideremos

$$\boldsymbol{V} := \left\{ \boldsymbol{v} \in H^1(\Omega)^2;\ v_i = 0 \text{ em } \Gamma_{qi} \right\},$$
$$\boldsymbol{H} := \left\{ \boldsymbol{v} \in H^1(\Omega)^2;\ v_i = q_i \text{ em } \Gamma_{qi} \right\},$$

onde $H^1(\Omega)$ é o espaço usual de Sobolev. É claro que $\boldsymbol{V}$ é subespaço vetorial de $H^1(\Omega)^2$, mas $\boldsymbol{H}$ é um subespaço afim de $H^1(\Omega)^2$. É claro também que $\boldsymbol{V}$ é um espaço de Hilbert se munido da norma usual de $H^1(\Omega)^2$, ou seja,

$$\|\boldsymbol{v}\|_{\boldsymbol{V}} := \left(\sum_{i=1}^{2} \|v_i\|^2_{H^1(\Omega)} \right)^{1/2} \quad \text{e} \quad \|v_i\|^2_{H^1(\Omega)} := \int_\Omega \left(|v_i|^2 + \sum_{j=1}^{2} \left| \frac{\partial v_i}{\partial x_j} \right|^2 \right) dx,$$

e $\boldsymbol{H}$ é espaço métrico completo se munido com a mesma norma $\| \ \|_{\boldsymbol{V}}$.

Multipliquemos então a equação (5.5) por $\boldsymbol{v} \in \boldsymbol{V}$ e integremos em Ω. Então, com a convenção usual dos índices repetidos, temos formalmente

$$-\int_\Omega \frac{\partial \sigma_{ij}}{\partial x_j} v_i \, dx = \int_\Omega f_i v_i \, dx. \tag{5.6}$$

Pela regra da derivada do produto de funções, obtemos

$$\int_\Omega \frac{\partial \sigma_{ij}}{\partial x_j} v_i \, dx = \int_\Omega \frac{\partial (\sigma_{ij} v_i)}{\partial x_j} \, dx - \int_\Omega \sigma_{ij} \frac{\partial v_i}{\partial x_j} \, dx.$$

Aplicando o Teorema da Divergência,

$$\int_\Omega \frac{\partial}{\partial x_j} (\sigma_{ij} v_i) \, dx = \int_\Gamma \sigma_{ij} n_j v_i \, dx$$

e substituindo em (5.6), obtemos

$$\int_\Omega \sigma_{ij} \frac{\partial v_i}{\partial x_j} \, dx = \int_\Omega f_i v_i \, d\Omega + \int_\Gamma \sigma_{ij} n_j v_i \, dx.$$

Como $v_i = 0$ em Γ_{qi} e $\sigma_{ij} n_j = p_i$ em Γ_{pi}, tem-se

$$\int_\Omega \sigma_{ij} \frac{\partial v_i}{\partial x_j} \, dx = \int_\Omega f_i v_i \, dx + \int_{\Gamma_{pi}} p_i v_i \, dx.$$

Assim, a formulação fraca é dada por

(II) Formulação fraca. *Dadas as funções* $f_i : \Omega \to \mathbb{R}$, $q_i : \Gamma_{qi} \to \mathbb{R}$ *e* $p_i : \Gamma_{pi} \to \mathbb{R}$, *determinar* $\boldsymbol{u} \in \boldsymbol{H}$ *tal que*

$$\int_\Omega \sigma_{ij} \frac{\partial v_i}{\partial x_j}\, dx = \int_\Omega f_i v_i\, dx + \int_{\Gamma_{pi}} p_i v_i\, dx, \quad \forall\, \boldsymbol{v} \in \boldsymbol{V}.$$

A formulação fraca é também denominada *princípio do trabalho virtual* e $\boldsymbol{v}$ é o *deslocamento virtual*.

De (5.1) e (5.2) e da simetria de $C_{ijk\ell}$, tem-se

$$\sigma_{ij} = C_{ijk\ell}\, \epsilon_{k\ell} = \frac{1}{2} C_{ijk\ell} \left(\frac{\partial u_k}{\partial x_\ell} + \frac{\partial u_\ell}{\partial x_k} \right) = C_{ijk\ell} \frac{\partial u_k}{\partial x_\ell}.$$

Substituindo na formulação fraca, obtemos

$$\int_\Omega C_{ijk\ell} \frac{\partial u_k}{\partial x_\ell} \frac{\partial v_i}{\partial x_j}\, dx = \int_\Omega f_i v_i\, dx + \int_{\Gamma_{pi}} p_i v_i\, dx.$$

Podemos então reescrever a formulação fraca em forma vetorial e manter uma semelhança com a notação dos capítulos anteriores, se introduzimos a seguinte formas (bilinear e lineares):

$$\begin{aligned} a(\boldsymbol{u}, \boldsymbol{v}) &:= \int_\Omega C_{ijk\ell} \frac{\partial u_k}{\partial x_\ell} \frac{\partial v_i}{\partial x_j}\, dx, \\ (\boldsymbol{f} : \boldsymbol{v}) &:= \int_\Omega f_i v_i\, dx, \\ (\boldsymbol{p} : \boldsymbol{v})_{\Gamma_p} &:= \int_{\Gamma_{p1}} p_1 v_1\, ds + \int_{\Gamma_{p21}} p_2 v_2\, ds \end{aligned}$$

onde ds denota a medida de comprimento de arco de curva em $\mathbb{R}^2$ e $\Gamma_p = \Gamma_{p1} \cup \Gamma_{p2}$. Assim, a formulação fraca do problema (II) toma a forma:

(II′) Formulação fraca. *Dados* $\boldsymbol{f}$ *e* $\boldsymbol{p}$, *determinar* $\boldsymbol{u} \in \boldsymbol{H}$ *tal que*

$$a(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{f} : \boldsymbol{v}) + (\boldsymbol{p} : \boldsymbol{v})_{\Gamma_p}, \quad \forall\, \boldsymbol{v} \in \boldsymbol{V}. \tag{5.7}$$

5.1.1 Formulação de Galerkin

Os argumentos do capítulo anterior podem ser adaptados no atual contexto se considerarmos $\boldsymbol{V}_h$ um subespaço de $\boldsymbol{V}$ de dimensão finita e $\boldsymbol{H}_h = \boldsymbol{V}_h + \{\widetilde{\boldsymbol{q}}\}$, onde $\widetilde{\boldsymbol{q}} = (\widetilde{q}_1, \widetilde{q}_2) \in \boldsymbol{H}^1(\Omega)^2$ é tal que $\widetilde{q}_i(x) = q_i(x)$ para todo $x \in \Gamma_{qi}$. A existência de $\widetilde{\boldsymbol{q}}$ é garantida pelo fato de que o operador de traço $\gamma_i : H^1(\Omega)^2 \to H^{1/2}(\Gamma_{qi})$ é sobrejetor.

Assim, se $\boldsymbol{u}_h \in \boldsymbol{H}_h$, a função $\boldsymbol{w}_h$ definida por

$$\boldsymbol{w}_h(x) = \boldsymbol{u}_h(x) - \widetilde{\boldsymbol{q}}(x), \tag{5.8}$$

pertence ao subespaço $\boldsymbol{V}_h$, de modo que $\boldsymbol{w}_h(x) = \boldsymbol{0}$ para $x \in \Gamma_q = \Gamma_{q1} \cup \Gamma_{q2}$.

Da mesma forma, definimos a função $\widetilde{\boldsymbol{p}} = (\widetilde{p}_1, \widetilde{p}_2) \in H^1(\Omega)^2$ tal que $\widetilde{p}_i(x) = p_i(x)$ para $x \in \Gamma_{pi}$. Nessas condições, podemos considerar o problema variacional aproximado associado ao problema (5.7), i.e., a formulação de Galerkin, dada por: *determinar* $\boldsymbol{u}_h \in \boldsymbol{H}_h$ *tal que*

$$a(\boldsymbol{u}_h, \boldsymbol{v}_h) = (\boldsymbol{f} : \boldsymbol{v}_h) + (\boldsymbol{p} : \boldsymbol{v}_h)_{\Gamma_p}, \quad \forall \boldsymbol{v}_h \in \boldsymbol{V}_h,$$

ou de forma equivalente, *determinar* $\boldsymbol{w}_h \in \boldsymbol{V}_h$ *tal que*,

$$a(\boldsymbol{w}_h, \boldsymbol{v}_h) = (\boldsymbol{f}, \boldsymbol{v}_h) + (\boldsymbol{p}, \boldsymbol{v}_h)_{\Gamma_p} - a(\widetilde{\boldsymbol{q}}, \boldsymbol{v}_h), \quad \forall \boldsymbol{v}_h \in \boldsymbol{V}_h. \tag{5.9}$$

De forma análoga ao que apresentamos anteriormente, consideremos N o conjunto de nós da malha e N_q o conjunto de nós sobre Γ_q onde os valores de fronteria são prescritos. Para cada nó A, considere φ_A a função de interpolação, tal como introduzida no capítulo anterior. Podemos então representar a solução aproximada $\boldsymbol{w}_h \in \boldsymbol{V}_h$ por

$$\boldsymbol{w}_h(x) = \sum_{A=1}^{\texttt{Nno}} \boldsymbol{d}_A \varphi_A(x), \quad \boldsymbol{d}_A = (d_{A,1}, \ldots, d_{A,n}). \tag{5.10}$$

A solução aproximada (5.10) é uma função vetorial, ou seja,

$$\boldsymbol{w}_h(x) = (w_{h,1}(x), w_{h,2}(x), \ldots, w_{h,n}(x)), \quad \forall x \in \Omega,$$

de modo que, em termos de coordenadas,

$$w_{h,j}(x) = \sum_{A \in N} d_{A,j} \varphi_A(x) = \sum_{A \in N - N_{q,j}} d_{A,j} \varphi_A(x), \quad 1 \leq j \leq n, \tag{5.11}$$

pois $w_{h,j}(x) = 0$, se $x \in N_{q,j}$.

De forma análoga, consideremos a interpolação da função $\widetilde{\boldsymbol{q}}$, definindo $\boldsymbol{q}_h(x) = (q_{h,1}(x), \ldots, q_{h,n}(x))$, onde, para $j = 1, \ldots, n$ e $x_A \in \Gamma_q$,

$$q_{h,j}(x) = \sum_{A \in N_{q,j}} q_{A,j} \varphi_A(x), \quad q_{A,j} = q_j(x_A). \tag{5.12}$$

Consideremos a base canônica do $\mathbb{R}^2$ dada por $\boldsymbol{e}_j = (0, \ldots, 0, 1, 0, \ldots, 0)$, onde a j-ésima coordenada é única coordenada não nula. Então os vetores $\boldsymbol{w}_h$ e $\boldsymbol{q}_h$ podem ser expressos na forma

$$\boldsymbol{w}_h = w_{h,j} \boldsymbol{e}_j \qquad \boldsymbol{q}_h = q_{h,j} \boldsymbol{e}_j,$$

com $w_{h,j}$ e $q_{h,j}$ são definidos em (5.11) e (5.12). Assim, podemos denotar a forma bilinear $a(\cdot,\cdot)$ por

$$a(w_{h,j}\boldsymbol{e}_j, \boldsymbol{v}_h) = a\left(\sum_{A\in N-N_{q,j}} d_{A,j}\varphi_A \boldsymbol{e}_j, \boldsymbol{v}_h\right) = \sum_{A\in N-N_{q,j}} d_{A,j}\, a(\varphi_A \boldsymbol{e}_j, \boldsymbol{v}_h) \quad \forall\, \boldsymbol{v}_h \in \boldsymbol{V}_h.$$

Como a igualdade é válida para todo $\boldsymbol{v}_h$, podemos tomar em particular $\boldsymbol{v}_h \in \boldsymbol{V}_h$ na forma $\boldsymbol{v}_h = \varphi_B \boldsymbol{e}_i$. Portanto,

$$a(\boldsymbol{w}_h, \varphi_B \boldsymbol{e}_i) = \sum_{j=1}^{n} \sum_{A\in N-N_{qj}} d_{A,j}\, a(\varphi_A \boldsymbol{e}_j, \varphi_B \boldsymbol{e}_i), \quad 1 \le i \le n.$$

Substituindo as relações acima em (5.9), obtemos o sistema linear

$$\sum_{j=1}^{n} \sum_{B\in N-N_{qj}} d_{B,j}\, a(\varphi_A \boldsymbol{e}_i, \varphi_B \boldsymbol{e}_j) = (\boldsymbol{f} : \varphi_A \boldsymbol{e}_i) + (\boldsymbol{p}_h : \varphi_A \boldsymbol{e}_i)_\Gamma - \sum_{j=1}^{n} \sum_{B\in N_{qj}} q_{B,j}\, a(\varphi_A \boldsymbol{e}_i, \varphi_B \boldsymbol{e}_j),$$

para todo $A \in N - N_{qi}$.

Definindo

$$K_{AB} := a(\varphi_A \boldsymbol{e}_i, \varphi_B \boldsymbol{e}_j), \tag{5.13}$$

$$F_A := (\varphi_A \boldsymbol{e}_i : \boldsymbol{f}) + (\varphi_A \boldsymbol{e}_i : \boldsymbol{p}_h)_\Gamma - \sum_{j=1}^{n} \sum_{B\in N_{qj}} a(\varphi_A \boldsymbol{e}_i, \varphi_B \boldsymbol{e}_j) q_{jB}. \tag{5.14}$$

Então,

$$K_{AB}\, d_{jB} = F_A. \tag{5.15}$$

Assim, a solução aproximada $u_h(x)$ tem como coordenadas $u_{h,1}(x)$ e $u_{h,2}(x)$, isto é,

$$\boldsymbol{u}_h(x) = (u_{h,1}(x),\, u_{h,2}(x)),$$

onde $u_{h,1}$ e $u_{h,2}$ são denominados respectivamente deslocamentos horizontal e vertical.

A próxima etapa é estabelecer uma relação entre os nós globais A e o número de equações no sistema linear. Neste sentido introduzimos a subrotina `EqNo` para gerar o número de equação, I, correspondente à componente i do nó global A e `Neq` o número total de equações do sistema linear. Assim,

$$\texttt{eqn}[i, A] = \begin{cases} I & \text{se } \ A \in N - N_{qi} \\ 0 & \text{se } \ A \in N_{qi} \end{cases}$$

onde $N_{qi} = \{A \in \Gamma_{qi}\}$. Sabemos que se $A \in N_{qi}$, então $\boldsymbol{u}_h(x_A) = (u_{h,1}(x_A), u_{h,2}(x_A))$ é conhecida e portanto não é necessário gerar uma equação para cada deslocamento $u_{h,i}$. Poderemos ter nós A prescritos numa direção ou em duas direções.

Por exemplo, na tabela abaixo mostramos o caso em que temos 5 nós globais com 6 equações.

$i\backslash A$	1	2	3	4	5
1	1	2	4	0	0
2	0	3	5	6	0

O nó $A = 1 \in \Gamma_{q2}$, logo $u_{h,1}(x_A)$ não é prescrito, mas $u_{h,2}(x_A)$ é prescrito.

O nó $A = 4 \in \Gamma_{q1}$, logo $u_{h,1}(x_A)$ é prescrito e $u_{h,2}(x_A)$ não é prescrito.

O nó $A = 5 \in \Gamma_{q1} \cap \Gamma_{q2}$, é prescrito em ambas as direções, $u_{h,1}$ e $u_{h,2}$.

Os nós $A = 2, 3 \notin \Gamma_{q1} \cup \Gamma_{q2}$, logo não são prescritos.

Assim, para esse exemplo, temos 6 equações para se determinar a solução aproximada. De um modo geral, definindo

$$M = \texttt{eqn}[i, A] \quad \text{e} \quad N = \texttt{eqn}[j, B],$$

com $1 \leq M, N \leq \texttt{Neq}$, o sistema linear (5.15) pode ser escrito por

$$K_{MN} D_N = F_M,$$

onde

$$K_{MN} =: a(\varphi_A \boldsymbol{e}_i, \varphi_B \boldsymbol{e}_j), \tag{5.16}$$

$$F_M := (\varphi_A \boldsymbol{e}_i : \boldsymbol{f}) + (\varphi_A \boldsymbol{e}_i : \boldsymbol{p}_h)_\Gamma - \sum_{j=1}^{2} \sum_{B \in N_{qj}} a(\varphi_A \boldsymbol{e}_i, \varphi_B \boldsymbol{e}_j) q_{jB} \tag{5.17}$$

A matriz $K = [K_{MN}]$ é denominada matriz rigidez global ou matriz global, $F = [F_M]$ é denominado vetor força e $D = [D_N]$ é o vetor deslocamento (incógnita).

• Propriedades da Matriz Global

1. $\underline{K_{MN} \text{ é simétrica}}$. De fato, como $C_{k\ell ij} = C_{ijk\ell}$, temos

$$\begin{aligned} K_{MN} &= a(\varphi_A \boldsymbol{e}_i, \varphi_B \boldsymbol{e}_k) = \int_\Omega \frac{\partial(\varphi_A \boldsymbol{e}_i)}{\partial x_j} C_{ijk\ell} \frac{\partial(\varphi_B \boldsymbol{e}_k)}{\partial x_\ell} \, d\Omega \\ &= \int_\Omega \frac{\partial \varphi_A}{\partial x_j} \boldsymbol{e}_i C_{ijk\ell} \frac{\partial \varphi_B}{\partial x_\ell} \boldsymbol{e}_k \, d\Omega = \int_\Omega \frac{\partial \varphi_B}{\partial x_\ell} \boldsymbol{e}_k C_{k\ell ij} \frac{\partial \varphi_A}{\partial x_j} \boldsymbol{e}_i \, d\Omega = K_{NM}. \end{aligned}$$

2. $\underline{K_{MN} \text{ é definida positiva}}$. Se definimos $\boldsymbol{d} = (d_1, d_2, \cdots, d_{\texttt{Neq}})$, então

$$\begin{aligned} \boldsymbol{d}^T K \boldsymbol{d} &= \sum_{M,N=1}^{\texttt{Neq}} d_M K_{MN} d_N = \sum_{\substack{A \in N - N_{qi} \\ B \in n - n_{qj}}} d_{iA} a(\varphi_A \boldsymbol{e}_i, \varphi_B \boldsymbol{e}_j) d_{jB} \\ &= a\Big(\sum_{A \in N - N_{qi}} d_{iA} \varphi_A \boldsymbol{e}_i, \sum_{B \in N - N_{qj}} d_{jB} \varphi_B \boldsymbol{e}_j \Big) = a(\boldsymbol{u}_h, \boldsymbol{u}_h) \geq 0. \end{aligned}$$

Por outro lado,

$$0 = a(\boldsymbol{u}_h, \boldsymbol{u}_h) = \int_\Omega \frac{\partial u_{h,i}}{\partial x_j} C_{ijk\ell} \frac{\partial u_{h,k}}{\partial x_\ell} \, dx.$$

De (5.3) e (5.4), $C_{ijk\ell}$ é positivo definido para todas as matrizes simétricas, logo,

$$\frac{\partial u_{h,i}}{\partial x_j} + \frac{\partial u_{h,j}}{\partial x_i} = 0,$$

Isto implica que $u_{h,i}$ é um movimento rígido, que pode ser representado por,

$$u_{h,i} = W_{ij} x_j + C,$$

onde C é uma constante e W_{ij} uma matriz anti-simétrica, cujo valor deveria ser pequeno, representando uma rotação infinitesimal para a teoria linear. Se $\Gamma_q \neq \emptyset$, pois $u_{h,i} = 0$ em Γ_q, então

$$W_{ij} = 0 \qquad \text{e} \qquad C = 0,$$

isto é, o corpo tem um movimento rígido nulo. Assim $u_{h,i} = 0$ e K é uma matriz definida positiva. Caso $\Gamma_q = \emptyset$, então K é uma matriz definida positiva se o corpo satisfaz a condição adicional de ter movimento rígido nulo.

5.2 Matriz Rigidez e Vetor Força

Trocando os índices (i, j), por (r, s) em (5.16) temos

$$K_{AB}^{rs} = a(\varphi_A \boldsymbol{e}_r, \varphi_B \boldsymbol{e}_s), \;\; 1 \leq A, B \leq N - N_{qr}$$

A forma bilinear $a(\cdot, \cdot)$ é definida por

$$a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \frac{\partial u_i}{\partial x_k} C_{ikj\ell} \frac{\partial v_j}{\partial x_\ell} \, dx.$$

Na forma componente, temos

$$\begin{aligned}\boldsymbol{u} = \varphi_A \boldsymbol{e}_r &\Longrightarrow u_i = (u, e_i) = (\varphi_A \boldsymbol{e}_r, \boldsymbol{e}_i) = \varphi_A \delta_{ir} \\ \boldsymbol{v} = \varphi_B \boldsymbol{e}_s &\Longrightarrow v_j = (v, e_j) = (\varphi_B \boldsymbol{e}_s, \boldsymbol{e}_j) = \varphi_B \delta_{js}\end{aligned}$$

Substituindo na forma bilinear temos

$$a(\varphi_A \boldsymbol{e}_r, \varphi_B \boldsymbol{e}_s) = \int_\Omega \frac{\partial \varphi_A}{\partial x_k} \frac{\partial \varphi_B}{\partial x_\ell} C_{ikj\ell} \delta_{ir} \delta_{js}\, dx.$$

Assim, a matriz é dada na forma

$$K_{AB}^{rs} = \delta_{ir}\delta_{js} \int_\Omega C_{ikj\ell} \frac{\partial \varphi_A}{\partial x_k} \frac{\partial \varphi_B}{\partial x_\ell}\, dx = \delta_{rk}\delta_{sl} \int_\Omega C_{ikj\ell} \frac{\partial \varphi_A}{\partial x_i} \frac{\partial \varphi_B}{\partial x_j}\, dx. \tag{5.18}$$

Para o vetor força definido em (5.14) e substituindo os índices (i, j) por (r, s) tem-se

$$F_A^r = (\varphi_A \boldsymbol{e}_r : \boldsymbol{f}) + (\varphi_A \boldsymbol{e}_r : \boldsymbol{p}_h)_\Gamma - \sum_{s=1}^{2} \sum_{B \in N_{qs}} a(\varphi_A \boldsymbol{e}_r, \varphi_B \boldsymbol{e}_s)\, q_{sB},$$

onde

$$\begin{aligned}(\varphi_A \boldsymbol{e}_r : \boldsymbol{f}) &= \int_\Omega \varphi_A (\boldsymbol{f} : \boldsymbol{e}_r)\, dx = \int_\Omega \varphi_A f_r\, dx, \\ (\varphi_A \boldsymbol{e}_r : \boldsymbol{p}_h)_\Gamma &= \int_{\Gamma_{pr}} \varphi_A (\boldsymbol{p}_h : \boldsymbol{e}_r)\, ds = \int_{\Gamma_{pr}} \varphi_A p_r\, ds,\end{aligned}$$

e

$$\sum_{s=1}^{2} \sum_{B \in N_{qs}} a(\varphi_A \boldsymbol{e}_r, \varphi_B \boldsymbol{e}_s) q_{sB} = \sum_{B \in N_{qs}} K_{AB}^{rs}\, q_{sB}.$$

Assim a força é dada por

$$F_A^r = \int_\Omega \varphi_A f_r\, dx + \int_{\Gamma_{pr}} \varphi_A p_r\, d\Gamma - \sum_{s=1}^{2} \sum_{B \in N_{qs}} K_{AB}^{rs}\, q_{sB}. \tag{5.19}$$

De forma análoga ao capítulo anterior, f_r e p_r serão interpolados usando a função φ_A como interpolante, mais precisamente,

$$f_r(x) = \sum_{A \in N - N_{qs}} \varphi_A(x) f_{rA}, \tag{5.20}$$

$$p_r(x) = \sum_{A \in N_{ps}} \varphi_A(x) p_{rA} \tag{5.21},$$

onde $f_{rA} = f_r(\boldsymbol{x}_A)$ e $p_{rA} = p_r(\boldsymbol{x}_A), r = 1, 2$. Substituindo em (5.19) obtemos

$$\begin{aligned} F_A^r &= \sum_{B\in N-N_{qs}} \int_\Omega (\varphi_A\varphi_B) f_{rB}\, dx + \sum_{B\in N_{ps}} \int_{\Gamma_{ps}} (\varphi_A\varphi_B) p_{rB}\, d\Gamma - \sum_{s=1}^{2} \sum_{B\in N_{qs}} K_{AB}^{rs}\, q_{sB} \\ &= \sum_{B\in N-N_{qs}} (\varphi_A : \varphi_B) f_{rB} + \sum_{B\in N_{ps}} (\varphi_A : \varphi_B)_\Gamma\, p_{rB} - \sum_{s=1}^{2} \sum_{B\in N_{qs}} K_{AB}^{rs}\, q_{sB}. \end{aligned} \tag{5.22}$$

De (5.18) e (5.22) o sistema linear global é dado por

$$K_{AB}^{rs} d_{sB} = F_A^r. \tag{5.23}$$

Como $M = \texttt{eqn}[r, A]$ e $N = \texttt{eqn}[s, B]$, o sistema linear pode ser escrito na forma

$$K_{MN} = \delta_{rk}\delta_{s\ell} \int_\Omega C_{ikj\ell} \frac{\partial\varphi_A}{\partial x_i} \frac{\partial\varphi_B}{\partial x_j}\, dx = \int_\Omega C_{irjs} \frac{\partial\varphi_A}{\partial x_i} \frac{\partial\varphi_B}{\partial x_j}\, dx, \tag{5.24}$$

$$F_M = \sum_{B\in N-N_{qs}} (\varphi_A : \varphi_B) f_{rB} + \sum_{B\in N_{ps}} (\varphi_A : \varphi_B)_\Gamma\, p_{rB} - \sum_{B\in N_{qs}} K_{AB}^{rs}\, q_{sB}. \tag{5.25}$$

5.2.1 Matriz Local e Força Local

Considere a discretização do domínio em elementos finitos Ω_e satisfazendo

$$\Omega = \left(\bigcup_{e=1}^{\texttt{Nel}} \overline{\Omega}_e \right)^\circ \quad \text{e} \quad \{x_e\} \cap \Omega_k = \emptyset \quad \text{se} \quad e \neq k.$$

Para cada elemento finito Ω_e temos a função base φ_A^e com suporte compacto em $\overline{\Omega}_e$. Dessa forma, a matriz global e o vetor força podem ser obtidos em termos locais através da soma

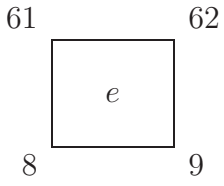
$$K = \sum_{e=1}^{\texttt{Nel}} K^e \quad \text{e} \quad F = \sum_{e=1}^{\texttt{Nel}} F^e,$$

onde $K^e = [K_{MN}^e]$ e $F^e = [F_M^e]$ são dados por

$$\begin{aligned} K_{MN}^e &= \delta_{rk}\delta_{s\ell} \int_{\Omega_e} C_{ikj\ell} \frac{\partial\varphi_A}{\partial x_i} \frac{\partial\varphi_B}{\partial x_j}\, dx_e = \int_{\Omega_e} C_{irjs} \frac{\partial\varphi_A}{\partial x_i} \frac{\partial\varphi_B}{\partial x_j} dx_e, \\ F_M^e &= \sum_{B\in N-N_{qs}} \int_{\Omega_e} \varphi_A\varphi_B f_B\, dx + \sum_{B\in N_{ps}} \int_{\Gamma_p^e} \varphi_A\varphi_B p_{rB}\, d\Gamma - \sum_{B\in N_{qs}} K_{MN}^e q_{sB} \end{aligned}$$

e $q_{sB} = q_s(x_B)$.

As matrizes K^e_{MN} e F^e_M têm ordem $\texttt{Neq}\times\texttt{Neq}$ e $\texttt{Neq}\times 1$, respectivamente. Do suporte compacto de φ_A em $\overline{\Omega}_e$, conclui-se que para todo nó global $B \notin \Omega_e$ os elementos da matriz e do vetor força são nulos. Por exemplo, considere um elemento com os seguintes nós globais:



onde $A = \{8, 9, 61, 62\}$.

Como vimos anteriormente, o número de equações no caso vetorial é dado por $\texttt{eqn}[i, A]$. Considere a seguinte tabela relacionando o número da equação com as componentes do nó global.

$i\backslash A$	8	9	61	62
1	5	7	47	49
2	6	8	48	50

Para este elemento e, os nós globais na direção horizontal $i = 1$ e na direção vertical $i = 2$ não estão prescritos, somente os coeficientes de K^e_{MN} e F^e_M, para $M, N = \{5, 6, 7, 8, 47, 48, 49, 50\}$, são não necessariamente nulos. Assim, no lugar de armazenar a matriz K^e_{MN} e F^e_M de ordem $\texttt{Neq} \times \texttt{Neq}$ e $\texttt{Neq} \times 1$ é conveniente armazenar apenas uma submatriz que denotaremos por K^e_{mn} de ordem 8×8 e F^e_m de ordem 8×1. Observe que no caso escalar bidimensional do capítulo anterior, a matriz K^e_{MN} e o vetor F^e_M são de ordem 4×4 e 4×1, respectivamente.

A matriz K^e_{mn} e o vetor força F^e_m serão chamados de matriz local e força local para os quais são introduzidos os nós locais $\underline{\mathbf{a}}$ associados aos nós globais A. Observe que os nós locais do elemento e no caso vetorial são definidos de forma análoga ao caso escalar, ou seja, se Ω_e é um retângulo, os quatro vértices do retângulo são os nós locais $a = 1, 2, 3, 4$ para o elemento e, enumerados no sentido anti-horário. A associação entre o nó local e o nó global é dada por $A = \texttt{NoLG}(e, a)$.

Para cada nó local a, define-se a função de interpolação local satisfazendo a condição:

$$\varphi^e_a(b) = \begin{cases} 1 & \text{se} \quad a = b, \\ 0 & \text{se} \quad a \neq b. \end{cases}$$

Utilizaremos, em particular, a função $\varphi^e_a(x, y)$ como um polinômio interpolante linear definida em (4.31).

Com a introdução da função de interpolação local φ^e_a podemos então definir a matriz

local e o vetor força local por

$$K^e_{mn} = \int_{\Omega_e} C_{irjs} \frac{\partial \varphi^e_a}{\partial x_i} \frac{\partial \varphi^e_b}{\partial x_j}\, dx_e \tag{5.26}$$

$$F^e_m = \sum_{b=1}^{4} \int_{\Omega_e} \varphi^e_a \varphi^e_b f_{rb}\, d\Omega_e + \sum_{b=1}^{4} \int_{\Gamma^e_p} \varphi^e_a \varphi^e_b p_{rb}\, d\Gamma - \sum_{b=1}^{4} K^e_{mn}\, q_{sb} \tag{5.27}$$

onde $1 \leq m, n \leq 8$ são os coeficientes da matriz local K^e e $1 \leq a, b \leq 4$ são os nós locais do elemento e. Os coeficientes da matriz local são dados por

$$(m, n) = (2(a-1) + r,\ 2(b-1) + s)\,, \tag{5.28}$$

onde r e s são as componentes horizontal e vertical, ou seja, $1 \leq r, s \leq 2$.

Mais explicitamente, variando os nós locais a, b e variando as componentes r e s obtém-se os 64 coeficientes da matriz K^e_{mn}. Por exemplo, se $(a, b) = (1, 2)$, variando r e s entre 1 e 2, obtém-se os coeficientes $\{K^e_{13}\ ,\ K^e_{23}\ ,\ K^e_{14}\ ,\ K^e_{24}\}$. De forma análoga, para $(a, b) = (2, 4)$ temos $\{K^e_{37}\ ,\ K^e_{38}\ ,\ K^e_{47}\ ,\ K^e_{48}\}$.

Um outro procedimento fundamental é a forma como os coeficientes K^e_{mn} da matriz local são alocados na matriz global $K = [K_{MN}]$, quando variamos o elemento $e = 1, 2, \cdots, \texttt{Nel}$, ou seja, como os coeficientes da matriz local contribuem para os coeficientes da matriz global. Para isso, considere

$$P = \texttt{eqn}[i, A], \quad 1 \leq i \leq 2,\ A \in N - N_{qi},$$

onde P é o número da equação, i.e.,

$$P = \texttt{eqn}[i, \texttt{NoLG}(a, e)].$$

Como os coeficientes da matriz local K^e_{mn} estão associados aos nós locais pela relação (5.28), podemos associar para cada elemento e, os coeficientes da matriz global K_{MN} com os coeficientes K^e_{mn} e analogamente para F_M.

Por exemplo, considere o elemento e que tem como nós globais $A = \{8, 9, 61, 62\}$ e o número de equações é dado pela tabela anterior. Para o elemento e fixo tem-se

$$\begin{aligned} &\texttt{NoLG}(1, e) = 8,\\ &\texttt{NoLG}(2, e) = 9,\\ &\texttt{NoLG}(3, e) = 61,\\ &\texttt{NoLG}(4, e) = 62. \end{aligned}$$

Localmente temos a matriz K^e_{mn} de ordem 8×8.

Então a contribuição da matriz local na matriz global é dada por

$$\begin{array}{rcl}
K_{55} & \leftarrow & K_{55} + K^e_{11}, \\
K_{56} & \leftarrow & K_{56} + K^e_{12}, \\
K_{57} & \leftarrow & K_{57} + K^e_{13}, \\
K_{58} & \leftarrow & K_{58} + K^e_{14}, \\
K_{5,47} & \leftarrow & K_{5,47} + K^e_{15}, \\
\vdots & \vdots & \vdots \\
K_{5,50} & \leftarrow & K_{5,50} + K^e_{18}, \\
K_{6,6} & \leftarrow & K_{6,6} + K^e_{22}, \\
K_{6,7} & \leftarrow & K_{6,7} + K^e_{2,3}, \\
\vdots & \vdots & \vdots \\
K_{6,50} & \leftarrow & K_{6,50} + K^e_{28}, \\
\vdots & \vdots & \vdots \\
K_{49,49} & \leftarrow & K_{49,49} + K^e_{77}, \\
K_{49,50} & \leftarrow & K_{49,50} + K^e_{78}, \\
K_{50,50} & \leftarrow & K_{50,50} + K^e_{88}.
\end{array}$$

e mais os elementos simétricos da matriz.

Variando $e = 1, 2, \cdots, \texttt{Nel}$, obtém-se todos os coeficientes K_{MN} da matriz global K, onde

$$M = \texttt{eqn}[i, \texttt{NoLG}(a, e)].$$

De forma análoga para a Força Global F,

$$\begin{array}{rcl}
F_5 & \leftarrow & F_5 + F^e_1, \\
F_6 & \leftarrow & F_6 + F^e_2, \\
\vdots & \vdots & \vdots \\
F_{50} & \leftarrow & F_{50} + F^e_8.
\end{array}$$

• Cálculo de Matriz Local

Para calcular a matriz local utilizaremos o mesmo procedimento do capítulo anterior, isto é, fazemos a transformação isoparamétrica entre o elemento finito Ω_e para o quadrado biunitário Ω_b e então utilizamos as funções base φ^e_a definidas em (4.32) por

$$\varphi_a(\xi, \eta) = \frac{1}{4}(1 + \xi_a\xi)(1 + \eta_a\eta), \quad a = 1, 2, 3, 4. \tag{5.29}$$

Entaão, o gradiente é dado por

$$\nabla\varphi_a(\xi, \eta) = \frac{1}{4}\left(\xi_a(1 + \eta_a\eta),\ \eta_a(1 + \xi_a\xi)\right),$$

onde $(\xi, \eta) \in \Omega_b$. Para $a = 1, 2, 3, 4$ tem-se, respectivamente,

$$(\xi_a, \eta_a) = (-1, -1),\ (1, -1),\ (1, 1),\ (-1, 1).$$

Utilizando a forma componente, temos

$$\frac{\partial \varphi_a}{\partial x_k} = \frac{\partial \varphi_a}{\partial \xi_i} \frac{\partial \xi_i}{\partial x_k}.$$

Substituindo na matriz local (5.26), obtemos

$$K^e_{mn} = \int_{\Omega_e} C_{irjs}\ \frac{\partial \varphi^e_a}{\partial \xi_k}\ \frac{\partial \xi_k}{\partial x_i}\ \frac{\partial \varphi^e_b}{\partial \xi_\ell}\ \frac{\partial \xi_\ell}{\partial x_j}\, dx_e = C^e_{irjs} \int_{\Omega_b} \left[\frac{\partial \varphi^e_a}{\partial \xi_k}\ \frac{\partial \xi_k}{\partial x_i}\ \frac{\partial \varphi^e_b}{\partial \xi_\ell}\ \frac{\partial \xi_\ell}{\partial x_j} \right] |J|\, dx_b,$$

onde estamos assumindo que o coeficiente de elasticidade $C_{ikj\ell}$ é constante sobre cada elemento e, com J denotando o Jacobiano da transformação entre os domínios Ω_e e Ω_b, definido por

$$J = \det \begin{bmatrix} \dfrac{\partial x_1}{\partial \xi_1} & \dfrac{\partial x_1}{\partial \xi_2} \\ \dfrac{\partial x_2}{\partial \xi_1} & \dfrac{\partial x_2}{\partial \xi_2} \end{bmatrix}.$$

Assim, considerando os coeficientes

$$\widehat{Q}^e_{abij} = \int_{\Omega_b} \frac{\partial \varphi^e_a}{\partial \xi_k}\ \frac{\partial \varphi^e_b}{\partial \xi_\ell}\ \frac{\partial \xi_k}{\partial x_i}\ \frac{\partial \xi_\ell}{\partial x_j}\, dx_b \tag{5.30}$$

e o Jacobiano constante para cada elemento Ω_e, podemos escrever os coeficientes da matriz local na forma

$$K^e_{mn} = |J^e|\ C^e_{irjs} \widehat{Q}^e_{abij}.$$

Em particular, se Ω_e é um retângulo, obtemos

$$\frac{\partial \xi_i}{\partial x_k} = \begin{cases} \dfrac{2}{\Delta x_i} & \text{se } i = k, \\ 0 & \text{se } i \neq k. \end{cases}$$

Substituindo em (5.31), obtemos

$$\widehat{Q}^e_{abij} = \frac{4}{\Delta x_i \Delta x_j}\ Q^e_{abij},$$

onde $\Delta x_i = x_i - x_{i-1}$ e

$$Q^e_{abij} = \int_{\Omega_b} \frac{\partial \varphi^e_a}{\partial \xi_k}\ \frac{\partial \varphi^e_b}{\partial \xi_\ell}\, dx_b. \tag{5.32}$$

Além disso, o Jacobiano da transformação é dado por

$$J^e = \frac{1}{4}\Delta x^e \Delta y^e, \tag{5.33}$$

onde $\Delta x^e = (x_2^e - x_1^e)$ e $\Delta y^e = (y_2^e - y_1^e)$. Assim, a matriz local pode ser calculada por

$$K_{mn}^e = |J^e| C_{irjs}^e \, \frac{4}{|Deltax_i \Delta x_j} \, Q_{abij}^e. \tag{5.34}$$

Observe que os únicos termos dependentes do elemento e são o Jacobiano e o coeficiente de elasticidade $C_{ikj\ell}$, onde em (5.34) estamos omitindo o somatório dos índices $1 \leq i, j, k, l;\ r, s \leq 2$ e $1 \leq a, b \leq 4$.

A matriz local dada acima é similar a matriz local dada em (4.44) e está definida na subrotina `LocalSystem`, onde a integral (5.32) foi calculada pela quadratura Gaussiana.

• **Cálculo da Força Local** F_m^e

Podemos verificar de (5.27) que a força local é dada por

$$F_m^e = f_m^e + p_m^e - q_m^e, \quad \text{para} \quad m = 2(a-1) + r, \tag{5.35}$$

onde

$$\begin{aligned} f_m^e &= \sum_{b=1}^{4} \int_{\Omega^e} (\varphi_a^e \varphi_b^e) f_{br}^e \, d\Omega_e, \\ p_m^e &= \sum_{b=1}^{4} \int_{\Gamma_p^e} (\varphi_a^e \varphi_b^e) p_{br}^e \, d\Gamma, \\ q_m^e &= \sum_{b=1}^{4} K_{mn}^e \, q_{br}^e. \end{aligned}$$

As igualdades acima são válidas para todo nó local $b \in \Omega_e$, cujo correspondente nó global $B = \texttt{NoLG}(e, b)$ satisfaça as respectivas condições $B \in N - N_{qi}$, $B \in N_{pi}$ e $B \in N_{qi}$.

• **Cálculo de** f_m^e

O cálculo é análogo ao caso escalar feito em (4.59). Assim, usando o mesmo procedimento, obtém-se

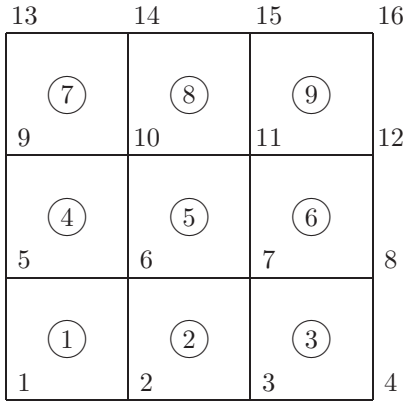
$$f_m^e = \sum_{b=1}^{4} \int_{\Omega^e} \varphi_a^e \varphi_b^e f_{br}^e \, dx = J^e \sum_{b=1}^{4} f_{br}^e \int_{\Omega_b} \varphi_a^e \varphi_b^e \, dx = J^e \sum_{b=1}^{e} f_{br}^e \, Q_{ab}, \tag{5.36}$$

onde $f^e_{br} = f(x^e_{br}) = f_r(x^e_b) = f_r(B)$, $B = \texttt{NoLG}(e, b)$, $r = 1, 2$, com $m = 2(b-1) + r$ e

$$Q_{ab} = \int_{\Omega b} \varphi^e_a \varphi^e_b \, dx.$$

Observe que a matrix Q_{ab}, calculada em (4.51), é independente do elemento e.

A título de exemplo, consideremos a seguinte malha:



$$\Gamma_{q1} = \{1, 2, 5, 8, 13, 16\}, \quad \Gamma_{q2} = \{2, 3, 4, 8, 13, 16\},$$
$$\Gamma_{p1} = \{3, 4, 9, 12, 14, 15\}, \quad \Gamma_{p2} = \{1, 5, 9, 12, 14, 15\},$$

onde Γ_{qi} é o conjunto dos nós globais nos quais u_i está prescrita e Γ_{pi} é o conjunto dos nós globais nos quais a derivada normal de u_i está prescrita. Note que deve ser satisfeita a seguinte relação: $\Gamma = \Gamma_{p1} \cup \Gamma_{q1}$ e $\Gamma = \Gamma_{p2} \cup \Gamma_{q2}$.

• **Número de Equações:** $\texttt{eqn}[i, A]$

$i \backslash A$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	2	3	0	5	7	0	9	11	13	15	0	17	19	0
2	1	0	0	0	4	6	8	0	10	12	14	16	0	18	20	0

Assim, temos $\texttt{Neq} = 20$ e a matriz global $K = [K_{MN}]$ é de ordem 20×20.

Por outro lado, para cada nó global A, existe um nó local a dado pela relação $A = \texttt{NoLG}(e, a)$, de modo que o número de equações locais m é dado pela relação $m = 2(a-1) + r$, $r = 1, 2,\ a = 1, 2, 3, 4$.

Definimos em (5.35) que a força local é calculada por

$$F^e_m = f^e_m + p^e_m - q^e_m.$$

Vamos então determinar F^e_m para cada $e = 1, 2, \ldots, 9$, localmente.

(1) Se $e = 5$, os nós locais $\{1, 2, 3, 4\}$ estão associados aos nós globais $\{6, 7, 11, 10\}$ através da relação $A = \texttt{NoLG}(e, a) = \texttt{NoLG}(5, a)$. Para este elemento, não existe nó global $A \in \Gamma_{pi}$ ou $A \in \Gamma_{qi}$. Assim,

$$F_m^5 = f_m^5 \quad \text{se} \quad m = 1, 2, \ldots, 8,$$

pois $p_m^5 = q_m^5 = 0$.

(2) Se $e = 8$, temos a associação $\{1, 2, 3, 4\}$ com $\{10, 11, 15, 14\}$, onde $A = 14, 15 \in \Gamma_{p1} \cap \Gamma_{p2}$. Então $(r, 4)$ e $(r, 3) \in \Gamma_{p1} \cap \Gamma_{p2}$. Assim, para as equações $m = \{5, 6, 7, 8\}$, temos

$$\begin{array}{rcll} F_m^8 & = & f_m^8 & \text{se } m = 1, 2, 3, 4, \\ F_m^8 & = & f_m^8 + p_m^8 & \text{se } m = 5, 6, 7, 8, \end{array}$$

pois $(r, 1)$ e $(r, 2) \notin \Gamma_{pr} \cup \Gamma_{qr}$.

(3) Se $e = 2$, então $\{1, 2, 3, 4\}$ está associado com $\{2, 3, 7, 6\}$ onde $2 \in \Gamma_{q1} \cap \Gamma_{q2}$ e $3 \in \Gamma_{q2}$. Para $r = 1, 2$ e $a = 1$ temos

$$(r, \texttt{NoLG}(e, a_1)) \in \Gamma_{q1} \cap \Gamma_{q2} \quad \text{e} \quad (2, \texttt{NoLG}(2, 2)) \in \Gamma_{q2}.$$

Como $q_{br}^e = q_r(x_b^e)$, onde x_b^e são os nós prescritos, tem-se

$$\begin{array}{rcll} F_1^2 & = & f_1^2 - q_1^2, & \\ F_2^2 & = & f_2^2 - q_2^2, & \\ F_3^2 & = & f_3^2, & \\ F_4^2 & = & f_4^2 - q_4^2, & \\ F_m^2 & = & f_m^2, & \text{se } m = 5, 6, 7, 8. \end{array}$$

Observe que sendo $3 \notin \Gamma_{q1}$, então $q_1(x_2^3) = 0$. Analogamente, os nós globais $6, 7 \notin \Gamma_{q1} \cup \Gamma_{q2} \Rightarrow q_r(x_b^3) = 0$. Como $A = \texttt{NoLG}(2, a) \notin \Gamma_{p1} \bigcup \Gamma_{p2} \Rightarrow p_r(x_b^3) = 0$.

(4) Seja $e = 6$. Neste caso, os componentes locais prescritos são

$$\begin{array}{rcl} 8 & = & \texttt{NoLG}(6, 2) \in \Gamma_{q1}, \\ 8 & = & \texttt{NoLG}(6, 2) \in \Gamma_{q2}, \\ 12 & = & \texttt{NoLG}(6, 3) \in \Gamma_{p1}, \\ 12 & = & \texttt{NoLG}(6, 3) \in \Gamma_{p2}. \end{array}$$

Assim a contribuição desse elemento é dado por

$$\begin{array}{rcll} F_m^6 & = & f_m^6 & \text{se } m = 1, 2, 7, 8, \\ F_m^6 & = & f_m^6 - q_m^6 & \text{se } m = 3, 4, \\ F_m^6 & = & f_m^6 + p_m^6 & \text{se } m = 5, 6. \end{array}$$

(5) De forma análoga temos para $e = 1$

$$\begin{aligned} F_m^1 &= f_m^1 - q_m^1 \quad \text{se } m = 1,3,4,7, \\ F_m^1 &= f_m^1 \quad \text{se } m = 2,5,6, \\ F_m^1 &= f_m^1 - p_m^1 \quad \text{se } m = 8. \end{aligned}$$

Se $e = 3$

$$\begin{aligned} F_m^3 &= f_m^3 \quad \text{se } m = 1,7,8, \\ F_m^3 &= f_m^3 - q_m^3 \quad \text{se } m = 2,4,5,6, \\ F_m^3 &= f_m^3 + p_m^3 \quad \text{se } m = 3. \end{aligned}$$

Se $e = 4$

$$\begin{aligned} F_m^4 &= f_m^4 \quad \text{se } m = 3,4,5,6, \\ F_m^4 &= f_m^4 - q_m^4 \quad \text{se } m = 1, \\ F_m^4 &= f_m^4 + p_m^4 \quad \text{se } m = 2,7,8. \end{aligned}$$

Se $e = 7$

$$\begin{aligned} F_m^7 &= f_m^7 \quad \text{se } m = 3,4, \\ F_m^7 &= f_m^7 - q_m^7 \quad \text{se } m = 7,8, \\ F_m^7 &= f_m^7 + p_m^7 \quad \text{se } m = 1,2,5,6. \end{aligned}$$

Se $e = 9$

$$\begin{aligned} F_m^9 &= f_m^9 \quad \text{se } m = 1,2, \\ F_m^9 &= f_m^9 - q_m^9 \quad \text{se } m = 5,6, \\ F_m^9 &= f_m^9 + p_m^9 \quad \text{se } m = 3,4,7,8. \end{aligned}$$

• **Cálculo de q_m^e**

Por definição, temos

$$q_m^e = \sum_{b=1}^{4} K_{mn}^e \, q_{br}^e.$$

Consideremos então $B = \texttt{NoLG}(e,b)$. Se $B \in \Gamma_{qr}$, então $u_r(B) = q_r(B) = q_r(x_b^e) = q_{br}^e$. Se $B \notin \Gamma_{qr}$, então $u_r(B)$ não está prescrita, ou seja, $q_{br}^e = 0$.

Este procedimento é encontrado na subrotina `CondFront`, onde os nós $B \in \Gamma_{qr}$ são denominados do tipo 1.

• **Cálculo de p_m^e**

$$p_m^e = \sum_{b=1}^{4} \int_{\Gamma_{pr}^e} (\varphi_a^e \varphi_b^e) p_{br}^e \, d\Gamma = \sum_{b=1}^{4} M_{ab} p_{br}^e \tag{5.37}$$

onde $p_{br}^e = p_r(x_b^e) = p_r^e = p_r(B)$ e a matriz M_{ab} é a mesma definida em (4.61).

Queremos identificar todos os nós globais $A \in \Gamma_p$ para os quais o tensor de tensão σ_{rj} na direção da normal exterior n_j é conhecido. O procedimento é análogo para o

caso escalar do capítulo anterior, com cada componente $r = 1$ ou $r = 2$ fixada. Assim, para $r = 1, 2$ tem-se

$$\begin{aligned} \sigma_{1j} n_j &= p_1 \text{ em } \Gamma_{p1} \\ \sigma_{2j} n_j &= p_2 \text{ em } \Gamma_{p2} \end{aligned}$$

Os nós $A \in \Gamma_{pr}$ são identificados pela subrotina `CondFront`, ou seja, o tensor está definido nas direções x e y respectivamente. Assim, para cada componente, utilizam-se as matrizes definidas em (4.63) e (4.64), i.e.,

$$M_{ab} = \begin{cases} \frac{1}{3}\Delta x^e & \text{se } a = b, \\ \frac{1}{6}\Delta x^e & \text{se } a \neq b, \end{cases} \tag{5.38}$$

ou

$$M_{ab} = \begin{cases} \frac{1}{3}\Delta y^e & \text{se } a = b, \\ \frac{1}{6}\Delta y^e & \text{se } a \neq b. \end{cases} \tag{5.39}$$

Para exemplificar, considere a malha na página 176. Assim,

1. Para o elemento $e = 3$, temos os nós globais $\{3, 4, 7, 8\}$. O único nó global em Γ_p é o nó $A = 4 = \texttt{NoLG}(e, a) = \texttt{NoLG}(3, 2)$. Para este nó, está definido o valor de $p(x_a^e)$. Mais especificamente, como $A = 4 \in \Gamma_{p1}$, então $p_1(x_2^3) = p_{12}^3$. Para todos os outros nós do elemento $e = 3$, tem-se $p_r(x_a^3) = 0$. Dado que $m = 2(a-1) + r$, a única equação a assumir um valor não necessariamente nulo é $m = 3 (a = 2, r = 1)$. Como

   $$p_m^e = \sum_{b=1}^{4} M_{ab} p_{br}^e,$$

   segue que

   $$p_3^3 = M_{22} p_{21}^3.$$

   Dependendo do tipo do nó global $A = 4$, temos

   $$M_{22} = \frac{\Delta y^e}{3} \text{ ou } M_{22} = \frac{\Delta x^e}{3}.$$

2. De forma análoga, para o elemento $e = 1$, temos como único nó prescrito o nó $A = 5 \in \Gamma_{p2}$, ou seja, $r = 2$. Sendo $5 = \texttt{NoLG}(1, 4)$, apenas o valor de p_{42}^1 está definido, para os demais valores, $p_{br}^1 = 0$. Logo, para $r = 2$ e $a = 4$, temos $m = 8$ e

   $$p_8^1 = M_{44} p_{42}^1.$$

3. Considere o elemento $e = 7$. Os nós $A = 9$ e $A = 12$ estão prescritos em Γ_{p1} e Γ_{p2}, ou seja, $(r, 9)$ e $(r, 12)$ estão prescritos para $r = 1$ e $r = 2$. Como $9 = \texttt{NoLG}(7, 1)$ e $12 = \texttt{NoLG}(7, 3)$, então $p^7_{br} = p_r(x^7_b)$ não é necessariamente nulo para $r = 1, 2$ e $b = 1, 3$. Assim, da relação $m = 2(b-1) + r$, somente p^7_m para $m = 1, 2, 5, 6$ não é necessariamente nulo. Assim, temos:

$$\begin{aligned}
&\text{para } r = 1 \text{ e } b = 1, \quad p^7_1 = \sum_{b=1}^{4} M_{ab} p_1(x^7_b) = M_{11} p_1(x^7_1) + M_{31} p_1(x^7_1),\\
&\text{para } r = 1 \text{ e } b = 1, \quad p^7_2 = M_{11} p_2(x^7_1) + M_{31} p_2(x^7_1),\\
&\text{para } r = 1 \text{ e } b = 3, \quad p^7_5 = M_{13} p_1(x^7_3) + M_{33} p_1(x^7_3),\\
&\text{para } r = 2 \text{ e } b = 3, \quad p^7_6 = M_{13} p_2(x^7_3) + M_{33} p_2(x^7_3),
\end{aligned}$$

onde M_{ab} está definida em (5.39).

4. Os elementos $e = 4, 6, 8, 9$ também têm nós globais prescritos em Γ_{pr} e os cálculos são semelhantes.

• **Valores de fronteira:** p^e_m **e** q^e_m

Os valores, p^r_b e q^r_b são contribuições de fronteira do tipo Neumann e Dirichlet em cada componente $r = 1, 2$. Para identificar os nós globais da fronteira e o tipo de fronteira são usadas as subrotinas `Fronteira, CondFront`. A subrotina `Fronteira` tem a função de identificar todos os nós das fronteiras do domínio, ou seja $\{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4\}$ e a expressão `Nbn[i]` quantifica o número de nós de cada fronteira Γ_i.

Uma vez identificados todos os nós da fronteira, a subrotina `CondFront` classifica as componentes horizontais e verticais de cada nó, como sendo do tipo Neumann ou Dirichlet, a partir da interferência do operador do programa nessa subrotina. Assim, por exemplo, se na fronteira Γ_j as componentes vertical e horizontal são do tipo de Neumann, a derivada da solução na variável x ou y é prescrita, ou seja,

$$\frac{\partial u}{\partial x}(r, A) = u_x(r, A) \quad \text{ou} \quad \frac{\partial u}{\partial y}(r, A) = u_y(r, A).$$

onde $r = 0$ indica a direção horizontal e $r = 1$ a vertical e $A \in N_{pi}$ um nó global da fronteira. Caso seja conhecido o valor $u_x(r, A)$, basta colocar na subrotina `CondFront` o que segue:

$$\texttt{Bv}[j][i][r] = u_x(r, A), \qquad r = 0, 1.$$

onde j corresponde à fronteira Γ_j e i corresponde aos nós indexados da fronteira.

Por outro lado, se $A \in N_{qi}$ é um nó global da fronteira do tipo Dirichlet, ou seja $u(r, A)$ é conhecida para $r = 0, 1$, então o procedimento para inserir essa condição na subrotina `CondFront` é dado por:

$$\texttt{typ}[A][r] = 1, \quad u[A][r] = u_r(A).$$

Quando uma componente tem um valor prescrito e outra tem a outra derivada prescrita, basta concatenar as inserções acima. Por exemplo, se temos conhecido $u_x(0, A)$ na direção horizontal e $u(1, A)$ na direção vertical, o procedimento pode ser feito por

$$\texttt{Bv}[j][i][0] = u_x(0, A),$$

$$\texttt{typ}[A][1] = 1, \quad u[A][1] = u_1(A).$$

Sabemos que quando a componente do nó é do tipo 1, então a solução é prescrita e não gera equação no sistema linear.

• **Contribuição da Fronteira de Neumann:** `TractionBoundary`

Como anteriormente, a subroutina `TractionBoundary` tem a função de inserir apropriadamente a contribuição dos valores de fronteira de Neumann na força, usando as matrizes definidas em (4.63) e (4.64), ou seja calcular o valor de p_a^e em $\Gamma_1,\ \Gamma_2,\ \Gamma_3,\ \Gamma_4$, quando essas fronteiras são do tipo Neumann.

5.2.2 Construção da Matriz Global

Por definição a matriz global é obtida por

$$K = \sum_{e=1}^{N_{e\ell}} K^e.$$

A ordem da matriz K é $\texttt{Neq} \times \texttt{Neq}$. Cada matriz local K^e tem ordem 8×8 e a forma de alocação de cada uma das matrizes K^e, para $e = 1, 2, \cdots, \texttt{Nel}$, é um processo fundamental do método de elementos finitos. A forma de contribuição é dada nas seguintes etapas:

(1) Identificação entre os nós locais e os nós globais dada por

$$A = \texttt{NoLG}(e, a).$$

(2) Identificação entre o nó global e a sua correspondente equação do sistema dado por

$$P = \texttt{eqn}[i, A] \ \ i = 1, 2.$$

Tomando como exemplo a malha da página 176, vemos que os nós globais na primeira etapa são dados na Tabela 5.1 abaixo e, para a segunda etapa, temos a Tabela 5.2, que mostra a relação entre os nós globais A e o número de equações.

$a\backslash e$	1	2	3	4	5	6	7	8	9
1	1	2	3	5	6	7	9	10	11
2	2	3	4	6	7	8	10	11	12
3	6	7	8	10	11	12	14	15	16
4	5	6	7	9	10	11	13	14	15

Tabela 5.1: $A = \texttt{NoLG}(e, a)$

$i\backslash A$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	2	3	0	5	7	0	9	11	13	15	0	17	19	0
2	1	0	0	0	4	6	8	0	10	12	14	16	0	18	20	0

Tabela 5.2: $P = \texttt{eqn}[i, A]$

Assim, para os 15 nós globais, temos 20 equações e portanto a matriz global tem ordem 20×20. Consideremos agora a contribuição de cada nó global A para os coeficientes da matriz global K em ordem crescente, onde aplicamos a relação

$$j = \texttt{eqn}[i, A] = \texttt{eqn}[i, \texttt{NoLG}(e, a)]. \tag{5.40}$$

Se $j = 0$, (i, A) não gera equação e portanto não contribui para os coeficientes da matriz K. Se $j \neq 0$, (i, A) gera a j-ésima equação e portanto, contribui para j-ésima linha da matriz global K, onde $j \leq M$. Relembremos que os coeficientes da matriz local K^e_{mn} satisfazem a relação

$$\begin{array}{ll} m = 2(a-1) + i, & a, b = 1, 2, 3, 4, \\ n = 2(b-1) + i, & i = 1, 2. \end{array} \tag{5.41}$$

Temos então,

1. $e = 1$: Os nós globais desse elemento são $A = 1, 2, 6, 5$. Observando a Tabela 5.2 e a relação (5.41) vemos que os coeficientes K^e_{mn} da matriz local não são necessariamente nulos. Mais precisamente, os elementos

   $$\{K^1_{22}, K^1_{25}, K^1_{26}, K^1_{28}, K^1_{55}, K^1_{56}, K^1_{58}, K^1_{66}, K^1_{68}, K^1_{88}\}$$

   e também os respectivos simétricos.

   A alocação destes coeficientes nos coeficientes da matriz global é dada por

K_{11+}	K_{14+}	K_{15+}	K_{16+}	K_{44+}	K_{45+}	K_{46+}	K_{55+}	K_{56+}	K_{66+}
K^1_{22}	K^1_{28}	K^1_{25}	K^1_{26}	K^1_{88}	K^1_{85}	K^1_{86}	K^1_{55}	K^1_{56}	K^1_{66}

onde em cada coluna o segundo membro representa a contribuição local para o coeficiente global do primeiro membro. Por exemplo, na primeira coluna, temos

$$K_{11} \leftarrow K_{11} + K^1_{22}.$$

Todos os outros coeficientes da matriz global não recebem contribuição do elemento $e = 1$.

2. $e = 2$: Neste caso os nós globais são: $A = \{2, 3, 7, 6\}$. Observando a Tabela 5.2 e a relação (5.41) concluímos que os coeficientes da matriz local que contribuem para a matriz global são

$$\begin{cases} K^2_{33}, K^2_{35}, K^2_{36}, K^2_{37}, K^2_{38}, K^2_{55}, K^2_{56}, K^2_{57}, K^2_{58}, \\ \\ K^2_{66}, K^2_{67}, K^2_{68}, K^2_{77}, K^2_{78}, K^2_{88}, \ + \text{simétricos.} \end{cases}$$

Dessa forma temos

K_{22+}	K_{25+}	K_{26+}	K_{27+}	K_{28+}	K_{55+}	K_{56+}	K_{57+}
K^2_{33}	K^2_{37}	K^2_{38}	K^2_{35}	K^2_{36}	K^2_{77}	K^2_{78}	K^2_{75}
K_{58+}	K_{66+}	K_{67+}	K_{68+}	K_{77+}	K_{78+}	K_{88+}	
K^2_{76}	K^2_{88}	K^2_{85}	K^2_{86}	K^2_{55}	K^2_{56}	K^2_{66}	

Fazendo $e = 3, 4, \cdots, 9$ de forma análoga obtém-se a matriz global K_{MN} do sistema linear, onde M e N são os números de equações correspondentes aos nós globais (i, A). A matriz global K para este exemplo, de ordem (20×20), está representada na Fig. 5.1 abaixo, onde omitimos os coeficientes simétricos e denotamos por $*$ os coeficiente não necessariamente nulos.

5.2.3 • Construção da Força Global

A força global é definida por

$$F = \sum_{e=1}^{N_{e\ell}} F^e_m \,,$$

onde $F = [F_M]$ com $M = 1, 2, \cdots,$ `Neq`.

Usando novamente a malha anterior, analisaremos a contribuição da força local F^e_m para a força global F_M. A força local é dada por

$$F^e_m = f^e_m + p^e_m - q^e_m \,, \qquad m = 2(a-1) + r \,, \qquad r = 1, 2 \qquad a = 1, 2, 3, 4$$

Para $e = 1, 2, 3, \cdots, 9$ os valores de F^e_m foram calculados anteriormente. Então, a forma de alocação das forças locais para a força global, onde apenas estão listadas as componentes que recebem contribuição do elemento especificado, é dada por

$$\begin{bmatrix}
* & 0 & 0 & * & * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & * & * & 0 & * & * & * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & * & 0 & 0 & 0 & * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & * & * & * & 0 & 0 & * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & * & * & * & * & * & * & * & * & * & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & * & * & * & * & * & * & * & * & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & * & * & 0 & 0 & * & * & * & * & * & * & 0 & 0 & 0 & 0 \\
 & & & & & & & * & 0 & 0 & * & * & * & * & * & * & 0 & 0 & 0 & 0 \\
 & & & & & & & & * & * & * & * & 0 & 0 & 0 & 0 & * & * & 0 & 0 \\
 & & & & & & & & & * & * & * & 0 & 0 & 0 & 0 & * & * & 0 & 0 \\
 & & & & & & & & & & * & * & * & * & 0 & 0 & * & * & * & * \\
 & & & & & & & & & & & * & * & * & 0 & 0 & * & * & * & * \\
 & & & & & & & & & & & & * & * & * & * & * & * & * & * \\
 & & & & & & & & & & & & & * & * & * & * & * & * & * \\
 & & & & & & & & & & & & & & * & * & 0 & 0 & * & * \\
 & & & & & & & & & & & & & & & * & 0 & 0 & * & * \\
 & & & & & & & & & & & & & & & & * & * & * & * \\
 & & & & & & & & & & & & & & & & & * & * & * \\
 & & & & & & & & & & & & & & & & & & * & * \\
 & & & & & & & & & & & & & & & & & & & *
\end{bmatrix}$$

Figura 5.1: Matriz K

1. $e = 1$: Usando as Tabelas 5.1, 5.2 e a relação (5.40) tem-se

$$\begin{aligned}
F_1 &\leftarrow F_1 + F_2^1, \\
F_4 &\leftarrow F_4 + F_8^1, \\
F_5 &\leftarrow F_5 + F_5^1, \\
F_6 &\leftarrow F_6 + F_6^1.
\end{aligned}$$

2. $e = 2$:

$$\begin{aligned}
F_2 &\leftarrow F_2 + F_3^2, \\
F_5 &\leftarrow F_5 + F_7^2, \\
F_6 &\leftarrow F_6 + F_8^2, \\
F_7 &\leftarrow F_7 + F_5^2, \\
F_8 &\leftarrow F_8 + F_6^2.
\end{aligned}$$

3. $e = 5$: Temos os nós globais $A = 6, 7, 11, 10$ associados aos nós locais $a = 1, 2, 3, 4$.

Usando a Tabela 5.2 e a relação (5.40) temos respectivamente a força dada por

$$\begin{array}{rcrcl} F_5 & \leftarrow & F_5 & + & F_1^5, \\ F_6 & \leftarrow & F_6 & + & F_2^5, \\ F_7 & \leftarrow & F_7 & + & F_3^5, \\ F_8 & \leftarrow & F_8 & + & F_4^5, \\ F_{13} & \leftarrow & F_{13} & + & F_5^5, \\ F_{14} & \leftarrow & F_{14} & + & F_6^5, \\ F_{11} & \leftarrow & F_{11} & + & F_7^5, \\ F_{12} & \leftarrow & F_{12} & + & F_8^5. \end{array}$$

De forma análoga pode-se calcular toda a contribuição para a força global dos elementos e da malha.

Os procedimentos da matriz global e a da Força Global estão na subrotina do programa `GlobalSystem`.

5.3 Sistema Linear

Na resolução do sistema global $K\boldsymbol{d} = \boldsymbol{F}$, utilizaremos o algoritmo de Crout [9]. Basicamente, a única diferença é o tamanho da matriz, visto que, neste caso, a função vetorial $\boldsymbol{u} = (u_1, u_2)$ tem as componentes horizontais e verticais. A partir da solução do sistema linear, obtemos $\boldsymbol{d} = (d_1, \ldots, d_N)$, onde $N = \texttt{eqn}(j, B)$, $j = 1, 2$, $B = 1, 2, \ldots \texttt{Neq}$, obtemos a solução $u_j(B)$, pois

$$u_j(x_B) = \sum_{B \in N - N_{qj}} d_{jB}\varphi_B(x_B) = d_{jB}.$$

O procedimento da resolução do sistema linear é encontrado na subrotina `Solver`.

Sejam $\boldsymbol{u}^h(x)$ a solução numérica e $\boldsymbol{u}(x)$ a solução exata. Determinamos os erros nas norma $L^2(\Omega)$ e na seminorma $H^1(\Omega)$, defiidos por

$$\|E\|_0 = \|\boldsymbol{u} - \boldsymbol{u}^h\|_0, \qquad \|E\|_1 = \|\nabla\boldsymbol{u} - \nabla\boldsymbol{u}^h\|_0 \tag{5.42}$$

e os percentuais de erro

$$E_0\% = \frac{\|E\|_0}{\|\boldsymbol{u}\|_0} \times 100, \qquad E_1\% = \frac{\|E\|_1}{\|\nabla\boldsymbol{u}\|_0} \times 100. \tag{5.43}$$

5.4 Exemplos Numéricos

Os elementos K^e_{mn} da matriz local são definidos por

$$K^e_{mn} = \delta_{rk}\delta_{s\ell}\frac{4}{\Delta\xi_i\Delta\xi_j}J^e C^e_{ikj\ell}Q_{abij}, \tag{5.44}$$

onde

$$\begin{aligned} Q_{abij} &= \int_{\Omega_b}\frac{\partial\varphi_a}{\partial\xi_i}\frac{\partial\varphi_b}{\partial\xi_j}\,dx, \\ \Delta\xi_1 &= \xi_2 - \xi_1 = 2, \\ \Delta\xi_2 &= \eta_2 - \eta_1 = 2. \end{aligned}$$

O único termo a ser definido em (5.44) é o coeficiente de elasticidade $\{C_{ikj\ell}\}$, que depende da posição. Como $1 \le i,k,j \le \ell \le 2$, devemos conhecer os 16 coeficientes.

Como exemplo numérico, vamos considerar o corpo isotrópico, onde o coeficiente de elasticidade pode ser escrito por

$$C_{ijk\ell}(x) = \mu(x)(\delta_{ik}\delta_{j\ell} + \delta_{i\ell}\delta_{jk}) + \lambda(x)\delta_{ij}\delta_{k\ell}, \tag{5.45}$$

sendo μ e λ os parâmetros de Lamé. Os coeficientes não necessariamente nulos são

$$\begin{aligned} &C_{1111} = C_{2222} = \lambda + 2\mu, \\ &C_{1122} = C_{2211} = \lambda, \\ &C_{1212} = C_{1221} = C_{2112} = C_{2121} = \mu. \end{aligned} \tag{5.46}$$

Supondo o corpo homogêneo, $C_{ijk\ell}$ não depende de x e podemos definir os coeficientes na forma

$$\left\{ \begin{array}{c} C_{1111}, C_{1112}, C_{1121}, C_{1122}, C_{1211}, C_{1212}, C_{1221}, C_{1222}, \\ C_{2111}, C_{2112}, C_{2121}, C_{2122}, C_{2211}, C_{2212}, C_{2221}, C_{2222} \end{array} \right\} = \left\{ \begin{array}{c} \lambda+2\mu, 0, 0, \lambda, 0, \mu, \mu, 0, \\ 0, \mu, \mu, 0, \lambda, 0, 0, \lambda+2\mu \end{array} \right\}$$

Nessas condições, o problema geral de elasticidade linear

$$-\frac{\partial}{\partial x_j}\Big(C_{ijk\ell}\frac{\partial u_k}{\partial x_\ell}\Big) = f_i$$

toma a forma

$$-(\lambda+\mu)\left(\frac{\partial^2 u_1}{\partial x\partial y} + \frac{\partial^2 u_2}{\partial y^2}\right) - \mu\left(\frac{\partial^2 u_2}{\partial x^2} + \frac{\partial^2 u_2}{\partial y^2}\right\} = f_2, \tag{5.47}$$

ou, mais sucintamente,

$$-(\lambda+\mu)\,\nabla(\operatorname{div} u) - \mu\Delta u = f,$$

onde $u = (u_1, u_2)$ e $f = (f_1, f_2)$.

5.4.1 Exemplo 1: Condições de Fronteira do tipo Dirichlet

Consideremos $\mu = 1$ e $\lambda = 1.5$ e as componentes horizontal e vertical da força dadas por

$$(f_1, f_2) = 2(\cos(\pi x)\cos(\pi y),\, \mathrm{sen}(\pi x)\,\mathrm{sen}(\pi y)). \tag{5.48}$$

Para o domínio $D := \{(x,y)\,;\, 0 \le y \le 2, \quad 0 \le x \le 1\}$, definimos as seguintes condições de fronteira em D:

$$\begin{aligned} u(x,0) &= \frac{1}{\pi^2}(\cos(\pi x),\, 0) \in \Gamma_1, \qquad & u(x,2) &= \frac{1}{\pi^2}(\cos(\pi x),\, 0) \in \Gamma_3, \\ u(0,y) &= \frac{1}{\pi^2}(\cos(\pi y),\, 0) \in \Gamma_4, \qquad & u(1,y) &= \frac{1}{\pi^2}(-\cos(\pi y),\, 0) \in \Gamma_2. \end{aligned} \tag{5.49}$$

A solução exata do problema (5.47) com essas condições de fronteira é

$$u = (u_1, u_2) = \frac{1}{\pi^2}(\cos(\pi x)\cos(\pi y),\, \mathrm{sen}(\pi x)\,\mathrm{sen}(\pi y)). \tag{5.50}$$

O procedimento para a determinação da solução numérica é o mesmo feito no capítulo anterior. A construção da malha e a entrada de dados são similares, exceto nas condições de fronteira, que são definidas para as componentes vertical e horizontal, ou seja, definindo a divisão horizontal `dx` por $1/$`Nelx` e a divisão vertical `dy` por $2/$`Nely`.

O número total de nós globais é `Nno` $= ($`Nelx` $+1)($`Nely` $+1)$, o número total de nós globais nas fronteiras Γ_j, $j = 1,2,3,4$ é dada pelo número `Nbn`$[j]$ e as fronteiras são representadas por `bdy`$[j]$ dentro da subrotina `CondFront`. A inserção dos valores de fronteira do tipo Dirichlet (5.49) em cada componente do exemplo é feita da seguinte forma (devido à restrição da linguagem C, a componente horizontal é representada no programa por $r = 0$ e vertical $r = 1$):

$$\begin{aligned}
\Gamma_1 \;=\; & \begin{cases} \mathtt{n} = \mathtt{bdy}[1][\mathtt{i}], \quad p = \mathtt{NoPos}(n), \\ \mathtt{typ}[\mathtt{n}][0] = 0,\ \mathtt{u}[\mathtt{n}][0] = \cos(\pi * x[p.v[0]])/(\pi^2), \\ \mathtt{typ}[\mathtt{n}][1] = 1,\ \mathtt{u}[\mathtt{n}][1] = 0, \end{cases} \\
\Gamma_2 \;=\; & \begin{cases} \mathtt{n} = \mathtt{bdy}[2][\mathtt{i}], \quad p = \mathtt{NoPos}(n), \\ \mathtt{typ}[\mathtt{n}][0] = 0,\ \mathtt{u}[\mathtt{n}][0] = -c\cos(\pi * y[p.v[1]])/(\pi^2), \\ \mathtt{typ}[\mathtt{n}][1] = 1,\ \mathtt{u}[\mathtt{n}][1] = 0, \end{cases} \\
\Gamma_3 \;=\; & \begin{cases} \mathtt{n} = \mathtt{bdy}[3][\mathtt{i}], \quad p = \mathtt{NoPos}(n), \\ \mathtt{typ}[\mathtt{n}][0] = 0,\ \mathtt{u}[\mathtt{n}][0] = \cos(\pi * x[p.v[0]])/(\pi^2), \\ \mathtt{typ}[\mathtt{n}][1] = 1,\ \mathtt{u}[\mathtt{n}][1] = 0, \end{cases} \\
\Gamma_4 \;=\; & \begin{cases} \mathtt{n} = \mathtt{bdy}[4][\mathtt{i}], \quad p = \mathtt{NoPos}(n), \\ \mathtt{typ}[\mathtt{n}][0] = 0,\ \mathtt{u}[\mathtt{n}][0] = \cos(\pi * y[p.v[1]])/(\pi^2), \\ \mathtt{typ}[\mathtt{n}][1] = 1,\ \mathtt{u}[\mathtt{n}][1] = 0. \end{cases}
\end{aligned}$$

Γ_1 : Para $n = 1, 2, \ldots, \texttt{Nelx} + 1$ e $r = 0, 1$,

$$\texttt{typ}[n][r] = 1, \qquad \texttt{u}[n][r] = \frac{1}{\pi^2}(\cos(\pi x), 0).$$

Γ_3 : Para $n = \texttt{Nno} - \texttt{Nelx}, \ldots, \texttt{Nno}$ e $r = 0, 1$,

$$\texttt{typ}[n][r] = 1, \qquad \texttt{u}[n][r] = \frac{1}{\pi^2}(\cos(\pi x), 0).$$

Em Γ_1 e Γ_3 estamos identificando os nós da fronteira para $y = 0$ e $y = 2$, respectivamente, que são nós prescritos nas duas componentes.

Γ_2 : Para $n = \texttt{Nelx} + \texttt{1}, \ldots, \texttt{Nno}$ com incremento de $\texttt{Nelx} + 1$ e $r = 0, 1$,

$$\texttt{typ}[n][r] = 1, \qquad \texttt{u}[n][r] = \frac{1}{\pi^2}(-\cos(\pi y), 0).$$

Γ_4 : Para $n = 1, \ldots, \texttt{Nno} - \texttt{Nelx}$ com incremento de $\texttt{Nelx} + 1$ e $r = 0, 1$

$$\texttt{typ}[n][r] = 1, \qquad \texttt{u}[n][r] = \frac{1}{\pi^2}(\cos(\pi y), 0).$$

Em Γ_4 e Γ_2 estamos identificando respectivamente os nós prescritos na fronteira para $x = 0$ e $x = 1$. Por $\texttt{u}[n][r]$ estamos denotando o valor prescrito na componente r do nó global n.

A matrix global K, construída a partir da matriz local, é de ordem $\texttt{Neq} \times \texttt{Neq}$ e a força local F_m^e é definida por

$$F_m^e = f_m^e + p_m^e - q_m^e.$$

Como a fronteira é do tipo Dirichlet, $p_m^e = 0$. Assim,

$$F_m^e = f_m^e - q_m^e.$$

Na tabela que mostraremos a seguir para a malha 40×80, temos 3.321 nós, 3.200 elementos e 6.162 equações, ou seja, $\{\texttt{Nno} = 3.321,\ \texttt{Nel} = 3.200,\ \texttt{Neq} = 6.162\}$. Dessa forma a matriz global é de ordem 6.162×6.162.

5.4.2 Exemplo 2: Condições de Fronteira do tipo Neumann

Consideremos $\mu = 1$, $\lambda = 1.5$ e a força dada por

$$f_1(x, y) = f_2(x, y) = 4.5\,\text{sen}(\pi x)\ \text{sen}(\pi y) - 2.5\cos(\pi x)\cos(\pi y),$$

o domínio

$$D = \{(x, y)\,,\, 0 \leq y \leq 1\,,\, 0 \leq x \leq 1\}$$

e as condições de fronteira do tipo Neumann

$$\sigma_{ij} n_j = C_{ijk\ell} \frac{\partial u_k}{\partial x_\ell} n_j = p_i,$$

com

$$\begin{aligned} (p_1(x,0), p_2(x,0) = (p_1(x,1), p_2(x,1)) &= -\frac{1}{\pi}\,\mathrm{sen}(\pi x)\,(1,\ 3.5), \\ (p_1(0,y), p_2(0,y) = (p_1(1,y), p_2(1,y)) &= -\frac{1}{\pi}\,\mathrm{sen}(\pi y)\,(3.5,\ 1), \end{aligned} \tag{5.51}$$

o que fisicamente significa que a tração p_i é prescrita na fronteira.

A solução exata nestas condições é dada por

$$u_1(x, y) = u_2(x, y) = \frac{1}{\pi^2}\,\mathrm{sen}(\pi x)\,\mathrm{sen}(\pi y). \tag{5.52}$$

A matriz local K^e_{mn} é a mesma do Exemplo 1, mas a matriz global K é de ordem $\texttt{Neq} \times \texttt{Neq}$, onde `Neq` é o número de equações que dependerá da escolha do procedimento descrito na seção unicidade do problema de Neumann.

Para inserir os dados da fronteira de Neumann (5.51) do exemplo na subrotina `CondFront`, o procedimento é o dado a seguir:

$$\begin{aligned}
\Gamma_1 \;=\; & \begin{cases} \texttt{n} = \texttt{bdy[1][i]}, \quad p = \texttt{NoPos}(n), \\ \texttt{Bv[1][i][0]} = -\,\mathrm{sen}(\pi * x[p.v[0]])/\pi, \\ \texttt{Bv[1][i][1]} = -3.5 * \mathrm{sen}(\pi * x[p.v[0]])/\pi, \end{cases} \\
\Gamma_2 \;=\; & \begin{cases} \texttt{n} = \texttt{bdy[2][i]}, \quad p = \texttt{NoPos}(n), \\ \texttt{Bv[2][i][0]} = -3.5 * \mathrm{sen}(\pi * y[p.v[1]])/\pi, \\ \texttt{Bv[2][i][1]} = -\,\mathrm{sen}(\pi * y[p.v[1]])/\pi, \end{cases} \\
\Gamma_3 \;=\; & \begin{cases} \texttt{n} = \texttt{bdy[3][i]}, \quad p = \texttt{NoPos}(n), \\ \texttt{Bv[3][i][0]} = -\,\mathrm{sen}(\pi * x[p.v[0]])/\pi, \\ \texttt{Bv[3][i][1]} = -3.5 * \mathrm{sen}(\pi * x[p.v[0]])/\pi, \end{cases} \\
\Gamma_4 \;=\; & \begin{cases} \texttt{n} = \texttt{bdy[4][i]}, \quad p = \texttt{NoPos}(n), \\ \texttt{Bv[4][i][0]} = -3.5 * \mathrm{sen}(\pi * y[p.v[1]])/\pi, \\ \texttt{Bv[4][i][1]} = -\,\mathrm{sen}(\pi * y[p.v[1]])/\pi. \end{cases}
\end{aligned}$$

De forma similar ao problema do calor, o problema com dados na fronteira do tipo Neumann não tem solução única devido aos movimentos de rotação e translação. Dessa forma, na próxima seção, descrevemos três procedimentos para obter a unicidade de solução e serão mostrados os erros absolutos e relativos para um dos procedimentos.

No procedimento 1 escolhemos dentro da subroutina `Restriction` a solução cujo deslocamento e momento de deslocamento médios são nulos.

No procedimento 2 fixamos três componentes (duas verticais e uma horizontal) no final da subroutina `CondFront`, forçando a solução a passar pela origem, ou seja, no exemplo 2 estamos fixando, para o nó global $A = 1$, as duas componentes horizontal e vertical e para o último nó (escolha arbitrária) do eixo x, $A = Nelx + 1$ estamos fixando a componente vertical, da seguinte forma:

$\texttt{typ[1][0]} = 1, \quad \texttt{u[1][0]} = 0,$ (comp. horiz.do nó global A=1),
$\texttt{typ[1][1]} = 1, \quad \texttt{u[1][1]} = 0,$ (comp. vert. do nó global A=1),
$\texttt{typ[Nelx + 1][1]} = 1, \ \texttt{u[Nelx + 1][1]} = 0,$ (comp. vert. do nó global A=Nelx+1).

No procedimento 3 fixamos inicialmente apenas duas componentes verticais da seguinte forma:

$\texttt{typ[1][1]} = 1, \quad \texttt{u[1][1]} = 0,$ (comp. horiz. do nó global A=1),
$\texttt{typ[Nelx + 1][1]} = 1, \ \texttt{u[Nelx + 1][1]} = 0,$ (comp. vert. do nó global A=Nelx+1).

Nesse caso, obtemos uma infinidade de soluções, que se diferenciam apenas na translação em relação ao eixo-x. Depois, podemos fixar uma delas dentro da subroutina `GlobalSystem` (como, por exemplo, aquela que passa pela origem) da seguinte forma:

$$\texttt{u[n][0]} -= \texttt{u[1][0]}, \quad n = 2, 3, \ldots, \qquad \texttt{u[1][0]} = 0.$$

Esses foram os procedimentos e os nós escolhidos no programa, mas evidentemente existem muitas formas diferentes de obter uma solução.

• **Força Local**

Em geral, a força local é calculada por:

$$F_m^e = f_m^e + p_m^e - q_m^e\,, \qquad m = 2(a-1) + r\,, \qquad r = 1, 2 \qquad a = 1, 2, 3, 4.$$

Mas nesse caso, $q_m^e = 0$ para todos os elementos, exceto para os nós fixados $A = 1$ (nas duas componentes) e $A = Nelx$ (componente vertical), o que significa que

$$F_1^1 = f_1^1 - q_1^1, \qquad F_2^1 = f_2^1 - q_2^1,$$

$$F_4^{Nelx-1} = f_4^{Nelx-1} - q_4^{Nelx-1}.$$

Note que para a componente horizontal ($r = 1$) do nó $A = Nelx$, temos

$$F_3^{Nelx-1} = f_3^{Nelx-1} - p_3^{Nelx-1}.$$

5.4.3 Exemplo 3: Fronteira mista

Nas mesmas condições do Exemplo 2, onde o domínio D é o retângulo $[0,1] \times [0,1]$, consideremos os valores de $\boldsymbol{u}$ nulos, prescritos nos nós da malha pertencentes ao intervalo com extremidades em $y = 0$ e $y = 1$, isto é,

$$(u_1(x,0), u_2(x,0)) = (u_1(x,1), u_2(x,1)) = (0,0),$$

onde $(u_1(x,0),\, u_2(x,0)) \in \Gamma_1$ e $(u_1(x,1),\, u_2(x,1)) \in \Gamma_3$.

Para os nós da malha pertencentes ao intervalo com extremidades $x = 0$ e $x = 1$, a tração $\boldsymbol{p}$ é prescrita, ou seja

$$(p_1(0,y), p_2(0,y)) = (p_1(1,y), p_2(1,y)) = -\frac{1}{\pi}(3.5\,\mathrm{sen}(\pi y),\, \mathrm{sen}(\pi y)),$$

onde $(p_1(0,y),\, p_2(0,y)) \in \Gamma_4$ e $(p_1(1,y),\, p_2(1,y)) \in \Gamma_2$.

O procedimento para inserir essas condições de fronteira no subrotina `CondFront` é

$$\begin{aligned}
\Gamma_1 &= \begin{cases} \mathtt{n} = \mathtt{bdy[1][i]}, \quad p = \mathtt{NoPos}(n), \\ \mathtt{typ[n][0]} = 0,\ \mathtt{u[n][0]} = 0, \\ \mathtt{typ[n][1]} = 1,\ \mathtt{u[n][1]} = 0, \end{cases} \\
\Gamma_2 &= \begin{cases} \mathtt{n} = \mathtt{bdy[2][i]}, \quad p = \mathtt{NoPos}(n), \\ \mathtt{typ[n][0]} = 0,\ \mathtt{u[n][0]} = -3.5 * \mathrm{sen}(\pi y)/\pi, \\ \mathtt{typ[n][1]} = 1,\ \mathtt{u[n][1]} = -\,\mathrm{sen}(\pi y)/\pi, \end{cases} \\
\Gamma_3 &= \begin{cases} \mathtt{n} = \mathtt{bdy[3][i]}, \quad p = \mathtt{NoPos}(n), \\ \mathtt{typ[n][0]} = 0,\ \mathtt{u[n][0]} = 0, \\ \mathtt{typ[n][1]} = 1,\ \mathtt{u[n][1]} = 0, \end{cases} \\
\Gamma_4 &= \begin{cases} \mathtt{n} = \mathtt{bdy[4][i]}, \quad p = \mathtt{NoPos}(n), \\ \mathtt{typ[n][0]} = 0,\ \mathtt{u[n][0]} = -3.5 * \mathrm{sen}(\pi y)/\pi, \\ \mathtt{typ[n][1]} = 1,\ \mathtt{u[n][1]} = -\,\mathrm{sen}(\pi y)/\pi. \end{cases}
\end{aligned}$$

Nesse caso a força definida por

$$F_m^e = f_m^e + p_m^e - q_m^e, \quad m = 2(a-1) + r, \quad r = 1,2 \quad a = 1,2,3,4$$

será determinada em Γ_2 e Γ_4 por,

$$F_m^e = f_m^e + p_m^e$$

e em Γ_1 e Γ_3 por

$$F_m^e = f_m^e + q_m^e.$$

Usando a força $f(x,y)$ do Exemplo 2, a solução é a mesma do exemplo anterior, isto é,

$$u_1(x,y) = u_2(x,y) = \frac{1}{\pi^2}\,\mathrm{sen}(\pi x)\,\mathrm{sen}(\pi y).$$

Nesse exemplo, para uma malha 50×50, temos $\{\mathtt{Neq} = 4.998, \mathtt{Nno} = 2.601, \mathtt{Nel} = 2.500\}$.

5.4.4 Problema de Neumann: unicidade

Seja u_i uma solução do problema e v_i definido por

$$v_i = u_i + W_{ij}x_j + C_i, \tag{5.53}$$

onde C_i é uma constante de translação e W_{ij} é uma matriz anti-simétrica definida por

$$[W] = \begin{bmatrix} 0 & \theta \\ -\theta & 0 \end{bmatrix}.$$

Verifica-se que v_i também é uma solução do problema. Observamos que por hipótese, na formulação do problema de elasticidade linear, o gradiente do deslocamento é infinitesimal e portanto o valor de θ é teoricamente pequeno e consequentemente a matriz W representa uma rotação infinitesimal de um ângulo θ. Entretanto, matematicamente, a não-unicidade da solução do problema é válida para qualquer valor de θ. Para obter a unicidade descrevemos a seguir três procedimentos para o problema 2.

• **Procedimento 1**

Nesse procedimento vamos impor uma restrição após a resolução do sistema, ou seja, primeiro obtemos uma solução qualquer, pois a matriz rigidez K é singular e consequentemente existem infinitas soluções, em seguida, determinamos as rotação e translação apropriadas para a solução com a referida restrição, como descrito a seguir:

Para assegurar a unicidade de solução podemos determinar a rotação do ângulo θ e a translação C_i de tal forma que satisfaçam as seguintes relações:

$$\int_\Omega v_1(x,y)\,dxdy = 0, \quad \int_\Omega v_2(x,y)\,dxdy = 0, \quad \int_\Omega (v_1(x,y)y - v_2(x,y)x)\,dxdy = 0. \tag{5.54}$$

Essas condições representam a escolha, dentre as infinitas soluções do problema, aquela que tenha o deslocamento médio e momento médio de deslocamento nulos e são equivalentes ao sistema linear abaixo.

$$\begin{bmatrix} I_2 & I_0 & 0 \\ -I_1 & 0 & I_0 \\ I_3 & I_2 & -I_1 \end{bmatrix} \begin{bmatrix} \theta \\ C_1 \\ C_2 \end{bmatrix} = - \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix},$$

onde

$$b_1 = \int_\Omega u_1(x,y)\,dxdy, \quad b_2 = \int_\Omega u_2(x,y)\,dxdy, \quad b_3 = \int_\Omega (u_1(x,y)y - u_2(x,y)x)\,dxdy,$$

$$I_0 = \int_\Omega dxdy, \quad I_1 = \int_\Omega x\,dxdy, \quad I_2 = \int_\Omega y\,dxdy, \quad I_3 = \int_\Omega (x^2+y^2)\,dxdy.$$

Resolvendo o sistema obtemos os valores da rotação θ e da translação C_1, C_2 dados por

$$\begin{bmatrix} \theta \\ C_1 \\ C_2 \end{bmatrix} = \frac{1}{D} \begin{bmatrix} -I_0 I_2 & I_0 I_1 & I_0^2 \\ I_0 I_3 - I_1^2 & -I_1 I_2 & -I_0 I_2 \\ -I_1 I_2 & I_0 I_3 - I_2^2 & I_0 I_1 \end{bmatrix} \begin{bmatrix} -b_1 \\ -b_2 \\ -b_3 \end{bmatrix}, \tag{5.55}$$

onde $D = I_0(I_0 I_3 - I_1^2 - I_2^2)$.

Entretanto, se a solução exata u_i^{ex} do problema é conhecida, em geral, ela não satisfaz as condições de deslocamento e seu momento médio nulos. Então, a fim de poder comparar a solução numérica com a solução exata, em vez das condições (5.54), podemos impor condições tais que as duas soluções tenham os mesmos valores médios de deslocamento e seu momento, ou seja

$$\begin{aligned} &\int_\Omega v_1(x,y)\,dxdy = \int_\Omega u_1^{\text{ex}}(x,y)\,dxdy, \\ &\int_\Omega v_2(x,y)\,dxdy = \int_\Omega u_2^{\text{ex}}(x,y)\,dxdy, \\ &\int_\Omega (v_1(x,y)y - v_2(x,y)x)\,dxdy = \int_\Omega (u_1^{\text{ex}}(x,y)y - u_2^{\text{ex}}(x,y)x)\,dxdy. \end{aligned} \tag{5.56}$$

Assim, a partir de uma solução numérica u_i, obtemos a solução procurada pela determinação da rotação e translação desejadas para a comparação:

$$\begin{bmatrix} \theta \\ C_1 \\ C_2 \end{bmatrix} = \frac{1}{D} \begin{bmatrix} -I_0 I_2 & I_0 I_1 & I_0^2 \\ I_0 I_3 - I_1^2 & -I_1 I_2 & -I_0 I_2 \\ -I_1 I_2 & I_0 I_3 - I_2^2 & I_0 I_1 \end{bmatrix} \begin{bmatrix} e_1 - b_1 \\ e_2 - b_2 \\ e_3 - b_3 \end{bmatrix}, \tag{5.57}$$

onde

$$e_1 = \int_\Omega u_1^{\text{ex}}(x,y)\,dxdy, \;\; e_2 = \int_\Omega u_2^{\text{ex}}(x,y)\,dxdy, \;\; e_3 = \int_\Omega (u_1^{\text{ex}}(x,y)y - u_2^{\text{ex}}(x,y)x)\,dxdy.$$

A solução v_i calculada pela equação (5.53) é a solução procurada que satisfaz as condições (5.56). Porém, na prática, tal solução pode ser insatisfatória, devido ao erros de arredondamento. Neste caso, o procedimento pode ser repetido algumas vezes, até que os valores das rotação e translação, do sistema (5.57) estejam suficientemente próximos de zero. Esse procedimento é feito no programa computacional, dentro da subrotina `Restriction`, sempre que nenhuma componente é fixada, como indica a condicional dentro da subrotina `GlobalSystem`.

Note que o procedimento 1 tem uma importância mais teórica do que prática, pois em geral não conhecemos a solução exata.

• **Procedimento 2**

Nesse procedimento, podemos prescrever as três componentes dentro da subrotina `CondFront`. Observe que as restrições (5.54) ou (5.56) envolvem 3 graus de liberdade, caracterizadas pelas constantes θ, C_1 e C_2. Podemos prefixar 3 valores de deslocamento, ou seja, u_1 e u_2 de um nó, e uma componente (por exemplo u_1) de um outro nó, o que assegura que a matriz rigidez seja positiva definida.

• **Procedimento 3**

Esse procedimento é similar ao anterior, mas ao invés de fixar três componentes, podemos fixar inicialmente apenas duas componentes verticais, permitindo, nesse caso, uma infinidade de soluções que se diferenciam somente pela translação no eixo x, ou seja, se $u(x, y)$ é solução do problema, então $\hat{u}(x, y) = u(x, y) + c$, também é uma solução. Ao final podemos então escolher o valor de c "conveniente", que no exemplo 2 foi tomado $c = 0$, o que na subrotina `GlobalSystem` pode ser feita a seguinte translação:

Para $(n = 2, n <= Nno, n++)$

$\mathtt{u[n][0]} -= \mathtt{u[1][0]}, \mathtt{u[1][0]} = \mathtt{0}$, (Translação da solução passando pela origem)

Os três procedimentos anteriores são para estabelecer, dentre as infinitas soluções possíveis do problema original com fronteira de Neumann, a mais adequada ao modelo físico. No procedimento 1, estamos escolhendo aquela solução cujo deslocamento médio e momento médio de deslocamento são nulos (5.54). Nos procedimentos 2 e 3, estamos escolhendo a solução que passa pela origem.

Para o exemplo 2, geramos soluções para os três tipos de procedimentos, cujos erros relativos e nas normas L^2 e H^1 são dados nas tabelas a seguir:

Erros numéricos: Procedimento 1

Malha	$\| E \|_0$	$\| E \|_1$	$E_0\%$	$E_1\%$
10×10	8.54×10^{-3}	5.88×10^{-2}	12.11	18.76
20×20	1.28×10^{-3}	7.3×10^{-3}	1.79	2.28
50×50	2.33×10^{-5}	1.4×10^{-4}	0.03	0.045

Erros numéricos: Procedimento 2 (Três pontos fixos)

Malha	$\| E \|_0$	$\| E \|_1$	$E_0\%$	$E_1\%$
10×10	2.42×10^{-2}	2.76×10^{-2}	42.45	8.79
20×20	7.13×10^{-3}	7.73×10^{-3}	10.7	2.43
50×50	1.72×10^{-3}	1.8×10^{-3}	2.45	0.57

Erros numéricos: Procedimento 3 (Dois pontos fixos+translação)

Malha	$\| E \|_0$	$\| E \|_1$	$E_0\%$	$E_1\%$
10×10	1.12×10^{-2}	1.37×10^{-2}	16.07	4.39
20×20	3.68×10^{-3}	4.41×10^{-3}	5.21	1.39
50×50	1.14×10^{-3}	1.43×10^{-3}	1.60	0.45

Note que para uma malha 50×50 temos 2.500 elementos, 2.601 nós e o número de equações é igual 5.202 ou 5.199 ou 5.201, dependente respectivamente do procedimento, $\{1,\ 3,\ 2\}$.

5.5 Resultados Numéricos

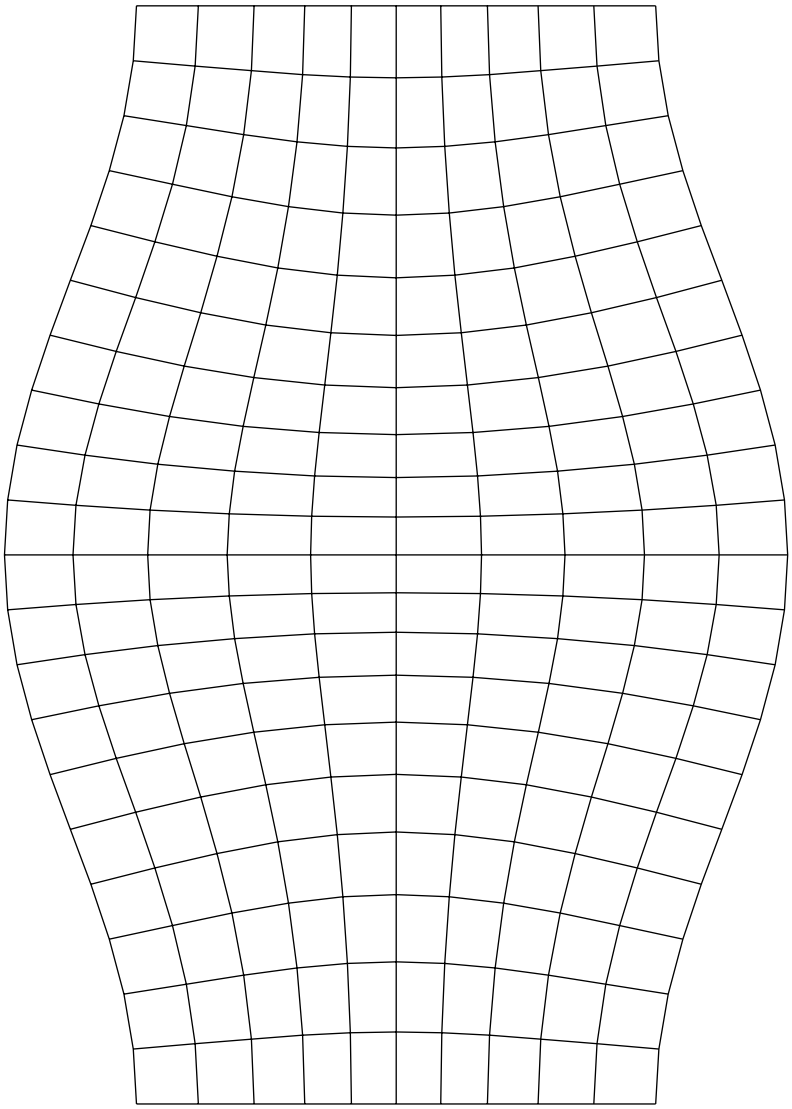
No que segue apresentamos os erros e percentuais de erros na norma $L^2(\Omega)$ e na seminorma $H^1(\Omega)$ para os exemplos considerados. Apresentamos também os gráficos da malha deformada. A posição deformada (x', y') do nó na posição (x, y) da malha é dada por

$$(x', y') = (x + u_1^h(x),\ y + u_2^h(x)),$$

onde $(u_1^h(x),\ u_2^h(x))$ é a solução numérica do deslocamento.

Exemplo 1: Problema de Dirichlet

Malha deformada (10×20) de $\Omega = [0,1] \times [0,2]$

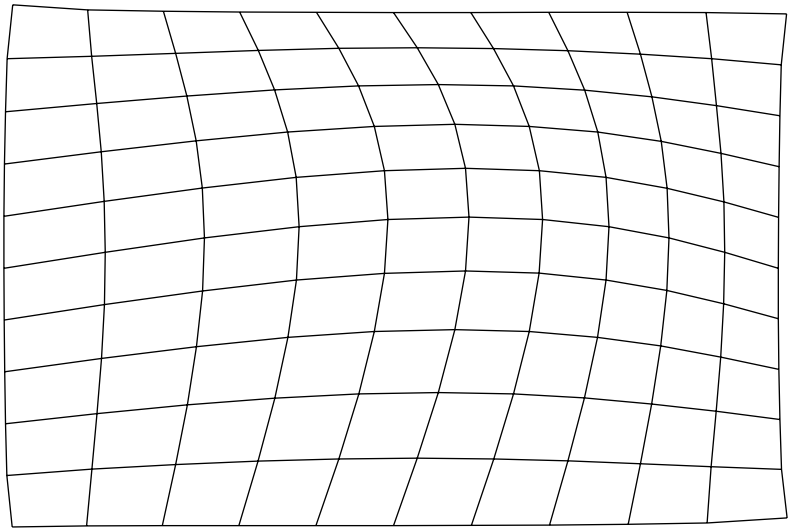


Erros numéricos

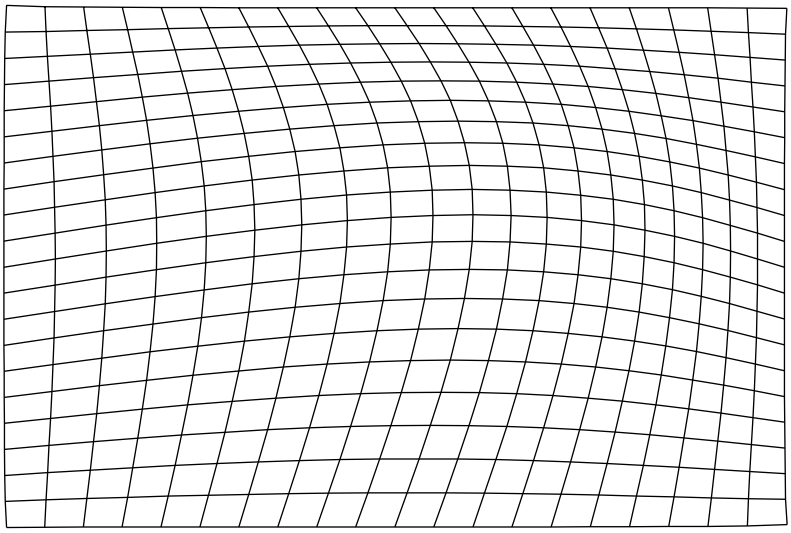
Malha	$\| E \|_0$	$\| E \|_1$	$E_0\%$	$E_1\%$
10×20	7.24×10^{-4}	3.62×10^{-3}	0.73	0.82
20×40	1.85×10^{-4}	9.28×10^{-4}	0.18	0.21
40×80	5.15×10^{-5}	2.59×10^{-4}	0.05	0.06

Exemplo 2: Problema de Neumann

Malha deformada (10×10) de $\Omega = [0,1] \times [0,1]$



Malha deformada (20×20) de $\Omega = [0,1] \times [0,1]$

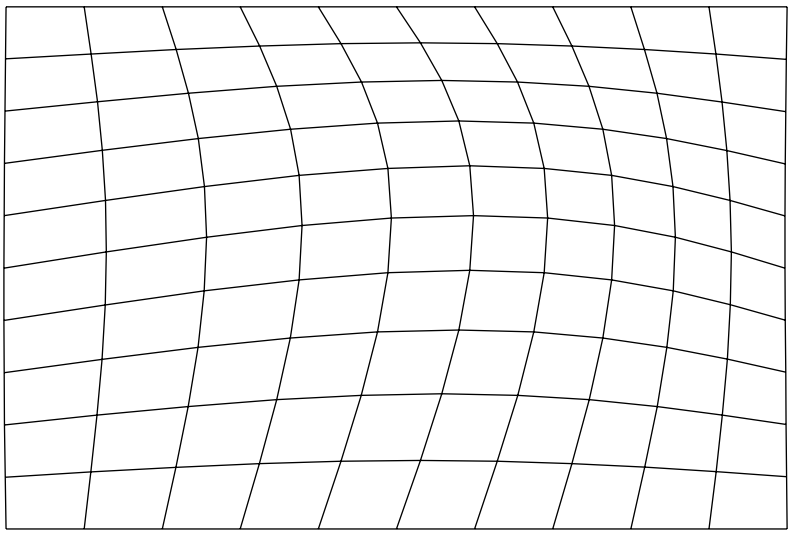


Erros numéricos: Procedimento 3 (Dois pontos fixos+translação)

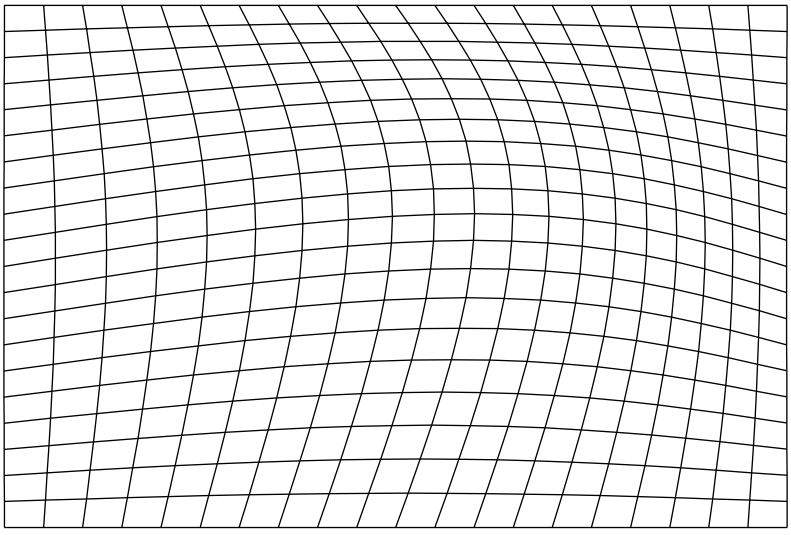
Malha	$\| E \|_0$	$\| E \|_1$	$E_0\%$	$E_1\%$
10×10	1.12×10^{-2}	1.37×10^{-2}	16.07	4.39
20×20	3.68×10^{-3}	4.41×10^{-3}	5.21	1.39
50×50	1.14×10^{-3}	1.43×10^{-3}	1.60	0.45

Exemplo 3: Problema misto

Malha deformada (10×10) de $\Omega = [0,1] \times [0,1]$



Malha deformada (20×20) de $\Omega = [0,1] \times [0,1]$



Erros numéricos

Malha	$\| E \|_0$	$\| E \|_1$	$E_0\%$	$E_1\%$
10×10	1.99×10^{-3}	6.64×10^{-3}	2.89	2.14
20×20	5.13×10^{-4}	1.71×10^{-3}	0.72	0.54
50×50	1.12×10^{-4}	3.74×10^{-4}	0.16	0.12

5.6 Exercícios

Nos Exemplos 1, 2 e 3 deste capítulo, consideramos um corpo isotrópico. No caso não não é isotrópico, a relação (5.46) não é válida. Suponha que o tensor de elasticidade C_{ijkl} seja caracterizado pelos seguintes coeficientes não nulos:

i - Isotropia Transversal:

$C_{1111} = C_{2222} = C_1,\ C_{1122} = C_2,\ C_{1133} = C_3,\ C_{3333} = C_4,$
$C_{1212} = C_1 - C_2,\ C_{2323} = C_{3131} = C_5,$

onde $C_1 = 3.5,\ C_2 = 1.5,\ C_3 = 1.25,\ C_4 = 2.5,$ e $C_5 = 1.0.$

ii - Simetria cúbica:

$C_{1111} = C_{2222} = C_{3333} = C_1,\ C_{1122} = C_{1133} = C_{2233} = C_2,$
$C_{1212} = C_{2323} = C_{3131} = C_3,$

onde $C_1 = 3.5,\ C_2 = 1.5$ e $C_3 = 1.25\,.$

iii - Simetria Tetragonal:

$C_{1111} = C_{2222} = C_1,\ C_{1122} = C_2,\ C_{1133} = C_{2233} = C_3,\ C_{3333} = C_4,$
$C_{1212} = C_5,\ C_{2323} = C_{3131} = C_6,$

onde $C_1 = 3.5,\ C_2 = 1.5,\ C_3 = 1.25,\ C_4 = 2.5,\ C_5 = 2.25$ e $C_6 = 2.0.$

Considere então a formulação variacional (5.9) do problema (5.5), o programa computacional (`elast.c`), a força horizontal e vertical, os valores de fronteira e a solução exata dos Exemplo 1, Exemplo 2 e Exemplo 3. Note que a hipótese (5.3) permanece válida.

1. Utilizando o programa (`elast.c`), refaça o Exemplo 1, o Exemplo 2 e o Exemplo 3 com a isotropia transversal, simetria cúbica e a simetria tetragonal, respectivamente.

2. Compare a solução numérica com a solução exata do corpo isotrópico (veja (5.50) e (5.52)).

CAPÍTULO 6

Métodos Numéricos e Algoritmos: Equação do Calor

Vamos tratar neste capítulo equações de evolução do tipo parabólico, tomando como modelo a equação do calor. O sistema de equações diferencias ordinárias, consequência da utilização do método de elementos finitos, será resolvido pelo método das diferenças finitas para a discretização do tempo.

A parte de análise matemática sobre existência e unicidade de solução e da análise dos métodos numéricos, tais como estimativas de erro e convergência para o problema semi-discreto e totalmente discreto, será discutido nos próximos capítulos, usando o método da energia. Por simplicidade, trabalharemos no caso unidimensional.

6.1 Equação do Calor

Consideremos o problema (2.1) com a introdução da derivada temporal u_t, ou seja, o seguinte problema parabólico modelo:

$$\begin{cases} u_t(x,t) - \alpha u_{xx}(x,t) + \beta u(x,t) = f(x,t), & \forall\,(x,t) \in (0,1) \times [0,T], \\ u(0,t) = u(1,t) = 0, & \forall t \in [0,T], \\ u(x,0) = u_0(x), & \forall x \in (0,1), \end{cases} \tag{6.1}$$

onde α e β são constantes reais positivas e $f(x,t)$ é uma fonte de calor, $u(0,t)$ e $u(1,t)$ representam as temperaturas na fronteira em cada instante, ou seja, a temperatura nos extremos do intervalo são fixas e $u_0(x)$ é a temperatura inicial da barra. Vamos supor que $f(x,t)$ e $u_0(x)$ sejam funções suficientemente regulares.

A solução $u(x,t)$ modela a temperatura em cada instante $t \in [0,T]$ de um fio fino e totalmente isolado, cuja temperatura nas extremidades é mantida constante a zero graus.

Aqui continuaremos adotando a notação introduzida anteriormente, embora as funções sejam dependentes do tempo. Assim, $w(\cdot,t) \in V$ significa que, para cada t, $w(\cdot,t)$ é uma função definida em Ω. Mais precisamente,

$$\begin{array}{rccc} w: & [0,T] & \longrightarrow & V \\ & t & \longmapsto & w(t) := w(\cdot,t). \end{array}$$

Se V é um espaço de Banach para a norma $\|\ \|_V$, definimos $L^p(0,T;V)$ o espaço das funções v tais que, para quase todo cada $t \in [0,T]$, $v(t) \in V$ e a função real

$$t \mapsto \|v(t)\|_V$$

pertença a $L^p(0,T)$. Nesse caso,

$$\|w\|_{L^p(0,T;V)} = \left(\int_0^T \|w(t)\|_V^p dt \right)^{1/p}, \tag{6.2}$$

define uma norma em $L^p(0,T;V)$. Além disso, se V é Banach para a norma $\|\ \|_V$, o espaço $L^p(0,T;V)$ também é Banach para a norma (6.2).

No contexto do que aqui abordaremos, V será subtituído pelos espaços $H_0^1(\Omega)$ ou $L^2(\Omega)$, com Ω sendo o intevalo $(0,1)$, $p \in [1,\infty)$ ou $p = \infty$. Para $p = \infty$, o espaço $L^\infty(0,T;V)$ denota as funções w tais que $\|w(t)\|_V$ são essencialmente limitadas em $[0,T]$, e nesse caso a norma é definida por

$$\|w\|_{L^\infty(0,T;V)} = \sup_{t\in[0,T]} \|w(t)\|_V < \infty.$$

Vale observar que para uma força $f \in L^2(0,T;L^2(0,1))$, por exemplo, a solução do problema (6.1) pode ser obtida pelo método de separação de variáveis (método de Fourier), no qual a solução $u(x,t)$ é representada por uma série infinita. A dificuldade numérica nesse caso é a convergência lenta da série de Fourier, mesmo quando usada a transformada rápida de Fourier. Essa é uma das razões para introdução do método de elementos finitos.

Como fizemos anteriormente, definimos a forma bilinear definida em $H^1(0,1) \times H^1(0,1)$

$$a(u,v) = \alpha \int_0^1 u_x v_x \, dx + \beta \int_0^1 uv \, dx, \tag{6.3}$$

à qual está associada a norma

$$\|u\|_a^2 = \alpha \|\nabla u\|_0^2 + \beta \|u\|_0^2.$$

Tomando $\nu_1 = \min\{\alpha,\beta\}$ e $\nu_2 = \max\{\alpha,\beta\}$, temos

$$\nu_1 \|u\|_1^2 \le \|u\|_a^2 \le \nu_2 \|u\|_1^2, \tag{6.4}$$

ou seja, as normas $\|\,.\,\|_a$ e $\|\,.\,\|_1$ são equivalentes.

6.1.1 Formulação Variacional para a Equação do Calor

Seja $\mathcal{D}(0,1)$ o espaço das funções de classe C^∞ com suporte compacto em $(0,1)$, que denominamos *espaço das funções testes de* $(0,1)$. Multiplicando a equação (6.1) por $v \in \mathcal{D}(0,1)$ e integrando em $(0,1)$, obtém-se

$$\int_0^1 u_t(t)v\,dx - \alpha\int_0^1 u_{xx}(t)v\,dx + \beta\int_0^1 u(t)v\,dx = \int_0^1 f(t)\,v\,dx, \quad \forall\, v \in \mathcal{D}(0,1). \tag{6.5}$$

Integrando por partes, usando (6.3) e denotando

$$(f : v) = \int_0^1 f(t)\,v\,dx, \tag{6.6}$$

a equação (6.5) é a versão variacional de (6.1), e o problema pode ser formulado como: *determinar uma função* $u(t)$ *satisfazendo*

$$\begin{cases} (u_t : v) + a(u,v) = (f : v)\,, \ \forall\, v \in \mathcal{D}(0,1), \\ (u(0) : v) = (u_0 :, v), \quad \forall\, v \in \mathcal{D}(0,1), \end{cases} \tag{6.7}$$

onde, para simplifcar a notação, estamos omitindo a variável x, isto é, estamos denotando por $u(t)$ a função $u(\cdot,t)$ e $u(0) = u(\cdot,0)$. Como $\mathcal{D}(0,1)$ é denso em $H_0^1(0,1)$, a formulação variacional (6.7) pode ser expressa por

$$\begin{cases} (u_t : v) + a(u,v) = (f : v)\,, \ \forall\, v \in H_0^1(0,1), \\ (u(0) : v) = (u_0 : v), \quad \forall\, v \in H_0^1(0,1), \end{cases} \tag{6.8}$$

Nesse caso, sob certas condições de regularidade, pode-se mostrar que uma solução de (6.8) é solução de (6.1).

6.1.2 Problema Aproximado

Seja $T > 0$ e $V_m = [w_1, w_2, ..., w_m] \subset V = H_0^1(0,1)$, o subespaço vetorial gerado pelos m primeiros elementos da base do espaço de Hilbert V (a existência da base é uma consequência do espaço de Hilbert V ser um espaço separável). Sabemos que toda função $u_m(t) \in V_m$ pode ser escrita como combinação linear dos elementos da base, ou mais precisamente.

$$u_m(t) = \sum_{i=1}^{m} d_i(t) w_i(x). \tag{6.9}$$

Podemos então estabelecer o problema aproximado restringindo (6.7) ao subespaço V_m, de modo a transformar o problema original em um sistema de equações diferenciais ordinárias. Mais precisamente,

> **Problema Aproximado:** *Para $m \in \mathbb{N}$, determinar uma função $u_m : [0,T] \to V_m$, $t \mapsto u_m(t) \in V_m$, solução da seguinte equação aproximada*:

$$\begin{cases} (u_m'(t) : v_m) + a(u_m(t), v_m) = (f(t) : v_m), \ \forall\, v \in V_m, \\ u_m(0) = u_{0m} \in V_m, \end{cases} \tag{6.10}$$

Para que a solução do problema (6.10) (com $m \in \mathbb{N}$ fixado) seja uma aproximação do problema original, devemos escolher a condição inicial de tal forma que a sequência $\{u_{0m}\}_m$ seja convergente para u_0 em V. Em particular, podemos escolher essa sequência da forma

$$u_{0m} = \sum_{i=1}^{m} (u_0 : w_i) w_i.$$

Observemos que u_{0m} assim definida é a projeção ortogonal de u_0 sobre o subespaço V_m, de modo que temos, necessariamente,

$$\lim_{m\to\infty} u_{0m} = \sum_{i=1}^{\infty} (u_0 : w_i) w_i = u_0 \quad \text{em } V.$$

Como veremos posteriormente, a sequência $\{u_m\}_{m\in\mathbb{N}}$ formada pela soluções dos problemas aproximados converge para uma solução u do problema (6.7). Entretanto, interessa-nos aqui desenvolver métodos para resolver numericamente o problema aproximado (6.10).

Substituindo (6.9) em (6.10), obtemos das propriedades do produto escalar e da forma bilinear a,

$$\sum_{i=1}^{m} d_i'(t)(w_i : v_m) + \sum_{i=1}^{m} d_i(t) a(w_i, v_m) = (f(t) : v_m), \ \forall\, v_m \in V.$$

Como a igualdade é válida para todo $v_m \in V_m$, podemos tomar em particular $v_m = w_j$, de modo que

$$\sum_{i=1}^{m} d_i'(t)(w_i : w_j) + \sum_{i=1}^{m} d_i(t) a(w_i, w_j) = (f(t) : w_j).$$

Definindo as matrizes simétricas A e B, cujos coeficientes são definidos respectiva-

mente por

$$\begin{aligned} A_{ij} &= (w_i : w_j) = \textstyle\int_0^1 w_i(x) w_j(x)\, dx, \\ B_{ij} &= a(w_i, w_j) = \alpha \displaystyle\int_0^1 w_i'(x) w_j'(x)\, dx + \beta \int_0^1 w_i(x) w_j(x)\, dx, \end{aligned} \tag{6.11}$$

obtemos o seguinte sistema de m equações diferenciais ordinárias

$$\sum_{i=1}^{m} d_i'(t)\; a_{ij} + \sum_{i=1}^{m} d_i(t)\; b_{ij} = F_j(t), \quad \text{para} \quad j = 1, \ldots, m,$$

onde $\boldsymbol{F}(t)$ é o vetor de compontents

$$F_j(t) = (f(t) : w_j) = \int_0^1 f(x,t) w_j(x)\, dx, \quad j = 1, \ldots, m.$$

Assim, o sistema acima pode ser escrito na forma matricial

$$\begin{cases} A\boldsymbol{d}'(t) + B\boldsymbol{d}(t) = \boldsymbol{F}(t), & \forall t \in (0,T) \\ \boldsymbol{d}(0) = \Big((u_0, w_1), (u_0, w_2), \ldots, (u_0, w_m)\Big) = \boldsymbol{d}_0, \end{cases} \tag{6.12}$$

onde $\boldsymbol{d}(0)$ é a condição inicial e $\boldsymbol{d}(t) = (d_1(t), d_2(t), \ldots, d_m(t))^T$ é o vetor incógnita. No contexto da difusão de calor, a matriz A é chamada de matriz capacitância, a matriz B é chamada de matriz de condutividade e $\boldsymbol{F}(t)$ é uma fonte de calor. Note que $A + B$ é a mesma matriz rigidez K definida em (2.13), tomando $\alpha = \beta = 1$. Além disso, como provado em (2.15) e (2.16), as matrizes A e B são simétricas e definidas positiva. Em particular, como a matriz A é uma matriz de Gram para os vetores linearmente independentes $w_1, \ldots, w_m$, A é invertível e assim podemos escrever o sistema de equações diferenciais ordinárias na forma

$$\boldsymbol{d}'(t) + A^{-1}B\boldsymbol{d}(t) = A^{-1}\boldsymbol{F}(t), \quad \forall t > 0, \quad \text{com} \quad \boldsymbol{d}(0) = \boldsymbol{d}_0,$$

e portanto tem solução única $\boldsymbol{d}(t)$ para $t \in [0,T]$. Assim, por (6.9), a solução aproximada $u_m(x,t)$ do problema (6.10) pode ser calculada.

A solução $\boldsymbol{d}(t)$ pode ser expressa em termos da matriz exponencial de $A^{-1}B$, que é diagonalizável, mas na prática uma solução numérica aproximada é mais adequada.

Vamos então retornar ao sistema de equações diferenciais ordinárias (6.12), obtido a partir do método de elementos finitos para a aproximação espacial. Esse sistema será então resolvido para tempos discretos t_n, utilizando-se o método das diferenças finitas.

Para isso, consideramos o sistema de m equações diferenciais ordinárias nos tempos discretos t_n, onde $t_n = n\Delta t$ para $n = 1, 2, \ldots, N$. Assim, temos

$$\begin{cases} A\boldsymbol{d}'(t_n) + B\boldsymbol{d}(t_n) = \boldsymbol{F}(t_n), \\ \boldsymbol{d}(0) = d_0. \end{cases} \tag{6.13}$$

6.2 Algoritmos para a Equação do Calor

Vamos agora introduzir alguns dos métodos numéricos mais conhecidos da literatura (ver em [9, 10]) para a resolução numérica do sistema (6.13), ou seja, para obtenção do vetor $\boldsymbol{d}^n = (d_1^n, d_2^n, \ldots, d_m^n)$, $n = 1, 2, \ldots, N$.

Uma vez conhecida a função base $\{w_i\}_{i=1}^m$ e calculado o vetor $\boldsymbol{d}^n$, podemos expressar a solução aproximada $u_m(x, t_n)$, $n = 1, 2, \ldots, N$, por

$$u_m^n(x) := u_m(x, t_n) = \sum_{i=1}^{m} d_i(t_n) w_i(x) = \sum_{i=1}^{m} d_i^n w_i(x). \tag{6.14}$$

6.2.1 Método de Euler Regressivo

Da diferença regressiva no tempo (veja (1.62)) temos

$$(d_i'(t))_{t=t_n} \approx \frac{1}{\Delta t}\Big(d_i^n - d_i^{n-1}\Big),$$

onde cada função d_i, como sabemos, depende somente da variável tempo t. Substituindo a aproximação no sistema (6.13), obtemos

$$\frac{1}{\Delta t} A\Big(\boldsymbol{d}^n - \boldsymbol{d}^{n-1}\Big) + B\boldsymbol{d}^n = \boldsymbol{F}^n, \qquad n = 1, 2 \ldots N \tag{6.15}$$

ou equivalentemente

$$(A + \Delta t B)\boldsymbol{d}^n = \Delta t \boldsymbol{F}^n + A\boldsymbol{d}^{n-1} = \boldsymbol{b}^n, \qquad n = 1, 2, \ldots N. \tag{6.16}$$

O vetor independente $\boldsymbol{b}^n = (b_1^n, b_2^n, \ldots b_m^n)^T$ está definido, pois A, B, $\boldsymbol{F}$ e o incremento Δt são conhecidos e, além disso, as matrizes A e B são independentes do tempo.

Note que o método de Euler é também um método implícito (trivialmente explicitável se as matrizes A e B são matrizes diagonais, isto é, os vetores bases w_i formam uma base ortogonal).

O vetor incógnita $\boldsymbol{d}^n = (d_1^n, d_2^n, \ldots, d_m^n)^T$ pode ser determinado de forma única, pois a matriz $A + \Delta t B$ é simétrica e definida positiva, portanto não singular. Logo, o sistema pode ser resolvido para cada n fixo usando, por exemplo, os métodos diretos de *Cholesky* (somente quando a matriz é simétrica e definida positiva), *Crout* (LDL^t) (para matriz simétrica) ou o método de eliminação de Gauss.

Observe que a matriz ser esparsa depende diretamente da escolha da base $w_i(x)$. Por exemplo, se $w_i(x)$ for a base linear ou a base cúbica, a matriz $A + \Delta t B$ será tridiagonal ou pentagonal, respectivamente.

Para cada $n = 1, 2, \ldots N$, temos um sistema linear e assim, para se determinar a solução $u(x,t)$ nos tempos discretos t_n, são necessários resolver N sistemas lineares, onde $t_1 = \Delta t$ e $t_N = T$.

• Algoritmo

Para inicialização do método iterativo, faz-se $n = 1$ em (6.16), obtendo-se

$$(A + \Delta t B)\boldsymbol{d}^1 = \Delta t \boldsymbol{F}^1 + A\boldsymbol{d}^0 = \boldsymbol{b}^1.$$

Observe que $F_j^1 = \Big(f(\cdot, t_1) : w_j\Big)$ é dado e $\boldsymbol{d}^0 = \boldsymbol{d}(0)$ é a condição inicial dada. Logo, resolvendo o sistema, determina-se $\boldsymbol{d}^1$, e assim, sucessivamente, obtém-se $\{\boldsymbol{d}^2, \boldsymbol{d}^3, \ldots, \boldsymbol{d}^N\}$, a partir do sistema linear

$$(A + \Delta t B)\boldsymbol{d}^n = \Delta t \boldsymbol{F}^n + A\boldsymbol{d}^{n-1} = \boldsymbol{b}^n, \qquad n = 2, 3, \ldots N. \tag{6.17}$$

Para o método de Euler regressivo, a aproximação no tempo por diferenças finitas tem ordem de convergência proporcional a Δt, o que significa dizer que o erro tem ordem de convergência Δt, denotado por $\mathcal{O}(\Delta t)$.

6.2.2 Método de Euler Progressivo

Usando a aproximação da derivada pela diferença progressiva no tempo (1.62), obtemos

$$(\boldsymbol{d}'(t))_{t=t_n} \approx \frac{1}{\Delta t}\Big(\boldsymbol{d}^{n+1} - \boldsymbol{d}^n\Big).$$

De forma análoga ao método anterior, substituindo a aproximação em (6.13), obtemos o seguinte sistema linear:

$$A\boldsymbol{d}^{n+1} = (A - \Delta t B)\boldsymbol{d}^n + \Delta t \boldsymbol{F}^n, \qquad n = 0, 1, \ldots N. \tag{6.18}$$

Assim, para cada $n = 0, 1, \ldots, N$, obtém-se os valores de $\boldsymbol{d}^n$ e portanto a solução aproximada. Note que a matriz A não é singular e portanto o sistema tem solução única.

Mostraremos adiante que o método iterativo (6.16) (Euler regressivo) é *incondicionalmente* convergente. Por outro lado, o método iterativo (6.18) (Euler progressivo) é *condicionalmente* convergente (ver [23]). Além disso, poder-se-ia perguntar por que não usar a diferença central (1.61) para a aproximação da derivada temporal, que tem uma precisão de $\mathcal{O}(\Delta t^2)$? A resposta é que o método não é convergente (ver [19]).

O método que descreveremos a seguir tem um precisão melhor, isto é, a ordem de convergência é $\mathcal{O}(\Delta t^2)$.

6.2.3 Método de Crank-Nicolson

O método consiste em considerar a diferença progressiva no tempo (1.62) para o termo $\boldsymbol{d}'(t)$ e a média aritmética nos demais termos que dependem de t e não envolvem derivas no tempo. Mais precisamente,

$$\boldsymbol{d}^{n+\frac{1}{2}} = \frac{1}{2}(\boldsymbol{d}^{n+1} + \boldsymbol{d}^n) \qquad \text{e} \qquad \boldsymbol{F}^{n+\frac{1}{2}} = \frac{1}{2}(\boldsymbol{F}^{n+1} + \boldsymbol{F}^n).$$

Fazendo as substituições no sistema (6.13) obtemos

$$\frac{A}{\Delta t}\Big(\boldsymbol{d}^{n+1} - \boldsymbol{d}^n\Big) + \frac{B}{2}(\boldsymbol{d}^{n+1} + \boldsymbol{d}^n) = \frac{1}{2}(\boldsymbol{F}^{n+1} + \boldsymbol{F}^n),$$

que é equivalente a

$$(2A + \Delta t\ B)\boldsymbol{d}^{n+1} = (2A - \Delta t B)\boldsymbol{d}^n + \Delta t(\boldsymbol{F}^{n+1} + \boldsymbol{F}^n), \qquad n = 0, 1, \ldots, (N-1). \quad (6.19)$$

Como a função $f(x,t)$ é dada para todo x e t, $F_j^n = (f(\cdot, t_n) : w_j)$ está bem definido para todo n e j.

Para cada $n = 0, 1, \ldots (N-1)$, temos que resolver o sistema linear (6.19) e assim determinar a solução aproximada nos tempos discretos $t_n = n\Delta t$.

• Algoritmo

Para inicialização do método iterativo, faz-se $n = 0$ em (6.19), obtendo-se

$$(2A + \Delta t B)\boldsymbol{d}^1 = (2A - \Delta t B)\boldsymbol{d}^0 + \Delta t(\boldsymbol{F}^1 + \boldsymbol{F}^0) = \boldsymbol{b}^0,$$

sendo $\boldsymbol{F}^0, \boldsymbol{F}^1$ conhecidos e $\boldsymbol{d}^0 = \boldsymbol{d}(0)$ é a condição inicial dada. Como $(A + \Delta t B)$ é não singular, determina-se de forma única $\boldsymbol{d}^1$ e, sucessivamente, $\boldsymbol{d}^2, \boldsymbol{d}^3, \ldots, \boldsymbol{d}^N$, resovendo-se o sistema (6.19).

Nas seções posteriores mostraremos que a ordem de convergência é $\mathcal{O}(\Delta t^2)$, que pode ser visto também em {[9],[19],[20]}.

6.2.4 Método Generalizado Trapezoidal: (θ-método)

Considere o procedimento similar ao Método de Crank-Nicolson, mas ao invés de usar a média aritmética, usaremos a seguinte média ponderada com o peso θ:

$$\boldsymbol{d}^{n+\theta} = \theta\boldsymbol{d}^{n+1} + (1-\theta)\boldsymbol{d}^n \qquad \text{e} \qquad \boldsymbol{F}^{n+\theta} = \theta\boldsymbol{F}^{n+1} + (1-\theta)\boldsymbol{F}^n, \qquad \theta \in [0,1].$$

Usando a derivada progressiva para o termo da primeira derivada e a média ponderada para os outros termos, obtemos a seguinte família de equações indexada por θ:

$$\Big(A+\theta\Delta tB\Big)\boldsymbol{d}^{n+1}=\Big(A-(1-\theta)\Delta tB\Big)\boldsymbol{d}^{n}+\Delta t\Big(\boldsymbol{F}^{n+\theta}\Big),\quad n=0,1,\ldots N-1 \tag{6.20}$$

Podemos notar que:

(i) Se $\theta=0$, obtemos o método de Euler progressivo (6.18).

(ii) Se $\theta=1/2$, obtemos o método de Crank-Nicolson (6.19).

(iii) Se $\theta=1$, obtemos o método de Euler regressivo (6.16).

• Algoritmo 1

A inicialização do método é semelhante ao que foi feito anteriormente, ou seja, faz-se $n=0$ em (6.20) e resolve-se o sistema linear, obtendo-se o vetor solução $\boldsymbol{d}^1$, a partir do qual obtém-se sucessivamente $\boldsymbol{d}^2,\boldsymbol{d}^3,\ldots,\boldsymbol{d}^N$.

Note que no algoritmo anterior, para (6.20), não há nenhuma preocupação em calcular o valor da derivada de $\boldsymbol{d}^n$, ou seja, calcular $\boldsymbol{d}'(t_n)$.

A seguir, apresentamos um algoritmo, tipo preditor-corretor, que determina a solução de $\boldsymbol{d}^n$ e sua derivada $\boldsymbol{v}^n=\boldsymbol{d}'(t_n)$.

• Algoritmo 2: Método Generalizado Trapezoidal

Considere o seguinte algoritmo:

$$\begin{cases} A\boldsymbol{v}^{n+1}+B\boldsymbol{v}^{n+1}=\boldsymbol{F}^{n+1},\\ \boldsymbol{d}^{n+1}=\boldsymbol{d}^{n}+\Delta t\boldsymbol{v}^{n+\theta},\\ \boldsymbol{v}^{n+\theta}=\theta\boldsymbol{v}^{n+1}+(1-\theta)\boldsymbol{v}^{n}, \end{cases} \tag{6.21}$$

onde $\boldsymbol{v}^n$ e $\boldsymbol{d}^n$ são aproximações de $\boldsymbol{d}'(t_n)$ e $\boldsymbol{d}(t_n)$ respectivamente e $\boldsymbol{F}^{n+1}=\boldsymbol{F}(t_{n+1})$, (Δt) é o incremento do tempo e $\theta\in[0,1]$ é um parâmetro.

O problema computacional é para determinar $\boldsymbol{d}^{n+1}$ e $\boldsymbol{v}^{n+1}$ a partir dos valores conhecidos do tempo anterior $\boldsymbol{d}^n$ e $\boldsymbol{v}^n$.

• Inicialização do Algoritmo

No tempo $t=0$, a temperatura inicial $\boldsymbol{d}^0$ é conhecida, de modo que $\boldsymbol{v}^0=d'(0)$ pode ser determinado fazendo $t=t_0=0$ na equação discreta (6.13), ou seja

$$A\boldsymbol{v}^0=\boldsymbol{F}^0-B\boldsymbol{d}^0.$$

Resolvendo este sistema linear, determina-se $\boldsymbol{v}^0 = \boldsymbol{d}'(0)$.

Para $n = 0, 1 \ldots, (N-1)$ o procedimento para determinar a solução aproximada é dividido nas seguintes etapas:

(i) Definimos um preditor para $\boldsymbol{d}^{n+1}$ na forma:

$$\widetilde{\boldsymbol{d}}^{n+1} = \boldsymbol{d}^n + (1-\theta)\Delta t \boldsymbol{v}^n.$$

(ii) De $(6.21)_2$ e $(6.21)_3$ tem-se que

$$\boldsymbol{d}^{n+1} = \boldsymbol{d}^n + \Delta t \boldsymbol{v}^{n+\theta} = \boldsymbol{d}^n + \Delta t\Big(\theta \boldsymbol{v}^{n+1} + (1-\theta)\boldsymbol{v}^n\Big) = \widetilde{\boldsymbol{d}}^{n+1} + \theta \Delta t \boldsymbol{v}^{n+1}.$$

(iii) Substituindo (ii) em $(6.21)_1$, temos

$$\Big(A + \theta \Delta t B\Big)\boldsymbol{v}^{n+1} = \boldsymbol{F}^{n+1} - B\widetilde{\boldsymbol{d}}^{n+1}.$$

Resolvendo o sistema linear obtém-se $\boldsymbol{v}^{n+1}$ e com este calculado, os valores de $\boldsymbol{d}^{n+1}$ são determinados pela relação (ii), ou seja,

$$\boldsymbol{d}^{n+1} = \widetilde{\boldsymbol{d}}^{n+1} + \theta \Delta t \boldsymbol{v}^{n+1},$$

e assim sucessivamente para $n = 0, 1 \ldots (N-1)$.

6.3 Simulação Numérica: Equação do Calor

Nessa seção faremos simulações numéricas do problema parabólico (6.1) utilizando os diferentes métodos apresentados. Para verificar a eficiência dos métodos, serão construídos problemas onde a solução exata é conhecida, possibilitando então comparar com as soluções numéricas em diferentes normas. Além disso, o conhecimento da solução exata possibilita o cálculo do erro definido por

$$E(x_i, t_n) := u(x_i, t_n) - u_h(x_i, t_n). \tag{6.22}$$

As normas $L^\infty(0,T;L^2(0,1))$ e $L^\infty(0,T;H^1(0,1))$ do erro são definidas por

$$\begin{aligned} \|E\|_{L^\infty(0,\ T;L^2(0,\ 1))} &:= \max_{t\in[0,T]} \left(\int_0^1 |E(x,t)|^2 dx\right)^{1/2}, \\ \|E\|_{L^\infty(0,T;H^1(0,\ 1))} &:= \max_{t\in[0,T]} \left(\int_0^1 \left(|E(x,t)|^2 + \left|\frac{\partial E}{\partial x}(x,t)\right|^2\right) dx\right)^{1/2}, \end{aligned} \tag{6.23}$$

com $n = 1, \ldots, N$.

Para o caso discreto com a malha uniforme, usando a definição dada em (**??**) e denotando $E_i^n = E(x_i, t_n)$ temos, para cada $n = 1, \ldots, N$, as normas

$$\begin{aligned} \|E\|_{L^\infty(0,\ T;L^2(0,\ 1))} &:= \max_{1\le n\le N}\left(h\sum_{i=1}^{m}|E_i^n|^2\right)^{1/2}, \\ \|E\|_{L^\infty(0,T;H^1(0,\ 1))} &:= \max_{1\le n\le N}\left(h\sum_{i=1}^{m}|E_i^n|^2+\frac{1}{h}\sum_{i=1}^{m}|E_{i+1}^n-E_i^n|^2\right)^{1/2}. \end{aligned} \tag{6.24}$$

Como normas em espaços de dimensão finita são equivalentes, podemos considerar

$$\|E\|_{L^\infty(0,T;H^1(0,\ 1))} = \max_{1\le n\le N}\left[\left(h\sum_{i=1}^{m}|E_i^n|^2\right)^{1/2}+\left(\frac{1}{h}\sum_{i=1}^{m}|E_{i+1}^n-E_i^n|^2\right)^{1/2}\right],$$

ou ainda a norma discreta do método dos trapézios $\|\cdot\|_0$ e $\|\cdot\|_1$, definidos em (2.96).

Como veremos no teorema (8.6), sob condições suficientemente regulares, as estimativas de erro para o método generalizado trapezoidal são:

1) para $\theta \in [1/2,\ 1]$,

$$\begin{cases} \|E\|_{L^\infty(0,T;L^2(0,1))} = \|u-u_h\|_{L^\infty(0,T;L^2(0,1))} \le c_1\left(h^{k+1}+(\Delta t)^m\right) \\ \|E\|_{L^\infty(0,T;H_0^1(0,1))} = \|u-u_h\|_{L^\infty(0,T;H_0^1(0,1))} \le c_2\left(h^{k}+(\Delta t)^m\right). \end{cases} \tag{6.25}$$

20 para $\theta \in [0, 1/2)$, tem-se $\Delta t/h^2$ dependente de θ, ou seja, o método é condicionalmente convergente.

Por conta de (6.25), dizemos que os erros $\|E\|_{L^\infty(0,T;L^2(0,1))}$ e $\|E\|_{L^\infty(0,T;H_0^1(0,1))}$ são respectivamente de ordem $\mathcal{O}(h^{k+1}+\Delta t^m)$ e $\mathcal{O}(h^k+\Delta t^m)$.

Note que para $k = 1$, ou seja, quando as funções bases são polinômios lineares por partes, a ordem no espaço é de $\mathcal{O}(h)$ em $H_0^1(0,1)$ e $\mathcal{O}(h^2)$ em $L^2(0,1)$. Além disso, se $\theta = 1/2$, tem-se $m = 2$, e se $\theta \neq 1/2$, $m = 1$. As constantes c_1 e c_2 são positivas e independentes de h e Δt, onde k é o grau do polinômio interpolador da base do subespaço V_m^k.

Do resultado podemos concluir que:

i) O método de Crank-Nicolson, $\theta = 1/2$, tem convergência quadrática no tempo, i.e., $\mathcal{O}(\Delta t)^2$.

ii) O método de Euler progressivo, $\theta = 0$, é condicionalmente convergente.

Para os exemplos numéricos a seguir, consideramos os intervalos discretos $t_n = n/N \in [0,1]$ e $k = 1$ em (6.25). Serão comparados os seguintes métodos:

i) Método de Euler regressivo , $\theta = 1$,

ii) Método de Crank-Nicolson, $\theta = 1/2$,

iii) Método generalizado trapezoidal, $\theta = 1/4$,

iv) Método de Euler progressivo, $\theta = 0$.

Considremos então o problema

$$\begin{cases} u_t(x,t) - \alpha u_{xx}(x,t) + \beta u(x,t) = f(x,t), & \forall\,(x,t) \in (0,1)\times[0,1], \\ u(0,t) = u(1,t) = 0, & \forall t \in [0,1], \\ u(x,0) = (1/\pi^2)\,\mathrm{sen}(\pi x), & \forall x \in (0,1). \end{cases} \tag{6.26}$$

Usando o método de separação de variáveis (método de Fourier), a solução do problema é dada por $u(x,t) = (1/\pi^2)\,\mathrm{sen}(\pi x)\exp(-\lambda t)$, para a força $f(x,t) = (-\lambda + \alpha\pi^2 + \beta)u(x,t)$, onde λ é um parâmetro a ser escolhido. Então

$$f(x,t) = \begin{cases} 0, & \text{se } \lambda = \alpha\pi^2 + \beta, \\ (-\lambda + \alpha\pi^2 + \beta)u(x,t) & \text{se } \lambda \neq \alpha\pi^2 + \beta. \end{cases}$$

Nos exemplos numéricos, consideramos os dois casos em separados, que serão denominados Exemplo 1 e Exemplo 2.

Observação: O resultados numéricos referentes a estes dois exemplos podem ser obtidos usando o Programa Computacional, denominado *Calor.cpp*, que está anexado no final do livro.

Exemplo 1: Nesse exemplo, consideramos $f \equiv 0$, $\alpha = \beta = 1$ e $\lambda = \pi^2 + 1$.

A Tabela 6.1 e a Figura 6.1 mostram o erro na norma $L^2(0,1)$ para cada tempo discreto $t_n = n\Delta t$, tomando $h = \Delta t = 0.05$.

São comparados os θ-métodos com os parâmetros, $\theta = 0.5$ (Crank - Nicolson), $\theta = 0.75$, $\theta = 1.0$ (Euler regressivo), sendo $E_{0.5}, E_{0.75}, E_{1.0}$, representando respectivamente os erros na norma $L^2(0,1)$.

Na Figura 6.1 estão sendo representados por E_{50}, E_{75} e E_{100}. Observe que o método de Crank-Nicolson, como esperado, tem o menor erro em razão da convergência quadrática no tempo, como pode ser observado na Tabela 6.3.

Para $\theta < 0.5$, os métodos numéricos são condicionalmente convergentes e nas mesmas condições de malha, todos os θ-métodos testados divergiram.

A Tabela 6.2 mostra a dependência dos erros em relação ao incremento de tempo Δt, para os θ-métodos, com os parâmetros, $\theta = 0.5$ (Crank-Nicolson), $\theta = 0.75$, $\theta = 1.0$ (Euler regressivo).

n	t	$h = \Delta t$	$E_{0.5}$	$E_{0.75}$	$E_{1.0}$
2	0.1	0.05	0.000620	0.002495	0.005398
4	0.2	0.05	0.000343	0.001484	0.003409
6	0.3	0.05	0.000160	0.000742	0.001816
8	0.4	0.05	0.000068	0.000341	0.000892
10	0.5	0.05	0.000028	0.000149	0.000418

Tabela 6.1: Tabela de erro em cada tempo

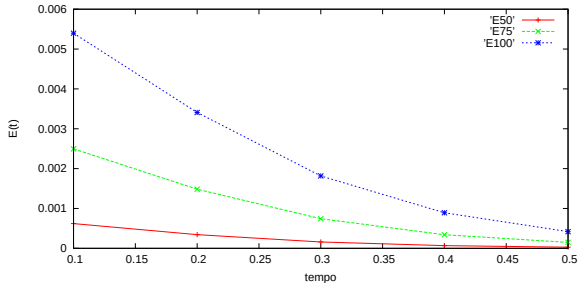


Figura 6.1: Decaimento do erro $\|E(t)\|_{L^2(0,1)}$, $t \in [0.0,\ 0.5]$

• Ordem de Convergência

Vamos analisar numericamente a ordem de convergência do Exemplo 1, onde a solução é conhecida e dada por $u(x,t) = (1/\pi^2)\,\mathrm{sen}(\pi x)\exp(-(\pi^2+1)t)$.

Consideremos

$$E_i := \max_{t\in[0,1]} \|E_i(t)\|_{L^2(0,1)},$$

o erro associado a malha $h = (5\times 2^{i+1})^{-1}$, para $i = 0, 1, \ldots N$ e. em particular, $\Delta t = h$. Então, como vimos em (3.55), a ordem de convergência é $p = \ln(E_i/E_{i+1})/\ln(2)$. Vamos agora calcular a ordem de convergência p para os métodos de Crank-Nicolson e Euler regressivo. A tabela 6.3 mostra que para o método de Crank-Nicolson, $p \approx 2$, e para o Euler regressivo $p \approx 1$.

	Δt	$E_{L^\infty(0,T;L^2(0,\ 1))}$	$E_{L^\infty(0,1;H^1(0,1))}$
$\theta = 1.0$	0.1	0.009911	0.013051
$h = 0.1$	0.05	0.005683	0.007483
$T = 1$	0.025	0.003016	0.003972
$\theta = 0.75$	0.1	0.004351	0.005730
$h = 0.1$	0.05	0.002627	0.003459
$T = 1$	0.005	0.001383	0.001821
$\theta = 0.5$	0.1	0.003176	0.004182
$h = 0.1$	0.05	0.000862	0.001136
$T = 1$	0.025	0.000357	0.000470

Tabela 6.2: Tabela de erros: $E(\Delta t)$

	$\Delta t = h$	$E_{L^\infty(0,T;L^2(0,\ 1))}$	p
$\theta = 1.0$	0.1	0.009911	–
(Euler	0.05	0.005854	0.760
Regressivo)	0.025	0.003215	0.865
	0.0125	0.001691	0.927
	0.00625	0.000868	0.962
$\theta = 0.5$	0.1	0.003176	–
(Crank	0.05	0.000717	2.147
Nicolson)	0.025	0.000175	2.035
	0.0125	0.000044	1.992
	0.00625	0.000011	2.000

Tabela 6.3: Ordem de convergência

Exemplo 2: Considere o problema:

$$\begin{cases} u_t(x,t) - \alpha u_{xx}(x,t) + \beta u(x,t) = f(x,t), & \forall\,(x,t) \in (0,1)\times[0,1], \\ u(0,t) = u(1,t) = 0, & \forall t \in [0,1], \\ u(x,0) = (1/\pi^2)\,\mathrm{sen}(\pi x), & \forall x \in (0,1). \end{cases} \tag{6.27}$$

Para $\lambda = \alpha = \beta = 1$ e $f(x,t) = \mathrm{sen}(\pi x)\exp(-t)$, vimos que a solução do problema é dada por $u(x,t) = (1/\pi^2)\,\mathrm{sen}(\pi x)\exp(-t)$. A Tabela 6.4 mostra a ordem de convergência do método na norma $L^\infty(0,T;L^2(0,\ 1))$.

A Tabela 6.5 compara os erros numéricos entre a solução exata e aproximada do exemplo 2, na norma $L^\infty(0,T;L^2(0,1))$ e $L^\infty(0,T;H^1(0,1))$ aplicando o método de Euler progressivo e regressivo. Note que para $r = \Delta t/h^2 = 0.001/(0.5)^2 = 0.4$, o método de Euler progressivo é divergente. Contudo é convergente para $r = \Delta t/h^2 = 0.001/(0.1)^2 = 0.1$. Como veremos na estimativas de erro, para qualquer θ satisfazendo

	$\Delta t = h$	$E_{L^\infty(0,T;L^2(0,\ 1))}$	p
$\theta = 1.0$	0.1	0.000234	–
(Euler	0.05	0.000123	0.928
Regressivo)	0.025	0.000063	0.965
	0.0125	0.000032	0.977
	0.00625	0.000015	1.093
$\theta = 0.5$	0.1	0.00004	–
(Crank	0.05	0.00001	2.000
Nicolson)	0.025	0.00000	–

Tabela 6.4: Ordem de convergência

a desigualdade, $0.5 \leq \theta \leq 1.0$, o método é convergente.

	Δt	h	$E_{L^\infty(0,T;L^2(0,\ 1))}$	$E_{L^\infty(0,T;H^1(0,\ 1))}$
$\theta = 1.0$	0.001	0.1	0.000003	0.000004
(Euler Regressivo)	0.001	0.05	0.000002	0.000003
$\theta = 0$	0.001	0.1	0.000002	0.000003
(Euler Progressivo)	0.001	0.05	diverge	diverge

Tabela 6.5: Tabela de erros

A figura 6.2 mostra o erro na norma $L^2(0,1)$ para cada tempo $t \in [0, 1.0]$, considerando $\Delta t = 0.1$, $h = 0.05$. As curvas dos erros E_{50}, E_{75}, E_{100}, representam, respectivamente, o erro na norma $L^2(0,1)$, para o $\theta = 0.5$ (Crank-Nicolson), $\theta = 0.75$, $\theta = 1.0$ (Euler regressivo).

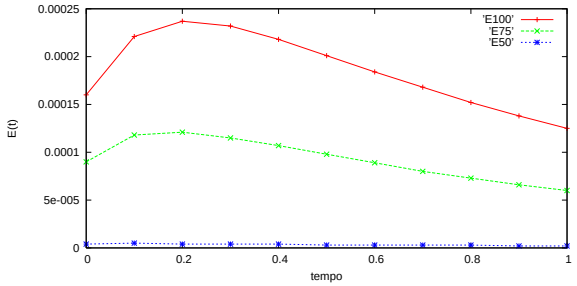


Figura 6.2: Erro $\|E(t)\|_{L^2(0,1)}$, $t \in [0.0,\ 1.0]$

6.4 Exercícios

1. **Bases cúbicas e Hermite:** Considere o problema (6.27) do Exemplo 2.

   (a) Determine as solução numérica aproximada para os métodos definidos na tabela 6.2, usando como função base ($\varphi_i \in V_m$) as funções B-splines, definidas em (3.8). Como as B-splines são polinômios de grau $k \leq 3$, podemos esperar que os erros sejam menores em ambas as normas $L^\infty(0,T,L^2(0,1))$ e $L^\infty(0,T,H_0^1(0,1))$, definidas em (6.24).

   (b) Nas mesmas condições do item anterior, refaça o problema usando como função base os polinômios de Hermite definidos em (3.20).

   (c) Suponha que a solução exata do problema (6.27) seja desconhecida. Verifique a ordem de convergência numérica, definida em (3.55), para o método de Crank-Nicolson e Euler regressivo, usando como função base as funções dos ítens anteriores e compare com a estimativa de erro esperado em (8.89).

2. **Fronteira mista:** Considere o problema (6.27) com $\alpha = \beta = 1$. Sabendo que a solução exata é $u(x,t) = (1/\pi^2)\,\mathrm{sen}(\pi x)\exp(-(\pi^2+1)t)$, o problema com as condições de fronteira de Neumann ($x=0$) e Dirichlet ($x=1$) é dado por

$$\begin{cases} u_t(x,t) - \alpha u_{xx}(x,t) + \beta u(x,t) = 0, & \forall\,(x,t) \in (0,1)\times[0,1], \\ u_x(0,t) = (1/\pi)\exp(-(\pi^2+1)t), & \forall t \in [0,1], \\ u(1,t) = 0, & \forall t \in [0,1], \\ u(x,0) = (1/\pi^2)\,\mathrm{sen}(\pi x), & \forall x \in (0,1). \end{cases} \tag{6.28}$$

   Determine as soluções numéricas aproximadas, usando os (θ-métodos) numéricos, com $\theta = \{0,\ 0.25,\ 0.5,\ 1.0\}$, variando a malha $h = \{0.1,\ 0.02,\ 0.01,\ 0.001\}$, para $\Delta t = 0.001$ fixo , conforme tabela 6.2. Verifique a convergência dos métodos nas diferentes malhas.

3. **Problema Bidimensional:** Seja $\Omega = (0,1)\times(0,1)$ e considere o problema homogêneo

$$\begin{cases} u_t(x,y,t) - \Delta u(x,y,t) + u(x,y,t) = f(x,y,t), & \forall\,(x,y,t) \in \Omega\times[0,1], \\ u(x,y,t) = 0, & \forall (x,y) \in \partial\Omega \quad \text{e} \quad \forall t \in [0,1], \\ u(x,y,0) = (1/\pi^2)\,\mathrm{sen}(\pi x)\,\mathrm{sen}(\pi y), & \forall (x,y) \in \Omega, \end{cases} \tag{6.29}$$

onde $\partial\Omega = \bigcup_{i=1}^{4} \Gamma_i$,

$$\begin{aligned}
\Gamma_1 &= \{(x,0) \in \partial\Omega;\ 0 \leq x \leq 1\},\\
\Gamma_2 &= \{(1,y) \in \partial\Omega;\ 0 \leq y \leq 1\},\\
\Gamma_3 &= \{(x,1) \in \partial\Omega;\ 0 \leq x \leq 1\},\\
\Gamma_4 &= \{(0,y) \in \partial\Omega;\ 0 \leq y \leq 1\}.
\end{aligned}$$

Considere os modelos dos exemplos numéricos dados no Capítulo IV, onde as matrizes foram explicitamente calculadas, usando como função base, a função linear por partes. Para o problema com fronteira do tipo de Dirichlet, considere em particular $f(x,y,t) = 0$.

Nessas condições, a solução exata do problema (6.29) é dada por

$$u(x,y,t) = (1/\pi^2)\,\mathrm{sen}(\pi x)\,\mathrm{sen}(\pi y)\exp(-(2\pi^2+1)t).$$

Definindo a malha uniforme $h_1 = 0.05$ no eixo x, $h_2 = 0.05$ no eixo y e o passo $\Delta t = 0.001$, determine uma solução numérica do problema e compare com a solução exata, usando o mesmo procedimento do modelo unidimensional.

CAPÍTULO 7

Métodos Numéricos e Algoritmos: Equação da Onda

Neste capítulo, vamos tratar a equação da onda, o exemplo típico de equação hiperbólica. Por razões numéricas, as derivadas em t serão, como anteriormente, aproximadas pelo método das diferenças finitas.

A parte de análise matemática (existência e unicidade de solução) e da análise dos métodos numéricos, tais como estimativas de erro e convergência para os problemas semi-discreto e totalmente discreto, serão discutidas nos próximos capítulos, usando o método da energia. Por simplicidades, vamos nos restringir ao caso unidimensional.

Existe um grande número de métodos numéricos para a resolução de equações diferenciais que podem ser encontrados, por exemplo, em [1, 10].

O problema hiperbólico modelo que estudaremos é:

$$\begin{cases} u_{tt}(x,t) - \alpha u_{xx}(x,t) + \beta u(x,t) = f(x,t), & \forall\,(x,t) \in (0,1)\times[0,T], \\ u(0,t) = u(1,t) = 0, & \forall t \in [0,T], \\ u(x,0) = u_0(x), & \forall x \in (0,1), \\ u_t(x,0) = u_1(x), & \forall x \in (0,1), \end{cases} \tag{7.1}$$

onde $\alpha > 0$ e $\beta \geq 0$ são constantes.

Sob certas condições (por exemplo $\beta = 0$), a solução $u(x,t)$ descreve as pequenas vibrações de uma corda elástica cujos extremos estão fixados e sobre a qual atua uma força externa $f(x,t)$. As funções $u_0(x)$ e $u_1(x)$ descreverm respectivamente a posição inicial e a velocidade inicial de cada ponto da corda.

• Formulação Variacional da Equação da Onda

Seja $\mathcal{D}$ o espaço das funções testes, i.e., as funções de classe C^∞ com suporte compacto em $(0,1)$ e $f \in L^2([0,1]\times[0,T])$. Multiplicando a equação (7.1) por $v \in \mathcal{D}(0,1)$ e integrando em $(0,1)$ obtém-se

$$\int_0^1 u''(x,t)v(x)\,dx - \alpha\int_0^1 u_{xx}(x,t)v(x)\,dx + \beta\int_0^1 u(x,t)v(x)\,dx = \int_0^1 f(x,t)v(x)\,dx. \tag{7.2}$$

Usando a forma bilinear (6.3), podemos escrever a seguinte forma variacional

$$\begin{cases} (u''(t):v) + a(u(t),v) = (f(t):v), \\ (u(0):v) = (u_0:v), \qquad \forall v \in \mathcal{D}(0,1). \\ (u'(0):v) = (u_1:v), \end{cases} \tag{7.3}$$

7.1 Problema Aproximado

Sejam $T>0$ e $\{w_k\}_{k\in\mathbb{N}}$ uma base do espaço de Hilbert V. Fixado $m\in\mathbb{N}$, definimos $V_m = [w_1, w_2, ..., w_m]$ o subespaço vetorial gerado pelos m primeiros elementos da base de V. Então, toda função $u_m(t) \in V_m$ pode ser escrita como combinação linear dos elementos dessa base, isto é,

$$u_m(t) = \sum_{i=1}^{m} d_i(t)w_i(x). \tag{7.4}$$

Consideremos o problema variacional (7.3) restrito ao subespaço V_m. Mais precisamente, queremos determinar uma função

$$\begin{array}{rccc} u_m: & [0,T] & \longrightarrow & V_m, \\ & t & \longmapsto & u_m(t), \end{array}$$

solução do seguinte sistema aproximado,

$$\begin{cases} (u_m''(t):v) + a(u_m(t),v) = (f(t):v), \\ u_m(0) = u_{0m}, \qquad\qquad \forall\, v \in V_m, \\ u_m'(0) = u_{1m}, \end{cases} \tag{7.5}$$

onde as sequências $\{u_{0m}\}_{\mathbb{N}}$ e $\{u_{1m}\}_{\mathbb{N}}$ convergem respectivamente para u_0 e u_1 em V.

Como veremos adiante, (vide Teorema 10.4) sobre existência e unicidade de solução), os problemas aproximados (7.5) definem uma sucessão de soluções $\{u_m\}_{m\in\mathbb{N}}$ que converge para solução exata u.

O problema (7.5) é um sistema linear de equações diferenciais ordinárias em relação às coordenadas $d_i(t)$. De fato, substituindo (7.4) em (7.5), obtemos

$$\Big(\sum_{i=1}^{m} d_i''(t)w_i : v_m\Big) + a\Big(\sum_{i=1}^{m} d_i(t)w_i, v_m\Big) = (f : v_m),\ \forall\, v_m \in V_m$$

e pela bilinearidade do produto escalar de V e da forma $a(\cdot,\cdot)$, podemos escrever

$$\sum_{i=1}^{m} d_i''(t)(w_i : v_m) + \sum_{i=1}^{m} d_i(t)a(w_i, v_m) = (f : v_m),\ \forall\, v_m \in V_m.$$

Como a igualdade é válida para todo $v_m \in V_m$, podemos tomar em particular $v_m = w_j$, resultando em

$$\sum_{i=1}^{m} d_i''(t)(w_i : w_j) + \sum_{i=1}^{m} d_i(t)a(w_i, w_j) = \Big(f(t) : w_j).$$

Assim, definindo-se as matrizes simétricas A e B, e o vetor $\boldsymbol{F}$, cujas coordenadas são

$$A_{ij} := (w_i, w_j), \quad B_{ij} := a(w_i, w_j) \quad \text{e} \quad F_j := (f(t), w_j), \tag{7.6}$$

obtemos o seguinte sistema com m equações diferenciais ordinárias

$$\sum_{i=1}^{m} d_i''(t)\ a_{ij} + \sum_{i=1}^{m} d_i(t)\ b_{ij} = F_j, \quad \text{para} \quad j = 1, \cdots, m$$

que pode ser escrito na seguinte forma matricial:

$$\begin{cases} A\boldsymbol{d}''(t) + B\boldsymbol{d}(t) = \boldsymbol{F}(t), \qquad \forall t \in [0,T] \\ \\ \boldsymbol{d}(0) = \boldsymbol{d}_0 := \Big((u_0, w_1), (u_0, w_2), \ldots, (u_0, w_m)\Big), \\ \boldsymbol{d}'(0) = \boldsymbol{v}_0 := \Big((u_1, w_1), (u_1, w_2), \ldots, (u_1, w_m)\Big). \end{cases} \tag{7.7}$$

onde $\boldsymbol{d}(t) = (d_1(t), d_2(t), \ldots, d_m(t))^T$ é o vetor incógnita, $\boldsymbol{d}_0$ é a posição inicial e $\boldsymbol{v}_0$ é a velocidade inicial da onda, dados *à priori*. A matriz A é chamada de matriz massa, enquanto que a matriz B é a já conhecida matriz rigidez. Note que a matriz A é simétrica e definida positiva, como provado no Capítulo 2. Em particular a matriz A é invertível e assim podemos escrever o sistema linear de equações diferenciais ordinárias na forma

$$\boldsymbol{d}''(t) + A^{-1}B\boldsymbol{d}(t) = A^{-1}\boldsymbol{F}(t), \quad \forall t > 0, \quad \text{com} \quad \boldsymbol{d}(0) = \boldsymbol{d}_0,\ \ \boldsymbol{d}'(0) = \boldsymbol{v}_0 \tag{7.8}$$

e, portanto, tem solução única $\boldsymbol{d}(t)$ para $t \in [0, T]$. Assim, por (7.4), a solução aproximada $u_m(x, t)$ do problema (7.5) pode ser calculada.

Em geral, a solução $\boldsymbol{d}(t)$ não é conhecida explicitamente e, dessa forma, fazem-se necessários métodos numéricos para se obter uma solução aproximada para tempos discretos t_n. O método numérico que aplicaremos para resolver o sistema de equações diferenciais ordinárias (7.7) ou (7.8) é o método das Diferenças Finitas.

7.2 Algoritmos para a Equação da Onda

Se considerarmos os tempos discretos t_n, com $t_n = n\Delta t$, $n = 0, 1, \ldots N$, o sistema (7.8) toma a forma

$$\begin{cases} \boldsymbol{d}''(t_n) + A^{-1}B\boldsymbol{d}(t_n) = A^{-1}\boldsymbol{F}(t_n), \quad n = 0, 1, \ldots, N, \\ \boldsymbol{d}(0) = \boldsymbol{d}_0 \qquad \boldsymbol{d}'(0) = \boldsymbol{v}_0, \end{cases} \tag{7.9}$$

No que segue abordaremos alguns esquemas de diferenças finitas para aproximar o termo $\boldsymbol{d}''(t_n)$.

7.2.1 O Método da Diferença Central

Usando a diferença central (1.61), obtemos

$$\frac{1}{(\Delta t)^2}\Big(\boldsymbol{d}^{n+1} - 2\boldsymbol{d}^n + \boldsymbol{d}^{n-1}\Big) + A^{-1}B\boldsymbol{d}^n = A^{-1}\boldsymbol{F}^n, \quad n = 0, 1, \ldots, (N-1) \tag{7.10}$$

que é equivalente a

$$\boldsymbol{d}^{n+1} = \Big(2 - (\Delta t)^2 A^{-1}B\Big)\boldsymbol{d}^n + (\Delta t)^2 A^{-1}\boldsymbol{F}^n - \boldsymbol{d}^{n-1}, \quad n = 0, 1, \ldots, (N-1) \tag{7.11}$$

Assim, resolvendo o sistema linear para cada $t_n = n\Delta t$, obtém-se a única solução do sistema e, consequentemente, a solução aproximada $u_m(x, t_n)$ por (7.4).

• Algoritmo

Para inicialização do método iterativo, faz-se $n = 0$ em (7.11), obtendo-se

$$\boldsymbol{d}^1 = \Big(2 - (\Delta t)^2 A^{-1}B\Big)\boldsymbol{d}^0 + (\Delta t)^2 A^{-1}\boldsymbol{F}^0 - \boldsymbol{d}^{-1} \tag{7.12}$$

Note-se que o termo $\boldsymbol{d}^{-1}$ não está definido, o que nos exige um tratamento especial. Lembrando a aproximação da derivada pela diferença central (1.57), ou seja,

$$(d'(t))_n = \frac{d^{n+1} - d^{n-1}}{2\Delta t},$$

podemos obter o termo $\boldsymbol{d}^{-1}$ a partir da relação

$$\boldsymbol{d}^{-1} = \boldsymbol{d}^1 - 2\Delta t \boldsymbol{d}'(0) = \boldsymbol{d}^1 - 2\Delta t \boldsymbol{v}_0. \tag{7.13}$$

Substituindo em (7.12), obtemos para a primeira iteração do esquema a seguinte expressão

$$\boldsymbol{d}^1 = \left(1 - \frac{(\Delta t)^2}{2} A^{-1} B\right) \boldsymbol{d}^0 + \frac{(\Delta t)^2}{2} A^{-1} \boldsymbol{F}^0 - \Delta t \boldsymbol{v}_0. \tag{7.14}$$

Observe que (7.14) é, na prática, o seguinte sistema linear a ser resolvido:

$$A\boldsymbol{d}^1 = \left(1 - \frac{(\Delta t)^2}{2} B\right) \boldsymbol{d}^0 + \frac{(\Delta t)^2}{2} \boldsymbol{F}^0 - \Delta t A \boldsymbol{v}_0. \tag{7.15}$$

Uma vez calculado $\boldsymbol{d}^1$, as soluções nos tempos $n = 1, 2, \ldots, N$, são obtidas por (7.11), ou seja, pelas soluções dos sistemas lineares

$$A\boldsymbol{d}^{n+1} = \Big(2 - (\Delta t)^2 B\Big)\boldsymbol{d}^n + (\Delta t)^2 \boldsymbol{F}^n - A\boldsymbol{d}^{n-1} \quad n = 1, 2, \ldots, (N-1).$$

As aproximações para a primeira e segunda derivadas em relação ao tempo são ambas de ordem $\mathcal{O}(\Delta t^2)$. Entretanto, o método da Diferença Central é condicionalmente convergente, como mostraremos nas estimativas de erro nos próximos capítulos.

Os métodos que trataremos a seguir são incondicionalmente convergentes para o parâmetro $\theta \geq 1/4$.

7.2.2 O Método de Newmark

Considere o sistema (7.9) e a seguinte aproximação:

$$\frac{1}{(\Delta t)^2} A(\boldsymbol{d}^{n+1} - 2\boldsymbol{d}^n + \boldsymbol{d}^{n-1}) + B\boldsymbol{d}_\theta^n = \boldsymbol{F}_\theta^n, \quad \theta \geq 0, \tag{7.16}$$

onde estamos definindo $\boldsymbol{d}_\theta^n := \boldsymbol{d}^n + \theta\Big(\boldsymbol{d}^{n+1} - 2\boldsymbol{d}^n + \boldsymbol{d}^{n-1}\Big)$.

Fazendo a substituição em (7.16), obtemos o seguinte sistema algébrico de equações lineares

$$\begin{aligned}\Big(A + \theta(\Delta t)^2 B\Big)\boldsymbol{d}^{n+1} = \Big(2A - (1-2\theta)(\Delta t)^2 B\Big)\boldsymbol{d}^n - \Big(A + \theta(\Delta t)^2 B\Big)\boldsymbol{d}^{n-1} \\ + (\Delta t)^2\Big(\theta(\boldsymbol{F}^{n+1} + \boldsymbol{F}^{n-1}) + (1-2\theta)\boldsymbol{F}^n\Big).\end{aligned} \tag{7.17}$$

Definindo as matrizes $M_\theta := \Big(A + \theta(\Delta t)^2 B\Big)$, $L_\theta := \Big(2A - (1-2\theta)(\Delta t)^2 B\Big)$ e o vetor $\boldsymbol{F}_\theta^n := \theta(\boldsymbol{F}^{n+1} + \boldsymbol{F}^{n-1}) + (1-2\theta)\boldsymbol{F}^n$, podemos escrever o sistema linear (7.17) na forma

$$M_\theta \boldsymbol{d}^{n+1} = L_\theta \boldsymbol{d}^n - M_\theta \boldsymbol{d}^{n-1} + (\Delta t)^2 \boldsymbol{F}_\theta^n. \tag{7.18}$$

O método de aproximação acima é conhecido como método de Newmark. Para que o esquema numérico (7.18) seja incondicionalmente estável no tempo, é necessário que $\theta \geq 1/4$, como veremos na análise de convergência. Note-se que o método de Newmark se reduz ao método da diferença central (7.11) no caso $\theta = 0$.

• Algoritmo

Para inicialização do processo, tomemos $n = 0$ em (7.18), obtendo-se

$$M_\theta \boldsymbol{d}^1 = L_\theta \boldsymbol{d}^0 - M_\theta \boldsymbol{d}^{-1} + (\Delta t)^2 \boldsymbol{F}_\theta^0. \tag{7.19}$$

O valor de $\boldsymbol{d}^{-1}$ pode ser obtido pela aproximação (7.13). Se o termo $\boldsymbol{F}^{-1}$ não é conhecido (caso em que a função força $f(x,t)$ não esteja definida para tempos negativos), podemos simplesmente substituir nesta primeira etapa $\boldsymbol{F}_\theta^0$ por $\boldsymbol{F}^0$. Assim, o algoritmo pode ser inicializado por

$$M_\theta \boldsymbol{d}^1 = \frac{1}{2} L_\theta \boldsymbol{d}^0 + \Delta t M_\theta \boldsymbol{v}_0 + \frac{(\Delta t)^2}{2} \boldsymbol{F}_\theta^0. \tag{7.20}$$

Resolvendo-se o sistema linear acima, obtemos a solução $\boldsymbol{d}^1$ e, sucessivamente para cada $n = 1, 2, \ldots, N-1$ obtemos $\boldsymbol{d}^2, \boldsymbol{d}^3, \ldots, \boldsymbol{d}^N$ pelo sistema linear (7.18).

7.2.3 O θ-método

O θ-método, também conhecido como método-θ, foi aplicado anteriormente para a equação do calor, que é um problema de primeira ordem no tempo. Vamos agora aplicá-lo à equação da onda.

Considerando a mudança de variável $v(x,t) = u'(x,t)$, podemos decompor o sistema (7.3) nas seguintes equações:

$$P1: \begin{cases} (v:w) + a(u,w) = (f:w), \\ (v(0):w) = (u'(0):w) = (u_1:w), \end{cases} \quad \forall\, w \in V, \tag{7.21}$$

$$P2: \begin{cases} (u':w) - (v:w) = 0, \\ (u(0):w) = (u_0:w), \end{cases} \quad \forall\, w \in V. \tag{7.22}$$

Definindo,

$$v_m(x,t) = \sum_{i=1}^{m} c_i(t)w_i(x) \quad \text{e} \quad u_m(x,t) = \sum_{i=1}^{m} d_i(t)w_i(x), \tag{7.23}$$

substituindo nos sistemas (7.21) e (7.21) e considerando as matrizes definidas em (7.6), obtemos dois sistemas de equações ordinárias de primeira ordem, a saber,

$$P1: \begin{cases} A\boldsymbol{c}'(t) + B\boldsymbol{d}(t) = \boldsymbol{F}(t), \\ \boldsymbol{c}(0) = \boldsymbol{d}'(0) = \boldsymbol{v}_0, \end{cases} \quad \text{e} \quad P2: \begin{cases} A[\boldsymbol{d}'(t) - \boldsymbol{c}(t)] = \boldsymbol{0}, \\ \boldsymbol{d}(0) = \boldsymbol{d}_0. \end{cases} \tag{7.24}$$

Observe que devido a não singularidade da matriz A, a primeira equação no problema P2 é trivialmente equivalente a $\boldsymbol{d}'(t) - \boldsymbol{c}(t) = \boldsymbol{0}$.

Para cada um dos problemas, associamos no tempo discreto $t_{n+1} = (n+1)\Delta t$, os pesos θ e σ; e no tempo $t_n = n\Delta t$, os pesos $(1-\theta)$ e $(1-\sigma)$, onde os parâmetros θ e σ são pertencentes ao intervalo $[0,\ 1]$, para os quais consideramos as médias ponderadas

$$\boldsymbol{d}^{n+\theta} = \theta\boldsymbol{d}^{n+1} + (1-\theta)\boldsymbol{d}^n \quad \text{e} \quad \boldsymbol{c}^{n+\sigma} = \sigma\boldsymbol{c}^{n+1} + (1-\sigma)\boldsymbol{c}^n, \quad \theta, \sigma \in [0,1].$$

Aplicando aos Problemas (P1) e (P2) os θ-métodos para cada dos problemas separadamente, obtemos:

$$P1: \begin{cases} A\dfrac{(\boldsymbol{c}^{n+1} - \boldsymbol{c}^n)}{\Delta t} + B\boldsymbol{d}^{n+\theta} = \boldsymbol{F}^{n+\theta}, \\ \boldsymbol{c}(0) = \boldsymbol{d}'(0) = \boldsymbol{v}_0, \end{cases} \quad P2: \begin{cases} \dfrac{(\boldsymbol{d}^{n+1} - \boldsymbol{d}^n)}{\Delta t} = \boldsymbol{c}^{n+\sigma}, \\ \boldsymbol{d}(0) = \boldsymbol{d}_0. \end{cases} \tag{7.25}$$

Desenvolvendo os termos, obtemos o sistema para $n = 1, 2, \ldots,$

$$\begin{cases} A\boldsymbol{c}^{n+1} = A\boldsymbol{c}^n - \Delta t B(\theta\boldsymbol{d}^{n+1} + (1-\theta)\boldsymbol{d}^n) + \Delta t(\theta\boldsymbol{F}^{n+1} + (1-\theta)\boldsymbol{F}^n) \\ \boldsymbol{d}^{n+1} = \boldsymbol{d}^n + \Delta t(\sigma\boldsymbol{c}^{n+1} + (1-\sigma)\boldsymbol{c}^n) \end{cases} \tag{7.26}$$

Note que o sistema (7.26) é um sistema acoplado que pode ser expresso na forma

$$\mathcal{K}\boldsymbol{\Phi}^{n+1} = \mathcal{L}\boldsymbol{\Phi}^n + \boldsymbol{F}^{n+\theta}, \tag{7.27}$$

onde

$$\mathcal{K} = \begin{pmatrix} A & \theta\Delta t B \\ -\sigma\Delta t I & I \end{pmatrix}, \qquad \mathcal{L} = \begin{pmatrix} A & (\theta-1)\Delta t B \\ (1-\sigma)\Delta t I & I \end{pmatrix}, \tag{7.28}$$

$$\boldsymbol{\Phi}^n = \begin{pmatrix} \boldsymbol{c}^n \\ \boldsymbol{d}^n \end{pmatrix}, \qquad \boldsymbol{F}^{n+\theta} = \begin{pmatrix} \theta\boldsymbol{F}^{n+1} + (1-\theta)\boldsymbol{F}^n \\ 0 \end{pmatrix}. \tag{7.29}$$

As matrizes A e B são as mesmas definidas em (6.11) e I é a matriz identidade de mesma ordem de A (e B). Logo, a matriz dos coeficientes $\mathcal{K}$, denominada matriz bloco do sistema linear (7.27), tem ordem $2m \times 2m$.

O sistema linear (7.27) tem solução única, pois a matriz bloco $\mathcal{K}$ é não singular. Nessas condições o sistema linear pode ser resolvido pelo método de eliminação de Gauss, fatoração LU ou o método de Uzwa (see [8]).

• Algoritmo

Fazendo $n = 0$, o sistema linear (7.27) toma a forma

$$\mathcal{K}\mathbf{\Phi}^1 = \mathcal{L}\mathbf{\Phi}^0 + \boldsymbol{F}^{0+\theta}.$$

O vetor $\mathbf{\Phi}^0 = (\boldsymbol{c}^0, \boldsymbol{d}^0) = (\boldsymbol{v}_0, \boldsymbol{d}_0)$ é conhecido pelos dados iniciais; e como a força $\boldsymbol{F}$ é conhecido em todo tempo, $\boldsymbol{F}^{0+\theta} = \theta\boldsymbol{F}^1 + (1-\theta)\boldsymbol{F}^0$ é está bem definido. Assim, sendo as matrizes blocos conhecidas, determina-se o vetor $\mathbf{\Phi}^1$ e, daí em diante, para $n = 1, 2, \ldots, (N-1)$.

Observações:

1. Quando $\theta,\ \sigma \geq 1/2$, o método é incondicionalmente estável, como consequência do Teorema 8.6. Além disso, se $\theta = \sigma = 1/2$ (método de Crank-Nicolson), o método tem ordem quadrática no tempo, ou seja, $\mathcal{O}(\Delta t)^2$.

2. Para $(\theta, \sigma) = (0, 1)$, o método é o conhecido método da diferença central, dado em (7.11), e é condicionalmente estável.

3. Para $\theta = \sigma = 1$, obtemos um método implícito de primeira ordem no tempo, $\mathcal{O}(\Delta t)$, similar ao Método de Euler Regressivo.

7.3 Simulação Numérica: Equação da Onda

Nessa seção faremos algumas simulações numéricas do problema hiperbólico modelo, utilizando o método de Newmark para diversos valores de θ. Para os exemplos numéricos, utilizamos como funções base os polinômios lineares por partes. Consideramos exemplos cuja solução exata é conhecida para calcular os erros entre as soluções numérica e exata nas normas definidas em (6.24), ou seja, $L^\infty(0, T; L^2(0, 1))$ e $L^\infty(0, T; H^1(0, 1))$. Os resultados numéricos destes dois exemplos foram obtidos usando o Programa Computacional *Onda.cpp*, cujo arquivo fonte está disponível no final do livro.

Consideremos o problema:

$$\begin{cases} u_{tt}(x,t) - \alpha u_{xx}(x,t) + \beta u(x,t) = f(x,t), & \forall\,(x,t) \in (0,1)\times[0,1], \\ u(0,t) = u(1,t) = 0, & \forall t \in [0,1], \\ u(x,0) = \operatorname{sen}(\pi x) & \forall x \in (0,1), \\ u_t(x,0) = 0 & \forall x \in (0,1), \end{cases} \tag{7.30}$$

α e β reais positivos, com a força $f(x,t)$ definida *a posteriori* de tal forma que a solução exata seja

$$u(x,t) = \cos(\lambda\pi t)\operatorname{sen}(\pi x), \quad \lambda \in \mathbb{R}. \tag{7.31}$$

Assim, substituindo na equação, obtemos

$$f(x,t) = (\beta + (\alpha - \lambda^2)\pi^2)\cos(\lambda\pi t)\operatorname{sen}(\pi x), \quad \lambda \in \mathbb{R}. \tag{7.32}$$

Exemplo 1: Dados $\alpha = \beta = 1$ e $\lambda_1 = \sqrt{1 + 1/\pi^2} \approx 1.05$, a função $u(x,t) = \cos(\lambda_1\pi t)\operatorname{sen}(\pi x)$ é solução de

$$\begin{cases} u_{tt}(x,t) - u_{xx}(x,t) + u(x,t) = 0, & \forall\,(x,t) \in (0,1)\times[0,1], \\ u(0,t) = u(1,t) = 0, & \forall t \in [0,1], \\ u(x,0) = \operatorname{sen}(\pi x), & \forall x \in (0,1), \\ u_t(x,0) = 0, & \forall x \in (0,1). \end{cases} \tag{7.33}$$

Na Tabela 7.1, são mostrados os erros nas normas $L^\infty(0,T;L^2(0,1))$ e $L^\infty(0,T;H^1(0,1))$ referentes ao método de Newmark para diversos valores de θ, com os parâmetros $\Delta t = 0.01$, $h = 0.05$ e $T = 0.5$ fixados.

	θ	$E_{L^\infty(0,T;L^2(0,\ 1))}$	$E_{L^\infty(0,T;H^1(0,\ 1))}$
	0	0.001113	0.001288
$\Delta t = 0.01$	0.25	0.000957	0.001108
$h = 0.05$	0.5	0.000801	0.000927
$T = 0.5$	0.75	0.000660	0.000764
	1.0	0.000536	0.000620

Tabela 7.1: Variando θ

Podemos observar que se θ cresce, o erro decresce nas duas normas. Os resultados teóricos não permitem afirmar que esse resultado é válido em geral, mas esperado sob o ponto de vista numérico, degido à robustez da matriz rigidez.

A Figura 7.1 mostra o erro na norma $E_{L^\infty(0,T;L^2(0,\ 1))}$, para cada parâmetro θ.

Nas tabelas de erros, Tabela 7.2 e Tabela 7.3, são mostradas a dependência dos erros em relação aos parâmetros de discretização espacial h e do parâmetro do passo temporal Δt.

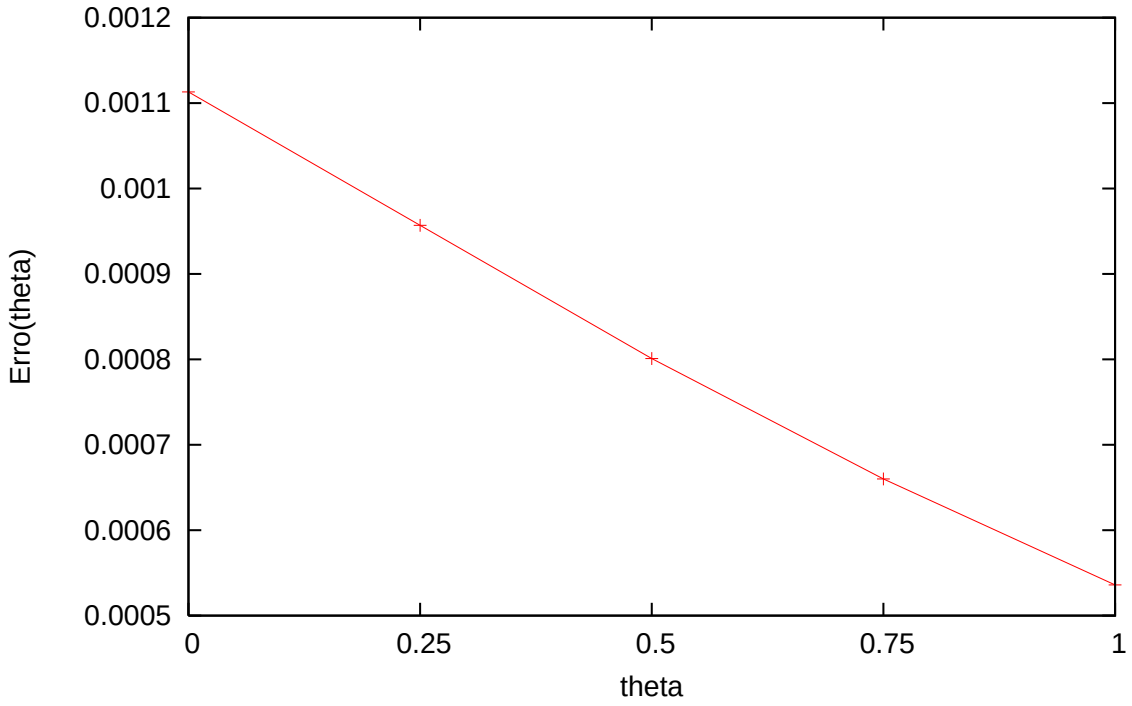


Figura 7.1: $E_{L^\infty(0,T;L^2(0,\ 1))}$

	h	$E_{L^\infty(0,1;L^2(0,1))}$	$E_{L^\infty(0,1;H^1(0,1))}$
$\Delta t = 0.01$	0.1	0.003853	0.005074
$\theta = 0.75$	0.05	0.000660	0.000764
$T = 0.5$	0.025	0.000142	0.000153

Tabela 7.2: Tabela de erro: $E(h)$

	Δt	$E_{L^\infty(0,1;L^2(0,1))}$	$E_{L^\infty(0,1;H^1(0,1))}$
$h = 0.01$	0.05	0.0104	0.010727
$\theta = 0.75$	0.025	0.002637	0.002720
$T = 0.5$	0.0125	0.000697	0.000719

Tabela 7.3: Tabela de erro:$E(\Delta t)$

Para analisar a ordem de convergência, seja, como já visto anteriormente, $E_i = \max_{t\in[0,1]} \|E_i(t)\|_{L^2(0,1)}$, para $i = 0, 1, \ldots, N$, o erro associado a malha $h = (5 \times 2^{i+1})^{-1}$. Tomando em particular $\Delta t = h$, a ordem de convergência é calculada por

$$p = \ln(E_i/E_{i+1})/\ln(2).$$

A Tabela 7.4 mostra que a ordem de convergência do método de Newmark é quadrática e permanece válida para todo $\theta \in [0.25, 1.0]$.

	$\Delta t = h$	$E_{L^\infty(0,T;L^2(0,\ 1))}$	p
$\theta = 0.25$	0.1	0.006704	–
	0.05	0.001698	1.981
	0.025	0.000426	1.995
	0.0125	0.000107	1.995
$\theta = 1.0$	0.1	0.055636	
	0.05	0.014566	1.933
	0.025	0.003687	1.982
	0.0125	0.000922	1.999

Tabela 7.4: Ordem de convergência

Nas estimativas de erros que apresentaremos nos próximos capítulos, mostraremos que para $\theta \in [0,\ 1/4)$, o método de Newmark é condicionalmente convergente, ou seja, o método converge sob a condição de que $\Delta t < h$. Dessa forma, o método da diferença central ($\theta = 0$) é um método condicionalmente convergente. Esse resultado é conhecido como condição CFL (Courant-Friedrichs-Lewy), ou seja, o método é convergente sob a condição $r := (\Delta t/h) < 1$.

Esses resultados podem ser comprovados na Tabela 7.5, onde estão exibidos os erros para alguns valores de θ, mostrando que se $r = (\Delta t/h) = 1$, o método diverge independentemente do tamanho da malha.

A seguir vamos mostrar que o método da diferença central também tem ordem de convergência quadrática, desde que $r = (\Delta t/h) < 1$. Em particular, são mostrados na Tabela 7.6 os erros e ordens de convergência, tomando $\Delta t = h/2$.

A Figura 7.2 foi construída para ilustrar geometricamente, no exemplo 1, as diferentes soluções numéricas associadas ao método de Newmark, com parâmetros $\theta = 0,\ 1/4,\ 1/2,\ 3/4,\ 1.0$. Em razão da precisão do método visto nas tabelas de erros, e da escala usada na elaboração do gráfico, os erros são imperceptíveis, mas há 5 gráficos sobrepostos, um para cada valor de θ.

Na Figura 7.3, apresentamos o erro absoluto entre as soluções exatas e numéricas,

	Δt	h	$E_{L^\infty(0,T;L^2(0,\ 1))}$
$\theta = 0$	0.01	0.1	0.00479
	0.01	0.05	0.00123
	0.01	0.01	**diverge**
$\theta = 0.25$	0.01	0.1	0.00472
	0.01	0.05	0.00106
	0.01	0.01	0.00011
$\theta = 0.5$	0.01	0.1	0.00445
	0.01	0.05	0.00089
	0.01	0.01	0.00024

Tabela 7.5: Tabela de erro

	$\Delta t = h/2$	h	$E_{L^\infty(0,T;L^2(0,\ 1))}$	p
$\theta = 0.0$	0.05	0.1	0.006188	–
	0.025	0.05	0.001561	1.987
	0.0125	0.025	0.000387	1.997
	0.00625	0.0125	0.000097	1.997

Tabela 7.6: Ordem de convergência

obtidas pelo método de Newmark para os valores de θ, denotados por $E_0, E_{25}, E_{50}, E_{75}$ e E_{100}, respectivamente. Note que $E_0 > E_{25} > E_{50} > E_{75} > E_{100}$.

Para elaboração dos gráficos, foram considerados $T = 10.0$, $\Delta t = 0.01$ e $h = 0.05$. Para analisar a evolução no tempo da solução, fixamos o ponto médio $x = 0.5$, ou seja, os gráficos são as soluções aproximadas $u_h(0.5, t)$ da solução exata $u(0.5, t) = cos(\lambda \pi t)$, onde $\lambda_1^2 = (1 + 1/\pi^2)$. Note que foram necessários 1.000 iterações para obter a solução aproximada $u(x; T)$.

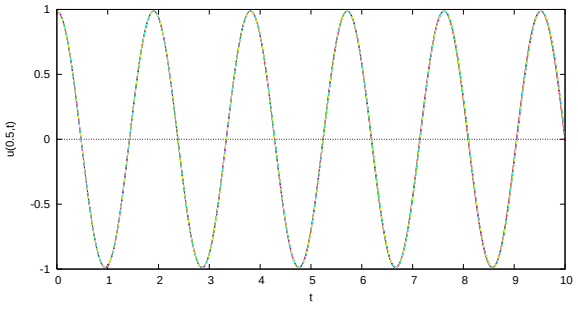


Figura 7.2: Solução aproximada $u_h(0.5, t)$

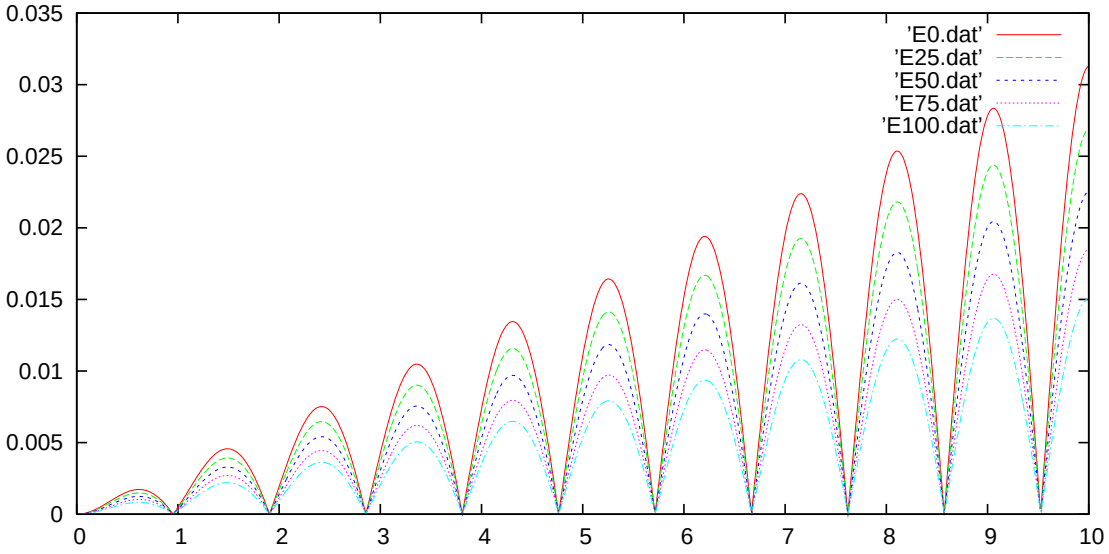


Figura 7.3: Erro absoluto do método de Newmark

Exemplo 2: Dados $\alpha = \beta = 1$ e $\lambda_2 = 1$, a função $u(x,t) = \cos(\pi t)\,\mathrm{sen}(\pi x)$ é solução do problema (7.33) co a forą $f(x,t) = \cos(\pi t)\,\mathrm{sen}(\pi x)$.

Para esse exemplo, os valores consignados na Tabela 7.7 são os erros nas normas $L^\infty(0,T;L^2(0,1))$ e $L^\infty(0,T;H^1(0,1))$, relativamente ao parâmetro θ do método de Newmark, com $\Delta t = 0.01$, $h = 0.05$ e $T = 0.5$. Observamos novamente, que ao

	θ	$E_{L^\infty(0,T;L^2(0,\ 1))}$	$E_{L^\infty(0,T;H^1(0,\ 1))}$
	0	0.001039	0.001203
$\Delta t = 0.01$	0.25	0.000906	0.001049
$h = 0.05$	0.5	0.000772	0.000894
$T = 0.5$	0.75	0.000652	0.000755
	1.0	0.000553	0.000640

Tabela 7.7: Variando θ

aumentar o valor de θ, o erro nas duas normas diminuem e os mesmos resultados do exemplo anterior se verificam.

7.4 Exercícios

1. **Bases Cúbicas e Hermite:** Considere o problema (7.30) como no Exemplo 2.

   (a) Determine a solução numérica para o Método da Diferença Central, usando como função base as funções B-splines, definidas em (3.8). Como as B-splines são polinômios de grau $k \leq 3$, podemos esperar que os erros sejam menores nas normas $L^\infty(0,T,L^2(0,1))$ e $L^\infty(0,T,H_0^1(0,1))$.

   (b) Nas mesmas condições do item anterior, refaça o problema usando como função base os polinômios de Hermite, definidos em (3.20) e (3.21).

   (c) Suponha que a solução exata do problema (7.30) seja desconhecida. Verifique a ordem de convergência numérica, definida em (3.55), no caso do Método da Diferença Central, usando as bases dos ítens anteriores, e verifique se satisfaz a estimativa de erro (9.39).

2. **Fronteira de Neumann:** Considere o mesmo problema (7.30), com $\alpha = 3$ e $\beta = \pi^2$. Sabendo que a solução exata é $u(x,t) = \operatorname{sen}(\pi x)\cos(2\pi t)$, o problema com as condições de fronteira do tipo Neumann é dado por

$$\begin{cases} u_{tt}(x,t) - 3u_{xx}(x,t) + \pi^2 u(x,t) = 0, & \forall\,(x,t) \in (0,1)\times[0,1], \\ u_x(0,t) = \pi\cos(2\pi t) = -u_x(1,t), & \forall t \in [0,1], \\ u(x,0) = \operatorname{sen}(\pi x), \quad u_t(x,0) = 0, & \forall x \in (0,1). \end{cases} \tag{7.34}$$

   Como observado anteriormente, para o problema ter solução única é necessário conhecer a solução em pelo menos um ponto, como por exemplo, $u(0,0) = 0$.

   (a) Verifique que o Método de Newmark, para $\theta \in \{0.25,\ 0.5,\ 1.0\}$, é incondicionalmente estável, mas o Método da Diferença Central ($\theta = 0$) é condicionalmente estável.

   (b) Com $\theta \in \{0,\ 0.25,\ 0.5,\ 1.0\}$, $h \in \{0.1,\ 0.01,\ 0.001\}$, e com o passo de tempo $\Delta t = 0.01$, faça uma tabela determinando os erros nas normas $L^2(0,1;L^2(0,1))$ entre a solução numérica e a solução exata.

3. **Problema Bidimensional:** Considere o problema homogêneo

$$\begin{cases} u_{tt}(x,y,t) - \Delta u(x,y,t) + u(x,y,t) = 0, & \forall\,(x,y,t) \in \Omega\times[0,1], \\ u(x,y,t) = 0, & \forall (x,y) \in \partial\Omega \quad \text{e} \quad \forall t \in [0,1], \\ u(x,y,0) = \operatorname{sen}(\pi x)\operatorname{sen}(\pi y), \quad u_t(x,y,0) = 0, & \forall x,y \in Q. \end{cases} \tag{7.35}$$

com $(x, y) \in \Omega = (0, 1) \times (0, 1)$ e a fronteira $\partial\Omega = \bigcup_{i=1}^{4} \Gamma_i$, onde

$$\begin{aligned}
\Gamma_1 &= \{(x, 0) \in \mathbb{R}^2 ;\ 0 \leq x \leq 1\}, \\
\Gamma_2 &= \{(1, y) \in \mathbb{R}^2 ;\ 0 \leq y \leq 1\}, \\
\Gamma_3 &= \{(x, 1) \in \mathbb{R}^2 ;\ 0 \leq x \leq 1\}, \\
\Gamma_4 &= \{(0, y) \in \mathbb{R}^2 ;\ 0 \leq y \leq 1\}.
\end{aligned}$$

Determine a solução numérica, usando o método de Newmark para $\theta = 0.5$, com a malha uniforme $h_1 = 0.05$ no eixo-x e $h_2 = 0.05$ no eixo-y e o passo de tempo $\Delta t = 0.001$. Sabendo que a solução exata do problema (7.35) é

$$u(x, y, t) = \operatorname{sen}(\pi x) \operatorname{sen}(\pi y) \cos\left(\sqrt{2 + 1/\pi^2}\ \pi t\right),$$

determine o erro nas normas $L^2(0, 1; L^2(\Omega))$ e $L^2(0, 1; H_0^1(\Omega))$.

CAPÍTULO 8

Análise Numérica da Equação do Calor

Nesse capítulo provaremos os resultados de convergência dos métodos numéricos e algoritmos para a Equação do Calor discutidos anteriormente tanto no contexto do problema semidiscreto quanto no caso discreto. Para maior aprofudamento, sugerimos ao leitor consultar, por exemplo, as referências [1, 9, 11, 17, 21].

8.1 Estimativas de Erro: Problema Semidiscreto

Nessa seção vamos estabelecer estimativas de erro para os problemas semidiscretos referentes à equação do calor. Mais precisamene, vamos considerar as estimativas de erros nos Espaços de Sobolev entre a solução exata $u(x,t)$ e a solução aproximada $u_h(x,t)$ de forma análoga à que foi feita na análise de erro do problema estacionário, mas sem a discretização da variável t.

Como vimos anteriormente, as funções bases w_i do subespaço V_m das soluções aproximadas são polinômios de grau k definidas em cada elemento finito Ω_e (para $k = 1, 2, 3$ temos respectivamente as funções bases lineares, quadráticas e splines cúbicas). Para considerar explicitamente a dependência do grau k do polinômio, escrevemos o espaço de elementos finitos V_m como

$$V_m^k(\Omega) := \{v_h \in V; v_h^e \in P_k(\Omega_e)\}, \tag{8.1}$$

onde v_h^e denota a restrição de v_h ao elemento e e P_k é o conjunto dos polinômios definidos em Ω_e, com grau menor ou igual a k na variável x.

O problema semidiscreto é formulado por: *determinar* $u_h : [0,T] \to V_m$, *solução do sistema*

$$\begin{cases} (u_h'(t) : v_h) + a(u_h(t), v_h) = (f(t) : v_h), \quad \forall\, v_h \in V_m^k, \\ (u_h(0) : v_h) = (u_{0h} : v_h). \end{cases} \tag{8.2}$$

A formulação variacional do problema contínuo, no caso de condições de fronteira do tipo Dirichlet (ver (6.1)), é dado por: *determinar* $u : [0,T] \to H_0^1(\Omega)$, *solução de*

$$\begin{cases} (u'(t) : v) + a(u(t), v) = (f(t) : v), \quad \forall\, v \in H_0^1(0,1), \\ (u(0) : v) = (u_0 : v), \quad \forall\, v \in H_0^1(0,1). \end{cases} \tag{8.3}$$

Tendo como objetivo estabelecer estimativa para o erro entre $u(t)$ e $u_h(t)$, onde $u_h(t)$ e $u(t)$ são respectivamente as soluções dos problemas (8.2) e (8.3) e lembrando que V_m^k é um subespaço do $H_0^1(\Omega)$, a igualdade (8.3) também é válida para $v_h \in V_m^k$. Assim, subtraindo as equações, obtemos

$$\begin{cases} (u'(t) - u_h'(t) : v_h) + a(u(t) - u_h(t), v_h) = 0, \quad \forall\, v_h \in V_m^k, \\ (u(0) - u_h(0) : v_h) = 0, \quad \forall\, v_h \in V_m^k. \end{cases} \tag{8.4}$$

Vale observar que, sendo a forma bilinear $a(\cdot,\cdot)$ contínua, simétrica e coerciva, ela define um produto interno em $V = H_0^1(\Omega)$. Assim, se u e u_h são respectivamente soluções dos problemas estacionários

$$a(u,v) = (f : v), \quad \forall v \in V \quad \text{e} \quad a(u_h, v_h) = (f : v_h),\ \forall v_h \in V_h,$$

temos a relação $a(u - u_h, v_h) = 0$, para todo $v_h \in V_h$, o que significa dizer que u_h é a projeção ortogonal de u sobre o subespaço V_h com relação ao produto interno definido pela forma bilinear $a(\cdot,\cdot)$. Entretanto, como é óbio verificar, isso não é verdade se $u(t)$ e $u_h(t)$ são respectivamente as solução dos problemas (8.2) e (8.3), isto é, não podemos afirmar que $a(u(t) - u_h(t), v_h) = 0$ para todo v_h e todo t. Em vista disso, consideremos:

Definição: *Sejam V um espaço de Hilbert, W um subespaço de V e $a : V \times V \longrightarrow \mathbb{R}$ uma forma bilinear contínua, simétrica e coerciva. Se $w(t) \in V$ para todo $t \in [0,T]$, definimos a projeção de Rayleigh-Ritz com respeito a forma bilinear $a(\cdot,\cdot)$ por*

$$\begin{array}{rccc} P_W : & V & \longrightarrow & W \\ & v(t) & \longmapsto & P_W v(t) \end{array}$$

tal que

$$a(v(t) - P_W v(t), w) = 0, \quad \forall w \in W. \tag{8.5}$$

É claro que a projeção de Rayleigh-Ritz coincide com a projeção usual no caso estacionário, isto é, se u é a solução de $a(u,v) = (f,v)$, $\forall v \in V$, temos $P_W u = P_h u = u_h$, onde $u_h \in V_h$ é a solução obtida pelo método de Galerkin. Como consequência direta do Lema de Céa (veja Teorema 3.1), a projeção de Rayleigh-Ritz satisfaz

$$\|u(t) - P_W u(t)\|_V \leq \gamma \|u(t) - w\|_V,\ \forall w \in W,$$

onde $\gamma = \sqrt{\delta}$, com $\delta > 0$ definido em (3.32) e $\|\cdot\|_V$ denota a norma de V associada à forma bilinear a.

No caso particular em que $W = V_m^k$, vamos manter a notação P_h, i.e., $P_{V_m^k} = P_h$, de modo que se $\widetilde{u}(t)$ é uma interpolação de $u(t)$ em V_m^k, obtemos

$$\|u(t) - P_h u(t)\|_V \leq \gamma \|u(t) - \widetilde{u}(t)\|_V. \tag{8.6}$$

Além disso, se $[\varphi_1, \varphi_2, \cdots \varphi_m]$ é um base ortonormal de V_m^k a projeção ortogonal sobre V_m^k satisfaz

$$P_h v = \sum_{i=1}^{m} a(\varphi_i, v)\varphi_i, \quad \forall v \in V. \tag{8.7}$$

Assim, para cada t fixo, podemos escrever a projeção de $u(t)$ na forma

$$P_h u(t) = \sum_{i=1}^{m} a(\varphi_i, u(t))\varphi_i = \sum_{i=1}^{m} c_i(t)\varphi_i. \tag{8.8}$$

Analogamente,

$$\widetilde{u}(x,t) = \sum_{i=1}^{m} u_i(t)\varphi_i. \tag{8.9}$$

Como as funções $\widetilde{u}(x,t)$ e $P_h u(x,t)$ não são idênticas, os coeficientes $c_i(t)$ e $u_i(t)$ definidos em (8.8) e (8.9) são diferentes. Entretanto, vale a estimativa de interpolação dada em (8.6), que será útil nas estimativas de erro.

A principal razão de introduzir a projeção ortogonal $P_h u(t)$ é porque podemos decompor o erro em duas partes na forma: $\|u - u_h\| \leq \|u - P_h u\| + \|P_h u - u_h\|$ e são conhecidas estimativas de erro do primeiro termo do lado direito, restando apenas estabelecer estimativas para o segundo termo no subespaço V_m^k. De fato, de acordo com o Corolário (3.4), temos para cada t fixado,

$$\|u(t) - P_h u(t)\|_m \leq \|u(t) - \widetilde{u}(t)\|_m \leq c h^{k+1-m} \|u(t)\|_{k+1}, \tag{8.10}$$

com $u(t) \in H^{k+1}(\Omega)$, $\widetilde{u}(t) \in V_m^k$ $(k \geq 1)$ e $m \leq k$.

Em [6], Douglas & Dupont provaram que valem as mesmas estimativas para as derivadas no tempo, como podemos ver no seguinte Lema, cuja demonstração pode ser vista em [6].

Lema 8.1. *Sejam $\{u(t), u'(t), u''(t)\} \subset H^{k+1}(\Omega)$, para todo $t \in [0,T]$ e $\widetilde{u}(t)$ o interpolador de $u(t)$ em V_m^k. Então, para cada $t \in [0,T]$, $u'(t)$ e $u''(t)$ satisfazem a mesma estimativa para o erro da interpolação de $u(t)$, ou seja*

$$\begin{aligned} \|u'(t) - \widetilde{u}'(t)\|_m &\leq c_1 h^{k+1-m} \|u'(t)\|_{k+1} \\ \|u''(t) - \widetilde{u}''(t)\|_m &\leq c_2 h^{k+1-m} \|u''(t)\|_{k+1} \end{aligned} \tag{8.11}$$

onde $m \leq k$, $k \geq 1$, c_1 *e* c_2 *são constantes positivas independentes de* $u(t)$ *e* h.

Consequentemente, dos resultados obtidos em (8.10) e (8.11), temos

$$\begin{aligned} &\|u'(t) - P_h u'(t)\|_m \leq \|u'(t) - \widetilde{u}'(t)\|_m \leq ch^{k+1-m}\|u'(t)\|_{k+1}, \quad \forall t \in [0,T], \\ &\|u''(t) - P_h u''(t)\|_m \leq \|u''(t) - \widetilde{u}''(t)\|_m \leq ch^{k+1-m}\|u''(t)\|_{k+1}, \quad \forall t \in [0,T] \end{aligned} \tag{8.12}$$

Com o auxílio de (8.11) e (8.12), podemos estabelecer as estimativas de erro entre a solução exata $u(t)$ e a solução aproximada $u_h(t)$, como veremos a seguir.

8.1.1 Estimativa na norma de $L^2(\Omega)$

Teorema 8.1. *Seja* $\{u, u'\} \subset L^\infty(0,T; H_0^1 \cap H^{k+1})$. *Então o erro entre a solução aproximada* $u_h(t)$ *e a solução exata* $u(t)$ *satisfaz*

$$\|u(t) - u_h(t)\|_0 \leq \widehat{C}h^{k+1}\|u(t)\|_{k+1}, \text{ qtp } t \in [0,T],$$

onde $k \geq 1$ *e* $\widehat{C}$ *é uma constante que depende de* T.

Demonstração: Considere a seguinte decomposição do erro:

$$e(t) := u(t) - u_h(t) = \Big(u(t) - P_h u(t)\Big) + \Big(P_h u(t) - u_h(t)\Big) =: \rho(t) + \xi(t),$$

onde $P_h u(t)$ é a projeção ortogonal de $u(t)$ em V_m^k.

Somando e subtraindo $P_h u(t)$ em (8.4), obtemos

$$(\rho'(t) + \xi'(t) : v_h) + a(\rho(t) + \xi(t), v_h) = 0, \quad \forall\, v_h \in V_m^k. \tag{8.13}$$

De (8.5), temos que $a(\rho(t), v_h) = 0$ e assim a equação (8.13) pode ser escrita na forma:

$$(\xi'(t) : v_h) + (\rho'(t) : v_h) + a(\xi(t), v_h) = 0, \quad \forall\, v_h \in V_m^k. \tag{8.14}$$

Em particular, podemos tomar $v_h = \xi(t) \in V_m^k$ em (8.14), obtendo-se

$$(\xi'(t) : \xi(t)) + (\rho'(t) : \xi(t)) + a(\xi(t), \xi(t)) = 0. \tag{8.15}$$

Temos também que,

$$\begin{cases} (\xi'(t) : \xi(t)) = \dfrac{1}{2}\dfrac{d}{dt}\|\xi(t)\|_0^2, \\ a(\xi(t), \xi(t)) = \alpha\|\xi(t)\|_1^2 + \beta\|\xi(t)\|_0^2 \geq 0. \end{cases} \tag{8.16}$$

Substituindo em (8.15) e desprezando o termo positivo $a(\xi(t),\xi(t))$, obtemos

$$\frac{1}{2}\frac{d}{dt}\|\xi(t)\|_0^2 \leq -(\rho'(t):\xi(t)).$$

Da desigualdade de Schwarz, temos $|(\rho'(t):\xi(t))| \leq \|\rho'(t)\|_0|\xi(t)\|_0$, de modo que

$$\frac{1}{2}\frac{d}{dt}\|\xi(t)\|_0^2 \leq \|\rho'(t)\|_0|\xi(t)\|_0. \tag{8.17}$$

Observemos de (8.16) que sendo a aplicação $t \mapsto \|\xi(t)\|_0^2$ diferenciável, $t \mapsto \|\xi(t)\|_0$ também diferenciável para os valores de t tais que $\|\xi(t)\|_0 \neq 0$, pois

$$\frac{d}{dt}\left(\int_0^1 \xi(x,t)^2\,dx\right)^{1/2} = \frac{1}{2}\left(\int_0^1 \xi(x,t)^2\,dx\right)^{-1/2}\frac{d}{dt}\int_0^1 \xi(x,t)^2\,dx,$$

de modo que

$$(\xi'(t),\xi(t)) = \frac{1}{2}\frac{d}{dt}\|\xi(t)\|_0^2 = \|\xi(t)\|_0\frac{d}{dt}\|\xi(t)\|_0. \tag{8.18}$$

Substituindo (8.18) em (8.17), temos após integrar 0 a t,

$$\|\xi(t)\|_0 \leq \|\xi(0)\|_0 + \int_0^t \|\rho'(\tau)\|_0\,d\tau \tag{8.19}$$

Para o primeira parcela do lado direito de (8.19),

$$\rho(0) = u(0) - u_h(0) - \xi(0) \quad \Rightarrow \quad \|\xi(0)\|_0 \leq \|u(0) - u_h(0)\|_0 + \|\rho(0)\|_0.$$

Lembrando que, para $m = 0$ em (8.10), vale a estimativa $\|\rho(0)\|_0 \leq ch^{k+1}\|u(0)\|_{k+1}$, $k \geq 1$, obtemos

$$\|\xi(0)\|_0 \leq ch^{k+1}\|u(0)\|_{k+1}, \qquad k \geq 1 \tag{8.20}$$

para a escolha de $u_h(0)$ tal que $\|u(0) - u_h(0)\|_0 \leq ch^{k+1}$

Por outro lado, usando (8.11) com $m = 0$, tem-se

$$\|\rho'(t)\|_0 = \|u'(t) - P_h u'(t)\|_0 \leq c_1\|u'(t) - \widetilde{u}'(t)\|_0 \leq c_2 h^{k+1}\|u'(t)\|_{k+1}. \tag{8.21}$$

Retornando a (8.19) e definindo $C = \max\{c, c_2\}$, obtemos

$$\|\xi(t)\|_0 \leq Ch^{k+1}\Big(\|u(0)\|_{k+1} + \int_0^t \|u'(s)\|_{k+1}\,ds\Big). \tag{8.22}$$

Por hipótese $u' \in L^\infty(0,T;H^{k+1})$. Logo,

$$\int_0^T \|u'(s)\|_{k+1}\,ds \leq T\|u'\|_{L^\infty(0,T;H^{k+1})}.$$

Portanto,

$$\|\xi(t)\|_0 \leq Ch^{k+1}\Big(\|u(0)\|_{k+1} + T\|u'\|_{L^\infty(0,T;H^{k+1})}\Big) \leq c_3(T)h^{k+1}. \tag{8.23}$$

Usando a desigualdade triangular, tem-se

$$\|u(t) - u_h(t)\|_0 \leq \|\rho(t)\|_0 + \|\xi(t)\|_0 \leq \widehat{C}(T)h^{k+1} \tag{8.24}$$

com o que conclui-se a demonstração do teorema. □

Note que por hipótese $u(t) \in H^{k+1}(\Omega)$ e $\xi(t) \in L^2(\Omega)$, para quase todo $t \in [0,T]$. Assim podemos tomar o supremo essencial em relaçaão a t para obter a seguinte estimativa:

$$\|u - u_h\|_{L^\infty(0,T;L^2(\Omega))} \leq \widehat{C}(T)h^{k+1}, \tag{8.25}$$

onde estamos enfatizando que a constante $\widehat{C}$ depende de T.

8.1.2 Estimativa na norma de $H_0^1(\Omega)$

Teorema 8.2. *Suponhamos $\{u, u'\} \subset L^\infty(0,T; H_0^1 \cap H^{k+1})$. Então o erro no instante $t \in [0,T]$ entre a solução aproximada $u_h(t)$ e a solução exata $u(t)$ satisfaz*

$$\|u(t) - u_h(t)\|_1 \leq \widehat{C}(T)h^k\|u(t)\|_{k+1},$$

onde $k \geq 1$.

Demonstração: Considere em (8.14), $v_h = \xi'(t) \in V_m^k$. Então,

$$\|\xi'(t)\|_0^2 + \frac{1}{2}\frac{d}{dt}\|\xi(t)\|_1^2 + (\rho'(t) : \xi'(t)) = 0.$$

Usando as desigualdades de Schwarz e Young, obtemos

$$(\rho'(t) : \xi'(t)) \leq \|\rho'(t)\|_0\|\xi'(t)\|_0 \leq \frac{1}{2}\Big(\|\rho'(t)\|_0^2 + \|\xi'(t)\|_0^2\Big).$$

Substituindo na identidade acima, segue

$$\|\xi'(t)\|_0^2 + \frac{d}{dt}\|\xi(t)\|_1^2 \leq \|\rho'(t)\|_0^2,$$

de onde se obtém após intergrar de 0 a t,

$$\int_0^t \|\xi'(\tau)\|_0^2\, d\tau + \|\xi(t)\|_1^2 - \|\xi(0)\|_1^2 \leq \int_0^t \|\rho'(t)\|_0^2\, d\tau,$$

de modo que

$$\|\xi(t)\|_1^2 \leq \|\xi(0)\|_1^2 + \int_0^t \|\rho'(\tau)\|_0^2 \, d\tau. \tag{8.26}$$

Usando o fato de que $\sqrt{a^2+b^2} \leq \sqrt{2}(a+b)$, quaisquer que sejam $a, b \geq 0$, temos

$$\|\xi(t)\|_1 \leq \sqrt{2} \left(\|\xi(0)\|_1 + \sqrt{\int_0^t \|\rho'(s)\|_0^2 \, ds} \right).$$

De forma análoga ao caso do $L^2(\Omega)$, temos na norma $H_0^1(\Omega)$,

$$\|\xi(0)\|_1 \leq \|u(0) - u_h(0)\|_1 + \|\rho(0)\|_1.$$

Tomando $m = 1$ em (8.10) temos a estimativa na norma $H_0^1(\Omega)$ dada por

$$\|\rho(0)\|_1 \leq ch^k \|u(0)\|_{k+1}, \ k \geq 1.$$

Ainda procedendo como no caso anterior com a escolha adequada de $u_h(0)$, obtemos

$$\|\xi(0)\|_1 = \|P_h u(0) - u_h(0)\|_1 \leq c_1 h^k \|u(0)\|_{k+1}, \quad k \geq 1.$$

Portanto, para $C = c + c_1$,

$$\|\xi(0)\|_1 \leq (ch^k + c_1 h^k)\|u(0)\|_{k+1} = Ch^k \|u(0)\|_{k+1}, \ k \geq 1.$$

Vale aqui observar a perda de precisão de ordem $\mathcal{O}(h)$ no presente caso quando comparamos com caso anterior.

Por outro lado, segue de (8.21), $\|\rho'(t)\|_0 \leq ch^{k+1}\|u'(t)\|_{k+1}$, de modo que

$$\|\xi(t)\|_1 \leq \sqrt{2} \left(Ch^k \|u(0)\|_{k+1} + ch^{k+1} \sqrt{\int_0^t \|u'(s)\|_0^2 \, ds} \right).$$

Por hipótese $u' \in L^\infty(0, T; H^{k+1})$. Logo,

$$\int_0^T \|u'(s)\|_{k+1}^2 \, ds \leq T \|u'\|_{L^\infty(0,T;H^{k+1})}^2.$$

Portanto, para $h < 1$,

$$\|\xi(t)\|_1 \leq Ch^k \Big(\|u(0)\|_{k+1} + \sqrt{T}\|u'\|_{L^\infty(0,T;H^{k+1})} \Big) \leq c_3(T) h^k. \tag{8.27}$$

Usando a desigualdade triangular, tem-se

$$\|u(t) - u_h(t)\|_1 \leq \|\rho(t)\|_1 + \|\xi(t)\|_1 \leq \widehat{C}(T)h^k, \tag{8.28}$$

com o que conclui-se a demonstração do teorema. □

Como anteriormente, podemos tomar o supremo essencial no tempo para obter a estimativa

$$\|u - u_h\|_{L^\infty(0,T;H^1(\Omega))} \leq \widehat{C}(T)h^k(\|u\|_{L^\infty(0,T;H^{k+1}(\Omega))}). \tag{8.29}$$

As estimativas (8.24), (8.25) e (8.28), (8.29) respectivamente nas normas L^2 e H_0^1 não são estimativas ótimas, em razão do último termo ser acumulativo, ou seja, o erro aumenta como função do tempo T. A seguir apresentamos uma estimativa ótima para a norma L^2.

8.1.3 Estimativa Ótima na Norma de $L^2(\Omega)$

Teorema 8.3. *Nas mesmas condições do Teorema (8.1) temos para $k \geq 1$,*

$$\|u(t) - u_h(t)\|_0 \leq \widehat{C}h^{k+1}\|u(t)\|_{k+1},$$

onde $\widehat{C}$ independe de T.

Demonstração: Seja $\{\lambda_i\}_{i\in\mathcal{N}}$ a sucessão de autovalores associados ao operador $\partial^2/\partial x^2$ e os correpondentes autovetores w_i no espaço $H_0^1(\Omega)$. Mais precisamente,

$$-\Delta w_i = \lambda_i w_i, \quad w_i \in H_0^1(\Omega).$$

Multiplicando ambos os lados, integrando em Ω e lembrando que $\lambda_1 < \lambda_2 \leq \lambda_3 \leq \ldots$, temos

$$\|\nabla w_i\|_0^2 = \lambda_i\|w_i\|_0^2 \geq \lambda_1\|w_i\|_0^2. \tag{8.30}$$

Observação: Lembremos que a sequência dos autovetores $\{w_i\}_{i\in\mathbb{N}}$ formam uma base ortonormal e completa em $H_0^1(\Omega)$.

Considere agora a equação (8.17). Usando a equivalência de normas em $H_0^1(\Omega)$ e a desigualdade (8.18), obtém-se

$$\frac{1}{2}\frac{d}{dt}\|\xi(t)\|_0^2 + \alpha\|\nabla\xi(t)\|_0^2 + \beta\|\xi(t)\|_0^2 \leq \|\rho'(t)\|_0\|\xi(t)\|_0. \tag{8.31}$$

Substituindo (8.30) em (8.31), temos a seguinte forma

$$\|\xi(t)\|_0\frac{d}{dt}\|\xi(t)\|_0 + (\alpha\lambda_1 + \beta)\|\xi(t)\|_0^2 \leq \|\rho'(t)\|_0\|\xi(t)\|_0, \tag{8.32}$$

que é equivalente a

$$\frac{d}{dt}\|\xi(t)\|_0 + \lambda\|\xi(t)\|_0 \leq \|\rho'(t)\|_0, \tag{8.33}$$

onde $\lambda = \alpha\lambda_1 + \beta$. Multiplicando (8.33) por $e^{\lambda t}$, obtemos

$$\frac{d}{dt}\Big(e^{\lambda t}\|\xi(t)\|_0\Big) \leq e^{\lambda t}\|\rho'(t)\|_0. \tag{8.34}$$

Integrando em relação $[0, t]$, obtemos

$$e^{\lambda t}\|\xi(t)\|_0 - \|\xi(0)\|_0 \leq \int_0^t e^{\lambda s}\|\rho'(s)\|_0\, ds,$$

que é equivalente a

$$\|\xi(t)\|_0 \leq e^{-\lambda t}\|\xi(0)\|_0 + \int_0^t e^{-\lambda(t-s)}\|\rho'(s)\|_0\, ds.$$

Usando as estimativas (8.20) e (8.21), obtemos

$$\|\xi(t)\|_0 \leq ch^{k+1}\Big(e^{-\lambda t}\|u(0)\|_{k+1} + \int_0^t e^{-\lambda(t-s)}\|u(s)\|_{k+1}\, ds\Big). \tag{8.35}$$

Temos agora uma estimativa ótima para o erro na norma $L^2(\Omega)$, pois a dependência das constantes em relação ao tempo T só aparecem no fator $\|u\|_{L^\infty(0,T;H^{k+a})}$. □

8.2 Estimativas de Erro: Problema Discreto

Nas estimativas anteriores para o problema semi-discreto, consideramos o tempo t não discretizado, o que não correspomde à prática, quando devemos discretizá-lo para o cálculo numérico das soluções. Em geral, nas equações de evolução do tipo parabólico ou hiperbólico, são usados o método de elementos finitos para a discretização das variáveis espaciais e o método das diferenças finitas para a variável temporal. A questão natural que se coloca é por que não utilizar também os elementos finitos na variável tempo. Matematicamente, é perfeitamente razoável aplicar o método de elementos finitos nas duas variáveis, o que tem sido usado por vários autores. Entretanto, estaremos acoplando o sistema ao aplicar o método de Galerkin, e isso destrói propriedades importantes de propagação (ver [20]). Sendo assim, utilizaremos nesta seção o Método das Diferenças Finitas como aproximação para a derivada no tempo e o método dos elementos finitos nas variáveis espaciais, para determinar a solução numérica do problema (8.3) e obter as estimativas de erro $\|u-u_h\|$ na normas em espaços de Sobolev.

8.2.1 Resultados Preliminares

No que segue, vamos considerar $\Omega = (0,1) \subset \mathbb{R}$. Para uma malha uniforme de $(0,1)$, com $h = x_{i+1} - x_i$, $i = 1, 2, \ldots, N$, definimos a norma discreta em $L^2(0,1)$ e $H^1(0,1)$ respectivamene por

$$\|w\|_{L^2(0,1)} := \Big(h \sum_{i=1}^{m} |w(x_i)|^2\Big)^{1/2},$$

$$\|w\|_{H^1(\Omega)} := \Big(h \sum_{i=1}^{m} |w(x_i)|^2\Big)^{1/2} + \Big(h \sum_{i=1}^{m} \Big|\frac{\partial w(x_i)}{\partial x}\Big|^2\Big)^{1/2}.$$

Por outro lado, se w é uma função na variável t, denotaremos $w^n := w(t_n)$, onde $t_n = n\Delta t$. Podemos assim introduzir:

$$\begin{aligned}
w^{n+1/2} &:= (w^{n+1} + w^n)/2,\\
w^{n+\theta} &:= \theta w^{n+1} + (1-\theta) w^n,\\
\star_\theta\, w^n &:= \theta w^{n+1} + (1-2\theta) w^n + \theta w^{n-1},\\
\delta w^{n+\frac{1}{2}} &:= (w^{n+1} - w^n)/\Delta t,\\
\delta w^{n-\frac{1}{2}} &:= (w^n - w^{n-1})/\Delta t,\\
\delta w^n &:= w^{n+\frac{1}{2}} - w^{n-1/2})/\Delta t = (w^{n+1} - w^{n-1})/2\Delta t,\\
\delta^2 w^n &:= (w^{n+1} - 2w^n + w^{n-1})/(\Delta t)^2,
\end{aligned} \tag{8.36}$$

onde $\theta \in [0,1]$ denota o parâmetro de Newmark. Por simplicidade, denotaremos também $w^n_\theta := \star_\theta\, w^n$.

Com base na notação introduzida em (8.36), se H é um espaço de Hilbert e $w : [0,T] \to H$, definimos as "normas discretas" para w pelas seguintes expressões:

$$\begin{aligned}
\|w\|_{L^\infty_0(0,T;H)} &:= \max\{\|w^n\|_H,\ n = 0, \ldots, N-1\}\\
\|w\|_{L^\infty_{1/2}(0,T;H)} &:= \max\{\|w^{n+\frac{1}{2}}\|_H,\ n = 0, \ldots, N-1\},\\
\|w\|^2_{L^2_0(0,T;H)} &= \Delta t \sum_{n=0}^{N-1} \|w^n\|^2_H,\\
\|w\|^2_{L^2_{1/2}(0,T;H)} &:= \Delta t \sum_{n=0}^{N-1} \|w^{n+\frac{1}{2}}\|^2_H.
\end{aligned} \tag{8.37}$$

Combinando apropriadamente a notação introduzida em (8.36), podemos definir várias normas, como por exemplo

$$\|\delta w\|^2_{L^2_{1/2}(0,T;W)} := \sum_{n=0}^{N-1} \|\delta w^{n+\frac{1}{2}}\|^2_W \tag{8.38}$$

Para estabelecer estimativas de erro no tempo discreto, aplicaremos os seguintes Lemas:

Lema 8.2. *Sejam φ e ψ funções não negativas definidas em $\{t_n, \ ; n = 0, \cdots, N\}$, com ψ não decrescente. Se*

$$\varphi^\tau \leq \psi^\tau + c\Delta t \sum_{n=0}^{\tau-1} \varphi^n, \quad \forall \tau = 0, \ldots, N \ , \tag{8.39}$$

então

$$\varphi^\tau \leq \psi^\tau \exp(ct_\tau), \quad \forall \tau = 0, \ldots, N; \tag{8.40}$$

onde c é uma constante positiva independente de τ.

Note que o Lema anterior, conhecido como Lema de Belman, é o análogo discreto do Lema de Gronwall.

Lema 8.3. (Dupont)*Sejam $w : [0, T] \to L^2(0, 1)$ e $w^n = w(t_n)$, $n = 0, \ldots, N$. Se o incremento no tempo $\Delta t < 1$, então*

$$\begin{aligned} \|w^{n-1/2}\|_0^2 &\leq c\Big(\|w^{1/2}\|_0^2 + \|\delta w\|^2_{L_0^2(0,T;L^2(0,1))} \Big), \\ \|w^n\|_0^2 &\leq \overline{c}\Big(\|w^0\|_0^2 + \|\delta w\|^2_{L_0^2(0,T;L^2(0,1))} \Big), \end{aligned} \tag{8.41}$$

onde c e $\overline{c}$ são constantes positivas independentes de w.

Os resultados da teoria de aproximação que seguem serão aplicados para estabelecer estimativas de erro no tempo discreto. Nesses resultados, p pode assumir os valores 2 ou ∞, e H denota qualquer um dos espaços de Hilbert, $H_0^1(0, 1)$, $H^2(0, 1)$ ou $L^2(0, 1)$.

(i) O operador δ, como definido em $(8.36)_6$, é uma aproximação para a primeira derivada no tempo e vale a seguinte estimativa relacionando δw com w':

$$\|\delta w\|_{L_0^p(0,T;H)} \leq c\|w'\|_{L^p(0,T;H)}. \tag{8.42}$$

(ii) De forma análoga, o operador δ^2, como definido em $(8.36)_7$, é uma aproximação para a segunda derivada no tempo, e tem-se a seguinte estimativa relacinando $\delta^2 w$ com w'':

$$\|\delta^2 w\|_{L_0^p(0,T;H)} \leq c\|w''\|_{L^p(0,\,T;H)}. \tag{8.43}$$

(iii) O operador $\star_\theta$, definido em $(8.36)_3$, com pesos θ, $(1-2\theta)$ e θ em nós consecutivos, pode ser estimada em função de w por

$$\| \star_\theta w\|_{L_0^p(0,T;H)} \leq c\|w\|_{L^p(0,T;H)}. \tag{8.44}$$

(iv) Usando (8.42) e (8.44) podemos compor os operadores δ e $\star_\theta$ e estimar a norma de $(\delta \circ \star_\theta)\, w$ em função de w'.

$$\|(\delta \circ \star_\theta)\, w\|_{L_0^p(0,T;H)} \leq c\|w'\|_{L^p(0,\,T;H)}. \tag{8.45}$$

Note que, ao aplicar as normas $(8.37)_{1,3}$ em $\star_\theta w$, faz-se necessário introduzir o valor de $\star_\theta w$ em $t = 0$. Por simplicidade, toma-se $\star_\theta w^0 = w^0$.

(v) (Somatório por partes) Finalizando os resultados preliminares, consideremos as seguintes fórmulas de somatório por partes, que são análogos discretos da integração por partes:

$$\begin{aligned} \Delta t \sum_{n=1}^{N_1-1} \varphi^n \delta\psi^{n-1/2} &= \varphi^{N_1-1}\psi^{N_1-1} - \varphi^1\psi^0 - \Delta t \sum_{n=1}^{N_1-2} \psi^n \delta\varphi^{n+\frac{1}{2}}, \\ \Delta t \sum_{n=1}^{N_1-1} \varphi^n \delta\psi^{n+\frac{1}{2}} &= \varphi^{N_1-1}\psi^{N_1} - \varphi^1\psi^1 - \Delta t \sum_{n=2}^{N_1-1} \psi^n \delta\varphi^{n-1/2}, \end{aligned} \tag{8.46}$$

onde $2 \leq N_1 \leq N$.

Para finalizar os preliminares, lembremos:

Teorema de Taylor: *Seja $f \in C^{n+1}(a,b)$, $n \geq 0$. Então f pode ser escrito na forma*

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n + R_n, \tag{8.47}$$

onde o resto R_n pode ser expresso por

$$R_n = \int_a^x \frac{f^{(n+1)}(s)}{n!}(x-s)^{(n)}\, ds \quad \text{ou} \quad R_n = \frac{f^{(n+1)}(\vartheta)}{(n+1)!}(x-a)^{(n+1)},$$

para algum $\vartheta \in [a,x]$.

• Decomposição do Erro

Nas estimativas de erro que seguem, considamos a forma bilinear $a(\cdot,\cdot)$ definida em (6.3) por

$$a(u,v) = \alpha \int_0^1 u_x v_x\, dx + \beta \int_0^1 uv\, dx. \tag{8.48}$$

Considere a decomposição do erro na forma

$$e = u - u_h = \Big(u - P_h u\Big) + \Big(P_h u - u_h\Big) = \rho + \xi, \tag{8.49}$$

onde $P_h u$ é a projeção ortogonal de u em V_m^k com relação a forma bilinear $a(\cdot,\cdot)$, ou seja, a projeção ortogonal da solução exata u sobre V_m^k com respeito ao produto escalar definido pela forma bilinear

$$a(u(t) - P_h u(t), v_h) = 0, \quad \forall t \in [0,T], \quad \forall v_h \in V_m^k. \tag{8.50}$$

No Teorema (10.1) do Capítulo 10, apresentamos resultados de existência e unicidade de solução para o Problema (6.1). Entretanto, para obter as estimativas de erro no tempo discreto, faz-se necessário supor mais regularidade da solução, que admitiremos sob a seguinte hipótese:

HC: Suponha que a força f dada seja tal que a solução u do Problema(6.1) tenham a seguinte regularidade:

$$\left| \begin{array}{l} f \in L^2(0,\ T; H^{k+1}(0,1)) \quad \text{e} \quad u \in L^\infty(0,T; H_0^1(0,1) \cap H^{k+1}(0,1)), \\ \text{com as derivadas } u_t,\ u_{tt},\ u_{ttt} \in L^\infty(0,T; H^{k+1}(0,1)). \end{array} \right. \tag{8.51}$$

8.2.2 Método de Euler Regressivo

Do sistema de equações diferenciais ordinárias (6.13), deduzimos o método iterativo (6.16) dado por

$$(A + \Delta t B)\boldsymbol{d}^n = A\boldsymbol{d}^{n-1} + \Delta t \boldsymbol{F}^n = \boldsymbol{b}^n, n = 1, 2, \ldots, N.$$

No próximo teorema mostraremos que o Método de Euler regressivo é incondicionalmente estável, ou seja, é estável independentemente da relação entre o Δt e h. Além disso, mostraremos a ordem de convergência do método, isto é, as estimativas de erro em normas de espaços de Sobolev.

Teorema 8.4. *Sob a hipótese (8.51), se o dado inicial $u_0 \in H_0^1(0,1) \cap H^{k+1}(0,1)$, então a estimativa de erro para o problema discreto obtida pelo Método de Euler regressivo (6.16), é dado por:*

$$\begin{aligned} \|e\|_{L_0^\infty(0,T;L^2(0,1))} &:= \|u - u_h\|_{L_0^\infty(0,T;L^2(0,1))} \leq c_1 \left(h^{k+1} + \Delta t\right) \\ \|e\|_{L_0^\infty(0,T;H_0^1(0,1))} &:= \|u - u_h\|_{L^\infty(0,T;H_0^1(0,1))} \leq c_2 \left(h^k + \Delta t\right) \end{aligned} \tag{8.52}$$

,

onde c_1 e c_2 são constantes positivas, independentes de h e Δt.

Demonstração: Seja e^n o erro entre $u_h^n = u_h(t_n)$ e $u^n = u(t_n)$, onde u é a solução exata e u_h é a solução aproximada, isto é,

$$e^n = u^n - u_h^n, \quad n = 0, 1, \cdots N.$$

Como anteriormente, introduzimos a projeção ortogonal $P_h u^n$, $n = 0, 1, \ldots, N$ em V_m^k satisfazendo a condição de ortogonalidade (8.50),

$$a(u^n - P_h u^n, v) = 0, \quad \forall v \in V_m^k \quad \text{e para todo} \quad n = 0, 1, \ldots, N. \tag{8.53}$$

Podemos decompor o erro na forma

$$e^n = u^n - u_h^n = (u^n - P_h u^n) + (P_h u^n - u_h^n) = \rho^n + \xi^n \tag{8.54}$$

Sendo as estimativas de erro para o termo ρ^n conhecidas (veja (8.11)), é suficiente estimar o termo ξ^n para $n = 0, 1, \ldots, N$.

A formulação fraca do problema (8.3) calculada no tempo discreto t_n é dada por:

$$(u'(t_n) : v) + a(u^n, v) = (f^n : v), \quad \forall\, v \in H_0^1(0, 1).$$

Para simplificar a notação, vamos denotar $(u')^n = u'(t_n)$, de modo que a equação acima toma forma

$$((u')^n : v) + a(u^n, v) = (f^n : v), \quad \forall\, v \in H_0^1(0, 1). \tag{8.55}$$

Então, somando e subtraindo o termo $(\delta u^{n-\frac{1}{2}}, v)$ na equação acima, obtemos

$$(\delta u^{n-\frac{1}{2}} : v) + a(u^n, v) = (f^n : v) + (\delta u^{n-\frac{1}{2}} - (u')^n : v). \tag{8.56}$$

Consideremos agora o equivalente aproximado (regressivo) de (8.55), isto é,

$$(\delta u_h^{n-\frac{1}{2}} : v_h) + a(u_h^n, v_h) = (f^n : v_h), \quad \forall v_h \in V_m^k, \;\; n = 1, 2, \ldots, N. \tag{8.57}$$

Tomando em particular $v = v_h \in V_m^k$ em (8.56) e fazendo a diferença com (8.57), obtemos

$$(\delta u^{n-\frac{1}{2}} - \delta u_h^{n-\frac{1}{2}} : v_h) + a(u^n - u_h^n, v_h) = (\delta u^{n-\frac{1}{2}} - (u')^n : v_h) \tag{8.58}$$

Somando e subtraindo o termo $\delta P_h u^{n-\frac{1}{2}}$ e $P_h u^n \in V_m^k$ no primeiro e segundo termo respectivamente e usando a decomposição do erro (8.54), obtemos

$$(\delta \xi^{n-\frac{1}{2}} : v_h) + a(\xi^n, v_h) = (\delta u^{n-\frac{1}{2}} - (u')^n : v_h) - (\delta \rho^{n-\frac{1}{2}} : v_h), \quad \forall v_h \in V_m^k. \tag{8.59}$$

Note que de (8.53), tem-se $a(\rho^n, v_h) = 0$.

Tomando em particular $v_h = \xi^n$ em (8.59), obtemos

$$(\delta \xi^{n-\frac{1}{2}} : \xi^n) + a(\xi^n, \xi^n) = -(\delta \rho^{n-\frac{1}{2}} : \xi^n) - ((u')^n - \delta u^{n-\frac{1}{2}} : \xi^n) \tag{8.60}$$

Além disso, observando que

$$\begin{aligned} (\delta \xi^{n-\frac{1}{2}} : \xi^n) &= \left(\frac{\xi^n - \xi^{n-1}}{\Delta t} : \xi^n \right) = \frac{\|\xi^n\|_0^2 - (\xi^n : \xi^{n-\frac{1}{2}})}{\Delta t} \\ &\geq \frac{\|\xi^n\|_0^2 - \|\xi^n\|_0 \|\xi^{n-\frac{1}{2}}\|_0}{\Delta t} = \frac{1}{\Delta t}(\|\xi^n\|_0 - \|\xi^{n-\frac{1}{2}}\|_0)\|\xi^n\|_0 \end{aligned} \tag{8.61}$$

e

$$-\Big((\delta\rho^{n-\frac{1}{2}}:\xi^n)+((u')^n-\delta u^{n-\frac{1}{2}}:\xi^n)\Big)\leq\Big(\|\delta\rho^{n-\frac{1}{2}}\|_0+\|(u')^n-\delta u^{n-\frac{1}{2}}\|_0\Big)\|\xi^n\|_0,$$

obtemos, após substituir os dois últimos termos em (8.60),

$$\frac{1}{\Delta t}\left(\|\xi^n\|_0-\|\xi^{n-1}\|_0\right)\|\xi^n\|_0+a(\xi^n,\xi^n)\leq\Big(\|\delta\rho^{n-\frac{1}{2}}\|_0+\|(u')^n-\delta u^{n-\frac{1}{2}}\|_0\Big)\|\xi^n\|_0. \quad (8.62)$$

Visto que $a(\xi^n,\xi^n)=\alpha\|\xi^n\|_1^2+\beta\|\xi^n\|_0^2\geq 0$, obtemos após multiplicar a equação (8.62) por Δt e eliminar o termo $\|\xi^n\|_0$,

$$\|\xi^n\|_0\leq\|\xi^{n-1}\|_0+\Delta t\Big(\|\delta\rho^{n-\frac{1}{2}}\|_0+\|(u')^n-\delta u^{n-\frac{1}{2}}\|_0\Big). \quad (8.63)$$

Fazendo o somatório para $n=1,2,\ldots,N$ e as necessárias simplificações, obtemos

$$\|\xi^N\|_0\leq\|\xi^0\|_0+\Delta t\Big(\sum_{n=1}^{N}\|\delta\rho^{n-\frac{1}{2}}\|_0+\sum_{n=1}^{N}\|(u')^n-\delta u^{n-\frac{1}{2}}\|_0\Big). \quad (8.64)$$

Note que

$$\xi^0=\widetilde{u}^0-u_h^0=\widetilde{u}^0-u^0+u^0-u_h^0=-\rho^0+u^0-u_h^0.$$

Logo,

$$\|\xi^0\|_0\leq\|\rho^0\|_0+\|u^0-u_h^0\|_0\leq ch^{k+1}\|u_0\|_{k+1}+\|u^0-u_h^0\|_0.$$

Como a função $u^0=u(0)$ é conhecida, podemos tomar $u_h(0)$ igual a $u(0)$, ou se necessário, satisfazendo a estimativa:

$$\|u^0-u_h^0\|_0\leq ch^{k+1}\|u_0\|_{k+1},\qquad k\geq 1.$$

Assim, temos

$$\|\xi^N\|_0\leq ch^{k+1}\|u_0\|_{k+1}+\Delta t\Big(\sum_{n=1}^{N-1}\|\delta\rho^{n-\frac{1}{2}}\|_0+\sum_{n=1}^{N-1}\|(u')^n-\delta u^{n-\frac{1}{2}}\|_0\Big). \quad (8.65)$$

Vamos agora analisar os dois termos com somatório em (8.65). Note que

$$\delta\rho^{n-\frac{1}{2}}=\frac{1}{\Delta t}(\rho^n-\rho^{n-1})=\frac{1}{\Delta t}\int_{t_{n-1}}^{t_n}\rho'(s)\,ds.$$

Somando $n=1,2,\ldots,N$, multiplicando por Δt e usando usando (8.11), obtemos

$$\Delta t\sum_{n=1}^{N}\|\delta\rho^{n-\frac{1}{2}}\|_0\leq\int_{t_0}^{t_N}\|\rho'(s)\|_0\,ds\leq ch^{k+1}\int_{t_0}^{t_N}\|u'(s)\|_{k+1}\,ds. \quad (8.66)$$

Para o segundo termo com somatório, temos

$$(u')^n - \delta u^{n-\frac{1}{2}} = (u')^n - \frac{1}{\Delta t}(u^n - u^{n-1}).$$

Pela expansão de Taylor, temos

$$u(t_n) = u(t_{n-1}) + \Delta t u'(t_{n-1}) + \int_{t_{n-1}}^{t_n} (t_n - s)u''(s)\, ds,$$
$$u(t_{n-1}) = u(t_n) - \Delta t u'(t_n) + \int_{t_n}^{t_{n-1}} (t_{n-1} - s)u''(s)\, ds,$$

que podemos escrever de forma equivalente,

$$\begin{aligned} \delta u^{n-\frac{1}{2}} &= \frac{1}{\Delta t}\Big(u^n - u^{n-1}\Big) = (u')^{n-1} + \frac{1}{\Delta t}\int_{t_{n-1}}^{t_n} (t_n - s)u''(s)\, ds, \\ \delta u^{n-\frac{1}{2}} &= \frac{1}{\Delta t}\Big(u^n - u^{n-1}\Big) = (u')^n + \frac{1}{\Delta t}\int_{t_{n-1}}^{t_n} (t_{n-1} - s)u''(s)\, ds. \end{aligned} \tag{8.67}$$

Integrando por partes, obtemos

$$\begin{aligned} \frac{1}{\Delta t}\int_{t_{n-1}}^{t_n} (t_{n-1} - s)u''(s)\, ds &= \frac{1}{\Delta t}\Big((t_{n-1} - s)u'(s)|_{t_{n-1}}^{t_n} + \int_{t_{n-1}}^{t_n} u'(s)\, ds\Big) \\ &= -(u')^n + \frac{1}{\Delta t}\Big(u^n - u^{n-1}\Big) = -(u')^n + \delta u^{n-\frac{1}{2}}. \end{aligned}$$

Por outro lado, como a função $f(s) = (t_{n-1} - s)$ não muda de sinal no intervalo $[t_{n-1}, t_n]$, temos pelo Teorema do Valor Médio para integrais, a existência de $\zeta \in [t_{n-1}, t_n]$, tal que

$$\int_{t_{n-1}}^{t_n} (t_{n-1} - s)u''(s)\, ds = u''(\zeta)\int_{t_{n-1}}^{t_n} (t_{n-1} - s)\, ds = u''(\zeta)\frac{(\Delta t)^2}{2}. \tag{8.68}$$

Dessa forma, segue que

$$\begin{aligned} \|(u')^n - \delta u^{n-\frac{1}{2}}\|_0 = \frac{\Delta t}{2}\|u''(\zeta)\|_0 &\le \frac{\Delta t}{2}\max_{\tau\in[t_{n-1},t_n]}\|u''(\tau)\|_0 \\ &\le \frac{\Delta t}{2}\|u''\|_{L^\infty(t_{n-1},t_n;L^2(0,1))}. \end{aligned}$$

Fazendo a soma para $n = 1, 2, \ldots, N$, obtemos

$$\sum_{n=1}^{N}\|(u')^n - \delta u^{n-\frac{1}{2}}\|_0 \le \frac{\Delta t}{2}\sum_{n=1}^{N}\|u''\|_{L^\infty(t_{n-1},t_n L^2(0,1))} \le C\|u''\|_{L^\infty(0,T;L^2(0,1))}. \tag{8.69}$$

Substituindo (8.66) e (8.69) em (8.65), obtemos

$$\|\xi^N\|_0 \leq ch^{k+1}\|u_0\|_{k+1} + \hat{C}\Big\{h^{k+1}\|u'\|_{L^\infty(0,T;H^{k+1})} + \Delta t\|u''\|_{L^\infty(0,T;L^2(0,1))}\Big\}. \tag{8.70}$$

Usando a hipótese (8.51), conclui-se que

$$\|\xi^N\|_0 = \|P_h(u)^N - u_h^N\|_0 \leq \|\widetilde{u}^N - u_h^N\|_0 \leq c_1 h^{k+1} + c_2\Delta t = \mathcal{O}(h^{k+1} + \Delta t). \tag{8.71}$$

Por outro lado, fazendo $m = 0$ em (8.11),

$$\|\rho^N\|_0 = \|u(t) - P_h u(t)\|_0 \leq \|u(t) - \widetilde{u}(t)\|_0 \leq ch^{k+1}\|u(t)\|_{k+1}, \text{ para t fixo em } [0,T],$$

Então, usando a decomposição do erro (8.49), obtemos a seguinte estimativa,

$$\|e^N\|_0 = \|u^N - P_h u^N\|_0 + \|P_h u^N - u_h^N\|_0 = \|\rho^N\|_0 + \|\xi^N\|_0 = \mathcal{O}(h^{k+1} + \Delta t), \tag{8.72}$$

o que conclui a estimativa na norma $L^\infty(0,T;L^2(0,1))$, ou seja,

$$\|e\|_{L^\infty(0,T;L^2(0,1))} = \|u - u_h\|_{L^\infty(0,T;L^2(0,1))} \leq c_2\left(h^{k+1} + \Delta t\right).$$

De forma análoga obtém-se a estimativa $L_0^\infty(0,T;H_0^1(0,1)$. Observe que, por causa da relação

$$\|\rho^N\|_1 = \|u(t) - P_h u(t)\|_1 \leq ch^k\|u(t)\|_{k+1},$$

tem-se uma perda de precisão de ordem h e concluímos assim a prova do teorema. □

8.2.3 Método de Crank-Nicolson

Teorema 8.5. *Sob a hipótese (8.51), e os dados iniciais $u_0 \in H_0^1(0,1) \cap H^{k+1}(0,1)$, a solução numérica obtida pelo método de Crank-Nicolson (6.19) tem a seguinte estimativa de erro:*

$$\begin{aligned} \|e\|_{L_0^\infty(0,T;L^2(0,1))} &= \|u - u_h\|_{L^\infty(0,T;L^2(0,1))} \leq c_2\left(h^{k+1} + (\Delta t)^2\right), \\ \|e\|_{L_0^\infty(0,T;H_0^1(0,1))} &= \|u - u_h\|_{L^\infty(0,T;H_0^1(0,1))} \leq c_1\left(h^{k} + (\Delta t)^2\right), \end{aligned} \tag{8.73}$$

onde c_1 e c_2 são constantes positivas independentes de h e Δt.

Demonstração: A formulação fraca do problema é dada por

$$(u'(t) : v) + a(u(t), v) = (f(t) : v), \quad \forall v \in H_0^1(0,1). \tag{8.74}$$

Fazendo $t_n = (n + \frac{1}{2})\Delta t$ em (8.74) e considerando $u^{n+\frac{1}{2}}$ e $f^{n+\frac{1}{2}}$ respectivamente como aproximações de $u(t_n)$ e $f(t_n)$, a equação acima toma a forma

$$(u'(t_n) : v) + a(u^{n+\frac{1}{2}}, v) = (f^{n+\frac{1}{2}} : v), \quad \forall v \in H_0^1(0,1). \tag{8.75}$$

Somando e subtraindo o termo $(\delta u^{n+\frac{1}{2}}, v)$ em (8.75), obtemos

$$(\delta u^{n+\frac{1}{2}}, v) + a(u^{n+\frac{1}{2}}, v) = (f^{n+\frac{1}{2}} + g^n, v), \quad \forall v \in H_0^1(0,1) \tag{8.76}$$

onde $g^n = \delta u^{n+\frac{1}{2}} - u'(t_n)$ e sendo $\delta u^{n+\frac{1}{2}} = \frac{1}{\Delta t}(u^{n+1} - u^n)$ uma aproximação para $u'(t_n)$ com erro de $\mathcal{O}(\Delta t)$.

Por outro lado, para o problema aproximado, a formulação é

$$(\delta u_h^{n+\frac{1}{2}} : v_h) + a(u_h^{n+\frac{1}{2}}, v_h) = (f^{n+\frac{1}{2}} : v_h), \quad \forall v_h \in V_m^k. \tag{8.77}$$

Tomando em particular $v = v_h \in V_m^k$ em (8.76) e subtraindo-a de (8.77), obtemos

$$(\delta e^{n+\frac{1}{2}} : v_h) + a(e^{n+\frac{1}{2}}, v_h) = (g^n : v_h), \quad \forall v_h \in V_m^k. \tag{8.78}$$

Somando e subtraindo $\delta P_h u^{n+\frac{1}{2}}$ e $P_h u^{n+\frac{1}{2}} \in V_m^k$ respectivamente no primeiro e segundo termos da equação acima e usando a decomposição do erro (8.54), obtém-se

$$(\delta \xi^{n+\frac{1}{2}} : v_h) + (\delta \rho^{n+\frac{1}{2}} : v_h) + a(\xi^{n+\frac{1}{2}}, v_h) = (g^n : v_h), \quad \forall v_h \in V_m^k, \tag{8.79}$$

onde usamos a ortogonalidade da projeção em V_m^k, ou seja,

$$a(\rho^{n+\frac{1}{2}}, v_h) = \frac{1}{2}\, a(\rho^{n+1} + \rho^n, v_h) = 0.$$

Assim, para $v_h = \xi^{n+\frac{1}{2}}$ em (8.79), obtemos

$$(\delta \xi^{n+\frac{1}{2}} : \xi^{n+\frac{1}{2}}) + a(\xi^{n+\frac{1}{2}}, \xi^{n+\frac{1}{2}}) = (g^n - \delta \rho^{n+\frac{1}{2}} : \xi^{n+\frac{1}{2}}). \tag{8.80}$$

Tendo em vista que

$$\begin{aligned}
(\delta \xi^{n+\frac{1}{2}} : \xi^{n+\frac{1}{2}}) &= \frac{1}{2\Delta t}(\xi^{n+1} - \xi^n : \xi^{n+1} + \xi^n) = \frac{1}{2\Delta t}\Big(\|\xi^{n+1}\|_0^2 - \|\xi^n\|_0^2\Big) \\
&= \frac{1}{2\Delta t}\Big(\|\xi^{n+1}\|_0 - \|\xi^n\|_0\Big)\Big(\|\xi^{n+1}\|_0 + \|\xi^n\|_0\Big), \\
a(\xi^{n+\frac{1}{2}}, \xi^{n+\frac{1}{2}}) &= \alpha \|\xi^{n+\frac{1}{2}}\|_1^2 + \beta \|\xi^{n+\frac{1}{2}}\|_0^2 \geq 0,
\end{aligned} \tag{8.81}$$

segue de (8.80),

$$\frac{1}{2\Delta t}\Big(\|\xi^{n+1}\|_0 - \|\xi^n\|_0\Big)\Big(\|\xi^{n+1}\|_0 + \|\xi^n\|_0\Big) \leq (g^n - \delta \rho^{n+\frac{1}{2}} : \xi^{n+\frac{1}{2}}).$$

Aplicando a desigualdade de Schwarz no lado direito da equação acima, obtemos

$$\begin{aligned}
\Big(\|\xi^{n+1}\|_0 - \|\xi^n\|_0\Big)\Big(\|\xi^{n+1}\|_0 + \|\xi^n\|_0\Big) &\leq 2\Delta t\Big(\|g^n\|_0 + \|\delta \rho^{n+\frac{1}{2}}\|_0\Big)\|\xi^{n+\frac{1}{2}}\|_0 \\
&\leq 2\Delta t\Big(\|g^n\|_0 + \|\delta \rho^{n+\frac{1}{2}}\|_0\Big)\Big(\|\xi^{n+1}\|_0 + \|\xi^n\|_0\Big),
\end{aligned}$$

de onde se deduz que

$$Vert\xi^{n+1}\|_0 - \|\xi^n\|_0 \leq \Delta t\Big(\|g^n\|_0 + \|\delta\rho^{n+\frac{1}{2}}\|_0\Big).$$

Assim, somando de $n = 0$ a $n = N-1$, temos

$$\|\xi^N\|_0 \leq \|\xi^0\|_0 + \Delta t\Big(\sum_{n=0}^{N-1}\|\delta\rho^{n+\frac{1}{2}}\|_0 + \sum_{n=0}^{N-1}\|g^n\|_0\Big). \tag{8.82}$$

Como estamos estimando na norma $L^2(0,1)$, devemos tomar $m = 0$ em (8.12), obtendo-se

$$\|\xi^0\|_0 \leq c_1 h^{(k+1)}\|u_0\|_{k+1}, \tag{8.83}$$

$$\begin{aligned}\Delta t\sum_{n=0}^{N-1}\|\delta\rho^{n+\frac{1}{2}}\|_0 \;&\leq \int_{t_0}^{t_N}\|\rho'(s)\|_0\,ds \leq c_2 h^{(k+1)}\int_{t_0}^{t_N}\|u'(s)\|_{k+1}\,ds \\ &= c_2 h^{(k+1)}\|u'\|_{L^1(0,T;H^{k+1})}.\end{aligned} \tag{8.84}$$

Por outro lado, como $t_n = (n+\frac{1}{2})\Delta t$, consideremos a aproximação de ordem $\mathcal{O}(\Delta t)$

$$u'(t_n) \approx \frac{u'((n+1)\Delta t) + u'(n\Delta t)}{2} = \frac{(u')^{n+1} + (u')^n}{2} = (u')^{n+\frac{1}{2}},$$

de modo que

$$\begin{aligned}g^n &= \delta u^{n+\frac{1}{2}} - u'(t_n) = \delta u^{n+\frac{1}{2}} - \frac{(u')^{n+1} + (u')^n}{2} \\ &= \frac{u^{n+1} - u^n}{\Delta t} - \frac{(u')^{n+1} + (u')^n}{2}.\end{aligned} \tag{8.85}$$

Aplicando intergração por partes, é imediato verificar a seguinte identidade:

$$\frac{1}{2(b-a)}\int_a^b (b-s)(s-a)\varphi'''(s)\,ds = \frac{\varphi(b)-\varphi(a)}{b-a} - \frac{\varphi'(b)-\varphi'(a)}{2}.$$

Utilisando esta identidade com $\varphi(s) = u(s,x)$, com $x \in (0,1)$ fixado, obtemos

$$g(t_n,x) = -\frac{1}{2\Delta t}\int_{t_n}^{t_{n+1}}(s-t_n)(t_{n+1}-s)u'''(s,x)\,ds,$$

de modo que

$$|g(t_n,x)| \leq \frac{1}{2\Delta t}\int_{t_n}^{t_{n+1}}|(s-t_n)(t_{n+1}-s)||u'''(s,x)|\,ds.$$

Logo, segue desta acima e da desigualdade de Schwarz na variável s,

$$|g(t_n,x)| \leq \frac{1}{2\Delta t}\left(\int_{t_n}^{t_{n+1}} (s-t_n)(t_{n+1}-s)\,ds\right)^{1/2}\left(\int_{t_n}^{t_{n+1}} |u'''(s,x)|^2\,ds\right)^{1/2}.$$

Elevando ao quadrado ambos os lados da desiguladade acima, obtemos

$$|g(t_n,x)|^2 \leq \frac{1}{4(\Delta t)^2}\left(\int_{t_n}^{t_{n+1}} (s-t_n)(t_{n+1}-s)\,ds\right)\left(\int_{t_n}^{t_{n+1}} |u'''(s,x)|^2\,ds\right).$$

Integrando os dois lados da desigualdade em $x \in (0,1)$ e aplicando o Teorema de Fubini, obtemos

$$\begin{aligned}\|g^n\|_0^2 &\leq \frac{1}{4(\Delta t)^2}\left(\int_{t_n}^{t_{n+1}} (s-t_n)(t_{n+1}-s)\,ds\right)\left(\int_{t_n}^{t_{n+1}} \|u'''(s)\|_0^2\,ds\right)\\ &\leq \frac{1}{4(\Delta t)}\left(\int_0^{\Delta t} \tau(\Delta t-\tau)\,d\tau\right)\|u'''\|^2_{L^\infty(0,T;L^2(0,1))}\\ &= \frac{1}{24}(\Delta t)^2\|u'''\|^2_{L^\infty(t_n,t_{n+1};L^2(0,1))},\end{aligned}$$

de modo que

$$\|g^n\|_0 \leq \frac{\Delta t}{2\sqrt{6}}\|u'''\|_{L^\infty(t_n,t_{n+1};L^2(0,1))}.$$

Assim, temos a estimativa do segundo termo em somatório em (8.82),

$$\Delta t\sum_{n=0}^{N-1}\|g^n\|_0 \leq \frac{(\Delta t)^2}{2\sqrt{6}}\sum_{n=0}^{N-1}\|u'''\|_{L^\infty(t_n,t_{n+1};L^2(0,1))} = \frac{(\Delta t)^2}{2\sqrt{6}}\|u'''\|_{L^\infty(0,T;L^2(0,1))}. \tag{8.86}$$

Substituindo (8.83), (8.84) e (8.86) em (8.82), obtemos

$$\|\xi^N\|_0 \leq C\Big(h^{k+1}\|u_0\|_{k+1} + h^{k+1}\|u'\|_{L^1(0,T;H^{k+1})} + (\Delta t)^2\|u'''\|_{L^\infty(0,T,L^2(0,1))}\Big). \tag{8.87}$$

Como por hipótese $u_0 \in H^{k+1}$, $u' \in L^1(0,T,H^{k+1}(0,1))$ e $u''' \in L^\infty(0,T,L^2(0,1))$, podemos afirmar que existe um constante $\widehat{C} > 0$ que depende de T, tal que

$$\|\xi^N\|_0 \leq \widehat{C}(h^{k+1} + (\Delta t)^2),$$

ou seja, dizemos que o erro tem ordem $(h^{k+1} + (\Delta t)^2)$, usualmente escrito na forma

$$\|\xi^N\|_0 \leq \mathcal{O}(h^{k+1} + (\Delta t)^2). \tag{8.88}$$

Por outro lado, tomando $m = 0$ em (8.11), obtemos

$$\|\rho^N\|_0 \leq \|u(t) - \widetilde{u}(t)\|_0 \leq ch^{k+1}\|u(t)\|_{k+1},$$

para cada t fixo em $[0,T]$.

Logo, da decomposição do erro obtemos

$$\|e^N\|_0 = \|u^N - P_h u^N\|_0 + \|P_h u^N - u_h^N\|_0 = \|\rho^N\|_0 + \|\xi^N\|_0 = \mathcal{O}(h^{k+1} + (\Delta t)^2).$$

Tomando o máximo em $t \in [0,T]$,

$$\|e\|_{L^\infty(0,T;L^2(0,1))} = \|u - u_h\|_{L^\infty(0,T;L^2(0,1))} \leq C\left(h^{k+1} + (\Delta t)^2\right).$$

Como a estimativa de erro no espaço $L^\infty(0,T;H_0^1(0,1))$ é obtida no próximo teorema (como caso particular com $\theta = 1/2$), concluímos a presente demostração. □

8.2.4 A Família θ-Métodos

Nesta seção demonstramos uma estimativa de erro para o método generalizado trapezoidal ou θ-métodos, descrito em (6.20). Como mencionado anteriormente, essa família engloba os métodos de Euler Progressivo ($\theta = 0$), Euler Regressivo ($\theta = 1$) e Crank-Nicolson ($\theta = 1/2$).

Teorema 8.6. *Sob a hipótese (8.51) e dados iniciais $u_0 \in H_0^1(0,1) \cap H^{k+1}(0,\ 1))$, a solução numérica obtida pelo Método Generalizado Trapezoidal (6.20) para $\theta \in [1/2, 1]$, tem a seguinte estimativa de erro:*

$$\begin{aligned} \|e\|_{L^\infty(0,T;L^2(0,1))} &= \|u - u_h\|_{L^\infty(0,T;L^2(0,1))} \leq C\left(h^{k+1} + (\Delta t)^\alpha\right) \\ \|e\|_{L^\infty(0,T;H_0^1(0,1))} &= \|u - u_h\|_{L^\infty(0,T;H_0^1(0,1))} \leq C\left(h^{k} + (\Delta t)^\alpha\right), \end{aligned} \tag{8.89}$$

onde C é uma constante positiva independente de h e Δt, com $\alpha = 2$ se $\theta = 1/2$ e $\alpha = 1$ se $\theta \neq 1/2$.

Demonstração: Vamos dividir a prova em duas etapas.

Etapa 1: *Estimativa em $L^\infty(0,T;L^2(0,1))$*

Seja $u(t)$ a solução exata na formulação fraca (8.3) e denotemos para $\theta \in [0,1]$,

$$\begin{aligned} u^{n+\theta} &= \theta u^{n+1} + (1-\theta)u^n = \theta u((n+1)\Delta t) + (1-\theta)u(n\Delta t), \\ u_t^{n+\theta} &= \theta (u')^{n+1} + (1-\theta)(u')^n = \theta u'((n+1)\Delta t) + (1-\theta)u'(n\Delta t). \end{aligned}$$

Então,

$$(u_t^{n+\theta} : v) + a(u^{n+\theta}, v) = (f^{n+\theta} : v), \quad \forall v \in H_0^1(0,1). \tag{8.90}$$

Somando e subtraindo $\delta u^{n+\frac{1}{2}}$ no primeiro membro de (8.90), obtemos

$$(\delta u^{n+\frac{1}{2}} : v) + a(u^{n+\theta}, v) = (f^{n+\theta} + g^n : v), \quad \forall v \in H_0^1(0,1), \tag{8.91}$$

onde $g^n = \delta u^{n+\frac{1}{2}} - u_t^{n+\theta}$.

Consideremos agora o sistema aproximado em V_m^k dado pela seguinte equação discreta

$$(\delta u_h^{n+\frac{1}{2}} : v_h) + a(u_h^{n+\theta}, v_h) = (f^{n+\theta} : v_h), \quad \forall v_h \in V_m^k. \tag{8.92}$$

Tomando em particular $v = v_h \in V_m^k$ em (8.91) e subtraindo de (8.92), obtemos

$$(\delta u^{n+\frac{1}{2}} - \delta u_h^{n+\frac{1}{2}} : v_h) + a(u^{n+\theta} - u_h^{n+\theta}, v_h) = (g^n : v_h), \quad \forall\, v_h \in V_m^k. \tag{8.93}$$

Somando e subtraindo $\delta P_h u^{n+\frac{1}{2}}$ e $P_h u^{n+\theta} \in V_m^k$ nos primeiro e segundo termos na identidade acima e usando a decomposição do erro (8.54), obtém-se

$$(\delta\xi^{n+\frac{1}{2}} : v_h) + (\delta\rho^{n+\frac{1}{2}} : v_h) + a(\xi^{n+\theta}, v_h) + a(\rho^{n+\theta}, v_h) = (g^n : v_h), \quad \forall v_h \in V_m^k. \tag{8.94}$$

Como $a(\rho^{n+\theta}, v_h) = a(\theta\rho^{n+1} + (1-\theta)\rho^n, v_h) = 0$, segue que

$$(\delta\xi^{n+\frac{1}{2}} : v_h) + a(\xi^{n+\theta}, v_h) = (g^n : v_h) - (\delta\rho^{n+\frac{1}{2}} : v_h), \quad \forall v_h \in V_m^k. \tag{8.95}$$

Tomando em particular $v_h = \xi^{n+\theta}$ em (8.95), obtemos

$$(\delta\xi^{n+\frac{1}{2}} : \xi^{n+\theta}) + a(\xi^{n+\theta}, \xi^{n+\theta}) = (g^n : \xi^{n+\theta}) - (\delta\rho^{n+\frac{1}{2}} : \xi^{n+\theta}), \quad \forall v_h \in V_m^k. \tag{8.96}$$

Observando que

$$\begin{aligned}
(\delta\xi^{n+\frac{1}{2}} : \xi^{n+\theta}) &= \frac{1}{\Delta t}(\xi^{n+1} - \xi^n : \theta\xi^{n+1} + (1-\theta)\xi^n) \\
&= \frac{1}{\Delta t}\Big(\theta\|\xi^{n+1}\|_0^2 + (1-2\theta)(\xi^{n+1} : \xi^n) - (1-\theta)\|\xi^n\|_0^2\Big), \\
a(\xi^{n+\theta}, \xi^{n+\theta}) &\geq 0,
\end{aligned}$$

obtemos de (8.96) a seguine estimativa

$$\begin{aligned}
\theta\|\xi^{n+1}\|_0^2 + (1-2\theta)(\xi^{n+1} : \xi^n) - (1-\theta)\|\xi^n\|_0^2 &\leq \Delta t\Big(g^n - \delta\rho^{n+\frac{1}{2}}, \xi^{n+\theta}\Big) \\
&\leq \Delta t\Big(\|g^n\|_0 + \|\delta\rho^{n+\frac{1}{2}}\|_0\Big)\|\xi^{n+\theta}\|_0.
\end{aligned} \tag{8.97}$$

Como $\theta \in [1/2, 1]$ implica que $(1-2\theta) \leq 0$, segue da desiguladade de Cauchy-Schwarz,

$$(1-2\theta)(\xi^{n+1} : \xi^n) \geq (1-2\theta)\|\xi^{n+1}\|_0\|\xi^n\|_0.$$

Além disso, como $\|\xi^{n+\theta}\|_0 \leq \theta\|\xi^{n+1}\|_0 + (1-\theta)\|\xi^n\|_0$, obtemos de (8.97)

$$\Big(\|\xi^{n+1}\|_0 - \|\xi^n\|_0\Big)\Big[\theta\|\xi^{n+1}\|_0 + (1-\theta)\|\xi^n\|_0\Big] \leq \Delta t\Big(\|g^n\|_0 + \|\delta\rho^{n+\frac{1}{2}}\|_0\Big)\Big[\theta\|\xi^{n+1}\|_0 + (1-\theta)\|\xi^n\|_0\Big].$$

Cancelando os termos em comum no dois lados da desigualdade e fazendo a soma para $n = 0, 1, \ldots, N-1$, obtemos

$$\|\xi^N\|_0 \leq \|\xi^0\|_0 + \Delta t\Big(\sum_{n=0}^{N-1} \|\delta\rho^{n+\frac{1}{2}}\|_0 + \sum_{n=0}^{N-1} \|g^n\|_0\Big). \tag{8.98}$$

Como vimos anteriormente, para $m = 0$ em (8.11), obtém-se

$$\|\xi^0\|_0 \leq c_1 h^{k+1} \|u_0\|_{k+1} \tag{8.99}$$

Vamos agora a analisar os dois termos restantes do lado direito de (8.98). Por definição

$$\delta\rho^{n+\frac{1}{2}} = \frac{1}{\Delta t}\Big(\rho^{n+1} - \rho^n\Big) = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \rho'(s)\, ds.$$

Então, tomando a norma em $L^2(0,1)$, obtemos

$$\|\delta\rho^{n+\frac{1}{2}}\|_0 \leq \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \|\rho'(s)\|_0\, ds,$$

fazendo a soma para $n = 0, 1, \ldots, N-1$ e multiplicando por Δt, temos

$$\begin{aligned} \Delta t \sum_{n=0}^{N-1} \|\delta\rho^{n+\frac{1}{2}}\|_0 &\leq \sum_{n=0}^{N-1} \int_{t_n}^{t_{n+1}} \|\rho'(s)\|_0\, ds = \int_0^{t_N} \|\rho'(s)\|_0\, ds \\ &\leq h^{k+1} \int_0^{t_N} \|u'(s)\|_{k+1}\, ds = h^{k+1} \|u'\|_{L^1(0,T;H^{k+1})}, \end{aligned} \tag{8.100}$$

onde usamos (8.11) com $m = 0$ na última desigualdade.

Para o segundo termo de (8.98), observemos que, somando e subtraindo $u_t^{n+\frac{1}{2}}$ na expressão que define g^n, obtemos a seguinte decomposição,

$$\begin{aligned} g^n &= \delta u^{n+\frac{1}{2}} - u_t^{n+\theta} = \frac{u^{n+1} - u^n}{\Delta t} - \theta u_t^{n+1} - (1-\theta) u_t^n \\ &= \frac{1}{\Delta t}\Big(u^{n+1} - u^n\Big) - \frac{1}{2}\Big(u_t^{n+1} + u_t^n\Big) - \Big(\theta - \frac{1}{2}\Big)\Big(u_t^{n+1} - u_t^n\Big) \\ &= g_1^n + g_2^n. \end{aligned} \tag{8.101}$$

Para estimar os termos g_1^n e g_2^n, procedemos da mesma forma que em (8.85). Assim,

$$g_1^n = \frac{1}{\Delta t}(u^{n+1} - u^n) - \frac{1}{2}(u_t^{n+1} + u_t^n) = -\frac{1}{2\Delta t}\int_{t_n}^{t_{n+1}} (s - t_n)(t_{n+1} - s) u'''(s)\, ds \tag{8.102}$$

e obtemos de (8.86)

$$\Delta t \sum_{n=0}^{N-1} \|g_1^n\|_0 \leq c_1 (\Delta t)^2 \|u'''\|_{L^\infty(0,T;L^2(0,1))}. \tag{8.103}$$

Por outro lado,

$$g_2^n = -\Big(\theta - \frac{1}{2}\Big)(u_t^{n+1} - u_t^n) = -\Big(\theta - \frac{1}{2}\Big) \int_{t_n}^{t_{n+1}} u''(s)\, ds.$$

Logo,

$$\|g_2^n\|_0 \leq \Delta t \Big(\theta - \frac{1}{2}\Big) \|u''\|_{L^\infty(t_n, t_{n+1}; L^2(0,1))}.$$

Somando de $n = 1, \ldots, N-1$ e multiplicando por Δt, obtemos

$$\Delta t \sum_{n=1}^{N-1} \|g_2^n\|_0 \leq \Delta t \Big(\theta - \frac{1}{2}\Big) \|u''\|_{L^\infty(0,T;L^2(0,1))}. \tag{8.104}$$

Substituindo (8.103) e 8.104) em (8.101), obtemos

$$\begin{aligned} \Delta t \sum_{n=1}^{N-1} \|g^n\|_0 &\leq \Delta t \sum_{n=1}^{N-1} \|g_1^n\|_0 + \Delta t \sum_{n=1}^{N-1} \|g_2^n\|_0 \\ &\leq c_1 (\Delta t)^2 \|u'''\|_{L^\infty(0,T,L^2(0,1))} + \Delta t \Big(\theta - \frac{1}{2}\Big) \|u''\|_{L^\infty(0,T;L^2(0,1))} \end{aligned} \tag{8.105}$$

e sbstituindo (8.99), (8.105) e (8.100) em (8.98), segue

$$\begin{aligned} \|\xi^N\|_0 \leq\ & C h^{k+1} \Big(\|u_0\|_{k+1} + \|u'\|_{L^1(0,T;H^{k+1})} \Big) \\ & + c_1 (\Delta t)^2 \|u'''\|_{L^\infty(0,T,L^2(0,1))} + \Delta t \Big(\theta - \frac{1}{2}\Big) \|u''\|_{L^\infty(0,T;L^2(0,1))}. \end{aligned} \tag{8.106}$$

Portanto, podemos afirmar que

$$\|\xi\|_{L^\infty(0,T;L^2(0,1))} \leq C(T) \Big(h^{k+1} + (\theta - 1/2)\Delta t + (\Delta t)^2 \Big), \tag{8.107}$$

onde

$$C(T) := \max \Big\{ \|u_0\|_{k+1}, \|u'\|_{L^1(0,T;H^{k+1})}, \|u'''\|_{L^\infty(0,T,L^2(0,1))}, \|u''\|_{L^\infty(0,T;L^2(0,1))} \Big\}.$$

Observemos que, para $\theta = 1/2$ (o que corresponde ao método de Crank-Nicolson), temos a ordem de convergência $\mathcal{O}\Big(h^{k+1}, (\Delta t)^2\Big)$, mas para $\theta > 1/2$, a ordem de convergência é $\mathcal{O}(h^{k+1}, \Delta t)$.

Por outro lado, fazendo $m = 0$ em (8.11), obtemos para cada t fixo em $[0, T]$,

$$\|\rho^N\|_0 = \|u(t) - P_h u(t)\|_0 \leq \|u(t) - \widetilde{u}(t)\|_0 \leq ch^{k+1}\|u(t)\|_{k+1}.$$

Logo,

$$\|\rho\|_{L^\infty(0,T;L^2(0,1))} \leq C_1 h^{k+1}\|u\|_{L^\infty(0,T;H^{k+1})}. \tag{8.108}$$

Da decomposição do erro e das estimativas (8.107) e (8.108), concluímos que:

1) para $\theta \neq 1/2$,

$$\|e\|_{L^\infty(0,T;L^2(0,1))} \leq \|\rho\|_{L^\infty(0,T;L^2(0,1))} + \|\xi\|_{L^\infty(0,T;L^2(0,1))} \leq C(h^{k+1} + \Delta t). \tag{8.109}$$

2) para $\theta = 1/2$,

$$\|e\|_{L^\infty(0,T;L^2(0,1))} \leq \|\rho\|_{L^\infty(0,T;L^2(0,1))} + \|\xi\|_{L^\infty(0,T;L^2(0,1))} \leq C\Big(h^{k+1} + (\Delta t)^2\Big), \tag{8.110}$$

com o que concluímos a etapa 1 da demostração.

Etapa 2: *Estimativa em* $L^\infty(0, T; H_0^1(0, 1))$.

Considere a igualdade dada por (8.95), ou seja,

$$(\delta\xi^{n+\frac{1}{2}} : v_h) + a(\xi^{n+\theta}, v_h) = (g^n : v_h) - (\delta\rho^{n+\frac{1}{2}} : v_h), \quad \forall v_h \in V_m^k. \tag{8.111}$$

Tomando em particular $v_h = \delta\xi^{n+\frac{1}{2}}$ em (8.111), obtemos

$$\|\delta\xi^{n+\frac{1}{2}}\|_0^2 + a(\xi^{n+\theta}, \delta\xi^{n+\frac{1}{2}}) = (g^n - \delta\rho^{n+\frac{1}{2}} : \delta\xi^{n+\frac{1}{2}}), \tag{8.112}$$

Para o primeiro termo de (8.112), temos

$$\begin{aligned} a(\xi^{n+\theta}, \delta\xi^{n+\frac{1}{2}}) &= \frac{1}{\Delta t} a(\xi^{n+1} - \xi^n, \theta\xi^{n+1} + (1-\theta)\xi^n) \\ &= \frac{1}{\Delta t}\Big(\theta\|\xi^{n+1}\|_a^2 + (1-2\theta)a(\xi^{n+1}, \xi^n) - (1-\theta)\|\xi^n\|_a^2\Big), \end{aligned} \tag{8.113}$$

onde estamos denotando $\|\cdot\|_a^{1/2}$ e $(\cdot,\cdot)_a$ respectivamente a norma e o produto interno associados à forma bilinear $a(\cdot,\cdot)$. Como $(1-2\theta) \leq 0$, temos das desigualdades de Cauchy-Schwarz e Young,

$$(1-2\theta)(\xi^{n+1} : \xi^n)_a \geq (1-2\theta)\|\xi^{n+1}\|_a\|\xi^n\|_a \geq \frac{(1-2\theta)}{2}\Big(\|\xi^{n+1}\|_a^2 + \|\xi^n\|_a^2\Big).$$

Substituindo-a em (8.113), obtemos

$$a(\xi^{n+\theta}, \delta\xi^{n+\frac{1}{2}}) \geq \frac{1}{2\Delta t}\Big(\|\xi^{n+1}\|_a^2 - \|\xi^n\|_a^2\Big).$$

Substituindo na identidade (8.112), temos a desigualdade

$$\begin{aligned}\|\delta\xi^{n+\frac{1}{2}}\|_0^2 + \frac{1}{2\Delta t}\Big(\|\xi^{n+1}\|_a^2 - \|\xi^n\|_a^2\Big) &\leq (g^n - \delta\rho^{n+\frac{1}{2}}, \delta\xi^{n+\frac{1}{2}}) \leq \|g^n - \delta\rho^{n+\frac{1}{2}}\|_0 \|\delta\xi^{n+\frac{1}{2}}\|_0 \\ &\leq \frac{1}{2}\Big(\|g^n - \delta\rho^{n+\frac{1}{2}}\|_0^2 + \|\delta\xi^{n+\frac{1}{2}}\|_0^2\Big),\end{aligned}$$

de onde se conclui que

$$\|\xi^{n+1}\|_a^2 - \|\xi^n\|_a^2 \leq \Delta t\|g^n - \delta\rho^{n+\frac{1}{2}}\|_0^2 \leq 2\Delta t\Big(\|g^n\|_0^2 + \|\delta\rho^{n+\frac{1}{2}}\|_0^2\Big). \tag{8.114}$$

Somando em n de 0 a $N-1$, temos

$$\|\xi^N\|_a^2 \leq \|\xi^0\|_a^2 + 2\Delta t \sum_{n=0}^{N-1}\Big(\|g^n\|_0^2 + \|\delta\rho^{n+\frac{1}{2}}\|_0^2\Big). \tag{8.115}$$

Devido a equivalência das normas $\|.\|_a$ e $\|.\|_1$, podemos usar a desigualdade (8.11) com $m = 1$ para obter

$$\|\xi^0\|_a \leq ch^k\|u_0\|_{k+1}.$$

Para os dois termos restantes (veja (8.100) e (8.105)),

$$\begin{aligned}&\Delta t\sum_{n=0}^{N-1}\|\delta\rho^{n+\frac{1}{2}}\|_0 \leq h^{k+1}\|u'\|_{L^1(0,T;H^{k+1})}, \\ &\Delta t\sum_{n=1}^{N-1}\|g^n\|_0 \leq \frac{(\Delta t)^2}{2}\|u'''\|_{L^\infty(0,T,L^2(0,1))} + \Delta t\Big(\theta - \frac{1}{2}\Big)\|u''\|_{L^\infty(0,T;L^2(0,1))}.\end{aligned}$$

Substituindo esses três termos em (8.115), considerando a equivalência das normas $\|\cdot\|_a$ e $\|\cdot\|_1$ e observando que

$$\big(a_0^2 + \cdots + a_k^2\big)^{1/2} \leq \sqrt{k}(|a_0| + \cdots + |s_k|), \quad \forall k \in \mathbb{N}, \quad \forall a_0, \ldots, a_k \in \mathbb{R},$$

podemos determinar $C > 0$ tal que

$$\begin{aligned}\|\xi^N\|_1 \leq C\Bigg(&h^k\|u_0\|_{k+1} + h^{k+1}\|u'\|_{L^1(0,T;H^{k+1})} + \frac{(\Delta t)^2}{2}\|u'''\|_{L^\infty(0,T,L^2(0,1))} \\ &+ \Delta t\Big(\theta - \frac{1}{2}\Big)\|u''\|_{L^\infty(0,T;L^2(0,1))}\Bigg).\end{aligned} \tag{8.116}$$

Para a conclusão da prova, basta observar a decomposição do erro. Logo, tomando $m = 1$ em (8.11), obtemos para cada t fixo em $[0, T]$,

$$\|\rho^N\|_1 \leq \|u(t) - \widetilde{u}(t)\|_1 \leq ch^k \|u(t)\|_{k+1}.$$

Tomando o máximo $t \in [0, T]$ na última desigualdade e em (8.116), obtemos as estimativas

$$\|e\|_{L^\infty(0,T;H_0^1(0,1))} \leq \|\rho\|_{L^\infty(0,T;H_0^1(0,1))} + \|\xi\|_{L^\infty(0,T;H_0^1(0,1))} \leq \begin{cases} C(h^k + \Delta t) & \text{se } \theta \neq 1/2, \\ C(h^k + (\Delta t)^2) & \text{se } \theta = 1/2, \end{cases}$$

com o que concluímos a prova do teorema. □

• **Observações:**

1 - No que se refere ao θ-método com $0 \leq \theta < 1/2$, temos a convergência condicionada à relação $\Delta t \leq h^2/2(1-2\theta)$ no caso de diferenças finitas (veja ([19])). De forma análoga, no método de elementos finitos temos uma restrição similar, ou seja $\Delta t/h^2 \leq f(\theta)$. As estimativas de erro nesse intervalo são as mesmas de (8.89). Note também que o Método de Euler progressivo ($\theta = 0$) é condicionalmente estável, como já mencionado.

2 - Note que as estimativas obtidas dependem do grau do polinômio interpolador, além da regularidade da solução, como visto no Teorema (3.4). Do citado teorema, sabemos que o erro de interpolação (3.48) é dado por

$$\|u - \widetilde{u}\|_s \leq ch^\alpha \|u\|_m,$$

onde $\alpha := \min\{k+1-s, m-s\}$, k é o grau do polinômio interpolador e $h = \max\{h^e\}$. Em particular, se o polinômio interpolador é o polinômio linear por partes, isto é, $k = 1$, o erro nas normas L^2 e H_0^1 são de ordem $\mathcal{O}(h^2)$ e $\mathcal{O}(h)$ respectivamente, para toda solução regular com $m \geq 2$. Por outro lado, se a solução $u \in H_0^1 \cap H^2$, a ordem de convergência tem a mesma ordem anterior, independentemente do grau do polinômio interpolador definido em V_m^k.

8.3 Exercício

Mostre que o método de Euler progressivo (6.18) é condicionalmente estável, ou seja, existe um $\delta > 0$ suficientemente pequeno tal que $\Delta t \leq \delta h^2$.

Sugestão: Usando o mesmo procedimento feito a partir de (8.54), somando e subtraindo os termos $\delta\widetilde{u}^{n+\frac{1}{2}}$, obtém-se que

$$(\delta\xi^{n+\frac{1}{2}} : v_h) + a(\xi^n, v_h) = (\delta u^{n+\frac{1}{2}} - \frac{d}{dt}u^n : v_h) - (\delta\rho^{n+\frac{1}{2}} : v_h), \quad \forall v_h \in V_m^k. \quad (8.117)$$

Tomando em particular $v_h = \xi^n$ em (8.117), obtenha estimativas semelhantes às obtidas em (8.63) e (8.72).

CAPÍTULO 9

Análise Numérica da Equação da Onda

Nesse capítulo provaremos os resultados de convergência dos métodos numéricos e algoritmos para a Equação da Onda, tanto no contexto do problema semidiscreto quanto no caso discreto. Para maior aprofudamento no assunto, sugerimos ao leitor consultar, por exemplo, as referências [1, 10, 11, 17].

9.1 Estimativas de Erro: Problema Semidiscreto

Nessa seção vamos estabelecer estimativas de erro para os problemas semidiscretos referentes à equação da onda. Como anteriormente, vamos considerar as estimativas de erros nos espaços $L^2(\Omega)$ e $H_0^1(\Omega)$ no caso unidimensional, ou seja, $\Omega = (0,1)$.

Vimos em (7.3) que a formulação fraca para a equação da onda é dada por

$$\begin{cases} (u'' : v) + a(u,v) = (f : v), \\ (u(0) : v) = (u_0 : v), \qquad\qquad \forall\, v \in H_0^1(0,1), \\ (u'(0) : v) = (u_1 : v), \end{cases} \tag{9.1}$$

onde $a(\cdot,\cdot)$ é a forma bilinear definida em $H_0^1(0,1) \times H_0^1(0,1)$ por

$$a(u,v) = \alpha \int_0^1 u_x v_x \, dx + \beta \int_0^1 uv \, dx, \quad \alpha > 0,\ \beta \geq 0 \tag{9.2}$$

e

$$(f,v) = \int_0^1 f(t)\, v \, dx. \tag{9.3}$$

Como $a(\cdot,\cdot)$ é contínua e coerciva, ela define um produto escalar em $H_0^1(0,1)$, denotado por $(\cdot,\cdot)_a$, sendo $\|\cdot\|_a$ a norma associada.

Consideremos as soluções aproximadas $u_h \in V_m$, com V_m definido em (8.1), sendo o problema aproximado definido por

$$\begin{cases} (u_h'' : v_h) + a(u_h, v_h) = (f : v_h), \\ (u_h(0) : v_h) = (u_{0h} : v_h), \qquad \forall\, v \in V_m. \\ (u_h'(0) : v_h) = (u_{1h} : v_h), \end{cases} \tag{9.4}$$

Considerando $v = v_h$ em (9.1) e fazendo a diferença entre (9.1) e (9.4), obtemos

$$\begin{cases} (u'' - u_h'' : v_h) + a(u - u_h : v_h) = 0, \\ (u(0) - u_h(0) : v_h) = (u_0 - u_{0h} : v_h), \quad \forall\, v_h \in V_m. \\ (u'(0) - u_h'(0) : v_h) = (u_1 - u_{1h} : v_h), \end{cases} \tag{9.5}$$

Consideremos a projeção P_h definida anteriormente em (8.5), ou seja,

$$a(u(t) - P_h u(t), v) = 0, \quad \forall v \in V_m, \tag{9.6}$$

de modo que, para cada t fixo, vale a estimativa (8.10), onde

$$P_h u(t) = \sum_{i=1}^{m} c_i(t)\varphi_i, \quad \widetilde{u}(x,t) = \sum_{i=1}^{m} u_i(t)\varphi_i.$$

9.1.1 Estimativa na norma de $H_0^1(0,1)$

Nunca é demais lembrar que os dados iniciais $u_h(0) = u_{0h} \in V_m$ e $u_h'(0) = u_{1h} \in V_m$ são essenciais para a determinação da solução aproximada $u_h(x,t)$, obtida após a resolução de um sistema de equações diferenciais ordinárias. É evidente que diferentes dados iniciais implicam em diferentes soluções aproximadas $u_h(x,t)$. Como o objetivo é mostrar que a solução aproximada $u_h(x,t)$ converge para a solução exata $u(x,t)$ em alguma norma, quando $h \to 0$, é necessário que $u_h(0)$ e $u_h'(0)$ convirjam, respectivamente, para $u(0)$ e $u'(0)$ em alguma norma, quando $h \to 0$.

Tendo em vista os resultados que paresentaremos, vamos supor que os dados iniciais do problema aproximado satisfaçam as seguintes estimativas:

$$\|u_h(0) - P_h u(0)\|_1 \le c_1 h, \qquad \|u_h'(0) - P_h u'(0)\|_0 \le c_2 h. \tag{9.7}$$

onde $P_h u(0)$ e $P_h u'(0)$ são respectivamente as projeções ortogonais de u_0 e u_1 em V_m.

Teorema 9.1. *Se os dados iniciais satisfazem (9.7) e $\{u, u', u''\} \subset L^\infty(0,T; H_0^1 \cap H^2)$, o erro entre a solução aproximada $u_h(x,t)$ e a solução exata $u(x,t)$ na norma $H_0^1(0,1)$ é de ordem $\mathcal{O}(h)$. Mais precisamente,*

$$\|u - u_h\|_{L^\infty(0,T;H_0^1(0,1))} + \|u' - u_h'\|_{L^\infty(0,T;L^2(0,1))} \le ch, \tag{9.8}$$

onde c é uma constante positiva independente de h.

Notemos que, nas condições do Teorema 9.1, a seguinte estimativa é suficiente para concluir (9.8):

$$\|u(t) - u_h(t)\|_1 + \|u'(t) - u_h'(t)\|_0 \leq ch, \quad \forall t \in [0, T]. \tag{9.9}$$

Notemos também que a constante c, embota não dependente de h, depende do tempo final T, de modo que pode tender ao infinito quando $T \to \infty$.

Demonstração: Consideremoos a seguinte decomposição do erro $e(t)$:

$$e(t) := u(t) - u_h(t) = (u(t) - P_h u(t)) + (P_h u(t) - u_h(t)) =: \rho(t) + \xi(t).$$

Somando e subtraindo $P_h u(t)$ em (9.5) e lembrando que $a(\rho(t), v_h) = 0$ (veja (9.6), obtemos

$$(\rho''(t) : v_h) + (\xi''(t) : v_h) + a(\xi(t) : v_h) = 0, \quad \forall v_h \in V_m^k. \tag{9.10}$$

Como $\xi(t)$ e $\xi'(t)$ pertencem a V_m para todo $t \in [0, T]$, podemos substituir v_h por $\xi'(t)$ em (9.10), de modo que

$$(\rho''(t) : \xi'(t)) + (\xi''(t) : \xi'(t)) + a(\xi(t), \xi'(t)) = 0, \tag{9.11}$$

que é equivalente a

$$\frac{1}{2}\frac{d}{dt}\left\{\|\xi'(t)\|_0^2 + \|\xi(t)\|_a^2\right\} = -(\rho''(t) : \xi'(t)). \tag{9.12}$$

Integrando (9.12) com respeito a s e usando as desigualdades de Schwarz e Young, obtemos

$$\begin{aligned}
\|\xi'(t)\|_0^2 + \|\xi(t)\|_a^2 &\leq \|\xi'(0)\|_0^2 + \|\xi(0)\|_a^2 + 2\int_0^t \|\rho''(s)\|_0 \|\xi'(s)\|_0 \, ds \\
&\leq \|\xi'(0)\|_0^2 + \|\xi(0)\|_a^2 + \int_0^t \left(\|\rho''(s)\|_0^2 + \|\xi'(s)\|_0^2\right) \, ds \\
&\leq \|\xi'(0)\|_0^2 + \|\xi(0)\|_a^2 + \int_0^t \left(\|\rho''(s)\|_0^2 + \|\xi'(s)\|_0^2 + \|\xi(s)\|_1^2\right) \, ds
\end{aligned} \tag{9.13}$$

e como consequência da desigualdade de Gronwall,

$$\|\xi'(t)\|_0^2 + \|\xi(t)\|_a^2 \leq \left(\|\xi'(0)\|_0^2 + \|\xi(0)\|_a^2 + \int_0^t \|\rho''(s)\|_0^2 \, ds\right) \exp(t), \quad \forall t \in [0, T]. \tag{9.14}$$

Pela hipótese (9.7), $\|\xi'(0)\|_0 \leq c_2 h$ e $\|\xi(0)\|_a \leq c_1 h$. Além disso, considerando $m = 0$ e $k = 1$ em (8.11), segue que

$$\|\rho''\|_{L^\infty(0,T;L^2(\Omega))} \leq c_3 h^2 \|u''\|_2.$$

Assim, para $t \in [0,T]$, temos

$$\|\xi'(t)\|_0^2 + \alpha\|\xi(t)\|_1^2 + \beta\|\xi(t)\|_0^2 = \|\xi'(t)\|_0^2 + \|\xi(t)\|_a^2 \leq C(T)h^2$$

e lembrando que

$$\left(\|\xi'(t)\|_0 + \|\xi(t)\|_a\right)^2 \leq 2\|\xi'(t)\|_0^2 + 2\|\xi(t)\|_a^2 \leq 2C(T)h^2, \quad \forall t \in [0,T],$$

segue que

$$\|\xi'(t)\|_0 + \|\xi(t)\|_a \leq \sqrt{2C(T)}h.$$

Sendo

$$\min\{\sqrt{\alpha}, \sqrt{\beta}\}\left(\|\xi(t)\|_1^2 + \|\xi(t)\|_0^2\right)^{1/2} \leq \|\xi(t)\|_a,$$

obtemos,

$$\min\{1, \sqrt{\alpha}, \sqrt{\beta}\}\left(\|\xi'(t)\|_0 + \left(\|\xi(t)\|_1^2 + \|\xi(t)\|_0^2\right)^{1/2}\right) \leq \sqrt{2C(T)}h.$$

Portanto, tomado o máximo em $t \in [0,T]$, concluímos

$$\|\xi'\|_{L^\infty(0,T;L^2(0,1))} + \|\xi\|_{L^\infty(0,T;H_0^1(0,1))} \leq \frac{\widehat{C}}{\min\{1, \sqrt{\alpha}, \sqrt{\beta}\}}h,$$

com o que terminamos a demonstração. □

9.1.2 Estimativa na norma de $L^2(0,1)$

Para obter a estimativa de erro na norma de $L^2(0,1)$, é necessário supor maior precisão nas aproximações dos dados iniciais. Assim, vamos supor que

$$\|u_h(0) - P_h u(0)\|_0 \leq c_1 h^2; \qquad \|u_h'(0) - P_h u'(0)\|_0 \leq c_2 h^2, \tag{9.15}$$

onde, como definido anteriormente, $P_h u(0)$ e $P_h u'(0)$ são respectivamente as projeções ortogonais de u_0 e u_1 em V_m^k.

Teorema 9.2. *Se os dados iniciais satisfazem (9.15) e $\{u, u', u''\} \subset L^\infty(0,T; H_0^1 \cap H^2)$, o erro entre a solução aproximada $u_h(x,t)$ e a solução exata $u(x,t)$ na norma de $L^2(0,1)$ é de ordem $\mathcal{O}(h^2)$. Mais precisamente,*

$$\|u - u_h\|_{L^\infty(0,T;L^2(\Omega))} + \|u' - u_h'\|_{L^\infty(0,T;L^2(\Omega))} \leq ch^2, \tag{9.16}$$

onde c é uma constante positiva independente de h.

Demonstração: Note que, pela definição da norma $L^\infty(0,T)$, para obter a estimativa (9.16) é suficiente estabelecer a desigualdade

$$\|u(t) - u_h(t)\|_0 + \|u'(t) - u_h'(t)\|_0 \leq ch^2, \quad \forall t \in [0,T]. \tag{9.17}$$

Com efeito, considere a decomposição do erro,

$$e(t) = u(t) - u_h(t) = (u(t) - P_h u(t)) + (P_h u(t) - u_h(t)) = \rho(t) + \xi(t).$$

Fazendo mesmo procedimento de (9.10) e usando a projeção ortogonal (9.6), obtemos a seguinte relação equivalente a (9.13),

$$\|\xi'(t)\|_0^2 + \|\xi(t)\|_a^2 \leq \|\xi'(0)\|_0^2 + \|\xi(0)\|_a^2 + \int_0^t \left(\|\rho''(s)\|_0^2 + \|\xi'(s)\|_0^2\right) ds. \tag{9.18}$$

Para a estimativa $L^2(\Omega)$, podemos descarta positivo $\|\xi(t)\|_a$ no lado esquerdo da desigualdade (9.18). Assim, segue da desigualdade de Gronwall,

$$\|\xi'(t)\|_0^2 \leq \left(\|\xi'(0)\|_0^2 + \|\xi(0)\|_a^2 + \|\rho''\|^2_{L^2((0,T)\times(0,1))}\right) e^T. \tag{9.19}$$

Lembrando que se $w \in H^1(0,T,L^2(0,1))$, vale o Teorema Fundamental do Cálculo, isto é, vale a seguinte igualdade em $L^2(0,1)$,

$$w(t) = w(0) + \int_0^t w'(s)\, ds, \quad \forall t \in [0,T].$$

Assim, temos

$$\|w(t)\|_0 \leq \|w(0)\|_0 + \int_0^t \|w'(s)\|_0\, ds$$

e, em especial

$$\|w(t)\|_0^2 \leq 2\|w(0)\|_0^2 + 2\left(\int_0^t \|w'(s)\|_0\, ds\right)^2. \tag{9.20}$$

Pela desigualdade de Cauchy-Schwarz, temos

$$\int_0^t \|w'(s)\|_0\, ds \leq \sqrt{T}\left(\int_0^t \|w'(s)\|_0^2\, ds\right)^{1/2},$$

que, substiuído em (9.20), nos dá

$$\|w(t)|_0^2 \leq c\left(\|w(0)\|_0^2 + \int_0^t \|w'(s)\|_0^2\, ds\right), \quad \forall t \in [0,T]. \tag{9.21}$$

Assim, considerando a regularidade de u, podemos tomar $w = \rho'$ na desigualdade (9.21), de modo que

$$\|\rho'(t)\|_0^2 \leq c\left(\|\rho_0'\|_0^2 + \int_0^t \|\rho''(s)\|_0^2\, ds\right), \quad \forall t \in [0,T]. \tag{9.22}$$

Sendo $e(t) = \rho(t) + \xi(t)$, temos

$$\|e'(t)\|_0^2 \leq 2\|\rho'(t)\|_0^2 + 2\|\xi'(t)\|_0^2,$$

de onde segue, juntando (9.19) e (9.22), a seguinde desigualdade:

$$\|e'(t)\|_0^2 \leq C_3\Big(\|\rho''\|^2_{L^\infty(0,T;L^2)} + \|\xi'(0)\|_0^2 + \|\xi(0)\|_a^2 + \|\rho'(0)\|_0^2\Big). \tag{9.23}$$

Considerando agora $w = e$ em (9.21), obtemos

$$\|e(t)\|_0^2 \leq C_4\Big(\|e(0)\|_0^2 + \int_0^t \|e'(s)\|_0^2\, ds\Big), \quad \forall t \in [0,T]. \tag{9.24}$$

Somando as equações (9.23) e (9.24) obtemos

$$\begin{aligned}
\|e(t)\|_0^2 + \|e'(t)\|_0^2 &\leq C_5\Bigg(\|\rho''\|^2_{L^\infty(0,T;L^2)} + \|\xi'(0)\|_0^2 + \|\xi(0)\|_a^2 + \|\rho'(0)\|_0^2 \\
&\qquad + \|e(0)\|_0^2 + \int_0^t \|e'(s)\|_0^2\, ds\Bigg) \\
&\leq C_5\Bigg(\|\rho''\|^2_{L^\infty(0,T;L^2)} + \|\xi'(0)\|_0^2 + \|\xi(0)\|_a^2 + \|\rho'(0)\|_0^2 \\
&\qquad + \|e(0)\|_0^2 + \int_0^t \Big(\|e(s)\|_0^2 + \|e'(s)\|_0^2\Big)\, ds\Bigg).
\end{aligned} \tag{9.25}$$

Assim, definindo

$$\varphi(t) = \|e(t)\|_0^2 + \|e'(t)\|_0^2,$$
$$\widehat{C} = C_5\Big(\|\rho''\|^2_{L^\infty(0,T;L^2)} + \|\xi'(0)\|_0^2 + \|\xi(0)\|_a^2 + \|\rho'(0)\|_0^2 + \|\rho(0)\|_0^2 + \|\xi(0)\|_0^2\Big),$$

a desigualdade (9.25) pode ser escrita na forma

$$\varphi(t) \leq \widehat{C} + C_5 \int_0^t \varphi(s)\, ds,$$

de onde se obtém pelo Lema de Gronwal

$$\|e(t)\|_0^2 + \|e'(t)\|_0^2 \leq \widehat{C} e^{C_5 T}. \tag{9.26}$$

Para concluir, vamos mostrar que cada parcela que compõe a constante $\widehat{C}$ é de ordem $\mathcal{O}(h^2)$. De fato, observemos inicalmente que

$$\|\rho''(t)\|_0 = \|u''(t) - P_h u''\|_0 \leq ch^{k+1}\|u''(t)\|_{k+1}$$

e da hipótese $u'' \in L^\infty(0,T;H^{k+1}) \subset L^2(0,T;H^{k+1})$, segue que

$$\|u''\|^2_{L^2(0,T;H^{k+1})} = \int_0^T \|u''(s)\|^2_{k+1} ds < \infty.$$

Portanto, para $m = 0$ e $k = 1$, temos

$$\|\rho''\|^2_{L^\infty(0,T;L^2)} \leq \widehat{c}_1 h^4 \|u''\|^2_{L^\infty(0.T;H^2)}.$$

Por outro lado, de (8.12) com $t = 0$, temos

$$\|\rho'(0)\|^2_0 \leq \widehat{c}_2 h^4 \|u'(0)\|^2_2$$

e de (9.15),

$$\|\xi(0)\|^2_1 = \|u_h(0) - P_h u(0)\|^2_1 \leq \widehat{c}_3 h^4 \quad \text{e} \quad \|\xi'(0)\|^2_0 = \|u_h'(0) - P_h u'(0)\|^2_0 \leq \widehat{c}_3 h^4.$$

Assim, como

$$\Big(\|e(t)\|_0 + \|e'(t)\|_0\Big)^2 \leq 2\|e(t)\|^2_0 + 2\|e'(t)\|^2_0 \leq \widehat{C}_7 h^4, \quad \forall t \in [0,T],$$

obtemos aoós extração da raiz quadrada,

$$\|e(t)\|_0 + \|e'(t)\|_0 \leq \widehat{C}_8 h^2, \qquad \forall t \in [0,T],$$

onde $\widehat{C}_8$ é constante que depende de T. Logo, passando ao supremo em $t \in [0,T]$, temos

$$\|e\|_{L^\infty(0,T;L^2)} + \|e'\|_{L^\infty(0,T;L^2)} \leq C(T)h^2 \tag{9.27}$$

com o que concluímos a prova. □

9.2 Estimativa de Erro: Problema Discreto

Nesta seção vamos estabelecer os erros nas normas discretas de $L^\infty(0,T;H^1_0(0,1))$ e $L^\infty(0,T;L^2(0,1))$ para a discretização uniforme no tempo, *i.e.*, $t = n\Delta t$, $n = 0,1,\ldots,N$, referentes aos métodos de Newmark (também conhecido como método θ-Newmark) e θ-método.

• **Método de Newmark**

Consideremos o sistema (7.9) e a seguinte aproximação (veja (7.16)),

$$\frac{1}{(\Delta t)^2}A(\boldsymbol{d}^{n+1}-2\boldsymbol{d}^n+\boldsymbol{d}^{n-1})+B\boldsymbol{d}_\theta^n=\boldsymbol{F}_\theta^n,\quad \theta\geq 0, \tag{9.28}$$

onde (ver em (8.36)) $\boldsymbol{d}_\theta^n:=\boldsymbol{d}^n+\theta(\boldsymbol{d}^{n+1}-2\boldsymbol{d}^n+\boldsymbol{d}^{n-1})$ e $\boldsymbol{d}=(d_1,d_2,\ldots,d_m)^T$, com $d_i=u(t_i)$.

Multiplicando a equação por $(\Delta t)^2$, obtemos

$$A(\boldsymbol{d}^{n+1}-2\boldsymbol{d}^n+\boldsymbol{d}^{n-1})+(\Delta t)^2B\boldsymbol{d}_\theta^n=(\Delta t)^2\boldsymbol{F}_\theta^n, \tag{9.29}$$

que pode ser reescrito na seguinte forma

$$\begin{aligned}\Big(A+\theta(\Delta t)^2B\Big)\boldsymbol{d}^{n+1}=&\Big(2A-(1-2\theta)(\Delta t)^2B\Big)\boldsymbol{d}^n-\Big(A+\theta(\Delta t)^2B\Big)\boldsymbol{d}^{n-1}\\&+(\Delta t)^2\Big(\theta\boldsymbol{F}^{n+1}+\theta\boldsymbol{F}^{n-1}+(1-2\theta)\boldsymbol{F}^n\Big).\end{aligned} \tag{9.30}$$

Note que (9.30) se reduz ao método da diferença central no caso $\theta=0$.

Para inicializar o algoritmo (9.30), precisamos fornecer os dados $\boldsymbol{d}^1$ e $\boldsymbol{d}^0$ correspondentes aos dados iniciais $u_h(0)=u_{0h}$ e $u_h'(0)=u_{1h}$ com aproximações adequadas. O seguinte resultado fornece as condições suficientes se u é suficientemente regular.

Lema 9.1. *Se $u'''\in L^\infty(0,T;H_0^1(0,1))$, temos as seguintes estimativas:*

$$\begin{aligned}&\|P_hu^0-u_h^0\|_1+\|P_hu^1-u_h^1\|_1+\|\delta(P_hu-u_h)^{1/2}\|_0\leq C(h+(\Delta t)^2),\\&\|P_hu^0-u_h^0\|_0+\|P_hu^1-u_h^1\|_0+\|\delta(P_hu-u_h)^{1/2}\|_0\leq C(h^2+(\Delta t)^2),\end{aligned} \tag{9.31}$$

onde C representa diferentes constantes positivas independente de h e (Δt).

Demonstração: Os dados iniciais são conhecidos, $u(x,0)=u_0(x)$ e $u'(x,0)=u_1(x)$. Então, das desigualdades (8.10), (9.8), temos

$$\begin{aligned}\|\xi^0\|_1=\|P_hu^0-u_h^0\|_1&\leq\|P_hu^0-u^0\|_1+\|u^0-u_h^0\|_1=\|\rho^0\|_1+\|e^0\|_1\\&\leq c_1h\|u_0\|_2+c_2h\leq Ch.\end{aligned}$$

Por outro lado, no tempo $t_1=\Delta t$, a solução exata $u(x,t_1)=u^1(x)$ não é conhecida e assim não é possível determinar diretamente a sua projeção $P_hu(x,t_1)=P_hu^1(x)$. Entretanto, podemos fazer uma aproximação $\widehat{u}(x)$ para a solução exata $u(x,t_1)=u^1(x)$, a partir de uma expansão de Taylor de segunda ordem da solução exata u, na vizinhança do tempo $t_0=0$, ou seja,

$$\widehat{u}^1(x):=u(x,0)+u'(x,0)\Delta t+u''(x,0)\frac{(\Delta t)^2}{2!}. \tag{9.32}$$

A posição inicial $u(x,0)$ e a velocidade inicial $u'(x,0)$ são dados do problema e a aceleração inicial $u''(x,0)$ pode ser obtida da equação (7.1), para $t=0$, isto é,

$$u''(x,0) = \alpha u_{xx}(x,0) - \beta u(x,0) + f(x,0).$$

Assim, o erro da aproximação de $\widehat{u}$ no tempo $t=t_1$ para u^1 é dado por

$$u^1(x) - \widehat{u}^1(x) = \int_0^{\Delta t} \frac{(\Delta t - s)^2}{2!} u'''(x,s)\, ds. \tag{9.33}$$

Com a regularidade que estamos supondo para a solução u, i.e., $u''' \in L^\infty(0,T,H_0^1(0,1))$, temos

$$\|u^1 - \widehat{u}^1\|_1 \leq \int_0^{\Delta t} \frac{(\Delta t - s)^2}{2!} \|u'''(s)\|_1\, ds \leq \frac{1}{3!}\|u'''\|_{L^\infty(0,T,H_0^1(0,1))}(\Delta t)^3. \tag{9.34}$$

Por (9.32), temos $\widehat{u}^1 \in H^2(0,1)\cap H_0^1(0,1)$. Logo, podemos tomar $u_h^1(x) = \widehat{u}^1(x)$ como solução aproximada no tempo t_1, ou satisfazendo a condição $\|u_h^1 - \widehat{u}^1\|_1 \leq c_2 h$. Assim, usando a desigualdade triangular,

$$\begin{aligned} \|P_h u^1 - u_h^1\|_1 &\leq \|P_h u^1 - u^1\|_1 + \|u^1 - u_h^1\|_1 \leq \|\rho^1\|_1 + \|u^1 - u_h^1\|_1 \\ &\leq c_0 h + \|u^1 - \widehat{u}^1\|_1 + \|\widehat{u}^1 - u_h^1\|_1 \\ &\leq c_0 h + c_1(\Delta t)^3 + c_2 h \leq c_3((\Delta t)^3 + h). \end{aligned} \tag{9.35}$$

De forma análogo, supondo que $\|u_h^1 - \widehat{u}^1\|_0 \leq c_2 h^2$ e substituindo a norma $\|\ \|_1$ pela norma $\|\ t\|_0$, obtemos a estimativa

$$\begin{aligned} \|P_h u^1 - u_h^1\|_0 &\leq \|P_h u^1 - u^1\|_0 + \|u^1 - u_h^1\|_0 \leq \|\rho^1\|_0 + \|u^1 - u_h^1\|_0 \\ &\leq c_0 h + \|u^1 - \widehat{u}^1\|_0 + \|\widehat{u}^1 - u_h^1\|_0 \\ &\leq c_0 h^2 + c_1(\Delta t)^3 + c_2 h^2 \leq c_3((\Delta t)^3 + h^2). \end{aligned} \tag{9.36}$$

Assim de (9.35) e (9.36) concluímos que:

$$\|\xi^1\|_1 \leq c_1\Big(h + (\Delta t)^3\Big) \quad \text{e} \quad \|\xi^1\|_0 \leq c_2\Big(h^2 + (\Delta t)^3\Big).$$

Além disso, sabemos que $\delta e^{1/2} = \frac{1}{\Delta t}(e^1 - e^0)$. Logo

$$\|\delta\xi^{1/2}\|_0 \leq \frac{1}{\Delta t}\Big(\|\xi^1\|_0 + \|\xi^0\|_0\Big) \leq c_3\Big(h^2 + (\Delta t)^2\Big). \tag{9.37}$$

Portanto,

$$\|\xi^1\|_1 + \|\xi^0\|_1 + \|\delta\xi^{1/2}\|_0 \leq c\Big(h + (\Delta t)^2\Big).$$

De forma análoga, observando (9.36), tem-se para a norma $L^2(0,1)$,

$$\|\xi^1\|_0 + \|\xi^0\|_0 + \|\delta\xi^{1/2}\|_0 \leq c\Big(h^2 + (\Delta t)^2\Big).$$

Finalmente, concluímos a prova do Lema a partir da decomposição do erro. □

Uma vez estabelecidas aproximações para os dados inciais, podemos tratar a estimativa de erro (ou ordem de convergência) para os métodos de Neumark e θ-método nas normas discretas de $L^\infty(0,T;H_0^1(\Omega))$ e $L^\infty(0,T;L^2(\Omega))$ (i.e., nas normas $L_0^\infty(0,T;H_0^1(\Omega))$ e $L_0^\infty(0,T;L^2(\Omega))$), quando o tempo t varia discretamente na forma $t_n = n\Delta t$, $n = 0,1,\ldots,N$.

Suponhamos a seguinte regularidade para a força f, a solução u e sua derivada u' do Problema (9.1).

$$\left|\begin{array}{l} f \in L^2(0,\ T;H^2(0,1)), \quad u \in L^\infty(0,T;H_0^1(0,1)\cap H^2(0,1)) \\ \text{com as derivadas} \quad \{u_t,\ u_{tt},\ u_{ttt},\ u_{tttt}\} \subset L^\infty(0,T;L^2(0,1)). \end{array}\right. \tag{9.38}$$

O resultado a seguir é o análogo discreto dos Teoremas (9.1) e (9.2).

Teorema 9.3. *Supondo (9.38), se $u_0 \in H_0^1(0,1)\cap H^2(0,1))$ e $u_1 \in H_0^1(0,1)$, o erro entre a solução exata u do Problema (9.1) e a solução aproximada u_h do Problema (9.4), obtida pelo método de Neumark (9.30), satisfaz as seguintes estimativas:*

$$\left\{\begin{array}{l} (1)\ \ \|u-u_h\|_{L_0^\infty(0,T;H_0^1(0,1))} + \|\delta(u-u_h)\|_{L_{1/2}^\infty(0,T;L^2(0,1))} \leq C(h+(\Delta t)^2), \\ (2)\ \ \|u-u_h\|_{L_0^\infty(0,T;L^2(0,1))} + \|\delta(u-u_h)\|_{L_{1/2}^\infty(0,T;L^2(0,1))} \leq C(h^2+(\Delta t)^2), \end{array}\right. \tag{9.39}$$

onde a constante $C>0$ é independente da malha h e da discretização Δt e:

(a) *para $\theta > 1/4$, o sistema é incondicionalmente estável, qualquer que seja $\Delta t > 0$;*

(b) *para $\theta = 1/4$, o sistema é condicionalmente estável.*

Demonstração: A ideia central da prova é obter estimativas que nos permitam aplicar o Lema 8.2, que é o análogo discreto do Lema de Gronwall. Faremos a prova em detalhes de (9.39) no caso (1); a prova do caso (2) segue com argumento semelhante.

O problema em questão tem sua formulação fraca dada por

$$(u_{tt} : v) + a(u,v) = (f : v), \quad \forall v \in H_0^1(\Omega). \tag{9.40}$$

Considerando em (9.40) os tempos discretos $(n+1)\Delta t$, $n\Delta t$ e $(n-1)\Delta t$ respectivamente com pesos θ, $(1-2\theta)$ e θ, somando os termos obtidos e considerando a notação (8.36), obtemos

$$(\star_\theta u_{tt}^n : v) + a(\star_\theta u^n, v) = (\star_\theta f^n : v), \quad \forall v \in H_0^1(0,1).$$

Somando e subtraindo o termo $(\delta^2 u^n : v)$, obtemos

$$(\delta^2 u^n : v) + a(\star_\theta u^n, v) = (\star_\theta f^n + g^n : v), \quad \forall v \in H_0^1(0,1), \tag{9.41}$$

onde $g^n = (\delta^2 u^n - \star_\theta u_{tt}^n)$ e $\delta^2 u^n$ é a aproximação de u_{tt} como definido em (8.36).

Por outro lado, para o problema aproximado, a formulação é dada por

$$(\delta^2 u_h^n : v_h) + a(\star_\theta u_h^n, v_h) = (\star_\theta f * n, v_h)_0, \quad \forall v_h \in V_m^k. \tag{9.42}$$

Substituindo $v = v_h \in V_m^k$ em (9.41) e subtraindo da equação (9.42), obtemos

$$(\delta^2 e^n : v_h) + a(\star_\theta e^n, v_h) = (g^n : v_h), \quad \forall v_h \in V_m^k$$

e usando a decomposição do erro, $e = \xi + \rho$,

$$(\delta^2 \xi^n : v_h) + (\delta^2 \rho : v_h) + a(\star_\theta \xi^n, v_h) + a(\star_\theta \rho^n, v_h) = (g^n : v_h), \quad \forall v_h \in V_m. \tag{9.43}$$

No que segue, vamos considerar em (9.43)

$$v_h = \frac{1}{2}\Big(\delta\xi^{n+\frac{1}{2}} + \delta\xi^{n-\frac{1}{2}}\Big) = \frac{1}{2\Delta t}\Big(\xi^{n+1} - \xi^{n-1}\Big).$$

Assim, observando que

$$\begin{aligned}\delta^2\xi^n &= \frac{1}{(\Delta t)^2}\Big(\xi^{n+1} - 2\xi^n + \xi^{n-1}\Big) = \frac{1}{(\Delta t)^2}\Big((\xi^{n+1} - \xi^n) - (\xi^n - \xi^{n-1})\Big) \\ &= \frac{1}{\Delta t}\Big(\delta\xi^{n+\frac{1}{2}} - \delta\xi^{n-1/2}\Big),\end{aligned}$$

obtemos para a primeira parcela do lado esquerdo de (9.43),

$$\begin{aligned}(\delta^2\xi^n : v_h) &= \Big(\frac{1}{\Delta t}(\delta\xi^{n+\frac{1}{2}} - \delta\xi^{n-\frac{1}{2}}) : \frac{1}{2}(\delta\xi^{n+\frac{1}{2}} + \delta\xi^{n-1/2})\Big) \\ &= \frac{1}{2\Delta t}\Big(\|\delta\xi^{n+\frac{1}{2}}\|_0^2 - \|\delta\xi^{n-\frac{1}{2}}\|_0^2\Big).\end{aligned} \tag{9.44}$$

Para o terceiro termo de (9.43) e lembrando que a forma bilinear $a(\cdot,\cdot)$ define o produto escalar $(\cdot : \cdot)_a$ em $H_0^1(0,1)$, podemos escrever

$$\begin{aligned}(\star_\theta \xi^n : v_h)_a = \frac{1}{2\Delta t}\Big(\star_\theta \xi^n : \xi^{n+1} - \xi^{n-1}\Big)_a &= \frac{\theta}{2\Delta t}\Big(\xi^{n+1} : \xi^{n+1} - \xi^{n-1}\Big)_a \\ &+ \Big(\frac{1-2\theta}{2\Delta t}\Big)\Big(\xi^n : \xi^{n+1} - \xi^{n-1}\Big)_a + \frac{\theta}{2\Delta t}\Big(\xi^{n-1} : \xi^{n+1} - \xi^{n-1}\Big)_a.\end{aligned}$$

Somando e subtraindo o termo ξ^n convenientemente, obtemos

$$\begin{aligned}(\star_\theta \xi^n : v_h)_a &= \frac{1}{2\Delta t}\Big(\xi^n : \xi^{n+1} - \xi^{n-1}\Big)_a + \frac{\theta}{2\Delta t}\Big(\xi^{n+1} - 2\xi^n + \xi^{n-1} : \xi^{n+1} - \xi^{n-1}\Big)_a \\ &= \frac{1}{2\Delta t}\Big((\xi^{n+1} : \xi^n)_a - (\xi^n : \xi^{n-1})_a\Big) + \frac{\theta}{2\Delta t}\Big(\xi^{n+1} - \xi^n : \xi^{n+1} - \xi^n\Big)_a \\ &\quad + \frac{\theta}{2\Delta t}\Big(\xi^{n+1} - \xi^n : \xi^n - \xi^{n-1}\Big)_a - \frac{\theta}{2\Delta t}\Big(\xi^n - \xi^{n-1} : \xi^{n+1} - \xi^n\Big)_a \\ &\quad - \frac{\theta}{2\Delta t}\Big(\xi^n - \xi^{n-1} : \xi^n - \xi^{n-1}\Big)_a \\ &= \frac{1}{2\Delta t}\Big((\xi^{n+1} : \xi^n)_a - (\xi^n : \xi^{n-1})_a\Big) + \frac{\theta}{2\Delta t}\Big(\|\xi^{n+1} - \xi^n\|_a^2 - \|\xi^n - \xi^{n-1}\|_a^2\Big).\end{aligned} \tag{9.45}$$

Substituindo (9.44) e (9.45) em (9.43),

$$\begin{aligned}&\frac{1}{2\Delta t}\Big(\|\delta\xi^{n+\frac{1}{2}}\|_0^2 - \|\delta\xi^{n-\frac{1}{2}}\|_0^2\Big) + \frac{1}{2\Delta t}\Big((\xi^{n+1} : \xi^n)_a - (\xi^n : \xi^{n-1})_a\Big) \\ &\quad + \frac{\theta}{2\Delta t}\Big(\|\xi^{n+1} - \xi^n\|_a^2 - \|\xi^n - \xi^{n-1}\|_a^2\Big) \\ &\qquad = \frac{1}{2}\Big(g^n : \delta\xi^{n+\frac{1}{2}} + \delta\xi^{n-\frac{1}{2}}\Big) - \frac{1}{2}\Big(\delta^2\rho^n : \delta\xi^{n+\frac{1}{2}} + \delta\xi^{n-\frac{1}{2}}\Big).\end{aligned} \tag{9.46}$$

Observe que $a(\star_\theta \rho^n, v_h) = a(\star_\theta u^n - P_h \star_\theta u^n, v_h) = 0$, pois $u(t) - P_h u(t)$ é ortogonal ao subespaço V_m^k com relação ao produto interno $(\cdot : \cdot)_a$, para qualquer que seja o valor de t.

Multiplicando a identidade (9.46) por $2\Delta t$ e fazendo a soma em $n = 1, \ldots, k$, com $k \leq N - 1$, obtemos após simplificações evidentes,

$$\begin{aligned}&\|\delta\xi^{k-\frac{1}{2}}\|_0^2 - \|\delta\xi^{\frac{1}{2}}\|_0^2 + (\xi^k : \xi^{k-1})_a - (\xi^1 : \xi^0)_a + \theta\|\xi^k - \xi^{k-1}\|_a^2 \\ &\qquad - \theta\|\xi^1 - \xi^0\|_a^2 = \Delta t \sum_{n=1}^{k-1}\Big(g^n - \delta^2\rho^n : \delta\xi^{n+\frac{1}{2}} + \delta\xi^{n-\frac{1}{2}}\Big).\end{aligned} \tag{9.47}$$

Usando as desigualdades de Schwarz e Young para os termos $(\xi^1 : \xi^0)_a$ e $\theta\|\xi^1 - \xi^0\|_a^2$ na dentidade acima, obtemos

$$\begin{aligned}&\|\delta\xi^{k-\frac{1}{2}}\|_0^2 + (\xi^k : \xi^{k-1})_a + \theta\|\xi^k - \xi^{k-1}\|_a^2 \leq \|\delta\xi^{\frac{1}{2}}\|_0^2 + (2\theta + \frac{1}{2})(\|\xi^1\|_a^2 + \|\xi^0\|_a^2) \\ &\quad + \Delta t \sum_{n=1}^{k-1}\Big(g^n : \delta\xi^{n+\frac{1}{2}} + \delta\xi^{n-\frac{1}{2}}\Big) - \Delta t \sum_{n=1}^{k-1}\Big(\delta^2\rho^n : \delta\xi^{n+\frac{1}{2}} + \delta\xi^{n-\frac{1}{2}}\Big).\end{aligned} \tag{9.48}$$

Analisemos agora os dois termos envolvendo somatórios em (9.48). Usando a Desigualdade de Schwarz, o primeiro toma a forma

$$\sum_{n=1}^{k-1}\Big(g^n:\delta\xi^{n+\frac{1}{2}}+\delta\xi^{n-\frac{1}{2}}\Big)\leq\sum_{n=1}^{k-1}\|g^n\|_0\Big(\|\delta\xi^{n+\frac{1}{2}}\|_0+\|\delta\xi^{n-\frac{1}{2}}\|_0\Big).$$

Usando agora a desigualdade de Young, obtemos

$$\begin{aligned}\sum_{n=1}^{k-1}\|g^n\|_0\|\delta\xi^{n+\frac{1}{2}}\|_0&\leq\frac{1}{2\varepsilon}\sum_{n=1}^{k-1}\|g^n\|_0^2+\frac{\varepsilon}{2}\sum_{n=1}^{k-1}\|\delta\xi^{n+\frac{1}{2}}\|_0^2\\&\leq\frac{1}{2\varepsilon\Delta t}\|g\|^2_{L^2_0(0,T;L^2)}+\frac{\varepsilon}{2}\sum_{n=1}^{k-1}\|\delta\xi^{n-\frac{1}{2}}\|_0^2+\frac{\varepsilon}{2}\|\delta\xi^{k-\frac{1}{2}}\|_0^2,\end{aligned}$$

onde $\varepsilon>0$ é uma constante a ser definida.

Analogamente,

$$\sum_{n=1}^{k-1}\|g^n\|_0\|\delta\xi^{n-\frac{1}{2}}\|_0\leq\frac{1}{2\varepsilon\Delta t}\|g\|^2_{L^2_0(0,T;L^2)}+\frac{\varepsilon}{2}\sum_{n=1}^{k-1}\|\delta\xi^{n-\frac{1}{2}}\|_0^2.$$

Assim, temos a seguinte estimativa para a primeira parcela que envolve somatório em (9.48),

$$\sum_{n=1}^{k-1}\Big(g^n:\delta\xi^{n+\frac{1}{2}}+\delta\xi^{n-\frac{1}{2}}\Big)\leq\frac{1}{\varepsilon\Delta t}\|g\|^2_{L^2_0(0,T;L^2)}+\varepsilon\sum_{n=1}^{k-1}\|\delta\xi^{n-\frac{1}{2}}\|_0^2+\frac{\varepsilon}{2}\|\delta\xi^{k-\frac{1}{2}}\|_0^2.\quad(9.49)$$

Para o segundo termo com o somatório em (9.48), usando a desigualdade de Young e (8.43), obtém-se que

$$\begin{aligned}-\sum_{n=1}^{k-1}(\delta^2\rho^n:\delta\xi^{n+\frac{1}{2}}-\delta\xi^{n-\frac{1}{2}})&\leq\frac{1}{2\varepsilon}\sum_{n=1}^{k-1}\|\delta^2\rho^n\|_0^2+\frac{\varepsilon}{2}\sum_{n=1}^{k-1}\|\delta\xi^{n+\frac{1}{2}}\|_0^2\\&\quad+\frac{1}{2\varepsilon}\sum_{n=1}^{k-1}\|\delta^2\rho^n\|_0^2+\frac{\varepsilon}{2}\sum_{n=1}^{k-1}\|\delta\xi^{n-\frac{1}{2}}\|_0^2\\&\leq\frac{1}{\varepsilon}\sum_{n=1}^{k-1}\|\delta^2\rho^n\|_0^2+\varepsilon\sum_{n=1}^{k-1}\|\delta\xi^{n-\frac{1}{2}}\|_0^2+\frac{\varepsilon}{2}\|\delta\xi^{k-\frac{1}{2}}\|_0^2\\&\leq\frac{1}{\varepsilon\Delta t}\|\rho_{tt}\|^2_{L^2_0(0,T;L^2)}+\varepsilon\sum_{n=1}^{k-1}\|\delta\xi^{n-\frac{1}{2}}\|_0^2+\frac{\varepsilon}{2}\|\delta\xi^{k-\frac{1}{2}}\|_0^2.\end{aligned}\quad(9.50)$$

Substituindo as desigualdades (9.49) e (9.50) em (9.48), obtém-se

$$(1-\varepsilon\Delta t)\|\delta\xi^{k-\frac{1}{2}}\|_0^2+(\xi^k:\xi^{k-1})_a+\theta\|\xi^k-\xi^{k-1}\|_a^2\leq\Psi^2+2\varepsilon\Delta t\sum_{n=1}^{k-1}\|\delta\xi^{n-\frac{1}{2}}\|_0^2, \quad (9.51)$$

onde

$$\Psi^2:=\frac{1}{\varepsilon}\|\rho_{tt}\|^2_{L^2_0(0,T;L^2)}+\frac{1}{\varepsilon}\|g\|^2_{L^2_0(0,T;L^2)}+\|\delta\xi^{1/2}\|_0^2+(2\theta+\frac{1}{2})(\|\xi^1\|_a^2+\|\xi^0\|_a^2). \quad (9.52)$$

Sendo $(\cdot:\cdot)_a$ um produto interno em $H_0^1(0,1)$, vale a *Identidade Polar*,

$$(\xi^k:\xi^{k-1})_a=\frac{1}{4}\Big(\|\xi^k+\xi^{k-1}\|_a^2-\|\xi^k-\xi^{k-1}\|_a^2\Big),$$

que substituída em (9.51) nos fornece

$$\begin{aligned}(1-\varepsilon\Delta t)\|\delta\xi^{k-\frac{1}{2}}\|_0^2+\frac{1}{4}\|\xi^k+\xi^{k-1}\|_a^2+\left(\theta-\frac{1}{4}\right)\|\xi^k-\xi^{k-1}\|_a^2\\ \leq\Psi^2+2\varepsilon\Delta t\sum_{n=1}^{k-1}\|\delta\xi^{n-\frac{1}{2}}\|_0^2,\end{aligned} \quad (9.53)$$

para todo $k=1,2,\ldots,N$.

Para usar o Lema 8.2, faz-se necessário supor que as constantes no lado esquerdo de (9.53) sejam positivas. Como $\varepsilon>0$ foi fixado arbitrariamente, vamos fixá-lo de forma que $1-\varepsilon\Delta t>0$. Temos então dois casos a considerar, a saber: $\theta>1/4$ e $\theta=1/4$.

Caso 1: $\theta>1/4$. Para $\varepsilon<1/\Delta t$ fixado, definimos

$$C_\theta:=\min\Big\{1-\varepsilon\Delta t,\ \theta-1/4,\ 1/4\Big\}>0.$$

Então, segue de (9.53) a seguinte estimativa:

$$\|\delta\xi^{k-\frac{1}{2}}\|_0^2+\|\xi^k+\xi^{k-1}\|_a^2+\|\xi^k-\xi^{k-1}\|_a^2\leq\frac{1}{C_\theta}\left(\Psi^2+2\varepsilon\Delta t\sum_{n=1}^{k-1}\|\delta\xi^{n-\frac{1}{2}}\|_0^2\right).$$

Observando que $\|\xi^k\|_a^2+\|\xi^{k-1}\|_a^2\leq\|\xi^k+\xi^{k-1}\|_a^2+\|\xi^k-\xi^{k-1}\|_a^2$, temos

$$\|\delta\xi^{k-\frac{1}{2}}\|_0^2+\|\xi^k\|_a^2+\|\xi^{k-1}\|_a^2\leq\frac{1}{C_\theta}\left(\Psi^2+2\varepsilon\Delta t\sum_{n=1}^{k-1}\|\delta\xi^{n-\frac{1}{2}}\|_0^2.\right). \quad (9.54)$$

Assim, denotando

$$\varphi^k:=\|\delta\xi^{k-\frac{1}{2}}\|_0^2+\|\xi^k\|_a^2+\|\xi^{k-1}\|_a^2,\quad \psi=\Psi^2/C_\theta\quad\text{e}\quad C=\frac{2\varepsilon}{C_\theta},$$

a desigualdade (9.54) toma a formac $\varphi^k \leq \psi + C\Delta t \sum_{n=1}^{k-1} \varphi^n$, de modo que, pelo Lema 8.2, temos

$$\|\delta\xi^{k-\frac{1}{2}}\|_0^2 + \|\xi^k\|_a^2 + \|\xi^{k-1}\|_a^2 \leq \frac{e^{Ct_k}}{C_\theta}\Psi^2, \quad \forall\, k = 1, 2, \ldots, N. \tag{9.55}$$

Passando ao máximo em $k \in \{1, 2, \ldots, N\}$, obtemos

$$\max_{k\leq 1\leq N} \|\delta\xi^{k-\frac{1}{2}}\|_0^2 + \|\xi^k\|_a^2 + \|\xi^{k-1}\|_a^2 \leq \frac{e^{CT}}{C_\theta}\Psi^2, \tag{9.56}$$

e da definição das normas discretas, obtemos

$$\|\xi\|_{L_0^\infty(0,T;H_0^1)} + \|\delta\xi\|_{L_{1/2}^\infty(0,T;L^2)} \leq \frac{e^{CT}}{C_\theta}\Psi^2.$$

Lembrando que $e = \xi + \rho$, segue da desigualdade triangular,

$$\|e\|_{L_0^\infty(0,T;H_0^1)} + \|\delta e\|_{L_{1/2}^\infty(0,T;L^2)} \leq \frac{e^{CT}}{C_\theta}\Psi^2 + \|\rho\|_{L_0^\infty(0,T;H_0^1)} + \|\delta\rho\|_{L_{1/2}^\infty(0,T;L^2)}.$$

Sabendo que $\|\rho\|_{L_0^\infty(0,T;H_0^1)} \leq c_1 h$ e $\|\delta\rho\|_{L_{1/2}^\infty(0,T;L^2)} \leq c_2 h^2$, e que $e = u - u_h$, temos

$$\|u - u_h\|_{L_0^\infty(0,T;H_0^1)} + \|\delta(u - u_h)\|_{L_{1/2}^\infty(0,T;L^2)} \leq C(\Psi^2 + h + h^2), \tag{9.57}$$

restando-nos então estimar o termo Ψ^2 para concluir a prova da primeira desigualdade de (9.39) no caso $\theta > 1/4$.

Estimativa de Ψ^2: (1) Lembrando que $\xi(t) = P_h u(t) - u_h(t)$, segue do Lema (9.1) que existe uma constante $c_1 > 0$ tal que

$$\|\delta\xi^{1/2}\|_0^2 + (2\theta + \frac{1}{2})(\|\xi^1\|_a^2 + \|\xi^0\|_a^2) \leq c_1\left(h + (\Delta t)^2\right). \tag{9.58}$$

(2) Além disso, segue de (8.10) para $m = 0$ e $k = 1$,

$$\|\rho''(t)\|_0 \leq ch^2\|u''(t)\|_2, \quad \forall t \in [0, T],$$

de modo que

$$\frac{1}{\varepsilon}\|\rho_{tt}\|_{L_0^2(0,T;L^2)}^2 \leq c_2 h^4 \|u''\|_{L_0^\infty(0,T;H^2)}^2. \tag{9.59}$$

(3) Para concluir, precisamos estimar a parcela $\|g\|_{L_0^2(0,T;L^2)}^2$.

A função g^n, definida em (9.41), é dada por $g^n = \delta^2 u^n - \star_\theta u_{tt}^n$. Usando a definição do operador $*_\theta$, temos

$$g^n = \delta^2 u^n - (\theta u_{tt}^{n+1} + (1 - 2\theta)u_{tt}^n + \theta u_{tt}^{n-1} = (\delta^2 u^n - u_{tt}^n) - \theta(u_{tt}^{n+1} - 2u_{tt}^n + u_{tt}^{n-1}).$$

Após sucessivas integrações por partes, vemos que

$$\delta^2 u^n - u^n_{tt} = \frac{1}{(\Delta t)^2}\left(\int_{t_n}^{t_{n+1}} \frac{(t_{n+1}-s)^3}{3!} u_{tttt}(s)\,ds + \int_{t_{n-1}}^{t_n} \frac{(s-t_{n-1})^3}{3!} u_{tttt}(s)\,ds\right).$$

De forma análoga, integrando por partes duas vezes, obtém-se

$$u^{n+1}_{tt} - 2u^n_{tt} + u^{n-1}_{tt} = \int_{t_n}^{t_{n+1}} (t_{n+1}-s) u_{tttt}(s)\,ds + \int_{t_{n-1}}^{t_n} (s-t_{n-1}) u_{tttt}(s)\,ds.$$

Segue da definição de g^n,

$$\begin{aligned} g^n &= \frac{1}{(\Delta t)^2}\left(\int_{t_n}^{t_{n+1}} \frac{(t_{n+1}-s)^3}{3!} u_{tttt}(s)\,ds + \int_{t_n}^{t_{n-1}} \frac{(s-t_{n-1})^3}{3!} u_{tttt}(s)\,ds\right) \\ &\quad - \theta\left(\int_{t_n}^{t_{n+1}} (t_{n+1}-s) u_{tttt}(s)\,ds + \int_{t_n}^{t_{n-1}} (s-t_{n-1}) u_{tttt}(s)\,ds\right) \\ &= \Phi_1 + \Phi_2 - (\Phi_3 + \Phi_4). \end{aligned}$$

Para estimar $\|g^n\|_0^2$, vamos analisar cada um dos termos Φ_i definidos acima. Da definição de Φ_1 e do fato de que $0 \le t_{n+1} - s \le \Delta t$, para $s \in [t_n, t_{n+1}]$, temos

$$|\Phi_1| \le \frac{\Delta t}{3!}\int_{t_n}^{t_{n+1}} |u_{tttt}(\cdot, s)|\,ds.$$

Pela desigualdade de Schwarz, temos

$$|\Phi_1|^2 \le \frac{(\Delta t)^2}{(3!)^2}\left(\int_{t_n}^{t_{n+1}} |u_{tttt}(\cdot,s)|\,ds\right)^2 \le \frac{(\Delta t)^3}{(3!)^2}\int_{t_n}^{t_{n+1}} |u_{tttt}(\cdot,s)|^2\,ds.$$

Pelo Teorema de Fubini, temos

$$\|\Phi_1\|_0^2 = \int_0^1 |\Phi_1|^2\,dx \le \frac{(\Delta t)^3}{(3!)^2}\|u_{tttt}(s)\|^2_{L^2(t_n,t_{n+1};L^2)}$$

e de forma análoga, temos

$$\|\Phi_2\|_0^2 = \int_0^1 |\Phi_1|^2\,dx \le \frac{(\Delta t)^3}{(3!)^2}\|u_{tttt}(s)\|^2_{L^2(t_{n-1},t_n;L^2)}.$$

Por outro lado, da definição de Φ_3, temos

$$|\Phi_3| \le \theta\Delta t\int_{t_n}^{t_{n+1}} |u_{tttt}(\cdot,s)|\,ds \quad \Rightarrow \quad |\Phi_3|^2 \le \theta^2(\Delta t)^3\int_{t_n}^{t_{n+1}} |u_{tttt}(\cdot,s)|^2\,ds$$

e consequentemente,

$$\|\Phi_3\|_0^2 \leq \theta^2(\Delta t)^3\|u_{tttt}(s)\|^2_{L^2(t_n,t_{n+1};L^2)}.$$

Da mesma forma,

$$\|\Phi_4\|_0^2 \leq \theta^2(\Delta t)^3\|u_{tttt}(s)\|^2_{L^2(t_{n-1},t_n;L^2)}.$$

Com isso concluímos que

$$\|g^n\|_0^2 \leq c_3(\Delta t)^3\Big(\|u_{tttt}(s)\|^2_{L^2(t_n,t_{n+1};L^2)} + \|u_{tttt}(s)\|^2_{L^2(t_{n-1},t_n;L^2)}\Big),$$

onde $c_3 = 4(\theta^2 + (1/3!)^2)$. Assim, multiplicando ambos os lados da desigualdade acima por Δt e somando em $n = 1, \ldots, N-1$, obtemos

$$\|g\|^2_{L^2_0(0,T;L^2)} = \Delta t\sum_{n=1}^{N-1}\|g^n\|_0^2 \leq 2c_3(\Delta t)^4\|u_{tttt}\|^2_{L^2(0,T;L^2(0,1))},$$

Logo,

$$\frac{1}{\varepsilon}\|g\|^2_{L^2_0(0,T;L^2)} \leq c_4(\Delta t)^4\|u_{tttt}\|^2_{L^2(0,T;L^2(0,1))}. \tag{9.60}$$

Portanto, de (9.58), (9.59) e (9.60), obtemos a desigualdade

$$\Psi^2 \leq C\left(h + (\Delta t)^2 + h^4 + (\Delta t)^4\right) \leq \widehat{C}\left(h + (\Delta t)^2\right)$$

que substituída em (9.57), fornece a estimativa (9.39) no caso $\theta > 1/4$.

Caso 2: $\theta = 1/4$. Consideremos a formulação (9.43) e como anteriormente, tomemos

$$v_h = \frac{1}{2}\Big(\delta\xi^{n+\frac{1}{2}} + \delta\xi^{n-\frac{1}{2}}\Big) = \frac{1}{2\Delta t}\Big(\xi^{n+1} - \xi^{n-1}\Big).$$

Então, o primeiro termo de (9.44) não se altera. Entretanto, multiplicando o segundo termo por $2\Delta t$, obtemos

$$a(\star_\theta\xi^n, \xi^{n+1} - \xi^{n-1}) = \Big(\xi^{n+1/2} + \xi^{n-1/2} : \xi^{n+1/2} - \xi^{n-1/2}\Big)_a = \|\xi^{n+1/2}\|_a^2 - \|\xi^{n-1/2}\|_a^2. \tag{9.61}$$

Assim, multiplicando (9.43) por $2\Delta t$ e em seguida, substituindo (9.61) e (9.44), obtemos após somar em n de 1 até k, $k \leq N-1$,

$$\|\delta\xi^{k-1/2}\|_0^2 + \|\xi^{k-1/2}\|_a^2 = \|\delta\xi^{1/2}\|_0^2 + \|\xi^{1/2}\|_a^2 + \Delta t\sum_{n=1}^{k-1}\Big(g^n - \delta^2\rho^n : \delta\xi^{n+\frac{1}{2}} + \delta\xi^{k-1/2}\Big)_0 \tag{9.62}$$

Com os mesmos cálculos se aplicam para obter (9.49) e (9.50), isto é

$$\Delta t\sum_{n=1}^{k-1}\left(g^n:\delta\xi^{n+\frac{1}{2}}+\delta\xi^{n-1/2}\right)\leq\frac{1}{\varepsilon}\|g\|^2_{L^2_0(0,T;L^2)}+\varepsilon\Delta t\sum_{n=1}^{k-1}\|\delta\xi^{n-1/2}\|_0^2 + \frac{\varepsilon\Delta t}{2}\|\delta\xi^{k-1/2}\|_0^2,$$

$$-\Delta t\sum_{n=1}^{k-1}(\delta^2\rho^n:\delta\xi^{n+\frac{1}{2}}+\delta\xi^{n-\frac{1}{2}})\leq\frac{1}{\varepsilon}\|\rho''\|^2_{L^2_0(0,T;L^2)}+\varepsilon\Delta t\sum_{n=1}^{k-1}\|\delta\xi^{n-\frac{1}{2}}\|_0^2 + \frac{\varepsilon\Delta t}{2}\|\delta\xi^{k-\frac{1}{2}}\|_0^2,$$

obtemos a seguinte limitação, após substitur os dois últimos termos acima em (9.62),

$$(1-\varepsilon\Delta t)\|\delta\xi^{k-\frac{1}{2}}\|_0^2+\|\xi^{k-\frac{1}{2}}\|_a^2\leq\|\delta\xi^{1/2}\|_0^2+\|\xi^{1/2}\|_a^2+\frac{1}{\varepsilon}\|g\|^2_{L^2_0(0,T;L^2)} + \frac{1}{\varepsilon}\|\rho''\|^2_{L^2_0(0,T;L^2)}+2\varepsilon\Delta t\sum_{n=1}^{k-1}\|\delta\xi^{n-\frac{1}{2}}\|_0^2.$$

Fixando $\varepsilon<1/\Delta t$, denotando $C_{1/4}:=(1-\Delta t\varepsilon)>0$ e desprezando o termo positivo $\|\xi^{k-1/2}\|_a^2$, obtemos

$$\|\delta\xi^{k-\frac{1}{2}}\|_0^2+\|\xi^{k-\frac{1}{2}}\|_a^2\leq\Psi_2+c_1\Delta t\sum_{n=1}^{k-1}\|\delta\xi^{n-\frac{1}{2}}\|_0^2, \tag{9.63}$$

para todo $k\in\{1,2,\ldots,N\}$, onde

$$\Psi^2:=c_1\left(\|\delta\xi^{1/2}\|_0^2+\|\xi^{1/2}\|_a^2+\|g\|^2_{L^2_0(0,T;L^2)}+\|\rho''\|^2_{L^2_0(0,T;L^2)}\right)$$

e $c_1=\max\{C_{1/4},\,1/\varepsilon\}$. Aplicando o Lema 8.2 (o análogo discreto do Lema de Gronwal), obtemos

$$\|\delta\xi^{N-\frac{1}{2}}\|_0^2\leq\Psi^2\exp(c_1T),$$

de onde se deduz, com os mesmos argumentos usados em (1),(2) e (3) no parágrafo sobre a estimativa de Ψ^2, a estimativa

$$\|\delta\xi^{N-\frac{1}{2}}\|_0^2+\|\xi^{N-\frac{1}{2}}\|_a^2\leq c_2\left(h+(\Delta t)^2\right)^2,$$

com o que conclímos a demonstração. □

O próximo teorema estabelece o erro da solução do θ-método para o problema dado pelo sistema (7.26).

Teorema 9.4. *Sob a hipótese (9.38), se* $u_0 \in H_0^1(0,1) \cap H^2(0,1))$ *e* $u_1 \in H_0^1(0,1)$, *a solução aproximada* $\{w_h, u_h\}$ *do Problema (7.21) obtida pelo* θ*-método (7.26) com* $\min\{\sigma,\theta\} \geq 1/2$, *satisfaz a seguinte estimativa:*

$$\|w - w_h\|_{L^\infty(0,T;L^2(0,1))} + \|u - u_h\|_{L^\infty(0,T;H_0^1(0,1))} \leq C\Big(h + (\Delta t)^p\Big), \tag{9.64}$$

onde C *é uma constante positiva independente de* h *e de* Δt *e*

$$p = \begin{cases} 2 & se \;\; \theta = \sigma = 1/2, \\ 1 & se \;\; \min\{\theta,\sigma\} > 1/2. \end{cases}$$

Demonstração: A formulação fraca do problema é

$$(u''(t) : v) + a(u(t), v) = (f(t) : v), \quad \forall v \in H_0^1(0,1),$$

de onde se obtém as identidades

$$(w'(t) : v) + a(u(t), v) = (f(t) : v) \quad \text{e} \quad (u'(t) : v) - (w(t) : v) = 0,$$

fazendo-se a mudança de variável $w(t) = u'(t)$. Assim, para os tempos discretos $t_{n+1} = (n+1)\Delta t$ e $t_n = n\Delta t$, com pesos θ, $(1-\theta)$ e σ, $(1-\sigma)$ respectivamente na primeira e na segunda equação, obtemos, somando-se as equações resultantes,

$$(w_t^{n+\theta} : v) + a(u^{n+\theta}, v) = (f^{n+\theta} : v) \quad \text{e} \quad (u_t^{n+\sigma} : v) - (w^{n+\sigma} : v) = 0, \quad \forall v \in H_0^1(0,1). \tag{9.65}$$

Somando e subtraindo $\delta w^{n+\frac{1}{2}}$ e $\delta u^{n+\frac{1}{2}}$ respectivamente na primeira e na segunda equação de (9.65), obtemos, para todo $v \in H_0^1(0,1)$,

$$\begin{cases} (\delta w^{n+\frac{1}{2}} : v) + a(u^{n+\theta}, v) = (f^{n+\theta} + g^n : v), \\ (\delta u^{n+\frac{1}{2}} : v) - (w^{n+\sigma} : v) = (\lambda^n : v_h), \end{cases} \tag{9.66}$$

onde $g^n = \delta w^{n+\frac{1}{2}} - w_t^{n+\theta}$ e $\lambda^n = \delta u^{n+\frac{1}{2}} - u_t^{n+\sigma}$.

Por outro lado, para o problema aproximado, a formulação é dada por

$$\begin{cases} (\delta w_h^{n+\frac{1}{2}} : v_h) + a(u_h^{n+\theta}, v_h) = (f^{n+\theta} : v_h), \\ (\delta u_h^{n+\frac{1}{2}} : v_h) - (w_h^{n+\sigma} : \varphi_h) = 0, \end{cases} \quad \forall v_h \in V_m^k. \tag{9.67}$$

Tomando em particular $v = v_h$ em (9.66) e subtraindo a equação em (9.67) daquela em (9.66), obtemos

$$\begin{cases} (\delta e_1^{n+\frac{1}{2}} : v_h) + a(e_2^{n+\theta}, v_h), = (g^n : v_h) \\ (\delta e_2^{n+\frac{1}{2}} : v_h) - (e_1^{n+\sigma} : v_h) = (\lambda^n : v_h), \end{cases} \quad \forall v_h \in V_m^k, \tag{9.68}$$

onde denotamos $e_1 = w - w_h$ e $e_2 = u - u_h$.

Se tomarmos em particular $v_h = (e_2^{n+1} - e_2^n)$ na primeira equação de (9.68), $v_h = (e_1^{n+1} - e_1^n)$ na segunda e subtrairmos a segunda da primeira, obtemos após as necessárias e evidentes simplificações,

$$a(e_2^{n+\theta}, e_2^{n+1} - e_2^n) + (e_1^{n+\sigma} : e_1^{n+1} - e_1^n) = (g^n : e_2^{n+1} - e_2^n) - (\lambda^n : e_1^{n+1} - e_1^n). \quad (9.69)$$

Observando que $\chi^{n+\theta} = \chi^{n+1/2} + (\theta - \frac{1}{2})(\chi^{n+1} - \chi^n)$, podemos reescreer a equação (9.69) na forma

$$\begin{aligned} a(e_2^{n+\frac{1}{2}}, e_2^{n+1} - e_2^n) + \Big(\theta - \frac{1}{2}\Big)\|e_2^{n+1} - e_2^n\|_a^2 + (e_1^{n+\frac{1}{2}} : e_1^{n+1} - e_1^n) \\ + \Big(\sigma - \frac{1}{2}\Big)\|e_1^{n+1} - e_1^n\|_0^2 \;=\; (g^n : e_2^{n+1} - e_2^n) - (\lambda^n : e_1^{n+1} - e_1^n). \end{aligned}$$

Lembrando que $e_i^{n+\frac{1}{2}} = (e_i^{n+1} + e_i^n)/2$, obtemos da identidade acima multiplicada por 2 e, para quaisquer que sejam as constantes positivas γ_1 e γ_2,

$$\begin{aligned} &\|e_2^{n+1}\|_a^2 - \|e_2^n\|_a^2 + 2(\theta - 1/2)\|e_2^{n+1} - e_2^n\|_a^2 \\ &+\|e_1^{n+1}\|^2 - \|e_1^n\|^2 + 2(\sigma - 1/2)\|e_1^{n+1} - e_1^n\|^2 \\ &\le \frac{1}{\gamma_2}\|g^n\|^2 + \frac{1}{\gamma_1}\|\lambda^n\|^2 + \gamma_2\|e_2^{n+1} - e_2^n\|_a^2 + \gamma_1\|e_1^{n+1} - e_1^n\|^2. \end{aligned} \quad (9.70)$$

Assim, com a escolha $\gamma_1 = (\sigma - 1/2)$ e $\gamma_2 = (\theta - 1/2)$ que são estritamente positivas se $\min\{\theta, \sigma\} > 1/2$, obtemos

$$\|e_2^{n+1}\|_a^2 - \|e_2^n\|_a^2 + \|e_1^{n+1}\|_0^2 - \|e_1^n\|_0^2 \le \frac{1}{\gamma_2}\|g^n\|^2 + \frac{1}{\gamma_1}\|\lambda^n\|^2.$$

Somando em n de 1 a $N-1$, segue que

$$\|e_2^N\|_a^2 + \|e_1^N\|_0^2 \le \|e_2^0\|_a^2 + \|e_1^0\|_0^2 + \frac{1}{\gamma_2}\sum_{n=0}^{N-1}\|g^n\|^2 + \frac{1}{\gamma_1}\sum_{n=0}^{N-1}\|\lambda^n\|^2. \quad (9.71)$$

Sabemos de (8.105) que

$$\begin{aligned} \|g\|^2_{L_0^2(0,T;L^2)} &= \Delta t \sum_{n=1}^{N-1}\|g^n\|_0^2 \\ &\le c_a(\Delta t)^4\|w'''\|^2_{L^2(0,T;H^2)} + 2\Big(\theta - \frac{1}{2}\Big)^2(\Delta t)^2\|w''\|^2_{L^2(0,T;H^2)}, \\ \|\lambda\|^2_{L_0^2(0,T;H_0^1)} &= \Delta t \sum_{n=1}^{N-1}\|\lambda^n\|_1^2 \\ &\le c_b(\Delta t)^4\|u'''\|^2_{L^2(0,T;H^2)} + 2\Big(\sigma - \frac{1}{2}\Big)^2(\Delta t)^2\|u''\|^2_{L^2(0,T;H^2)} \end{aligned} \quad (9.72)$$

e que para os dados iniciais, temos ao menos a seguinte estimativa:

$$\|e_2^0\|_a^2 + \|e_1^0\|_0^2 \leq c_3 h^2.$$

Assim, substituindo os dados acima em (9.71) e majorando as constantes por $C > 0$ adequada (por exemplo, a maior das contantes), temos

$$\|e_2^N\|_a^2 + \|e_1^N\|_0^2 \leq C\Big(h^2 + (\Delta t)^4 + (\sigma + \theta - 1)(\Delta t)^2\Big).$$

Como $(a+b)^2 \leq 2(a^2+b^2)$, segue da desigualdade acima,

$$\begin{aligned}\|e_2^N\|_a + \|e_1^N\|_0 &\leq \sqrt{2C}\Big(h^2 + (\Delta t)^4 + (\sigma + \theta - 1)(\Delta t)^2\Big)^{1/2} \\ &\leq \sqrt{2C}\Big(h + (\Delta t)^2 + \sqrt{\sigma + \theta - 1}\Delta t\Big)^{1/2}.\end{aligned}$$

Portanto, $\|e_2^N\|_a + \|e_1^N\|_0$ é de ordem $\mathcal{O}(h + \Delta t)$. Observe que se $\sigma = \theta = 1/2$, a parcela em Δt na desigualdade acima desaparece, e teríamos a estimativa $\mathcal{O}(h + (\Delta t)^2)$. Entretanto, para a validade do argumento usado até aqui, no qual fizemos uso da desigualdade de Hölder em (9.70), fez-se necessária a hipótese $\min\{\sigma, \theta\} > 1/2$. Entretanto, se $\theta = \sigma = 1/2$ a equação (9.69) se simplifica e as estimativas decorrem de forma mais direta. □

9.2.1 Exercícios

1. Prove que o método da diferença central ($\theta = 0$) em (9.30) é condicionalmente estável, ou seja, $\Delta t/h \leq \delta$ com δ suficientemente pequeno.

2. Prove com detalhes, a estimativa de erro (9.39) na norma $L^2(0,1)$ (ou seja, o caso (2)) do problema totalmente discreto.

CAPÍTULO 10

Análise Matemática

Nesse capítulo abordaremos algumas questões básicas sobre a análise matemática das equações do calor e da onda, tais como existência, unicidade e propriedades gerais das respectivas soluções. Embora essas propriedades sejam importantes até para o estudo das soluções numéricas, não é impressindível numa primeira abordagem saber provás-las, razão pela qual os tópicos tratados neste capítulo não precisam ser abordados num primeiro curso sobre os métodos numéricos em Equações Diferenciais Parciais.

10.1 Equação do Calor

No que segue demonstraremos a existência de solução do problema variacional (6.1) pelo método de Galerkin, também conhecido como método de Faedo-Galerkin (ver [13]), a unicidade e, na sequência, algumas propriedades gerais da solução.

10.1.1 Existência e unicidade de solução

Teorema 10.1. *Se $f \in L^2(0,T;L^2(0,1))$ e $u_0 \in H_0^1(0,1)$, o problema (6.1) admite uma única solução $u : [0,T] \times (0,1) \to \mathbb{R}$, satisfazendo as seguintes condições:*

$$\begin{array}{ll} \text{(a)} & u \in L^2(0,T;H_0^1(0,1)) \cap C([0,T];L^2(0,1)), \\ \text{(b)} & u' \in L^2(0,T;L^2(0,1)), \\ \text{(c)} & (u'(t):v) + a(u(t),v) = (f:v), \quad \forall v \in H_0^1(0,1), \\ \text{(d)} & u(0) = u_0. \end{array} \tag{10.1}$$

Demonstração: No que se refere à existência, a demonstração segue o mesmo procedimento geral já utilizado na construção da solução numérica, mas o cerne da questão é, como veremos a seguir, o processo de limite das soluções aproximadas.

Como $H_0^1(0,1)$ é um espaço de Hilbert, consideremos $\{w_i\}_{i\in\mathbb{N}}$ uma base hilbertiana (cuja existência é consequência do fato de $H_0^1(0,1)$ ser separável). Seja V_m o subespaço de $H_0^1(0,1)$ gerado pelos m primeiros vetores dessa base, isto é $V_m = [w_1,\ldots,w_m]$. Para cada $t\in[0,T]$, seja $u_m(t)\in V_m$ definida por

$$u_m(t) := \sum_{i=1}^{m} g_{im}(t)w_i, \tag{10.2}$$

onde o coeficiente $\boldsymbol{g}_m(t) := (g_{1m}(t),\ldots,g_{mm}(t))$ é a solução do problema de valor inicial para o seguinte sistema de equações diferenciais:

$$\begin{cases} (u_m'(t):v) + a(u_m(t),v) = (f(t):v),\ \forall\, v\in V_m, \\ u_m(0) = u_{0m}, \end{cases} \tag{10.3}$$

com a condição inicial u_{0m} tal que a sequência $\{u_{0m}\}_{m\in\mathbb{N}}$ seja convergente para u_0 em $H_0^1(0,1)$, isto é,

$$u_{0m} \longrightarrow u_0 \quad \text{em} \quad H_0^1(0,1) \quad \text{quando} \quad m\to\infty.$$

A idéia da demonstração consiste em mostrar que existe uma sequência de soluções $\{u_m\}_{m\in\mathbb{N}}$ do problema aproximado (10.3) pertencentes ao subespaço V_m que são limitadas independentemente de m, possibilitando assim mostrar que ela admite uma subsequência que converge para uma solução exata $u(x,t)$. Mais precisamente, substituindo (10.2) em (10.3), obtemos

$$\Big(\sum_{i=1}^{m} g_{im}'(t)w_i : v\Big) + a\Big(\sum_{i=1}^{m} g_{im}(t)w_i, v\Big) = (f(t):v),\ \forall v\in V_m,$$

ou de forma equivalente,

$$\sum_{i=1}^{m} g_{im}'(t)(w_i:v) + \sum_{i=1}^{m} g_{im}(t)a(w_i,v) = (f(t):v),\ \forall v\in V_m.$$

Sendo $v\in V_m$ arbitrário, podemos tomar $v=w_j$, de onde resulta

$$\sum_{i=1}^{m} g_{im}'(t)(w_i:w_j) + \sum_{i=1}^{m} g_{im}(t)a(w_i,w_j) = (f(t):w_j).$$

Temos assim definidas as matrizes A, B e o vetor $\boldsymbol{F}(t)$, cujos coeficientes são definidos respectivamente por

$$A_{ij} = (w_i:w_j), \quad B_{ij} = a(w_i,w_j) \quad \text{e} \quad F_j(t) = (f(t):w_j),$$

o que nos permite escrever o seguinte sistema de equações diferenciais ordinárias

$$\sum_{i=1}^{m} g'_{im}(t)\, a_{ij} + \sum_{i=1}^{m} g_{im}(t)\, b_{ij} = F_j, \quad \text{para} \quad j = 1, \cdots, m$$

que pode ser escrito na seguinte forma matricial:

$$\begin{cases} A\boldsymbol{g}'_m(t) + B\boldsymbol{g}_m(t) = \boldsymbol{F}(t), \quad \forall t \in [0,T] \\ \boldsymbol{g}_m(0) = \Big((u_0, w_1), (u_0, w_2), \ldots, (u_0, w_m)\Big) = \boldsymbol{g}_{0m} \end{cases} \tag{10.4}$$

onde $\boldsymbol{g}_m(0)$ é a condição inicial, $\boldsymbol{g}_m(t) = (g_{1m}(t), g_{2m}(t), \ldots, g_{mm}(t))^T$ é o vetor incógnita de um sistema linear de m equações diferenciais ordinárias.

Vale observar que A é uma matriz de Gram, portanto simétrica e invertível (já que as funções w_i são linearmente independentes em $L^2(0,1)$). Assim, para cada m, o sistema (10.4) tem uma única solução

$$\boldsymbol{g}_m \in H^1(0,T;\mathbb{R}^m) \cap C([0,T];\mathbb{R}^m),$$

o que nos assegura a existência da solução u_m do problema (10.3), a saber

$$\begin{aligned} &u_m \in H^1(0,T;V_m) \cap C([0,T];V_m), \\ &u_m(t) = \sum_{i=1}^{m} g_{im}(t) w_i(x), \quad \forall t \in [0,T], \end{aligned}$$

com $\boldsymbol{g}_m$ dado pela fórmula de variação dos parâmetros, isto é,

$$\boldsymbol{g}_m(t) = \exp(-tA^{-1}B)\boldsymbol{g}_0 + \int_0^t \exp((t-s)A^{-1}B)A^{-1}B\boldsymbol{F}(s)\, ds.$$

O próximo passo é deteminar estimativas uniformes adequadas para as soluções u_m que nos permitam obter a convergência (eventualmente de uma subsequência) cujo limite u seja solução do problema (10.1).

Consideremos $v = u'_m(t) \in V_m$ em (10.3). Então,

$$(u'_m(t) : u'_m(t)) + a(u_m(t), u'_m(t)) = (f(t) : u'_m(t)). \tag{10.5}$$

É claro que $(u'_m(t) : u'_m(t)) = \|u'_m(t)\|_0^2$ e como a forma bilinear a é simétrica, temos

$$a(u_m(t), u'_m(t)) = \frac{1}{2}\frac{d}{dt} a(u_m(t), u_m(t)).$$

Usando sucessivamente as desigualdades de Schwarz e Young, obtemos

$$(f(t) : u'_m(t)) \leq \frac{1}{2}\Big(\|f(t)\|_0^2 + \|u'_m(t)\|_0^2\Big).$$

de modo que após substituição em (10.5), concluímos que

$$\|u'_m(t)\|_0^2 + \frac{d}{dt}a(u_m(t), u_m(t)) \leq \|f(t)\|_0^2, \quad \forall t \in [0, T].$$

Integrando de 0 a $t \leq T$, obtemos

$$\int_0^t \|u'_m(s)\|_0^2\, ds + a(u_m(t), u_m(t)) \leq a(u_m(0), u_m(0)) + \int_0^t \|f(s)\|_0^2\, ds.$$

Como a forma bilinear a é contínua em $H_0^1(0,1)$, i.e., $|a(u,v)| \leq C\|u\|_1\|v\|_1$ para todo $u, v \in H_0^1(0,1)$, e $u_m(0)$ é convergente nesse mesmo espaço, a sequência numérica $\{a(u_m(0), u_m(0))\}_{m\in\mathbb{N}}$ é limitada, de modo que podemos garantir a existência de uma constante positiva c_1 independente de m tal que

$$\int_0^t \|u'_m(s)\|_0^2\, ds + a(u_m(t), u_m(t)) \leq c_1, \quad \forall t \in [0, T],\ \forall m \in \mathbb{N},$$

o que nos permite concluir que

$$\begin{array}{ll} \text{(i)} & \{u_m\}_{m\in\mathbb{N}} \text{ limitada em } L^\infty(0, T; H_0^1(0,1)), \\ \text{(ii)} & \{u'_m\}_{m\in\mathbb{N}} \text{ limitada em } L^2(0, T; L^2(0,1)), \end{array} \tag{10.6}$$

O Teorema de Dunfort-Pettis nos garante que o espaço $L^\infty(0, T; H_0^1(0,1))$ é o dual topológico de $L^1(0, T; H^{-1}(0,1))$. Portanto, de (ii) e do Teorema de Banach-Alaoglu podemos extrair uma subsequência $\{u_{m_j}\}_{j\in\mathbb{N}}$ que converge fraco-$*$ para u em $L^\infty(0, T; H_0^1(0,1))$. Além disso, de (ii) e passando a uma nova subsequência se necessário, podemos garantir que $\{u'_{m_j}\}_{j\in\mathbb{N}}$ converge fraco para v em $L^2(0, T; L^2(0,1)) = L^2(Q)$, onde $Q = (0,1) \times (0,T)$.

Para mostrar que $v = u'$, basta observar que a convergência fraco-$*$ de u_m em $L^\infty(0, T; H_0^1(0,1))$, e consequentemente a convergência fraca em $L^2(0, T; L^2(0,1)))$ implica na convergência em $\mathcal{D}'(0, T; L^2(0,1))$ (sentido das distribuições vetoriais), que por sua vez nos garante que u'_m converge para u' no sentido das distribuições. Portanto, pelo Lema de Du Bois-Raymond (veja [13]), $v = u'$.

Temos assim respectivamente de (i) e (ii)

$$\text{(iii)}\ \ (u'_{m_j}(t) : v) \rightharpoonup (u'(t) : v) \text{ em } L^2(0,T),\ \forall v \in L^2(0,1),$$

$$\text{(iv)}\ \ a(u_{m_j}(t), v) \underset{*}{\rightharpoonup} a(u(t), v) \text{ em } L^\infty(0,T),\ \forall v \in H_0^1(0,1).$$

Multiplicando por $\theta \in \mathcal{D}(0,T)$ ambos os lados da equação (10.3) com $v = w_k$, obtemos

$$\int_0^T \Big[(u'_{m_j}(s) : w_k) + a(u_{m_j}(s), w_k) - (f(s) : w_k)\Big]\theta(s)\, ds = 0, \quad \forall m_j \geq k.$$

Passando ao limite quando m_j tende ao infinito, obtemos

$$\int_0^T \Big[(u'(s) : w_k) + a(u(s), w_k) - (f(s) : w_k)\Big]\theta(s)\, ds = 0.$$

Como k foi fixado arbitrariamente, temos

$$\int_0^T \Big[(u'(s) : v) + a(u(s), v) - (f(s) : v)\Big]\theta(s)\, ds = 0, \quad \forall v \in H_0^1(0,1).$$

Como todas as parcelas envolvendo o fator v na integral acima estão definidas em $L^2(0,T)$, segue da densidade de $\mathcal{D}(0,T)$ em $L^2(0,T)$,

$$(u'(t) : v) + a(u(t), v) = (f(t) : v) \quad \text{em} \quad L^2(0,T), \ \ \forall v \in H_0^1(0,1),$$

com o que concluímos o item (c) do Teorema 10.1. Por outro lado, segue dos itens (i) e (ii) acima,

$$u, u' \in L^2(0,T; L^2(0,1)) \quad \Rightarrow \quad u \in H^1(0,T; L^2(0,1)).$$

Como $H^1(0,T;L^2(0,1))$ está contido[1] em $C^{0,1/2}(0,T;L^2(0,1)) \subset C([0,T];L^2(0,1))$, temos os itens (a) e (b) do enunciado.

Vamos então mostrar que u satisfaz a condição inicial definida em (d). Como $u \in C([0,T];L^2(0,1))$, segue que $u(0)$ está bem definida em $L^2(0,1)$. Além disso, Como $u_{m_j}, u \in H^1(0,T;L^2(0,1)) \cap C([0,T];L^2(0,1))$, temos[2],

$$\left.\begin{aligned} u_{m_j}(t) &= u_0 + \int_0^t u'_{m_j}(\xi)\, d\xi, \\ u(t) &= u(0) + \int_0^t u'(\xi)\, d\xi, . \end{aligned}\right\} \quad \forall t \in [0,T].$$

Das identidades acima, obtemos para todo $v \in L^2(0,1)$,

$$(u(0) - u_0 : v) = (u_{m_j}(t) - u(t) : v) - \int_0^t (u'_{m_j}(\xi) - u'(\xi) : v)\, d\xi$$

[1] Se X é espaço de Benach reflexivo, tem-se $W^{1,p}(0,T;X) \subset C^{0,\alpha}([0,T];X)$, com $\alpha = (p-1)/p$.

[2] Se X é espaço de Banach, $1 \leq p \leq \infty$, então $f \in W^{1,p}(0,T;X)$ se, e somente se, para quase todo $t, s \in [0,T]$, $u(t) = u(s) + \int_s^t u'(\xi)\, d\xi$

Logo, para todo $v \in L^2(0,1)$,

$$|(u(0) - u_0 : v)| \leq |(u_{m_j}(t) - u(t) : v)| + \int_0^T |(u'_{m_j}(\xi) - u'(\xi) : v)|\, d\xi.$$

Fazendo $j \to +\infty$ e considerando as convergências obtidas acima, temos

$$\lim_{j \to +\infty} \left[|(u_{m_j}(t) - u(t) : v)| + \int_0^T |(u'_{m_j}(\xi) - u'(\xi) : v)|\, d\xi \right] = 0,$$

de onde se conclui que $u(0) = u_0$.

Para concluir a demostração, mostremos que a solução encontrada é única. Suponhamos que existam duas soluções u_1, u_2 satisfazendo as condições do do Teorema 10.1. Então,

$$(u'_1(t) - u'_2(t) : v) + a(u_1(t) - u_2(t), v) = 0, \quad \forall v \in H_0^1(0,1)$$

e como $u_1, u_2 \in L^2(0,T; H_0^1(0,1))$, podemos substituir v por u_1, u_2, de modo que

$$(u'_1(t) - u'_2(t) : u_1(t) - u_2(t)) + a(u_1(t) - u_2(t), u_1(t) - u_2(t)) = 0, \forall t \in [0,T].$$

Como $a(v,v) \geq 0$ para todo $v \in H_0^1(0,1)$, obtemos da equação acima

$$\frac{d}{dt} \|u_1(t) - u_2(t)\|_0^2 \leq 0, \forall t \in [0,T]$$

de onde se conclui que $u_1 = u_2$ após integrar a desegualdade acima. □

Observação: Vale o resultado do Teorema 10.1 para condições de contorno do tipo Dirichlet, Neumann ou Mista. Além disso, a regularidade da solução u depende da regularidade de f. Por exemplo, se $f \in C^k([0,T] \times (0,1))$, $k \in \mathbb{N}$ ou mesmo $k = \infty$, podemos mostrar que $u \in C^k([\varepsilon, T] \times [0,1])$ qualquer que seja $\varepsilon > 0$, independentemente da regularidade de u_0.

10.1.2 Propriedades das soluções

Nessa seção vamos mostrar alguns resultados gerais importantes das soluções da equação do calor homogênea ($f \equiv 0$), a saber: (1) o Princípio do Máximo, que garante que os valores máximo e mínimo da solução u no compacto $Q := [0,1] \times [0,T]$ são atingidos necessariamente sobre a parte da fronteira

$$\Gamma := \Big([0,1] \times \{0\} \Big) \cup \Big(\{0,1\} \times (0,T) \Big) \subset \partial Q \tag{10.7}$$

e (2) o Decaimento Exponencial, que garante que a "energia" do sistema decai exponencialmente quando o tempo cresce indefinidamente.

Embora esses resultados sejam válidos para a equação do calor num domínio limitado de $\mathbb{R}^n$, faremos a demostração no caso unidimensional, já que os argumentos são os mesmos em qualquer dimensão.

• **Princípio do Máximo**

Teorema 10.2. (Princípio do Máximo) *Seja $u(x,t)$ solução da equação*

$$u_t(x,t) - u_{xx}(x,t) = 0, \quad (x,t) \in (0,1) \times (0,T), \tag{10.8}$$

Então u satisfaz as seguintes condições,

$$\begin{aligned} \sup\{u(x,t)\,;\,(x,t) \in Q\} &= \sup\{u(x,t)\,;\,(x,t) \in \Gamma\} \\ \inf\{u(x,t)\,;\,(x,t) \in Q\} &= \inf\{u(x,t)\,;\,(x,t) \in \Gamma\}. \end{aligned} \tag{10.9}$$

Demonstração: Sendo a equação homogênea, segue da Observação acima que u é de classe C^∞ no interior de Q. Consideremos então a função

$$v(x,t) := u(x,t) + \varepsilon x^2, \quad \varepsilon > 0.$$

Então, para todo $(x,t) \in (0,1) \times (0,T)$,

$$v_t(x,t) - v_{xx}(x,t) = -2\epsilon < 0. \tag{10.10}$$

Fixemos $0 < T_1 < T$ e suponhamos que v admita um ponto de máximo absoluto $(x_\varepsilon, t_\varepsilon)$ no interior do retângulo $Q_1 = [0,1] \times [0,T_1]$. Então,

$$v_t(x_\varepsilon, t_\varepsilon) = v_x(x_\varepsilon, t_\varepsilon) = 0. \tag{10.11}$$

Como x_ε é necessariamente ponto de máximo da função $x \mapsto v(x,t_\varepsilon)$, $x \in [0,1]$, vale a desigualdade

$$v_{xx}(x_\varepsilon, t_\varepsilon) \leq 0. \tag{10.12}$$

Como 10.11 e 10.12 estão em contradição com 10.10, concluímos que o máximo de $v(x,t)$ ocorre na fronteira ∂Q_1.

Suponhamos então que o máximo possa ocorrer no ponto (x_ε, T_1). Nesse caso (x_ε, T_1) seria o ponto de máximo absoluto da função $t \mapsto v(x_\varepsilon, t)$ no intervalo $[0, T_1]$. Assim, teríamos necessariamente $v_t(x_\varepsilon, T_1) \geq 0$ e $v_{xx}(x_\varepsilon, T_1) \leq 0$, o que novamente contraria 10.10. Logo,

$$\begin{aligned} \sup\{u(x,t)\,;\,(x,t) \in Q_1\} &\leq \sup\{v(x,t)\,;\,(x,t) \in Q_1\} = \sup\{v(x,t)\,;\,(x,t) \in \partial Q_1\} \\ &\leq \sup\{u(x,t)\,;\,(x,t) \in \partial Q_1\} + \varepsilon. \end{aligned}$$

Fazendo $\varepsilon \to 0$ e $T_1 \to T$, obtemos

$$\sup\{u(x,t)\,;\,(x,t) \in Q_1\} \leq \sup\{u(x,t)\,;\,(x,t) \in \partial Q_1\}. \tag{10.13}$$

Como $\partial Q_1 \subset Q_1$, a desigualdade contrária em (10.13) é evidente. Logo, temos demonstrada a primeira condição de (10.9). Para obter a segunda condição, basta substituir u por $-u$ e repetir exatamente os mesmos argumentos da demonstração. □

Observação: Observe que nenhuma condição de contorno e condição inicial foi estabelecida no enunciado do Teorema 10.8. Observe também que o mesmo resultado vale para a equação $u_t(x,t) - \alpha u_{xx}(x,t) + \beta u(x,t) = 0$, com $\alpha > 0$ e $\beta \in \mathbb{R}$. De fato, se considerarmos a função $v(x,t) := \exp(\beta t)u(x,t)$, então v é solução de (10.8).

• **Decaimento Exponencial**

Teorema 10.3. (Decaimento exponencial) *Sejam $u_0 \in H_0^1(0,1)$ e $u \in L^2(0,+\infty); H_0^1(0,1)) \cap C([0,+\infty); L^2(0,1))$ a solução de*

$$\begin{cases} u_t(x,t) - \alpha u_{xx}(x,t) + \beta u(x,t) = 0, & x \in (0,1),\ t > 0, \\ u(0,t) = u(1,t) = 0, & t \geq 0, \\ u(x,0) = u_0(x), & x \in (0,1), \end{cases} \tag{10.14}$$

onde $\alpha > 0$ e $\beta \geq 0$ são constantes. Então existe $k > 0$ tal que

$$E(t) := \frac{1}{2}\int_0^1 |u(x,t)|^2\,dx \leq E(0)\exp(-kt). \tag{10.15}$$

Demonstração: Multiplicando a equação (10.14) por u e integrando em $(0,1)$, obtemos

$$\int_0^1 \Big[u_t(x,t)u(x,t) + \alpha u_x(x,t)^2 + \beta u(x,t)^2\Big]\,dx = 0. \tag{10.16}$$

Como

$$\int_0^1 u_t(x,t)u(x,t)\,dx = (u_t(t) : u(t)) = \frac{1}{2}\frac{d}{dt}\|u(t)\|_0^2$$

a identidade (10.16) toma a forma

$$\frac{d}{dt}E(t) = -\alpha\int_0^1 u_x(x,t)^2\,dx - \beta\int_0^1 u(x,t)^2\,dx. \tag{10.17}$$

Como $u(t) \in H_0^1(0,1)$ para todo $t > 0$, segue da desigualdade de Poincaré,

$$\int_0^1 u_x(x,t)^2\,dx \geq \pi^2\int_0^1 u(x,t)^2\,dx,$$

de modo que, substiuindo em (10.17), obtemos

$$\frac{d}{dt}E(t) \leq -(\alpha\pi^2 + \beta)E(t), \quad \forall t > 0.$$

Integrado a desigualdade acima, obtemos

$$E(t) \leq E(0)e^{-(\alpha\pi^2+\beta)t}, \quad \forall t > 0,$$

o que demonstra o decaimento exponecial em (10.15) com $k = \alpha\pi^2 + \beta$. □

Observação: O resultado do Teorema 10.3 vale para a equação do calor em um domínio limitado $\Omega \subset \mathbb{R}^n$, com a mesma demonstração. A única diferença aparece na constante da desigualdade de Poincaré, que em uma dimensão espacial é π^2 e no caso n-dimensional é $\lambda_1(\Omega) > 0$, o primeiro auto valor do Laplaciano em Ω. Além disso, veja que o decaimento exponecial da energia $E(t)$ continua válido para valores de $\beta \in (-\alpha\pi^2, 0]$.

10.2 Equação da Onda

Como na seção anterior, vamos demonstrar a existência e unicidade de solução da equação da onda pelo método de Galerkin, assim como algumas propriedades gerais da solução.

10.2.1 Existência e unicidade de solução

Teorema 10.4. *Se $f \in L^2(0,T;L^2(0,1))$, $u_0 \in H_0^1(0,1)$ e $u_1 \in L^2(0,1)$, existe uma única função $u : Q = [0,T] \times (0,1) \to \mathbb{R}$ satisfazendo as seguintes condições:*

$$\begin{aligned}
&\text{(a)} \quad u \in L^\infty(0,T;H_0^1(0,1)) \cap C([0,T];L^2(0,1)),\\
&\text{(b)} \quad u' \in L^\infty(0,T;L^2(0,1)),\\
&\text{(c)} \quad \frac{d}{dt}(u'(t):v) + a(u(t),v) = (f(t):v) \text{ em } L^2(0,T),\ \forall v \in H_0^1(0,1),\\
&\text{(d)} \quad u(0) = u_0 \text{ e } u'(0) = u_1.
\end{aligned} \tag{10.18}$$

Demonstração: Para provar a existência, repetimos o mesmo argumento no caso da equação do calor. Seja $\{w_i\}_{i\in\mathbb{N}}$ uma base de Hilbert do espaço $H_0^1(0,1)$. Consideremos $V_m := [w_1, w_2, \ldots, w_m]$ o subespaço gerado pelos m primeiros elementos da base. Então, qualquer função $u_m(t) \in V_m$ pode ser expressa na forma

$$u_m(t) = \sum_{i=1}^{m} g_{im}(t)w_i. \tag{10.19}$$

Consideremos agora o problema variacional (7.3), restrito ao subespaço V_m, onde pro-

curamos uma solução $u_m : [0,T] \to V_m$, tal que

$$\begin{cases} (u_m''(t) : v) + a(u_m(t), v) = (f(t) : v), \quad \forall v \in V_m, \\ u_m(0) = u_{0m} \to u_0 \text{ forte em } H_0^1(0,1), \\ u_m'(0) = u_{1m} \to u_1 \text{ forte em } L^2(0,1). \end{cases} \tag{10.20}$$

Substituindo sucessivamente v pelos elementos da base de V_m, podemos reescrever o problema (10.20) na forma

$$\begin{cases} A\boldsymbol{g}''(t) + B\boldsymbol{g}(t) = \boldsymbol{F}(t), \quad t \in (0,T), \\ \boldsymbol{g}(0) = \boldsymbol{g}_0, \quad \boldsymbol{g}'(0) = \boldsymbol{g}_1, \end{cases} \tag{10.21}$$

onde $\boldsymbol{g}(t) = (g_{1m}(t), \ldots, g_{mm}(t))$ e $\boldsymbol{g}_i = ((u_i : w_1), \ldots, (u_i : w_m))$, $i = 0, 1$.

O sistema (10.21) é um sistema linear de m equações diferenciais de segunda ordem, que pode ser expresso como um sistema linear de $2m$ equações de primeira ordem. De fato, como a matriz A é invertível (é uma matriz de Gram com relação à base de V_m), o sistema pode ser reescrito na forma

$$\begin{pmatrix} \boldsymbol{g_m}'(t) \\ \boldsymbol{h_m}'(t) \end{pmatrix} = \begin{pmatrix} O & -I \\ -A^{-1}B & O \end{pmatrix} \begin{pmatrix} \boldsymbol{g_m}(t) \\ \boldsymbol{h_m}(t) \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} \\ A^{-1}\boldsymbol{F}(t) \end{pmatrix}.$$

Pelo Teorma de Caratheodory, o sistema acima possui uma única solução

$$\boldsymbol{g}_m \in H^2((0,T); \mathbb{R}^m) \cap C^1([0,T]; \mathbb{R}^m).$$

o que corresponde a $u_m \in H^2((0,T); V_m) \cap C^1([0,T]; V_m)$.

Para analisar a convergência da sequência $\{u_m\}_{m\in\mathbb{N}}$, devemos estabelecer estimativas que independam de m. Para isso, tomemos $v = u_m'(t) \in V_m^k$ em $(10.20)_1$. Então

$$(u_m''(t) : u_m'(t)) + a(u_m(t), u_m'(t)) = (f(t), u_m'(t)).$$

Lembrando que

$$\frac{1}{2}\frac{d}{dt}\|u_m'(t)\|_0^2 = (u_m''(t) : u_m'(t)) \quad \text{e} \quad \frac{1}{2}\frac{d}{dt}\|u_m(t)\|_a^2 = a(u_m(t), u_m'(t)),$$

segue das desigualdades de Schwarz e Young,

$$(f(t) : u_m'(t)) \leq \frac{1}{2}\Big(\|f(t)\|_0^2 + \|u_m'(t)\|_0^2\Big)$$

de onde se conlui que

$$\frac{d}{dt}\Big(\|u_m'(t)\|_0^2 + \|u_m(t)\|_a^2\Big) \leq \|f(t)\|_0^2 + \|u_m'(t)\|_0^2.$$

Integrando a inequação acima em $[0,t]$, com $t \leq T$, obtemos

$$\|u'_m(t)\|_0^2 + \|u_m(t)\|_a^2 \leq \|u'_m(0)\|_0^2 + \|u_m(0)\|_a^2 + \int_0^t \left(\|f(s)\|_0^2 + \|u'_m(s)\|_0^2\right) ds$$

Da hipótese de convergência das sequências de dados inicias em (10.20), podemos encontrar um constante $C_1 > 0$ independente de m (mas dependente de $\|f\|_{L^2(Q)}$) tal que

$$\|u'_m(t)\|_0^2 + \|u_m(t)\|_a^2 \leq C_1 + \int_0^t \|u'_m(s)\|_0^2 \, ds, \tag{10.22}$$

o que nos permite deduzir pela desigualdade de Gronwall,

$$\|u'_m(t)\|_0^2 \leq C_1 e^t, \quad \forall t \in [0,T].$$

Voltando à desigualdade (10.22), obtemos para todo $t \in [0,T]$,

$$\|u_m(t)\|_a^2 \leq C_1 + \int_0^t \|u'_m(s)\|_0^2 \, ds \leq C_1 + C_1(e^t - 1) \leq C_1 e^t.$$

Sendo assim, temos

$$\begin{array}{ll} \text{(i)} & \{u_m\}_{m\in\mathbb{N}} \text{ limitada em } L^\infty(0,T;H_0^1(0,1)), \\ \text{(ii)} & \{u'_m\}_{m\in\mathbb{N}} \text{ limitada em } L^\infty(0,T;L^2(0,1)), \end{array} \tag{10.23}$$

As estimativas (i) e (ii) acima implicam nas condições obtidas para a equação do calor em (10.6). Portanto, com argumentos análogos aos usados nos dois parágrafos após as limitações em (10.6), podemos concluir que

$$\begin{array}{ll} \text{(iii)} & (u'_m(t) : v) \underset{*}{\rightharpoonup} (u'(t) : v) \text{ em } L^\infty(0,T), \ \forall v \in L^2(0,1), \\ \text{(iv)} & a(u_m(t), v) \underset{*}{\rightharpoonup} a(u(t), v) \text{ em } L^\infty(0,T), \ \forall v \in H_0^1(0,1). \end{array} \tag{10.24}$$

Das convergências acima, obtemos

$$(u''_m(t) : v) = \frac{d}{dt}(u'_m(t) : v) \longrightarrow \frac{d}{dt}(u'(t) : v) \text{ em } \mathcal{D}'(0,T), \ \forall v \in H_0^1(0,1).$$

Assim, para $k \in \mathbb{N}$ fixado, temos

$$(u''_m(t) : w_k) + a(u_m(t), w_k) - (f(t) : w_k) = 0 \text{ em } \mathcal{D}'(0,T), \quad \forall m \geq k$$

de onde se conclui, massando ao limite quando $m \to +\infty$,

$$\frac{d}{dt}(u'(t) : w_k) + a(u(t), w_k) - (f(t) : w_k) = 0 \text{ em } \mathcal{D}'(0,T).$$

Como k foi fixado arbitrariamente, temos como conclusão,

$$\frac{d}{dt}(u'(t):v)+a(u(t),v)-(f(t):v)=0 \text{ em } \mathcal{D}'(0,T), \forall v\in H_0^1(0,1).$$

Como a função $t\mapsto a(u(t),v)-(f(t):v)$ pertence a $L^2(0,T)$, segue da equação acima e do item (iii) de (10.24),

$$(u'(t):v)\in L^2(0,T),\ \frac{d}{dt}(u'(t):v)\in L^2(0,T),\ \forall v\in H_0^1(0,1) \tag{10.25}$$

e em consequência, $(u'(t):v)\in H^1(0,T)$, $\forall v\in H_0^1(0,1)$. Assim concluímos que a aplicação $t\mapsto(u'(t):v)$ é contínua em $[0,T]$.

Com argumentos análogos aos usados na seção anterior, podemos mostrar que a solução u satisfaz as condições iniciais do item (d).

Para mostrar a unicidade, suponhamos u_1 e u_2 duas funções satisfazendo os itens (a)–(d) de (10.18). Então $w:=u_1-u_2$ satisfaz

$$\begin{array}{ll}
\text{(a')} & w\in L^\infty(0,T;H_0^1(0,1))\cap C([0,T];L^2(0,1)),\\
\text{(b')} & w'\in L^\infty(0,T;L^2(0,1)),\\
\text{(c')} & (w''(t):v)+a(w(t),v)=0 \text{ em } L^2(0,T),\ \forall v\in H_0^1(0,1),\\
\text{(d')} & w(0)=0 \text{ e } w'(0)=0.
\end{array} \tag{10.26}$$

Infelizmente não podemos simplesmente imitar aqui o argumento usado na equação do calor. Mais precisamente, não podemos substituir $v(x)$ por $w'(x,t)$ na equação (item (c')) para recuperar derivadas nulas de $\|u'(t)\|_0^2$ e $a(u(t),u(t))$, visto que, com as hipóteses feitas, não podemos garantir $u'(t)\in H_0^1(0,1)$. Para contornar essa dificuldade, consideremos a seguinte função: para cada $s\in(0,T]$, seja

$$v(t,s,x):=\begin{cases}-\int_t^s w(x,\xi)\,d\xi & \text{se } t<s,\\ 0 & \text{se } t\geq s.\end{cases}$$

Como $w\in C([0,T];H_0^1(0,1))$, segue que $v(t,s)\in C^1([0,T];H_0^1(0,1))$ para todo s. Substituindo v por $v(t,s)$ em (10.26) e integrando em $t\in[0,s]$, temos

$$\int_0^s (w''(t):v(t,s))\,dt+\int_0^s a(w(t),v(t,s))\,dt=0.$$

Integrando por partes a primeira das integrais acima, temos

$$w'(s)v(s,s)-w'(0)v(0,s)-\int_0^s (w'(t):v'(t,s))\,dt+\int_0^s a(w(t),v(t,s))\,dt=0.$$

Como $v(s,s)=0$, $w'(0)=0$ e $v'(t,s)=\partial_t v(t,s)=w(t)$, a identidade acima se reduz a

$$\int_0^s (w'(t):w(t))\,dt - \int_0^s a(v'(t,s),v(t,s))\,dt = 0,$$

ou seja

$$\frac{1}{2}\int_0^s \frac{d}{dt}\Big(\|w'(t)\|_0^2 - a(v(t,s),v(t,s))\Big)\,dt = 0.$$

Portanto, $\|w'(s)\|_0^2 \le \|w'(s)\|_0^2 + a(v(0,s),v(0,s)) = 0$, de onde se conclui que $w'(s)=0$ para todo s, e consequentemente $w(s)=w(0)=0$. □

10.2.2 Conservação de Energia

Se u é solução da equação onda (10.18), definimos $E(t)$, a "energia" de u no instante t (no caso unidimensional), por:

$$E(t) := \frac{1}{2}\int_0^1 u'(x,t)^2\,dx + \frac{1}{2}\int_0^1 a(u(x,t),u(x,t)))\,dx - \int_0^1 f(x)u(x,t)\,dx \qquad (10.27)$$

Essa denominação tem o sentido físico da palavra energia. De fato, considerando, por exemplo, o caso das oscilações de uma corda elástica e flexível, onde $u(x,t)$ denota a posição x da corda no instante t, as duas primeiras integrais (com valores unitários das constantes físicas da corda) representam respectivamente a energia cinética e a energia potencial devida às forças elásticas internas; e a última integral, a energia potencial devida à distribuição $f(x)$ de forças externas atuando no ponto x.

Observemos que na definição acima, estamos supondo que $u'(t) \in L^2(0,1)$, para que a energia potencial esteja bem definida como função de t, o que nos garante o item (b) do Teorema 10.4.

Dizemos que a energia $E(t)$ se conserva se $E(t)=E(0)$ para todo t e para mostrar que a energia se conserva, vamos supor que a solução seja suficientemente regular para que os cálculos que seguem sejam justificáveis.

Multiplicando a equação por $u'(x,t)$ e integrando em x, obtemos

$$\frac{1}{2}\frac{d}{dt}\Big(\|u'(t)\|_0^2 + a(u(t),u(t))\Big) - \frac{d}{dt}(f:u(t)) = 0$$

Integrando de 0 a t, obtemos

$$\begin{aligned} E(t) &= \frac{1}{2}\Big(\|u'(t)\|_0^2 + a(u(t),u(t)\Big) - (f:u(t)) \\ &= \frac{1}{2}\Big(\|u_1\|_0^2 + a(u_0,u_0)\Big) - (f:u_0) = E(0). \end{aligned}$$

Observemos que se f depende do tempo, não pode haver conservação da energia, pois as forças externas estariam acrescentando ou retirando energia do sistema.

Os argumentos formais apresentados acima podem ser matematicamente justificáveis se considerarmos mais regularidade para a solução da equação da onda. Por exemplo, nas condições do teorema que segue, cuja demonstração será omitida, os cálculos formais são justificados.

Teorema 10.5. *Se além das hipóteses do Teorema 10.4 supusermos que*

$$\frac{\partial f}{\partial t} \in L^2(0,T;L^2(0,1)), \quad u_0 \in H_0^1(0,1)\cap H^2(0,1), \quad u_1 \in H_0^1(0,1),$$

a única solução de (10.18) satisfaz as seguintes condições:

$$\begin{aligned}
&\text{(i)} \ \ u \in L^\infty(0,T;H_0^1(0,1)\cap H^2(0,1)),\\
&\text{(ii)} \ \ u' \in L^\infty(0,T;H_0^1(0,1)),\\
&\text{(iii)} \ \ u'' \in L^2(0,T;L^2(0,1)),\\
&\text{(iv)} \ \ u'' - \alpha u_{xx} + \beta u = f, \quad \textit{quase sempre em } Q,\\
&\text{(v)} \ \ u(x,0) = u_0(x), \quad u'(x,0) = u_1(x), \quad \forall x \in (0,1).
\end{aligned}$$

Para a demonstração, além dos os mesmos para as estimativas do Teorema 10.4, usamos as condições mais regulares para obter novas estimativas, derivando a equação aproximada em relação a t.

APÊNDICE A

Programas computacionais: linguagem C

`PEU.cpp`	Problema Estacionário Unidimensional
`typdef.h`	Header file, definições dos variáveis vetorial e matricial
`grid.h`	Header file, malha do domínio quadrado
`solver.h`	Header file, linear solver - algoritmo de Crout
`PEB.cpp`	Problema Estacionário Bidimensional
`elast.cpp`	Elasticidade Linear Bidimensional
`Calor.cpp`	Equação do Calor Unidimensional
`Onda.cpp`	Equação da Onda Unidimensional

A.1 Problema estacionário unidimensional – PEU.cpp

```
/* Problema Estacionário Unidimensional(PEU)
   Metodo de Elementos finitos
    - (Alpha)u'' + (Beta)u = f(x)
   tipo=1: u() prescrita
   tipo=2: du() prescrita
   Linear solver: Tri-diagonal matrix
*/

#include <stdio.h>
#include <math.h>
#include <string.h>

#define  Nosm   102
#define  Nelm   101
#define  Xi1    -sqrt(3)/3
#define  Xi2    sqrt(3)/3

FILE  *f1;
int   Nel,Nos;
int   tipo1,tipo2;
float X[Nosm], Xe[3];
float H[Nelm];
float K[Nosm][4];
float F[Nosm],D[Nosm];
float Alpha,Beta;
float A1,A2,dl,v[2];
float Uex[Nosm];


void InputData()
{
   printf("Constante Alpha = ");
   scanf("%f", &Alpha);
   printf("Constante Beta = ");
   scanf("%f", &Beta);
   fprintf(f1,"Alpha = %f\n", Alpha);
   fprintf(f1,"Beta  = %f\n", Beta);
   printf("\n\nNumeros de elementos = ");
   scanf("%d", &Nel);
```

```
   fprintf(f1,"Numero de elementos = %d\n", Nel);
}


void InputCondFront()
{
   printf("\nCond. Contorno:(tipo: 1-Dirichlet 2-Neumann)\n");
   printf("Cond. do no %d (tipo, valor):", 1);
   scanf("%d %f", &tipo1, &A1);
   printf("Cond. do no %d (tipo, valor):", Nos);
   scanf("%d %f", &tipo2, &A2);
}


void CoordGlobal()
{  int i;

   /* divisao uniforme */
   printf("Entre com a coord. do No' esquerda:\n");
   scanf("%f", &X[1]);
   printf("Entre com a coord. do No' direta:\n");
   scanf("%f", &X[Nos]);
   dl = (X[Nos]-X[1])/Nel;
   for (i=2; i<Nos; i++) X[i]=X[1]+dl*(i-1);

   printf("\nCoord. Global:\n");
   fprintf(f1,"\nCoord. Global:\n");
   for (i=1; i<=Nos; i++)
   {
     printf(" No %d = %f\n", i, X[i]);
     fprintf(f1," No %d = %f\n", i, X[i]);
   }
}


void CoordLocal(int e)
{
   Xe[1] = X[e];
   Xe[2] = X[e+1];
   H[e] = Xe[2] - Xe[1];
}
```

```
int NoLG(int e,int a)
{
   if (a==1)  return  e;
   else       return  (e+1);
}


float Xi(int e, float x)  /* Transf. isoparametrica */
{
   return (Xe[1] + H[e]/2*(x+1));
}


void MatrizRigidez()
{  int e;
   float K11,K22,K12,K21;
   for (e=1; e<=Nel; e++)
   {
      CoordLocal(e);
      K11 = Alpha/H[e] + Beta/3*H[e];
      K22 = Alpha/H[e] + Beta/3*H[e];
      K12 = -Alpha/H[e] + Beta/6*H[e];
      K21 = -Alpha/H[e] + Beta/6*H[e];
      /* Matriz Rigidez */
      K[e][2] += K11;
      K[e][3]  = K12;
      K[e+1][1]  = K21;
      K[e+1][2]  = K22;
   }
}

float forca(float x)
{
   return x;   /* funcao de forca prescrita */
}


float g1(int e, float x)
{
   return  (.5*forca(Xi(e,x))*(1 - x));
}
```

```
float g2(int e, float x)
{
   return  (.5*forca(Xi(e,x))*(1 + x));
}


void VetorForca()
{
   int e;
   float Fe1,Fe2;
   for (e=1; e<=Nel; e++)
   {
   /* Forca local - Quadratura Gauss 2-pts */
      CoordLocal(e);
      Fe1 = H[e]/2*(g1(e,Xi1)+g1(e,Xi2));
      Fe2 = H[e]/2*(g2(e,Xi1)+g2(e,Xi2));
   /* Forca global */
      F[NoLG(e,1)] += Fe1;
      F[NoLG(e,2)]  = Fe2;
}  }


void CondFront(float A, float B)
{
   switch (tipo1)
   { case 1:{
        K[1][2] = 1;
        K[1][3] = 0;
        F[1]  = A;
        F[2] += -K[2][1]*A;
        K[2][1] = 0;
        break;}
     case 2:{
        F[1] = F[1] - Alpha*A;
        break;}
   }
   switch (tipo2)
   { case 1:{
        F[Nos-1] += -K[Nos-1][3]*B;
        F[Nos] = B;
        K[Nos-1][3] = 0;
```

```
        K[Nos][1] = 0;
        K[Nos][2] = 1;
        break;}
     case 2:{
        F[Nos] = F[Nos] + Alpha*B;
        break;}
   }
}


void LinearSolver(float M[Nosm][4], float F[Nosm],float X[Nosm])
{  int i;
   float y;
   for (i=1; i<Nos; i++)
   {
       y = M[i+1][1]/M[i][2];
       M[i+1][2] += -y*M[i][3];
       F[i+1] += -y*F[i];
   }
   /* Retro-Substituicao */
   X[Nos] = F[Nos]/M[Nos][2];
   for (i=Nos-1; i>=1; i--)
       X[i] = (F[i]-M[i][3]*X[i+1])/M[i][2];
}


void SolExata()
{ int i;
  float Gama =  sqrt(Beta/Alpha);
  float c1 = exp(Gama);
  float c2 = exp(-Gama);
  float c3 =(A2-(1./Beta));
  float c4 =(A1*c2);
  float c5 =(A1*c1);
  float c6 =((1./Beta)-A1);
  float c8 =(Gama*(c1+c2));
  float U1[Nosm],U2,U3;

  for (i=1; i<=Nos; i++)
  U1[i]=1./Beta*X[i];
```

```
switch (tipo1)
{ case 1:
  {
   switch (tipo2)
   { case 1:
     /*cond. fronteira:u(x[1])=A1 e u(x[Nos])=A2*/
     /*tipo1-case1,tipo2-case1*/
     { for (i=1; i<=Nos; i++)
       { U2= (c3-A1*c2)*pow(c1,X[i]);
         U3= (A1*c1-c3)*pow(c2,X[i]);
         Uex[i]= U1[i]+1./(c1-c2)*(U2+U3);
       }
       break;
     }
     case 2:
     /*cond. fronteira:u(x[1])=A1 e Du(x[Nos]=A2*/
     /*tipo1-case1, tipo2-case2*/
     { for (i=1; i<=Nos; i++)
       { U2=(c3+Gama*c4)*pow(c1,X[i]);
         U3=(Gama*c5-c3)*pow(c2,X[i]);
         Uex[i]=U1[i]+1./c8*(U2+U3);
       }
       break;
  }} }
  break;
  case 2:
  {
   switch (tipo2)
   { case 1:
     /*cond. fronteira:Du(x[1])=A1 e u(x[Nos]=A2*/
     /*tipo1-case2,tipo2-case1*/
     { for (i=1; i<=Nos; i++)
       { U2= (-c2*c6+Gama*c3)*pow(c1,X[i]);
         U3= ( c1*c6+Gama*c3)*pow(c2,X[i]);
         Uex[i]= U1[i]+1./c8*(U2+U3);
       }
       break;
     }
     case 2:
     /*cond. fronteira:Du(x[1])=A1 e Du(x[Nos]=A2*/
     /*tipo1-case2,tipo2-case2*/
```

```
       { for (i=1; i<=Nos; i++)
         { U2=(-c6/Gama)*pow(c1,X[i]);
           U3=1./Gama/(c1-c2)*(c6*c1+c3)*(pow(c1,X[i])+pow(c2,X[i]));
           Uex[i]= U1[i]+(U2+U3);
         }
         break;
  } }} }
}



void Norma(float *f)
{  int   i;
   float L2=0, H1=0, x1, x2;
   for (i=1; i<Nos; i++)
   {
       x1 = f[i+1] + f[i];
       x2 = f[i+1] - f[i];
       L2 += x1*x1;
       H1 += x2*x2;
   }
   v[0] = sqrt(L2/4*dl);/* Norma L^2*/
   v[1] = sqrt(L2/4*dl+H1/dl);*/Norma H^1*/
   }



void OutputData()
{  int i,j;
   fprintf(f1,"\nMatriz Rigidez:\n");
   for (i=1; i<=Nos; i++)
   {
       for (j=1; j<=3; j++)
         fprintf(f1,"  K(%d,%d)=%f", i,j, K[i][j]);
         fprintf(f1,"\n");
   }
   fprintf(f1,"\nForca Prescrita:\n");
   for (i=1; i<=Nos; i++)
       fprintf(f1,"  F(%d)=%f", i, F[i]);
       fprintf(f1,"\n");
}
```

```
void OutputResultado()
{  int i;
   fprintf(f1,"\n  No     Aproximada       Exata");
   for (i=1; i<=Nos; i++)
   {
       fprintf(f1,"\n %3d    %+.8f    %+.8f ", i, D[i], Uex[i]);
   }
       fprintf(f1,"\n");
}

void main()
{  int   i;
   float a0, a1;

   f1=fopen("fem1d.out","w");
   InputData();
   Nos = Nel + 1;
   CoordGlobal();
   InputCondFront();
   MatrizRigidez();
   VetorForca();
   /* Cond. Contorno: */
   CondFront(A1,A2);
   OutputData();
   LinearSolver(K,F,D);
   SolExata();
   OutputResultado();
   Norma(Uex);
   a0 = v[0]; a1 = v[1];
   for (i=1; i<=Nos; i++) D[i] = D[i] - Uex[i];
   Norma(D);
   fprintf(f1,"\n err(L2) = %.4f \%", v[0]/a0*100);
   fprintf(f1,"\n err(H1) = %.4f \%", v[1]/a1*100);
   fclose(f1);
}
```

A.2 Header file – `typdef.h`

```
/*
Definition of vector and matrix type variables
*/

typedef struct{float v[2];}vec_2;
typedef struct{int   v[2];}vec_2i;
typedef struct{float v[4];}vec_4;
typedef struct{float v[8];}vec_8;
typedef struct{float m[2][2];}mat_2;
typedef struct{float m[2][4];}mat_24;
typedef struct{float m[4][4];}mat_4;
typedef struct{float m[8][8];}mat_8;

float *v_alloc(int n)           /* float n-vetor */
{
   float *v;
   v= (float *) calloc(n,sizeof(float));
   if (v==NULL)
   {
      fprintf(stderr, "nao pode alocar memoria");
      exit(1);
   }
   return v;
}

float **m_alloc(int m,int n)    /* float mxn-matriz */
{
  int i;
  float **a;
  a=(float **) calloc(m,sizeof(float *));
  if (a==NULL)
  {
     fprintf(stderr,"Nao pode alocar memoria");
     exit(1);
  }
  for (i=1;i<=m;i++)
  {
     a[i-1]=(float *) calloc(n, sizeof(float));
     if (a[i-1]==NULL)
```

```
     {
        fprintf(stderr,"Nao pode alocar memoria");
        exit(1);
     }
  }
  return a;
}

int *vi_alloc(int n)        /* int n-vetor */
{
   int *v;
   v= (int *) calloc(n,sizeof(int));
   if (v==NULL)
   {
      fprintf(stderr, "nao pode alocar memoria");
      exit(1);
   }
   return v;
}

int **mi_alloc(int m,int n)    /* int mxn-matriz */
{
  int i;
  int **a;
  a=(int **) calloc(m,sizeof(int *));
  if (a==NULL)
  {
     fprintf(stderr,"Nao pode alocar memoria");
     exit(1);
  }
  for (i=1;i<=m;i++)
  {
     a[i-1]=(int *) calloc(n, sizeof(int));
     if (a[i-1]==NULL)
     {
        fprintf(stderr,"Nao pode alocar memoria");
        exit(1);
     }
  }
  return a;
}
```

```
mat_24 Prod22_24(mat_2 u, mat_24 v) /* Prod. mat (2x2).(2X4) */
{ int j;
  mat_24 x;
  for (j=0; j<=3; j++)
  {
      x.m[0][j]=u.m[0][0]*v.m[0][j]+u.m[0][1]*v.m[1][j];
      x.m[1][j]=u.m[1][0]*v.m[0][j]+u.m[1][1]*v.m[1][j];
  }
  return x;
}

vec_4 Prod44_41(mat_4 u, vec_4 v) /* Prod. mat (4x4).(4X1) */
{ int j;
  vec_4 x;
  for (j=0; j<=3; j++)
  {
      x.v[j]=u.m[j][0]*v.v[0]+u.m[j][1]*v.v[1]+
             u.m[j][2]*v.v[2]+u.m[j][3]*v.v[3];
  }
  return x;
}
```

A.3 Header file – `grid.h`

```
/*
    Regiao: unit square.
    Square element
    Types of knot:
       tipo=0: outside the domain
       tipo=1: boundary knot
       tipo=2:
       tipo=3:
       tipo=4: interior knot

Functions:
vec_2i NoPos(int);               No'global -> Posicao
vec_2i ElmPos(int e);            Posicao (No'1) de Elemento
int    PosNo(vec_2i);            Posicao -> No'global
int    NoLG(int a, int e);       No'local(elmento) -> no' global
vec_4  Phi(float,float);         Funcao interpolacao
mat_24 DPhi(float,float);        Gradiente da funcao interpolacao
void   PhiMatriz(void);          Integr (DPhi),(Phi) do elemento
*/

#define  Xi0   0.5773502691896  /* 1/sqrt(3) */


int   Nelx,Nel,Nno,Neq;
int   band;
float *x,*y;           /* x/y-coord of knot */
int   *typ,*eqn;    /* type and eqno of knot */
float Jacob;        /* Jacobiano do (x,y)/(xi,eta) */
float Qabij[4][4][2][2];
float Qab[4][4];


vec_2i NoPos(int n)
{  vec_2i x;
   x.v[0] = (n-1)%(Nelx+1);
   x.v[1] = (n-1)/(Nelx+1);
   return x;
}
```

```
int PosNo(vec_2i x)
{
   return (x.v[1]*(Nelx+1) + x.v[0] + 1);
}


vec_2i ElmPos(int e)
{  vec_2i x;
   x.v[0] = (e-1)%Nelx;
   x.v[1] = (e-1)/Nelx;
   return x;
}


int NoLG(int a, int e)
{  vec_2i x;
   x=ElmPos(e);
   if (a==1) { x.v[0]++;}
   if (a==2) { x.v[0]++; x.v[1]++; }
   if (a==3) { x.v[1]++; }
   return PosNo(x);
}


vec_4 Phi(float xi, float eta)
{  int a;
   float x[4][2]={{-1,-1},{1,-1},{1,1},{-1,1}};
   vec_4 v;
   for (a=0; a<=3; a++)
   {
       v.v[a] = (1.+x[a][0]*xi)*(1.+x[a][1]*eta)/4;
   }
   return v;
}


mat_24 DPhi(float xi, float eta)
{  int a;
   float x[4][2]={{-1,-1},{1,-1},{1,1},{-1,1}};
   mat_24 g;
   for (a=0; a<=3; a++)
```

```
   {
       g.m[0][a] = (x[a][0]*(1.+x[a][1]*eta)/4);
       g.m[1][a] = (x[a][1]*(1.+x[a][0]*xi )/4);
   }
   return g;
}


void PhiMatriz()
{  int i,j,k,a,b;
   float  xi,eta;
   mat_24 B;
   vec_4  g;
   float  q[4][2]={{-Xi0,-Xi0},{Xi0,-Xi0},{Xi0,Xi0},{-Xi0,Xi0}};

   for (k=0; k<=3; k++)   /* Gaussian integration 4-pts */
   {
       xi = q[k][0];
       eta= q[k][1];

       B  = DPhi(xi,eta);
       for (a=0; a<=3; a++)
       for (b=0; b<=3; b++)
       for (i=0; i<=1; i++)
       for (j=0; j<=1; j++)
           Qabij[a][b][i][j] += B.m[i][a] * B.m[j][b];

       g = Phi(xi,eta);
       for (a=0; a<=3; a++)
       for (b=0; b<=3; b++)
           Qab[a][b] += g.v[a] * g.v[b];
   }
}

float L2(float *u, float *v)
{  int    e,a,b,na,nb;
   float  z={0};

   for (e=1; e<=Nel; e++)
   {
       for (a=0; a<=3; a++)
```

```
        {
            na = NoLG(a,e);
            for (b=0; b<=3; b++)
            {
                nb = NoLG(b,e);
                z += Jacob*Qab[a][b]*u[na]*v[nb];
            }
        }
    }
    return z;
}


void Grid(int m)
{  int   n,i;
   float d;

   Nelx = m;
   x = v_alloc(Nelx+1);
   d = 1./Nelx;
   for (i=0; i<=Nelx; i++)  x[i] = d*i;
   Jacob = d*d/4;

   Nel = Nelx*Nelx;
   Nno = (Nelx+1)*(Nelx+1);
   typ = vi_alloc(Nno+1);
   eqn = vi_alloc(Nno+1);

   for (n=1; n<=Nno; n++) typ[n]=4;
   for (n=1; n<=Nelx+1; n++)              typ[n] = 1;
   for (n=Nno-Nelx; n<=Nno; n++)          typ[n] = 1;
   for (n=1; n<=Nno-Nelx; n += Nelx+1)    typ[n] = 1;
   for (n=Nelx+1; n<=Nno; n += Nelx+1)    typ[n] = 1;

   for (i=1,n=1; n<=Nno; n++)
   {
     if (typ[n]>1)
     {
        eqn[n]=i; Neq=i++;
     }
     else eqn[n]=0;
```

```
   }
   PhiMatriz();
}


void GridData(FILE *f)
{  int e,n,a;

   fprintf(f,"\nNo's dos elementos:");
   for (e=1; e<=Nel; e++)
   {
       fprintf(f,"\nElm %4d:", e);
       for (a=0; a<=3; a++)
           fprintf(f,"   %4d", NoLG(a,e));
   }
   fprintf(f,"\nCondicao de contorno:");
   fprintf(f,"\n No'    Tipo   Eqno ");
   for (n=1; n<=Nno; n++)
   {
       fprintf(f,"\n%4d     %1d     %4d", n, typ[n], eqn[n]);
       if (n%(Nelx+1)==0)  fprintf(f,"\n");
   }
   fprintf(f,"\nNel =%4d  Nno =%4d  Neq =%4d ", Nel, Nno, Neq);
}


void PlotData(FILE *f1, float *f)
{ int n;
  vec_2i p;

  for (n=1; n<=Nno; n++)
  {
      p = NoPos(n);
      fprintf(f1,"\n%+f  %+f  %+f", x[p.v[0]], y[p.v[1]], f[n]);
      if (n%(Nelx+1)==0) fprintf(f1,"\n");
  }
}
```

A.4 Header file – `solver.h`

```
/*
Linear solver for AX=B
  Algorithm - Crout  A-symmetric
  LU decomposition: U'DU -> A
  Solution X -> B -> V
*/
#define tol 1e-8

void LU_Decomp(float **A)
{  int   i,j,k;
   float t;

   for (j=1; j<Neq; j++)
   {
       for (i=((j-band+2)>0)? j-band+2 : 1; i<=j-1; i++)
       for (k=((j-band+1)>0)? j-band+1 : 0; k<=i-1; k++)
           A[i][j-i] -= A[k][i-k]*A[k][j-k];
       for (i=((j-band+1)>0)? j-band+1 : 0; i<=j-1; i++)
       {
          t = A[i][j-i];
          if (fabs(A[i][0])<tol)
            { printf("\nerr 1: A[%d,%d] < %e",i+1,i+1,tol); exit(1);}
          A[i][j-i] = t/A[i][0];
          A[j][0] -= t*A[i][j-i];
          if (fabs(A[j][0])<tol)
            { printf("\nerr 2: A[%d,%d] < %e",j+1,j+1,tol); exit(1);}
       }
   }
}

/* Back substitution
   Solution -> V */
void Solver(float **A, float *B, float *V)
{  int   i,j,n;

   for (j=1; j<Neq; j++)
   for (i=((j-band+1)>0)? j-band+1 : 0; i<=j-1; i++)
       B[j] -= A[i][j-i]*B[i];
```

```
   for (j=0; j<Neq; j++)
       B[j] = B[j]/A[j][0];

   for (j=Neq-1; j>=1; j--)
   for (i=((j-band+1)>0)? j-band+1 : 0; i<=j-1; i++)
       B[i] -= A[i][j-i]*B[j];

   /* set solution -> V */
   for (n=1; n<=Nno; n++)
       V[n] = (typ[n]==1)? 0 : B[eqn[n]-1];
}
```

A.5 Problema estacionário bidimensional – PEB.cpp

```
/*
    Problema Estacionário Bidimensional(PEB)
    Metodo de Elementos Finitos
    Condução de calor, meio anisotropico
       -(K_ij (U_,j)),i = f(x)
    bi-dimensional: elementos retangulos
    tipo=1: U    deslocamento prescrito
    tipo=0:  deslocamento nao prescrito
    Arquivo de Saida:
       PEB.out   dados e solucoes
       PEB.dat   3D-plot arquivo (para fazer o grafico)
      Esse programa permite que na entrada de dados a escolha
      dos três tipos de fronteira Dirichlet, Neumann e misto
      para o problema, cuja solucao exata é:
      u = sen(pi*x)sen(pi*y),  com x,y no intervalo[0,1].
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

FILE  *f1;
int   Nely;         /* Numero de elementos y*/
int   Ex;           /* Escolha do exemplo a ser rodado */
int   band;         /*Banda da matriz*/
float *y;
float **K;          /* Matriz Rigidez */
float *F;           /* Vetor  Carregamento-> vetor solucao-> erro */
float *u;           /* Vetor deslocamento */
int   **bdy,Nbn[5]; /*Identificando os nós da fronteira*/
float **Bv;         /* Valor de fronteira*/
float *uv;
float *Du[2];

#include "typdef.h"
#include "grid.h"
#include "solver.h"
```

```
/* Função protótipo*/
void   DataInput(void);        /* Entrada de dados */
void   InputData(void);        /* Dados de entrada */
void   Fronteira(void);        /*Identificando os nos da fronteira*/
void   CondFront(int);         /* Condicao de fronteira */
void   EqNo(void);             /* Número de No global -> Variavel */
mat_2  Hess(int);              /* Hessiano D(xi,eta)/D(x,y) */
float  Jacobi(int);            /* Jacobiano do Hess^-1, retangulo */
mat_2  Qm(int e);              /* Matriz condutividade do elemento */
float  Load(float,float);      /* Funcao do carregamento externo */
void   LocalSystem(int);       /* Rigidez-força local */
void   GlobalSystem(void);     /* Rigidez-força global */
void   TractionBoundary(void); /* Para os nos da fronteira de Neumann*/
vec_2i NoPos(int);             /* No global -> Posicao  */
void   PhiMatriz(void);        /* Integ (Dphi),(phi) do elemento */
void   System(void);           /* Kab, Fa, Sistema Kab Xb = Fa */
void   Solution(void);         /* Solucao Xb */
void   Sol_Nodal(void);

#define  Xi0 0.5773502691896 /* 1/sqrt(3)-Gaussiana com 2 pontos*/
#define  Pi  3.14159265359
mat_4 Kab;
vec_4 Fa;


void main()
{
   int    n;
   vec_2i p;
   f1 = fopen("PEB.out","w");   /*Arquivo de  saida*/
   DataInput();
   Nel = Nelx*Nely;
   Nno = (Nelx+1)*(Nely+1);
   band= Nelx+3;
   eqn = vi_alloc(Nno+1);
   typ = vi_alloc(Nno+1);
   u  = v_alloc(Nno+1);
   uv =  v_alloc(Nno+1);
   n = (Nelx>=Nely)? Nelx+2 : Nely+2;
   bdy = mi_alloc(5,n);
   Bv  = m_alloc(5,n);
```

```
Fronteira();
   CondFront(Ex);
   EqNo();
   K = m_alloc(Neq,band);
   F = v_alloc(Nno+1);
   Du[0] = v_alloc(Nno+1);
   Du[1] = v_alloc(Nno+1);

   PhiMatriz();
   GlobalSystem();
   LU_Decomp(K);
   Solver(K,F,uv);

   GridData(f1);
   /* u[n] é a solução */
   for (n=1; n<=Nno; n++)
       u[n] = (typ[n]==1)? u[n] : F[eqn[n]-1];
    /* solução exata: sen(Pi*x)*cos(Pi*y)
   /* F[n] é o erro , U[n]-exact[n] */
   for (n=1; n<=Nno; n++)
   {
       p  = NoPos(n);
       F[n] = u[n] - sin(Pi*x[p.v[0]])*cos(Pi*y[p.v[1]]);
   }
   Sol_Nodal();
   fclose(f1);
   f1=fopen("PEB.dat","w");  /*Arquivo para fazer o grafico*/
   PlotData(f1,u);
   fclose(f1);
}


void DataInput(void)
{  int i;
   float d;

   printf("\n\nNumeros de x-divisao = ");
   scanf("%d", &Nelx);
   x = v_alloc(Nelx+1);
   d = 1./Nelx;
   for (i=0; i<=Nelx; i++)  x[i] = d*i;
```

```
   printf("Numeros de y-divisao = ");
   scanf("%d", &Nely);
   y = v_alloc(Nely+1);
   d = 1./Nely;
   for (i=0; i<=Nely; i++)  y[i] = d*i;

   printf("Fronteira de Dirichlet(1), Neumann(2), Misto(3).\\
    Digite o tipo de fronteira = ");
   scanf("%d", &Ex);
}


void Fronteira(void)
{  int n,i;

   for (n=1,i=1; n<=Nelx+1; n++,i++)
   { bdy[1][i] = n;  Nbn[1] = i; }
   for (n=Nelx+1,i=1; n<=Nno; n+=Nelx+1,i++)
   { bdy[2][i] = n;  Nbn[2] = i; }
   for (n=Nno-Nelx,i=1; n<=Nno; n++,i++)
   { bdy[3][i] = n;  Nbn[3] = i; }
   for (n=1,i=1; n<=Nno-Nelx; n+=Nelx+1,i++)
   { bdy[4][i] = n;  Nbn[4] = i; }
 }
void CondFront(int Ex)
{  int i,n;
   vec_2i p;

   for (n=1; n<=Nno; n++)
   {typ[n]=0;  u[n]=0;}

  for (n=1; n<=4; n++)
   for (i=1; i<=Nbn[n]; i++)
    Bv[n][i]=0;

if (Ex==1)
  {
    for (i=1; i<=Nbn[1]; i++)
   {
     n = bdy[1][i];  p = NoPos(n);
```

```
      typ[n] = 1;     u[n] = sin(Pi*x[p.v[0]]); /*Fronteira prescrita*/
   }

   for (i=1; i<=Nbn[2]; i++)
   {
     n = bdy[2][i];  p = NoPos(n);
     typ[n] = 1;     u[n] = 0;                 /*Fronteira prescrita*/
   }
   for (i=1; i<=Nbn[3]; i++)
   {
     n = bdy[3][i];  p = NoPos(n);
     typ[n] = 1;     u[n] = -sin(Pi*x[p.v[0]]); /* Fronteira prescrita*/
   }
   for (i=1; i<=Nbn[4]; i++)
   {
     n = bdy[4][i];  p = NoPos(n);
      typ[n] = 1;     u[n] = 0;                 /* Fronteira prescrita*/
   }
  }
  if (Ex==2)
  {
   for (n=1; n<=Nno; n++) typ[n]=0;

   for (i=1; i<=Nbn[1]; i++)
   {
     n = bdy[1][i];  p = NoPos(n);
     Bv[1][i] =  -Pi*cos(Pi*x[p.v[0]]);         /* Fronteira nao prescrita*/
   }
 for (i=1; i<=Nbn[2]; i++)
   {
     n = bdy[2][i];  p = NoPos(n);
     Bv[2][i] = -Pi*cos(Pi*y[p.v[1]])*2;       /* Fronteira nao prescrita*/
   }

   for (i=1; i<=Nbn[3]; i++)
   {
     n = bdy[3][i];  p = NoPos(n);
     Bv[3][i] = -Pi*cos(Pi*x[p.v[0]]);         /* Fronteira nao prescrita*/
   }

   for (i=1; i<=Nbn[4]; i++)
```

```
  {
    n = bdy[4][i];  p = NoPos(n);
    Bv[4][i] = -Pi*cos(Pi*y[p.v[1]])*2;      /* Fronteira nao prescrita*/
  }

    typ[1] = 1;     u[1] = 0;     /* Fixando o no global, para unicidade*/
  }

  if (Ex==3)
 {
  for (n=1; n<=Nno; n++) typ[n]=0;

  for (i=1; i<=Nbn[1]; i++)
  {
    n = bdy[1][i];  p = NoPos(n);
    // typ[n] = 1;     u[n] = sin(Pi*x[p.v[0]]); /* Fronteira prescrita*/
    Bv[1][i] =  -Pi*cos(Pi*x[p.v[0]]);       /* Fronteira nao prescrita*/
  }

  for (i=1; i<=Nbn[2]; i++)
  {
    n = bdy[2][i];  p = NoPos(n);
    typ[n] = 1;     u[n] = 0;              /* Fronteira prescrita*/
     //Bv[2][i] = -Pi*cos(Pi*y[p.v[1]])*2; /* Fronteira nao prescrita*/
  }

  for (i=1; i<=Nbn[3]; i++)
  {
    n = bdy[3][i];  p = NoPos(n);
    //typ[n] = 1;     u[n] = -sin(Pi*x[p.v[0]]); /* Front. prescrita*/
     Bv[3][i] = -Pi*cos(Pi*x[p.v[0]]);       /* Fronteira nao prescrita*/
  }

  for (i=1; i<=Nbn[4]; i++)
  {
    n = bdy[4][i];  p = NoPos(n);
    typ[n] = 1;     u[n] = 0;              /* Fronteira prescrita*/
    //Bv[4][i] = -Pi*cos(Pi*y[p.v[1]])*2;  /* Front. nao prescrita*/
  }
 }
}
```

```
mat_2 Qm(int e)
{  mat_2  q;
   vec_2i p;
   float  x1,x2;
   p = ElmPos(e);
   x1 = x[p.v[0]];  x2 = y[p.v[1]];
   q.m[0][0] = 2 + 0*x1;
   q.m[1][1] = 2 + 0*x2;
   q.m[0][1] = 1;
   q.m[1][0] = q.m[0][1];
   return q;
}


float Load(float x, float y)       /*Forca prescrita*/
{
   return (2*Pi*Pi*(2*sin(Pi*x)*cos(Pi*y)+cos(Pi*x)*sin(Pi*y)));
}


void EqNo()
{  int n,e;
   for (e=1,n=1; n<=Nno; n++)
   {
     if (typ[n]!=1)
     {
        eqn[n]=e; Neq=e; e++;
     }
     else eqn[n]=0;
}  }


mat_2 Hess(int e)        /* Matriz Hessiana*/
{  float dx,dy;
   vec_2i x1,x2,x3;
   mat_2 m={{{0,0},{0,0}}};
   x1 = NoPos(NoLG(0,e));
   x2 = NoPos(NoLG(1,e));
   x3 = NoPos(NoLG(2,e));
   dx = x[x2.v[0]]-x[x1.v[0]];
   dy = y[x3.v[1]]-y[x2.v[1]];
```

```
   m.m[0][0]= 2./dx;
   m.m[1][1]= 2./dy;
   return m;
}


float Jacobi(int e)
{  float J;
   mat_2 m;
   m = Hess(e);
   J = 1./(m.m[0][0]*m.m[1][1]);
   return J;
}


void GlobalSystem(void)
{  int e,i,j,a,b;

   for (e=1; e<=Nel; e++)
   {


       LocalSystem(e);
       for (a=0; a<=3; a++)
       {
         i = eqn[NoLG(a,e)];  if (i==0) goto ipass;
         F[i-1] += Fa.v[a];
         for (b=0; b<=3; b++)
         {
           j = eqn[NoLG(b,e)];  if (j==0 || j<i) goto jpass;
           K[i-1][j-i] += Kab.m[a][b];
           jpass:;
         }
         ipass:;
   }    }
   TractionBoundary();
}

void LocalSystem(int e)
{  int i,j,a,b;
   float  J;
```

```
   vec_2i p;
   mat_2  m,Q;
   float  dH[2];
   mat_4  X={{{0,0,0,0},{0,0,0,0},{0,0,0,0},{0,0,0,0}}};
   vec_4  qa={{0,0,0,0}},z={{0,0,0,0}};

   m = Hess(e);
   dH[0] = m.m[0][0];
   dH[1] = m.m[1][1];
   J  = Jacobi(e);
   Q = Qm(e);

   for (a=0; a<=3; a++)
   for (b=a; b<=3; b++)
   for (i=0; i<=1; i++)
   for (j=0; j<=1; j++)
       X.m[a][b] += J* Qabij[a][b][i][j] * Q.m[i][j] *dH[i]*dH[j];

   for (a=0; a<=3; a++)
   for (b=0; b<=3; b++)
   {
       p = NoPos(NoLG(b,e));
       z.v[a] += J*Qab[a][b]*Load(x[p.v[0]],y[p.v[1]]);
   }

   for (a=1; a<=3; a++)
   for (b=0; b<a; b++)
       X.m[a][b] = X.m[b][a];
   Kab = X;
   Fa  = z;
                          /* Termos da fronteira de Dirichlet */
   for (a=0; a<=3; a++)
   {  i = NoLG(a,e);
      if (typ[i]==1)  qa.v[a]=u[i];
}

   z = Prod44_41(Kab,qa);
   for (a=0; a<=3; a++)  Fa.v[a] -= z.v[a];
}

void TractionBoundary(void)
```

```
{
   int   j,n,n1,n2,i1,i2;
   float dx,dy,dl;
   vec_2i p1,p2;

   for (n=1; n<=4; n++)
   {
      if (typ[bdy[n][2]]==0)
      {
         for (j=1; j<Nbn[n]; j++)
         {
            n1 = bdy[n][j];   n2 = bdy[n][j+1];
            p1 = NoPos(n1);   p2 = NoPos(n2);
            dx = x[p1.v[0]]-x[p2.v[0]];
            dy = y[p1.v[1]]-y[p2.v[1]];
            dl = sqrt(dx*dx+dy*dy);
            i1 = eqn[n1];  i2 = eqn[n2];
            if (i1!=0)
               F[i1-1] += (Bv[n][j]/3+Bv[n][j+1]/6)*dl;
            if (i2!=0)
               F[i2-1] += (Bv[n][j+1]/3+Bv[n][j]/6)*dl;
   }  }  }
}
void System(void)
{  int i,j;
   fprintf(f1,"\n\nMatriz rigidez K:");
   for (i=0; i<Neq; i++)
   {
       fprintf(f1,"\n");
       for (j=0; j<band; j++) fprintf(f1,"%+6.5f  ", K[i][j]);
   }
   fprintf(f1,"\n\nVetor forca F:\n");
   for (i=0; i<Neq; i++)
       fprintf(f1,"%+6.5f  ", F[i]);
   fprintf(f1,"\n");
}

void Solution(void)
{  int i;
   fprintf(f1,"\nVetor solucao X:");
   for (i=0; i<Neq; i++)
```

```
   {
      if ((i%5)==0)  fprintf(f1,"\n%4d: ", i+1);
      fprintf(f1,"%+6.5f  ", F[i]);
   }
   fprintf(f1,"\n");
}
/* Calcula a derivada de u em relacao as variaveis x e y*/
void Grad_U(float *u)
{
   int    i,k,n,m,e;
   mat_24 A;
   mat_2  B;
   for (i=1; i<=Nno; i++)
   {   Du[0][i] = 0;  Du[1][i] = 0; }

   for (e=1; e<=Nel; e++)
   {
       A = DPhi(-1,-1);
       B = Hess(e);
       m = NoLG(0,e);

       for (k=0; k<=1; k++)
       for (i=0; i<=3; i++)
       {
         n = NoLG(i,e);
         Du[k][m] += (B.m[0][k]*A.m[0][i] + B.m[1][k]*A.m[1][i])*u[n];
}  }   }


vec_2 Norm(float *u)   /* (Norma L2 e H1) */
{  int e,i,a,b,na,nb;
   float  J,da;
   vec_2i x1,x2,x3;
   float  dl[2];
   vec_2  X={{0,0}};

   for (e=1; e<=Nel; e++)
   {
       x1 = NoPos(NoLG(0,e));
       x2 = NoPos(NoLG(1,e));
       x3 = NoPos(NoLG(3,e));
```

```
        dl[0] = x[x2.v[0]]-x[x1.v[0]];
        dl[1] = y[x3.v[1]]-y[x1.v[1]];
        da = dl[0]*dl[1];
        J = da/4;

        for (a=0; a<=3; a++)
        {  na = NoLG(a,e);
           for (b=0; b<=3; b++)
           { nb = NoLG(b,e);
             X.v[0] += J*Qab[a][b]*u[na]*u[nb];
             for (i=0; i<=1; i++)
             X.v[1] += J*Qabij[a][b][i][i]*4/dl[i]/dl[i]*u[na]*u[nb];
   }   }  }
   X.v[0] = sqrt(X.v[0]);
   X.v[1] = sqrt(X.v[1]);
   return X;
}
void Sol_Nodal(void)
{  int n;
   vec_2  er;
   vec_2i p;

   Grad_U(u);
   fprintf(f1,
       "\n No'   Valor    Coord.(x, y)     Err     Du/Dx    Du/Dy");
   for (n=1; n<=Nno; n++)
   {
       p = NoPos(n);
       fprintf(f1,"\n%4d  %+6.5f  (%+.3f, %+.3f)   ",
               n,   u[n],   x[p.v[0]], y[p.v[1]]);
       fprintf(f1,"%+6.5f  %+6.5f  %+6.5f", F[n], Du[0][n], Du[1][n]);
       if (n%(Nelx+1)==0)  fprintf(f1,"\n");
   }

   er = Norm(F);
   fprintf(f1,"\n L^2 error = %e   H^1 error = %e", er.v[0], er.v[1]);
}
```

A.6 Elasticidade linear bidimensional – `elast.cpp`

```
/* Problema de Elasticidade Linear Bidimensional
   Metodo de Elementos finitos
   Elasticidade Linear:  - (C_ijkl(u_k,l)),j = f_i(x)
   bi-dimensional, elementos retangulos
   typ=1: u_i deslocamento prescrito
   typ=0: u_i deslocamento nao prescrito
   Arquivos de Saída:
     elast.out    (dados e solucoes )
     elast.dat (para fazer o grafico): malha deformada
      Esse programa permite que na entrada de dados a escolha
      dos tres tipos de fronteira Dirichlet, Neumann e Misto
      para o problema, cuja solucao exata é:
      u_1 = u_2  = 1\pi^2(\sen\pi x\sen\pi y),
      com x,y no intervalo[0,1].
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include "typdef.h"

#define  Xi0   0.5773502691896  /* 1/sqrt(3)-Gaussiana com 2 pontos */
#define  Pi    3.14159265359
#define  c1    1.5              /* lambda */
#define  c2    1.0              /* mu */
#define  c3    (c1+2*c2)

/* Funcao protótipo */
void   DataInput(void);         /* Entrada de dados */
void   InputData(void);         /* Dados de entrada */
void   Fronteira(void);         /*Identificando os nos da fronteira*/
void   CondFront(int);         /* Condicao de fronteira */
void   TractionBoundary(void);  /* Para os nos da fronteira de Neumann*/
vec_2i NoPos(int);              /* no global -> Posicao  */
vec_2i ElmPos(int e);           /* Posicao no do Elemento */
int    PosNo(vec_2i);           /* Posicao -> no global */
int    NoLG(int a, int e);      /* no local(elemento) -> no global */
void   EqNo(void);              /* Número de no global -> Variável */
```

```
vec_4  Phi(float,float);        /* Funcao teste */
mat_24 DPhi(float,float);       /* Xi-Gradiente da funcao teste */
mat_2  Hess(int);               /* Hessiano D(xi,eta)/D(x,y) */
float  Jacob(int);              /* Jacobiano do Hess^-1, retangulo */
vec_2  Force(float,float);      /* Funcao do carregamento externo */
void   PhiMatriz(void);         /* Integ (Dphi),(phi) do elemento */
void   LocalSystem(int);        /* Rigidez-forca local */
void   GlobalSystem(void);      /* Rigidez-forca global */
void   Restriction(void);       /* Unicidade do Problema Neumann*/
void   Sol_Nodal(void);
void   Ex_f(float **);
void   Solver(float **, float *);
void   PlotData(FILE *);
float  ***m3d_alloc(int,int,int);



FILE  *f1;
int   Nel,Nno,Neq;
int   Nelx,Nely;
int   Ex;               /* Exemplo de fronteira*/
int   band;             /* banda da Matriz Global*/
float *x,*y;            /* x/y-coord do no */
float **u;              /* vetor deslocamento */
float **K;              /* Matriz Rigidez  */
float *F;               /* Carregamento do vetor -> vetor solucao */
int   **bdy,Nbn[5];     /* nos da fronteira e número total de nos da fronteira */
float ***Bv;            /* Valor de fronteira de Neumann */
int   **eqn,**typ;      /* Numero de equacoes e tipo do no*/
mat_8 Kmn;              /* Matriz Rigidez local, (a,i)->m=2a+i */
vec_8 Fm;               /* Vetor local, (a=0..3, i=0..1) */
float Qabij[4][4][2][2];
float Qab[4][4];

/*
Constantes de Elasticidade:
    C[2][2][2][2]=
    { C1111,C1112, C1121,C1122, C1211,C1212, C1221,C1222,
      C2111,C2112, C2121,C2122, C2211,C2212, C2221,C2222 };
*/
float C[2][2][2][2]=
```

```
    { {{{c3,0}, {0,c1}}, {{0,c2}, {c2,0}}},
      {{{0,c2}, {c2,0}}, {{c1,0}, {0,c3}}} };

void main()
{  int  n,i;

   f1 = fopen("elast.out","w");
   DataInput();

   Nel = Nelx*Nely;
   Nno = (Nelx+1)*(Nely+1);
   band= 2*Nelx+6;
   eqn = mi_alloc(Nno+1,2);
   typ = mi_alloc(Nno+1,2);
   u  = m_alloc(Nno+1,2);
   n = (Nelx>=Nely)? Nelx+2 : Nely+2;
   bdy = mi_alloc(5,n);
   Bv  = m3d_alloc(5,n,2);
   Fronteira();
   CondFront(Ex);
   EqNo();
   K = m_alloc(Neq,band);
   F = v_alloc(Neq);
   PhiMatriz();
   InputData();



   GlobalSystem();
   Solver(K,F);
   /* considerando u[n][i] sendo  a solucao U[n][i] */
   for (n=1; n<=Nno; n++)
   for (i=0; i<=1; i++)
       u[n][i] = (typ[n][i]==1)? u[n][i] : F[eqn[n][i]-1];

   if (Ex==2)  /* translacao para u[1][0]=0  */
    {
       for (n=2; n<=Nno; n++)  /*Translacao-Unicidade:Procedimento 3 */
       u[n][0] -= u[1][0];
       u[1][0] = 0;
   }
```

```
   /* Seja K[n][i] sendo o erro, U[n][i]-Ex[n][i] */
   Ex_f(K);
   for (n=1; n<=Nno; n++)
   {
       K[n][0] = u[n][0]-K[n][0];
       K[n][1] = u[n][1]-K[n][1];
   }

   if (Neq==2*Nno) Restriction();
   Sol_Nodal();
   fclose(f1);

   f1=fopen("elast.dat","w");
   PlotData(f1);
   fclose(f1);
}


void DataInput()
{  int i;
   float d;

   printf("\nNumeros de divisao horizontal = ");
   scanf("%d", &Nelx);
   x = v_alloc(Nelx+1);
   d = 1./Nelx;
   for (i=0; i<=Nelx; i++)  x[i] = d*i;

   printf("Numeros de divisao vertical   = ");
   scanf("%d", &Nely);
   y = v_alloc(Nely+1);
   d = 1./Nely;
   for (i=0; i<=Nely; i++)  y[i] = d*i;

   printf("Fronteira de Dirichlet(1), Neumann(2), Misto(3). \\
   Digite o tipo de fronteira = ");
   scanf("%d", &Ex);
}


void Fronteira(void)
```

```
{  int n,i;

   for (n=1,i=1; n<=Nelx+1; n++,i++)
   { bdy[1][i] = n;  Nbn[1] = i; }
   for (n=Nelx+1,i=1; n<=Nno; n+=Nelx+1,i++)
   { bdy[2][i] = n;  Nbn[2] = i; }
   for (n=Nno-Nelx,i=1; n<=Nno; n++,i++)
   { bdy[3][i] = n;  Nbn[3] = i; }
   for (n=1,i=1; n<=Nno-Nelx; n+=Nelx+1,i++)
   { bdy[4][i] = n;  Nbn[4] = i; }
 }
 void CondFront(int Ex)
 {  int i,n;
   vec_2i p;

   for (n=1; n<=Nno; n++)
   for (i=0; i<2; i++)
   { typ[n][i]=0; u[n][i]=0; }

   for (n=1; n<=4; n++)
   for (i=1; i<=Nbn[n]; i++)
   { Bv[n][i][0]=0; Bv[n][i][1]=0; }

   if (Ex==1)
  {
   for (i=1; i<=Nbn[1]; i++)
   {
     n = bdy[1][i];  p = NoPos(n);
     typ[n][0] = 1;  u[n][0] = 0;
     typ[n][1] = 1;  u[n][1] = 0;
   }
   for (i=1; i<=Nbn[2]; i++)
   {
     n = bdy[2][i];  p = NoPos(n);
     typ[n][0] = 1;  u[n][0] = 0;
     typ[n][1] = 1;  u[n][1] = 0;
   }
   for (i=1; i<=Nbn[3]; i++)
   {
     n = bdy[3][i];  p = NoPos(n);
     typ[n][0] = 1;  u[n][0] = 0;
```

```
    typ[n][1] = 1;  u[n][1] = 0;
  }
  for (i=1; i<=Nbn[4]; i++)
  {
    n = bdy[4][i];  p = NoPos(n);
    typ[n][0] = 1;  u[n][0] = 0;
    typ[n][1] = 1;  u[n][1] = 0;
  }
 }

 if (Ex==2)
 {
  for (i=1; i<=Nbn[1]; i++)
  {
    n = bdy[1][i];  p = NoPos(n);
    Bv[1][i][0] = -sin(Pi*x[p.v[0]])/Pi;
    Bv[1][i][1] = -sin(Pi*x[p.v[0]])/Pi*3.5;
  }
  for (i=1; i<=Nbn[2]; i++)
  {
    n = bdy[2][i];  p = NoPos(n);
    Bv[2][i][0] = -sin(Pi*y[p.v[1]])/Pi*3.5;
    Bv[2][i][1] = -sin(Pi*y[p.v[1]])/Pi;
  }
  for (i=1; i<=Nbn[3]; i++)
  {
    n = bdy[3][i];  p = NoPos(n);
    Bv[3][i][0] = -sin(Pi*x[p.v[0]])/Pi;
    Bv[3][i][1] = -sin(Pi*x[p.v[0]])/Pi*3.5;
  }
  for (i=1; i<=Nbn[4]; i++)
  {
    n = bdy[4][i];  p = NoPos(n);
    Bv[4][i][0] = -sin(Pi*y[p.v[1]])/Pi*3.5;
    Bv[4][i][1] = -sin(Pi*y[p.v[1]])/Pi;
  }
   /* Unicidade de solucao*/
   // typ[1][0] = 1;  u[1][0] = 0;      /*Procedimento 2*/
   //typ[1][1] = 1;  u[1][1] = 0;
   //typ[Nelx][1] = 1;  u[Nelx][1] = 0;
   typ[1][1] = 1;  u[1][1] = 0;           /*Procedimento 3:Unicidade*/
```

```
   typ[Nelx][1] = 1;  u[Nelx][1] = 0;
  }

   if (Ex==3)
  {
   for (i=1; i<=Nbn[1]; i++)
   {
     n = bdy[1][i];  p = NoPos(n);
     typ[n][0] = 1;  u[n][0] = 0;
     typ[n][1] = 1;  u[n][1] = 0;
     //Bv[1][i][0] = -sin(Pi*x[p.v[0]])/Pi;
     //Bv[1][i][1] = -sin(Pi*x[p.v[0]])/Pi*3.5;
   }
   for (i=1; i<=Nbn[2]; i++)
   {
     n = bdy[2][i];  p = NoPos(n);
     //typ[n][0] = 1;  u[n][0] = -cos(Pi*y[p.v[1]])/Pi/Pi;
     //typ[n][1] = 1;  u[n][1] = 0;
     Bv[2][i][0] = -sin(Pi*y[p.v[1]])/Pi*3.5;
     Bv[2][i][1] = -sin(Pi*y[p.v[1]])/Pi;
   }
   for (i=1; i<=Nbn[3]; i++)
   {
     n = bdy[3][i];  p = NoPos(n);
     typ[n][0] = 1;  u[n][0] = 0;
     typ[n][1] = 1;  u[n][1] = 0;
     //Bv[3][i][0] = -sin(Pi*x[p.v[0]])/Pi;
     //Bv[3][i][1] = -sin(Pi*x[p.v[0]])/Pi*3.5;
   }
   for (i=1; i<=Nbn[4]; i++)
   {
     n = bdy[4][i];  p = NoPos(n);
    // typ[n][0] = 1;  u[n][0] = cos(Pi*y[p.v[1]])/Pi/Pi;
    // typ[n][1] = 1;  u[n][1] = 0;
     Bv[4][i][0] = -sin(Pi*y[p.v[1]])/Pi*3.5;
     Bv[4][i][1] = -sin(Pi*y[p.v[1]])/Pi;
   }
  }
 }

 void Ex_f(float **f)
```

```
{  int    n;
   float  X,Y;
   vec_2i p;

   for (n=1; n<=Nno; n++)
   {
     p = NoPos(n); X = x[p.v[0]]; Y = y[p.v[1]];
     f[n][0] =sin(Pi*X)*sin(Pi*Y)/Pi/Pi; /*Solucao exata na comp. horizontal*/
     f[n][1] =sin(Pi*X)*sin(Pi*Y)/Pi/Pi; /*Solucao exata na comp. vertical*/
}  }

vec_2 Force(float x, float y)
{  vec_2 v;
    v.v[0]= ((c1+3*c2)*sin(Pi*x)*sin(Pi*y)-
            (c1+c2)*cos(Pi*x)*cos(Pi*y));   /* Forca na comp. orizontal*/
   v.v[1]= ((c1+3*c2)*sin(Pi*x)*sin(Pi*y)-
            (c1+c2)*cos(Pi*x)*cos(Pi*y));   /* Forca na comp. vertical*/
   return v;
}

vec_2i NoPos(int n)
{  vec_2i x;
   x.v[0] = (n-1)%(Nelx+1);
   x.v[1] = (n-1)/(Nelx+1);
   return x;
}

int PosNo(vec_2i x)
{
   return (x.v[1]*(Nelx+1) + x.v[0] + 1);
}

vec_2i ElmPos(int e)
{  vec_2i x;
   x.v[0] = (e-1)%Nelx;
   x.v[1] = (e-1)/Nelx;
   return x;
}

int NoLG(int a, int e)
{  vec_2i x;
```

```
   x=ElmPos(e);
   if (a==1) { x.v[0]++;}
   if (a==2) { x.v[0]++; x.v[1]++; }
   if (a==3) { x.v[1]++; }
   return PosNo(x);
}

void EqNo()
{  int n,e,i;
   for (e=1,n=1; n<=Nno; n++)
   for (i=0; i<=1; i++)   /* i=0 componente horiz. i=1 comp. vertical */
   {
     if (typ[n][i]!=1)
     {
        eqn[n][i]=e; Neq=e; e++;
     }
     else eqn[n][i]=0;
   }
}

vec_4 Phi(float xi, float eta)
{  int a;
   float x[4][2]={{-1,-1},{1,-1},{1,1},{-1,1}};
   vec_4 v;

   for (a=0; a<=3; a++)
   {
      v.v[a] = (1.+x[a][0]*xi)*(1.+x[a][1]*eta)/4;
   }
   return v;
}

mat_24 DPhi(float xi, float eta)
{  int a;
   float x[4][2]={{-1,-1},{1,-1},{1,1},{-1,1}};
   mat_24 g;

   for (a=0; a<=3; a++)
   {
       g.m[0][a] = (x[a][0]*(1.+x[a][1]*eta)/4);
       g.m[1][a] = (x[a][1]*(1.+x[a][0]*xi )/4);
```

```
   }
   return g;
}


mat_2 Hess(int e)
{  float dx,dy;
   vec_2i x1,x2,x3;
   mat_2  m={{{0,0},{0,0}}};

   x1 = NoPos(NoLG(0,e));
   x2 = NoPos(NoLG(1,e));
   x3 = NoPos(NoLG(2,e));
   dx = x[x2.v[0]]-x[x1.v[0]];
   dy = y[x3.v[1]]-y[x2.v[1]];
   m.m[0][0]= 2./dx;
   m.m[1][1]= 2./dy;
   return m;
}


float Jacob(int e)
{  float J;
   mat_2 m;

   m = Hess(e);
   J = 1./(m.m[0][0]*m.m[1][1]);
   return J;
}

void PhiMatriz()
{  int i,j,k,a,b;
   float  xi,eta;
   mat_24 B;
   vec_4  g;
   float  q[4][2]={{-Xi0,-Xi0},{Xi0,-Xi0},{Xi0,Xi0},{-Xi0,Xi0}};

   for (k=0; k<=3; k++)   /* Integracao Gaussiana com 4 pontos */
   {
      xi = q[k][0];
      eta= q[k][1];
```

```
      B  = DPhi(xi,eta);
      for (a=0; a<=3; a++)
      for (b=0; b<=3; b++)
      for (i=0; i<=1; i++)
      for (j=0; j<=1; j++)
          Qabij[a][b][i][j] += B.m[i][a] * B.m[j][b];

      g = Phi(xi,eta);
      for (a=0; a<=3; a++)
      for (b=0; b<=3; b++)
          Qab[a][b] += g.v[a] * g.v[b];
} }


void GlobalSystem()
{  int e,i,j,a,b,r,s;

   for (e=1; e<=Nel; e++)
   {
     LocalSystem(e);
     for (a=0; a<=3; a++)
     for (r=0; r<=1; r++)
     {
       i = eqn[NoLG(a,e)][r];  if (i==0) goto ipass;
       F[i-1] += Fm.v[2*a+r];

       for (b=0; b<=3; b++)
       for (s=0; s<=1; s++)
       {
         j = eqn[NoLG(b,e)][s];  if (j==0 || j<i) goto jpass;
         K[i-1][j-i] += Kmn.m[2*a+r][2*b+s];
         jpass:;
       }
       ipass:;
     }
   }
   TractionBoundary();
}

void LocalSystem(int e)
```

```
{  int i,j,k,l,a,b,r,s;
   float  J;
   vec_2  fi;
   vec_2i p;
   mat_2  m;
   float  dH[2];
   mat_8  X={{{0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0},
              {0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0},
              {0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0},
              {0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0}}};
   vec_8  qa={{0,0,0,0,0,0,0,0}};
   vec_8  z={{0,0,0,0,0,0,0,0}},zq={{0,0,0,0,0,0,0,0}};
   float  ei[2][2]={{1,0},{0,1}};    /* base canonica */

   m = Hess(e);
   dH[0] = m.m[0][0];
   dH[1] = m.m[1][1];
   J  = Jacob(e);

   for (a=0; a<=3; a++)
   for (r=0; r<=1; r++)
   for (b=0; b<=3; b++)
   for (s=0; s<=1; s++)
   {
     for (i=0; i<=1; i++)
     for (j=0; j<=1; j++)
     for (k=0; k<=1; k++)
     for (l=0; l<=1; l++)
         X.m[2*a+r][2*b+s] += Qabij[a][b][i][j] * C[i][k][j][l]
            * J*dH[i]*dH[j] * ei[r][k] * ei[s][l];
   }
   Kmn = X;

   for (a=0; a<=3; a++)
   for (i=0; i<=1; i++)
   for (b=0; b<=3; b++)
   {
     p = NoPos(NoLG(b,e));
     fi= Force(x[p.v[0]],y[p.v[1]]);
     z.v[2*a+i] += J*Qab[a][b]* fi.v[i];
   }
```

```
   Fm  = z;

   /* Termos da Fronteira de Dirichlet */
   for (a=0; a<=3; a++)
   { k = NoLG(a,e);
     for (r=0; r<=1; r++)
     if (typ[k][r]==1) qa.v[2*a+r] = u[k][r];
   }
   for (i=0; i<=7; i++)
   for (j=0; j<=7; j++)
       zq.v[i] += Kmn.m[i][j]*qa.v[j];
   for (i=0; i<=7; i++)
   Fm.v[i] -= zq.v[i];
}


/* Contribuicao da Fronteira de Neumann na Forca */
void TractionBoundary(void)
{
   int   j,s,n,n1,n2,i1,i2;
   float dx,dy,dl;
   vec_2i p1,p2;

   for (n=1; n<=4; n++)
   for (s=0; s<=1; s++)
   for (j=1; j<Nbn[n]; j++)
   {
      n1 = bdy[n][j];   n2 = bdy[n][j+1];
      p1 = NoPos(n1);   p2 = NoPos(n2);
      dx = x[p1.v[0]]-x[p2.v[0]];
      dy = y[p1.v[1]]-y[p2.v[1]];
      dl = sqrt(dx*dx+dy*dy);
      i1 = eqn[n1][s];  i2 = eqn[n2][s];
      if (i1!=0)
         F[i1-1] += (Bv[n][j][s]/3+Bv[n][j+1][s]/6)*dl;
      if (i2!=0)
         F[i2-1] += (Bv[n][j+1][s]/3+Bv[n][j][s]/6)*dl;
   }
}
```

```
/*
Sistema Linear AX=B, solucao X -> B
Algoritmo de  Crout  A e simetrica
*/
void Solver(float **A, float *B)
{  int   i,j,k;
   float t;
   float tol=1.e-8;

   /* U'DU - Decomposicao de Crout */
   for (j=1; j<Neq; j++)
   {
      for (i=((j-band+2)>0)? j-band+2 : 1; i<=j-1; i++)
      for (k=((j-band+1)>0)? j-band+1 : 0; k<=i-1; k++)
        A[i][j-i] -= A[k][i-k]*A[k][j-k];
      for (i=((j-band+1)>0)? j-band+1 : 0; i<=j-1; i++)
      {
        t = A[i][j-i];
        if (fabs(A[i][0])<tol)
           { printf("err 1: A[%d,%d] < %e",i+1,i+1,tol); exit(1);}
        A[i][j-i] = t/A[i][0];
        A[j][0] -= t*A[i][j-i];
        if (fabs(A[j][0])<tol)
           { printf("err 2: A[%d,%d] < %e",j+1,j+1,tol); exit(1);}
   }  }
   /* Solucao do Sistema, considerando a banda */
   for (j=1; j<Neq; j++)
   for (i=((j-band+1)>0)? j-band+1 : 0; i<=j-1; i++)
       B[j] -= A[i][j-i]*B[i];

   for (j=0; j<Neq; j++)
       B[j] = B[j]/A[j][0];

   for (j=Neq-1; j>=1; j--)
   for (i=((j-band+1)>0)? j-band+1 : 0; i<=j-1; i++)
       B[i] -= A[i][j-i]*B[j];
}

void InputData()
{  int e,n,a,i;
   fprintf(f1,"\nNo's dos elementos:");
```

```
   for (e=1; e<=Nel; e++)
   {
     fprintf(f1,"\nElm %4d:", e);
     for (a=0; a<=3; a++)
     {
       fprintf(f1,"   %4d", NoLG(a,e));
   } }
   fprintf(f1,"\n");
   fprintf(f1,"\n No'  H&V  Eqn  Tipo");
   for (n=1; n<=Nno; n++)
   for (i=0; i<=1; i++)
   {
     fprintf(f1,"\n%4d   %d   %4d    %d",
            n, i+1, eqn[n][i],typ[n][i]);
   }
   for (n=1; n<=4; n++)
   {
     fprintf(f1,"\n bdy: %d", n);
     fprintf(f1,"\n  No      typ(x,y)   valor Bv(x,y)");
     for (i=1; i<=Nbn[n]; i++)
     {
       fprintf(f1,"\n %2d: %3d  (%d, %d)    (%+f, %+f)", i, bdy[n][i],
       typ[bdy[n][i]][0], typ[bdy[n][i]][0], Bv[n][i][1], Bv[n][i][1]);
     }
}  }


vec_2 Norm(float **u)   /* (Normas L2 e H1 ) */
{  int e,i,a,b,na,nb;
   float  J,da;
   vec_2i x1,x2,x3;
   float  dl[2];
   vec_2  X={{0,0}};

   for (e=1; e<=Nel; e++)
   {
     x1 = NoPos(NoLG(0,e));
     x2 = NoPos(NoLG(1,e));
     x3 = NoPos(NoLG(2,e));
     dl[0] = x[x2.v[0]]-x[x1.v[0]];
     dl[1] = y[x3.v[1]]-y[x2.v[1]];
```

```
     da = dl[0]*dl[1];
     J = da/4;

     for (a=0; a<=3; a++)
     {
       na = NoLG(a,e);
       for (b=0; b<=3; b++)
       {
         nb = NoLG(b,e);
         X.v[0] += J* Qab[a][b] *
         (u[na][0]*u[nb][0] + u[na][1]*u[nb][1]);
     for (i=0; i<=1; i++)
         X.v[1] += J * Qabij[a][b][i][i] * 4/dl[i]/dl[i] *
         (u[na][0]*u[nb][0] + u[na][1]*u[nb][1]);
   } } }
   X.v[0] = sqrt(X.v[0]);
   X.v[1] = sqrt(X.v[1]);
   return X;
}

float Integral(float *f)
{  int e,a,n;
   float  J;
   float  X=0;

   for (e=1; e<=Nel; e++)
   {
     J = Jacob(e);
     for (a=0; a<=3; a++)
     {
       n = NoLG(a,e);
       X += J * f[n];
     }
   }
   return X;
}

/* Procedimento 1:Unicidade para o Problema de Neumann */
void Restriction()
{  int n;
   vec_2i p;
```

```
float b1,b2,b3,I0,I1,I2,I3;
float D,W[3][3],C1,C2;

for (n=1; n<=Nno; n++) F[n] = 1;
I0 = Integral(F);
for (n=1; n<=Nno; n++)
{
  p = NoPos(n);  F[n] = x[p.v[0]];
}
I1 = Integral(F);
for (n=1; n<=Nno; n++)
{
  p = NoPos(n);  F[n] = y[p.v[1]];
}
I2 = Integral(F);
for (n=1; n<=Nno; n++)
{
  p = NoPos(n);
  F[n] = x[p.v[0]]*x[p.v[0]]+y[p.v[1]]*y[p.v[1]];
}
I3 = Integral(F);
D = I0*(I0*I3-I1*I1-I2*I2);
W[0][0] = -I0*I2/D; W[1][2] = W[0][0];
W[0][1] =  I0*I1/D; W[2][2] = W[0][1];
W[0][2] =  I0*I0/D;
W[1][1] = -I1*I2/D; W[2][0] = W[1][1];
W[1][0] = (I0*I3-I1*I1)/D;
W[2][1] = (I0*I3-I2*I2)/D;

do{
    for (n=1; n<=Nno; n++) F[n] = -K[n][0];
    b1 = Integral(F);
    for (n=1; n<=Nno; n++) F[n] = -K[n][1];
    b2 = Integral(F);
    for (n=1; n<=Nno; n++)
    {
       p = NoPos(n);
   F[n] = -K[n][0]*y[p.v[1]]+K[n][1]*x[p.v[0]];
    }
    b3 = Integral(F);
```

```
        I0 = W[0][0]*b1 + W[0][1]*b2 + W[0][2]*b3;
        I1 = W[1][0]*b1 + W[1][1]*b2 + W[1][2]*b3;
        I2 = W[2][0]*b1 + W[2][1]*b2 + W[2][2]*b3;
        printf("\nw = %+e   c1 = %+e   c2 = %+e", I0,I1,I2);

        for (n=1; n<=Nno; n++)
        {
          p = NoPos(n);
          C1 =  I0*y[p.v[1]]+I1;
          C2 = -I0*x[p.v[0]]+I2;
          u[n][0] = u[n][0] + C1;
          u[n][1] = u[n][1] + C2;
          K[n][0] = K[n][0]+C1;
          K[n][1] = K[n][1]+C2;
        }
    } while (fabs(I0)+fabs(I1)+fabs(I2) > 1e-4);
}

void Sol_Nodal()
{  int n;
   float  a1,a2;
   vec_2  er;
   vec_2i p;

   fprintf(f1,
   "\n No'   Coord.(x, y)      u_1       u_2        err_1        err_2");
   for (n=1; n<=Nno; n++)
   {
     p = NoPos(n);
     fprintf(f1,"\n%4d  (%+.3f, %+.3f)   %+.5f  %+.5f",
            n,  x[p.v[0]], y[p.v[1]], u[n][0], u[n][1]);
     fprintf(f1,"   %+.4e  %+.4e", K[n][0], K[n][1]);
     if (n%(Nelx+1)==0)  fprintf(f1,"\n");
   }
   er = Norm(K);
   fprintf(f1,"\nL^2 err = %e   H^1 err = %e", er.v[0], er.v[1]);
   a1 = er.v[0]; a2 = er.v[1];
   er = Norm(u);
   fprintf(f1,"\nL^2 sol = %e   H^1 sol = %e", er.v[0], er.v[1]);
   fprintf(f1,"\nError relativo em L^2  = %.3f \%", a1/er.v[0]*100);
   fprintf(f1,"\nError relativo em H^1  = %.3f \%", a2/er.v[1]*100);
```

```
}

void PlotData(FILE *f1)
{ int n,i,j;

  for (j=0; j<=Nely; j++)
  {
    for (i=0; i<=Nelx; i++)
    {
      n = j*(Nelx+1)+i+1;
      fprintf(f1,"\n %+f %+f", x[i]+u[n][0], y[j]+u[n][1]);
    }
    fprintf(f1,"\n");
  }
  for (i=0; i<=Nelx; i++)
  {
    for (j=0; j<=Nely; j++)
    {
      n = j*(Nelx+1)+i+1;
      fprintf(f1,"\n %+f %+f", x[i]+u[n][0], y[j]+u[n][1]);
    }
    fprintf(f1,"\n");
  }
}
```

A.7 Equação do calor unidimensional –Calor.cpp

```
/*
    Metodo de Elementos Finitos
    Condução de calor, meio anisotropico
       -(K_ij (U_,j)),i = f(x)
    bi-dimensional: elementos retangulos
    tipo=1: U    deslocamento prescrito
    tipo=0:  deslocamento nao prescrito
    Arquivo de Saida:
       calor.out   dados e solucoes
       calor.dat   3D-plot arquivo (para fazer o grafico)
      Esse programa permite que na entrada de dados a escolha dos tres tipos
      de fronteira Dirichlet, Neumann e Misto para o problema, cuja solucao
      exata é:u = sen(pi*x)sen(pi*y),  com x,y no intervalo[0,1].
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

FILE  *f1;
int   Nely;              /* Numero de elementos y*/
int   Ex;               /* Escolha do exemplo a ser rodado */
int   band;             /*Banda da matriz*/
float *y;
float **K;              /* Matriz Rigidez */
float *F;               /* Vetor  Carregamento-> vetor solucao-> erro */
float *u;              /* Vetor deslocamento */
int   **bdy,Nbn[5];     /*Identificando os nós da fronteira*/
float **Bv;            /* Valor de fronteira*/
float *uv;
float *Du[2];

#include "typdef.h"
#include "grid.h"
#include "solver.h"

/* Função protótipo*/
void   DataInput(void);         /* Entrada de dados */
```

```
void   InputData(void);          /* Dados de entrada */
void   Fronteira(void);          /*Identificando os nos da fronteira*/
void   CondFront(int);          /* Condicao de fronteira */
void   EqNo(void);               /* Número de No global -> Variavel */
mat_2  Hess(int);                /* Hessiano D(xi,eta)/D(x,y) */
float  Jacobi(int);              /* Jacobiano do Hess^-1, retangulo */
mat_2  Qm(int e);                /* Matriz condutividade do elemento */
float  Load(float,float);        /* Funcao do carregamento externo */
void   LocalSystem(int);         /* Rigidez-força local */
void   GlobalSystem(void);       /* Rigidez-força global */
void   TractionBoundary(void);   /* Para os nos da fronteira de Neumann*/
vec_2i NoPos(int);               /* No global -> Posicao  */
void   PhiMatriz(void);          /* Integ (Dphi),(phi) do elemento */
void   System(void);             /* Kab, Fa, Sistema Kab Xb = Fa */
void   Solution(void);           /* Solucao Xb */
void   Sol_Nodal(void);

#define  Xi0   0.5773502691896  /* 1/sqrt(3)- Gaussiana com dois pontos*/
#define  Pi    3.14159265359
mat_4 Kab;
vec_4 Fa;



void main()
{
   int    n;
   vec_2i p;
   f1 = fopen("calor.out","w");   /*Arquivo de  saida*/
   DataInput();
   Nel = Nelx*Nely;
   Nno = (Nelx+1)*(Nely+1);
   band= Nelx+3;
   eqn = vi_alloc(Nno+1);
   typ = vi_alloc(Nno+1);
   u  = v_alloc(Nno+1);
   uv =  v_alloc(Nno+1);
   n = (Nelx>=Nely)? Nelx+2 : Nely+2;
   bdy = mi_alloc(5,n);
   Bv  = m_alloc(5,n);
```

```
Fronteira();
   CondFront(Ex);
   EqNo();
   K = m_alloc(Neq,band);
   F = v_alloc(Nno+1);
   Du[0] = v_alloc(Nno+1);
   Du[1] = v_alloc(Nno+1);

   PhiMatriz();
   GlobalSystem();
   LU_Decomp(K);
   Solver(K,F,uv);

   GridData(f1);
   /* u[n] é a solução */
   for (n=1; n<=Nno; n++)
       u[n] = (typ[n]==1)? u[n] : F[eqn[n]-1];
    /* solução exata: sen(Pi*x)*cos(Pi*y)
   /* F[n] é o erro , U[n]-exact[n] */
   for (n=1; n<=Nno; n++)
   {
       p  = NoPos(n);
       F[n] = u[n] - sin(Pi*x[p.v[0]])*cos(Pi*y[p.v[1]]);
   }

   Sol_Nodal();
   fclose(f1);
   f1=fopen("calor.dat","w");    /*Arquivo para fazer o grafico*/
   PlotData(f1,u);
   fclose(f1);
}


void DataInput(void)
{  int i;
   float d;

   printf("\n\nNumeros de x-divisao = ");
   scanf("%d", &Nelx);
   x = v_alloc(Nelx+1);
   d = 1./Nelx;
```

```
   for (i=0; i<=Nelx; i++)  x[i] = d*i;

   printf("Numeros de y-divisao = ");
   scanf("%d", &Nely);
   y = v_alloc(Nely+1);
   d = 1./Nely;
   for (i=0; i<=Nely; i++)  y[i] = d*i;

   printf("Fronteira de Dirichlet(1), Neumann(2), Misto(3).\\
    Digite o tipo de fronteira = ");
   scanf("%d", &Ex);
}


void Fronteira(void)
{  int n,i;

   for (n=1,i=1; n<=Nelx+1; n++,i++)
   { bdy[1][i] = n;  Nbn[1] = i; }
   for (n=Nelx+1,i=1; n<=Nno; n+=Nelx+1,i++)
   { bdy[2][i] = n;  Nbn[2] = i; }
   for (n=Nno-Nelx,i=1; n<=Nno; n++,i++)
   { bdy[3][i] = n;  Nbn[3] = i; }
   for (n=1,i=1; n<=Nno-Nelx; n+=Nelx+1,i++)
   { bdy[4][i] = n;  Nbn[4] = i; }
 }

void CondFront(int Ex)
{  int i,n;
   vec_2i p;

   for (n=1; n<=Nno; n++)
   {typ[n]=0;  u[n]=0;}

  for (n=1; n<=4; n++)
   for (i=1; i<=Nbn[n]; i++)
    Bv[n][i]=0;

if (Ex==1)
  {
    for (i=1; i<=Nbn[1]; i++)
```

```
  {
    n = bdy[1][i];  p = NoPos(n);
     typ[n] = 1;     u[n] = sin(Pi*x[p.v[0]]);   /* Fronteira prescrita*/
  }

  for (i=1; i<=Nbn[2]; i++)
  {
    n = bdy[2][i];  p = NoPos(n);
    typ[n] = 1;     u[n] = 0;                    /* Fronteira prescrita*/
  }
  for (i=1; i<=Nbn[3]; i++)
  {
    n = bdy[3][i];  p = NoPos(n);
    typ[n] = 1;     u[n] = -sin(Pi*x[p.v[0]]);     /* Fronteira prescrita*/
  }
  for (i=1; i<=Nbn[4]; i++)
  {
    n = bdy[4][i];  p = NoPos(n);
     typ[n] = 1;     u[n] = 0;                   /* Fronteira prescrita*/
  }
 }

 if (Ex==2)
 {
  for (n=1; n<=Nno; n++) typ[n]=0;

  for (i=1; i<=Nbn[1]; i++)
  {
    n = bdy[1][i];  p = NoPos(n);
    Bv[1][i] =  -Pi*cos(Pi*x[p.v[0]]);          /* Fronteira nao prescrita*/
  }
for (i=1; i<=Nbn[2]; i++)
  {
    n = bdy[2][i];  p = NoPos(n);
    Bv[2][i] = -Pi*cos(Pi*y[p.v[1]])*2;         /* Fronteira nao prescrita*/
  }

  for (i=1; i<=Nbn[3]; i++)
  {
    n = bdy[3][i];  p = NoPos(n);
    Bv[3][i] = -Pi*cos(Pi*x[p.v[0]]);          /* Fronteira nao prescrita*/
```

```
  }

  for (i=1; i<=Nbn[4]; i++)
  {
    n = bdy[4][i];  p = NoPos(n);
    Bv[4][i] = -Pi*cos(Pi*y[p.v[1]])*2;        /* Fronteira nao prescrita*/
  }

    typ[1] = 1;     u[1] = 0;         /* Fixando o no global, para unicidade*/
  }

  if (Ex==3)
 {
  for (n=1; n<=Nno; n++) typ[n]=0;

  for (i=1; i<=Nbn[1]; i++)
  {
    n = bdy[1][i];  p = NoPos(n);
    // typ[n] = 1;     u[n] = sin(Pi*x[p.v[0]]);    /* Fronteira prescrita*/
    Bv[1][i] =  -Pi*cos(Pi*x[p.v[0]]);              /* Fronteira nao prescrita*/
  }

  for (i=1; i<=Nbn[2]; i++)
  {
    n = bdy[2][i];  p = NoPos(n);
    typ[n] = 1;     u[n] = 0;                       /* Fronteira prescrita*/
     //Bv[2][i] = -Pi*cos(Pi*y[p.v[1]])*2;          /* Fronteira nao prescrita*/
  }

  for (i=1; i<=Nbn[3]; i++)
  {
    n = bdy[3][i];  p = NoPos(n);
    //typ[n] = 1;     u[n] = -sin(Pi*x[p.v[0]]);   /* Fronteira prescrita*/
     Bv[3][i] = -Pi*cos(Pi*x[p.v[0]]);             /* Fronteira nao prescrita*/
  }

  for (i=1; i<=Nbn[4]; i++)
  {
    n = bdy[4][i];  p = NoPos(n);
    typ[n] = 1;     u[n] = 0;                    /* Fronteira prescrita*/
    //Bv[4][i] = -Pi*cos(Pi*y[p.v[1]])*2;        /* Fronteira nao prescrita*/
```

```
   }
  }
 }


mat_2 Qm(int e)
{  mat_2  q;
   vec_2i p;
   float  x1,x2;
   p = ElmPos(e);
   x1 = x[p.v[0]];  x2 = y[p.v[1]];
   q.m[0][0] = 2 + 0*x1;
   q.m[1][1] = 2 + 0*x2;
   q.m[0][1] = 1;
   q.m[1][0] = q.m[0][1];
   return q;
}


float Load(float x, float y)       /*Forca prescrita*/
{
   return (2*Pi*Pi*(2*sin(Pi*x)*cos(Pi*y)+cos(Pi*x)*sin(Pi*y)));
}


void EqNo()
{  int n,e;
   for (e=1,n=1; n<=Nno; n++)
   {
     if (typ[n]!=1)
     {
        eqn[n]=e; Neq=e; e++;
     }
     else eqn[n]=0;
}  }


mat_2 Hess(int e)        /* Matriz Hessiana*/
{  float dx,dy;
   vec_2i x1,x2,x3;
   mat_2 m={{{0,0},{0,0}}};
```

```
   x1 = NoPos(NoLG(0,e));
   x2 = NoPos(NoLG(1,e));
   x3 = NoPos(NoLG(2,e));
   dx = x[x2.v[0]]-x[x1.v[0]];
   dy = y[x3.v[1]]-y[x2.v[1]];
   m.m[0][0]= 2./dx;
   m.m[1][1]= 2./dy;
   return m;
}


float Jacobi(int e)
{  float J;
   mat_2 m;
   m = Hess(e);
   J = 1./(m.m[0][0]*m.m[1][1]);
   return J;
}


void GlobalSystem(void)
{  int e,i,j,a,b;

   for (e=1; e<=Nel; e++)
   {


       LocalSystem(e);
       for (a=0; a<=3; a++)
       {
         i = eqn[NoLG(a,e)];  if (i==0) goto ipass;
         F[i-1] += Fa.v[a];
         for (b=0; b<=3; b++)
         {
           j = eqn[NoLG(b,e)];  if (j==0 || j<i) goto jpass;
           K[i-1][j-i] += Kab.m[a][b];
           jpass:;
         }
         ipass:;
   }    }
   TractionBoundary();
```

```
}

void LocalSystem(int e)
{  int i,j,a,b;
   float  J;
   vec_2i p;
   mat_2  m,Q;
   float  dH[2];
   mat_4  X={{{0,0,0,0},{0,0,0,0},{0,0,0,0},{0,0,0,0}}};
   vec_4  qa={{0,0,0,0}},z={{0,0,0,0}};

   m = Hess(e);
   dH[0] = m.m[0][0];
   dH[1] = m.m[1][1];
   J  = Jacobi(e);
   Q = Qm(e);

   for (a=0; a<=3; a++)
   for (b=a; b<=3; b++)
   for (i=0; i<=1; i++)
   for (j=0; j<=1; j++)
       X.m[a][b] += J* Qabij[a][b][i][j] * Q.m[i][j] *dH[i]*dH[j];

   for (a=0; a<=3; a++)
   for (b=0; b<=3; b++)
   {
       p = NoPos(NoLG(b,e));
       z.v[a] += J*Qab[a][b]*Load(x[p.v[0]],y[p.v[1]]);
   }

   for (a=1; a<=3; a++)
   for (b=0; b<a; b++)
       X.m[a][b] = X.m[b][a];
   Kab = X;
   Fa  = z;

   /* Termos da fronteira de Dirichlet */
   for (a=0; a<=3; a++)
   {  i = NoLG(a,e);
      if (typ[i]==1)  qa.v[a]=u[i];
}
```

```
   z = Prod44_41(Kab,qa);
   for (a=0; a<=3; a++)  Fa.v[a] -= z.v[a];
}

void TractionBoundary(void)
{
   int   j,n,n1,n2,i1,i2;
   float dx,dy,dl;
   vec_2i p1,p2;

   for (n=1; n<=4; n++)
   {
      if (typ[bdy[n][2]]==0)
      {
         for (j=1; j<Nbn[n]; j++)
         {
            n1 = bdy[n][j];   n2 = bdy[n][j+1];
            p1 = NoPos(n1);   p2 = NoPos(n2);
            dx = x[p1.v[0]]-x[p2.v[0]];
            dy = y[p1.v[1]]-y[p2.v[1]];
            dl = sqrt(dx*dx+dy*dy);
            i1 = eqn[n1];  i2 = eqn[n2];
            if (i1!=0)
               F[i1-1] += (Bv[n][j]/3+Bv[n][j+1]/6)*dl;
            if (i2!=0)
               F[i2-1] += (Bv[n][j+1]/3+Bv[n][j]/6)*dl;
   }  }  }
}
void System(void)
{  int i,j;
   fprintf(f1,"\n\nMatriz rigidez K:");
   for (i=0; i<Neq; i++)
   {
       fprintf(f1,"\n");
       for (j=0; j<band; j++) fprintf(f1,"%+6.5f  ", K[i][j]);
   }
   fprintf(f1,"\n\nVetor forca F:\n");
   for (i=0; i<Neq; i++)
       fprintf(f1,"%+6.5f  ", F[i]);
   fprintf(f1,"\n");
```

```
}

void Solution(void)
{  int i;
   fprintf(f1,"\nVetor solucao X:");
   for (i=0; i<Neq; i++)
   {
      if ((i%5)==0)  fprintf(f1,"\n%4d: ", i+1);
      fprintf(f1,"%+6.5f  ", F[i]);
   }
   fprintf(f1,"\n");
}
/* Calcula a derivada de u em relacao as variaveis x e y*/
void Grad_U(float *u)
{
   int    i,k,n,m,e;
   mat_24 A;
   mat_2  B;
   for (i=1; i<=Nno; i++)
   {   Du[0][i] = 0;  Du[1][i] = 0; }

   for (e=1; e<=Nel; e++)
   {
       A = DPhi(-1,-1);
       B = Hess(e);
       m = NoLG(0,e);

       for (k=0; k<=1; k++)
       for (i=0; i<=3; i++)
       {
         n = NoLG(i,e);
         Du[k][m] += (B.m[0][k]*A.m[0][i] + B.m[1][k]*A.m[1][i])*u[n];
}  }   }


vec_2 Norm(float *u)   /* (Norma L2 e H1) */
{  int e,i,a,b,na,nb;
   float  J,da;
   vec_2i x1,x2,x3;
   float  dl[2];
   vec_2  X={{0,0}};
```

```
   for (e=1; e<=Nel; e++)
   {
       x1 = NoPos(NoLG(0,e));
       x2 = NoPos(NoLG(1,e));
       x3 = NoPos(NoLG(3,e));
       dl[0] = x[x2.v[0]]-x[x1.v[0]];
       dl[1] = y[x3.v[1]]-y[x1.v[1]];
       da = dl[0]*dl[1];
       J = da/4;

       for (a=0; a<=3; a++)
       {  na = NoLG(a,e);
          for (b=0; b<=3; b++)
          { nb = NoLG(b,e);
            X.v[0] += J*Qab[a][b]*u[na]*u[nb];
            for (i=0; i<=1; i++)
            X.v[1] += J*Qabij[a][b][i][i]*4/dl[i]/dl[i]*u[na]*u[nb];
   }   }  }
   X.v[0] = sqrt(X.v[0]);
   X.v[1] = sqrt(X.v[1]);
   return X;
}
void Sol_Nodal(void)
{  int n;
   vec_2  er;
   vec_2i p;

   Grad_U(u);
   fprintf(f1,
       "\n No'   Valor    Coord.(x, y)     Err     Du/Dx    Du/Dy");
   for (n=1; n<=Nno; n++)
   {
       p = NoPos(n);
       fprintf(f1,"\n%4d  %+6.5f  (%+.3f, %+.3f)   ",
               n,   u[n],   x[p.v[0]], y[p.v[1]]);
       fprintf(f1,"%+6.5f  %+6.5f  %+6.5f", F[n], Du[0][n], Du[1][n]);
       if (n%(Nelx+1)==0)  fprintf(f1,"\n");
   }

   er = Norm(F);
```

```
   fprintf(f1,"\n L^2 error = %e   H^1 error = %e", er.v[0], er.v[1]);
}
```

A.8 Equação da onda unidimensional –Onda.cpp

```
/*
   Problema Hiperbólico da Equação da Onda Unidimensional
   Metodo de Elementos Finitos
   U_tt(x,t) - (Alpha)U_xx(x,t) + (Beta)U(x,t) = f(x,t)
   tipo=1: u() prescrita
   tipo=2: du() prescrita

   Problema1: Alpha=1 e Beta=0 então
            f=0 e solução exata u(x,t)=sen(Pi*x)cos(lambda*Pi*t)
            u(x,0)= sen(Pi*x) e u'(x,0)=0
           onde lambda=sqrt(alpha+beta/Pi^2)

   Problema2:  Alpha=Beta=1 então (tomando lambda=1)
             f(x,t)=u(x,t)=sen(Pi*x)cos(lambda*Pi*t)(força=solução exata)
              u(x,0)= sen(Pi*x) e u'(x,0)=0

      Método de NEUMARK: U^(*n)=teta*(U^{n-1}+U^{n+1})+(1-2*teta)*U^n
   Se teta=0 então Médodo da Diferença Central
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

#define  Nosm   1002
#define  Nelm   1001
#define  Xi1    -sqrt(3)/3
#define  Xi2    sqrt(3)/3
#define  Pi     3.14159265359
#define  T      1.0
//#define lambda   1.049438509


FILE  *f1,*f2;
int i,j;
int maxj;
int Nel,Nos;                        /*Número de elementos e número de nós */
int tipo1,tipo2;
```

```
float g0[Nosm];                     /*Posição inicial da corda */
float g[Nosm];                      /*Solução aproximada do problema*/
float ganterior[Nosm];              /*g(elevado a menos 1)*/
float g1[Nosm];                     /* Velocidade inicial da corda g´(0)= g1*/
float b[Nosm];                      /*Vetor b do sistema Ag[n+1] = b*/
float G[Nosm];                      /*Solução exata do problema dado*/
float X[Nosm], Xe[3];
float H[Nelm];
float A[Nosm][4];                           /*Matriz A do sistema Ag[n+1] = b */
float B[Nosm][4];                           /*Matriz B do sistema Ag[n+1] = b */
float F[Nosm],FNeg[Nosm],F0[Nosm],F1[Nosm];/*Vetor F (Força) */
float Alpha,Beta,Teta,lambda;
float A1,A2,dl,v[2],k,Max;
float EAbs[Nosm];      /*Erro absoluto */
float E[Nosm];                      /*Diferença da solução aproximada e exata */
float MaxH1;     /*Erro maximo no tempo da norma L2 */
float MaxLInf; /*Erro máximo no tempo da norma H1*/
void InputData()
{
   printf("Constante Alpha = ");
   scanf("%f", &Alpha);
   printf("\nConstante Beta = ");
   scanf("%f", &Beta);
   printf("\nConstante Teta (Teta entre 0 e 1) = ");
   scanf("%f", &Teta);
   printf("\nConstante k (passo no tempo) = ");  /*k=delta t=Passo no tempo*/
   scanf("%f", &k);
   printf("\n\nNumeros de elementos = ");
   scanf("%d", &Nel);
   fprintf(f1,"Alpha = %f\n", Alpha);
   fprintf(f1,"Beta  = %f\n", Beta);
   lambda = sqrt(Alpha+Beta/Pi/Pi);    //valor de lambda p/ Problema 1
    // lambda =1                    // calor de lambda para o problema 2
   fprintf(f1,"Teta  = %f\n", Teta);
   fprintf(f1,"\nConstante k= %f\n", k);
   fprintf(f1,"\nNumero de elementos = %d\n", Nel);
   fprintf(f1,"\nSolução Exata =  sin(Pi*x)*cos(lambda*Pi*t)\n");
}
void InputCondFront()
{
   printf("\nCond. Contorno:(tipo: 1-Dirichlet 2-Neumann)\n");
```

```
   printf("Cond. do no %d (tipo, valor):", 1);
   scanf("%d %f", &tipo1, &A1);
   printf("Cond. do no %d (tipo, valor):", Nos);
   scanf("%d %f", &tipo2, &A2);
   system("pause");
}
 void CoordGlobal()
{
 /* divisao uniforme */
   printf("Entre com a coord. do No' esquerdo:\n");
   scanf("%f", &X[1]);
   printf("Entre com a coord. do No' direito:\n");
   scanf("%f", &X[Nos]);
   dl = (X[Nos]-X[1])/Nel;    /*dl = h,onde h é o passo no espaço */
   for (i=2; i<Nos; i++) X[i]=X[1]+dl*(i-1);

   printf("\nCoord. Global:\n");
   fprintf(f1,"\nCoord. Global:\n");
   for (i=1; i<=Nos; i++)
   {
     printf(" No %d = %f\n", i, X[i]);
     fprintf(f1," No %d = %f\n", i, X[i]);
   }
}
 void CoordLocal(int e)
{
   Xe[1] = X[e];
   Xe[2] = X[e+1];
   H[e] = Xe[2] - Xe[1];
}
 int NoLG(int e,int a)
{
   if (a==1)  return  e;
   else       return  (e+1);
}

float Xi(int e, float x)  /* Transf. perimetrica */
{
   return (Xe[1] + H[e]/2*(x+1));
}
void MatrizA()
```

```
{  int e;
   float A11,A22,A12,A21;
   for (e=1; e<=Nel; e++)
   {
      CoordLocal(e);
      A11 = H[e]/3;
      A22 = H[e]/3;
      A12 = H[e]/6;
      A21 = H[e]/6;
      /* Matriz A */
      A[e][2] += A11;
      A[e][3]  = A12;
      A[e+1][1]  = A21;
      A[e+1][2]  = A22;
   }
}
void MatrizRigidez()
{  int e;
   float B11,B22,B12,B21;
   for (e=1; e<=Nel; e++)
   {
      CoordLocal(e);
      B11 = Alpha/H[e] + Beta/3*H[e];
      B22 = Alpha/H[e] + Beta/3*H[e];
      B12 = -Alpha/H[e] + Beta/6*H[e];
      B21 = -Alpha/H[e] + Beta/6*H[e];
      /* Matriz Rigidez */
      B[e][2] += B11;
      B[e][3]  = B12;
      B[e+1][1]  = B21;
      B[e+1][2]  = B22;
   }
}
float forca(float x)
{
   return 0;                              /*funcao de forca prescrita Problema1*/
/*return sin(Pi*x)*cos(Pi*j*k);*/   /*funcao de forca prescrita Problema2*/
}

float h1(int e, float x)
{
```

```
   return (.5*forca(Xi(e,x))*(1-x));
}

float h2(int e, float x)
{
   return (.5*forca(Xi(e,x))*(1+x));
}
void VetorForca()
{  int e;
   float Fe1,Fe2;
   for (e=1; e<=Nel; e++)
   {
   /* Forca local - Quadratura Gauss com 2 pontos */
      CoordLocal(e);
      Fe1 = H[e]/2*(h1(e,Xi1)+h1(e,Xi2));
      Fe2 = H[e]/2*(h2(e,Xi1)+h2(e,Xi2));
   /* Forca global */
      F[NoLG(e,1)] += Fe1;
      F[NoLG(e,2)]  = Fe2;
   }
}
void CondFront(float M, float N)
{
   switch (tipo1)
   { case 1:{
        A[1][2] = 1;
        A[1][3] = 0;
        B[1][2] = 1;
        B[1][3] = 0;
        F[1]  = M;
        F[2] += -B[2][1]*N;
        A[2][1] = 0;
        B[2][1] = 0;
        break;}
     case 2:{
        F[1] = F[1] - Alpha*M;
        break;}
   }
   switch (tipo2)
   { case 1:{
        F[Nos-1] += -B[Nos-1][3]*N;
```

```
        F[Nos] = N;
        A[Nos-1][3] = 0;
        A[Nos][1] = 0;
        A[Nos][2] = 1;
        B[Nos-1][3] = 0;
        B[Nos][1] = 0;
        B[Nos][2] = 1;
        break;}
     case 2:{
        F[Nos] = F[Nos] + Alpha*N;
        break;}
   }
}
void LinearSolver(float M[Nosm][4], float b[Nosm],float g[Nosm])
{  /*Resolução do Sistema Linear Ag[n+1] = b pelo MEG*/
   float y;
   for (i=1; i<Nos; i++)
   {
       y = M[i+1][1]/M[i][2];     /*Cálculo do multiplicador*/
       M[i+1][2] += -y*M[i][3];   /* transforma A  em triangular superior*/
       b[i+1] += -y*b[i];         /*Atualização do vetor b*/
   }

   /* Retro-Substituicao */
   g[Nos] = b[Nos]/M[Nos][2];
   for (i=Nos-1; i>=1; i--)
       g[i] = (b[i]-M[i][3]*g[i+1])/M[i][2];
}
void Norma(float *f)
{  float L2=0, H1=0, x1, x2;
   for (i=1; i<Nos; i++)
   {
       x1 = f[i+1] + f[i];
       x2 = f[i+1] - f[i];
       L2 += x1*x1;
       H1 += x2*x2;
   }
   v[0] = sqrt(L2/4*dl);                  /* Norma em L2*/
   v[1] = sqrt(L2/4*dl)+sqrt(H1*dl);      /*Norma em H1*/
}
void OutputData()
```

```
{  fprintf(f1,"\nMatriz A:\n");
   for (i=1; i<=Nos; i++)
   {
       for (j=1; j<=3; j++)
         fprintf(f1,"  A(%d,%d)=%f", i,j, A[i][j]);
         fprintf(f1,"\n");
   }
   fprintf(f1,"\nMatriz B (Matriz Rigidez):\n");
   for (i=1; i<=Nos; i++)
   {
       for (j=1; j<=3; j++)
         fprintf(f1,"  B(%d,%d)=%f", i,j, B[i][j]);
         fprintf(f1,"\n");
   }
}
void main()
{
   float a0, a1;
   float M[Nosm][4];
   float L[Nosm][4];
   float Fteta[Nosm];
   f1=fopen("Newmark.out","w");
   f2=fopen("Newmark.dat","w");
   InputData();
   Nos = Nel + 1;
   CoordGlobal();
   InputCondFront();
   MatrizA();
   MatrizRigidez();
   CondFront(A1,A2);                          /* Cond. Contorno: */
   OutputData();
   maxj= int(T/k + 0.5);
   fprintf(f2, "(x =   )   ");
    for(i=1; i<=Nos; ++i)
   {
    fprintf(f2, "%3.4f   ", X[i]);
   }
      for(i=1; i<=Nos; ++i)
   {
  /* A posição g(0) e a velocidade g1 são iguais nos Problemas 1 e 2*/
    fprintf(f2, "%3.4f   ", X[i]);
```

```
   g0[i]=(sin(X[i]*Pi));          /* posição inicial g(0)= g0  */
   g1[i]=(0);                    /* velocidade inicial g´(0) = g1 */
}

 fprintf(f2, "\n\n0.0000     ");
fprintf(f1,"\n\nResultados:\n");
fprintf(f1,"\n  X[i]     Tempo    Aprox.  Exata     Erro    Força \n");

j=-1;
VetorForca();
CondFront(A1,A2);
for (i=1; i<=Nos; i++)
{
 FNeg[i]=F[i];
}

j=0;
VetorForca();
CondFront(A1,A2);
for (i=1; i<=Nos; i++)
{
 F0[i]=F[i];
}

j=1;
VetorForca();
CondFront(A1,A2);
for (i=1; i<=Nos; i++)
{
 F1[i]=F[i];
}
/* Calculo da Matriz Rigidez*/
  for(i=1; i<=Nos; i++)
{
    M[i][1]=A[i][1]+Teta*(k*k)*B[i][1];
    M[i][2]=A[i][2]+Teta*(k*k)*B[i][2];
    M[i][3]=A[i][3]+Teta*(k*k)*B[i][3];
}
  for(i=1; i<=Nos; i++)
{
    L[i][1]=2*A[i][1]-(1-2*Teta)*(k*k)*B[i][1];
```

```
    L[i][2]=2*A[i][2]-(1-2*Teta)*(k*k)*B[i][2];
    L[i][3]=2*A[i][3]-(1-2*Teta)*(k*k)*B[i][3];
}
  /* Tomando a media ponderada da Força inicial com pesos teta*/
  Fteta[1]=Teta*(F1[1]+FNeg[1])+(1-2*Teta)*F0[1];

for(i=2; i<=Nos-1; i++)
{
  Fteta[i]=Teta*(F1[i]+FNeg[i])+(1-2*Teta)*F0[i];
}
  Fteta[Nos]=Teta*(F1[Nos]+FNeg[Nos])+(1-2*Teta)*F0[Nos];
       /* Montagem do vetor b para solução aproximada em j = 0 */

b[1]=0.5*L[1][2]*g0[1]+
0.5*L[1][3]*g0[2]+k*M[1][2]*g1[1]+
k*M[1][3]*g1[2]+.5*k*k*(Fteta[1]);

for (i=2; i<=Nos-1; i++)
{
   b[i]=0.5*L[i][1]*g0[i-1]+0.5*L[i][2]*g0[i]+
   0.5*L[i][3]*g0[i+1]+k*M[i][1]*g1[i-1]+k*M[i][2]*g1[i]
   +k*A[i][3]*g1[i+1]+.5*k*k*(Fteta[i]);
}
b[Nos]=0.5*L[Nos][1]*g0[Nos-1]+ 0.5*L[Nos][2]*g0[Nos]+ M[Nos][1]*g1[Nos-1]
        +M[Nos][2]*g1[Nos]+.5*k*k*(Fteta[Nos]);

LinearSolver(M, b, g);

MaxH1 = 0;
MaxLInf = 0;
for (i=1; i<=Nos; i++)
{
  g[1]=0.0;
    g[Nos]=0.0;
 G[i] =sin(Pi*X[i])*cos(lambda*Pi*j*k);    /* Solução exata inicial para t=0*/
    E[i] = g[i]-G[i];
    EAbs[i]= sqrt((g[i]-G[i])*(g[i]-G[i]));               //Erro absoluto
    fprintf(f1,"%f %f %f %f %f %f\n", X[i], j*k, g[i], G[i], EAbs[i],  Fteta[i]);
    fprintf(f2, "%3.4f   ", g[i]);
    ganterior[i]=g[i];
}
```

```
   for (i=1; i<=Nos; i++)
{
      fprintf(f2, "%3.4f   ", G[i]);
   }

   Norma(G);
   a1 = v[1]; a0 = v[0];
   Norma(E);
   if (MaxLInf<= v[0]) MaxLInf = v[0];
if (MaxH1<= v[1]) MaxH1 = v[1];
   fprintf(f1,"\n erro(L2) = %f", v[0]); // Erro na norma  L2 em cada tempo t
   fprintf(f1,"\n erro(H1) = %f \n", v[1]); //Erro na norma  H1 em cada  tempo t

   fprintf(f1, "\n");
   //Calcula a solução exata e aproximada no ponto medio x=0.5
   //fprintf(f2,"\n %f %f %f %f %f", j*k, v[0],(g[(Nos-1)/2]-G[(Nos-1)/2]),
   //                  g[(Nos-1)/2], G[(Nos-1)/2]);

   for(j=1; j<=maxj; j++)
   {
      j=j+1;
      VetorForca();
   CondFront(A1,A2);
      fprintf(f2, "\n%1.4f     ", (j-1)*k);


       for(i=1; i<=Nos; i++)
      {
          M[i][1]=A[i][1]+Teta*(k*k)*B[i][1];
          M[i][2]=A[i][2]+Teta*(k*k)*B[i][2];
          M[i][3]=A[i][3]+Teta*(k*k)*B[i][3];
      }
       for(i=1; i<=Nos; i++)
      {
          L[i][1]=2*A[i][1]-(1-2*Teta)*(k*k)*B[i][1];
          L[i][2]=2*A[i][2]-(1-2*Teta)*(k*k)*B[i][2];
          L[i][3]=2*A[i][3]-(1-2*Teta)*(k*k)*B[i][3];
      }

       /* Tomando a media ponderada da Força com pesos teta*/
```

```
      Fteta[1]=Teta*(F[1]+F0[1])+(1-2*Teta)*F1[1];

   for(i=2; i<=Nos-1; i++)
   {
      Fteta[i]=Teta*(F[i]+F0[i])+(1-2*Teta)*F1[i];
   }

     Fteta[Nos]=Teta*(F[Nos]+F0[Nos])+(1-2*Teta)*F1[Nos];

    /*Vetor b para solução aproximada em j = 1,2,*/
    b[1]=L[1][2]*ganterior[1]+L[1][3]*ganterior[2]-
       M[1][2]*g0[1]-M[1][3]*g0[2]+ k*k*(Fteta[1]);
   for (i=2; i<=Nos-1; i++)
   {
     b[i]=L[i][1]*ganterior[i-1]+L[i][2]*ganterior[i]+
       L[i][3]*ganterior[i+1]-M[i][1]*g0[i-1]
       -M[i][2]*g0[i]-M[i][3]*g0[i+1]
       +k*k*(Fteta[i]);
   }
  b[Nos]=L[Nos][1]*ganterior[Nos-1]+
    L[Nos][2]*ganterior[Nos]-M[Nos][1]*g0[Nos-1]-
    M[Nos][2]*g0[Nos]+k*k*(Fteta[Nos]);

    LinearSolver(M, b, g);

    for (i=1; i<=Nos; i++)
   {
     g[1]=0.0;
       g[Nos]=0.0;
       G[i] = (sin(Pi*X[i])*cos(lambda*Pi*j*k)); /*Sol.exata para Ploblema 1 e 2*/
     E[i] = g[i]-G[i];
       EAbs[i]= sqrt((g[i]-G[i])*(g[i]-G[i]));                //Erro absoluto
       fprintf(f1,"%f   %f   %f   %f   %f   %f\n" , X[i], (j-1)*k,
       g[i], G[i], EAbs[i], Fteta[i]);
       g0[i]=ganterior[i];
       ganterior[i]=g[i];
       F0[i]=F1[i];
       F1[i]=F[i];
    }
    for (i=1; i<=Nos; i++)
   {
```

```
       fprintf(f2, "%3.4f   ", G[i]);
      }
        Norma(G);
    a1 = v[1]; a0 = v[0];
      Norma(E);
if (MaxLInf<= v[0]) MaxLInf = v[0];
if (MaxH1<= v[1]) MaxH1 = v[1];
      fprintf(f1,"\n erabs(L2) = %f", v[0]);    // Erro norma L^2 em cada tempo t
      fprintf(f1,"\n erabs(H1) = %f \n", v[1]); //Erro na norma H^1 em cada tempo t

      fprintf(f1, "\n");
     //Calcula a solução exata e aproximada no ponto medio x=0.5
   //fprintf(f2,"\n %f %f %f %f %f", j*k, v[0],(g[(Nos-1)/2]-G[(Nos-1)/2]),
   //                  g[(Nos-1)/2], G[(Nos-1)/2]);
      j=j-1;
   }
   fprintf(f1,"\n Erro maximo no tempo em L2 = %f", MaxLInf);
   fprintf(f1,"\n Erro maximo no tempo em H1 = %f", MaxH1);fclose(f1);
   fclose(f2);
}
```

Bibliografia

[1] Atkinson, K., Han, Weimin.: Theoretical Numerical Analysis. A functional Analisys Framework. Springer (2005)

[2] Brézis, H.: Analyse Fonctionelle, Masson Paris (1983).

[3] Burden, R. L., Faires, D. J.: Numerical Analysis, PWS Publishing Company (1993).

[4] Chou, S.-I., Wang, C.-C.: Error estimates of finite element approximations for problems in linear elasticity, Part I. Problems in elastostatics, Arch. Rational Mech. Anal., 72, 41-60 (1979).

[5] Ciarlet, P.: The Finite Element Method for Elliptic Problems, Studies in Mathematics and its Applications, North-Holland, (1978)

[6] Douglas, J. , Dupont, T.: Galerkin methods for parabolic equations, SIAM J. Numer. Anal. 7, 575-626 (1970)

[7] Fichera, G.: Existence Theorems in Elasticity, in Handbuch der Physik, Band VIa/2, Edited by C. Truesdell, Springer-Verlag (1972).

[8] G.H. Golub ; C.F Van Loan: *Matrix Computations*, 3 rd ed., Johns Hopkins U. Press, Bltimore, (1996).

[9] Hughes, T. J. R.: The finite element method - Linear static and dynamic finite element analysis. Prentice Hall (2000)

[10] Johnson, C.: Numerical solution of partial differential equations by the finite element method. Springer (2009)

[11] Larsson, S., Thomée, V.: Partial Differential Equations with Numerical Methods. Springer (2000).

[12] Liu, I-Shih: Continuum Mechanics. Springer (2002).

[13] Medeiros, L. A.: Equações Diferencias Parciais. IM/UFRJ (1981)

[14] Medeiros, L. A., Milla Miranda, M: Espaços de Sobolev (Iniciação aos Problemas Elíticos não homogêneo). IM/UFRJ (2000)

[15] Oden, J. T., Reddy, J. N.: Variational Methods in Theoretical Mechanics, Spring-Verlag (1976).

[16] Oden, J. T., Reddy, J. N.: The mathematical Theory of Finite Elements. New York; Wiley-Interscience, (1976).

[17] Reddy,B. Daya: Introductory Functional Analysis - with applications to boundary value problems and finite elements.Springer,(1998)

[18] Silva, J. A: Equações de Evolução. Dissertação de Mestrado do PPGI/IM/UFRJ, (2009)

[19] Smith, G.D.: Numerical Solution of Partial Differential Equations: Finite Difference Methods. Clarendon Press, Oxford, (1978).

[20] Strang, G.; Fix, G. J.: An analysis of the finite element method, Prentice Hall (1973)

[21] Thomée, V.: From finite differences to finite elements. A short history of numerical analysis of partial differencial equations, Journal of Computational and Applied Mathematics 128, 1-54 (2001).

[22] Thomée, V.: Galerkin Finite Element Methods for Parabolic Problems, Springer-Verlag, 1984.

[23] Larsson,S.; Thomée, V.: Partial Differential Equations with Numerical Methods, Springer, 2009.

[24] Zienkiewicz, O. C.: The Finite Element Method, McGraw-Hill, (1977)

[25] Wheeler, M. F.: A priori L^2 error estimates for Galerkin approximations to parabolic partial differential equations, SIAM J. Numer. Anal. 10, $n^0$4 723-759 (1973)

Índice

Índice