

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

2013

Prefácio

A origem do presente livro data de 1997, quando os autores objetivavam a obtenção de soluções numéricas aproximadas de equações diferenciais parciais do tipo elíptico descrevendo modelos matemáticos da teoria de elasticidade linear. Por meio de seminários foram estudadas existência e unicidade de soluções de alguns problemas e a correspondente formulação variacional. Para obter a solução numérica aproximada, optamos pelo método de elementos finitos. Inicialmente, utilizamos programas computacionais em linguagem Fortran desenvolvidos por outros autores. Tais programas por sua generalidade, não eram didáticos e conseqüentemente dificultavam o entendimento do conteúdo matemático e a adaptação a problemas específicos. Deste modo, desenvolvemos nossos próprios programas computacionais utilizando linguagem C, especificamente, para atender aos nossos objetivos. Eles são simples e de fácil entendimento.

No capítulo 1 são deduzidos alguns modelos da física matemática, tais como, a equação da condução do calor e a teoria de elasticidade linear, que serão estudados, no contexto do método de elementos finitos, nos próximos capítulos deste livro. Para 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 com o objetivo de 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 dos capítulos 6 e 7.

O capítulo 2 trata de um problema modelo estacionário unidimensional de equações diferenciais parciais com condições na fronteira. Alguns exemplos numéricos são exibidos, utilizando vários tipos de valores de fronteira, comparando-se os resultados obtidos com aqueles da solução exata do problema. Os erros nas normas de $L^2(\Omega)$ e $H^1(\Omega)$ são mostrados em cada um dos exemplos dados.

O capítulo 3 introduzimos as mais usuais funções base de ordem superior, ou seja, a função base quadrática, a base spline cúbica e a função base de Hermite. Estimativas de erros em espaços de Sobolev, são feitas para o problema modelo elíptico.

O capítulo 4 apresentamos o problema estacionário bidimensional. São estudadas as formulações forte e fraca, existência e as condições para a unicidade de soluções.

Vários tipos de valores fronteira são considerados e alguns exemplos numéricos são dados juntamente com as respectivas soluções exatas. Para resolver o sistema linear, associado ao método de elementos finitos, utilizamos o algoritmo de Crout. Concluimos o capítulo, mostrando os gráficos das soluções e os erros associados nas normas $L^2(\Omega)$ e $H^1(\Omega)$.

No capítulo 5 apresentamos o problema modelo de elasticidade linear para o caso bidimensional. Os mesmos tópicos do capítulo 4 são abordados, considerando-se, no entanto, a solução vetorial do problema. Deste modo, a dimensão do sistema linear correspondente é aproximadamente o dobro, uma vez que temos, neste caso, as componentes verticais e horizontais. Simulações numéricas, erros e gráficos são mostrados.

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

No capítulo 7, são dados alguns métodos numéricos, muito conhecidos da literatura e seus algoritmos para resolver o problema hiperbólico modelo: **Equação da onda**. São dados alguns exemplos numéricos e comparações entre os diversos métodos numéricos. São também apresentados gráficos e tabelas de erros.

Os capítulos 8 e 9 são capítulos complementares e mais indicados para alunos que querem se aprofundar na análise matemática da equação. Assim não são necessários num primeiro curso para o entendimento dos métodos numéricos desenvolvidos nos capítulos anteriores. São apresentados os resultados teóricos das equações.

No apêndice A, contém com os programas computacionais utilizados para 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 a fonte *typewriter*, por exemplo: **Nel**, **Phi**, **Solver**, *etc*.

Esperamos que este livro, devido à forma simples, porém detalhada com a qual foi escrito, possa constituir um primeiro curso do método de elementos finitos, para alunos 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

1	Introdução	1
1.1	Condução do Calor	1
1.1.1	Equação da Energia	1
1.1.2	Equações Constitutivas	2
1.1.3	Tensor de Condutividade Térmica	3
1.1.4	Condições de Fronteira	4
1.2	Elasticidade Linear	5
1.2.1	Pequena Deformação e Rotação Infinitesimal	6
1.2.2	Equação do Movimento	8
1.2.3	Lei de Hooke	10
1.2.4	Problemas Elastostáticos	11
1.3	Convenção de Somatório	12
1.4	Métodos Numéricos	14
1.4.1	Método da Colocação	15
1.4.2	Método de Galerkin	16
1.4.3	Elementos Finitos	18
1.5	Problemas Variacionais Abstratos	19
1.5.1	Formulação Variacional Abstrata	20
1.5.2	Espaço das Funções Testes	21
1.6	Aproximação por Diferenças Finitas	25
2	Problema Estacionário Unidimensional	29
2.1	Formulação do Problema	29
2.2	Função de Interpolação	33
2.3	Sistema Linear	38
2.4	Matriz Local e Força Local	40
2.5	Matriz Global e Força Global	42
2.6	Integração Numérica	48
2.7	Condições de Fronteira	52
2.8	Programa Computacional	60
2.9	Exemplos	64
2.10	Exercícios	73

3	Função Base e Estimativa de Erro	75
3.1	Função Base de Ordem Superior	75
3.1.1	Base Quadrática	76
3.1.2	Base Cúbica	78
3.1.3	Base de Hermite	84
3.2	Análise de Erro do Problema Estacionário	89
3.2.1	Erro de Interpolação	92
3.2.2	Erro na Norma $H^1(\Omega)$	95
3.2.3	Erro na Norma $L^2(\Omega)$	95
3.2.4	Erro na Norma $H^m(\Omega)$	98
3.3	Erro Numérico	99
3.4	Exercícios	100
4	Problema Estacionário Bidimensional	101
4.1	Formulação do Problema	101
4.2	Discretização do Domínio	107
4.3	Interpolação dos Dados Iniciais	113
4.4	Propriedades da Matriz Rigidez	114
4.5	Função de Interpolação	118
4.6	Quadratura Gaussiana	124
4.7	Construção da Matriz Global e Força Global	136
4.8	Resolução do Sistema Linear	140
4.9	Sistema Linear Global	146
4.10	Erro da Solução Numérica	147
4.11	Entrada e Saída de Dados	149
4.12	Exemplos Numéricos	151
4.13	Unicidade: Problema de Neumann	153
4.14	Exemplos Numéricos	154
4.15	Resultados Numéricos	162
4.16	Exercícios	166
5	Problema de Elasticidade Linear - Caso Bidimensional	167
5.1	Formulação do Problema	167
5.2	Matriz Rigidez e Vetor Força	175
5.3	Sistema Linear	192
5.4	Exemplos Numéricos	192
5.5	Unicidade: Problema de Neumann	199
5.6	Resultados Numéricos	202
5.7	Exercícios	206

6	Métodos Numéricos e Algoritmos: Equação do Calor	207
6.1	Equação Parabólica	207
6.2	Algoritmos para a Equação do Calor	211
6.2.1	Método de Euler Regressivo	212
6.2.2	Método de Euler Progressivo	213
6.2.3	Método de Crank-Nicolson	214
6.2.4	Método Generalizado Trapezoidal: (θ -método)	214
6.3	Simulação Numérica: Equação do Calor	216
6.4	Exercícios	222
7	Métodos Numéricos e Algoritmos: Equação da Onda	224
7.1	Problema Aproximado	225
7.2	Algoritmos para a Equação da Onda	227
7.2.1	Método da Diferença Central	227
7.2.2	Métodos Implícitos - Método de Newmark	228
7.2.3	Método Numérico: θ -método: Equação da Onda	230
7.3	Simulação Numérica: Equação da Onda	232
7.4	Exercícios	237
8	Análise Numérica:Equação do Calor	239
8.1	Estimativa de erro	239
8.1.1	Problema Semidiscreto	239
8.1.2	Problema Discreto	247
8.1.3	Exercício	265
8.2	Existência e unicidade de solução	266
8.3	Propriedades e decaimento assintótico	272
9	Análise Numérica:Equação da Onda	277
9.1	Estimativa de Erro	277
9.1.1	Problema Semidiscreto	277
9.1.2	Problema Discreto	283
9.1.3	Exercícios	298
9.2	Existência e unicidade de solução	299
9.3	Regularidade e Conservação de Energia	303
A	Programas computacionais: linguagem C	305
A.1	Problema estacionário unidimensional – PEU.cpp	306
A.2	Header file – typedef.h	314
A.3	Header file – grid.h	317
A.4	Header file – solver.h	322
A.5	Problema estacionário bidimensional – PEB.cpp	324

A.6 Elasticidade linear bidimensional – <code>elast.cpp</code>	336
A.7 Equação do calor unidimensional – <code>Calor.cpp</code>	355
A.8 Equação da onda unidimensional – <code>Onda.cpp</code>	365
Bibliografia	377
Índice	379

CAPÍTULO 1

Introdução

Neste livro, para desenvolver o método de elementos finitos, utilizaremos equações diferenciais parciais do tipo elíptica. Embora, os modelos da Física-Matemática para a equação de condução do calor e elasticidade linear na sua forma mais geral são do tipo parabólico e hiperbólico respectivamente, ambas se reduzem a problemas do tipo elíptico para problemas estacionários.

A formulação do problema de valor de contorno para a condução do calor será governada por uma equação escalar da temperatura do corpo e para elasticidade linear será governada por um sistema de equações para o vetor deslocamento do corpo.

1.1 Condução do Calor

Consideremos um corpo rígido ocupando uma região $V \subset \mathbb{R}^3$. Seja ϱ e e denotando a densidade da massa e a densidade da energia (interna) do corpo. Sejam $\Omega \subset V$ 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 passando da fronteira para dentro da região e o suprimento de energia r dentro da região devido a fontes externas. Esta relação pode ser representada por

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

Seja \mathbf{n} denotando o vetor normal unitário externo na fronteira e \mathbf{h} o vetor fluxo do calor, então o fluxo de energia entrando no corpo q pode ser expressado como

$$q = -\mathbf{h} \cdot \mathbf{n}. \quad (1.2)$$

Desde que Ω é uma região fixa, então usando o teorema da divergência e assumindo que a fronteira do corpo seja suficientemente suave, obtemos de (1.1) e (1.2) que

$$\int_{\Omega} \left\{ \varrho \frac{\partial e}{\partial t} + \operatorname{div} \mathbf{h} - \varrho r \right\} d\Omega = 0. \quad (1.3)$$

Note que desde que o corpo é rígido, portanto indeformável, a densidade ϱ é independente do tempo. Para obter a relação acima na sua forma local utilizaremos o seguinte teorema:

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

$$\int_{\Omega} f d\Omega = 0 \quad \forall \Omega \subset V,$$

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

Como a relação (1.3) é válida para qualquer $\Omega \subset V$, pelo teorema acima, obtemos

$$\varrho \frac{\partial e}{\partial t} + \operatorname{div} \mathbf{h} - \varrho r = 0. \quad (1.4)$$

Esta equação é conhecida como equação da energia.

1.1.2 Equações Constitutivas

Para problemas de condução de calor, a mais importante quantidade física 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 $u(\mathbf{x}, t)$ de uma maneira dependente do material. As relações lineares, muito usadas em aplicações práticas, são dadas por

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

e a lei de Fourier para a condução do calor,

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

onde c é chamado calor específico e Q o tensor de condutividade térmica. Por argumentos da termodinâmica, assumiremos que $c > 0$ e a matriz Q é simétrica e definida positiva, *i.e.*, para qualquer vetor não nulo \mathbf{v} ,

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

Por outro lado, a energia suplementar r não é uma quantidade constitutiva. Ela pode representar a energia devido a radiação pelo meio ambiente e a fonte de energia dentro do corpo. Seja \hat{u} a temperatura do meio ambiente, então, podemos escrever

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

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

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

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

onde $f = \beta \hat{u} + \gamma$. 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 um parâmetro do material. Ele representa as propriedades características do material. A grosso modo, o material é chamado *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 que o tensor de condutividade térmica é invariante sob qualquer rotação ou em termos matemáticos,

$$RQR^T = Q \quad (1.7)$$

para qualquer matriz ortogonal R . Essa matriz é chamada matriz isotrópica. Pode ser facilmente mostrado que a 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 chamado de 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 considere por simplicidade o caso bidimensional. Seja R a rotação do ângulo θ dado por

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix},$$

então de (1.7), temos

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

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

$$Q_{11} = Q_{22} = \alpha, \quad Q_{12} = -Q_{21} = 0,$$

onde α é chamado de coeficiente de condutividade térmica. A segunda relação resulta da simetria da matriz Q_{ij} . Além disso, como a matriz é definida positiva, o coeficiente de condutividade α é uma quantidade positiva, $\alpha > 0$.

Um corpo é chamado *homogêneo* se as propriedades do material do corpo são independentes da posição \mathbf{x} do estado de referência.

Para um corpo homogêneo, ambos α e β são constantes do material e de (1.6), então, obtemos 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

De (1.6) o problema estacionário para a equação do calor é governada pela seguinte equação diferencial parcial do tipo elíptico,

$$-\sum_{i,j} \frac{\partial}{\partial x_i} \left(Q_{ij} \frac{\partial u}{\partial x_j} \right) + \beta u = f, \quad (1.8)$$

numa região fixa V ocupada pelo corpo, onde a condutividade térmica $Q_{ij}(\mathbf{x})$ é uma matriz definida positiva e $\beta(\mathbf{x}) > 0$, se o corpo é isotrópico, então $Q_{ij}(\mathbf{x}) = \alpha(\mathbf{x}) \delta_{ij}$ e a equação (1.8) torna-se

$$-\sum_i \frac{\partial}{\partial x_i} \left(\alpha \frac{\partial u}{\partial x_i} \right) + \beta u = f. \quad (1.9)$$

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

$$-\alpha \sum_i \frac{\partial^2 u}{\partial x_i \partial x_i} + \beta u = f, \quad (1.10)$$

onde α e β são constantes positivas.

Na fronteira ∂V usualmente a temperatura u ou o fluxo de calor q são prescritos. Mais especificamente existem dois tipos de condições de fronteira:

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

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

onde $u_0(\mathbf{x})$ e $u_1(\mathbf{x})$ são funções prescritas e o fluxo de calor q é dado por (1.2). Os valores da fronteira pode consistir de partes separadas de tipos diferentes. Um problema com condições de fronteira do primeiro tipo são usualmente chamados de problema de Dirichlet, enquanto problemas com condições de fronteira do segundo tipo são chamados de problema de Neumann.

Usando a lei de Fourier (1.5), a condição de fronteira do tipo Neumann pode ser explicitamente expressado na forma componente

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

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 $B \subset \mathbb{R}^3$ uma região ocupada por um corpo no seu estado de referência, e $\xi : B \times \mathbb{R} \rightarrow \mathbb{R}^3$ uma aplicação bijetora e regular,

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

chamada de movimento do corpo. O movimento é uma deformação do estado de referência, dependente do tempo. Introduzimos o gradiente da deformação F , a velocidade $\dot{\mathbf{x}}$ e a aceleração $\ddot{\mathbf{x}}$ definidas como

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

respectivamente. Seja (X_1, X_2, X_3) e (x_1, x_2, x_3) o sistema cartesiano do corpo no estado de referência e no estado atual do corpo respectivamente, então o gradiente da deformação F pode ser expressado na forma componente por

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

O gradiente da deformação F é uma transformação linear de B a $B_t = \xi(B, t)$. Pela hipótese, F é não singular, e em particular, podemos assumir que $\det F > 0$. O gradiente da deformação F é uma aproximação linear da deformação ξ , que geralmente é uma função não linear. Com efeito, se considerarmos um pequeno segmento linear $d\mathbf{X}$ no estado de referência, então no estado deformado temos

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

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

1.2.1 Pequena Deformação e Rotação Infinitesimal

Denotamos por

$$\mathbf{u}(\mathbf{X}, t) = \xi(\mathbf{X}, t) - \mathbf{X} \quad (1.13)$$

o vetor deslocamento do material no ponto $\mathbf{X} \in B$ no tempo t e denotamos o gradiente do deslocamento por H . Obviamente, temos

$$F = I + H,$$

onde I é o tensor identidade, e $H = 0$ quando não há deformação. Além disso, na teoria linear para pequenas deformações, é assumido que o gradiente do deslocamento é uma pequena quantidade, *i.e.*, $|H| \ll 1$ é pequena.

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

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

Para interpretar fisicamente estas definições, em primeiro lugar, observamos que

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

Seja $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ a base canônica do \mathbb{R}^3 e sejam $d\mathbf{X}_1$ e $d\mathbf{X}_2$ dois pequenos segmentos lineares no estado de referência, e $d\mathbf{x}_1$ e $d\mathbf{x}_2$ os segmentos correspondentes no estado deformado. Por (1.12), segue que

$$d\mathbf{x}_1 \cdot d\mathbf{x}_2 = F^T F d\mathbf{X}_1 \cdot d\mathbf{X}_2 = d\mathbf{X}_1 \cdot d\mathbf{X}_2 + 2E d\mathbf{X}_1 \cdot d\mathbf{X}_2, \quad (1.15)$$

negligenciando os termos de ordem superior de $|H|$.

Tomando $d\mathbf{X}_1 = d\mathbf{X}_2 = d\mathbf{X} = s_0 \mathbf{e}_1$ e $|d\mathbf{x}| = s$, e sabendo que $(E\mathbf{e}_i \cdot \mathbf{e}_j)$ 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 a percentagem do alongamento do segmento na direção e_1 com relação ao comprimento original, também conhecido como deformação Lagrangeano. Observamos que as demais componentes da diagonal de E tem a mesma interpretação.

Para interpretar o significado das componentes fora da diagonal, considere $d\mathbf{X}_1 = s_0 \mathbf{e}_1$ e $d\mathbf{X}_2 = s_0 \mathbf{e}_2$, e seja θ o ângulo entre $d\mathbf{x}_1$ e $d\mathbf{x}_2$. 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 $(d\mathbf{X}_1, d\mathbf{X}_2, d\mathbf{X}_3)$. Então de (1.12), o volume dv do elemento no estado deformado é dado por

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

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

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

Por definição,

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

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,

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

e o gradiente do deslocamento H ,

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

Como na teoria linear, os termos não lineares são insignificantes, portanto, não é necessário distinguir as coordenadas do estado de referência e atual. Logo, usualmente não é necessário introduzir o estado de referência na teoria linear. Assim, na forma componente, podemos escrever o gradiente do deslocamento como o gradiente com respeito ao estado atual e de (1.14), o tensor e a rotação infinitesimal podem ser escritos como

$$E_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad W_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right). \quad (1.16)$$

1.2.2 Equação do Movimento

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

$$\int_{\Omega} \rho \ddot{\mathbf{x}} d\Omega = \int_{\partial\Omega} \mathbf{t} d\Gamma + \int_{\Omega} \mathbf{f} d\Omega, \quad (1.17)$$

onde o lado direito representa as forças agindo sobre o corpo Ω . Existem dois tipos de forças, a primeira chamada de força do corpo \mathbf{f} que é devido a forças externas, tal como a gravitação. A segunda chamada de tração de superfície \mathbf{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 $\mathbf{x} \in \partial\Omega \subset V$, a tração \mathbf{t} é a força agindo sobre a parte Ω pela parte restante do corpo $V \setminus \Omega$ através da superfície $\partial\Omega$ no ponto \mathbf{x} . Para as superfícies de duas partes diferentes com um ponto em comum \mathbf{x} , os valores da tração \mathbf{t} no \mathbf{x} são geralmente diferentes nas respectivas superfícies. A idéia 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 \mathbf{x} então os valores da tração são iguais em \mathbf{x} . A consequência principal da hipótese de Cauchy, também conhecida como o *teorema de Cauchy*, assegura que a tração tem que ser linear na normal, *i.e.*,

$$\mathbf{t} = \sigma \mathbf{n} \quad \text{or} \quad t_i = \sum_{j} \sigma_{ij} n_j, \quad (1.18)$$

onde \mathbf{n} é a normal unitária externa de $\partial\Omega$ e σ é chamado de 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, para qualquer $\mathbf{x} \in V$,

$$\rho \ddot{\mathbf{x}} - \text{div } \sigma = \mathbf{f}, \quad (1.19)$$

ou na forma componente,

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

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

$$\sigma^T = \sigma \quad \text{ou} \quad \sigma_{ij} = \sigma_{ji}. \quad (1.20)$$

De fato, a conservação do momento angular é dada por

$$\int_{\Omega} \mathbf{r} \times \rho \ddot{\mathbf{x}} d\Omega = \int_{\partial\Omega} \mathbf{r} \times \sigma \mathbf{n} d\Gamma + \int_{\Omega} \mathbf{r} \times \mathbf{f} d\Omega, \quad (1.21)$$

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

$$\mathbf{a} \cdot \mathbf{r} \times \rho \ddot{\mathbf{x}} - \operatorname{div}(\sigma^T(\mathbf{a} \times \mathbf{r})) = \mathbf{a} \cdot \mathbf{r} \times \mathbf{f}, \quad (1.22)$$

onde usamos a identidade,

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

Por outro lado de (1.19), segue que

$$\mathbf{a} \cdot \mathbf{r} \times \rho \ddot{\mathbf{x}} - \mathbf{a} \cdot \mathbf{r} \times \operatorname{div} \sigma = \mathbf{a} \cdot \mathbf{r} \times \mathbf{f}. \quad (1.23)$$

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

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

que após alguns cálculos nos leva a simetria do tensor de tensão, $\sigma^T = \sigma$. Os cálculos estão feitos na Sec. 1.3.

Para ver o significado físico do tensor de tensão, considere a superfície $\partial\Omega$ com normal $\mathbf{n} = \mathbf{e}_1$. Então, a componente da tração \mathbf{t} normal à superfície é dada por

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

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

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

Assim, σ_{11} e σ_{21} são as forças normal e tangencial por área unitária 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 de materiais elásticos para pequenas deformações pode ser expressada como uma relação linear entre a tensão e a deformação:

$$\sigma(\mathbf{x}, t) = C(\mathbf{x})E(\mathbf{x}, t), \quad \text{or} \quad \sigma_{ij} = \sum_{k,l} C_{ijkl} E_{kl}, \quad (1.25)$$

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

$$C_{ijkl} = C_{jikl} = C_{ijlk}. \quad (1.26)$$

Além disso, por argumentos da termodinâmicas, 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,

$$C_{ijkl} = C_{klij}. \quad (1.27)$$

A relação linear (1.25) é chamada de lei de Hooke para materiais elásticos. O tensor de elasticidade envolve muitas constantes de material. Com efeito, considerando a simetria (1.26) e (1.27), em geral para materiais não isotrópicos, 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 elasticidade deve ser invariante sob qualquer orientação e como uma consequência, podemos provar que o número de constantes do material se reduz a somente duas, a saber,

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

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

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

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

Usualmente é assumido, por considerações da termodinâmica, que o tensor de elasticidade é definido positivo, *i.e.*,

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

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

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

1.2.4 Problemas Elastostáticos

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

$$-\sum_{j,k,l} \frac{\partial}{\partial x_j} \left(C_{ijkl} \frac{\partial u_k}{\partial x_l} \right) = f_i \quad i = 1, 2, 3, \quad (1.31)$$

em V . 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 \mathbf{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

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

Se o corpo é isotrópico, então, de (1.29) a equação de equilíbrio é dada por

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

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

$$-(\lambda + \mu) \operatorname{grad}(\operatorname{div} \mathbf{u}) - \mu \Delta \mathbf{u} = \mathbf{f}.$$

Na fronteira ∂V , usualmente ou o vetor deslocamento u_i ou a tração na superfície t_i são prescritas. Mais especificamente por (1.18), existem dois tipos de condições de fronteira:

$$(1) \quad u_i(\mathbf{x}) = q_i(\mathbf{x}),$$

$$(2) \quad \sum_j \sigma_{ij} n_j = p_i(\mathbf{x}),$$

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

As condições de fronteira são de dois tipos diferentes. A primeira condição é chamada de tipo Dirichlet, enquanto a segunda é chamada do tipo Neumann. A condição de fronteira do tipo Neumann pode ser escrita na forma

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

Finalmente, observamos que desde que as equações são lineares, se \mathbf{u} é uma solução de (1.31), temos que $k\mathbf{u}$ é solução para qualquer $k \in \mathbb{R}$, quando a força externa é $k\mathbf{f}$. Apesar da teoria de elasticidade linear ser válida somente para pequenas deformações, para grandes deformações é possível obter soluções matemáticas, mas tais soluções não têm sentido físico. Logo, é importante formular o problema de valor de fronteira com condições de fronteira e forças externas apropriadas para que a solução \mathbf{u} satisfaça a hipótese de pequenas deformações, $|\nabla \mathbf{u}| \ll 1$.

Para dar um exemplo, consideremos o problema de Dirichlet unidimensional na região $V = (a, b) \subset \mathbb{R}$, e seja $L = b - a$. Então, o valor da função prescrita $|\mathbf{q}(\mathbf{x})|$ deve ser pequena comparado a L . Ao contrário, a deformação não será pequena, a menos que a solução seja uma grande translação.

1.3 Convenção de Somatório

Frequentemente é usada a convenção de somatório, para simplificar as expressões envolvendo somatório com respeito a repetição de índices:

Convenção de somatório. Na expressão de uma soma, quando aparece um par de índices, i.e., o índice é repetido exatamente uma vez, então isto significa um somatório sobre o índice no seu domínio sem explicitar o símbolo do somatório.

O domínio do índice é 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 de índice que realmente aparece somente um única vez ou mais de duas vezes, o símbolo 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):

$$\begin{cases} -\frac{\partial}{\partial x_i} \left(Q_{ij} \frac{\partial u}{\partial x_j} \right) + \beta u = f & \text{in } V, \\ u = u_0 \quad \text{or} \quad Q_{ij} \frac{\partial u}{\partial x_j} n_i = u_1 & \text{on } \partial V. \end{cases}$$

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

$$\begin{cases} -\frac{\partial}{\partial x_j} \left(C_{ijkl} \frac{\partial u_k}{\partial x_l} \right) = f_i & \text{in } V, \\ u_i = q_i \quad \text{or} \quad C_{ijkl} \frac{\partial u_k}{\partial x_l} n_j = p_i & \text{on } \partial V. \end{cases}$$

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 $\mathbf{v} \times \mathbf{u}$ pode ser escrito na forma componente como

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

Agora podemos escrever a relação (1.24) como

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

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

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

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

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

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 à equação diferencial com as condições de contorno dadas 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 a derivada de $u(x)$ por $u'(x)$. 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 idéia básica do método das diferenças finitas é transformar o problema de resolver uma equação diferencial num problema de resolver um sistema de equações algébricas, usando aproximações das derivadas da equação por diferenças finitas. O método das projeções consiste em obter uma solução aproximada da solução da equação diferencial, usando uma combinação linear finita de funções conhecidas, usualmente chamadas funções bases. Conceitualmente, se considerarmos que a solução do problema pertence a algum espaço de função 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.

Nós ilustraremos as idéias gerais do método das projeções com um problema de segunda ordem simples, dado por

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

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

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

onde as funções bases φ_j satisfazem as condições de fronteira:

$$\varphi_j(0) = \varphi_j(1) = 0, \quad j = 1, \dots, m. \quad (1.38)$$

Nestas condições, a solução aproximada $u_m(x)$, dada por (1.37), satisfaz a condição de fronteira.

Dadas as funções bases, o problema é como determinar os coeficientes C_j da solução aproximada 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, \dots, 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 nestes m pontos. Portanto substituindo a $u_m(x)$ definida em (1.37) na equação diferencial obtemos

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

onde estamos assumindo que as funções bases são duas vezes diferenciáveis. Podemos isolar os coeficientes C_j para obter o seguinte sistema linear:

$$\sum_{j=1}^m C_j (-\varphi_j''(x_i) + \varphi_j(x_i)) = f(x_i), \quad i = 1, \dots, m. \quad (1.39)$$

Definindo

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

então obtemos o sistema linear $AC = F$, onde $A = (a_{ij})$ é uma matriz quadrada de ordem m , $C = (C_1, \dots, C_m)^T$ é o vetor incógnita do sistema e $F = (f(x_1), \dots, f(x_m))^T$ é o vetor força nodal conhecido. Determinando os coeficientes C_j do sistema linear, então a solução aproximada da equação diferencial (1.36) é obtida através de (1.37). A matriz A depende somente das funções bases consideradas e, geralmente, é uma matriz de componentes cheias 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

$$(u, v) = \int_0^1 u(x)v(x)dx = 0. \quad (1.41)$$

Consideramos o problema modelo (1.36) e introduzimos a função resíduo $r(x)$ definida por

$$r(x) = -u''(x) + u(x) - f(x) \quad \forall x \in (0, 1). \quad (1.42)$$

É fácil ver que se $u(x)$ é a solução exata do problema (1.36), então $r(x) = 0$, e neste caso, a função resíduo é ortogonal a qualquer função, e, em particular, é ortogonal as funções bases. No entanto, se a solução $u(x) = u_m(x)$ é dada por (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 $\{\varphi_1, \varphi_2, \dots, \varphi_m\}$, ou seja,

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

Substituindo (1.37) na integral acima, obtemos

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

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

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

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

De (1.44) é suficiente assumir que as funções $\varphi_j(x)$ e suas derivadas $\varphi'_j(x)$ sejam quadrado-integrável, *i.e.*, de $L^2(0, 1)$, como por exemplo, funções contínuas e seccionalmente lineares. Logo, as classes de funções disponíveis para funções bases são mais amplas para método de Galerkin do que para método da colocação.

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 uma 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 uso das autofunções do operador do problema (1.36), $D = -d^2/dx^2 + I$,

$$\varphi_j(x) = \text{sen } j\pi x, \quad j = 1, \dots, m. \quad (1.45)$$

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

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

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

$$\begin{aligned} a_{jj} &= \frac{1}{2} \left(1 + (j\pi)^2 \right) \\ f_j &= \int_0^1 f(x) \text{sen } j\pi x \, dx, \end{aligned} \quad j = 1, \dots, m. \quad (1.46)$$

Portanto, os valores de C_j são dados por

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

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

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

Observamos que esta solução corresponde a série truncada da solução clássica obtida pelo *método de séries de Fourier*, ou seja

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

onde \hat{f}_j são os coeficientes de Fourier da função $f(x)$,

$$\hat{f}_j = 2 \int_0^1 f(x) \sin j\pi x \, dx.$$

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

Evidentemente, as autofunções definem uma base ideal para o cálculo da solução, mas infelizmente nem sempre isto é possível. Porém, como veremos, tomando uma base "local" dos elementos finitos, a matriz dos coeficientes, convenientemente ordenada, é uma matriz banda e isto reduz muito 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 $\{\varphi_1(x), \dots, \varphi_m(x)\}$ que geram o subespaço onde estamos procurando a solução aproximada $u_m(x)$.

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

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

Neste caso, o suporte da função $\varphi_j(x)$ é somente 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 e reduz o número de operações para resolução do sistema linear para uma ordem proporcional a m .

Note que a função base $\varphi_j(x)$, contínua e seccionalmente linear, definida em (1.47) é de $L^2(0, 1)$. Ela não pode ser usada como função base para o problema (1.36) utilizando o método de colocação, pois, neste caso, a função base tem que ser duas vezes

diferenciável. Entretanto, funções mais regulares constituídas pelos polinômios nos elementos (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. Neste livro, estaremos exclusivamente considerando elementos finitos, utilizando o Método de Galerkin.

1.5 Problemas Variacionais Abstratos

Para um melhor compreensão dos conceitos usados nos próximos capítulos, introduziremos algumas definições e teoremas fundamentais para obter a existência e unicidade de solução para uma ampla classe de problemas variacionais. Para isto consideremos os espaços de Hilbert V e H com produto interno e norma representados, respectivamente, por $((\cdot, \cdot)); \|\cdot\|$ e $(\cdot, \cdot); |\cdot|$.

Definição 1. A forma $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$, ou seja, a função numérica a definida em $V \times V$ é bilinear em V , se é linear em cada uma das componentes.

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

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

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

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

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

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

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

$$a(u, v) = \langle f, v \rangle, \quad \forall v \in V. \quad (1.48)$$

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

Teorema (Lax-Milgram). *Se $a(\cdot, \cdot)$ for uma forma bilinear, contínua e coerciva e f uma forma linear e contínua em V , então 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 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, denominado funcional energia, $E : V \rightarrow \mathbb{R}$ dado por

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

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 $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$ uma forma bilinear, contínua, coerciva, simétrica e $f : V \rightarrow \mathbb{R}$ 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 que:

$$\begin{aligned} a(u, u - v) &\leq \langle f, u - v \rangle + \frac{1}{2} a(u - v, u - v) \iff \\ a(u, u) - a(u, v) &\leq \langle f, u \rangle - \langle f, v \rangle + \frac{1}{2} a(u, u) - a(u, v) - \frac{1}{2} a(v, v), \quad \forall v \in V. \end{aligned}$$

Logo $E(u) \leq E(v)$, $\forall v \in V$. e portanto a solução u é a função que minimiza o funcional Energia. Falta mostrar que esta solução é única. De fato, considere uma outra solução $\tilde{u} \in V$ que minimiza o funcional Energia. Assim por definição de mínimo temos:

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

Usando a definição e desenvolvendo cada um dos termos obtemos que:

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

Logo $\frac{1}{4} a(u - \tilde{u}, u - \tilde{u}) \leq 0$. Desde que $a(., .)$ é coerciva, conclui-se que $u = \tilde{u}$.

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

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

Fazendo o limite:

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

Daí conclui-se que $a(u, v) - \langle f, v \rangle \geq 0$, $\forall v \in V$. 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 a solução do problema (1.48). \square

1.5.2 Espaço das Funções Testes

No que segue, daremos noções básicas da teoria das distribuições para facilitar a compreensão do texto.

Seja Ω um conjunto aberto e limitado do \mathbb{R}^n com fronteira suave. Definimos por $C^\infty(\Omega)$ o espaço vetorial das funções reais 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 compacto $K \subset \Omega$.
2. A sequência (φ_ν) e de todas as suas derivadas convergem uniformemente para zero em K , ou seja,

$$(D^j \varphi_\nu) \rightarrow 0,$$

para todo multi-índice $j = (j_1, j_2, \dots, j_N)$.

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 a noção acima podemos então definir o espaço das distribuições.

Definição 7. Denomina-se distribuição sobre Ω a toda forma linear T , contínua em $\mathcal{D}(\Omega)$, ou seja, uma distribuição é um funcional $T : \mathcal{D}(\Omega) \rightarrow \mathbb{R}$ satisfazendo as condições:

1. $T(\alpha\theta + \beta\varphi) = \alpha T(\theta) + \beta T(\varphi) \quad \alpha, \beta \in \mathbb{R}; \quad \theta, \varphi \in \mathcal{D}(\Omega)$
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 $\mathcal{D}'(\Omega)$.

Pode-se mostrar que as funções localmente integráveis, ou seja, as funções $u \in L^1_{loc}(\Omega)$ definem univocamente uma distribuição dada por

$$\langle T_u, \varphi \rangle = \int_{\Omega} u(x)\varphi(x)dx.$$

Desde que podemos identificar de forma única a distribuição T_u com a função $u \in L^1_{loc}(\Omega)$, quando não houver ambiguidade escreve-se u em lugar de T_u e assim temos a dualidade definida por

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

Para $u \in \mathcal{D}'(\Omega)$, $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$ e $|\alpha| = \alpha_1 + \dots + \alpha_n$, definimos a derivada de ordem α de u no sentido das distribuições do seguinte modo:

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

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

Com a noção de distribuição e suas derivadas, podemos definir, os espaços de Sobolev $H^m(\Omega)$.

Seja Ω um aberto do \mathbb{R}^n , definimos o espaço $L^2(\Omega)$ por

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

cujo produto interno e norma são definidos por

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

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 $u \in L^2(\Omega)$ e, deste modo, as funções deste espaço são considerados distribuições sobre Ω e portanto as funções $u \in L^2(\Omega)$ possuem derivadas de todas as ordens no sentido das distribuições. É importante observar que as derivadas das funções $u \in L^2(\Omega)$ não necessariamente são funções do $L^2(\Omega)$.

Definimos o espaço $H^m(\Omega)$, $m = 0, 1, \dots$ com $m \in \mathbb{N}$, por

$$H^m(\Omega) = \left\{ u : \Omega \rightarrow \mathbb{R}; u \in L^2(\Omega), \frac{\partial^\alpha u}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}} \in L^2(\Omega) \right\}$$

onde $\alpha = \alpha_1 + \dots + \alpha_n$, $1 \leq \alpha \leq m$, $\alpha_i \in \mathbb{N}$.

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

$$(u, v) = \sum_{\alpha=0}^m \int_{\Omega} \frac{\partial^\alpha u}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}} \frac{\partial^\alpha v}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}} \quad \forall u, v \in H^m(\Omega). \quad (1.52)$$

Observe que o espaço de Sobolev $H^m(\Omega)$ com o produto interno definido em (1.52) é um espaço de Hilbert, para $m = 0, 1 \dots m \in \mathbb{N}$. Em particular, quando $m = 0$, temos espaço $H^0(\Omega) = L^2(\Omega)$ e para $m = 1$, temos

$$H^1(\Omega) = \left\{ u \in L^2(\Omega); \frac{\partial u}{\partial x_i} \in L^2(\Omega) \right\}.$$

cujo produto interno e norma são dados por

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

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,

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

Definição 8. Sejam X e Y espaço de Banach reais, sendo X um espaço reflexivo e com injeção contínua de X em Y . Diz-se que a função $u(t) \in Y$ é fracamente contínua se $\langle u(t), y' \rangle$ é contínua, $\forall y' \in Y'$. O espaço das funções fracamente contínuas em Y é denotado por $C_s([0, T]; Y)$.

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

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

Teorema de Aubin-Lions. Sejam os espaços de Banach B_0 , B e B_1 com imersão compacta entre B_0 e B e imersão contínua entre B e B_1 . Dada uma sucessão satisfazendo as condições:

$$u_m \in L^P(0, T; B_0) \quad \text{e} \quad \frac{du_m}{dt} \in L^P(0, T; B_1).$$

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

Um exemplo comum de espaços satisfazendo as imersões acima são: $B_0 = H_0^1$, $B = L^2$ e $B_1 = H^{-1}$ ou $B_1 = L^2$.

Desigualdade de Gronwall. Seja φ uma função real e integrável em $[0, b)$, tal que $\varphi(t) \geq 0$, $\forall t \in [0, b)$. Se

$$\varphi(t) \leq C + \int_0^t \varphi(s) \, ds$$

então $\varphi(t)$ é limitada.

Algumas propriedades importantes do espaço $H_0^1(\Omega)$, 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 espaço dual do espaço $H_0^1(\Omega)$, ou seja, $H^{-1}(\Omega)$ é o espaço dos operadores lineares e contínuos definidos em $H_0^1(\Omega)$. Deste 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, \dots, 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 $\{u_\nu\} \in \mathcal{D}(\Omega)$ convergente para u em $H^1(\Omega)$.

Obs: Identificando o espaço $L^2(\Omega)$ com seu dual $(L^2(\Omega))'$ e das duas propriedades anteriores, podemos concluir 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 $L^2(\Omega)$. 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 , $C^k(\bar{\Omega})$ são densos em $H^m(\Omega)$ $m = 0, 1, \dots$; 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 $C^k(\bar{\Omega})$. Neste caso, as derivadas são no sentido clássico.
5. **Teorema do Traço.** A aplicação linear e contínua, $\gamma_0 : H^1(\Omega) \rightarrow L^2(\Gamma)$, 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

$$|\gamma_0 u|_0 \leq \|u\|_1.$$

1.6 Aproximação por Diferenças Finitas

Usualmente para resolver uma equação diferencial de evolução, do tipo parabólica ou hiperbólica pelo método de elementos finitos, a derivada no tempo é aproximada pelo método das diferenças finitas por razões de estabilidade. Dessa forma, nessa seção faremos algumas considerações sobre a aproximação da derivada por diferenças finitas e a ordem de convergência.

Seja $f \in C^{n+1}(a, b)$. Do teorema de Taylor podemos expandir a função $f(x)$ na vizinhança de $x \in (x - h, x + h)$, $h \ll 1$, na forma:

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

e de forma análoga,

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

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

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

onde $\mathcal{O}(h^4)$ denota todos os termos de potência quatro ou superior de h . Assumindo que estes termos são pequenos quando comparados com potências inferiores de h ($h \ll 1$) então negligenciando os termos do lado direito, temos a seguinte aproximação para a segunda derivada da função,

$$f''(x) \simeq \frac{1}{h^2} \left(f(x+h) - 2f(x) + f(x-h) \right), \quad (1.56)$$

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

$$f'(x) \simeq \frac{1}{2h} \left(f(x+h) - f(x-h) \right), \quad (1.57)$$

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:

$$f'(x) \simeq \frac{1}{h} \left(f(x+h) - f(x) \right) \quad (\text{Diferença progressiva ou adiantada}), \quad (1.58)$$

$$f'(x) \simeq \frac{1}{h} \left(f(x) - f(x-h) \right) \quad (\text{Diferença regressiva ou atrasada}), \quad (1.59)$$

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

Notação:

Suponhamos que f é uma função das variáveis independentes $x \in [a, b]$ e $t \in [0, T]$ e seja seguinte discretização uniforme: $a = x_0 < x_1 < \dots < x_M = b$ e $0 = t_0 < t_1 < \dots < t_N = T$, 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,

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

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

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

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

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

com erro de ordem $\mathcal{O}(\Delta t^2)$.

Para a diferença progressiva (regressiva) temos

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

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 estudamos abaixo 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 erros associados. As etapas do programa computacional que se encontram no apêndice são mostradas passo a passo.

2.1 Formulação do Problema

O problema modelo que estudaremos será o de determinar no espaço de soluções H , uma função $u = u(x)$, tal que,

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

onde α e β são constantes reais positivas e $f = f(x)$ uma função regular.

A solução $u = u(x)$ do problema (2.1) acima representa a posição de equilíbrio de uma barra que está presa nos extremos e $f = f(x)$ é a força distribuída atuando sobre a barra. Outros tipos de fronteiras também serão considerados neste texto.

Formulação Variacional

O Método de elementos finitos não é aplicável diretamente no problema (2.1). Assim, é necessário expressar o problema numa forma mais conveniente para que seja possível aplicar o método de Galerkin, que introduziremos adiante.

Seja $\mathcal{D}(\Omega) = \{v \in C_0^\infty(\Omega); v(\partial\Omega) = 0\}$ denominado espaço das funções testes com suporte compacto em Ω . Multiplicando a equação (2.1) por $v \in \mathcal{D}(\Omega)$ e integrando obtém-se

$$\int_0^1 -\alpha u_{xx} v \, dx + \int_0^1 \beta uv \, dx = \int_0^1 f v \, dx \quad \forall v \in \mathcal{D}(\Omega). \quad (2.2)$$

Integrando por partes a primeira integral obtemos

$$-\alpha \int_0^1 u_{xx} v \, dx = -\alpha(u_x v) \Big|_0^1 + \alpha \int_0^1 u_x v_x \, dx = \alpha \int_0^1 u_x v_x \, dx$$

pois $v(0) = v(1) = 0$. Substituindo em (2.2) temos

$$\alpha \int_0^1 u_x v_x \, dx + \beta \int_0^1 uv \, dx = \int_0^1 f v \, dx \quad \forall v \in \mathcal{D}(\Omega). \quad (2.3)$$

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

$$a(u, v) = \alpha \int_0^1 u_x v_x \, dx + \beta \int_0^1 uv \, dx, \quad (2.4)$$

e

$$(f, v) = \int_0^1 f v \, dx, \quad (2.5)$$

então (2.3) é equivalente a determinar $u = u(x)$ satisfazendo:

$$a(u, v) = (f, v), \quad \forall v \in V. \quad (2.6)$$

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

Com efeito

1. $a(., .)$ é bilinear e simétrica. Sejam u ; v e w funções pertencentes ao espaço de Hilbert V . Usando a definição (2.4) é simples mostrar que:

$$a(u + w, v) = a(u, v) + a(w, v) \quad \text{e} \quad a(u, v + w) = a(u, v) + a(u, w)$$

e a simetria $a(u, v) = a(v, u)$.

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

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

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

3. $a(.,.)$ é coerciva. Temos que

$$a(u, u) \geq \delta_2 \left\{ \int_0^1 u_x^2 dx + \int_0^1 u^2 dx \right\} \geq \delta_2 \|u\|_1^2, \quad \forall u \in V$$

onde $\delta_2 = \min\{\alpha; \beta\} > 0$. Note que a condição das constantes α e β serem positivas é essencial para mostrar que a forma bilinear definida seja coerciva em $H_0^1(0, 1)$.

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

$$\|\langle f, v \rangle\| \leq \int_0^1 \|f v\| dx \leq c_3 \|v\|_1 \quad \forall v \in V,$$

onde $c_3 = \|f\|_0$.

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

$$\|u\|_2 \leq c \|f\|_0, \quad (2.7)$$

onde c somente depende de $\Omega = (0, 1)$.

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

Método de Galerkin

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

$$V_m = [\varphi_1, \varphi_2, \dots, \varphi_m] \quad (2.8)$$

onde $\{\varphi_i, i \in N\}$ é uma base de $H_0^1(\Omega)$. Agora, buscamos uma solução aproximada $u^h = u^h(x)$ do problema (2.6) no subespaço V_m .

Problema Aproximado

Aproximamos o problema (2.6) por

$$a(u^h, v) = (f, v), \quad \forall v \in V_m \quad (2.9)$$

onde

$$u^h(x) = \sum_{j=1}^m C_j \varphi_j(x), \quad \varphi_j \in V_m \quad (2.10)$$

é uma solução aproximada de $u = u(x)$.

Para se obter a solução aproximada $u^h \in V_m$ é necessário determinar os coeficientes C_j . Substituindo u^h dado por (2.10) em (2.9) tem-se

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

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

$$\sum_{j=1}^m a(\varphi_i, \varphi_j) C_j = (\varphi_i, f), \quad \text{para } i = 1, \dots, m \quad (2.11)$$

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 $C = (C_1, \dots, C_m)$ o vetor incógnita, obtemos o sistema linear

$$KC = F \quad (2.12)$$

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 por;

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

$$F_i = \int_0^1 f \varphi_i dx \quad (2.14)$$

Propriedades da Matriz Rigidez

(i) K é simétrica.

De fato:

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

(ii) K é definida positiva.

De fato: Seja $d = (d_1, \dots, d_m)$. Então,

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

pois

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

para $\alpha, \beta > 0$, onde $\delta_2 = \min\{\alpha, \beta\}$. Por outro lado $d^T K d = 0 \Leftrightarrow a(v, v) = 0$. Assim $\|v\|_{H_0^1(\Omega)}^2 = 0$. Como a base é linearmente independente, temos $d = 0$.

2.2 Função de Interpolação

Para o cálculo da matriz global K_{ij} e do vetor força F_i precisamos definir explicitamente as funções φ_i , base do subespaço V_m . A escolha de φ_i é essencial para a otimização do sistema linear. Desde que os elementos da matriz K_{ij} são funções deriváveis, escolhemos as φ_i 's como sendo funções trigonométricas ou polinomiais, para as quais as condições de fronteira são satisfeitas. Se na escolha da base φ_i resultar uma matriz K_{ij} cheia, então à medida que queremos melhorar a solução aproximada $u^h(x)$ devemos aumentar a dimensão do subespaço V_m de $H_0^1(\Omega)$. Assim, o número m de vetores φ_i seria grande e portanto a matriz K_{ij} de ordem $m \times m$ aumentaria quadraticamente, tornando o sistema de difícil resolução.

O objetivo principal na escolha de φ_i é fazer com que a matriz K_{ij} seja uma matriz especial com muitos elementos nulos obedecendo uma certa ordem. Este tipo de matriz é denominada matriz esparsa e o sistema linear resultante, em geral, é bem condicionado.

As funções φ_i escolhidas são funções de interpolação linear por partes satisfazendo a seguinte condição:

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

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

$$h_i = x_{i+1} - x_i, \quad i = 1, \dots, m \quad (2.18)$$

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

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

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

Geometricamente, as funções φ_i podem ser representadas por

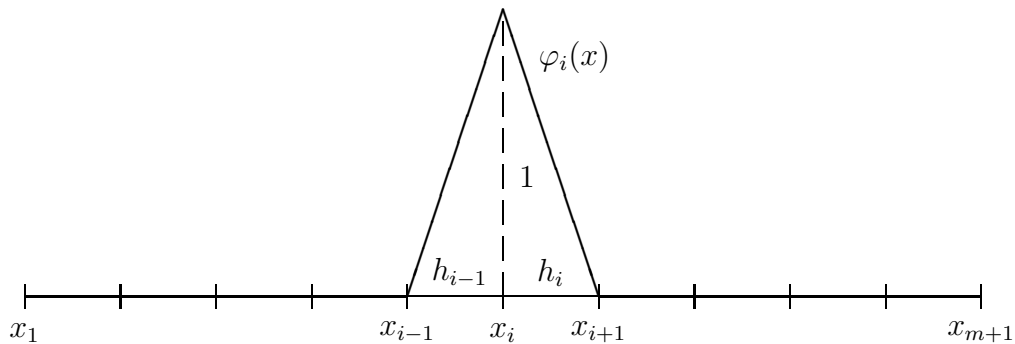


Figura 2.1: Função base

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

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

A derivada da função $\varphi_i(x)$ é descontínua no ponto x_i , mas isto não afeta o cálculo da matriz K_{ij} , pois um ponto tem medida zero.

Note que o espaço V_m gerado pelas funções contínuas e linear por partes φ_i é um subespaço do espaço $V = H^1(0, 1)$, pois $\varphi_i \in C^0[0, 1] \Leftrightarrow \varphi_i \in H^1(0, 1)$.

Matriz Rigidez

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

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

e $\varphi_i \varphi_j = 0$ 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;

$$K = \begin{bmatrix} * & * & 0 & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 & 0 \\ 0 & * & * & * & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & * & * & * & 0 \\ 0 & 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & * & * \end{bmatrix} \quad (2.21)$$

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

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

Calculando cada uma das integrais obtemos

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

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

$$\begin{aligned} K_{i,i+1} &= \alpha \int_0^1 \frac{d\varphi_i}{dx} \frac{d\varphi_{i+1}}{dx} dx + \beta \int_0^1 \varphi_i \varphi_{i+1} dx = \alpha \int_{x_i}^{x_{i+1}} \frac{d\varphi_i}{dx} \frac{d\varphi_{i+1}}{dx} dx + \beta \int_{x_i}^{x_{i+1}} \varphi_i \varphi_{i+1} dx = \\ &= \alpha \int_{x_i}^{x_{i+1}} \left(-\frac{1}{h_i} \frac{1}{h_i} \right) dx + \beta \int_{x_i}^{x_{i+1}} \frac{1}{h_i} (x_{i+1} - x) \left(\frac{x - x_i}{h_i} \right) dx = -\frac{\alpha}{h_i} + \frac{\beta h_i}{6} \end{aligned}$$

Logo,

$$K_{i,i+1} = K_{i+1,i} = -\frac{\alpha}{h_i} + \frac{\beta h_i}{6} \quad (2.23)$$

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;

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

onde

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

e

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

Se a malha é uniforme, $h = h_i$, assim

$$a = \frac{2\alpha}{h} + \frac{2\beta h}{3} \quad (2.27)$$

e

$$b = -\frac{\alpha}{h} + \frac{\beta h}{6} \quad (2.28)$$

Vetor Força

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

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

para $i = 2, 3, \dots, m$. Por exemplo, tomando $f(x) = x$ para todo $x \in [0, 1]$, então;

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

Para a malha uniforme temos

$$F_i = hx_i, \quad i = 2, 3, \dots, m \quad (2.30)$$

Uma outra forma para o cálculo da força F_i é através da interpolação de $f(x)$ usando as funções base φ_i como interpoladores. Este procedimento é adequado quando para o cálculo da integral F_i é necessário o uso de métodos numéricos. Com efeito, a função $f(x)$ pode ser interpolada pela função φ_i da seguinte forma:

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

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

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

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

$$F_i = \sum_{j=i-1}^{i+1} \int_{x_{i-1}}^{x_{i+1}} (\varphi_i \varphi_j) f_j dx = f_{i-1} \int_{x_{i-1}}^{x_i} (\varphi_i \varphi_{i-1}) dx + f_i \int_{x_{i-1}}^{x_{i+1}} (\varphi_i)^2 dx + f_{i+1} \int_{x_i}^{x_{i+1}} (\varphi_i \varphi_{i+1}) dx \quad (2.32)$$

Fazendo os cálculos das integrais, obtemos

$$\begin{aligned}\int_{x_{i-1}}^{x_i} (\varphi_i \varphi_{i-1}) dx &= \frac{h_{i-1}}{6} \\ \int_{x_{i-1}}^{x_{i+1}} (\varphi_i)^2 dx &= \frac{1}{3}(h_{i-1} + h_i) \\ \int_{x_i}^{x_{i+1}} (\varphi_i \varphi_{i+1}) dx &= \frac{h_i}{6}\end{aligned}$$

Substituindo em (2.32) obtemos:

$$F_i = f_{i-1} \frac{h_{i-1}}{6} + f_i \frac{h_i + h_{i-1}}{3} + f_{i+1} \frac{h_i}{6} \quad (2.33)$$

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

$$F_i = \frac{h}{6} \{f_{i-1} + 4f_i + f_{i+1}\} \quad (2.34)$$

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 $C = (C_2, \dots, C_m)^t$ 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 $i = 2, \dots, m$. Com efeito, fazendo $x = x_i$ e substituindo em (2.10) temos:

$$u^h(x_i) = \sum_{j=2}^m C_j \varphi_j(x_i) = C_i \quad (2.35)$$

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

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

Definimos:

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

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

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

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

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

e o vetor força

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

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

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

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

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

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 dimensão 2 ou 3.

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

$$x_{i+1} = x_i + h_i, \quad i = 1, 2, \dots, m.$$

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

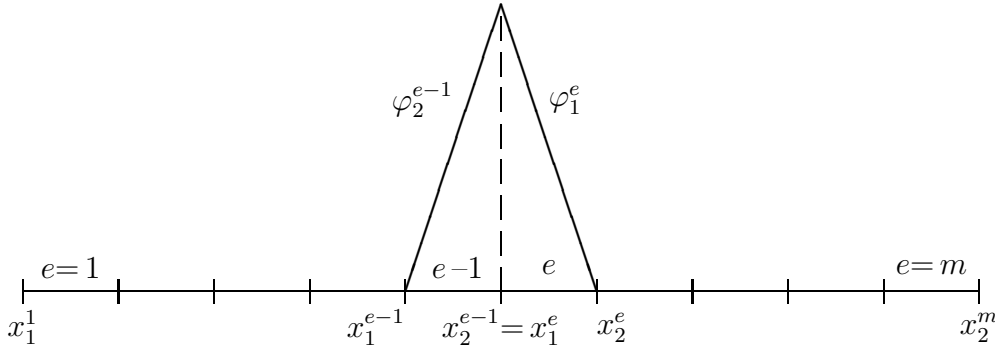


Figura 2.2: Função base local

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

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

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

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

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

$$\begin{aligned} K_{ij} &= \int_0^1 \left(\alpha \frac{d\varphi_i}{dx} \frac{d\varphi_j}{dx} + \beta \varphi_i \varphi_j \right) dx \\ F_i &= \int_0^1 f \varphi_i dx \end{aligned}$$

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

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

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

e

$$F_a^e = \int_{x_1^e}^{x_2^e} f \varphi_a^e dx \quad 1 \leq a \leq 2 \quad (2.40)$$

que denominamos de matriz local e força local, respectivamente.

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

$$K = \sum_{e=1}^m K^e \quad \text{e} \quad F = \sum_{e=1}^m F^e \quad (2.41)$$

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. Temos, então, que:

$$K^e = \begin{array}{c} \begin{array}{cc} & e \quad e+1 \\ \begin{bmatrix} 0 & \downarrow & \downarrow & 0 \\ & K_{11}^e & K_{12}^e & \leftarrow \\ & K_{21}^e & K_{22}^e & \leftarrow \\ 0 & & & 0 \end{bmatrix} & \begin{array}{l} e \\ e+1 \end{array} \end{array} \end{array}$$

Os elementos $K_{11}^e, K_{12}^e, K_{21}^e$ e K_{22}^e podem ser representados por $K_{ee}^e, K_{e,e+1}^e, K_{e+1,e}^e$ e $K_{e+1,e+1}^e$, respectivamente. Análogamente, para a força F^e os únicos elementos não necessariamente nulos são F_1^e e F_2^e . Assim

$$F^e = [0, \dots, 0, \underbrace{F_1^e, F_2^e}, 0, \dots, 0]^t$$

onde $F_1^e = F_e^e$ e $F_2^e = F_{e+1}^e$.

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 formada pelos coeficientes e coordenadas não nulos, ou seja,

$$K^e = \begin{bmatrix} K_{11}^e & K_{12}^e \\ K_{21}^e & K_{22}^e \end{bmatrix} \quad \text{e} \quad F^e = \begin{bmatrix} F_1^e \\ F_2^e \end{bmatrix} \quad (2.42)$$

A matriz local K^e é uma matriz padrão dos elementos e somente serão diferentes se $h_e = x_2^e - x_1^e$ 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 $\varphi_a^e(x)$ definida por (2.38) com $[x_1^e, x_2^e] = [x_e, x_{e+1}]$. Para calcular o coeficiente K_{ee} da matriz global K basta fazer $i = j = e$. Então:

$$K_{ee} = \int_0^1 \left(\alpha \frac{d\varphi_e}{dx} \frac{d\varphi_e}{dx} + \beta \varphi_e \varphi_e \right) dx$$

$$= \int_{x_{e-1}}^{x_e} \left(\alpha \frac{d\varphi_e}{dx} \frac{d\varphi_e}{dx} + \beta \varphi_e \varphi_e \right) dx + \int_{x_e}^{x_{e+1}} \left(\alpha \frac{d\varphi_e}{dx} \frac{d\varphi_e}{dx} + \beta \varphi_e \varphi_e \right) dx$$

Em termos da contribuição local, obtemos

$$\begin{aligned} K_{ee} &= \int_{x_1^{e-1}}^{x_2^{e-1}} \left(\alpha \frac{d\varphi_2^{e-1}}{dx} \frac{d\varphi_2^{e-1}}{dx} + \beta \varphi_2^{e-1} \varphi_2^{e-1} \right) dx \\ &\quad + \int_{x_1^e}^{x_2^e} \left(\alpha \frac{d\varphi_1^e}{dx} \frac{d\varphi_1^e}{dx} + \beta \varphi_1^e \varphi_1^e \right) dx \\ &= K_{22}^{e-1} + K_{11}^e \end{aligned} \quad (2.43)$$

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

$$K_{ee} = K_{22}^{e-1} + K_{11}^e, \quad e = 2, 3, \dots, m \quad (2.44)$$

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

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

$$\begin{aligned} K_{e,e+1} &= \int_0^1 \left(\alpha \frac{d\varphi_e}{dx} \frac{d\varphi_{e+1}}{dx} + \beta \varphi_e \varphi_{e+1} \right) dx = \int_{x_e}^{x_{e+1}} \left(\alpha \frac{d\varphi_e}{dx} \frac{d\varphi_{e+1}}{dx} + \beta \varphi_e \varphi_{e+1} \right) dx \\ &= \int_{x_1^e}^{x_2^e} \left(\alpha \frac{d\varphi_1^e}{dx} \frac{d\varphi_2^e}{dx} + \beta \varphi_1^e \varphi_2^e \right) dx = K_{12}^e \end{aligned} \quad (2.45)$$

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 $e = 2, 3, \dots, m$, temos:

$$\begin{aligned} K_{ee} &= K_{22}^{e-1} + K_{11}^e \\ K_{e,e+1} &= K_{12}^e \end{aligned} \quad (2.46)$$

Para os coeficientes K_{11} , $K_{m+1,m+1}$ relativo aos nós extremos, consideremos o seguinte:

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

$$= \int_{x_1^1}^{x_2^1} \left(\alpha \frac{d\varphi_1^1}{dx} \frac{d\varphi_1^1}{dx} + \beta \varphi_1^1 \varphi_1^1 \right) dx = K_{11}^1 \quad (2.47)$$

pois,

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

De forma análoga, temos:

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

logo,

$$K_{m+1,m+1} = \int_{x_1^m}^{x_2^m} \left(\alpha \frac{d\varphi_2^m}{dx} \frac{d\varphi_2^m}{dx} + \beta \varphi_2^m \varphi_2^m \right) dx = K_{22}^m \quad (2.48)$$

De (2.46), (2.47) e (2.48) obtemos a matriz global K dada pelo algoritmo:

$$\begin{aligned} K_{11} &= K_{11}^1 \\ K_{ee} &= K_{22}^{e-1} + K_{11}^e, & e = 2, 3, \dots, m \\ K_{e,e+1} &= K_{e+1,e} = K_{12}^e, & e = 1, 2, 3, \dots, m \\ K_{m+1,m+1} &= K_{22}^m \end{aligned} \quad (2.49)$$

Como vimos anteriormente, as condições de fronteira implicam que os coeficientes da matriz $K_{11} = K_{m+1,m+1} = 0$. Pela forma como foi introduzida a interpolação local, não teremos a condição acima satisfeita, mas isto não causa problema, como veremos adiante. O algoritmo anterior possibilita a resolução de problemas com outros tipos de fronteira. Por outro lado, usando a definição de φ_a^e em (2.38) com $[x_e, x_{e+1}] = [x_1^e, x_2^e]$ no vetor força temos

$$\begin{aligned} F_e &= \int_0^1 f(x) \varphi_a^e(x) dx = \int_{x_{e-1}}^{x_{e+1}} f(x) \varphi_a^e(x) dx = \int_{x_1^{e-1}}^{x_2^{e-1}} f(x) \varphi_2^{e-1}(x) dx + \int_{x_1^e}^{x_2^e} f(x) \varphi_1^e(x) dx \\ &= F_2^{e-1} + F_1^e \end{aligned} \quad (2.50)$$

Para F_1 e F_{m+1} , temos

$$F_1 = \int_0^1 f(x) \varphi_1(x) dx = \int_{x_1^1}^{x_2^1} f(x) \varphi_1^1(x) dx = F_1^1 \quad (2.51)$$

$$F_{m+1} = \int_0^1 f(x) \varphi_{m+1}(x) dx = \int_{x_1^m}^{x_2^m} f(x) \varphi_2^m(x) dx = F_2^m \quad (2.52)$$

De (2.50), (2.51) e (2.52) temos,

$$\begin{aligned} F_1 &= F_1^1, \\ F_e &= F_2^{e-1} + F_1^e, \quad e = 2, 3, \dots, m \\ F_{m+1} &= F_2^m \end{aligned} \quad (2.53)$$

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

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

onde,

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

e

$$K_{21}^e = K_{12}^e = \int_{x_1^e}^{x_2^e} \left(\alpha \frac{d\varphi_1^e}{dx} \frac{d\varphi_2^e}{dx} + \beta \varphi_1^e \varphi_2^e \right) dx$$

De (2.38), tem-se que

$$\varphi_1^e = \frac{x_2^e - x}{h_e} \quad \varphi_2^e = \frac{x - x_1^e}{h_e}$$

Logo,

$$\frac{d\varphi_1^e}{dx} = -\frac{1}{h_e} \quad \frac{d\varphi_2^e}{dx} = \frac{1}{h_e}$$

Substituindo e fazendo os cálculos obtém-se:

$$K^e = \begin{bmatrix} \frac{\alpha}{h_e} + \frac{\beta h_e}{3} & \frac{-\alpha}{h_e} + \frac{\beta h_e}{6} \\ \frac{-\alpha}{h_e} + \frac{\beta h_e}{6} & \frac{\alpha}{h_e} + \frac{\beta h_e}{3} \end{bmatrix} \quad (2.54)$$

Usando o algoritmo (2.49), obtemos a matriz global que é tridiagonal e simétrica:

$$K = \begin{bmatrix} K_{11} & K_{12} & 0 & \cdots & 0 & 0 \\ K_{21} & K_{22} & K_{23} & \cdots & 0 & 0 \\ 0 & K_{32} & K_{33} & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \ddots & K_{m,m} & K_{m,m+1} \\ 0 & 0 & 0 & 0 & K_{m+1,m} & K_{m+1,m+1} \end{bmatrix} \quad (2.55)$$

É claro que a matriz global K calculada em (2.24) e (2.55) 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 (2.54) é válida para todo elemento $e = 1, 2, \dots, m$. Logo, para obtenção da matriz global K é suficiente o uso do algoritmo (2.49). No próximo capítulo mostraremos que para o caso bidimensional a sistemática é a mesma e portanto 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 (2.53), tem-se que

$$F_1 = F_1^1 \quad \text{e} \quad F_{m+1} = F_2^m$$

De (2.50), para $e = 2, 3, \dots, m$,

$$F_e = F_2^{e-1} + F_1^e$$

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

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

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

$$F_1^e = \int_{x_1^e}^{x_2^e} f(x) \varphi_1^e(x) dx = f_1^e \int_{x_1^e}^{x_2^e} (\varphi_1^e(x) \varphi_1^e(x)) dx + f_2^e \int_{x_1^e}^{x_2^e} (\varphi_2^e(x) \varphi_1^e(x)) dx \quad (2.56)$$

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,

$$\int_{x_1^e}^{x_2^e} (\varphi_1^e(x))^2 dx = \frac{h_e}{3} \quad (2.57)$$

e

$$\int_{x_1^e}^{x_2^e} (\varphi_2^e(x) \varphi_1^e(x)) dx = \frac{h_e}{6} \quad (2.58)$$

Substituindo em (2.56) conclui-se que:

$$F_1^e = \frac{h_e}{3} f_1^e + \frac{h_e}{6} f_2^e \quad (2.59)$$

De forma análoga, temos,

$$F_2^e = \int_{x_1^e}^{x_2^e} f(x) \varphi_2^e(x) dx = \int_{x_1^e}^{x_2^e} (\varphi_1^e(x) \varphi_2^e(x)) f_1^e(x) dx + \int_{x_1^e}^{x_2^e} (\varphi_2^e(x))^2 f_2^e(x) dx = \frac{h_e}{6} f_1^e + \frac{h_e}{3} f_2^e \quad (2.60)$$

Portanto, a força F_e é dada por

$$F_e = F_2^{e-1} + F_1^e = \frac{h_{e-1}}{6} (f_1^{e-1} + 2f_2^{e-1}) + \frac{h_e}{6} (2f_1^e + f_2^e), \quad e = 2, 3, \dots, m \quad (2.61)$$

De (2.51) e (2.52), F_1 e F_{m+1} são calculados por

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

isto é,

$$F_1 = F_1^1 = \frac{h_1}{6} (2f_1^1 + f_2^1) \quad (2.62)$$

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,

$$F_{m+1} = F_2^m = \frac{h_m}{6} (f_1^m + 2f_2^m) \quad (2.63)$$

Portanto, o vetor global F é dado por

$$F = [F_1, \dots, F_{m+1}]^T \quad (2.64)$$

Observe que a força F dada 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:

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

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, a solução $u^h = u^h(x)$, definida acima, do problema aproximado

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

também é um polinômio de grau 1 por partes.

Em geral, podemos melhorar a solução aproximada aumentando o grau do polinômio interpolador. Assim cada elemento finito e pode ser interpolado, por exemplo,

por splines cúbicos. Consequentemente, a solução aproximada $u^h(x)$ é um polinômio cúbico por partes e, em geral, se espera que a solução numérica se aproxime melhor da solução exata $u = u(x)$ do que a obtida por interpolação linear. Neste 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.

Neste contexto um método apropriado para o cálculo da integral é 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 $N = 2$, então, $\xi_1 = \frac{-\sqrt{3}}{3} = -\xi_2$ 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

$$\int_{-1}^1 g(\xi) d\xi = g\left(\frac{-\sqrt{3}}{3}\right) + g\left(\frac{\sqrt{3}}{3}\right) \quad (2.65)$$

Desde que o intervalo de integração da função g é o intervalo fechado $[-1, 1]$, então, 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:

$$\begin{aligned} \xi &: [x_1^e, x_2^e] \rightarrow [-1, 1] \\ x &\mapsto \xi(x) = \frac{2x - x_1^e - x_2^e}{h_e} \end{aligned}$$

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

$$\begin{aligned} x^e &: [-1, 1] \rightarrow [x_1^e, x_2^e] \\ \xi &\mapsto x^e(\xi) = \frac{1}{2}(x_1^e + x_2^e + h_e \xi) \end{aligned} \quad (2.66)$$

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

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

Porém,

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

logo,

$$K_{ab}^e = \int_{-1}^1 \left(\frac{4\alpha}{h_e^2} \frac{d\varphi_a^e}{d\xi} \frac{d\varphi_b^e}{d\xi} + \beta \varphi_a^e \varphi_b^e \right) \frac{h_e}{2} d\xi = \int_{-1}^1 g(\xi) d\xi = g\left(\frac{-\sqrt{3}}{3}\right) + g\left(\frac{\sqrt{3}}{3}\right), \quad (2.67)$$

onde,

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

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

$$\varphi_a^e(\xi) = \begin{cases} \varphi_1^e(\xi) &= \frac{1}{2}(1 - \xi), & \forall \xi \in [-1, 1] \\ \varphi_2^e(\xi) &= \frac{1}{2}(1 + \xi), & \forall \xi \in [-1, 1] \\ 0, & \forall \xi \notin [-1, 1] \end{cases} \quad (2.68)$$

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 (2.68) em (2.67), obtém-se a matriz local K^e dada por (2.54).

Para problemas unidimensionais que utilizam funções de interpolação linear, a transformação isoparamétrica não é vantajosa em virtude da facilidade dos cálculos dos coeficientes da matriz local K^e . Se a função $\varphi_a^e(\xi)$ for um spline cúbico, então, os cálculos seriam trabalhosos e sua aplicação juntamente com a quadratura Gaussiana facilitaria os cálculos de K^e . O uso de transformação isoparamétrica é muito útil em problemas de dimensão 2 ou 3.

De forma análoga, podemos calcular F_e definida em (2.50) diretamente usando a transformação isoparamétrica e a quadratura Gaussiana. Com efeito:

$$\begin{aligned} F_1^e &= \int_{x_1^e}^{x_2^e} f(x) \varphi_1^e(x) dx = \int_{-1}^1 f\left(\frac{1}{2}(x_1^e + x_2^e + h_e \xi)\right) \varphi_1^e(\xi) \frac{h_e}{2} d\xi = \int_{-1}^1 g_1(\xi) d\xi \\ &= g_1\left(\frac{-\sqrt{3}}{3}\right) + g_1\left(\frac{\sqrt{3}}{3}\right) \end{aligned} \quad (2.69)$$

onde usamos a aplicação $x^e(\xi)$ definida em (2.66) e a função $g_1(\xi)$ é dada por

$$g_1(\xi) = f\left(\frac{1}{2}(x_1^e + x_2^e + h_e\xi)\right)\varphi_1^e(\xi)\frac{h_e}{2} \quad (2.70)$$

onde $\varphi_1^e(\xi)$ é a função de interpolação do intervalo $[-1, 1]$. Se tomarmos, em particular, a função de interpolação linear (2.68) obtemos

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

Analogamente, para F_2^e tem-se

$$F_2^e = \int_{-1}^1 g_2(\xi) d\xi = g_2\left(\frac{-\sqrt{3}}{3}\right) + g_2\left(\frac{\sqrt{3}}{3}\right), \quad (2.71)$$

onde,

$$g_2(\xi) = f\left(\frac{1}{2}(x_1^e + x_2^e + h_e\xi)\right)\varphi_2^e(\xi)\frac{h_e}{2}, \quad (2.72)$$

ou se tomarmos (2.68),

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

Usando o algoritmo (2.53), 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:

$$f(x^e(\xi)) = f(\xi) = \sum_{a=1}^2 f_a^e \varphi_a(\xi) \quad (2.73)$$

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

$$F_a^e = \int_{x_1^e}^{x_2^e} f(x)\varphi_a^e(x) dx = \int_{-1}^1 f(\xi) \frac{\partial x^e}{\partial \xi} \varphi_a(x^e(\xi)) d\xi = \int_{-1}^1 \left(\sum_{b=1}^2 \varphi_a(\xi)\varphi_b(\xi) \frac{\partial x^e}{\partial \xi} f_b^e \right) d\xi \quad (2.74)$$

Em particular, para a interpolação linear, temos

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

Por outro lado, $\frac{dx^e}{d\xi} = \frac{h_e}{2}$. Logo

$$F_a^e = \frac{h_e}{2} \left(\int_{-1}^1 \varphi_a(\xi) \varphi_1(\xi) f_1^e + \varphi_a(\xi) \varphi_2(\xi) f_2^e \right) d\xi \quad (2.75)$$

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 (2.60) e (2.61). A diferença consiste no uso da transformação isoparamétrica.

2.7 Condições de Fronteira

A matriz global K e o vetor força F obtidos através da matriz local K^e e da força local F^e são de ordem $(m+1) \times (m+1)$ e $(m+1) \times 1$, respectivamente. Nesta formulação, as condições de fronteira foram negligenciadas.

Vamos agora introduzir quatro tipos de condições de fronteira e analisar a influência dessas condições na matriz K e no vetor F . Subdividindo o intervalo $[0,1]$ em m -partes, obtém-se $(m+1)$ 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 $[x_i, x_{i+1}] = [x_e, x_{e+1}] = [x_1^e, x_2^e]$ definimos um elemento finito e e a função de interpolação local $\varphi_a^e(x)$ dada em (2.38), para $e = 1, 2, \dots, m$.

Consideremos, agora, para a mesma equação quatro tipos de condições de fronteira:

Exemplo 1

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

Para este caso a solução u é conhecida em $x_1 = 0$ e $x_{m+1} = 1$. Assim, temos $(m-1)$ incógnitas (nós) para os quais queremos determinar a solução. Agora,

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

Então, para $x = x_1 = 0$, temos $p = u^h(x_1) = C_1\varphi_1(x_1) = C_1$. Logo, $C_1 = p$. Analogamente, para $x = x_{m+1} = 1$ obtemos $C_{m+1} = q$. Portanto, do 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, \dots, C_m .

Como a matriz global K tem ordem $(m+1) \times (m+1)$ e o vetor força F tem $(m+1)$ componentes, devemos transformar 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 (2.55) e o vetor força F é dado por (2.64). Assim, a primeira linha do sistema $KC = F$,

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

é dada por

$$K_{11}C_1 + K_{12}C_2 = F_1$$

Tomando $K_{11} = 1$, $K_{12} = 0$ e $F_1 = p$, a igualdade é verdadeira e satisfaz a condição $C_1 = p$. Para a segunda linha, temos:

$$K_{21}C_1 + K_{22}C_2 + K_{23}C_3 = F_2 \Leftrightarrow K_{22}C_2 + K_{23}C_3 = F_2 - K_{21}C_1 = F_2 - K_{21}p$$

Após o cálculo de $F_2 - K_{21}p$, faça $K_{21} = 0$.

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

Para a penúltima linha temos $K_{m,m-1}C_{m-1} + K_{m,m}C_m = F_m - qK_{m,m+1}$. Após o cálculo de $F_m - qK_{m,m+1}$, faça $K_{m,m+1} = 0$. Portanto, o sistema linear $KC = F$ de ordem $(m+1) \times (m+1)$ que satisfaz as condições de fronteira é dado por

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & K_{22} & K_{23} & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & K_{m,m-1} & K_{m,m} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ \vdots \\ \vdots \\ C_m \\ C_{m+1} \end{bmatrix} = \begin{bmatrix} p \\ F_2 - pK_{21} \\ F_3 \\ \vdots \\ \vdots \\ F_m - qK_{m,m+1} \\ q \end{bmatrix} \quad (2.76)$$

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

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

Assim, a solução aproximada nos nós x_i é dada por

$$u^h(x_i) = C_i, \quad i = 1, 2, \dots, m, m+1$$

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

Uma forma mais rigorosa de obter diretamente o sistema linear, que assegura as condições de fronteira do problema 1, pode ser feita com a seguinte formulação:

Do problema aproximado temos

$$a(u^h, v) = (f, v)$$

Seja $u^h = w^h + p^h + q^h$, onde $w^h(0) = w^h(1) = p^h(1) = q^h(0) = 0$ e $p^h(0) = p$ e $q^h(1) = q$. Nestas condições temos $u^h(0) = p$ e $u^h(1) = q$. Substituindo no problema aproximado temos:

$$a(u^h, v) = a(w^h + p^h + q^h, v) = (f, v), \quad \forall v \in H_0^1(\Omega)$$

Logo,

$$a(w^h, v) = (f, v) - a(p^h, v) - a(q^h, v) \quad (2.77)$$

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

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

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

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

Definindo

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

e

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

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

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

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

Observe que a matriz global K é a mesma matriz dada por (2.55). Calculemos, agora, a força F , observando as igualdades (2.50), (2.51), (2.52), o algoritmo (2.53) e usando o fato da matriz K ser tridiagonal:

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

Dessa forma, tem-se

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

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

O sistema (2.82) é equivalente ao sistema (2.76) quando impomos linhas e colunas extras relativas à condição de $C_1 = p$ e $C_{m+1} = q$.

O procedimento acima será usado para o problema bidimensional do próximo capítulo, para valores de fronteira.

Exemplo 2

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

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 , $i = 1, \dots, m+1$.

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

$$\int_0^1 -\alpha u_{xx} v \, dx + \int_0^1 \beta u v \, dx = \int_0^1 f v \, dx, \quad \forall v \in H^1(\Omega).$$

Integrando por partes a 1ª integral, tem-se

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

Assim,

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

Definindo

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

e

$$(f, v) = \int_0^1 f(x) v(x) \, dx \quad (2.84)$$

a formulação variacional é dada por:

Determinar $u \in H^1(\Omega)$, tal que

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

O problema aproximado é formulado por

Determinar $u^h \in V_m$, tal que

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

onde $V_m = [\varphi_1, \dots, \varphi_m]$ é um subespaço de $H^1(\Omega)$ gerado pelos m -primeiros 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

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

e

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

para $i = 1, \dots, m+1$, obtém-se um sistema linear da forma $KC = F$. A matriz global K é a mesma dada por (2.55). 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

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

Para $e = 1$,

$$\begin{aligned} F_1^1 &= f_1^1 - \alpha p \\ F_2^1 &= f_2^1 \end{aligned} \quad (2.88)$$

pois $\varphi_1^1(x_1) = \varphi_1^1(0) = 1$, $\varphi_2^1(x_1) = \varphi_2^1(0) = 0$ e $\varphi_a^1(x_{m+1}) = \varphi_a^1(1) = 0$.

Para $e = 2, \dots, m-1$,

$$F_a^e = f_a^e$$

pois $\varphi_a^e(0) = \varphi_a^e(1) = 0$.

Para $e = m$,

$$F_1^m = f_1^m$$

e

$$F_2^m = f_2^m + \alpha q \quad (2.89)$$

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 (2.53),

$$\begin{aligned} F_1 &= F_1^1, \\ F_e &= F_2^{e-1} + F_1^e, \quad e = 2, \dots, m \quad \text{e} \\ F_{m+1} &= F_2^m \end{aligned}$$

De (2.59) e (2.60) obtém-se para $e = 2, \dots, m$,

$$\begin{aligned} F_1^e &= \frac{h_e}{3} f_1^e + \frac{h_e}{6} f_2^e \\ F_2^e &= \frac{h_e}{6} f_1^e + \frac{h_e}{3} f_2^e \end{aligned}$$

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

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

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

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

onde $f_1^e = f_2^{e-1} = f(x_e)$ e $f_2^e = f_1^{e+1} = f(x_{e+1})$. Portanto, $F = [F_1, \dots, F_{m+1}]^T$.

Utilizando a matriz global K dada em (2.55) e o vetor força dado acima, obtém-se, então, o sistema linear de ordem $(m+1)$, cuja solução $C = (C_1, \dots, C_{m+1})$ 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

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

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, \dots, C_m , onde $C_i = u^h(x_i)$, $i = 1, \dots, m$.

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

$$\begin{aligned} K_{ij} &= a(\varphi_i, \varphi_j) = \int_0^1 \left(\alpha \frac{d\varphi_i}{dx} \frac{d\varphi_j}{dx} + \beta \varphi_i \varphi_j \right) dx \quad \text{e} \\ F_i &= \int_0^1 f \varphi_i dx - \alpha p \varphi_i(0) \end{aligned}$$

Para $1 \leq i, j \leq m$, 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,

$$\begin{bmatrix} K_{11} & K_{12} & \cdots & 0 & 0 \\ K_{21} & K_{22} & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & K_{mm} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \\ C_{m+1} \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m - qK_{m,m+1} \\ q \end{bmatrix} \quad (2.90)$$

onde $K_{m,m+1} = \frac{-\alpha}{h_m} + \frac{\beta h_m}{6}$ é dado por (2.54) e F_m é dado por (2.61).

Exemplo 4

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

onde p e q são números reais.

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

$$\begin{bmatrix} K_{11} & K_{12} & \cdots & 0 & 0 \\ K_{21} & K_{22} & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & K_{mm} & K_{m,m+1} \\ 0 & 0 & \cdots & K_{m+1,m} & K_{m+1,m+1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \\ C_{m+1} \end{bmatrix} = \begin{bmatrix} p \\ F_2 - pK_{12} \\ \vdots \\ F_m \\ F_{m+1} \end{bmatrix} \quad (2.91)$$

onde todos os elementos K_{ij} , com $2 \leq i, j \leq m+1$ são iguais ao do exemplo 2, bem como o vetor F_j , $j = 2, \dots, m+1$. Além disso, por (2.54), $K_{12} = \frac{-\alpha}{h_1} + \frac{\beta h_1}{6}$.

A substituição dos elementos do sistema original é para assegurar a condição de fronteira $u(0) = u(x_1) = p$.

2.8 Programa Computacional

Descrevemos abaixo as subrotinas usadas para o desenvolvimento do programa computacional usando linguagem C para o cálculo da solução aproximada nos problemas que daremos a seguir.

1. InputData

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

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, ou seja,

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

Se o nó x_1 é do tipo 2, então, a derivada da solução u é conhecida, ou seja,

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

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 $X[i]$, $i = 1, 2, \dots, \mathbf{Nos}$, do intervalo são as coordenadas globais. Para a malha uniforme é suficiente introduzir os extremos do intervalo e o número de elementos (\mathbf{Nel}) da discretização, que 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 $e = 1, 2, \dots, m$, define-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 nós globais A da discretização:

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

Assim, 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 (2.66) a função

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

onde

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

onde 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 (2.54). Desde que NoLG($e, 1$) = e e NoLG($e, 2$) = $e + 1$, para alocar a matriz local para cada e devemos usar o algoritmo:

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

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^o, 2^o e 3^o colunas, respectivamente, ou seja,

$$\begin{cases} K_{e+1,1} = K_{e,e-1} \\ K_{e,2} = K_{e,e} \\ K_{e,3} = K_{e,e+1} \end{cases}$$

8. VetorForca

Para calcular a força local $f(x)$ utilizando a quadratura Gaussiana, onde definimos as funções g_1 e g_2 em (2.70) e (2.72), 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:

$$\begin{aligned} F_1 &= F_1^1 \\ F_e &= F_2^{e-1} + F_1^e, \quad e = 2, 3, \dots, m \\ F_{m+1} &= F_2^m \end{aligned}$$

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

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

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:

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

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,

$$F_1 \leftarrow F_1 - \alpha p$$

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

$$F_{m+1} \leftarrow F_{m+1} + \alpha q$$

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

10. LinearSolver

Para qualquer que seja o tipo de fronteira, 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 2.96 e 2.97.

12. OutputData

Imprime a matriz rigidez e o vetor força

13. SolExata

As soluções exatas (2.92), (2.93), (2.94) e (2.95) 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 daremos exemplos para o problema modelo com diferentes tipos de fronteira. Estimativas de erros de soluções numéricas em geral serão dadas no próximo capítulo. Porém, se conhecemos a solução exata do problema podemos calcular os erros, nas normas do $H^1(0, 1)$ e $L^2(0, 1)$, entre a solução numérica e exata diretamente.

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,

$$u_1(x) = \frac{x}{\beta} + \frac{1}{e^\eta - e^{-\eta}} \left\{ (q - pe^{-\eta} - \frac{1}{\beta})e^{\eta x} + (pe^\eta - q + \frac{1}{\beta})e^{-\eta x} \right\} \quad (2.92)$$

$$u_2(x) = \frac{x}{\beta} + (p - \frac{1}{\beta})\frac{e^{\eta x}}{\eta} + \left\{ (q - \frac{1}{\beta}) + (\frac{1}{\beta} - p)e^\eta \right\} \frac{e^{\eta x} + e^{-\eta x}}{\eta(e^\eta - e^{-\eta})} \quad (2.93)$$

$$u_3(x) = \frac{x}{\beta} + (p - \frac{1}{\beta})\frac{e^{\eta(x-1)} - e^{\eta(1-x)}}{\eta(e^\eta + e^{-\eta})} + \eta(q - \frac{1}{\beta})\frac{e^{\eta x} + e^{-\eta x}}{\eta(e^\eta + e^{-\eta})} \quad (2.94)$$

$$u_4(x) = \frac{x}{\beta} + p\eta\frac{e^{\eta(x-1)} + e^{\eta(1-x)}}{\eta(e^\eta + e^{-\eta})} + (q - \frac{1}{\beta})\frac{e^{\eta x} - e^{-\eta x}}{\eta(e^\eta + e^{-\eta})} \quad (2.95)$$

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 $L^2(\Omega)$, $H^1(\Omega)$ e a seminorma $H_0^1(\Omega)$ do erro E são definidas por

$$\|E\|_0 = \left\{ \int_{\Omega} |E|^2 dx \right\}^{1/2}$$

$$\|E\|_1 = \left\{ \int_{\Omega} \left(|E|^2 + \left| \frac{dE}{dx} \right|^2 \right) dx \right\}^{1/2}.$$

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

$$\|E\|_0 = \left\{ h \sum_{i=1}^m E_i^2 \right\}^{1/2}, \quad (2.96)$$

$$\|E\|_1 = \left\{ h \sum_{i=1}^m E_i^2 + h \sum_{i=1}^m \frac{\partial E_i^2}{\partial x} \right\}^{1/2} \quad (2.97)$$

A norma discreta $\|E\|_1$ pode ser escrita na forma equivalente,

$$\begin{aligned}\|E\|_1 &= \left\{ h \sum_{i=1}^m E_i^2 \right\}^{1/2} + \left\{ h \sum_{i=1}^m \left(\frac{\partial E_i}{\partial x} \right)^2 \right\}^{1/2} \\ &= \left\{ h \sum_{i=1}^m E_i^2 \right\}^{1/2} + \left\{ \frac{1}{h} \sum_{i=1}^m (E_{i+1} - E_i)^2 \right\}^{1/2}\end{aligned}\quad (2.98)$$

onde denotamos $E_i = E(x_i)$. As normas discretas anteriores 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. Assim por exemplo a norma discreta em $L^2(\Omega)$ e $H_0^1(\Omega)$ é dada por:

$$\begin{aligned}\|E\|_0 &= \left\{ \frac{h}{2} \left(E_1^2 + 2 \sum_{i=2}^m E_i^2 + E_{m+1}^2 \right) \right\}^{1/2} \\ \|E\|_1 &= \|E\|_0 + \left\{ \frac{1}{2h} \left((E_2 - E_1)^2 + 2 \sum_{i=2}^{m-1} (E_{i+1} - E_i)^2 + \right. \right. \\ &\quad \left. \left. (E_{m+1} - E_m)^2 \right) \right\}^{1/2}\end{aligned}\quad (2.99)$$

Solução Numérica

Exemplo 1 Problema de Dirichlet

$$\begin{cases} -\alpha u_{xx} + \beta u = f \\ u(0) = p, \quad u(1) = q \end{cases}\quad (2.100)$$

(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

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

Para efeito de comparação, para mostrar que o erro depende da malha, segue abaixo

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

os erros absolutos e relativos para 5 e 10 elementos, respectivamente,

$$\left\{ \begin{array}{ll} \|E_a\|_0 = 1.2064 \times 10^{-4}, & \|E_a\|_1 = 4.2306 \times 10^{-4}, \\ \|E_r\|_0 = 0.3045\%, & \|E_r\|_1 = 0.3057\%. \end{array} \right.$$

$$\left\{ \begin{array}{ll} \|E_a\|_0 = 3.131 \times 10^{-5}, & \|E_a\|_1 = 1.0779 \times 10^{-4}, \\ \|E_r\|_0 = 0.0758\%, & \|E_r\|_1 = 0.0762\%, \end{array} \right.$$

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

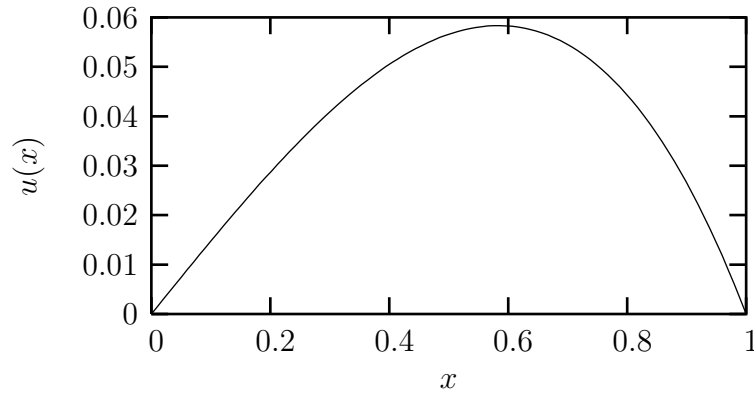
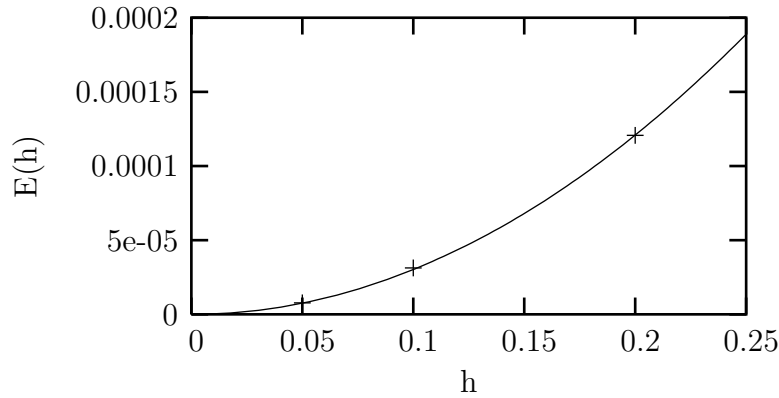


Figura 2.3: Solução exata

Figura 2.4: Erro absoluto na norma $L^2(0, 1)$

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

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

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

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 .

(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 $\|E_r\|_0 = 0.0000\%$ e $\|E_r\|_1 = 0.0000\%$.

Mas, devido aos erros de arredondamento, foram encontrados os seguintes erros absolutos e relativos para 20 elementos:

$$\begin{aligned}\|E_a\|_0 &= 1.452 \times 10^{-6}, & \|E_r\|_1 &= 5.075 \times 10^{-6}, \\ \|E_r\|_0 &= 0.0003\%, & \|E_r\|_1 &= 0.0004\%.\end{aligned}$$

(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

$$\begin{aligned}\|E_a\|_0 &= 2.577 \times 10^{-6}, & \|E_a\|_1 &= 1.555 \times 10^{-5}, \\ \|E_r\|_0 &= 0.0005\%, & \|E_r\|_1 &= 0.0013\%.\end{aligned}$$

Exemplo 2 *Problema de Neumann*

$$\begin{cases} -\alpha u_{xx} + \beta u = f \\ u'(0) = p, \quad u'(1) = q \end{cases} \quad (2.101)$$

(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

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

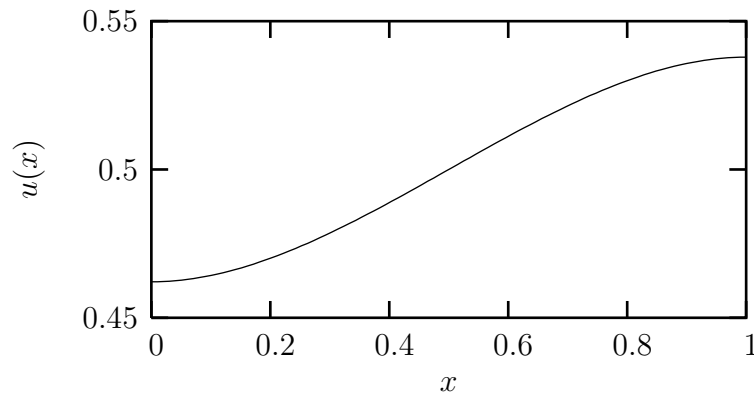


Figura 2.5: Solução exata

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)

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

Os erros absolutos e relativos obtidos para 20 elementos são

$$\begin{aligned}\|E_a\|_0 &= 1.4875 \times 10^{-5}, \quad \|E_a\|_1 = 2.025 \times 10^{-5}, \\ \|E_r\|_0 &= 0.0030\%, \quad \|E_r\|_1 = 0.0040\%.\end{aligned}$$

(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

$$\begin{aligned}\|E_a\|_0 &= 2.2002 \times 10^{-4}, \quad \|E_a\|_1 = 2.2020 \times 10^{-4}, \\ \|E_r\|_0 &= 0.0274\%, \quad \|E_r\|_1 = 0.0419\%.\end{aligned}$$

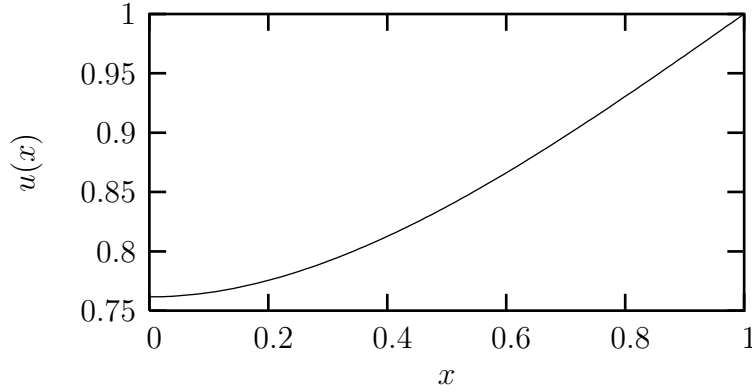


Figura 2.6: Solução exata

Exemplo 3 *Problema misto*

(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

$$\begin{aligned}\|E_a\|_0 &= 2.9346 \times 10^{-5}, \quad \|E_a\|_1 = 5.47 \times 10^{-5}, \\ \|E_r\|_0 &= 0.0034\%, \quad \|E_r\|_1 = 0.0061\%.\end{aligned}$$

(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

$$\begin{aligned}\|E_a\|_0 &= 2.2557 \times 10^{-5}, \quad \|E_a\|_1 = 4.4658 \times 10^{-5}, \\ \|E_r\|_0 &= 0.0025\%, \quad \|E_r\|_1 = 0.0048\%.\end{aligned}$$

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

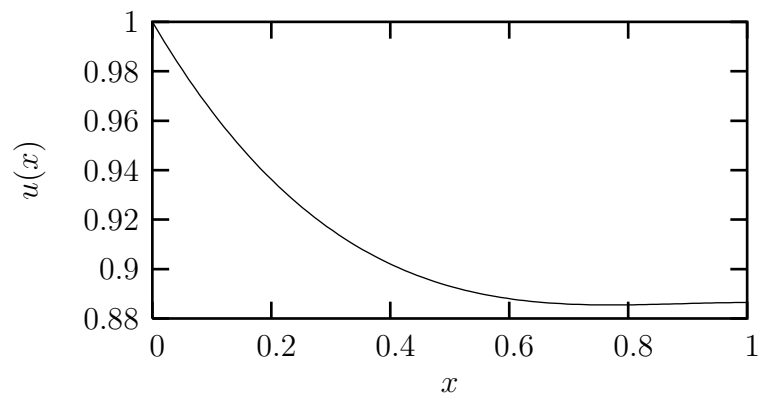


Figura 2.7: Solução exata

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 diminuimos o valor de α , o erro aumenta nas normas $L^2(0, 1)$ e $H^1(0, 1)$.
2. Define-se por malha geométrica, a malha cujos pontos nodais são definidos da seguinte forma no intervalo $[0, 1]$,

$$\begin{cases} x_i = (\alpha)^{m-i}, & i = 1, \dots, m \\ x_0 = 0 \end{cases} \quad (2.102)$$

onde α é uma constante positiva tal que $0 < \alpha < 1$. Diz-se neste caso que a malha tem razão α . Usando a malha geométrica, com $\alpha = 4/5$, altere o programa `PEU.cpp` de forma a determinar a solução aproximada do Exemplo 2 do Cap.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:

$$\begin{cases} -\left(p(x)u'(x)\right)' + q(x)u(x) = f, & \forall x \in (0, 1) \\ u(0) = 0, & u'(1) = 0, \end{cases} \quad (2.103)$$

onde u' denota $\frac{du}{dx}$, $p(x)$ e $q(x)$ são funções que representam características físicas do material da corda. Assumiremos que $p, q \in C^1(0, 1)$ e além disso, existem constantes positivas $\alpha_1, \alpha_2, \beta_1, \beta_2$ e δ tais que:

- (i) $\alpha_1 \geq p(x) \geq \alpha_2 > 0, \quad \beta_1 \geq q(x) \geq \beta_2 \geq 0, \quad \forall x \in [0, 1]$
- (ii) $|p'(x)| \leq \delta \quad \forall x \in [0, 1]$

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:

$$\int_0^1 (pu'v' + quv)dx = \int_0^1 fv \, dx, \quad \forall v \in V$$

- (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 a a força local para cada elemento \mathbf{e} .

Sugestão: Considere as funções p e q constantes sobre cada elemento \mathbf{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. $p(x) = 1 + x$, $q(x) = 2$ e $f(x) = 2x(4 - x)$.

ii. $p(x) = 1 + \frac{1}{2}x^2$, $q(x) = \frac{1 + 3x}{2 - x}$ e $f(x) = 6x^2 - x + 2$.

- (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 do $L^2(0, 1)$ e $H^1(0, 1)$.

Função Base e Estimativa de Erro

No capítulo anterior solução aproximada é gerada pelas funções bases φ_i , onde φ_i são funções lineares por partes em cada intervalo $I_i = [x_{i-1}, x_{i+1}]$ e $\varphi_i(x_j) = 0$ se $x_j \notin I_i$. Neste capítulo são introduzidas outros tipos de funções bases φ_i , os polinômios por partes de grau 2 e grau 3, com os procedimentos para obtenção da solução aproximada, isto é, a matriz e força local para cada elemento e algoritmos para obtenção da matriz e força global. Introduzimos também estimativas de erros para espaços de Sobolev e a relação entre o erro e o grau do polinômio gerador do espaço das soluções aproximadas V_m . Além das considerações do erro de interpolação, o uso de funções base de ordem superior é indispensável para alguns problemas, como o problema da viga elástica que apresentaremos.

3.1 Função Base de Ordem Superior

Em (2.19), definimos a função $\varphi_i(x)$, que é linear por partes em cada intervalo $I_i = [x_{i-1}, x_{i+1}]$ e com essa base, gera-se o subespaço $V_m = [\varphi_1, \varphi_2, \dots, \varphi_m]$, dado por

$$V_m = \{v_h \in V; v_h|_{I_i} = v_h^i \in C^0 = P_1(I_i)\},$$

onde $P_1(I_i)$ é um polinômio de grau 1 em I_i .

É natural a questão de saber se aumentando o grau do polinômio interpolador então melhoraremos a ordem de convergência da solução aproximada? Para responder a esta questão, primeiro vamos introduzir exemplos de bases interpoladoras de grau $\nu = 2$ grau $\nu = 3$.

Considere o espaço de elementos finitos V_m^k definido por,

$$V_m^k = \{v_h \in V; v_h^i \in P_k(I_i)\},$$

onde por v_h^i estamos denotando a restrição de v_h ao intervalo (elemento) $I_i = [x_{i-1}, x_i]$, V é um espaço de Hilbert e P_k é o conjunto dos polinômios definidos em I_i , com graus

menores ou iguais a k na variável x . Note que cada v_h deve satisfazer as mesmas condições de fronteira do espaço V . A seguir serão definidos subespaços V_m^k do espaço V , usando duas bases clássicas de polinômios de grau $k = 2$ e $k = 3$, definidos em cada elemento I_i . Sem perda de generalidade, consideraremos o espaço de Hilbert $V = H_0^1(0, 1)$. Quando as condições de fronteira não são de Dirichlet então $V = H^1(0, 1)$ e as mudanças das funções bases nos extremos seguem o mesmo procedimento do capítulo anterior.

3.1.1 Base Quadrática

Considere o subespaço

$$V_m^2 = \{v_h \in V; v_h|_{I_i} = v_h^i \in P_2(I_i)\},$$

onde P_2 são polinômios de grau 2, definidos em cada elemento finito I_i . Para gerar cada um dos polinômios interpoladores por partes de grau 2 são necessários três pontos e desta forma o intervalo $\Omega = (0, 1)$ deve ser discretizado num número par de pontos, ou seja, considerando o intervalo $I_i = [x_{2i}, x_{2(i+1)}]$ e um outro ponto adicional, denotado por $x_{2i+1} \in I_i$, gerando os subintervalos $[x_{2i}, x_{2i+1}]$ e $[x_{2i+1}, x_{2(i+1)}]$ para $i = 0, 1, \dots, n-1$, com $x_0 = 0$ e $x_{2m} = 1$. Dessa forma, definimos $h_i = x_{2(i+1)} - x_{2i}$, o passo de cada intervalo. Então o subespaço V_m^2 é gerado pelas funções testes $[\varphi_0, \varphi_1, \dots, \varphi_{2m}]$. Estas funções são definidas da seguinte forma:

As funções testes pares φ_{2i} satisfazem as condições,

$$\begin{aligned} \varphi_{2i}(x_{2i}) &= 1, & \varphi_{2i}(x_{2i+1}) &= \varphi_{2i}(x_{2i+2}) = 0, & \text{em } I_i, \\ \varphi_{2i}(x_{2i-1}) &= \varphi_{2i}(x_{2i-2}) = 0 & & & \text{em } I_{i-1}. \end{aligned}$$

As funções testes ímpares φ_{2i+1} satisfazem as condições,

$$\varphi_{2i+1}(x_{2i+1}) = 1, \quad \varphi_{2i+1}(x_{2i}) = \varphi_{2i+1}(x_{2i+2}) = 0, \quad \text{em } I_i.$$

Portanto podemos defini-las explicitamente por

$$\varphi_{2i}(x) = \begin{cases} -\frac{1}{h_i}(x - x_{2i-2}) + \frac{2}{h_i^2}(x - x_{2i-2})^2, & \forall x \in [x_{2i-2}, x_{2i}], \\ 1 - \frac{3}{2h_i}(x - x_{2i}) + \frac{1}{2h_i^2}(x - x_{2i})^2, & \forall x \in [x_{2i}, x_{2i+2}], \\ 0, & \forall x \notin [x_{2i-2}, x_{2i+2}]. \end{cases} \quad (3.1)$$

$$\varphi_{2i+1}(x) = \begin{cases} -\frac{4}{h_i^2}(x - x_{2i})(x - x_{2i+2}), & \forall x \in [x_{2i}, x_{2i+2}], \\ 0, & \forall x \notin [x_{2i}, x_{2i+2}]. \end{cases} \quad (3.2)$$

Nos extremos $x_0 = 0$ e $x_{2m} = 1$ temos respectivamente,

$$\begin{aligned} \varphi_0 &= \begin{cases} 1 - \frac{3}{2h_i} x + \frac{1}{2h_i^2} x^2, & \forall x \in [x_0, x_2], \\ 0, & \forall x \notin [x_0, x_2] \end{cases}, \\ \varphi_{2n} &= \begin{cases} 1 + \frac{3}{h_i}(x-1) + \frac{2}{h_i^2}(x-1)^2, & \forall x \in [x_{2m-2}, x_{2m}], \\ 0, & \forall x \notin [x_{2m-2}, x_{2m}]. \end{cases} \end{aligned}$$

Note que no nó x_{2i} , temos que $(\varphi_{2i}, \varphi_{2i+j}) = 0$ se $|j| \geq 3$, logo a matriz global K é uma matriz pentagonal.

Matriz Rigidez Local

Considere a matriz do problema exemplo, definida por,

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

Em termos de matriz local temos, como anteriormente, definimos o elemento e em cada intervalo $I_i = [x_{2i}, x_{2i+2}]$, para $i = 0, 1, \dots, m-1$, com $x_0 = 0$ e $x_{2m} = 1$. Assim redefinindo os intervalos em relação as coordenadas locais do elemento e tem-se que, $[x_{2i}, x_{2i+2}] = [x_1^e, x_2^e]$. Desta forma em cada elemento e estão definidas as seguintes funções de interpolação local $\varphi_a^e(x)$, do elemento e em $[x_1^e, x_2^e]$, (veja Fig. 3.1)

$$\begin{cases} \varphi_1^e(x) = 1 - \frac{3}{h_e}(x - x_1^e) + \frac{2}{h_e^2}(x - x_1^e)^2, \\ \varphi_2^e(x) = -\frac{4}{h_e^2}(x - x_1^e)(x - x_2^e), \\ \varphi_3^e(x) = -\frac{1}{h_e}(x - x_1^e) + \frac{2}{h_e^2}(x - x_1^e)^2, \end{cases} \quad (3.3)$$

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

Assim cada matriz local K_{ab}^e de ordem 3×3 e Força Local F_a^e são dadas por

$$\begin{aligned} K_{ab}^e &= \int_{x_1^e}^{x_2^e} \left(\alpha \frac{d\varphi_a^e}{dx} \frac{d\varphi_b^e}{dx} + \beta \varphi_a^e \varphi_b^e \right) dx & 1 \leq a, b \leq 3, \\ F_a^e &= \int_{x_1^e}^{x_2^e} f \varphi_a^e dx & 1 \leq a \leq 3. \end{aligned} \quad (3.4)$$

Explicitamente os termos da matriz local (3.4) são dados por

$$\begin{aligned} k_{11}^e &= k_{33}^e = \alpha \frac{7}{3h_e} + \beta \frac{2h_e}{15}, & k_{12}^e &= k_{21}^e = k_{23}^e = k_{32}^e = -\alpha \frac{8}{3h_e} + \beta \frac{h_e}{15}, \\ k_{22}^e &= \alpha \frac{16}{3h_e} + \beta \frac{8h_e}{15}, & k_{13}^e &= k_{31}^e = \alpha \frac{1}{3h_e} - \beta \frac{h_e}{30}. \end{aligned}$$

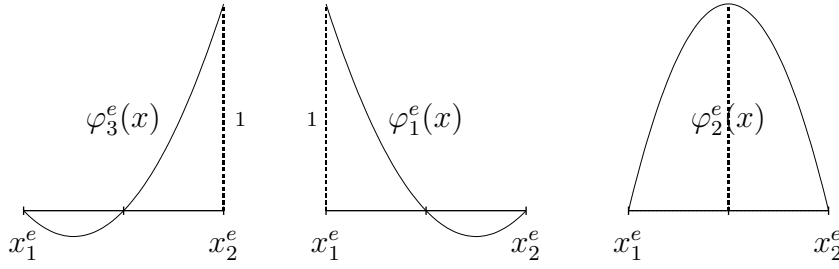


Figura 3.1: Função base local: quadrática

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:

$$\begin{aligned}
 K_{00} &= k_{11}^1, & K_{01} &= k_{12}^1, & K_{02} &= k_{13}^1, \\
 K_{11} &= k_{22}^1, & K_{12} &= k_{23}^1, & K_{2m-2m} &= k_{33}^{m-1}.
 \end{aligned}$$

$$\left\{ \begin{array}{ll}
 K_{(2e-1),(2e+1)} &= K_{(2e+1),(2e-1)} = 0 \\
 K_{(2e),(2e)} &= k_{33}^e + k_{11}^{e+1} \\
 K_{(2e),(2e+1)} &= k_{12}^e \\
 K_{(2e),(2e+2)} &= k_{13}^e \\
 K_{(2e+1),(2e+1)} &= k_{22}^e \\
 K_{(2e+1),(2e+2)} &= k_{23}^e
 \end{array} \right. \quad e = 1, 2, 3, \dots, m-1. \quad (3.5)$$

A contribuição do vetor força local f_a^e para o vetor força global F de ordem $(2m + 1)$ é dado pelo algoritmo:

$$\begin{aligned}
 F_{2e} &= f_3^{e-1} + f_1^e, & F_{2e+1} &= f_2^e, & e &= 1, 2, 3, \dots, m-1, \\
 F_0 &= f_1^1, & F_1 &= f_2^1, \\
 F_{2m-1} &= f_2^{m-1}, & F_{2m} &= f_3^{m-1}.
 \end{aligned}$$

A introdução dos valores de fronteira depende do tipo de fronteira do problema, ou seja, se são do tipo de Dirichlet, Neumann ou misto e a inserção segue o mesmo procedimento da interpolação linear.

3.1.2 Base Cúbica

Um spline cúbico, denotado por $C(x)$, é um polinômio interpolador por partes de grau 3 em cada intervalo $I_i = [x_i, x_{i+1}]$, que interpola uma dada função $g(x)$ e é duas vezes diferenciável em cada nó x_i . A função $C(x)$ pode ser representada por

$$C(x) = C_i(x) = a_{i3}x^3 + a_{i2}x^2 + a_{i1}x^1 + a_{i0}, \quad x \in I_i, \quad i = 1, 2, \dots, m-1$$

Para que $C(x)$ satisfaça as propriedades de interpolação e de ser duas vezes diferenciável é necessário determinar os coeficientes a_{ij} de $C(x)$ de forma que:

1. $C(x_i) = g(x_i), \quad i = 1, 2, \dots, m$
2. $C_{i-1}(x_i) = C_i(x_i), \quad C'_{i-1}(x_i) = C'_i(x_i), \quad C''_{i-1}(x_i) = C''_i(x_i), \quad i = 2, 3, \dots, m-1.$

Os coeficientes $a_{i3}, a_{i2}, a_{i1}, a_{i0}$ a ser determinados em $C(x) = C_i(x)$ totalizam $4(m-1)$ incógnitas e o número de condições requeridas são: $m + 3(m-2) = 4m - 6$. Assim são necessários impor mais duas condições adicionais para que $C(x)$ seja determinado de forma única. Uma das possibilidades é tomar as seguintes condições adicionais :

$$C''(x_1) = C''(x_m) = 0, \quad (3.6)$$

e nesse caso o spline é denominado *spline cúbico natural*. Mas este spline cúbico não é a melhor escolha para a base, pois não satisfaz a condição:

$$C'_j(x_i) = C_j(x_i) = 0, \quad \text{se } |i - j| \geq 2, \quad (3.7)$$

ou seja, queremos determinar um spline cúbico que se anula fora do intervalo $[x_{i-2}, x_{i+2}]$ e dessa forma obter uma matriz global que tenha a propriedade especial de ser heptagonal, ou seja,

$$K_{ij} = \int_0^1 \left(\alpha \frac{d\varphi_i}{dx} \frac{d\varphi_j}{dx} + \beta \varphi_i \varphi_j \right) dx = 0 \quad \text{se } |i - j| > 3,$$

onde tomamos $\varphi_j(x) = C_j(x)$.

Inicialmente consideramos uma função par $\theta(x)$ tal que

$$\theta(x) = \begin{cases} \theta_1(x) & \forall x \in [0, 1], \\ \theta_2(x) & \forall x \in [1, 2], \\ \theta_1(-x) & \forall x \in [-1, 0], \\ \theta_2(-x) & \forall x \in [-2, -1], \\ 0 & \forall x \notin [-2, 2]. \end{cases}$$

Esta função, que é um spline cúbico, onde $\theta_1(x) = a_0 + a_1x + a_2x^2 + a_3x^3$ e $\theta_2(x) = b_0 + b_1x + b_2x^2 + b_3x^3$, devem satisfazer as seguintes condições:

$$\begin{cases} \theta_1(0) = 1, & \theta'_1(0) = 0, \\ \theta_1(1) = \theta_2(1), & \theta'_1(1) = \theta'_2(1), & \theta''_1(1) = \theta''_2(1), \\ \theta_2(2) = 0, & \theta'_2(2) = 0, & \theta''_2(2) = 0. \end{cases}$$

Com estas 8 condições podemos determinar os 8 coeficientes $\{a_i, b_i\}$ por um sistema linear. Assim $\theta_1(x)$ e $\theta_2(x)$ são dadas por

$$\begin{cases} \theta_1(x) = 1 - \frac{3}{2}x^2 + \frac{3}{4}x^3, \\ \theta_2(x) = 2 - 3x + \frac{3}{2}x^2 - \frac{1}{4}x^3. \end{cases} \quad (3.8)$$

Considere agora, a discretização uniforme do intervalo $[a, b]$ na forma $a = x_1 < x_2, \dots < x_{m+1} = b$, com $h = x_{i+1} - x_i$. Então, para cada nó x_i , com $i = 1, 2, \dots, m+1$, definimos o elemento de função base spline cúbico, conhecido como *B-spline*,

$$B_i(x) = \theta\left(\frac{x - x_i}{h}\right).$$

Nestas condições a função $B_i(x)$ pode ser reescrita explicitamente na seguinte forma:

$$B_i(x) = \begin{cases} \frac{1}{4h^3}(x - x_{i-2})^3, & \forall x \in [x_{i-2}, x_{i-1}], \\ \frac{1}{4} + \frac{3}{4h}(x - x_{i-1}) + \frac{3}{4h^2}(x - x_{i-1})^2 - \frac{3}{4h^3}(x - x_{i-1})^3, & \forall x \in [x_{i-1}, x_i], \\ \frac{1}{4} + \frac{3}{4h}(x_{i+1} - x) + \frac{3}{4h^2}(x_{i+1} - x)^2 - \frac{3}{4h^3}(x_{i+1} - x)^3, & \forall x \in [x_i, x_{i+1}], \\ \frac{1}{4h^3}(x_{i+2} - x)^3, & \forall x \in [x_{i+1}, x_{i+2}] \\ 0, & \forall x \notin [x_{i-2}, x_{i+2}] \end{cases}$$

Pode se mostrar que $B_i(x)$ de fato é 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 de Dirichlet. Mas especificamente considere que os valores de fronteira da equação diferencial associada sejam nulos, ou seja, $u(x_1) = u(x_{m+1}) = 0$. As funções B_3, \dots, 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

$$\begin{cases} \varphi_i(x) = B_i(x), & i = 3, \dots, m-1, \\ \varphi_1(x) = B_1(x) - 4B_0(x), \\ \varphi_2(x) = B_2(x) - B_0(x), \\ \varphi_m(x) = B_m(x) - B_{m+2}(x), \\ \varphi_{m+1}(x) = B_{m+1}(x) - 4B_{m+2}(x), \end{cases} \quad (3.9)$$

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 que

$$\begin{aligned}\varphi_i(x_i) &= 1, \quad i = 2, 3, \dots, m; & \varphi_i(x_{i+1}) &= 1/4, \quad i = 1, 2, \dots, m-1, \\ \varphi_i(x_{i-1}) &= 1/4, \quad i = 3, 4, \dots, m+1 \\ \varphi_1(x_1) &= \varphi_2(x_1) = \varphi_m(x_{m+1}) = \varphi_{m+1}(x_{m+1}) = 0.\end{aligned}$$

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 $V_m = [\varphi_2, \varphi_3, \dots, \varphi_m]$, dada por

$$u_h(x) = \sum_{j=2}^m d_j \varphi_j(x), \quad \varphi_j \in V_m.$$

Note que as funções φ_1 e φ_{m+1} são linearmente dependentes das funções φ_i , $i = 2, 3, \dots, m$.

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,

$$\left\{ \begin{array}{l} u_h(x_2) = d_2 + \frac{1}{4}d_3 \\ u_h(x_i) = \frac{1}{4}d_{i-1} + d_i + \frac{1}{4}d_{i+1}, \quad \text{para } i = 3, 4, \dots, m-1 \\ u_h(x_m) = \frac{1}{4}d_{m-1} + d_m \end{array} \right.$$

que pode ser representada na forma matricial $\mathbf{A}\mathbf{d} = \mathbf{u}_h$, onde $\mathbf{d} = (d_2, d_3, \dots, d_m)^t$, $\mathbf{u}_h = (u_h(x_2), u_h(x_3), \dots, u_h(x_m))^t$ e \mathbf{A} é uma matriz quadrada de ordem $m-1 \times m-1$, tridiagonal, ou seja,

$$A = \begin{bmatrix} 1 & 1/4 & 0 & \dots & 0 & 0 \\ 1/4 & 1 & 1/4 & \dots & 0 & 0 \\ 0 & 1/4 & 1 & 1/4 & \dots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & 1/4 & 1 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1 \end{bmatrix} \quad (3.10)$$

Matriz Local

Usando os mesmos argumentos anteriores, considere o elemento e definido em cada intervalo $I_i = [x_i, x_{i+1}]$, para $i = 1, \dots, m$. Então em cada elemento e do intervalo $[x_i, x_{i+1}] = [x_1^e, x_2^e]$, com $h = x_2^e - x_1^e$, estão definidas as seguintes funções de interpolação local, (ver Fig. 3.2)

$$\left\{ \begin{array}{l} \varphi_1^e(x) = \frac{1}{4} + \frac{3}{4h}(x_2^e - x) + \frac{3}{4h^2}(x_2^e - x)^2 - \frac{3}{4h^3}(x_2^e - x)^3 \\ \varphi_2^e(x) = \frac{1}{4h^3}(x_2^e - x)^3 \\ \varphi_3^e(x) = \frac{1}{4} + \frac{3}{4h}(x - x_1^e) + \frac{3}{4h^2}(x - x_1^e)^2 - \frac{3}{4h^3}(x - x_1^e)^3 \\ \varphi_4^e(x) = \frac{1}{4h^3}(x - x_1^e)^3 \end{array} \right. \quad (3.11)$$

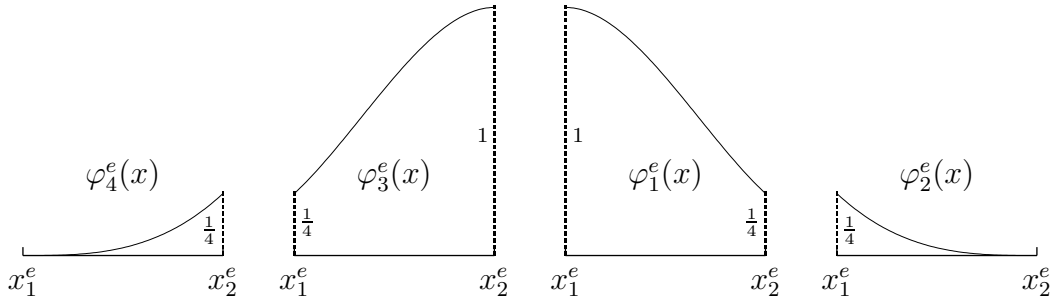


Figura 3.2: Função base local: spline cúbico

Agora cada matriz local k_{ab}^e é de ordem 4×4 e Força Local f_a^e é de ordem 4×1 dadas por

$$\begin{aligned} k_{ab}^e &= \int_{x_1^e}^{x_2^e} \left(\alpha \frac{d\varphi_a^e}{dx} \frac{d\varphi_b^e}{dx} + \beta \varphi_a^e \varphi_b^e \right) dx \quad 1 \leq a, b \leq 4, \\ f_a^e &= \int_{x_1^e}^{x_2^e} f \varphi_a^e dx \quad 1 \leq a \leq 4. \end{aligned} \quad (3.12)$$

Explicitamente os termos da matriz local (3.12) 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 (3.12), pode ser obtida pelo seguinte algoritmo:

$$\left\{ \begin{array}{ll} K_{ee} &= k_{44}^{e-2} + k_{33}^{e-1} + k_{11}^e + k_{22}^{e+1}, & e = 3, 4, \dots, m-1 \\ K_{e(e+1)} &= k_{34}^{e-1} + k_{13}^e + k_{12}^{e+1}, & e = 3, 4, \dots, m-2 \\ K_{(e+1)e} &= k_{43}^{e-1} + k_{31}^e + k_{21}^{e+1}, & e = 3, 4, \dots, m-2 \\ K_{e(e+2)} &= k_{14}^e + k_{23}^{e+1}, & e = 2, 3, \dots, m-2 \\ K_{(e+2)e} &= k_{41}^e + k_{32}^{e+1}, & e = 2, 3, \dots, m-2 \\ K_{e(e+3)} &= k_{24}^{e+1}, & e = 1, 2, \dots, m-2 \\ K_{(e+3)e} &= k_{42}^{e+1}, & e = 1, 2, \dots, m-2. \end{array} \right. \quad (3.13)$$

Considerando as condições (3.9), para satisfazer a condição de fronteira nula então o algoritmo associado para obter os elementos da matriz global K referentes a essas restrições são dados respectivamente por

$$\left\{ \begin{array}{ll} K_{11} = k_{11}^1 + 17k_{22}^2 - 4(k_{12}^2 + k_{21}^2), & K_{22} = k_{33}^1 + k_{11}^2 + k_{22}^3 - k_{32}^1 - k_{23}^1 + k_{22}^1 \\ K_{12} = k_{13}^1 + k_{21}^2 - k_{12}^1 - 4(k_{23}^1 - k_{22}^1), & K_{21} = k_{31}^1 + k_{12}^2 - k_{21}^1 - 4(k_{32}^1 - k_{22}^1), \\ K_{13} = k_{14}^1 + k_{23}^2 - 4k_{24}^1, & K_{31} = k_{41}^1 + k_{32}^2 - 4k_{42}^1, \\ K_{23} = k_{34}^1 + k_{13}^2 + k_{21}^3 - k_{24}^1, & K_{32} = k_{43}^1 + k_{31}^2 + k_{12}^3 - k_{42}^1 \end{array} \right.$$

$$\left\{ \begin{array}{ll} K_{m-1,m+1} = k_{14}^{m-2} + k_{23}^{m-1} - 4k_{24}^m, & K_{m+1,m-1} = k_{41}^{m-2} + k_{32}^{m-1} - 4k_{42}^m \\ K_{m-1,m} = k_{34}^{m-2} + k_{13}^{m-1} + k_{21}^m - k_{24}^m, & K_{m,m-1} = k_{43}^{m-2} + k_{31}^{m-1} + k_{12}^m - k_{42}^m \\ K_{m,m} = k_{44}^{m-2} + k_{33}^{m-1} + k_{11}^m - k_{14}^m - k_{41}^m + k_{44}^m, & \\ K_{m,m+1} = k_{34}^{m-1} + k_{13}^m - 4(k_{14}^m - k_{44}^m) - k_{43}^m, & \\ K_{m+1,m} = k_{43}^{m-1} + k_{31}^m - 4(k_{41}^m - k_{44}^m) - k_{34}^m, & \\ K_{m+1,m+1} = k_{33}^m + 17k_{44}^{m-1} - 4(k_{34}^{m-1} + k_{43}^{m-1}) & \end{array} \right.$$

A contribuição do vetor força local f_a^e para o vetor força global F de ordem n é dado pelo algoritmo:

$$F_e = f_4^{e-2} + f_3^{e-1} + f_1^e + f_2^{e+1} \quad \text{para } e = 3, 4, \dots, m-1$$

e nos extremos temos

$$\begin{array}{ll} F_2 = f_3^1 + f_1^2 + f_2^3, & F_1 = f_1^1 + f_2^2 \\ F_m = f_4^{m-2} + f_3^{m-1} + f_1^m, & F_{m+1} = f_4^{m-1} + f_3^m \end{array}$$

Note que até agora, foi negligenciado os valores de fronteira. Assim o sistema linear $\mathbf{Kd} = \mathbf{F}$, tem $m+1$ incógnitas $\mathbf{d} = (d_1, d_2, \dots, d_{m+1})$. A forma de inserção dos valores de fronteira é semelhante ao procedimento que obtivemos para os polinômios linear por partes (ver Seção 2.7), e dessa forma se as condições de fronteira são tipo Dirichlet, Neumann ou misto teremos, respectivamente, sistemas quadrados de ordem $m-1$, $m+1$ e m . Também observamos que o algoritmo (3.13) foi deduzido considerando a simetria da matriz local.

3.1.3 Base de Hermite

Nesta seção, vamos introduzir uma função base conveniente para determinar a solução aproximada do problema da viga elástica, formulada por Bernoulli-Euler.

Seja $\Omega = (0, 1)$. A formulação forte do problema da viga elástica com extremos encaixadas na parede, pode ser dada, formalmente, como um problema de valores de fronteira de equação diferencial de quarta ordem: Dada $f : (0, 1) \rightarrow \mathbb{R}$, queremos determinar uma função $u : [0, 1] \rightarrow \mathbb{R}$ tal que

$$\begin{cases} u_{xxxx} = f & \text{em } (0, 1), \\ u(0) = u(1) = u_x(0) = u_x(1) = 0. \end{cases} \quad (3.14)$$

Usando o mesmo procedimento feito no problema (2.1), ou seja multiplicando por uma função teste e integrando por partes duas vezes obtemos o seguinte formulação variacional:

O problem (3.14) é equivalente a determinar $u(x)$ satisfazendo

$$a(u, v) = (f, v), \quad \forall v \in V, \quad (3.15)$$

onde

$$\begin{aligned} a(u, v) &= \int_0^1 u_{xx} v_{xx} dx, \\ (f, v) &= \int_0^1 f v dx. \end{aligned} \quad (3.16)$$

Note que, para obter a formulação (3.15) é necessário que u , u_x e u_{xx} deve pertencer ao espaço $L^2(0, 1)$, isto significa que u deve pertencer ao espaço $H^2(0, 1)$. Além disso devemos requerer que $u(0) = u_x(0) = u(1) = u_x(1) = 0$, pelas condições de fronteira, ou seja, o $H_0^2(0, 1)$ é o espaço conveniente para a solução u . Também é necessário que a função f deve pertencer ao espaço $L^2(0, 1)$.

Seguindo o mesmo raciocínio, queremos determinar a solução aproximada da forma:

$$u_h(x) = \sum_{j=1}^m d_j \varphi_j(x), \quad \varphi_j \in V_m \quad (3.17)$$

onde $V_m = [\varphi_1, \varphi_2, \dots, \varphi_m]$ e $\varphi_j(0) = \varphi_j(1) = \varphi_j'(0) = \varphi_j'(1) = 0$. Vimos que determinar a solução aproximada consiste de resolver o sistema linear $Kd = F$ onde a matriz global K é dada por

$$K_{ij} = \int_0^1 \left(\frac{d^2 \varphi_i}{dx^2} \frac{d^2 \varphi_j}{dx^2} \right) dx. \quad (3.18)$$

Logo as funções testes precisam ser de $H_0^2(0, 1)$.

Obviamente, a função base de polinômios de grau 1 ou grau 2 por partes não é adequada para o problema de viga, porque a segunda derivada da função não pertence ao espaço $L^2(0, 1)$. Para este problema, uma possível escolha para a função base são os splines cúbicos, que pertencem ao espaço $C^2(0, 1)$, fazendo sentido a integral (3.18). Entretanto, para usar os splines cúbicos φ_i como elementos da função base, devemos modificá-las nos extremos $x = 0$ e $x = 1$, de tal forma que a solução aproximada definida em (3.17) satisfaça os valores de fronteira essenciais, $u_h(0) = u'_h(0) = u_h(1) = u'_h(1) = 0$.

Um outro tipo de funções testes, de interpolação cúbica, são os polinômios de Hermite que descrevemos a seguir. Note que sendo $H_0^2(0, 1)$ o espaço da soluções admissíveis então qualquer subespaço $V_m \subset H_0^2(0, 1)$ deve ser pelo menos de classe $C^1(0, 1)$, ou seja, que a função base φ_i e sua primeira derivada $(\varphi_i)_x$ seja contínua em $[0, 1]$. Os polinômios de Hermite satisfaz essa condição de continuidade e é portanto muito conveniente para o problema da viga (3.14).

Considere a discretização do intervalo $[a, b]$ na forma $a = x_0 < x_1 < \dots < x_m = b$, e para cada nó x_i , definimos dois elementos de função base, $\phi_i(x)$ e $\psi_i(x)$, satisfazendo

$$\begin{cases} \phi_i(x_i) = 1, & \phi_i(x_{i+1}) = 0, & \phi_i(x_{i-1}) = 0, \\ \phi'_i(x_i) = 0, & \phi'_i(x_{i+1}) = 0, & \phi'_i(x_{i-1}) = 0, \end{cases} \quad (3.19)$$

$$\begin{cases} \psi_i(x_i) = 0, & \psi_i(x_{i+1}) = 0, & \psi_i(x_{i-1}) = 0, \\ \psi'_i(x_i) = 1, & \psi'_i(x_{i+1}) = 0, & \psi'_i(x_{i-1}) = 0, \end{cases} \quad (3.20)$$

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 4 coeficientes com as 4 condições dadas em (3.19) e (3.20), 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:

$$\phi_i(x) = \begin{cases} \left(\left|\frac{x - x_i}{h}\right| - 1\right)^2 \left(2\left|\frac{x - x_i}{h}\right| + 1\right), & \forall x \in [x_{i-1}, x_{i+1}], \\ 0, & \forall x \notin [x_{i-1}, x_{i+1}], \end{cases} \quad (3.21)$$

$$\psi_i(x) = \begin{cases} (x - x_i) \left(\left|\frac{x - x_i}{h}\right| - 1\right)^2, & \forall x \in [x_{i-1}, x_{i+1}], \\ 0, & \forall x \notin [x_{i-1}, x_{i+1}]. \end{cases} \quad (3.22)$$

Observe que com relação a função base de Hermite, a solução aproximada $u_h(x)$ pode ser escrita na seguinte forma,

$$u_h(x) = \sum_{i=0}^m \left(c_i \phi_i(x) + d_i \psi_i(x) \right). \quad (3.23)$$

De (3.21) e (3.22), tem-se que $u_h(x_i) = c_i$ e $u'_h(x_i) = d_i$.

Note que temos $2(m+1)$ incógnitas $c = (c_0, c_1, \dots, c_m)$ e $d = (d_0, d_1, \dots, d_m)$ para se determinar a solução aproximada $u_h(x)$ e também a sua derivada $\partial u_h / \partial x$.

Para gerar as $2(m+1)$ equações correspondentes, podemos fazer o seguinte procedimento:

Considere o problema aproximado de (3.15), com a formulação (3.16), tomando u_h na forma (3.23)

$$\left(\sum_{i=0}^m (c_i \phi_i(x) + d_i \psi_i(x))_{xx}, v_{xx} \right) = (f, v), \quad \forall v \in V_m$$

Tomando $v = \phi_j(x)$ e depois tomando $v = \psi_j(x)$, obtemos dois sistemas com $(m+1)$ incógnitas cada, dado por,

$$\begin{aligned} \sum_{i=0}^m c_i ((\phi_i)_{xx}, (\phi_j)_{xx}) + \sum_{i=0}^m d_i ((\psi_i)_{xx}, (\phi_j)_{xx}) &= (f, \phi_j), \quad j = 0, 1, \dots, m \\ \sum_{i=0}^m c_i ((\phi_i)_{xx}, (\psi_j)_{xx}) + \sum_{i=0}^m d_i ((\psi_i)_{xx}, (\psi_j)_{xx}) &= (f, \psi_j), \quad j = 0, 1, \dots, m \end{aligned}$$

Definindo as matrizes:

$$\begin{aligned} K_{ij} &= ((\phi_i)_{xx}, (\phi_j)_{xx}) = \int_0^1 (\phi_i)_{xx} (\phi_j)_{xx} dx \\ L_{ij} &= ((\psi_i)_{xx}, (\phi_j)_{xx}) = \int_0^1 (\psi_i)_{xx} (\phi_j)_{xx} dx \\ M_{ij} &= ((\psi_i)_{xx}, (\psi_j)_{xx}) = \int_0^1 (\psi_i)_{xx} (\psi_j)_{xx} dx \\ F_j^1 &= \int_0^1 f(x) \phi_j(x) dx \quad F_j^2 = \int_0^1 f(x) \psi_j(x) dx \end{aligned}$$

Então podemos escrever os dois sistemas na forma

$$\begin{cases} Kc + L^T d = F^1 \\ Lc + Md = F^2 \end{cases} \quad (3.24)$$

Note que os dois sistemas são dependentes e podem ser escritos na seguinte forma, denominado de sistema bloco:

$$\begin{bmatrix} K & L^T \\ L & M \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} F^1 \\ F^2 \end{bmatrix} \quad (3.25)$$

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. Cada vetor $\{c, d, F^1, F^2\}$ tem $(m+1)$ componentes.

Como vimos em (2.31) as forças podem ser interpoladas, usando como base a função ϕ_i na forma

$$f(x) = \sum_{i=0}^m f_i \phi_i(x) \quad (3.26)$$

Assim, podemos escrever, como em (2.32)

$$\begin{aligned} F_j^1 &= \int_0^1 f(x) \phi_j(x) dx = \int_0^1 \sum_{i=0}^m f_i \phi_i(x) \phi_j(x) dx \\ &= f_{i-1} \int_{x_{i-1}}^{x_i} \phi_i \phi_{i-1} dx + f_i \int_{x_{i-1}}^{x_{i+1}} (\phi_i)^2 dx + f_{i+1} \int_{x_i}^{x_{i+1}} \phi_i \phi_{i+1} dx \end{aligned}$$

$$\begin{aligned} F_j^2 &= \int_0^1 f(x) \psi_j(x) dx = \int_0^1 \sum_{i=0}^m f_i \phi_i(x) \psi_j(x) dx \\ &= f_{i-1} \int_{x_{i-1}}^{x_i} \phi_i \psi_{i-1} dx + f_i \int_{x_{i-1}}^{x_{i+1}} \phi_i \psi_i dx + f_{i+1} \int_{x_i}^{x_{i+1}} \phi_i \psi_{i+1} dx \end{aligned}$$

Note que também podemos interpolar a função $f(x)$ usando como função base a função $\psi'_i(x)$, ou seja, $f(x) = \sum_{i=0}^m f_i \psi'_i(x)$ e calcular F^1 e F^2 de forma análoga.

Condições de Fronteira: O sistema linear bloco $\widehat{\mathbf{K}}\widehat{\mathbf{C}} = \widehat{\mathbf{F}}$ tem infinitas soluções, ou seja a matriz $\widehat{\mathbf{K}}$ é singular. Para obter uma única solução $\widehat{\mathbf{C}}$ temos que introduzir os valores de fronteira dados pelo problema para garantir que a solução aproximada satisfaça os valores de fronteira. Em particular, para o exemplo (3.14) devemos ter a seguinte condição 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 seja,

$$\begin{aligned} u_h(0) &= \sum_{i=0}^m \left(c_i \phi_i(x) + d_i \psi_i(x) \right) = c_0 = 0 \\ u_h(1) &= \sum_{i=0}^m \left(c_i \phi_i(x) + d_i \psi_i(x) \right) = c_m = 0 \\ u'_h(0) &= \sum_{i=0}^m \left(c_i \phi'_i(x) + d_i \psi'_i(x) \right) = d_0 = 0 \\ u'_h(1) &= \sum_{i=0}^m \left(c_i \phi'_i(x) + d_i \psi'_i(x) \right) = d_m = 0 \end{aligned}$$

Ou seja, a matriz bloco $\widehat{\mathbf{K}}$ do sistema linear deve ser modificada de forma a garantir que a solução seja da forma, $\widehat{\mathbf{C}} = (0, c_1, c_2, \dots, c_{m-1}, 0, 0, d_1, d_2, \dots, d_{m-1}, 0)$, isto significa que o sistema passa a ter 4 incógnitas a menos, ou seja, o sistema é de ordem $2(m-1)$. Para outros tipos de fronteira e valores diferentes de zero, o procedimento é similar ao da Seção 2.7.

Portanto, usando a base de Hermite, o valor da função e sua primeira derivada no nó x_i são os coeficiente de interpolação da função base $\phi_i(x)$ e $\psi_i(x)$ respectivamente, ou seja $u_h(x_i) = c_i$ e $u'_h(x_i) = d_i$.

Matriz Local

Usando os mesmos argumentos anteriores, no intervalo $[x_1^e, x_2^e]$, definimos a função base local por (ver Fig. 3.3)

$$\begin{cases} \varphi_1^e(x) = \left(\frac{x - x_1^e}{h} - 1\right)^2 \left(\frac{2(x - x_1^e)}{h} + 1\right), \\ \varphi_2^e(x) = \left(\frac{x_2^e - x}{h} - 1\right)^2 \left(\frac{2(x_2^e - x)}{h} + 1\right), \\ \psi_1^e(x) = (x - x_1^e) \left(\frac{x - x_1^e}{h} - 1\right)^2, \\ \psi_2^e(x) = (x - x_2^e) \left(\frac{x_2^e - x}{h} - 1\right)^2. \end{cases} \quad (3.27)$$

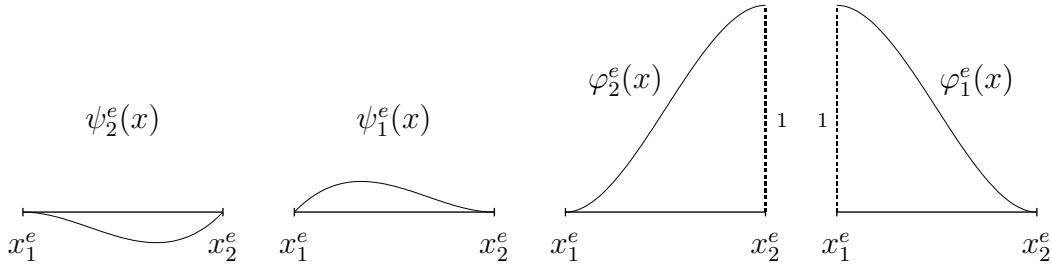


Figura 3.3: Função base local: Hermite

Então em cada elemento e , a matriz local K_{ab}^e de ordem 4×4 e vetor força local f_a^e de 4×1 podem ser calculados por

$$K_{ab}^e = \int_{x_1^e}^{x_2^e} \left(\frac{d^2 \varphi_a^e}{dx^2} \frac{d^2 \varphi_b^e}{dx^2} \right) dx, \quad 1 \leq a, b \leq 4.$$

$$f_a^e = \int_{x_1^e}^{x_2^e} f \varphi_a^e dx. \quad 1 \leq a \leq 4$$

As quantidades globais K_{ij} e f_i podem ser obtidas somando as contribuições locais de todos elementos nos nós correspondentes, da maneira semelhante.

3.2 Análise de Erro do Problema Estacionário

Nesta seção estabeleceremos a convergência da solução aproximada obtida pelo método de elementos finitos para a solução exata. A convergência depende do grau do polinômio interpolador de u_h no interior de cada elemento Ω_e e do método de aproximação utilizado.

Considere, sem perda de generalidade, Ω um conjunto aberto e limitado do \mathbb{R}^2 , V um espaço de Hilbert real com norma $\|\cdot\|_V$ induzida pelo produto interno $(\cdot, \cdot)^{1/2}$. Seja $f : V \rightarrow \mathbb{R}$ um funcional linear e contínuo em V e uma aplicação $a : V \times V \rightarrow \mathbb{R}$. Assumiremos que a forma variacional abstrata ou forma fraca do problema seja dado da seguinte forma:

Encontrar $u \in V$ tal que

$$a(u, v) = (f, v) + (p, v)_\Gamma \quad \forall v \in V. \quad (3.28)$$

Considerando o método de Galerkin, temos a seguinte forma variacional aproximada:

Encontrar $u_h \in V_m$ tal que

$$a(u_h, v_h) = (f, v_h) + (p, v_h)_\Gamma \quad \forall v_h \in V_m, \quad (3.29)$$

onde $V_m \subset V$ é um subespaço de dimensão finita gerado pelo conjunto de m vetores da base de V .

Queremos estimar o erro $\|u - u_h\|_V$, entre a solução exata u e a solução aproximada u_h , ou seja $\lim_{h \rightarrow 0} \|u - u_h\|_V \rightarrow 0$. No método de elementos finitos as funções u_h são interpolantes locais que em geral, são polinômios definidos em subdomínios Ω_e do domínio Ω . A escolha do subespaço V_m é fundamental para a melhor aproximação da solução. Definimos o espaço de elementos finitos $V_m^k(\Omega) \cap C^0(\Omega)$ de grau k na variável espacial, onde

$$V_m^k(\Omega) = \left\{ v_h \in V; v_h^e \in P_k(\Omega_e) \right\}, \quad (3.30)$$

e 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 graus menores ou iguais a k nas variáveis x_1 e x_2 . Os elementos Ω_e satisfazem a condição

$$\Omega = \left(\bigcup_{e=1}^{\text{Nel}} \overline{\Omega_e} \right)^\circ \quad \text{e} \quad \Omega_e \cap \Omega_k = \emptyset, \quad \text{se} \quad e \neq k,$$

onde N_{el} é o número total de elementos.

Considere as seguintes hipóteses:

(H1) $a(\cdot, \cdot), (\cdot, \cdot)$ e $(\cdot, \cdot)_\Gamma$ são bilineares em V , i.e., linear em cada argumento.

(H2) $a(\cdot, \cdot)$ é contínua e coerciva em V , i.e., existem constantes positivas δ_1 e δ_2 tais que:

$$\begin{aligned} \|a(u, v)\| &\leq \delta_1 \|u\|_V \|v\|_V & \forall u, v \in V, \\ a(v, v) &\geq \delta_2 \|v\|_V^2 & \forall v \in V. \end{aligned} \quad (3.31)$$

Segue então o seguinte teorema,

Teorema 3.1. (Lema de Céa). *Seja $e = u - u_h$, denotando o erro absoluto da aproximação do método de elementos finitos. Então sob as hipóteses (H1) e (H2), existe uma constante δ , independente do subespaço V_m tal que,*

1. $a(e, v_h) = a(u - u_h, v_h) = 0$,
2. $\|u - u_h\|_V \leq \delta \|u - v_h\|_V$,

para todo $v_h \in V_m$, onde $\delta = \delta_1/\delta_2$.

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 $u \in V$, $\lim_{h \rightarrow 0} \inf_{v_h \in V_m} \|u - v_h\|_V = 0$.

Demonstração:

1. Como $V_m \subset V$ então a solução exata (3.28) pode ser escrita por

$$a(u, v_h) = (f, v_h) + (p, v_h)_\Gamma \quad \forall v_h \in V_m. \quad (3.32)$$

Subtraindo (3.32) de (3.29) e usando a bilinearidade de $a(\cdot, \cdot)$ obtemos

$$a(u - u_h, v_h) = a(e, v_h) = 0 \quad \forall v_h \in V_m. \quad (3.33)$$

2. Como $a(\cdot, \cdot)$ é coerciva e bilinear temos

$$\begin{aligned} \delta_2 \|u - u_h\|_V^2 &\leq a(u - u_h, u - v_h + v_h - u_h) \\ &\leq a(u - u_h, u - v_h) + a(u - u_h, v_h - u_h) \\ &= a(u - u_h, u - v_h), \end{aligned}$$

onde usamos a propriedade de ortogonalidade (3.33). Da continuidade (3.31) obtemos

$$\|u - u_h\|_V \leq \delta \|u - v_h\|_V, \quad \forall v_h \in V_m. \quad (3.34)$$

onde $\delta = \delta_1/\delta_2 > 0$. \square

Corolário 3.1. *Se, além das hipóteses (H1) e (H2), a forma $a(\cdot, \cdot)$ é simétrica em V , ou seja, $a(u, v) = a(v, u)$, $\forall u, v \in V$, então,*

$$\|u - u_h\|_V \leq \sqrt{\delta} \|u - v_h\|_V \quad \forall v_h \in V_m. \quad (3.35)$$

Demonstração:

Considerando a identidade e a simetria de $a(\cdot, \cdot)$ temos

$$\begin{aligned} a(u - v_h, u - v_h) &= a(u - u_h + u_h - v_h, u - u_h + u_h - v_h) \\ &= a(u - u_h, u - u_h) + 2a(u - u_h, u_h - v_h) + a(u_h - v_h, u_h - v_h). \end{aligned}$$

Usando a coercividade (3.31) e ortogonalidade (3.33) obtém-se

$$a(u - u_h, u - u_h) \leq a(u - v_h, u - v_h) \quad \forall v_h \in V_m.$$

Da continuidade e coercividade (3.31),

$$\delta_2 \|u - u_h\|_V^2 \leq \delta_1 \|u - v_h\|_V^2,$$

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 do que a obtida por (3.34). \square

Essencialmente o Lema de Céa, mostra que o erro $\|u - u_h\|$ se reduz ao cálculo da distância $d(u, V_m) = \inf_{v_h \in V_m} \|u - v_h\|$, entre a função $u \in V$ e o subespaço $V_m \subset V$.

Corolário 3.2. *Nas mesmas condições do Corolário 3.1, temos*

$$\|u_h\|_V \leq \sqrt{\delta} \|u\|_V. \quad (3.36)$$

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 (3.31) obtemos

$$\|u_h\|_V \leq \sqrt{\delta} \|u\|_V. \quad \square$$

Os resultados anteriores nos permite 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, queremos determinar uma estimativa de erro da convergência de u_h para u quando $h \rightarrow 0$, em normas de espaços de Sobolev, ou seja, estimar $\|u - u_h\|_1$ e $\|u - u_h\|_0$, quando $h \rightarrow 0$. Antes porém, são necessários fazer estimativas de erro para a função de interpolação.

3.2.1 Erro de Interpolação

A seguir provaremos esses resultados para o problema (2.6), ou seja,

$$a(u, v) = (f, v), \quad \forall v \in V \quad (3.37)$$

onde

$$a(u, v) = \alpha \int_0^1 u_x v_x dx + \beta \int_0^1 uv dx, \quad \forall v \in V \quad (3.38)$$

e

$$(f, v) = \int_0^1 f v dx, \quad \forall v \in V \quad (3.39)$$

Definição 1. Denomina-se por interpolante ao polinômio interpolador de grau k , denotado por $\tilde{u}(x)$ que interpola a solução exata u nos pontos nodais $\{x_i\}$, i.e, $u(x_i) = \tilde{u}(x_i)$ e em cada intervalo, $I_i = [x_i, x_{i+1}]$, $i = 0, 1, \dots, m$, $\tilde{u}(x)$ é um polinômio de grau k .

No problema aproximado, definimos o subespaço V_m gerado pelas funções lineares por partes (2.19) e desta forma uma maneira natural é escolher a interpolante tendo como base a função $\varphi_i(x)$, ou seja,

$$\tilde{u}(x) = \sum_{i=1}^m u(x_i) \varphi_i(x),$$

onde $u = u(x_i)$ é a solução exata nos pontos nodais. Sendo $\varphi_i(x)$ uma base de V_m então $\tilde{u} \in V_m$, $\tilde{u}(x_i) = u(x_i)$ e \tilde{u} é um polinômio de grau $k = 1$, em cada elemento finito I_i . Assim \tilde{u} é um polinômio interpolador. O interpolante \tilde{u} não deve ser confundido com a solução aproximada u_h obtida variacionalmente pelo método de Galerkin.

Inicialmente vamos estabelecer uma estimativa de erro para o termo $\|u - \tilde{u}\|_1$.

Teorema 3.2. *Se $f \in L^2(0, 1)$ então a interpolante \tilde{u} converge para a solução $u \in H_0^1(0, 1) \cap H^2(0, 1)$. do problema (2.6), na norma $H^1(0, 1)$, i.e.,*

$$\|u - \tilde{u}\|_1 = \|u - \tilde{u}\|_0 + \|u' - \tilde{u}'\|_0 \leq \frac{h}{\pi} \left(1 + \frac{h^2}{\pi^2}\right)^{\frac{1}{2}} \|u\|_2, \quad (3.40)$$

onde estamos denotando a derivada u_x por u' , \tilde{u}_i é a restrição de \tilde{u} em cada elemento I_i e $\|\cdot\|_{0, I_i}$ denota a norma de $L^2(I_i)$, isto é,

$$\|v\|_{0, I_i}^2 = \int_{x_i}^{x_{i+1}} \|v(s)\|^2 ds$$

Demonstração: Tem se que:

$$\|u - \tilde{u}\|_1^2 = \sum_{i=1}^m (\|u - \tilde{u}_i\|_{0, I_i}^2 + \|u' - \tilde{u}'_i\|_{0, I_i}^2)$$

Usaremos no que segue o fato de que a solução u do problema (3.37), para $f \in L^2(0, 1)$, pertence ao espaço $V = H_0^1(0, 1) \cap H^2(0, 1)$.

Seja $\xi = u - \tilde{u}$, tal que, $\xi_i = \xi|_{I_i}$. Desde que \tilde{u}_i é uma interpolante de u em I_i , então nos pontos nodais x_i e x_{i+1} temos que, $\xi_i(x_i) = \xi_i(x_{i+1}) = 0$. Desta forma podemos considerar a expansão ξ_i em série de senos de Fourier, i.e.,

$$\xi_i(x) = \sum_{m=1}^{\infty} c_{mi} \sin \frac{m\pi(x - x_i)}{h}, \quad \forall c_{mi} \in \mathbb{R}$$

Fazendo a troca de variável $y = x - x_i$, obtemos que

$$\|\xi_i\|_{0, I_i}^2 = \int_{x_i}^{x_{i+1}} \xi_i(x)^2 dx = \int_0^h \xi_i(y)^2 dy = \frac{h}{2} \sum_{m=1}^{\infty} (c_{mi})^2,$$

onde o termo c_{mi} é denominado, coeficiente de Fourier. A última igualdade é conhecida como identidade de Parseval, que é uma consequência da seguinte ortogonalidade:

$$\int_0^h \cos \frac{m\pi y}{h} \cos \frac{n\pi y}{h} dy = \int_0^h \sin \frac{m\pi y}{h} \sin \frac{n\pi y}{h} dy = \frac{h}{2} \delta_{mn}, \quad \forall m, n \in \mathbb{Z}.$$

Temos também que

$$\xi'_i(x) = \xi'_i(y) = \sum_{m=1}^{\infty} c_{mi} \frac{m\pi}{h} \cos \frac{m\pi y}{h}, \quad \forall c_{mi} \in \mathbb{R}$$

Logo, usando ortogonalidade anterior obtemos,

$$\|\xi'_i\|_{0,I_i}^2 = \int_{x_i}^{x_{i+1}} \xi'_i(x)^2 dx = \frac{h}{2} \sum_{m=1}^{\infty} (c_{mi})^2 \frac{m^2 \pi^2}{h^2}$$

e de forma análoga;

$$\|\xi''_i\|_{0,I_i}^2 = \int_{x_i}^{x_{i+1}} \xi''_i(x)^2 dx = \frac{h}{2} \sum_{m=1}^{\infty} (c_{mi})^2 \frac{m^4 \pi^4}{h^4}$$

Note que, sendo a função interpolante \tilde{u}_i linear em cada intervalo $I_i = [x_i, x_{i+1}]$, temos $\tilde{u}''_i = 0$ e então $\xi''_i(x) = u''(x)$, $\forall x \in I_i$. Logo, como $m \geq 1$, obtém-se

$$\|\xi'_i\|_{0,I_i}^2 \leq \sum_{m=1}^{\infty} (c_{mi})^2 \frac{m^4 \pi^2}{h^2} = \frac{h^2}{\pi^2} \|u''\|_{0,I_i}^2, \quad i = 1, 2, \dots, m.$$

Fazendo o somatório de $i = 1$ até $i = m$, obtemos

$$\sum_{i=1}^m \|\xi'_i\|_0^2 = \|u' - \tilde{u}'\|_0^2 \leq \frac{h^2}{\pi^2} \|u''\|_0^2$$

Extraindo a raiz quadrada, obtemos a estimativa de erro para a derivada ξ'_i na norma $L^2(0, 1)$,

$$\|\xi'\|_0 = \|u' - \tilde{u}'\|_0 \leq \frac{h}{\pi} \|u''\|_0 \quad (3.41)$$

Diz-se neste caso que o erro é de ordem h e denota-se por $\mathcal{O}(h)$.

De forma análoga, temos

$$\|\xi_i\|_{0,I_i}^2 = \frac{h}{2} \sum_{m=1}^{\infty} (c_{mi})^2 \leq \frac{h}{2} \sum_{m=1}^{\infty} (c_{mi})^2 m^4 = \frac{h^4}{\pi^4} \|u''\|_{0,I_i}^2, \quad i = 1, 2, \dots, m.$$

Logo, somando de $i = 1$ até $i = m$, obtemos

$$\sum_{i=1}^m \|\xi_i\|_0^2 = \|u - \tilde{u}\|_0^2 \leq \frac{h^4}{\pi^4} \|u''\|_0^2$$

Extraindo a raiz quadrada, obtemos a estimativa de erro para ξ_i na norma $L^2(0, 1)$,

$$\|\xi\|_0 = \|u - \tilde{u}\|_0 \leq \frac{h^2}{\pi^2} \|u''\|_0 \quad (3.42)$$

Diz-se neste caso que o erro de interpolação é de ordem h^2 e denota-se por $\mathcal{O}(h^2)$.

Das desigualdades (3.41) e (3.42), conclui-se a estimativa de erro na norma $H^1(0, 1)$,

$$\|\xi\|_1^2 = \|u - \tilde{u}\|_1^2 = \|u - \tilde{u}\|_0^2 + \|u' - \tilde{u}'\|_0^2 \leq \left(\frac{h^4}{\pi^4} + \frac{h^2}{\pi^2} \right) \|u''\|_0^2$$

Ou seja

$$\|\xi\|_1 = \|u - \tilde{u}\|_1 = \|u - \tilde{u}\|_0 + \|u' - \tilde{u}'\|_0 \leq \frac{h}{\pi} \left(1 + \frac{h^2}{\pi^2}\right)^{\frac{1}{2}} \|u\|_2, \quad (3.43)$$

onde $\|\cdot\|_2$ denota a norma de $H^2(0, 1)$ e por definição, $\|u\|_2 \geq \|u''\|_0$. \square

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 (3.41).

A seguir serão demonstradas 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 a solução u_h , projeção ortogonal de u com respeito a forma $a(\cdot, \cdot)$ no espaço V_m satisfaz a desigualdade (3.34), dada por,

$$\|u - u_h\|_V \leq \delta \|u - v_h\|_V, \quad \forall v_h \in V_m.$$

onde $\delta = \delta_1/\delta_2$, com $\delta_1 = \max\{\alpha, \beta\}$, $\delta_2 = \min\{\alpha, \beta\}$ e estamos denotando $V = H_0^1(0, 1) \cap H^2(0, 1)$. Tomando em particular $v_h = \tilde{u} \in V_m$, obtemos a estimativa de erro na norma $H^1(0, 1)$,

$$\|u - u_h\|_1 \leq \delta \|u - \tilde{u}\|_1 \leq \frac{\delta h}{\pi} \left(1 + \frac{h^2}{\pi^2}\right)^{\frac{1}{2}} \|u\|_2, \quad (3.44)$$

com erro de ordem $\mathcal{O}(h)$. Por outro lado, usando (2.7) obtemos a seguinte estimativa em termos da função f , dada por

$$\|u - u_h\|_1 \leq \delta \|u - \tilde{u}\|_1 \leq \frac{c\delta h}{\pi} \left(1 + \frac{h^2}{\pi^2}\right)^{\frac{1}{2}} \|f\|_0, \quad (3.45)$$

com erro de ordem $\mathcal{O}(h)$.

Quando a forma bilinear $a(\cdot, \cdot)$ é simétrica, então de (3.35) obtemos a mesma estimativa, substituindo-se δ por $\sqrt{\delta}$.

3.2.3 Erro na Norma $L^2(\Omega)$

A desigualdade (3.42) mostra que a estimativa do erro de interpolação na norma L^2 é de ordem $\mathcal{O}(h^2)$. Mas ao calcular a estimativa de erro de aproximação na norma

$L^2(\Omega)$, $\|u - u_h\|_0$, a única relação que temos entre o erro de interpolação e o erro de aproximação é dada na norma $H^1(\Omega)$, ou seja

$$\|u - u_h\|_1 \leq \delta \|u - \tilde{u}\|_1 \leq \frac{\delta h}{\pi} \left(1 + \frac{h^2}{\pi^2}\right)^{\frac{1}{2}} \|u\|_2, \quad (3.46)$$

Neste caso, como $H_0^1(\Omega)$ esta imerso continuamente em $L^2(\Omega)$, obtemos

$$\|u - u_h\|_0 \leq \|u - u_h\|_1 \leq \delta \|u - \tilde{u}\|_1 \leq \frac{\delta h}{\pi} \left(1 + \frac{h^2}{\pi^2}\right)^{\frac{1}{2}} \|u\|_2, \quad (3.47)$$

Portanto estamos perdendo a ordem quadrática do erro de interpolação, obtendo-se somente a convergência linear do erro $\|u - u_h\|_0$. Uma outra forma é usar a equivalência das normas $\|u\|_1$ e $\|u'\|_0$ no espaço $H_0^1(\Omega)$. Mas neste caso, a estimativa de erro da derivada da função de interpolação na norma $L^2(\Omega)$, dada por (3.41), também é de ordem linear. De qualquer forma, obtemos somente uma estimativa de erro linear para a norma $L^2(\Omega)$. Em razão da estimativa (3.42), gostaríamos de obter um ordem de convergência quadrática. Para isso vamos usar um argumento devido a Nitsche, para mostrar que o erro em $L^2(\Omega)$ é de ordem $\mathcal{O}(h^2)$.

Teorema 3.3. *Se $f \in L^2(0, 1)$ então a interpolante \tilde{u} converge para a solução $u \in H^2(0, 1) \cap H_0^1(0, 1)$ do problema (2.6), na norma $L^2(0, 1)$, i.e.,*

$$\|u - u_h\|_0 \leq \hat{c} h^2 \|f\|_0, \quad (3.48)$$

onde \hat{c} é uma constante positiva independente de h e f .

Demonstração:

Com efeito, considere o problema (3.37), substituindo a força $f \in L^2(0, 1)$ pela diferença $(u - u_h)$ que também pertence ao espaço $L^2(0, 1)$. Então

$$a(w, v) = (u - u_h, v), \quad \forall v \in H_0^1(0, 1) \subset L^2(0, 1),$$

onde w é a solução a ser determinada. Como a igualdade é verdadeira para todo v , em particular podemos tomar $v = u - u_h$, obtendo-se

$$a(w, u - u_h) = (u - u_h, u - u_h) = \|u - u_h\|_0^2$$

Considere agora a solução aproximada w_h de w em V_m . Então pela condição de ortogonalidade, ver Teorema 3.1, temos que

$$a(w_h, u - u_h) = 0$$

Combinando os dois resultados, obtemos que

$$a(w - w_h, u - u_h) = \|u - u_h\|_0^2$$

Usando a desigualdade de Schwarz, obtemos

$$\|u - u_h\|_0^2 \leq \|u - u_h\|_1 \|w - w_h\|_1 \quad (3.49)$$

Considerando $f = u - u_h$ e também $f = w - w_h$ na estimativa (3.45), temos

$$\begin{aligned} \|u - u_h\|_1 &\leq Ch\|f\|_0 = Ch\|u - u_h\|_0 \\ \|w - w_h\|_1 &\leq Ch\|f\|_0 = Ch\|w - w_h\|_0 \end{aligned}$$

Substituindo na desigualdade (3.49) resulta que,

$$\|u - u_h\|_0^2 \leq (Ch)^2 \|u - u_h\|_0 \|w - w_h\|_0,$$

que é equivalente a,

$$\|u - u_h\|_0 \leq (Ch)^2 \|f\|_0 \quad (3.50)$$

obtendo-se assim o resultado. \square

Diz-se neste caso que a ordem de convergência é quadrática, ou seja, de ordem $\mathcal{O}(h^2)$.

Note que, numericamente podemos ter problemas de convergência, quando α é muito pequeno, ou seja, considerando em particular $\alpha = \varepsilon > 0$ e $\beta = 1$, então de (3.45) 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.

No que segue, mostraremos que a solução aproximada u_h converge em norma para a solução exata u quando $h \rightarrow 0$.

Teorema 3.4. *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.,*

$$\lim_{h \rightarrow 0} \|u - u_h\|_1 = 0 \quad e \quad \lim_{h \rightarrow 0} \|u - u_h\|_0 = 0.$$

Demonstração: A demonstração é uma consequência imediata da estimativas (3.45) e (3.48).

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 $\mathcal{O}(h^2)$. Mostraremos a seguir que podemos melhorar a ordem de convergência se a solução exata pertence a um espaço mais regular que o espaço $H^2(0, 1) \cap H_0^1(0, 1)$. Nesta seção enunciaremos alguns resultados mais gerais de estimativa de erro em normas de espaço de Sobolev, válidos para Ω aberto, regular e limitado do \mathbb{R}^n $n \leq 3$, que depende da regularidade da solução exata e do grau do polinômio interpolador definido no espaço V_m^k , cuja demonstração pode ser encontrada, em [5, 16, 24].

Teorema 3.5. *Seja $u \in H^m$, $m \geq 0$ então existe uma função interpolante $\tilde{u} \in V_m^k$, tal que para todo s , $0 \leq s \leq r$,*

$$\|u - \tilde{u}\|_s \leq ch^\alpha \|u\|_m, \quad (3.51)$$

onde $\alpha = \min\{k+1-s, m-s\}$, k é o grau do polinômio interpolador, c é uma constante independente de u e $h = \max\{h^e\}$.

Corolário 3.3. (Estimativa Ótima)

Seja $u \in H^{k+1}$ e $\tilde{u} \in V_m^k$ então,

$$\|u - \tilde{u}\|_m \leq ch^{k+1-m} \|u\|_{k+1}, \quad (3.52)$$

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 (3.3) é que a estimativa depende da regularidade da solução, isto significa, por exemplo, que se a solução u do problema pertence ao espaço $H^2(\Omega)$ então a melhor estimativa a ser obtida é de ordem $\mathcal{O}(h)$ e $\mathcal{O}(h^2)$ nas normas do $H^1(\Omega)$ e $L^2(\Omega)$, independente do grau k do polinômio interpolador.

Pode-se mostrar que para o problema (2.1), é válido o seguinte resultado de regularidade (ver [2]): Para $f \in H^m(0, 1)$ a solução $u \in H^{m+2}(0, 1)$ e além disso

$$\|u\|_{m+2} \leq \|f\|_m.$$

onde $m \geq 1$ é um inteiro.

Nestas condições 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 seguintes estimativas em $L^2(0, 1) = H^0(0, 1)$ e $H^1(0, 1)$:

$$\begin{aligned} \|u - \tilde{u}\|_0 &\leq ch^2 \|u\|_2, \quad \|u - \tilde{u}\|_0 \leq ch^3 \|u\|_3, \quad \|u - \tilde{u}\|_0 \leq ch^4 \|u\|_4, \\ \|u - \tilde{u}\|_1 &\leq ch^1 \|u\|_2, \quad \|u - \tilde{u}\|_1 \leq ch^2 \|u\|_3, \quad \|u - \tilde{u}\|_1 \leq ch^3 \|u\|_4. \end{aligned}$$

Para a conclusão da análise do erro precisamos estabelecer uma relação entre o erro $\|e\|_m = \|u - u_h\|_m$ e $\|u - \tilde{u}\|_m$, onde u é a solução exata, u_h é a solução aproximada e \tilde{u} é a função interpolante. Para isto é suficiente tomar $\tilde{u} = v_h \in V_m$ no Teorema (3.1), obtendo-se:

$$\|u - u_h\|_m \leq \delta \|u - \tilde{u}\|_m. \quad (3.53)$$

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 (3.52) e (3.53) obtemos:

$$\|e\|_m = \|u - u_h\|_m \leq \delta \|u - \tilde{u}\|_m \leq Ch^{k+1-m} \|u\|_{k+1}. \quad (3.54)$$

onde $C = c\delta$.

3.3 Erro Numérico

Os resultados de convergência obtidos, são as estimativas de erro esperado entre a solução exata e a solução numérica aproximada. Entretanto, a solução exata não é conhecida em geral, e o propósito dessa seção é introduzir um dos procedimentos numéricos para se certificar dos resultados numéricos obtidos fazendo um paralelo aos resultados teóricos de estimativa de erro.

Como em geral, não conhecemos a solução exata u então considere em seu lugar a solução numérica u_N com N suficientemente grande (ou equivalentemente h muito pequeno), onde N é o número de nós da discretização, ou seja, u_N será entendida como a "solução exata".

Construiremos uma sucessão de soluções numéricas $\{u_i\}$ associada a diversos tamanhos de malha h_i com os respectivos erros $\|\hat{E}_i\|_m = \|u_i - u_N\|_m$, para $i = 1, 2, \dots, (N-1)$.

Observe então a estimativa (3.54) associada a solução u_N

$$\|\hat{E}_i\|_m = \|u_i - u_N\|_m \leq c_1 h^{k+1-m} \|u_N\|_{k+1} \approx c_2 h_i^p, \quad \text{para } i = 1, 2, \dots, (N-1). \quad (3.55)$$

onde $p = (k+1-m)$ é a taxa de convergência. Note que, $\|u_N\|_{k+1}$ é um número real conhecido para algum k .

Assim podemos escrever

$$\alpha_i = \frac{\|\hat{E}_i\|_m}{\|\hat{E}_{i+1}\|_m} = \left(\frac{h_i}{h_{i+1}} \right)^p, \quad \forall i = 1, 2, \dots. \quad (3.56)$$

onde $h_{i+1} < h_i$. Através da relação (3.56), dizemos que p é a **taxa de convergência numérica**. Na teoria o valor de p é constante, mas na prática obteremos 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

$$p = \frac{\ln \alpha_i}{\ln \left(\frac{h_i}{h_{i+1}} \right)}, \quad \text{para } i = 1, 2, \dots. \quad (3.57)$$

Exemplo de construção da sucessão da solução numérica

Considere as soluções numéricas aproximadas $u_1, u_2, \dots, u_{N/2}$, obtidas usando o passo $h_i = (5 \times 2^{i+1})^{-1}$, ou seja, $h_i = 2h_{i+1}$, $i = 0, 1, \dots, (N-2)/2$. Considere \hat{E}_i o erro associado a uma norma, comparando as soluções u_i com u_N . Para essa sucessão temos, por exemplo:

$$p = \frac{\ln \alpha_i}{\ln \left(\frac{h_i}{h_{i+1}} \right)} = \frac{\ln \alpha_i}{\ln 2}, \quad \text{para } i = 0, 1, \dots, (N-2)/2 \quad (3.58)$$

3.4 Exercícios

1. Determine os elementos da matriz local do problema (3.14), 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 dos d_i , solução do sistema linear $Kd = F$. Considerando o polinômio interpolador quadrático, qual é a relação entre a solução aproximada $u_h(x_i)$ e o vetor solução d do sistema linear $Kd = F$.
4. Mostre que o conjunto de funções B-splines $\{\varphi_1, \varphi_2, \dots, \varphi_m, \varphi_{m+1}\}$ definidas em (3.9) formam um conjunto linearmente dependente.

Sugestão Mostre que o sistema $Ad = 0$ tem infinitas soluções, quando consideradas as duas funções no conjunto.

CAPÍTULO 4

Problema Estacionário Bidimensional

Neste capítulo desenvolveremos um programa utilizando o método de elementos finitos para a equação elíptica, caracterizada pelo problema do calor em estado de equilíbrio. São estudadas as formulações forte e fraca, as condições para existência e unicidade de solução para alguns tipo de condições de fronteira e sua influência na obtenção do sistema linear. São apresentados também o método de Crout para resolver o sistema linear, alguns exemplos numéricos com solução gráfica e os erros da solução aproximada utilizando diversos tamanhos de malha, bem como as condições para existência e unicidade de solução. 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 . Os elementos de Ω são denotados por x e o vetor normal unitário exterior a Γ é denotado por n . Dessa forma, $x = (x_1, x_2)$ e $n = (n_1, n_2) = (n_{x_1}, n_{x_2})$. Assumimos que a fronteira Γ admite a decomposição $\Gamma = \overline{\Gamma_q} \cup \overline{\Gamma_p}$, $\Gamma_q \cap \Gamma_p = \emptyset$. Denotamos por $q_i = q_i(x)$ o fluxo de calor e por $u = u(x)$ a temperatura. A fonte de calor por unidade de volume é denotada por $f = f(x)$. Assumimos que o fluxo de calor é dado pela lei de Fourier:

$$q_i = -Q_{ij} \frac{\partial u}{\partial x_j},$$

onde Q_{ij} é uma matriz simétrica definida positiva em todo $x \in \Omega$ e é denominada condutividade térmica. Quando o corpo é homogêneo a matriz Q_{ij} é constante para todo $x \in \Omega$. No caso isotrópico $Q_{ij}(x) = Q(x)\delta_{ij}$. Assumimos que a matriz Q_{ij} é definida positiva. A solução clássica do problema de calor é dada por:

(I) Formulação forte. Dadas as funções $f : \Omega \rightarrow \mathbb{R}$, $q : \Gamma_q \rightarrow \mathbb{R}$ e $p : \Gamma_p \rightarrow \mathbb{R}$, queremos encontrar $u : \overline{\Omega} \rightarrow \mathbb{R}$ tal que

$$\begin{cases} \frac{\partial}{\partial x_i} \left(-Q_{ij} \frac{\partial u}{\partial x_j} \right) = f, & \text{em } \Omega, \\ u = q, & \text{em } \Gamma_q, \\ -q_i n_i = Q_{ij} \frac{\partial u}{\partial x_j} n_i = p, & \text{em } \Gamma_p, \end{cases}$$

onde a função $q(x)$ é conhecida em Γ_q e a função $p(x)$ é conhecida em Γ_p . Além disso, considere $p(x)$ e $q(x)$ suficientemente regulares de tal forma que pelo menos a solução $u \in H^2(\Omega)$.

Quando o corpo é isotrópico e homogêneo a primeira equação do problema (I) pode ser escrita como

$$-\Delta u = f, \quad (4.1)$$

que é conhecida como equação de Poisson. Se $\Gamma = \Gamma_q$, então a condição de fronteira é conhecida como condição de Dirichlet. Se $\Gamma = \Gamma_p$, a condição de fronteira é conhecida como sendo do tipo Neumann. Se $\Gamma = \overline{\Gamma_p \cup \Gamma_q}$, $\Gamma_p \neq \emptyset$ e $\Gamma_q \neq \emptyset$ então a condição de fronteira é conhecida como do tipo misto.

Formulação Fraca

Sejam

$$H = \{u \in H^1(\Omega); u = q \text{ em } \Gamma_q\} \quad (4.2)$$

$$V = \{v \in H^1(\Omega); v = 0 \text{ em } \Gamma_q\}. \quad (4.3)$$

A formulação fraca é dada por

(II) Formulação fraca. Dados $f : \Omega \rightarrow \mathbb{R}$, $q : \Gamma_q \rightarrow \mathbb{R}$, e $p : \Gamma_p \rightarrow \mathbb{R}$, encontrar $u \in H$ tal que

$$\int_{\Omega} Q_{ij} \frac{\partial v}{\partial x_i} \frac{\partial u}{\partial x_j} = \int_{\Omega} v f \, d\Omega + \int_{\Gamma_p} v p \, d\Gamma, \quad \forall v \in V.$$

O próximo teorema estabelece uma relação entre as duas formulações:

Teorema 1 (Equivalência de Solução). Suponhamos que as funções sejam suficientemente regulares. Então os problemas (I) e (II) são equivalentes.

Demonstração:

(I) \Rightarrow (II): Seja $u \in H$ a solução do problema (I) e considere $v \in V$. Multiplicando a primeira equação de (I) por v , integrando em Ω e usando o **teorema da divergência**,

$$\int_{\Omega} v \frac{\partial q_i}{\partial x_i} d\Omega = - \int_{\Omega} \frac{\partial v}{\partial x_i} q_i d\Omega + \int_{\Gamma} v q_i n_i d\Gamma \quad (4.4)$$

conclui-se que

$$\int_{\Omega} \frac{\partial v}{\partial x_i} \left(Q_{ij} \frac{\partial u}{\partial x_j} \right) d\Omega = \int_{\Omega} v f d\Omega + \int_{\Gamma_p} v p d\Gamma$$

onde usamos que $q_i n_i = -Q_{ij} \frac{\partial u}{\partial x_j} n_i = p(x)$ em Γ_p e que $v = 0$ em Γ_q . Portanto, u é solução do problema (II).

(II) \Rightarrow (I): Seja $u \in H$, solução do problema (II) e $v \in V$. Do problema (II), temos

$$0 = \int_{\Omega} \frac{\partial v}{\partial x_i} q_i d\Omega + \int_{\Omega} v f d\Omega + \int_{\Gamma_p} v p d\Gamma.$$

Usando (4.4) no primeiro termo, obtemos

$$\int_{\Omega} \frac{\partial v}{\partial x_i} q_i d\Omega = - \int_{\Omega} v \frac{\partial q_i}{\partial x_i} d\Omega + \int_{\Gamma_p} v q_i n_i d\Gamma.$$

Logo,

$$0 = \int_{\Omega} v \left(-\frac{\partial q_i}{\partial x_i} + f \right) d\Omega + \int_{\Gamma_p} v (q_i n_i + p) d\Gamma$$

Definimos:

$$\begin{aligned} \alpha &= -\frac{\partial q_i}{\partial x_i} + f = -\frac{\partial}{\partial x_i} \left(Q_{ij} \frac{\partial u}{\partial x_j} \right) + f \quad \text{em } \Omega, \\ \beta &= q_i n_i + p \quad \text{em } \Gamma_p. \end{aligned}$$

Assim, $u \in H$ é solução do problema se $\alpha = 0$ e $\beta = 0$. De fato, seja $v = \alpha\phi$, onde

- i) $\phi > 0$ em Ω ,
- ii) $\phi = 0$ em Γ ,
- iii) ϕ é suficientemente regular.

Nestas condições, $v \in V = \{v \in H^1(\Omega); v = 0 \text{ em } \Gamma_q\}$. Além disso,

$$0 = \int_{\Omega} v \alpha d\Omega + \int_{\Gamma_p} v \beta d\Gamma = \int_{\Omega} \phi \alpha^2 d\Omega$$

Desde que $\phi > 0$ em Ω , então $\alpha = 0$ em Ω . Por outro lado, tomemos em particular, $v = \beta\varphi$, onde

- i) $\varphi > 0$ em Γ_p
- ii) $\varphi = 0$ em Γ_q
- iii) φ é suficientemente suave.

Nestas condições, $v \in V = \{v \in H^1(\Omega); v = 0 \text{ em } \Gamma_q\}$. Desde que $\alpha = 0$,

$$0 = \int_{\Gamma_p} v\beta \, d\Gamma = \int_{\Gamma_p} \varphi\beta^2 \, d\Gamma$$

Logo, necessariamente, $\beta = 0$ em Γ_p . Como $\alpha = 0$ e $\beta = 0$, então

$$-\frac{\partial}{\partial x_i} \left(Q_{ij} \frac{\partial u}{\partial x_j} \right) = f \quad \text{em } \Omega,$$

$$-Q_{ij} \frac{\partial u}{\partial x_j} n_i = p \quad \text{em } \Gamma_p.$$

Portanto u é solução do problema (I). Conclui-se, então, a equivalência de solução entre os problemas (I) e (II), se a solução é suficientemente regular.

Existência e Unicidade de Solução

Mostraremos a existência e a unicidade da solução fraca do problema II, quando os dados iniciais f , p e q são regulares. Considere

$$a(v, u) = \int_{\Omega} \frac{\partial v}{\partial x_i} Q_{ij} \frac{\partial u}{\partial x_j} \, d\Omega, \quad \forall v \in V \quad (4.5)$$

$$(v, f) = \int_{\Omega} v f \, d\Omega, \quad \forall v \in V \quad (4.6)$$

e

$$(v, p)_{\Gamma} = \int_{\Gamma_p} v p \, d\Gamma, \quad \forall v \in V \quad (4.7)$$

Dessa forma, a formulação fraca pode ser escrita por

$$a(v, u) = (v, f) + (v, p)_{\Gamma}, \quad \forall v \in V. \quad (4.8)$$

Por hipótese, a matriz condutividade Q_{ij} é simétrica e definida positiva.

Considere a norma do subespaço V de $H^1(\Omega)$ dada por

$$\|v\|_V^2 = \int_{\Omega} |\nabla v|^2 \, d\Omega.$$

Pode se mostrar que no subespaço V , as normas $\|\cdot\|_V$ e $\|\cdot\|_{H^1(\Omega)}$ são equivalentes. Mostraremos que a forma bilinear $a : V \times V \rightarrow R$ satisfaz as condições de Lax-Milgram:

i) $a(v, u) = a(u, v)$, pois Q_{ij} é simétrica.

ii) $|a(v, u)| = |((v, u))_V| \leq C\|v\|_V\|u\|_V$.

Assim, $\frac{|a(v, u)|}{\|u\|} \leq C\|v\|_V \Rightarrow a(\cdot, \cdot)$ é contínua em V .

iii) $|a(v, v)| = \int_{\Omega} \frac{\partial v}{\partial x_i} Q_{ij} \frac{\partial v}{\partial x_j} \geq C \int_{\Omega} \left| \frac{\partial v}{\partial x_i} \right|^2 = C\|v\|_V^2$,

pois, por hipótese, Q_{ij} é definida positiva.

Assim, $a(\cdot, \cdot)$ é coerciva em V . Por outro lado, a aplicação linear

$$v \in V \mapsto (v, f) + (v, p)_{\Gamma} \in \mathbb{R}$$

é contínua para $p \in H^1(\Omega)$ e $f \in L^2(\Omega)$.

De fato, do Teorema do Traço, temos que o traço de p , denotado por $(\gamma \circ p) \in L^2(\Gamma)$, satisfaz

$$\begin{aligned} \int_{\Omega} v f \, d\Omega + \int_{\Gamma_p} v p \, d\Gamma &\leq \|v\|_{L^2(\Omega)} \|f\|_{L^2(\Omega)} + \|\gamma \circ v\|_{L^2(\Gamma)} \|\gamma \circ p\|_{L^2(\Gamma)} \\ &\leq \|v\|_{H^1(\Omega)} (\|f\|_{L^2(\Omega)} + \|\gamma \circ p\|_{L^2(\Gamma)}) \leq C\|v\|_{H^1(\Omega)}. \end{aligned}$$

Como $a : V \times V \longrightarrow \mathbb{R}$ é uma forma bilinear, contínua e coerciva no espaço de $H^1(\Omega)$ e f uma função contínua, pelo Teorema de Lax-Milgram, existe uma única solução $u \in V$, quando $f \in L^2(\Omega)$, $p \in H^1(\Omega)$ e $q \in L^2(\Omega)$.

Formulação de Galerkin

Para usar o método de Galerkin, as funções u e v devem pertencer ao mesmo espaço. Sejam $H^h \subset H$, $V^h \subset V$ subespaços de dimensão finita dos espaços de Hilbert H e V . Definimos uma função

$$w^h(x) = u^h(x) - q^h(x), \quad (4.9)$$

onde $u^h \in H^h$ e $q^h \in H^h$ é uma função que satisfaz a mesma condição de fronteira de u , ou seja,

$$q^h(x) = \begin{cases} q(x), & \forall x \in \Gamma_q \\ 0, & \forall x \notin \Gamma_q \end{cases} \quad (4.10)$$

Nestas condições, temos

$$w^h(x) = u^h(x) - q^h(x) = q(x) - q(x) = 0, \quad \forall x \in \Gamma_q.$$

Assim $w^h \in V^h \subset V = \{v \in H^1(\Omega); v = 0 \text{ em } \Gamma_q\}$.

Definimos também a função

$$p^h(x) = \begin{cases} p(x), & \forall x \in \Gamma_p \\ 0, & \forall x \notin \Gamma_p \end{cases} \quad (4.11)$$

A formulação variacional é dada por

$$a(v, u) = (v, f) + (v, p)_\Gamma, \quad \forall v \in V. \quad (4.12)$$

As funções $v^h \in V^h$ são, em geral, funções lineares por partes. Assim, para domínios em geral, a fronteira $\partial\Omega$ de Ω pode não coincidir com a fronteira aproximada Γ_h . Restringindo a formulação (4.12) aos subespaços V^h e H^h , temos

$$a(v^h, u^h) = (v^h, f) + (v^h, p^h)_\Gamma, \quad \forall v^h \in V^h.$$

Por (4.9),

$$a(v^h, w^h) = (v^h, f) + (v^h, p^h)_\Gamma - a(v^h, q^h), \quad \forall v^h \in V^h. \quad (4.13)$$

Podemos então formular o seguinte problema:

Problema aproximado. Dadas as funções f , p e q , queremos determinar a função $w^h = u^h - q^h \in V^h$ solução da formulação (4.13).

Seja $\{\varphi_1, \varphi_2, \dots, \varphi_n\}$ uma base do subespaço V^h . Dessa forma, todo elemento $w^h \in V^h$ pode ser representado por

$$w^h = \sum_{j=1}^m C_j \varphi_j. \quad (4.14)$$

Substituindo em (4.13), tem-se

$$a(v^h, \sum_{j=1}^m C_j \varphi_j) = (v^h, f) + (v^h, p^h)_\Gamma - a(v^h, q^h), \quad \forall v^h \in V^h.$$

A igualdade permanece válida tomando em particular $v^h = \varphi_i$, ou seja,

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

Da linearidade da forma $a(\cdot, \cdot)$, segue que

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

Denotando

$$K_{ij} = a(\varphi_i, \varphi_j), \quad 1 \leq i, j \leq m \quad (4.15),$$

$$F_i = (\varphi_i, f) + (\varphi_i, p^h)_\Gamma - a(\varphi_i, q^h) \quad (4.16),$$

a formulação (4.13) pode ser escrita na forma matricial,

$$KC = F. \quad (4.17)$$

A solução do sistema linear de ordem $m \times m$, permitirá calcular a solução numérica $u^h \in V^h$ do problema aproximado. Como $u^h \in V^h$ é uma solução aproximada da solução $u \in H$, então se m é grande, melhor será a solução aproximada. Por outro lado, a matriz K do sistema linear aumenta quadraticamente, o que significa custo operacional. Neste aspecto, é então introduzido o Método de Elementos Finitos que, através de uma escolha conveniente da base $\{\varphi_1, \dots, \varphi_m\}$ de V^h , tornará a matriz K do sistema uma matriz banda e em geral de fácil resolução.

4.2 Discretização do Domínio

Considere uma partição do domínio Ω em subregiões Ω_e , de tal forma a satisfazer as condições:

$$\Omega = \left(\bigcup_{e=1}^{\text{Nel}} \overline{\Omega_e} \right)^\circ \quad \text{e} \quad \Omega_e \cap \Omega_k = \emptyset, \quad \text{se} \quad e \neq k,$$

onde Nel é o número total de elementos. Na partição do domínio definimos os nós globais A , $A = 1, 2, \dots, \text{Nno}$, onde Nno é o número total de nós da malha.

As subregiões, no caso bidimensional, geralmente são triângulos ou retângulos, consistindo de 3 ou 4 nós locais para cada elemento finito Ω_e .

Construção da Malha

Para a geração da malha, consideremos no domínio Ω um retângulo da forma $(a, b) \times (c, d)$. 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 subdividir os intervalos $[a, b]$ e $[c, d]$ e fazer o produto cartesiano. 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 considere o seguinte procedimento:

Define-se $h = (b - a)/\text{Nelx}$ e $k = (d - c)/\text{Nely}$, onde Nelx e Nely são os números de elementos nas direções x e y respectivamente. Logo,

$$\begin{aligned} x_i &= x_o + ih, & i &= 1, 2, \dots, \text{Nelx}, & \text{com } x_o &= a, \\ y_j &= y_o + jk, & j &= 1, 2, \dots, \text{Nely}, & \text{com } y_o &= c. \end{aligned}$$

Assim,

$$[a, b] \times [c, d] = \bigcup_{i=0}^{\text{Nelx}} [x_i, x_{i+1}] \times \bigcup_{j=0}^{\text{Nely}} [y_j, y_{j+1}]$$

A subrotina **DataInput** faz a geração da malha uniforme, onde em particular, tomamos $a = c = 0$ e $b = d = 1$. Dessa forma, $h = 1/\text{Nelx}$ e $k = 1/\text{Nely}$.

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 $(x[i], y[j])$, onde $i = 0, 1, \dots, \text{Nelx}$ e $j = 0, 1, \dots, \text{Nely}$.

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 $(\text{Nelx} + 1)$ e a posição j é o quociente da divisão de $(A - 1)$ por $(\text{Nelx} + 1)$, que serão representados na linguagem C por

$$(i, j) = \text{NoPos}(A),$$

onde

$$\begin{aligned} i &= (A - 1) \% (\text{Nelx} + 1), \\ j &= (A - 1) / (\text{Nelx} + 1). \end{aligned}$$

Por exemplo, considere a malha dada na Fig 4.1: Temos portanto 25 nós globais, 16 elementos (representados por um círculo) e $\text{Nelx} = \text{Nely} = 4$. 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 ,

$$A = \text{PosNo}(i, j),$$

através da relação:

$$A = j(\text{Nelx} + 1) + i + 1.$$

Assim, na posição $(1, 4)$ temos o nó $A = 22$, considerando a malha anterior.

21	22	23	24	25
16	17	18	19	20
11	12	13	14	15
6	7	8	9	10
1	2	3	4	5

Figura 4.1: Malha de elementos retangulares

Elemento \mapsto Posição: Subrotina ElmPos

Para cada elemento e , temos 4 nós globais. Para identificá-lo 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 pela relação:

Dado um elemento e , a sua posição (i, j) é dada pelo resto e o quociente da divisão. Temos

$$(i, j) = \text{ElmPos}(e),$$

onde

$$\begin{aligned} i &= (e - 1) \% \text{Nelx}, \\ j &= (e - 1) / \text{Nelx}. \end{aligned}$$

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

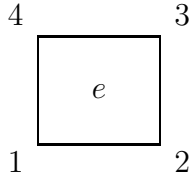
A subrotina **NoLG** identifica os nós locais $a = 1, 2, 3, 4$ de cada elemento (retângulo) com os nós globais A da malha,

$$A = \text{NoLG}(a, e)$$

da seguinte forma:

Dado um elemento e , o primeiro passo é identificar sua posição através da subrotina anterior, $(i, j) = \text{ElmPos}(e)$.

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,

$$\begin{aligned} \text{NoLG}(1, 10) &= 12, & \text{NoLG}(2, 10) &= 13, \\ \text{NoLG}(3, 10) &= 18, & \text{NoLG}(4, 10) &= 17. \end{aligned}$$

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 $\text{eqn}[A]$ no sistema e o número total de equações Neq do sistema. Assim, temos

$$I = \text{eqn}[A],$$

onde $I = 1, \dots, \text{Neq}$. Por conveniência, para os nós onde os valores são prescritos, tomamos $I = 0$. É claro que o número de equação é menor ou igual ao número de nós globais, $\text{Neq} \leq \text{Nno}$.

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
$\text{eqn}[A]$	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	
$\text{eqn}[A]$	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 $\text{Neq} = 19$. 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 $\text{Neq} = \text{Nno}$, 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:

$$\begin{aligned} \Gamma_1 &= \{(x, c) \in \partial\Omega; a \leq x \leq b\} \\ \Gamma_2 &= \{(b, y) \in \partial\Omega; c \leq y \leq d\} \\ \Gamma_3 &= \{(x, d) \in \partial\Omega; a \leq x \leq b\} \\ \Gamma_4 &= \{(a, y) \in \partial\Omega; c \leq y \leq d\} \end{aligned}$$

então precisamos identificar os nós pertencentes a cada uma das fronteiras Γ_i , independentes se são do tipo Neumann ou Dirichlet. A subrotina (**Fronteira**) tem essa função, além de quantificar os nós de cada fronteira pela expressão ($\text{Nbn}[i]$) "Número

de nós da fronteira $[i]$ ". Assim por exemplo, para a malha Fig. 4.1, temos os seguintes nós globais em cada fronteira:

$$\begin{aligned}\Gamma_1 &= \{1, 2, 3, 4, 5\} & \text{Nbn}[1] &= 5 \\ \Gamma_2 &= \{5, 10, 15, 20, 25\} & \text{Nbn}[2] &= 5 \\ \Gamma_3 &= \{21, 22, 23, 24, 25\} & \text{Nbn}[3] &= 5 \\ \Gamma_4 &= \{1, 6, 11, 16, 21\} & \text{Nbn}[4] &= 5\end{aligned}$$

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 as formas de contribuição de cada tipo de fronteira na obtenção da solução numérica do problema serão mostradas nas seções seguintes.

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:

$$\varphi_A(B) = \begin{cases} 1, & \text{se } A = B, \\ 0, & \text{se } A \neq B, \end{cases} \quad \forall B \in N.$$

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

$$w^h(x) = \sum_{B \in N - N_q} C_B \varphi_B(x). \quad (4.18)$$

De forma análoga, para a obtenção do sistema linear tem-se que:

$$\sum_{B \in N - N_q} a(\varphi_A, \varphi_B) C_B = (\varphi_A, f) + (\varphi_A, p^h)_\Gamma - a(\varphi_A, q^h), \quad \forall A \in N - N_q. \quad (4.19)$$

Variando os nós globais A em $N - Nq$ 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

$$I = \text{eqn}[A] \quad e \quad J = \text{eqn}[B],$$

onde $1 \leq I, J \leq \text{Neq}$. Então o sistema linear é definido por

$$\sum_{J=1}^{\text{Neq}} K_{IJ} C_J = F_I,$$

onde

$$K_{IJ} = a(\varphi_A, \varphi_B), \quad (4.20)$$

$$F_I = (\varphi_A, f) + (\varphi_A, p^h)_\Gamma - a(\varphi_A, q^h). \quad (4.21)$$

A matriz K é denominada matriz rigidez global do sistema linear.

4.3 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 todo $x \in \Gamma_q$ e $x \in \Gamma_p$ 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,

$$f^h(x) = \sum_{A \in N} \varphi_A(x) f_A, \quad (4.22)$$

$$q^h(x) = \sum_{A \in Nq} \varphi_A(x) q_A, \quad (4.23)$$

$$p^h(x) = \sum_{A \in Np} \varphi_A(x) p_A, \quad (4.24)$$

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,

$$f^h(x_A) = f_A, \quad q^h(x_A) = q_A \quad e \quad p^h(x_A) = p_A.$$

Substituindo em (4.21) obtém-se

$$F_I = \sum_{B \in N} (\varphi_A, \varphi_B) f_B + \sum_{B \in Np} (\varphi_A, \varphi_B) p_B - \sum_{B \in Nq} a(\varphi_A, \varphi_B) q_B.$$

Usando a definição dada em (4.6), (4.7) e substituindo em (4.21) obtém-se

$$F_I = \sum_{B \in N} \int_{\Omega} \varphi_A \varphi_B f_B d\Omega + \sum_{B \in N_p} \int_{\Gamma_p} \varphi_A \varphi_B p_B d\Gamma_p - \sum_{B \in N_q} K_{AB} q_B, \quad (4.25)$$

onde $I = \text{eqn}[A]$, para $1 \leq I \leq \text{Neq}$.

4.4 Propriedades da Matriz Rigidez

Analogamente, usando (4.5) e (4.20) temos

$$K_{IJ} = a(\varphi_A, \varphi_B) = \int_{\Omega} (\nabla \varphi_A)^T Q (\nabla \varphi_B) d\Omega, \quad (4.26)$$

onde $I = \text{eqn}[A]$ e $J = \text{eqn}[B]$. A matriz K_{IJ} tem seguintes propriedades:

Teorema 2. *A matriz K definida em (4.26) 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, fazendo com que a forma bilinear $a(\cdot, \cdot)$ também seja simétrica. Portanto K é simétrica. Por outro lado, seja $C = (C_1, C_2, \dots, C_{\text{Neq}})$, então,

$$\begin{aligned} C^T K C &= \sum_{I, J=1}^{\text{Neq}} C_I K_{IJ} C_J = \sum_{A, B \in N - N_q} a(\hat{C}_A \varphi_A, \hat{C}_B \varphi_B) \\ &= a(w^h, w^h) = \int_{\Omega} Q_{ij} \frac{\partial w^h}{\partial x_i} \frac{\partial w^h}{\partial x_j} d\Omega \geq 0, \end{aligned}$$

onde estamos usando a hipótese da matriz Q ser definida positiva e denotamos

$$w^h = \sum_{A, B \in N - N_q} \hat{C}_A \varphi_A.$$

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 $C = 0$. Portanto, 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.

Matriz Rigidez Local e Força Local

A formulação (4.26) e (4.25) para obtenção da matriz Rigidez K e do vetor Força F em todo o domínio Ω não é a mais conveniente. Um apropriado procedimento 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

$$K_{IJ}^e = a(\varphi_A^e, \varphi_B^e) = \int_{\Omega_e} (\nabla \varphi_A^e)^T Q (\nabla \varphi_B^e) d\Omega, \quad (4.27)$$

$$F_I^e = \sum_{B \in N} \int_{\Omega_e} \varphi_A^e \varphi_B^e f_B d\Omega + \sum_{B \in Np} \int_{\Gamma_p^e} \varphi_A^e \varphi_B^e p_B d\Gamma - \sum_{B \in Nq} K_{AB}^e q_B, \quad (4.28)$$

onde $I = \text{eqn}[A]$, $J = \text{eqn}[B]$ e $1 \leq I, J \leq \text{Neq}$. Logo a matriz global K e o vetor força F são obtidos por

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

A matriz K_{IJ}^e e o vetor Força F_I^e definidos em (4.27) e (4.28) têm ordem $\text{Neq} \times \text{Neq}$ e $\text{Neq} \times 1$. Mas a função de interpolação φ_A^e 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_{IJ}^e tem somente uma submatriz de ordem 4×4 cujos elementos são não necessariamente nulos e o vetor força F_I^e tem somente quatro coordenadas não necessariamente nulas. Considere, por exemplo um elemento e dado por

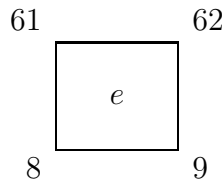


Figura 4.2: Um elemento e

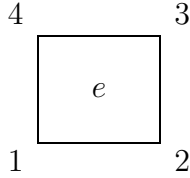
onde $A = \{8, 9, 61, 62\}$ são os nós globais. Suponhamos que a estes nós correspondem as equações $\text{eqn}(A) = \{5, 6, 47, 48\}$, onde esta correspondência é dada pela subrotina **EqNo**. Então os elementos da matriz e do vetor força $K_{5,5}^e, K_{5,6}^e, K_{5,47}^e, K_{5,48}^e; K_{6,6}^e, K_{6,47}^e, K_{6,48}^e; K_{47,47}^e, K_{47,48}^e, K_{48,48}^e$ e $F_5^e, F_6^e, F_{47}^e, F_{48}^e$ 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 $\varphi_A^e \equiv 0$ 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_{IJ}^e 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.

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

$$\varphi_a^e(\mathbf{x}_b^e) = \begin{cases} 1, & \text{se } a = b, \\ 0, & \text{se } a \neq b, \end{cases}$$

onde \mathbf{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

$$K_{ab}^e = a(\varphi_a^e, \varphi_b^e) = \int_{\Omega_e} (\nabla \varphi_a^e)^t Q (\nabla \varphi_b^e) d\Omega, \quad 1 \leq a, b \leq 4 \quad (4.29)$$

$$F_a^e = \sum_{b=1}^4 \int_{\Omega_e} \varphi_a^e \varphi_b^e f_b^e d\Omega + \sum_{b=1}^4 \int_{\Gamma_p^e} \varphi_a^e \varphi_b^e p_b^e d\Gamma - \sum_{b=1}^4 K_{ab}^e q_b^e. \quad (4.30)$$

Seja $B = \text{NoLG}(b, e)$. Então a primeira, segunda e terceira integrais são válidas para $B \in N - Nq$, $B \in Np$ e $B \in Nq$, 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

$$A = \text{NoLG}(a, e).$$

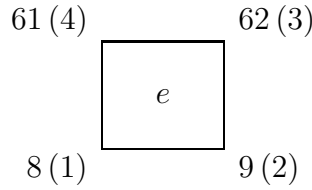
Além disso, é necessário relacionar o nó global A e a correspondente equação do sistema. Isto é feito pela subrotina **EqNo**, definida por

$$I = \text{eqn}[A], \quad 1 \leq I \leq \text{Neq}.$$

A composição entre as duas relações nos permite relacionar o nó local a com o número da equação I correspondente, através de

$$I = \text{eqn}[\text{NoLG}(a, e)] = \text{eqn}[A].$$

Considere o exemplo dado pela Fig. 4.2, com a introdução dos nós locais no sentido anti-horário.



Através de (4.29) e (4.30) calcula-se a matriz local K_{ab}^e e F_a^e dadas genericamente por

$$K_{ab}^e = \begin{bmatrix} K_{11}^e & K_{12}^e & K_{13}^e & K_{14}^e \\ & K_{22}^e & K_{23}^e & K_{24}^e \\ & & K_{33}^e & K_{34}^e \\ & & & K_{44}^e \end{bmatrix} \quad \text{e} \quad F_a^e = \begin{bmatrix} F_1^e \\ F_2^e \\ F_3^e \\ F_4^e \end{bmatrix}$$

K_{ab} é simétrica. Tem-se que

$$\begin{aligned} \text{eqn}[\text{NoLG}(1, e)] &= \text{eqn}[8] = 5 \\ \text{eqn}[\text{NoLG}(2, e)] &= \text{eqn}[9] = 6 \\ \text{eqn}[\text{NoLG}(3, e)] &= \text{eqn}[62] = 48 \\ \text{eqn}[\text{NoLG}(4, e)] &= \text{eqn}[61] = 47 \end{aligned}$$

Portanto, a contribuição do elemento e , na montagem da matriz global K e do vetor força são dados por

$$\begin{array}{llll}
K_{55} & \leftarrow & K_{55} & + & K_{11}^e \\
K_{56} & \leftarrow & K_{56} & + & K_{12}^e \\
K_{5\,48} & \leftarrow & K_{5\,48} & + & K_{13}^e \\
K_{5\,47} & \leftarrow & K_{5\,47} & + & K_{14}^e \\
K_{66} & \leftarrow & K_{66} & + & K_{22}^e \\
K_{6\,48} & \leftarrow & K_{6\,48} & + & K_{23}^e \\
K_{6\,47} & \leftarrow & K_{6\,47} & + & K_{24}^e \\
K_{48\,48} & \leftarrow & K_{48\,48} & + & K_{33}^e \\
K_{48\,47} & \leftarrow & K_{48\,47} & + & K_{34}^e \\
K_{47\,47} & \leftarrow & K_{47\,47} & + & K_{44}^e
\end{array}
\quad
\begin{array}{llll}
F_5 & \leftarrow & F_5 & + & F_1^e \\
F_6 & \leftarrow & F_6 & + & F_2^e \\
F_{48} & \leftarrow & F_{48} & + & F_3^e \\
F_{47} & \leftarrow & F_{47} & + & F_4^e
\end{array}$$

Variando $e = 1, 2, \dots, \text{Nel}$, a matriz global K e o vetor força F do sistema linear são obtidos por

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

onde K^e e F^e estão definidos em (4.29) e (4.30).

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.

4.5 Função de Interpolação

Consideremos um quadrilátero $\Omega_e \subset \Omega$ representado como abaixo:

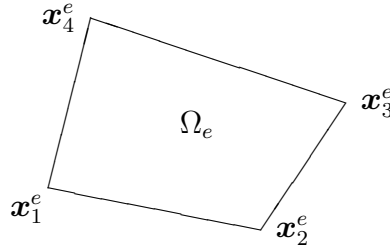


Figura 4.3: Um quadrilátero

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 $e = 1, 2, \dots$, \mathbf{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:

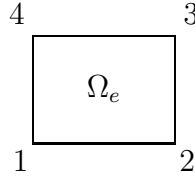
$$\varphi_a^e(\mathbf{x}_b^e) = \begin{cases} 1 & \text{se } a = b, \\ 0 & \text{se } a \neq b, \end{cases} \quad (4.31)$$

onde $1 \leq a, b \leq 4$ e $\mathbf{x}_b^e = (x_b^e, y_b^e)$.

Uma função bilinear em Ω_e é dada por

$$\varphi_a^e(x, y) = C_1 + C_2x + C_3y + C_4xy; \quad a = 1, 2, 3, 4,$$

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 $\varphi_1^e, \varphi_2^e, \varphi_3^e, \varphi_4^e$, interpoladores lineares satisfazendo (4.31). 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 $\mathbf{x}_1^e = (x_1^e, y_1^e)$, $\mathbf{x}_2^e = (x_2^e, y_1^e)$, $\mathbf{x}_3^e = (x_2^e, y_2^e)$, $\mathbf{x}_4^e = (x_1^e, y_2^e)$. A função de interpolação φ_a^e pode ser representada graficamente por

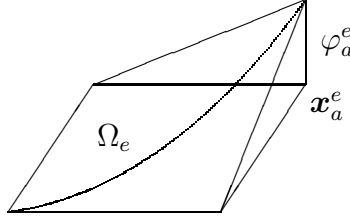


Figura 4.4: Função de interpolação

Transformação Isoparamétrica

Em particular, quando o domínio é o quadrado biunitário $[-1, 1] \times [-1, 1]$, a função de interpolação φ_a tem a forma simples, ou seja, denotando

$$\xi_1 = (-1, -1), \quad \xi_2 = (1, -1), \quad \xi_3 = (1, 1), \quad \xi_4 = (-1, 1).$$

temos

$$\varphi_1(\xi, \eta) = \frac{1}{4}(1 - \xi)(1 - \eta), \quad (4.32)$$

$$\varphi_2(\xi, \eta) = \frac{1}{4}(1 + \xi)(1 - \eta), \quad (4.33)$$

$$\varphi_3(\xi, \eta) = \frac{1}{4}(1 + \xi)(1 + \eta), \quad (4.34)$$

$$\varphi_4(\xi, \eta) = \frac{1}{4}(1 - \xi)(1 + \eta), \quad (4.35)$$

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 biunitário e principalmente por usar elementos finitos uniformes é conveniente fazer uma parametrização entre os elementos Ω_e e o quadrado biunitário, $[-1, 1] \times [-1, 1]$, denotado por Ω_b .

Considere a aplicação:

$$(\xi, \eta) \in \Omega_b \mapsto (x, y) \in \Omega_e$$

definida por

$$x(\xi, \eta) = \sum_{a=1}^4 \varphi_a(\xi, \eta) x_a^e, \quad (4.36)$$

$$y(\xi, \eta) = \sum_{a=1}^4 \varphi_a(\xi, \eta) y_a^e. \quad (4.37)$$

As funções (4.36) e (4.37) são biunívocas entre o quadrilátero Ω_e e o quadrado biunitário Ω_b . De fato, definindo

$$\begin{aligned} (\xi_1, \eta_1) &= (-1, -1) & (\xi_2, \eta_2) &= (1, -1) \\ (\xi_3, \eta_3) &= (1, 1) & (\xi_4, \eta_4) &= (-1, 1) \end{aligned} \quad (4.38)$$

então,

$$\begin{aligned} x(\xi_b, \eta_b) &= \sum_{a=1}^4 \varphi_a(\xi_b, \eta_b) x_a^e = \begin{cases} x_a^e, & \text{se } a = b, \\ 0, & \text{se } a \neq b. \end{cases} \\ y(\xi_b, \eta_b) &= \sum_{a=1}^4 \varphi_a(\xi_b, \eta_b) y_a^e = \begin{cases} y_a^e, & \text{se } a = b, \\ 0, & \text{se } a \neq b. \end{cases} \end{aligned}$$

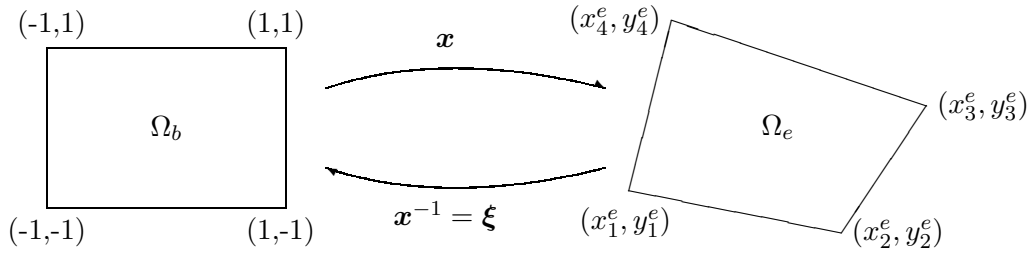


Figura 4.5: Transformação isoparamétrica

Na Fig. 4.5, denotamos $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}) = (x, y)$ e $\boldsymbol{\xi} = \boldsymbol{\xi}(\mathbf{x}) = (\xi, \eta)$.

Com a notação vetorial, as aplicações (4.36) e (4.37) podem ser dadas por

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{a=1}^4 \varphi_a(\boldsymbol{\xi}) \mathbf{x}_a^e. \quad (4.39)$$

Para verificar a existência da função

$$\boldsymbol{\xi} = \mathbf{x}^{-1} : \Omega_e \rightarrow \Omega_b$$

usaremos o Teorema da Função Inversa. A função φ_a é diferenciável e portanto $\mathbf{x}(\boldsymbol{\xi})$ é diferenciável. Assim podemos calcular o Jacobiano da transformação isoparamétrica, (a positividade é devido ao sentido anti-horário do mapeamento) dado por

$$J = \det \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} > 0 \quad (4.40)$$

Como $\mathbf{x} : \Omega_b \rightarrow \Omega_e$ é bijetora e de classe C^1 (pois φ_a tem classe C^1) e o Jacobiano é positivo, então, o Teorema da Função Inversa garante a existência da função inversa $\mathbf{x}^{-1} = \boldsymbol{\xi} : \Omega_e \rightarrow \Omega_b$, com \mathbf{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,

$$\varphi_a(\xi, \eta) = \frac{1}{4}(1 + \xi_a \xi)(1 + \eta_a \eta), \quad a = 1, 2, 3, 4, \quad (4.41)$$

com (ξ_a, η_a) definida em (4.38). Além disso o gradiente da função φ_a , é dado por

$$\nabla \varphi_a(\xi, \eta) = \frac{1}{4} \left(\xi_a(1 + \eta_a \eta), \eta_a(1 + \xi_a \xi) \right), \quad a = 1, 2, 3, 4. \quad (4.42)$$

As funções φ_a e $\nabla \varphi_a$ estão definidas nas subrotinas **Phi** e **DPhi** do programa.

Para o cálculo do Jacobiano (4.40) são usadas as funções $\mathbf{x}(\boldsymbol{\xi})$ em (4.39) e $\nabla \varphi_a(\xi)$ em (4.42) da seguinte forma:

$$\left(\frac{\partial x}{\partial \xi}, \frac{\partial x}{\partial \eta} \right) = \sum_{a=1}^4 x_a^e \left(\frac{\partial \varphi_a}{\partial \xi}(\xi, \eta), \frac{\partial \varphi_a}{\partial \eta}(\xi, \eta) \right) = \sum_{a=1}^4 x_a^e \nabla \varphi_a(\xi, \eta), \quad (4.43)$$

$$\left(\frac{\partial y}{\partial \xi}, \frac{\partial y}{\partial \eta} \right) = \sum_{a=1}^4 y_a^e \left(\frac{\partial \varphi_a}{\partial \xi}(\xi, \eta), \frac{\partial \varphi_a}{\partial \eta}(\xi, \eta) \right) = \sum_{a=1}^4 y_a^e \nabla \varphi_a(\xi, \eta), \quad (4.44)$$

Utilizando $\nabla \varphi_a(\xi, \eta)$ podemos calcular o Jacobiano da transformação entre o quadrado biunitário Ω_b e o quadrilátero Ω_e . Quando Ω_e é um retângulo, o Jacobiano pode ser simplesmente calculado por

$$J = \frac{1}{4}(x_2^e - x_1^e)(y_2^e - y_1^e) = \frac{1}{4}dx^e dy^e. \quad (4.45)$$

Se os retângulos são uniformes então 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 (4.45) 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 biunitário 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

$$\begin{bmatrix} \frac{\partial \varphi_a}{\partial x} \\ \frac{\partial \varphi_a}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial \varphi_a}{\partial \xi} \\ \frac{\partial \varphi_a}{\partial \eta} \end{bmatrix}. \quad (4.46)$$

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 (4.46), mas eles podem ser calculados usando a matriz Hessiana \mathbf{x}_ξ definida por

$$\mathbf{x}_\xi = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}. \quad (4.47)$$

Da definição de $\mathbf{x}(\xi, \eta)$ dada por (4.36) e (4.37), calcula-se a matriz \mathbf{x}_ξ . Então, a inversa de \mathbf{x}_ξ é dada por

$$(\mathbf{x}_\xi)^{-1} = \boldsymbol{\xi}_x = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} = \frac{1}{J} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial x}{\partial \eta} \\ -\frac{\partial y}{\partial \xi} & \frac{\partial x}{\partial \xi} \end{bmatrix} = \frac{1}{J} \bar{\mathbf{x}}_\xi, \quad (4.48)$$

onde $J = J(\xi, \eta) = \det(\mathbf{x}_\xi)$ e $\bar{\mathbf{x}}_\xi$ é a matriz adjunta de \mathbf{x}_ξ .

Substituindo em (4.46) temos

$$\begin{bmatrix} \frac{\partial \varphi_a}{\partial x} \\ \frac{\partial \varphi_a}{\partial y} \end{bmatrix} = \frac{1}{J} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} \begin{bmatrix} \frac{\partial \varphi_a}{\partial \xi} \\ \frac{\partial \varphi_a}{\partial \eta} \end{bmatrix}. \quad (4.49)$$

A matriz $(\mathbf{x}_\xi)^{-1}$ calculada em (4.48) 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

$$\xi(x, y) = \frac{1}{dx_e}((x - x_1^e) + (x - x_2^e)), \quad (4.50)$$

$$\eta(x, y) = \frac{1}{dy_e}((y - y_1^e) + (y - y_2^e)). \quad (4.51)$$

É fácil notar que (ξ, η) é um mapeamento entre o retângulo Ω_e e o quadrado biunitário Ω_b . Assim temos

$$(\mathbf{x}_\xi)^{-1} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{2}{dx_e} & 0 \\ 0 & \frac{2}{dy_e} \end{bmatrix}$$

Portanto,

$$\left(\frac{\partial \varphi_a}{\partial x}, \frac{\partial \varphi_a}{\partial y}\right) = \left(\frac{\partial \varphi_a}{\partial \xi} \frac{2}{dx_e}, \frac{\partial \varphi_a}{\partial \eta} \frac{2}{dy_e}\right). \quad (4.52)$$

4.6 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 $g : \Omega_b \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ uma função integrável, então,

$$\int_{\Omega_b} g d\Omega = \int_{-1}^1 \int_{-1}^1 g(\xi, \eta) d\xi d\eta. \quad (4.53)$$

Para o número de pontos interiores igual a dois nas direções ξ e η , os pontos são

$$\begin{aligned} (\xi_1, \eta_1) &= \left(\frac{-\sqrt{3}}{3}, \frac{-\sqrt{3}}{3}\right), & (\xi_2, \eta_2) &= \left(\frac{\sqrt{3}}{3}, \frac{-\sqrt{3}}{3}\right), \\ (\xi_3, \eta_3) &= \left(\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right), & (\xi_4, \eta_4) &= \left(\frac{-\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right), \end{aligned}$$

com pesos

$$w_1 = w_2 = w_3 = w_4 = 1.$$

Logo, temos

$$\int_{\Omega_e} g(\xi, \eta) d\Omega = \sum_{k=1}^4 g(\xi_k, \eta_k). \quad (4.54)$$

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 $Q = Q(x, y)$, $\forall (x, y) \in \Omega \subset \mathbb{R}^2$ 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 (4.29) pode ser escrita como

$$K_{ab}^e = \int_{\Omega_e} (\nabla \varphi_a(x, y))^T Q(x, y) (\nabla \varphi_b) d\Omega = \int_{\Omega_b} B_a^T Q(\xi, \eta) B_b |J(\xi, \eta)| d\Omega_b, \quad (4.55)$$

onde por B_a estamos denotando o lado direito de (4.49), ou

$$B_a = \begin{bmatrix} \xi_x \end{bmatrix}_{2 \times 2} \cdot \begin{bmatrix} \frac{\partial \varphi_a}{\partial \xi} \\ \frac{\partial \varphi_a}{\partial \eta} \end{bmatrix}_{2 \times 4}$$

que é uma matriz de ordem 2×4 .

Para cada Ω_e é necessário o cálculo da integral (4.55) que envolve o cálculo do Jacobiano $J(\xi, \eta)$ e a matriz adjunta \bar{x}_ξ . Assim, para quadriláteros, a matriz local K_{ab}^e pode ser obtida utilizando um método de integração numérica.

Como vimos, se Ω_e é um retângulo então o Jacobiano é dado por (4.45) e assim temos a seguinte matriz local:

$$K_{ab}^e = J \int_{\Omega_b} (B_a^T Q B_b) d\Omega_b. \quad (4.56)$$

Para o cálculo da matriz local K_{ab}^e dada por (4.56) é suficiente definir

$$g(\xi, \eta) = B_a^T(\xi, \eta) Q(\xi, \eta) B_b(\xi, \eta)$$

e usar a quadratura Gaussiana (4.54).

Um outro procedimento para calcular a matriz local K_{ab}^e é dado na forma componente, como descreveremos a seguir:

A matriz K_{ab}^e definida por (4.29) pode ser escrita por

$$K_{ab}^e = \int_{\Omega_e} (\nabla \varphi_a)^T Q (\nabla \varphi_b) d\Omega = \sum_{k,l=1}^2 \int_{\Omega_e} Q_{kl} \frac{\partial \varphi_a}{\partial x_k} \frac{\partial \varphi_b}{\partial x_l} d\Omega. \quad (4.57)$$

Como

$$\frac{\partial \varphi_a}{\partial x_k} = \frac{\partial \varphi_a}{\partial \xi_i} \frac{\partial \xi_i}{\partial x_k},$$

tem-se

$$K_{ab}^e = \sum_{i,j,k,l=1}^2 \int_{\Omega_b} Q_{kl} \frac{\partial \varphi_a}{\partial \xi_i} \frac{\partial \xi_i}{\partial x_k} \frac{\partial \varphi_b}{\partial \xi_j} \frac{\partial \xi_j}{\partial x_l} |J| d\Omega_b. \quad (4.58)$$

Usando (4.52), obtém-se

$$K_{ab}^e = \sum_{i,j=1}^2 \frac{4|J|}{dx_i^e dx_j^e} \int_{\Omega_b} Q_{ij} \frac{\partial \varphi_a}{\partial \xi_i} \frac{\partial \varphi_b}{\partial \xi_j} d\Omega_b. \quad (4.59)$$

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,

$$A = \text{NoLG}(1, e).$$

Substituindo em (4.59) obtém-se

$$K_{ab}^e = \sum_{i,j=1}^2 \frac{4|J|}{dx_i^e dx_j^e} Q_{ij}^e(A) \int_{\Omega_b} \frac{\partial \varphi_a}{\partial \xi_i} \frac{\partial \varphi_b}{\partial \xi_j} d\Omega_b.$$

Definindo a matriz

$$Q_{abij} = \int_{\Omega_b} \frac{\partial \varphi_a}{\partial \xi_i} \frac{\partial \varphi_b}{\partial \xi_j} d\Omega_b = \int_{-1}^1 \int_{-1}^1 \frac{\partial \varphi_a}{\partial \xi_i} \frac{\partial \varphi_b}{\partial \xi_j} d\xi d\eta. \quad (4.60)$$

que é independente do elemento e , temos

$$K_{ab}^e = 4|J| \sum_{i,j=1}^2 \frac{1}{dx_i^e dx_j^e} Q_{ij}^e(A) Q_{abij}, \quad (4.61)$$

onde $1 \leq a, b \leq 4$.

Desde que $1 \leq i, j \leq 2$ tem-se então as matrizes Q_{ab11} , Q_{ab12} , Q_{ab21} e Q_{ab22} de ordem 4 definidas por (4.60) 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 (4.60) pode se calculada por

$$Q_{abij} = \sum_{l=1}^4 g_{abij}(\xi_l, \eta_l). \quad (4.62)$$

Estas matrizes podem ser calculadas diretamente e são dadas por

$$\begin{aligned}
 Q_{ab11} &= \frac{1}{6} \begin{bmatrix} 2 & -2 & -1 & 1 \\ -2 & 2 & 1 & -1 \\ -1 & 1 & 2 & -2 \\ 1 & -1 & -2 & 2 \end{bmatrix}, \\
 Q_{ab12} &= \frac{1}{4} \begin{bmatrix} 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \end{bmatrix}, \\
 Q_{ab21} &= \frac{1}{4} \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \end{bmatrix}, \\
 Q_{ab22} &= \frac{1}{6} \begin{bmatrix} 2 & 1 & -1 & -2 \\ 1 & 2 & -2 & -1 \\ -1 & -2 & 2 & 1 \\ -2 & -1 & 1 & 2 \end{bmatrix}.
 \end{aligned}$$

O cálculo de Q_{abij} está contido na subrotina **PhiMatrix** e da matriz local está contido na subrotina **LocalSystem**.

Para o cálculo da matriz local K_{ab}^e quando Ω_e é um retângulo temos as fórmulas (4.56) e (4.61).

Em (4.56) temos de calcular 4 integrais para cada elemento e , enquanto que em (4.61), somente são necessários o cálculo de 4 integrais.

Assim, por exemplo, se o número total de elementos de uma malha $\text{Nel} = 1000$, então temos que calcular 4000 integrais usando a formula (4.56). Além disso, a matriz condutividade é variável, enquanto que pela formulação (4.61) a matriz é assumida constante por elemento.

Cálculo da Força Local

A força local definida por (4.30) pode ser escrita por

$$F_a^e = f_a^e + p_a^e - q_a^e, \quad a = 1, 2, 3, 4, \quad (4.63)$$

onde

$$f_a^e = \sum_{b=1}^4 \int_{\Omega_e} \varphi_a^e \varphi_b^e f_b^e d\Omega_e, \quad (4.64)$$

$$p_a^e = \sum_{b=1}^4 \int_{\Gamma_p} \varphi_a^e \varphi_b^e p_b^e d\Gamma_e, \quad (4.65)$$

$$q_a^e = \sum_{b=1}^4 K_{ab}^e q_b^e. \quad (4.66)$$

As igualdades (4.64), (4.65) e (4.66) são válidas para todo nó local $b \in \Omega_e$ tal que o correspondente nó global B , dado por $B = \text{NoLG}(b, e)$ satisfaça as respectivas condições: $B \in N$, $B \in N_p$ e $B \in N_q$.

Cálculo de f_a^e

Consideremos Ω_e um retângulo. De (4.50) e (4.51) as funções $\xi(x, y)$ e $\eta(x, y)$ são aplicações isoparamétricas entre Ω_e e o quadrado biunitário Ω_b .

Dessa forma tem-se

$$f_a^e = \sum_{b=1}^4 f_b^e \int_{\Omega_e} \varphi_a^e \varphi_b^e d\Omega_e = \sum_{b=1}^4 f_b^e \int_{\Omega_b} \varphi_a^e \varphi_b^e J d\Omega_b.$$

Definindo

$$Q_{ab} = \int_{\Omega_b} \varphi_a^e \varphi_b^e d\Omega_b, \quad (4.67)$$

então,

$$f_a^e = J \sum_{b=1}^4 f_b^e Q_{ab}, \quad a = 1, 2, 3, 4. \quad (4.68)$$

Usando as definições de $\varphi_a(\xi, \eta)$ dada por (4.41), a matriz Q_{ab} pode ser calculada diretamente ou usando-se quadratura Gaussiana.

A matriz é dada por

$$Q_{ab} = \frac{1}{9} \begin{bmatrix} 4 & 2 & 1 & 2 \\ 2 & 4 & 2 & 1 \\ 1 & 2 & 4 & 2 \\ 2 & 1 & 2 & 4 \end{bmatrix}. \quad (4.69)$$

O cálculo de Q_{ab} está contido na subrotina **PhiMatriz**.

Para as funções de interpolação linear φ_a o método de quadratura Gaussiana com dois pontos interiores em cada direção para o cálculo de (4.67) é exato. 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 (4.67).

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 $A = \text{NoLG}(a, e)$. 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

$$F_a^e = f_a^e, \quad a = 1, 2, 3, 4.$$

$$\text{pois, } p_a^e = q_a^e = 0.$$

- 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

$$A = \text{NoLG}(a, e), \quad \text{onde } A \in \Gamma_p \text{ ou } A \in \Gamma_q.$$

Por exemplo para o elemento $e = 4$ temos

$$\begin{aligned} 5 &= \text{NoLG}(2, 4) \in \Gamma_q \\ 10 &= \text{NoLG}(3, 4) \in \Gamma_p \\ 4 &= \text{NoLG}(1, 4) \notin \Gamma_q \cup \Gamma_p \\ 9 &= \text{NoLG}(4, 4) \notin \Gamma_q \cup \Gamma_p \end{aligned}$$

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

$$q_a^e = \sum_{b=1}^4 K_{ab}^e q_b^e, \quad a = 1, 2, 3, 4.$$

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 $A = \text{NoLG}(a, e)$.

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 e portanto,

$$u(B) = q(B) = q_b^e$$

Se $B \notin \Gamma_q \Rightarrow q_b^e = 0$. Assim 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, $\text{typ}(B) = 1$. Se existe pelo menos um nó $B = \text{NoLG}(b, e)$ do tipo $\text{typ}(B) = 1$, 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 $p_b^e = p(x_b^e) = p(B)$.

Definimos dois tipos de nó $B \in \Gamma_p$. Os nós com fluxo na direção x são os nós pertencentes as fronteiras $\Gamma_1 = \Gamma_{p1}$ e $\Gamma_3 = \Gamma_{p3}$ e os nós com fluxo na direção y são os nós da fronteira pertencentes a $\Gamma_2 = \Gamma_{p2}$ e $\Gamma_4 = \Gamma_{p4}$. Geometricamente mostramos

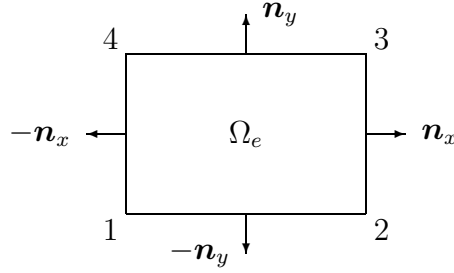


Figura 4.6: Normais externas

a fronteira Γ_p na Fig. 4.6, onde Γ_{p1} , Γ_{p2} , Γ_{p3} , Γ_{p4} são 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 o qual o fluxo normal na direção y e direção x são conhecidos, respectivamente.

Definimos a matriz M_{ab} por

$$M_{ab} = \int_{\Gamma} \varphi_a^e \varphi_b^e d\Gamma, \quad (4.70)$$

então,

$$p_a^e = \sum_{b=1}^4 M_{ab} p_b^e. \quad (4.71)$$

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} , observe que a função de interpolação linear só depende de x . Assim:

Em Γ_{p1} as funções $\varphi_a(x, y) = \varphi_a(x, y_1^e)$ são dadas por

$$\varphi_1(x, y_1^e) = \frac{x - x_2^e}{x_1^e - x_2^e}, \quad \varphi_2(x, y_1^e) = \frac{x - x_1^e}{x_2^e - x_1^e}, \quad \varphi_3(x, y_1^e) = \varphi_4(x, y_1^e) = 0.$$

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 dadas acima são obtidas das funções de interpolação (4.32) a (4.35), definidas sobre retângulos.

Usando a definição de M_{ab} , conclui-se em Γ_{p1} , que os coeficientes não necessariamente nulos são M_{11} , M_{12} , M_{21} , M_{22} e em Γ_{p3} são M_{33} , M_{34} , M_{43} , M_{44} .

Os coeficientes são calculados por

$$\begin{aligned} M_{11} &= \int_{x_1^e}^{x_2^e} \varphi_1(x) \varphi_1(x) dx = \frac{dx^e}{3} = M_{22}, \\ M_{12} &= \int_{x_1^e}^{x_2^e} \varphi_1(x) \varphi_2(x) dx = \frac{dx^e}{6} = M_{21}. \end{aligned}$$

Então 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

$$M_{ab} = \frac{dx^e}{6} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

O cálculo da matriz M_{ab} para as fronteiras Γ_{p2} e Γ_{p4} são análogas ao caso anterior, exceto que as funções de interpolação são agora constantes em relação a x . Assim 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

$$M_{ab} = \frac{dy^e}{6} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

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:

$$M_{ab} = \frac{dx^e}{6} \begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix} \quad (4.72)$$

$$M_{ab} = \frac{dy^e}{6} \begin{bmatrix} 2 & 0 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix} \quad (4.73)$$

Considere o nó global A associado ao nó local a do elemento e pela relação,

$$A = \text{NoLG}(a, e).$$

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 (4.73).

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 (4.72).

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

$$p_b^e = 0 \quad \text{se } b = 1, 4,$$

Então,

$$\begin{aligned} p_1^e &\leftarrow 0 \\ p_2^e &\leftarrow p_2^e M_{22} + p_3^e M_{23} = \frac{dy^e}{6} (2p_2^e + p_3^e) \\ p_3^e &\leftarrow p_2^e M_{32} + p_3^e M_{33} = \frac{dy^e}{6} (p_2^e + 2p_3^e) \\ p_4^e &\leftarrow 0 \end{aligned}$$

Desde que $\Gamma_p \cap \Gamma_q = \phi$, então, neste caso 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 $A_1, A_4 \in \Gamma_{p4}$. Neste 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 dado 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

$$\begin{aligned} p_1^e &\leftarrow p_1^e M_{11} + p_2^e M_{12} = \frac{dx^e}{6}(2p_1^e + p_2^e) \\ p_2^e &\leftarrow p_1^e M_{21} + p_2^e M_{22} = \frac{dx^e}{6}(p_1^e + 2p_2^e) \\ p_3^e &\leftarrow 0 \\ p_4^e &\leftarrow 0 \end{aligned}$$

Assim,

$$\begin{aligned} F_1^e &\leftarrow f_1^e + p_1^e \\ F_2^e &\leftarrow f_2^e + p_2^e \end{aligned}$$

Os valores $p_a^e = p(x_a^e, y_a^e) = p(A)$ onde a função $p = p(x, y)$ são prescritos no problema. Quando calculamos o valor de p_a^e , no exemplo anterior, consideramos dois nós globais para o qual o fluxo é prescrito. Mas é claro que, a quantidade de nós prescritos em cada elemento pode ser diferente de 2 (dois) 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 de Neumann, são definidos pelas matrizes (4.72) e (4.73), quando o fluxo está definido na direção x e y . A partir da identificação de todos os nós de fronteira feita pela subrotina **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 $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$. Esse procedimento é feito pela subrotina **TractionBoundary**, que obviamente somente 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 (4.70), usamos o fato de que os elementos são retângulos. De um modo mais geral, se os elementos da malha são quadriláteros, então um dos procedimentos para o calcular os elementos da matriz é através da parametrização. Para isso considere o segmento de reta $(tx_b + (1-t)x_a)$ e a seguinte função de interpolação:

$$\varphi_a(tx_b + (1-t)x_a) = 1 - t.$$

Então, tem-se que

$$\varphi_a(x) = \begin{cases} 1 & \text{se } x = x_a \\ 0 & \text{se } x = x_b \end{cases}$$

Logo

$$\begin{aligned} M_{ab} &= \int_{\Gamma} \varphi_a^e \varphi_b^e d\Gamma = \int_0^1 \varphi_a(tx_b + (1-t)x_a) \varphi_b(tx_b + (1-t)x_a) |\Gamma_{ab}| dt \\ &= \int_0^1 (1-t)t |\Gamma_{ab}| dt = \frac{|\Gamma_{ab}|}{6} \end{aligned}$$

$$\begin{aligned}
M_{aa} = M_{bb} &= \int_{\Gamma} \varphi_b^e \varphi_b^e d\Gamma = \int_0^1 \varphi_b(tx_b + (1-t)x_a) \varphi_b(tx_b + (1-t)x_a) |\Gamma_{ab}| dt \\
&= \int_0^1 t^2 |\Gamma_{ab}| dt = \frac{|\Gamma_{ab}|}{3}
\end{aligned}$$

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.

4.7 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 $e = 1, 2, \dots, \text{Nel}$ e $1 \leq a, b \leq 4$, serão utilizados na obtenção da matriz global e do vetor força global através da relação:

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

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 $\text{Neq} \times 1$, 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 dado nas seguintes etapas:

I) Identificação do nó local (e, a) com o nó global A , usando a subrotina:

$$A = \text{NoLG}(e, a).$$

II) Identificação do nó global A com o número da equação correspondente no sistema, $\text{eqn}[A]$, 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 $A = \text{NoLG}(a, e)$, $A = 1, 2, \dots, 25$, $a = 1, 2, 3, 4$, $e = 1, 2, \dots, 16$, então

$$K = \sum_{e=1}^{16} K^e, \quad F = \sum_{e=1}^{16} F^e.$$

Na 1ª 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 2ª 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 $\text{Neq} = 20$, com o número total de nós $\text{Nno} = 25$, 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 serão 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 1ª equação do sistema e portanto precisamos calcular a 1ª linha do sistema: $K_{11}, K_{12}, \dots, K_{1,20}$. Mas o nó $A = 2$ pertence aos elementos $e = 1$ e $e = 2$, mais precisamente $2 = \text{NoLG}(1, 2) = \text{NoLG}(2, 1)$. Assim:

$$K_{11} \leftarrow K_{22}^1 + K_{11}^2$$

$$K_{13} \leftarrow K_{24}^1$$

$$K_{14} \leftarrow K_{23}^1 + K_{14}^2$$

$$K_{15} \leftarrow K_{13}^2$$

3) O nó $A = 3$ é prescrito e portanto semelhante ao nó $A = 1$.

4) Para o nó $A = 4$ temos a 2ª linha do sistema, logo:

$$\begin{aligned} K_{22} &\leftarrow K_{22}^3 + K_{11}^4 \\ K_{25} &\leftarrow K_{24}^3 \\ K_{26} &\leftarrow K_{23}^3 + K_{14}^4 \\ K_{27} &\leftarrow K_{13}^4 \end{aligned}$$

5) O nó $A = 5$ é semelhante ao nó $A = 3$.

6) Para o nó $A = 6$ temos a 3ª linha do sistema dado por

$$\begin{aligned} K_{33} &\leftarrow K_{44}^1 + K_{11}^5 \\ K_{34} &\leftarrow K_{34}^1 + K_{12}^5 \\ K_{38} &\leftarrow K_{14}^5 \\ K_{39} &\leftarrow K_{13}^5 \end{aligned}$$

7) Para o nó $A = 7$ temos a 4ª linha do sistema dado por

$$\begin{aligned} K_{44} &\leftarrow K_{22}^5 + K_{11}^6 + K_{33}^1 + K_{44}^2 \\ K_{45} &\leftarrow K_{12}^6 + K_{34}^2 \\ K_{48} &\leftarrow K_{2,4}^5 \\ K_{49} &\leftarrow K_{23}^5 + K_{14}^6 \\ K_{49} &\leftarrow K_{23}^5 + K_{14}^6 \\ K_{410} &\leftarrow K_{13}^6 \end{aligned}$$

8) Para o nó $A = 8$ temos a quinta linha do sistema dada por

$$\begin{aligned} K_{55} &\leftarrow K_{33}^2 + K_{44}^3 + K_{22}^6 + K_{11}^7 \\ K_{56} &\leftarrow K_{34}^3 + K_{12}^7 \\ K_{59} &\leftarrow K_{24}^6 \\ K_{510} &\leftarrow K_{23}^6 + K_{14}^7 \\ K_{511} &\leftarrow K_{13}^7 \end{aligned}$$

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 $i \geq j$.

[illegible]

Figura 4.7: Matriz global K

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 $F = [F_1, F_2, \dots, F_{20}]$.

Desde que K_{ab}^e e F_a^e são conhecidos então 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)]**.

4.8 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 nos métodos iterativos não existe o problema de armazenamento, mas têm a desvantagem de que o raio espectral da matriz seja menor que um (1) para convergência da solução.

A matriz K , obtida pelo método de elementos finitos, tem uma estrutura especial, isto é, uma matriz do tipo banda no qual permite uma compactação minimizando o problema de armazenamento. Por essa razão, o sistema linear será 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, então, 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. Se a matriz Q_{ij} é positiva definida, a matriz K também é positiva definida. Assumimos que a matriz Q_{ij} é somente simétrica e portanto K é apenas simétrica.

Assim, a matriz K do sistema linear é uma matriz banda e simétrica. Dessa forma, não podemos usar o método de Cholesky. Usaremos um método variante da decomposição LU , denominado $U^T D U$ que utiliza o algoritmo de Crout, e que é uma consequência do teorema:

Teorema. *Seja A uma matriz simétrica, então existe uma matriz triangular superior U com diagonal principal unitária e uma matriz diagonal D tal que*

$$A = U^T D U$$

Observação:

- (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 qual 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,

$$A = U^T D U = U^T D^{1/2} D^{1/2} U = (D^{1/2} U)^T (D^{1/2} U) = L^T L.$$

- (3) Uma matriz A não singular, pode ser decomposta na forma:

$$A = \hat{L} \hat{U}$$

onde \hat{L} é uma matriz triangular inferior com diagonal unitária e \hat{U} uma matriz triangular superior. Além disso, se A é simétrica,

$$LU = A = A^T = (\hat{L} \hat{U})^T = \hat{U}^T \hat{L}^T = U^T D U$$

isto é, a decomposição de Crout é um caso particular da decomposição $\hat{L} \hat{U}$ no caso simétrico, onde

$$\hat{U} = U, \quad \hat{L}^T = D U.$$

- (4) Usando a 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, então são armazenados na diagonal de U os elementos da matriz D .

- (5) Após a decomposição de Crout, o sistema linear $Ax = b$ pode ser resolvido por

$$Ax = (U^T D U)(x) = (U^T D)y = U^T z = b,$$

onde

$$Ux = y, \quad Dy = z, \quad U^T z = b.$$

Algoritmo de Crout

A matriz simétrica A pode ser fatorada por

$$A_{ij} = \sum_{k=1}^j U_{ki} D_{kk} U_{kj}, \quad 1 \leq i \leq j$$

onde

$$U_{ii} = 1 \quad \text{e} \quad U_{ij} = 0 \text{ se } i > j.$$

Queremos determinar a matriz DU . Como a matriz U é triangular superior e $U_{ii} = 1$, então é suficiente calcular os elementos U_{ij} , para os quais $i < j$. Sem perda de generalidade, consideremos a matriz A com ordem 4.

A obtenção do algoritmo diferente do Método de Eliminação de Gauss, consiste em fixar as colunas j 's e variar as linhas i 's até a diagonal. Com efeito:

Para $j = 1$ e $i = 1$, temos

$$A_{11} = U_{11} D_{11} U_{11} = D_{11}$$

Para $j = 2$ e $i = 1, 2$, temos

$$A_{12} = U_{11} D_{11} U_{12} + U_{21} D_{22} U_{22} = D_{11} U_{12}$$

$$A_{22} = U_{12} D_{11} U_{12} + U_{22} D_{22} U_{22} = U_{12} D_{11} U_{12} + D_{22}$$

Para $j = 3$ e $i = 1, 2, 3$, temos

$$A_{13} = D_{11} U_{13}$$

$$A_{23} = U_{12} D_{11} U_{13} + D_{22} U_{23}$$

$$A_{33} = U_{13} D_{11} U_{13} + U_{23} D_{22} U_{23} + D_{33}$$

Para $j = 4$ e $i = 1, 2, 3, 4$, temos

$$A_{14} = D_{11} U_{14}$$

$$A_{24} = U_{12} D_{11} U_{14} + D_{22} U_{24}$$

$$A_{34} = U_{13} D_{11} U_{14} + U_{23} D_{22} U_{24} + D_{33} U_{34}$$

$$A_{44} = U_{14} D_{11} U_{14} + U_{24} D_{22} U_{24} + U_{34} D_{33} U_{34} + D_{44}$$

Logo, as matrizes D e U podem ser obtidas pelas relações acima, onde

$$D_{11} = A_{11}$$

$$\begin{aligned}
U_{12} &= A_{12}/D_{11} \\
D_{22} &= A_{22} - D_{11}U_{12}^2 \\
U_{13} &= A_{13}/D_{11} \\
U_{23} &= (A_{23} - U_{12}D_{11}U_{13})/D_{22} \\
D_{33} &= A_{33} - D_{11}U_{13}^2 - D_{22}U_{23}^2 \\
U_{14} &= A_{14}/D_{11} \\
U_{24} &= (A_{24} - U_{12}D_{11}U_{14})/D_{22} \\
U_{34} &= (A_{34} - U_{13}D_{11}U_{14} - U_{23}D_{22}U_{24})/D_{33} \\
U_{44} &= (A_{44} - D_{11}U_{14}^2 - D_{22}U_{24}^2 - D_{33}U_{34}^2)/D_{44}
\end{aligned}$$

usando o fato de $U_{ij} = 0$ para $i > j$ e $U_{ii} = 1$.

Para facilitar a programação computacional será introduzida uma matriz auxiliar definida por

$$L_{ji} = D_{ii}U_{ij}.$$

Então 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 com *algoritmo de Crout* ou o algoritmo U^tDU e pode

ser descrito na forma geral por

$$\begin{aligned}
 &\text{para } j = 1, 2, \dots, n \\
 &\quad \text{para } i = 1, 2, \dots, j-1 \\
 &\quad \quad L_{ji} = A_{ij} - \sum_{k=1}^{i-1} U_{ki} L_{jk} \\
 &\quad \quad U_{ij} = L_{ji} / D_{ii} \\
 &\quad \quad D_{jj} = A_{jj} - \sum_{i=1}^{j-1} L_{ji} U_{ij}
 \end{aligned}$$

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

$$U = \begin{bmatrix} 1 & -1 & 0 & 0 \\ & 1 & -1 & 0 \\ & & 1 & -1 \\ & & & 1 \end{bmatrix} \text{ e } D = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Assim, verifica-se que

$$A = U^T D U$$

Desde que os elementos da diagonal de U são unitários, podemos acoplar a matriz diagonal D na matriz U , isto é

$$U_{ii} \leftarrow D_{ii}, \quad i = 1, 2, \dots$$

Dessa forma, não é necessário o armazenamento da matriz D . Além disso, para evitar a proliferação de armazenamento de matrizes, a matriz U será substituída pela matriz A e a matriz auxiliar L não precisa ser armazenada. Assim, o algoritmo de Crout pode ser reescrito por

$$\begin{aligned}
 &\text{Para } j = 2, 3, \dots, n \\
 &\quad \text{para } i = 2, 3, \dots, j-1 \\
 &\quad \quad A_{ij} \leftarrow A_{ij} - \sum_{k=1}^{i-1} A_{ki} A_{kj} \\
 &\quad \text{Para } i = 1, 2, \dots, j-1 \\
 &\quad \quad T \leftarrow A_{ij} \\
 &\quad \quad A_{ij} \leftarrow T / A_{ii} \\
 &\quad \quad A_{jj} \leftarrow A_{jj} - T A_{ij}
 \end{aligned}$$

No final do procedimento obteremos 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 serão feitas no algoritmo de Crout a fim de otimizá-lo, pois a matriz K é uma matriz banda, como veremos adiante.

Considere o algoritmo de Crout para o sistema linear $Ax = b$, dado por

$$Ax = (U^T D U)x = b.$$

Então, determinar a solução $x = (x_1, x_2, \dots, x_n)$ é equivalente a resolver três sistemas lineares ordenados por

$$U^T z = b, \quad Dy = z, \quad Ux = y.$$

1. Sistema $U^T z = b$.

O sistema pode ser escrito por

$$\sum_{i=1}^n U_{ij} z_i = b_j.$$

Como a matriz U^t é triangular inferior, a solução $z = (z_1, z_2, \dots, z_n)$ é dada por

$$\begin{aligned} z_1 &= b_1 \\ \text{para } j &= 2, 3, \dots, n \\ z_j &= b_j - \sum_{i=1}^{j-1} U_{ij} z_i \end{aligned}$$

Podemos eliminar a variável z no algoritmo acima através de

$$\begin{aligned} \text{para } j &= 2, 3, \dots, n \\ b_j &= b_j - \sum_{i=1}^{j-1} A_{ij} b_i \end{aligned}$$

2. Sistema Diagonal $Dy = z$.

Como D é uma matriz diagonal, temos

$$D_{jj} y_j = z_j.$$

Portanto, fazemos

$$\begin{aligned} \text{para } j &= 1, 2, \dots, n \\ b_j &\leftarrow \frac{b_j}{A_{jj}}. \end{aligned}$$

Se $A_{jj} = 0$, a matriz original A é singular e, portanto, o sistema linear não tem solução única.

3. Sistema $Ux = y$.

O sistema 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 retrossubstituição dado por

$$\begin{aligned} x_n &= y_n \\ \text{para } i &= n-1, n-2, \dots, 1 \\ x_i &= y_i - \sum_{j=i+1}^n U_{ij}x_j \end{aligned}$$

Uma forma equivalente é dada por

$$\begin{aligned} \text{Para } j &= n, n-1, \dots, 2 \\ \text{para } i &= 1, 2, \dots, j-1 \\ b_i &\leftarrow b_i - A_{ij}b_j \end{aligned}$$

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 .

Número de Operações.

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.

4.9 Sistema Linear Global

Com a introdução do algoritmo de Crout, estamos interessados em resolver o sistema linear $KC = F$ obtido pelo método de elementos finitos.

A matriz K tem uma propriedade de ser esparsa, e com isso, podemos fazer algumas modificações no Algoritmo de Crout para evitar cálculos desnecessários entre zeros. A estrutura da matriz K , quando utilizamos quadriláteros lineares, pode ser representada na forma mostrada na pag. 139, onde as diagonais representam elementos da matriz possivelmente diferentes de zero e estes elementos são isométricos em relação à diagonal principal.

A região triangular q , quando se usa o Algoritmo de Crout, não é alterada para a obtenção da matriz triangular superior. Sendo assim, os elementos K_{ij} que são nulos 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 $\mathbf{band} = \mathbf{Nelx} + 3$, onde \mathbf{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 $\mathbf{Neq} \times R$.

Resolvendo o sistema linear

$$KC = F,$$

obtem-se o vetor solução $C = (C_1, \dots, C_{\mathbf{Neq}})$. 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

$$I = \mathbf{eqn}(A), \quad 1 \leq I \leq \mathbf{Neq},$$

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

$$u^h(x) = \sum_{A=1}^{\mathbf{Nno}} u^h(A) \varphi_A(x).$$

4.10 Erro da Solução Numérica

Para o cálculo numérico do erro nas normas dos espaços $L^2(\Omega)$ e $H^1(\Omega)$ ou $H_0^1(\Omega)$ será utilizada a solução exata, conhecida a priori em alguns casos particulares.

Para o cálculo do erro em $H^1(\Omega)$ ou $H_0^1(\Omega)$ é necessário o cálculo do gradiente de uma função $v(x)$, dada por

$$v(x) = \sum_{A=1}^{\mathbf{Nno}} d_A \varphi_A(x),$$

Assim o gradiente ∇v é dado por

$$\nabla v(x) = \sum_{A=1}^{\text{Nno}} d_A \nabla \varphi_A(x).$$

O gradiente $\nabla \varphi_A(x)$ é calculado na subrotina **Dphi**.

Denotemos por \bar{u} e u a solução exata e aproximada, respectivamente, e por $\nabla \bar{u}$ e ∇u os respectivos gradientes. Definimos o erro E na norma $L^2(\Omega)$ e na seminorma $H^1(\Omega)$ por

$$\|E\|_0 = \left(\int_{\Omega} |u - \bar{u}|^2 d\Omega \right)^{1/2},$$

$$\|E\|_1 = \left(\int_{\Omega} |\nabla u - \nabla \bar{u}|^2 d\Omega \right)^{1/2}$$

O cálculo das normas podem então ser efetuados por um método de integração numérica, como por exemplo, a quadratura Gaussiana.

Norma $L^2(\Omega)$ e Seminorma $H^1(\Omega)$

Seja a interpolação da função $v(x)$ representada em termos de elementos e coordenadas locais por

$$v(x) = \sum_{e=1}^{\text{Nel}} \sum_{a=1}^4 d_a^e \varphi_a^e(x)$$

Para o cálculo da norma do $L^2(\Omega)$ considere

$$\begin{aligned} \|v\|_0^2 &= \int_{\Omega} v(x)^2 dx = \int_{\Omega} \sum_{e=1}^{\text{Nel}} \sum_{a,b=1}^4 (d_a^e d_b^e) \varphi_a^e(x) \varphi_b^e(x) dx \\ &= \sum_{e=1}^{\text{Nel}} \sum_{a,b=1}^4 \left(d_a^e d_b^e \int_{\Omega_e} \varphi_a^e(x) \varphi_b^e(x) dx \right). \end{aligned}$$

Mas

$$\int_{\Omega_e} \varphi_a^e(x) \varphi_b^e(x) dx = \int_{\Omega_b} \varphi_a^e(\xi) \varphi_b^e(\xi) J d\Omega_b,$$

onde J é o Jacobiano da transformação isoparamétrica entre Ω_e e Ω_b , definido anteriormente. Então, obtemos

$$\|v\|_0^2 = \sum_{a,b=1}^4 Q_{ab} \sum_{e=1}^{\text{Nel}} J d_a^e d_b^e, \quad (4.74)$$

onde os coeficientes Q_{ab} da matriz local foram anteriormente calculados usando quadratura Gaussiana para obtenção em (4.67) da força local f_a^e .

Por outro lado, a seminorma é dada por

$$\|v\|_1^2 = \int_{\Omega} |\nabla v(x)|^2 dx = \sum_{e=1}^{\text{Nel}} \sum_{a,b=1}^4 d_a^e d_b^e \left(\sum_{i=1}^2 \int_{\Omega_e} \nabla_i \varphi_a^e(x) \nabla_i \varphi_b^e(x) d\Omega \right).$$

Usando a transformação isoparamétrica entre Ω_e e Ω_b temos

$$\int_{\Omega_e} \nabla_i \varphi_a^e \nabla_i \varphi_b^e dx = \sum_{i=1}^2 \frac{4J}{(dx_i^e)^2} \int_{\Omega_b} \frac{\partial \varphi_a}{\partial \xi_i} \frac{\partial \varphi_b}{\partial \xi_i} d\xi d\eta = \sum_{i=1}^2 \frac{4J}{(dx_i^e)^2} Q_{abii},$$

onde Q_{abii} é a matriz definida em (4.60) para $i = j$. Então, obtemos

$$\|v\|_1^2 = \sum_{i=1}^2 \sum_{a,b=1}^4 \sum_{e=1}^{\text{Nel}} d_a^e d_b^e \frac{4J}{(dx_i^e)^2} Q_{abii}. \quad (4.75)$$

O cálculo das normas é dado na subrotina **Norma**.

4.11 Entrada e Saída de Dados

Entrada 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 todas as outras coordenadas uniformemente.

Por exemplo: Seja o intervalo $[a, b]$ do eixo x e $[c, d]$ do eixo y . Definimos:

$$h = \frac{b-a}{\text{Nelx}}, \quad k = \frac{d-c}{\text{Nely}}, \quad x_o = a, \quad y_o = c$$

então

$$x_i = x_o + ih, \quad i = 1, 2, \dots, \text{Nelx}$$

$$y_j = y_o + jk, \quad j = 1, 2, \dots, \text{Nely}$$

quando $a = 0$ e $b = d = 1$ então:

$$x_i = ih \quad , \quad i = 1, 2, \dots, \text{Nelx}$$

$$y_j = ik \quad , \quad j = 1, 2, \dots, \text{Nely}$$

Para o caso da malha não uniforme devemos entrar com todos os dados das coordenadas x_i e y_j .

Todos os nós A da fronteira da malha, são $\text{typ}[A] = 1$ (Dirichlet) ou $\text{typ}[A] \neq 1$ (Neumann) e a forma de contribuição na força são bastantes diferentes, dentro da subrotina **CondFront**.

Saída de Dados

A critério do operador, podem ser inseridas algumas saídas de dados para a verificação da correção do programa. Este procedimento é aconselhável em várias etapas do programa. Neste programa, as saídas de dados são:

1. Elementos:

Todos os elementos com sua conectividade, ou seja, os elementos $e = 1, 2, \dots, \text{Nel}$, 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 $L^2(\Omega)$ e $H^1(\Omega)$ ou $H_0^1(\Omega)$.

4.12 Exemplos Numéricos

Consideremos agora, um exemplo numérico para o problema do calor:

$$\begin{cases} \frac{\partial}{\partial x_i} \left(-Q_{ij} \frac{\partial u}{\partial x_j} \right) = f, & \text{em } \Omega, \\ u = q, & \text{em } \Gamma_q, \\ -q_i n_i = Q_{ij} \frac{\partial u}{\partial x_j} n_i = p, & \text{em } \Gamma_p, \end{cases}$$

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

$$f(x, y) = 2\pi^2(2 \sin \pi x \cos \pi y + \cos \pi x \sin \pi y).$$

Sob estas condições, verifica-se que a função $u = u(x, y)$ dada por

$$u(x, y) = \sin \pi x \cos \pi y$$

é uma solução exata da equação diferencial do calor, independente do tipo de fronteira que definiremos nos três exemplos numéricos que se seguem.

Considere as fronteiras denotadas por

$$\begin{aligned} \Gamma_1 &= \{(x, 0) \in \partial\Omega; 0 \leq x \leq 1\} \\ \Gamma_2 &= \{(1, y) \in \partial\Omega; 0 \leq y \leq 1\} \\ \Gamma_3 &= \{(x, 1) \in \partial\Omega; 0 \leq x \leq 1\} \\ \Gamma_4 &= \{(0, y) \in \partial\Omega; 0 \leq y \leq 1\} \end{aligned}$$

Para a unicidade de solução do problema em Ω vamos considerar alguns tipos de fronteira correspondendo à solução exata $u(x, y)$.

Exemplo 1. *Fronteira 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

$$q(x) = \begin{cases} \sin \pi x & \text{em } \Gamma_{q_1}, \\ 0 & \text{em } \Gamma_{q_2}, \\ -\sin \pi x & \text{em } \Gamma_{q_3}, \\ 0 & \text{em } \Gamma_{q_4}. \end{cases}$$

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

$$p = Q_{ij} \frac{\partial u}{\partial x_j} n_i.$$

Basta relembrar que o vetor normal unitário da fronteira é dado por $n = (0, -1); (1, 0); (0, 1); (-1, 0)$ em $\Gamma_{p_1}, \Gamma_{p_2}, \Gamma_{p_3}$ e Γ_{p_4} , respectivamente. 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 fronteira 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 $\Gamma_p \neq \emptyset; \Gamma_q \neq \emptyset$, e $\Gamma_p \cap \Gamma_q = \emptyset$.

4.13 Unicidade: Problema de Neumann

O problema do calor com condições de fronteira de Neumann, como no Exemplo 2 não tem solução única. Pode-se comprovar que se $u = u(x, y)$ é uma solução então

$$v(x, y) = u(x, y) + C,$$

para qualquer constante C , também é solução. Para a obtenção da unicidade é necessário impor alguma restrição à solução.

Uma das formas de assegurar a unicidade é restringir o conjunto das possíveis soluções, por exemplo, restringir o espaço $H^1(\Omega)$ na forma

$$V = \{u \in H^1(\Omega); \int_{\Omega} u \, d\Omega = 0\}.$$

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,

$$\int_{\Omega} (v(x, y) + C) d\Omega = \int_{\Omega} v(x, y) d\Omega + \mu(\Omega)C = 0,$$

que implica

$$C = -\frac{1}{\mu(\Omega)} \int_{\Omega} v(x, y) d\Omega.$$

Uma outra forma de assegurar a unicidade é prescrever a solução em um ponto, quando possível, tal como,

$$u(\bar{x}, \bar{y}) = \alpha, \quad (\bar{x}, \bar{y}) \in \partial\Omega, \quad \alpha \text{ fixo.}$$

Esta maneira também assegura uma única solução das infinitas possíveis soluções.

Este último procedimento será o adotado no programa computacional, por ser mais simples. Com a condição acima, a fronteira não é mais uma fronteira de Neumann no sentido estrito, pois neste caso existe pelo menos um ponto $(x, y) \in \Gamma_q$.

A não unicidade da solução do problema do calor, somente com condição de fronteira de Neumann, também pode ser verificada através do sistema linear:

$$KC = F$$

Pode se verificar, para uma malha suficientemente pequena com razoáveis dígitos significativos, que a matriz rigidez K é singular. Sendo assim, o sistema possui infinitas soluções. Quando fixamos um ponto em Γ_q , a matriz de Rigidez K torna-se não singular e o sistema admite uma única solução.

Etapas da Solução Numérica.

Para os 3 exemplos dados, construiremos a solução numérica desenvolvendo todas as etapas, para melhor compreensão deste texto, 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.

4.14 Exemplos Numéricos

Exemplo 1

1. Geração e enumeração da malha

Assumiremos que $\Omega = [0, 1] \times [0, 1]$ e a discretização uniforme em ambos os eixos com passo $h = 1/\text{Nelx}$ e $k = 1/\text{Nely}$. Logo, o número de elementos e na direção- x e direção- y é igual a 4, isto é

$$\text{Nelx} = \text{Nely} = 4$$

Consequentemente, a malha tem 5 nós no eixo- x e 5 nós no eixo- y e portanto, o número total de nós $\text{Nno} = 25$ e o número total de elementos $\text{Nel} = 16$, 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

$$q(x) = \begin{cases} \text{sen } \pi x, & \text{em } \Gamma_{q_1} = \Gamma_1, \\ 0, & \text{em } \Gamma_{q_2} = \Gamma_2, \\ -\text{sen } \pi x, & \text{em } \Gamma_{q_3} = \Gamma_3, \\ 0, & \text{em } \Gamma_{q_4} = \Gamma_4. \end{cases}$$

são inseridos da seguinte forma:

$$\begin{aligned} \Gamma_1 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[1][i], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[\mathbf{n}] = 1; \mathbf{u}[\mathbf{n}] = \sin(\pi * x[p.v[0]]); \end{cases} \\ \Gamma_2 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[2][i], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[\mathbf{n}] = 1; \mathbf{u}[\mathbf{n}] = 0; \end{cases} \\ \Gamma_3 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[3][i], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[\mathbf{n}] = 1; \mathbf{u}[\mathbf{n}] = \sin(\pi * x[p.v[0]]); \end{cases} \\ \Gamma_4 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[4][i], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[\mathbf{n}] = 1; \mathbf{u}[\mathbf{n}] = 0; \end{cases} \end{aligned}$$

Como todos os nós da fronteira são prescritos a solução $\mathbf{u}[\mathbf{n}]$ é conhecida nesses nós. O índice $[i]$ da função \mathbf{bdy} percorre todos os nós de cada fronteira, identificando-o 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 $\mathbf{typ}[A] = 1$ são os nós cujos valores são prescritos e a subrotina **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, $q(x) = \pm \text{sen } \pi x$. 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, como vimos, as incógnitas são os nós: $A = \{7, 8, 9, 12, 13, 14, 17, 18, 19\}$, que corresponderão as equações: $\text{eqn}[A] = 1, 2, \dots, 9$ da matriz rigidez do sistema linear, desde que, para os outros nós a solução é conhecida. Assim, $N_{eq} = 9$ e a matriz rigidez $[K]_{9 \times 9}$.

3. Matriz local e força local

Para cada elemento $e = 1, 2, 3, \dots, 16$, é calculada a matriz local K_{ab}^e dada por (4.61). Como estamos usando uma malha uniforme, o Jacobiano \dot{j} é constante e, além disso, $dx_k = dx_l = 1/4$.

Fazendo o cálculo, obtém-se para cada elemento a matriz local,

$$K_{ab}^e = \frac{1}{6} \begin{bmatrix} 11 & -2 & -7 & -2 \\ -2 & 5 & -2 & -1 \\ -7 & -2 & 11 & -2 \\ -2 & -1 & -2 & 5 \end{bmatrix}$$

A força local F_a^e é definida em (4.63) por

$$F_a^e = f_a^e + p_a^e - q_a^e,$$

onde f_a^e , p_a^e e q_a^e são definidos, respectivamente, por (4.64), (4.65) e (4.66).

No Exemplo 1, $\Gamma_p = \phi$, logo, $p_a^e = 0$. Assim,

$$F_a^e = f_a^e - q_a^e$$

Os elementos $e = \{6, 7, 10, 11\}$ não possuem nós na fronteira. Logo, para esses elementos

$$F_a^e = f_a^e.$$

Para todos os outros elementos restantes o vetor q_a^e contribui pra a força local F_a^e , $a = 1, 2, 3, 4$. Para $e = 5$ e $e = 6$ o vetor força local são dados por

$$F_a^5 = \begin{bmatrix} 0.292508 \\ 0.336815 \\ 0.292508 \\ 0.285411 \end{bmatrix}, \quad F_a^6 = \begin{bmatrix} 0.351010 \\ 0.320898 \\ 0.196797 \\ 0.248201 \end{bmatrix}.$$

4. Matriz global e força global

A matriz Global K e o vetor Força F são dados por

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

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

$$u(x, y) = \sin \pi x \cdot \cos \pi y, \quad 0 \leq x \leq 1 \quad \text{e} \quad 0 \leq y \leq 1$$

Comparando com a solução numérica aproximada $u^h(x, y)$ na norma do $L^2(\Omega)$ e $H^1(\Omega)$ obtemos

$$\begin{aligned} \|E\|_{L^2(\Omega)} &= \|u - u^h\|_{L^2(\Omega)} = 6.853220 \times 10^{-3} \\ \|E\|_{H^1(\Omega)} &= \|u - u^h\|_{H^1(\Omega)} = 5.253452 \times 10^{-2} \end{aligned}$$

Para o mesmo problema, usando $\text{Nelx} = \text{Nely} = 8 \Rightarrow h = k = 1/8$ e $\text{Nelx} = \text{Nely} = 16 \Rightarrow h = k = 1/16$, que correspondem respectivamente a $\{Nno = 81; Nel = 64; Neq = 49\}$ e $\{Nno = 289; Nel = 256; Neq = 225\}$ obtém-se os erros:

$$\begin{aligned} \|E\|_{L^2(\Omega)} &= 2.185674 \times 10^{-3}, & \|E\|_{H^1(\Omega)} &= 1.601428 \times 10^{-2}, \\ \|E\|_{L^2(\Omega)} &= 5.785226 \times 10^{-4}, & \|E\|_{H^1(\Omega)} &= 4.192837 \times 10^{-3}. \end{aligned}$$

Exemplo 2

1. Condição de fronteira

Considere a malha para $\text{Nelx} = \text{Nely} = 4$, 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 $\text{typ}[1] = 1$ e que $q(1) = 0$.

Observação: Qualquer um dos 25 nós poderia ser fixado. O número total de nós da malha $\text{Nno} = 25$. Em Γ_p é a derivada da solução (fluxo) que 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:

$$\begin{aligned}\Gamma_{p_1} &= \{2, 3, 4, 5\} \\ \Gamma_{p_2} &= \{10, 15, 20\} \\ \Gamma_{p_3} &= \{21, 22, 23, 24, 25\} \\ \Gamma_{p_4} &= \{6, 11, 16\}\end{aligned}$$

Tem-se 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 **bdy[1][i]**, **bdy[2][i]**, **bdy[3][i]** e **bdy[4][i]**, respectivamente. Para esse exemplo temos

$$\begin{aligned}\Gamma_1 &= \left\{ \begin{array}{l} n = \text{bdy}[1][i], \quad p = \text{NoPos}(n) \\ \text{Bv}[1][i] = -\pi * \cos(\pi * x[p.v[0]]); \end{array} \right. \\ \Gamma_2 &= \left\{ \begin{array}{l} n = \text{bdy}[2][i], \quad p = \text{NoPos}(n) \\ \text{Bv}[2][i] = -2\pi * \cos(\pi * y[p.v[1]]); \end{array} \right. \\ \Gamma_3 &= \left\{ \begin{array}{l} n = \text{bdy}[3][i], \quad p = \text{NoPos}(n) \\ \text{Bv}[3][i] = -\pi * \cos(\pi * x[p.v[0]]); \end{array} \right. \\ \Gamma_4 &= \left\{ \begin{array}{l} n = \text{bdy}[4][i], \quad p = \text{NoPos}(n) \\ \text{Bv}[4][i] = -2\pi * \cos(\pi * y[p.v[1]]). \end{array} \right.\end{aligned}$$

Quando a função **Bv** esta 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, àquela cujo valor no nó global $A = 1$ seja zero ($u(A) = 0$), devemos inserir na subrotina **CondFront** do programa, após Γ_4 o que segue:

$$\text{typ}[1] = 1, \quad u[1] = 0$$

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:

$$\begin{aligned} \|E\|_{L^2(\Omega)} &= 9.047955 \times 10^{-2}, \\ \|E\|_{H^1(\Omega)} &= 1.797705 \times 10^{-1}. \end{aligned}$$

Para esse exemplo temos $\{Nno = 25, Nel = 16, Neq = 24\}$.

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 $\Gamma_1, \Gamma_2, \Gamma_3$ 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:

$$\begin{aligned} \Gamma_1 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[1][i], & p = \mathbf{NoPos}(n) \\ \mathbf{Bv}[1][i] = -\pi * \cos(\pi * x[p.v[0]]); \end{cases} \\ \Gamma_3 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[3][i], & p = \mathbf{NoPos}(n) \\ \mathbf{Bv}[3][i] = -\pi * \cos(\pi * x[p.v[0]]); \end{cases} \\ \Gamma_2 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[2][i], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[n] = 1; \mathbf{u}[n] = 0; \end{cases} \\ \Gamma_4 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[4][i], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[n] = 1; \mathbf{u}[n] = 0; \end{cases} \end{aligned}$$

Considerando a malha do exemplo temos os seguintes nós globais em $\Gamma_p = \Gamma_1 \cup \Gamma_3$ e $\Gamma_q = \Gamma_2 \cup \Gamma_4$ são dados por

$$\begin{aligned} \Gamma_p &= \{2, 3, 4, 22, 23, 24\} \\ \Gamma_q &= \{5, 10, 15, 20, 25, 1, 6, 11, 16, 21, \} \end{aligned}$$

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 $e = 1, 11, 16$ são dadas abaixo, 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. Os erros, neste caso, são

$$\begin{aligned} \|E\|_{L^2(\Omega)} &= 2.658667 \times 10^{-2}, \\ \|E\|_{H^1(\Omega)} &= 1.372426 \times 10^{-1}. \end{aligned}$$

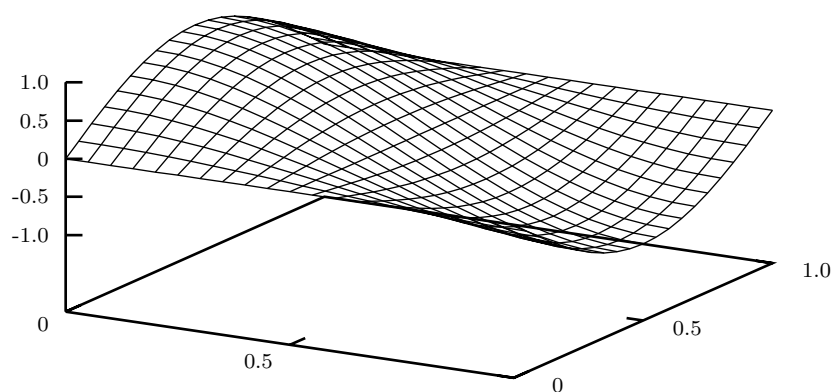
Para esse exemplo temos $\{Nno = 25, Nel = 16, Neq = 15\}$. Note que $Neq = Nno - \Gamma_q$

4.15 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

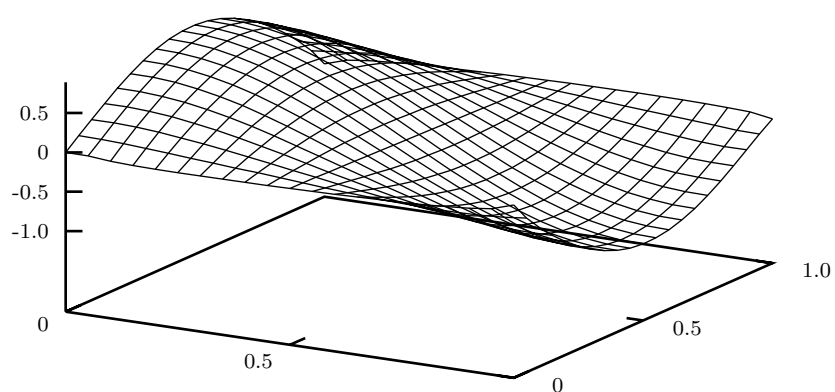
Solução numérica $u^h(x, y)$



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}

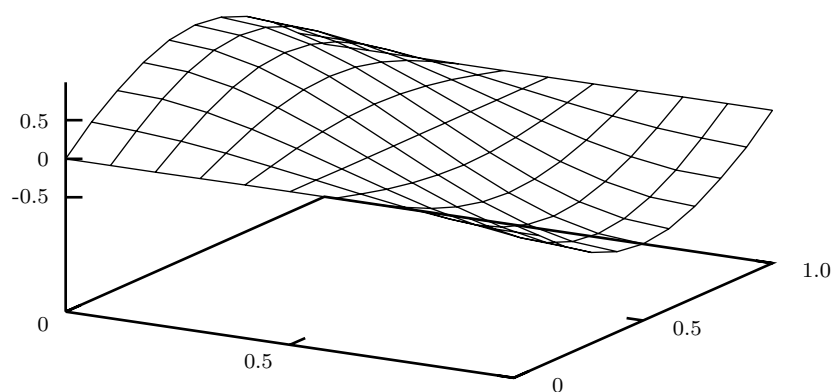
Para a malha 40×40 tem-se $\{N_{no} = 1681; N_{el} = 1600; N_{eq} = 1521\}$

Exemplo 2: Problema de NeumannSolução 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 $\{N_{no} = 1681; N_{el} = 1600; N_{eq} = 1680\}$

Exemplo 3: Problema mistoSolução 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 $\{N_{no} = 1681; N_{el} = 1600; N_{eq} = 1599\}$

4.16 Exercícios

Na literatura, é mais usual discretizar o domínio Ω por elementos triangulares Ω_e , desde que sua geometria é melhor ajustada geometricamente para domínios mais gerais do que os elementos retangulares. Considere então a formulação variacional (4.13) 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 neste capítulo será aplicado ao problema bidimensional vetorial representado pelo problema modelo de elasticidade linear. Neste caso temos uma solução vetorial e algumas mudanças serão introduzidas em relação ao problema estudado no capítulo anterior. De forma análoga, mostraremos as várias etapas para 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

Considere $\Omega \subset \mathbb{R}^n$ um conjunto aberto e limitado com fronteira $\partial\Omega$ suave e os sub-índices $1 \leq i, j, k, \ell \leq n$, com $n = 2$ ou $n = 3$ para problemas bidimensionais ou tridimensionais respectivamente. Denotamos por

- σ_{ij} : Tensor de tensão de Cauchy definido por $\sigma = (\sigma_{ij}) : \Omega \rightarrow \mathbb{R}^n \times \mathbb{R}^n$
- u_i : vetor deslocamento definido por $\mathbf{u} = (u_i) : \Omega \rightarrow \mathbb{R}^n$
- f_i : força prescrita por unidade de volume definida por $\mathbf{f} = (f_i) : \Omega \rightarrow \mathbb{R}^n$
- ϵ_{ij} : tensor de deformação infinitesimal definido por

$$\epsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \quad (5.1)$$

Pela lei de Hooke, o tensor de deformação infinitesimal ϵ_{ij} e o tensor de tensão σ_{ij} estão relacionados por

$$\sigma_{ij} = C_{ijkl} \epsilon_{kl} \quad (5.2)$$

onde C_{ijkl} é denominado coeficiente de elasticidade definido para todo $x \in \Omega$. Se o corpo é homogêneo então C_{ijkl} é constante. Note que usamos a convenção de somatório, isto é, os índices repetidos representam um somatório sobre o índice aos seus valores.

Sobre o tensor de elasticidade assumimos as hipóteses:

1. O tensor elasticidade C_{ijkl} satisfaz a condição:

$$C_{ijkl} = C_{jikl} = C_{ijlk} = C_{klij} \quad (5.3)$$

2. Para qualquer matriz simétrica S_{ij} , existem constantes positivas C e \overline{C} , tais que

$$CS_{ij}S_{ij} \leq C_{ijkl}S_{ij}S_{kl} \leq \overline{C}S_{ij}S_{ij} \quad (5.4)$$

A hipótese (1) é devido a existência da função energia potencial e da simetria do tensor de tensão e de deformação, enquanto a hipótese (2) estabelece que o tensor elasticidade é limitado e estritamente positivo definido.

Assumiremos que a fronteira $\Gamma = \partial\Omega$ admite a seguinte decomposição:

$$\begin{aligned} \overline{\Gamma}_{pi} \cup \overline{\Gamma}_{qi} &= \Gamma, \\ \Gamma_{pi} \cap \Gamma_{qi} &= \emptyset, \end{aligned}$$

para $1 \leq i \leq n$, 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. Nestas condições o problema elastostático linear é formulado por

(I) Formulação forte. Dadas as funções $f_i : \Omega \rightarrow \mathbb{R}$, $q_i : \Gamma_{qi} \rightarrow \mathbb{R}$ e $p_i : \Gamma_{pi} \rightarrow \mathbb{R}$. Determinar $u_i : \overline{\Omega} \rightarrow \mathbb{R}$ tal que

$$\begin{cases} -\frac{\partial \sigma_{ij}}{\partial x_j} = f_i & \text{em } \Omega \\ u_i = q_i & \text{em } \Gamma_{qi} \\ \sigma_{ij}n_j = p_i & \text{em } \Gamma_{pi} \end{cases} \quad (5.5)$$

onde q_i é a componente i do deslocamento prescrito e p_i é a componente i da tração prescrito na fronteira.

Se $\Gamma = \cup \Gamma_{qi}$, a condição de fronteira é conhecida como condição de Dirichlet ou fronteira de deslocamento. Se $\Gamma = \cup \Gamma_{pi}$, a condição de fronteira é conhecida como do tipo Neumann ou fronteira de tração. Se $\Gamma_{pi} \neq \emptyset$ e $\Gamma_{qi} \neq \emptyset$, a condição de fronteira é conhecida como do tipo misto.

Considere

$$H = \{\mathbf{u} \in (H^1(\Omega))^n; u_i = q_i \text{ em } \Gamma_{qi}\}$$

e

$$V = \{\mathbf{v} \in (\mathcal{D}(\Omega))^n; v_i = 0 \text{ em } \Gamma_{qi}\},$$

onde $H^1(\Omega)$ é um espaço com norma:

$$\|\mathbf{u}\|_H = \left(\int_{\Omega} (|u_i|^2 + \left| \frac{\partial u_i}{\partial x_j} \right|^2) d\Omega \right)^{1/2}$$

e $\mathcal{D}(\Omega)$ é denominado espaço das funções testes.

Agora, consideramos a formulação fraca. Multiplicando a equação (5.5) por $\mathbf{v} \in V$ e integrando sobre Ω temos

$$- \int_{\Omega} \frac{\partial \sigma_{ij}}{\partial x_j} v_i d\Omega = \int_{\Omega} f_i(x) v_i(x) d\Omega, \quad (5.6)$$

Usando a derivada do produto de funções obtemos

$$\int_{\Omega} \frac{\partial \sigma_{ij}}{\partial x_j} v_i d\Omega = \int_{\Omega} \frac{\partial (\sigma_{ij} v_i)}{\partial x_j} d\Omega - \int_{\Omega} \sigma_{ij} \frac{\partial v_i}{\partial x_j} d\Omega$$

Aplicando o Teorema da Divergência temos

$$\int_{\Omega} \frac{\partial}{\partial x_j} (\sigma_{ij} v_i) d\Omega = \int_{\Gamma} v_i \sigma_{ij} n_j d\Gamma$$

Substituindo em (5.6) obtemos

$$\int_{\Omega} \sigma_{ij} \frac{\partial v_i}{\partial x_j} d\Omega = \int_{\Omega} f_i v_i d\Omega + \int_{\Gamma} v_i \sigma_{ij} n_j d\Gamma$$

Desde que $v_i = 0$ em Γ_{qi} e $\sigma_{ij} n_j = p_i$ em Γ_{pi} tem-se

$$\int_{\Omega} \sigma_{ij} \frac{\partial v_i}{\partial x_j} d\Omega = \int_{\Omega} f_i v_i d\Omega + \int_{\Gamma_{pi}} v_i p_i d\Gamma$$

Assim, a formulação fraca é dada por

(II) Formulação fraca. Dadas as funções $f_i : \Omega \rightarrow \mathbb{R}$, $q_i : \Gamma_{qi} \rightarrow \mathbb{R}$ e $p_i : \Gamma_{pi} \rightarrow \mathbb{R}$. Determinar $\mathbf{u} \in H$ tal que

$$\int_{\Omega} \frac{\partial v_i}{\partial x_j} \sigma_{ij} d\Omega = \int_{\Omega} f_i v_i d\Omega + \int_{\Gamma_{pi}} v_i p_i d\Gamma \quad \forall \mathbf{v} \in V$$

A formulação fraca também é denominada princípio do trabalho virtual e \mathbf{v} é o deslocamento virtual.

De (5.1) e (5.2) e da simetria de C_{ijkl} , tem-se

$$\sigma_{ij} = C_{ijkl} \epsilon_{kl} = \frac{1}{2} C_{ijkl} \left(\frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k} \right) = C_{ijkl} \frac{\partial u_k}{\partial x_l}.$$

Substituindo na formulação fraca obtemos

$$\int_{\Omega} C_{ijk\ell} \frac{\partial v_i}{\partial x_j} \frac{\partial u_k}{\partial x_\ell} d\Omega = \int_{\Omega} f_i v_i d\Omega + \int_{\Gamma_{pi}} v_i p_i d\Gamma$$

Definindo por

$$a(\mathbf{v}, \mathbf{u}) = \int_{\Omega} C_{ijk\ell} \frac{\partial v_i}{\partial x_j} \frac{\partial u_k}{\partial x_\ell} d\Omega \quad (5.7)$$

$$(\mathbf{v}, \mathbf{f}) = \int_{\Omega} v_i f_i d\Omega \quad (5.8)$$

$$(\mathbf{v}, \mathbf{p})_{\Gamma} = \int_{\Gamma_{pi}} v_i p_i d\Gamma \quad (5.9)$$

podemos redefinir o problema (II) por

(II') Formulação fraca. Dados \mathbf{f} e \mathbf{p} , determinar $\mathbf{u} \in H$ tal que

$$a(\mathbf{v}, \mathbf{u}) = (\mathbf{v}, \mathbf{f}) + (\mathbf{v}, \mathbf{p})_{\Gamma}, \quad \forall \mathbf{v} \in V. \quad (5.10)$$

Formulação de Galerkin

Considere o subespaço $H^h \subset H$ e $V^h \subset V$ de dimensão finita, gerado pelos m -primeiros vetores da base do espaço de Hilbert H e V .

Seja $\mathbf{u}^h \in H^h$, tal que, $\mathbf{u}^h = \mathbf{w}^h + \mathbf{q}^h$, onde $\mathbf{w}^h \in V^h$ e

$$q_i^h(\mathbf{x}) = \begin{cases} q_i(\mathbf{x}); & \forall \mathbf{x} \in \Gamma_{qi} \\ 0 & \forall \mathbf{x} \notin \Gamma_{qi} \end{cases} \quad (5.11)$$

Definimos também a função

$$p_i^h(\mathbf{x}) = \begin{cases} p_i(\mathbf{x}); & \forall \mathbf{x} \in \Gamma_{pi} \\ 0 & \forall \mathbf{x} \notin \Gamma_{pi} \end{cases} \quad (5.12)$$

Temos que

$$u_i^h(\mathbf{x}) = w_i^h(\mathbf{x}) + q_i^h(\mathbf{x}) = q_i, \quad \forall \mathbf{x} \in \Gamma_{qi}$$

Como a formulação é válida para todo $\mathbf{v} \in V$, então, em particular, é válida em V^h , isto é,

$$a(\mathbf{v}^h, \mathbf{u}) = (\mathbf{v}^h, \mathbf{f}) + (\mathbf{v}^h, \mathbf{p})_{\Gamma}, \quad \forall \mathbf{v}^h \in V^h.$$

A formulação de Galerkin é dada por

Formulação de Galerkin. Dados \mathbf{f} , \mathbf{p} e \mathbf{q} , determinar $\mathbf{u}^h \in H^h$, tal que,

$$a(\mathbf{v}^h, \mathbf{u}^h) = (\mathbf{v}^h, \mathbf{f}) + (\mathbf{v}^h, \mathbf{p}^h)_\Gamma, \quad \forall \mathbf{v}^h \in V^h,$$

ou usando a decomposição de \mathbf{u}^h tem-se

$$a(\mathbf{v}^h, \mathbf{w}^h) = (\mathbf{v}^h, \mathbf{f}) + (\mathbf{v}^h, \mathbf{p}^h)_\Gamma - a(\mathbf{v}^h, \mathbf{q}^h), \quad \forall \mathbf{v}^h \in V^h. \quad (5.13)$$

onde $\mathbf{w}^h \in V^h$.

De forma análoga ao capítulo anterior, considere N o conjunto de nós da malha e N_q o conjunto de nós prescritos. Para cada nó A , considere φ_A a função de interpolação da base de V^h . Assim, como a solução aproximada $\mathbf{w}^h \in V^h$, podemos representá-la por

$$\mathbf{w}^h(x) = \sum_{A=1}^{\text{Nno}} \mathbf{d}_A \varphi_A(x) \quad (5.14)$$

A solução aproximada (5.14) é uma função vetorial, ou seja,

$$\mathbf{w}^h = (w_1^h, w_2^h, \dots, w_n^h),$$

onde n é a dimensão do espaço. Podemos escrever (5.14) na forma coordenada dada por

$$w_j^h(x) = \sum_{B \in N} d_{jB} \varphi_B(x) = \sum_{B \in N - N_{qj}} d_{jB} \varphi_B(x), \quad 1 \leq j \leq n. \quad (5.15),$$

pois $w_j^h(x) = 0$, se $x \in N_{qj}$.

De forma análoga, na fronteira Γ_{qj} temos

$$q_j^h(x) = \sum_{B \in N_{qj}} q_{jB} \varphi_B(x), \quad 1 \leq j \leq n. \quad (5.16)$$

Considere a base canônica do \mathbb{R}^n dada por $\mathbf{e}_j = (0, \dots, 0, 1, 0, \dots, 0)$, onde o número 1 está na j -ésima posição. Então os vetores \mathbf{w}^h e \mathbf{q}^h podem ser escritos por

$$\mathbf{w}^h = w_j^h \mathbf{e}_j \quad \mathbf{q}^h = q_j^h \mathbf{e}_j$$

onde w_j^h e q_j^h são dados por (5.15) e (5.16). Assim a forma bilinear $a(\cdot, \cdot)$ é dada por

$$a(\mathbf{v}^h, w_j^h \mathbf{e}_j) = a(\mathbf{v}^h, \sum_{B \in N - N_{qj}} d_{jB} \varphi_B \mathbf{e}_j), \quad \forall \mathbf{v}^h \in V^h$$

Como a igualdade é válida para todo $\mathbf{v}^h = v_i^h \mathbf{e}_i$, então, em particular, podemos tomar $\mathbf{v}^h \in V^h$ na forma $\mathbf{v}^h = \varphi_A \mathbf{e}_i$. Substituindo tem-se

$$a(\mathbf{v}^h, \mathbf{w}^h) = \sum_{j=1}^n \left(\sum_{B \in N - N_{qj}} a(\varphi_A \mathbf{e}_i, \varphi_B \mathbf{e}_j) d_{jB} \right), \quad 1 \leq i \leq n.$$

De forma análoga,

$$\begin{aligned} (\mathbf{v}^h, \mathbf{f}) &= (\varphi_A \mathbf{e}_i, \mathbf{f}), \\ (\mathbf{v}^h, \mathbf{p}^h)_\Gamma &= (\varphi_A \mathbf{e}_i, \mathbf{p}^h)_\Gamma, \\ a(\mathbf{v}^h, \mathbf{q}^h) &= a(\varphi_A \mathbf{e}_i, \mathbf{q}^h). \end{aligned}$$

Como anteriormente podemos interpolar a função \mathbf{q}^h pela função base φ_A , ou seja,

$$q_j^h = \sum_{A \in N_{qj}} \varphi_A(x) q_{jA}$$

onde

$$q_{jA} = q_j^h(\mathbf{x}_A), \quad \mathbf{x}_A \in \Gamma_{qj}.$$

Logo

$$a(\mathbf{v}^h, \mathbf{q}^h) = \sum_{j=1}^n \left(\sum_{B \in N_{qj}} a(\varphi_A \mathbf{e}_i, \varphi_B \mathbf{e}_j) \right) q_{jB}$$

Substituindo as relações acima em (5.13) obtemos o sistema linear

$$\sum_{j=1}^n \sum_{B \in N - N_{qj}} a(\varphi_A \mathbf{e}_i, \varphi_B \mathbf{e}_j) d_{jB} = (\varphi_A \mathbf{e}_i, \mathbf{f}) + (\varphi_A \mathbf{e}_i, \mathbf{p}^h)_\Gamma - \sum_{j=1}^n \sum_{B \in N_{qj}} a(\varphi_A \mathbf{e}_i, \varphi_B \mathbf{e}_j) q_{jB}$$

onde $A \in N - N_{qi}$.

Definindo

$$K_{AB} = a(\varphi_A \mathbf{e}_i, \varphi_B \mathbf{e}_j) \quad (5.17)$$

$$F_A = (\varphi_A \mathbf{e}_i, \mathbf{f}) + (\varphi_A \mathbf{e}_i, \mathbf{p}^h)_\Gamma - \sum_{j=1}^n \sum_{B \in N_{qj}} a(\varphi_A \mathbf{e}_i, \varphi_B \mathbf{e}_j) q_{jB}. \quad (5.18)$$

Então,

$$K_{AB} d_{jB} = F_A. \quad (5.19)$$

Fixaremos, de agora em diante, a dimensão $n = 2$. Assim, a solução aproximada $u^h(\mathbf{x})$ tem como coordenadas $u_1^h(\mathbf{x})$ e $u_2^h(\mathbf{x})$, isto é,

$$\mathbf{u}^h(\mathbf{x}) = (u_1^h(\mathbf{x}), u_2^h(\mathbf{x}))$$

onde u_1^h e u_2^h são denominados 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,

$$\mathbf{eqn}[i, A] = \begin{cases} I & \text{se } A \in N - N_{qi} \\ 0 & \text{se } A \in N_{qi} \end{cases}$$

onde $N_{qi} = \{A \in \Gamma_{qi}\}$. Sabemos que se $A \in N_{qi}$, então $\mathbf{u}^h(\mathbf{x}_A) = (u_1^h(\mathbf{x}_A), u_2^h(\mathbf{x}_A))$ é conhecida e portanto não é necessário gerar uma equação para cada deslocamento u_i^h . Poderemos ter nós A prescritos numa direção ou em duas direções.

Por exemplo,

$i \backslash A$	1	2	3	4	5
1	1	2	4	0	0
2	0	3	5	6	0

Nesse caso, temos 5 nós globais com 6 equações.

O nó $A = 1 \in \Gamma_{q2}$, logo $u_1^h(\mathbf{x}_A)$ não é prescrito, mas $u_2^h(\mathbf{x}_A)$ é prescrito.

O nó $A = 4 \in \Gamma_{q1}$, logo $u_1^h(\mathbf{x}_A)$ é prescrito e $u_2^h(\mathbf{x}_A)$ não é prescrito.

O nó $A = 5 \in \Gamma_{q1} \cap \Gamma_{q2}$, é prescrito em ambas as direções, u_1^h e u_2^h .

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

$$M = \mathbf{eqn}[i, A] \quad N = \mathbf{eqn}[j, B]$$

com $1 \leq M, N \leq \mathbf{Neq}$, o sistema linear (5.19) pode ser escrito por

$$K_{MN} D_N = F_M,$$

onde

$$K_{MN} = a(\varphi_A \mathbf{e}_i, \varphi_B \mathbf{e}_j), \quad (5.20)$$

$$F_M = (\varphi_A \mathbf{e}_i, \mathbf{f}) + (\varphi_A \mathbf{e}_i, \mathbf{p}^h)_\Gamma - \sum_{j=1}^2 \sum_{B \in N_{qj}} a(\varphi_A \mathbf{e}_i, \varphi_B \mathbf{e}_j) q_{jB} \quad (5.21)$$

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. K_{MN} é simétrica:

$$\begin{aligned} K_{MN} &= a(\varphi_A \mathbf{e}_i, \varphi_B \mathbf{e}_k) = \int_{\Omega} \frac{\partial(\varphi_A \mathbf{e}_i)}{\partial x_j} C_{ijkl} \frac{\partial(\varphi_B \mathbf{e}_k)}{\partial x_\ell} d\Omega \\ &= \int_{\Omega} \frac{\partial \varphi_A}{\partial x_j} \mathbf{e}_i C_{ijkl} \frac{\partial \varphi_B}{\partial x_\ell} \mathbf{e}_k d\Omega = \int_{\Omega} \frac{\partial \varphi_B}{\partial x_\ell} \mathbf{e}_k C_{klij} \frac{\partial \varphi_A}{\partial x_j} \mathbf{e}_i d\Omega = K_{NM} \end{aligned}$$

pois $C_{klij} = C_{ijkl}$.

2. K_{MN} é definida positiva:

Seja $\mathbf{d} = (d_1, d_2, \dots, d_{\text{Neq}})$. Então,

$$\begin{aligned} \mathbf{d}^T K \mathbf{d} &= \sum_{M,N=1}^{\text{Neq}} d_M K_{MN} d_N = \sum_{\substack{A \in N-N_{qi} \\ B \in n-n_{qj}}} d_{iA} a(\varphi_A \mathbf{e}_i, \varphi_B \mathbf{e}_j) d_{jB} \\ &= a\left(\sum_{A \in N-N_{qi}} d_{iA} \varphi_A \mathbf{e}_i, \sum_{B \in N-N_{qj}} d_{jB} \varphi_B \mathbf{e}_j \right) = a(\mathbf{u}^h, \mathbf{u}^h) \geq 0. \end{aligned}$$

Por outro lado,

$$0 = a(\mathbf{u}^h, \mathbf{u}^h) = \int_{\Omega} \frac{\partial u_i^h}{\partial x_j} C_{ijkl} \frac{\partial u_k^h}{\partial x_\ell} d\Omega.$$

De (5.3) e (5.4), C_{ijkl} é positivo definido para todas as matrizes simétricas, logo,

$$\frac{\partial u_i^h}{\partial x_j} + \frac{\partial u_j^h}{\partial x_i} = 0,$$

Isto implica que u_i^h é um movimento rígido, que pode ser representado por,

$$u_i^h = W_{ij} x_j + C,$$

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 $u_i^h = 0$ em Γ_q , então

$$W_{ij} = 0 \quad \text{e} \quad C = 0,$$

isto é, o corpo tem um movimento rígido nulo. Assim $u_i^h = 0$ 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 (5.20) temos

$$K_{AB}^{rs} = a(\varphi_A \mathbf{e}_r, \varphi_B \mathbf{e}_s), \quad 1 \leq A, B \leq N - N_{qr}$$

A forma bilinear $a(\cdot, \cdot)$ é definida por

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \frac{\partial u_i}{\partial x_k} C_{ikj\ell} \frac{\partial v_j}{\partial x_\ell} d\Omega.$$

Na forma componente, temos

$$\begin{aligned} \mathbf{u} = \varphi_A \mathbf{e}_r &\implies u_i = (u, \mathbf{e}_i) = (\varphi_A \mathbf{e}_r, \mathbf{e}_i) = \varphi_A \delta_{ir} \\ \mathbf{v} = \varphi_B \mathbf{e}_s &\implies v_j = (v, \mathbf{e}_j) = (\varphi_B \mathbf{e}_s, \mathbf{e}_j) = \varphi_B \delta_{js} \end{aligned}$$

Substituindo na forma bilinear temos

$$a(\varphi_A \mathbf{e}_r, \varphi_B \mathbf{e}_s) = \int_{\Omega} \frac{\partial \varphi_A}{\partial x_k} \frac{\partial \varphi_B}{\partial x_\ell} C_{ikj\ell} \delta_{ir} \delta_{js} d\Omega.$$

Assim, a matriz é dada na forma

$$K_{AB}^{rs} = \delta_{ir} \delta_{js} \int_{\Omega} C_{ikj\ell} \frac{\partial \varphi_A}{\partial x_k} \frac{\partial \varphi_B}{\partial x_\ell} d\Omega = \delta_{rk} \delta_{sl} \int_{\Omega} C_{ikj\ell} \frac{\partial \varphi_A}{\partial x_i} \frac{\partial \varphi_B}{\partial x_j} d\Omega. \quad (5.22)$$

Para o vetor força definido em (5.18) e substituindo os índices (i, j) por (r, s) tem-se que

$$F_A^r = (\varphi_A \mathbf{e}_r, \mathbf{f}) + (\varphi_A \mathbf{e}_r, \mathbf{p}^h)_\Gamma - \sum_{s=1}^2 \sum_{B \in N_{qs}} a(\varphi_A \mathbf{e}_r, \varphi_B \mathbf{e}_s) q_{sB}.$$

Agora,

$$\begin{aligned} (\varphi_A \mathbf{e}_r, \mathbf{f}) &= \int_{\Omega} \varphi_A (\mathbf{f}, \mathbf{e}_r) d\Omega = \int_{\Omega} \varphi_A f_r(x) d\Omega, \\ (\varphi_A \mathbf{e}_r, \mathbf{p}^h)_\Gamma &= \int_{\Gamma_{pr}} \varphi_A (\mathbf{p}^h, \mathbf{e}_r) d\Gamma = \int_{\Gamma_{pr}} \varphi_A p_r d\Gamma, \\ \sum_{s=1}^2 \sum_{B \in N_{qs}} a(\varphi_A \mathbf{e}_r, \varphi_B \mathbf{e}_s) q_{sB} &= \sum_{B \in N_{qs}} K_{AB}^{rs} q_{sB}. \end{aligned}$$

Assim a força é dada por

$$F_A^r = \int_{\Omega} \varphi_A f_r d\Omega + \int_{\Gamma_{pr}} \varphi_A p_r d\Gamma - \sum_{s=1}^2 \sum_{B \in N_{qs}} K_{AB}^{rs} q_{sB}. \quad (5.23)$$

De forma análoga ao capítulo anterior, f_r e p_r serão interpolados usando a função φ_A como interpolante. Logo,

$$f_r(x) = \sum_{A \in N - N_{qs}} \varphi_A(x) f_{rA} \quad (5.24)$$

$$p_r(x) = \sum_{A \in N_{ps}} \varphi_A(x) p_{rA} \quad (5.25)$$

onde $f_{rA} = f_r(\mathbf{x}_A)$ e $p_{rA} = p_r(\mathbf{x}_A)$, $r = 1, 2$. Substituindo em (5.23) obtemos

$$\begin{aligned} F_A^r &= \sum_{B \in N - N_{qs}} \int_{\Omega} (\varphi_A \varphi_B) f_{rB} d\Omega + \sum_{B \in N_{ps}} \int_{\Gamma_{ps}} (\varphi_A \varphi_B) p_{rB} d\Gamma - \sum_{s=1}^2 \sum_{B \in N_{qs}} K_{AB}^{rs} q_{sB} \\ &= \sum_{B \in N - N_{qs}} (\varphi_A, \varphi_B) f_{rB} + \sum_{B \in N_{ps}} (\varphi_A, \varphi_B)_{\Gamma} p_{rB} - \sum_{s=1}^2 \sum_{B \in N_{qs}} K_{AB}^{rs} q_{sB}. \end{aligned} \quad (5.26)$$

De (5.22) e (5.26) o sistema linear global é dado por

$$K_{AB}^{rs} d_{sB} = F_A^r. \quad (5.27)$$

Desde que

$$M = \text{eqn}[r, A] \text{ e } N = \text{eqn}[s, B]$$

o sistema linear pode ser escrito por

$$K_{MN} = \delta_{rk} \delta_{sl} \int_{\Omega} C_{ikjl} \frac{\partial \varphi_A}{\partial x_i} \frac{\partial \varphi_B}{\partial x_j} d\Omega = \int_{\Omega} C_{irjs} \frac{\partial \varphi_A}{\partial x_i} \frac{\partial \varphi_B}{\partial x_j} d\Omega, \quad (5.28)$$

$$F_M = \sum_{B \in N - N_{qs}} (\varphi_A, \varphi_B) f_{rB} + \sum_{B \in N_{ps}} (\varphi_A, \varphi_B)_{\Gamma} p_{rB} - \sum_{B \in N_{qs}} K_{AB}^{rs} q_{sB}. \quad (5.29)$$

Matriz Local e Força Local

Considere a discretização do domínio em elementos finitos Ω_e satisfazendo

$$\Omega = \left(\bigcup_{e=1}^{\text{Nel}} \overline{\Omega_e} \right)^{\circ} \quad \text{e} \quad \Omega_e \cap \Omega_k = \emptyset \quad \text{se} \quad e \neq k$$

Para cada elemento finito Ω_e temos a função base φ_A^e com suporte compacto em Ω_e . Dessa forma a matriz global e o vetor força podem ser obtidos em termos locais através da soma;

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

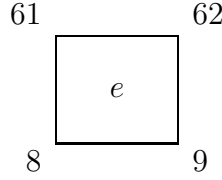
onde $K^e = [K_{MN}^e]$ e $F^e = [F_M^e]$ são dados por

$$K_{MN}^e = \delta_{rk} \delta_{sl} \int_{\Omega_e} C_{ikj\ell} \frac{\partial \varphi_A}{\partial x_i} \frac{\partial \varphi_B}{\partial x_j} d\Omega_e = \int_{\Omega_e} C_{irjs} \frac{\partial \varphi_A}{\partial x_i} \frac{\partial \varphi_B}{\partial x_j} d\Omega_e$$

$$F_M^e = \sum_{B \in N - N_{qs}} \int_{\Omega_e} \varphi_A \varphi_B f_B d\Omega_e + \sum_{B \in N_{ps}} \int_{\Gamma_p^e} \varphi_A \varphi_B p_{rB} d\Gamma - \sum_{B \in N_{qs}} K_{MN}^e q_{sB}$$

e $q_{sB} = q_s(x_B)$.

As matrizes K_{MN}^e e F_M^e têm ordem $\mathbf{Neq} \times \mathbf{Neq}$ e $\mathbf{Neq} \times 1$, respectivamente. Do suporte compacto de φ_A em Ω_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 $\mathbf{eqn}[i, A]$. Considere a seguinte tabela relacionado 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_{MN}^e e F_M^e para $M, N = \{5, 6, 7, 8, 47, 48, 49, 50\}$ são não necessariamente nulos. Assim, no lugar de armazenar a matriz K_{MN}^e e F_M^e de ordem $\mathbf{Neq} \times \mathbf{Neq}$ e $\mathbf{Neq} \times 1$ é conveniente armazenar apenas uma submatriz que denotaremos por K_{mn}^e de ordem 8×8 e F_m^e de ordem 8×1 . Observe que no caso escalar bidimensional do capítulo anterior, a matriz K_{MN}^e e o vetor F_M^e são de ordem 4×4 e 4×1 , respectivamente.

A matriz K_{mn}^e e o vetor força F_m^e serão chamados de matriz local e força local para os quais são introduzidos os nós locais \underline{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 4 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

$$A = \mathbf{NoLG}(e, a)$$

Para cada nó local a , define-se a função de interpolação local satisfazendo a condição:

$$\varphi_a^e(b) = \begin{cases} 1 & \text{se } a = b \\ 0 & \text{se } a \neq b \end{cases}$$

Utilizaremos, em particular, a função $\varphi_a^e(x, y)$ como um polinômio interpolante linear definida em (4.40).

Com a introdução da função de interpolação local φ_a^e podemos então definir a matriz local e o vetor força local por

$$K_{mn}^e = \int_{\Omega_e} C_{irjs} \frac{\partial \varphi_a^e}{\partial x_i} \frac{\partial \varphi_b^e}{\partial x_j} d\Omega_e \quad (5.30)$$

$$F_m^e = \sum_{b=1}^4 \int_{\Omega_e} \varphi_a^e \varphi_b^e f_{rb} d\Omega_e + \sum_{b=1}^4 \int_{\Gamma_p^e} \varphi_a^e \varphi_b^e p_{rb} d\Gamma - \sum_{b=1}^4 K_{mn}^e q_{sb} \quad (5.31)$$

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

$$(m, n) = (2(a-1) + r, 2(b-1) + s), \quad (5.32)$$

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_{mn}^e . Por exemplo, se $(a, b) = (1, 2)$, então, variando r e s entre 1 e 2, obtém-se os coeficientes:

$$\{K_{13}^e; K_{23}^e; K_{14}^e; K_{24}^e\}$$

De forma análoga para $(a, b) = (2, 4)$ temos

$$\{K_{37}^e; K_{38}^e; K_{47}^e; K_{48}^e\}.$$

Um outro procedimento fundamental é a forma como os coeficientes K_{mn}^e da matriz local são alocados na matriz global $K = [K_{MN}]$, quando variamos o elemento $e = 1, 2, \dots, \text{Nel}$, ou seja, como os coeficientes da matriz local contribuem para os coeficientes da matriz global. Para isso considere

$$P = \text{eqn}[i, A] \quad 1 \leq i \leq 2; \quad A \in N - N_{qi},$$

onde P é o número da equação. Temos

$$P = \text{eqn}[i, \text{NoLG}(a, e)]$$

Como os coeficientes da matriz local K_{mn}^e estão associados aos nós locais pela relação (5.32), podemos associar para cada elemento e , os coeficientes da matriz global K_{MN} com os coeficientes K_{mn}^e 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

$$\begin{aligned}\text{NoLG}(1, e) &= 8 \\ \text{NoLG}(2, e) &= 9 \\ \text{NoLG}(3, e) &= 61 \\ \text{NoLG}(4, e) &= 62\end{aligned}$$

Localmente temos a matriz K_{mn}^e de ordem 8×8 .

Então a contribuição da matriz local na matriz global é dada por

$$\begin{aligned}K_{55} &\leftarrow K_{55} + K_{11}^e \\ K_{56} &\leftarrow K_{56} + K_{12}^e \\ K_{57} &\leftarrow K_{57} + K_{13}^e \\ K_{58} &\leftarrow K_{58} + K_{14}^e \\ K_{5,47} &\leftarrow K_{5,47} + K_{15}^e \\ \vdots &\quad \quad \quad \vdots \\ K_{5,50} &\leftarrow K_{5,50} + K_{18}^e \\ K_{6,6} &\leftarrow K_{6,6} + K_{22}^e \\ K_{6,7} &\leftarrow K_{6,7} + K_{2,3}^e \\ \vdots &\quad \quad \quad \vdots \\ K_{6,50} &\leftarrow K_{6,50} + K_{28}^e \\ \vdots &\quad \quad \quad \vdots \\ K_{49,49} &\leftarrow K_{49,49} + K_{77}^e \\ K_{49,50} &\leftarrow K_{49,50} + K_{78}^e \\ K_{50,50} &\leftarrow K_{50,50} + K_{88}^e\end{aligned}$$

e mais os elementos simétricos da matriz.

Variando $e = 1, 2, \dots, \text{Nel}$, obtém-se todos os coeficientes K_{MN} da matriz global K , onde

$$M = \text{eqn}[i, \text{NoLG}(a, e)].$$

De forma análoga para a Força Global F ,

$$\begin{aligned}F_5 &\leftarrow F_5 + F_1^e \\ F_6 &\leftarrow F_6 + F_2^e \\ \vdots &\quad \quad \quad \vdots \\ F_{50} &\leftarrow F_{50} + F_8^e\end{aligned}$$

Cálculo de Matriz Local

Para calcular a matriz local utilizaremos o mesmo procedimento do capítulo anterior, isto é, faz-se a transformação isoparamétrica entre o elemento finito Ω_e para o quadrado biunitário Ω_b e então utiliza-se as funções base φ_a^e definida em (4.41) por

$$\varphi_a(\xi, \eta) = \frac{1}{4}(1 + \xi_a \xi)(1 + \eta_a \eta), \quad a = 1, 2, 3, 4. \quad (5.33)$$

Segue que o gradiente é dado por

$$\nabla \varphi_a(\xi, \eta) = \frac{1}{4}(\xi_a(1 + \eta_a \eta), \eta_a(1 + \xi_a \xi))$$

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 que

$$\frac{\partial \varphi_a}{\partial x_k} = \frac{\partial \varphi_a}{\partial \xi_i} \frac{\partial \xi_i}{\partial x_k}$$

Substituindo na matriz local (5.30), obtemos que:

$$K_{mn}^e = \int_{\Omega_e} C_{irjs} \frac{\partial \varphi_a^e}{\partial \xi_k} \frac{\partial \xi_k}{\partial x_i} \frac{\partial \varphi_b^e}{\partial \xi_\ell} \frac{\partial \xi_\ell}{\partial x_j} d\Omega_e = C_{irjs}^e \int_{\Omega_b} \left[\frac{\partial \varphi_a^e}{\partial \xi_k} \frac{\partial \xi_k}{\partial x_i} \frac{\partial \varphi_b^e}{\partial \xi_\ell} \frac{\partial \xi_\ell}{\partial x_j} \right] |J| d\Omega_b$$

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 para Ω_b , definido por

$$J = \det \begin{bmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} \end{bmatrix}.$$

Definindo a matriz

$$\hat{Q}_{abij}^e = \int_{\Omega_b} \frac{\partial \varphi_a^e}{\partial \xi_k} \frac{\partial \varphi_b^e}{\partial \xi_\ell} \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_\ell}{\partial x_j} d\Omega_b, \quad (5.34)$$

e considerando o Jacobiano constante para cada elemento Ω_e , podemos escrever a matriz local na forma simplificada

$$K_{mn}^e = |J^e| C_{irjs}^e \hat{Q}_{abij}^e$$

Em particular, se Ω_e é um retângulo temos,

$$\frac{\partial \xi_i}{\partial x_k} = \begin{cases} \frac{2}{dx_i} & \text{se } i = k \\ 0 & \text{se } i \neq k \end{cases}$$

Substituindo em (5.35), obtemos

$$\widehat{Q}_{abij}^e = \frac{4}{dx_i dx_j} Q_{abij}^e,$$

onde

$$Q_{abij}^e = \int_{\Omega_b} \frac{\partial \varphi_a^e}{\partial \xi_k} \frac{\partial \varphi_b^e}{\partial \xi_\ell} d\Omega_b. \quad (5.36)$$

Além disso, o Jacobiano da transformação é dado por

$$J^e = \frac{1}{4} dx^e dy^e \quad (5.37)$$

onde $dx^e = (x_2^e - x_1^e)$ e $dy^e = (y_2^e - y_1^e)$. Assim, a matriz local pode ser calculada por

$$K_{mn}^e = |J^e| C_{irjs}^e \frac{4}{dx_i dx_j} Q_{abij}^e \quad (5.38)$$

Observe que os únicos termos dependentes do elemento e são o Jacobiano e o coeficiente de elasticidade C_{ijkl} . Em (5.38) 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 (4.53) e está definida na subrotina `LocalSystem`, onde a integral (5.36) foi calculada pela quadratura Gaussiana.

Cálculo da Força Local F_m^e

Por (5.31) a força local é dada por

$$F_m^e = f_m^e + p_m^e - q_m^e, \quad \text{para } m = 2(a-1) + r \quad (5.39)$$

onde

$$f_m^e = \sum_{b=1}^4 \int_{\Omega^e} (\varphi_a^e \varphi_b^e) f_{br}^e d\Omega_e \quad (5.40)$$

$$p_m^e = \sum_{b=1}^4 \int_{\Gamma_p^e} (\varphi_a^e \varphi_b^e) p_{br}^e d\Gamma \quad (5.41)$$

$$q_m^e = \sum_{b=1}^4 K_{mn}^e q_{br}^e \quad (5.42)$$

As igualdades acima são válidas para todo nó local $b \in \Omega_e$, tal que, o correspondente nó global $B = \text{NoLG}(e, b)$ 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 (4.68). Assim, usando o mesmo procedimento, obtém-se

$$f_m^e = \sum_{b=1}^4 \int_{\Omega^e} (\varphi_a^e \varphi_b^e) f_{br}^e d\Omega_e = J^e \sum_{b=1}^4 f_{br}^e \int_{\Omega_b} (\varphi_a^e \varphi_b^e) d\Omega_b = J^e \sum_{b=1}^e f_{br}^e Q_{ab} \quad (5.43)$$

onde

$$Q_{ab} = \int_{\Omega_b} \varphi_a^e \varphi_b^e d\Omega_b$$

$$f_{br}^e = f(x_{br}^e) = f_r(x_b^e) = f_r(B), \quad B = \text{NoLG}(e, b), \quad r = 1, 2$$

com $m = 2(b - 1) + r$.

Observe que a matrix Q_{ab} , calculada em (4.60), é independente do elemento e .

Considere a seguinte malha dada a seguir:

13	14	15	16
9	10	11	12
5	6	7	8
1	2	3	4

$$\begin{aligned} \Gamma_{q1} &= \{1, 2, 5, 8, 13, 16\} \\ \Gamma_{q2} &= \{2, 3, 4, 8, 13, 16\} \\ \Gamma_{p1} &= \{3, 4, 9, 12, 14, 15\} \\ \Gamma_{p2} &= \{1, 5, 9, 12, 14, 15\} \end{aligned}$$

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: $\text{eqn}[i, A]$

$i \backslash A$	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 $\text{Neq} = 20$ 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: $A = \text{NoLG}(e, a)$ e 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 (5.39) que a força local é calculada por

$$F_m^e = f_m^e + p_m^e - q_m^e.$$

Vamos estabelecer o cálculo de F_m^e para cada $e = 1, 2, \dots, 9$, localmente. Com efeito,

- (1) Seja $e = 5$. Então, os nós locais $\{1, 2, 3, 4\}$ estão associados aos nós globais $\{6, 7, 11, 10\}$ através da relação: $A = \text{NoLG}(e, a) = \text{NoLG}(5, a)$. Para este elemento não existe nó global $A \in \Gamma_{pi}$ ou $A \in \Gamma_{qi}$. Assim

$$F_m^5 = f_m^5, \quad \text{se } m = 1, 2, \dots, 8,$$

pois $p_m^5 = q_m^5 = 0$.

- (2) Seja $e = 8$. Neste caso, 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

$$\begin{aligned} F_m^8 &= f_m^8 & \text{se } m &= 1, 2, 3, 4 \\ F_m^8 &= f_m^8 + p_m^8 & \text{se } m &= 5, 6, 7, 8, \end{aligned}$$

pois $(r, 1)$ e $(r, 2) \notin \Gamma_{pr} \cup \Gamma_{qr}$.

- (3) Seja $e = 2$. 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 $(r, \text{NoLG}(e, a_1)) \in \Gamma_{q1} \cap \Gamma_{q2}$ e $(2, \text{NoLG}(2, 2)) \in \Gamma_{q2}$.

Como

$$q_{br}^e = q_r(x_b^e)$$

onde x_b^e são os nós prescritos, tem-se

$$\begin{aligned} F_1^2 &= f_1^2 - q_1^2 \\ F_2^2 &= f_2^2 - q_2^2 \\ F_3^2 &= f_3^2 \\ F_4^2 &= f_4^2 - q_4^2 \\ F_m^2 &= f_m^2; \quad \text{se } m = 5, 6, 7, 8 \end{aligned}$$

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 $A = \text{NoLG}(2, a) \notin \Gamma_{p1} \cup \Gamma_{p2} \Rightarrow p_r(x_b^3) = 0$.

(4) Seja $e = 6$. Neste caso, os componentes locais prescritos são

$$\begin{aligned} 8 &= \text{NoLG}(6, 2) \in \Gamma_{q1} \\ 8 &= \text{NoLG}(6, 2) \in \Gamma_{q2} \\ 12 &= \text{NoLG}(6, 3) \in \Gamma_{p1} \\ 12 &= \text{NoLG}(6, 3) \in \Gamma_{p2} \end{aligned}$$

Assim a contribuição desse elemento é dado por

$$\begin{aligned} F_m^6 &= f_m^6 \quad \text{se } m = 1, 2, 7, 8 \\ F_m^6 &= f_m^6 - q_m^6 \quad \text{se } m = 3, 4 \\ F_m^6 &= f_m^6 + p_m^6 \quad \text{se } m = 5, 6. \end{aligned}$$

(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 $e = 9$

$$\begin{aligned} F_m^9 &= f_m^9 & \text{se } m = 1, 2 \\ F_m^9 &= f_m^9 - q_m^9 & \text{se } m = 5, 6 \\ F_m^9 &= f_m^9 + p_m^9 & \text{se } m = 3, 4, 7, 8. \end{aligned}$$

Cálculo de q_m^e

Por definição,

$$q_m^e = \sum_{b=1}^4 K_{mn}^e q_{br}^e$$

e considere

$$B = \text{NoLG}(e, b).$$

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

$$p_m^e = \sum_{b=1}^4 \int_{\Gamma_{pr}^e} (\varphi_a^e \varphi_b^e) p_{br}^e d\Gamma = \sum_{b=1}^4 M_{ab} p_{br}^e \quad (5.44)$$

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 (4.70).

Queremos identificar todos os nós globais $A \in \Gamma_p$ para o qual 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, para cada componente $r = 1$ ou $r = 2$ fixa. 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 na direção x e na direção y respectivamente. Assim, para cada componente, utiliza-se as matrizes definidas em (4.72) e (4.73).

Então,

$$M_{ab} = \begin{cases} \frac{1}{3} dx^e & \text{se } a = b \\ \frac{1}{6} dx^e & \text{se } a \neq b \end{cases} \quad (5.45)$$

ou

$$M_{ab} = \begin{cases} \frac{1}{3} dy^e & \text{se } a = b \\ \frac{1}{6} dy^e & \text{se } a \neq b \end{cases} \quad (5.46)$$

Para exemplificar, considere a malha na pag. 182.

1. Para o elemento $e = 3$, temos os nós globais $\{3, 4, 7, 8\}$. O único nó global em Γ_p é o nó $A = 4 = \text{NoLG}(e, a) = \text{NoLG}(3, 2)$. 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 $m = 3(a = 2, r = 1)$. Mas

$$p_m^e = \sum_{b=1}^4 M_{ab} p_{br}^e,$$

logo,

$$p_3^3 = M_{22} p_{21}^3.$$

Dependendo do tipo do nó global $A = 4$ teremos

$$M_{22} = \frac{dy^e}{3} \quad \text{ou} \quad M_{22} = \frac{dx^e}{3}.$$

2. Para o elemento $e = 1$, de forma análoga, temos como único nó prescrito o nó $A = 5 \in \Gamma_{p2}$ ou seja $r = 2$. Sendo $5 = \text{NoLG}(1, 4)$ então apenas o valor de p_{42}^1 está definido e para os demais valores, $p_{br}^1 = 0$. Então, para $r = 2$ e $a = 4$, temos $m = 8$. Logo

$$p_8^1 = M_{44} p_{42}^1$$

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 $r = 2$. Desde que $9 = \text{NoLG}(7, 1)$ e $12 = \text{NoLG}(7, 3)$, então $p_{br}^7 = p_r(x_b^7)$ não são necessariamente nulos para $r = 1, 2$ e $b = 1, 3$. Assim, da relação $m = 2(b - 1) + r$, somente p_m^7 para $m = 1, 2, 5, 6$ não são necessariamente nulas. Logo para $r = 1$ e $b = 1$ tem-se

$$p_1^7 = \sum_{b=1}^4 M_{ab} p_1(x_b^7) = M_{11} p_1(x_1^7) + M_{31} p_1(x_3^7)$$

para $r = 2$ e $b = 1$

$$p_2^7 = M_{11} p_2(x_1^7) + M_{31} p_2(x_3^7)$$

para $r = 1$ e $b = 3$:

$$p_5^7 = M_{13} p_1(x_3^7) + M_{33} p_1(x_3^7)$$

para $r = 2$ e $b = 3$:

$$p_6^7 = M_{13}p_2(x_3^7) + M_{33}p_2(x_3^7)$$

onde M_{ab} está definida em (5.46).

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_m^e e q_m^e

Os valores, p_b^r e q_b^r 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 .

Identificados todos os nós da fronteira, então a subrotina **CondFront** vai classificar as componentes horizontais e verticais de cada nó, como sendo de Neumann ou Dirichlet, a partir da interferência do operador do programa nessa subrotina. Assim por exemplo, suponhamos que na fronteira Γ_j as componentes vertical e horizontal são do tipo de Neumann, então 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. Suponha que é conhecida o valor $u_x(r, A)$. Então basta colocar na subrotina **CondFront** o que segue:

$$\mathbf{Bv}[j][i][r] = u_x(r, A), \quad r = 0, 1$$

onde j representam a fronteira Γ_j , i representa todos os nós da fronteira.

Por outro lado, se $A \in N_{qi}$ um nó global da fronteira do tipo de 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:

$$\mathbf{typ}[A][r] = 1; \quad u[A][r] = u_r(A)$$

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:

$$\mathbf{Bv}[j][i][0] = u_x(0, A)$$

$$\mathbf{typ}[A][1] = 1; \quad u[A][1] = u_1(A)$$

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 subrotina **TractionBoundary**, tem a função de inserir apropriadamente a contribuição dos valores de fronteira de Neumann na força, usando as matrizes definidas em (4.72) e (4.73), ou seja calcular o valor de p_a^e , em $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$, quando essas fronteiras são do tipo de Neumann.

Construção da Matriz Global

Por definição a matriz global é obtida por

$$K = \sum_{e=1}^{N_{el}} K^e.$$

A ordem da matriz K é $\mathbf{Neq} \times \mathbf{Neq}$. Cada matriz local K^e tem ordem 8×8 e a forma de alocação de cada uma das matrizes K^e , para $e = 1, 2, \dots, \mathbf{Nel}$ 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

$$A = \mathbf{NoLG}(e, a)$$

- (2) Identificação entre o nó global e a sua correspondente equação do sistema dado por

$$P = \mathbf{eqn}[i, A] \quad i = 1, 2$$

Com novamente a malha dada na pag. 182. Os nós globais na 1ª etapa são dados na Tabela 5.1

$a \backslash e$	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: $A = \mathbf{NoLG}(e, a)$

Na 2ª etapa temos a relação entre os nós globais A e o número de equações dada pela Tabela 5.2:

$i \backslash A$	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: $P = \text{eqn}[i, A]$

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. Aplicaremos a seguinte relação:

$$j = \text{eqn}[i, A] = \text{eqn}[i, \text{NoLG}(e, a)]. \quad (5.47)$$

Se $j = 0$ então (i, A) não gera equação e portanto não contribui para os coeficientes da matriz K_{MN} .

Se $j \neq 0$, (i, A) gera a j -ésima equação e portanto, contribui para j -ésima linha da matriz global K_{MN} , onde $j \leq M$. Relembramos que os coeficientes da matriz local K_{mn}^e satisfazem a relação:

$$\begin{aligned} m &= 2(a - 1) + i & a, b &= 1, 2, 3, 4 \\ n &= 2(b - 1) + i & i &= 1, 2 \end{aligned} \quad (5.48)$$

Temos então que;

1. $e = 1$: Os nós globais desse elemento são $A = 1, 2, 6, 5$. Observando a Tabela 5.2 e a relação (5.48) temos que os coeficientes da matriz local K_{mn}^e não necessariamente nulos são

$$\{K_{22}^1, K_{25}^1, K_{26}^1, K_{28}^1, K_{55}^1, K_{56}^1, K_{58}^1, K_{66}^1, K_{68}^1, K_{88}^1\}$$

e também os coeficientes 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_{22}^1	K_{28}^1	K_{25}^1	K_{26}^1	K_{88}^1	K_{85}^1	K_{86}^1	K_{55}^1	K_{56}^1	K_{66}^1

onde em cada coluna o segundo membro representa a contribuição local para o coeficiente global do primeiro membro. Por exemplo, na primeira coluna, temos

$$K_{11} \leftarrow K_{11} + K_{22}^1$$

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 (5.48) concluímos que os coeficientes da matriz local que irão contribuir para a matriz global são

$$\begin{cases} K_{33}^2; K_{35}^2; K_{36}^2; K_{37}^2; K_{38}^2; K_{55}^2; K_{56}^2; K_{57}^2; K_{58}^2; \\ K_{66}^2; K_{67}^2; K_{68}^2; K_{77}^2; K_{78}^2; K_{88}^2; + \text{simétricos} \end{cases}$$

Dessa forma temos

K_{22+}	K_{25+}	K_{26+}	K_{27+}	K_{28+}	K_{55+}	K_{56+}	K_{57+}
K_{33}^2	K_{37}^2	K_{38}^2	K_{35}^2	K_{36}^2	K_{77}^2	K_{78}^2	K_{75}^2
K_{58+}	K_{66+}	K_{67+}	K_{68+}	K_{77+}	K_{78+}	K_{88+}	
K_{76}^2	K_{88}^2	K_{85}^2	K_{86}^2	K_{55}^2	K_{56}^2	K_{66}^2	

Fazendo $e = 3, 4, \dots, 9$ 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_{MN} para o exemplo anterior de ordem (20×20) omitindo-se os coeficientes simétricos podem ser representada pela Fig. 5.1, onde * são coeficientes não necessariamente nulos.

Construção da Força Global

A força global é definida por

$$F = \sum_{e=1}^{N_{el}} F_m^e,$$

onde $F = [F_M]$ com $M = 1, 2, \dots, N_{eq}$.

Usando novamente a malha anterior, analisaremos a contribuição da força local F_m^e para a força global F_M . A força local é dada por

$$F_m^e = f_m^e + p_m^e - q_m^e, \quad m = 2(a-1) + r, \quad r = 1, 2 \quad a = 1, 2, 3, 4$$

Para $e = 1, 2, 3, \dots, 9$ os valores de F_m^e 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

1. $e = 1$: Usando as Tabelas 5.1, 5.2 e a relação (5.47) tem-se

$$\begin{aligned} F_1 &\leftarrow F_1 + F_2^1 \\ F_4 &\leftarrow F_4 + F_8^1 \\ F_5 &\leftarrow F_5 + F_5^1 \\ F_6 &\leftarrow F_6 + F_6^1 \end{aligned}$$

$$\begin{bmatrix}
 * & 0 & 0 & * & * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & * & * & 0 & * & * & * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & * & 0 & 0 & 0 & * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & * & * & * & 0 & 0 & * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & * & * & * & * & * & * & * & * & * & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & * & * & * & * & * & * & * & * & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & * & * & 0 & 0 & * & * & * & * & * & 0 & 0 & 0 & 0 \\
 & & & & & & & * & 0 & 0 & * & * & * & * & * & 0 & 0 & 0 & 0 \\
 & & & & & & & & * & * & * & * & 0 & 0 & 0 & 0 & * & * & 0 & 0 \\
 & & & & & & & & & * & * & * & 0 & 0 & 0 & 0 & * & * & 0 & 0 \\
 & & & & & & & & & & * & * & * & 0 & 0 & * & * & * & * \\
 & & & & & & & & & & & * & * & 0 & 0 & * & * & * & * \\
 & & & & & & & & & & & & * & * & * & * & * & * & * \\
 & & & & & & & & & & & & & * & * & * & * & * & * \\
 & & & & & & & & & & & & & & * & * & 0 & 0 & * & * \\
 & & & & & & & & & & & & & & & * & 0 & 0 & * & * \\
 & & & & & & & & & & & & & & & & * & * & * & * \\
 & & & & & & & & & & & & & & & & & * & * & * \\
 & & & & & & & & & & & & & & & & & & * & * \\
 & & & & & & & & & & & & & & & & & & & * \\
 & *
 \end{bmatrix}$$

Figura 5.1: Matriz K_{MN}

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 (5.47) temos respectivamente a força dada por

$$\begin{aligned}
 F_5 &\leftarrow F_5 + F_1^5 \\
 F_6 &\leftarrow F_6 + F_2^5 \\
 F_7 &\leftarrow F_7 + F_3^5 \\
 F_8 &\leftarrow F_8 + F_4^5 \\
 F_{13} &\leftarrow F_{13} + F_5^5 \\
 F_{14} &\leftarrow F_{14} + F_6^5 \\
 F_{11} &\leftarrow F_{11} + F_7^5 \\
 F_{12} &\leftarrow F_{12} + F_8^5
 \end{aligned}$$

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 $Kd = F$ utilizaremos o algoritmo de Crout [9]. Basicamente a única diferença é o tamanho da matriz, desde que neste caso a função vetorial $u = (u_1, u_2)$ tem as componentes horizontais e verticais. A partir da solução do sistema linear,

$$d = \{d_N\}$$

onde $N = \text{eqn}(j, B)$, $j = 1, 2$, $B = 1, 2, \dots, \text{Neq}$.

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 $\mathbf{u}^h(\mathbf{x})$ a solução numérica e $\mathbf{u}(\mathbf{x})$ a solução exata. Determinamos os seguintes erros na norma $L^2(\Omega)$ e na seminorma $H^1(\Omega)$:

$$\|E\|_0 = \|\mathbf{u} - \mathbf{u}^h\|_0, \quad \|E\|_1 = \|\nabla \mathbf{u} - \nabla \mathbf{u}^h\|_0, \quad (5.49)$$

e os percentagens de erros:

$$E_0\% = \frac{\|E\|_0}{\|\mathbf{u}\|_0} \times 100, \quad E_1\% = \frac{\|E\|_1}{\|\nabla \mathbf{u}\|_0} \times 100. \quad (5.50)$$

5.4 Exemplos Numéricos

A matriz local K_{mn}^e é definida por

$$K_{mn}^e = \delta_{rk} \delta_{sl} \frac{4}{d\xi_i d\xi_j} J^e C_{ikj\ell}^e Q_{abij} \quad (5.51)$$

onde

$$Q_{abij} = \int_{\Omega_b} \frac{\partial \varphi_a}{\partial \xi_i} \frac{\partial \varphi_b}{\partial \xi_j} d\Omega_b$$

$$\begin{aligned} d\xi_1 &= \xi_2 - \xi_1 = 2 \\ d\xi_2 &= \eta_2 - \eta_1 = 2 \end{aligned}$$

O único termo a ser definido em (5.51) é o coeficiente de elasticidade $\{C_{ikj\ell}\}$ que depende da posição. Como $1 \leq i, k, j \leq \ell \leq 2$ então devemos conhecer os 16 coeficientes.

Em particular, no exemplo numérico vamos considerar o corpo isotrópico. Neste caso o coeficiente de elasticidade pode ser escrito por

$$C_{ijkl}(x) = \mu(x)(\delta_{ik}\delta_{j\ell} + \delta_{i\ell}\delta_{jk}) + \lambda(x)\delta_{ij}\delta_{k\ell} \quad (5.52)$$

onde μ e λ são conhecidos como parâmetros de Lamé. Os coeficientes não necessariamente nulos são

$$\begin{aligned} C_{1111} &= C_{2222} = \lambda + 2\mu \\ C_{1122} &= C_{2211} = \lambda \\ C_{1212} &= C_{1221} = C_{2112} = C_{2121} = \mu \end{aligned} \quad (5.53)$$

Supondo que o corpo é homogêneo então C_{ijkl} não depende de x , assim podemos definir os coeficientes na forma

$$\left\{ \begin{array}{l} C_{1111}, C_{1112}, C_{1121}, C_{1122}, C_{1211}, C_{1212}, C_{1221}, C_{1222}, \\ C_{2111}, C_{2112}, C_{2121}, C_{2122}, C_{2211}, C_{2212}, C_{2221}, C_{2222} \end{array} \right\} = \left\{ \begin{array}{l} \lambda + 2\mu, 0, 0, \lambda, 0, \mu, \mu, 0, \\ 0, \mu, \mu, 0, \lambda, 0, 0, \lambda + 2\mu \end{array} \right\}$$

Nestas condições o problema de elasticidade linear

$$-\frac{\partial}{\partial x_j} \left(C_{ijk\ell} \frac{\partial u_k}{\partial x_\ell} \right) = f_i, \quad \forall x \in \Omega$$

pode ser escrito na forma

$$-(\lambda + \mu) \nabla(\operatorname{div} u) - \mu \Delta u = f, \quad \forall x \in \Omega.$$

Na forma componente obtém-se

$$-(\lambda + \mu) \left\{ \frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_2}{\partial x \partial y} \right\} - \mu \left\{ \frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_1}{\partial y^2} \right\} = f_1 \quad (5.54)$$

$$-(\lambda + \mu) \left\{ \frac{\partial^2 u_1}{\partial x \partial y} + \frac{\partial^2 u_2}{\partial y^2} \right\} - \mu \left\{ \frac{\partial^2 u_2}{\partial x^2} + \frac{\partial^2 u_2}{\partial y^2} \right\} = f_2 \quad (5.55)$$

onde $u = (u_1, u_2)$ e $f = (f_1, f_2)$.

Exemplo 1: Fronteira de Dirichlet

Consideremos $\mu = 1$ e $\lambda = 1.5$ e a força horizontal e vertical dadas por

$$(f_1, f_2) = 2(\cos \pi x \cos \pi y, \sin \pi x \sin \pi y) \quad (5.56)$$

e o domínio $D = \{(x, y); 0 \leq y \leq 2; 0 \leq x \leq 1\}$.

Definimos as seguintes condições de fronteira no retângulo D :

$$\begin{aligned} u(x, 0) &= \frac{1}{\pi^2}(\cos \pi x, 0) \in \Gamma_1, & u(x, 2) &= \frac{1}{\pi^2}(\cos \pi x, 0) \in \Gamma_3, \\ u(0, y) &= \frac{1}{\pi^2}(\cos \pi y, 0) \in \Gamma_4, & u(1, y) &= \frac{1}{\pi^2}(-\cos \pi y, 0) \in \Gamma_2. \end{aligned} \quad (5.57)$$

A solução exata do problema (5.54) e (5.55) com essas condições de fronteira é dada por

$$u = (u_1, u_2) = \frac{1}{\pi^2}(\cos \pi x \cos \pi y, \sin \pi x \sin \pi y). \quad (5.58)$$

O procedimento para a determinação da solução numérica é a mesma feita no capítulo anterior. A construção da malha e a entrada de dados é similar, exceto as condições de fronteiras que são definidas para a posição vertical e horizontal, ou seja, definindo a divisão horizontal \mathbf{dx} por $1/\mathbf{Nelx}$ e a divisão vertical \mathbf{dy} por $2/\mathbf{Nely}$. O número total de nós globais $\mathbf{Nno} = (\mathbf{Nelx} + 1)(\mathbf{Nely} + 1)$. O número total de nós globais nas fronteiras Γ_j , $j = 1, 2, 3, 4$ é dada pelo número $\mathbf{Nbn}[j]$ e as fronteiras são representadas por $\mathbf{bdy}[j]$, dentro da subrotina **CondFront**. A inserção dos valores de fronteira de Dirichlet (5.57) em cada componente (no programa, devido a restrição da linguagem C, a componente horizontal é representada por $r = 0$ e vertical $r = 1$) do exemplo, é feito da seguinte forma:

$$\begin{aligned} \Gamma_1 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[1][\mathbf{i}], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[\mathbf{n}][0] = 0; \mathbf{u}[\mathbf{n}][0] = \cos(\pi * x[p.v[0]])/(\pi^2); \\ \mathbf{typ}[\mathbf{n}][1] = 1; \mathbf{u}[\mathbf{n}][1] = 0; \end{cases} \\ \Gamma_2 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[2][\mathbf{i}], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[\mathbf{n}][0] = 0; \mathbf{u}[\mathbf{n}][0] = -\cos(\pi * y[p.v[1]])/(\pi^2); \\ \mathbf{typ}[\mathbf{n}][1] = 1; \mathbf{u}[\mathbf{n}][1] = 0; \end{cases} \\ \Gamma_3 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[3][\mathbf{i}], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[\mathbf{n}][0] = 0; \mathbf{u}[\mathbf{n}][0] = \cos(\pi * x[p.v[0]])/(\pi^2); \\ \mathbf{typ}[\mathbf{n}][1] = 1; \mathbf{u}[\mathbf{n}][1] = 0; \end{cases} \\ \Gamma_4 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[4][\mathbf{i}], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[\mathbf{n}][0] = 0; \mathbf{u}[\mathbf{n}][0] = \cos(\pi * y[p.v[1]])/(\pi^2); \\ \mathbf{typ}[\mathbf{n}][1] = 1; \mathbf{u}[\mathbf{n}][1] = 0; \end{cases} \end{aligned}$$

Γ_1 : Para $n = 1, 2, \dots, \text{Nelx} + 1$ e $r = 0, 1$

$$\text{typ}[n][r] = 1, \quad \text{u}[n][r] = \frac{1}{\pi^2}(\cos \pi x, 0)$$

Γ_3 : Para $n = \text{Nno} - \text{Nelx}, \dots, \text{Nno}$ e $r = 0, 1$

$$\text{typ}[n][r] = 1, \quad \text{u}[n][r] = \frac{1}{\pi^2}(\cos \pi x, 0)$$

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 $n = \text{Nelx} + 1, \dots, \text{Nno}$ com incremento de $\text{Nelx} + 1$ e $r = 0, 1$

$$\text{typ}[n][r] = 1, \quad \text{u}[n][r] = \frac{1}{\pi^2}(-\cos \pi y, 0)$$

Γ_4 : Para $n = 1, \dots, \text{Nno} - \text{Nelx}$ com incremento de $\text{Nelx} + 1$ e $r = 0, 1$

$$\text{typ}[n][r] = 1, \quad \text{u}[n][r] = \frac{1}{\pi^2}(\cos \pi y, 0)$$

Em Γ_4 e Γ_2 estamos identificando, respectivamente, os nós prescritos na fronteira para $x = 0$ e $x = 1$. Por $\text{u}[n][r]$ 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 $\text{Neq} \times \text{Neq}$.

A força local F_m^e definida por

$$F_m^e = f_m^e + p_m^e - q_m^e$$

Como a fronteira é de 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 tem-se 3.321 nós, 3.200 elementos e 6.162 equações, ou seja, $\{\text{Nno} = 3.321; \text{Nel} = 3.200; \text{Neq} = 6.162\}$. Dessa forma a matriz global é de ordem 6.162×6.162 .

Exemplo 2: Fronteira de Neumann

Consideremos $\mu = 1$, $\lambda = 1.5$ e a força dada por

$$f_1(x, y) = f_2(x, y) = 4.5 \sin \pi x \sin \pi y - 2.5 \cos \pi x \cos \pi y.$$

Definimos o domínio $D = \{(x, y); 0 \leq y \leq 1; 0 \leq x \leq 1\}$. A fronteira de Neumann é dada por

$$\sigma_{ij}n_j = C_{ijkl}\frac{\partial u_k}{\partial x_\ell}n_j = p_i,$$

isto é, na fronteira a tração p_i é prescrita. Consideremos a tração p_i dada por

$$\begin{aligned} (p_1(x, 0), p_2(x, 0) = (p_1(x, 1), p_2(x, 1)) &= -\frac{1}{\pi} \sin \pi x (1, 3.5) \\ (p_1(0, y), p_2(0, y) = (p_1(1, y), p_2(1, y)) &= -\frac{1}{\pi} \sin \pi y (3.5, 1) \end{aligned} \quad (5.59)$$

A solução exata nestas condições é dada por

$$u_1(x, y) = u_2(x, y) = \frac{1}{\pi^2} \sin \pi x \sin \pi y. \quad (5.60)$$

A matriz local K_{mn}^e é a mesma do Exemplo 1, mas a matriz global K é de ordem $\text{Neq} \times \text{Neq}$, 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 (5.59) do exemplo na subrotina **CondFront**, o procedimento é dado a seguir:

$$\begin{aligned} \Gamma_1 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[1][\mathbf{i}], & p = \text{NoPos}(n) \\ \mathbf{Bv}[1][\mathbf{i}][0] = -\sin(\pi * x[p.v[0]])/\pi; \\ \mathbf{Bv}[1][\mathbf{i}][1] = -3.5 * \sin(\pi * x[p.v[0]])/\pi; \end{cases} \\ \Gamma_2 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[2][\mathbf{i}], & p = \text{NoPos}(n) \\ \mathbf{Bv}[2][\mathbf{i}][0] = -3.5 * \sin(\pi * y[p.v[1]])/\pi; \\ \mathbf{Bv}[2][\mathbf{i}][1] = -\sin(\pi * y[p.v[1]])/\pi; \end{cases} \\ \Gamma_3 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[3][\mathbf{i}], & p = \text{NoPos}(n) \\ \mathbf{Bv}[3][\mathbf{i}][0] = -\sin(\pi * x[p.v[0]])/\pi; \\ \mathbf{Bv}[3][\mathbf{i}][1] = -3.5 * \sin(\pi * x[p.v[0]])/\pi; \end{cases} \\ \Gamma_4 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[4][\mathbf{i}], & p = \text{NoPos}(n) \\ \mathbf{Bv}[4][\mathbf{i}][0] = -3.5 * \sin(\pi * y[p.v[1]])/\pi; \\ \mathbf{Bv}[4][\mathbf{i}][1] = -\sin(\pi * y[p.v[1]])/\pi; \end{cases} \end{aligned}$$

De forma similar ao problema do calor, o problema com fronteira completamente de 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, estamos escolhendo aquela solução cujo deslocamento e momento de deslocamento médios são nulos, descritos dentro da subrotina **Restriction**.

No procedimento 2, estamos fixando três componentes (duas verticais e uma horizontal), no final da subrotina **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:

```
typ[1][0] = 1;  u[1][0] = 0; (comp. horiz.do nó global  A=1)
typ[1][1] = 1;  u[1][1] = 0; (comp. vert. do nó global  A=1)
typ[Nelx + 1][1] = 1; 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:

```
typ[1][1] = 1;  u[1][1] = 0; (comp. horiz. do nó global  A=1)
typ[Nelx + 1][1] = 1; 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, como por exemplo àquela que passa pela origem, da seguinte forma

$$u[n][0] - u[1][0], \quad n = 2, 3, \dots; \quad u[1][0] = 0;$$

dentro da subrotina **GlobalSystem**.

Esses foram os procedimentos e os nós escolhidos no programa, mas evidentemente existem muitas formas diferentes de obter a solução.

Força Local

Em geral a força local é calculada por:

$$F_m^e = f_m^e + p_m^e - q_m^e, \quad m = 2(a - 1) + r, \quad r = 1, 2 \quad a = 1, 2, 3, 4$$

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), que significa:

$$F_1^1 = f_1^1 - q_1^1, \quad F_2^1 = f_2^1 - q_2^1 \\ F_4^{Nelx-1} = f_4^{Nelx-1} - q_4^{Nelx-1}$$

Note que para a componente horizontal ($r = 1$) do nó $A = Nelx$, temos

$$F_3^{Nelx-1} = f_3^{Nelx-1} - p_3^{Nelx-1}$$

Exemplo 3: Fronteira mista

Nas mesmas condições do Exemplo 2, consideremos os valores de \mathbf{u} dos nós da reta $y = 0$, $y = 1$ e pertencentes à malha prescritos ou seja

$$(u_1(x, 0), u_2(x, 0)) = (u_1(x, 1), u_2(x, 1)) = (0, 0),$$

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 pertencentes à reta $x = 0$ e $x = 1$ da malha, a tração \mathbf{p} é prescrita ou seja

$$(p_1(0, y), p_2(0, y)) = (p_1(1, y), p_2(1, y)) = -\frac{1}{\pi} \sin \pi y (3.5, 1),$$

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 dado por:

$$\begin{aligned} \Gamma_1 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[1][i], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[\mathbf{n}][0] = 0; \mathbf{u}[\mathbf{n}][0] = 0; \\ \mathbf{typ}[\mathbf{n}][1] = 1; \mathbf{u}[\mathbf{n}][1] = 0; \end{cases} \\ \Gamma_2 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[2][i], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[\mathbf{n}][0] = 0; \mathbf{u}[\mathbf{n}][0] = -3.5 * \sin \pi y / \pi; \\ \mathbf{typ}[\mathbf{n}][1] = 1; \mathbf{u}[\mathbf{n}][1] = -\sin \pi y / \pi; \end{cases} \\ \Gamma_3 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[3][i], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[\mathbf{n}][0] = 0; \mathbf{u}[\mathbf{n}][0] = 0; \\ \mathbf{typ}[\mathbf{n}][1] = 1; \mathbf{u}[\mathbf{n}][1] = 0; \end{cases} \\ \Gamma_4 &= \begin{cases} \mathbf{n} = \mathbf{bdy}[4][i], & p = \mathbf{NoPos}(n) \\ \mathbf{typ}[\mathbf{n}][0] = 0; \mathbf{u}[\mathbf{n}][0] = -3.5 * \sin \pi y / \pi; \\ \mathbf{typ}[\mathbf{n}][1] = 1; \mathbf{u}[\mathbf{n}][1] = -\sin \pi y / \pi; \end{cases} \end{aligned}$$

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,

$$u_1(x, y) = u_2(x, y) = \frac{1}{\pi^2} \sin(\pi x) \sin(\pi y).$$

Nesse exemplo, para uma malha 50×50 temos $\{\mathbf{Neq} = 4.998, \mathbf{Nno} = 2.601, \mathbf{Nel} = 2.500\}$

5.5 Unicidade: Problema de Neumann

Seja u_i uma solução do problema e v_i definido por

$$v_i = u_i + W_{ij}x_j + C_i, \quad (5.61)$$

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:

$$\int_{\Omega} v_1 d\Omega = 0, \quad \int_{\Omega} v_2 d\Omega = 0, \quad \int_{\Omega} (v_1 x_2 - v_2 x_1) d\Omega = 0. \quad (5.62)$$

Fisicamente, essas condições representam a escolha, dentre as infinitas soluções do problema, aquela que tenha o deslocamento médio e momento médio de deslocamento nulos. Estas condições 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

$$b_1 = \int_{\Omega} u_1 d\Omega, \quad b_2 = \int_{\Omega} u_2 d\Omega, \quad b_3 = \int_{\Omega} (u_1 x_2 - u_2 x_1) d\Omega,$$

$$I_0 = \int_{\Omega} d\Omega, \quad I_1 = \int_{\Omega} x_1 d\Omega, \quad I_2 = \int_{\Omega} x_2 d\Omega, \quad I_3 = \int_{\Omega} (x_1^2 + x_2^2) d\Omega.$$

Resolvendo o sistema obtemos os valores da rotação θ e da translação C_1, C_2 dados por

$$\begin{bmatrix} \theta \\ C_1 \\ C_2 \end{bmatrix} = \frac{1}{D} \begin{bmatrix} -I_0 I_2 & I_0 I_1 & I_0^2 \\ I_0 I_3 - I_1^2 & -I_1 I_2 & -I_0 I_2 \\ -I_1 I_2 & I_0 I_3 - I_2^2 & I_0 I_1 \end{bmatrix} \begin{bmatrix} -b_1 \\ -b_2 \\ -b_3 \end{bmatrix}, \quad (5.63)$$

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 (5.62) podemos impor as condições tal que as duas soluções tenham os mesmos valores médios de deslocamento e seu momento, ou sejam

$$\begin{aligned} \int_{\Omega} v_1 d\Omega &= \int_{\Omega} u_1^{\text{ex}} d\Omega, \\ \int_{\Omega} v_2 d\Omega &= \int_{\Omega} u_2^{\text{ex}} d\Omega, \\ \int_{\Omega} (v_1 x_2 - v_2 x_1) d\Omega &= \int_{\Omega} (u_1^{\text{ex}} x_2 - u_2^{\text{ex}} x_1) d\Omega. \end{aligned} \quad (5.64)$$

Assim, a partir de uma solução numérica u_i obtemos a solução procurada pela determinação da rotação e translação desejadas para a comparação:

$$\begin{bmatrix} \theta \\ C_1 \\ C_2 \end{bmatrix} = \frac{1}{D} \begin{bmatrix} -I_0 I_2 & I_0 I_1 & I_0^2 \\ I_0 I_3 - I_1^2 & -I_1 I_2 & -I_0 I_2 \\ -I_1 I_2 & I_0 I_3 - I_2^2 & I_0 I_1 \end{bmatrix} \begin{bmatrix} e_1 - b_1 \\ e_2 - b_2 \\ e_3 - b_3 \end{bmatrix}, \quad (5.65)$$

onde

$$e_1 = \int_{\Omega} u_1^{\text{ex}} d\Omega, \quad e_2 = \int_{\Omega} u_2^{\text{ex}} d\Omega, \quad e_3 = \int_{\Omega} (u_1^{\text{ex}} x_2 - u_2^{\text{ex}} x_1) d\Omega.$$

A solução v_i calculada pela equação (5.61) é a solução procurada que satisfaz as condições (5.64). 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 (5.65) tendam a 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 (5.62) ou (5.64) envolvem 3 graus de liberdade caracterizada pelas constantes θ , C_1 e C_2 . Podemos prefixar 3 valores de deslocamento, ou sejam u_1 e u_2 de um nó e uma componente, por exemplo u_1 , de um outro nó, e consequentemente a matriz rigidez será 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 assim uma infinidade de soluções que se diferenciam somente pela translação no eixo-x, nesse caso, 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 desse contexto foi tomado $c = 0$, ou seja, dentro da subrotina **GlobalSystem** pode ser feita a seguinte translação :

Para ($n = 2; n \leq Nno; n++$)

$u[n][0] = 0; u[1][0] = 0$; (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 (5.62). 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, $\{1, 3, 2\}$.

5.6 Resultados Numéricos

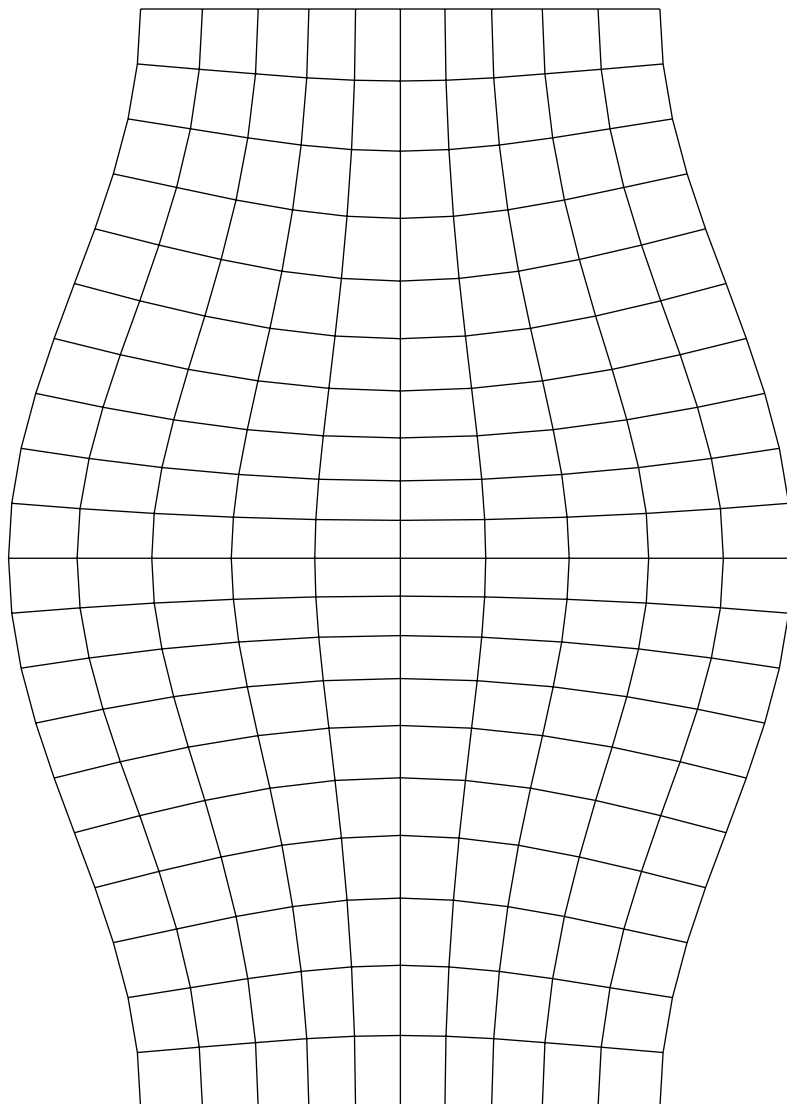
Para os exemplos, apresentamos os erros e percentagens de erros na norma $L^2(\Omega)$ e na seminorma $H^1(\Omega)$ e os gráficos da malha deformada do domínio. A posição deformada (x', y') do nó na posição (x, y) da malha é dada por

$$(x', y') = (x + u_1^h(x), y + u_2^h(x)),$$

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 $\Omega = [0, 1] \times [0, 2]$

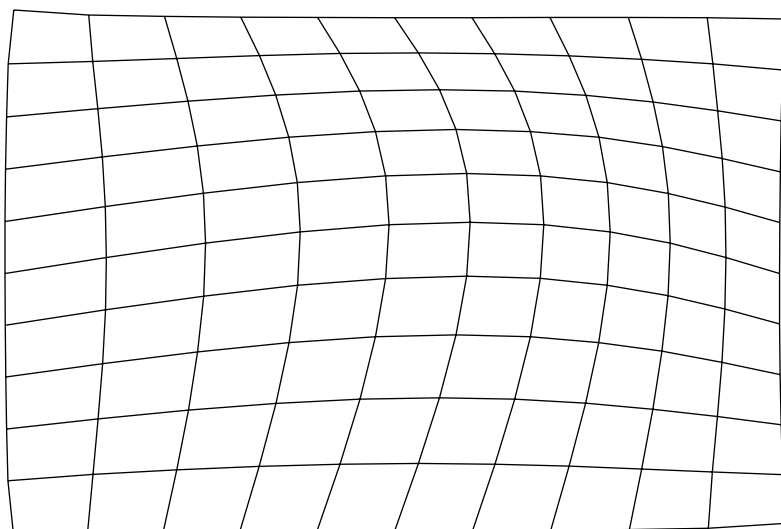


Erros numéricos

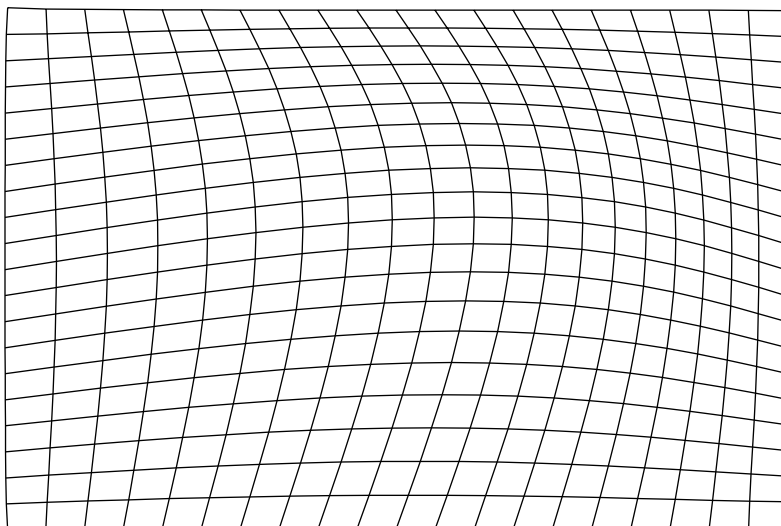
Malha	$\ E\ _0$	$\ E\ _1$	$E_0\%$	$E_1\%$
10×20	7.24×10^{-4}	3.62×10^{-3}	0.73	0.82
20×40	1.85×10^{-4}	9.28×10^{-4}	0.18	0.21
40×80	5.15×10^{-5}	2.59×10^{-4}	0.05	0.06

Exemplo 2: Problema de Neumann

Malha deformada (10×10) de $\Omega = [0, 1] \times [0, 1]$



Malha deformada (20×20) de $\Omega = [0, 1] \times [0, 1]$

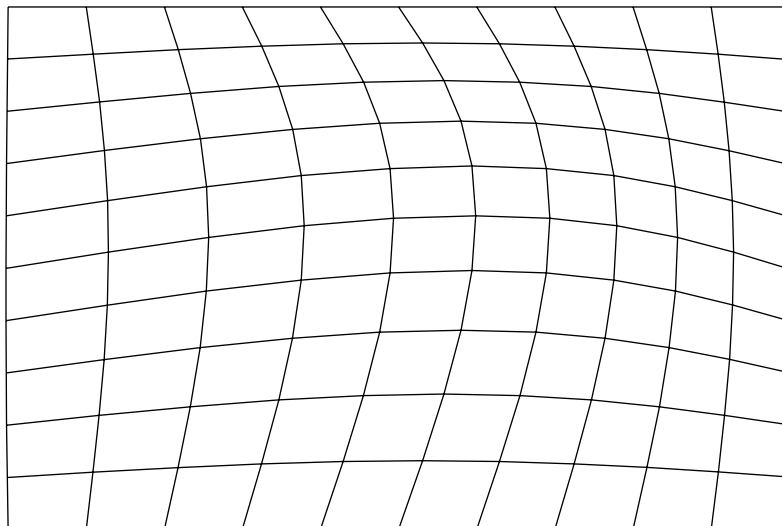


Erros numéricos: Procedimento 3 (Dois pontos fixos+translação)

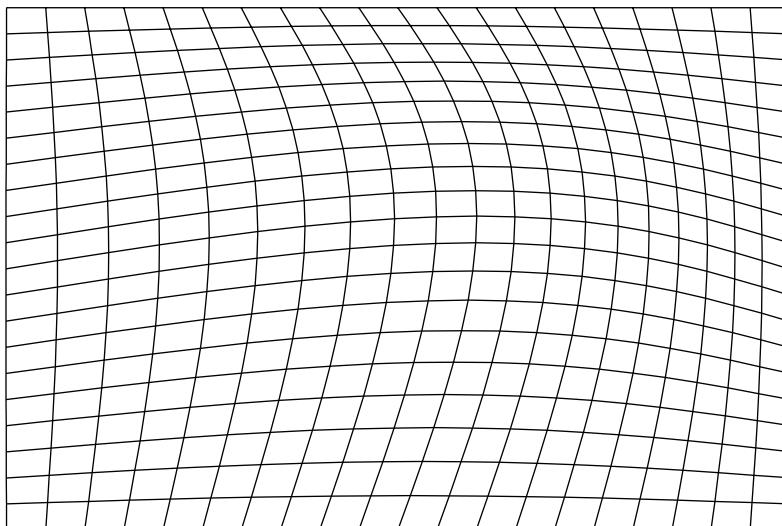
Malha	$\ E\ _0$	$\ E\ _1$	$E_0\%$	$E_1\%$
10×10	1.12×10^{-2}	1.37×10^{-2}	16.07	4.39
20×20	3.68×10^{-3}	4.41×10^{-3}	5.21	1.39
50×50	1.14×10^{-3}	1.43×10^{-3}	1.60	0.45

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

5.7 Exercícios

No Exemplo 1, Exemplo 2 e Exemplo 3 deste capítulo, o corpo considerado é isotrópico. Se o corpo não é isotrópico a relação (5.53) não é válida. Considere que o tensor de elasticidade C_{ijkl} seja caracterizada pelos seguintes coeficientes não nulos:

i - Isotropia Transversal:

$$C_{1111} = C_{2222} = C_1, \quad C_{1122} = C_2, \quad C_{1133} = C_3, \quad C_{3333} = C_4,$$

$$C_{1212} = C_1 - C_2, \quad C_{2323} = C_{3131} = C_5,$$

$$\text{onde } C_1 = 3.5, \quad C_2 = 1.5, \quad C_3 = 1.25, \quad C_4 = 2.5, \quad \text{e } C_5 = 1.0.$$

ii - Simetria cúbica:

$$C_{1111} = C_{2222} = C_{3333} = C_1, \quad C_{1122} = C_{1133} = C_{2233} = C_2,$$

$$C_{1212} = C_{2323} = C_{3131} = C_3,$$

$$\text{onde } C_1 = 3.5, \quad C_2 = 1.5 \quad \text{e } C_3 = 1.25.$$

iii - Simetria Tetragonal:

$$C_{1111} = C_{2222} = C_1, \quad C_{1122} = C_2, \quad C_{1133} = C_{2233} = C_3, \quad C_{3333} = C_4,$$

$$C_{1212} = C_5, \quad C_{2323} = C_{3131} = C_6,$$

$$\text{onde } C_1 = 3.5, \quad C_2 = 1.5, \quad C_3 = 1.25, \quad C_4 = 2.5, \quad C_5 = 2.25 \quad \text{e } C_6 = 2.0.$$

Considere então a formulação variacional (5.13) 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, 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 (5.58) e (5.60)).

PS: Embora a solução exata para um corpo isotrópico, seja diferente a solução do correspondente problema para um corpo anisotrópico, mas com os valores dados para o tensor de elasticidade acima, é esperado que as soluções sejam próximas.

Métodos Numéricos e Algoritmos: Equação do Calor

Neste capítulo, vamos introduzir a equação de evolução do tipo parabólico, tomando como modelo a equação do calor. O sistema de equações diferenciais ordinárias, consequência da utilização do método de elementos finitos, será resolvido pelo método das diferenças finitas para os tempos discretos.

A parte de análise matemática, 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 Parabólica

Considere o problema (2.1) com a introdução da derivada temporal u_t , ou seja, o problema parabólico modelo que estudaremos é da forma:

$$\begin{cases} u_t(x, t) - \alpha u_{xx}(x, t) + \beta u(x, t) = f(x, t), & \forall (x, t) \in (0, 1) \times [0, T], \\ u(0, t) = u(1, t) = 0, & \forall t \in [0, T], \\ u(x, 0) = u_0(x), & \forall x \in (0, 1), \end{cases} \quad (6.1)$$

onde α e β são constantes reais positivas e $f = f(x, t)$ 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 $f = f(x, t)$ e $u_0(x)$ sejam funções suficientemente regulares.

A solução $u(x, t)$ representa a temperatura da barra em cada instante $t \in [0, T]$, que está presa nos extremos.

Neste capítulo, continuaremos a usar o sistema de notações adotada anteriormente, entretanto, agora as funções são também dependentes do tempo, e assim, por exemplo $w = w(., t) \in V$ significa que w é uma função dependente do tempo em Ω , ou seja

$$\begin{aligned} w : [0, T] &\longrightarrow V \\ t &\longmapsto w(t) \end{aligned}$$

Seja V um espaço normado, então pelo símbolo $L^p(0, T; V)$ denotamos o espaço dependente do tempo, com norma dada por:

$$\|w\|_{L^p(0, T; V)}^p = \int_0^T \|w(t)\|_V^p dt < \infty,$$

onde $\|w(t)\|_V$ denota a V -norma, que pertence ao espaço $L^p(0, T)$. Usualmente V é o espaço $H_0^1(\Omega)$ ou $L^2(\Omega)$ e $p = 2$ ou $p = \infty$. Para $p = \infty$, o espaço $L^\infty(0, T; V)$ denota as funções $\|w(t)\|_V$ essencialmente limitadas em $[0, T]$ e nesse caso a norma é definida por

$$\|w\|_{L^\infty(0, T; V)} = \sup_{t \in [0, T]} \|w(t)\|_V < \infty.$$

A solução do problema (6.1), para uma força $f \in L^2((0, T; L^2(0, 1)))$, por exemplo, pode ser obtida usando o método de separação de variáveis (método de Fourier), no qual a solução $u(x, t)$ é representada por uma série infinita. O problema numérico é a convergência lenta da série de Fourier, mesmo quando usada a denominada transformada rápida de Fourier. Essa é uma das razões para introdução do método de elementos finitos.

Definimos a forma bilinear

$$a(u, v) = \alpha \int_0^1 u_x v_x dx + \beta \int_0^1 uv dx, \quad (6.2)$$

associado a norma

$$\|u\|_a^2 = \alpha \|\nabla u\|_0^2 + \beta \|u\|_0^2.$$

Tomando $\nu_1 = \min\{\alpha, \beta\}$ e $\nu_2 = \max\{\alpha, \beta\}$ então

$$\nu_1 \|u\|_1^2 \leq \|u\|_a^2 \leq \nu_2 \|u\|_1^2, \quad (6.3)$$

ou seja, normas $\|\cdot\|_a$ e $\|\cdot\|_1$ são equivalentes.

Formulação Variacional para a Equação do Calor

Seja $\mathcal{D}(0, 1) = \{v \in C_0^\infty(0, 1); v(0) = v(1) = 0\}$ o espaço das funções testes com suporte compacto em $(0, 1)$. Note que a função $v = v(x)$ somente depende da variável x .

Multiplicando a equação (6.1) por $v \in \mathcal{D}(0, 1)$ e integrando em $(0, 1)$ obtém-se

$$\int_0^1 u_t(t)v \, dx - \alpha \int_0^1 u_{xx}(t)v \, dx + \beta \int_0^1 u(t)v \, dx = \int_0^1 f(t)v \, dx, \quad \forall v \in \mathcal{D}(0, 1). \quad (6.4)$$

Integrando por partes, usando (6.2) e denotando,

$$(f, v) = \int_0^1 f(t)v \, dx, \quad (6.5)$$

então (6.4) é equivalente a determinar uma função

$$\begin{aligned} u : [0, T] &\longrightarrow V \\ t &\longmapsto u(t) \end{aligned}$$

satisfazendo:

$$\begin{cases} (u_t, v) + a(u, v) = (f, v), \quad \forall v \in V \\ (u(0), v) = (u_0, v), \quad \forall v \in V, \end{cases} \quad (6.6)$$

onde $(6.6)_2$ representa a temperatura inicial de uma haste e estamos denotando $u(t)$ por u e $u(x, 0) = u(0)$.

Problema Aproximado

Seja $T > 0$ e $V_m = [w_1, w_2, \dots, w_m] \subseteq V$, um 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, representada por

$$u_m(t) = \sum_{i=1}^m d_i(t)w_i(x). \quad (6.7)$$

Considere o problema (6.6), redefinido no subespaço V_m . Então, no subespaço V_m queremos determinar uma função

$$\begin{aligned} u_m : [0, T] &\longrightarrow V_m \\ t &\longmapsto u_m(t) \end{aligned}$$

solução do seguinte sistema aproximado,

$$\begin{cases} (u'_m(t), v_m) + a(u_m(t), v_m) = (f(t), v_m), \quad \forall v \in V_m, \\ u_m(0) = u_{0m} \longrightarrow u_0 \quad \text{em } V. \end{cases} \quad (6.8)$$

A condição inicial significa que:

$$u_{0m} = \sum_{i=1}^m (u_0, w_i) w_i \longrightarrow u_0 = \sum_{i=1}^{\infty} (u_0, w_i) w_i, \quad \text{em } V, \quad \text{quando } m \rightarrow \infty,$$

onde $d_i(0) = (u_0, w_i)$. Assim é natural a escolha de u_{0m} como condição inicial do problema em V_m .

No capítulo de análise numérica (existência e unicidade de solução), provaremos que existe uma sucessão de soluções $u_m(x, t)$ do problema aproximado (6.8) pertencentes ao subespaço V_m que são limitadas, independente de m , possibilitando mostrar que essa sucessão converge para solução exata $u(x, t)$, quando $m \rightarrow \infty$, ou seja, quando aumentamos o número de vetores $\{w_i\}_{i=1}^m$ da base de V_m a solução do problema aproximado converge para a solução u do problema do problema (6.6).

Nesta seção estamos principalmente interessados em desenvolver métodos para obter as soluções numéricas $u_m(x, t)$ do problema aproximado (6.8) para cada instante de tempo t .

Como a solução procurada é da forma (6.7) então substituindo em (6.8) obtemos

$$\left(\sum_{i=1}^m d'_i(t) w_i(x), v_m \right) + a \left(\sum_{i=1}^m d_i(t) w_i(x), v_m \right) = (f, v_m), \quad \forall v_m \in V_m.$$

Como (\cdot, \cdot) são definidos em V_m , podemos escrever

$$\sum_{i=1}^m d'_i(t) \left(w_i(x), v_m \right) + \sum_{i=1}^m d_i(t) a \left(w_i(x), v_m \right) = (f, v_m), \quad \forall v_m \in V_m,$$

Como a igualdade é válida para todo $v_m \in V_m$, podemos tomar em particular $v_m = w_j$, resultando em

$$\sum_{i=1}^m d'_i(t) \left(w_i(x), w_j(x) \right) + \sum_{i=1}^m d_i(t) a \left(w_i(x), w_j(x) \right) = \left(f(t), w_j(x) \right).$$

Definindo as matrizes

$$\begin{aligned} A &= \left(w_i, w_j \right) = \int_0^1 w_i(x) w_j(x) dx, \\ B &= a \left(w_i, w_j \right) = \alpha \int_0^1 (w_i)_x(x) (w_j)_x(x) dx + \beta \int_0^1 (w_i)(x) (w_j)(x) dx, \\ F &= F_j = \left(f(t), w_j(x) \right) = \int_0^1 f(x, t) w_j(x) dx, \end{aligned} \quad (6.9)$$

obtemos o seguinte sistema de m -equações diferenciais ordinárias

$$\sum_{i=1}^m d'_i(t) a_{ij} + \sum_{i=1}^m d_i(t) b_{ij} = F_j, \quad \text{para } j = 1, \dots, m$$

que pode ser escrito na seguinte forma matricial:

$$\begin{cases} Ad'(t) + Bd(t) = F(t), & \forall t \in [0, T] \\ d(0) = \left((u_0, w_1), (u_0, w_2), \dots, (u_0, w_m) \right) = d_0, \end{cases} \quad (6.10)$$

onde $d(0)$ é a condição inicial e $d(t) = (d_1(t), d_2(t), \dots, d_m(t))^T$ é o vetor incógnita. A matriz A é chamada de matriz capacitância enquanto que a matriz B é chamada de matriz de condutividade e F 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), a matriz A é simétrica e definida positiva. Em particular a matriz A é invertível e assim podemos escrever o sistema de equações diferenciais ordinárias na forma

$$d'(t) + A^{-1}Bd(t) = A^{-1}F(t), \quad \forall t > 0, \quad \text{com } d(0) = d_0,$$

e portanto tem solução única $d(t)$ para $t \in [0, T]$. Assim, por (6.7), a solução aproximada $u_m(x, t)$ do problema (6.8) pode ser calculada.

Em geral, a solução $d(t)$ não é conhecida explicitamente para todo $t \geq 0$, e dessa forma é necessário utilizar um método numérico para obter uma solução aproximada. O método numérico para resolver numericamente o sistema de equações diferenciais ordinárias (6.10) a ser empregado é o método das diferenças finitas.

Vamos então retornar ao sistema de equações diferenciais ordinárias (6.10). Note que ao aplicar o método de elementos finitos no espaço obtemos um sistema de equações diferenciais ordinárias cuja variável é o tempo t . Esse sistema será então resolvido para tempos discretos t_n , utilizando métodos numéricos baseado no método das diferenças finitas.

Considere então o sistema de m equações diferenciais ordinárias nos tempos discretos t_n , onde $t_n = n\Delta t$ para $n = 1, 2, \dots, N$. Então

$$\begin{cases} Ad'(t_n) + Bd(t_n) = F(t_n), \\ d(0) = d_0 \end{cases} \quad (6.11)$$

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 (6.11), ou seja, para obtenção

do vetor $d^n = (d_1^n, d_2^n, \dots, d_m^n)$, para $n = 1, 2, \dots, N$. Conhecida a função base $\{w_i\}_{i=1}^m$ e determinando o vetor d^n para todo n , então podemos calcular a solução aproximada $u_m(x, t_n)$, para $n = 1, 2, \dots, N$, por

$$u_m^n(x) = u_m(x, t_n) = \sum_{i=1}^m d_i(t_n) w_i(x) = \sum_{i=1}^m d_i^n w_i(x). \quad (6.12)$$

6.2.1 Método de Euler Regressivo

Da diferença regressiva no tempo (1.62) temos que,

$$(d'(t))_{t=t_n} = \left(\frac{\partial d(t)}{\partial t} \right)_n = \frac{1}{\Delta t} (d^n - d^{n-1}),$$

onde a função d , como sabemos, depende somente da variável tempo $t = t_n$. Substituindo a aproximação no sistema (6.11) e usando a notação dada, obtemos;

$$\frac{1}{\Delta t} A (d^n - d^{n-1}) + B d^n = F^n, \quad n = 1, 2, \dots, N \quad (6.13)$$

ou equivalentemente

$$(A + \Delta t B) d^n = \Delta t F^n + A d^{n-1} = b^n, \quad n = 1, 2, \dots, N. \quad (6.14)$$

Note que o método de Euler é também um método implícito, exceto se as matrizes A e B são matrizes diagonais, isto é, os vetores bases w_i formam uma base ortogonal.

O vetor independente $b^n = (b_1^n, b_2^n, \dots, b_m^n)^t$ é conhecido, pois A , B , F e o incremento Δt são conhecidos e além disso as matrizes A e B são independentes de t .

O vetor incógnita $d^n = (d_1^n, d_2^n, \dots, d_m^n)^t$ pode ser determinado de forma única, pois a matriz dos coeficientes $(A + \Delta t B)$, onde A e B definidas em (6.9), são simétricas e definida positiva. Logo $(A + \Delta t B)$ é não singular e o sistema para cada n fixo pode ser resolvido, usando por exemplo, os métodos diretos de **Cholesky** (somente quando a matriz é simétrica e definida positiva), Crout (LDL^t) (para matriz simétrica) e o método de eliminação de Gauss.

Observe que a esparsidade da matriz depende diretamente da escolha da base $w_i(x)$, isto é, se $w_i(x)$ for a base linear ou a base cúbica, por exemplo, a matriz $(A + \Delta t B)$ será tridiagonal ou pentagonal, respectivamente.

Para cada $n = 1, 2, \dots, N$, temos um sistema linear e assim para determinar a solução $u(x, t)$ nos tempos discretos t_n , são necessários resolver N sistemas lineares, onde $t_1 = 1$ e $t_N = T$.

Algoritmo

Para inicialização do método iterativo, faz-se $n = 1$ em (6.14), obtendo-se

$$(A + \Delta t B)d^1 = \Delta t F^1 + Ad^0 = b^1,$$

Mas $F^1 = \left(f(x, t_1), w_j(x)\right)$ é conhecido para todo x e $d^0 = d(0)$ é a condição inicial dada. Assim resolvendo o sistema determina-se d^1 . Para obter $\{d^2, d^3, \dots, d^N\}$ basta resolver o sistema linear:

$$(A + \Delta t B)d^n = \Delta t F^n + Ad^{n-1} = b^n, \quad n = 2, 3, \dots, N. \quad (6.15)$$

Para o método de Euler regressivo, a aproximação no tempo por diferenças finitas, tem ordem de convergência proporcional a Δt , ou seja, diz-se nesse caso que o erro tem ordem de convergência Δt representado por $\mathcal{O}(\Delta t)$.

6.2.2 Método de Euler Progressivo

Usando a aproximação da derivada pela diferença progressiva no tempo (1.62), em lugar da diferença regressiva, obtemos que

$$(d'(t))_{t=t_n} = \left(\frac{\partial d(t)}{\partial t}\right)_n = \frac{1}{\Delta t}(d^{n+1} - d^n).$$

De forma análoga ao método anterior, substituindo a aproximação no sistema (6.11) obtemos o seguinte sistema linear,

$$Ad^{n+1} = (A - \Delta t B)d^n + \Delta t F^n, \quad n = 0, 1, \dots, N. \quad (6.16)$$

Assim para cada $n = 0, 1, \dots, N$, resolvendo o sistema linear obtém-se os valores de d^n e portanto a solução aproximada. Note que a matriz A não singular e portanto o sistema tem solução única.

Mostraremos adiante, que o método iterativo (6.14) (Euler regressivo) é *incondicionalmente* convergente. Por outro lado, o método iterativo (6.16) (Euler progressivo) é *condicionalmente* convergente (ver [23]). Além disso, uma outra questão importante, é porque não usar a diferença central (1.61) que tem uma precisão de $\mathcal{O}(\Delta t)^2$ em vez de usar a diferença regressiva (1.6)? Prova-se também nesse caso, que o método não é convergente, (ver [19]).

O método que descreveremos a seguir tem um precisão melhor, a ordem de convergência é $\mathcal{O}(\Delta t)^2$

6.2.3 Método de Crank-Nicolson

O método considera a diferença progressiva no tempo (1.62) para o termo $d'(t)$, mas os restantes dos termos que dependem de t e não envolvem derivas no tempo é feita a seguinte aproximação pela média aritmética,

$$d^{n+\frac{1}{2}} = \frac{1}{2}(d^{n+1} + d^n) \quad \text{e} \quad F^{n+\frac{1}{2}} = \frac{1}{2}(F^{n+1} + F^n)$$

Fazendo as substituições no sistema (6.11) obtemos;

$$\frac{A}{\Delta t}(d^{n+1} - d^n) + \frac{B}{2}(d^{n+1} + d^n) = \frac{1}{2}(F^{n+1} + F^n),$$

que é equivalente a

$$(2A + \Delta t B)d^{n+1} = (2A - \Delta t B)d^n + \Delta t(F^{n+1} + F^n), \quad n = 0, 1, \dots, (N-1) \quad (6.17)$$

Como a função $f(x, t)$ para todo x e t é conhecida então $F^n = (f(x, t_n), w_j(x))$ é conhecida para todo n . Para cada $n = 0, 1, \dots, (N-1)$, temos que resolver o sistema linear (6.17) e assim para determinar a solução $u(x, t)$ nos tempos discretos $t_n = n\Delta t$.

Algoritmo

Para inicialização do método iterativo, faz-se $n = 0$ em (6.17), obtendo-se

$$(2A + \Delta t B)d^1 = (2A - \Delta t B)d^0 + \Delta t(F^1 + F^0) = b^0.$$

Mas F^0, F^1 são conhecidos para todo x e $d^0 = d(0)$ é a condição inicial dada. Como $(A + \Delta t B)$ é não singular, então resolvendo o sistema determina-se d^1 e sucessivamente obtemos de forma única d^2, d^3, \dots, d^N , através do sistema (6.17). 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 os pesos θ :

$$d^{n+\theta} = \theta d^{n+1} + (1 - \theta)d^n \quad \text{e} \quad F^{n+\theta} = \theta F^{n+1} + (1 - \theta)F^n, \quad \theta \in [0, 1].$$

Usando a derivada progressiva para o termo da primeira derivada e usando a média ponderada para os outros termos, com os pesos respectivos, obtemos a seguinte família de métodos:

$$(A + \theta \Delta t B)d^{n+1} = (A - (1 - \theta)\Delta t B)d^n + \Delta t(F^{n+\theta}), \quad n = 0, 1, \dots, N-1 \quad (6.18)$$

Podemos notar que;

- (i) Se $\theta = 0$, obtemos o método de Euler progressivo (6.16).
- (ii) Se $\theta = 1/2$, obtemos o método de Crank-Nicolson (6.17).
- (iii) Se $\theta = 1$, obtemos o método de Euler regressivo (6.14).

Algoritmo 1

A inicialização do método é semelhante aos métodos anteriores, ou seja, faz-se $n = 0$ em (6.18) e então resolve-se o sistema linear obtendo-se o vetor solução d^1 . A partir desse vetor solução determina-se d^2, d^3, \dots, d^N .

Note que no algoritmo anterior para o (6.18), não há nenhuma preocupação em calcular o valor da derivada de d^n , ou seja, $d_t^n = d'(t_n)$.

A seguir, apresentamos um algoritmo, tipo preditor-corretor, que determina a solução de d^n e sua derivada $v^n = d'(t_n)$.

Algoritmo 2: Método Generalizado Trapezoidal

Considere o seguinte algoritmo;

$$\begin{cases} Av^{n+1} + Bd^{n+1} = F^{n+1} \\ d^{n+1} = d^n + \Delta t v^{n+\theta} \\ v^{n+\theta} = \theta v^{n+1} + (1 - \theta)v^n, \end{cases} \quad (6.19)$$

onde v^n e d^n são aproximações de $d'(t_n)$ e $d(t_n)$ respectivamente e $F^{n+1} = F(t_{n+1})$, (Δt) é o incremento do tempo e $\theta \in [0, 1]$ é um parâmetro.

O problema computacional é para determinar d^{n+1} e v^{n+1} a partir dos valores conhecidos do tempo anterior d^n e v^n .

Inicialização do Algoritmo

No tempo $t = 0$, a temperatura inicial d^0 é conhecida então $v^0 = d'(0)$ pode ser determinado fazendo $t = t_0 = 0$ na equação discreta (6.11), ou seja

$$Av^0 = F^0 - Bd^0$$

Resolvendo o sistema linear, determina-se $v^0 = d'(0)$.

Para $n = 0, 1, \dots, (N - 1)$ o procedimento para determinar a solução aproximada é dividido nas seguintes etapas:

(i) Definimos um preditor para d^{n+1} na forma:

$$\tilde{d}^{n+1} = d^n + (1 - \theta)\Delta t v^n$$

(ii) De (6.19)₂ e (6.19)₃ tem-se que

$$d^{n+1} = d^n + \Delta t v^{n+\theta} = d^n + \Delta t \left(\theta v^{n+1} + (1 - \theta)v^n \right) = \tilde{d}^{n+1} + \theta \Delta t v^{n+1}$$

(iii) Substituindo (ii) em (6.19)₁, temos

$$(A + \theta \Delta t B) v^{n+1} = F^{n+1} - B \tilde{d}^{n+1}$$

Resolvendo o sistema linear obtém-se v^{n+1} . Com v^{n+1} calculado, então os valores de d^{n+1} são determinados pela relação (ii), ou seja,

$$d^{n+1} = \tilde{d}^{n+1} + \theta \Delta t v^{n+1},$$

e assim sucessivamente para $n = 0, 1, \dots, (N - 1)$.

6.3 Simulação Numérica: Equação do Calor

Nessa seção faremos algumas simulações numéricas do problema parabólico, utilizando os diferentes métodos apresentados. Para verificar a eficiência dos métodos, serão construídos problemas onde a solução exata do problema (6.1) é conhecida, possibilitando então comparar com as soluções numéricas em diferentes normas.

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

$$E(x_i, t_n) = u(x_i, t_n) - u_h(x_i, t_n). \quad (6.20)$$

As normas $L^\infty(0, T; L^2(0, 1))$ e $L^\infty(0, T; H^1(0, 1))$ do erro são definidas por

$$\begin{aligned} \|E\|_{L^\infty(0, T; L^2(0, 1))} &= \max_{t_n \in [0, T]} \left(\int_0^1 |E(x)|^2 dx \right)^{1/2} \\ \|E\|_{L^\infty(0, T; H^1(0, 1))} &= \max_{t_n \in [0, T]} \left(\int_0^1 (|E(x)|^2 + \left| \frac{dE(x)}{dx} \right|^2) dx \right)^{1/2} \end{aligned} \quad (6.21)$$

para $n = 1, \dots, N$.

Para o caso discreto com a malha uniforme $h = h_i = x_{i+1} - x_i$, usando a definição dada em (2.98), e denotando $E_i^n = E(x_i, t_n)$ temos para cada $n = 1, \dots, N$, as normas

$$\begin{aligned} \|E\|_{L^\infty(0, T; L^2(0, 1))} &= \max_{t_n \in [0, T]} \left(h \sum_{i=1}^m |E_i^n|^2 \right)^{1/2}, \\ \|E\|_{L^\infty(0, T; H^1(0, 1))} &= \max_{t_n \in [0, T]} \left(h \sum_{i=1}^m |E_i^n|^2 + h \sum_{i=1}^m \left| \frac{\partial E_i^n}{\partial x} \right|^2 \right)^{1/2} \\ &=^* \max_{t_n \in [0, T]} \left(h \sum_{i=1}^m |E_i^n|^2 \right)^{1/2} + \left(\frac{1}{h} \sum_{i=1}^m |E_{i+1}^n - E_i^n|^2 \right)^{1/2}. \end{aligned} \quad (6.22)$$

(* Normas equivalentes em espaço de dimensão finita). Note que também poderíamos ter usado a norma discreta dos métodos dos trapézios $\|\cdot\|_{L^2(0,1)}$ e $\|\cdot\|_{H_0^1(0,1)}$ definidos em (2.99).

Como veremos no teorema (8.6), as estimativas de erro para o método generalizado trapezoidal, sobre condições suficientemente regulares são dadas por:

Para $\theta \in [1/2, 1]$

$$\begin{cases} \|E\|_{L^\infty(0, T; L^2(0,1))} = \|u - u_h\|_{L^\infty(0, T; L^2(0,1))} \leq c_1 (h^{k+1} + (\Delta t)^m) \\ \|E\|_{L^\infty(0, T; H_0^1(0,1))} = \|u - u_h\|_{L^\infty(0, T; H_0^1(0,1))} \leq c_2 (h^k + (\Delta t)^m), \end{cases} \quad (6.23)$$

Para $\theta \in [0, 1/2) \Rightarrow \Delta t/h^2 \leq f(\theta)$, ou seja, o método é condicionalmente convergente.

De outra forma dizemos que o erro $\|E\|_{L^\infty(0, T; L^2(0,1))}$ é de ordem $\mathcal{O}(h^{k+1} + \Delta t^m)$ e $\|E\|_{L^\infty(0, T; H_0^1(0,1))}$ tem ordem $\mathcal{O}(h^k + \Delta t^m)$.

Note que, em particular para $k = 1$, ou seja, quando as funções bases são polinômios lineares por partes então a ordem no espaço é de $\mathcal{O}(h)$ em $H_0^1(0, 1)$ e ordem $\mathcal{O}(h^2)$ em $L^2(0, 1)$.

Além disso, se $\theta = 1/2 \Rightarrow m = 2$ e se $\theta \neq 1/2 \Rightarrow m = 1$. As constantes c_1 e c_2 são positivas independentes de h e Δt e 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 $\mathcal{O}(\Delta t)^2$.
- ii O método de Euler progressivo, $\theta = 0$, é condicionalmente convergente.

Para os exemplos numéricos a seguir, considere os intervalos discretos $t_n \in [0, 1]$, e as funções bases $\varphi \in V_m^k$ como as funções lineares por partes, ou seja, $k = 1$ em (6.23). Serão comparados os 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, $\theta = 0$.

Considere o problema,

$$\begin{cases} u_t(x, t) - \alpha u_{xx}(x, t) + \beta u(x, t) = f(x, t), & \forall (x, t) \in (0, 1) \times [0, 1] \\ u(0, t) = u(1, t) = 0, & \forall t \in [0, 1] \\ u(x, 0) = (1/\pi^2) \sin(\pi x) & \forall x \in (0, 1), \end{cases} \quad (6.24)$$

Usando o método de separação de variáveis (método de Fourier), a solução do problema é dada por $u(x, t) = (1/\pi^2) \sin(\pi x) \exp(-\lambda t)$, 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, considere os dois casos separados, que serão chamados de exemplo 1 e exemplo 2.

Programa Computacional O resultados numéricos destes dois exemplos podem ser obtidos usando o Programa Computacional, denominado *Calor.cpp*, que está anexo no final do livro.

Exemplo 1: Nesse exemplo, considere $f \equiv 0$, e $\alpha = \beta = 1$ e vamos considerar $\lambda = \pi^2 + 1$.

A tabela 6.1 e a figura 6.1, mostram o erro na norma $L^2(0, 1)$ para cada tempo discreto $t_n = n \times \Delta t$, 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

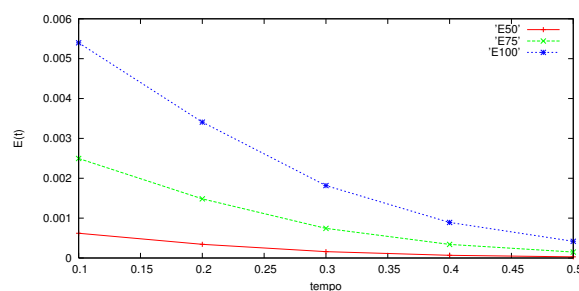


Figura 6.1: Decaimento do erro $\|E(t)\|_{L^2(0,1)}$, $t \in [0.0, 0.5]$

Ordem de Convergência

Vamos analisar numericamente a ordem de convergência do exemplo 1, onde a solução é conhecida e dada por: $u(x, t) = (1/\pi^2) \sin(\pi x) \exp(-(\pi^2 + 1)t)$. Considere $E_i = \max_{t \in [0,1]} \|E_i(t)\|_{L^2(0,1)}$, o erro associado a malha $h = (5 \times 2^{i+1})^{-1}$, para $i = 0, 1, \dots, N$ e considere em particular $\Delta t = h$. Então, como vimos em (3.58), a ordem de convergência $p = \ln(E_i/E_{i+1})/\ln(2)$. Vamos agora calcular, a ordem de convergência p para o método de Crank-Nicolson e o método de 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
$\theta = 1.0$ (Euler Regressivo)	0.1	0.009911	—
	0.05	0.005854	0.760
	0.025	0.003215	0.865
	0.0125	0.001691	0.927
	0.00625	0.000868	0.962
$\theta = 0.5$ (Crank Nicolson)	0.1	0.003176	—
	0.05	0.000717	2.147
	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:

$$\begin{cases} u_t(x, t) - \alpha u_{xx}(x, t) + \beta u(x, t) = f(x, t), & \forall (x, t) \in (0, 1) \times [0, 1] \\ u(0, t) = u(1, t) = 0, & \forall t \in [0, 1] \\ u(x, 0) = (1/\pi^2) \sin(\pi x) & \forall x \in (0, 1), \end{cases} \quad (6.25)$$

Seja $\lambda = \alpha = \beta = 1$. Então tomando $f(x, t) = \sin(\pi x) \exp(-t)$, vimos que a solução do problema é dada por $u(x, t) = (1/\pi^2) \sin(\pi x) \exp(-t)$. 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
$\theta = 1.0$ (Euler Regressivo)	0.1	0.000234	—
	0.05	0.000123	0.928
	0.025	0.000063	0.965
	0.0125	0.000032	0.977
	0.00625	0.000015	1.093
$\theta = 0.5$ (Crank Nicolson)	0.1	0.00004	—
	0.05	0.00001	2.000
	0.025	0.00000	—

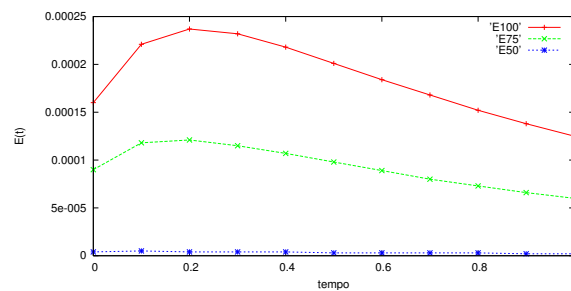
Tabela 6.4: Ordem de convergência

a desigualdade, $0.5 \leq \theta \leq 1.0$, o método é convergente. A figura 6.2 mostra o erro

	Δt	h	$E_{L^\infty(0,T;L^2(0,1))}$	$E_{L^\infty(0,T;H^1(0,1))}$
$\theta = 1.0$ (Euler Regressivo)	0.001	0.1	0.000003	0.000004
	0.001	0.05	0.000002	0.000003
$\theta = 0$ (Euler Progressivo)	0.001	0.1	0.000002	0.000003
	0.001	0.05	diverge	diverge

Tabela 6.5: Tabela de erros

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 (6.25) do Exemplo 2.
 - (a) Determine a 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 (3.9). 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 (6.22).
 - (b) Nas mesmas condições do item anterior, refaça o problema usando como função base os polinômios de Hermite, definidos em (3.21).
 - (c) Suponha que a solução exata do problema (6.25) seja desconhecida. Verifique a ordem de convergência numérica, definida em (3.58), para o método de Crank-Nicolson e Euler regressivo, usando como função base as funções dos itens anteriores e compare com a estimativa de erro esperado em (8.85).
2. **Fronteira mista:** Considere o problema (6.25), com $\alpha = \beta = 1$. Sabendo que a solução exata é $u(x, t) = (1/\pi^2) \sin(\pi x) \exp(-(\pi^2 + 1)t)$, então o problema com as condições de fronteira de Neumann ($x = 0$) e Dirichlet ($x = 1$) é dado por

$$\begin{cases} u_t(x, t) - \alpha u_{xx}(x, t) + \beta u(x, t) = 0, & \forall (x, t) \in (0, 1) \times [0, 1] \\ u_x(0, t) = (1/\pi) \exp(-(\pi^2 + 1)t), & u(1, t) = 0, \quad \forall t \in [0, 1] \\ u(x, 0) = (1/\pi^2) \sin(\pi x), & \forall x \in (0, 1). \end{cases} \quad (6.26)$$

Determine as soluções numéricas aproximadas, usando os $(\theta$ -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:** Definimos o problema homogêneo

$$\begin{cases} u_t(\mathbf{x}, t) - \Delta u(\mathbf{x}, t) + u(\mathbf{x}, t) = f(\mathbf{x}, t), & \forall (\mathbf{x}, t) \in Q \times [0, 1] \\ u(\mathbf{x}, t) = 0, & \forall \mathbf{x} \in \partial\Omega \quad \text{e} \quad \forall t \in [0, 1] \\ u(\mathbf{x}, 0) = (1/\pi^2) \sin(\pi x) \sin(\pi y), & \forall \mathbf{x} \in Q \end{cases} \quad (6.27)$$

onde $\mathbf{x} = (x, y) \in Q = [0, 1] \times [0, 1]$. Por $\partial\Omega = \bigcup_{i=1}^4 \Gamma_i$, onde

$$\begin{aligned} \Gamma_1 &= \{(x, 0) \in \partial\Omega; 0 \leq x \leq 1\} \\ \Gamma_2 &= \{(1, y) \in \partial\Omega; 0 \leq y \leq 1\} \\ \Gamma_3 &= \{(x, 1) \in \partial\Omega; 0 \leq x \leq 1\} \\ \Gamma_4 &= \{(0, y) \in \partial\Omega; 0 \leq y \leq 1\} \end{aligned}$$

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 $f(\mathbf{x}, t) = 0$.

Nessas condições, a solução exata do problema (6.27) é dada por $u(\mathbf{x}, t) = (1/\pi^2) \sin(\pi x) \sin(\pi y) \exp(-(2\pi^2 + 1)t)$. Definindo a malha uniforme $h_1 = 0.05$ no eixo-x e $h_2 = 0.05$ 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.

Métodos Numéricos e Algoritmos: Equação da Onda

Neste capítulo, vamos introduzir a equação de evolução do tipo hiperbólico, tendo como modelo a equação da onda. Por razões numéricas, vamos usar, como anteriormente, a aproximação por diferenças finitas para resolução de um sistema de equações diferenciais ordinárias.

A parte de análise matemática, 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.

Existe um grande número de métodos numéricos para a resolução da equação diferencial, que podem ser encontrados, por exemplo, em [1, 10].

Considere o problema (2.1) com a introdução da derivada temporal u_{tt} , ou seja, o problema hiperbólico modelo que estudaremos é da forma:

$$\left\{ \begin{array}{ll} u_{tt}(x, t) - \alpha u_{xx}(x, t) + \beta u(x, t) = f(x, t), & \forall (x, t) \in (0, 1) \times [0, T] \\ u(0, t) = u(1, t) = 0, & \forall t \in [0, T] \\ u(x, 0) = u_0(x), & \forall x \in (0, 1), \\ u_t(x, 0) = u_1(x), & \forall x \in (0, 1), \end{array} \right. \quad (7.1)$$

onde α e β são constantes reais positivas e $f = f(x, t)$ é a força atuando sobre a barra em cada instante $t \in [0, T]$, $u(0, t)$ e $u(1, t)$ representam as condições de fronteira em cada instante, $u(x, 0) = u_0(x)$ é a posição inicial da onda e $u_t(x, 0) = u_1(x)$ é a velocidade inicial da onda.

A solução $u = u(x, t)$ representa a posição da onda em cada instante $t \in [0, T]$, que está presa nos extremos.

Formulação Variacional da Equação da Onda

Seja $\mathcal{D}(0, 1) = \{v \in C_0^\infty(0, 1); v(0) = v(1) = 0\}$ o espaço das funções testes com suporte compacto em $(0, 1)$. Note que a função $v = v(x)$ somente depende da variável x . Multiplicando a equação (7.1) por $v \in \mathcal{D}(0, 1)$ e integrando em $(0, 1)$ obtém-se

$$\int_0^1 u''(t)v \, dx - \alpha \int_0^1 u_{xx}(t)v \, dx + \beta \int_0^1 u(t)v \, dx = \int_0^1 f(t)v \, dx, \quad v \in \mathcal{D}(0, 1). \quad (7.2)$$

Usando a forma bilinear (6.2), podemos escrever a seguinte forma variacional

$$\begin{cases} (u'', v) + a(u, v) = (f, v), \quad \forall v \in V \\ (u(0), v) = (u_0, v), \quad \forall v \in V \\ (u'(0), v) = (u_1, v), \quad \forall v \in V. \end{cases} \quad (7.3)$$

7.1 Problema Aproximado

Seja $T > 0$ e $V_m = [w_1, w_2, \dots, w_m] \subseteq V$, um subespaço vetorial gerado pelos m -primeiros elementos da base do espaço de Hilbert V . Então toda função $u_m(t) \in V_m$ pode ser escrita como combinação linear dos elementos da base, representada por

$$u_m(t) = \sum_{i=1}^m d_i(t)w_i(x). \quad (7.4)$$

Considere o problema (7.3), redefinido no subespaço V_m . Então, no subespaço V_m queremos determinar uma função

$$\begin{aligned} u_m : [0, T] &\longrightarrow V_m \\ t &\longmapsto u_m(t) \end{aligned}$$

solução do seguinte sistema aproximado,

$$\begin{cases} (u_m''(t), v) + a(u_m(t), v) = (f(t), v), \quad \forall v \in V_m, \\ u_m(0) = u_{0m} \rightarrow u_0 \text{ forte em } V \\ u_m'(0) = u_{1m} \rightarrow u_1 \text{ forte em } H \end{cases} \quad (7.5)$$

Provaremos no teorema (9.5) sobre existência e unicidade de solução, que existe uma sucessão de soluções $u_m(x, t)$ do problema aproximado (7.5) pertencentes ao subespaço

V_m que são limitadas, independente de m , possibilitando mostrar que essa sucessão convirja para solução exata $u(x, t)$, quando $m \rightarrow \infty$, ou seja, quando aumentamos o número de vetores $\{w_i\}_{i=1}^m$ da base de V_m a solução do problema aproximado converge para a solução u do problema do problema (7.3).

Substituindo (7.4) em (7.5), obtemos

$$\left(\sum_{i=1}^m d_i''(t) w_i(x), v_m \right) + a \left(\sum_{i=1}^m d_i(t) w_i(x), v_m \right) = (f, v_m), \quad \forall v_m \in V_m$$

Como (\cdot, \cdot) são definidos em V_m , podemos escrever

$$\sum_{i=1}^m d_i''(t) \left(w_i(x), v_m \right) + \sum_{i=1}^m d_i(t) a \left(w_i(x), v_m \right) = (f, v_m), \quad \forall v_m \in V_m,$$

Como a igualdade é válida para todo $v_m \in V_m$, podemos tomar em particular $v_m = w_j$, resultando em

$$\sum_{i=1}^m d_i''(t) \left(w_i(x), w_j(x) \right) + \sum_{i=1}^m d_i(t) a \left(w_i(x), w_j(x) \right) = \left(f(t), w_j(x) \right).$$

Definindo as matrizes

$$A = \left(w_i(x), w_j(x) \right), \quad B = a \left(w_i(x), w_j(x) \right) \quad \text{e} \quad F = F_j = \left(f(t), w_j(x) \right), \quad (7.6)$$

obtemos o seguinte sistema de m-equações diferenciais ordinárias

$$\sum_{i=1}^m d_i''(t) a_{ij} + \sum_{i=1}^m d_i(t) b_{ij} = F_j, \quad \text{para } j = 1, \dots, m$$

que pode ser escrito na seguinte forma matricial:

$$\begin{cases} Ad''(t) + Bd(t) = F(t), & \forall t \in [0, T] \\ d(0) = \left((u_0, w_1), (u_0, w_2), \dots, (u_0, w_m) \right) = d_0 \\ d'(0) = \left((u'_0, w_1), (u'_0, w_2), \dots, (u'_0, w_m) \right) = d_1 \end{cases} \quad (7.7)$$

onde $d(t) = (d_1(t), d_2(t), \dots, d_m(t))^T$ é o vetor incógnita, $d(0)$ é a posição inicial e $d'(0)$ é a velocidade inicial da onda. 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 de equações diferenciais ordinárias na forma

$$d''(t) + A^{-1}Bd(t) = A^{-1}F(t), \quad \forall t > 0, \quad \text{com} \quad d(0) = d_0, \quad d'(0) = d_1$$

e portanto tem solução única $d(t)$ 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 $d(t)$ não é conhecida explicitamente para todo t positivo, e dessa forma é necessário desenvolver métodos numéricos para obter uma solução aproximada para tempos discretos t_n . O método numérico para resolver o sistema numericamente o sistema de equações diferenciais ordinárias (7.7) para cada instante é o método das Diferenças Finitas.

7.2 Algoritmos para a Equação da Onda

Considere os tempos discretos t_n , onde $t_n = n\Delta t$, $n = 0, 1, \dots, N$, temos

$$\begin{cases} Ad''(t_n) + Bd(t_n) = F(t_n), & \forall t \in [0, T] \\ d(0) = d_0 & d'(0) = d_1 \end{cases} \quad (7.8)$$

7.2.1 Método da Diferença Central

Usando a diferença central (1.61) no sistema de equações para aproximar a segunda derivada obtemos que,

$$A \frac{(d^{n+1} - 2d^n + d^{n-1}))}{(\Delta t)^2} + Bd^n = F^n, \quad n = 0, 1, \dots, (N-1) \quad (7.9)$$

que é equivalente a

$$Ad^{n+1} = (2A - (\Delta t)^2 B)d^n + (\Delta t)^2 F^n - Ad^{n-1}, \quad n = 0, 1, \dots, (N-1) \quad (7.10)$$

Como a matriz A é não singular e o lado direito do sistema é conhecido então resolvendo o sistema linear para cada $t_n = n\Delta t$, obtém-se uma única solução do sistema e portanto o vetor d^n é determinado e consequentemente a solução aproximada $u_m(x, t_n)$ por (7.4).

Algoritmo

Para inicialização do método iterativo, faz-se $n = 0$ em (7.10), obtendo-se

$$Ad^1 = (2A - (\Delta t)^2 B)d^0 + (\Delta t)^2 F^0 - Ad^{-1} \quad (7.11)$$

As matrizes A, B do lado direito são conhecidas e independentes de t , o passo Δt é dado, F^0 é a força, conhecida para todo tempo t e d^0 é dado pela posição inicial da onda.

Cálculo do termo d^{-1}

O termo d^{-1} é obtido através da velocidade inicial u_1 . De fato considere a diferença central (1.57), ou seja,

$$(d'(t))_n = \frac{d^{n+1} - d^{n-1}}{2\Delta t}$$

Fazendo $n = 0$, temos

$$g'(0) = \frac{d^1 - d^{-1}}{2\Delta t} = v_1$$

onde g_1 é a velocidade inicial dada. Assim podemos obter o termo d^{-1} da relação

$$d^{-1} = d^1 - 2\Delta t v_1 \quad (7.12)$$

Substituindo em (7.11), obtemos para primeira iteração o sistema

$$Ad^1 = (A - \frac{(\Delta t)^2}{2}B)d^0 + \frac{(\Delta t)^2}{2}F^0 + \Delta t Av_1 \quad (7.13)$$

Como a matriz é não singular então o sistema linear de m equações tem uma única solução $d^1 = (d_1^1, d_2^1, \dots, d_m^1)$. Para calcular as soluções nos tempos $n = 1, 2, \dots, N$, basta resolver o sistema (7.10) para cada n , ou seja,

$$Ad^{n+1} = (2A - (\Delta t)^2 B)d^n + (\Delta t)^2 F^n - Ad^{n-1} \quad n = 1, 2, \dots, (N-1)$$

As aproximações para a primeira e segunda derivada 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 capítulo. A família de métodos a seguir são incondicionalmente convergentes, para o parâmetro $\theta \geq 1/4$.

7.2.2 Métodos Implícitos - Método de Newmark

Considere o sistema (7.8) e a seguinte aproximação para (7.8)₁

$$A \frac{(d^{n+1} - 2d^n + d^{n-1}))}{(\Delta t)^2} + Bd^{*n} = F^{*n} \quad (7.14)$$

onde estamos definindo a média geométrica $d^{*n} = \theta d^{n+1} + (1 - 2\theta)d^n + \theta d^{n-1}$. Fazendo as substituições, obtemos o seguinte sistema algébrico de equações lineares

$$\begin{aligned} \left(A + \theta(\Delta t)^2 B\right) d^{n+1} &= \left(2A - (1 - 2\theta)(\Delta t)^2 B\right) d^n - \left(A + \theta(\Delta t)^2 B\right) d^{n-1} \\ &+ (\Delta t)^2 \left(\theta(F^{n+1} + F^{n-1}) + (1 - 2\theta)F^n\right) \end{aligned} \quad (7.15)$$

Definindo as matrizes $M = \left(A + \theta(\Delta t)^2 B\right)$ e $L = \left(2A - (1 - 2\theta)(\Delta t)^2 B\right)$, então podemos escrever o sistema linear (7.15) por

$$Md^{n+1} = Ld^n - Md^{n-1} + (\Delta t)^2 \left(\theta(F^{n+1} + F^{n-1}) + (1 - 2\theta)F^n\right) \quad (7.16)$$

O método de aproximação acima é conhecido como método de Newmark. Para que o esquema numérico (7.16) seja incondicionalmente estável no tempo é necessário que $\theta \geq 1/4$, como veremos na análise de convergência do método. Note que quando $\theta = 0$ obtemos o método da diferença central (7.10).

Algoritmo

Para inicialização do processo tomemos $n = 0$ em (7.16), obtendo-se

$$Md^1 = Ld^0 - Md^{-1} + (\Delta t)^2 \left(\theta(F^1 + F^{-1}) + (1 - 2\theta)F^0\right) \quad (7.17)$$

O valor de d^{-1} pode ser obtido pela aproximação (7.12). Para a força, no processo de inicialização ($n = 0$) podemos tomar a aproximação

$$F^{*0} = \left(\theta(F^1 + F^{-1}) + (1 - 2\theta)F^0\right),$$

desde que o valor de F^{-1} seja conhecido. Caso contrário podemos tomar a aproximação inicial $F^0 = F^{*0}$. Assim o algoritmo pode ser inicializado por

$$Md^1 = \frac{1}{2}Ld^0 + \Delta t M v_1 + \frac{(\Delta t)^2}{2} F^{*0} \quad (7.18)$$

Resolvendo-se o sistema linear obtemos a solução d^1 . Sucessivamente para cada $n = 1, 2, \dots$ obtemos d^2, d^3, \dots, d^N resolvendo para cada n o sistema linear (7.16),

$$Md^{n+1} = Ld^n - Md^{n-1} + (\Delta t)^2 \left(\theta(F^{n+1} + F^{n-1}) + (1 - 2\theta)F^n\right).$$

7.2.3 Método Numérico: θ -método: Equação da Onda

O método- θ ou também conhecido como θ -método foi aplicado anteriormente para a equação do calor, que é um problema de primeira ordem no tempo. Vamos aplicar o mesmo método a equação da onda, ou seja, a um problema de segunda ordem no tempo.

Considere o sistema (7.3) e a seguinte mudança de variável: $u'(x, t) = w(x, t)$. Assim temos as equações:

$$\begin{cases} (w', v) + a(u, v) = (f, v) & \forall v \in V \\ (w(0), v) = (u'(0), v) = (u_1, v), \end{cases} \quad \begin{cases} (u', v) + (w, v) = 0, & \forall v \in V \\ (u(0), v) = (u_0, v), & \forall v \in V. \end{cases} \quad (7.19)$$

Definindo,

$$w_m(x, t) = \sum_{i=1}^m d_i(t) \psi_i(x) \quad u_m(x, t) = \sum_{i=1}^m \phi_i(t) \psi_i(x). \quad (7.20)$$

Substituindo no sistema aproximado do sistema (7.19), usando a definição de matriz (7.6), obtemos dois sistemas de equações ordinárias de primeira ordem, dados por:

$$P1 : \begin{cases} A\phi'(t) + Bd(t) = F(t) \\ \phi(0) = d'(0) = d_1 \end{cases} \quad P2 : \begin{cases} d'(t) - \phi(t) = 0, \\ d(0) = d_0 \end{cases} \quad (7.21)$$

Para cada um dos problemas, considere no tempo discreto $t_{n+1} = (n+1)\Delta t$ os pesos θ e σ e no tempo $t_n = n\Delta t$ o peso $(1-\theta)$ e $(1-\sigma)$, onde os parâmetros θ e σ são pertencentes ao intervalo $[0, 1]$. Assim nos tempos discretos $t_{n+\theta}$ e $t_{n+\sigma}$, definimos as médias ponderadas:

$$d^{n+\theta} = \theta d^{n+1} + (1-\theta)d^n \quad \text{e} \quad \phi^{n+\sigma} = \sigma \phi^{n+1} + (1-\sigma)\phi^n, \quad \theta, \sigma \in [0, 1].$$

Aplicando aos Problemas (P1) e (P2), que são de primeira ordem, os θ -métodos para cada dos problemas separadamente, obtemos:

$$P1 : \begin{cases} A \frac{(\phi^{n+1} - \phi^n)}{\Delta t} + Bd^{n+\theta} = F^{n+\theta} \\ \phi(0) = d'(0) = d_1 \end{cases} \quad P2 : \begin{cases} \frac{(d^{n+1} - d^n)}{\Delta t} = \phi^{n+\sigma}, \\ d(0) = d_0. \end{cases} \quad (7.22)$$

Desenvolvendo os termos, obtemos o sistema:

$$\begin{cases} A\phi^{n+1} = A\phi^n - \Delta t B(\theta d^{n+1} + (1-\theta)d^n) + \Delta t(\theta F^{n+1} + (1-\theta)F^n) \\ d^{n+1} = d^n + \Delta t(\sigma \phi^{n+1} + (1-\sigma)\phi^n), \quad \text{para } n = 0, 1, \dots \end{cases} \quad (7.23)$$

Sistema Linear

Note que o sistema (7.23) é um sistema acoplado, ou seja determina-se $\{d^{n+1}, \phi^{n+1}\}$ a partir do conhecimento da solução no tempo anterior $\{d^n, \phi^n\}$. Na forma matricial podemos escrever o seguinte sistema linear:

$$\mathbf{K}\Phi^{n+1} = \mathbf{L}\Phi^n + \mathbf{F}^{n+\theta}, \quad (7.24)$$

onde

$$\mathbf{K} = \begin{bmatrix} A & \theta\Delta t B \\ -\sigma\Delta t I & I \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} A & (\theta-1)\Delta t B \\ (1-\sigma)\Delta t I & I \end{bmatrix} \quad (7.25)$$

$$\Phi^n = \begin{bmatrix} \phi^n \\ d^n \end{bmatrix}, \quad \mathbf{F}^{n+\theta} = \begin{bmatrix} \theta F^{n+1} + (1-\theta)F^n \\ 0 \end{bmatrix}$$

Note que A e B são as mesmas matrizes definidas em (6.9) e I é a matriz identidade de mesma ordem de A e B . Logo a matriz dos coeficientes \mathbf{K} , denominada de matriz bloco, do sistema linear (7.24), tem ordem máxima de $2m \times 2m$.

O sistema linear (7.24) tem solução única, pois a matriz bloco \mathbf{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 $n = 0$ no sistema linear (7.24), temos

$$\mathbf{K}\Phi^1 = \mathbf{L}\Phi^0 + \mathbf{F}^{0+\theta}. \quad (7.26)$$

O vetor $\Phi^0 = [\phi^0, d^0]^t = [d_1, d_0]^t$ é conhecido pelo dados iniciais e como a força F é conhecido em todo tempo então $F^{0+\theta} = \theta F^1 + (1-\theta)F^0$ é conhecido e portanto podemos determinar o vetor $\mathbf{F}^{0+\theta}$. Sendo as matrizes blocos conhecidas então resolvendo o sistema linear determina-se o vetor $\Phi^1 = [\phi^1, d^1]^t$. E assim sucessivamente para $n = 1, 2, \dots, (N-1)$, obtém-se a solução em cada nível de tempo resolvendo o sistema linear bloco (7.24).

Observação

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 (7.10), mas que é condicionalmente estável.

3. Para $(\theta, \sigma) = (1, 1)$, 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, utilizando o método de Newmark para diversos valores de θ . Para os exemplos numéricos, o programa computacional utilizou como função base os polinômios lineares por partes. Nos exemplos a solução exata é conhecida e podemos então calcular o erro entre a solução numérica aproximada e a solução exata, nas normas definidas em (6.22), ou seja $L^\infty(0, T; L^2(0, 1))$ e $L^\infty(0, T; H^1(0, 1))$.

Considere o problema:

$$\begin{cases} u_{tt}(x, t) - \alpha u_{xx}(x, t) + \beta u(x, t) = f(x, t), & \forall (x, t) \in (0, 1) \times [0, 1] \\ u(0, t) = u(1, t) = 0, & \forall t \in [0, 1] \\ u(x, 0) = \sin(\pi x) & \forall x \in (0, 1), \\ u_t(x, 0) = 0 & \forall x \in (0, 1), \end{cases} \quad (7.27)$$

Para a força $f(x, t)$ é definida por,

$$f(x, t) = \begin{cases} 0, & \text{se } \lambda^2 = \alpha + \beta/(\pi)^2 \\ \left((-\lambda^2 + \alpha)\pi^2 + \beta \right) u(x, t) & \text{se } \lambda^2 \neq \alpha + \beta/(\pi)^2. \end{cases}$$

Nessas condições o problema (7.27) tem como solução única a função $u(x, t) = \sin(\pi x) \cos(\lambda \pi t)$. Nos exemplos numéricos, considere os dois casos separados, que serão chamados de exemplo 1 e exemplo 2.

Programa Computacional O resultados numéricos destes dois exemplos podem ser obtidos usando o Programa Computacional, denominado *Onda.cpp*, que está anexo no final do livro.

Exemplo 1: Considere o problema:

$$\begin{cases} u_{tt}(x, t) - \alpha u_{xx}(x, t) + \beta u(x, t) = 0, & \forall (x, t) \in (0, 1) \times [0, 1] \\ u(0, t) = u(1, t) = 0, & \forall t \in [0, 1] \\ u(x, 0) = \sin(\pi x) & \forall x \in (0, 1), \\ u_t(x, 0) = 0 & \forall x \in (0, 1), \end{cases} \quad (7.28)$$

Nesse caso a solução exata é $u(x, t) = \sin(\pi x) \cos(\lambda \pi t)$, com $\lambda = (\alpha + \beta/(\pi)^2)^{1/2}$. Para o exemplo numérico, foram considerados $\alpha = \beta = 1$. Assim $\lambda = (1 + 1/(\pi)^2)^{1/2} \approx 1.05$.

Na tabela 7.1, são mostrados os erros nas normas $L^\infty(0, T; L^2(0, 1))$ e $L^\infty(0, T; H^1(0, 1))$, do método de Newmark para diversos θ , com os parâmetros $\Delta t = 0.01$, $h = 0.05$ e $T = 0.5$ fixos.

	θ	$E_{L^\infty(0,T;L^2(0,1))}$	$E_{L^\infty(0,T;H^1(0,1))}$
$\Delta t = 0.01$ $h = 0.05$ $T = 0.5$	0	0.001113	0.001288
	0.25	0.000957	0.001108
	0.5	0.000801	0.000927
	0.75	0.000660	0.000764
	1.0	0.000536	0.000620

Tabela 7.1: Variando θ

Podemos notar, nas duas normas, que para θ crescente o erro decresce. Os resultados teóricos, não permitem afirmar que esse resultado é válido em geral, mas sob o ponto de vista numérico, podemos esperar um melhor resultado pois a matriz rigidez é "mais" bem condicionada, ou seja, é mais robusta. A figura 7.1 mostra o erro na norma $E_{L^\infty(0,T;L^2(0,1))}$, para cada parâmetro θ .

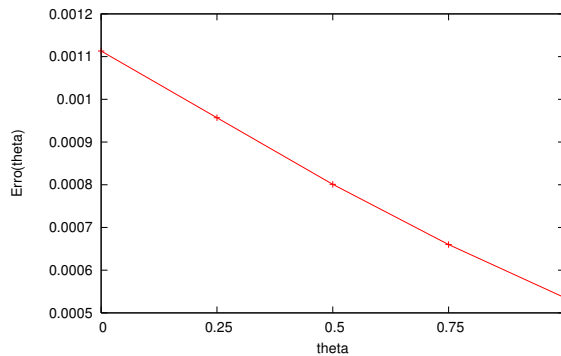


Figura 7.1: $E_{L^\infty(0,T;L^2(0,1))}$

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 .

	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: $E(\Delta t)$

Ordem de convergência

Seja, como anteriormente, $E_i = \max_{t \in [0,1]} \|E_i(t)\|_{L^2(0,1)}$, o erro associado a malha $h = (5 \times 2^{i+1})^{-1}$, para $i = 0, 1, \dots, N$ e 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

Métodos condicionalmente convergentes

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 de $r = (\Delta t/h) < 1$.

Esses resultados podem ser comprovados na tabela 7.5, onde estão exibidos o erro para alguns valores de θ , mostrando que quando $r = (\Delta t/h) = 1$, o método diverge e independe do tamanho da malha.

	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

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 a tabela 7.6 são mostrados os erros e ordem de convergência, tomando $\Delta = h/2$.

	$\Delta t = h/2$	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

A figura (7.2) foi construída para ilustrar geometricamente, no exemplo 1 as diferentes soluções aproximadas associadas ao método de Newmark, com parâmetros $\theta = 0; 1/4, 1/2; 3/4; 1.0$. Como esperado, 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 θ .

Na figura 7.3, apresentamos o erro absoluto entre a solução exata e as soluções numérica aproximada 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 $\lambda^2 = (\alpha + \beta/(\pi)^2)$. Note que foram necessários 1.000 iterações para obter a solução aproximada $u(x; T)$.

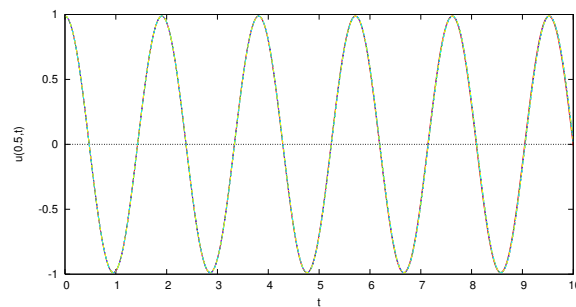


Figura 7.2: Solução aproximada $u_h(0.5, t)$

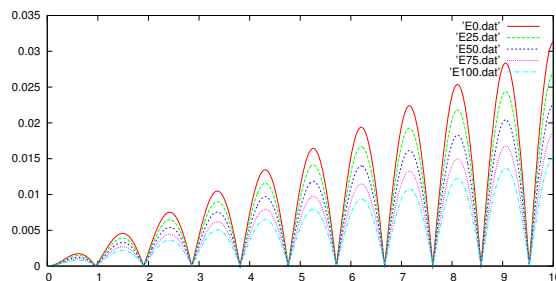


Figura 7.3: Erro absoluto do método de Newmark

Exemplo 2: Considere o problema:

$$\begin{cases} u_{tt}(x, t) - \alpha u_{xx}(x, t) + \beta u(x, t) = f(x, t), & \forall (x, t) \in (0, 1) \times [0, 1] \\ u(0, t) = u(1, t) = 0, & \forall t \in [0, 1] \\ u(x, 0) = \text{sen}(\pi x) & \forall x \in (0, 1), \\ u_t(x, 0) = 0 & \forall x \in (0, 1), \end{cases} \quad (7.29)$$

Considere em particular $\lambda = \alpha = \beta = 1$. Nesse caso para $f(x, t) = \text{sen}(\pi x) \cos(\pi t)$ a solução exata é igual a força, ou seja, $u(x, t) = \text{sen}(\pi x) \cos(\pi t)$.

Para esse exemplo, os valores consignados na tabela 7.7 são os erros nas normas $L^\infty(0, T; L^2(0, 1))$ e $L^\infty(0, T; H^1(0, 1))$, dependente do parâmetro θ do método de Newmark, com os parâmetros fixos $\Delta t = 0.01$, $h = 0.05$ e $T = 0.5$. Observamos novamente,

	θ	$E_{L^\infty(0,T;L^2(0,1))}$	$E_{L^\infty(0,T;H^1(0,1))}$
$\Delta t = 0.01$ $h = 0.05$ $T = 0.5$	0	0.001039	0.001203
	0.25	0.000906	0.001049
	0.5	0.000772	0.000894
	0.75	0.000652	0.000755
	1.0	0.000553	0.000640

Tabela 7.7: Variando θ

que ao aumentar o valor de θ diminuimos o erro nas duas normas. Essencialmente, valem os mesmos resultados do exemplo anterior.

7.4 Exercícios

1. **Bases Cúbicas e Hermite:** Considere o problema (7.27) do Exemplo 2.
 - (a) Determine a solução numérica aproximada para o Método da Diferença Central, usando como função base a função B-splines, definidas em (3.9). 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 (3.21) e (3.22).
 - (c) Suponha que a solução exata do problema (7.27) seja desconhecida. Verifique a ordem de convergência numérica, definida em (3.58) do Método da Diferença Central usando as bases dos itens anteriores e verifique se satisfaz a estimativa de erro (9.40).
2. **Fronteira de Neumann:** Considere o mesmo problema (7.27), com $\alpha = 3$ e $\beta = \pi^2$. Sabendo que a solução exata é $u(x, t) = \sin(\pi x) \cos(2\pi t)$ então o problema com as condições de fronteira de Neumann é dado por

$$\begin{cases} u_{tt}(x, t) - 3u_{xx}(x, t) + \pi^2 u(x, t) = 0, & \forall (x, t) \in (0, 1) \times [0, 1] \\ u_x(0, t) = \pi \cos(2\pi t) = -u_x(1, t), & \forall t \in [0, 1] \\ u(x, 0) = \sin(\pi x), & u_t(x, 0) = 0 \quad \forall x \in (0, 1), \end{cases} \quad (7.30)$$

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 $\theta = \{0.25, 0.5, 1.0\}$, é incondicionalmente estável, mas o Método da Diferença Central, ($\theta = 0$) é condicionalmente estável.
- (b) Para $\theta = \{0, 0.25, 0.5, 1.0\}$ e para as malhas com $h = \{0.1, 0.01, 0.001\}$, 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:** Definimos o problema homogêneo

$$\begin{cases} u_{tt}(\mathbf{x}, t) - \Delta u(\mathbf{x}, t) + u(\mathbf{x}, t) = 0, & \forall (\mathbf{x}, t) \in Q \times [0, 1] \\ u(\mathbf{x}, t) = 0, & \forall \mathbf{x} \in \partial\Omega \text{ e } \forall t \in [0, 1] \\ u(\mathbf{x}, 0) = \sin(\pi x) \sin(\pi y); \quad u_t(\mathbf{x}, 0) = 0 \quad \forall \mathbf{x} \in Q \end{cases} \quad (7.31)$$

onde $\mathbf{x} = (x, y) \in Q = [0, 1] \times [0, 1]$. Por $\partial\Omega = \bigcup_{i=1}^4 \Gamma_i$, onde

$$\begin{aligned} \Gamma_1 &= \{(x, 0) \in \partial\Omega; 0 \leq x \leq 1\} \\ \Gamma_2 &= \{(1, y) \in \partial\Omega; 0 \leq y \leq 1\} \\ \Gamma_3 &= \{(x, 1) \in \partial\Omega; 0 \leq x \leq 1\} \\ \Gamma_4 &= \{(0, y) \in \partial\Omega; 0 \leq y \leq 1\} \end{aligned}$$

Determine a solução numérica aproximada, usando o método de Newmark para $\theta = 0.5$, usando a malha uniforme $h_1 = 0.05$ no eixo-x e $h_2 = 0.05$ no eixo-y e o passo $\Delta t = 0.001$. Sabendo que a solução exata do problema (7.31) é dada por

$$u(\mathbf{x}, t) = \sin(\pi x) \sin(\pi y) \cos \sqrt{2 + (1/\pi^2)} \pi t,$$

determine o erro nas normas $L^2(0, 1; L^2(Q))$ e $L^2(0, 1; H_0^1(Q))$

Análise Numérica: Equação do Calor

Nesse capítulo provaremos os resultados de convergência dos métodos numéricos e algoritmos discutidos anteriormente, tanto no contexto do problema semidiscreto quanto no problemas totalmente discretos. Os resultados de convergência dessa seção, também são encontrados, por exemplo, nas referências [1, 9, 11, 17, 21].

8.1 Estimativa de erro

Nessa seção estamos interessado em fazer estimativas de erro para o problema semidiscreto e Problema Discreto da equação do calor.

8.1.1 Problema Semidiscreto

Vamos considerar as estimativas de erros em espaços de Sobolev, de forma análoga à feita em análise de erro do problema elíptico ou estacionário. Note que agora vamos considerar estimativas de erro entre a solução exata $u(x, t)$ e solução aproximada $u_h(x, t) = u_m(x, t)$, com a introdução da variável tempo t . No problema semidiscreto, considera-se a discretização do espaço, mas a variável tempo é considerada contínua. Dessa forma nessa seção vamos analisar a estimativa de erro $\|u(t) - u_h(t)\|$ para cada $t \in [0, T]$ fixo. Como vimos anteriormente, as funções bases w_i do subespaço V_m das soluções aproximadas são polinômios de grau k (Se $k = 1, 2$ e 3 temos, respectivamente as funções bases lineares, quadráticas e splines cúbicas) definidas em cada elemento finitos Ω_e . Para considerar explicitamente a dependência do grau do polinômio k , definimos o espaço de elementos finitos $V_m^k(\Omega) \cap C^0(\Omega)$, onde

$$V_m = V_m^k(\Omega) = \{v_h \in V; v_h^e \in P_k(\Omega_e)\}, \quad (8.1)$$

e 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 graus menores ou iguais a k na variável x .

O problema semidiscreto é formulado por: Determinar $u_h : [0, T] \rightarrow V_m$, solução do seguinte sistema,

$$\begin{cases} (u'_h(t), v_h) + a(u_h(t), v_h) = (f(t), v_h), \quad \forall v_h \in V_m^k, \\ (u_h(0), v_h) = (u_{0h}, v_h). \end{cases} \quad (8.2)$$

A formulação variacional do problema contínuo, ver (6.1), é dado por: Determinar $u : [0, T] \rightarrow H_0^1(0, 1)$, solução de

$$\begin{cases} (u'(t), v) + a(u(t), v) = (f(t), v), \quad \forall v \in H_0^1(0, 1) \\ (u(0), v) = (u_0, v), \quad \forall v \in H_0^1(0, 1). \end{cases} \quad (8.3)$$

Tempo Contínuo

O objetivo principal é fazer estimativa do erro entre $\|u(t) - u_h(t)\|$ em elementos finitos, onde $u_h(t)$ e $u(t)$ são as soluções dos problemas (8.2) e (8.3). Como V_m^k é um subespaço do $H_0^1(0, 1)$ então a igualdade (8.3) também é válida para $v_h \in V_m^k$. Subtraindo as equações obtemos

$$\begin{cases} (u'(t) - u'_h(t), v_h) + a(u(t) - u_h(t), v_h) = 0, \quad \forall v_h \in V_m^k \\ (u(0) - u_h(0), v_h) = 0, \quad \forall v_h \in V_m^k. \end{cases} \quad (8.4)$$

No problema elíptico (estacionário), a forma bilinear $a(., .)$ é independente do tempo e Céa (3.1), provou que a solução aproximada $u_h \in V_m^k$ obtida pelo método de Galerkin, é a projeção ortogonal de u no subespaço V_m^k com respeito a forma bilinear $a(., .)$, ou seja $a(u - u_h, v) = 0, \quad \forall v \in V_m^k$. Como a solução $u = u(x, t)$ do problema (8.3) é dependente do tempo, esse resultado não é mais verdadeiro. Mas ainda podemos definir a projeção ortogonal, denominada projeção de Rayleigh-Ritz, com respeito a forma bilinear $a(., .)$ da seguinte forma:

$$\begin{aligned} P : V &\longrightarrow V_m^k \\ u(t) &\longmapsto Pu(t) = \tilde{u}(t) \end{aligned}$$

satisfazendo,

$$a(u(t) - \tilde{u}(t), v_h) = 0, \quad \forall v_h \in V_m^k. \quad (8.5)$$

Em outras palavras, se o problema fosse estacionário (independente do tempo) da forma:

$$a(u, v) = (f, v), \quad \forall v \in V$$

a projeção \tilde{u} seria exatamente a solução $u_h \in V_m^k$, obtida pelo método de Galerkin.

A projeção $\tilde{u}(t)$ será usada como vetor intermediário entre a solução exata $u(t)$ do problema (8.3) e a solução aproximada $u_h(t)$ do problema aproximado (8.2). Note que sendo a projeção $\tilde{u}(t) \in V_m^k$ e $\varphi_i(x)$ uma base de $V_m = V_m^k$ podemos escrever,

$$\tilde{u}(x, t) = \sum_{i=1}^m u_i(t) \varphi_i(x), \quad (8.6)$$

e dessa forma a projeção $\tilde{u}(t)$ é também denominada interpolante de u . A principal razão de introduzir a projeção $\tilde{u}(t)$ é porque podemos decompor o erro em duas partes na forma: $\|u - u_h\| \leq \|u - \tilde{u}\| + \|\tilde{u} - u_h\|$ e são conhecidas estimativas de erro do primeiro termo do lado direito, restando apenas estimativas para o segundo termo no subespaço V_m^k . De fato, vimos no Corolário (3.3), (Estimativa Ótima em Espaços de Sobolev) o seguinte resultado:

$$\|u(t) - \tilde{u}(t)\|_m \leq ch^{k+1-m} \|u(t)\|_{k+1}, \quad \text{para cada } t \text{ fixo em } [0, T], \quad (8.7)$$

com $u(t) \in H^{k+1}(\Omega)$, $\tilde{u}(t) \in V_m^k$, onde $k \geq 1$ é o grau do polinômio interpolador da função base que gera o subespaço vetorial de dimensão finita e $m \leq k$.

Em [6], Douglas & Dupont provaram que valem as mesmas estimativas para as derivadas no tempo, como podemos ver no seguinte Lema:

Lema 8.1. *Sejam $\{u(t), u'(t), u''(t)\} \in H^{k+1}(\Omega)$, $\forall t \in [0, T]$ e seja $\tilde{u}(t)$ o interpolador de $u(t)$ em V_m^k , para cada $t \in [0, T]$. Então $u'(t)$ e $u''(t)$ satisfazem a mesma estimativa para o erro da interpolação de $u(t)$, ou seja*

$$\begin{aligned} \|u'(t) - \tilde{u}'(t)\|_m &\leq c_1 h^{k+1-m} \|u'(t)\|_{k+1} \\ \|u''(t) - \tilde{u}''(t)\|_m &\leq c_2 h^{k+1-m} \|u''(t)\|_{k+1} \end{aligned} \quad (8.8)$$

onde $m \leq k$, $k \geq 1$, c_1 e c_2 são constantes positivas independentes de $u(t)$ e h .

Demonstração: Ver em [6].

Com o auxílio dos resultados anteriores podemos agora fazer estimativas de erro entre a solução exata $u(., t)$ e a solução aproximada $u_h(., t)$.

Estimativa na norma $L^2(\Omega)$

Teorema 8.1. *Sejam $\{u, u'\} \in L^\infty(0, T; H_0^1 \cap H^{k+1})$. Então o erro entre a solução aproximada $u_h(x, t)$ e a solução exata $u(x, t)$ é dado por*

$$\|e(t)\|_0 = \|u(t) - u_h(t)\|_0 \leq c_1 h^{k+1} \|u(t)\|_{k+1}, \quad \text{para cada } t \in [0, T] \quad \text{onde } k \geq 1.$$

Demonstração:

Considere a decomposição do erro dado por

$$e(t) = u(t) - u_h(t) = \left(u(t) - \tilde{u}(t)\right) + \left(\tilde{u}(t) - u_h(t)\right) = \rho(t) + \xi(t),$$

onde $\tilde{u}(t)$ é o interpolante de $u(t)$ em V_m^k . Somando e subtraindo $\tilde{u}(t)$ em (8.4), obtemos

$$(\rho'(t) + \xi'(t), v_h) + a(\rho(t) + \xi(t), v_h) = 0, \quad \forall v_h \in V_m^k \quad (8.9)$$

De (8.5), temos que $a(\rho(t), v_h) = 0$ e assim a equação (8.9) pode ser escrita na forma:

$$\left(\xi'(t), v_h\right) + \left(\rho'(t), v_h\right) + a\left(\xi(t), v_h\right) = 0, \quad \forall v_h \in V_m^k. \quad (8.10)$$

Em particular podemos tomar $v_h = \xi(t) \in V_m^k$ em (8.10) obtendo-se;

$$\left(\xi'(t), \xi(t)\right) + \left(\rho'(t), \xi(t)\right) + a\left(\xi(t), \xi(t)\right) = 0, \quad \forall \xi(t) \in V_m^k. \quad (8.11)$$

Temos também que,

$$|\xi(t)|_0 \frac{d}{dt} |\xi(t)|_0 = \frac{1}{2} \frac{d}{dt} |\xi(t)|_0^2 = \left(\xi'(t), \xi(t)\right)$$

e

$$a\left(\xi(t), \xi(t)\right) = \alpha \|\xi(t)\|_1^2 + \beta |\xi(t)|_0^2 \geq 0.$$

Substituindo em (8.11) e desprezando o termo positivo $a(\cdot)$, obtemos

$$|\xi(t)|_0 \frac{d}{dt} |\xi(t)|_0 \leq \left(\rho'(t), \xi(t)\right) \quad (8.12)$$

Da desigualdade de Schwarz, temos

$$\left(\rho'(t), \xi(t)\right) \leq \int_0^1 |\rho'(t) \xi(t)| dx = |\rho'(t)|_0 |\xi(t)|_0$$

Substituindo em (8.12) e integrando de 0 a t , obtemos

$$|\xi(t)|_0 \leq |\xi(0)|_0 + \int_0^t |\rho'(t)|_0 \quad (8.13)$$

Vamos analisar os dois termos do lado direito da desigualdade (8.13).

De fato: Para o primeiro termo temos;

$$u(0) - u_h(0) = \xi(0) + \rho(0) \quad \Longleftrightarrow \quad |\xi(0)| \leq |u(0) - u_h(0)| + |\rho(0)|.$$

Mas, para $m = 0$ em (8.7) temos, $|\rho(0)|_0 \leq ch^{k+1}\|u(0)\|_{k+1}$, para $k \geq 1$. Em relação aos dados iniciais, podemos escolher $u(0) = u_h(0)$, mas isso não é necessário, é suficiente tomar os dados iniciais satisfazendo a condição:

$$|\xi(0)|_0 = |u(0) - u_h(0)|_0 \leq ch^{k+1}\|u(0)\|_{k+1}, \quad k \geq 1. \quad (8.14)$$

Por outro lado, usando (8.8) com $m = 0$, tem-se que

$$|\rho'(t)|_0 = |u'(t) - \tilde{u}'(t)|_0 \leq c_1 h^{k+1}\|u'(t)\|_{k+1} \quad (8.15)$$

Retornando a (8.13) e definindo $C = \max\{c, c_1\}$ obtemos

$$|\xi(t)|_0 \leq Ch^{k+1} \left(\|u(0)\|_{k+1} + \int_0^t \|u'(s)\|_{k+1} ds \right). \quad (8.16)$$

Note que do teorema fundamental do cálculo, temos:

$$u(t) = u(0) + \int_0^t u'(s) ds.$$

Segue então que

$$\|u(t)\|_{k+1} = \|u(0) + \int_0^t u'(s) ds\|_{k+1} \leq \|u(0)\|_{k+1} + \int_0^t \|u'(s)\|_{k+1} ds. \quad (8.17)$$

Substituindo em (8.16), obtém-se

$$|\xi(t)|_0 \leq Ch^{k+1} \left(\|u(t)\|_{k+1} \right), \quad (8.18)$$

Usando a desigualdade triangular tem-se que:

$$|u(t) - u_h(t)|_0 \leq |\rho(t)|_0 + |\xi(t)|_0 \leq \hat{C}h^{k+1} \left(\|u(t)\|_{k+1} \right),$$

o que conclui o teorema \square .

Note que para todo $t \in [0, T]$, por hipótese $u(t) \in H^{k+1}(\Omega)$ e $\xi(t) \in L^2(\Omega)$. Assim podemos tomar super essencial no tempo para obter a seguinte estimativa:

$$\|u - u_h\|_{L^\infty(0,T;L^2(\Omega))} \leq \hat{C}h^{k+1} \left(\|u\|_{L^\infty(0,T;H^{k+1}(\Omega))} \right). \quad (8.19)$$

Nas estimativas (8.16) e (8.18) para $\xi(t)$ na norma $L^2(\Omega)$, observe que o erro aumenta quando o tempo finito t cresce, desde que todos os termos do lado direito são positivos.

Estimativa na norma $H_0^1(\Omega)$

Vamos agora fazer uma estimativa para $\xi(t)$ na norma $H_0^1(\Omega)$.

Teorema 8.2. *Sejam $\{u, u'\} \in L^\infty(0, T; H_0^1 \cap H^{k+1})$. Então o erro entre a solução aproximada $u_h(x, t)$ e a solução exata $u(x, t)$ é dado por*

$$\|u(t) - u_h(t)\|_1 \leq c_1 h^k \|u(t)\|_{k+1}, \quad \text{para cada } t \in [0, T] \quad \text{onde } k \geq 1.$$

Demonstração:

Considere em (8.10), $v_h = \xi'(t) \in V_m^k$. Então,

$$|\xi'(t)|_0^2 + \frac{1}{2} \frac{d}{dt} \|\xi(t)\|_1^2 + (\rho'(t), \xi'(t)) = 0$$

Usando a desigualdade de Schwarz e elementar, obtém-se

$$(\rho'(t), \xi'(t)) \leq |\rho'(t)|_0 |\xi'(t)|_0 \leq \frac{1}{2} (|\rho'(t)|_0^2 + |\xi'(t)|_0^2)$$

Substituindo

$$|\xi'(t)|_0^2 + \frac{d}{dt} \|\xi(t)\|_1^2 \leq |\rho'(t)|_0^2.$$

Como $|\xi'(t)|^2 > 0$ então integrando os termos restantes de 0 a t, obtemos

$$\|\xi(t)\|_1^2 \leq \|\xi(0)\|_1^2 + \int_0^t |\rho'(t)|^2 \quad (8.20)$$

De forma análoga ao caso do $L^2(\Omega)$, temos na norma $H_0^1(\Omega)$ que

$$\|\xi(0)\|_1 \leq \|u(0) - u_h(0)\|_1 + \|\rho(0)\|_1.$$

Tomando $m = 1$ em (8.7) então temos a estimativa na norma $H_0^1(\Omega)$, dada por

$$\|\rho(0)\|_1 \leq ch^k \|u(0)\|_{k+1}, \quad k \geq 1.$$

Para os dados iniciais, como anteriormente tomamos,

$$\|\xi(0)\|_1 = \|u(0) - u_h(0)\|_1 \leq c_1 h^{k+1} \|u(0)\|_{k+1}, \quad k \geq 1.$$

Portanto temos que

$$\|\xi(0)\|_1 \leq (ch^k + c_1 h^{k+1}) \|u(0)\|_{k+1} \leq Ch^k \|u(0)\|_{k+1}, \quad k \geq 1,$$

onde podemos notar uma perda de precisão de ordem $\mathcal{O}(h)$ na norma $H_0^1(\Omega)$. Retornando a (8.20) obtemos

$$\|\xi(t)\|_1 \leq Ch^k \left(\|\xi(0)\|_1 + \int_0^t \|u'(s)\|_{k+1} ds \right) \quad (8.21)$$

Usando o teorema fundamental do cálculo e (8.17) obtemos

$$\|\xi(t)\|_1 \leq \widehat{C}h^k \|u(t)\|_{k+1} + Ch^{k+1} \|u(t)\|_{k+1} \leq \widehat{C}h^k \|u(t)\|_{k+1} \quad (8.22)$$

Portanto, usando a desigualdade triangular;

$$\|u(t) - u_h(t)\|_1 \leq \widehat{C}h^k \|u(t)\|_{k+1}, \quad (8.23)$$

e dessa forma concluímos o teorema \square

Como anteriormente, tomando super essencial no tempo para obter a seguinte estimativa:

$$\|u - u_h\|_{L^\infty(0,T;H^1(\Omega))} \leq \widehat{C}h^k \left(\|u\|_{L^\infty(0,T;H^{k+1}(\Omega))} \right). \quad (8.24)$$

As estimativas (8.16) e (8.18) e (8.21), (8.23) nas normas $L^2(\Omega)$ e $H_0^1(\Omega)$ respectivamente, não são estimativas ótimas, em razão do último termo ser acumulativo, ou seja, o erro aumenta quando o tempo t cresce. A seguir será feita uma estimativa ótima para a norma $L^2(\Omega)$.

Estimativa Ótima na Norma $L^2(\Omega)$

Teorema 8.3. *Nas mesmas condições do Teorema (8.1) temos,*

$$\|u(t) - u_h(t)\|_0 \leq c_1 h^{k+1} \|u(t)\|_{k+1} \quad \text{onde } k \geq 1.$$

Demonstração:

Seja λ_i uma sucessão de autovalores associados aos autovetores w_i do espaço $H_0^1(0,1)$, satisfazendo a seguinte condição:

$$-\frac{\partial^2 w_i}{\partial x^2} = \lambda_i w_i$$

Multiplicando ambos os lados e integrando em 0 a 1, temos:

$$|\nabla w_i|_0^2 = \lambda_i |w_i|_0^2 \geq \lambda_1 |w_i|_0^2, \quad (8.25)$$

pois os autovalores forma uma sucessão crescente e divergente para o infinito ∞ .

Observação: Os autovetores $w_i = \sqrt{2} \sin(i\pi x)$, por exemplo, é uma base ortonormal e completa em $H_0^1(0, 1)$ e os autovalores associados são $\lambda_i = (i\pi)^2$.

Considere agora a equação (8.12). Usando a equivalência de normas em $H_0^1(\Omega)$ e a desigualdade elementar (8.120) obtém-se

$$\frac{1}{2} \frac{d}{dt} |\xi(t)|_0^2 + \alpha |\nabla \xi(t)|_0^2 + \beta |\xi(t)|_0^2 \leq |\rho'(t)|_0 |\xi(t)|_0. \quad (8.26)$$

Substituindo (8.25) em (8.26), temos a seguinte forma

$$|\xi(t)|_0 \frac{d}{dt} |\xi(t)|_0 + (\alpha \lambda_1 + \beta) |\xi(t)|_0^2 \leq |\rho'(t)|_0 |\xi(t)|_0, \quad (8.27)$$

que é equivalente a

$$\frac{d}{dt} |\xi(t)|_0 + \lambda |\xi(t)|_0 \leq |\rho'(t)|_0, \quad (8.28)$$

onde $\lambda = \alpha \lambda_1 + \beta$.

Multiplicando (8.28) por $e^{\lambda t}$, obtemos

$$\frac{d}{dt} \left(e^{\lambda t} |\xi(t)|_0 \right) \leq e^{\lambda t} |\rho'(t)|_0. \quad (8.29)$$

Integrando em relação s ente 0 e 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 (8.14) e (8.15), obtemos

$$|\xi(t)|_0 \leq ch^{k+1} \left(e^{-\lambda t} \|u(0)\|_{k+1} + \int_0^t e^{-\lambda(t-s)} \|u(s)\|_{k+1} ds \right). \quad (8.30)$$

Temos agora uma estimativa ótima para o erro na norma $L^2(\Omega)$, pois o erro decai exponencialmente quando o tempo t cresce, concluindo a estimativa ótima \square

8.1.2 Problema Discreto

Nas estimativas anteriores para o problema semi-discreto, consideramos que o tempo t variava continuamente, mas na prática o tempo t varia discretamente. Em geral, nas equações de evolução do tipo parabólico ou hiperbólico, são usados o método de elementos finitos para o espaço e o método das diferenças finitas no tempo. Uma questão natural, é porque não utilizar também os elementos finitos na variável tempo. Matematicamente é perfeitamente razoável aplicar o método de elementos finitos nas duas variáveis e 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, nesta seção utilizaremos o Método das Diferenças Finitas como aproximação para a derivada no tempo e o método dos elementos finitos na variável espacial x para determinar a solução numérica do problema (8.3), e obter as estimativas de erro $\|u - u_h\|$ na normas em espaços de Sobolev.

Resultados Preliminares

Notação e definição

Inicialmente vamos introduzir o conceito de norma discreta para uma função w dependente do tempo t . Denotaremos por $w^n = w(t_n)$ o valor de w no tempo discreto $t_n = n\Delta t$ e

$$\begin{aligned} w^{n+1/2} &= (w^{n+1} + w^n)/2, \\ w^{*n} &= \theta w^{n+1} + (1 - 2\theta)w^n + \theta w^{n-1}, \\ w^{n+\theta} &= \theta w^{n+1} + (1 - \theta)w^n, \\ \delta w^{n+\frac{1}{2}} &= (w^{n+1} - w^n)/\Delta t; \\ \delta w^{n-\frac{1}{2}} &= (w^n - w^{n-1})/\Delta t; \\ \delta w^n &= (w^{n+\frac{1}{2}} - w^{n-\frac{1}{2}})/\Delta t = (w^{n+1} - w^{n-1})/2\Delta t; \\ \delta^2 w^n &= (w^{n+1} - 2w^n + w^{n-1})/(\Delta t)^2; \end{aligned} \tag{8.31}$$

onde $\theta \in [0, 1]$ denota o parâmetro de Newmark.

Para $i = 1, 2, \dots, m$ e a malha uniforme $h = x_{i+1} - x_i$ no espaço, definimos a norma discreta em $L^2(\Omega)$ e $H^1(\Omega)$ por

$$\begin{aligned} \|w\|_{L^2(\Omega)} &= \left(h \sum_{i=1}^m |w(x_i)|^2 \right)^{1/2} \\ \|w\|_{H^1(\Omega)} &= \left(h \sum_{i=1}^m |w(x_i)|^2 \right)^{1/2} + \left(h \sum_{i=1}^m \left| \frac{\partial w(x_i)}{\partial x} \right|^2 \right)^{1/2} \end{aligned}$$

Definimos as normas discretas dos espaços $L^2(0, T; H)$ e $L^\infty(0, T; H)$, onde H pode ser $L^2(0, 1)$, $H_0^1(0, 1)$ ou $H^2(0, 1)$,

$$\begin{aligned} \|w\|_{\widehat{L}^\infty(0, T; H)} &= \max\{\|w^n\|_H, \quad n = 0, 1, \dots, N-1\}, \\ \|w\|_{\widetilde{L}^\infty(0, T; H)} &= \max\{\|w^{n+\frac{1}{2}}\|_H, \quad n = 0, 1, \dots, N-1\}, \\ \|w\|_{\widehat{L}^2(0, T; H)}^2 &= \Delta t \sum_{n=0}^{N-1} \|w^n\|_H^2, \\ \|w\|_{\widetilde{L}^2(0, T; H)}^2 &= \Delta t \sum_{n=0}^{N-1} \|w^{n+\frac{1}{2}}\|_H^2, \end{aligned} \quad (8.32)$$

onde os valores para as normas \widehat{L} são definidos para os valores $(8.31)_1$.

Combinando as notações apropriadamente, podemos por exemplo, definir a seguinte norma

$$\|\delta w\|_{\widetilde{L}^2(0, T; W)}^2 = \sum_{n=0}^{N-1} \|\delta w^{n+\frac{1}{2}}\|_W^2 \quad (8.33)$$

Para estabelecer estimativas de erro no tempo discreto, vamos enunciar os seguintes Lemas:

Lema 8.2. *Sejam φ e ψ funções não negativas definidas no conjunto discreto $\{t^n, n = 0, \dots, N\}$ e ψ uma função não decrescente. Se*

$$\varphi^\tau \leq \psi^\tau + c\Delta t \sum_{n=0}^{\tau-1} \varphi^n, \quad \forall \tau = 0, \dots, N, \quad (8.34)$$

então

$$\varphi^\tau \leq \psi^\tau \exp(c\tau), \quad \forall \tau = 0, \dots, N; \quad (8.35)$$

onde c é uma constante positiva independente de τ .

Note que o Lema anterior é o análogo discreto do Lema de Gronwall. O próximo resultado são para funções definidas somente num conjunto de tempos discretos.

Lema 8.3. *(Dupont) Seja w uma vetor dependente do tempo tal que $w^n \in L^2(\Omega)$, para $n = 0, \dots, N$ e suponha que o incremento no tempo $\Delta t < 1$. Então*

$$\|w^{n-1/2}\|_0^2 \leq c \left(\|w^{1/2}\|_0^2 + \|\delta w\|_{\widetilde{L}^2(0, t^n; L^2(\Omega))}^2 \right), \quad (8.36)$$

$$\|w^n\|_0^2 \leq \bar{c} \left(\|w^0\|_0^2 + \|\delta w\|_{\widetilde{L}^2(0, t^n; L^2(\Omega))}^2 \right), \quad (8.37)$$

onde c e \bar{c} são constantes positivas independentes de w .

Alguns resultados da teoria de aproximação serão introduzidos com o objetivo de estabelecer estimativas de erro no tempo discreto. Nesses resultados, p pode ser 2 ou ∞ e H qualquer um dos espaços de Hilbert, $H_0^1(\Omega)$, $H^2(\Omega)$ ou $L^2(\Omega)$.

- (i) O operador δ , como definido em (8.31)₆ é uma aproximação para a primeira derivada no tempo, então a norma $\|\delta w\|$ pode ser estimada em termos de w' , na forma

$$\|\delta w\|_{\hat{L}^p(0, T; H)} \leq c \|w'\|_{L^p(0, T; H)}. \quad (8.38)$$

- (ii) De forma análoga, o operador δ^2 , como definido em (8.31)₇ é uma aproximação para a segunda derivada no tempo, então a norma $\|\delta^2 w\|$ pode ser estimada em termos de w'' , na forma

$$\|\delta^2 w\|_{\hat{L}^p(0, T; H)} \leq \bar{c} \|w''\|_{L^p(0, T; H)}. \quad (8.39)$$

- (iii) Foi introduzido o operador $*$ em (8.31)₂, com pesos θ , $(1 - 2\theta)$ e θ em nós consecutivos, então a norma w^* pode ser estimada em função de w da seguinte forma

$$\|w^*\|_{\hat{L}^p(0, T; H)} \leq \hat{c} \|w\|_{L^p(0, T; H)}. \quad (8.40)$$

- (iv) Usando (8.38) e (8.40) podemos estimar a norma $\|\delta w^*\|$ em função de w' , por

$$\|\delta w^*\|_{\hat{L}^p(0, T; H)} \leq \tilde{c} \|w'\|_{L^p(0, T; H)}. \quad (8.41)$$

Note que ao aplicar as normas (8.31)_{1,3} em w^* , é necessário conhecer o valor de w^{*0} , ou seja w^* em $t = 0$. Por simplicidade, toma-se $w^{*0} = w^0$.

- (v) (Somatório por partes)

Finalizando os resultados preliminares considere as seguintes fórmulas de somatório por partes, que é o análogo discreto da integração por partes dadas por:

$$\Delta t \sum_{n=1}^{N_1-1} \varphi^n \delta \psi^{n-1/2} = \varphi^{N_1-1} \psi^{N_1-1} - \varphi^1 \psi^0 - \Delta t \sum_{n=1}^{N_1-2} \psi^n \delta \varphi^{n+\frac{1}{2}} \quad (8.42)$$

e

$$\Delta t \sum_{n=1}^{N_1-1} \varphi^n \delta \psi^{n+\frac{1}{2}} = \varphi^{N_1-1} \psi^{N_1} - \varphi^1 \psi^1 - \Delta t \sum_{n=2}^{N_1-1} \psi^n \delta \varphi^{n-1/2}, \quad (8.43)$$

onde $2 \leq N_1 \leq N$.

Decomposição do Erro

Nas estimativas de erro que se seguem, considere a forma bilinear $a(.,.)$, definida em (6.2) por

$$a(u, v) = \alpha \int_0^1 u_x v_x dx + \beta \int_0^1 uv dx. \quad (8.44)$$

Considere a decomposição do erro na forma

$$e = u - u_h = (u - \tilde{u}) + (\tilde{u} - u_h) = \rho + \xi, \quad (8.45)$$

onde \tilde{u} é o interpolante de u em V_m^k , ou equivalentemente \tilde{u} é conhecida como projeção de Rayleigh-Ritz, ou seja, a projeção ortogonal da solução exata u em V_m^k , com respeito a forma bilinear $a(.,.)$ da seguinte forma,

$$a(u(t) - \tilde{u}(t), v_h) = 0, \quad \forall t \in [0, T], \quad \forall v_h \in V_m^k, \quad (8.46)$$

No Teorema (8.7) da próxima seção, são mostrados resultados de existência e unicidade de solução para o Problema (6.1). Para obter as estimativas de erro no tempo discreto, são necessários resultados adicionais mais regulares que daremos na seguinte hipótese:

HC: Suponha que a força f dada, a solução u do Problema (6.1) e suas derivadas tenham a seguinte regularidade:

$$\left| \begin{array}{l} f \in L^2(0, T; H^{k+1}(0, 1)), \quad u \in L^\infty(0, T; H_0^1(0, 1) \cap H^{k+1}(0, 1)) \\ \text{com as derivadas } \{u_t, u_{tt}, u_{ttt}\} \in L^\infty(0, T; H^{k+1}(0, 1)). \end{array} \right. \quad (8.47)$$

Método de Euler Regressivo

Do sistema de equações diferenciais ordinárias (6.11), deduzimos o método iterativo (6.14) dado por

$$(A + \Delta t B)u^n = Au^{n-1} + \Delta t F^n = b^n, \quad n = 1, 2, \dots, N,$$

onde estamos substituindo a variável d por u .

No próximo teorema mostraremos que o Método de Euler regressivo é incondicionalmente estável, isto é, é estável independente da relação entre o Δt e h . Além disso, mostraremos a ordem de convergência do método, ou seja, as estimativas de erro em normas de espaço de Sobolev.

Teorema 8.4. *Sob a hipótese (8.47), e os dados iniciais $u_0 \in H_0^1(0, 1) \cap H^{k+1}(0, 1)$ então a estimativa de erro para o problema discreto obtida pelo Método de Euler regressivo (6.14), é dado por:*

$$\begin{aligned} \|e\|_{L^\infty(0,T;L^2(0,1))} &= \|u - u_h\|_{L^\infty(0,T;L^2(0,1))} \leq c_2 (h^{k+1} + \Delta t) \\ \|e\|_{L^\infty(0,T;H_0^1(0,1))} &= \|u - u_h\|_{L^\infty(0,T;H_0^1(0,1))} \leq c_1 (h^k + \Delta t), \end{aligned} \quad (8.48)$$

onde c_1 e c_2 são constantes positivas, independentes de h e Δt .

Demonstração: Vamos demonstrar inicialmente a estimativa na norma $L^\infty(0, T; L^2(0, 1))$. De fato, o erro entre a solução aproximada u_h^n e a solução exata u^n é definido por:

$$e^n = u^n - u_h^n, \quad n = 0, 1, \dots, N \quad (8.49)$$

Como anteriormente, introduzimos a função de interpolação \tilde{u}^n , $n = 0, 1, \dots, N$ em V_m^k satisfazendo a condição de ortogonalidade (8.46)

$$a(u^n - \tilde{u}^n, v) = 0, \quad \forall v \in V_m^k \quad \text{e para todo } n = 0, 1, \dots, N. \quad (8.50)$$

Podemos decompor o erro na forma

$$e^n = u^n - u_h^n = (u^n - \tilde{u}^n) + (\tilde{u}^n - u_h^n) = \rho^n + \xi^n \quad (8.51)$$

Desde que estimativas de erro para o termo ρ^n são conhecidos por (8.8) então o objetivo é estimar o termo ξ^n , para $n = 0, 1, \dots, N$.

A formulação fraca do problema (8.3) calculada no tempo discreto t_n é dada por:

$$\left(\frac{d}{dt} u^n, v \right) + a(u^n, v) = (f^n, v), \quad \forall v \in H_0^1(0, 1)$$

Somando e subtraindo o termo $(\delta u^{n-\frac{1}{2}}, v)$, temos

$$(\delta u^{n-\frac{1}{2}}, v) + a(u^n, v) = (f^n, v) + \left(\delta u^{n-\frac{1}{2}} - \frac{d}{dt} u^n, v \right) \quad (8.52)$$

Considere agora o sistema aproximado no subespaço V_m^k , dado por

$$(\delta u_h^{n-\frac{1}{2}}, v_h) + a(u_h^n, v_h) = (f^n, v_h), \quad \forall v_h \in V_m^k, \quad \text{para } n = 1, 2, \dots, N \quad (8.53)$$

Tomando em particular $v = v_h \in V_m^k$ em (8.52) e fazendo a diferença com (8.53), obtemos

$$(\delta u^{n-\frac{1}{2}} - \delta u_h^{n-\frac{1}{2}}, v_h) + a(u^n - u_h^n, v_h) = \left(\delta u^{n-\frac{1}{2}} - \frac{d}{dt} u^n, v \right) \quad (8.54)$$

Somando e subtraindo o interpolante $\delta\tilde{u}^{n-\frac{1}{2}}$ e $\tilde{u}^n \in V_m^k$ no primeiro e segundo termo respectivamente e usando a decomposição do erro (8.51), obtemos

$$(\delta\xi^{n-\frac{1}{2}}, v_h) + a(\xi^n, v_h) = (\delta u^{n-\frac{1}{2}} - \frac{d}{dt} u^n, v_h) - (\delta\rho^{n-\frac{1}{2}}, v_h), \quad \forall v_h \in V_m^k. \quad (8.55)$$

Note que da projeção elíptica (8.50), $a(\rho^n, v_h) = 0$.

Tomando em particular $v_h = \xi^n$ em (8.55), obtemos

$$(\delta\xi^{n-\frac{1}{2}}, \xi^n) + a(\xi^n, \xi^n) = -\left((\delta\rho^{n-\frac{1}{2}}, \xi^n) + \left(\frac{d}{dt} u^n - \delta u^{n-\frac{1}{2}}, \xi^n\right)\right) \quad (8.56)$$

Mas

$$(\delta\xi^{n-\frac{1}{2}}, \xi^n) = \left(\frac{1}{\Delta t}(\xi^n - \xi^{n-1}), \xi^n\right) \geq \frac{1}{\Delta t} (\|\xi^n\|_0 - \|\xi^{n-1}\|_0) \|\xi^n\|_0, \quad (8.57)$$

e temos também

$$-\left((\delta\rho^{n-\frac{1}{2}}, \xi^n) + \left(\frac{d}{dt} u^n - \delta u^{n-\frac{1}{2}}, \xi^n\right)\right) \leq \left(\|\delta\rho^{n-\frac{1}{2}}\|_0 + \left\|\frac{d}{dt} u^n - \delta u^{n-\frac{1}{2}}\right\|_0\right) \|\xi^n\|_0.$$

Substituindo os dois últimos termos em (8.56), obtém-se

$$\frac{1}{\Delta t} (\|\xi^n\|_0 - \|\xi^{n-1}\|_0) \|\xi^n\|_0 + a(\xi^n, \xi^n) \leq \left(\|\delta\rho^{n-\frac{1}{2}}\|_0 + \left\|\frac{d}{dt} u^n - \delta u^{n-\frac{1}{2}}\right\|_0\right) \|\xi^n\|_0 \quad (8.58)$$

Desde que

$$a(\xi^n, \xi^n) = \alpha\|\xi^n\|_1^2 + \beta|\xi^n|_0^2 \geq 0,$$

então multiplicando a equação (8.58) por Δt e eliminando o termo comum $\|\xi^n\|_0$, obtém-se

$$\|\xi^n\|_0 \leq \|\xi^{n-1}\|_0 + \Delta t \left(\|\delta\rho^{n-\frac{1}{2}}\|_0 + \left\|\frac{d}{dt} u^n - \delta u^{n-\frac{1}{2}}\right\|_0\right) \quad (8.59)$$

Fazendo o somatório para $n = 1, 2, \dots, N$, observando o cancelamento dos termos comuns nos somatórios obtemos

$$\|\xi^N\|_0 \leq \|\xi^0\|_0 + \Delta t \left(\sum_{n=1}^N \|\delta\rho^{n-\frac{1}{2}}\|_0 + \sum_{n=1}^N \left\|\frac{d}{dt} u^n - \delta u^{n-\frac{1}{2}}\right\|_0\right) \quad (8.60)$$

Note que

$$\xi^0 = \tilde{u}^0 - u_h^0 = \tilde{u}^0 - u^0 + u^0 - u_h^0 = -\rho^0 + u^0 - u_h^0$$

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 então podemos tomar a solução aproximada $u_h^0 = u_h(0)$ igual a $u(0)$, ou satisfazendo pelo menos a estimativa:

$$\|u^0 - u_h^0\|_0 \leq ch^{k+1} \|u_0\|_{k+1}, \quad k \geq 1.$$

Assim temos que

$$\|\xi^N\|_0 \leq ch^{k+1} \|u_0\|_{k+1} + \Delta t \left(\sum_{n=1}^{N-1} \|\delta \rho^{n-\frac{1}{2}}\|_0 + \sum_{n=1}^{N-1} \left\| \frac{d}{dt} u^n - \delta u^{n-\frac{1}{2}} \right\|_0 \right) \quad (8.61)$$

Vamos analisar os dois termos do somatório, definidos em (8.61).

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 $n = 1, 2, \dots, N$, multiplicando por Δt e usando usando (8.8), obtemos

$$\Delta t \sum_{n=1}^N \|\delta \rho^{n-\frac{1}{2}}\|_0 \leq \int_{t_0}^{t_N} \|\rho'(s)\|_0 ds \leq ch^{k+1} \int_{t_0}^{t_N} \|u'(s)\|_{k+1} ds. \quad (8.62)$$

Para o segundo termo do somatório, temos

$$\frac{d}{dt} u^n - \delta u^{n-\frac{1}{2}} = \frac{d}{dt} u^n - \frac{1}{\Delta t} (u^n - u^{n-1}).$$

Pela expansão de Taylor, temos

$$\begin{aligned} u(t_n) &= u(t_{n-1}) + \Delta t u'(t_{n-1}) + \int_{t_{n-1}}^{t_n} (t_n - s) u''(s) ds, \\ u(t_{n-1}) &= u(t_n) - \Delta t u'(t_n) + \int_{t_n}^{t_{n-1}} (t_{n-1} - s) u''(s) ds. \end{aligned}$$

Logo de forma equivalente, podemos escrever

$$\begin{aligned} \delta u^{n-\frac{1}{2}} &= \frac{1}{\Delta t} (u^n - u^{n-1}) = u_t^{n-1} + \frac{1}{\Delta t} \int_{t_{n-1}}^{t_n} (t_n - s) u''(s) ds, \\ \delta u^{n-\frac{1}{2}} &= \frac{1}{\Delta t} (u^n - u^{n-1}) = u_t^n + \frac{1}{\Delta t} \int_{t_{n-1}}^{t_n} (t_{n-1} - s) u''(s) ds. \end{aligned} \quad (8.63)$$

Note que integrando por partes, obtemos

$$\begin{aligned} \frac{1}{\Delta t} \int_{t_{n-1}}^{t_n} (t_{n-1} - s) u''(s) ds &= \frac{1}{\Delta t} \left((t_{n-1} - s) u'(s) \Big|_{t_{n-1}}^{t_n} + \int_{t_{n-1}}^{t_n} u'(s) ds \right) \\ &= -u_t^n + \frac{1}{\Delta t} (u^n - u^{n-1}) = -u_t^n + \delta u^{n-\frac{1}{2}} \end{aligned}$$

Dessa forma temos que

$$\begin{aligned} \|u_t^n - \delta u^{n-\frac{1}{2}}\|_0 &\leq \frac{1}{\Delta t} \int_{t_{n-1}}^{t_n} \|(s - t_{n-1})u''(s)\|_0 ds \\ &\leq \|u''\|_{L^\infty(t_{n-1}, t_n; L^2(0,1))} \frac{1}{\Delta t} \int_{t_{n-1}}^{t_n} \|s - t_{n-1}\|_0 ds \leq \frac{\Delta t}{2} \|u''\|_{L^\infty(t_{n-1}, t_n; L^2(0,1))} \end{aligned}$$

onde usamos que $|s - t_{n-1}| \leq \Delta t$, para $s \in [t_{n-1}, t_n]$.

Fazendo a soma para $n = 1, 2, \dots, N$ obtemos

$$\sum_{n=1}^N \|u_t^n - \delta u^{n-\frac{1}{2}}\|_0 \leq \frac{\Delta t}{2} \sum_{n=1}^N \|u''\|_{L^\infty(t_{n-1}, t_n; L^2(0,1))} \leq C \|u''\|_{L^\infty(0, T; L^2(0,1))} \quad (8.64)$$

Substituindo (8.62) e (8.64) em (8.61) obtemos

$$\|\xi^N\| \leq ch^{k+1} \|u_0\|_{k+1} + \hat{C} \left\{ h^{k+1} \|u'\|_{L^\infty(0, T; H^{k+1})} + \Delta t \|u''\|_{L^\infty(0, T; L^2(0,1))} \right\} \quad (8.65)$$

onde usamos a relação (8.17) na última desigualdade.

Usando a hipótese (8.47), conclui-se que

$$\|\xi^N\|_0 = \|\tilde{u}^N - u_h^N\|_0 \leq c_1 h^{k+1} + c_2 \Delta t = \mathcal{O}(h^{k+1} + \Delta t). \quad (8.66)$$

Por outro lado, fazendo $m = 0$ em (8.8), temos

$$\|\rho^N\|_0 = \|u(t) - \tilde{u}(t)\|_0 \leq ch^{k+1} \|u(t)\|_{k+1}, \quad \text{para cada } t \text{ fixo em } [0, T],$$

Então usando a decomposição do erro (8.45), obtemos a seguinte estimativa,

$$\|e^N\|_0 = \|u^N - \tilde{u}^N\|_0 + \|\tilde{u}^N - u_h^N\|_0 = \|\rho^N\|_0 + \|\xi^N\|_0 = \mathcal{O}(h^{k+1} + \Delta t) \quad (8.67)$$

o que conclui a estimativa na norma $L^\infty(0, T; L^2(0, 1))$, ou seja,

$$\|e\|_{L^\infty(0, T; L^2(0,1))} = \|u - u_h\|_{L^\infty(0, T; L^2(0,1))} \leq c_2 (h^{k+1} + \Delta t)$$

De forma análoga obtém-se a estimativa $L^\infty(0, T; H_0^1(0, 1))$. Observe nesse caso, que por causa da relação

$$\|\rho^N\|_1 = \|u(t) - \tilde{u}(t)\|_1 \leq ch^k \|u(t)\|_{k+1}$$

tem-se uma perda de precisão de ordem h . Assim conclui-se a prova do teorema \square

Método de Crank-Nicolson

Teorema 8.5. *Sob a hipótese (8.47), e os dados iniciais $u_0 \in H_0^1(0, 1) \cap H^{k+1}(0, 1)$ então a solução numérica obtida pelo método de Crank-Nicolson (6.17), tem a seguinte estimativa de erro:*

$$\begin{aligned} \|e\|_{L^\infty(0,T;L^2(0,1))} &= \|u - u_h\|_{L^\infty(0,T;L^2(0,1))} \leq c_2 (h^{k+1} + (\Delta t)^2) \\ \|e\|_{L^\infty(0,T;H_0^1(0,1))} &= \|u - u_h\|_{L^\infty(0,T;H_0^1(0,1))} \leq c_1 (h^k + (\Delta t)^2), \end{aligned} \quad (8.68)$$

onde c_1 e c_2 são constantes positivas, independentes de h e Δt .

Demonstração:

Vamos inicialmente demonstrar a estimativa em $L^\infty(0, T; L^2(0, 1))$.

A formulação fraca do problema é dada por

$$(u_t(t), v)_0 + a(u(t), v) = (f, v)_0, \quad \forall v \in H_0^1(0, 1) \quad (8.69)$$

Fazendo $t = (n + \frac{1}{2})$ em (8.69), obtemos

$$(u_t^{n+\frac{1}{2}}, v)_0 + a(u^{n+\frac{1}{2}}, v) = (f^{n+\frac{1}{2}}, v)_0, \quad \forall v \in H_0^1(0, 1)$$

Somando e subtraindo o termo $(\delta u^{n+\frac{1}{2}}, v)$, obtemos

$$(\delta u^{n+\frac{1}{2}}, v)_0 + a(u^{n+\frac{1}{2}}, v) = (f^{n+\frac{1}{2}} + g^n, v)_0, \quad \forall v \in H_0^1(0, 1) \quad (8.70)$$

onde $g^n = (\delta u^{n+\frac{1}{2}} - u_t^{n+\frac{1}{2}})$ e $\delta u^{n+\frac{1}{2}} = \frac{1}{\Delta t}(u^{n+1} - u^n)$ é uma aproximação para u_t com erro de $\mathcal{O}(\Delta t)$.

Por outro lado, para o problema aproximado, a formulação é dada por

$$(\delta u_h^{n+\frac{1}{2}}, v_h)_0 + a(u_h^{n+\frac{1}{2}}, v_h) = (f^{n+\frac{1}{2}}, v_h)_0, \quad \forall v_h \in V_m^k \quad (8.71)$$

Tomando, em particular, $v = v_h \in V_m^k$ em (8.70) e subtraindo da equação (8.71), obtemos

$$(\delta e^{n+\frac{1}{2}}, v_h)_0 + a(e^{n+\frac{1}{2}}, v_h) = (g^n, v_h)_0, \quad \forall v_h \in V_m^k \quad (8.72)$$

Somando e subtraindo os termo interpolantes, $\delta \tilde{u}^{n+\frac{1}{2}}$ e $\tilde{u}^{n+\frac{1}{2}} \in V_m^k$ no primeiro e segundo termo, respectivamente e usando a decomposição do erro (8.51), obtém-se

$$(\delta \xi^{n+\frac{1}{2}}, v_h)_0 + (\delta \rho^{n+\frac{1}{2}}, v_h)_0 + a(\xi^{n+\frac{1}{2}}, v_h) = (g^n, v_h)_0, \quad \forall v_h \in V_m^k \quad (8.73)$$

onde usamos a ortogonalidade da projeção em V_m^k , ou seja,

$$a(\rho^{n+\frac{1}{2}}, v_h) = \frac{1}{2} a(\rho^{n+1} + \rho^n, v_h) = 0.$$

Tomando em particular $v_h = \xi^{n+\frac{1}{2}}$ em (8.73), obtemos

$$(\delta \xi^{n+\frac{1}{2}}, \xi^{n+\frac{1}{2}})_0 + a(\xi^{n+\frac{1}{2}}, \xi^{n+\frac{1}{2}}) = (g^n - \delta \rho^{n+\frac{1}{2}}, \xi^{n+\frac{1}{2}})_0 \quad (8.74)$$

Do primeiro termo de (8.74) temos

$$\begin{aligned} (\delta \xi^{n+\frac{1}{2}}, \xi^{n+\frac{1}{2}})_0 &= \frac{1}{2\Delta t} (\xi^{n+1} - \xi^n, \xi^{n+1} + \xi^n)_0 = \frac{1}{2\Delta t} (\|\xi^{n+1}\|_0^2 - \|\xi^n\|_0^2) \\ &= \frac{1}{2\Delta t} (\|\xi^{n+1}\|_0 - \|\xi^n\|_0) (\|\xi^{n+1}\|_0 + \|\xi^n\|_0) \end{aligned} \quad (8.75)$$

Temos também que

$$a(\xi^{n+\frac{1}{2}}, \xi^{n+\frac{1}{2}}) = \alpha \|\xi^{n+\frac{1}{2}}\|_1^2 + \beta \|\xi^{n+\frac{1}{2}}\|_0^2 \geq 0 \quad (8.76)$$

Assim, descartando o termo positivo (8.76), obtemos que

$$\frac{1}{2\Delta t} (\|\xi^{n+1}\|_0 - \|\xi^n\|_0) (\|\xi^{n+1}\|_0 + \|\xi^n\|_0) \leq (g^n - \delta \rho^{n+\frac{1}{2}}, \xi^{n+\frac{1}{2}})_0$$

Usando a desigualdade elementar tem-se:

$$\begin{aligned} (g^n, \xi^{n+\frac{1}{2}})_0 &\leq \frac{1}{2} \|g^n\|_0 \|\xi^{n+\frac{1}{2}}\|_0 \\ (-\delta \rho^{n+\frac{1}{2}}, \xi^{n+\frac{1}{2}})_0 &\leq \frac{1}{2} \|\delta \rho^{n+\frac{1}{2}}\|_0 \|\xi^{n+\frac{1}{2}}\|_0 \end{aligned}$$

Substituindo e multiplicando a desigualdade por $2\Delta t$ obtemos:

$$\begin{aligned} (\|\xi^{n+1}\|_0 - \|\xi^n\|_0) (\|\xi^{n+1}\|_0 + \|\xi^n\|_0) &\leq \Delta t (\|g^n\|_0 + \|\delta \rho^{n+\frac{1}{2}}\|_0) \|\xi^{n+\frac{1}{2}}\|_0 \\ &\leq \Delta t (\|g^n\|_0 + \|\delta \rho^{n+\frac{1}{2}}\|_0) (\|\xi^{n+1}\|_0 + \|\xi^n\|_0) \end{aligned}$$

Cancelando o termo comum obtemos;

$$(\|\xi^{n+1}\|_0 - \|\xi^n\|_0) \leq \Delta t (\|g^n\|_0 + \|\delta \rho^{n+\frac{1}{2}}\|_0)$$

Fazendo a soma de $n = 0$ a $n = N - 1$, temos

$$\|\xi^N\|_0 \leq \|\xi^0\|_0 + \Delta t \left(\sum_{n=0}^{N-1} \|\delta \rho^{n+\frac{1}{2}}\|_0 + \sum_{n=0}^{N-1} \|g^n\|_0 \right) \quad (8.77)$$

Como estamos estimando na norma $L^2(0, 1)$, então devemos tomar $m = 0$ em (8.8), obtendo-se

$$\|\xi^0\|_0 \leq c_1 h^{(k+1)} \|u_0\|_{k+1} \quad (8.78)$$

$$\begin{aligned} \Delta t \sum_{n=0}^{N-1} \|\delta \rho^{n+\frac{1}{2}}\|_0 &\leq \int_{t_0}^{t_N} \|\rho'(s)\|_0 ds \leq c_2 h^{(k+1)} \int_{t_0}^{t_N} \|u'(s)\|_{k+1} ds \\ &= c_2 h^{(k+1)} \|u'\|_{L^1(0,T;H^{k+1})} \end{aligned} \quad (8.79)$$

Temos também que,

$$\begin{aligned} g^n &= \left(\delta u^{n+\frac{1}{2}} - \frac{1}{2}(u_t^{n+1} + u_t^n) \right) = \frac{1}{\Delta t}(u^{n+1} - u^n) - \frac{1}{2}(u_t^{n+1} + u_t^n) \\ &= -\frac{1}{2\Delta t} \left\{ \int_{t_n}^{t_{n+1}} (s - t_n)(t_{n+1} - s) u'''(s) ds \right\} \end{aligned} \quad (8.80)$$

Para obter a última igualdade, basta integrar por partes duas vezes.

Como $s \in [t_n, t_{n+1}]$ então $|s - t_n| \leq \Delta t$ e $|s - t_{n+1}| \leq \Delta t$. Além disso, por hipótese $u \in L^\infty(0, T; L^2(0, 1))$, assim obtemos

$$\begin{aligned} \|g^n\|_0 &\leq \|u'''\|_{L^\infty(t_n, t_{n+1}; L^2(0,1))} \frac{1}{2\Delta t} \left\{ \int_{t_n}^{t_{n+1}} (s - t_n)(t_{n+1} - s) ds \right\} \\ &\leq c(\Delta t)^2 \|u'''\|_{L^\infty(t_n, t_{n+1}; L^2(0,1))} \end{aligned} \quad (8.81)$$

Fazendo a soma de $n = 0$ a $n = N - 1$ e multiplicando por Δt , obtemos

$$\Delta t \sum_{n=0}^{N-1} \|g^n\|_0 \leq c_1 (\Delta t)^2 \|u'''\|_{L^\infty(0,T;L^2(0,1))} \quad (8.82)$$

Substituindo (8.78), (8.79) e (8.82) em (8.77) obtemos

$$\|\xi^N\|_0 \leq C \left(h^{k+1} \|u_0\|_{k+1} + h^{k+1} \|u'\|_{L^1(0,T;H^{k+1})} + (\Delta t)^2 \|u'''\|_{L^\infty(0,T;L^2(0,1))} \right) \quad (8.83)$$

Como por hipótese $u_0 \in H^{k+1}$, $u' \in L^1(0, T, H^{k+1}(0, 1))$ e $u''' \in L^\infty(0, T, L^2(0, 1))$ então existe um constante $\widehat{C} > 0$ 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

$$\|\xi^N\|_0 \leq \mathcal{O}(h^{k+1} + (\Delta t)^2). \quad (8.84)$$

Por outro lado, tomando $m = 0$ em (8.8), obtemos

$$\|\rho^N\|_0 = \|u(t) - \tilde{u}(t)\|_0 \leq ch^{k+1} \|u(t)\|_{k+1}, \quad \text{para cada } t \text{ fixo em } [0, T],$$

Logo da decomposição do erro temos

$$\|e^N\|_0 = \|u^N - \tilde{u}^N\|_0 + \|\tilde{u}^N - u_h^N\|_0 = \|\rho^N\|_0 + \|\xi^N\|_0 = \mathcal{O}(h^{k+1} + (\Delta t)^2).$$

Tomando o máximo em $t \in [0, T]$,

$$\|e\|_{L^\infty(0,T;L^2(0,1))} = \|u - u_h\|_{L^\infty(0,T;L^2(0,1))} \leq C(h^{k+1} + (\Delta t)^2)$$

A demonstração da estimativa de erro para o espaço $L^\infty(0, T; H_0^1(0, 1))$ pode ser vista no próximo teorema, quando em particular, $(\theta = 1/2)$.

Podemos constatar um perda de precisão no espaço, obtendo a seguinte estimativa:

$$\|e\|_{L^\infty(0,T;H_0^1(0,1))} = \|u - u_h\|_{L^\infty(0,T;H_0^1(0,1))} \leq C(h^k + (\Delta t)^2)$$

Família θ -Métodos

Nesta seção faremos uma estimativa de erro para o método generalizado trapezoidal ou θ -métodos, descrito em (6.18). Como mencionado anteriormente, essa família de métodos engloba os métodos de Euler Progressivo ($\theta = 0$), Euler Regressivo ($\theta = 1$) e Crank-Nicolson ($\theta = 1/2$).

Teorema 8.6. *Sob a hipótese (8.47), e os dados iniciais $u_0 \in H_0^1(0, 1) \cap H^{k+1}(0, 1)$ então a solução numérica obtida pelo Método Generalizado Trapezoidal (6.18), para $\theta \in [1/2, 1]$, tem a seguinte estimativa de erro:*

$$\begin{aligned} \|e\|_{L^\infty(0,T;L^2(0,1))} &= \|u - u_h\|_{L^\infty(0,T;L^2(0,1))} \leq C(h^{k+1} + (\Delta t)^m) \\ \|e\|_{L^\infty(0,T;H_0^1(0,1))} &= \|u - u_h\|_{L^\infty(0,T;H_0^1(0,1))} \leq C(h^k + (\Delta t)^m), \end{aligned} \quad (8.85)$$

onde C é uma constante positiva e independentes de h e Δt . Além disso, $m = 2$ se $\theta = \frac{1}{2}$ e $m = 1$ se $\theta \neq \frac{1}{2}$.

Demonstração: Provaremos inicialmente a estimativa em $L^\infty(0, T; L^2(0, 1))$.

Fazendo $t = (n + 1)$ e $t = n$ com pesos θ e $(1 - \theta)$ respectivamente em (8.3) e somando os termos, obtemos

$$(u_t^{n+\theta}, v) + a(u^{n+\theta}, v) = (f^{n+\theta}, v), \quad \forall v \in H_0^1(0, 1) \quad (8.86)$$

onde estamos denotando $u^{n+\theta} = \theta u^{n+1} + (1 - \theta)u^n$, $\forall \theta \in [0, 1]$.

Somando e subtraindo $\delta u^{n+\frac{1}{2}}$ no primeiro membro de (8.86), obtemos

$$(\delta u^{n+\frac{1}{2}}, v) + a(u^{n+\theta}, v) = (f^{n+\theta} + g^n, v), \quad \forall v \in H_0^1(0, 1) \cap H^{k+1}(0, 1), \quad (8.87)$$

onde $g^n = \delta u^{n+\frac{1}{2}} - u_t^{n+\theta}$.

Considere agora o sistema aproximado em V_m^k , dado pela seguinte discretização

$$(\delta u_h^{n+\frac{1}{2}}, v_h) + a(u_h^{n+\theta}, v_h) = (f^{n+\theta}, v_h), \quad \forall v_h \in V_m^k. \quad (8.88)$$

Tomando em particular $v = v_h \in V_m^k \subset H_0^1(0, 1) \cap H^{k+1}(0, 1)$ em (8.87) e subtraindo de (8.88), obtemos

$$(\delta u^{n+\frac{1}{2}} - \delta u_h^{n+\frac{1}{2}}, v_h) + a(u^{n+\theta} - u_h^{n+\theta}, v_h) = (g^n, v_h), \quad \forall v_h \in V_m^k. \quad (8.89)$$

Somando e subtraindo os termos interpolantes, $\delta \tilde{u}^{n+\frac{1}{2}}$ e $\tilde{u}^{n+\theta} \in V_m^k$ no primeiro e segundo termo, respectivamente e usando a decomposição do erro (8.51), obtém-se

$$(\delta \xi^{n+\frac{1}{2}}, v_h)_0 + (\delta \rho^{n+\frac{1}{2}}, v_h)_0 + a(\xi^{n+\theta}, v_h) + a(\rho^{n+\theta}, v_h) = (g^n, v_h)_0, \quad \forall v_h \in V_m^k \quad (8.90)$$

Como pela projeção elíptica $a(\rho^{n+\theta}, v_h) = a(\theta \rho^{n+1} + (1 - \theta) \rho^n, v_h) = 0$, então

$$(\delta \xi^{n+\frac{1}{2}}, v_h)_0 + a(\xi^{n+\theta}, v_h) = (g^n, v_h)_0 - (\delta \rho^{n+\frac{1}{2}}, v_h)_0, \quad \forall v_h \in V_m^k \quad (8.91)$$

Tomando em particular $v_h = \xi^{n+\theta}$ em (8.91), obtemos

$$(\delta \xi^{n+\frac{1}{2}}, \xi^{n+\theta}) + a(\xi^{n+\theta}, \xi^{n+\theta})_0 = (g^n, \xi^{n+\theta})_0 - (\delta \rho^{n+\frac{1}{2}}, \xi^{n+\theta})_0, \quad \forall v_h \in V_m^k \quad (8.92)$$

Para o primeiro termo de (8.108) temos,

$$\begin{aligned} (\delta \xi^{n+\frac{1}{2}}, \xi^{n+\theta})_0 &= \frac{1}{\Delta t} (\xi^{n+1} - \xi^n, \theta \xi^{n+1} + (1 - \theta) \xi^n) \\ &= \frac{1}{\Delta t} \left(\theta \|\xi^{n+1}\|_0^2 + (1 - 2\theta)(\xi^{n+1}, \xi^n) - (1 - \theta) \|\xi^n\|_0^2 \right). \end{aligned}$$

Além disso, para o segundo termo do lado esquerdo temos que

$$a(\xi^{n+\theta}, \xi^{n+\theta}) \geq 0$$

Substituindo em (8.108) e multiplicando a igualdade por Δt , obtemos

$$\begin{aligned} &\theta \|\xi^{n+1}\|_0^2 + (1 - 2\theta)(\xi^{n+1}, \xi^n) - (1 - \theta) \|\xi^n\|_0^2 \\ &\leq \Delta t \left(g^n - \delta \rho^{n+\frac{1}{2}}, \xi^{n+\theta} \right) \leq \Delta t \left(\|g^n\|_0 + \|\delta \rho^{n+\frac{1}{2}}\|_0 \right) \|\xi^{n+\theta}\|_0 \end{aligned} \quad (8.93)$$

Note que, por hipótese $\theta \in [1/2, 1]$ então o termo $(1 - 2\theta) \leq 0$. Assim podemos escrever a seguinte desigualdade:

$$(1 - 2\theta)(\xi^{n+1}, \xi^n) \geq (1 - 2\theta)\|\xi^{n+1}\|_0 \|\xi^n\|_0.$$

Além disso, temos que

$$\|\xi^{n+\theta}\|_0 \leq \theta\|\xi^{n+1}\|_0 + (1 - \theta)\|\xi^n\|_0$$

Substituindo as duas últimas relações em (8.93) obtemos

$$\left(\|\xi^{n+1}\|_0 - \|\xi^n\|_0\right) \left[\theta\|\xi^{n+1}\|_0 + (1 - \theta)\|\xi^n\|_0\right] \leq \Delta t \left(\|g^n\|_0 + \|\delta\rho^{n+\frac{1}{2}}\|_0\right) \left[\theta\|\xi^{n+1}\|_0 + (1 - \theta)\|\xi^n\|_0\right]$$

Cancelando os termos comuns, e fazendo a soma para $n = 0, 1 \dots N - 1$, obtemos

$$\|\xi^N\|_0 \leq \|\xi^0\|_0 + \Delta t \left(\sum_{n=0}^{N-1} \|\delta\rho^{n+\frac{1}{2}}\|_0 + \sum_{n=0}^{N-1} \|g^n\|_0 \right) \quad (8.94)$$

Note que

$$\sum_{n=0}^{N-1} \left(\|\xi^{n+1}\|_0 - \|\xi^n\|_0 \right) = \|\xi^N\|_0 - \|\xi^0\|_0$$

Fazendo $m = 0$ em (8.8) obtém-se

$$\|\xi^0\|_0 \leq c_1 h^{k+1} \|u_0\|_{k+1} \quad (8.95)$$

Vamos agora analisar os dois termos restantes do lado direito de (8.102). Por definição

$$\delta\rho^{n+\frac{1}{2}} = \frac{1}{\Delta t} (\rho^{n+1} - \rho^n) = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \rho'(s) ds$$

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 $n = 0, 1 \dots N - 1$ e multiplicando por Δt , temos

$$\begin{aligned} \Delta t \sum_{n=0}^{N-1} \|\delta\rho^{n+\frac{1}{2}}\|_0 &\leq \sum_{n=0}^{N-1} \int_{t_n}^{t_{n+1}} \|\rho'(s)\|_0 ds = \int_0^{t_N} \|\rho'(s)\|_0 ds \\ &\leq h^{k+1} \int_0^{t_N} \|u'(s)\|_{k+1} ds = h^{k+1} \|u'\|_{L^1(0, T; H^{k+1})} \end{aligned} \quad (8.96)$$

onde temos usado (8.8) com $m = 0$, na última desigualdade.

Para o segundo termo de (8.102), observe que a função g^n definida em (8.87) pode ser decomposta na forma

$$g^n = \delta u^{n+\frac{1}{2}} - u_t^{n+\theta} = \left(\delta u^{n+\frac{1}{2}} - u_t^{n+1/2} \right) - \left((\theta - \frac{1}{2})(u_t^{n+1} - u_t^n) \right) = g_1^n + g_2^n \quad (8.97)$$

$$\text{onde } g_1^n = \frac{1}{\Delta t}(u^{n+1} - u^n) - \frac{1}{2}(u_t^{n+1} + u_t^n) \quad \text{e} \quad g_2^n = -(\theta - \frac{1}{2})(u_t^{n+1} - u_t^n).$$

Estimativa para os termos g_1^n e g_2^n .

Procedendo da mesma forma que em (8.80) temos que:

$$\begin{aligned} g_1^n &= \left(\delta u^{n+\frac{1}{2}} - \frac{1}{2}(u_t^{n+1} + u_t^n) \right) = \frac{1}{\Delta t}(u^{n+1} - u^n) - \frac{1}{2}(u_t^{n+1} + u_t^n) \\ &= -\frac{1}{2\Delta t} \left\{ \int_{t_n}^{t_{n+1}} (s - t_n)(t_{n+1} - s) u'''(s) ds \right\}, \end{aligned} \quad (8.98)$$

onde a última igualdade é obtida integrando por partes duas vezes. Usando (8.82), obtemos que

$$\Delta t \sum_{n=0}^{N-1} \|g_1^n\|_0 \leq c_1 (\Delta t)^2 \|u'''\|_{L^\infty(0,T;L^2(0,1))}. \quad (8.99)$$

Por outro lado

$$g_2^n = -(\theta - \frac{1}{2})(u_t^{n+1} - u_t^n) = -(\theta - \frac{1}{2}) \int_{t_n}^{t_{n+1}} u''(s) ds$$

Logo

$$\|g_2^n\|_0 \leq \Delta t \left| \theta - \frac{1}{2} \right| \|u''\|_{L^\infty(t_n, t_{n+1}; L^2(0,1))}$$

Somando de $n = 1, \dots, N-1$ e multiplicando por Δt , temos

$$\Delta t \sum_{n=1}^{N-1} \|g_2^n\|_0 \leq \Delta t \left| \theta - \frac{1}{2} \right| \|u''\|_{L^\infty(0,T;L^2(0,1))} \quad (8.100)$$

Substituindo (8.99) e 8.100) em (8.97) obtemos

$$\begin{aligned} \Delta t \sum_{n=1}^{N-1} \|g^n\|_0 &\leq \Delta t \sum_{n=1}^{N-1} \|g_1^n\|_0 + \Delta t \sum_{n=1}^{N-1} \|g_2^n\|_0 \\ &\leq c_1 (\Delta t)^2 \|u'''\|_{L^\infty(0,T;L^2(0,1))} + \Delta t \left| \theta - \frac{1}{2} \right| \|u''\|_{L^\infty(0,T;L^2(0,1))} \end{aligned} \quad (8.101)$$

Substituindo (8.95), (8.101) e (8.96) em (8.102) obtemos que

$$\begin{aligned} \|\xi^N\|_0 &\leq Ch^{k+1} \left(\|u_0\|_{k+1} + \|u'\|_{L^1(0,T;H^{k+1})} \right) \\ &\quad + c_1 (\Delta t)^2 \|u'''\|_{L^\infty(0,T;L^2(0,1))} + \Delta t \left| \theta - \frac{1}{2} \right| \|u''\|_{L^\infty(0,T;L^2(0,1))} \end{aligned} \quad (8.102)$$

Tomando o máximo para $t \in [0, T]$ em ambos os lados e como os termos do lado direito são independentes de N , então existe uma constante C tal que

$$\|\xi\|_{L^\infty(0,T;L^2(0,1))} \leq C \left(h^{k+1} + \left| \theta - \frac{1}{2} \right| \Delta t + (\Delta t)^2 \right). \quad (8.103)$$

Portanto para $\theta = \frac{1}{2}$, '**Método de Crank-Nicolson**', temos que a ordem de convergência é $\mathcal{O}(h^{k+1}, (\Delta t)^2)$ e para $\theta \neq \frac{1}{2}$ a ordem de convergência é $\mathcal{O}(h^{k+1}, \Delta t)$.

Por outro lado, fazendo $m = 0$ em (8.8), obtemos

$$\|\rho^N\|_0 = \|u(t) - \tilde{u}(t)\|_0 \leq ch^{k+1} \|u(t)\|_{k+1}, \quad \text{para cada } t \text{ fixo em } [0, T],$$

Logo

$$\|\rho\|_{L^\infty(0,T;L^2(0,1))} \leq C_1 h^{k+1} \|u\|_{L^\infty(0,T;H^{k+1})} \quad (8.104)$$

Da decomposição do erro e das estimativas (8.103), (8.104), concluímos que:

Para $\theta \neq \frac{1}{2}$,

$$\|e\|_{L^\infty(0,T;L^2(0,1))} \leq \|\rho\|_{L^\infty(0,T;L^2(0,1))} + \|\xi\|_{L^\infty(0,T;L^2(0,1))} \leq C(h^{k+1} + \Delta t). \quad (8.105)$$

Para $\theta = \frac{1}{2}$,

$$\|e\|_{L^\infty(0,T;L^2(0,1))} \leq \|\rho\|_{L^\infty(0,T;L^2(0,1))} + \|\xi\|_{L^\infty(0,T;L^2(0,1))} \leq C(h^{k+1} + (\Delta t)^2), \quad (8.106)$$

o que conclui a primeira parte do teorema.

Estimativa para $L^\infty(0, T; H_0^1(0, 1))$

Vamos estimar o erro entre a solução exata e aproximada na norma $L^\infty(0, T; H_0^1(0, 1))$, completando a demonstração do teorema.

Considere a igualdade dada por (8.91)

$$(\delta\xi^{n+\frac{1}{2}}, v_h)_0 + a(\xi^{n+\theta}, v_h) = (g^n, v_h)_0 - (\delta\rho^{n+\frac{1}{2}}, v_h)_0, \quad \forall v_h \in V_m^k \quad (8.107)$$

Tomando em particular $v_h = \delta\xi^{n+\frac{1}{2}}$ em (8.107), obtemos

$$\|\delta\xi^{n+\frac{1}{2}}\|_0^2 + a(\xi^{n+\theta}, \delta\xi^{n+\frac{1}{2}}) = (g^n, \delta\xi^{n+\frac{1}{2}})_0 - (\delta\rho^{n+\frac{1}{2}}, \delta\xi^{n+\frac{1}{2}})_0, \quad (8.108)$$

Para o primeiro termo de (8.108) temos,

$$\begin{aligned} a(\xi^{n+\theta}, \delta\xi^{n+\frac{1}{2}}) &= \frac{1}{\Delta t} a(\xi^{n+1} - \xi^n, \theta\xi^{n+1} + (1-\theta)\xi^n) \\ &= \frac{1}{\Delta t} \left(\theta \|\xi^{n+1}\|_a^2 + (1-2\theta)(\xi^{n+1}, \xi^n)_a - (1-\theta)\|\xi^n\|_a^2 \right). \end{aligned} \quad (8.109)$$

Por hipótese $\theta \in [1/2, 1]$ então o termo $(1-2\theta) \leq 0$. Assim podemos escrever a seguinte desigualdade:

$$(1-2\theta)(\xi^{n+1}, \xi^n)_a \geq (1-2\theta)\|\xi^{n+1}\|_a \|\xi^n\|_a \geq \frac{(1-2\theta)}{2} \left(\|\xi^{n+1}\|_a^2 + \|\xi^n\|_a^2 \right).$$

Substituindo em (8.109), obtemos

$$a(\xi^{n+\theta}, \delta\xi^{n+\frac{1}{2}}) \geq \frac{1}{2\Delta t} \left(\|\xi^{n+1}\|_a^2 - \|\xi^n\|_a^2 \right)$$

Por outro usando a desigualdade de Schwarz e elementar para os dois termos do lado direito de (8.108), obtemos

$$\begin{aligned} (g^n, \delta\xi^{n+\frac{1}{2}})_0 &\leq \frac{1}{2} \left(\|g^n\|_0^2 + \|\delta\xi^{n+\frac{1}{2}}\|_0^2 \right), \\ (\delta\rho^{n+\frac{1}{2}}, \delta\xi^{n+\frac{1}{2}})_0 &\leq \frac{1}{2} \left(\|\delta\rho^{n+\frac{1}{2}}\|_0^2 + \|\delta\xi^{n+\frac{1}{2}}\|_0^2 \right). \end{aligned}$$

Substituindo em (8.108) e multiplicando por 2 obtemos,

$$\left(\|\xi^{n+1}\|_a^2 - \|\xi^n\|_a^2 \right) \leq \left(\|g^n\|_0^2 + \|\delta\rho^{n+\frac{1}{2}}\|_0^2 \right). \quad (8.110)$$

Multiplicando por Δt , fazendo a soma para $n = 0, 1, \dots, N-1$ então

$$\sum_{n=0}^{N-1} \left(\|\xi^{n+1}\|_a^2 - \|\xi^n\|_a^2 \right) \leq \Delta t \sum_{n=0}^{N-1} \left(\|g^n\|_0^2 + \|\delta\rho^{n+\frac{1}{2}}\|_0^2 \right)$$

Ou seja

$$\|\xi^N\|_a^2 \leq \|\xi^0\|_a^2 + \Delta t \sum_{n=0}^{N-1} \left(\|g^n\|_0^2 + \|\delta\rho^{n+\frac{1}{2}}\|_0^2 \right) \quad (8.111)$$

Devido a equivalência das normas $\|\cdot\|_a$ e $\|\cdot\|_1$, então podemos usar a desigualdade (8.8) como $m = 1$ para obter

$$\|\xi^0\|_1 \leq ch^k \|u_0\|_{k+1}$$

Para os dois termos restantes, sabemos de (8.96) e de (8.101) que

$$\Delta t \sum_{n=0}^{N-1} \|\delta\rho^{n+\frac{1}{2}}\|_0 \leq h^{k+1} \|u'\|_{L^1(0,T;H^{k+1})}$$

$$\Delta t \sum_{n=1}^{N-1} \|g^n\|_0 \leq \frac{(\Delta t)^2}{2} \|u'''\|_{L^\infty(0,T;L^2(0,1))} + \Delta t \left| \theta - \frac{1}{2} \right| \|u''\|_{L^\infty(0,T;L^2(0,1))}$$

Substituindo os três termos em (8.111) e observando a equivalência de normas em espaços de dimensão finita, podemos extrair os quadrados das normas e obter

$$\begin{aligned} \|\xi^N\|_a^2 \leq ch^k \|u_0\|_{k+1}^2 &+ h^{k+1} \|u'\|_{L^1(0,T;H^{k+1})}^2 + \frac{(\Delta t)^2}{2} \|u'''\|_{L^\infty(0,T;L^2(0,1))}^2 \\ &+ \Delta t \left| \theta - \frac{1}{2} \right| \|u''\|_{L^\infty(0,T;L^2(0,1))}^2 \end{aligned} \quad (8.112)$$

Para a conclusão, basta observar a decomposição do erro. Tomando $m = 1$ em (8.8), obtemos

$$\|\rho^N\|_1 = \|u(t) - \tilde{u}(t)\|_1 \leq ch^k \|u(t)\|_{k+1}, \quad \text{para cada } t \text{ fixo em } [0, T],$$

Tomando o máximo $t \in [0, T]$ na última desigualdade e em (8.112) obtemos a estimativa:

Para $\theta \neq \frac{1}{2}$,

$$\|e\|_{L^\infty(0,T;H_0^1(0,1))} \leq \|\rho\|_{L^\infty(0,T;H_0^1(0,1))} + \|\xi\|_{L^\infty(0,T;H_0^1(0,1))} \leq C(h^k + \Delta t). \quad (8.113)$$

Para $\theta = \frac{1}{2}$,

$$\|e\|_{L^\infty(0,T;H_0^1(0,1))} \leq \|\rho\|_{L^\infty(0,T;H_0^1(0,1))} + \|\xi\|_{L^\infty(0,T;H_0^1(0,1))} \leq C\left(h^k + (\Delta t)^2\right), \quad (8.114)$$

o que conclui o teorema \square

Observação

1 - Para concluir as estimativas para o θ -Métodos, faremos algumas considerações, sem demonstrar, sobre o método quando $\theta \in [0, 1/2)$. No método das diferenças finitas, ver por exemplo em ([19]), que nesse intervalo temos a convergência condicionada à $r = \Delta t \leq \frac{h^2}{2(1-2\theta)}$. De forma análoga, no método de elementos finitos temos uma restrição similar, ou seja $\frac{\Delta t}{h^2} \leq f(\theta)$. As estimativas de erro nesse intervalo, são as mesmas de (8.85). Note também que, o Método de Euler progressivo ($\theta = 0$), como já mencionado, é condicionalmente estável.

2 - Note que as estimativas dependem do grau do polinômio interpolador, além da regularidade da solução, como visto no Teorema(3.5). Do citado teorema, sabemos que o erro de interpolação (3.51) é dado por:

$$\|u - \tilde{u}\|_s \leq ch^\alpha \|u\|_m,$$

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, $k = 1$ então o erro na norma L^2 e H_0^1 será de ordem $\mathcal{O}(h^2)$ e $\mathcal{O}(h)$ respectivamente, para toda solução regular $m \geq 2$. Por outro lado, se a solução $u \in H_0^1 \cap H^2$ então a ordem de convergência tem a mesma ordem anterior, independente do grau do polinômio interpolador definido em V_m^k .

8.1.3 Exercício

1. Mostre que o método de Euler progressivo (6.16) 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 (8.51), somando e subtraindo os termos $\delta \tilde{u}^{n+\frac{1}{2}}$, obtém-se que

$$(\delta \xi^{n+\frac{1}{2}}, v_h) + a(\xi^n, v_h) = (\delta u^{n+\frac{1}{2}} - \frac{d}{dt} u^n, v_h) - (\delta \rho^{n+\frac{1}{2}}, v_h), \quad \forall v_h \in V_m^k. \quad (8.115)$$

Tomando em particular $v_h = \xi^n$ em (8.115), obtenha estimativas semelhantes ao obtida em (8.59) e finalmente (8.67).

8.2 Existência e unicidade de solução

É essencial saber da existência e unicidade de solução e suas propriedades para então determinar a solução aproximada usando um método numérico. No entanto, não é imprescindível saber provar esses resultados para se obter a solução numérica aproximada. No que se segue, ver Medeiros ([13]), demonstraremos a existência e unicidade de solução do problema variacional (6.1), usando o método de Galerkin, ou também conhecido como método de Faedo-Galerkin.

Teorema 8.7. *Seja $f \in L^2(0, T; L^2(0, 1))$ e $u_0 \in H_0^1(0, 1)$, então existe uma única solução $u : [0, T] \times (0, 1) \rightarrow \mathbb{R}$ para o problema (6.1) satisfazendo a seguinte condição:*

- (i) $u \in L^2(0, T; H_0^1(0, 1)) \cap C^0(0, T; L^2(0, 1))$
- (ii) $\frac{d}{dt}(u(t), v) + a(u(t), v) = (f, v), \quad \forall v \in H_0^1(0, 1) \text{ em } L^2(0, T).$
- (iii) $u(0) = u_0.$

Demonstração:

Para provar o teorema, é introduzido um problema aproximado ao problema (6.1), definido num subespaço de dimensão finita, que denotaremos por V_m . Determina-se uma sucessão de soluções aproximadas $u_m \in V_m$ e a dificuldade principal é mostrar que essa sucessão converge para a solução u do problema original quando $m \rightarrow \infty$. Esse método foi primeiro empregado por Faedo-Galerkin. Seja $T > 0$ e $V_m = [w_1, w_2, \dots, w_m]$, um subespaço vetorial gerado, onde $\{w_i, i = 1, \dots, m\}$ são os m -primeiros elementos da base do espaço de Hilbert $H_0^1(0, 1)$ (A existência da base é uma consequência do espaço de Hilbert $H_0^1(0, 1)$ ser um espaço separável). Se $u_m(t) \in V_m$ então ele pode ser representado por

$$u_m(t) = \sum_{i=1}^m g_{im}(t) w_i(x). \quad (8.116)$$

Considere o problema (6.1), redefinido no subespaço V_m . Queremos determinar uma solução $u_m : [0, T] \rightarrow V_m$, solução do seguinte sistema,

$$\begin{cases} (u'_m(t), v) + a(u_m(t), v) = (f(t), v), \quad \forall v \in V_m, \\ u_m(0) = u_{0m} \longrightarrow u_0 \quad \text{em } H_0^1(0, 1). \end{cases} \quad (8.117)$$

A condição inicial (8.117)₂ significa que:

$$u_{0m} = \sum_{i=1}^m (u_0, w_i) w_i \longrightarrow u_0 = \sum_{i=1}^{\infty} (u_0, w_i) w_i, \quad \text{em } H_0^1(0, 1), \quad \text{quando } m \rightarrow \infty,$$

onde $g_{im}(0) = (u_0, w_i)$. Assim é natural a escolha de u_{0m} como condição inicial do problema em V_m . Observe que no problema aproximado a variável tempo ainda é contínua e que os produtos internos (\cdot, \cdot) são definidos somente na variável espacial x . A idéia da demonstração consiste em mostrar que existe uma sequência de soluções $u_m(x, t)$ do problema aproximado (8.117) pertencentes ao subespaço V_m que são limitadas, independente de m , possibilitando mostrar que essa sequência converge para solução exata $u(x, t)$, quando $m \rightarrow \infty$, significando que cada vez mais o subespaço V_m se aproxime do espaço $H_0^1(0, 1)$.

Substituindo (8.116) em (8.117), obtemos

$$\left(\sum_{i=1}^m g'_{im}(t) w_i(x), v \right) + a \left(\sum_{i=1}^m g_{im}(t) w_i(x), v \right) = (f, v), \quad \forall v \in V_m$$

Como (\cdot, \cdot) são definidos em V_m , podemos escrever

$$\sum_{i=1}^m g'_{im}(t) (w_i(x), v) + \sum_{i=1}^m g_{im}(t) a(w_i(x), v) = (f, v), \quad \forall v \in V_m,$$

Como a igualdade é válida para todo $v \in V_m$, podemos tomar em particular $v = w_j$, resultando em

$$\sum_{i=1}^m g'_{im}(t) (w_i(x), w_j(x)) + \sum_{i=1}^m g_{im}(t) a(w_i(x), w_j(x)) = (f(t), w_j(x))$$

Definindo as matrizes

$$A = a_{ij} = (w_i(x), w_j(x)), \quad B = b_{ij} = a(w_i(x), w_j(x)) \quad \text{e} \quad F = F_j = (f(t), w_j(x)),$$

obtemos o seguinte sistema de equações diferenciais ordinárias

$$\sum_{i=1}^m g'_{im}(t) a_{ij} + \sum_{i=1}^m g_{im}(t) b_{ij} = F_j, \quad \text{para } j = 1, \dots, m$$

que pode ser escrito na seguinte forma matricial:

$$\begin{cases} Ag'_m(t) + Bg_m(t) = F(t), & \forall t \in [0, T] \\ g_m(0) = ((u_0, w_1), (u_0, w_2), \dots, (u_0, w_m)) = g_{0m} \end{cases} \quad (8.118)$$

onde $g_m(0)$ é a condição inicial, $g_m(t) = (g_{1m}(t), g_{2m}(t), \dots, g_{mm}(t))^T$ é um vetor incógnita. Assim temos um sistema de m equações diferenciais ordinárias.

Para cada m o sistema (8.118) tem uma única solução $g_m(t)$ e portanto obtemos a solução $u_m(t) = \sum_{i=1}^m g_{im}(t)w_i(x)$ do problema (8.117) em V_m . Note que a solução $u_m(t)$ é local, desde que, o tempo $t \in [0, T_m[$, $T_m > 0$. O próximo passo é demonstrar que as soluções são limitadas independente de m e t , que permitirão estender a solução em todo intervalo $[0, T]$ e obter a convergência da sequência u_m para u , solução do problema original no sentido do teorema. Para isso são necessários as seguintes estimativas:

Estimativa 1: Considere $v = u_m(t) \in V_m$ em (8.117). Então obtemos;

$$(u'_m(t), u_m(t)) + a(u_m(t), u_m(t)) = (f(t), u_m(t)). \quad (8.119)$$

Mas

$$\frac{d}{dt}|u_m(t)|^2 = 2(u'_m(t), u_m(t))$$

e

$$a(u_m(t), u_m(t)) = \alpha \|u_m(t)\|^2 + \beta |u_m(t)|^2;$$

onde estamos usando a equivalência entre as normas $H^1(0, 1)$ e $L^2(0, 1)$ no espaço $H_0^1(0, 1)$. Por outro lado, da desigualdade de Schwarz temos

$$\begin{aligned} (f(t), u_m(t)) &= \int_0^1 f(t)u_m(t)dx \leq \left(\int_0^1 |f(t)|^2 \right)^{1/2} \left(\int_0^1 |u_m(t)|^2 \right)^{1/2} \\ &\leq |f(t)|_{L^2(0,1)} |u_m(t)|_{L^2(0,1)} \leq \frac{1}{4\beta} |f(t)|^2 + \beta |u_m(t)|^2; \end{aligned}$$

na última desigualdade usamos a seguinte desigualdade elementar:

$$ab \leq \frac{a^2}{4\varepsilon} + \varepsilon b^2, \quad (8.120)$$

onde a e b são constantes reais, ε é uma constante positiva e considerando $\varepsilon = \beta$, $a = |f(t)|$ e $b = |u_m(t)|$. Substituindo os resultados em (8.119), obtemos

$$\frac{1}{2} \frac{d}{dt} |u_m(t)|^2 + \alpha \|u_m(t)\|^2 \leq \frac{1}{4\beta} |f(t)|^2.$$

Integrando a desigualdade em 0 a t , $t < T_m$, obtemos

$$\frac{1}{2} |u_m(t)|^2 + \alpha \int_0^t \|u_m(t)\|^2 \leq \frac{1}{2} |u_m(0)|^2 + \frac{1}{4\beta} \int_0^t |f(t)|^2.$$

Considere as constantes positivas $\{c_1, c_2, c_3, c_4\}$, definidas por:

$$c_1 = \min\{1/2, \alpha\}, \quad |u_m(0)|^2 \leq c_2 \|u_m(0)\|^2, \quad c_3 = \max\{c_2, 1/(4\beta)\} \text{ e } c_4 = c_3/c_1.$$

Então a desigualdade acima pode ser escrita na forma:

$$|u_m(t)|^2 + \int_0^t \|u_m(t)\|^2 \leq c_4 \left(\|u_m(0)\|^2 + \int_0^t |f(t)|^2 \right)$$

Mas por hipótese $u_m(0) = u_{0m}$ converge forte para $u_0 \in H_0^1(0, 1)$, u_{0m} é limitada em $H_0^1(0, 1)$, para m suficientemente grande. Temos também que $f \in L^2(0, T; L^2(0, 1))$ e portanto o segundo termo do lado direito é limitado. Dessa forma podemos escrever a desigualdade na forma:

$$|u_m(t)|^2 + \int_0^t \|u_m(t)\|^2 \leq c_5, \quad (8.121)$$

a constante c_5 é positiva e independe de t e m .

De (8.121) obtemos do primeiro e segundo termo respectivamente que

- (i) u_m é limitada em $L^\infty(0, T; L^2(0, 1))$
- (ii) u_m é limitada em $L^2(0, T; H_0^1(0, 1))$,

independente de m . Note que podemos estender a limitação ao intervalo $(0, T)$, pois a limitação independe do tempo t .

Estimativa 2: Considere $v = u'_m(t) \in V_m$ em (8.117). Então obtemos;

$$\left(u'_m(t), u'_m(t) \right) + a \left(u_m(t), u'_m(t) \right) = \left(f(t), u'_m(t) \right). \quad (8.122)$$

Mas

$$\begin{aligned} \left(u'_m(t), u'_m(t) \right) &= |u'_m(t)|^2, \\ a \left(u_m(t), u'_m(t) \right) &= \frac{1}{2\alpha} \frac{d}{dt} \|u_m(t)\|^2 + \frac{1}{2\beta} \frac{d}{dt} |u_m(t)|^2 = \frac{1}{2} \frac{d}{dt} a \left(u_m(t), u_m(t) \right). \end{aligned}$$

e usando a desigualdade de Schwarz, juntamente com a desigualdade elementar (8.120) com $\varepsilon = 1/2$, obtemos

$$\left(f(t), u'_m(t) \right) \leq \frac{1}{2} \left(|f(t)|^2 + |u'_m(t)|^2 \right)$$

Substituindo temos;

$$|u'_m(t)|^2 + \frac{d}{dt} a \left(u_m(t), u_m(t) \right) \leq |f(t)|^2$$

Integrando a desigualdade em 0 a t , $t < T_m$, obtemos

$$\int_0^t |u'_m(t)|^2 + a(u_m(t), u_m(t)) \leq a(u_m(0), u_m(0)) + \int_0^t |f(t)|^2$$

Note que

$$a(u_m(0), u_m(0)) = \alpha \|u_m(0)\|^2 + \beta |u_m(0)|^2 \leq c_6 \|u_m(0)\|^2,$$

onde $c_6 = \max\{\alpha, \hat{c}\beta\}$ sendo \hat{c} é a constante da desigualdade $|u_m(0)|^2 \leq \hat{c} \|u_m(0)\|^2$.

Por hipótese $u_m(0) = u_{0m}$ é limitada em $H_0^1(0, 1)$ e $f \in L^2(0, T; L^2(0, 1))$. Então o lado direito é limitada por uma constante, independente de m e t , permitindo concluir que:

(iii) u'_m é limitada em $L^2(0, T; L^2(0, 1))$

(iv) u_m é limitada em $L^\infty(0, T; L^2(0, 1) \cap H_0^1(0, 1)) = L^\infty(0, T; H_0^1(0, 1))$

O espaço $H_0^1(0, 1)$ tem imersão compacta em $L^2(0, 1)$ e das estimativas (ii) e (iii), segue do Teorema de Aubin-Lions, que a sequência $u_m \in C^0([0, T; L^2(0, 1)])$. Como esse espaço é um espaço métrico completo, então da sequência limitada u_m , podemos extrair uma subsequência u_ν , que converge forte para u quase sempre em $[0, T] \times (0, 1)$, ou seja,

$$\lim_{\nu \rightarrow \infty} \max_{0 \leq t \leq T} |u_\nu(t) - u(t)|_{L^2(0,1)} = 0$$

Note que se a subsequência u_ν converge então toda sequência u_m também converge. Desde que $u \in C^0([0, T; L^2(0, 1)])$ então faz sentido tomar $u(0)$ e além disso

$$u_0 = \sum_{i=1}^{\infty} (u_0, w_i) w_i = u(0)$$

Vimos que $u \in L^2(0, T; H_0^1(0, 1)) \cap C^0(0, T; L^2(0, 1))$ e que $u_0 = u(0)$.

Convergência do Sistema Aproximado

Para concluir o teorema, resta mostrar, que a sequência u_m converge para u em $L^2(0, T; H_0^1(0, 1))$.

De fato, como u_m é limitada em $L^2(0, T; H_0^1(0, 1))$, então podemos extrair uma subsequência de u_m também denotada por u_m , a qual converge para uma função $u \in L^2(0, T; H_0^1(0, 1))$. Isto significa que

$$\int_0^T ((u_m(t), v)) \theta(t) dt \rightarrow \int_0^T ((u(t), v)) \theta(t) dt, \quad \forall v \in H_0^1(0, 1) \text{ e } \forall \theta \in L^2(0, T).$$

Usando a forma bilinear temos a forma equivalente

$$\int_0^T a(u_m(t), v)\theta(t)dt \rightarrow \int_0^T a(u(t), v)\theta(t)dt, \quad \forall v \in H_0^1(0, 1) \text{ e } \forall \theta \in L^2(0, T) \quad (8.123)$$

Para o outro termo envolvendo a derivada no tempo temos da estimativa (iii), que u'_m é limitada em $L^2(0, T; L^2(0, 1))$, o que significa

$$\int_0^T \frac{d}{dt}(u_m(t), v)\theta(t) dt \rightarrow \int_0^T \frac{d}{dt}(u(t), v)\theta(t)dt, \quad (8.124)$$

para todo $v \in H_0^1(0, 1)$ e $\theta \in L^2(0, T)$.

De (8.123) e (8.124), fazendo $m \rightarrow \infty$ tem-se que

$$\int_0^T \left(\frac{d}{dt}(u_m(t), v)\theta(t) + a(u_m(t), v)\theta(t) \right) dt \rightarrow \int_0^T \left(\frac{d}{dt}(u(t), v)\theta(t) + a(u(t), v)\theta(t) \right) dt$$

para todo $v \in H_0^1(0, 1)$ e para todo $\theta \in L^2(0, T)$. Do Lema Du Bois Raymond, ver Medeiros ([13]) obtemos que

$$\frac{d}{dt}(u_m(t), v) + a(u_m(t), v) \rightarrow \frac{d}{dt}(u(t), v) + a(u(t), v), \quad \forall v \in H_0^1(0, 1)$$

em $L^2(0, T)$. Como o subespaço V_m é denso em $H_0^1(0, 1)$, concluí-se que o sistema aproximado (8.117) converge quando $m \rightarrow \infty$ para o sistema (6.1), ou seja

$$\frac{d}{dt}(u_m(t), v) + a(u_m(t), v) - (f, v) \rightarrow \frac{d}{dt}(u(t), v) + a(u(t), v) - (f, v), \quad \forall v \in H_0^1(0, 1)$$

em $L^2(0, T)$.

Unicidade de Solução

Para concluir a demonstração do teorema resta provar a unicidade de solução, que nesse caso é bem simples, dado que o problema (6.1) é linear. Suponha que existam duas soluções distintas $u_1(x, t)$ e $u_2(x, t)$ e considere $w(x, t) = u_1(x, t) - u_2(x, t)$. Substituindo no problema (6.1), obtemos

$$\begin{cases} w_t(x, t) - \alpha w_{xx}(x, t) + \beta w(x, t) = 0, & \forall (x, t) \in (0, 1) \times [0, T] \\ w(0, t) = w(1, t) = 0, & \forall t \in [0, T] \\ w(x, 0) = 0, & \forall x \in (0, 1). \end{cases} \quad (8.125)$$

Que é equivalente a forma,

$$\begin{cases} (w_t, v) + a(w, v) = 0, & \forall v \in H_0^1(0, 1) \\ w(0) = 0, & \forall v \in H_0^1(0, 1). \end{cases} \quad (8.126)$$

Tomando $v = w$ em (8.126) e integrando de 0 a t , obtemos

$$\frac{1}{2}|w(t)|^2 + \int_0^t \|w(s)\|^2 ds = 0$$

Todos os dois termos do lado esquerdo são positivos, assim $w = 0$ é a única solução possível para igualdade. Ou seja, $u_1(x, t) = u_2(x, t)$ \square

8.3 Propriedades e decaimento assintótico

Nessa seção, vamos mostrar alguns resultados importantes para equação parabólica homogênea, que pode auxiliar em simulações numéricas e o decaimento assintótico.

O princípio do máximo, garante que a solução u não pode ser maior e nem menor que os dados iniciais e de fronteira. O decaimento exponencial garante que a "energia" do sistema decai exponencialmente quando o tempo cresce indefinidamente.

Consideraremos $\Omega \subset \mathbb{R}^n$, um conjunto aberto e limitado com fronteira representada por $\partial\Omega$ e o cilindro $Q = (0, T) \times \Omega$ e denotemos o conjunto $C^\infty(Q)$, espaço das funções infinitamente continua e derivável nas variáveis x e t .

Teorema 8.8. (*Princípio do máximo*) *Seja u a solução do seguinte problema homogêneo;*

$$\begin{cases} u_t(x, t) - \Delta u(x, t) = 0, & \forall (x, t) \in Q \\ u(x, t) = 0, & \forall x \in \partial\Omega \text{ e } \forall t \in [0, T] \\ u(x, 0) = u_0(x) & \forall x \in \Omega \end{cases} \quad (8.127)$$

onde $u \in C^2(Q)$. Então a solução satisfaz a seguinte condição,

$$\min \left\{ \inf_{\partial\Omega} u, 0 \right\} \leq u(x, t) \leq \max \left\{ \sup_{\partial\Omega} u, 0 \right\}.$$

Demonstração:

Se um dos extremos da função u pertence a fronteira então o resultado é imediato.

Suponhamos que o ponto $(x_0, t_0) \in \Omega^\circ$ seja um extremo da função não pertence a fronteira e provaremos o resultado por absurdo.

De fato, considere a função $v = u + \epsilon|x|^2$. Então $v = v(x, t)$ satisfaz a seguinte igualdade:

$$v_t - \Delta v = -2\epsilon N < 0,$$

onde N é a dimensão espaço. Em particular seja $(x_0, t_0) \in \Omega^\circ$, um ponto de máximo da função. Então

$$(v_t, \nabla v) = (0, \mathbf{0}).$$

Como (x_0, t_0) é um ponto de máximo então a matriz Hessiana

$$\sum_{i=1}^N \sum_{j=1}^N \frac{\partial^2 v}{\partial x_i \partial x_j} \xi_i \xi_j \leq 0$$

é definida negativa e por definição cada termo da diagonal principal é negativo. Assim conclui-se que o laplaciano de v no ponto de máximo (x_0, t_0) é negativo, ou seja $\Delta v(x_0, t_0) \leq 0$. Logo

$$v_t(x_0, t_0) - \Delta v(x_0, t_0) \geq 0,$$

o que contradiz a hipótese. Portanto o máximo deve pertencer a fronteira e

$$v(x, t) \leq \left\{ \sup_{\Omega} u_0 + \epsilon |x|^2, \epsilon c \right\}$$

Fazendo $\epsilon \rightarrow 0$ obtemos

$$u(x, t) \leq \left\{ \sup_{\Omega} u_0, 0 \right\}.$$

De forma análoga, para mostrar o lado esquerdo da desigualdade, basta tomar o ponto (x_0, t_0) como o mínimo. \square

No próximo teorema vamos mostrar que a solução do problema (8.127) decai exponencialmente;

Teorema 8.9. (*Decaimento exponencial*) Considere o problema ,

$$\begin{cases} u_t(x, t) - \alpha \Delta u(x, t) + \beta u(x, t) = 0, & \forall (x, t) \in Q \\ u(x, t) = 0, & \forall x \in \partial\Omega \text{ e } \forall t \in [0, T] \\ u(x, 0) = u_0(x) & \forall x \in \Omega \end{cases} \quad (8.128)$$

onde $u \in C^2(Q)$, α e β são constantes positivas. Então a energia associada a solução satisfaz:

$$E(t) := \frac{1}{2} \int_{\Omega} |u(s, t)|^2 ds \leq E(0) \exp^{-kt} \quad (8.129)$$

onde k é uma constante positiva dependente de Ω .

Demonstração:

Multiplicando a equação (8.128) por u e integrando em Ω , obtemos;

$$\int_{\Omega} u_t u + \alpha \int_{\Omega} (\nabla u)^2 + \beta \int_{\Omega} u^2 = 0$$

Mas

$$\int_{\Omega} u_t u = (u_t, u) = \frac{1}{2} \frac{d}{dt} |u|^2$$

Logo

$$\frac{d}{dt} E(t) = - \left(\alpha \int_{\Omega} (\nabla u)^2 + \beta \int_{\Omega} u^2 \right)$$

Pela desigualdade de Poincaré, temos que

$$\|u\| \leq C(\Omega) |\nabla u|$$

Então obtemos que

$$\frac{d}{dt} E(t) \leq - \frac{(\alpha + \beta)}{C(\Omega)} \int_{\Omega} |u|^2 = - \frac{(\alpha + \beta)}{C(\Omega)} E(t) \leq 0$$

Note que o lado direito é negativo, pois os termos $E(t)$, α , β e $C(\Omega)$ são positivos. Assim a função energia $E(t)$ é decrescente. Resolvendo a equação diferencial ordinária na variável t , obtemos que

$$E(t) \leq E(0) \exp^{-t(\alpha + \beta)/C(\Omega)}$$

Note que $\lim_{t \rightarrow \infty} E(t) = 0$, isto é, a equação do calor tem decaimento assintótico.

Em particular, no caso unidimensional com $x \in (0, L)$, tem-se

$$E(t) \leq E(0) \exp^{-(\pi^2/L^2)t}$$

Decaimento Assintótico: Caso Discreto

Pelo Teorema(8.9), a energia associada a solução do problema (8.128) satisfaz

$$E(t) := \frac{1}{2} \int_{\Omega} |u(s, t)|^2 ds \leq E(0) \exp^{-kt} \quad (8.130)$$

onde k é uma constante positiva dependente de Ω . Como em geral, a solução exata do problema não é conhecida, então não temos como calcular a integral acima, e devemos usar algum método numérico para calcular a integral, pois somente a solução numérica é conhecida. Para isso vamos calcular a integral e portanto a energia usando o método dos trapézios por sua simplicidade. Para isso considere o tempo discreto $t_n = n\Delta t$, para $n = 0, 1, \dots, N$ e $x_{i+1} = x_i + h$, $i = 1, \dots, m$. Por simplicidade vamos considerar o caso unidimensional no espaço e $x \in [0, 1]$. Logo a energia no tempo discreto é dada por

$$E(t_n) = \frac{1}{2} \int_{\Omega} |u(s, t_n)|^2 ds \leq E(0) \exp^{-kt_n} \quad (8.131)$$

Calculando a integral pelo método dos trapézios, para n fixo, temos

$$E(t_n) = \frac{1}{2} \left\{ \frac{h}{2} \left(u(x_1, t_n)^2 + 2 \sum_{i=2}^m u(x_i, t_n)^2 + u(x_{m+1}, t_n)^2 \right) \right\}, \quad n = 0, 1, \dots, N$$

Como a solução aproximada é conhecida em todos os pontos (x_i, t_n) , podemos então obter a energia $E(0), E(t_1), \dots, E(t_N)$.

O próximo teorema garante que a equação do calor (8.128) tem um efeito regularizante sobre o dado inicial $u(x, 0)$ e a demonstração pode ser encontrada em [2].

Teorema 8.10. (*Efeito regularizante*) *Considere o problema (8.127). Então a solução $u(t) \in C^\infty(\Omega)$, para cada $t > 0$, mesmo que a temperatura inicial $u(x, 0)$ seja descontínua.*

Uma interessante consequência do teorema anterior é que o problema com dado final, ou seja, com a temperatura conhecida no final do processo, dado por,

$$\begin{cases} u_t(x, t) - \alpha \Delta u(x, t) + \beta u(x, t) = 0, & \forall (x, t) \in Q \\ u(x, t) = 0, & \forall x \in \partial\Omega \text{ e } \forall t \in [0, T] \\ u(x, T) = u_T(x), & \forall x \in \Omega, \end{cases} \quad (8.132)$$

não tem solução. Por outro lado se substituirmos o sinal do termo u_t na primeira equação de (8.132)₁, ou seja, $-u_t(x, t) - \alpha \Delta u(x, t) + \beta u(x, t) = 0$, $\forall (x, t) \in Q$ então teremos o mesmos resultados anteriores, bastando para isso fazer a troca de variável t por $(T - t)$.

Análise Numérica: Equação da Onda

Nesse capítulo provaremos os resultados de convergência dos métodos numéricos discutidos para o problema modelo hiperbólico de segunda ordem, tanto o problema semi-discreto quanto o problema totalmente discreto.

Muitos dos resultados de convergência para o caso semidiscreto e discreto podem ser encontrados nas referências [1, 10, 11, 17].

9.1 Estimativa de Erro

Nessa seção serão feitas as estimativas de erro para as normas $L^2(\Omega)$ e $H_0^1(\Omega)$, para o problema semidiscreto e problema discreto. Consequentemente, em razão da estimativa ótima e da regularidade da solução, consideraremos $k = 1$ o grau de polinômio interpolador e representaremos $V_m^k = V_m$.

9.1.1 Problema Semidiscreto

Vimos em (7.3) que a formulação fraca para a equação da onda é dada por

$$\left\{ \begin{array}{l} (u'', v) + a(u, v) = (f, v), \quad \forall v \in H_0^1(0, 1) \\ (u(0), v) = (u_0, v), \quad \forall v \in H_0^1(0, 1) \\ (u'(0), v) = (u_1, v), \quad \forall v \in H_0^1(0, 1). \end{array} \right. \quad (9.1)$$

onde

$$a(u, v) = (u, v)_a = \alpha \int_0^1 u_x(t) v_x dx + \beta \int_0^1 u(t) v dx, \quad (9.2)$$

cujas normas serão denotadas por $\|\cdot\|_a = \alpha\|\cdot\|_1 + \beta|\cdot|$, e

$$(f, v) = \int_0^1 f(t) v \, dx, \quad (9.3)$$

Considere agora o espaço das soluções aproximadas $u_h \in V_m$, com V_m definido em (8.1). Assim o problema aproximado é dado por

$$\begin{cases} (u_h'', v_h) + a(u_h, v_h) = (f, v_h), \quad \forall v_h \in V_m \\ (u_h(0), v_h) = (u_{0h}, v_h), \quad \forall v_h \in V_m \\ (u_h'(0), v_h) = (u_{1h}, v_h), \quad \forall v_h \in V_m. \end{cases} \quad (9.4)$$

Tomando $v = v_h$ e fazendo a diferença entre os problemas (9.1) e (9.4) obtemos

$$\begin{cases} (u'' - u_h'', v_h) + a(u - u_h, v_h) = 0, \quad \forall v_h \in V_m \\ (u(0) - u_h(0), v_h) = (u_0 - u_{0h}, v_h), \quad \forall v_h \in V_m \\ (u'(0) - u_h'(0), v_h) = (u_1 - u_{1h}, v_h), \quad \forall v_h \in V_m \end{cases} \quad (9.5)$$

Considere a interpolação (8.5) definida como anteriormente por

$$a(u(t) - \tilde{u}(t), v) = 0, \quad \forall v \in V_m, \quad (9.6)$$

onde $\tilde{u}(t)$ é definido por (8.6) e são válidas as estimativas (8.7).

Norma $H_0^1(\Omega)$

Os dados iniciais $u_h(0) = u_{0h} \in V_m$ e $u_h'(0) = u_{1h} \in V_m$ são fundamentais para a determinação da solução aproximada $u_h(x, t)$, obtida após a resolução de um sistema de equações diferenciais ordinárias cujos valores iniciais são exatamente a posição $u_h(0)$ e a velocidade inicial $u_h'(0)$. 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 $h \rightarrow 0$, então é fundamental que $u_h(0)$ e $u_h'(0)$ convirjam, respectivamente, para $u(0)$ e $u'(0)$ em alguma norma, quando $h \rightarrow 0$. Para conseguir a convergência, requeremos que

$$\|u_h(0) - \tilde{u}(0)\|_1 \leq c_1 h; \quad |u_h'(0) - \tilde{u}'(0)|_0 \leq c_2 h, \quad (9.7)$$

onde $\tilde{u}(0)$ e $\tilde{u}'(0)$ são os interpoladores de $u(0)$ e $u'(0)$ em V_m . Lembrando que poderíamos ter tomado os dados iniciais $u_h(0)$ e $u_h'(0)$ como os próprios interpoladores, ou seja $u_h(0) = \tilde{u}(0)$ e $u_h'(0) = \tilde{u}'(0)$, desde que são dados do problema.

Teorema 9.1. *Sejam os dados iniciais satisfazendo (9.7) e $\{u, u', u''\} \in L^\infty(0, T; H_0^1 \cap H^2)$. Então 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 $\mathcal{O}(h)$, ou seja, satisfaz a seguinte estimativa,*

$$\|u - u_h\|_{L^\infty(0, T; H_0^1(0, 1))} + \|u' - u'_h\|_{L^\infty(0, T; L^2(0, 1))} \leq ch, \quad (9.8)$$

onde c é uma constante positiva independente de h .

Note que, pela definição da norma $L^\infty(0, T)$, para obter a estimativa (9.8) é suficiente obter a seguinte estimativa

$$\|u - u_h\|_1(t) + |u' - u'_h|_0(t) \leq ch, \quad \forall t \in [0, T] \quad (9.9)$$

Observação A constante c não depende de h , mas depende da solução $u(x, t)$ e também depende do intervalo $t \in [0, T]$. Assim a sequência dos valores de c associados com a sucessão de intervalos do tempo $[0, T]$ pode tender ao infinito quando $t \rightarrow \infty$ e dessa forma nada podemos afirmar sobre o comportamento assintótico do erro neste caso.

Demonstração:

Seja $e(t) = u(t) - u_h(t) = (u(t) - \tilde{u}(t)) + (\tilde{u}(t) - u_h(t)) = \rho(t) + \xi(t)$, onde $\tilde{u}(t)$ é o interpolante de $u(t)$ em V_m . Somando e subtraindo $\tilde{u}(t)$ em (9.5), obtemos

$$(\rho''(t), v_h) + (\xi''(t), v_h) + a(\xi(t), v_h) = 0, \quad \forall v_h \in V_m^k, \quad (9.10)$$

onde, por (9.6), $a(\rho(t), v_h) = 0$. Como $\xi(t)$ e $\xi'(t)$ pertencem a V_m para todo $t \in [0, T]$, podemos escolher $v_h = \xi'(t)$ na forma fraca (9.10), obtendo

$$(\rho''(t), \xi'(t)) + (\xi''(t), \xi'(t)) + a(\xi(t), \xi'(t)) = 0, \quad (9.11)$$

que é equivalente a

$$\frac{1}{2} \frac{d}{dt} \left\{ |\xi'(t)|_0^2 + \|\xi(t)\|_a^2 \right\} = -(\rho''(t), \xi'(t)), \quad (9.12)$$

Integrando (9.12) com respeito a $s \in [0, T]$ e usando a desigualdade de Schwarz e elementar, obtemos

$$\begin{aligned} |\xi'(t)|_0^2 + \|\xi(t)\|_a^2 &\leq |\xi'(0)|_0^2 + \|\xi(0)\|_a^2 + 2 \int_0^T |\rho''| |\xi'| \, ds \\ &\leq |\xi'(0)|_0^2 + \|\xi(0)\|_a^2 + \int_0^T |\rho''|^2 \, ds + \int_0^T |\xi'|^2 \, ds \end{aligned} \quad (9.13)$$

Note que $\|\xi(t)\|_a^2 = \alpha\|\xi(t)\|_1^2 + \beta\|\xi(t)\|_0^2$ e usando a hipótese (9.7) obtemos que $|\xi'(0)| \leq c_2 h$ e $\|\xi(0)\|_a \leq c_1 h$. Temos também para $m = 0$ e $k = 1$ em (8.8), que

$$|\rho''|_{L^\infty(0,T;L^2(\Omega))} \leq \hat{c}_2 h^2 \|u''\|_2.$$

Assim temos que

$$|\xi'(t)|^2 + \|\xi(t)\|_a^2 \leq (Ch)^2 + \int_0^t |\xi'(s)|^2 ds, \quad \forall t \in [0, T] \quad (9.14)$$

Usando a desigualdade de Gronwall, a integral do lado direito pode ser removida, obtendo-se que

$$|\xi'(t)|^2 + \|\xi(t)\|_a^2 \leq (Ch)^2, \quad \forall t \in [0, T] \quad (9.15)$$

Mas

$$(|\xi'(t)|_0 + \|\xi(t)\|_a)^2 \leq 2|\xi'(t)|_0^2 + 2\|\xi(t)\|_a^2 \leq 2(Ch)^2 \quad \forall t \in [0, T]$$

Extraindo a raiz em ambos os membros obtemos a estimativa

$$|\xi'(t)| + \|\xi(t)\|_a \leq \hat{C}h.$$

Note que

$$\min\{\alpha, \beta\} (\|\xi(t)\|_1 + |\xi(t)|_0) \leq \|\xi(t)\|_a = \alpha\|\xi\|_1 + \beta|\xi|_0 \leq \hat{C}h$$

Portanto, desde que o lado direito é independente do tempo $t \in [0, T]$ então

$$|\xi'|_{L^\infty(0,T;L^2(0,1))} + \|\xi\|_{L^\infty(0,T;H_0^1(0,1))} \leq \frac{\hat{C}}{\min\{\alpha, \beta\}} h$$

concluindo assim a estimativa de erro \square

Norma $L^2(\Omega)$

Para obter a Estimativa na erro na norma $L^2(\Omega)$, é necessário que os valores iniciais sejam mais precisos, devem ser tomados a satisfazer pelo menos a estimativa,

$$\|u_h(0) - \tilde{u}(0)\|_1 \leq c_1 h^2; \quad |u'_h(0) - \tilde{u}'(0)|_0 \leq c_2 h^2, \quad (9.16)$$

onde, como definido anteriormente, $\tilde{u}(0)$ e $\tilde{u}'(0)$ são os interpoladores de $u(0)$ e $u'(0)$ em V_m .

Teorema 9.2. *Sejam os dados iniciais satisfazendo (9.16) e $\{u, u', u''\} \in L^\infty(0, T; H_0^1 \cap H^2)$. Então o erro entre a solução aproximada $u_h(x, t)$ e a solução exata $u(x, t)$ na norma $L^2(\Omega)$ é de ordem $\mathcal{O}(h^2)$, ou seja, satisfaz a seguinte estimativa,*

$$\|u - u_h\|_{L^\infty(0,T;L^2(\Omega))} + \|u' - u'_h\|_{L^\infty(0,T;L^2(\Omega))} \leq ch^2, \quad (9.17)$$

onde c é uma constante positiva independente de h .

Demonstração:

Note que, pela definição da norma $L^\infty(0, T)$, para obter a estimativa (9.17) é suficiente obter a seguinte estimativa,

$$\|u(t) - u_h(t)\|_0 + |u'(t) - u'_h(t)|_0 \leq ch^2, \quad \forall t \in [0, T]. \quad (9.18)$$

Com efeito, considere a decomposição do erro,

$$e(t) = u(t) - u_h(t) = (u(t) - \tilde{u}(t)) + (\tilde{u}(t) - u_h(t)) = \rho(t) + \xi(t).$$

Fazendo mesmo procedimento, como em (9.10) e usando a projeção ortogonal (9.6) obtemos a seguinte relação, equivalente a (9.13);

$$|\xi'(t)|_0^2 + \|\xi(t)\|_a^2 \leq |\xi'(0)|_0^2 + \|\xi(0)\|_a^2 + \int_0^t |\rho''|^2 ds + \int_0^t |\xi'|^2 ds. \quad (9.19)$$

Para obter a estimativa em $L^2(\Omega)$, não estamos interessado numa limitação para o termo $\|\xi(t)\|_a$ que é equivalente a $|\nabla \xi(t)|_0$. Esse termo positivo pode ser removido do lado esquerdo da desigualdade (9.19). Então da desigualdade de Gronwall, obtém-se

$$|\xi'(t)|_0^2 \leq C_2 \left(|\xi'(0)|_0^2 + \|\xi(0)\|_1^2 + |\rho''|_{L^2(Q)}^2 \right). \quad (9.20)$$

Da mesma forma como obtemos (8.18) usando o Teorema Fundamental do Cálculo, para $w(0) \in L^2(\Omega)$, $w' \in L^2(\Omega)$, existe uma constante positiva c tal que;

$$|w(t)|_0^2 \leq c \left(|w_0|_0^2 + \int_0^t |w'(s)|_0^2 ds \right); \quad \forall t \in [0, T]. \quad (9.21)$$

Assim para $w = \rho'$ tem-se

$$|\rho'(t)|_0^2 \leq C_2 \left(|\rho'(0)|_0^2 + \int_0^t |\rho''(s)|_0^2 ds \right); \quad \forall t \in [0, T]. \quad (9.22)$$

Somando com (9.20), obtemos

$$|e'(t)|_0^2 \leq C_3 \left(|\rho''|_{L^2(Q)}^2 + |\xi'(0)|_0^2 + \|\xi(0)\|_1^2 + |\rho'(0)|_0^2 \right) \quad (9.23)$$

Considerando agora $w = e$ em (9.21), obtemos

$$|e(t)|_0^2 \leq C_4 \left(|e_0|_0^2 + \int_0^t |e'(s)|_0^2 ds \right); \quad \forall t \in [0, T] \quad (9.24)$$

Somando as equações (9.23) e (9.24) obtemos

$$\begin{aligned} |e(t)|_0^2 + |e'(t)|_0^2 \leq & C_5 \left(|\rho''|_{L^2(Q)}^2 + |\xi'(0)|^2 + \|\xi(0)\|_1^2 + |\rho'(0)|_0^2 + |\rho(0)|_0^2 \right. \\ & \left. + |\xi(0)|_0^2 + \int_0^t (|e(s)|_0^2 + |e'(s)|_0^2) ds \right) \end{aligned} \quad (9.25)$$

onde foi adicionado o termo positivo $\int_0^t |e(s)|^2 ds$ e usamos que $|e_0|_0^2 \leq |\rho(0)|_0^2 + |\xi(0)|_0^2$.

Definindo

$$\begin{aligned} \varphi(t) &= |e(t)|_0^2 + |e'(t)|_0^2 \\ \hat{C} &= (|\rho''|_{L^2(Q)}^2 + |\xi'(0)|^2 + \|\xi(0)\|_1^2 + |\rho'(0)|_0^2 + |\rho(0)|_0^2 + |\xi(0)|_0^2, \end{aligned}$$

então a inequação (9.25) pode ser escrita por

$$\varphi(t) \leq \hat{C} + \int_0^T \varphi(s) ds.$$

Aplicando o lemma de Gronwall, obtém-se que $\varphi(t)$ é limitada, ou seja

$$\begin{aligned} |e(t)|_0^2 + |e'(t)|_0^2 &\leq C_6 \left(|\rho''|_{L^2(Q)}^2 + |\xi'(0)|_0^2 + \|\xi(0)\|_1^2 \right. \\ &\quad \left. + |\rho'(0)|_0^2 + |\rho(0)|_0^2 + |\xi(0)|_0^2 \right) \leq C_7 h^4 \end{aligned} \quad (9.26)$$

Note que todos os termos do lado são de ordem $\mathcal{O}(h^2)$.

De fato;

1. Fazendo $m = 0$ e $k = 1$ em (8.8) temos

$$|\rho''|_0^2 \leq \hat{c}_2 h^4 \|u''\|_2^2 \quad \text{e} \quad |\rho'(0)|_0^2 \leq C h^4 \|u'\|_2^2$$

2. De (9.16),

$$\|\xi(0)\|_1^2 = \|u_h(0) - \tilde{u}(0)\|_1^2 \leq c_1 h^4 \quad \text{e} \quad |\xi'(0)|_0^2 = |u'_h(0) - \tilde{u}'(0)|_0^2 \leq c_2 h^4.$$

Assim, como a norma Euclidiana em qualquer espaço de dimensão finita é equivalente a norma definida pela soma dos valores absolutos das componentes dos vetores, segue que

$$\left(|e(t)|_0 + |e'(t)|_0 \right)^2 \leq 2|e(t)|_0^2 + 2|e'(t)|_0^2 \leq C_7 h^4, \quad \forall t \in [0, T]$$

Logo, extraindo a raiz quadrada obtemos

$$|e(t)|_0 + |e'(t)|_0 \leq \hat{C}h^2, \quad \forall t \in [0, T]$$

Como o lado direito independe da variável t , podemos tomar o supremo em t e obter a estimativa:

$$|e|_{L^\infty(0,T;L^2(\Omega))} + |e'|_{L^\infty(0,T;L^2(\Omega))} \leq \hat{C}h^2 \quad \square \quad (9.27)$$

9.1.2 Problema Discreto

Método de Newmark

Considere o sistema (7.8) e a seguinte aproximação para $(7.8)_1$

$$A \frac{(u^{n+1} - 2u^n + u^{n-1}))}{(\Delta t)^2} + Bu^{*n} = F^{*n}.$$

Multiplicando a equação por $(\Delta t)^2$, obtemos

$$A(u^{n+1} - 2u^n + u^{n-1}) + (\Delta t)^2 Bu^{*n} = (\Delta t)^2 F^{*n}, \quad (9.28)$$

onde u^{*n} representa a seguinte média aritmética ponderada, $u^{*n} = \theta u^{n+1} + (1 - 2\theta)u^n + \theta u^{n-1}$, (ver em (8.31)). Fazendo as substituições, obtemos o seguinte sistema de equações lineares

$$\begin{aligned} (A + \theta(\Delta t)^2 B)u^{n+1} &= (2A - (1 - 2\theta)(\Delta t)^2 B)u^n - (A + \theta(\Delta t)^2 B)u^{n-1} \\ &+ (\Delta t)^2 \left(\theta(F^{n+1} + F^{n-1}) + (1 - 2\theta)F^n \right) \end{aligned} \quad (9.29)$$

O método de aproximação acima é conhecido como método θ –Newmark e requer que $\theta \geq \frac{1}{4}$. Note que quando $\theta = 0$, obtemos o método da diferença central.

Dados Iniciais

Para inicializar o algoritmo (9.29), para obtenção das soluções aproximadas u^2, u^3, \dots, u^N , correspondentes aos tempos $2\Delta t; 3\Delta t; \dots; N\Delta t = T$ precisamos conhecer os valores u^1 e u^0 . Para obter a estimativa de erro, são necessários estimar os valores iniciais. Dessa forma segue o Lema:

Lema 9.1. *Os dados iniciais satisfazem a seguinte estimativa em $H_0^1(\Omega)$ e $L^2(\Omega)$:*

$$\begin{aligned} \|\tilde{u}^0 - u_h^0\|_1 + \|\tilde{u}^1 - u_h^1\|_1 + \|\delta(\tilde{u} - u_h)^{1/2}\|_0 &= \|\xi^0\|_1 + \|\xi^1\|_1 + \|\delta\xi^{1/2}\|_0 \\ &\leq C(h + (\Delta t)^2) \end{aligned} \quad (9.30)$$

$$\begin{aligned} \|\tilde{u}^0 - u_h^0\|_0 + \|\tilde{u}^1 - u_h^1\|_0 + \|\delta(\tilde{u} - u_h)^{1/2}\|_0 &= \|\xi^0\|_0 + \|\xi^1\|_0 + \|\delta\xi^{1/2}\|_0 \\ &\leq C(h^2 + (\Delta t)^2) \end{aligned} \quad (9.31)$$

onde C representa diferentes constantes positivas independente de h e (Δt) .

Demonstração:

Os dados iniciais são conhecidos, ou seja, $u(x, 0) = u_0(x)$ e $u'(x, 0) = u_1(x)$. Assim podemos definir, como polinômio interpolador de $u_0(x)$ a função definida por:

$$\tilde{u}_0(x) = \sum_{i=1}^m u_0(x_i) \varphi_i(x) \in V_m$$

Pela ortogonalidade da função base $\varphi_i(x)$, então para $x = x_i$, $\tilde{u}_0(x_i) = u_0(x_i)$, ou seja a função \tilde{u}_0 interpola a função $u_0(x)$ nos pontos x_i . Nesse caso podemos tomar a solução aproximada $u_{0h}(x) = \tilde{u}_0(x)$ em V_m . Consequentemente, temos a estimativa no tempo inicial $t_0 = 0$, com $m \leq k$ dada por,

$$\|u_0 - u_{0h}\|_m \leq \|u_0 - \tilde{u}_0\|_m + \|\tilde{u}_0 - u_{0h}\|_m = \|\rho(0)\|_m + \|\xi(0)\|_m = \|\rho(0)\|_m \leq ch^{k+1-m}.$$

Por outro lado, no tempo $t_1 = \Delta t$, a solução exata $u(x, t_1)$ não é conhecida e assim não é possível determinar diretamente a sua interpolante $\tilde{u}(x, t_1) = \tilde{u}^1(x)$. Entretanto, podemos determinar uma aproximação $\hat{u}(x)$ para a solução exata $u(x, t_1) = u^1(x)$, fazendo uma expansão de segunda ordem pela série de Taylor da solução exata u , na vizinhança do tempo $t_0 = 0$, ou seja

$$\hat{u}^1 = u_0 + \Delta t u'_0 + \frac{(\Delta t)^2}{2!} u''_0. \quad (9.32)$$

A posição inicial u_0 e a velocidade inicial $u'(0)$ são dados do problema e a aceleração inicial $u''(0)$ pode ser obtida fazendo $t = 0$ no problema (7.1)

$$u''(0) = \alpha \frac{\partial^2 u}{\partial x^2}(0) - \beta u(0) + f(0).$$

Assim, o erro da aproximação de \hat{u} para u^1 é dado por

$$u^1(x, \cdot) - \hat{u}^1(x, \cdot) = \int_0^{\Delta t} \frac{(\Delta t - s)^2}{2!} u'''(x, s) ds. \quad (9.33)$$

Supondo que a solução u seja suficientemente regular, tal que $u''' \in L^\infty(0, T, H_0^1(\Omega))$ então temos que

$$\|u^1 - \hat{u}^1\|_1 \leq \|u'''\|_{L^\infty(0, T, H_0^1(\Omega))} \int_0^{\Delta t} \frac{(\Delta t - s)^2}{2!} \leq \hat{c}(\Delta t)^3 \quad (9.34)$$

onde \hat{c} independe de Δt .

Por (9.32) a extrapolação $\hat{u}^1(\cdot)$ é conhecida e além disso pertence ao espaço

$H^2(0, 1) \cap H_0^1(0, 1)$. Assim, também podemos calcular a sua interpolação em V_m na forma

$$\tilde{u}^1(x) = \sum_{i=1}^m \hat{u}^1(x_i) \varphi_i(x) \in V_m, \quad (9.35)$$

então para $x = x_i$, $\tilde{u}^1(x_i) = \hat{u}^1(x_i)$. Logo, podemos tomar, como solução aproximada no tempo t_1 , a função $u_h^1(x) = \tilde{u}^1(x)$.

Então, usando a desigualdade triangular,

$$\begin{aligned} \|u^1 - u_h^1\|_1 &\leq \|u^1 - \hat{u}^1\|_1 + \|\hat{u}^1 - u_h^1\|_1 \leq c(\Delta t)^3 + \|\hat{u}^1 - \tilde{u}^1\|_1 + \|\tilde{u}^1 - u_h^1\|_1 \\ &\leq c(\Delta t)^3 + \|\rho^1\|_1 \leq c_1((\Delta t)^3 + h) \end{aligned} \quad (9.36)$$

De forma equivalente, substituindo a norma $\|\cdot\|_1$ pela norma $\|\cdot\|_0$ temos a seguinte estimativa:

$$\begin{aligned} \|u^1 - u_h^1\|_0 &\leq \|u^1 - \hat{u}^1\|_0 + \|\hat{u}^1 - u_h^1\|_0 \leq c(\Delta t)^3 + \|\hat{u}^1 - \tilde{u}^1\|_0 + \|\tilde{u}^1 - u_h^1\|_0 \\ &\leq c(\Delta t)^3 + \|\rho^1\|_0 \leq c_2((\Delta t)^3 + h^2) \end{aligned} \quad (9.37)$$

Assim de (9.36) e (9.37) concluímos que:

$$\|e^1\|_1 \leq c_1(h + (\Delta t)^3) \quad \text{e} \quad \|e^1\|_0 \leq c_2(h^2 + (\Delta t)^3)$$

Além disso, sabemos que $\delta e^{1/2} = \frac{1}{\Delta t}(e^1 - e^0)$. Logo

$$\|\delta e^{1/2}\|_0 \leq \frac{1}{\Delta t}(\|e^1\|_0 + \|e^0\|_0) \leq c_3(h^2 + (\Delta t)^2) \quad (9.38)$$

Portanto

$$\|e^1\|_1 + \|e^0\|_1 + \|\delta e^{1/2}\|_0 \leq c(h + (\Delta t)^2).$$

De forma análoga, observando (9.37), tem-se para a norma $L^2(\Omega)$ que

$$\|e^1\|_0 + \|e^0\|_0 + \|\delta e^{1/2}\|_0 \leq c(h^2 + (\Delta t)^2).$$

Finalmente, usando a decomposição do erro, conclui-se o Lema. \square

Estimativa de Erro no Tempo Discreto

Nessa seção faremos a estimativa de erro, ou ordem de convergência na norma discreta $L^\infty(0, T; H_0^1(\Omega))$ e $L^\infty(0, T; L^2(\Omega))$, quando o tempo t varia discretamente na forma $t_n = n\Delta t$, $n = 0, 1, \dots, N$, para o método de Neumark.

No Teorema(9.5) do final desse capítulo, são mostrados resultados de existência e unicidade de solução para o Problema(9.1). Para obter as estimativas de erro no tempo discreto, são necessários resultados adicionais mais regulares que daremos na seguinte hipótese: são mostrados resultados de existência e unicidade de solução para o Problema(9.1). Para obter as estimativas de erro no tempo discreto, são necessários resultados adicionais mais regulares que daremos na seguinte hipótese:

HO: Suponha que a força f dada, a solução u do Problema(9.1) e suas derivadas tenham a seguinte regularidade:

$$\left\{ \begin{array}{l} f \in L^2(0, T; H^2(0, 1)), \quad u \in L^\infty(0, T; H_0^1(0, 1) \cap H^2(0, 1)) \\ \text{com as derivadas } \{u_t, u_{tt}, u_{ttt}, u_{tttt}\} \in L^\infty(0, T; L^2(0, 1)). \end{array} \right. \quad (9.39)$$

Os resultados a seguir são os análogos discretos dos Teorema(9.1) e Teorema(9.2).

Teorema 9.3. *Sob a hipótese (9.39) e os dados iniciais $u_0 \in H_0^1(0, 1) \cap H^2(0, 1)$ e $u_1 \in H_0^1(0, 1)$ então o erro entre a solução exata u do Problema(9.1) e da solução aproximada u_h do Problema(9.4) obtida pelo método de Neumark (9.29), tem as seguintes estimativas nas normas $H_0^1(0, 1)$ e $L^2(0, 1)$ respectivamente:*

$$\left\{ \begin{array}{l} \|u - u_h\|_{\hat{L}^\infty(0, T; H_0^1(\Omega))} + \|\delta(u - u_h)\|_{\tilde{L}^\infty(0, T; L^2(\Omega))} \leq C(h + (\Delta t)^2) \\ \|u - u_h\|_{\hat{L}^\infty(0, T; L^2(\Omega))} + \|\delta(u - u_h)\|_{\tilde{L}^\infty(0, T; L^2(\Omega))} \leq C(h^2 + (\Delta t)^2) \end{array} \right. \quad (9.40)$$

onde

(a) Para $\theta \geq 1/4$, o sistema é incondicionalmente estável para todo $\Delta t > 0$.

(b) Para $\theta < 1/4$, o sistema é condicionalmente estável.

A constante C é positiva e independe da malha h e da discretização Δt .

Demonstração:

Estimativa de Erro: Norma $H_0^1(\Omega)$

Como vimos anteriormente as normas são definidas como segue

$$\begin{aligned} \|w\|_{\hat{L}^\infty(0,T;H^1(\Omega))} &= \max \left\{ \|w^n\|_{H^1(\Omega)}, \quad n = 0, 1, \dots, N-1 \right\}, \\ \|w\|_{\tilde{L}^\infty(0,T;L^2(\Omega))} &= \max \left\{ \|w^{n+\frac{1}{2}}\|_{L^2(\Omega)}, \quad n = 0, 1, \dots, N-1 \right\} \end{aligned} \quad (9.41)$$

Considere a decomposição do erro na forma $e = u - u_h = (u - \tilde{u}) + (\tilde{u} - u_h) = \rho + \xi$, onde \tilde{u} é o interpolante de u em V_m , ou equivalentemente \tilde{u} é a projeção da solução exata u em V_m , com respeito ao produto interno da energia $(\cdot, \cdot)_a$, ou seja,

$$a(u - \tilde{u}, v) = a(\rho, v) = 0, \quad \forall t \in [0, T]; \quad \forall v \in V_m,$$

onde a forma bilinear $a(\cdot, \cdot)$ está definida em (6.2).

Para demonstrar o teorema, inicialmente mostraremos a seguinte estimativa preliminar:

$$\begin{aligned} \|\xi\|_{\hat{L}^\infty(0,T;H^1(\Omega))} + \|\delta\xi\|_{\tilde{L}^\infty(0,T;L^2(\Omega))} &\leq c \left(\|\rho_{tt}\|_{L^2(0,T;L^2(\Omega))} + \|\xi^0\|_1 + \|\xi^1\|_1 \right. \\ &\quad \left. + \|\delta\xi^{-1/2}\|_0 + (\Delta t)^2 \|u_{tttt}\|_{L^2(Q)} \right) \end{aligned} \quad (9.42)$$

A formulação fraca do problema é dada por

$$(u_{tt}, v)_0 + a(u, v) = (f, v)_0, \quad \forall v \in H_0^1(\Omega). \quad (9.43)$$

Fazendo $t = (n+1), n, (n-1)$ com pesos $\theta, (1-2\theta), \theta$ respectivamente em (9.43), e somando os termos, obtemos

$$\begin{aligned} \theta(u_{tt}^{n+1}, v)_0 + \theta a(u^{n+1}, v) + (1-2\theta)(u_{tt}^n, v)_0 + (1-2\theta)a(u^n, v) + \theta(u_{tt}^{n-1}, v)_0 \\ + \theta a(u^{n-1}, v) = \theta(f^{n+1}, v)_0 + (1-2\theta)(f^n, v)_0 + \theta(f^{n-1}, v)_0 \end{aligned}$$

Usando a notação (8.31) obtém-se,

$$(u_{tt}^{*n}, v)_0 + a(u^{*n}, v) = (f^{*n}, v)_0, \quad \forall v \in H_0^1(\Omega)$$

Somando e subtraindo o termo $(\delta^2 u^n, v)$, obtemos

$$(\delta^2 u^n, v)_0 + a(u^{*n}, v) = (f^{*n} + g^n, v)_0, \quad \forall v \in H_0^1(\Omega) \quad (9.44)$$

onde $g^n = (\delta^2 u^n - u_{tt}^{*n})$ e $\delta^2 u^n$ é a aproximação do termo (u_{tt}) no tempo por diferenças finitas, dado em (8.31).

Por outro lado, para o problema aproximado, a formulação é dada por

$$(\delta^2 u_h^n, v_h)_0 + a(u_h^{*n}, v_h) = (f^{*n}, v_h)_0, \quad \forall v_h \in V_m \quad (9.45)$$

Tomando, em particular, $v = v_h \in V_m$ em (9.44) e subtraindo da equação (9.45), obtemos

$$(\delta^2 e^n, v_h)_0 + a(e^{*n}, v_h) = (g^n, v_h)_0, \quad \forall v_h \in V_m^k \quad (9.46)$$

Usando a decomposição do erro, obtém-se

$$(\delta^2 \xi^n, v_h)_0 + (\delta^2 \rho^n, v_h)_0 + a(\xi^{*n}, v_h) + a(\rho^{*n}, v_h) = (g^n, v_h)_0, \quad \forall v_h \in V_m \quad (9.47)$$

Considere em particular,

$$v_h = \frac{1}{2}(\delta \xi^{n+\frac{1}{2}} + \delta \xi^{n-1/2}) = \frac{1}{2\Delta t}(\xi^{n+1} - \xi^{n-1}).$$

Substituindo no primeiro termo de (9.47) e observando que

$$\begin{aligned} \delta^2 \xi^n &= \frac{1}{(\Delta t)^2}(\xi^{n+1} - 2\xi^n + \xi^{n-1}) = \frac{1}{(\Delta t)^2}((\xi^{n+1} - \xi^n) - (\xi^n - \xi^{n-1})) \\ &= \frac{1}{\Delta t}(\delta \xi^{n+\frac{1}{2}} - \delta \xi^{n-1/2}), \end{aligned}$$

temos

$$\begin{aligned} (\delta^2 \xi^n, v_h)_0 &= \left(\frac{1}{\Delta t}(\delta \xi^{n+\frac{1}{2}} - \delta \xi^{n-1/2}), \frac{1}{2}(\delta \xi^{n+\frac{1}{2}} + \delta \xi^{n-1/2}) \right)_0 \\ &= \frac{1}{2\Delta t}(\|\delta \xi^{n+\frac{1}{2}}\|_0^2 - \|\delta \xi^{n-1/2}\|_0^2) \end{aligned} \quad (9.48)$$

Para o terceiro termo de (9.47), usando a notação (9.2) temos

$$\begin{aligned} a(\xi^{*n}, v_h) &= (\xi^{*n}, v_h)_a = \frac{1}{2\Delta t}(\xi^{*n}, \xi^{n+1} - \xi^{n-1})_a = \frac{\theta}{2\Delta t}(\xi^{n+1}, \xi^{n+1} - \xi^{n-1})_a \\ &\quad + \left(\frac{1-2\theta}{2\Delta t} \right) (\xi^n, \xi^{n+1} - \xi^{n-1})_a + \frac{\theta}{2\Delta t}(\xi^{n-1}, \xi^{n+1} - \xi^{n-1})_a \end{aligned}$$

Somando e subtraindo o termo ξ^n convenientemente, obtemos que

$$\begin{aligned} (\xi^{*n}, v_h)_a &= \frac{1}{2\Delta t}(\xi^n, \xi^{n+1} - \xi^{n-1})_a + \frac{\theta}{2\Delta t}(\xi^{n+1} - 2\xi^n + \xi^{n-1}, \xi^{n+1} - \xi^{n-1})_a \\ &= \frac{1}{2\Delta t}((\xi^{n+1}, \xi^n)_a - (\xi^n, \xi^{n-1})_a) + \frac{\theta}{2\Delta t}(\xi^{n+1} - \xi^n, \xi^{n+1} - \xi^n)_a \\ &\quad + \frac{\theta}{2\Delta t}(\xi^{n+1} - \xi^n, \xi^n - \xi^{n-1})_a - \frac{\theta}{2\Delta t}(\xi^n - \xi^{n-1}, \xi^{n+1} - \xi^n)_a \\ &\quad - \frac{\theta}{2\Delta t}(\xi^n - \xi^{n-1}, \xi^n - \xi^{n-1})_a = \frac{1}{2\Delta t}((\xi^{n+1}, \xi^n)_a - (\xi^n, \xi^{n-1})_a) \\ &\quad + \frac{\theta}{2\Delta t}(\|\xi^{n+1} - \xi^n\|_a^2 - \|\xi^n - \xi^{n-1}\|_a^2) \end{aligned} \quad (9.49)$$

Substituindo (9.48) e (9.49) em (9.47) obtém-se que

$$\begin{aligned} & \frac{1}{2\Delta t} \left(\|\delta\xi^{n+\frac{1}{2}}\|_0^2 - \|\delta\xi^{n-1/2}\|_0^2 \right) + \frac{1}{2\Delta t} \left((\xi^{n+1}, \xi^n)_a - (\xi^n, \xi^{n-1})_a \right) \\ & + \frac{\theta}{2\Delta t} \left(\|\xi^{n+1} - \xi^n\|_a^2 - \|\xi^n - \xi^{n-1}\|_a^2 \right) = \frac{1}{2} \left(g^n, \delta\xi^{n+\frac{1}{2}} + \delta\xi^{n-1/2} \right)_0 \\ & - \frac{1}{2} \left(\delta^2 \rho^n, \delta\xi^{n+1/2} + \delta\xi^{n-1/2} \right)_0 \end{aligned} \quad (9.50)$$

Observe que $a(\rho^{*n}, v_h) = a(u^{*n} - \tilde{u}^{*n}, v_h) = 0$, pois $(u - \tilde{u})$ é ortogonal ao subespaço V_m com relação ao produto interno $(\cdot, \cdot)_a$, independente do tempo.

Multiplicando igualdade por $2\Delta t$ e fazendo a soma para $n = 1, \dots, (N-1)$ e observando o cancelamento dos termos comuns obtemos

$$\begin{aligned} & \|\delta\xi^{N-1/2}\|_0^2 - \|\delta\xi^{1/2}\|_0^2 + (\xi^N, \xi^{N-1})_a - (\xi^1, \xi^0)_a + \theta (\|\xi^N - \xi^{N-1}\|_a^2) \\ & - \theta (\|\xi^1 - \xi^0\|_a^2) = \Delta t \sum_{n=1}^{N-1} \left(g^n - \delta^2 \rho^n, \delta\xi^{n+\frac{1}{2}} + \delta\xi^{n-1/2} \right)_0 \end{aligned} \quad (9.51)$$

Usando a desigualdade de Schwarz para os termos do lado esquerdo da igualdade obtemos:

$$\begin{aligned} & \|\delta\xi^{N-1/2}\|_0^2 + (\xi^N, \xi^{N-1})_a + \theta \|\xi^N - \xi^{N-1}\|_a^2 \leq C \left\{ \|\delta\xi^{1/2}\|_0^2 + \|\xi^1\|_a^2 + \|\xi^0\|_a^2 \right. \\ & \left. + \Delta t \sum_{n=1}^{N-1} \left(g^n, \delta\xi^{n+\frac{1}{2}} + \delta\xi^{n-1/2} \right)_0 - \Delta t \sum_{n=1}^{N-1} \left(\delta^2 \rho^n, \delta\xi^{n+\frac{1}{2}} + \delta\xi^{n-1/2} \right)_0 \right\} \end{aligned} \quad (9.52)$$

Vamos analisar os termos do lado direito envolvendo os somatórios em (9.52).

Para o primeiro termo, temos

$$\begin{aligned} \Delta t \sum_{n=1}^{N-1} \left(g^n, \delta\xi^{n+\frac{1}{2}} + \delta\xi^{n-1/2} \right)_0 &= \Delta t \sum_{n=1}^{N-1} \left(g^n, \delta\xi^{n+\frac{1}{2}} \right)_0 + \Delta t \sum_{n=1}^{N-1} \left(g^n, \delta\xi^{n-1/2} \right)_0 \\ &\leq \Delta t \sum_{n=1}^{N-1} \|g^n\|_0 \left(\|\delta\xi^{n+\frac{1}{2}}\|_0 + \|\delta\xi^{n-1/2}\|_0 \right) \end{aligned}$$

Usando a desigualdade elementar obtemos,

$$\begin{aligned}
\Delta t \sum_{n=1}^{N-1} \|g^n\|_0 \|\delta \xi^{n+\frac{1}{2}}\|_0 &\leq \frac{\Delta t}{2\varepsilon} \sum_{n=1}^{N-1} \|g^n\|_0^2 + \frac{\Delta t \varepsilon}{2} \sum_{n=1}^{N-1} \|\delta \xi^{n+\frac{1}{2}}\|_0^2 \\
&\leq \frac{\Delta t}{2\varepsilon} \sum_{n=1}^{N-1} \|g^n\|_0^2 + \frac{\Delta t \varepsilon}{2} \sum_{n=1}^{N-1} \|\delta \xi^{n+\frac{1}{2}}\|_0^2 + \frac{\Delta t \varepsilon}{2} \|\delta \xi^{1/2}\|_0^2 \\
&= \frac{1}{2\varepsilon} \|g\|_{\tilde{L}^2(0,T;L^2)}^2 + \frac{\Delta t \varepsilon}{2} \sum_{n=1}^{N-1} \|\delta \xi^{n-1/2}\|_0^2 + \frac{\Delta t \varepsilon}{2} \|\delta \xi^{N-1/2}\|_0^2,
\end{aligned}$$

onde ε é uma constante positiva a ser definida.

Analogamente para o segundo termo do lado direito, tem-se

$$\Delta t \sum_{n=1}^{N-1} \|g^n\|_0 \|\delta \xi^{n-1/2}\|_0 \leq \frac{1}{2\varepsilon} \|g\|_{\tilde{L}^2(0,T;L^2)}^2 + \frac{\Delta t \varepsilon}{2} \sum_{n=1}^{N-1} \|\delta \xi^{n-1/2}\|_0^2$$

Logo o primeiro termo em (9.52) pode ser escrito na forma

$$\begin{aligned}
\Delta t \sum_{n=1}^{N-1} \left(g^n, \delta \xi^{n+\frac{1}{2}} + \delta \xi^{n-1/2} \right)_0 &\leq \frac{1}{\varepsilon} \|g\|_{\tilde{L}^2(0,T;L^2)}^2 + \Delta t \varepsilon \sum_{n=1}^{N-1} \|\delta \xi^{n-1/2}\|_0^2 \\
&\quad + \frac{\Delta t \varepsilon}{2} \|\delta \xi^{N-1/2}\|_0^2
\end{aligned} \tag{9.53}$$

Para o segundo termo do somatório em (9.52), usando a desigualdade elementar e (8.39), obtém-se que

$$\begin{aligned}
-\Delta t \sum_{n=1}^{N-1} (\delta^2 \rho^n, \delta \xi^{n+\frac{1}{2}}) - \Delta t \sum_{n=1}^{N-1} (\delta^2 \rho^n, \delta \xi^{n-1/2})_0 &\leq \frac{\Delta t}{2\varepsilon} \sum_{n=1}^{N-1} \|\delta^2 \rho^n\|^2 \\
&\quad + \frac{\Delta t \varepsilon}{2} \sum_{n=1}^{N-1} \|\delta \xi^{n+\frac{1}{2}}\|^2 + \frac{\Delta t}{2\varepsilon} \sum_{n=1}^{N-1} \|\delta^2 \rho^n\|^2 + \frac{\Delta t \varepsilon}{2} \sum_{n=1}^{N-1} \|\delta \xi^{n-1/2}\|^2 \\
&\leq \frac{\Delta t}{\varepsilon} \sum_{n=1}^{N-1} \|\delta^2 \rho^n\|^2 + \Delta t \varepsilon \sum_{n=1}^{N-1} \|\delta \xi^{n-1/2}\|^2 + \frac{\Delta t \varepsilon}{2} \|\delta \xi^{N-1/2}\|^2 \\
&\leq \frac{1}{\varepsilon} \|\rho_{tt}\|_{\tilde{L}^2(0,T;L^2)}^2 + \Delta t \varepsilon \sum_{n=1}^{N-1} \|\delta \xi^{n-1/2}\|^2 + \frac{\Delta t \varepsilon}{2} \|\delta \xi^{N-1/2}\|^2
\end{aligned} \tag{9.54}$$

Substituindo as desigualdades (9.53) e (9.54) em (9.52), obtém-se que

$$\begin{aligned}
(1 - \Delta t \varepsilon) \|\delta \xi^{N-1/2}\|^2 &+ (\xi^N, \xi^{N-1})_a + \theta \|\xi^N - \xi^{N-1}\|_a^2 \\
&\leq C \left\{ \Delta t \varepsilon \sum_{n=1}^{N-1} \|\delta \xi^{n-1/2}\|^2 + \Psi^2 \right\}
\end{aligned} \tag{9.55}$$

onde

$$\Psi^2 = \frac{1}{\varepsilon} \|\rho_{tt}\|_{\hat{L}^2(0,T;L^2)}^2 + \frac{1}{\varepsilon} \|g\|_{\hat{L}^2(0,T;L^2)}^2 + \|\delta\xi^{1/2}\|_0^2 + \|\xi^1\|_a^2 + \|\xi^0\|_a^2. \quad (9.56)$$

Usando a **Identidade polar**, podemos escrever o que segue:

$$(\xi^N, \xi^{N-1})_a = \frac{1}{4} \left(\|\xi^N + \xi^{N-1}\|_a^2 - \|\xi^N - \xi^{N-1}\|_a^2 \right)$$

Substituindo em (9.55), obtemos

$$\begin{aligned} (1 - \Delta t\varepsilon) \|\delta\xi^{N-1/2}\|^2 + \frac{1}{4} \|\xi^N + \xi^{N-1}\|_a^2 + (\theta - \frac{1}{4}) \|\xi^N - \xi^{N-1}\|_a^2 \\ \leq C \left\{ \Psi^2 + \Delta t\varepsilon \sum_{n=1}^{N-1} \|\delta\xi^{n-1/2}\|^2 \right\} \end{aligned} \quad (9.57)$$

Todos os procedimentos feitos tem por objetivo aplicar o Lema (8.2), para obter a limitação (8.35), que é o análogo discreto do Lema de Gronwall. Para isso são necessários que as constantes do lado esquerdo da desigualdade sejam todas positivas. Como ε é um número arbitrário, real e positivo então para que a constante $(1 - \Delta t\varepsilon)$ seja positiva é necessário que $\varepsilon < 1/\Delta t$. Por outro lado, para que a constante $(\theta - 1/4)$ seja não negativa é necessário que $\theta \geq 1/4$. Dessa forma para obter a convergência da família dos θ métodos, denominados métodos de Newmark, a constante θ tem que ser maior ou igual a $1/4$. Para concluir a demonstração, vamos considerar dois casos:

(i) $\theta > 1/4$

(ii) $\theta = 1/4$

Caso (i): $\theta > 1/4$

Definimos $\varepsilon < 1/\Delta t$ e $\hat{C} = \min \{1 - \Delta t\varepsilon, \theta - 1/4, 1/4\} > 0$. Tomando o mínimo em (9.57) e dividindo por \hat{C} ,

$$\|\delta\xi^{N-1/2}\|_0^2 + \|\xi^N + \xi^{N-1}\|_a^2 + \|\xi^N - \xi^{N-1}\|_a^2 \leq \frac{C}{\hat{C}} \left\{ \Psi^2 + \Delta t \sum_{n=1}^{N-1} \|\delta\xi^{n-1/2}\|^2 \right\}$$

Como

$$\|\delta\xi^{N-1/2}\|_0^2 + \|\xi^N\|_a^2 + \|\xi^{N-1}\|_a^2 \leq \|\delta\xi^{N-1/2}\|_0^2 + \|\xi^N + \xi^{N-1}\|_a^2 + \|\xi^N - \xi^{N-1}\|_a^2$$

Então

$$\|\delta\xi^{N-1/2}\|_0^2 + \|\xi^N\|_a^2 + \|\xi^{N-1}\|_a^2 \leq C_1 \left\{ \Psi^2 + \Delta t \sum_{n=1}^{N-1} \|\delta\xi^{n-1/2}\|_0^2 \right\} \quad (9.58)$$

Análise de Ψ

O objetivo é estimar cada termo de Ψ definido em (9.56), ou seja, mostrar que todos os termos são independentes de t .

A função g^n , definida em (9.44) é dada por $g^n = \delta^2 u^n - u_{tt}^{*n}$. Usando a definição do peso $*$ temos que

$$\delta^2 u^n - u_{tt}^{*n} = \delta^2 u^n - (\theta u_{tt}^{n+1} + (1-2\theta)u_{tt}^n + \theta u_{tt}^{n-1}) = \left(\delta^2 u^n - u_{tt}^n \right) - \theta \left(u_{tt}^{n+1} - 2u_{tt}^n + u_{tt}^{n-1} \right)$$

Mas, integrando por partes quatro vezes obtemos que:

$$\frac{1}{(\Delta t)^2} \left\{ \int_{t_n}^{t_{n+1}} \frac{(t_{n+1} - s)^3}{3!} u_{tttt}(s) ds + \int_{t_n}^{t_{n-1}} \frac{(t_{n-1} - s)^3}{3!} u_{tttt}(s) ds \right\} = \delta^2 u^n - u_{tt}^n$$

De forma análoga, integrando por partes duas vezes obtém-se que

$$\begin{aligned} \theta \left\{ \int_{t_n}^{t_{n+1}} (t_{n+1} - s) u_{tttt}(s) ds + \int_{t_n}^{t_{n-1}} (t_{n-1} - s) u_{tttt}(s) ds \right\} &= \theta \left(u_{tt}^{n+1} - 2u_{tt}^n + u_{tt}^{n-1} \right) \\ &= \theta (\Delta t)^2 \delta^2 u_{tt} \end{aligned}$$

Substituindo na função g , tem-se

$$\begin{aligned} g^n &= \frac{1}{(\Delta t)^2} \left\{ \int_{t_n}^{t_{n+1}} \frac{(t_{n+1} - s)^3}{3!} u_{tttt}(s) ds + \int_{t_n}^{t_{n-1}} \frac{(t_{n-1} - s)^3}{3!} u_{tttt}(s) ds \right\} \\ &\quad - \theta \left\{ \int_{t_n}^{t_{n+1}} (t_{n+1} - s) u_{tttt}(s) ds + \int_{t_n}^{t_{n-1}} (t_{n-1} - s) u_{tttt}(s) ds \right\} \end{aligned}$$

Tomando a norma $L^2(0, 1)$ em ambos os lados,

$$\begin{aligned} \|g^n\|_0 &\leq \frac{1}{(\Delta t)^2} \left\{ \int_{t_n}^{t_{n+1}} \left\| \frac{(t_{n+1} - s)^3}{3!} u_{tttt}(s) \right\|_0 ds + \int_{t_n}^{t_{n-1}} \left\| \frac{(t_{n-1} - s)^3}{3!} u_{tttt}(s) \right\|_0 ds \right\} \\ &\quad + \theta \left\{ \int_{t_n}^{t_{n+1}} \|(t_{n+1} - s) u_{tttt}(s)\|_0 ds + \int_{t_n}^{t_{n-1}} \|(t_{n-1} - s) u_{tttt}(s)\|_0 ds \right\} \end{aligned}$$

Mas por hipótese $u_{tttt} \in L^\infty(0, T; L^2(0, 1))$, então

$$\begin{aligned} \|g^n\|_0 &\leq \frac{1}{(\Delta t)^2} \left\{ \|u_{tttt}\|_{L^\infty(t_n, t_{n+1}; L^2(0, 1))} \int_{t_n}^{t_{n+1}} \left\| \frac{(t_{n+1} - s)^3}{3!} \right\|_0 ds \right. \\ &\quad \left. + \|u_{tttt}\|_{L^\infty(t_{n-1}, t_n; L^2(0, 1))} \int_{t_n}^{t_{n-1}} \left\| \frac{(t_{n-1} - s)^3}{3!} \right\|_0 ds \right\} \\ &\quad + \theta \left\{ \|u_{tttt}\|_{L^\infty(t_n, t_{n+1}; L^2(0, 1))} \int_{t_n}^{t_{n+1}} \|(t_{n+1} - s)\|_0 ds \right. \\ &\quad \left. + \|u_{tttt}\|_{L^\infty(t_{n-1}, t_n; L^2(0, 1))} \int_{t_n}^{t_{n-1}} \|(t_{n-1} - s)\|_0 ds \right\} \end{aligned}$$

Como $|t_n - s| \leq \Delta t \leq 1$, para $s \in [t_n, t_{n+1}]$ então obtemos que;

$$\|g^n\|_0 \leq (c + \theta)(\Delta t)^2 \left(\|u_{tttt}\|_{L^\infty(t_{n-1}, t_n; L^2(0,1))} + \|u_{tttt}\|_{L^\infty(t_n, t_{n+1}; L^2(0,1))} \right) \quad (9.59)$$

Multiplicando por Δt e fazendo a soma para $n = 1, \dots, (N-1)$ obtemos

$$\|g\|_{\tilde{L}^2(0,T;L^2)}^2 = \Delta t \sum_{n=1}^{N-1} \|g^n\|_0^2 \leq c_1 \Delta t^4 \|u_{tttt}\|_{L^\infty(0,T;L^2(0,1))}^2, \quad (9.60)$$

onde $c_1 = c + \theta$. Note que,

$$\|u_{tttt}\|_{L^\infty(0,T;L^2(0,1))}^2 = \Delta t \sum_{n=1}^{N-1} \|u_{tttt}\|_{L^\infty(t_n, t_{n+1}; L^2(0,1))}^2.$$

Do Lema(9.1), os dados iniciais em Ψ^2 satisfazem a seguinte estimativa:

$$\|\delta\xi^{1/2}\|_0 + \|\xi^1\|_1 + \|\xi^0\|_1 \leq c(h^2 + (\Delta t)^2) \quad (9.61)$$

Temos também para $m = 0$ e $k = 1$ em (8.8), que

$$|\rho''|_0 \leq \hat{c}_2 h^2 \|u_{tt}\|_2. \quad (9.62)$$

Substituindo as estimativas (9.60), (9.61) e (9.62), em (9.56) obtemos a estimativa:

$$\begin{aligned} 0 \leq \Psi^2 &= \|\rho''\|_{\tilde{L}^2(0,T;L^2)}^2 + \|g\|_{\tilde{L}^2(0,T;L^2)}^2 + \|\delta\xi^{1/2}\|_0^2 + \|\xi^1\|_1^2 + \|\xi^0\|_1^2 \\ &\leq \hat{C} \left\{ h^2 \|u_{tt}\|_{L^2(0,T;L^2(\Omega))}^2 + (h + (\Delta t)^2)^2 + \Delta t^4 \|u_{tttt}\|_{L^\infty(0,T;L^2(\Omega))}^2 \right\} \end{aligned} \quad (9.63)$$

Retornando a (9.58),

$$\begin{aligned} \|\delta\xi^{N-1/2}\|_0^2 + \|\xi^N\|_1^2 + \|\xi^{N-1}\|_1^2 &\leq C \left\{ h^2 \|u_{tt}\|_{L^2(0,T;L^2(\Omega))}^2 + (h + (\Delta t)^2)^2 \right. \\ &\quad \left. + \Delta t^4 \|u_{tttt}\|_{L^\infty(0,T;L^2(\Omega))}^2 + \Delta t \sum_{n=1}^{N-1} \|\delta\xi^{n-1/2}\|_0^2 \right\} \end{aligned} \quad (9.64)$$

Todos os termos do lado direito de (9.64) são limitados e independentes do tempo t , então podemos aplicar o Lema(8.2), análogo discreto do Lema de Gronwall para concluir que os termos do lado esquerdo de (9.64) são também limitados. Usando a equivalência de normas em espaço de dimensão finita podemos extrair a raiz quadrada dos termos para obter,

$$\|\delta\xi^{N-1/2}\|_0 + \|\xi^N\|_1 + \|\xi^{N-1}\|_1 \leq C_1 (h + (\Delta t)^2). \quad (9.65)$$

Substituindo os resultados obtidos em (9.56), (9.60) e (9.63), obtemos

$$\max_{1 \leq N \leq T} \left\{ \|\delta \xi^{N-1/2}\|_0 + \|\xi^N\|_1 + \|\xi^{N-1}\|_1 \right\} \leq \hat{C} (h + (\Delta t)^2). \quad (9.66)$$

Usando a definição de normas obtemos,

$$\|\xi\|_{\hat{L}^\infty(0,T;H_0^1(\Omega))} + \|\delta \xi\|_{\tilde{L}^\infty(0,T;L^2(\Omega))} \leq \hat{C}(h + (\Delta t)^2).$$

Mas pela decomposição do erro $e = \rho + \xi$. Usando a desigualdade triangular,

$$\begin{aligned} \|e\|_{\hat{L}^\infty(0,T;H_0^1(\Omega))} + \|\delta e\|_{\tilde{L}^\infty(0,T;L^2(\Omega))} &\leq C(h + (\Delta t)^2) + \|\rho\|_{\hat{L}^\infty(0,T;H_0^1(\Omega))} \\ &\quad + \|\delta \rho\|_{\tilde{L}^\infty(0,T;L^2(\Omega))} \leq C(h + (\Delta t)^2), \end{aligned}$$

pois $\|\rho\|_{\hat{L}^\infty(0,T;H_0^1(\Omega))} \leq ch$ e $\|\delta \rho\|_{\tilde{L}^\infty(0,T;L^2(\Omega))} \leq ch^2$, ou seja

$$\|u - u_h\|_{\hat{L}^\infty(0,T;H_0^1(\Omega))} + \|\delta u - u_h\|_{\tilde{L}^\infty(0,T;L^2(\Omega))} \leq C(h + (\Delta t)^2)$$

Vamos agora considerar, o caso de $\theta = 1/4$, para concluir a demonstração de (9.40)₁.

Caso (ii): $\theta = 1/4$

De fato: Considere a formulação (9.47), e como anteriormente tomamos

$$v_h = \frac{1}{2} \left(\delta \xi^{n+\frac{1}{2}} + \delta \xi^{n-1/2} \right) = \frac{1}{2\Delta t} \left(\xi^{n+1} - \xi^{n-1} \right).$$

Então para o primeiro termo permanece o mesmo resultado de (9.48).

Para o segundo termo, após multiplicarmos por $2\Delta t$ obtemos:

$$a(\xi^{*n}, \xi^{n+1} - \xi^{n-1}) = \left(\xi^{n+1/2} + \xi^{n-1/2}, \xi^{n+1/2} - \xi^{n-1/2} \right) = \|\xi^{n+1/2}\|_a^2 - \|\xi^{n-1/2}\|_a^2 \quad (9.67)$$

Multiplicando (9.47) por $2\Delta t$, substituindo (9.67), (9.48) e somando $n = 1, 2, \dots, N-1$ obtemos

$$\|\delta \xi^{N-1/2}\|_0^2 + \|\xi^{N-1/2}\|_a^2 = \|\delta \xi^{1/2}\|_0^2 + \|\xi^{1/2}\|_a^2 + \Delta t \sum_{n=1}^{N-1} \left(g^n - \delta^2 \rho^n, \delta \xi^{n+\frac{1}{2}} + \delta \xi^{n-1/2} \right)_0 \quad (9.68)$$

Mas de (9.53) e (9.54) tem-se

$$\begin{aligned} \Delta t \sum_{n=1}^{N-1} \left(g^n, \delta \xi^{n+\frac{1}{2}} + \delta \xi^{n-1/2} \right)_0 &\leq \frac{1}{\varepsilon} \|g\|_{\tilde{L}^2(0,T;L^2)}^2 + \Delta t \varepsilon \sum_{n=1}^{N-1} \|\delta \xi^{n-1/2}\|_0^2 \\ &\quad + \frac{\Delta t \varepsilon}{2} \|\delta \xi^{N-1/2}\|_0^2 \end{aligned}$$

e

$$\begin{aligned}
-\Delta t \sum_{n=1}^{N-1} (\delta^2 \rho^n, \delta \xi^{n+\frac{1}{2}} + \delta \xi^{n-1/2}) &\leq \frac{1}{\varepsilon} \|\rho''\|_{\tilde{L}^2(0,T;L^2)}^2 + \Delta t \varepsilon \sum_{n=1}^{N-1} \|\delta \xi^{n-1/2}\|_0^2 \\
&\quad + \frac{\Delta t \varepsilon}{2} \|\delta \xi^{N-1/2}\|_0^2
\end{aligned}$$

Substituindo os dois últimos termos em (9.68) e usando a equivalência de normas $\|\cdot\|_1$ e $\|\cdot\|_a$, obtemos a seguinte limitação:

$$\begin{aligned}
(1 - \Delta t \varepsilon) \|\delta \xi^{N-1/2}\|_0^2 + \|\xi^{N-1/2}\|_1^2 &\leq \|\delta \xi^{1/2}\|_0^2 + \|\xi^{1/2}\|_1^2 + \frac{1}{\varepsilon} \|g\|_{\tilde{L}^2(0,T;L^2)}^2 \\
&\quad + \frac{1}{\varepsilon} \|\rho''\|_{\tilde{L}^2(0,T;L^2)}^2 + 2\Delta t \varepsilon \sum_{n=1}^{N-1} \|\delta \xi^{n-1/2}\|_0^2
\end{aligned}$$

Tomando $\varepsilon < 1/\Delta t$, denotando $\hat{c} = (1 - \Delta t \varepsilon) > 0$ e desprezando o termo positivo $\|\xi^{N-1/2}\|_1^2 \geq 0$, obtemos

$$\|\delta \xi^{N-1/2}\|_0^2 \leq c_1 \left(\|\delta \xi^{1/2}\|_0^2 + \|\xi^{1/2}\|_1^2 + \|g\|_{\tilde{L}^2(0,T;L^2)}^2 + \|\rho''\|_{\tilde{L}^2(0,T;L^2)}^2 + \Delta t \sum_{n=1}^{N-1} \|\delta \xi^{n-1/2}\|_0^2 \right) \quad (9.69)$$

onde $c_1 = \max\{\hat{c}, 1/\varepsilon\}$. Todos os termos do lado direito de (9.69) são limitados e independentes do tempo t , então podemos aplicar o Lema(8.2), análogo discreto do Lema de Gronwall, para obter a seguinte limitação:

$$\|\delta \xi^{N-1/2}\|_0^2 \leq c_1 \left(\|\delta \xi^{1/2}\|_0^2 + \|\xi^{1/2}\|_1^2 + \|g\|_{\tilde{L}^2(0,T;L^2)}^2 + \|\rho''\|_{\tilde{L}^2(0,T;L^2)}^2 \right)$$

Mas usando (9.60), (9.66) e (9.62) obtemos a estimativa

$$\|\delta \xi^{N-1/2}\|_0^2 \leq c_2 (h + (\Delta t)^2).$$

Da definição de norma,

$$\|\delta \xi\|_{\tilde{L}^\infty(0,T;L^2(\Omega))} \leq c_1 (h + (\Delta t)^2).$$

Assim usando a decomposição do erro $e = \rho + \xi$ e a desigualdade triangular,

$$\begin{aligned}
\|\delta e\|_{\tilde{L}^\infty(0,T;L^2(\Omega))} &\leq \|\delta \xi\|_{\tilde{L}^\infty(0,T;L^2(\Omega))} + \|\delta \rho\|_{\tilde{L}^\infty(0,T;L^2(\Omega))} \\
&\leq c(h + (\Delta t)^2 + h^2) \leq \hat{c}(h + (\Delta t)^2),
\end{aligned}$$

Para concluir o teorema resta provar a estimativa na norma $L^2(\Omega)$ dada em (9.40).

Estimativa de Erro: Norma $L^2(\Omega)$

A demonstração segue o mesmo raciocínio da estimativa anterior. Mostra-se que

$$\begin{aligned} \|\delta\xi^{N-1/2}\|_0 + \|\xi^N\|_0 + \|\xi^{N-1}\|_0 &\leq c \left\{ \|\delta\xi^{1/2}\|_0 + \|\xi^1\|_0 + \|\xi^0\|_0 \right. \\ &\quad \left. + \|\rho''\|_{\hat{L}^2(0,T;L^2(\Omega))} + \|g\|_{\hat{L}^2(0,T;L^2(\Omega))} \right\} \end{aligned} \quad (9.70)$$

Mas para os dados iniciais temos de (9.31), que

$$\|\delta\xi^{1/2}\|_0 + \|\xi^1\|_0 + \|\xi^0\|_0 \leq \tilde{c} \left(h^2 + (\Delta t)^2 \right)$$

Temos também de (9.60) e (9.62) que

$$\|\rho''\|_{\hat{L}^\infty(0,T;L^2(\Omega))} \leq c_1 h^2 \quad \text{e} \quad \|g\|_{\hat{L}^2(0,T;L^2(\Omega))} \leq c h^2$$

Assim de forma similar à estimativa anterior obtemos que,

$$\|e\|_{\hat{L}^\infty(0,T;L^2(\Omega))} + \|\delta e\|_{\hat{L}^\infty(0,T;L^2(\Omega))} \leq c_2 (h^2 + (\Delta t)^2) \quad (9.71)$$

onde c_2 é uma constante positiva independente da malha h e da discretização Δt . \square

No próximo teorema, vamos estimar o erro da solução do θ -método para o problema de segunda ordem, dado pelo sistema (7.23).

Teorema 9.4. *Sob a hipótese (9.39) e os dados iniciais $u_0 \in H_0^1(0,1) \cap H^2(0,1)$ e $u_1 \in H_0^1(0,1)$ então a solução aproximada $\{w_h, u_h\}$ do Problema (7.19) obtida pelo θ -método (7.23), para $(\sigma, \theta) \geq 1/2$, tem a seguinte estimativa:*

$$\|w - w_h\|_{L^\infty(0,T;L^2(0,1))} + \|u - u_h\|_{L^\infty(0,T;H_0^1(0,1))} \leq C \left(h + (\Delta t)^m \right). \quad (9.72)$$

onde C é uma constante positiva, independente de h e Δt .

Além disso, se $\sigma = \theta = \frac{1}{2}$ então $m = 2$ e para $(\sigma, \theta) \geq 1/2$ temos $m = 1$.

Demonstração:

A formulação fraca do problema é: $(u'', v) + a(u, v) = (f, v)$.

Fazendo a mudança de variável $w = u'$ então temos os dois problemas

$$(w', v) + a(u, v) = (f, v) \quad \text{e} \quad (u', \psi) - (w, \psi) = 0$$

Fazendo $t = (n+1)$ e $t = n$ com pesos θ e $(1-\theta)$ e σ e $(1-\sigma)$ respectivamente na primeira e segunda equação e somando os termos, obtemos

$$(w_t^{n+\theta}, v) + a(u^{n+\theta}, v) = (f^{n+\theta}, v) \quad \text{e} \quad (u_t^{n+\sigma}, \varphi) - (w^{n+\sigma}, \varphi) = 0 \quad (9.73)$$

Somando e subtraindo $\delta w^{n+\frac{1}{2}}$ e $\delta u^{n+\frac{1}{2}}$ na primeira e segunda equação do primeiro membro (9.73), obtemos em particular para $\varphi_h, v_h \in V_m$

$$(\delta w^{n+\frac{1}{2}}, v) + a(u^{n+\theta}, v) = (f^{n+\theta} + g^n, v) \quad \text{e} \quad (\delta u^{n+\frac{1}{2}}, \varphi) - (w^{n+\sigma}, \varphi) = (\lambda^n, \varphi) \quad (9.74)$$

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

$$(\delta w_h^{n+\frac{1}{2}}, v_h) + a(u_h^{n+\theta}, v_h) = (f^{n+\theta}, v_h)_0, \quad \text{e} \quad (\delta u_h^{n+\frac{1}{2}}, \varphi_h) - (w_h^{n+\sigma}, \varphi_h) \quad (9.75)$$

Tomando em particular $v = v_h$ e $\varphi = \varphi_h$ e fazendo a respectiva diferença obtemos

$$(\delta e_1^{n+\frac{1}{2}}, v_h) + a(e_2^{n+\theta}, v_h) = (g^n, v_h) \quad \text{e} \quad (\delta e_2^{n+\frac{1}{2}}, \varphi_h) - (e_1^{n+\sigma}, \varphi_h) = (\lambda^n, \varphi_h) \quad (9.76)$$

onde $e_1 = w - w_h$ e $e_2 = u - u_h$.

Considere em particular $v_h = (e_2^{n+1} - e_2^n)$ e $\varphi_h = (e_1^{n+1} - e_1^n)$. Substituindo nas equações e fazendo a diferença e cancelando o primeiro elemento de cada equação, obtemos:

$$a(e_2^{n+\theta}, e_2^{n+1} - e_2^n) + (e_1^{n+\sigma}, e_1^{n+1} - e_1^n) = (g^n, e_2^{n+1} - e_2^n) - (\lambda^n, e_1^{n+1} - e_1^n) \quad (9.77)$$

Usando a igualdade, $\chi^{n+\theta} = \chi^{n+1/2} + (\theta - \frac{1}{2})(\chi^{n+1} - \chi^n)$, substituindo em (9.77), e usando a desigualdade elementar, obtemos;

$$\begin{aligned} & \|e_2^{n+1}\|_a^2 - \|e_2^n\|_a^2 + 2(\theta - 1/2)\|e_2^{n+1} - e_2^n\|_a^2 + \\ & + \|e_1^{n+1}\|^2 - \|e_1^n\|^2 + 2(\sigma - 1/2)\|e_1^{n+1} - e_1^n\|^2 \\ & \leq \frac{1}{\gamma_2}\|g^n\|^2 + \frac{1}{\gamma_1}\|\lambda^n\|^2 + \gamma_2\|e_2^{n+1} - e_2^n\|_a^2 + \gamma_1\|e_1^{n+1} - e_1^n\|^2 \end{aligned} \quad (9.78)$$

Para $\theta, \sigma \in [1/2; 1]$, podemos tomar $\gamma_1 = (\sigma - 1/2) \geq 0$ e $\gamma_2 = (\theta - 1/2) \geq 0$. Logo

$$\begin{aligned} & (\|e_2^{n+1}\|_a^2 - \|e_2^n\|_a^2) + (\|e_1^{n+1}\|^2 - \|e_1^n\|^2) + \gamma_2\|e_2^{n+1} - e_2^n\|_a^2 + \gamma_1\|e_1^{n+1} - e_1^n\|^2 \\ & \leq \frac{1}{\gamma_2}\|g^n\|^2 + \frac{1}{\gamma_1}\|\lambda^n\|^2 \end{aligned}$$

Como $\gamma_2\|e_2^{n+1} - e_2^n\|_a^2 \geq 0$ e $\gamma_1\|e_1^{n+1} - e_1^n\|^2 \geq 0$, então

$$\|e_2^{n+1}\|_a^2 - \|e_2^n\|_a^2 + \|e_1^{n+1}\|^2 - \|e_1^n\|^2 \leq \frac{1}{\gamma_2}\|g^n\|^2 + \frac{1}{\gamma_1}\|\lambda^n\|^2 \quad (9.79)$$

Fazendo a soma $n = 0, 1, \dots, N-1$, obtemos

$$\|e_2^N\|_a^2 + \|e_1^N\|^2 \leq \|e_2^0\|_a^2 + \|e_1^0\|^2 + \frac{1}{\gamma_2} \sum_{n=0}^{N-1} \|g^n\|^2 + \frac{1}{\gamma_1} \sum_{n=0}^{N-1} \|\lambda^n\|^2 \quad (9.80)$$

De (8.101)

$$\begin{aligned} \|g\|_{\hat{L}^2(0,T;L^2(\Omega))}^2 &= \Delta t \sum_{n=1}^{N-1} \|g^n\|_0^2 \\ &\leq c_a (\Delta t)^4 \|w'''\|_{L^2(0,T;H^2)}^2 + 2\left(\theta - \frac{1}{2}\right)^2 (\Delta t)^2 \|w''\|_{L^2(0,T;H^2)}^2 \end{aligned} \quad (9.81)$$

$$\begin{aligned} \|\lambda\|_{\hat{L}^2(0,T;H_0^1(\Omega))}^2 &= \Delta t \sum_{n=1}^{N-1} \|\lambda^n\|_1^2 \\ &\leq c_b (\Delta t)^4 \|u'''\|_{L^2(0,T;H^2)}^2 + 2\left(\sigma - \frac{1}{2}\right)^2 (\Delta t)^2 \|u''\|_{L^2(0,T;H^2)}^2 \end{aligned} \quad (9.82)$$

Para os dados iniciais, temos pelo menos, $\|e_2^0\|_a^2 + \|e_1^0\|_0^2 \leq c_3 h^2$.

Usando a equivalência de normas em dimensão finita e extraindo a raiz quadrada dos termos, supondo a regularidade da solução e suas derivadas, obtemos que;

$$\begin{aligned} \|e_2^N\|_1 + \|e_1^N\|_0 &\leq c_4 \left(h + (\Delta t)^2 \|w'''\|_{L^2(0,T;H^2)} + \left(\theta - \frac{1}{2}\right) \Delta t \|w''\|_{L^2(0,T;H^2)} \right. \\ &\quad \left. + (\Delta t)^2 \|u'''\|_{L^2(0,T;H^2)} + \left(\sigma - \frac{1}{2}\right) \Delta t \|u''\|_{L^2(0,T;H^2)} \right) \end{aligned} \quad (9.83)$$

Como os termos da direita são independentes de N , então

$$\|e_2\|_{L^\infty(0,T;H_0^1(0,1))} + \|e_1\|_{L^\infty(0,T;L^2(0,1))} \leq c_5 \left\{ h + \left(\left(\sigma - \frac{1}{2}\right) + \left(\theta - \frac{1}{2}\right) \right) \Delta t + (\Delta t)^2 \right\}. \quad (9.84)$$

ou seja o erro é de ordem $\mathcal{O}(h, \Delta t)$.

Note que se $\sigma = \theta = 1/2$, então de (9.84) obtemos,

$$\|e_2\|_{L^\infty(0,T;H_0^1(0,1))} + \|e_1\|_{L^\infty(0,T;L^2(0,1))} \leq c_5 (h + (\Delta t)^2). \quad \square \quad (9.85)$$

9.1.3 Exercícios

1. Prove que o método da diferença central ($\theta = 0$) em (9.29) é condicionalmente estável, ou seja, $\Delta t/h \leq \delta$ com δ suficientemente pequeno.
2. Prove com detalhes, a estimativa de erro (9.71) na norma $L^2(\Omega)$ para o problema totalmente discreto.

9.2 Existência e unicidade de solução

Teorema 9.5. *Dados $f \in L^2(0, T; L^2(0, 1))$, $u_0(x) \in H_0^1(0, 1)$ e $u_1(x) \in L^2(0, 1)$, então existe uma única solução $u : Q = [0, T] \times (0, 1) \rightarrow \mathbb{R}$ para o problema (6.1) satisfazendo a seguinte condição:*

$$(i) \quad u \in L^\infty(0, T; H_0^1(0, 1))$$

$$(ii) \quad u' \in L^\infty(0, T; L^2(0, 1))$$

$$(iii) \quad u'' \in L^2(0, T; H^{-1}(0, 1))$$

$$(iv) \quad \frac{d}{dt}(u'(t), v) + a(u(t), v) = (f, v), \quad \forall v \in H_0^1(0, 1).^*$$

$$(v) \quad u(x, 0) = u_0(x), \quad u'(x, 0) = u_1(x), \quad \forall x \in (0, 1).$$

* {A igualdade deve ser entendida no sentido das distribuições em $(0, T)$, representada por $\mathcal{D}'(0, T)$ }.

Demonstração:

Antes de iniciar a demonstração, vamos verificar se faz sentido os dados iniciais $u(x, 0) = u_0(x) \in H_0^1(0, 1)$ e $u'(x, 0) = u_1(x) \in L^2(0, 1)$. Com efeito, seja $u(x, t)$ satisfazendo as condições do teorema. Usando o (1.5.2) a as condições (i) e (ii), obtemos que $u \in C^0([0, T]; L^2(0, 1))$ e $u \in C_s([0, T]; H_0^1(0, 1))$ e de (ii) e (iii) obtemos que $u' \in C^0([0, T]; H^{-1}(0, 1))$ e $u' \in C_s([0, T]; H_0^1(0, 1))$, e assim faz sentido calcular $u(x, 0)$ e $u'(x, 0)$.

Sistema aproximado: Seja $T > 0$ e $\{w_i, i = 1, \dots, m\}$ uma base do espaço

de Hilbert $H_0^1(0, 1)$, que é um espaço separável. Considere $V_m = [w_1, w_2, \dots, w_m]$, um subespaço vetorial gerado pelos m-primeiros elementos da base w_i . Então qualquer função $u_m(t) \in V_m$ tem a seguinte representação

$$u_m(t) = \sum_{i=1}^m g_{im}(t) w_i(x). \quad (9.86)$$

Considere o problema (7.3), redefinido no subespaço V_m , e então deseja-se determinar

uma solução $u_m : [0, T] \rightarrow V_m$, solução do seguinte sistema,

$$\begin{cases} (u_m''(t), v) + a(u_m(t), v) = (f(t), v), \quad \forall v \in V_m, \\ u_m(0) = u_{0m} \rightarrow u_0 \text{ forte em } H_0^1(0, 1) \\ u_m'(0) = u_{1m} \rightarrow u_1 \text{ forte em } L^2(0, 1) \end{cases} \quad (9.87)$$

Pelo Teorema de Caratheodory, o sistema de equações diferenciais ordinárias possui uma solução local $u_m(t)$, $\forall t \in [0, t_m[$, $0 \leq t_m < T$. Para prolongar a solução ao intervalo $[0, T]$, são necessárias algumas estimativas, que se seguem.

Estimativa 1: Seja $v \in u_m'(t) \in V_m$ em $(9.87)_1$. Então

$$(u_m''(t), u_m'(t)) + a(u_m(t), u_m'(t)) = (f(t), u_m'(t)).$$

Mas

$$\frac{1}{2} \frac{d}{dt} |u_m'(t)|^2 = (u_m''(t), u_m'(t)),$$

e

$$\frac{1}{2} \frac{d}{dt} \|u_m(t)\|_a^2 = a(u_m(t), u_m'(t)),$$

usando a desigualdade de Schwarz e elementar temos

$$(f(t), u_m'(t)) \leq \frac{1}{2} (|f(t)|^2 + |u_m'(t)|^2)$$

Substituindo termo a termo e multiplicando por 2 obtemos;

$$\frac{d}{dt} |u_m'(t)|^2 + \frac{d}{dt} \|u_m(t)\|_a^2 \leq |f(t)|^2 + |u_m'(t)|^2$$

Integrando a inequação de 0 a $t < T$, obtemos

$$|u_m'(t)|^2 + \|u_m(t)\|_a^2 \leq |u_m'(0)|^2 + \|u_m(0)\|_a^2 + \int_0^t |f(s)|^2 ds + \int_0^t |u_m'(s)|^2 ds$$

Por hipótese

1. $u_m'(0) = u_{1m}$ converge forte para u_1 em $L^2(0, 1)$, o que significa $\lim_{m \rightarrow \infty} |u_{1m} - u_1| = 0$
2. $u_m(0) = u_{0m}$ converge forte para u_0 em $H_0^1(0, 1)$, então, $\lim_{m \rightarrow \infty} \|u_{0m} - u_0\| = 0$.
3. $f \in L^2(0, T; L^2(0, 1))$ então $\int_0^t |f(s)|^2 ds \leq \int_0^T |f(s)|^2 ds \leq c$.

Assim podemos escrever que

$$|u'_m(t)|^2 + \|u_m(t)\|_a^2 \leq c_1 \int_0^t |u'_m(s)|^2 ds \leq c_1 \int_0^T |u'_m(s)|^2 ds \quad (9.88)$$

Logo da desigualdade de Gronwall segue que:

$$|u'_m(t)| \leq c_2, \text{ independente de } m \text{ e } t$$

ou seja

$$(u'_m) \text{ é limitada no espaço } L^\infty(0, T; L^2(0, 1)) \subset L^2(0, T; L^2(0, 1)). \quad (9.89)$$

Da limitação de u'_m em (9.88), conclui-se também que

$$\min\{\alpha, \beta\}(\|u_m\| + |u_m|) \leq \|u_m\|_a = \alpha\|u_m\| + \beta|u_m| \leq c.$$

Logo,

$$(u_m) \text{ é limitada em } L^\infty(0, T; H_0^1(0, 1)) \subset L^2(0, T; H_0^1(0, 1)), \quad (9.90)$$

independente de m e t . As estimativas (9.89) e (9.90) são suficientes para prolongar a solução u_m ao intervalo $[0, T]$.

Passagem do Limite

De (9.89) e (9.90), sendo os espaços $L^2(0, T; L^2(0, 1))$ e $L^2(0, T; H_0^1(0, 1))$ completos, existem subsequências de $(u_m)_{m \in N}$, também denotadas por (u_m) tais que:

1. u_m converge fraco para u em $L^2(0, T; H_0^1(0, 1))$, denota-se, $u_m \rightharpoonup u$.
2. u'_m converge fraco para u' em $L^2(0, T; L^2(0, 1))$, denota-se, $u'_m \rightharpoonup u'$.

Observação: Deve-se tomar cuidado com a segunda afirmação, na verdade precisamos provar que realmente $u'_m \rightharpoonup u'$. Com efeito suponha que $u'_m \rightharpoonup \chi$ em $L^2(0, T; L^2(0, 1))$. Da primeira convergência temos que $u_m \rightharpoonup u$ em $L^2(0, T; H_0^1(0, 1)) \subset L^2(0, T; L^2(0, 1))$.

Então $u_m \rightharpoonup u$ em $\mathcal{D}'(0, T; L^2(0, 1))$. Por definição de distribuição, temos que $\frac{du_m}{dt} = u'_m \rightarrow \frac{du}{dt} = u'$ em $\mathcal{D}'(0, T; L^2(0, 1))$. Por outro lado $u'_m \rightharpoonup \chi$ em $L^2(0, T; L^2(0, 1)) \subset \mathcal{D}'(0, T; L^2(0, 1))$. Assim da unicidade do limite conclui-se que $\chi = u'$. \square

Mostremos agora, que o sistema aproximado (9.87) converge para

$$\frac{d}{dt}(u'(t), v) + a(u(t), v) = (f, v), \quad \forall v \in H_0^1(0, 1) \quad \text{em } \mathcal{D}'(0, T).$$

Com efeito, usando a definição da convergência fraca em (9.90), tem-se que

$$\int_0^T (u'_m(t), w) dt \rightarrow \int_0^T (u'(t), w) dt \quad \forall w \in L^2(0, T; L^2(0, 1))$$

Tomando em particular $w(x, t) = \theta(t)v(x)$, com $\theta \in \mathcal{D}(0, T) \subset L^2(0, T)$ e $v \in H_0^1(0, 1) \subset L^2(0, 1)$, e substituindo temos

$$\int_0^T (u'_m(t), v)\theta dt \rightarrow \int_0^T (u'(t), v)\theta dt \quad \forall v \in L^2(0, 1) \quad \text{e} \quad \forall \theta \in \mathcal{D}(0, T).$$

De onde se conclui que

$$(u'_m(t), v) \rightarrow (u'(t), v), \text{ quando } m \rightarrow \infty \text{ em } \mathcal{D}'(0, T), \forall v \in L^2(0, 1). \quad (9.91)$$

Logo

$$\frac{d}{dt}(u'_m(t), v) \rightarrow \frac{d}{dt}(u'(t), v), \text{ quando } m \rightarrow \infty \text{ em } \mathcal{D}'(0, T), \forall v \in L^2(0, 1). \quad (9.92)$$

De forma análoga, em (9.89), tomando $w(x, t) = \theta(t)v(x)$, com $\theta \in \mathcal{D}(0, T) \subset L^2(0, T)$ e $v \in H_0^1(0, 1) \subset H^{-1}(0, 1)$, e substituindo obtemos

$$\int_0^T ((u_m(t), v))\theta dt \rightarrow \int_0^T ((u(t), v))\theta dt \quad \forall v \in H_0^1(0, 1) \quad \text{e} \quad \forall \theta \in \mathcal{D}(0, T).$$

Ou seja

$$\int_0^T a(u_m(t), v)\theta dt \rightarrow \int_0^T a(u(t), v)\theta dt \quad \forall v \in H_0^1(0, 1) \quad \text{e} \quad \forall \theta \in \mathcal{D}(0, T).$$

Portanto

$$a(u_m(t), v) \rightarrow a(u(t), v), \text{ quando } m \rightarrow \infty \text{ em } \mathcal{D}'(0, T), \forall v \in H_0^1(0, 1). \quad (9.93)$$

De (9.92) e (9.93) temos a convergência procurada, ou seja,

$$\frac{d}{dt}(u'_m(t), v) + a(u_m(t), v) - (f, v) \rightarrow \frac{d}{dt}(u'(t), v) + a(u(t), v) - (f, v), \quad \forall v \in H_0^1(0, 1) \quad (9.94)$$

no sentido de $\mathcal{D}'(0, T)\square$.

9.3 Regularidade e Conservação de Energia

Para estabelecer estimativa de erro na norma $L^2(\Omega)$, é necessário mais regularidade para a solução do problema (6.1). Para isso anunciaremos dois teoremas de regularidade cuja demonstração será omitida:

Teorema 9.6. *Seja (Ω) um aberto e limitado do \mathbb{R}^n . Dados f e $\frac{\partial f}{\partial t} \in L^2(0, T; L^2(\Omega))$, $u_0(x) \in H_0^1(\Omega) \cap H^2(\Omega)$ e $u_1(x) \in H_0^1(\Omega)$, então existe uma única solução $u : Q = [0, T] \times \Omega \rightarrow \mathbb{R}$ para o problema (6.1) satisfazendo a seguinte condição:*

$$(i) \quad u \in L^\infty(0, T; H_0^1(\Omega) \cap H^2(\Omega))$$

$$(ii) \quad u' \in L^\infty(0, T; H_0^1(\Omega))$$

$$(iii) \quad u'' \in L^2(0, T; L^2(\Omega))$$

$$(iv) \quad u'' - \alpha \Delta u + \beta u = f, \quad \text{quase sempre em } Q = [0, T] \times \Omega$$

$$(v) \quad u(x, 0) = u_0(x), \quad u'(x, 0) = u_1(x), \quad \forall x \in \Omega.$$

Demonstração:(Idéia)

Para a demonstração, basta fazer as estimativas para a equação original como no Teorema(9.5) e depois para sua derivada em relação ao tempo, pois temos a hipótese de que $\frac{\partial f}{\partial t} \in L^2(0, T; L^2(\Omega))$. A seguir o resultado é obtido por teoremas de compacidade.

Teorema 9.7. *Seja (Ω) um aberto e limitado do \mathbb{R}^n . Dados $f \in L^2(0, T; H_0^1(\Omega))$, $u_0(x) \in H_0^1(\Omega) \cap H^2(\Omega)$ e $u_1(x) \in H_0^1(\Omega)$, então existe uma única solução $u : Q = [0, T] \times \Omega \rightarrow \mathbb{R}$ para o problema (6.1) satisfazendo a seguinte condição:*

$$(i) \quad u \in L^\infty(0, T; H_0^1(\Omega) \cap H^2(\Omega))$$

$$(ii) \quad u' \in L^\infty(0, T; H_0^1(\Omega))$$

$$(iii) \quad u'' \in L^2(0, T; L^2(\Omega))$$

$$(iv) \quad u'' - \alpha \Delta u + \beta u = f, \quad \text{quase sempre em } Q = [0, T] \times \Omega$$

$$(v) \quad u(x, 0) = u_0(x), \quad u'(x, 0) = u_1(x), \quad \forall x \in \Omega.$$

Demonstração: (Idéia)

Para a demonstração, a base escolhida para o subespaço V_m são os autovalores do operador Laplaciano, estimativas e resultados de compacidade.

A seguir vamos mostrar uma importante propriedade da equação da onda homogênea, ou seja, quando $f = 0$.

Conservação de Energia

Definimos a energia da equação da onda, como a soma da energia cinética e potencial por:

$$E(t) = \frac{1}{2} \left(\int_{\Omega} |u'|^2 + \alpha \int_{\Omega} |\nabla u|^2 + \beta \int_{\Omega} |u|^2 \right) \quad (9.95)$$

Nas mesmas condições do Teorema(9.5) então a equação da onda homogênea:

$$u'' - \Delta u + u = 0$$

conserva a energia, ou seja, $E(t) = E(0)$, $\forall t > 0$.

Para verificar a propriedade, multiplique a equação por u' , obtendo-se

$$\frac{1}{2} \frac{d}{dt} \left(|u'|^2 + \alpha |\nabla u|^2 + \beta |u|^2 \right) = 0$$

Integrando de 0 a t , obtemos

$$\frac{1}{2} \left(\int_{\Omega} |u(t)'|^2 + \alpha \int_{\Omega} |\nabla u(t)|^2 + \beta \int_{\Omega} |u(t)|^2 \right) - \frac{1}{2} \left(\int_{\Omega} |u_1|^2 + \alpha \int_{\Omega} |\nabla u_0|^2 + \beta \int_{\Omega} |u_0|^2 \right) = 0;$$

ou seja, da definição $E(t) = E(0)$, $\forall t > 0$.

APÊNDICE A

Programas computacionais: linguagem C

PEU.cpp	Problema Estacionário Unidimensional
typedef.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
Calortrapezoidal.cpp	Equação do Calor Unidimensional
MetodoNewmark.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,d1,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' direita:\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*d1); /* Norma L^2 */
    v[1] = sqrt(L2/4*d1+H1/d1); /* 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 – typedef.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(elemento) -> 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 "typedef.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_Decom(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 "typedef.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,d1;
    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]];
        d1 = 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)*d1;
        if (i2!=0)
            F[i2-1] += (Bv[n][j+1][s]/3+Bv[n][j][s]/6)*d1;
    }
}

/*
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

```

/*
  Problema Parabólico do Calor Unidimensional
  Metodo de Elementos finitos
   $U_t(x,t) - (\text{Alpha})U_{xx}(x,t) + (\text{Beta})U(x,t) = f(x,t)$ 
  tipo=1: u() prescrita
  tipo=2: du() prescrita
  Linear solver: Tri-diagonal matrix

  Para Alpha=Beta=1.0 então

  Problema 1:
  Solução Exata:  $u(x,t) = (1.0/\pi^2) \sin(\pi x) \exp^{-(\alpha \pi^2 + \beta)t}$ 
  para a força F=0

  Problema2:
  Solução Exata:  $u(x,t) = (1.0/\pi^2) \sin(\pi x) \exp^{-t}$ 
  para a força  $F = (\sin(\pi x) \exp(-t))$ 

  Programa também gera um arquivo de erros na norma  $L^2$  e  $H^1$ .
  Método generalizado trapezoidal, usando a diferença progressiva para  $g'(t)$ 
  e média ponderada nos termos dependentes de t e nao envolvem derivadas.
*/

#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 3.14159265358979
#define T 1.0

FILE *f1,*f2, *f3;
int i,j;
int maxj;
int Nel,Nos;

```

```

int tipo1,tipo2;
float g0[Nosm];           /*Solução inicial do problema dado */
float v0[Nosm];           /*Derivada inicial  $g'(0)$  do problema dado */
float g[Nosm];            /*Sol.aprox do sistema  $(A+Teta*k*B)g[n+1] = b$  */
float v[Nosm];            /*Derivada  $g'(tn)$  */
float gtil[Nosm];
float b[Nosm];            /*Vetor b do sistema  $(A+Teta*k*B)g[n+1] = b$  */
float c[Nosm];            /*Vetor c do sistema  $(A+Teta*k*B)v = c$  */
float G[Nosm];            /*Solução exata do problema dado */
float X[Nosm], Xe[3];
float H[Nelm];
float A[Nosm][4];         /*Matriz A do sistema  $(A+Teta*k*B)g[n+1] = b$  */
float B[Nosm][4];         /*Matriz B do sistema  $(A+Teta*k*B)g[n+1] = b$  */
float F[Nosm],F0[Nosm],F1[Nosm]; /*Vetor F (Força) */
float Alpha,Beta,Teta,lambda;
float A1,A2,d1,d[2],k, Max;
float EAbs[Nosm];         /*Erro absoluto */
float E[Nosm];            /*Erro da diferença sol. aprox. e exata */
float MaxH1;              /*Maximo do Erro para norma H1 */
float MaxLInf;            /*Maximo do Erro para norma L Infinito */

void InputData()
{
    printf("Constante Alpha = ");
    scanf("%f", &Alpha);
    printf("Constante Beta = ");
    scanf("%f", &Beta);
    printf("Constante Teta (Teta entre 0 e 1) = ");
    scanf("%f", &Teta);
    printf("\nConstante k = "); /* k = DeltaT = Passo no tempo */
    scanf("%f", &k);
    fprintf(f1,"Alpha = %f\n", Alpha);
    fprintf(f1,"Beta = %f\n", Beta);
    //lambda =(Alpha*Pi*Pi+Beta); //valor de lambda p/ Problema 2
    lambda =1; // calor de lambda para o problema 2
    fprintf(f1,"Teta = %f\n", Teta);
    fprintf(f1,"\nConstante k = %f\n", k);
    printf("\n\nNumeros de elementos = ");
    scanf("%d", &Nel);
    fprintf(f1,"\nNumero de elementos = %d\n", Nel);
    fprintf(f1,"\nSolução Exata =(1.0/\pi^2)*sin(Pi*x)*exp^(-(\pi^2+1)*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 = h, onde h é o passo no espaço */
    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. 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 (sin(Pi*x)*exp(-j*k)); /* funcao de forca dada*/
    //return 0;
}
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 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 Am[Nosm][4], float b[Nosm], float g[Nosm])
/*Resolução do Sistema Linear (A+kB)g = b pelo MEG*/
{ int i;
  float y;
  for (i=1; i<Nos; i++)
  {
    y = Am[i+1][1]/Am[i][2]; /*Cálculo do multiplicador*/
    Am[i+1][2] += -y*Am[i][3]; /*Transf. da matriz (A+kB) em Triang.Superior*/
    b[i+1] += -y*b[i]; /*Atualização do vetor b*/
  }
  /* Retro-Substituicao */
  g[Nos] = b[Nos]/Am[Nos][2];
  for (i=Nos-1; i>=1; i--)
    g[i] = (b[i]-Am[i][3]*g[i+1])/Am[i][2];
}

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*d1);          /* Norma em L2*/
    v[1] = sqrt(L2/4*d1)+sqrt(H1*d1); /*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 Am[Nosm][4];

    f1=fopen("ProgramaCalor_Trapezoidal.out", "w");
    f2=fopen("ProgramaCalor_Trapezoidal.dat", "w");
    f3=fopen("Erro.dat", "w");

    InputData();
    Nos = Nel + 1;
    CoordGlobal();
    InputCondFront();
    MatrizA();
    MatrizRigidez();
    CondFront(A1, A2);
    OutputData();
    maxj= int(T/k + 0.5);
    fprintf(f2, "

```

APROXIMADA

```

fprintf(f2, "(x =      )      ");
j=0;
VetorForca();
CondFront(A1,A2);

for(i=1; i<=Nos; ++i)
{
    fprintf(f2, "%3.4f      ", X[i]);
}
for(i=1; i<=Nos; ++i)
{
    fprintf(f2, "%3.4f      ", X[i]);
    /* condição inicial  $g(0) = 1/(Pi*Pi)*sin(Pi*x)$  */
    g0[i]=(1.0/(Pi*Pi))*sin(Pi*X[i]);
    F0[i]=F[i];
}
fprintf(f2, "\n");
fprintf(f1, "\n\nResultados:\n");
fprintf(f1, "\n  X[i]  Tempo Aproximada Exata  Erro Força Prescrita\n");
/*Montagem do vetor b para o calculo de  $v0 = g'(0)$ */
b[1]=F0[1]-(B[1][2]*g0[1]+B[1][3]*g0[2]);
for (i=2; i<=Nos-1; i++)
{
    b[i] = F0[i]-(B[i][1]*g0[i-1]+B[i][2]*g0[i]+B[i][3]*g0[i+1]);
}
b[Nos]=F0[Nos]-(B[Nos][1]*g0[Nos-1]+B[Nos][2]*g0[Nos]);

for(i=1; i<=Nos; i++)
{
    Am[i][1]=A[i][1];
    Am[i][2]=A[i][2];
    Am[i][3]=A[i][3];
}

LinearSolver(Am, b, v0);
MaxH1 = 0;
MaxLInf = 0;
for(j=0; j<=maxj; j++)
{
    //fprintf(f2, "\n(t=%1.4f)      ", j*k);
    fprintf(f2, "\n%1.4f      ", j*k);
}

```

```

j=j+1;
VetorForca();
CondFront(A1,A2);

for (i=1; i<=Nos; i++)
{
    gtil[i]=g0[i]+(1-Teta)*k*v0[i];
}
/*Montagem do vetor c para calculo da derivada v = g'(tn)*/
c[1]=F[1]-(B[1][2]*gtil[1]+B[1][3]*gtil[2]);
for (i=2; i<=Nos-1; i++)
{
c[i] = F[i]-(B[i][1]*gtil[i-1]+B[i][2]*gtil[i]+B[i][3]*gtil[i+1]);
}
c[Nos]=F[Nos]-(B[Nos][1]*gtil[Nos-1]+B[Nos][2]*gtil[Nos]);

for(i=1; i<=Nos; i++)
{
    Am[i][1]=A[i][1]+Teta*k*B[i][1];
    Am[i][2]=A[i][2]+Teta*k*B[i][2];
    Am[i][3]=A[i][3]+Teta*k*B[i][3];
}
LinearSolver(Am, c, v);
for (i=1; i<=Nos; i++)
{
    g[i]=gtil[i]+Teta*k*v[i];
    /*quando força nula f=0*/
    //G[i] = (1.0/(Pi*Pi))*sin(Pi*X[i])*exp(-(Pi*Pi+1)*(j)*k);
    /*quando a força F=(sin(Pi*x)*exp(-t))*/
    G[i] = (1.0/(Pi*Pi))*sin(Pi*X[i])*exp(-(j)*k);
    g[1]=0.0;
    g[Nos]=0.0;
    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-1)*k, g[i], G[i],
        EAbs[i], F0[i]);
    fprintf(f2, "%3.4f ", g[i]);
}
for (i=1; i<=Nos; i++)
{
    fprintf(f2, "%3.4f ", G[i]);
}

```

```

    }
    Norma(G);
    a1 = v[1]; a0 = v[0];
    Norma(E);
    /*Calcula os erros na norma L^2 em todo tempo*/
    fprintf(f3,"%f    %f\n", (j-1)*k, v[0]);

    if (MaxLInf<= v[0]) MaxLInf = v[0];
    if (MaxH1<= v[1]) MaxH1 = v[1];
    //Erro na norma L^2 em cada tempo t
    fprintf(f1,"\n erabs(L2) = %f", v[0]);
    //Erro na norma H^1 em cada tempo t
    fprintf(f1,"\n erabs(H1) = %f \n", v[1]);

    //fprintf(f1, "\n");
    for (i=1; i<=Nos; i++)
    {
        g0[i]=g[i];
        v0[i]=v[i];
        F0[i]=F[i];
    }
    j=j-1;
}
fprintf(f1,"\n Erro na norma LInfinito = %f", MaxLInf);
fprintf(f1,"\n Erro na norma H1 = %f", MaxH1);
fclose(f1);
fclose(f2);
fclose(f3);
}

```

A.8 Equação da onda unidimensional –Onda.cpp

```

/*
Problema Hiperbólico da Equação da Onda Unidimensional
Metodo de Elementos Finitos
 $U_{tt}(x,t) - (\text{Alpha})U_{xx}(x,t) + (\text{Beta})U(x,t) = f(x,t)$ 
tipo=1:  $u()$  prescrita
tipo=2:  $du()$  prescrita

Problema1: Alpha=1 e Beta=0 então
    f=0 e solução exata  $u(x,t)=\text{sen}(\text{Pi}*x)\cos(\text{lambda}*\text{Pi}*t)$ 
     $u(x,0)=\text{sen}(\text{Pi}*x)$  e  $u'(x,0)=0$ 
    onde  $\text{lambda}=\text{sqrt}(\text{alpha}+\text{beta}/\text{Pi}^2)$ 

Problema2: Alpha=Beta=1 então (tomando  $\text{lambda}=1$ )
     $f(x,t)=u(x,t)=\text{sen}(\text{Pi}*x)\cos(\text{lambda}*\text{Pi}*t)$  (força=solução exata)
     $u(x,0)=\text{sen}(\text{Pi}*x)$  e  $u'(x,0)=0$ 

Método de Newmark:  $U^{(n)}=\text{teta}*(U^{n-1}+U^{n+1})+(1-2*\text{teta})*U^n$ 
Se  $\text{teta}=0$  então Método 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,d1,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);
/*k=delta t=Passo no tempo*/
    printf("\nConstante k (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,"Constante k= %f\n", k);
    fprintf(f1,"Numero de elementos = %d\n", Nel);
    fprintf(f1,"Solução 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 = h, onde h é o passo no espaço */
    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. 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 Prob1*/
    /*return sin(Pi*x)*cos(Pi*j*k);*/ /*funcao de forca prescrita Prob2*/
}

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*d1);          /* Norma em L2*/
    v[1] = sqrt(L2/4*d1)+sqrt(H1*d1); /*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 Prob. 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;
        /* Solução exata inicial para t=0*/
        G[i] =sin(Pi*X[i])*cos(lambda*Pi*j*k);
        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];
// Erro na norma L2 em cada tempo t
fprintf(f1, "\n erro(L2) = %f", v[0]);
// Erro na norma H1 em cada tempo t
fprintf(f1, "\n erro(H1) = %f \n", v[1]);
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;
/*Sol.exata para Poblema 1 e 2*/
    G[i] = (sin(Pi*X[i])*cos(lambda*Pi*j*k));
    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];
// Erro norma L^2 em cada tempo t
    fprintf(f1,"\n erabs(L2) = %f", v[0]);
//Erro na norma H^1 em cada tempo t

    fprintf(f1,"\n erabs(H1) = %f \n", v[1]);
    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 Analysis Framework. Springer (2005)
- [2] Brézis, H.: Analyse Fonctionnelle, 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, Springer-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 differential 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^o4 723-759 (1973)

Índice

Algoritmos

- Crout, 140, 142
- equação da onda, 227
- equação do calor, 211

Análise numérica

- equação da onda, 277
- equação do calor, 239

band, 147, 317, 325, 337

bdy, 337

Bv, 325, 337

bv, 195

$C^\infty(\Omega)$, 21

CondFront, 327, 339

CondFront, 62, 111, 185, 309

Condição de fronteira, 102

- Dirichlet, 52, 152, 194

- mista, 58, 59, 152, 198

- Neumann, 55, 152, 196

Condutividade térmica, 2, 3

Conservação

- de energia, 304

- de energia, 1

- de momento angular, 9

- de momento linear, 8

Constante de Lamé, 10

Convenção de somatório, 12

Convergência

- funções testes, 21

CoordGlobal, 60, 307

CoordLocal, 60, 307

Corpo

- homogêneo, 4

- isotrópico, 3, 10

$\mathcal{D}'(\Omega)$, 22

DataInput, 339

DataInput, 108, 326

Decaimento

- assintótico, 272

- exponencial, 273

Decaimento assintótico

- caso discreto, 274

Decomposição

- U^tDU , 140

- LU, 140

- do erro, 250

Delta de Kronecker, 3

Densidade, 25

Desigualdade

- Gronwall, 24

- Poincaré-Friedricks, 25

Diferenças Finitas, 25

Dilatação, 7

Discretização do domínio, 107

DPhi, 122, 318

Elemento retangular, 107

ElmPos, 109, 318

eqn, 111, 173, 317, 337

EqNo, 110, 173, 330

Equação

- parabólica, 207

- de elasticidade linear, 168

- de energia, 2

- de movimento, 8
- de Poisson, 102
- Erro
 - de interpolação, 92
 - em H^1 , 95
 - em H^m , 98
 - em L^2 , 95
- Erro numérico, 64, 99, 148, 202
- Espaço
 - distribuições, $\mathcal{D}'(\Omega)$, 22
 - funções testes, 21
 - Sobolev, $H^m(\Omega)$, 23
- Estimativa Ótima, 98
- Estimativa ótima
 - norma $L^2(\Omega)$, 245
- Estimativa de Erro, 277
- Estimativa de erro, 239
 - norma $H_0^1(\Omega)$, 287
 - Norma $L^2(\Omega)$, 296
- Estimativa na norma $H_0^1(\Omega)$, 244
- Estimativa na norma $L^2(\Omega)$, 241
- Existência de solução, 104, 266, 299
- Forma bilinear, 19
 - coerciva, 19
 - contínua, 19
 - simétrica, 19
- Formulação Variacional
 - Equação da Onda, 225
 - Equação do Calor, 208
- Fronteira, 111, 326, 327, 339
- Fronteira de Neumann, 135, 188
- Função base
 - cúbica, 78
 - de ordem superior, 75
 - Hermite, 84
 - linear, 34
 - quadrática, 76
- Função de interpolação, 34, 118
 - dos dados, 113
 - gradiente de, 122
 - local, 41
- Função interpolante, 98
 - g_1 , 62, 308
 - g_2 , 62, 309
- Gauss-Legendre, 49
- Geração de malha, 107
- GlobalSystem, 331, 338, 346
- GlobalSystem, 139
- Gradiente
 - de deformação, 5
 - de deslocamento, 6
- Grid, 320
- GridData, 321
- $H^m(\Omega)$, 23
- $H_0^1(\Omega)$, 23
- Hess, 330, 345
- Identidade polar, 291
- InputCondFront, 60, 307
- InputData, 60, 306
- Interpolação de função, 113
- Interpolante, 92
- Jaco, 331
- Jacob, 122, 345
- Jacobiano, 121
- $L_{loc}^1(\Omega)$, 22
- $L^2(\Omega)$, 22
- Lei
 - de Fourier, 2, 101
 - de Hooke, 10, 167
- Lema
 - Céa, 90
 - Douglas, Dupont, 241
- LinearSolver, 63, 310
- Load, 330
- LocalSystem, 331
- LocalSystem, 127, 181
- Método

- Crank-Nicolson, 214, 255
- Newmark, 283
- Trapezoidal, 214
- Método
 - diferença central, 227
- Método de Euler
 - progressivo, 213
 - regressivo, 212, 250
- Método Numérico
 - θ -método, 214
 - θ -método: equação da onda, 230
- Método trapezoidal, 258
- Métodos Numéricos e Algoritmos
 - equação da onda, 224
 - equação do calor, 207
- Matriz
 - condutividade, 101, 104
 - rigidez, 33, 35
- Matriz rigidez
 - global, 42, 46, 115, 136, 175
 - local, 41, 115, 176
 - propriedades, 114, 174
- MatrizRigidez, 61, 308
- Método
 - de Cholesky, 140
 - de eliminação de Gauss, 38, 140
 - de colocação, 15
 - de Crout, 140, 142
 - de Galerkin, 16, 31, 105, 170
 - de projeção, 14
- Nbn, 325, 327, 337, 339
- Nel, 60, 90, 107, 317, 337
- Nelx, 108, 194, 317, 337
- Nely, 108, 194, 325, 337
- Neq, 111, 173, 317, 337
- Nno, 107, 171, 317, 337
- NoLG, 61, 110, 177, 308, 318
- NoPos, 108, 317
- Norm, 334, 350
- Norma, 63, 149, 312
- Norma $H_0^1(\Omega)$, 278
- Norma $L^2(\Omega)$, 280
- Nos, 60, 107
- Ordem de Convergência , 219
- Ortogonalidade de erro, 90
- OutputData, 63, 312
- OutputResultado, 63, 312
- Parametrização, 135
- Phi, 122, 318
- PhiMatriz, 127, 128, 319
- PlotData, 321
- PosNo, 108, 317
- Princípio do máximo, 272
- Problema
 - aproximado, 32, 106
 - discreto, 247, 283
 - semidiscreto, 239, 277
 - de calor, 13, 101
 - elastostático linear, 13, 168
- Programas computacionais, 305
- Qm, 330
- Quadratura Gaussiana, 49, 124
- Rotação infinitesimal, 199
- Séries de Fourier, 17
- Símbolo de permutação, 13
- Sistema aproximado, 270, 299
- SolExata, 63, 310
- Solução aproximada, 171
- Solver, 147, 192
- Spline cúbica, 78
- Taxa de Convergência, 99
- Tempo contínuo, 240
- Tensor
 - de deformação infinitesimal, 6, 167
 - de elasticidade, 10, 168
 - de rotação infinitesimal, 6
 - de tensão, 8, 167

Teorema

Aubin-Lions, 24

Caratheodory, 300

da divergência, 103

de regularidade de solução, 275

do traço, 25

Lax-Milgram, 20, 30, 31, 104

TractionBoundary, 135, 188, 348

Transformação isoparamétrica, 49, 121

typ, 317, 337

Unicidade de solução, 104, 271

problema de Neumann, 153, 199

Valor de fronteira, 111, 129

Vetor força, 37

global, 44, 48, 115, 136, 175

local, 41, 115, 176

VetorForca, 62, 309

Xi, 61, 308