

**ELEMENTOS DE CONTORNO  
DE GALERKIN SIMÉTRICO**

**WERNER HEISENBERG SANTOS FIGUEIREDO**

**PROJETO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**FACULDADE DE TECNOLOGIA**

**UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**ELEMENTOS DE CONTORNO  
DE GALERKIN SIMÉTRICO**

**WERNER HEISENBERG SANTOS FIGUEIREDO**

**ORIENTADOR: PROF. DR. ÉDER LIMA DE ALBUQUERQUE, ENM/UNB**

**PROJETO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA**

**PUBLICAÇÃO -  
BRASÍLIA-DF, 27 DE NOVEMBRO DE 2019.**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**ELEMENTOS DE CONTORNO  
DE GALERKIN SIMÉTRICO**

**WERNER HEISENBERG SANTOS FIGUEIREDO**

PROJETO DE GRADUAÇÃO ACADÊMICO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO MECÂNICO EM ENGENHARIA MECÂNICA.

APROVADA POR:

---

Prof. Dr. Éder Lima de Albuquerque, ENM/UnB  
Orientador

---

Prof. Dr. Marcus Vinicius Girão de Moraes, ENM/UnB  
Examinador interno

---

MSc. Álvaro Campos Ferreira, ENM/UnB  
Examinador interno

**BRASÍLIA, 27 DE NOVEMBRO DE 2019.**

## **FICHA CATALOGRÁFICA**

WERNER HEISENBERG SANTOS FIGUEIREDO

**Elementos de Contorno de Galerkin Simétrico**

**2019xv, 70p., 201x297 mm**

(ENM/FT/UnB, Engenheiro Mecânico, Engenharia Mecânica, 2019)

Projeto de Graduação - Universidade de Brasília

Faculdade de Tecnologia - Departamento de Engenharia Mecânica

## **REFERÊNCIA BIBLIOGRÁFICA**

WERNER HEISENBERG SANTOS FIGUEIREDO (2019) Elementos de Contorno de Galerkin Simétrico. Projeto de Graduação em Engenharia Mecânica, Publicação, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 70p.

## **CESSÃO DE DIREITOS**

AUTOR: Werner Heisenberg Santos Figueiredo

TÍTULO: Elementos de Contorno de Galerkin Simétrico.

GRAU: Engenheiro Mecânico ANO: 2019

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

---

Werner Heisenberg Santos Figueiredo

Correio Eletrônico: werner.hsfig@gmail.com

# Agradecimentos

*A minha família, pelo apoio constante e sem ela nada disso seria possível, em especial minha mãe, Luzia, meu pai, Jackson, e meu irmão, Gibson.*

*Ao professor Éder, pela paciência, disposição e entusiasmo ao repassar seu conhecimento.*

*A todos meus amigos e colegas de curso que tornaram esta uma jornada agradável e sensacional.*

*Werner Heisenberg Santos Figueiredo*

---

## ABSTRACT

This thesis presents a detailed approach to the formulation of symmetric Galerkin boundary element method for two-dimensional problems. A validation was performed using the numerical implementation in MatLab<sup>®</sup> language, which was applied to heat conduction problems and compared to collocation boundary element programs legitimized by the literature. The validity of the symmetric Galerkin code is confirmed. Since the two-dimensional formulation can be extended to a certain extent for the three-dimensional case, the Bempp open source library was studied because of the utility of the software to solve three-dimensional engineering problems using the Galerkin boundary element method. The verification of its effectiveness was explored from its application in a thermal problem, where it was observed its attractiveness to the university environment due to an efficiency to solve relatively small problems, however, presented underperformance in large and complex problems.

**Keywords:** Symmetric Galerkin boundary element method. Bempp. Tridimensional thermal analysis.

---

## RESUMO

Este trabalho apresenta uma abordagem detalhada para a formulação do método dos elementos de contorno simétrico de Galerkin para problemas bidimensionais. Sua validação foi realizada a partir da implementação numérica de um programa em linguagem MatLab<sup>®</sup> aplicado a problemas de condução de calor e comparado a programas de elementos de contorno por colocação reconhecidos pela literatura. A validade do código de Galerkin simétrico pôde ser confirmada. Como a formulação bidimensional pode ser estendida a certo ponto para o caso tridimensional, foi efetuado o estudo da biblioteca de código aberto Bempp devido a utilidade do programa para solucionar problemas tridimensionais de engenharia utilizando o método dos elementos de contorno de Galerkin simétrico. A verificação de sua eficácia foi explorada a partir de sua aplicação em um problema térmico onde foi observada sua atratividade para o meio universário devido a suficiência para resolver problemas relativamente pequenos. No entanto, apresentou performance abaixo do esperado em problemas grandes e complexos.

**Palavras-chave:** Método dos elementos de contorno simétrico de Galerkin. Bempp. Análise térmica tridimensional.

# Sumário

<b>1</b>	<b>Introdução</b> .....	<b>1</b>
1.1	O método dos elementos de contorno . . . . .	1
1.2	O método de Galerkin . . . . .	4
1.3	Objetivos e principais contribuições do trabalho . . . . .	5
1.4	Plano da dissertação . . . . .	5
<b>2</b>	<b>Método dos Resíduos Ponderados</b> .....	<b>7</b>
2.1	Soluções aproximadas . . . . .	7
2.2	Resíduo . . . . .	8
2.3	Diferenças entre Galerkin e colocação . . . . .	9
2.3.1	Método da colocação . . . . .	9
2.3.2	Método de Galerkin . . . . .	12
<b>3</b>	<b>A Equação de Laplace</b> .....	<b>15</b>
3.1	Equação de Poisson para transferência de calor . . . . .	15
3.1.1	Solução fundamental para a equação de Laplace . . . . .	18
3.1.2	Fluxo de calor através do contorno . . . . .	20
<b>4</b>	<b>Equações Integrais de Contorno</b> .....	<b>22</b>
4.1	Equações integrais de contorno . . . . .	22
4.1.1	Equação regular . . . . .	22
4.1.2	Equação hipersingular . . . . .	27
<b>5</b>	<b>Integração Analítica das Matrizes H e G</b> .....	<b>29</b>
5.1	Discretização das equações integrais de contorno . . . . .	29



5.2	Elementos de contorno constantes . . . . .	30
5.2.1	Integração analítica com elementos constantes . . . . .	31
5.3	Elementos de contorno lineares contínuos . . . . .	32
5.3.1	Integração analítica com elementos lineares contínuos . . . . .	35
<b>6</b>	<b>Método dos Elementos de Contorno de Galerkin Simétrico . . . . .</b>	<b>40</b>
6.1	Formulação de Galerkin para o MEC . . . . .	41
6.2	Consecução do sistema linear simétrico . . . . .	43
<b>7</b>	<b>Implementação Computacional . . . . .</b>	<b>45</b>
7.1	Descrição dos programas . . . . .	45
<b>8</b>	<b>Simulação Numérica e Resultados . . . . .</b>	<b>48</b>
8.1	Condução de calor na placa quadrada . . . . .	48
8.2	Condução de calor na placa trapezoidal . . . . .	53
<b>9</b>	<b>O Programa Bempp . . . . .</b>	<b>56</b>
9.1	Bempp e plataformas relacionadas . . . . .	56
9.1.1	O projeto FEniCS . . . . .	57
9.1.2	Paralelização MPI . . . . .	58
9.1.3	Matrizes hierárquicas . . . . .	58
9.2	Metodologia . . . . .	59
9.3	Resultados . . . . .	60
<b>10</b>	<b>Conclusões e Trabalhos Futuros . . . . .</b>	<b>63</b>
	<b>Referências Bibliográficas . . . . .</b>	<b>64</b>
	<b>Anexos . . . . .</b>	<b>66</b>
I.1	Programa Principal Polia - Python . . . . .	66

# Lista de Figuras

3.1	Transferência de calor de $u_0$ para $u_1$ . . . . .	16
3.2	Fluxo de calor unidirecional. . . . .	17
3.3	Distância $r$ entre o ponto fonte e o ponto campo. . . . .	18
3.4	Posições dos pontos fonte e campo. . . . .	20
4.1	Contornos original e adaptado. . . . .	25
4.2	Ângulo interno do contorno. . . . .	27
5.1	Discretização do contorno em $n$ partes. . . . .	30
5.2	Aproximação do contorno por segmentos de retas. . . . .	30
5.3	Sistema de referências usado para o cálculo das integrais analíticas. . . . .	32
5.4	Elementos lineares contínuos. . . . .	33
5.5	Sistema de coordenadas local. . . . .	35
5.6	Projeção escalar dos vetores $\mathbf{r}$ , $\mathbf{r}_1$ e $\mathbf{r}_2$ no vetor normal unitário e no vetor tangente unitário. . . . .	36
6.1	Contorno dividido em $\Gamma_1$ e $\Gamma_2$ . . . . .	43
7.1	Fluxograma dos programas de elementos de contorno. . . . .	47
8.1	Dimensões da placa quadrada. . . . .	49
8.2	Distribuição de temperatura ao longo do segmento $\overline{AD}$ da placa quadrada. . . . .	50
8.3	Distribuição de temperatura ao longo do segmento $\overline{BC}$ da placa quadrada. . . . .	50
8.4	Fluxo de calor ao longo do segmento $\overline{DC}$ da placa quadrada. . . . .	51
8.5	Fluxo de calor ao longo do segmento $\overline{AB}$ da placa quadrada. . . . .	51
8.6	Distribuição de temperatura ao longo do segmento $\overline{AD}$ da placa quadrada (malha refinada). . . . .	52

8.7	Distribuição de temperatura e fluxo de calor ao longo da placa quadrada. . .	52
8.8	Dimensões da placa trapezoidal. . . . .	53
8.9	Distribuição de temperatura ao longo do segmento $\overline{AB}$ da placa trapezoidal.	54
8.10	Fluxo de calor ao longo do segmento $\overline{BC}$ da placa trapezoidal. . . . .	55
8.11	Distribuição de temperatura e fluxo de calor ao longo da placa trapezoidal. .	55
9.1	Distribuição de temperatura na polia. . . . .	60

# Lista de Tabelas

2.1	Resultados para colocação . . . . .	11
2.2	Resultados para Galerkin . . . . .	14
9.1	Tempo consumido para diferentes tamanhos de bloco (malha 1). . . . .	61
9.2	Tempo consumido para diferentes números de threads (malha 1). . . . .	61
9.3	Tempo consumido para diferentes ranks máximos (malha 1). . . . .	61
9.4	Tempo consumido para diferentes tamanhos de bloco (malha 2). . . . .	62
9.5	Tempo consumido para diferentes números de threads (malha 2). . . . .	62
9.6	Tempo consumido para diferentes ranks máximos (malha 2). . . . .	62

# Capítulo 1

## Introdução

### 1.1 O método dos elementos de contorno

O estudo do comportamento das estruturas e outros problemas de engenharia é feito hoje usando computadores. A razão é bastante óbvia: o baixo custo da simulação numérica versus a dispendiosa análise experimental. A modelagem numérica pode ser usada para estudar uma ampla variedade de cargas e geometrias em uma estrutura e para determinar a melhor solução de projeto, antes de prosseguir para sua construção. O método mais popular usado para a análise de problemas de engenharia nos últimos 45 anos é o método dos elementos finitos (MEF). É o método com o qual problemas realistas de engenharia estão sendo resolvidos, isto é, a análise de elementos estruturais de geometria arbitrária, carga arbitrária, variedade de relações constitutivas, com comportamento linear ou não linear, em duas ou três dimensões. Justificadamente, o MEF tem sido considerado nos últimos 45 anos como uma ferramenta computacional moderna.

O método dos elementos de contorno (MEC) é um método computacional para a solução de sistemas de equações diferenciais, formuladas em forma integral. Assim como o MEF, também é aplicado em diversas áreas da engenharia, como em mecânica dos fluidos, acústica, eletromagnetismo e mecânica da fratura. A principal diferença entre o método dos elementos de contorno e o método dos elementos finitos é que no primeiro somente o contorno do problema é discretizado (dividido em elementos). Isto, em muitos casos, reduz drasticamente o tamanho do problema, além de ser mais simples a implementação de pré-processadores (geradores de malha).

A justificativa para a necessidade do MEC é que modelar com elementos finitos pode ser ineficaz e trabalhoso para certas classes de problemas. Assim, o MEF, apesar da generalidade de sua aplicação em problemas de engenharia, não está livre de inconvenientes, os mais importantes dos quais são:

1. A discretização é sobre todo o domínio ocupado pelo corpo. Assim, a geração e a inspeção da malha de elementos finitos apresentam dificuldade, são trabalhosas e de-

moradas, especialmente quando a geometria do corpo não é simples. Por exemplo, quando há furos, entalhes ou cantos, refinamento de malha e alta densidade de elementos são necessários nessas regiões críticas de grandes gradientes de solução.

2. A modificação do modelo discretizado para melhorar a precisão da solução ou para refletir as alterações de projeto pode ser complexa e requer muito esforço e tempo.
3. Para domínios infinitos, por exemplo, é necessária a introdução de limites fictícios fechados para poder aplicar o MEF. Isso reduz a precisão e, às vezes, pode resultar em soluções falsas ou incorretas.
4. Para os problemas descritos por equações diferenciais de quarta ordem ou superior (isto é, equações de placa ou equações de casca de sexta, oitava ou maior ordem), os requisitos de conformidade exigem um trabalho tão tedioso que o MEF pode se tornar impraticável.
5. Embora o MEF calcule com precisão a função de campo, que é o desconhecido do problema, é ineficaz na determinação de suas derivadas com a mesma precisão. A precisão cai consideravelmente em áreas de grandes gradientes.

As desvantagens 1 e 2 podem ser superadas usando programas comerciais de elementos finitos, como NASTRAN, ANSYS, ABAQUS, etc., que são equipados com geradores de malha automáticos e adaptáveis. A desvantagem desta abordagem é que estes códigos comerciais são fechados, exigindo o pagamento de licenças que podem ser bastante caras a depender do tamanho da empresa. Essencialmente, a tarefa de gerar uma malha de elementos finitos é um problema geométrico difícil e, em vários casos, pode revelar-se muito mais complexo do que o problema físico a ser resolvido pelo MEF.

Por outro lado, o MEC possui muitas vantagens, sendo as mais importantes:

1. A discretização é apenas sobre o contorno do corpo, facilitando a modelagem numérica e reduzindo o número de incógnitas em uma ordem. Assim, uma remodelação para refletir as mudanças no projeto torna-se bem mais simples.
2. Para domínios infinitos, o problema é formulado simplesmente como um exterior. A solução fundamental satisfaz as condições no infinito, como a condição de radiação de Sommerfeld para problemas de propagação de ondas. Desta forma, programas de computador desenvolvidos para domínios finitos podem ser usados, com apenas algumas modificações, para resolver problemas em domínios infinitos. Isso não é possível com o MEF.
3. O método é particularmente eficaz para calcular com precisão as derivadas da função de campo (por exemplo, fluxos, deformações, tensões e momentos). Ele pode lidar facilmente com forças e momentos concentrados, seja dentro do domínio ou no contorno.

4. O MEC permite a avaliação da solução e suas derivadas em qualquer ponto do domínio do problema e a qualquer instante no tempo. Isso é viável porque o método usa uma representação integral da solução como uma expressão matemática contínua, que pode ser diferenciada e utilizada como uma fórmula matemática. Isso é impossível com o MEF, já que a solução é obtida apenas nos pontos nodais.
5. O método é bem adequado para resolver problemas em domínios com peculiaridades geométricas, como trincas.
6. As malhas de elementos de contorno, especialmente as tridimensionais, podem ser facilmente ligadas a sistemas de CAE (computer-aided engineering), pois para estes sistemas também só são necessárias informações sobre o contorno do corpo. O processo de discretização é ainda mais simples quando se utilizam elementos descontínuos, que não são admissíveis em elementos finitos.

Entretanto, o MEC apresenta as seguintes principais desvantagens inerentes:

1. A aplicação do MEC requer o estabelecimento da representação integral da equação diferencial. Isso só é possível se o problema for linear e sua solução fundamental puder ser estabelecida. Portanto, o método não pode ser usado para problemas cuja solução fundamental é desconhecida ou não pode ser determinada. Tais são, por exemplo, problemas descritos por equações diferenciais com coeficientes variáveis. O método não é aplicável a problemas não-lineares para os quais uma identidade recíproca não pode ser estabelecida. Durante os últimos 25 anos, intensas pesquisas foram conduzidas e métodos eficientes foram desenvolvidos para superar esta desvantagem (método da integração radial, método da reciprocidade dual, dentre outros).
2. A implementação numérica do MEC resulta em sistemas de equações algébricas lineares cujas matrizes de coeficiente são totalmente povoadas (matrizes cheias) e não simétricas. Em um modelo MEF, no entanto, as matrizes correspondentes têm algumas propriedades muito boas: elas são esparsas (cheias de zeros) em bandas e simétricas. Embora as matrizes do MEC tenham tamanho menor, por serem cheias requerem mais operações para calcular os coeficientes e para resolver o sistema linear. Por isso, sua eficiência nas soluções tem sido um problema sério para a análise de modelos em grande escala. Por exemplo, o MEC se limita a resolver problemas com alguns milhares de graus de liberdade (DOFs) em um computador pessoal (cuja memória RAM atualmente dificilmente é superior a 16 GBytes).

O fato do método dos elementos de contorno ter matrizes cheias e não simétricas é apontado por pesquisadores como a principal desvantagem do método. Os métodos rápidos (fast multipole e matrizes hierárquicas em geral) aproximam as matrizes por blocos de baixo posto, que torna muito mais eficiente os algoritmos para cálculo do produto matriz-vetor com matriz cheia e, por sua vez, o método dos elementos de contorno como um todo.

Os métodos rápidos resolvem o problema de esparsidade das matrizes do MEC, porém não tratam a sua assimetria. A simetria das matrizes pode ser desejável devido as conhecidas propriedades de matrizes simétricas demonstradas pela álgebra linear. A mais importante delas é que matrizes simétricas são sempre positivo-definidas ou positivo-semidefinidas (todos os seus auto-valores são maiores ou iguais a zero). Alguns métodos iterativos bastante eficientes na solução de sistemas lineares só são aplicáveis a matrizes simétricas como, por exemplo, o método do gradiente conjugado.

## 1.2 O método de Galerkin

A aproximação do método de Galerkin aplicado às equações integrais de contorno tem muitas vantagens importantes em relação ao método da colocação, mais amplamente utilizado (KANE, 1994). Em primeiro lugar, é a capacidade de avaliar, sem ambiguidade, intervalos hipersingulares (FRANGI; NOVATI, 1996) usando o padrão de elementos contínuos (GRAY, 1991). Com o método da colocação, a existência destas integrais depende de uma restrição de diferenciabilidade (MARTIN; RIZZO, 1996), contudo, trabalhos recentes indicam que, em cálculos computacionais, esse problema pode ser amenizado (MARTIN; RIZZO; CRUSE, 1998) (RICHARDSON; CRUSE, 1998). Outros benefícios da utilização de Galerkin são a obtenção de maior precisão, uma análise simples e confiável nos cantos, e uma teoria matemática bem desenvolvida (SCHATZ; THOMÉE; WENDLAND, 1990). Além disso, com a utilização do método de Galerkin é possível formular um método de elementos de contorno naturalmente simétrico. No entanto, essas vantagens vêm por meio de uma segunda integração de contorno e, por isso, a principal desvantagem do método de Galerkin tem sido o maior tempo computacional consumido, aproximadamente uma ordem de grandeza maior que a colocação. Compreensivelmente, isso limitou severamente o uso do método de Galerkin em aplicações de engenharia.

O algoritmo do método simétrico de Galerkin para integrais de contorno, primeiro introduzido por Sirtori (1979), Hartmann, Katz e Protosaltis (1985), e posteriormente desenvolvido por Maier e Polizzotto (1987) e outros, parcialmente removeu a dramática diferença de desempenho entre os métodos de colocação e Galerkin (BALAKRISHNA; GRAY; KANE, 1994). Nessa abordagem, uma matriz de coeficientes simétricos é obtida utilizando o método de Galerkin juntamente com uma combinação apropriada de equações singulares e hipersingulares. A simetria da matriz pode então ser explorada para reduzir as operações de eliminação gaussianas de  $\frac{M^3}{3}$  para  $\frac{M^3}{6}$ , onde  $M$  é o tamanho do sistema linear. Assim, para problemas de grande escala, economias significativas de tempo são possíveis.

Para valores grandes de  $M$ , um algoritmo de solução direta é substituído por um esquema iterativo (KANE; KEYES; PRASAD, 1991). É claramente desejável a redução do tempo computacional ao usar o método de Galerkin, aproveitando ao máximo a simetria durante a fase de construção da matriz do cálculo. A abordagem mais natural é avaliar apenas o



triângulo superior (ou inferior), ignorando as integrações desnecessárias. Esta estratégia é parcialmente eficaz para algumas aplicações específicas (BUI, 1977), ou problemas de interface (MAIER; DILIGENTI; CARINI, 1991) mas, em geral, se comparado ao método de colocação, não se tem ganhos significativos ao explorar a simetria durante a montagem da matriz.

### 1.3 Objetivos e principais contribuições do trabalho

Os objetivos do trabalho são a exibição de uma abordagem detalhada para a formulação do MEC de Galerkin simétrico e o estudo da performance do programa Bempp, que é uma biblioteca open source de elementos de contorno que utiliza a formulação de Galerkin.

Foram apresentados os passos para a implementação numérica do método dos elementos de contorno simétrico de Galerkin. Dentre os principais tópicos, estão:

- A exposição das diferenças entre os métodos de Galerkin e Colocação, em que são usadas técnicas distintas para satisfazer as equações integrais de contorno;
- A dedução das equações integrais de contorno, a fim de mostrar como são obtidas as equações regular e hipersingular;
- A descrição da integração analítica das matrizes  $\mathbf{H}$  e  $\mathbf{G}$ , para elementos constantes e lineares contínuos;
- A implementação do MEC simétrico de Galerkin para problemas 2D;
- A comparação, por meio da análise de distribuição de temperatura e fluxo de calor em problemas bidimensionais, da formulação de elementos de contorno padrão com a formulação de Galerkin;
- O estudo do programa open source Bempp e sua aplicação para resolver problemas 3D.

### 1.4 Plano da dissertação

A sequência deste trabalho é dividida em outros 9 capítulos.

No capítulo 2, a teoria dos métodos dos resíduo ponderados é apresentada a fim de expor as diferenças entre os métodos de Galerkin e colocação, além de ser aplicada à equação de Laplace para a dedução da equação integral de contorno. No capítulo 3, a equação de Laplace é deduzida e sua solução fundamental é calculada, etapas necessárias para a obtenção da equação integral de contorno. O capítulo 4 apresenta uma breve revisão sobre a formulação clássica do MEC para problemas bidimensionais governados pela equação de Laplace e nele

é detalhada a dedução das equações integrais de contorno regular e hipersingular. No capítulo **5** é feita a discretização da equação integral de contorno, em que são considerados elementos constantes e lineares. O capítulo exibe a descrição da formulação de elementos de contorno onde a integração das matrizes de influência **H** e **G** é calculada analiticamente. O capítulo **6** retrata a formulação do MEC simétrico de Galerkin, onde são expostas as estratégias para a obtenção do sistema de equações simétrico. No capítulo **7** é feita uma descrição do algoritmo de Galerkin implementado para resolver problemas de condução de calor. O capítulo **8** contém os resultados numéricos obtidos com a implementação do código, onde a simulação computacional da distribuição de temperatura em placas planas de geometrias diferentes é realizada. O capítulo **9** expõe a análise relativa ao desempenho do programa Bempp, software utilizado para solucionar um problema térmico tridimensional. No capítulo **10** estão expressas as conclusões e sugestões para futuros trabalhos.

# Capítulo 2

## Método dos Resíduos Ponderados

O intuito deste capítulo é explicar o conceito de solução aproximada, explicar a definição de resíduo, introduzir o conceito de método dos resíduos ponderados e desenvolver dois desses métodos: colocação e Galerkin. Para este último, será aprofundado o seu conteúdo e aplicações no método dos elementos de contorno nos próximos capítulos deste trabalho. A redação deste capítulo foi baseada no livro de Brebbia e Dominguez (1994).

### 2.1 Soluções aproximadas

Em muitos problemas de engenharia, descobrir uma função que satisfaça uma equação diferencial pode apresentar um alto grau de complexidade. No entanto, muitas vezes basta obter uma solução para um determinado intervalo, e não para todo o domínio. Os métodos de resíduos ponderados encontram uma função que aproxima a solução exata em um dado intervalo com um grau de precisão satisfatório, e tanto melhor quanto menor o intervalo.

A fim de exemplificar a utilização do método, considere uma equação diferencial ordinária simples aplicada em um domínio unidimensional  $x$ :

$$\frac{d^2u}{dx^2} + \lambda^2u = b, \quad \text{em } x \in [0, 1], \quad (2.1)$$

onde  $u$  é a função governante da equação,  $\lambda^2$  é uma constante positiva conhecida e  $b$  é uma função conhecida de  $x$ .

As condições de contorno que correspondem a esta equação são:

$$u = \bar{u} \quad \text{em } x = 0 \quad (2.2)$$

e

$$q = \frac{du}{dx} = \bar{q} \quad \text{em } x = 1. \quad (2.3)$$

As derivadas de  $u$  são definidas como  $q$  e os termos  $\bar{u}$  e  $\bar{q}$  representam valores conhecidos da função e de sua derivada, respectivamente.

Ao introduzir uma solução aproximada  $\tilde{u}$  que satisfaça as condições de contorno para a função  $u(x)$  apresentada na Eq.(2.1), geralmente obtém-se algo na forma:

$$\frac{d^2\tilde{u}}{dx^2} + \lambda^2\tilde{u} - b \neq 0, \quad \text{para } x \in [0, 1]. \quad (2.4)$$

## 2.2 Resíduo

Apresentadas essas condições, agora é possível introduzir o conceito de função erro ou resíduo, que representa os erros que ocorrem no domínio pelo fato da aproximação não satisfazer a equação diferencial. A função erro no domínio é chamada de resíduo ( $R$ ) e é dada por:

$$R = \frac{d^2\tilde{u}}{dx^2} + \lambda^2\tilde{u} - b \neq 0. \quad (2.5)$$

Por mais que o caso apresentado seja para uma equação específica e relativamente simples, o mesmo ocorre para qualquer outro problema (BREBBIA; DOMINGUEZ, 1994).

A ideia da utilização dos resíduos ponderados é minimizar o  $R$ , e isso é obtido fazendo o resíduo ponderado ser igual a 0 para vários valores de funções pesos ( $\Psi_j$ ),

$$\int_0^1 R\Psi_j dx = 0, \quad j = 1, 2, \dots, N. \quad (2.6)$$

Uma outra maneira de escrever a equação (2.6) de uma forma mais compacta e fácil de operar, é definindo uma nova função  $\omega$ :

$$\omega = \beta_1\Psi_1 + \beta_2\Psi_2 + \dots + \beta_n\Psi_n = \sum_{j=1}^n \beta_j\Psi_j, \quad (2.7)$$

em que  $\beta_j$  são coeficientes arbitrários. Com isso, a equação (2.6) pode ser escrita da forma:

$$\int_0^1 R\omega dx = 0, \quad \text{em } 0 < x < 1. \quad (2.8)$$

Generalizando as Eqs.(2.6) e (2.8) para um domínio  $\Omega$ , têm-se:

$$\int_{\Omega} R\Psi_j d\Omega = 0, \quad j = 1, 2, \dots, N. \quad (2.9)$$

e

$$\int_{\Omega} R\omega \, d\Omega = 0, \quad \text{em } \Omega. \quad (2.10)$$

## 2.3 Diferenças entre Galerkin e colocação

Métodos de aproximação baseados na equação (2.10) são os chamados métodos dos resíduos ponderados e, dada a solução aproximada, o método varia de acordo com as funções usadas como funções de ponderação.

A seguir, serão descritos dois métodos: Galerkin e colocação. Suas diferenças serão esclarecidas a partir do desenvolvimento de uma equação diferencial proposta.

### 2.3.1 Método da colocação

Neste caso,  $N$  pontos  $x_1, x_2, \dots, x_n$  são escolhidos no domínio e assume-se que o resíduo é igual a 0 nesses pontos. Isto é equivalente a considerar a função de ponderação em termos do delta de Dirac:

$$\Psi_j = \Delta(x - x_j); \quad j = 1, 2, \dots, N. \quad (2.11)$$

$\Delta(x - x_j)$  no ponto  $x - x_j$  tem um valor infinito, mas sua integral tem um valor unitário:

$$\int_{\Omega} \Delta(x - x_j) d\Omega = 1; \quad j = 1, 2, \dots, N. \quad (2.12)$$

A função de Dirac pode ser interpretada como o limite de uma função regular quando sua base tende a zero.

Assim, a equação (2.9) pode ser escrita como:

$$\int_{\Omega} R\Delta(x - x_j) d\Omega = 0; \quad j = 1, 2, \dots, N \quad (2.13)$$

que simplesmente significa que o erro da função é zero em uma série de pontos, ou seja:

$$R|_{x=x_j} = 0; \quad j = 1, 2, \dots, N. \quad (2.14)$$

Esse método consiste em assumir o resíduo ou função erro igual a 0 em tantos pontos quantos forem os coeficientes desconhecidos na solução aproximada. A distribuição dos pontos de colocação a princípio é arbitrária, mas na prática se obtém melhores resultados se eles forem distribuídos uniformemente.

A fim de ilustrar a utilização desse método, considere a equação (2.1), com  $\lambda^2 = 1$  e

$b = -x$ , ou seja:

$$\frac{d^2u}{dx^2} + u + x = 0. \quad (2.15)$$

Com as seguintes condições de contorno:

$$u = 0 \quad \text{em} \quad x = 0$$

e

$$u = 0 \quad \text{em} \quad x = 1.$$

A solução exata de (2.15) pode ser facilmente obtida por integração, e é dada por:

$$u = \frac{\sin x}{\sin 1} - x. \quad (2.16)$$

Define-se uma solução aproximada, dada por:

$$v = a_1\phi_1 + a_2\phi_2 + a_3\phi_3 + \dots \quad (2.17)$$

onde  $\phi_i$  são termos polinomiais em  $x$ , ou seja,

$$\phi_1 = 1, \quad \phi_2 = x, \quad \phi_3 = x^2 \dots \quad (2.18)$$

Para que as condições de contorno sejam satisfeitas, é necessário que:

$$\text{em } x = 0, \quad v = 0 \quad \rightarrow \quad a_1 = 0 \quad (2.19)$$

$$\text{em } x = 1, \quad v = 0 \quad \rightarrow \quad a_1 + a_2 + \dots = 0 \quad (2.20)$$

Daí, tem-se que:

$$a_2 = -(a_3 + a_4 + \dots). \quad (2.21)$$

Desta forma, usando 4 termos,  $v$  é dado por:

$$v = -(a_3 + a_4)x + a_3x^2 + a_4x^3 \quad (2.22)$$

$$= x(1-x)(-a_3 + a_4) + x(1-x)(-a_4)x \quad (2.23)$$

$$= x(1-x)(\alpha_1 + \alpha_2x). \quad (2.24)$$

onde

$$\alpha_1 = (-a_3 + a_4),$$

$$\alpha_2 = -a_4.$$

Substituindo essa função na equação governante (2.15), tem-se que o resíduo é:

$$R = \frac{d^2v}{dx^2} + v + x \quad (2.25)$$

$$= (-2 + x + x^2)\alpha_1 + (2 - 6x + x^2 - x^3)\alpha_2 + x. \quad (2.26)$$

Para se resolver pelo método da colocação, o resíduo é considerado igual a zero em pontos específicos do domínio, logo, fazendo  $R = 0$  em  $x = \frac{1}{4}$  e  $x = \frac{1}{2}$ , e substituindo esses valores em (2.26) obtém-se o sistema linear:

$$\begin{bmatrix} \frac{29}{16} & \frac{-35}{64} \\ \frac{7}{9} & \frac{7}{8} \end{bmatrix} \times \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{2} \end{bmatrix}. \quad (2.27)$$

A solução deste sistema fornece

$$\alpha_1 = \frac{6}{31} \quad \text{e} \quad \alpha_2 = \frac{40}{217}. \quad (2.28)$$

A solução aproximada de  $v$ , dada pela equação (2.24), pode então ser escrita como:

$$v = \frac{x(1-x)}{217}(42 + 40x). \quad (2.29)$$

Nota-se que agora a função erro pode ser totalmente definida em função de  $x$ , e assim, tem-se:

$$R = \frac{1}{217}(-4 + 19x - 2x^2 - 40x^3). \quad (2.30)$$

Com esses resultados, pode-se comparar por meio da Tabela 2.1 as soluções aproximada e exata, além de mostrar o valor do resíduo.

Tabela 2.1: Resultados para colocação

$x$	$u$ (exato)	$v$ (aproximado)	$R$
0,1	0,018641	0,019078	-0,009953
0,3	0,051194	0,052258	+0,002027
0,5	0,069746	0,071428	+0,00000
0,7	0,065585	0,065806	-0,024884
0,9	0,030901	0,032350	-0,081474

### 2.3.2 Método de Galerkin

Neste método as funções de ponderação são iguais às funções de interpolação:

$$\phi_j = \Psi_j. \quad (2.31)$$

A partir disso e da Eq.(2.9), tem-se que:

$$\int_{\Omega} R\phi_j \, d\Omega = 0; \quad j = 1, 2, \dots, N. \quad (2.32)$$

Usando a mesma definição mostrada em (2.7), tem-se que:

$$\int_{\Omega} R\omega \, d\Omega = 0 \quad (2.33)$$

com

$$\omega = \beta_1\phi_1 + \beta_2\phi_2 + \dots + \beta_n\phi_n. \quad (2.34)$$

Este método é o ponto de partida de muitas formulações de elementos finitos para os quais se obtém matrizes algébricas simétricas.

Resolvendo o mesmo problema por Galerkin, a solução aproximada pode ser a mesma mostrada em (2.24), ou seja:

$$v = \alpha_1x(1-x) + \alpha_2x^2(1-x), \quad (2.35)$$

que pode ser escrita da forma

$$v = \alpha_1\phi_1 + \alpha_2\phi_2, \quad (2.36)$$

onde  $\phi_1$  e  $\phi_2$  são funções de forma,

$$\phi_1 = x(1-x) \quad \text{e} \quad \phi_2 = x^2(1-x). \quad (2.37)$$

O resíduo é, então,

$$R = \frac{d^2v}{dx^2} + v + x \quad (2.38)$$

$$= (-2 + x + x^2)\alpha_1 + (2 - 6x + x^2 - x^3)\alpha_2 + x. \quad (2.39)$$

Presume-se que a função de ponderação  $\omega$  em Galerkin tenha a mesma função de forma que a solução aproximada (2.36),

$$\omega = \beta_1\phi_1 + \beta_2\phi_2, \quad (2.40)$$

onde  $\beta_1$  e  $\beta_2$  são coeficientes arbitrários.



A equação do resíduo ponderado é então dada por:

$$\int_0^1 R(\beta_1\phi_1 + \beta_2\phi_2) dx = 0. \quad (2.41)$$

Como  $\beta_1$  e  $\beta_2$  são arbitrários, tem-se que:

$$\int_0^1 R\phi_1 dx = 0 \quad (2.42)$$

e

$$\int_0^1 R\phi_2 dx = 0. \quad (2.43)$$

Substituindo (2.39) em (2.42) e (2.43) e integrando, obtém-se o seguinte sistema:

$$\begin{bmatrix} \frac{3}{10} & \frac{3}{10} \\ \frac{3}{10} & \frac{13}{105} \end{bmatrix} \times \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{12} \\ \frac{1}{20} \end{bmatrix}. \quad (2.44)$$

Observa-se que a matriz é simétrica porque a equação é de uma ordem uniforme e as funções aproximada e de ponderação são as mesmas.

Resolvendo o sistema, obtém-se:

$$\alpha_1 = \frac{71}{369} \quad \text{e} \quad \alpha_2 = \frac{7}{41}. \quad (2.45)$$

Substituindo esses valores em (2.35), tem-se a solução aproximada:

$$v = x(1-x) \left( \frac{71}{369} + \frac{7}{41}x \right). \quad (2.46)$$

E o resíduo pode ser escrito totalmente em função de  $x$ :

$$R = \frac{1}{369}(-16 + 62x - 8x^2 + 63x^3). \quad (2.47)$$

Os resultados para  $v$  e  $R$  são apresentados na Tabela 2.2, onde eles são comparados com a solução exata de  $u$ . É possível observar que, embora a solução seja no geral mais precisa do que no caso do método da colocação, no método de Galerkin é preciso realizar algumas integrações, e esta operação não foi necessária para a situação em que foi aplicada a técnica da colocação.

Outra vantagem presente na técnica de Galerkin é a obtenção de um sistema simétrico, como apresentado na equação (2.44), o que não acontece sempre no método da colocação, como é possível ver na equação (2.27). Além de expor essas diferenças, necessárias para

um melhor entendimento do MEC, o método dos resíduos ponderados será aplicado para a dedução da equação integral de contorno, como será visto na seção **4.2** deste trabalho.

Tabela 2.2: Resultados para Galerkin

$x$	$u$ (exato)	$v$ (aproximado)	$R$
0,1	0,018641	0,018853	-0,026945
0,3	0,051194	0,051162	+0,000485
0,5	0,069746	0,069444	+0,013888
0,7	0,065585	0,065505	+0,005070
0,9	0,030901	0,031146	-0,034165

# Capítulo 3

## A Equação de Laplace

Neste capítulo, a equação governante para problemas de condução de calor, que é a equação de Laplace, é deduzida através de considerações da física do problema. Em seguida, a solução fundamental da equação de Laplace é calculada. Então, nos próximos capítulos, a equação integral de contorno é deduzida e discretizada, obtendo-se a formulação do método dos elementos de contorno. Portanto, é neste capítulo que se dará início à descrição da formulação do MEC para problemas bidimensionais de condução de calor, que será mais detalhada no capítulo posterior. A redação deste capítulo foi baseada com grande influência nos trabalhos de Braga (2012) e Albuquerque (2012).

### 3.1 Equação de Poisson para transferência de calor

A condução de calor é regida pela lei de Fourier, uma equação determinada experimentalmente que relaciona o gradiente da temperatura com a taxa de condução de calor no tempo.

Para o caso da condução de calor unidirecional, conforme mostra a Figura 3.1,  $Q_x$  é a quantidade de calor transmitida na direção  $x$  e  $t$  é o tempo.

O fluxo de calor na direção  $x$  é dado por:

$$\dot{Q}_x = \frac{\partial Q_x}{\partial t}. \quad (3.1)$$

Assim, a energia térmica que flui através de um corpo de seção transversal  $A$ , é definida como:

$$\dot{Q}_x = -kA \frac{(u_1 - u_0)}{\Delta x}, \quad (3.2)$$

onde  $k$  é a condutividade térmica do material,  $A$  é a área e  $u_1$  e  $u_0$  são as temperaturas nas faces paralelas da placa.

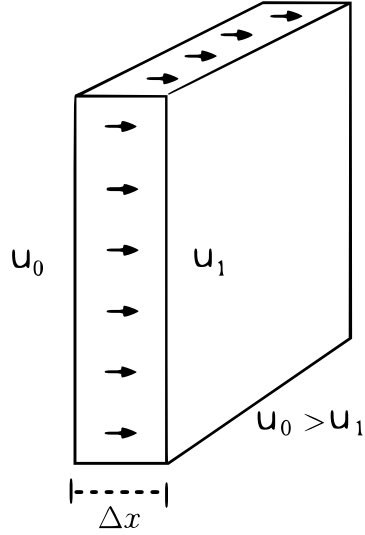


Figura 3.1: Transferência de calor de  $u_0$  para  $u_1$ .

Fazendo  $\Delta x \rightarrow 0$ , segue que:

$$\dot{Q}_x = -kA \lim_{\Delta x \rightarrow 0} \frac{u_1 - u_0}{\Delta x}, \quad (3.3)$$

da qual é obtida:

$$\dot{Q}_x = -kA \frac{\partial u}{\partial x}, \quad (3.4)$$

que é a equação da quantidade de calor transferida por condução na unidade de tempo.

Generalizando para um caso 3D e considerando a taxa de condução de calor por unidade de área, pode-se definir:

$$\vec{\dot{q}} = \frac{1}{dA} (\dot{Q}_x \vec{i} + \dot{Q}_y \vec{j} + \dot{Q}_z \vec{k}), \quad (3.5)$$

onde  $\dot{Q}_x$ ,  $\dot{Q}_y$  e  $\dot{Q}_z$  são os fluxos de calor nas direções  $x$ ,  $y$  e  $z$ , respectivamente. Assim

$$\begin{aligned} \vec{\dot{q}} &= \frac{1}{dA} \left[ -kdA \left( \frac{\partial u}{\partial x} \vec{i} + \frac{\partial u}{\partial y} \vec{j} + \frac{\partial u}{\partial z} \vec{k} \right) \right], \\ &= -k \left( \frac{\partial u}{\partial x} \vec{i} + \frac{\partial u}{\partial y} \vec{j} + \frac{\partial u}{\partial z} \vec{k} \right), \\ \vec{\dot{q}} &= -k \vec{\nabla} u, \end{aligned} \quad (3.6)$$

onde

$$\vec{\nabla} = \left( \frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k} \right) \quad (3.7)$$

é o operador gradiente.

Da primeira lei da termodinâmica, segue:

$$\dot{Q}_e + \dot{Q}_g = \dot{Q}_s + \dot{U}, \quad (3.8)$$

onde  $\dot{Q}_e$ ,  $\dot{Q}_g$  e  $\dot{Q}_s$  são os fluxos de calor que entra, que é gerado, e que sai do sistema,

respectivamente, e  $\dot{U}$  é a variação da energia térmica do sistema.

Para um caso unidimensional, a 1ª lei da termodinâmica é dada por:

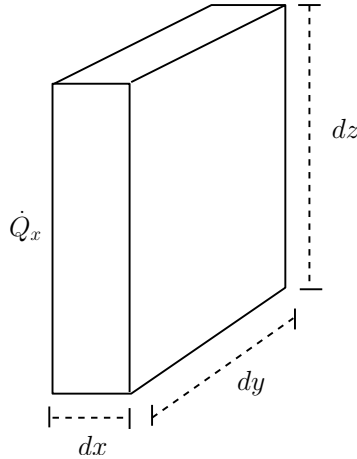


Figura 3.2: Fluxo de calor unidirecional.

$$\dot{Q}_x + \dot{q}_g dV = \left( \dot{Q} + \frac{\partial \dot{Q}}{\partial x} dx \right) + c\rho dV \dot{u}, \quad (3.9)$$

em que  $\dot{q}_g$  é a geração de calor por unidade de volume,  $c$  o calor específico,  $\rho$  a densidade do material e  $\dot{u} = \frac{\partial u}{\partial t}$ .

Isolando  $\dot{q}_g dV$  e considerando a temperatura como estacionária, isto é,  $\dot{u} = 0$ , segue:

$$\dot{q}_g dV = \frac{\partial \dot{Q}}{\partial x} dx. \quad (3.10)$$

Substituindo a Eq.(3.4) na Eq.(3.10), obtém-se:

$$\dot{q}_g dx dy dz = \frac{\partial}{\partial x} \left( -kA \frac{\partial u}{\partial x} \right) dx. \quad (3.11)$$

Como  $A = dy dz$  e considerando  $k$  constante em todo material, tem-se:

$$\dot{q}_g = -k \frac{\partial^2 u}{\partial x^2}. \quad (3.12)$$

Expandindo para o caso 3D, segue:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = -\frac{\dot{q}_g}{k}, \quad (3.13)$$

que é conhecida como equação de Poisson para condução de calor. Esta equação ainda pode ser escrita como:

$$\nabla^2 u = -\frac{\dot{q}_g}{k}, \quad (3.14)$$

onde

$$\nabla^2 = \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) \quad (3.15)$$

é o operador laplaciano.

Quando  $\dot{q}_g = 0$ , a equação de Poisson se reduz a  $\nabla^2 u = 0$ , que é conhecida como equação de Laplace.

### 3.1.1 Solução fundamental para a equação de Laplace

A solução fundamental, que é a base da formulação do MEC, para a equação de Laplace, corresponde à resposta da temperatura em um meio infinito quando a fonte de geração de calor é concentrada em um ponto. Matematicamente, ela corresponde à solução particular analítica da equação de Poisson quando o termo não homogêneo (termo referente à geração de calor) é igual ao delta de Dirac, ou seja:

$$\nabla^2 u^* = -\frac{\delta(x - x_d, y - y_d)}{k}. \quad (3.16)$$

Uma função que satisfaz essa equação é dada por:

$$u^* = A \ln r, \quad (3.17)$$

em que  $r$  é a distância entre o ponto onde a fonte de calor é aplicada (ponto fonte) e o ponto onde a temperatura é medida (ponto campo) e  $A$  é uma constante a se determinar.

Sem perder generalidade, considere o sistema de coordenadas com origem no ponto fonte, conforme a Fig. 3.3.

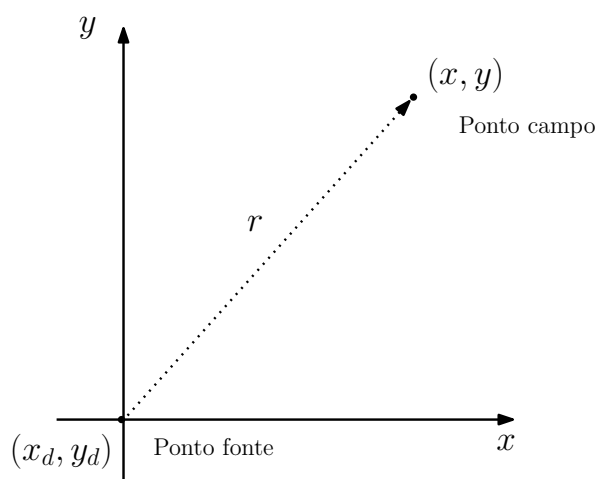


Figura 3.3: Distância  $r$  entre o ponto fonte e o ponto campo.

Da figura, pode-se notar

$$r = \sqrt{x^2 + y^2}. \quad (3.18)$$

Como

$$u^* = A \ln r,$$

$$\frac{\partial u^*}{\partial x} = A \frac{\partial(\ln r)}{\partial r} \frac{\partial r}{\partial x}. \quad (3.19)$$

Calculando o termo  $\frac{\partial r}{\partial x}$ , segue

$$\frac{\partial r}{\partial x} = \frac{\partial \sqrt{x^2 + y^2}}{\partial x} = \frac{x}{\sqrt{x^2 + y^2}} = \frac{x}{r}. \quad (3.20)$$

Assim:

$$\frac{\partial u^*}{\partial x} = A \frac{1}{r} \frac{x}{r} = \frac{Ax}{r^2}. \quad (3.21)$$

Da mesma forma, pode-se mostrar que:

$$\frac{\partial^2 u^*}{\partial x^2} = \frac{A}{r^2} - 2 \frac{Ax^2}{r^4}, \quad (3.22)$$

$$\frac{\partial u^*}{\partial y} = \frac{Ay}{r^2} \quad (3.23)$$

e

$$\frac{\partial^2 u^*}{\partial y^2} = \frac{A}{r^2} - 2 \frac{Ay^2}{r^4}. \quad (3.24)$$

Daí, segue que:

$$\frac{\partial^2 u^*}{\partial x^2} + \frac{\partial^2 u^*}{\partial y^2} = A \left( \frac{1}{r^2} - \frac{2x^2}{r^4} + \frac{1}{r^2} - \frac{2y^2}{r^4} \right), \quad (3.25)$$

$$\nabla^2 u^* = 2A \underbrace{\left( \frac{1}{r^2} - \frac{x^2 + y^2}{r^4} \right)}_{\delta(x-0)}. \quad (3.26)$$

Como  $\nabla^2 u^* = 0$  para qualquer ponto, exceto o ponto de aplicação da fonte, segue que:

$$\nabla^2 u^* = 2A\delta(x-0). \quad (3.27)$$

Igualando as equações (3.16) e (3.27) com  $(x_d, y_d) = (0, 0)$ , segue:

$$\frac{\delta(x-0)}{k} = 2A\delta(x-0) \implies 2A = -\frac{1}{k}, \quad (3.28)$$

da qual obtém-se o valor de  $A$ , dado por:

$$A = -\frac{1}{2k}. \quad (3.29)$$

Substituindo o valor de  $A$  na Eq.(3.17), segue que:

$$u^* = \frac{-1}{2k} \ln r. \quad (3.30)$$

Contudo, a fim de obter uma formulação numérica esteticamente melhor, é comum considerar:

$$u^* = \frac{-1}{2\pi k} \ln r. \quad (3.31)$$

Considerando que o sistema de referência tem origem em uma posição qualquer,  $r$  é dado por:

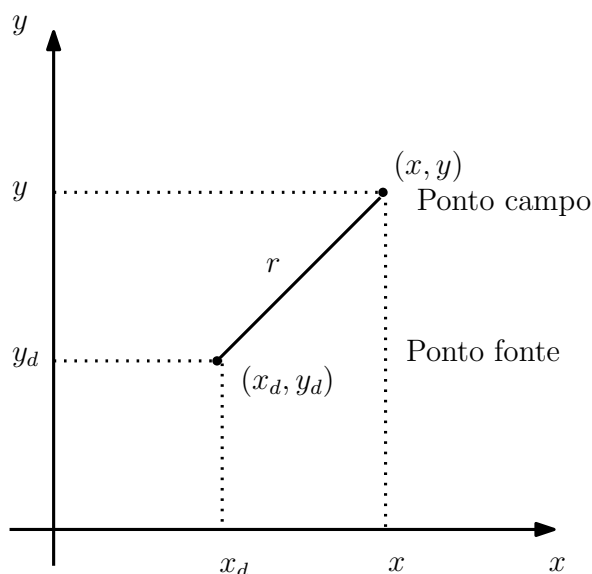


Figura 3.4: Posições dos pontos fonte e campo.

$$r = \sqrt{(x - x_d)^2 + (y - y_d)^2}, \quad (3.32)$$

onde  $(x_d, y_d)$  são coordenadas do ponto fonte.

### 3.1.2 Fluxo de calor através do contorno

Define-se fluxo de calor através do contorno pela expressão

$$q = \vec{q} \cdot \vec{n}, \quad (3.33)$$

onde  $q$  é a quantidade de calor que passa através do contorno por unidade de tempo e por unidade de área. Substituindo na Eq.(3.33) o valor de  $\vec{q}$  dado pela Eq.(3.6), obtém-se

$$q = \vec{q} \cdot \vec{n} = -k \vec{\nabla} \cdot \dots u \vec{n}, \quad (3.34)$$



$$q = -k \frac{\partial u}{\partial n}, \quad (3.35)$$

onde  $\frac{\partial u}{\partial n}$  é a derivada da temperatura na direção do vetor normal ao contorno  $\vec{n}$ . Dessa forma, é possível definir a solução fundamental para o fluxo de calor, dada por:

$$q^* = -k \frac{\partial u^*}{\partial n}. \quad (3.36)$$

Substituindo na Eq.(3.36) o valor de  $u^*$  dado pela Eq.(3.31), segue:

$$q^* = -k \frac{\partial}{\partial n} \left( -\frac{1}{2\pi k} \ln r \right), \quad (3.37)$$

$$q^* = \frac{1}{2\pi} \left[ \frac{\partial}{\partial x} (\ln r) n_x + \frac{\partial}{\partial y} (\ln r) n_y \right]. \quad (3.38)$$

Calculando o termo  $\frac{\partial}{\partial x} (\ln r)$ , obtém-se:

$$\frac{\partial}{\partial x} (\ln r) = \frac{\partial \ln r}{\partial r} \frac{\partial r}{\partial x}. \quad (3.39)$$

Substituindo na Eq.(3.39) o valor de  $r$  dado pela Eq.(3.32), tem-se:

$$\begin{aligned} \frac{\partial}{\partial x} (\ln r) &= \frac{1}{r} \frac{\partial}{\partial x} [(x - x_d)^2 + (y - y_d)^2]^{\frac{1}{2}}, \\ &= \frac{1}{r} \frac{(x - x_d)}{[(x - x_d)^2 + (y - y_d)^2]^{\frac{1}{2}}}, \\ &= \frac{1}{r} \frac{(x - x_d)}{r}, \\ \frac{\partial}{\partial x} (\ln r) &= \frac{x - x_d}{r^2}. \end{aligned} \quad (3.40)$$

De modo análogo, tem-se:

$$\frac{\partial}{\partial y} (\ln r) = \frac{y - y_d}{r^2}. \quad (3.41)$$

Substituindo a Eq.(3.40) e a Eq.(3.41) na Eq.(3.38), segue:

$$q^* = \frac{1}{2\pi r^2} [(x - x_d)n_x + (y - y_d)n_y], \quad (3.42)$$

que é a solução fundamental de fluxo.

# Capítulo 4

## Equações Integrais de Contorno

Este capítulo tem por finalidade mostrar as deduções das equações integrais de contorno, tanto regular quanto hipersingular. Essa verificação é feita a partir do desenvolvimento da equação integral de contorno para a equação de Laplace. Esta equação é posteriormente discretizada em elementos de contorno, obtendo-se então a formulação do MEC. A dedução da equação integral de contorno regular foi baseada nos trabalhos de Braga (2012) e Albuquerque (2012).

### 4.1 Equações integrais de contorno

#### 4.1.1 Equação regular

Foi demonstrado no capítulo 3 que a solução fundamental para a equação de Laplace é:

$$u^* = \frac{-1}{2\pi k} \ln r, \quad (4.1)$$

e a solução fundamental para o fluxo é dada por:

$$q^* = \frac{1}{2\pi r^2} [(x - x_d)n_x + (y - y_d)n_y]. \quad (4.2)$$

Dada a equação de Laplace

$$\nabla^2 u = 0, \quad (4.3)$$

multiplicando a equação (4.3) por uma função peso  $\omega(x, y)$  e integrando sobre o domínio  $A$ , assume-se que o resultado da integral é zero (método dos resíduos ponderados). Assim,

tem-se:

$$\begin{aligned}\iint_A (\nabla^2 u) \omega dA &= 0, \\ \iint_A \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \omega dA &= 0, \\ \iint_A \frac{\partial^2 u}{\partial x^2} \omega dA + \iint_A \frac{\partial^2 u}{\partial y^2} \omega dA &= 0.\end{aligned}\tag{4.4}$$

Pelo teorema de Gauss-Green, tem-se:

$$\int_s f(x, y) n_x ds = \int_A \frac{\partial f}{\partial x} dA,\tag{4.5}$$

onde  $f$  é uma função qualquer,  $n_x$  é a componente na direção  $x$  do vetor  $\vec{n}$  normal ao contorno  $s$  da área  $A$ . Aplicando o teorema dado na primeira parcela da Eq.(4.4), tem-se:

$$\int_s \frac{\partial u}{\partial x} \omega n_x ds = \int_A \frac{\partial}{\partial x} \left( \frac{\partial u}{\partial x} \omega \right) dA.\tag{4.6}$$

Aplicando a regra da derivada do produto de funções, tem-se:

$$\int_s \frac{\partial T}{\partial x} \omega n_x ds = \int_A \frac{\partial^2 u}{\partial x^2} \omega dA + \int_A \frac{\partial u}{\partial x} \frac{\partial \omega}{\partial x} dA.\tag{4.7}$$

Escrevendo os termos da equação de forma conveniente, segue:

$$\int_A \frac{\partial^2 u}{\partial x^2} \omega dA = \int_s \frac{\partial u}{\partial x} \omega n_x ds - \int_A \frac{\partial u}{\partial x} \frac{\partial \omega}{\partial x} dA.\tag{4.8}$$

De modo análogo, obtém-se:

$$\int_A \frac{\partial^2 u}{\partial y^2} \omega dA = \int_s \frac{\partial u}{\partial y} \omega n_y ds - \int_A \frac{\partial u}{\partial y} \frac{\partial \omega}{\partial y} dA.\tag{4.9}$$

Substituindo as Eqs.(4.8) e (4.9) na Eq.(4.4), tem-se:

$$\int_s \left( \frac{\partial u}{\partial x} \omega n_x + \frac{\partial u}{\partial y} \omega n_y \right) ds - \int_A \left( \frac{\partial u}{\partial x} \frac{\partial \omega}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial \omega}{\partial y} \right) dA = 0.\tag{4.10}$$

Simplificando:

$$\int_s \frac{\partial u}{\partial n} \omega ds - \int_A \left( \frac{\partial u}{\partial x} \frac{\partial \omega}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial \omega}{\partial y} \right) dA = 0.\tag{4.11}$$

Considerando as igualdades:

$$\int_A \frac{\partial u}{\partial x} \frac{\partial \omega}{\partial x} dA = \int_s u \frac{\partial \omega}{\partial x} n_x ds - \int_A u \frac{\partial^2 \omega}{\partial x^2} dA \quad (4.12)$$

e

$$\int_A \frac{\partial u}{\partial y} \frac{\partial \omega}{\partial y} dA = \int_s u \frac{\partial \omega}{\partial y} n_y ds - \int_A u \frac{\partial^2 \omega}{\partial y^2} n_y dA. \quad (4.13)$$

Substituindo as equações (4.12) e (4.13) na equação (4.11), tem-se:

$$\int_s \frac{\partial u}{\partial n} \omega ds - \int_s \left( u \frac{\partial \omega}{\partial x} n_x + u \frac{\partial \omega}{\partial y} n_y \right) ds + \int_A u \left( \frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right) dA = 0, \quad (4.14)$$

$$\int_s \frac{\partial u}{\partial n} \omega ds - \int_s u \frac{\partial \omega}{\partial n} ds + \int_A u \nabla^2 \omega dA = 0. \quad (4.15)$$

Com o objetivo de obter uma equação integral que não possua integrais de domínio (integrais de área), a função  $\omega$  deve ser escolhida de forma que a integral de domínio da equação desapareça. Qualquer função harmônica, ou seja, função que o laplaciano é igual a zero, satisfaz essa exigência. Contudo, por razões numéricas, a escolha mais adequada é a função cujo laplaciano é o delta de Dirac.

$$\nabla^2 \omega = -\frac{\delta(x - x_d)}{k}, \quad (4.16)$$

o que implica que  $\omega = u^*$ . Então tem-se:

$$\int_s \frac{\partial u}{\partial n} u^* ds - \int_s u \frac{\partial u^*}{\partial n} ds + \int_A u \frac{[-\delta(x - x_d)]}{k} dA, \quad (4.17)$$

onde  $x_d$  é a coordenada do ponto fonte. Tomando o ponto fonte no interior do domínio  $A$ , pela propriedade do delta de Dirac, tem-se:

$$\int_s \frac{\partial u}{\partial n} u^* ds - \int_s u \frac{\partial u^*}{\partial n} ds - \frac{u(x_d, y_d)}{k} = 0. \quad (4.18)$$

Multiplicando os termos por  $-k$ , obtém-se:

$$\int -k \frac{\partial u}{\partial n} u^* ds + \int_s u \left( \frac{k \partial u^*}{\partial n} \right) ds + u(x_d, y_d) = 0, \quad (4.19)$$

$$u(x_d, y_d) = \int_s u q^* ds - \int_s q u^* ds. \quad (4.20)$$

A Eq.(4.20) é a *equação integral de contorno* quando o ponto fonte encontra-se no interior do domínio.

A fim de considerar o ponto  $(x_d, y_d)$  no contorno, faz-se uma pequena modificação no

mesmo, conforme Fig. 4.1:

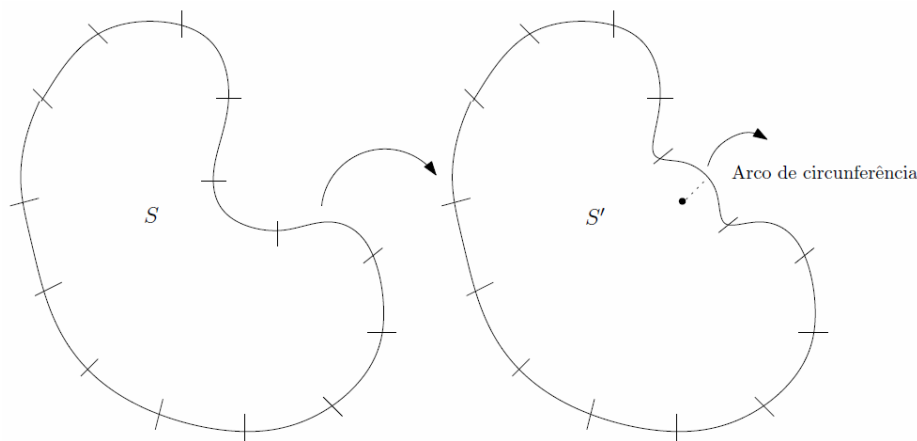


Figura 4.1: Contornos original e adaptado.

Assim, tem-se:

$$u(x_d, y_d) = \int_{s-s^*} uq^* ds - \int_{s-s^*} u^* q ds + \int_{s^*} uq^* ds - \int_{s^*} u^* q ds. \quad (4.21)$$

Conforme visto na Eq.(4.2):

$$q^* = \frac{1}{2\pi r^2} [(x - x_d)n_x + (y - y_d)n_y],$$

com

$$r = [(x - x_d)^2 + (y - y_d)^2],$$

logo:

$$\int_{s^*} uq^* ds = \int_{\theta_1}^{\theta_2} u \frac{1}{2\pi r^2} [(x - x_d)n_x + (y - y_d)n_y] ds. \quad (4.22)$$

Em \$s^\*\$, tem-se:

$$\vec{r} = (x - x_d)\vec{i} + (y - y_d)\vec{j}, \quad (4.23)$$

$$|\vec{r}| = r = \sqrt{(x - x_d)^2 + (y - y_d)^2}, \quad (4.24)$$

$$\vec{n} = \frac{(x - x_d)\vec{i} + (y - y_d)\vec{j}}{r}, \quad (4.25)$$

onde \$\vec{n}\$ é um vetor unitário. Quando \$r\_x = (x - x\_d)\$ e \$r\_y = (y - y\_d)\$ tem-se:

$$\vec{n} = \frac{r_x\vec{i} + r_y\vec{j}}{r} \quad (4.26)$$

com

$$n_x = \frac{r_x}{r} \quad \text{e} \quad n_y = \frac{r_y}{r}. \quad (4.27)$$

Assim:

$$\int_{s^*}^s uq^* ds = \int_{\theta_1}^{\theta_2} \frac{u}{2\pi r^2} \left( r_x \frac{r_x}{r} + r_y \frac{r_y}{r} \right) \varepsilon d\theta. \quad (4.28)$$

Observando que  $r = \varepsilon$  para qualquer  $\theta$ , portanto:

$$\int_{s^*}^s uq^* ds = \int_{\theta_1}^{\theta_2} \frac{u}{2\pi \varepsilon^2} \left( \frac{r_x^2 + r_y^2}{\varepsilon} \right) \varepsilon d\theta = \int_{\theta_1}^{\theta_2} \frac{u}{2\pi} d\theta. \quad (4.29)$$

Fazendo  $\varepsilon \rightarrow 0$ ,  $u$  assume o valor de  $u(d)$ . Por fim, tem-se:

$$\int_s uq^* ds = \frac{u(d)(\theta_2 - \theta_1)}{2\pi}. \quad (4.30)$$

A mesma análise deve ser feita para:

$$\int_{s^*} u^* q ds = \int_{\theta_1}^{\theta_2} \frac{-1}{2\pi r} \ln r q r d\theta. \quad (4.31)$$

Como  $r = \varepsilon = \text{constante}$ , tem-se:

$$\int_{s^*} u^* q ds = \frac{-1}{2\pi k} \varepsilon \ln \varepsilon \int_{\theta_1}^{\theta_2} q d\theta. \quad (4.32)$$

Fazendo  $\varepsilon \rightarrow 0$ , tem-se:

$$\int_{s^*} u^* q ds = \frac{-1}{2\pi k} \lim_{\varepsilon \rightarrow 0} \varepsilon \ln \varepsilon (\theta_2 - \theta_1), \quad (4.33)$$

$$\int_{s^*} u^* q ds = 0. \quad (4.34)$$

Voltando a equação original, segue:

$$u(x_d, y_d) = \int_s uq^* ds - \int_s u^* q ds + \frac{u(x_d, y_d)(\theta_2 - \theta_1)}{2\pi} - 0, \quad (4.35)$$

$$u(x_d, y_d) \left[ 1 - \frac{(\theta_2 - \theta_1)}{2\pi} \right] = \int_s uq^* ds - \int_s u^* q ds, \quad (4.36)$$

$$u(x_d, y_d) \left[ \frac{2\pi - (\theta_2 - \theta_1)}{2\pi} \right] = \int_s uq^* ds - \int_s u^* q ds. \quad (4.37)$$

Conforme mostra a Fig. 3.2,  $\theta_{int}$  é o ângulo interno do contorno. Assim:

$$\frac{\theta_{int}}{2\pi} u(x_d, y_d) = \int_s uq^* ds - \int_s u^* q ds, \quad (4.38)$$

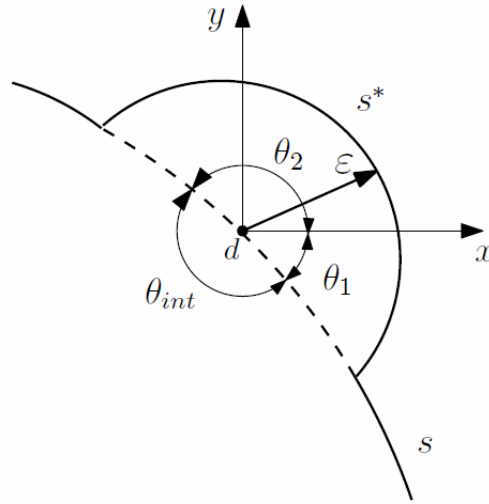


Figura 4.2: Ângulo interno do contorno.

que é a integral de contorno quando o ponto fonte pertence ao contorno.

Quando o ponto fonte não pertence ao contorno nem ao domínio, devido à propriedade do delta de Dirac, tem-se:

$$\int_s uq^* ds - \int_s u^* q ds = 0. \quad (4.39)$$

Generalizando, a equação integral de contorno pode ser escrita como:

$$cu(x_d, y_d) = \int_s uq^* ds - \int_s u^* q ds, \quad (4.40)$$

onde

$$c = \begin{cases} 1, & \text{se } (x_d, y_d) \in \text{ao domínio.} \\ \frac{\theta_{int}}{2\pi}, & \text{se } (x_d, y_d) \in \text{ao contorno.} \\ 0, & \text{se } (x_d, y_d) \notin \text{ao domínio ou ao contorno.} \end{cases}$$

Quando o ponto fonte encontra-se em ponto suave do contorno, isto é, não é um canto, tem-se:

$$c = \frac{\theta_{int}}{2\pi} = \frac{\pi}{2\pi} = \frac{1}{2}. \quad (4.41)$$

### 4.1.2 Equação hipersingular

A equação hipersingular é obtida derivando a equação fracamente singular em relação à posição do ponto fonte  $(x_d, y_d)$ :

$$c \frac{\partial u}{\partial m}(x_d, y_d) = \int_s \frac{\partial q^*}{\partial m} u ds - \int_s \frac{\partial u^*}{\partial m} q ds, \quad (4.42)$$

em que  $\vec{m}$  é o vetor normal no ponto fonte.

Esta equação é necessária para a obtenção de um sistema linear simétrico, como será demonstrado em capítulos posteriores.

Para usar a equação hipersingular, o ponto de colocação não pode ser um canto pois, neste caso, a integral não existe no sentido do valor principal de Hadamard (LIU, 2009).

Analisando os componentes núcleos da Eq.(4.42), tem-se:

$$\frac{\partial u^*}{\partial m} = \frac{1}{2\pi r} r_{,k} m_k \quad (4.43)$$

e

$$\frac{\partial q^*}{\partial m} = \frac{1}{2\pi r^2} [n_k m_k - 2r_{,k} m_k r_{,l} n_l]. \quad (4.44)$$

O termo núcleo na Eq.(4.43) é fortemente singular, e a segunda integral da Eq.(4.42) é uma integral no valor principal de Cauchy. O termo núcleo da Eq. (4.44) é hipersingular, e a primeira integral da Eq.(4.42) é uma integral de parte finita de Hadamard.



# Capítulo 5

## Integração Analítica das Matrizes $\mathbf{H}$ e $\mathbf{G}$

A fim de apresentar as matrizes de influência  $\mathbf{H}$  e  $\mathbf{G}$ , primeiramente é realizada a discretização do contorno e, posteriormente, a discretização utilizando elementos constantes e elementos lineares contínuos, para que então seja possível efetuar a integração analítica dessas matrizes para ambos os casos.

A realização da integração analítica é altamente benéfica, visto que ela trata automaticamente as singularidades e é exata. O contraponto é a alta complexidade de sua elaboração. Como neste trabalho os tipos de elementos que apresentam mais alto grau de dificuldade são os retilíneos, que são relativamente simples, sua aplicação pôde ser contemplada.

A discretização das equações integrais de contorno foi baseada nos trabalhos de Braga (2012) e Albuquerque (2012). A integração analítica com elementos constantes foi feita por Braga (2012) e a integração analítica com elementos lineares por Albuquerque e Izquierdo (2014).

### 5.1 Discretização das equações integrais de contorno

Basicamente, a formulação do MEC transforma equações diferenciais em equações integrais no contorno, eliminando, assim, a discretização do domínio. Essas integrais podem ser resolvidas de forma numérica e analítica com a integração feita ao longo do contorno, que é discretizado através da divisão do mesmo em elementos, chamados *elementos de contorno*, nos quais as condições de contorno são prescritas.

Uma vez obtida a integral de contorno, o próximo passo é a discretização desta equação, de forma que as integrais ao longo deste contorno sejam escritas como a soma das partes deste contorno, ou seja:

$$S = s_1 + s_2 + \dots + s_n.$$

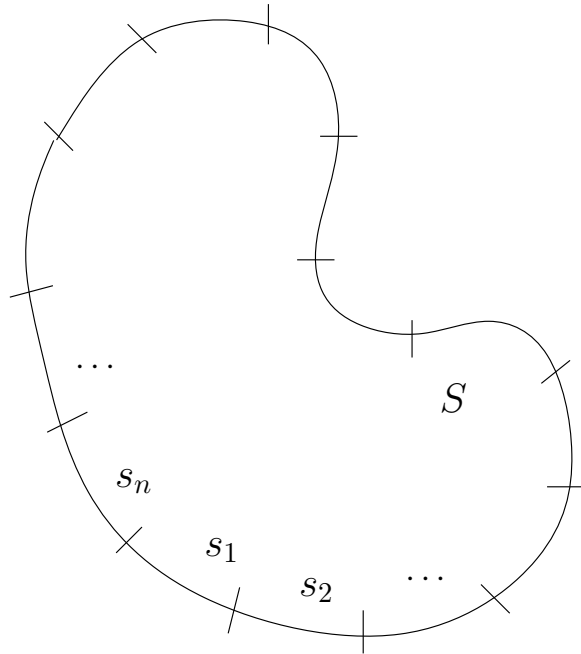


Figura 5.1: Discretização do contorno em  $n$  partes.

Desta forma, a equação integral de contorno (4.40) é escrita como:

$$cu(x_d, y_d) = \sum_{j=1}^n \int_{s_j} uq^* ds - \sum_{j=1}^n \int_{s_j} u^* q ds. \quad (5.1)$$

## 5.2 Elementos de contorno constantes

Na discretização utilizando elementos constantes, a geometria é aproximada por segmentos de retas com um nó no meio de cada elemento. Dessa forma, considere que as partes do contorno  $s_1, s_2, \dots, s_n$ , sejam aproximadas por segmentos retilíneos e que ao longo destes segmentos, tanto a temperatura quanto o fluxo sejam assumidos como constantes.

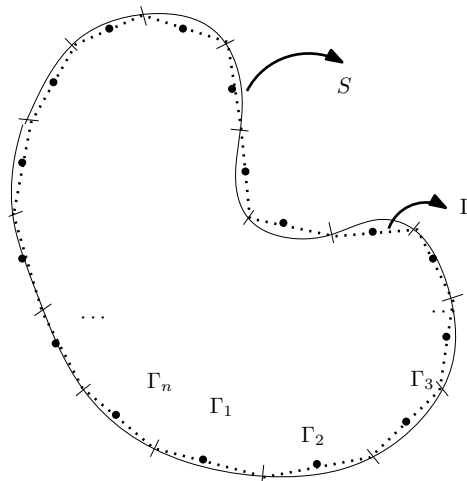


Figura 5.2: Aproximação do contorno por segmentos de retas.

O nó  $j$  sempre estará na posição central do elemento  $j$  (será sempre em uma região suave do contorno, portanto  $c = \frac{1}{2}$ ). A equação integral é aproximada por:

$$\frac{1}{2}u^{(i)}(x_d, y_d) = \sum_{j=1}^n \left[ u_j \int_{\Gamma_j} q^* d\Gamma \right] - \sum_{j=1}^n \left[ q_j \int_{\Gamma_j} u^* d\Gamma \right], \quad (5.2)$$

onde  $i$  corresponde ao nó do  $i$ -ésimo elemento. Daí, tem-se:

$$-\frac{1}{2}u^{(i)}(x_d, y_d) + \sum_{j=1}^n \left[ u_j \int_{\Gamma_j} q^* d\Gamma \right] = \sum_{j=1}^n \left[ q_j \int_{\Gamma_j} u^* d\Gamma \right]. \quad (5.3)$$

Chamando

$$H_{ij} = \begin{cases} \int_{\Gamma_j} q^* d\Gamma, & \text{se } i \neq j \\ -\frac{1}{2} + \int_{\Gamma_j} q^* d\Gamma, & \text{se } i = j \end{cases} \quad (5.4)$$

e

$$G_{ij} = \int_{\Gamma_j} u^* d\Gamma, \quad (5.5)$$

pode-se escrever a equação matricial da seguinte forma:

$$\sum_{j=1}^n [H_{ij}u_j] = \sum_{j=1}^n [G_{ij}q_j]. \quad (5.6)$$

## 5.2.1 Integração analítica com elementos constantes

Conforme mostrado por Banerjee (1994), a integração analítica dos elementos das matrizes  $[H]$  e  $[G]$  pode ser feita facilmente considerando um sistema de coordenadas  $x'y'$  com origem no ponto fonte e com o eixo  $y'$  paralelo ao elemento (Figura 5.3). Desta forma, das Eqs. (4.1) e (5.5), tem-se:

$$G_{ij} = \int_{\Gamma_j} u^* d\Gamma = \frac{-1}{2\pi k} \int_{\Gamma_j} \ln r d\Gamma. \quad (5.7)$$

Da Fig. 5.3, tem-se que:

$$\begin{cases} r = & h \sec \theta \\ \frac{\partial r}{\partial n} = & r_x n_x + r_y n_y = r \cos \theta = h \\ x' = & r \cos \theta \\ y' = & r \sin \theta \\ dy' = & dS = h \sec^2 \theta d\theta \end{cases}$$

$$G_{ij} = \frac{-1}{2\pi k} \int_{\Delta S} \ln r dy' = \frac{-1}{2\pi k} \int_{\theta_A}^{\theta_B} \ln(h \sec \theta) h \sec^2 \theta d\theta, \quad (5.8)$$

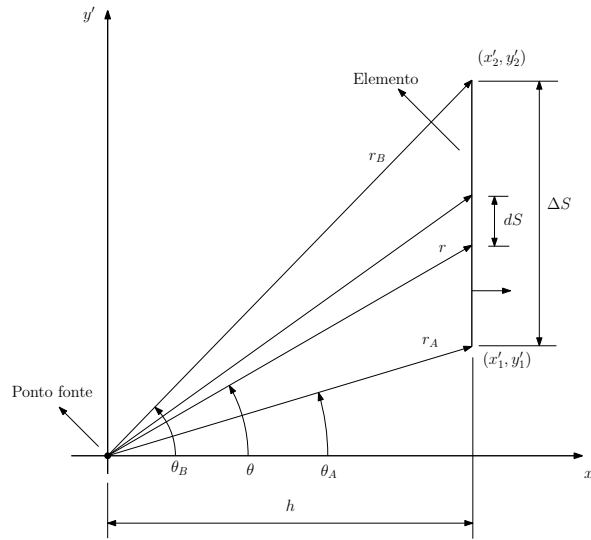


Figura 5.3: Sistema de referências usado para o cálculo das integrais analíticas.

$$G_{ij} = \frac{-h}{2\pi k} \left[ \tan \theta (\ln |(h \sec \theta) - 1|) + \theta \right] \Big|_{\theta_A}^{\theta_B}. \quad (5.9)$$

Da mesma forma, tem-se que:

$$H_{ij} = \int_{\Gamma_j} q^* d\Gamma = \frac{1}{2\pi} \int_{\Gamma_j} \frac{r_x n_x + r_y n_y}{r^2} dy' = \frac{1}{2\pi} \int_{\Delta S} \frac{r \cos \theta}{r^2} h \sec^2 \theta d\theta, \quad (5.10)$$

$$H_{ij} = \frac{1}{2\pi} \int_{\theta_A}^{\theta_B} \frac{h \sec \theta \cos \theta}{h^2 \sec^2 \theta} h \sec^2 \theta d\theta = \frac{1}{2\pi} \int_{\theta_A}^{\theta_B} d\theta = \frac{1}{2\pi} [\theta] \Big|_{\theta_A}^{\theta_B}. \quad (5.11)$$

### 5.3 Elementos de contorno lineares contínuos

Na discretização por elementos lineares, a geometria é aproximada por um polinômio do 1º grau, necessitando de dois nós em cada elemento, um em cada extremidade do elemento. A temperatura e o fluxo também são aproximados por um polinômio de 1º grau. A formulação é isoparamétrica, ou seja, as mesmas funções de forma usadas para interpolar a geometria são também usadas para interpolar as variáveis físicas (temperatura e fluxo).

Neste caso, a equação integral é dada por:

$$cu(x_d, y_d) = \int_s uq^* dS - \int_s u^* qdS.$$

Discretizando em elementos de contorno lineares contínuos, segue:

$$cu(x_d, y_d) = \sum_{j=1}^{n_{elem}} \left[ \int_{\Gamma_j} uq^* d\Gamma \right] - \sum_{j=1}^{n_{elem}} \left[ \int_{\Gamma_j} u^* qd\Gamma \right]. \quad (5.12)$$

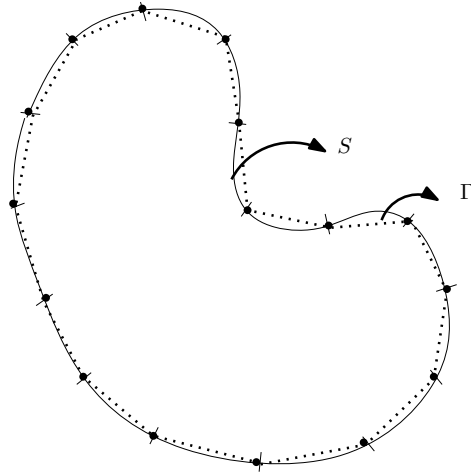


Figura 5.4: Elementos lineares contínuos.

Observando-se que  $u$  e  $q$  são assumidos com variação linear ao longo do elemento, ou seja,

$$u = N_1 u_1 + N_2 u_2 \quad (5.13)$$

e

$$q = N_1 q_1 + N_2 q_2, \quad (5.14)$$

onde  $u_1$  é a temperatura no nó local 1,  $u_2$  a temperatura no nó local 2,  $q_1$  é o fluxo no nó local 1 e  $q_2$  é o fluxo no nó local 2,  $N_1$  é a função de forma 1 e  $N_2$  é a função de forma 2.

Da mesma forma, segue:

$$\begin{cases} x = N_1 x_1 + N_2 x_2 \\ y = N_1 y_1 + N_2 y_2. \end{cases} \quad (5.15)$$

Escrevendo na forma matricial, segue:

$$u = \begin{bmatrix} N_1 & N_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (5.16)$$

e

$$q = \begin{bmatrix} N_1 & N_2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}. \quad (5.17)$$

A equação integral discretizada é então escrita como:

$$cu(x_d, y_d) = \sum_{j=1}^{n_{elem}} \left\{ \int_{\Gamma} \begin{bmatrix} N_1 & N_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_j q^* d\Gamma \right\} - \sum_{j=1}^{n_{elem}} \left\{ \int_{\Gamma} u^* \begin{bmatrix} N_1 & N_2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}_j d\Gamma \right\}. \quad (5.18)$$

Como  $u_1, u_2, q_1$  e  $q_2$  são valores nodais, segue:

$$cu(x_d, y_d) = \sum_{j=1}^{n_{elem}} \left\{ \int_{\Gamma_j} \begin{bmatrix} N_1 & N_2 \end{bmatrix} q^* d\Gamma \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_j \right\} - \sum_{j=1}^{n_{elem}} \left\{ \int_{\Gamma_j} \begin{bmatrix} N_1 & N_2 \end{bmatrix} u^* d\Gamma \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}_j \right\}, \quad (5.19)$$

que pode ser escrito da seguinte forma:

$$cu(x_d, y_d) = \sum_{j=1}^{n_{elem}} \left\{ \begin{bmatrix} h_1 & h_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_j - \begin{bmatrix} g_1 & g_2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}_j \right\}, \quad (5.20)$$

onde

$$h_1 = \int_{\Gamma_j} N_1 q^* d\Gamma, \quad (5.21)$$

$$h_2 = \int_{\Gamma_j} N_2 q^* d\Gamma \quad (5.22)$$

e

$$g_1 = \int_{\Gamma_j} N_1 u^* d\Gamma, \quad (5.23)$$

$$g_2 = \int_{\Gamma_j} N_2 u^* d\Gamma. \quad (5.24)$$

A Eq. (5.20) pode ser escrita como:

$$-cu_i + \begin{bmatrix} \widehat{H}_{i1} & \widehat{H}_{i2} & \dots & \widehat{H}_{iNE} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{NE} \end{Bmatrix} = \begin{bmatrix} G_{i1} & G_{i2} & \dots & G_{i2NE} \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{2NE} \end{Bmatrix}, \quad (5.25)$$

na qual

$$\widehat{H}_{ij} = h_1^j + h_2^{j-1}. \quad (5.26)$$

Portanto, a Eq.(5.25) pode ser escrita como

$$\sum_{j=1}^{NE} H_{ij} u_j = \sum_{j=1}^{2NE} G_{ij} q_j, \quad (5.27)$$

onde o termo  $H_{ij}$  é dado por

$$H_{ij} = \begin{cases} \widehat{H}_{ij}, & \text{se } i \neq j \\ -c + \widehat{H}_{ij}, & \text{se } i = j. \end{cases} \quad (5.28)$$

A Eq.(5.25) pode ser expressa em forma matricial como

$$\mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{q}, \quad (5.29)$$

onde  $\mathbf{H}$  e  $\mathbf{G}$  são conhecidas como as matrizes de influência do método dos elementos de contorno.

### 5.3.1 Integração analítica com elementos lineares contínuos

Nesta seção, para calcular os coeficientes  $H_{ij}$  e  $G_{ij}$ , as integrais de contorno ao longo de qualquer elemento  $j$  são avaliadas analiticamente por razões computacionais.

Assim como foi feito para elementos constantes na seção 5.2.1, foi considerado um novo sistema de coordenadas para a integração analítica dos elementos das matrizes  $[H]$  e  $[G]$ , como mostrado na Fig. 5.5.

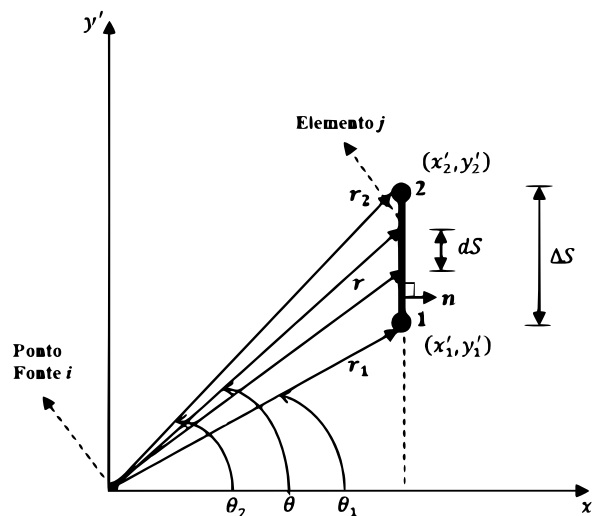


Figura 5.5: Sistema de coordenadas local.

Da Fig. 5.5, pode-se obter as seguintes relações:

$$x' = r \cos \theta, \quad (5.30)$$

$$y' = r \sin \theta, \quad (5.31)$$

$$y_1' = r_1 \sin \theta_1, \quad (5.32)$$

$$y'_2 = r_2 \sin \theta_2. \quad (5.33)$$

As relações obtidas a partir da projeção escalar dos vetores  $\mathbf{r}$ ,  $\mathbf{r}_1$  e  $\mathbf{r}_2$  no vetor normal unitário  $\mathbf{n}$  e no vetor unitário tangente  $\mathbf{t}$ , como mostrado na Fig. 5.6, são definidas como

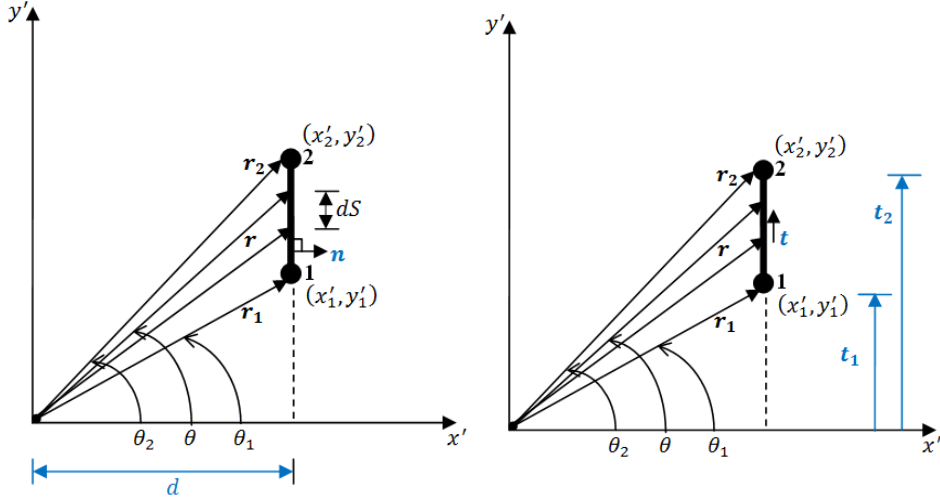


Figura 5.6: Projeção escalar dos vetores  $\mathbf{r}$ ,  $\mathbf{r}_1$  e  $\mathbf{r}_2$  no vetor normal unitário e no vetor tangente unitário.

$$d = \frac{\mathbf{r} \cdot \mathbf{n}}{|\mathbf{n}|} = r \cos \theta, \quad (5.34)$$

$$r = d \sec \theta, \quad (5.35)$$

$$r_1 = d \sec \theta_1, \quad (5.36)$$

$$r_2 = d \sec \theta_2, \quad (5.37)$$

$$dS = dy' = d \sec^2 \theta d\theta, \quad (5.38)$$

$$t_1 = \frac{\mathbf{r}_1 \cdot \mathbf{t}}{|\mathbf{t}|} = r_1 \cos(90 - \theta_1) = r_1 \sin \theta_1, \quad (5.39)$$

$$t_2 = \frac{\mathbf{r}_2 \cdot \mathbf{t}}{|\mathbf{t}|} = r_2 \cos(90 - \theta_2) = r_2 \sin \theta_2, \quad (5.40)$$

$$l = \sqrt{(x'_2 - x'_1)^2 + (y'_2 - y'_1)^2} = y'_2 - y'_1, \quad (5.41)$$

$$t_2 - t_1 = l, \quad (5.42)$$



$$\Delta\theta = \theta_2 - \theta_1 = \arctan\left(\frac{dl}{d^2 + t_1 t_2}\right), \quad (5.43)$$

onde  $l$  é o comprimento do elemento  $j$  e  $\Delta\theta$  é o ângulo entre  $\mathbf{r}_1$  e  $\mathbf{r}_2$ .

As funções de interpolação linear são definidas no sistema de coordenadas locais 2D

$$N_1 = \frac{y'_2 - y'}{y'_2 - y'_1} = \frac{r_2 \sin \theta_2 - r \sin \theta}{l}, \quad (5.44)$$

$$N_2 = \frac{y' - y'_1}{y'_2 - y'_1} = \frac{r \sin \theta - r_1 \sin \theta_1}{l}. \quad (5.45)$$

Com as relações matemáticas e as funções de interpolação definidas acima, as integrais de contorno para calcular os termos  $h_1$ ,  $h_2$ ,  $g_1$  e  $g_2$  podem ser determinadas analiticamente.

O termo  $g_1$  dado na Eq. (5.23) é computado como:

$$g_1 = \int_{\Gamma_j} N_1 u_{ij}^* d\Gamma = \int_{\Delta S} \left(\frac{r_2 \sin \theta_2 - r \sin \theta}{l}\right) \left(-\frac{1}{2\pi K} \ln r\right) dS. \quad (5.46)$$

Inserindo as Eqs. (5.34), (5.35) e (5.38) na Eq. (5.46) produz-se:

$$g_1 = \left(-\frac{1}{2\pi Kl}\right) \left[ \int_{\theta_1}^{\theta_2} r_2 \sin \theta_2 \ln(d \sec \theta) d \sec^2 \theta d\theta - \int_{\theta_1}^{\theta_2} d \sec \theta \sin \theta \ln(d \sec \theta) d \sec^2 \theta d\theta \right] \quad (5.47)$$

$$g_1 = \left(-\frac{1}{2\pi Kl}\right) \left[ dr_2 \sin \theta_2 \int_{\theta_1}^{\theta_2} \ln(d \sec \theta) \sec^2 \theta d\theta - d^2 \int_{\theta_1}^{\theta_2} \tan \theta \ln(d \sec \theta) \sec^2 \theta d\theta \right]. \quad (5.48)$$

Resolvendo separadamente as integrais que aparecem em  $g_1$ , tem-se:

$$\begin{aligned} I_1 &= dr_2 \sin \theta_2 \int_{\theta_1}^{\theta_2} \ln(d \sec \theta) \sec^2 \theta d\theta \\ &= dr_2 \sin \theta_2 \left[ \tan \theta (\ln |d \sec \theta| - 1) + \theta \right] \Big|_{\theta_1}^{\theta_2} \\ &= r_2 \sin \theta_2 \left[ d \tan \theta (\ln |d \sec \theta| - 1) + \theta d \right] \Big|_{\theta_1}^{\theta_2} \\ &= r_2 \sin \theta_2 \ln |d \sec \theta_2| - r_2 \sin \theta_2 + \theta_2 d - r_1 \sin \theta_1 \ln |d \sec \theta_1| + r_1 \sin \theta_1 - \theta_1 d \\ &= t_2 [t_2 \ln |r_2| - t_2 + \theta_2 d - t_1 \ln |r_1| + t_1 - \theta_1 d] \\ &= t_2 [ds \Delta\theta - l - t_1 \ln |r_1| + t_2 \ln |r_2|]. \end{aligned} \quad (5.49)$$

A segunda integral  $I_2$  é:

$$\begin{aligned}
I_2 &= d^2 \int_{\theta_1}^{\theta_2} \tan \theta \ln (d \sec \theta) \sec^2 \theta d\theta \\
&= d^2 \left[ \frac{(-1 + 2 \ln |d \sec \theta|) \sec^2 \theta}{4} \right] \Big|_{\theta_1}^{\theta_2} \\
&= \frac{1}{4} [-d^2 \sec^2 \theta + 2d^2 \sec^2 \theta \ln |d \sec \theta|] \Big|_{\theta_1}^{\theta_2} \\
&= \frac{1}{4} [-r^2 + 2r^2 \ln |r|] \Big|_{r_1}^{r_2} \\
&= \frac{1}{4} [-r_2^2 + 2r_2^2 \ln |r_2| + r_1^2 - 2r_1^2 \ln |r_1|] .
\end{aligned} \tag{5.50}$$

Incorporando as Eqs.(5.49) e (5.50) na Eq.(5.48) , obtém-se o seguinte resultado:

$$g_1 = \left( \frac{1}{8\pi Kl} \right) [4t_2 (-ds\Delta\theta + l + t_1 \ln |r_1| - t_2 \ln |r_2|) + 2r_2^2 \ln |r_2| - 2r_1^2 \ln |r_1| + r_1^2 - r_2^2] . \tag{5.51}$$

Da mesma forma, o termo  $g_2$  dado na Eq.(5.24) é calculado por:

$$g_2 = \int_{\Gamma_j} N_2 u_{ij}^* d\Gamma = \int_{\Delta S} \left( \frac{r \sin \theta - r_1 \sin \theta_1}{l} \right) \left( -\frac{1}{2\pi K} \ln r \right) dS. \tag{5.52}$$

Inserindo as Eqs.(5.36) e (5.39) na Eq.(5.52) produz-se:

$$\begin{aligned}
g_2 &= \left( -\frac{1}{2\pi Kl} \right) \left[ \int_{\theta_1}^{\theta_2} r \sin \theta \ln (d \sec \theta) d \sec^2 \theta d\theta - \int_{\theta_1}^{\theta_2} r_1 \sin \theta_1 \ln (d \sec \theta) d \sec^2 \theta d\theta \right] \\
&= \left( -\frac{1}{2\pi Kl} \right) \left[ \int_{\theta_1}^{\theta_2} d \tan \theta \ln (d \sec \theta) d \sec^2 \theta d\theta - \int_{\theta_1}^{\theta_2} r_1 \sin \theta_1 \ln (d \sec \theta) d \sec^2 \theta d\theta \right] \\
&= \left( -\frac{1}{2\pi Kl} \right) \left[ \frac{1}{4} (2r_2^2 \ln |r_2| - 2r_1^2 \ln |r_1| - r_2^2 + r_1^2) - t_1 (ds\Delta\theta - l - t_1 \ln |r_1| + t_2 \ln |r_2|) \right] \\
&= \left( \frac{1}{8\pi Kl} \right) [4t_1 (ds\Delta\theta - l - t_1 \ln |r_1| + t_2 \ln |r_2|) + 2r_1^2 \ln |r_1| - 2r_2^2 \ln |r_2| + r_2^2 - r_1^2] .
\end{aligned} \tag{5.53}$$

O termo  $h_1$  dado pela Eq. (5.21) pode ser facilmente computado como:

$$h_1 = \int_{\Gamma_j} N_1 q_{ij}^* d\Gamma = \int_{\Delta S} \left[ \frac{r_2 \sin \theta_2 - r \sin \theta}{l} \right] \left[ \frac{r_x n_x + r_y n_y}{2\pi r^2} \right] dS. \tag{5.54}$$

Inserindo a Eq.(5.38) na Eq.(5.54), tem-se:

$$\begin{aligned}
h_1 &= \int_{\theta_1}^{\theta_2} \left[ \frac{r_2 \sin \theta_2 - r \sin \theta}{l} \right] \left[ \frac{d^2 \sec^2 \theta}{2\pi d^2 \sec^2 \theta} \right] d\theta \\
&= \left( \frac{1}{2\pi l} \right) \left[ \int_{\theta_1}^{\theta_2} r_2 \sin \theta_2 d\theta - \int_{\theta_1}^{\theta_2} r \sin \theta d\theta \right] \\
&= \left( \frac{1}{2\pi l} \right) \left[ r_2 \sin \theta_2 (\theta_2 - \theta_1) + d \ln |\cos \theta| \Big|_{\theta_1}^{\theta_2} \right] \\
&= \left( \frac{1}{2\pi l} \right) \left[ r_2 \sin \theta_2 (\theta_2 - \theta_1) + d (\ln |\cos \theta_2| - \ln |\cos \theta_1|) \right] \\
&= \left( \frac{1}{2\pi l} \right) \left[ t_2 \Delta\theta + d \left( \ln \left| \frac{d}{r_2} \right| - \ln \left| \frac{d}{r_1} \right| \right) \right] \\
&= \left( \frac{1}{2\pi l} \right) \left[ t_2 \Delta\theta + d \ln \left| \frac{r_1}{r_2} \right| \right].
\end{aligned} \tag{5.55}$$

Repetindo o processo para calcular o termo  $h_2$  obtido pela Eq.(5.22), tem-se:

$$\begin{aligned}
h_2 &= \int_{\Gamma_j} N_2 q_{ij}^* d\Gamma = \int_{\Delta S} \left[ \frac{r \sin \theta - r_1 \sin \theta_1}{l} \right] \left[ \frac{r_x n_x + r_y n_y}{2\pi r^2} \right] dS \\
&= \int_{\theta_1}^{\theta_2} \left[ \frac{r \sin \theta - r_1 \sin \theta_1}{l} \right] \left[ \frac{d^2 \sec^2 \theta}{2\pi d^2 \sec^2 \theta} \right] d\theta \\
&= \left( \frac{1}{2\pi l} \right) \left[ \int_{\theta_1}^{\theta_2} r \sin \theta d\theta - \int_{\theta_1}^{\theta_2} r_1 \sin \theta_1 d\theta \right] \\
&= \left( \frac{1}{2\pi l} \right) \left[ \int_{\theta_1}^{\theta_2} d \tan \theta d\theta - r_1 \sin \theta_1 (\theta_2 - \theta_1) \right] \\
&= \left( \frac{1}{2\pi l} \right) \left[ -d \ln |\cos \theta| \Big|_{\theta_1}^{\theta_2} - r_1 \sin \theta_1 (\theta_2 - \theta_1) \right] \\
&= \left( \frac{1}{2\pi l} \right) \left[ -d (\ln |\cos \theta_2| - \ln |\cos \theta_1|) - r_1 \sin \theta_1 (\theta_2 - \theta_1) \right] \\
&= \left( \frac{1}{2\pi l} \right) \left[ d \left( \ln \left| \frac{d}{r_1} \right| - \ln \left| \frac{d}{r_2} \right| \right) - r_1 \sin \theta_1 (\theta_2 - \theta_1) \right] \\
&= \left( \frac{1}{2\pi l} \right) \left[ d \ln \left| \frac{r_2}{r_1} \right| - t_1 \Delta\theta \right].
\end{aligned} \tag{5.56}$$

## Capítulo 6

# Método dos Elementos de Contorno de Galerkin Simétrico

O método dos elementos de contorno pode ser derivado por diferentes metodologias, resultando em implementações computacionais distintas, que se baseiam em reduzir as equações integrais de contorno contínuas em sistemas de equações lineares. A formulação clássica do MEC é conhecida como método da colocação, onde se procura satisfazer as equações integrais de contorno de forma forte, diretamente em nós específicos do contorno do modelo, usualmente, os próprios nós de discretização do problema. Em contraste, no método de galerkin procura-se satisfazer as equações integrais de contorno de forma fraca. A estratégia utilizada pelo método consiste em aplicar a técnica de resíduos ponderados de Galerkin às equações integrais de contorno, distribuindo-se o erro cometido pela aproximação da melhor forma possível. Pode-se ainda, fazendo uso das equações hipersingulares de contorno, reduzir as equações integrais de contorno a um sistema simétrico de equações lineares. Denomina-se essa estratégia de método simétrico de Galerkin.

Apresenta-se neste capítulo os principais passos para a implementação numérica do MEC simétrico de Galerkin. São retratadas as estratégias para a obtenção do sistema de equações simétrico.

## 6.1 Formulação de Galerkin para o MEC

Como já mostrado nos capítulos anteriores, no MEC, tem-se a equação integral regular:

$$u(x_d) = \int_s \frac{\partial u^*}{\partial n} u ds - \int_s u^* \frac{\partial u}{\partial n} ds, \quad (6.1)$$

e a equação de fluxo (hipersingular):

$$\frac{\partial u}{\partial n}(x_d) = \int_s \frac{\partial^2 u^*}{\partial n^2} u ds - \int_s \frac{\partial u^*}{\partial n} \frac{\partial u}{\partial n} ds. \quad (6.2)$$

Aproximando a solução  $u$  por:

$$\tilde{u} = \sum_{k=1}^n N_k u_k, \quad (6.3)$$

que é válida em pequenos pedaços do contorno  $s$ , surge então o resíduo dado por:

$$R_1 = \sum_{j=1}^{ne} \int_{\Gamma_j} \frac{\partial u^*}{\partial n} \left( \sum_{k=1}^n N_k u_k \right) d\Gamma - \sum_{j=1}^{ne} \int_{\Gamma_j} u^* \left( \sum_{k=1}^n N_k \frac{\partial u_k}{\partial n} \right) d\Gamma - u(x_d), \quad (6.4)$$

$$R_2 = \sum_{j=1}^{ne} \int_{\Gamma_j} \frac{\partial^2 u^*}{\partial n^2} \left( \sum_{k=1}^n N_k u_k \right) d\Gamma - \sum_{j=1}^{ne} \int_{\Gamma_j} \frac{\partial u^*}{\partial n} \left( \sum_{k=1}^n N_k \frac{\partial u_k}{\partial n} \right) d\Gamma - \frac{\partial u}{\partial n}(x_d), \quad (6.5)$$

onde  $\Gamma_j$  é uma aproximação de  $s_j$ , e  $R_1$  e  $R_2$  são os resíduos para a equação potencial e de fluxo, respectivamente.

No MEC de Galerkin simétrico, a primeira equação do sistema linear é obtida fazendo a integral do resíduo  $R$  multiplicada pela primeira função de forma do primeiro elemento integrado sobre o contorno. Entretanto, como essa função de forma só é diferente de zero ao longo do primeiro elemento, esta integral só precisa ser feita no primeiro elemento:

$$\int_{\Gamma_1} N_1 R d\Gamma = 0. \quad (6.6)$$

Para a equação regular, tem-se:

$$\int_{\Gamma_1} N_1 R_1 d\Gamma = \int_{\Gamma_1} N_1 \left[ \sum_{j=1}^{ne} \int_{\Gamma_j} \frac{\partial u^*}{\partial n} \left( \sum_{k=1}^n N_k u_k \right) d\Gamma - \sum_{j=1}^{ne} \int_{\Gamma_j} u^* \left( \sum_{k=1}^n N_k \frac{\partial u_k}{\partial n} \right) d\Gamma - u(x_d) \right] d\Gamma = 0, \quad (6.7)$$

e para a hipersingular:

$$\int_{\Gamma_1} N_1 R_2 d\Gamma = \int_{\Gamma_1} N_1 \left[ \sum_{j=1}^{ne} \int_{\Gamma_j} \frac{\partial^2 u^*}{\partial n^2} \left( \sum_{k=1}^n N_k u_k \right) d\Gamma - \sum_{j=1}^{ne} \int_{\Gamma_j} \frac{\partial u^*}{\partial n} \left( \sum_{k=1}^n N_k \frac{\partial u_k}{\partial n} \right) d\Gamma - \frac{\partial u}{\partial n}(x_d) \right] d\Gamma = 0. \quad (6.8)$$

Considerando o ponto no contorno, para a equação regular, tem-se:

$$\int_{\Gamma_i} N_i \frac{1}{2} u_i d\Gamma - \int_{\Gamma_i} N_i \sum_{j=1}^{ne} \int_{\Gamma_j} \frac{\partial u^*}{\partial n} \left( \sum_{k=1}^n N_k u_k \right) d\Gamma d\Gamma + \int_{\Gamma_i} N_i \sum_{j=1}^{ne} \int_{\Gamma_j} u^* \left( \sum_{k=1}^n N_k \frac{\partial u_k}{\partial n} \right) d\Gamma d\Gamma = 0. \quad (6.9)$$

Em que o termo

$$[M] = \frac{1}{2} \int_{\Gamma_i} N_i [N_1 \dots N_n] d\Gamma \quad (6.10)$$

origina a chamada matriz de massa  $\mathbf{M}$ ,

$$[\bar{H}] = \int_{\Gamma_i} N_i \sum_{j=1}^{ne} \int_{\Gamma_j} \frac{\partial u^*}{\partial n} [N_1 \dots N_n] d\Gamma d\Gamma \quad (6.11)$$

é a matriz  $\bar{\mathbf{H}}$ , e

$$[G] = \int_{\Gamma_i} N_i \sum_{j=1}^{ne} \int_{\Gamma_j} u^* [N_1 \dots N_n] d\Gamma d\Gamma \quad (6.12)$$

é a matriz de influência  $\mathbf{G}$ . A matriz  $\mathbf{G}$  é simétrica na formulação de Galerkin mas não é na colocação, a menos que os elementos tenham o mesmo tamanho.

Usando a equação integral padrão, tem-se:

$$[M]\{u\} = [\bar{H}]\{u\} - [G] \left\{ \frac{\partial u}{\partial n} \right\}, \quad (6.13)$$

em que as matrizes  $\mathbf{M}$  e  $\mathbf{G}$  são simétricas.

Utilizando a equação integral hipersingular, tem-se:

$$[M]_h \left\{ \frac{\partial u}{\partial n} \right\} = [\bar{H}]_h \{u\} - [G]_h \left\{ \frac{\partial u}{\partial n} \right\}, \quad (6.14)$$

em que as matrizes  $[M]_h$  e  $[\bar{H}]_h$  são simétricas, e

$$[G]_h = -[\bar{H}]^T. \quad (6.15)$$

Reescrevendo as Eqs. (6.13) e (6.14) para uma forma mais simples de manejar, tem-se:

$$[M_1]\{u\} + [G_1]\{q\} = [\bar{H}_1]\{u\}, \quad (6.16)$$

$$[M_2]\{q\} + [G_2]\{q\} = [\overline{H}_2]\{u\}. \quad (6.17)$$

Considerando que a temperatura é contínua nos nós e o fluxo é descontínuo, é conveniente usar uma função de forma contínua para a temperatura e descontínua para o fluxo. Por exemplo, funções de forma lineares contínuas para aproximar a temperatura e funções de forma constantes para o fluxo.

## 6.2 Consecução do sistema linear simétrico

Em um problema de condução de calor, o contorno  $\Gamma$  pode ser dividido em  $\Gamma_1$ , onde a temperatura  $u^c$  é conhecida, e  $\Gamma_2$ , onde o fluxo  $q^c$  é conhecido.

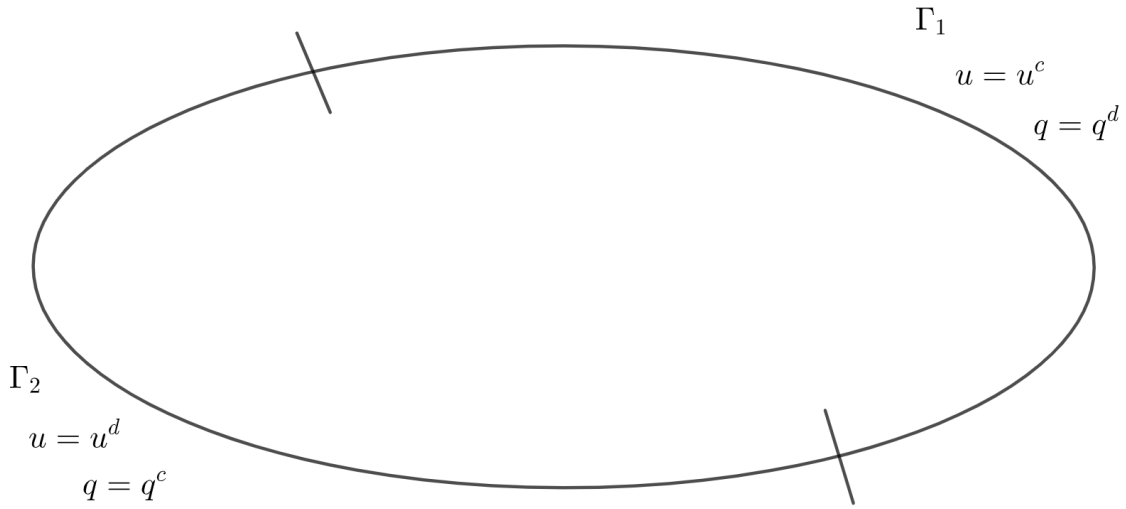


Figura 6.1: Contorno dividido em  $\Gamma_1$  e  $\Gamma_2$ .

onde  $u^d$  e  $q^d$  são valores desconhecidos.

As equações (6.16) e (6.17) podem ser reescritas respectivamente como:

$$\begin{bmatrix} M_1^{11} & M_1^{12} \\ M_1^{21} & M_1^{22} \end{bmatrix} \begin{Bmatrix} u^d \\ u^c \end{Bmatrix} + \begin{bmatrix} G_1^{11} & G_1^{12} \\ G_1^{21} & G_1^{22} \end{bmatrix} \begin{Bmatrix} q^c \\ q^d \end{Bmatrix} = \begin{bmatrix} \overline{H}_1^{11} & \overline{H}_1^{12} \\ \overline{H}_1^{21} & \overline{H}_1^{22} \end{bmatrix} \begin{Bmatrix} u^d \\ u^c \end{Bmatrix}, \quad (6.18)$$

$$\begin{bmatrix} M_2^{11} & M_2^{12} \\ M_2^{21} & M_2^{22} \end{bmatrix} \begin{Bmatrix} q^c \\ q^d \end{Bmatrix} + \begin{bmatrix} G_2^{11} & G_2^{12} \\ G_2^{21} & G_2^{22} \end{bmatrix} \begin{Bmatrix} q^c \\ q^d \end{Bmatrix} = \begin{bmatrix} \overline{H}_2^{11} & \overline{H}_2^{12} \\ \overline{H}_2^{21} & \overline{H}_2^{22} \end{bmatrix} \begin{Bmatrix} u^d \\ u^c \end{Bmatrix}. \quad (6.19)$$

Onde, para uma dada matriz qualquer:

$$A_c^{ab}$$

- O índice  $a$  indica se o elemento fonte encontra-se em  $\Gamma_1$ , onde  $a = 1$ , ou em  $\Gamma_2$ , onde  $a = 2$ ;
- O índice  $b$  indica se o elemento de integração encontra-se em  $\Gamma_1$ , onde  $b = 1$ , ou em  $\Gamma_2$ , onde  $b = 2$ ;
- O índice  $c$  indica se a equação é a equação regular, onde  $c = 1$ , ou hipersingular, onde  $c = 2$ .

Para se obter um sistema linear simétrico, utiliza-se a seguinte estratégia:

- Nos nós onde o fluxo é desconhecido, utiliza-se a equação de temperatura (equação integral de contorno fracamente singular);
- Nos nós onde a temperatura é desconhecida, utiliza-se a equação de fluxo (equação integral de contorno hipersingular).

Assim, o sistema linear simétrico é alcançado fazendo:

$$\begin{bmatrix} M_1^{21} & M_1^{22} \\ M_2^{11} & M_2^{12} \end{bmatrix} \begin{Bmatrix} u^d \\ u^c \end{Bmatrix} + \begin{bmatrix} G_1^{21} & G_1^{22} \\ G_2^{11} & G_2^{12} \end{bmatrix} \begin{Bmatrix} q^c \\ q^d \end{Bmatrix} = \begin{bmatrix} \overline{H}_1^{21} & \overline{H}_1^{22} \\ \overline{H}_2^{11} & \overline{H}_2^{12} \end{bmatrix} \begin{Bmatrix} u^d \\ u^c \end{Bmatrix}. \quad (6.20)$$

Chamando  $[H] = [\overline{H}] + [M]$ , tem-se:

$$\begin{bmatrix} G_1^{21} & G_1^{22} \\ G_2^{11} & G_2^{12} \end{bmatrix} \begin{Bmatrix} q^c \\ q^d \end{Bmatrix} = \begin{bmatrix} H_1^{21} & H_1^{22} \\ H_2^{11} & H_2^{12} \end{bmatrix} \begin{Bmatrix} u^d \\ u^c \end{Bmatrix}, \quad (6.21)$$

$$\begin{bmatrix} G_1^{22} & -H_1^{21} \\ G_2^{12} & -H_2^{11} \end{bmatrix} \begin{Bmatrix} q^d \\ u^d \end{Bmatrix} = \begin{bmatrix} H_1^{22} & -G_1^{21} \\ H_2^{12} & -G_2^{11} \end{bmatrix} \begin{Bmatrix} u^c \\ q^c \end{Bmatrix}. \quad (6.22)$$

Vale ressaltar que  $H_1 = -G_2$ . Desta forma, a equação (6.22) é um sistema linear do tipo

$$[A]\{x\} = \{b\}, \quad (6.23)$$

onde  $\mathbf{A}$  é uma matriz simétrica. Resolvendo esse sistema, as variáveis desconhecidas são então calculadas.



# Capítulo 7

## Implementação Computacional

Com as formulações das equações integrais de contorno usando elementos constantes (com equação tanto regular quanto hipersingular), elementos lineares, e Galerkin simétrico, foi possível implementar programas de elementos de contorno aplicados a problemas de condução de calor sem geração, em linguagem MatLab<sup>®</sup>, capazes de resolver o problema matematicamente e apresentar a solução através de geometrias que contêm pontos para indicar os nós, retas para indicar a direção e sentido do fluxo, e mapas de cores para designar a temperatura.

### 7.1 Descrição dos programas

Os programas de elementos de contorno implementados são constituídos das seguintes etapas:

- Definição de dados do problema a ser analisado, como geometria, condições de contorno e discretização;
- Visualização de características do problema que será analisado, como geometria, malha de elementos de contorno e condições de contorno;
- Criação de pontos internos à geometria;
- Construção das matrizes **H** e **G** e montagem do sistema linear;
- Aplicação das condições de contorno para temperatura e fluxo;
- Resolução do sistema linear, obtendo-se o vetor de soluções;
- Visualização dos resultados obtidos, através da representação da geometria do problema e mapas de cores para indicar grandezas.

Os programas operam da seguinte forma: um arquivo de dados é usado para definir o problema de condução de calor, em que a geometria é determinada por uma matriz que contém as coordenadas de todos os pontos e por uma matriz que define as linhas da geometria a partir dos pontos já definidos, que são os segmentos. A discretização é definida por uma matriz que contém o número de elementos em cada linha. Outra matriz abrange as condições de contorno de temperatura e fluxo, e por fim, a propriedade do material utilizado para análise é estipulada pela variável que indica o valor da condutividade térmica do material.

Em seguida, uma função executa a transcrição dos dados de entrada do problema (nós, elementos, etc.) em um conjunto de dados que é utilizado durante a análise. Assim, o conjunto de dados iniciais, que tem aspecto amigável para o usuário, é transformado em um conjunto de dados úteis ao programa. Dessa maneira, essa função cria as matrizes que contém as coordenadas nodais, as conectividades dos elementos e as condições de contorno por elemento.

Com essas informações, uma nova função exibe a geometria do problema e a malha de elementos de contorno através de representações gráficas do MatLab®. Posteriormente, neste mesmo gráfico, serão expostas as condições de contorno do problema, com o emprego do símbolo de seta para representar o fluxo, o de triângulo para indicar ausência de fluxo, e o de círculo para assinalar os segmentos onde a temperatura é conhecida.

Em seguida, opera-se uma função que gera um conjunto de pontos internos ao domínio do problema em estudo. De início, é concebida uma rede regular de pontos sobre a geometria. Seguidamente, cada ponto criado é testado por um algoritmo que verifica se o ponto é interno ou externo ao domínio e experimenta a proximidade do ponto ao contorno, pois ele não pode estar próximo ao contorno devido à dificuldade de integração. Caso o ponto esteja dentro da geometria e respeite a distância ao contorno, ele será adicionado à matriz que contém as coordenadas dos pontos internos.

Posteriormente, uma função irá construir as matrizes **H** e **G**. A partir delas, o algoritmo separa os termos conhecidos dos desconhecidos e, assim, monta-se o sistema linear.

Por fim, uma função aplica as condições de contorno para temperatura e fluxo, resolve o sistema linear e obtém o vetor de soluções para os nós de contorno e para os pontos internos. Deste modo, uma outra função aciona as reproduções gráficas e expõe os resultados obtidos através da representação da geometria do problema e mapa de cor para indicar o valor de temperatura em cada ponto, além de setas para indicar as direções do fluxo nos pontos internos.

A Figura 7.1 retrata um fluxograma com as principais funções dos programas.

Os códigos implementados estão disponíveis no repositório online <<https://github.com/wernerh7/PG-Galerkin-Simetrico>>.

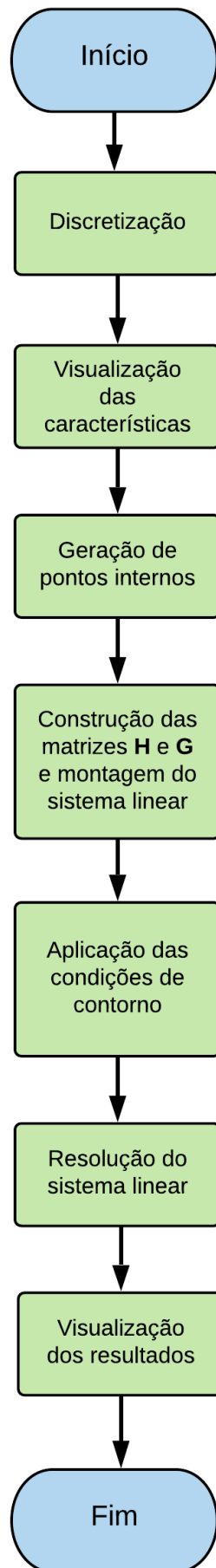


Figura 7.1: Fluxograma dos programas de elementos de contorno.

# Capítulo 8

## Simulação Numérica e Resultados

Para avaliar as formulações de elementos de contorno usando elementos constante (empregando a equação regular e a hipersingular), elementos lineares e Galerkin simétrico, foi analisada a distribuição de temperatura e fluxo de calor ao longo de uma placa plana quadrada e de uma placa plana com formato trapezoidal.

Serão apresentados os resultados obtidos com os programas desenvolvidos em linguagem MatLab<sup>®</sup> relatados no capítulo 7 deste trabalho.

### 8.1 Condução de calor na placa quadrada

A placa plana quadrada a ser analisada tem dimensões mostradas na Fig. 8.1. O problema foi discretizado com uma malha de 44 nós, sendo 11 nós em cada segmento. Foi considerada uma condutividade térmica de  $k = 1$ .

As condições de contorno do problema são:

- No segmento  $\overline{AB}$  a temperatura é igual a 0;
- No segmento  $\overline{BC}$  o fluxo de calor é nulo;
- No segmento  $\overline{CD}$  a temperatura é igual a 1;
- No segmento  $\overline{AD}$  o fluxo de calor é igual a 1.

Os resultados foram avaliados para todos os nós do contorno da placa. Para os segmentos  $\overline{AD}$  e  $\overline{BC}$ , onde os valores de fluxo são conhecidos, avaliou-se a distribuição de temperatura nesses trechos. Já para os segmentos  $\overline{AB}$  e  $\overline{DC}$ , em que os valores de temperatura são conhecidos, examinou-se a distribuição do fluxo de calor. Esses valores foram calculados com diferentes métodos e tipos de elementos e os resultados foram comparados entre si graficamente. As Figuras 8.2 e 8.3 mostram as comparações para os valores de temperatura e as Figuras 8.4 e 8.5 para os valores de fluxo.

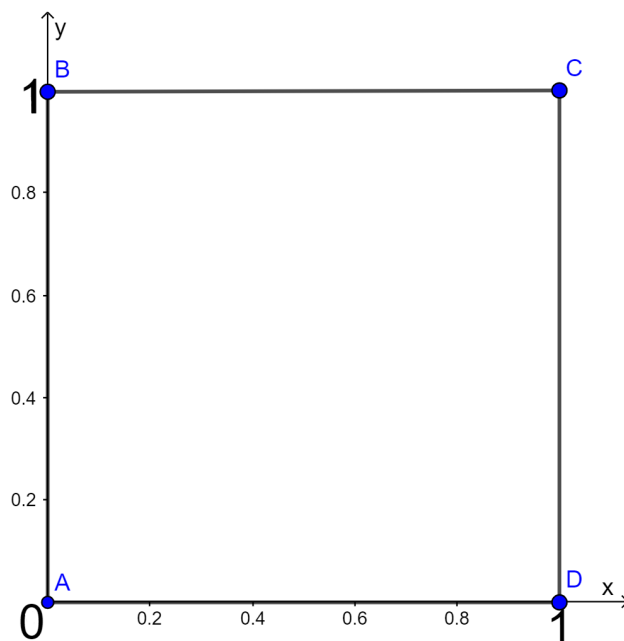


Figura 8.1: Dimensões da placa quadrada.

Observando os gráficos que contêm as comparações, nota-se que a curva representada pelo método de Galerkin simétrico teve uma boa concordância com relação aos outros métodos. Para os segmentos em que o fluxo de calor foi avaliado, contemplou-se valores muito próximos entre as curvas de Galerkin simétrico e do MEC padrão com elementos constantes, mesmo com uma malha grosseira, de apenas 44 nós.

Para os segmentos em que a temperatura é o objeto de estudo, observa-se que a curva de Galerkin simétrico apresentou valores levemente divergentes em relação às curvas dos outros métodos, mas seu formato ainda é qualitativamente semelhante. Impende salientar que a discretização desse problema foi grosseira, e que, para malhas mais refinadas, os métodos apresentam erros relativos quase nulos entre si, como é verificado na Figura 8.6, onde foi realizada a análise dos valores de temperatura dos nós no segmento  $\overline{AD}$ , com uma malha de contorno com 400 nós.

A Figura 8.7 mostra a distribuição de temperatura e o fluxo de calor na placa quadrada, visíveis pelo mapa de cor e setas. Neste problema, o número de pontos internos gerados na geometria foi 441.

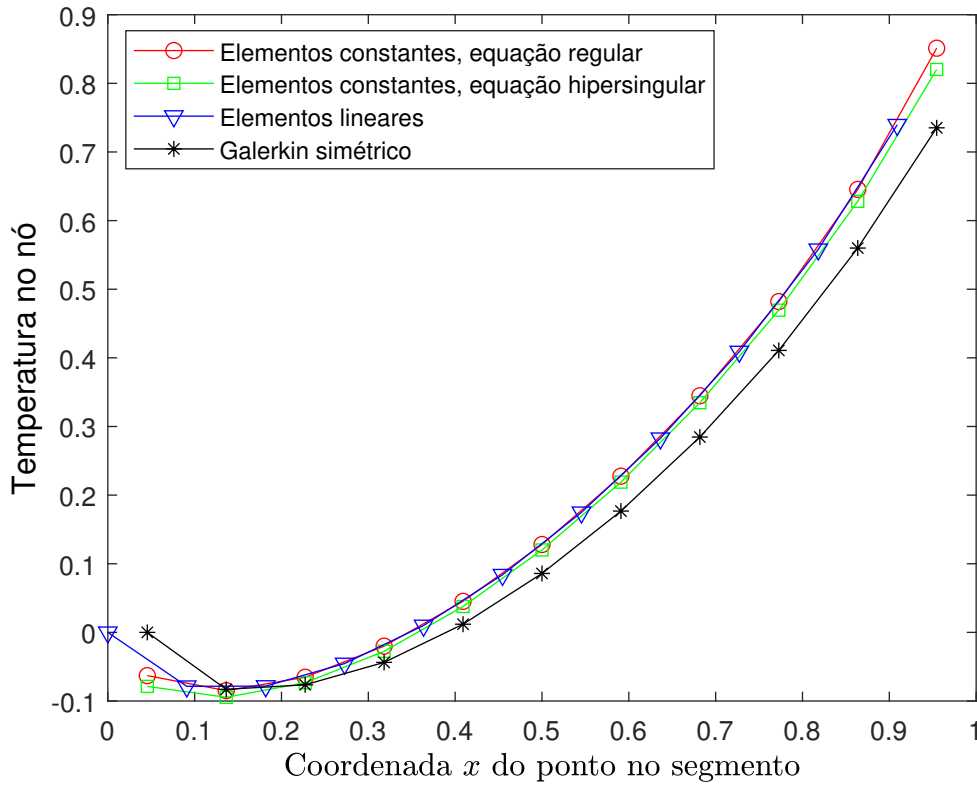


Figura 8.2: Distribuição de temperatura ao longo do segmento  $\overline{AD}$  da placa quadrada.

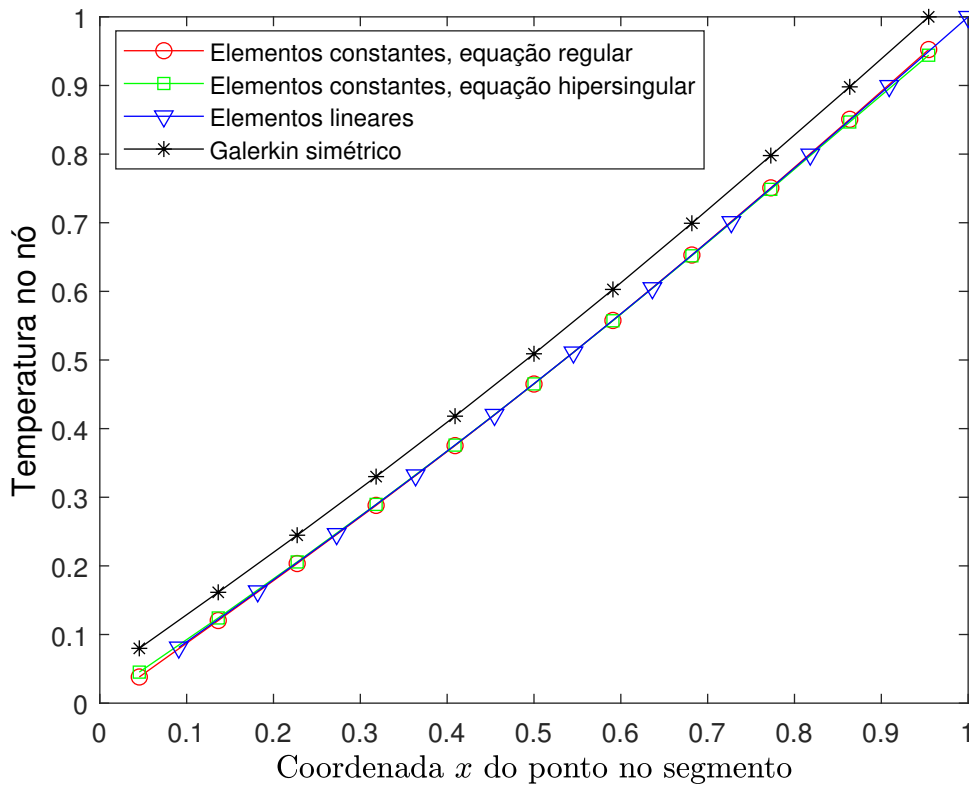


Figura 8.3: Distribuição de temperatura ao longo do segmento  $\overline{BC}$  da placa quadrada.

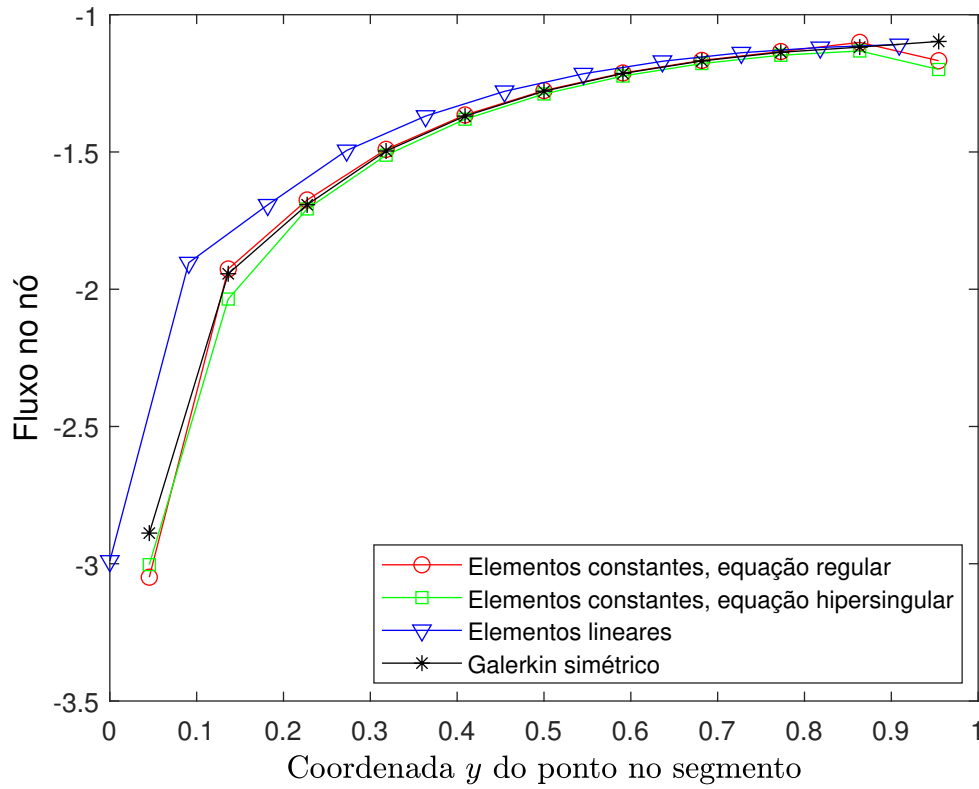


Figura 8.4: Fluxo de calor ao longo do segmento  $\overline{DC}$  da placa quadrada.

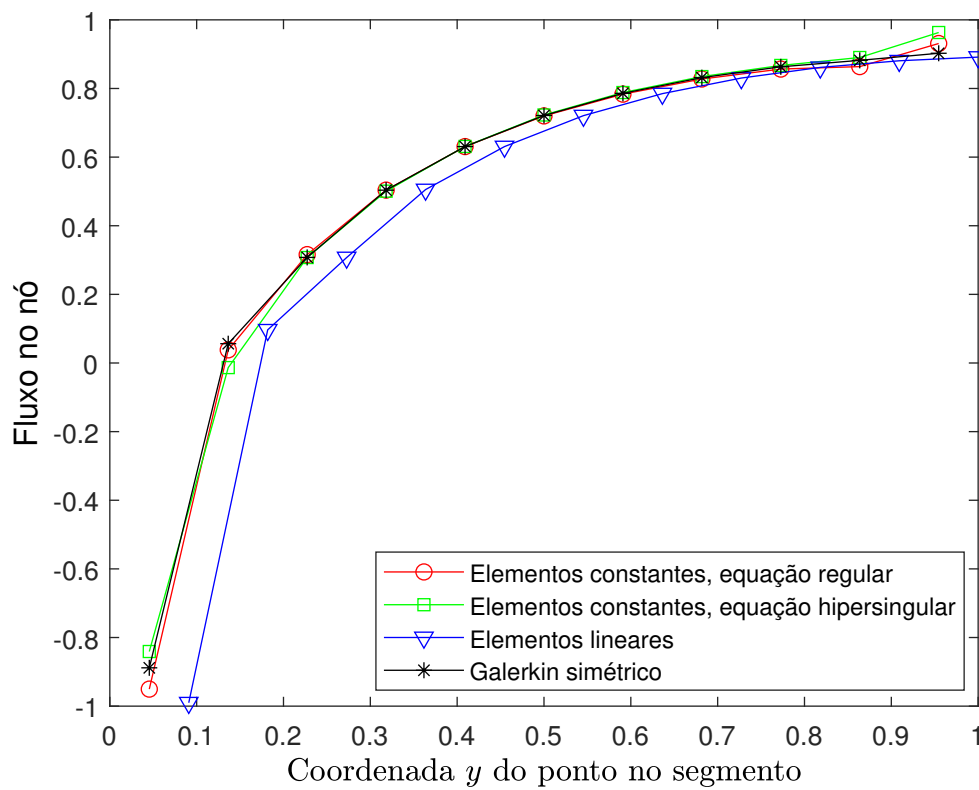


Figura 8.5: Fluxo de calor ao longo do segmento  $\overline{AB}$  da placa quadrada.

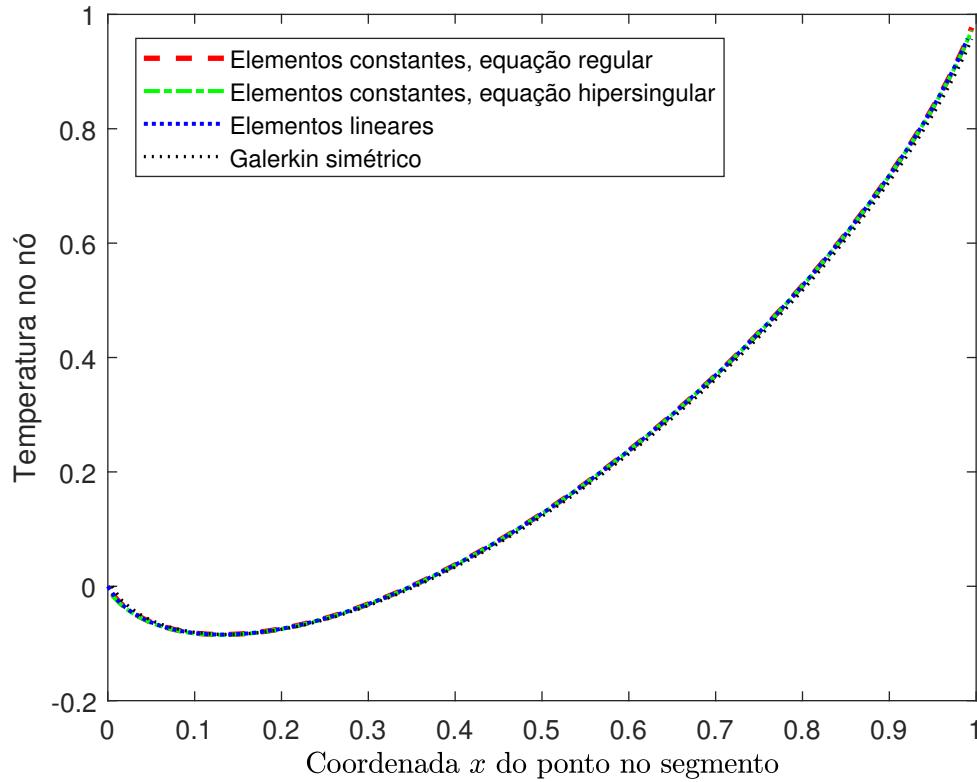


Figura 8.6: Distribuição de temperatura ao longo do segmento  $\overline{AD}$  da placa quadrada (malha refinada).

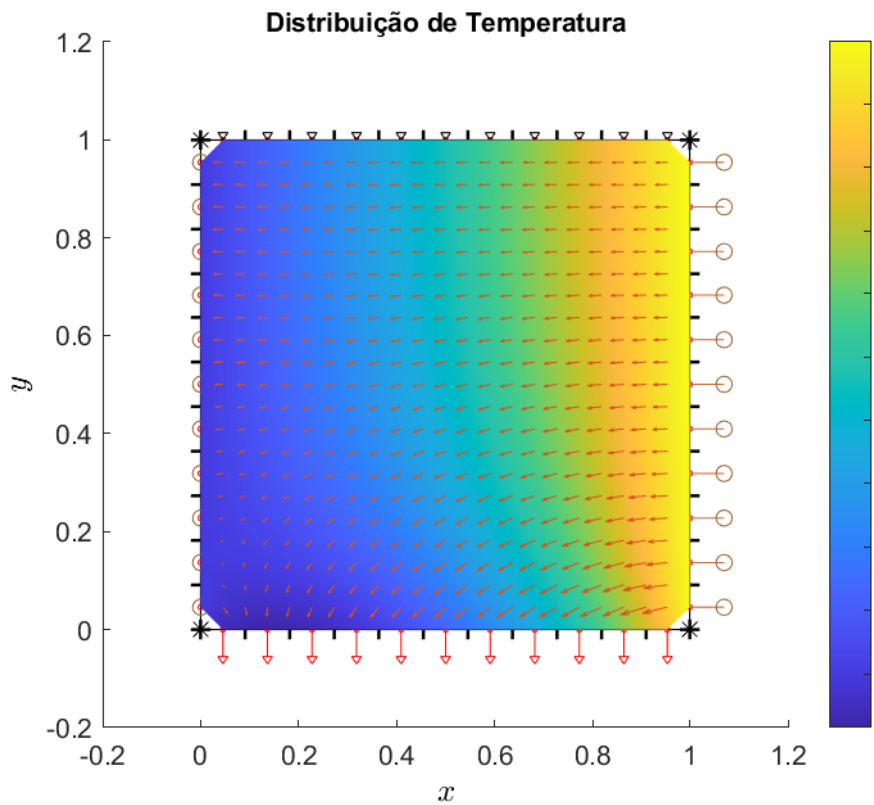


Figura 8.7: Distribuição de temperatura e fluxo de calor ao longo da placa quadrada.



## 8.2 Condução de calor na placa trapezoidal

A placa plana com formato trapezoidal a ser estudada tem dimensões mostradas na Fig. 8.8. O problema foi discretizado com uma malha de 84 nós, com 12 nós em cada segmento, com exceção do segmento  $\overline{AB}$ , que tem 24 nós. Foi considerada uma condutividade térmica de  $k = 1$ .

As condições de contorno do problema são:

- No segmento  $\overline{AB}$  o fluxo de calor é igual a -2;
- No segmento  $\overline{BC}$  a temperatura é igual a 6;
- No segmento  $\overline{CD}$  o fluxo de calor é igual a 3;
- No segmento  $\overline{DE}$  a temperatura é igual a 5;
- No segmento  $\overline{EF}$  o fluxo de calor é nulo;
- No segmento  $\overline{FA}$  a temperatura é igual a 7.

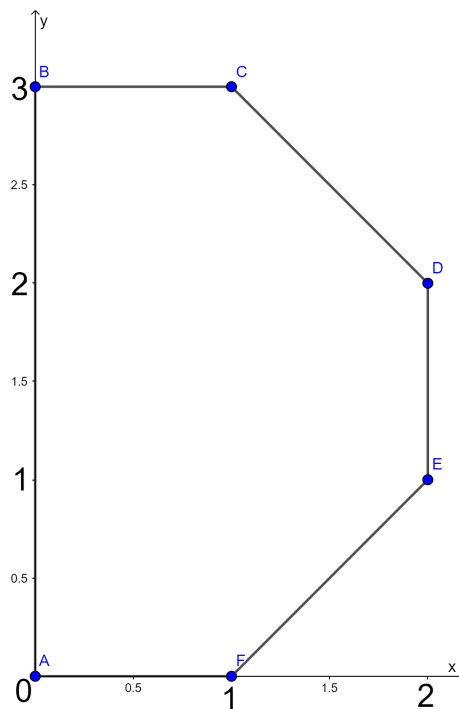


Figura 8.8: Dimensões da placa trapezoidal.

Os resultados foram avaliados para os nós dos segmentos  $\overline{AB}$ , onde avaliou-se a distribuição de temperatura, e  $\overline{BC}$ , onde examinou-se o fluxo de calor. Esses valores foram calculados com diferentes métodos e tipos de elementos e os resultados foram comparados entre si graficamente, assim como foi feito no problema da placa quadrada. As Figuras 8.9 e 8.10 mostram as comparações para os valores de temperatura e fluxo de calor, respectivamente.

Observa-se que, para este problema, a curva representada pelo método de Galerkin simétrico também teve uma boa concordância com relação aos outros métodos. Bem como no problema da placa quadrada, nos segmentos em que o fluxo de calor foi avaliado constatou-se valores muito próximos entre as curvas de Galerkin simétrico e do MEC padrão com elementos constantes. Para os segmentos em que a temperatura foi analisada, notou-se o mesmo comportamento presente no problema anterior: pequena divergência entre os valores da curva de Galerkin em relação à curva apresentada pelos outros métodos, embora formato parecido. Transtorno facilmente solucionável, bastando o uso de uma malha mais refinada.

A Figura 8.11 mostra a distribuição de temperatura e o fluxo de calor na placa trapezoidal, visíveis pelo mapa de cor e setas. Neste problema, o número de pontos internos gerados na geometria foi 359.

Os códigos de MEC padrão com elementos constantes e lineares que foram usados nesse trabalho já haviam sido aplicados por Braga (2012) para resolver outros problemas de condução de calor e, em seus resultados, verificada convergência com as soluções analíticas. Pode-se afirmar, portanto, a validação do código de Galerkin simétrico implementado, visto que suas curvas de resultados tiveram boa concordância em comparação com os outros códigos já validados pela literatura.

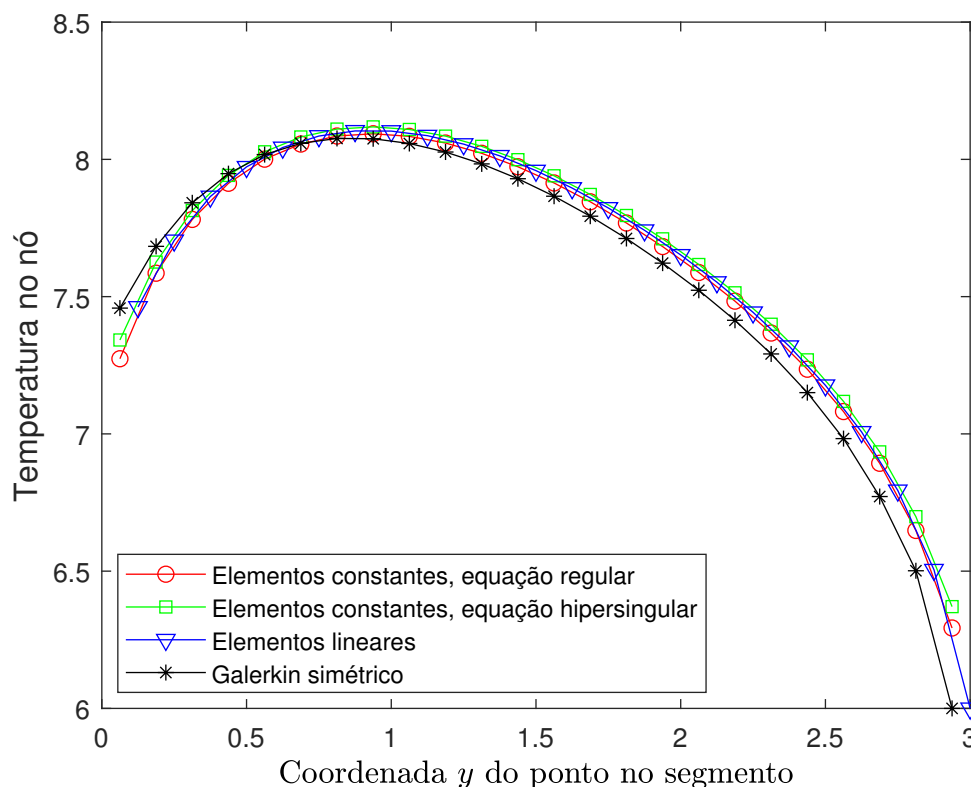


Figura 8.9: Distribuição de temperatura ao longo do segmento  $\overline{AB}$  da placa trapezoidal.

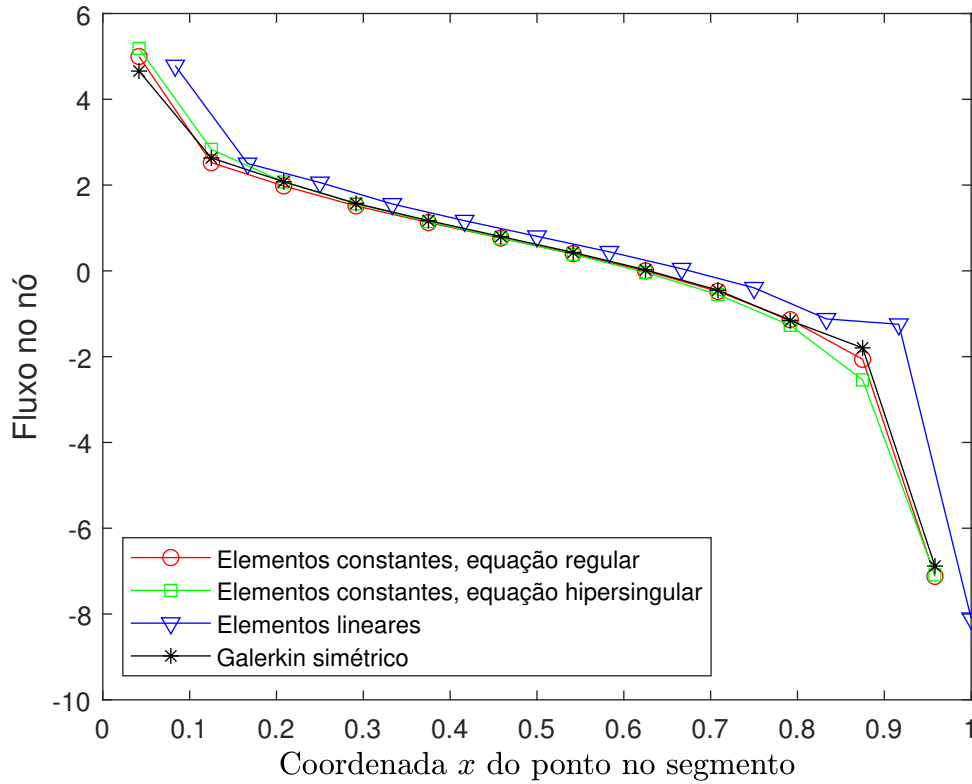


Figura 8.10: Fluxo de calor ao longo do segmento  $\overline{BC}$  da placa trapezoidal.

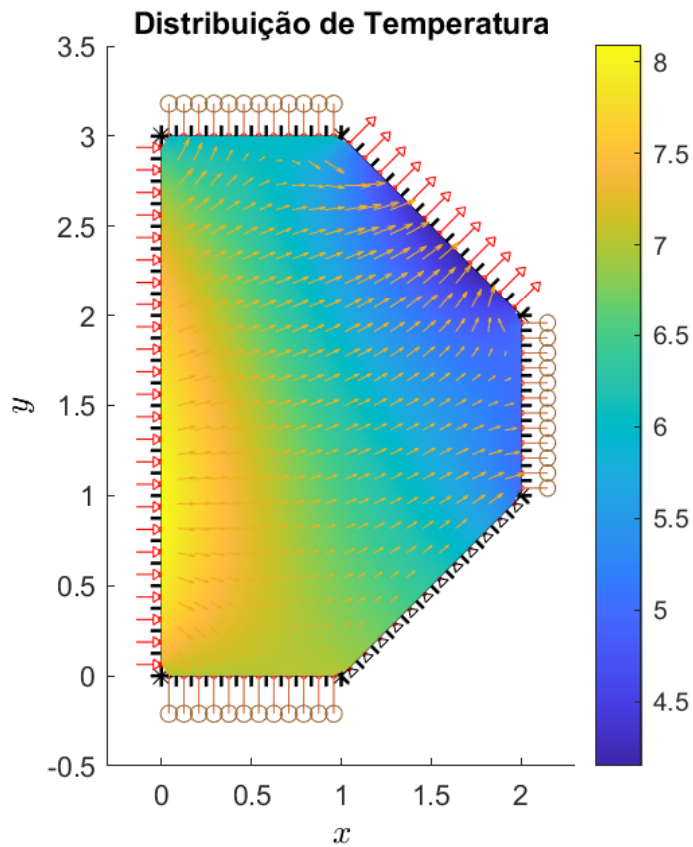


Figura 8.11: Distribuição de temperatura e fluxo de calor ao longo da placa trapezoidal.

# Capítulo 9

## O Programa Bempp

O MEC é um método bem estabelecido e fundamentado para a solução de diversos problemas em engenharia. Os algoritmos necessários para a sua execução (e também para a execução de outros métodos numéricos particularmente conhecidos em engenharia, tais como o MEF) são encontrados comumente em pacotes de software fornecidos comercialmente. Os pacotes de software comerciais possuem implementações testadas e otimizadas dos métodos matemáticos, fornecendo ao usuário uma interface de utilização das capacidades do programa. Entretanto, por serem soluções comerciais, a maior parte desses softwares possuem código fechado e não permitem ao usuário inspecionar o código fonte ou criar extensões de software de acordo com as suas necessidades individuais.

A formulação de elementos de contorno de Galerkin simétrico apresentada nos capítulos anteriores para problemas planos pode ser estendida para problemas tridimensionais. Um exemplo desta extensão é o programa Bempp, que será retratado neste capítulo.

O objetivo deste capítulo é estudar a viabilidade da biblioteca open source Bempp como uma alternativa para a solução de problemas de engenharia utilizando o método dos elementos de contorno, principalmente no domínio de problemas tridimensionais térmicos e elásticos. Esse estudo será realizado a partir da análise de um problema de condução de calor em uma polia.

### 9.1 Bempp e plataformas relacionadas

O Bempp é uma plataforma computacional de elementos de contorno para resolver problemas de engenharia. O programa foi desenvolvido pela University College London em conjunto com a Pontificia Universidad Católica de Chile. A biblioteca é distribuída através da licença MIT, que permite seu uso irrestrito para aplicações comerciais e de código aberto (Bempp Services, 2015).

Algumas das características do Bempp são uma interface Python de simples utilização e o fornecimento de uma imagem Docker que facilita a instalação da biblioteca em plataformas

Windows, Mac e Linux. Ademais, possui suporte para malhas de superfície triangulares e a possibilidade de importar e exportar no formato Gmsh. O pacote de software em si é escrito em C++, garantindo extensibilidade através do uso do paradigma de orientação a objetos fornecido pela linguagem. A versão atual da biblioteca baseia-se nas discretizações de Galerkin das equações integrais de contorno.

A biblioteca faz uso extensivo da plataforma open source FEniCS para a realização de tarefas computacionalmente intensivas, principalmente aplicando computação paralela. De fato, a plataforma FEniCS foi desenvolvida tendo em mente a computação paralela e a prototipagem rápida. Essa ferramenta permite que soluções para problemas complexos sejam esboçadas primeiramente em laptops ou computadores pessoais e que depois o mesmo código seja executado em plataformas de computação em larga escala.

O programa Bempp exerce uso de memória compartilhada e paralelização MPI (Message Passing Interface). A utilização dos recursos de computação paralela torna a execução do código mais eficiente.

Além disso, o software faz uso de matrizes hierárquicas (matrizes H) baseadas no procedimento de aproximação cruzada adaptativa (ACA), o que reduz a complexidade do MEC para uma relação quase linear entre o tamanho do sistema e os custos computacionais.

### **9.1.1 O projeto FEniCS**

O pacote Bempp utiliza as ferramentas de software disponibilizadas pelo projeto FEniCS para a computação de soluções para problemas acoplados MEC/MEF. O FEniCS é um projeto de pesquisa que permite a solução de equações diferenciais parciais (EDPs), tornando possível a solução de modelos científicos e de engenharia através de algoritmos de elementos finitos.

Uma das facilidades que o FEniCS apresenta é sua interface em linguagens de programação de alto nível, tais como Python e C++, permitindo que modelos sejam criados e resolvidos com simplicidade. Essas interfaces permitem ao programador abstrair dos blocos fundamentais de construção do software e focar apenas na utilização dos recursos que esse oferece.

O FEniCS é constituído dos módulos principais DOLFIN, FFC e FIAT. O DOLFIN é o backend em C++ responsável pelas computações em alta performance, implementando as estruturas de dados mais comumente utilizadas, como malhas, funções e algoritmos computacionalmente intensivos. O FFC é a engine de geração de código, uma espécie de compilador responsável por produzir eficientemente o código C++ que será executado pela máquina. Enquanto isso, FIAT é o backend responsável pela implementação do método dos elementos finitos.

### 9.1.2 Paralelização MPI

A tarefa de tornar algoritmos e programas mais rápidos e eficientes, viabilizando novas soluções para problemas complicados em diversas áreas do conhecimento, é um dos grandes desafios em computação. De fato, existe uma vasta gama de problemas que se beneficia de soluções computacionais, principalmente em engenharia. Os algoritmos utilizados tanto no MEF quanto no MEC são computacionalmente intensivos, exigindo uma grande quantidade de poder de processamento e podendo levar muito tempo para oferecerem resultados.

A computação em paralelo é uma das respostas para esse tipo de problema. Soluções utilizando o MEC se beneficiam consideravelmente da utilização de práticas de processamento em paralelo. O Bempp, como mencionado, utiliza a biblioteca MPI e memória compartilhada na otimização dos cálculos.

MPI (Message Passing Interface) é uma biblioteca definida para comunicação entre processos paralelos. O MPI é praticamente a norma para a programação de sistemas em paralelo. Vale ressaltar que o MPI é uma especificação, havendo no mercado uma série de implementações, desde as que exigem pagamento de licenças até alternativas open source, como o Open MPI.

De maneira simplificada, pode-se explicar o funcionamento da biblioteca MPI da seguinte forma:

1. O problema a ser resolvido é dividido em diversas partes;
2. As partes individuais são fracionadas entre os processadores;
3. Os resultados calculados são então coletados, agrupados e fornecidos ao usuário.

A grande vantagem de se usar paralelismo é que um problema complexo e grande pode ser fragmentado em múltiplos pequenos e simples, que podem ser resolvidos simultaneamente. O uso das funcionalidades do MPI torna-se ainda mais importante ao resolver problemas utilizando High-Performance Computers (HPC's).

### 9.1.3 Matrizes hierárquicas

Uma parte importante do método dos elementos de contorno é montar as matrizes de influência  $\mathbf{H}$  e  $\mathbf{G}$ . Esse processo em geral é computacionalmente intensivo, visto que as matrizes envolvidas geralmente são densas (poucos elementos nulos). Dessa maneira, se  $N$  é o número de elementos na malha, a complexidade dessa montagem cresce como  $O(N^3)$ . Isso torna a resolução de problemas grandes intratável em hardware convencional.

No entanto, em anos recentes, duas técnicas foram desenvolvidas para resolver esse problema. A primeira delas foi o método dos elementos de contorno com expansão em multipolos e a segunda foi o método das matrizes hierárquicas. Esses dois métodos diminuem a

complexidade computacional da montagem das matrizes para  $O(N \log N)$ , tornando possível a solução de problemas muito maiores (Bempp Services, 2017).

## 9.2 Metodologia

Para estudar a performance do Bempp na solução de um problema de engenharia, foi realizada a análise de condução de calor num modelo sólido de uma peça de alta complexidade geométrica. A peça escolhida foi uma polia na qual a temperatura foi especificada em duas superfícies: na região cilíndrica que entra em contato com a correia ( $T = 220 \text{ }^\circ\text{C}$ ) e na região cilíndrica que entra em contato com o eixo ( $T = 0 \text{ }^\circ\text{C}$ ). Nas demais superfícies foi assumido fluxo nulo. Esse modelo foi desenhado utilizando o FreeCAD, software de CAD open source, e sua malha gerada pelo Gmsh, software open source para geração de malhas. O intuito é submeter esse modelo a diversos testes dentro do pacote de software Bempp, variando parâmetros diversos para investigar sua performance ao resolver problemas complexos.

Foram gerados dois arquivos de malha diferentes, um com 4686 triângulos e outro mais refinado, com 18744 triângulos. As soluções computacionais foram implementadas no Bempp e os arquivos de malha carregados. A partir dos resultados fornecidos pelo programa, foram obtidos dados a respeito de memória utilizada e graus de liberdade para ambas as malhas.

O Bempp utiliza por padrão a construção por matrizes hierárquicas (hierarchical matrix assembly) para resolução de problemas. Uma série de parâmetros de código controla a matriz hierárquica usada para resolução do problema. No caso do Bempp, o algoritmo ACA (Adaptive Cross Approximation, ou método dos elementos de contorno com aproximação cruzada adaptativa) é empregado para reduzir o uso de memória e CPU.

Os parâmetros variados para o estudo foram:

- (a) o tamanho do bloco, que restringe o tamanho maior de matriz admissível aproximado por matrizes de posto (rank) mais baixo;
- (b) o número de threads, que designa a quantidade de processos paralelos durante a execução do programa;
- (c) o posto (rank) máximo dos blocos de matrizes aproximadas.

## 9.3 Resultados

A Figura 9.1 ilustra a análise em elementos de contorno que foi aplicada à polia para os estudos de performance do Bempp depois do pós-processamento.

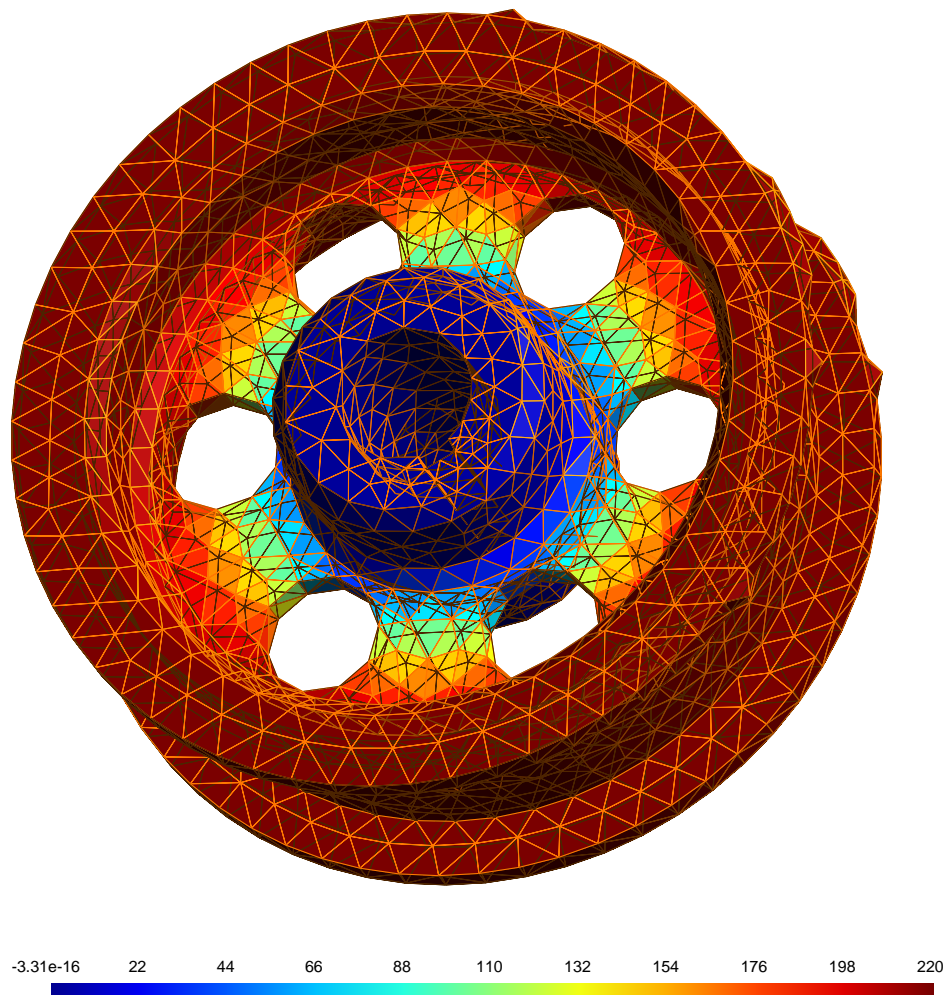


Figura 9.1: Distribuição de temperatura na polia.

A primeira análise foi feita para o problema com a malha menos refinada (malha 1), que possui 4686 triângulos.

Analisando a Tabela 9.1, é possível identificar uma interferência considerável tanto no tempo para montagem das matrizes quanto para a resolução do sistema quando se varia o tamanho do bloco das matrizes (através do algoritmo ACA). A documentação do Bempp recomenda um valor padrão de 2048 para o tamanho do bloco, pois a escolha de um valor muito grande pode ter um efeito negativo no balanceamento de carga entre os diferentes núcleos durante a montagem, e a escolha de um valor muito pequeno pode levar a um aumento significativo na complexidade computacional geral.



Tabela 9.1: Tempo consumido para diferentes tamanhos de bloco (malha 1).

Tamanho de bloco	Montagem das matrizes (s)	Resolução do sistema linear (s)	Tempo total (s)
2048	29,564	8,905	38,470
1024	29,672	8,960	38,633
512	30,582	9,287	39,870
256	32,951	10,322	43,274
128	38,925	11,671	50,596
64	54,503	17,418	71,921

A mesma variação não é encontrada, no entanto, quando se varia o número de threads e o rank máximo das matrizes, como ilustram as Tabelas 9.2 e 9.3. Em ambos os casos, o tempo para montagem e resolução dos sistemas apresenta uma variação quase insignificante em termos práticos.

Tabela 9.2: Tempo consumido para diferentes números de threads (malha 1).

Número de threads	Montagem das matrizes (s)	Resolução do sistema linear (s)	Tempo total (s)
2	29,562	8,875	38,430
3	30,428	8,939	39,367
4	29,509	8,867	38,376
5	29,505	8,860	38,365
6	29,537	8,867	38,404
7	29,573	8,862	38,435

Tabela 9.3: Tempo consumido para diferentes ranks máximos (malha 1).

Rank	Montagem das matrizes (s)	Resolução do sistema linear (s)	Tempo total (s)
10	26,984	8,492	35,477
30	30,012	9,149	39,161
40	29,924	9,404	39,329
80	30,159	9,397	39,557
160	30,206	9,315	39,522
320	30,270	9,205	39,475

Analisando o caso da malha mais refinada, com 18744 triângulos (malha 2), pode-se perceber que o Bempp se torna muito mais lento quando o número de elementos na malha do problema é aumentado, a troco de uma maior precisão nos resultados. O parâmetro que realmente afeta a performance do software é o tamanho do bloco, como se pode averiguar na Tabela 9.4. No entanto, aumentar o tamanho do bloco indefinitivamente não aumenta de maneira considerável a performance. Dobrando o tamanho do bloco de 4096 para 8192, o tempo para a montagem das matrizes foi reduzido em apenas 1 segundo. Assim como na malha mais grosseira, a variação do número de threads de execução do programa não afeta de maneira considerável o tempo de execução, como é possível apurar na Tabela 9.5, embora tenha sido observado um aumento considerável no tempo quando se eleva o rank máximo das matrizes, como visto na Tabela 9.6.

Tabela 9.4: Tempo consumido para diferentes tamanhos de bloco (malha 2).

Tamanho do bloco	Montagem das matrizes (s)	Resolução do sistema linear (s)	Tempo total (s)
256	206,563	90,828	297,391
4096	128,941	62,235	191,176
8192	127,002	61,113	188,116

Tabela 9.5: Tempo consumido para diferentes números de threads (malha 2).

Número de threads	Montagem das matrizes (s)	Resolução do sistema linear (s)	Tempo total (s)
1	131,646	61,654	193,300
2	129,739	62,419	192,158
4	129,484	62,295	191,780
8	129,551	62,533	192,084

Tabela 9.6: Tempo consumido para diferentes ranks máximos (malha 2).

Rank	Montagem das matrizes (s)	Resolução do sistema linear (s)	Tempo total (s)
5	85,744	35,257	121,032
10	112,978	53,896	166,874
30	131,575	64,368	195,943

A análise dos resultados e a experiência no uso do Bempp para a solução de problemas de engenharia mostrou que, apesar do software ser uma excelente ferramenta para o aprendizado e aplicação do método dos elementos de contorno em problemas relativamente pequenos, a performance do programa deixa a desejar em problemas maiores e mais complexos, que exigem malhas mais refinadas para que seja possível obter uma solução precisa e confiável.

No entanto, o software conta com uma extensa documentação e material de apoio para que seja possível aprender a usar as ferramentas que ele oferece. A instalação é simples, e o programa em si se baseia em diversos projetos de sucesso que já utilizam métodos numéricos para solução de problemas de engenharia. Tendo em mente que o Bempp é um projeto em andamento e que novas versões do programa estão sendo trabalhadas, é de se esperar que, em breve, o programa venha a se tornar mais competitivo dentro o cenário atual de softwares de engenharia. Além disso, a proposta open source do programa também é muito atrativa para o meio universitário, visto que é possível explorar o Bempp em conjunto com outras ferramentas também gratuitas, tais como o FreeCAD e o Gmsh, para aplicações simples do MEC.

# Capítulo 10

## Conclusões e Trabalhos Futuros

Este trabalho apresentou uma análise de problemas de condução de calor bidimensional através do método de elementos de contorno de Galerkin simétrico e também através do método dos elementos de contorno por colocação. Foi produzida uma comparação dos resultados para os métodos através da investigação da distribuição de temperatura e fluxo de calor ao longo do contorno da geometria dos problemas em estudo. A partir da observação das soluções obtidas, concluiu-se que a implementação numérica do método de Galerkin apresentou boa concordância com os resultados adquiridos pelo método da colocação, o que ratifica a eficácia do código.

A formulação do MEC simétrico de Galerkin validada pôde ser estendida para problemas tridimensionais, e a análise para esse tipo de problema foi feita com a utilização do programa Bempp. O estudo da performance do software foi realizado a partir da análise da condução de calor de uma peça de alta complexidade geométrica. Demonstrou-se que o programa é uma boa ferramenta para aprendizado e aplicação do método dos elementos de contorno em problemas relativamente pequenos, mas não apresentou boa performance em problemas grandes e complexos.

Como proposta para trabalhos futuros, é sugerida a implementação de código para resolver problemas de elasticidade no software Bempp.

# Referências Bibliográficas

ALBUQUERQUE, E. L. *Introdução ao método dos elementos de contorno*. [S.l.], 2012. 47–53 p.

ALBUQUERQUE, E. L.; IZQUIERDO, L. M. C. Analytical integration to compute the influence matrices in the bem simulation of oil extraction. p. 7–12, 2014.

BALAKRISHNA, C.; GRAY, L.; KANE, J. Efficient analytical integration of symmetric galerkin boundary integrals over curved elements: thermal conduction formulation. *Computer methods in applied mechanics and engineering*, Elsevier, v. 111, n. 3-4, p. 335–355, 1994.

BANERJEE, P. K. *The Boundary Element Methods in Engineering*. Second. London: MacGraw-Hill, 1994.

Bempp Services. *Bempp*. 2015. [Online; Acessado em 01/Novembro/2019]. Disponível em: <<https://bempp.com/>>.

Bempp Services. *Bempp*. 2017. [Online; Acessado em 02/Novembro/2019]. Disponível em: <<https://bempp.com/2017/07/12/fast-assembly-via-hierarchical-matrices/>>.

BRAGA, L. M. O método dos elementos de contorno rápido com expansão em multipólos aplicado a problemas de condução de calor. 2012.

BREBBIA, C. A.; DOMINGUEZ, J. *Boundary elements: an introductory course*. [S.l.]: WIT press, 1994.

BUI, H. An integral equations method for solving the problem of a plane crack of arbitrary shape. *Journal of the Mechanics and Physics of Solids*, Elsevier, v. 25, n. 1, p. 29–39, 1977.

FRANGI, A.; NOVATI, G. Symmetric be method in two-dimensional elasticity: evaluation of double integrals for curved elements. *Computational Mechanics*, Springer, v. 19, n. 2, p. 58–68, 1996.

GRAY, L. Evaluation of hypersingular integrals in the boundary element method. *Mathematical and Computer Modelling*, Elsevier, v. 15, n. 3-5, p. 165–174, 1991.

HARTMANN, F.; KATZ, C.; PROTOPSALTIS, B. Boundary elements and symmetry. *Ingenieur-Archiv*, Springer, v. 55, n. 6, p. 440–449, 1985.

KANE, J.; KEYES, D.; PRASAD, K. G. Iterative solution techniques in boundary element analysis. *International journal for numerical methods in engineering*, Wiley Online Library, v. 31, n. 8, p. 1511–1536, 1991.

KANE, J. H. *Boundary element analysis in engineering continuum mechanics*. [S.l.]: Prentice-Hall Englewood Cliffs, NJ, 1994.

LIU, Y. *Fast multipole boundary element method: theory and applications in engineering*. [S.l.]: Cambridge university press, 2009.

MAIER, G.; DILIGENTI, M.; CARINI, A. A variational approach to boundary element elastodynamic analysis and extension to multidomain problems. *Computer Methods in Applied Mechanics and Engineering*, Elsevier, v. 92, n. 2, p. 193–213, 1991.

MAIER, G.; POLIZZOTTO, C. A galerkin approach to boundary element elastoplastic analysis. *Computer Methods in Applied Mechanics and Engineering*, Elsevier, v. 60, n. 2, p. 175–194, 1987.

MARTIN, P.; RIZZO, F. Hypersingular integrals: how smooth must the density be? *International Journal for Numerical Methods in Engineering*, Wiley Online Library, v. 39, n. 4, p. 687–704, 1996.

MARTIN, P.; RIZZO, F.; CRUSE, T. Smoothness–relaxation strategies for singular and hypersingular integral equations. *International Journal for Numerical Methods in Engineering*, Wiley Online Library, v. 42, n. 5, p. 885–906, 1998.

RICHARDSON, J.; CRUSE, T. Nonsingular bem for fracture modeling. *Computers & structures*, Elsevier, v. 66, n. 5, p. 695–703, 1998.

SCHATZ, A. H.; THOMÉE, V.; WENDLAND, W. L. *Mathematical theory of finite and boundary element methods*. [S.l.]: Birkhäuser Basel, 1990.

SIRTORI, S. General stress analysis method by means of integral equations and boundary elements. *Meccanica*, Springer, v. 14, n. 4, p. 210–218, 1979.

# ANEXOS

## I.1 Programa Principal Polia - Python

```
1 from IPython import get_ipython
2 get_ipython().magic('reset -sf')
3
4 import time
5
6 import bempp.api
7 import numpy as np
8 bempp.api.global_parameters.assembly.boundary_operator_assembly_type="
9                                     dense"
10 bempp.api.global_parameters.quadrature.medium.double_order = 4
11 bempp.api.global_parameters.quadrature.far.double_order = 4
12
13 import Mesh
14 file = 'Polia1';
15 dirichlet_segments = [32,3,7]
16 mymesh = Mesh.Mesh()
17
18 mymesh.read_msh(file + '.msh')
19
20
21 XYZ = mymesh.Verts
22 tri = mymesh.Elmts[2][1]-1
23 surf = mymesh.Elmts[2][0]
24
25
26 domain_index=surf
27 segments=np.unique(surf)
28
29
30 def dirichlet_data_fun(x, domain_index) :
31     if(domain_index==32) :
32         return 220
33     else:
34         return 0
35
```

```

36 def neumann_data_fun(x):
37     return 0
38
39 order_neumann = 0
40 order_dirichlet = 1
41
42 neumann_segments=[iseq for iseg in segments if iseg not in
43                   dirichlet_segments]
44
45 grid= bempp.api.grid_from_element_data(XYZ.transpose(), tri.transpose(),
46                                       domain_index)
47
48 global_neumann_space = bempp.api.function_space(grid, "DP", order_neumann
49                                                )
50 global_dirichlet_space = bempp.api.function_space(grid, "P",
51                                                  order_dirichlet)
52
53 dirichlet_space_neumann_segment = bempp.api.function_space(
54     grid, "P", order_dirichlet, domains=neumann_segments, closed=False)
55
56 dual_dirichlet_space = bempp.api.function_space(
57     grid, "P", order_dirichlet, domains=dirichlet_segments,
58     closed=True, strictly_on_segment=True)
59
60 neumann_space_dirichlet_segment = bempp.api.function_space(
61     grid, "DP", order_neumann, domains=dirichlet_segments,
62     closed=True, element_on_segment=True)
63
64 neumann_space_neumann_segment = bempp.api.function_space(
65     grid, "DP", order_neumann, domains=neumann_segments,
66     closed=False, element_on_segment=True, reference_point_on_segment=
67     False)
68
69 dirichlet_space_dirichlet_segment = bempp.api.function_space(
70     grid, "P", order_dirichlet, domains=dirichlet_segments, closed=True)
71
72 slp_DD = bempp.api.operators.boundary.laplace.single_layer(
73     neumann_space_dirichlet_segment,
74     dirichlet_space_dirichlet_segment,
75     neumann_space_dirichlet_segment) # used in "blocked"
76
77 dlp_DN = bempp.api.operators.boundary.laplace.double_layer(
78     dirichlet_space_neumann_segment,
79     dirichlet_space_dirichlet_segment,
80     neumann_space_dirichlet_segment) # used in "blocked"
81
82 adlp_ND = bempp.api.operators.boundary.laplace.adjoint_double_layer(
83     neumann_space_dirichlet_segment,

```

```

80     neumann_space_neumann_segment,
81     dirichlet_space_neumann_segment) # used in "blocked"
82
83 hyp_NN = bempp.api.operators.boundary.laplace.hypersingular(
84     dirichlet_space_neumann_segment,
85     neumann_space_neumann_segment,
86     dirichlet_space_neumann_segment) # used in "blocked"
87
88 slp_DN = bempp.api.operators.boundary.laplace.single_layer(
89     neumann_space_neumann_segment,
90     dirichlet_space_dirichlet_segment,
91     neumann_space_dirichlet_segment) # not used in "blocked"
92
93 dlp_DD = bempp.api.operators.boundary.laplace.double_layer(
94     dirichlet_space_dirichlet_segment,
95     dirichlet_space_dirichlet_segment,
96     neumann_space_dirichlet_segment) # not used in "blocked"
97
98 id_DD = bempp.api.operators.boundary.sparse.identity(
99     dirichlet_space_dirichlet_segment,
100    dirichlet_space_dirichlet_segment,
101    neumann_space_dirichlet_segment) # not used in "blocked"
102
103 adlp_NN = bempp.api.operators.boundary.laplace.adjoint_double_layer(
104    neumann_space_neumann_segment,
105    neumann_space_neumann_segment,
106    dirichlet_space_neumann_segment) # not used in "blocked"
107
108 id_NN = bempp.api.operators.boundary.sparse.identity(
109    neumann_space_neumann_segment,
110    neumann_space_neumann_segment,
111    dirichlet_space_neumann_segment) # not used in "blocked"
112
113 hyp_ND = bempp.api.operators.boundary.laplace.hypersingular(
114    dirichlet_space_dirichlet_segment,
115    neumann_space_neumann_segment,
116    dirichlet_space_neumann_segment) # not used in "blocked"
117
118 blocked = bempp.api.BlockedOperator(2, 2)
119
120 blocked[0, 0] = slp_DD
121 blocked[0, 1] = -dlp_DN
122 blocked[1, 0] = adlp_ND
123 blocked[1, 1] = hyp_NN
124
125
126
127 def dirichlet_data(x, n, domain_index, res):
128     res[0] = dirichlet_data_fun(x, domain_index)

```



```

129
130
131 def neumann_data(x, n, domain_index, res):
132     res[0] = neumann_data_fun(x)
133
134 dirichlet_grid_fun = bempp.api.GridFunction(
135     dirichlet_space_dirichlet_segment,
136     fun=dirichlet_data, dual_space=dual_dirichlet_space)
137
138 neumann_grid_fun = bempp.api.GridFunction(
139     neumann_space_neumann_segment,
140     fun=neumann_data, dual_space=dirichlet_space_neumann_segment)
141 start_time = time.time()
142 rhs_fun1 = (.5 * id_DD + dlp_DD) * dirichlet_grid_fun \
143     - slp_DN * neumann_grid_fun
144
145 rhs_fun2 = - hyp_ND * dirichlet_grid_fun \
146     + (.5 * id_NN - adlp_NN) * neumann_grid_fun
147
148 lhs = blocked.weak_form()
149 rhs = np.hstack([rhs_fun1.projections(neumann_space_dirichlet_segment),
150     rhs_fun2.projections(dirichlet_space_neumann_segment)])
151 elapsed_time1= time.time() - start_time
152 from scipy.sparse.linalg import gmres
153 start_time = time.time()
154
155 x, info = gmres(lhs, rhs)
156 elapsed_time2= time.time() - start_time
157
158 nx0 = neumann_space_dirichlet_segment.global_dof_count
159
160 neumann_solution = bempp.api.GridFunction(
161     neumann_space_dirichlet_segment, coefficients=x[:nx0])
162 dirichlet_solution = bempp.api.GridFunction(
163     dirichlet_space_neumann_segment, coefficients=x[nx0:])
164
165 # bempp.api.PLOT_BACKEND = "ipython_notebook"
166
167 neumann_imbedding_dirichlet_segment = \
168     bempp.api.operators.boundary.sparse.identity(
169     neumann_space_dirichlet_segment,
170     global_neumann_space,
171     global_neumann_space)
172
173 neumann_imbedding_neumann_segment = \
174     bempp.api.operators.boundary.sparse.identity(
175     neumann_space_neumann_segment,
176     global_neumann_space,
177     global_neumann_space)

```

```

178
179 dirichlet_imbedding_dirichlet_segment = \
180     bempp.api.operators.boundary.sparse.identity(
181         dirichlet_space_dirichlet_segment,
182         global_dirichlet_space,
183         global_dirichlet_space)
184
185 dirichlet_imbedding_neumann_segment = \
186     bempp.api.operators.boundary.sparse.identity(
187         dirichlet_space_neumann_segment,
188         global_dirichlet_space,
189         global_dirichlet_space)
190
191 dirichlet = (dirichlet_imbedding_dirichlet_segment * dirichlet_grid_fun +
192             dirichlet_imbedding_neumann_segment * dirichlet_solution)
193
194 neumann = (neumann_imbedding_neumann_segment * neumann_grid_fun +
195            neumann_imbedding_dirichlet_segment * neumann_solution)
196 dirichlet.plot()
197
198 unknown_dofs=x.shape[0]
199 elements = grid.leaf_view.entity_count(0)
200 number_of_global_neumann_dofs = global_neumann_space.global_dof_count
201 number_of_global_dirichlet_dofs = global_dirichlet_space.global_dof_count
202 total_time=elapsed_time1+elapsed_time2
203
204 print("time to assembly matrices blocks: "+str(elapsed_time1))
205 print("time to solve linear system: "+str(elapsed_time2))
206 print("total time: "+str(total_time))
207 print("Number of elements "+str(elements))
208 print("Number of global Neumann dofs: "+str(number_of_global_neumann_dofs
209                                           ))
210 print("Number of global Dirichlet dofs: "+str(
211                                           number_of_global_dirichlet_dofs))
212 print("Number of unknown dofs: "+str(unknown_dofs))

```