

Universidade Federal Fluminense

BRUNO DE OLIVEIRA CHAGAS

**Métodos de Elementos Finitos e Diferenças Finitas  
para a equação de Helmholtz**

Volta Redonda

2013

BRUNO DE OLIVEIRA CHAGAS

# Métodos de Elementos Finitos e Diferenças Finitas para a equação de Helmholtz

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Modelagem Computacional.

Orientador:

Gustavo Benitez Alvarez

Coorientador:

Diomar Cesar Lobão  
Emerson Souza Freire

UNIVERSIDADE FEDERAL FLUMINENSE

Volta Redonda

2013

C434 Chagas, Bruno de Oliveira.

Métodos de elementos finitos e diferenças finitas para a equação de Helmholtz. / Bruno de Oliveira Chagas. – Volta Redonda, 2013.

107 f.

Dissertação (Mestrado em Engenharia Metalúrgica) –  
Universidade Federal Fluminense.

Orientador: Gustavo Benitez Alvarez.

1. Equações diferenciais parciais. 2. Equação de Helmholtz.  
3. Método de elementos finitos. 4. Elementos finitos estabilizados.  
5. Método de diferenças finitas. 6. Análise numérica. I. Alvarez,  
Gustavo Benitez. II. Título.

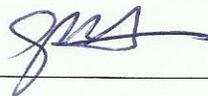
CDD 519.4

# Métodos de Elementos Finitos e Diferenças Finitas para a equação de Helmholtz

Bruno de Oliveira Chagas

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Modelagem Computacional.

Aprovada por:



---

Prof. Gustavo Benitez Alvarez, D.Sc / MCCT-UFF (Presidente)



---

Prof. Emerson Souza Freire, D.Sc. / MCCT-UFF



---

Prof. Wellington Gomes Dantas, D.Sc / MCCT-UFF



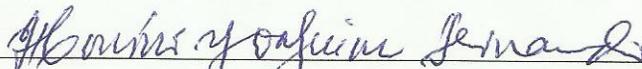
---

Prof. Panters Rodríguez Bermúdez, D.Sc. / MCCT-UFF



---

Prof. Abimael Fernando Dourado Loula, D.Sc. / LNCC



---

Prof. Honório Joaquim Fernando, D.Sc. / UFF

Volta Redonda, 28 de Agosto de 2013.

*À minha família, com muito orgulho e satisfação.*

# Agradecimentos

Dedico meu tempo investido e cada página dessa dissertação a Deus, o substrato da minha existência.

À minha família, em especial à minha mãe Maria da Penha, meu pai Francisco Chagas e meu irmão Júlio Victor.

Aos meus orientadores, pela paciência e inteligência.

Aos meus colegas e amigos do MCCT.

# Resumo

Os métodos clássicos de elementos finitos e diferenças finitas, quando aplicados à equação de Helmholtz, apresentam o que é chamado de efeito de *poluição* do erro, comprometendo seriamente a qualidade da solução aproximada. Em virtude desse desafio numérico, foram desenvolvidos, nas últimas décadas, uma série de métodos que são capazes de contornar esse problema, minimizando o erro gerado por este efeito. Inicialmente, mostra-se como a *poluição* se comporta no método de elementos finitos de Galerkin e diferenças finitas centradas. Posteriormente, são apresentados dois métodos que tratam, ou minimizam, o erro de *poluição*: GLS (*Galerkin Least Squares*) e QSFEM (*Quasi Stabilized Finite Element Method*). Todos os métodos apresentados são ilustrados com seus respectivos resultados numéricos e serão feitas as comparações devidas entre eles.

# Abstract

The classical methods of finite element and finite differences, when applied to Helmholtz equation, present what we call pollution effect, compromising seriously the quality of the approximated solution. Because that numerical challenge, it was developed in the last decades a serie of methods capable to outline that obstacle, minimizing the error generated by pollution effect. Initially we will show how the pollution effect behaves in the finite element method of Galerkin and centered finite differences. Posteriorly, we will present three methods that deal, or minimize, the pollution error: GLS (Galerkin Least Squares) e QSFEM (Quasi Stabilized Finite Element Method). All methods presented will be illustrated with their respectives numerical results and we will do the due comparisions to each other.

# Palavras-chave

1. Equações Diferenciais Parciais
2. Equação de Helmholtz
3. Método de Elementos Finitos
4. Elementos Finitos Estabilizados
5. Método de Diferenças Finitas
6. Análise Numérica

# Glossário

- MDFC : Método de Diferenças Finitas Centradas  
GLS : *Galerkin Least-Squares*  
QSFEM : *Quasi Stabilized Finite Element Method*  
GPR : *Galerkin Projected Residual Finite Element Method*

# Sumário

|   |            |
|---|------------|
| <b>Lista de Figuras</b>   | <b>xii</b> |
| <b>1 Introdução</b>   | <b>16</b>  |
| <b>2 O problema de Helmholtz</b>  | <b>18</b>  |
| 2.1 Ondas acústicas . . . . .   | 18         |
| 2.2 Formulação forte do problema . . . . .                                  | 20         |
| 2.2.1 Soluções em uma dimensão . . . . .                                    | 22         |
| 2.2.2 Soluções em duas dimensões - ondas planas . . . . .                   | 23         |
| 2.3 Formulação fraca do problema . . . . .                                  | 25         |
| 2.3.1 Condições de Dirichlet . . . . .                                      | 26         |
| 2.3.2 Condições de Robin . . . . .  | 26         |
| <b>3 Solução numérica por diferenças finitas e elementos finitos</b>        | <b>28</b>  |
| 3.1 Diferenças Finitas Centradas . . . . .                                  | 28         |
| 3.1.1 Em uma dimensão . . . . .   | 29         |
| 3.1.2 Em duas dimensões . . . . .   | 30         |
| 3.2 Elementos Finitos de Galerkin . . . . .                                 | 32         |
| 3.2.1 Formulação geral . . . . .  | 32         |
| 3.2.2 Espaço de funções lineares por partes - caso unidimensional . . . . . | 33         |
| 3.2.3 Espaço de funções lineares por partes - caso bidimensional . . . . .  | 36         |
| 3.3 Análise de Dispersão . . . . .  | 39         |
| 3.3.1 Caso unidimensional . . . . .   | 39         |

---

|          |  |           |
|----------|--|-----------|
| 3.3.2    | Caso bidimensional . . . . .                                     | 41        |
| 3.4      | Galerkin Mínimos Quadrados (GLS) . . . . .                       | 41        |
| 3.5      | Método de Elementos Finitos Quase Estabilizado (QSFEM) . . . . . | 43        |
| 3.6      | Ressonância . . . . .  | 46        |
| <b>4</b> | <b>Resultados Numéricos</b>                                      | <b>48</b> |
| 4.1      | Implementação Computacional . . . . .                            | 48        |
| 4.2      | Análise unidimensional . . . . .                                 | 50        |
| 4.2.1    | Interpolante e regra de aproximação . . . . .                    | 51        |
| 4.2.2    | Efeito de poluição do erro . . . . .                             | 53        |
| 4.2.3    | Análise de Erro . . . . .  | 59        |
| 4.3      | Análise bidimensional . . . . .                                  | 64        |
| 4.3.1    | Efeito de Poluição do erro . . . . .                             | 64        |
| 4.3.2    | Análise de Erro . . . . .  | 69        |
| <b>5</b> | <b>Conclusões e Trabalhos Futuros</b>                            | <b>73</b> |
| 5.1      | Conclusões . . . . .   | 73        |
| 5.2      | Trabalhos Futuros . . . . .                                      | 74        |
|          | <b>Referências</b>   | <b>75</b> |
|          | <b>Apêndice A - Elementos de Análise Funcional</b>               | <b>77</b> |
| A.1      | Norma e produto interno . . . . .                                | 77        |
| A.2      | Espaços de Lebesgue . . . . .                                    | 78        |
| A.3      | Espaços de Hilbert . . . . .                                     | 78        |
| A.4      | Derivadas forte e fraca . . . . .                                | 79        |
| A.5      | Espaço $H^1$ . . . . .   | 81        |
| A.6      | Construção de espaços por completamento . . . . .                | 81        |

---

|  |   |           |
|--|---|-----------|
| A.7  | Formas sesquilineares e operadores lineares . . . . . | 82        |
| A.8  | Existência e unicidade de soluções . . . . .          | 83        |
| <b>Apêndice B - Códigos Computacionais</b> |   | <b>85</b> |
| B.1  | Códigos dos métodos em uma dimensão . . . . .         | 85        |
| B.2  | Códigos dos métodos em duas dimensões . . . . .       | 90        |

# Lista de Figuras

|     |   |    |
|-----|---|----|
| 2.1 | Volume de controle . . . . .  | 19 |
| 3.1 | Domínio discreto . . . . .  | 29 |
| 3.2 | Domínio discreto com $h_x = h_y = h$ . . . . .  | 31 |
| 3.3 | Funções base para um elemento. . . . .  | 34 |
| 3.4 | Numeração dos elementos e suas coordenadas . . . . .  | 37 |
| 3.5 | Numeração Global e Local dos nós . . . . .  | 37 |
| 4.1 | Número de onda $k = 30$ no método de Galerkin 1D, com 30 simulações para cada resolução de malha, variando de 100 a 1000, em intervalos de 10. . . . .                              | 49 |
| 4.2 | Número de onda $k = 1$ no método de Galerkin 2D, com 15 simulações para cada resolução de malha, variando de 100 a 3600, em intervalos de 25. . . . .                               | 50 |
| 4.3 | Regra do <i>Thumb</i> para interpolação com $n_{res} = 8$ . . . . .   | 52 |
| 4.4 | Aproximação por MDFC com número de onda $k=30$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . A resolução da malha tem $n_{res} = 10$ , ou seja, $kh \approx 0.6$ . . . . .     | 53 |
| 4.5 | Aproximação por MDFC com número de onda $k=90$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . A resolução da malha tem $n_{res} = 10$ , ou seja, $kh \approx 0.6$ . . . . .     | 54 |
| 4.6 | Aproximação por MDFC com número de onda $k=120$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . A resolução da malha tem $n_{res} = 10$ , ou seja, $kh \approx 0.6$ . . . . .    | 54 |
| 4.7 | Aproximação por Galerkin com número de onda $k=30$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . A resolução da malha tem $n_{res} = 10$ , ou seja, $kh \approx 0.6$ . . . . . | 55 |

|      |  |    |
|------|--|----|
| 4.8  | Aproximação por Galerkin com número de onda $k=90$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . A resolução da malha tem $n_{res} = 10$ , ou seja, $kh \approx 0.6$ . . . . .  | 55 |
| 4.9  | Aproximação por Galerkin com número de onda $k=120$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . A resolução da malha tem $n_{res} = 10$ , ou seja, $kh \approx 0.6$ . . . . . | 56 |
| 4.10 | Aproximação por GLS com número de onda $k=30$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . A resolução da malha tem $n_{res} = 10$ , ou seja, $kh \approx 0.6$ . . . . .       | 56 |
| 4.11 | Aproximação por GLS com número de onda $k=90$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . A resolução da malha tem $n_{res} = 10$ , ou seja, $kh \approx 0.6$ . . . . .       | 57 |
| 4.12 | Aproximação por GLS com número de onda $k=120$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . A resolução da malha tem $n_{res} = 10$ , ou seja, $kh \approx 0.6$ . . . . .      | 57 |
| 4.13 | Aproximação por MDFC para o caso não homogêneo de $f(x) = k^2x$ , para $k=80$ , com condição de Dirichlet $u(0) = 0$ e $u(1) = -3$ , considerando $kh \approx 0.6$ . . . . .         | 58 |
| 4.14 | Aproximação por MDFC para o caso não homogêneo de $f(x) = k^2x$ , para $k=80$ , com condição de Dirichlet $u(0) = 0$ e $u(1) = -3$ , considerando $kh \approx 0.6$ . . . . .         | 58 |
| 4.15 | Aproximação GLS para o caso não homogêneo de $f(x) = k^2x$ , para $k=80$ , com condição de Dirichlet $u(0) = 0$ e $u(1) = -3$ , considerando $kh \approx 0.6$ . . . . .              | 59 |
| 4.16 | Gráfico do erro relativo do caso homogêneo na norma $L^2$ para $k=60$ . . . . .  | 59 |
| 4.17 | Gráfico do erro relativo do caso homogêneo na seminorma $H^1$ para $k=60$ . . . . .  | 60 |
| 4.18 | Gráfico do erro relativo do caso homogêneo na norma $H^1$ para $k = 200$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . . . . .  | 60 |
| 4.19 | Gráfico do erro relativo do caso não homogêneo, com $f(x) = k^2x$ na norma $H^1$ para $k=60$ em condição de Dirichlet $u(0) = 0$ e $u(1) = -3$ . . . . .                             | 61 |
| 4.20 | Erro na seminorma $H^1$ para o problema homogêneo mantendo-se a relação $kh = 0.2$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . . . . .  | 61 |
| 4.21 | Erro na seminorma $H^1$ para o problema homogêneo mantendo-se a relação $k^2h = 0.2$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . . . . .                                      | 62 |

|      |   |    |
|------|---|----|
| 4.22 | Erro na seminorma $H^1$ para o problema homogêneo mantendo-se a relação $k^3h^2 = 0.2$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . . . . .   | 62 |
| 4.23 | Erro na norma $L^2$ para o problema homogêneo mantendo-se a relação $k^3h^2 = 0.2$ em condição de Dirichlet $u(0) = 0$ e $u(1) = 1$ . . . . .   | 63 |
| 4.24 | As figuras (a) e (b) apresentam a solução exata para uma onda plana com $k = 30$ e $k = 50$ , respectivamente. As figuras (c) e (d) são para duas e três ondas planas, respectivamente, e ambas para $k = 30$ . . . . . | 64 |
| 4.25 | Aproximação por MDFC com número de onda $k=50$ em condição de Dirichlet, com corte em $y = 0,4792$ , para uma onda plana com direção $\theta_1 = 0$ e malha $80 \times 80$ . . . . .                                    | 65 |
| 4.26 | Aproximação por MDFC com número de onda $k=70$ em condição de Dirichlet, com corte em $y = 0,4792$ , para uma onda plana com direção $\theta_1 = 0$ e e malha $111 \times 111$ . . . . .                                | 66 |
| 4.27 | Aproximação de Galerkin com número de onda $k=50$ em condição de Dirichlet, com corte em $y = 0,4792$ , para uma onda plana com direção $\theta_1 = 0$ e malha $80 \times 80$ . . . . .                                 | 66 |
| 4.28 | Aproximação de GLS com número de onda $k=30$ em condição de Dirichlet, com corte em $y = 0,4792$ , para uma onda plana com direção $\theta_1 = 3\pi/8$ e malha $49 \times 49$ . . . . .                                 | 67 |
| 4.29 | Aproximação de GLS com número de onda $k=30$ em condição de Dirichlet, com corte em $y = 0,4792$ , para uma onda plana com direção $\theta_1 = 0$ e malha $49 \times 49$ . . . . .                                      | 67 |
| 4.30 | Aproximação de QSFEM com número de onda $k=30$ em condição de Dirichlet, com corte em $y = 0,4792$ , para uma onda plana com direção $\theta_1 = \pi/16$ e malha $49 \times 49$ . . . . .                               | 68 |
| 4.31 | Aproximação de QSFEM com número de onda $k=30$ em condição de Dirichlet, com corte em $y = 0,4792$ , para uma onda plana com direção $\theta_1 = \pi/8$ e malha $49 \times 49$ . . . . .                                | 68 |
| 4.32 | Gráfico do erro relativo na norma $H^1$ considerando três ondas planas na mesma direção, com $k = 80$ e malha $200 \times 200$ . O ângulo de direção da onda varia de $0$ a $\pi/2$ . . . . .                           | 69 |

- 
- 4.33 Resultado igual ao da figura (4.32) mas considerando apenas o interpolante e o método QSFEM . . . . . 70
- 4.34 Gráfico do erro relativo na norma  $L^2$  considerando três ondas planas, duas fixadas em  $\theta_1 = \pi/4$  e  $\theta_2 = 0$ , e uma variando de 0 a  $\pi/2$ , com  $k = 80$  e malha  $200 \times 200$  . . . . . 70
- 4.35 Gráfico do erro relativo, considerando caso homogêneo, do problema bidimensional para a norma  $H^1$ . E tem-se os parâmetros  $k = 30$ , uma onda plana na direção  $\theta_1 = 0$  e malha variando de  $50 \times 50$  até  $150 \times 150$ . . . . . 71
- 4.36 Gráfico do erro relativo, considerando caso homogêneo, do problema bidimensional para a norma  $H^1$ . E tem-se os parâmetros  $k = 30$ , três ondas planas nas direções  $\theta_1 = 0$ ,  $\theta_2 = \pi/8$ ,  $\theta_3 = \pi/4$  e malha variando de  $50 \times 50$  até  $150 \times 150$ . . . . . 71
- 4.37 Gráfico do erro relativo, considerando caso homogêneo, do problema bidimensional para a seminorma  $H^1$  em (a) e norma  $L^2$  em (b). E tem-se os parâmetros  $k = 110$ , uma onda plana com  $\theta = 0$  e malha variando de  $100 \times 100$  até  $400 \times 400$ . . . . . 72

# Capítulo 1

## Introdução

A equação de Helmholtz tem aplicações em problemas lineares de propagação de ondas harmônicas. Modela-se por esta equação ondas acústicas, ondas elásticas, interação fluido-sólido e sistemas/fenômenos eletromagnetismo [15]. De acordo com a aplicação, a equação de Helmholtz pode ser usada em problemas diretos ou inversos, por meio de soluções numéricas.

Todo fenômeno físico possui suas particularidades, características intrínsecas, que denotam o seu comportamento. Por se tratar de uma equação que modela fenômenos ondulatórios, é de se esperar uma natureza oscilatória das soluções. Quando se buscam soluções numéricas para este problema, deve-se então ajustar a distância  $h$  entre os nós da malha ao número de onda  $k$ , numa regra do tipo " $kh$  constante", para que em cada oscilação tenha-se um número mínimo de pontos que sejam capazes de capturar a solução numérica aproximada.

No entanto, as expectativas analíticas não são satisfeitas quando o dado problema é submetido ao método de Galerkin [17], em sua formulação clássica. Repara-se que para números elevados de  $k$  a regra  $kh$  constante não nos é suficiente para controlar o erro: testes computacionais, bem como a análise numérica, comprovam a verossimilhança do fato [16]. Este comportamento em relação ao  $k$  é chamado *efeito de poluição* do erro, pois o número de onda  $\tilde{k}$  aproximado difere-se do número de onda  $k$  exato. Não obstante, encontra-se esse mesmo desafio no método de diferenças finitas [13, 25]. Portanto, o objetivo deste trabalho é de analisar o *efeito de poluição* do erro, primeiramente, pelos métodos de diferenças finitas centradas e elementos finitos de Galerkin. Após essa análise, vê-se que estes dois primeiros métodos não possuem uma boa aproximação da solução tanto para problema em uma dimensão quanto em duas, mantendo-se a regra  $kh$  igual uma constante. Dessa forma, procuram-se métodos que eliminem esse *efeito de poluição*

do erro ou pelo menos minimizem.

Na busca por esses métodos, o capítulo 2 apresenta um caráter mais físico e matemático para o problema de Helmholtz. A primeira parte desse capítulo, consiste em se deduzir a equação de Helmholtz através da equação da onda, assumindo uma vertente mais de modelagem do fenômeno em si, chegando em sua formulação forte. Após delinear o fenômeno, é natural buscar-se soluções analíticas que porventura existam. Numa segunda parte deste capítulo, define-se o que é chamado de formulação fraca de uma equação, ou forma variacional, que será essencial para a formulação de elementos finitos.

O capítulo 3, por sua vez, é dedicado aos métodos trabalhados e implementados no presente problema. O primeiro método apresentado é o de diferenças finitas centradas, por se tratar de um método mais antigo, em relação aos outros que serão utilizados, e de fácil implementação. O segundo método é o de elementos finitos na formulação de Galerkin, necessitando da formulação fraca para que este seja definido. Outra parte é dedicada à análise de dispersão, mostrando claramente o efeito de poluição dos métodos anteriores. Conclui-se com dois métodos que são capazes de proporcionar melhores resultados, que são: GLS [12] e QSFEM [4].

O capítulo 4 tem o objetivo de mostrar os resultados numéricos pelos métodos tratados no capítulo anterior. Uma análise a ser considerada é a relação número de onda  $k$  e o refino da malha, ou seja, quantos elementos, ou nós, deve-se considerar para um controle robusto do erro. Sendo mais conclusivo, dado um  $k$  qualquer, procura-se descobrir o valor de  $h$  para que a solução convirja assintoticamente. Explora-se também o erro da solução aproximada, com relação às soluções exatas que foram obtidas no capítulo 2, nas normas dos espaços  $H^1$  e  $L^2$ , onde nosso problema está bem definido.

Por fim, o capítulo 5 tem o finalidade de mostrar os objetivos alcançados ao longo do trabalho e também elucidar alguns pontos de pesquisa para trabalhos futuros.

# Capítulo 2

## O problema de Helmholtz

Este capítulo visa estabelecer duas formulações matemáticas para o problema de Helmholtz e que serão base para a formulação dos métodos numéricos do capítulo 3. A primeira formulação será a forte, ou clássica, e que, para este trabalho, parte da equação da onda em fenômenos acústicos. A segunda formulação é conhecida como fraca, ou variacional, e que tem como ponto de partida a formulação forte. Serão conhecidas, também, algumas soluções analíticas para o presente problema e que serão necessárias no capítulo 4.

### 2.1 Ondas acústicas

Define-se ondas acústicas, ou som, como a variação de pressão em um fluido ideal, sendo este necessariamente compressível (densidade pode variar temporalmente). Para tanto, define-se um volume de controle  $V = \Omega$ , com contorno  $\partial V = \partial\Omega$  e um fluxo de um fluido com densidade  $\rho(x, t)$ , pressão  $P(x, t)$  e velocidade  $v(x, t)$  na direção do vetor unitário  $n(x)$ , exterior a  $V$ , onde  $x \in \mathbb{R}^d$  com  $d = 1, 2, 3$ , conforme ilustra a Fig. 2.1 para  $d = 3$ .

A velocidade do fluxo normal, através do contorno  $\partial V$ , é dada por  $v(x, t) \cdot n(x)$ . Por tais características, a conservação de massa por unidade de tempo é expressa pela relação

$$-\frac{\partial}{\partial t} \int_{\Omega} \rho d\Omega = \oint_{\partial\Omega} \rho(v \cdot n) d\partial\Omega. \quad (2.1)$$

A interpretação física da equação (2.1) é de que há um fluxo de entrada e outro de saída, por isso há sinais opostos nos dois lados da igualdade. O termo da esquerda da identidade representa a massa do volume de controle variando com o tempo. O termo da

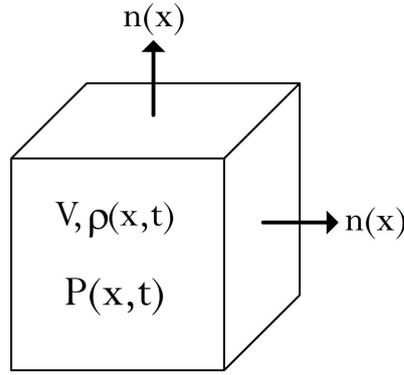


Figura 2.1: Volume de controle

direita mostra o fluxo de entrada, ou saída, através da fronteira. Já a igualdade refere-se justamente a esse balanço, a conservação propriamente dita.

A integral de superfície, na identidade acima, pode ser transformada em uma integral de volume, segundo o teorema de Gauss, na forma

$$\oint_{\partial\Omega} (\rho v) \cdot n d\Omega = \int_{\Omega} \text{div}(\rho v) d\Omega. \quad (2.2)$$

Segue-se diretamente das equações (2.1) e (2.2) que

$$\int_{\Omega} \left( \frac{\partial \rho}{\partial t} + \text{div}(\rho v) \right) d\Omega = 0.$$

Pela igualdade anterior obtém-se finalmente a equação da continuidade

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho v) = 0. \quad (2.3)$$

Assumindo agora que o mesmo volume de controle  $V$  está sujeito à pressão hidrostática  $P(x, t)$ , pode-se identificar a força ao longo de  $\partial V$  como

$$F = - \oint_{\partial\Omega} P n d\Omega.$$

E pela segunda lei de Newton ( $F = ma$ ) tem-se ainda que

$$- \oint_{\partial\Omega} P n d\Omega = \int_{\Omega} \rho \frac{dv}{dt} d\Omega. \quad (2.4)$$

De forma análoga, usa-se o teorema de Gauss para transformar a integral de superfície

em uma de volume. Assim, obtém-se a identidade

$$-\oint_{\partial\Omega} P n d\Omega = \int_{\Omega} \nabla P d\Omega, \quad (2.5)$$

onde  $\nabla$  é o operador (gradiente) em coordenadas cartesianas.

Sabe-se que a diferencial total é expressa por

$$\frac{dv}{dt} = \frac{\partial v}{\partial t} + (v \cdot \nabla)v.$$

Assumindo pequenas oscilações no campo de velocidades  $v$ , pode-se linearizar o diferencial, conforme a referência [15], ficando somente com a parte linear

$$\frac{dv}{dt} \approx \frac{\partial v}{\partial t}.$$

Pelas equações (2.4), (2.5) e pela linearização do diferencial total, obtém-se o que é chamado de equação de Euler ou de movimento:

$$\rho \frac{\partial v}{\partial t} = -\nabla P. \quad (2.6)$$

## 2.2 Formulação forte do problema

Por definição, som é uma pequena perturbação  $(P, \rho)$  dos campos de pressão e densidade de um estado constante  $(P_0, \rho_0)$  em um fluido ideal e compressível [15]. Em um certo ponto  $x$ , as funções  $P(x, t)$ ,  $\rho(x, t)$  representam vibrações com uma pequena amplitude. A relação entre velocidade de propagação de uma onda, densidade e pressão é dada por

$$P = c^2 \rho,$$

onde a constante  $c$  é chamada de velocidade do som [29]. Então, fazendo uso das versões linearizadas de (2.3) e (2.6), obtém-se

$$P_{tt} = c^2 \rho_{tt} = -c^2 \rho_0 \operatorname{div}(V_t) = c^2 \operatorname{div}(\nabla P),$$

que conduz à equação da onda

$$\Delta P - \frac{1}{c^2} P_{tt} = 0, \quad (2.7)$$

onde  $\Delta = \nabla \cdot \nabla$  é o operador Laplaciano em coordenadas espaciais.

Assume-se que a equação (2.7) tenha soluções do tipo

$$P(x, t) = u(x)e^{-i\omega t}, \quad (2.8)$$

chamadas de harmônicos temporais, onde  $u(x)$  é a parte espacial, com  $\omega > 0$  sendo a frequência. Desse modo, substituindo a solução na forma de harmônicos temporais (2.8) na equação (2.7), obtém-se:

$$\Delta(u(x)e^{-i\omega t}) - \frac{1}{c^2} \frac{\partial^2(u(x)e^{-i\omega t})}{\partial t^2} = 0.$$

Observa-se que o primeiro termo, com o operador Laplaciano  $\nabla$ , é essencialmente de derivadas espaciais, assim  $e^{-i\omega t}$  é tomada como constante, para cada  $t$  fixo. Já no segundo termo, com derivadas parciais temporais, a função  $u(x)$  é, desta vez, tomada como constante, para cada  $x$  fixo. Assim, é possível escrever:

$$e^{-i\omega t} \Delta(u(x)) + \frac{\omega^2}{c^2} u(x) e^{-i\omega t} = 0.$$

Então, a parte estacionária  $u(x) = u$  satisfaz a equação

$$\Delta u + k^2 u = 0 \quad (2.9)$$

chamada Equação de Helmholtz, onde  $k^2 = \omega^2/c^2$ . As definições anteriores, bem como as deduções, foram tomadas do texto base [15] escrito por Frank Ihlenburg, que deve ser consultado para maiores detalhes.

A equação (2.9) faz parte da formulação clássica, ou forte, do problema de Helmholtz. Nota-se que deve-se ter, no mínimo, segunda derivada da função  $u$ , para que o problema esteja bem definido. O termo, formulação forte, será melhor compreendido quando apresentada a formulação fraca na seção 3 deste mesmo capítulo.

Um ponto de extrema importância e que não foi levantado ainda é o das condições de contorno do problema. Porém, esses dados serão bem colocados juntamente da formulação fraca e lá ficarão estabelecidas tais condições, para que se possa trabalhar com os métodos numéricos do capítulo 3 e suas implementações no capítulo 4.

Propõe-se, agora, buscar soluções da equação de Helmholtz para o problema unidimensional. Recorrer-se ao estudo das Equações Diferenciais Ordinárias seguindo a referência [10]. Logo em seguida, serão estudadas as soluções de ondas planas para o caso bidimensional, sendo essa a base para a análise numérica em capítulos posteriores.

### 2.2.1 Soluções em uma dimensão

Escreve-se a equação de Helmholtz em uma notação típica para equações diferenciais ordinárias

$$u'' + k^2u = 0, \quad (2.10)$$

sendo

$$u'' = \frac{d^2u}{dx^2}.$$

Nota-se, dessa forma, que se trata de uma equação linear homogênea de segunda ordem com coeficientes constantes. Ainda mais, ela é da mesma forma da equação do oscilador harmônico simples. Assim, pode-se esperar soluções que envolvam senos e cossenos. Espera-se soluções com a forma [10]

$$u(x) = e^{\lambda x} \quad (2.11)$$

substituindo a equação (2.11) na (2.10), e já efetuando as derivadas, obtém-se

$$\lambda^2 e^{\lambda x} + k^2 e^{\lambda x} = 0.$$

Sabe-se que  $e^{\lambda x}$  é não nula, portanto pode-se dividir ambos os lados da identidade por ela, resultando

$$\lambda^2 + k^2 = 0.$$

Desta forma,  $\lambda$  pode assumir dois valores:  $\lambda_1 = -ik$  ou  $\lambda_2 = ik$ . Consecutivamente, tem-se duas soluções, uma para cada valor de  $\lambda$  encontrado,

$$u_1(x) = e^{\lambda_1 x} = e^{-ikx} \quad e$$

$$u_2(x) = e^{\lambda_2 x} = e^{ikx}.$$

Observa-se que as soluções encontradas acima são linearmente independentes entre si. Por isto, uma combinação linear delas também será solução da equação (2.10),

$$u(x) = c_1 e^{-ikx} + c_2 e^{ikx}. \quad (2.12)$$

Utilizando a fórmula de Euler,  $e^{ix} = \cos(x) + i\sin(x)$ , e desenvolvendo a equação (2.12), tem-se que

$$u(x) = c_1(\cos(kx) - i\sin(kx)) + c_2(\cos(kx) + i\sin(kx))$$

$$u(x) = (c_1 + c_2)\cos(kx) + i(c_2 - c_1)\sin(kx)$$

sendo  $B = c_1 + c_2$  e  $C = i(c_2 - c_1)$ , chega-se à solução com o formato

$$u(x) = B\cos(kx) + C\sin(kx). \quad (2.13)$$

Substituindo-se (2.13) em (2.10), observa-se que, de fato, ela é solução para o presente problema. E, como enunciado, a solução é uma composição de senos e cossenos. Entretanto a equação (2.13) ainda pode ser escrita, de uma forma mais compacta e simples, como

$$u(x) = A\cos(kx - \phi)$$

onde  $A$  é o afastamento máximo em relação à posição central, chamada de amplitude. O ângulo  $\phi$  é chamado ângulo de fase, que mostra o seu defasamento, ou seja, o quanto a onda está deslocada da origem.

Escolhemos o domínio  $(0, 1)$  por simplicidade e temos o problema como

$$u'' + k^2u = 0 \quad \text{em } (0, 1),$$

com condição de contorno de Dirichlet

$$u(0) = a \quad \text{e} \quad u(1) = b,$$

e de posse da solução geral (2.13), tem-se uma solução para esse dado problema, após as devidas manipulações algébricas, obtém-se, finalmente, a solução

$$u(x) = \frac{a\sin(k - kx) + b\sin(kx)}{\sin(k)}.$$

## 2.2.2 Soluções em duas dimensões - ondas planas

Essa seção é dedicada apenas às soluções na forma de ondas planas. As ondas planas apresentam frequência e amplitude constantes, além de terem uma direção especificada, assim elas são ditas não dispersivas. A solução será obtida através do método de separação de variáveis para coordenadas cartesianas.

Considerando-se a equação de Helmholtz  $\Delta u + k^2u = 0$  em  $\mathbb{R}^3$ , o trabalho será o de procurar soluções não nulas que possam ser escritas como  $u(x, y, z) = X(x)Y(y)Z(z)$ . Esta última identidade é o que caracteriza o método de separação de variáveis. Aceitando-se que será possível encontrar soluções desse tipo, a equação de Helmholtz é reescrita como

$$X''YZ + XY''Z + XYZ'' + k^2XYZ = 0.$$

Como são procuradas soluções não nulas, pode-se dividir pelo produto  $XYZ$  todos os termos da identidade anterior, obtém-se então:

$$-\frac{X''}{X} = \frac{Y''}{Y} + \frac{Z''}{Z} + k^2. \quad (2.14)$$

O lado direito da equação (2.14) não depende de  $x$ , pela forma como foi definido. Portanto, a igualdade segue se ambos os lados forem iguais a uma constante, tal como  $\lambda$ . Por estes fatos estabelecidos, tem-se um novo par de equações

$$X'' + \lambda X = 0, \quad (2.15)$$

$$\frac{Y''}{Y} + \frac{Z''}{Z} + k^2 - \lambda = 0 \quad (2.16)$$

que são satisfeitos simultaneamente. Fazendo o mesmo processo para a equação (2.16), e assumindo uma constante  $\nu$ , as funções  $X, Y$  e  $Z$  satisfazem

$$X'' + \lambda X = 0, \quad (2.17)$$

$$Y'' + \nu Y = 0, \quad (2.18)$$

$$Z'' + (k^2 - \lambda - \nu)Z = 0, \quad (2.19)$$

para certos valores de  $\lambda$  e  $\nu$ . Como procuram-se soluções em propagação de ondas, consideram-se somente valores reais e positivos para estas constantes, sendo assim  $\lambda := \alpha^2$ ,  $\nu := \beta^2$  com  $\alpha, \beta \in \mathbb{R}$ . Então a equação (2.19) pode ser reescrita como

$$Z'' + \gamma^2 Z = 0,$$

com

$$\gamma := \sqrt{k^2 - \alpha^2 - \beta^2}.$$

Para  $k^2 \geq \alpha^2 + \beta^2$ , o parâmetro  $\gamma$  será também real e, então, obtém-se soluções na forma de ondas planas

$$u(x, y, z) = e^{i(\alpha x + \beta y + \gamma z)} \quad (2.20)$$

onde os parâmetros  $\alpha, \beta, \gamma$  satisfazem a relação de dispersão

$$\alpha^2 + \beta^2 + \gamma^2 = k^2. \quad (2.21)$$

Uma forma alternativa de se escrever ondas planas em duas dimensões é considerando uma direção  $\sigma = (\cos(\theta), \sin(\theta))$ . Respeitando a relação de dispersão para duas

dimensões, pode-se escrever a solução para ondas planas como

$$u(x, y) = e^{ik(x\cos(\theta)+y\sin(\theta))}, \quad (2.22)$$

que é facilmente verificada numa relação análoga à (2.20) considerando  $\gamma = 0$ . Usa-se novamente a fórmula de Euler para encontrar uma forma de se escrever a solução (2.22) que será de grande utilidade. A solução apresenta-se como

$$u(x, y) = \cos(k(x\cos(\theta) + y\sin(\theta))) + i\sin(k(x\cos(\theta) + y\sin(\theta))). \quad (2.23)$$

Nota-se que ela possui uma parte real e uma imaginária, ou seja, a solução está no corpo dos complexos. Contudo, como o operador de Helmholtz é linear, somente a parte real da equação (2.23) ainda será solução da equação de Helmholtz. Assim,

$$u(x, y) = \cos(k(x\cos(\theta) + y\sin(\theta))) \quad (2.24)$$

também é solução para o problema de Helmholtz (2.9). E, de forma análoga, pode-se ter uma combinação de  $n$  ondas superpostas e é possível expressar como

$$u_n(x, y) = \sum_{i=1}^n \cos(k(x\cos(\theta_i) + y\sin(\theta_i))), \quad (2.25)$$

sendo  $\theta_i$  a direção de cada uma das ondas.

## 2.3 Formulação fraca do problema

O método de elementos finitos de Galerkin, que será apresentado no capítulo 3, necessita da forma fraca, ou variacional, do problema de Helmholtz. A existência de solução do problema em sua forma fraca é garantida por resultados, cujos enunciados necessitam de uma série de elementos de análise funcional. No apêndice A, encontram-se estas definições e resultados.

Pode-se escrever de uma forma geral o problema de Helmholtz, preservando aspectos dos funcionais, em uma notação usual:

$$\begin{cases} \text{Encontrar } u \in V_1 : \\ b(u, v) = f(v), \quad \forall v \in V_2 \end{cases}$$

sabendo-se que a forma sesquilinear  $b(u, v)$ , o funcional antilinear  $f(v)$  e os espaços  $V_1$  e  $V_2$  variam de acordo com as condições de contorno estabelecidas para o problema. Assim, apresenta-se a forma como é definida o problema de Helmholtz para certas condições de

contorno, bem como os espaços onde é definido.

### 2.3.1 Condições de Dirichlet

O problema de Helmholtz não homogêneo em sua forma forte (clássica), com condições de contorno de Dirichlet, consiste em encontrar  $u$  com segunda derivada contínua, isto é,  $u \in C^2(\Omega)$  tal que

$$-\Delta u - k^2 u = f \quad \text{em } \Omega, \quad (2.26)$$

$$u = g \quad \text{em } \partial\Omega, \quad (2.27)$$

sendo  $\Omega$  o domínio do problema,  $\partial\Omega$  a fronteira ou contorno de  $\Omega$  e  $f$  um termo fonte. Se  $f = 0$  a equação (2.26) é dita homogênea. As equações (2.26) e (2.27) podem ser formuladas numa versão fraca, ou variacional, aplicando-se a primeira identidade de Green (ou Integração por Partes). Assim, a solução do problema na forma fraca, consiste em encontrar  $u \in U$ ,  $\forall v \in V$ , tal que  $a(u, v) = f(v)$ , onde

$$a(u, v) = \int_{\Omega} (\nabla u \cdot \nabla v - k^2 uv) d\Omega \quad (2.28)$$

$$f(v) = \int_{\Omega} f v d\Omega \quad (2.29)$$

então deve-se encontrar  $u \in U$ ,  $\forall v \in V$ , e esses espaços são definidos da forma:

$$U = \{u \in H^1(\Omega) : u = g \quad \text{em } \partial\Omega\}, \quad (2.30)$$

$$V = \{v \in H^1(\Omega) : v = 0 \quad \text{em } \partial\Omega\}. \quad (2.31)$$

Apenas como ressalva, as funções  $v$  são de suporte compacto, isto é  $v \in C_0^\infty(\Omega)$  e que é definida no apêndice A, bem como o espaço  $H^1$ .

### 2.3.2 Condições de Robin

Considerando-se novamente  $u \in C^2(\Omega)$ , um outro tipo de condição de contorno para o problema de Helmholtz, na forma clássica, é dada por

$$-\Delta u - k^2 u = f \quad \text{em } \Omega \quad (2.32)$$

$$\frac{\partial u}{\partial \mathbf{n}} + iku = g \quad \text{em } \partial\Omega, \quad (2.33)$$

onde  $\mathbf{n}$  é um vetor unitário que aponta para fora do interior do domínio  $\Omega$  e  $\mathbf{i} = \sqrt{-1}$ . As equações (2.32) e (2.33) referem-se à formulação clássica (ou forte) do problema, podendo ser expressas na forma variacional como  $a(u, v) = f(v)$ , onde

$$a(u, v) = \int_{\Omega} (\nabla u \cdot \nabla v - k^2 uv) d\Omega + ik \int_{\partial\Omega} uv d\partial\Omega, \quad (2.34)$$

$$f(v) = \int_{\Omega} fvd\Omega + \int_{\partial\Omega} gvd\partial\Omega, \quad (2.35)$$

$$U = V = H^1(\Omega). \quad (2.36)$$

Ao longo dos experimentos numéricos, será usada condição de Dirichlet para os resultados. Essa condição de contorno oferece maiores dificuldades numéricas, devido ao fenômeno de ressonância que será visto no capítulo posterior.

# Capítulo 3

## Solução numérica por diferenças finitas e elementos finitos

Os capítulos anteriores tiveram como objetivo formalizar o problema de Helmholtz em aspectos físicos e matemáticos, necessários para análise e formulação dos métodos numéricos. Neste capítulo será feita uma análise do problema em uma dimensão utilizando o método de diferenças finitas, elementos finitos de Galerkin clássico e o Galerkin Mínimos Quadrados (GLS). Posteriormente, será feita a análise em duas dimensões usando o método de diferenças finitas, elementos finitos de Galerkin, GLS e Método de Elementos Finitos Quase Estabilizado (QSFEM).

### 3.1 Diferenças Finitas Centradas

A análise terá início com o método de diferenças finitas centradas (MDFC), pois trata-se de um método de mais simples implementação e historicamente anterior aos outros que serão apresentados. Partindo da equação de estudo, em nosso caso a de Helmholtz, necessita-se apenas da noção de derivadas no sentido forte para a construção desse método.

A análise será subdividida, primeiramente, em uma dimensão e, depois, para duas dimensões, com o intuito de que o trabalho fique estruturado de forma harmônica e concisa.

### 3.1.1 Em uma dimensão

Seja uma função  $u : \Omega \subset \mathbb{R} \rightarrow \mathbb{R}$  pertencente a  $C^\infty$ , sendo  $\Omega$  seu domínio,  $x$  um ponto no interior do domínio e  $x + h \in \Omega$ , então esta função pode ser escrita como [21]:

$$u(x + h) = u(x) + u'(x)h + \frac{u''(x)h^2}{2!} + \dots + \frac{u^{(n-1)}(x)h^{n-1}}{(n-1)!} + O(h^n), \quad (3.1)$$

onde

$$O(h^n) = \frac{u^{(n)}(x + \theta_n h)h^n}{n!}, \quad \text{com } 0 < \theta_n < 1, \quad \forall n \in \mathbb{N}. \quad (3.2)$$

A equação (3.1) interpreta-se como um polinômio de ordem  $n - 1$  que interpola a função  $u$  em torno do ponto  $x$  e que se chama polinômio de Taylor. A partir destas equações pode-se aproximar a derivada segunda da função  $u$  usando

$$u(x + h) \approx u(x) + u'(x)h + \frac{u''(x)h^2}{2}$$

e

$$u(x - h) \approx u(x) - u'(x)h + \frac{u''(x)h^2}{2},$$

que somadas termo a termo, ainda reposicionando  $u''$  a esquerda, obtém-se

$$u''(x) \approx D_{xx}U_i = \frac{u(x - h) - 2u(x) + u(x + h)}{h^2}. \quad (3.3)$$

Representa-se o domínio de forma particionada, onde analisa-se a solução de forma pontual. Cada ponto desse domínio chama-se nó e a distância entre eles será  $h$ . A esse conjunto discreto de nós dá-se o nome de malha, representado esquematicamente na figura abaixo:

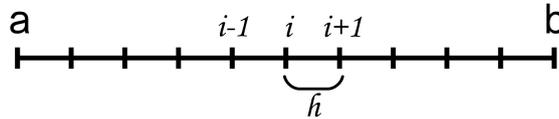


Figura 3.1: Domínio discreto

Com base na figura (3.1), será usada a notação  $u(x) = U_i$  para a função avaliada no nó  $i$ ,  $u(x - h) = U_{i-1}$  em  $i - 1$  e  $u(x + h) = U_{i+1}$  em  $i + 1$ . Com essa maneira de representar o domínio de forma discreta, escrevemos a derivada segunda como:

$$D_{xx}U_i = \frac{U_{i-1} - 2U_i + U_{i+1}}{h^2}$$

Assim, uma aproximação por diferenças finitas centradas de segunda ordem para o problema de Helmholtz unidimensional é definida como

$$D_{xx}U_i + k^2U_i = 0,$$

que de forma explícita é escrita como

$$R^{DF}U_{i-1} + 2S^{DF}U_i + R^{DF}U_{i+1} = 0 \quad (3.4)$$

com

$$R^{DF} = 1 \quad (3.5)$$

$$S^{DF} = \frac{(kh)^2}{2} - 1 \quad (3.6)$$

### 3.1.2 Em duas dimensões

Nossa análise para o MDFC em duas dimensões é análoga à unidimensional tratada na sessão (3.1.1), define-se a série de Taylor para a função nessas condições, será usada uma notação adequada à malha e estes conceitos serão aplicados ao problema de Helmholtz.

Seja uma função  $u : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$  pertencente à  $C^\infty$ , sendo  $x, y$  pontos interiores ao domínio  $\Omega$  e  $x + h_x, y + h_y \in \Omega$ . Esta função  $u$  pode ser escrita como [22]:

$$u(x + h_x, y) = u(x, y) + u_x(x, y)h_x + u_{xx}(x, y)\frac{h_x^2}{2} + O(h_x^3) \quad (3.7)$$

$$u(x - h_x, y) = u(x, y) - u_x(x, y)h_x + u_{xx}(x, y)\frac{h_x^2}{2} + O(h_x^3) \quad (3.8)$$

$$u(x, y + h_y) = u(x, y) + u_y(x, y)h_y + u_{yy}(x, y)\frac{h_y^2}{2} + O(h_y^3) \quad (3.9)$$

$$u(x, y - h_y) = u(x, y) - u_y(x, y)h_y + u_{yy}(x, y)\frac{h_y^2}{2} + O(h_y^3) \quad (3.10)$$

Necessita-se de expressões para  $u_{xx}$  e  $u_{yy}$ , tendo como ponto de partida as equações (3.7), (3.8), (3.9) e (3.10), assim como se fez para uma dimensão. Portanto, após certas manipulações algébricas, chega-se nas expressões que aproximam as derivadas de segunda ordem como:

$$u_{xx} \approx \frac{u(x + h_x, y) - 2u(x, y) + u(x - h_x, y)}{h_x^2}$$

$$u_{yy} \approx \frac{u(x, y + h_y) - 2u(x, y) + u(x, y - h_y)}{h_y^2}$$

O domínio discreto será referenciado junto com dois índices: um  $i$  que compete ao

eixo horizontal e um  $j$  para o vertical. A solução  $u$  será analisada sobre esses nós segundo a posição dele em  $i$  e em  $j$ , sendo da forma  $U_{i,j}$ .

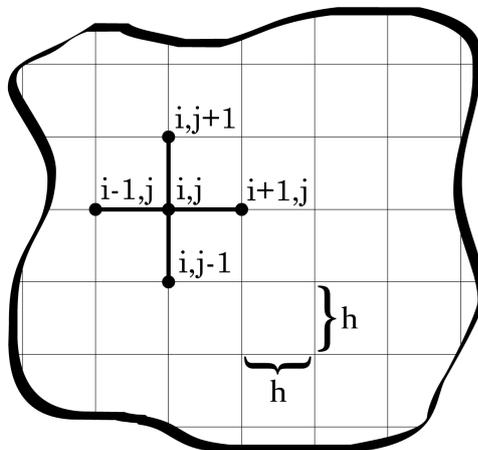


Figura 3.2: Domínio discreto com  $h_x = h_y = h$

Utiliza-se de uma notação baseada no domínio discreto conforme a figura (3.2). Sendo assim, tem-se  $u(x, y) = U_{i,j}$ ,  $u(x + h, y) = U_{i+1,j}$ ,  $u(x - h, y) = U_{i-1,j}$ ,  $u(x, y + h) = U_{i,j+1}$  e  $u(x, y - h) = U_{i,j-1}$ . A figura (3.2) representa uma malha uniforme, quadrangular, com estêncil de 5 pontos. Desta forma, pode-se escrever a equação de Helmholtz como

$$\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{h^2} + k^2 U_{i,j} = 0$$

e que multiplicando por  $h^2$  o lado esquerdo da igualdade tem-se ainda que

$$U_{i+1,j} - 2U_{i,j} + U_{i-1,j} + U_{i,j+1} - 2U_{i,j} + U_{i,j-1} + h^2 k^2 U_{i,j} = 0$$

sendo escrito de forma compacta como

$$A_1 U_{i-1,j} + A_1 U_{i,j-1} + A_2 U_{i,j} + A_1 U_{i,j+1} + A_1 U_{i+1,j} = 0 \quad (3.11)$$

com

$$A_1 = 1 \quad \text{e} \quad A_2 = -4 + h^2 k^2 \quad (3.12)$$

A forma como é escrito o método de diferenças finitas em (3.11) será de grande auxílio quando a análise de dispersão do problema para este método for explorada, procurando estimar o efeito de poluição.

## 3.2 Elementos Finitos de Galerkin

Alguns aspectos do método de elementos finitos de Galerkin são gerais, isto é, independem da dimensão e também do espaço de funções. Assim, esse primeiro olhar sobre o problema de Helmholtz, com base nesse método, será chamado de formulação geral. Após este preâmbulo, serão definidos os espaços de funções que variam segundo a dimensão do problema.

### 3.2.1 Formulação geral

Uma definição de extrema importância que será utilizada para enunciar o método de Elementos Finitos de Galerkin é o de espaço de dimensão finita [11].

**Definição 3.1.** Um espaço de funções  $V$  tem dimensão finita, se existe  $n \in \mathbb{N}$ , e um conjunto de funções  $\phi_i : \Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}$ ,  $i \in \{1, \dots, n\}$ , tal que qualquer função  $v \in V$  pode ser escrita como uma combinação linear das funções  $\phi_i$ . Isto é, existem  $n$  escalares  $\alpha_i$  tal que  $v = \sum_{i=1}^n \alpha_i \phi_i$ .

No capítulo 2, foi enunciado o problema de Helmholtz, em sua forma fraca, para um espaço de dimensão qualquer, desde que seja “Sobolev” (ver apêndice A). Entretanto, o método de elementos finitos, na formulação de Galerkin, aproximará a solução dentro de um espaço de dimensão finita, seguindo a definição 3.1. Desta forma, define-se o espaço de dimensão finita  $V^h$  que é usado para aproximar a solução exata  $u$ , sendo a solução aproximada  $u^h \in V^h$ . O problema com condições de contorno de Dirichlet, por exemplo, é o de encontrar  $u^h \in V^h$ , tal que

$$\int_{\Omega} (\nabla u^h \cdot \nabla v^h - k^2 u^h v^h) d\Omega = \int_{\Omega} f v^h d\Omega. \quad \forall v^h \in V^h. \quad (3.13)$$

Como o espaço onde é procurada a solução aproximada é de dimensão finita, é portanto gerado uma base  $\{\phi_i\}_i^{N^h}$ . Pode-se representar a solução aproximada  $u^h$  como combinação linear das funções base, isto é

$$u^h(x) = \sum_{i=1}^{N^h} \alpha_i \phi_i(x), \quad x \in \mathbb{R}^d \quad (3.14)$$

com os coeficientes  $\alpha_i$  a serem determinados.

Destaca-se que a solução aproximada agora pertence a um espaço de dimensão finita,

onde  $u^h$  é escrita segundo (3.14) e que  $v^h \in V^h$ . Substitui-se (3.14) em (3.13) e obtém-se

$$\int_{\Omega} [\nabla(\sum_{i=1}^{N^h} \alpha_i \phi_i(x)) \cdot \nabla \phi_j(x) - k^2 \sum_{i=1}^{N^h} \alpha_i \phi_i(x) \phi_j(x)] d\Omega = \int_{\Omega} f \phi_j(x) d\Omega \quad \forall j \in \{1, \dots, N^h\}$$

ou ainda

$$\sum_{i=1}^{N^h} \alpha_i \int_{\Omega} \nabla \phi_i(x) \cdot \nabla \phi_j(x) - k^2 \phi_i(x) \phi_j(x) d\Omega = \int_{\Omega} f \phi_j(x) d\Omega \quad \forall j \in \{1, \dots, N^h\}.$$

Define-se a matriz  $A = (a_{ij})$ , também conhecida com o nome de matriz de rigidez, como

$$a_{ij} = \int_{\Omega} \nabla \phi_i(x) \cdot \nabla \phi_j(x) d\Omega - k^2 \int_{\Omega} \phi_i(x) \cdot \phi_j(x) d\Omega \quad (3.15)$$

e os vetores  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_{N^h})^t$  e  $b = (b_j)$ , tal que

$$b_j = \int_{\Omega} f \phi_j(x) d\Omega.$$

Assim, o problema original em um espaço de dimensão infinita, quando se faz a suposição de que ele é solúvel em espaços de dimensão finita, tem a forma de um sistema linear de equações algébricas

$$A\alpha = b \quad (3.16)$$

### 3.2.2 Espaço de funções lineares por partes - caso unidimensional

A partir de agora, o trabalho será o de caracterizar as funções  $\phi$  no espaço de funções definido. Sendo  $\Omega \subset \mathbb{R}$  um aberto, particiona-se esse domínio em elementos finitos  $\Omega_e$  não degenerados, isto é, não se reduzem a um ponto. A união desses elementos gera todo o  $\Omega$  e a interseção desses elementos é vazia. Dessa forma, o domínio de cada elemento será o intervalo aberto  $\Omega_e = (x_1^e, x_2^e)$  e  $\Omega = \bigcup_{i=1}^{Nel} (c_{i-1}, c_i)$ , onde  $Nel$  é o número de elementos e  $c_i$  um vértice de um elemento.

Considerando essa partição do domínio, define-se o espaço das funções contínuas lineares por partes como

$$V^h = \{g \in C(\Omega) : g|_{(c_{i-1}, c_i)} \text{ é linear}\}, \quad (3.17)$$

isto é,  $g|_{(c_{i-1}, c_i)}(x) = \gamma_i x + \beta_i$ , sendo  $\gamma_i, \beta_i \in \mathbb{R}$  constantes apropriadas. Sabe-se ainda que

o conjunto das funções  $\phi_i$  forma uma base para o espaço  $V^h$  e define-se como

$$\phi_i(c_j) = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases} \quad (3.18)$$

e vê-se também que qualquer função  $v^h \in V^h$  pode ser escrita como

$$v^h(x) = \sum_{i=0}^N z_i \phi_i(x). \quad (3.19)$$

A forma como é definida a função  $\phi_i$  parece ser aleatória, entretanto, não é. Ela completa o sentido pelo qual foi definido o sistema em (3.16), pois o vetor  $\alpha$  deve conter exatamente a solução sobre o nó, ou vértice dos subintervalos. Portanto, se  $u(z_i)$  contém a solução sobre o vértice, a solução no vetor  $\alpha$  deve ser a mesma para que o sistema tenha sentido. E, claramente, observa-se que

$$\alpha_i = u(z_i) \quad (3.20)$$

pois a função  $\phi_i$  é não nula exatamente em cima do nó. Com isso, a forma como se define  $\phi_i$  é compatível com o espaço de funções e como encontra-se a solução no sistema linear definido em (3.16).

Com base nas formulações e definições anteriores, principalmente (3.17) e (3.18), faz-se uso de algumas funções de base bem específicas. Usa-se a notação para as funções base restritas a um elemento. No caso unidimensional serão duas funções para cada elemento, como ilustra a figura a baixo:

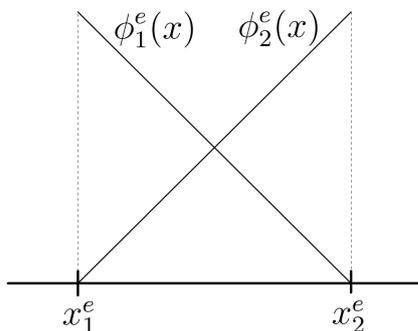


Figura 3.3: Funções base para um elemento.

Desta forma, as expressões que caracterizam tais funções são dadas por

$$\phi_1^e(x) = \frac{x_2^e - x}{x_2^e - x_1^e}, \quad (3.21)$$

$$\phi_2^e(x) = \frac{x - x_1^e}{x_2^e - x_1^e}, \quad (3.22)$$

sendo  $\phi_1^e$  e  $\phi_2^e$  as funções restritas a apenas um elemento,  $x_1^e$  e  $x_2^e$ , seu domínio, com  $x_1^e \leq x \leq x_2^e$ . A partir destas funções, pode-se construir a matriz do sistema, que é composta por matrizes para cada elemento, chamadas de matrizes locais.

A matriz local  $K^{loc}$  no caso unidimensional, com malha uniforme, é dada por uma mesma expressão para qualquer elemento do domínio. Por questões de ter-se uma malha uniforme, com parâmetro  $h$ , e de simetrias,  $K_{11}^{loc} = K_{22}^{loc}$  e  $K_{12}^{loc} = K_{21}^{loc}$ . Assim, as matrizes locais assumem a seguinte forma:

$$K^{loc} = \begin{bmatrix} \frac{1}{h} - \frac{k^2 h}{3} & \frac{-1}{h} - \frac{k^2 h}{6} \\ \frac{-1}{h} - \frac{k^2 h}{6} & \frac{1}{h} - \frac{k^2 h}{3} \end{bmatrix}. \quad (3.23)$$

Resta montar a matriz global, pois o que foi feito até o momento são matrizes locais, restritas a um único elemento do domínio. Repara-se a forma como foram definidas as funções de base em (3.17) e (3.18), ela contribui para dois elementos, sempre um a esquerda e outro a direita de cada nó. No entanto, quando as matrizes locais são calculadas, usa-se o esquema da figura (3.3) que dá a metade da contribuição da função base. Tendo em vista essa decomposição do domínio, a passagem das matrizes locais para a global é chamada *assembly* ou montagem. Lembrando ainda que o *assembly* respeita a forma como são escritas as soluções em (3.19) e (3.20).

No caso unidimensional essa superposição acontece nos elementos  $K_{11}^{loc}$  e  $K_{22}^{loc}$ , com exceção dos elementos que tenham nós no contorno. Portanto, a matriz de global assume a forma

$$K = \begin{bmatrix} S^{Gal} & R^{Gal} & 0 & \dots & \dots & \dots & \dots & 0 \\ R^{Gal} & 2S^{Gal} & R^{Gal} & \ddots & & & & \vdots \\ 0 & R^{Gal} & 2S^{Gal} & R^{Gal} & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & R^{Gal} & 2S^{Gal} & R^{Gal} & 0 \\ \vdots & & & & \ddots & R^{Gal} & 2S^{Gal} & R^{Gal} \\ 0 & \dots & \dots & \dots & \dots & 0 & R^{Gal} & S^{Gal} \end{bmatrix}, \quad (3.24)$$

e tendo que  $R^{Gal} = K_{12}^{loc}$  e  $S^{Gal} = K_{11}^{loc}$  de (3.24) tem-se ainda que

$$R^{Gal} = -1 - \frac{(kh)^2}{6} \quad \text{e} \quad S^{Gal} = 1 - \frac{(kh)^2}{3}. \quad (3.25)$$

Pode-se fazer uma interpretação da matriz  $K$  (3.24) por colunas ou linhas. Analisando por colunas vê-se exatamente a solução para cada nó como a combinação linear das funções base, como em (3.19) e (3.20). A segunda análise, por linhas, corrobora a representação de um nó e a influência de seus vizinhos laterais mais próximos. Este último fato é evidenciado quando efetua-se a multiplicação de uma das linhas da matriz  $K$  pelo vetor solução.

Finalmente, excluindo-se a primeira e última filas da matriz  $K$  por se tratar de contorno, pode-se escrever o método, para o caso homogêneo, ainda pela forma

$$R^{Gal}U_{i-1} + 2S^{Gal}U_i + R^{Gal}U_{i+1} = 0. \quad (3.26)$$

Nota-se ainda que os valores de  $R^{Gal}$  e  $S^{Gal}$  diferem um pouco de (3.23), mas é justamente porque (3.26) é homogênea, bastando dividir (3.26) por  $h$  para que seja igual às entradas em (3.23). Faz-se essa manipulação algébrica para mudar a aparência da matriz, a solução não é alterada.

### 3.2.3 Espaço de funções lineares por partes - caso bidimensional

Para o caso bidimensional, faz-se o mesmo trajeto de definições que foi feito em uma dimensão. Começa-se por particionar o domínio, visando os elementos retangulares que o compõe, fazendo-se uma numeração por elementos. Logo em seguida, caracteriza-se as funções base, sendo agora bilineares. Constrói-se então as matrizes locais e conclui-se com a matriz global.

Considere  $M^h = \{\Omega_1, \dots, \Omega_{Nel}\}$  uma partição, em elementos finitos  $\Omega_e$ , do domínio não degenerado e aberto  $\Omega \subset \mathbb{R}^2$ . Cada par de elementos satisfazem  $\Omega_e \cap \Omega_{e'} = \emptyset$ . Também, tem-se que  $\Omega \cup \Gamma = \bigcup_{e=1}^{Nel} (\Omega_e \cup \Gamma_e)$  onde  $\Gamma$  é a fronteira de  $\Omega$  e  $\Gamma_e$  a fronteira de  $\Omega_e$ .

Em um elemento retangular, embora tenha 4 pontos ou arestas, precisa-se de dois valores de  $x$ ,  $\{x_1^e, x_2^e\}$ , e dois de  $y$ ,  $\{y_1^e, y_2^e\}$ , para que suas quatro arestas sejam mapeadas. A figura (3.4) mostra essa partição do domínio  $\Omega$  e que, por exemplo, para o elemento  $\Omega_7$ , precisa-se de  $(x_1^7, y_1^7)$ ,  $(x_2^7, y_2^7)$ ,  $(x_1^7, y_2^7)$  e  $(x_2^7, y_1^7)$ .

As funções bilineares são definidas com restrição a um dado elemento  $\Omega_e$  do domínio

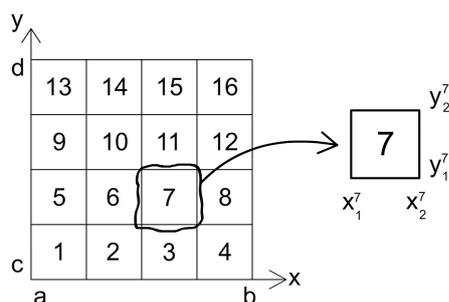


Figura 3.4: Numeração dos elementos e suas coordenadas

$\Omega$  que são escritas como

$$v^h|_{\Omega_e}(x, y) = axy + by + cx + d \quad (3.27)$$

onde  $a, b, c, d$  são constantes apropriadas dependentes de cada um dos elementos, sendo os graus de liberdade da função. Ainda impõe-se a mesma condição do caso unidimensional

$$\phi_i(c_j) = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases} \quad (3.28)$$

onde  $i$  e  $j$  são índices para os  $N$  vértices, ou nós, da partição, conforme figura (3.5).

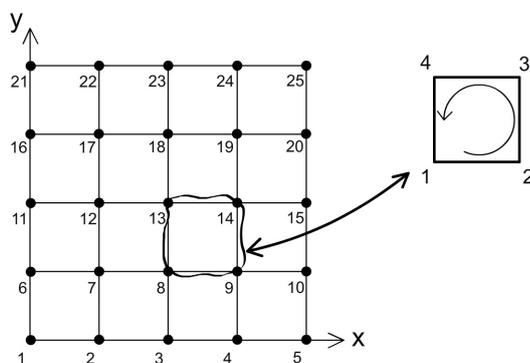


Figura 3.5: Numeração Global e Local dos nós

Nota-se que a figura (3.5) apresenta duas numerações. A primeira é global, com índices  $i$  e  $j$  variando dentro do conjunto  $\{1, 2, 3, \dots, N\}$ , permitindo definir as funções  $\phi_i$  segundo (3.28). A segunda numeração é local, com índices variando dentro do conjunto  $\{1, 2, 3, 4\}$  que é favorável à definição das funções  $\phi_i$  restritas a um elemento apenas, assim como se fez no caso unidimensional em (3.21) e (3.22).

Utilizando-se a numeração local proposta esquematicamente na figura (3.5) define-se

as funções base para um elemento específico, em uma malha uniforme conforme figura (3.5). Valendo-se de (3.27) e (3.28) tem-se que

$$\phi_1^e(x, y) = \frac{(x - x_2^e)(y - y_2^e)}{(x_1^e - x_2^e)(y_1^e - y_2^e)}, \quad (3.29)$$

$$\phi_2^e(x, y) = \frac{(x - x_1^e)(y - y_2^e)}{(x_2^e - x_1^e)(y_1^e - y_2^e)}, \quad (3.30)$$

$$\phi_3^e(x, y) = \frac{(x - x_1^e)(y - y_1^e)}{(x_2^e - x_1^e)(y_2^e - y_1^e)}, \quad (3.31)$$

$$\phi_4^e(x, y) = \frac{(x - x_2^e)(y - y_1^e)}{(x_1^e - x_2^e)(y_2^e - y_1^e)}. \quad (3.32)$$

Utilizando-se as funções de base descritas de (3.29) até (3.32), a forma integral (3.15) e reparando-se certas simetrias, a matriz local para cada elemento é dada por

$$K^{loc} = \begin{bmatrix} A_1 & A_2 & A_3 & A_2 \\ A_2 & A_1 & A_2 & A_3 \\ A_3 & A_2 & A_1 & A_2 \\ A_2 & A_3 & A_2 & A_1 \end{bmatrix}. \quad (3.33)$$

E sendo  $x_2^e - x_1^e = y_2^e - y_1^e = h$ , as entradas da matriz em (3.33) são dadas por

$$A_1 = \frac{2}{3h} - \frac{k^2 h}{9} \quad (3.34)$$

$$A_2 = \frac{-1}{6h} - \frac{k^2 h}{18} \quad (3.35)$$

$$A_3 = \frac{-1}{3h} - \frac{k^2 h}{36}. \quad (3.36)$$

Não foi efetuado nada mais que a integração com as funções base, sem nenhuma manipulação algébrica, coisa que será feita posteriormente para modificar a aparência das matrizes.

Ainda resta o *assembly*, ou montagem, da matriz global. O domínio foi decomposto em elementos quadrangulares e as matrizes locais são calculadas para cada um deles, até mesmo conforme a figura (3.4). Por outro lado, as funções base não são definidas para um elemento, e sim em função do nó. Portanto, quando se integra e computa as entradas da matriz local, é levada em consideração apenas uma parte da função base, restrita a um elemento. O *assembly* faz essa realocação das contribuições para cada nó, dentro da matriz global.

Escrevendo-se numa forma semelhante a (3.11) tem-se:

$$\begin{aligned}
& A_3^{Gal} U_{i-1,j+1} + A_2^{Gal} U_{i,j+1} + A_3^{Gal} U_{i+1,j+1} \\
& + A_2^{Gal} U_{i-1,j} + A_1^{Gal} U_{i,j} + A_2^{Gal} U_{i+1,j} \\
& + A_3^{Gal} U_{i-1,j-1} + A_2^{Gal} U_{i,j-1} + A_3^{Gal} U_{i+1,j-1} = 0
\end{aligned} \tag{3.37}$$

onde os coeficientes que acompanham a solução discreta valem

$$A_1^{Gal} = \frac{8}{3} - \frac{4k^2 h^2}{9} \tag{3.38}$$

$$A_2^{Gal} = \frac{-1}{3} - \frac{k^2 h^2}{9} \tag{3.39}$$

$$A_3^{Gal} = \frac{-1}{3} - \frac{k^2 h^2}{36}. \tag{3.40}$$

A equação (3.37) pode ser dividida por  $h$  em ambos os membros para que estes coeficiente sejam idênticos às entradas em (3.33). Foi feito esse mesmo procedimento no caso unidimensional também, alterando-se apenas a aparência da matriz global.

### 3.3 Análise de Dispersão

É preciso verificar agora qual é a relação existente entre a discretização dos métodos, seja por diferenças finitas, ou por elementos finitos, e a própria solução exata, já que é conhecida para certas condições de contorno. Para tanto, será utilizado o que é chamado de estêncil de (3.4), (3.11), (3.25) e (3.37).

#### 3.3.1 Caso unidimensional

Considere o problema unidimensional segundo

$$u'' + k^2 u = 0 \quad \text{em } (0, 1) \tag{3.41}$$

e faz-se a suposição de que o problema tem solução única para uma dada condição de contorno e um valor de  $k$ . Sabe-se que a solução exata para o problema é dada pela equação (2.12). É intuitivo procurar uma solução nodal aproximada com o mesmo formato da exata, com a forma

$$u^h(x_j) = e^{ik^h x_j} \tag{3.42}$$

sendo  $u^h(x_j)$  a solução aproximada em um nó  $j$ ,  $k^h$  o número de onda discreto e valendo-se da discretização uniforme  $x_j = (j-1)h$  com  $j = 1, 2, \dots, n$ . Segue-se, então, a idealização discreta de um esquema de diferenças finitas como

$$Ru(x_j - h) + 2Su(x_j) + Ru(x_j + h) = 0. \quad (3.43)$$

Substituindo (3.42) em (3.43), considerando  $u^h = u$ , tem-se a seguinte equação

$$Re^{ik^h(x_j-h)} + 2Se^{ik^hx_j} + Re^{ik^h(x_j+h)} = 0. \quad (3.44)$$

Multiplicando-se ambos os termos da equação anterior por  $e^{-ik^hx_j}$  tem-se

$$Re^{-ik^hh} + 2S + Re^{ik^hx_j} = 0, \quad (3.45)$$

e também por  $e^{ik^hh}$ , tem-se a equação de segundo grau

$$R\lambda^2 + 2S\lambda + R = 0 \quad (3.46)$$

onde  $\lambda = e^{ik^hh}$ . E, portanto, tem-se as soluções

$$\lambda = -\frac{S}{R} \pm \sqrt{\frac{S^2}{R^2} - 1} \quad (3.47)$$

Analisando (3.47) tem-se uma solução puramente real se  $|S/R| \geq 1$ , do contrário a solução será complexa com parte imaginária não nula. Voltando à equação (3.45) e usando a fórmula de Euler da mesma forma que (2.22) transformando-se em (2.23), observa-se que

$$2R\cos(k^hh) + 2S = 0. \quad (3.48)$$

Portanto, o número de onda aproximado, ou discreto, é dado por

$$k^h = \frac{1}{h} \arccos\left(-\frac{S}{R}\right). \quad (3.49)$$

Para o método de Galerkin, considera-se  $S$  e  $R$  conforme (3.5) e (3.6), e para o método de diferenças finitas, com  $S$  e  $R$  como calculados em (3.26), a estimativa para o erro de fase, seguindo a referência [9], é dada por:

$$\frac{k - k^h}{k} = \frac{(kh)^2}{24} + O((kh)^4). \quad (3.50)$$

### 3.3.2 Caso bidimensional

O caso bidimensional, segue a mesma ideia de discretização e análise do unidimensional. Considere uma malha uniforme com elementos quadrados, de lado  $h$ . Por questões de simetria e invariância em relação à translação [9], a equação do estêncil será dada pela forma

$$\begin{aligned} & A_3u(x_j - h, y_j + h) + A_2u(x_j, y_j + h) + A_3u(x_j + h, y_j + h) \\ & \quad + A_2u(x_j - h, y_j) + A_1u(x_j, y_j) + A_2u(x_j + h, y_j) \\ & + A_3u(x_j - h, y_j - h) + A_2u(x_j, y_j - h) + A_3u(x_j + h, y_j - h) = 0 \end{aligned} \quad (3.51)$$

e procura-se soluções que sejam ondas planas escritas sobre a forma

$$u^h(x_j, y_j) = e^{ik^h(x_j \cos(\theta) + y_j \sin(\theta))} \quad (3.52)$$

sendo  $u^h$  a solução aproximada, avaliada nos nós  $(x_j, y_j)$ , com o número de onda discreto  $k^h$ . Substituindo-se (3.52) em (3.51), tem-se a seguinte relação de dispersão para o problema discreto em duas dimensões:

$$A_1 + A_2(\cos(k^h h \cos \theta) + \cos(k^h h \sin \theta)) + 4A_3 \cos(k^h h \cos \theta) \cos(k^h h \sin \theta) = 0. \quad (3.53)$$

O método de Galerkin, e também Diferenças Finitas, apresenta o erro relativo de fase da forma

$$\frac{k - k^h}{k} = (3 + \cos(4\theta)) \frac{(kh)^2}{96} + O((kh)^4). \quad (3.54)$$

## 3.4 Galerkin Mínimos Quadrados (GLS)

O método agora enunciado consiste em se adicionar certos termos à formulação clássica de Galerkin, esses como resíduos na forma de mínimos quadrados. Pode-se mencionar um primeiro artigo de Hughes, Franca e Hulbert [14] em que esse método foi trabalhado para equações de advecção-difusão. Posteriormente, Harari e Hughes [12] estenderam essa formulação para o problema exterior de Helmholtz, usando o processo de DtN (Dirichletto-Neumann), consistente com a condição de radiação de Sommerfeld. Embora este último trabalho seja intitulado para um problema exterior, o problema interior também é tratado. Uma análise para o problema de Helmholtz em duas dimensões foi feita por Thompson e Pinsky [28].

Supondo-se novamente uma forma bilinear e uma malha quadrangular uniforme, tem-se que encontrar  $u_h \in V^h$  na formulação variacional, adicionando-se novos termos, que assume a forma

$$a(u^h, v^h) + \tau a_{GLS}(u^h, v^h) = f(v^h) + \tau f_{GLS}(v^h), \quad \forall v^h \in V^h,$$

onde

$$a_{GLS}(u^h, v^h) = \sum_{i=1}^{N^h} \int_{\Omega_e} (-\Delta u^h - k^2 u^h)(-\Delta v^h - k^2 v^h) d\Omega$$

$$f_{GLS}(v^h) = \sum_{i=1}^{N^h} \int_{\Omega_e} (-\Delta v^h - k^2 v^h) f d\Omega.$$

Os termos  $a_{GLS}(\cdot, \cdot)$  e  $f_{GLS}(\cdot)$  são os resíduos que são mencionados anteriormente e que são adicionados à formulação clássica de Galerkin. A matriz local para o método GLS assumirá a forma

$$K^{loc} = K_{GAL}^{loc} + K_{GLS}^{loc},$$

que para uma dimensão  $K_{GAL}^{loc}$  é dada por (3.23) e  $K_{GLS}^{loc}$  é

$$K_{GLS}^{loc} = \begin{bmatrix} A_1 & A_2 \\ A_2 & A_1 \end{bmatrix} \quad (3.55)$$

com,

$$A_1 = \tau k^4 h(1/3) \quad \text{e} \quad A_2 = \tau k^4 h(1/6)$$

e para o problema em duas dimensões  $K_{GAL}^{loc}$  é dada por (3.33) e  $K_{GLS}^{loc}$  e expressa como

$$K_{GLS}^{loc} = \begin{bmatrix} A_1 & A_2 & A_3 & A_2 \\ A_2 & A_1 & A_2 & A_3 \\ A_3 & A_2 & A_1 & A_2 \\ A_2 & A_3 & A_2 & A_1 \end{bmatrix} \quad (3.56)$$

onde as entradas da matriz são dadas por

$$A_1 = \tau k^4 h(1/9), \quad A_2 = \tau k^4 h(1/18) \quad \text{e} \quad A_3 = \tau k^4 h(1/36).$$

Efetua-se apenas as integrações para construir as matrizes locais. As entradas da matriz local para o método de Galerkin podem ser escritas como

$$A_1 = -\frac{8}{3} + \alpha_{GAL}, \quad A_2 = \frac{1}{3} + \frac{\alpha_{GAL}}{4}, \quad A_3 = \frac{1}{3} + \alpha_{GAL} \quad (3.57)$$

onde considera-se  $\alpha_{GAL} := (kh)^2/9$ . Para obter-se as entradas da matriz para o método GLS, basta que se troque  $\alpha_{GAL}$  por  $\alpha_{GLS} = \alpha_{GAL}(1 - \tau k^2)$ , o que resulta uma forma mais compacta de representação do estêncil para o método GLS.

Entretanto, observa-se ainda um parâmetro de estabilidade  $\tau$  na forma bilinear do GLS. Em duas dimensões ele segue como [12]

$$\tau = \frac{1}{k^2} \left( 1 - 6 \frac{4 - \cos(s_1) - \cos(s_2) - 2\cos(s_1)\cos(s_2)}{(2 + \cos(s_1))(2 + \cos(s_2))k^2h^2} \right) \quad (3.58)$$

onde,

$$s_1 = kh\cos(\theta) \quad e \quad s_2 = kh\sin(\theta).$$

A direção  $\theta = \frac{\pi}{8}$  é normalmente escolhida para duas dimensões [12]. O parâmetro de estabilização para uma dimensão é intuitivamente dado pela escolha de  $\theta = 0$ , observando a natureza da solução para o problema. Assim, tem-se que

$$\tau = \frac{1}{k^2} \left( 1 - \frac{6(1 - \cos(kh))}{k^2h^2(2 + \cos(kh))} \right)$$

minimiza o erro da solução aproximada ao erro do interpolante, em quaisquer normas. Resultado este que será visto no capítulo posterior.

Uma dificuldade que é encontrada a primeira vista é a dependência do parâmetro  $\tau$  com  $kh$ , o que acarreta a impossibilidade de estabilização do método para uma malha não uniforme, pois não há uma escolha única de  $h$ , apenas uma que minimize o erro e que depende da malha e sua distorção em relação à uniforme.

Na tentativa de deixar o texto conciso e mais objetivo quanto aos métodos, omitimos a forma de determinação do parâmetro  $\tau$ . Contudo, ele é determinado na inserção dos parâmetros em (3.57) na relação de dispersão (3.53) para duas dimensões.

O erro relativo de fase para o método GLS [9], para um ângulo diferente do  $\theta$  ótimo, é da mesma ordem que o de Galerkin

$$\frac{k - k^h}{k} = \cos(4\theta) \frac{(kh)^2}{24} + O((kh)^4). \quad (3.59)$$

### 3.5 Método de Elementos Finitos Quase Estabilizado (QSFEM)

O método QSFEM foi proposto por Babuska [4] com o intuito de minimizar o efeito de poluição do erro para o problema de Helmholtz em duas dimensões. Contudo, este método

não é construído sobre uma formulação variacional, assim como o GLS, ele é definido de forma similar a um método de diferenças finitas. Portanto, supõe-se um domínio discreto com uma malha uniforme, formada por quadrados e que o estêncil dos nós interiores possuem a forma

$$\begin{aligned} &G_3u(x_j - h, y_j + h) + G_2u(x_j, y_j + h) + G_3u(x_j + h, y_j + h) \\ &\quad + G_2u(x_j - h, y_j) + G_1u(x_j, y_j) + G_2u(x_j + h, y_j) \\ &+ G_3u(x_j - h, y_j - h) + G_2u(x_j, y_j - h) + G_3u(x_j + h, y_j - h) = 0 \end{aligned} \quad (3.60)$$

onde  $G_1$ ,  $G_2$  e  $G_3$  dependem de  $kh$  e  $u^h \in S_h(\Omega)$  a solução aproximada dada por

$$u^h(x) = \sum u^h(x_j)\phi_i(x). \quad (3.61)$$

Esse método agora utiliza um ferramental da teoria (integral) da transformada de Fourier para verificar a qualidade da solução na forma discreta. Essa parte da teoria é melhor detalhada em [5]. O símbolo do operador de Helmholtz é dado por

$$\sigma(\xi) = \|\xi\|^2 - k^2 \quad (3.62)$$

onde  $\xi \in \mathbb{R}$  e  $\|\xi\|^2 := \xi_1^2 + \xi_2^2$ , com  $\xi_1 = kh\cos\theta$  e  $\xi_2 = kh\sin\theta$ . Assume-se agora que o estêncil dos pontos interiores do domínio para o QSFEM, segundo (3.60), é dado na forma matricial

$$G = \begin{bmatrix} G_3 & G_2 & G_3 \\ G_2 & G_1 & G_2 \\ G_3 & G_2 & G_3 \end{bmatrix} \quad (3.63)$$

onde  $G_1, G_2$  e  $G_3$  dependem de  $k$  e  $h$ . O símbolo do operador diferença correspondente para o estêncil é

$$\sigma_{estncil}(\xi) := G_1 + 2G_2(\cos(\xi_1) + \cos(\xi_2)) + 4G_3\cos(\xi_1)\cos(\xi_2) = 0. \quad (3.64)$$

A ideia é de minimizar a distância entre o círculo descrito em (3.62) e a curva projetada em (3.64). O estêncil obtido por meio dos últimos cálculos faz com que essas duas curvas interceptem-se em 16 pontos

$$\theta = \frac{(2n-1)\pi}{16} \quad n = 1, \dots, 16. \quad (3.65)$$

Uma forma de se obter  $G_1$ ,  $G_2$  e  $G_3$  é predefinindo  $G_1 = 4$  e resolvendo um sistema

de duas varáveis,  $G_2$  e  $G_3$ , e duas equações

$$G_1 + 2G_2(\cos(hR_1) + \cos(hR_2)) + 4G_3\cos(hR_1)\cos(hR_2) = 0 \quad (3.66)$$

$$G_1 + 2G_2(\cos(hS_1) + \cos(hS_2)) + 4G_3\cos(hS_1)\cos(hS_2) = 0 \quad (3.67)$$

onde

$$R_1 = k\cos\frac{\pi}{16}, \quad (3.68)$$

$$R_2 = k\sen\frac{\pi}{16}, \quad (3.69)$$

$$S_1 = k\cos\frac{3\pi}{16}, \quad (3.70)$$

$$S_2 = k\sen\frac{3\pi}{16}. \quad (3.71)$$

$$(3.72)$$

Assumindo-se  $\alpha := kh$  e definindo os coeficientes do estêncil para pontos interiores do domínio do método QSFEM como

$$G_1 = 4 \quad (3.73)$$

$$G_2 = 2 \frac{c_1(\alpha)s_1(\alpha) - c_2(\alpha)s_2(\alpha)}{c_2(\alpha)s_2(\alpha)(c_1(\alpha) + s_1(\alpha)) - c_1(\alpha)s_1(\alpha)(c_2(\alpha) + s_2(\alpha))}, \quad (3.74)$$

$$G_3 = \frac{c_2(\alpha) + s_2(\alpha) - c_1(\alpha) - s_1(\alpha)}{c_2(\alpha)s_2(\alpha)(c_1(\alpha) + s_1(\alpha)) - c_1(\alpha)s_1(\alpha)(c_2(\alpha) + s_2(\alpha))}, \quad (3.75)$$

e assumindo funções auxiliares  $c_1, c_2, s_1$  e  $s_2$  que são definidas por

$$c_1(\alpha) = \cos\left(\alpha\cos\frac{\pi}{16}\right), \quad (3.76)$$

$$s_1(\alpha) = \cos\left(\alpha\sen\frac{\pi}{16}\right), \quad (3.77)$$

$$c_2(\alpha) = \cos\left(\alpha\cos\frac{3\pi}{16}\right), \quad (3.78)$$

$$s_2(\alpha) = \cos\left(\alpha\sen\frac{3\pi}{16}\right), \quad (3.79)$$

O erro relativo de fase para este método é da ordem de

$$\frac{k - k^h}{k} = -\frac{\cos 8\theta}{774144}(kh)^6 + O((kh)^8).$$

## 3.6 Ressonância

Um fenômeno analítico, e também numérico, que será explorado é o que chamamos de ressonância, seguindo a referência [9]. Para tanto, considera-se o problema unidimensional

$$u'' + k^2 u = 0 \quad \text{em } (0, 1) \quad (3.80)$$

$$u(0) = a, \quad u(1) = b \quad (3.81)$$

O problema (3.80) com condições de contorno de Dirichlet (3.55) tem a solução

$$u(x) = \frac{a.\text{sen}(k - kx) + b.\text{sen}(kx)}{\text{sen}(k)}. \quad (3.82)$$

Uma primeira análise, com rápida inspeção dos termos da solução (3.82), já aponta um  $\text{sen}(k)$  no denominador. Portanto, é trivial se pensar que a função possa ser nula para algum valor de  $k$ . A função  $\text{sen}(k)$  é nula para  $k = n\pi$ , para qualquer valor de  $n \in \mathbb{N}$ . Pode-se verificar também pelos autovalores do operador  $-\Delta$  em uma dimensão

$$\lambda_n = n^2 \pi^2. \quad (3.83)$$

Quando  $k^2 = \lambda_n$  surge ressonância.

Os métodos apresentados até agora permitem formular uma solução sobre a forma

$$\begin{aligned} A_1 u^h(x_{j-1}) + A_2 u^h(x_j) + A_1 u^h(x_{j+1}) + \\ k^2 (B_1 u^h(x_{j-1}) + B_2 u^h(x_j) + B_1 u^h(x_{j+1})) = 0. \end{aligned} \quad (3.84)$$

A solução desse problema sobre a forma discreta é dada por

$$u^h(x_j) = \frac{a.\text{sen}(k^h - k^h x_j) + b.\text{sen}(k^h x_j)}{\text{sen}(k^h)}. \quad (3.85)$$

onde  $k^h$  é o número de onda discreto:

$$k^h = \frac{1}{h} \arccos \left( -\frac{A_2 + k^2 B_2}{2A_1 + 2k^2 B_1} \right). \quad (3.86)$$

É possível visualizar a ressonância tanto no problema contínuo, quanto no discreto. Contudo, sabe-se que no método de elementos finitos de Galerkin e Diferenças Finitas o número de onda discreto difere-se do analítico, evidenciado pela equação (3.86). Portanto, mesmo que a solução exata esteja em ressonância a aproximada não estará, e vice-versa.

Da mesma forma, a ressonância acontecerá em

$$k^h = n\pi, \quad \text{para } n \in \mathbb{N} \quad (3.87)$$

e substituindo (3.87) em (3.86) tem-se os valores de  $k$  para os quais o problema discreto entra em ressonância:

$$k^2 = \frac{-A_2 - 2A_1 \cos(hn\pi)}{B_2 + 2B_1 \cos(hn\pi)} \quad (3.88)$$

É possível fazer a mesma análise para o problema bidimensional considerando um domínio  $(0, a) \times (0, b)$ , sendo que os autovalores do operador  $-\Delta$  são

$$\lambda_{mn} = \pi^2 \left[ \frac{m^2}{a} + \frac{n^2}{b} \right]. \quad (3.89)$$

Observa-se que o fenômeno de ressonância para o caso bidimensional é severo para efeito de erro, pois tem-se dois parâmetros que causam ressonância, ou seja, maior probabilidade de entrar em ressonância ou estar próximo dela.

Necessita-se, em primeira instância, de métodos que façam  $k^h = k$ , com a finalidade de se evitar a degradação da solução numérica com o agravante da ressonância numérica. O método GLS para uma dimensão é capaz de eliminar, tornando  $k^h = k$ . Para duas dimensões o GLS não minimiza esse efeito. O método QSFEM é apresentado para contornar e minimizar tais efeitos para o problema 2D, oriundos primariamente do efeito de poluição, diferença  $k^h - k$ .

# Capítulo 4

## Resultados Numéricos

Neste capítulo são apresentados os resultados para os métodos formulados através dos capítulos anteriores e os resultados são divididos em duas partes para melhor clareza. A primeira consiste de uma análise do problema em uma dimensão com três métodos: Galerkin, MDFC e GLS. A segunda parte consiste de uma análise bidimensional do problema, utilizando quatro métodos: Galerkin, MDFC, GLS e QSFEM.

### 4.1 Implementação Computacional

Todos os códigos ao longo desse trabalho, os quais encontram-se no apêndice B, foram implementados em Matlab versão 2010 64 bits [1], em ambiente Windows 7, com notebook Samsung de processador intel i3 e 4GB RAM.

Os códigos possuem parâmetros de entrada, tais como: número de onda, número de nós ou elementos, definições de malha (domínio e partição) e valores de contorno. Quanto a esse aspecto inicial eles não se diferem muito entre si. Há diferença do caso 2D para o 1D porque no primeiro acrescenta-se a direção da onda como parâmetro.

No método de diferenças finitas, tanto no 1D, quanto no 2D, a implementação foi similar. Dá-se os parâmetros de entrada, constrói-se a matriz do sistema, aplicam-se as condições de contorno, no caso Dirichlet, para os nós devidos e resolve-se o sistema.

Já no método de elementos finitos de Galerkin, GLS e QSFEM faz-se uma implementação com outras estruturas de dados, distintas do método de diferenças finitas. Neste método, usa-se integração numérica para a construção das matrizes locais, no caso, quadratura de Gauss com 3 pontos [18]. Usa-se no código um vetor de estruturas [24] para armazenar todas as informações de cada elemento, tais como: posição final e inicial, em

$x$  e  $y$ , valores das funções base e outros. Usa-se ao longo do código a função *sum* que é uma forma vetorizada do "loop" de "for".

Em ambos os códigos utiliza-se a estrutura de dados *sparse*, que armazena os vetores de maneira ótima com relação a memória do computador, pois todos os vetores são do tipo esparso, com uma quantidade alta de zeros. Pode-se inicializar os vetores com a função *zeros*, onde todos os zeros são armazenados na memória do computador. Uma desvantagem na estrutura *sparse* é sua inicialização, pois a função *zeros* é um pouco mais rápida pois não tem que efetuar a otimização dos elementos na memória. Contudo, torna-se inevitável não se usar a estrutura *sparse* e essa lentidão torna-se ínfima conforme o número de elementos cresce.

Uma particularidade do Matlab é que quando inicializamos uma estrutura do tipo *sparse* ele já otimiza a resolução do sistema linear algébrico. A figura (4.1) mostra que até 600 pontos na malha, para o método de Galerkin no caso unidimensional, a inicialização por *zeros* é vantajosa computacionalmente, mas depois desse valor a *sparse* é mais eficiente. Outro fator a ser levado em conta é que mesmo fazendo-se 30 simulações para cada valor de malha, a estrutura do tipo *sparse* é mais estável, para valores próximos de malha, com relação ao tempo de execução.

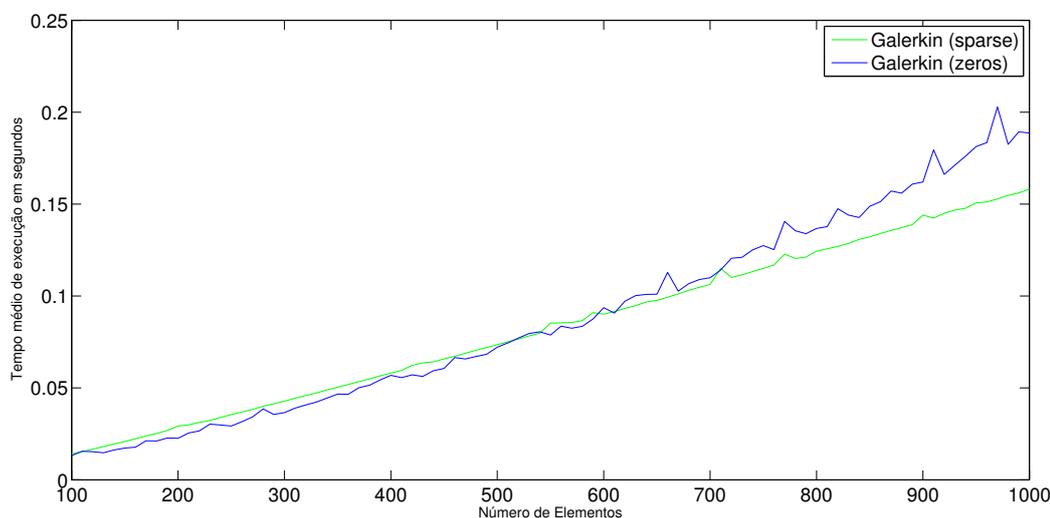


Figura 4.1: Número de onda  $k = 30$  no método de Galerkin 1D, com 30 simulações para cada resolução de malha, variando de 100 a 1000, em intervalos de 10.

Repara-se através do gráfico (4.1) que a complexidade do algoritmo é da ordem de  $O(n)$  [27] pois o tempo de execução aumenta linearmente conforme aumentamos a resolução da malha. A análise do tempo de execução não leva em consideração as funções de geração

gráfica da solução.

A figura (4.2) faz a mesma análise mas só que para o caso bidimensional. Nota-se para esse caso uma complexidade da ordem de  $O(n^2)$ , tanto para o caso *sparse* e *zeros*, que é evidenciado pelos dois "loops" de "for". Contudo, a curva da estrutura inicializada por *sparse* é bem menos acentuada que a por *zeros*. O ganho em desempenho acontece perto de 2200 elementos na malha, sem mencionar os ganhos em memória, podendo-se gerar malhas bem mais refinadas.

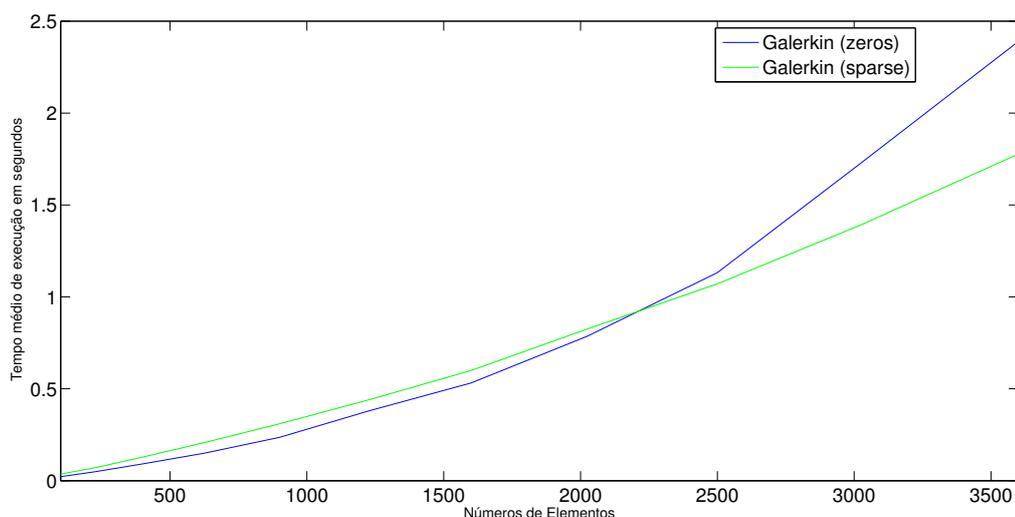


Figura 4.2: Número de onda  $k = 1$  no método de Galerkin 2D, com 15 simulações para cada resolução de malha, variando de 100 a 3600, em intervalos de 25.

O grande gasto computacional acontece dentro da resolução do sistema, pois a construção das matrizes em si tem um gasto bem inferior. Entretanto, o Matlab consegue contornar bem este fato, conseguindo excelentes resultados através da estrutura *sparse*.

É necessário dar ênfase a estrutura de dados para vetores esparsos pois os métodos numéricos que foram implementados exigem um gasto altíssimo de memória quando a malha é bastante refinada, até mesmo porque temos ainda a geração gráfica da solução do sistema.

## 4.2 Análise unidimensional

De início, introduz-se o que seria um interpolante para a solução exata e como se comporta em relação a uma solução analítica. É feita também uma análise sobre o efeito de *poluição* do erro, onde o refinamento da malha não garante uma convergência assintótica.

Analisa-se os métodos segundo a seminorma  $H^1$  e a norma  $L^2$  e que são definidas no Apêndice A.

Alguns aspectos levantados ao longo da análise unidimensional serão usados para o bidimensional. O tópico seguinte, por exemplo, relaciona o ajuste da malha ao número de onda. Pelo comportamento análogo em ambas dimensões, esse aspecto será levantado para o caso unidimensional e estendido para o bidimensional.

### 4.2.1 Interpolante e regra de aproximação

O interpolante é uma função que se ajusta a um conjunto de dados discretos. Por exemplo, no método de Galerkin tem-se soluções nos nós quando resolvido o sistema linear. Contudo, não há um valor entre os nós e para que se tenha usa-se o interpolante.

Para o caso unidimensional será usado o interpolante linear porque cada elemento do domínio possui dois nós. Dessa forma, por esses dois pontos passa-se uma única reta. Tendo a solução em cada nó, pode-se usar a relação (3.19) que é um somatório das funções base ponderadas pelas soluções em cada nó para interpolar a solução aproximada. O caso bidimensional é análogo, mas tem-se 4 nós por elemento e funções bilineares como base.

No capítulo 2 foi encontrada a solução do problema de Helmholtz com a característica oscilatória de um seno ou cosseno. Estas soluções são periódicas com comprimento de onda  $\lambda = 2\pi/k$ . Por outro lado, pode-se intuir que deve-se ter um número mínimo de pontos para cada oscilação ou ciclo, onde

$$n_{res} = \frac{\lambda}{h} \approx \text{constante} \quad (4.1)$$

dessa forma tem-se

$$n_{res} = \frac{2\pi}{kh} \quad (4.2)$$

ou ainda,

$$kh = \frac{2\pi}{n_{res}} = \text{constante} \quad (4.3)$$

sendo  $n_{res}$  a resolução da malha, indicando quantos elementos se tem por oscilação da solução.

Pode-se então escolher uma resolução de malha com  $n_{res} = 8$ , tendo assim um número de 8 elementos, ou 9 nós, para cada oscilação. Para esse mesmo valor de  $n_{res}$ , seguindo a relação (4.3), tem-se que  $kh \approx 0.8$ . Posteriormente, serão usados outros valores para a resolução da malha tais que consigam um erro menor, sendo um deles  $kh \approx 0.2$ . A figura

(4.3) exemplifica esse caso apresentado até aqui, tendo em mente um interpolante linear.

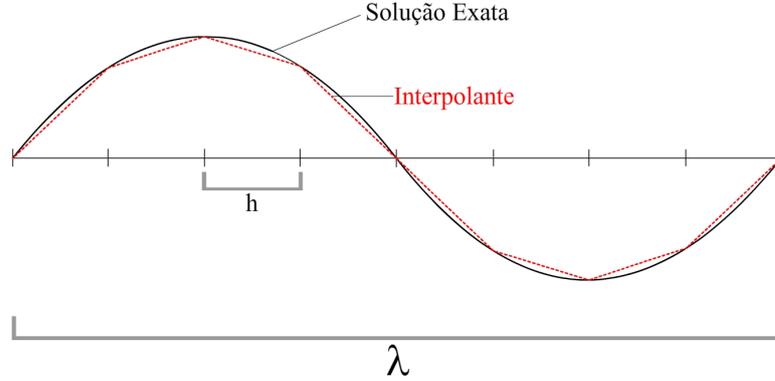


Figura 4.3: Regra do *Thumb* para interpolação com  $n_{res} = 8$ .

Essa ideia de interpolação que relaciona o número de onda  $k$  e a discretização do domínio, possui resultados que garantem o controle do erro nas normas  $H^1$  e  $L^2$ . Coloque-se assim um lema que se encontra em [26].

**Lema 4.1.** Seja  $u \in H^2(0, 1)$ , e seja  $u_I \in V^h(0, 1)$  um interpolante linear por partes de  $u$  em uma malha homogênea de intervalos  $h$ . Então

$$\|u - u_I\|_2 \leq \frac{h^2}{\pi} |u|_2, \quad (4.4)$$

$$|u - u_I|_1 \leq \frac{h}{\pi} |u|_2, \quad (4.5)$$

$$\|u - u_I\|_2 \leq \frac{h}{\pi} |u - u_I|_1. \quad (4.6)$$

sendo  $\|\cdot\|_2$  a norma no espaço  $L^2$ ,  $|\cdot|_1$  e  $|\cdot|_2$  seminormas nos espaços  $H^1$  e  $H^2$ , respectivamente.

Além disso, o caso unidimensional apresenta soluções como combinação linear de funções  $\text{sen}(kx)$  e  $\text{cos}(kx)$ , dessa forma é possível encontrar constantes que satisfazem as relações

$$\frac{|u|_2}{\|u\|} \leq C_1 k^2 \quad \text{e} \quad \frac{|u|_2}{|u|_1} \leq C_2 k \quad (4.7)$$

Assumindo que as funções  $u$  e  $u'$  não são identicamente nulas, podemos dividir (4.4) por  $\|u\|$  e (4.5) por  $|u|_1$  e que de posse de (4.7) tem-se a seguinte estimativa de erro para o interpolante:

$$\frac{\|u - u_I\|}{\|u\|} \leq C_3 h^2 k^2, \quad (4.8)$$

$$\frac{|u - u_I|_1}{|u|_1} \leq C_4 h k. \quad (4.9)$$

Portanto, conclui-se então que se a noção de resolução de malha do tipo  $kh = \text{constante}$  é seguida, o erro relativo é controlado nas normas  $H^1$  e  $L^2$ , conforme (4.8) e (4.9). A referência tomada como base para esta regra de aproximação é [15].

### 4.2.2 Efeito de poluição do erro

Na seção 3 do capítulo 3, quando tratamos da análise de dispersão, vimos que número de onda  $k$  difere-se do numérico  $k^h$ , o que compromete a qualidade da solução. Este fator torna-se crucial para o controle robusto do erro, portanto não podemos esperar que a resolução da malha do tipo  $kh = \text{constante}$  seja suficiente.

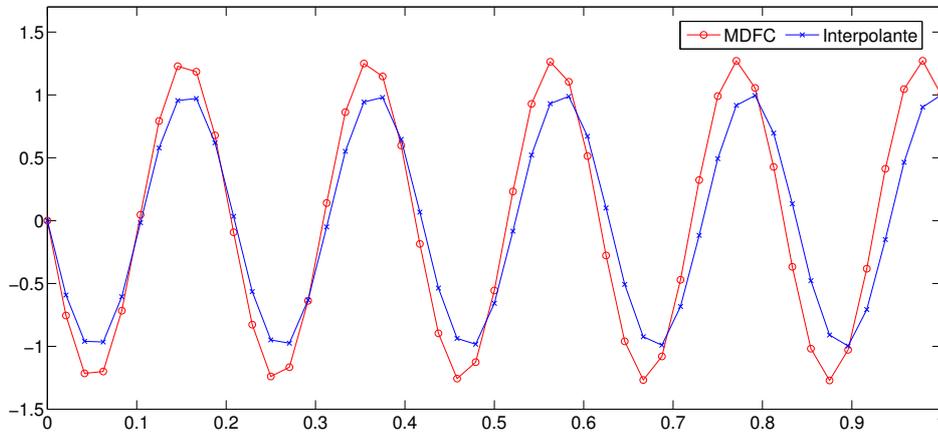


Figura 4.4: Aproximação por MDFC com número de onda  $k=30$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$ . A resolução da malha tem  $n_{res} = 10$ , ou seja,  $kh \approx 0.6$ .

O que se pretende mostrar agora é que o erro do interpolante é controlado para um  $kh = \text{constante}$ , entretanto, já para os métodos de diferenças finitas centradas e elementos finitos de Galerkin isso não acontece. Como já mencionado anteriormente, esse controle não robusto do erro mediante um refinamento da malha é o que leva o nome de poluição do erro. Mais claramente, dado um número de onda qualquer, não se pode garantir que o

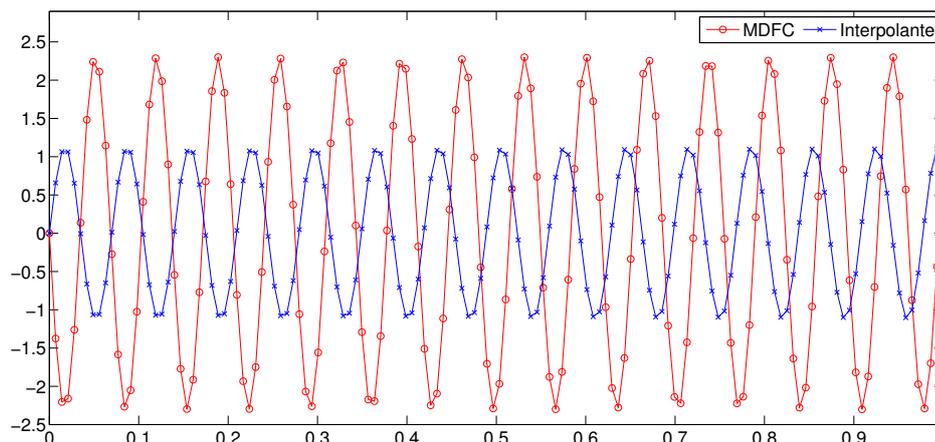


Figura 4.5: Aproximação por MDFC com número de onda  $k=90$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$ . A resolução da malha tem  $n_{res} = 10$ , ou seja,  $kh \approx 0.6$ .

refinamento diminua o erro em quaisquer normas. As figuras (4.4), (4.5) e (4.6) mostram que conforme aumenta-se o número de onda, mas mantendo a relação  $kh = constante$ , a solução aproximada pode estar bem longe do ideal.

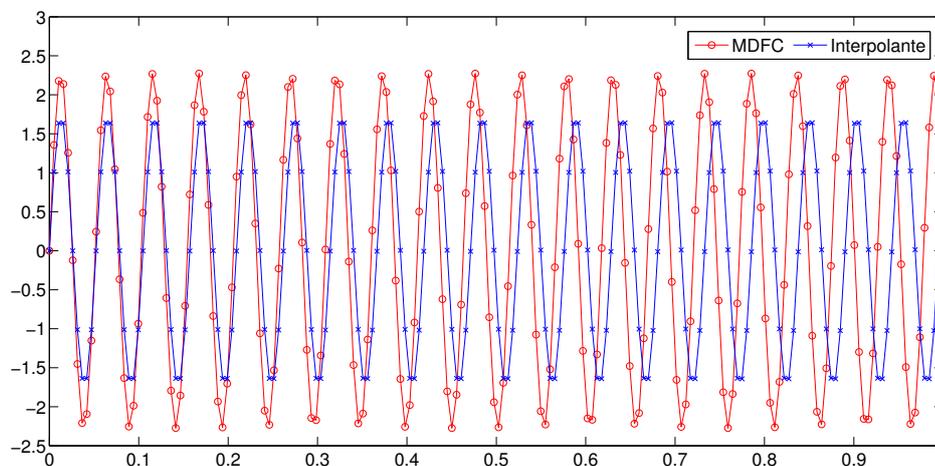


Figura 4.6: Aproximação por MDFC com número de onda  $k=120$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$ . A resolução da malha tem  $n_{res} = 10$ , ou seja,  $kh \approx 0.6$ .

Faz-se agora a mesma análise para o método de Galerkin. São escolhidos os mesmos números de onda ( $k = 30, 90, 120$ ), impondo-se as mesmas condições de contorno de Dirichlet, mantendo-se a resolução de malha com um  $kh$  constante, mostrando que o efeito de poluição também é presente neste método e não só no MDFC.

O resultado nas figuras (4.7), (4.8) e (4.9) mostram que realmente esse tipo de aproximação torna-se insuficiente em questões práticas, tornando-se necessário a busca de métodos que contornem esse problema numérico.

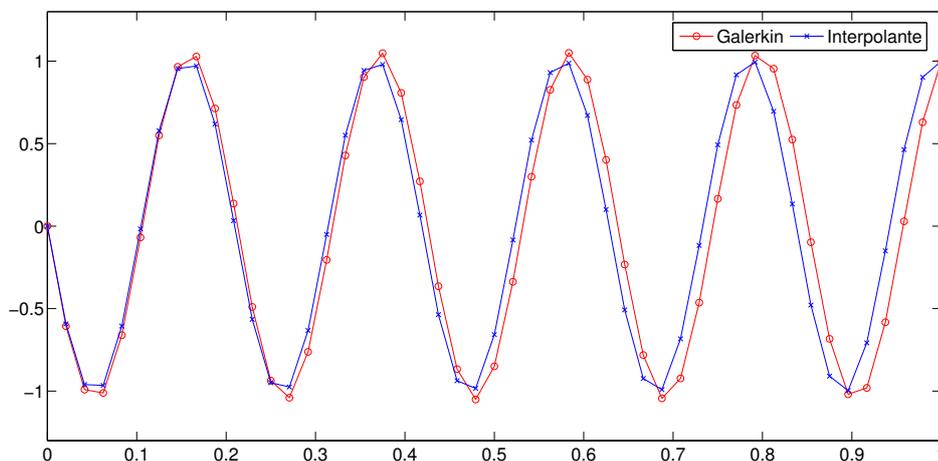


Figura 4.7: Aproximação por Galerkin com número de onda  $k=30$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$ . A resolução da malha tem  $n_{res} = 10$ , ou seja,  $kh \approx 0.6$ .

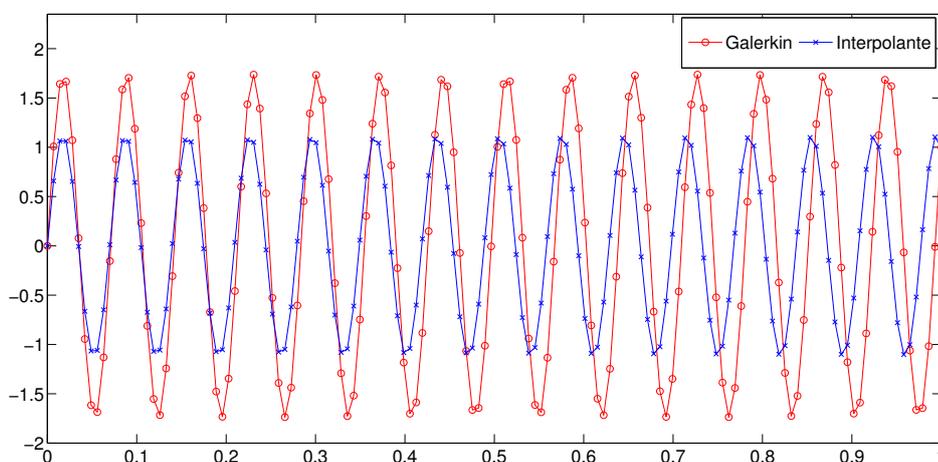


Figura 4.8: Aproximação por Galerkin com número de onda  $k=90$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$ . A resolução da malha tem  $n_{res} = 10$ , ou seja,  $kh \approx 0.6$ .

Os métodos de Galerkin e diferenças finitas são gravemente afetados pelo efeito de poluição como vimos. Para o problema homogêneo, isto é,  $f(x) = 0$ , o método GLS consegue eliminar o erro, fazendo com que a solução aproximada coincida com a do interpolante da solução exata.

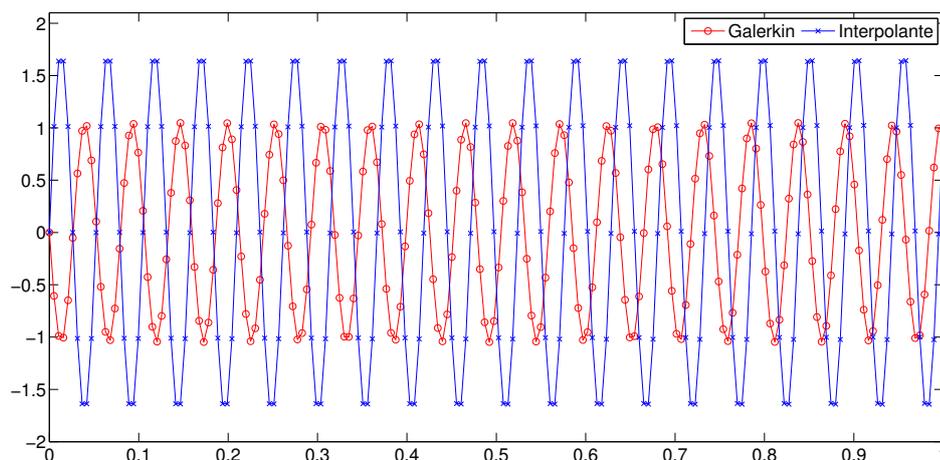


Figura 4.9: Aproximação por Galerkin com número de onda  $k=120$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$ . A resolução da malha tem  $n_{res} = 10$ , ou seja,  $kh \approx 0.6$ .

Manteve-se a resolução de malha, números de onda e condições de Dirichlet nos resultados para o GLS, mostrando como é contornado o efeito de poluição neste método nas figuras (4.10), (4.11) e (4.12).

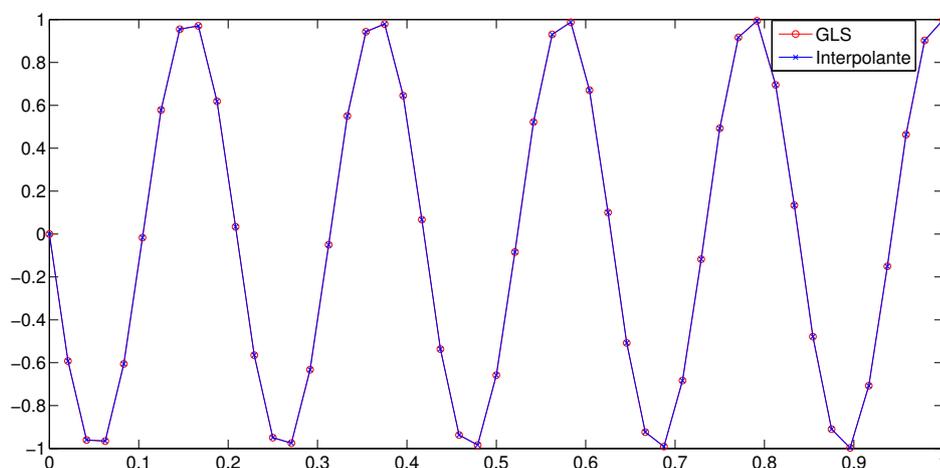


Figura 4.10: Aproximação por GLS com número de onda  $k=30$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$ . A resolução da malha tem  $n_{res} = 10$ , ou seja,  $kh \approx 0.6$ .

Assim, o caso homogêneo é suficientemente tratado com o método GLS, pois o erro é da mesma ordem do interpolante. Ainda resta avaliar o problema para o caso não homogêneo.

Escolhe-se fonte polinomial  $f(x) = k^2x$  para não influenciar na precisão integração

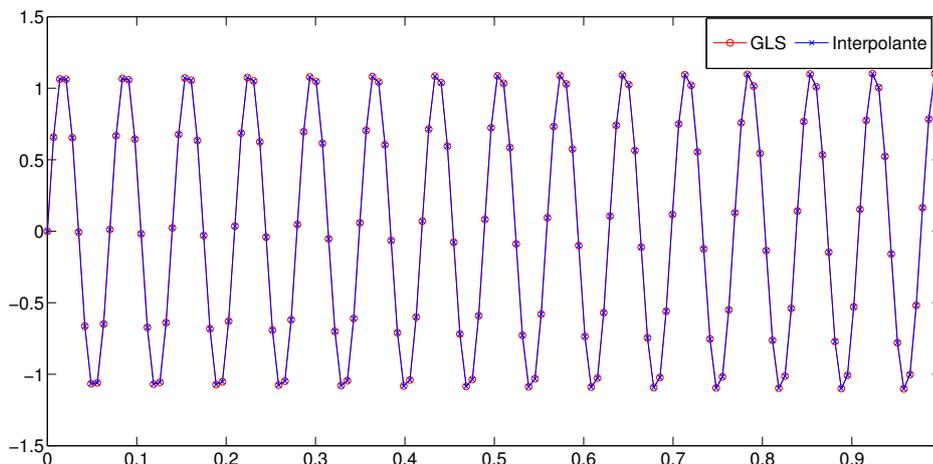


Figura 4.11: Aproximação por GLS com número de onda  $k=90$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$ . A resolução da malha tem  $n_{res} = 10$ , ou seja,  $kh \approx 0.6$ .

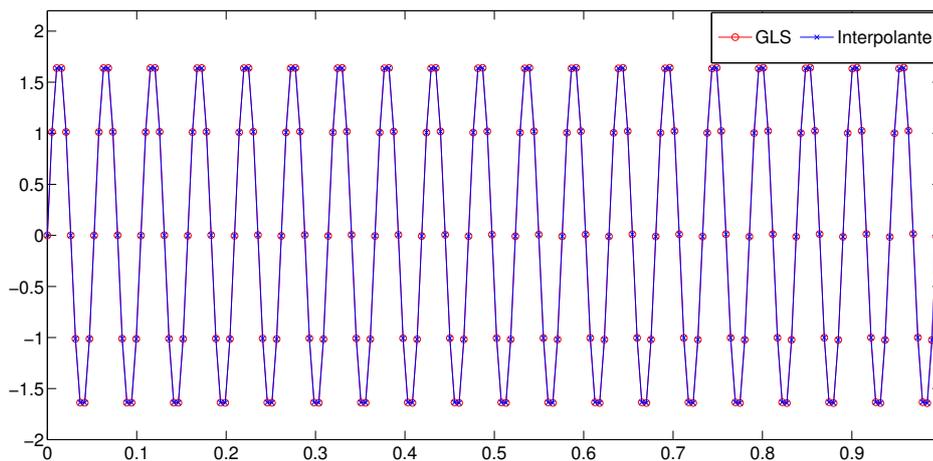


Figura 4.12: Aproximação por GLS com número de onda  $k=120$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$ . A resolução da malha tem  $n_{res} = 10$ , ou seja,  $kh \approx 0.6$ .

numérica, pretende-se avaliar o comportamento desses três métodos para um número de onda  $k = 80$ . Observa-se pelas figuras (4.13) e (4.14) que os métodos de Galerkin e diferenças finitas são insuficientes para  $kh$  constante, assim como para o problema homogêneo. Entretanto, o GLS mostra-se eficiente, conseguindo aproximar bem a solução, figura (4.15).

Nota-se que mesmo para o caso não homogêneo o método GLS é capaz de aproximar bem a solução em relação ao interpolante. Portanto, nitidamente o método de GLS tem um eficácia maior que os métodos de diferenças finitas e elementos finitos de Galerkin,

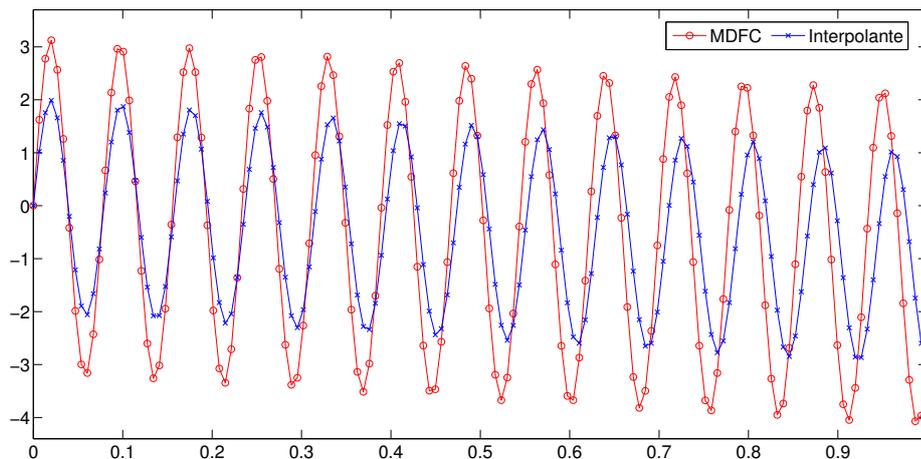


Figura 4.13: Aproximação por MDFC para o caso não homogêneo de  $f(x) = k^2x$ , para  $k=80$ , com condição de Dirichlet  $u(0) = 0$  e  $u(1) = -3$ , considerando  $kh \approx 0.6$ .

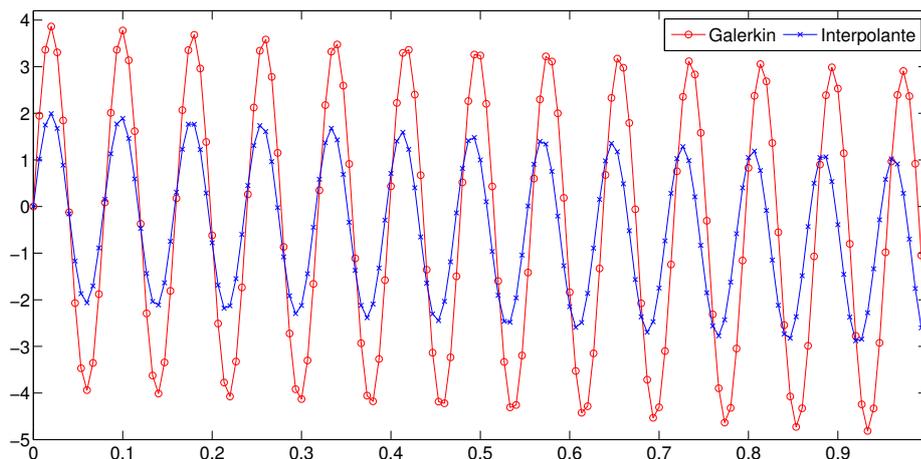


Figura 4.14: Aproximação por MDFC para o caso não homogêneo de  $f(x) = k^2x$ , para  $k=80$ , com condição de Dirichlet  $u(0) = 0$  e  $u(1) = -3$ , considerando  $kh \approx 0.6$ .

mesmo considerando a relação do  $kh \approx constante$ .

A análise até agora foi bastante visual, mostrando gráficos da solução aproximada dos métodos em relação ao interpolante. Dessa forma, resta apresentar uma análise de erro mais consistente que permita comparar os métodos e estudar sua convergência nas normas  $H^1$  e  $L^2$ .

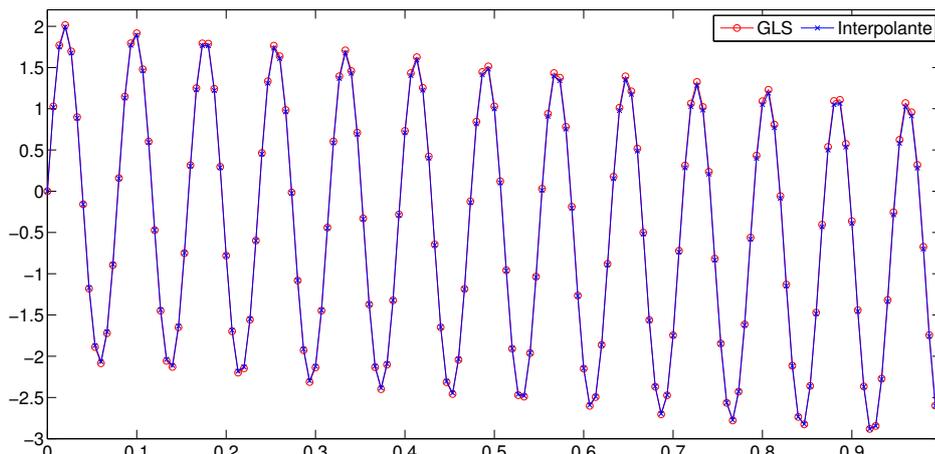


Figura 4.15: Aproximação GLS para o caso não homogêneo de  $f(x) = k^2x$ , para  $k=80$ , com condição de Dirichlet  $u(0) = 0$  e  $u(1) = -3$ , considerando  $kh \approx 0.6$ .

### 4.2.3 Análise de Erro

Uma forma de mostrar o efeito de poluição do erro é fixando um número de onda  $k$  qualquer e aumentando o refinamento da malha. Se não houvesse efeito poluição o erro diminuiria conforme o refinamento da malha, aumentando-se o número de elementos.

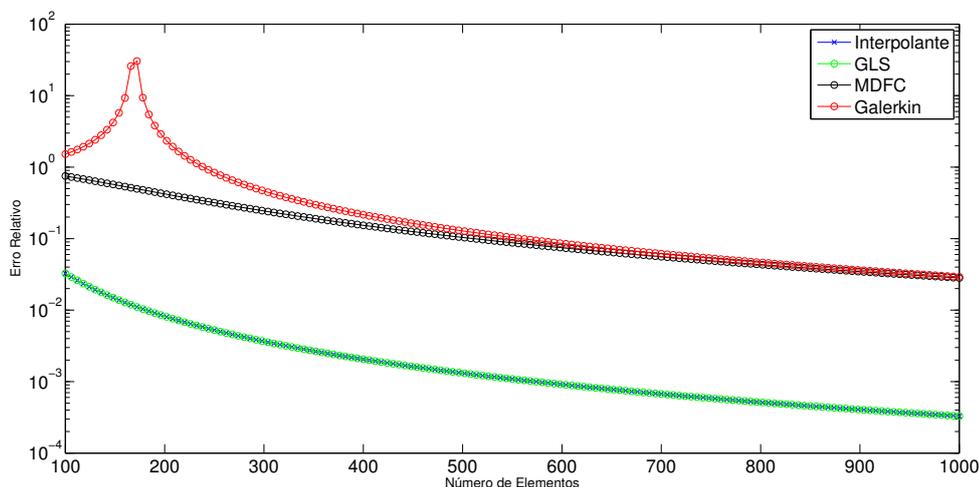


Figura 4.16: Gráfico do erro relativo do caso homogêneo na norma  $L^2$  para  $k=60$ .

Nas figuras (4.16) e (4.17) observa-se, tanto na norma  $H^1$  quanto na  $L^2$ , que o método de Galerkin apresenta um "pico" no erro relativo, o que caracteriza o efeito de ressonância. Acontece que aumentando-se o número de elementos na malha, o erro aumenta em certos pontos e só após 200 elementos a convergência é assintótica.

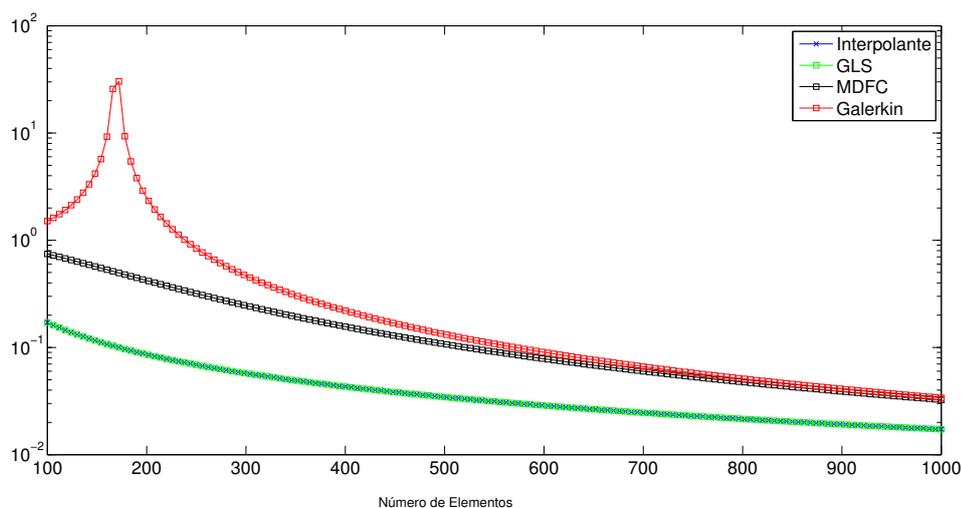


Figura 4.17: Gráfico do erro relativo do caso homogêneo na seminorma  $H^1$  para  $k=60$ .

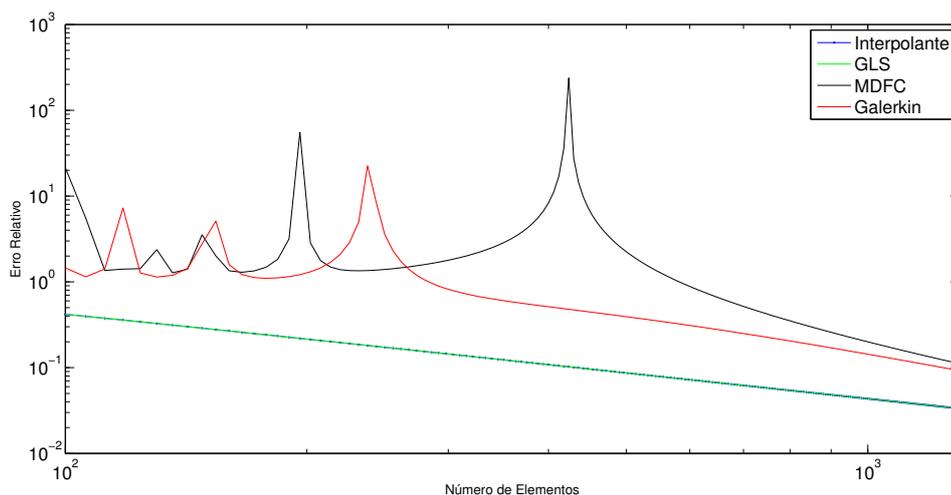


Figura 4.18: Gráfico do erro relativo do caso homogêneo na norma  $H^1$  para  $k = 200$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$

Na figura (4.18), em contraponto à figura (4.17), consegue-se ver dois aspectos interessantes. O primeiro é que para um número de onda mais baixo, na figura (4.17), o MDFC comporta-se melhor que o método de Galerkin, já para um mais alto vê-se o contrário, conforme figura (4.18). Um segundo ponto é que para um número de onda alto o efeito de poluição é mais grave, comprometendo seriamente a qualidade da solução.

O efeito de poluição do erro também acontece no caso não homogêneo conforme o resultado na figura (4.19), tanto na norma  $H^1$  quanto na  $L^2$ .

Foi dito anteriormente que mantendo-se uma relação  $kh$  igual uma constante o erro é

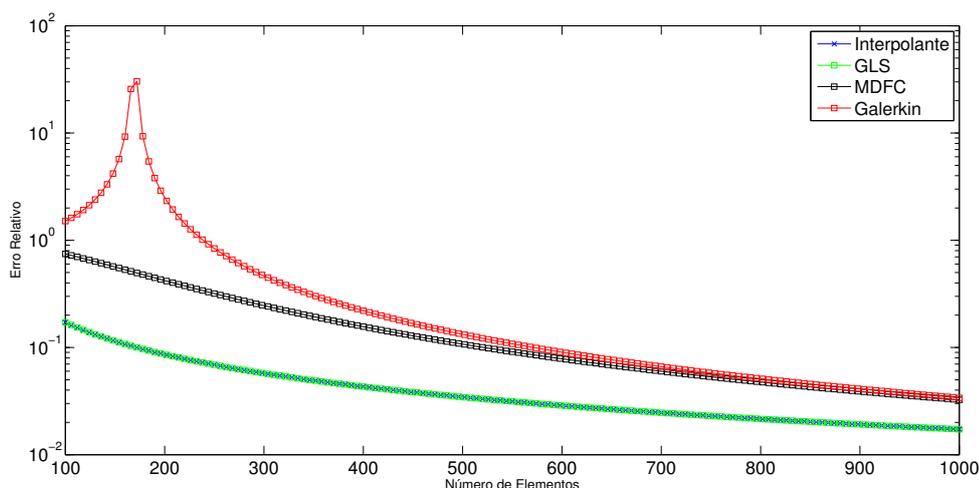


Figura 4.19: Gráfico do erro relativo do caso não homogêneo, com  $f(x) = k^2x$  na norma  $H^1$  para  $k=60$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = -3$

controlado para o interpolante mas não para os métodos de diferenças finitas e de Galerkin. A figura (4.20) mostra exatamente este efeito, onde mantém-se a relação  $kh \approx \text{constante}$  e aumenta-se o número de onda  $k$ , e vê-se que o erro relativo tende a aumentar.

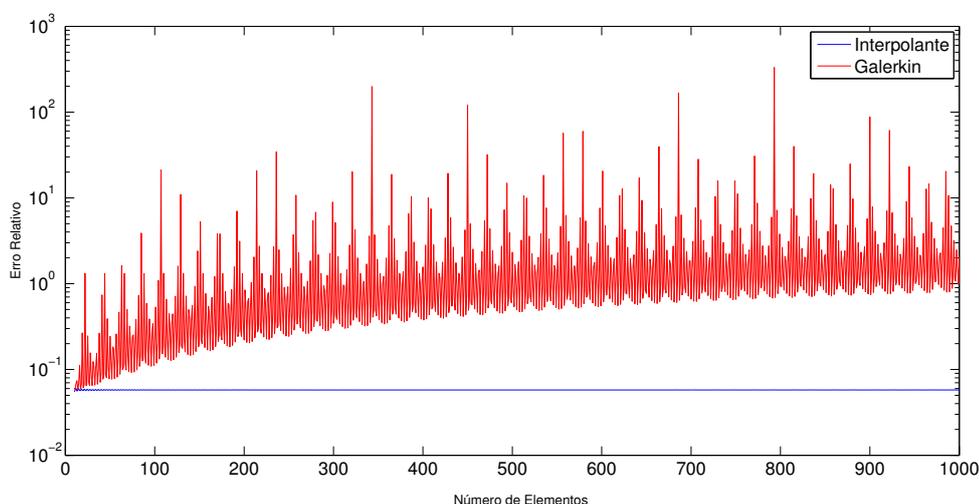


Figura 4.20: Erro na seminorma  $H^1$  para o problema homogêneo mantendo-se a relação  $kh = 0.2$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$ .

Se fizermos uma relação do tipo  $k^2h \approx \text{constante}$  o erro relativo é baixo e é controlado de forma satisfatória. Contudo, essa relação um pouco mais robusta é impraticável para um número de onda elevado já que exige muitos elementos na malha. A figura (4.21) mostra como o erro é controlado, mas o  $k$  não pode ser muito alto para que seja computável em tempo razoável e que a memória do computador permita alocar tantos nós/elementos.

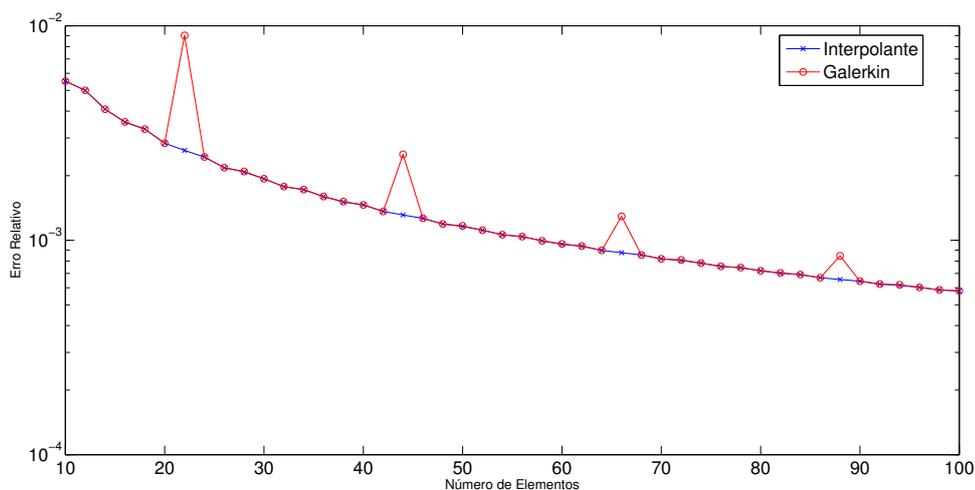


Figura 4.21: Erro na seminorma  $H^1$  para o problema homogêneo mantendo-se a relação  $k^2h = 0.2$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$ .

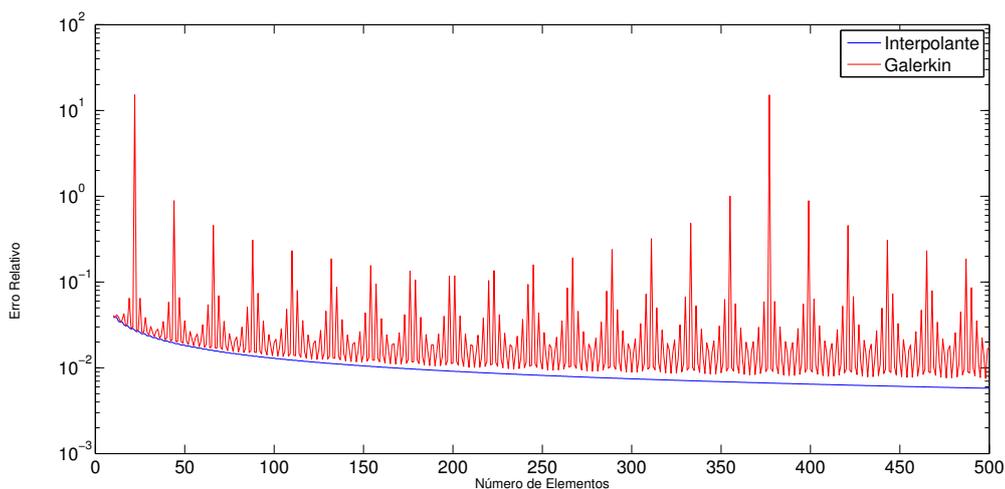


Figura 4.22: Erro na seminorma  $H^1$  para o problema homogêneo mantendo-se a relação  $k^3h^2 = 0.2$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$ .

Mas ainda, pode-se optar por uma relação do tipo  $k^3h^2 \approx \text{constante}$ , pois não exige um esforço computacional como a relação  $k^2h$ , mas também não é tão branda quanto a  $kh$ . A figura (4.22) mostra que há uma certa convergência que segue a do interpolante, embora apareça alguns picos elevados no erro relativo.

Pode-se esperar que esse comportamento do erro também ocorra na norma  $L^2$ , assim como foi observado nos gráficos anteriores na norma  $H^1$ , e também para o método de diferenças finitas. De fato esse fenômeno numérico aparece conforme o resultado da figura (4.23), para norma  $L^2$  com o método de diferenças finitas.

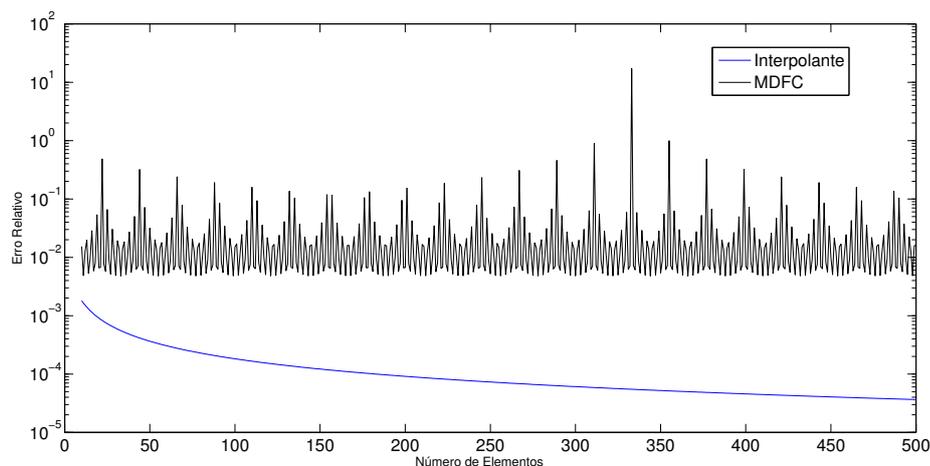


Figura 4.23: Erro na norma  $L^2$  para o problema homogêneo mantendo-se a relação  $k^3 h^2 = 0.2$  em condição de Dirichlet  $u(0) = 0$  e  $u(1) = 1$ .

|             | MDFC                 | Galerkin             | GLS                  | Interpolante         |
|-------------|----------------------|----------------------|----------------------|----------------------|
| Norma $L^2$ | $1,38 \cdot 10^{-1}$ | $1,48 \cdot 10^{-1}$ | $6,48 \cdot 10^{-3}$ | $6,48 \cdot 10^{-3}$ |
| Norma $H^1$ | $1,58 \cdot 10^{-1}$ | $1,68 \cdot 10^{-1}$ | $7,68 \cdot 10^{-2}$ | $7,68 \cdot 10^{-2}$ |

Tabela 4.1: Erro relativo para o problema homogêneo para  $k=80$ , com 300 pontos em condição de Dirichlet no contorno.

A tabela (4.1) mostra mais claramente a ordem dos erros para o caso homogêneo do problema de Helmholtz. É possível reparar que o erro para o método de Galerkin e MDFC são relativamente próximos e o GLS é exatamente o do interpolante, para as normas  $H^1$  e  $L^2$ . Foi considerado uma malha bastante refinada, com 300 pontos, com o objetivo de sair da zona de poluição numérica e comparar devidamente os métodos.

A tabela (4.2) mostra o mesmo da (4.1) mas com a diferença de que foi tratado nela o problema não homogêneo. Repara-se que o erro de Galerkin e MDFC não se altera muito, contudo o erro de GLS é maior que do interpolante para a norma  $L^2$ , o que é inevitável nessa norma quando tratamos de soluções que não coincidem. Contudo, na norma  $H^1$  o erro é o mesmo porque ela mede a variação da solução, o que torna possível o método ter

|             | MDFC                 | Galerkin             | GLS                  | Interpolante         |
|-------------|----------------------|----------------------|----------------------|----------------------|
| Norma $L^2$ | $1,28 \cdot 10^{-1}$ | $1,38 \cdot 10^{-1}$ | $6,01 \cdot 10^{-3}$ | $4,32 \cdot 10^{-3}$ |
| Norma $H^1$ | $1,58 \cdot 10^{-1}$ | $1,67 \cdot 10^{-1}$ | $7,68 \cdot 10^{-2}$ | $7,68 \cdot 10^{-2}$ |

Tabela 4.2: Erro relativo para o problema não homogêneo para  $k=80$ , com 300 pontos em condição de Dirichlet no contorno.

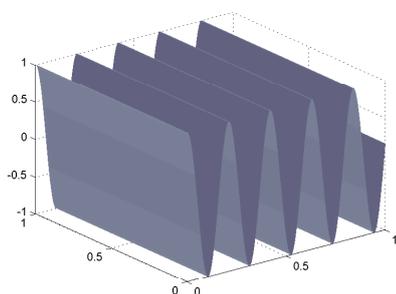
o mesmo erro do interpolante.

### 4.3 Análise bidimensional

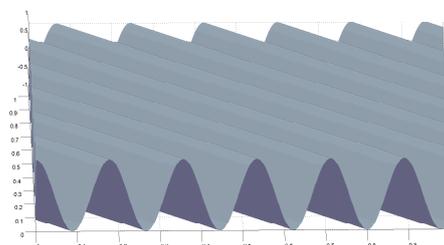
A análise bidimensional verificará o efeito de poluição, assim como no caso unidimensional, mostrando a relação entre refinamento da malha e o número de onda  $k$ , bem como o efeito da ressonância numérica em nossos resultados.

O problema em duas dimensões apresenta uma nova característica: direção da onda. Podemos ter uma superposição de ondas planas como solução do problema e essa direção interfere na solução aproximada do problema, como será visto posteriormente. A figura (4.24) mostra a solução exata para ondas planas e que há a possibilidade de maior oscilação na solução, o que pode comprometer ainda mais a solução aproximada.

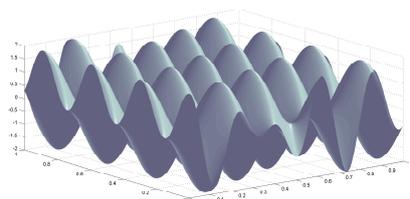
#### 4.3.1 Efeito de Poluição do erro



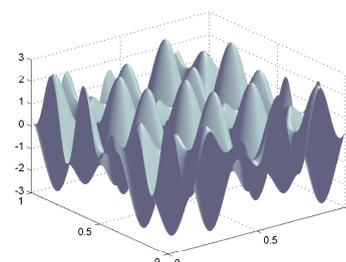
(a)  $\theta_1 = 0$



(b)  $\theta_1 = \frac{\pi}{4}$



(c)  $\theta_1 = 0$  e  $\theta_2 = \frac{\pi}{4}$



(d)  $\theta_1 = 0, \theta_2 = \frac{\pi}{4}$  e  $\theta_3 = \frac{\pi}{2}$

Figura 4.24: As figuras (a) e (b) apresentam a solução exata para uma onda plana com  $k = 30$  e  $k = 50$ , respectivamente. As figuras (c) e (d) são para duas e três ondas planas, respectivamente, e ambas para  $k = 30$ .

É possível intuir pelas soluções exatas, e por ser um problema em duas dimensões, que o efeito de poluição seja maior ou igual ao caso unidimensional. Vê-se que o caso bidimensional tem uma maior liberdade para oscilações, comparado ao caso unidimensional, por isso o efeito de poluição pode comprometer ainda mais a solução aproximada.

Observa-se numericamente que o efeito de poluição aparece no caso bidimensional, conforme os resultados nas figuras (4.25) e (4.26) a seguir. Esses resultados ilustram o comportamento do MDFC e que aumentando o número de onda  $k$  a solução aproximada pode afastar-se bastante do interpolante.

Dependendo de onde se faça o corte para analisar a solução, a aproximação pode ser melhor ou pior. Essa análise em corte é apenas uma primeira investigação do efeito de poluição para o problema em duas dimensões. Contudo, quando se calcula o erro, ele será computado no domínio todo.

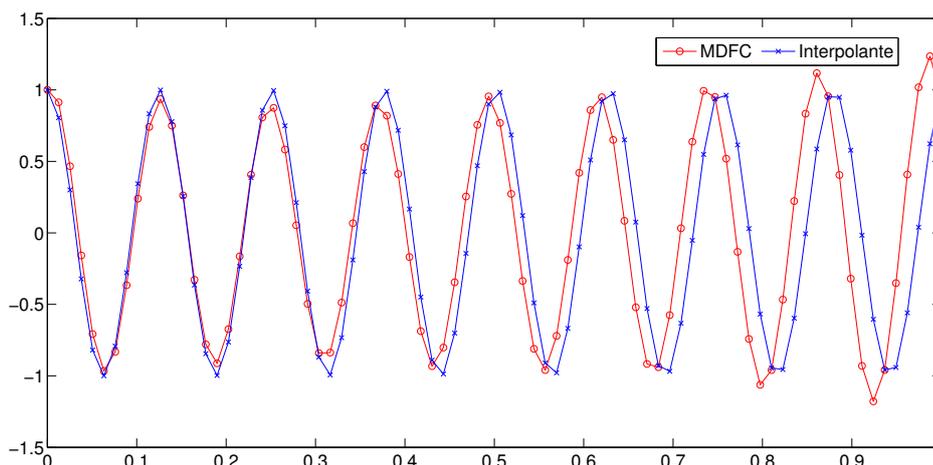


Figura 4.25: Aproximação por MDFC com número de onda  $k=50$  em condição de Dirichlet, com corte em  $y = 0,4792$ , para uma onda plana com direção  $\theta_1 = 0$  e malha  $80 \times 80$ .

O método de Galerkin também é afetado pelo efeito de poluição como é visto na figura (4.27). Vale mencionar que é usada uma regra análoga ao  $kh$  igual uma constante, porém estamos agora no caso bidimensional, fazendo essa divisão tanto para o eixo  $x$  quanto para o  $y$ . Será observado em análises posteriores que a relação  $kh$  também é insuficiente conforme o número de onda aumenta.

A aproximação pelo método GLS, para uma onda plana e malha uniforme, coincide com o interpolante em duas direções de onda, levando em consideração o primeiro quadrante do círculo trigonométrico, que são em  $\pi/4$  e  $3\pi/4$ . Mas se a escolha for feita para

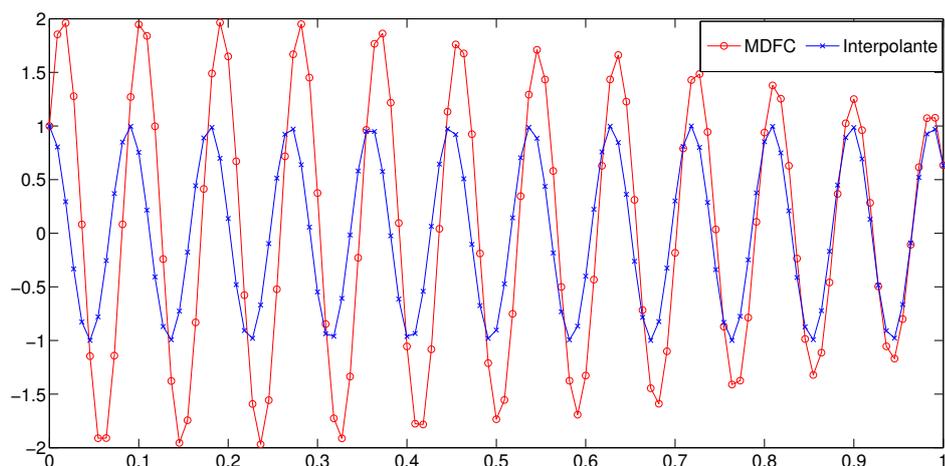


Figura 4.26: Aproximação por MDFC com número de onda  $k=70$  em condição de Dirichlet, com corte em  $y = 0,4792$ , para uma onda plana com direção  $\theta_1 = 0$  e malha  $111 \times 111$ .

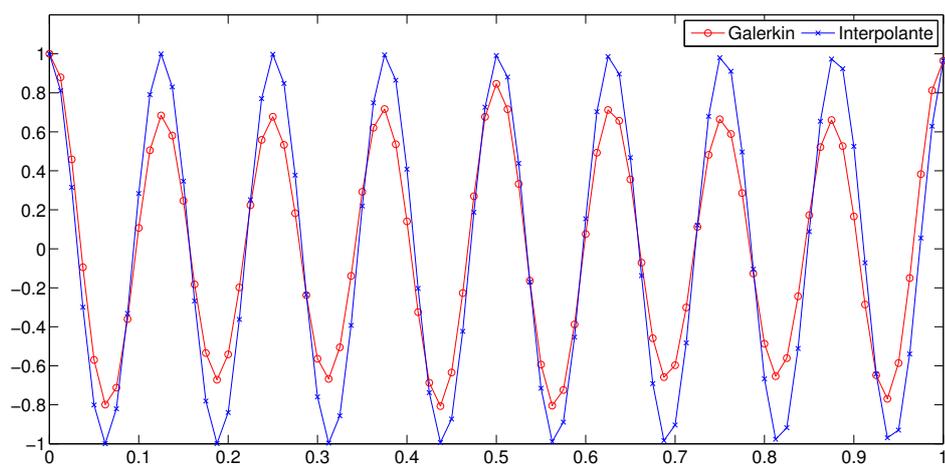


Figura 4.27: Aproximação de Galerkin com número de onda  $k=50$  em condição de Dirichlet, com corte em  $y = 0,4792$ , para uma onda plana com direção  $\theta_1 = 0$  e malha  $80 \times 80$ .

outras direções da onda, ou até mesmo uma superposições de ondas com direções distintas, o erro é da mesma ordem do método de Galerkin. Podemos ver a solução do método GLS conforme a figura (4.28), onde o método é igual ao interpolante.

O resultado apresentado na figura (4.29) é para o método GLS, onde a direção da onda na solução exata é  $\theta = 0$  e a condição de contorno é de Dirichlet com uma onda plana nessa mesma direção. Como já dito anteriormente e pela análise de dispersão do

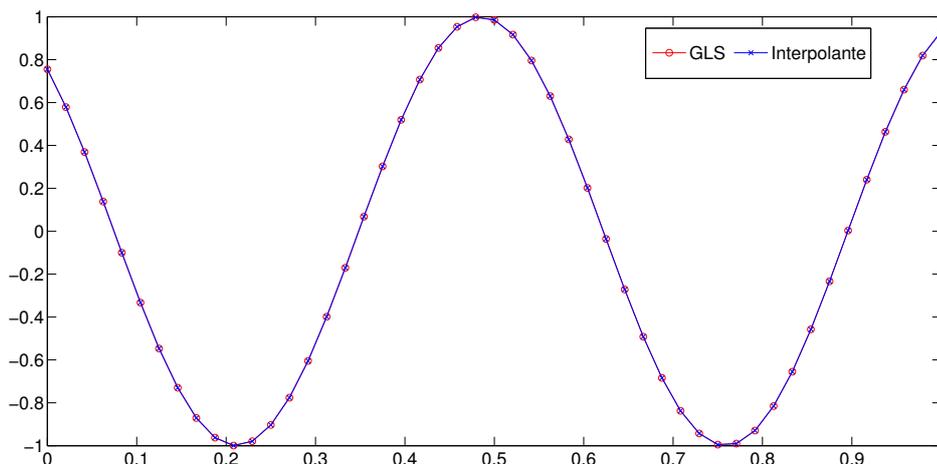


Figura 4.28: Aproximação de GLS com número de onda  $k=30$  em condição de Dirichlet, com corte em  $y = 0,4792$ , para uma onda plana com direção  $\theta_1 = 3\pi/8$  e malha  $49 \times 49$ .

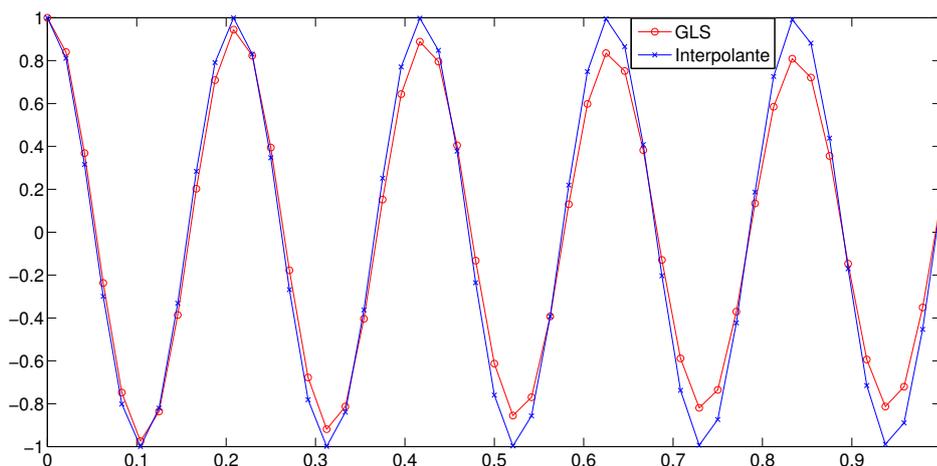


Figura 4.29: Aproximação de GLS com número de onda  $k=30$  em condição de Dirichlet, com corte em  $y = 0,4792$ , para uma onda plana com direção  $\theta_1 = 0$  e malha  $49 \times 49$ .

método, sabe-se que a solução não é igual a do interpolante.

Até o momento foram apresentados os mesmos métodos que foram levados em consideração na análise unidimensional do problema e vimos que eles não são suficientes para uma boa aproximação da solução. Enfatiza-se que, no problema bidimensional, a direção da onda terá um papel importante na análise de erro. Portanto, o erro sofrerá alterações dependendo da direção da onda.

O método QSFEM possui duas características, uma é semelhante a do GLS e a outra é

o que garante melhores resultados. A primeira característica é que a solução aproximada, para uma onda plana, coincide em 4 direções com a solução exata, o GLS era em duas. A segunda característica é que fora dessas direções onde a solução é igual a do interpolante, o erro do método é de ordem mais alta que a do Galerkin, MDFC e GLS.

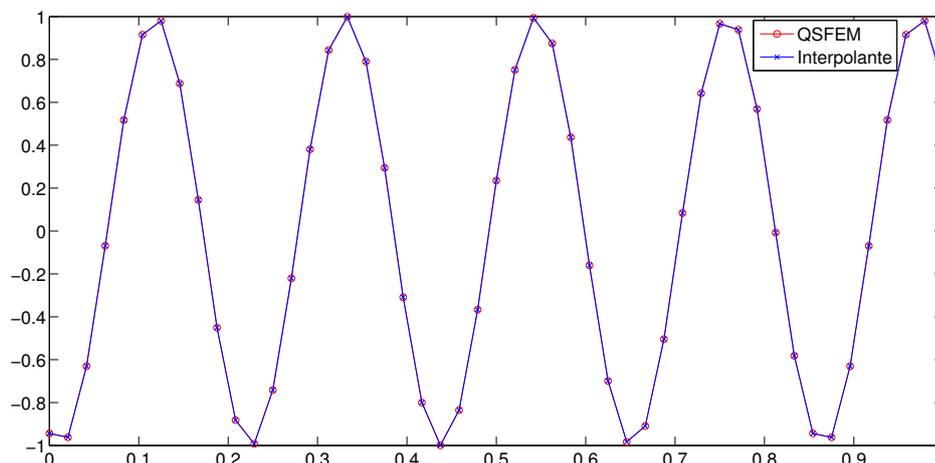


Figura 4.30: Aproximação de QSFEM com número de onda  $k=30$  em condição de Dirichlet, com corte em  $y = 0,4792$ , para uma onda plana com direção  $\theta_1 = \pi/16$  e malha  $49 \times 49$ .

A solução para o QSFEM para uma onda plana na direção  $\theta = \pi/16$  é igual a do interpolante, conforme a figura (4.30). A figura (4.31) mostra a solução para uma direção

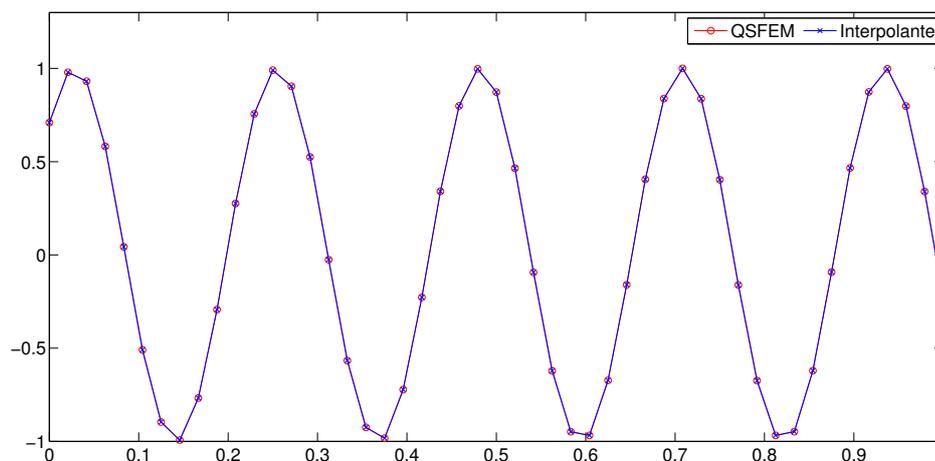


Figura 4.31: Aproximação de QSFEM com número de onda  $k=30$  em condição de Dirichlet, com corte em  $y = 0,4792$ , para uma onda plana com direção  $\theta_1 = \pi/8$  e malha  $49 \times 49$ .

onde a solução não é a do interpolante,  $\theta = \pi/8$ , entretanto não se vê diferença na inspeção visual.

### 4.3.2 Análise de Erro

Uma primeira análise que se pode fazer é a da dependência do erro e a da direção da onda plana da solução exata. A figura (4.32) mostra o método GLS coincidindo com o interpolante em dois pontos, como foi previsto anteriormente. O método QSFEM não apresenta muita diferença comparando-se com o interpolante, por isso inclui-se a figura (4.33), mostrando a ínfima diferença entre QSFEM e o interpolante.

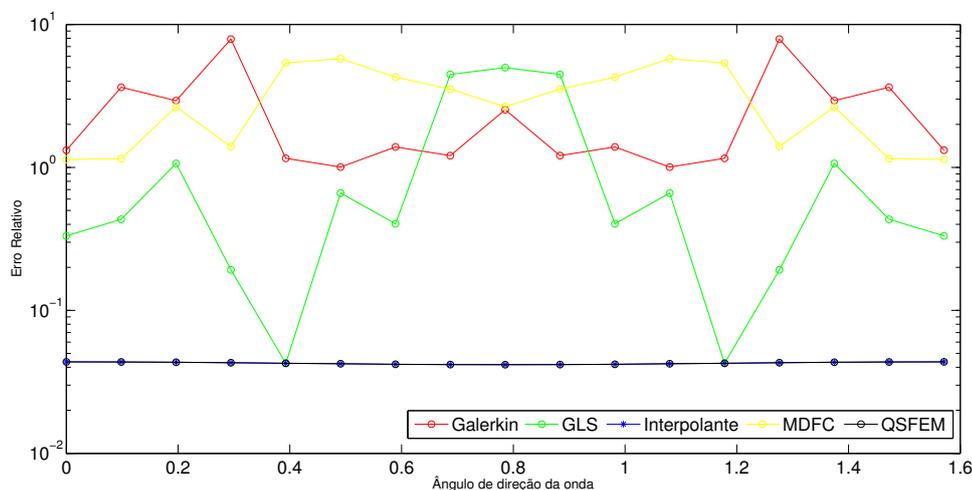


Figura 4.32: Gráfico do erro relativo na norma  $H^1$  considerando três ondas planas na mesma direção, com  $k = 80$  e malha  $200 \times 200$ . O ângulo de direção da onda varia de 0 a  $\pi/2$ .

A figura (4.34) exhibe o resultado para três ondas planas onde são fixadas duas direções e deixa-se uma outra variar. O método GLS não é igual ao interpolante, embora apresente pontos de mais baixo erro. O QSFEM mais uma vez tem o erro próximo ao do interpolante. Contudo, vê-se que o erro do interpolante depende da direção da onda, mas que não compromete gravemente a solução. Pode-se afirmar que a regra que controla o erro para uma onda não será a mesma para condições mais gerais de contorno, por exemplo três ondas planas.

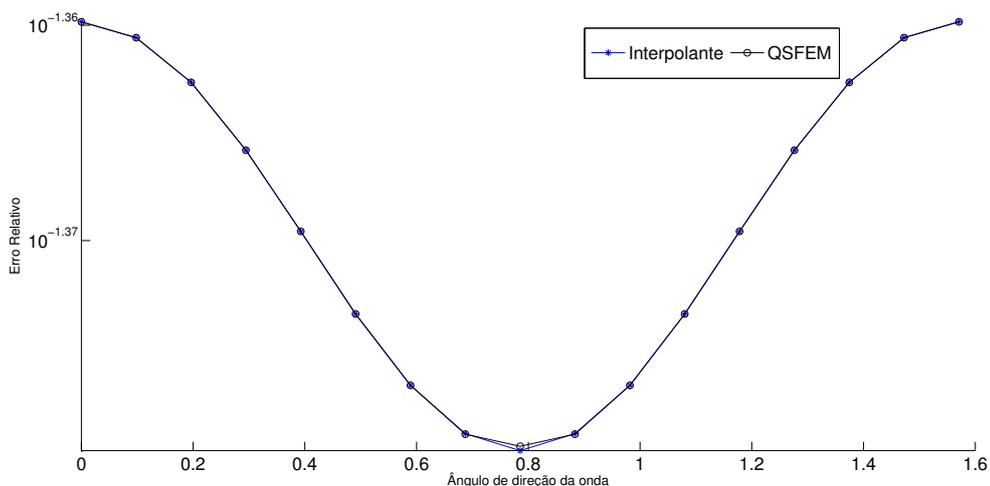


Figura 4.33: Resultado igual ao da figura (4.32) mas considerando apenas o interpolante e o método QSFEM

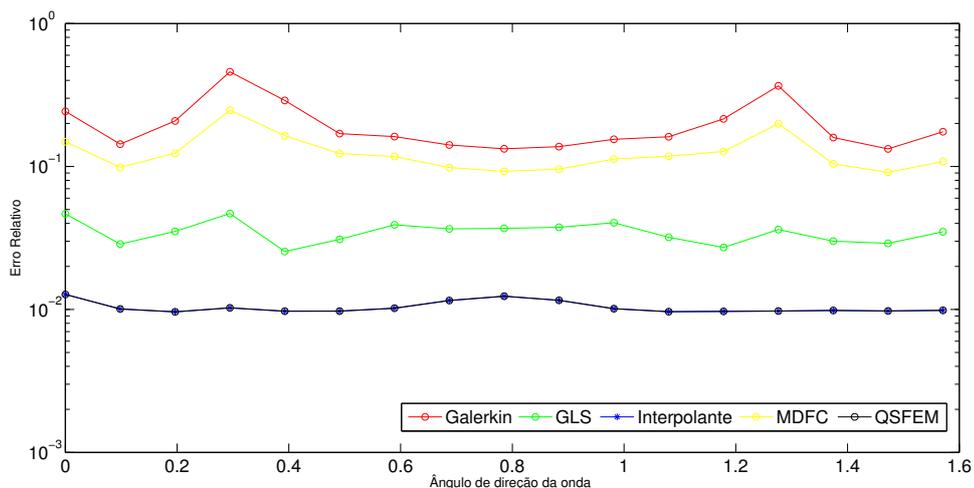


Figura 4.34: Gráfico do erro relativo na norma  $L^2$  considerando três ondas planas, duas fixadas em  $\theta_1 = \pi/4$  e  $\theta_2 = 0$ , e uma variando de 0 a  $\pi/2$ , com  $k = 80$  e malha  $200 \times 200$

As figuras (4.35) e (4.36) mostram os erros relativos para um número de onda relativamente baixo,  $k = 30$ , onde o efeito de poluição do erro aparece no método de Galerkin. Conforme aumenta-se o número de onda esse efeito tende a aumentar.

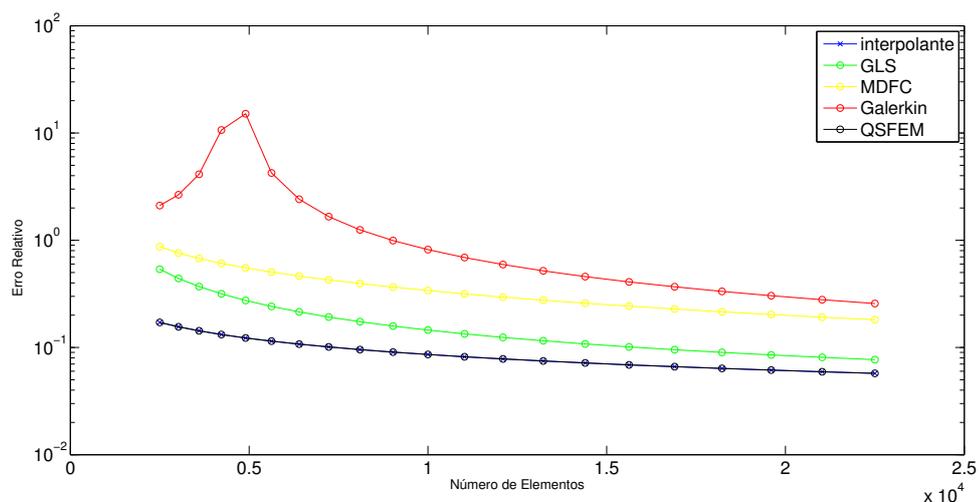


Figura 4.35: Gráfico do erro relativo, considerando caso homogêneo, do problema bidimensional para a norma  $H^1$ . E tem-se os parâmetros  $k = 30$ , uma onda plana na direção  $\theta_1 = 0$  e malha variando de  $50 \times 50$  até  $150 \times 150$ .

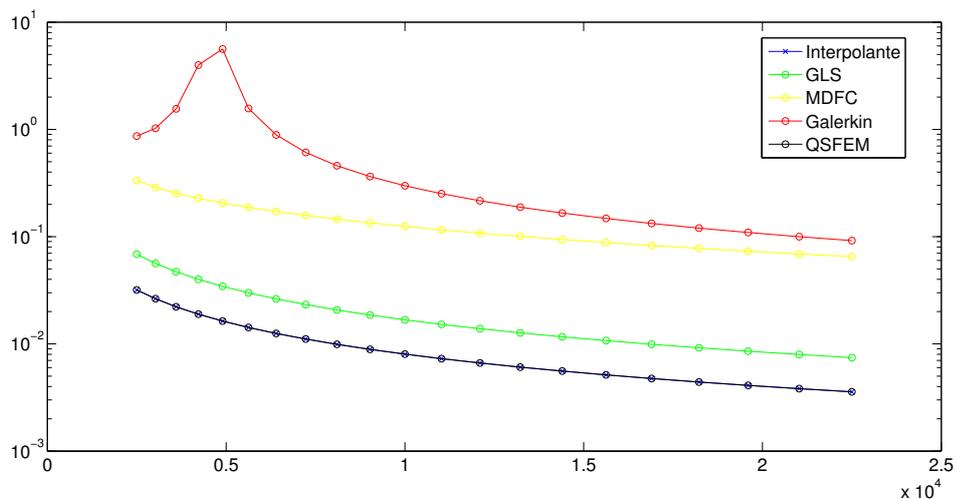


Figura 4.36: Gráfico do erro relativo, considerando caso homogêneo, do problema bidimensional para a norma  $H^1$ . E tem-se os parâmetros  $k = 30$ , três ondas planas nas direções  $\theta_1 = 0$ ,  $\theta_2 = \pi/8$ ,  $\theta_3 = \pi/4$  e malha variando de  $50 \times 50$  até  $150 \times 150$ .

A figura (4.37) mostra que para um número de onda elevado, no caso  $k = 110$ , o método de elementos finitos de Galerkin tende a ser mais eficiente que o método de diferenças finitas, assim como verificou-se no caso unidimensional. É possível notar que conforme aumentamos o número de onda o *efeito de poluição do erro* também aumenta, tal como no caso unidimensional.

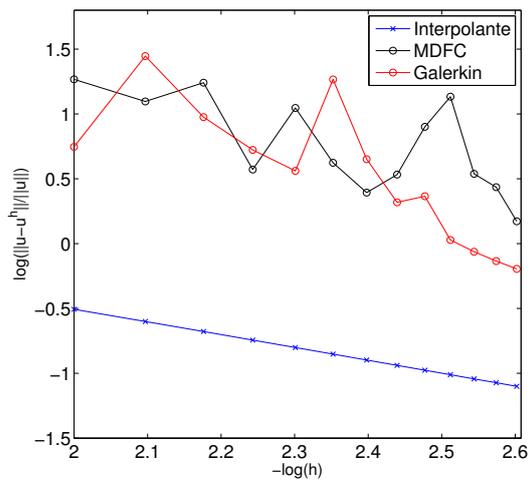
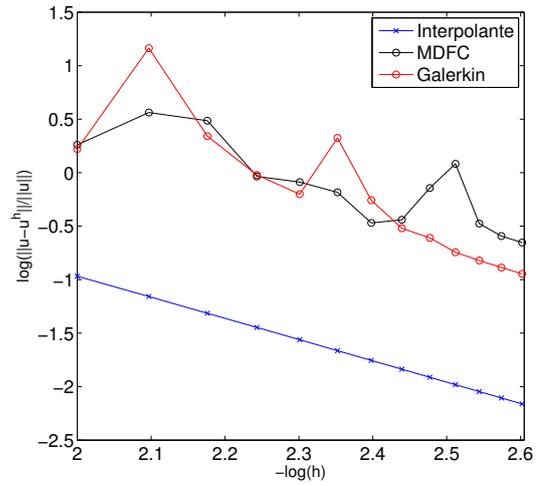
(a) Seminorma em  $H^1$ (b) Norma em  $L^2$ 

Figura 4.37: Gráfico do erro relativo, considerando caso homogêneo, do problema bidimensional para a seminorma  $H^1$  em (a) e norma  $L^2$  em (b). E tem-se os parâmetros  $k = 110$ , uma onda plana com  $\theta = 0$  e malha variando de  $100 \times 100$  até  $400 \times 400$ .

# Capítulo 5

## Conclusões e Trabalhos Futuros

### 5.1 Conclusões

Um primeiro objetivo dessa dissertação foi a análise de métodos clássicos para resolução de equações diferenciais parciais quando aplicados na equação de Helmholtz. Foram utilizados os métodos de Galerkin e diferenças finitas centradas de segunda ordem. Viu-se que estes métodos clássicos não são eficazes para bem aproximar a solução do problema. Com a finalidade de aproximar melhor a solução foram utilizados os métodos GLS e QSFEM, tanto no caso unidimensional, quanto no caso bidimensional.

Foram apresentados resultados, primeiramente, para o caso unidimensional. Nota-se que tanto os métodos de diferenças finitas, quanto o elementos finitos de Galerkin apresentam uma baixa qualidade da solução aproximada, bem como o *efeito de poluição* do erro, conforme figuras 4.4 até 4.9, para o caso homogêneo. Apresentou-se, com a finalidade de contornar esse problema, o método GLS que entrega a solução exata em cada nó, ou seja, erro da mesma ordem do interpolante, conforme figuras 4.10 até 4.12. No caso não homogêneo, ou seja, com termo fonte  $f(x)$ , o *efeito de poluição* do erro persiste para os dois primeiros métodos e a qualidade da solução também não é boa. Já para o GLS, visivelmente, não se encontra quase diferença entre o interpolante e a solução entregue pelo método, conforme figura 4.15.

Outra análise feita, ainda para o caso unidimensional, é a de relacionar o refino da malha, o número de onda  $k$  e *efeito de poluição* do erro. As figuras 4.16 e 4.17 mostram que mesmo refinando-se a malha o erro não diminui e vê-se assim o *efeito de poluição* do erro mais claramente e em ambas as normas. E este efeito no problema de Helmholtz é mais grave conforme aumentamos o número de onda, a figura 4.18 mostra diversos picos antes do erro começar a convergir. Um fato que se deve notar é que o método de Galerkin

é menos eficiente que o de diferenças finitas para um número de onda baixo, conforme figuras 4.16 e 4.17. Contudo o método de Galerkin é melhor quando o número de onda é mais alto, como se vê na figura 4.18.

Portanto, para o caso unidimensional, o GLS é suficiente para que se tenha uma excelente qualidade da solução aproximada, o que pode-se verificar nas tabelas 4.1 e 4.2 quando verificamos os erros nas normas  $L^2$  e  $H^1$ .

A análise prosseguiu para duas dimensões, já com a previsão de encontrar os mesmos entraves numéricos com relação ao *efeito de poluição* do erro. Além disso, para o caso bidimensional as soluções em ondas planas permitem uma maior oscilação, já que para uma dimensão a solução era composta por um seno. As figuras 4.25, 4.26 e 4.27 mostram que a qualidade da solução entregue não é boa. A figura 4.32 mostra que até mesmo o GLS não apresenta uma boa solução, pois a dependência da direção da onda interfere no erro. Dessa forma, foi apresentado o QSFEM que é bastante superior aos três métodos anteriores. As figuras 4.30 até 4.36 mostram o ótimo resultado que esse método fornece para o caso homogêneo.

## 5.2 Trabalhos Futuros

O QSFEM apresenta o mesmo resultado do GPR [8, 7] para malhas uniformes e sem termo fonte. Contudo, o GPR é feito sobre forma variacional o que permite a implementação em malhas mais gerais. Portanto, seria um resultado novo se ele fosse apresentado com malha irregular e com termo fonte. Dessa forma, essa seria uma primeira abordagem do método GPR. Um segundo ponto que poderia ser explorado é o estudo em três dimensões do problema, aplicando-se o GPR, já que ele apresenta bons resultados em relação aos demais métodos.

# Referências

- [1] *MATLAB 7 Getting Started Guide*. MathWorks, 2010.
- [2] ADAMS, R. A., FOURNIER, J. J. F. *Sobolev Spaces*, second edition ed., vol. 140. Academic Press, New York, 2003.
- [3] BABUŠKA, I., AZIS, A. K. *The mathematical foundations of the finite element method with applications to partial differential equations*. Prentice Hall, New York, 1972.
- [4] BABUŠKA, I., IHLENBURG, F., PAIK, E. T., SAUTER, S. A. A generalized finite element method for solving the Helmholtz equation in two dimensions with minimal pollution. *Computer Methods in Applied Mechanics and Engineering* 128 (1995), 325–359.
- [5] BABUŠKA, I., IHLENBURG, F., PAIK, E. T., SAUTER, S. A. Is the pollution effect of the fem avoidable for the Helmholtz equation considering high wave numbers. *SIAM* 42 (2000), 451–484.
- [6] BRENNER, S., SCOTT, L. R. *The Mathematical Theory of Finite Element Methods*. Springer, 2007.
- [7] CARMO, E. G. D., ALVAREZ, G. B., LOULA, A. F. D., ROCHINHA, F. A. A nearly optimal Galerkin projected residual finite element method for Helmholtz problem. *Computer Methods in Applied Mechanics and Engineering* 197 (2008), 1362–1375.
- [8] CARMO, E. G. D., ALVAREZ, G. B., ROCHINHA, F. A., LOULA, A. F. D. Galerkin projected residual method applied to diffusion-reaction problems. *Computer Methods in Applied Mechanics and Engineering* 197 (2008), 4559–4570.
- [9] FERNANDES, D. T. *Métodos de Elementos Finitos e Diferenças Finitas para o Problema de Helmholtz*. PhD thesis, Laboratório Nacional de Computação Científica - LNCC, 2009.
- [10] FIGUEIREDO, D. G., NEVES, A. F. *Equações Diferenciais Aplicadas*. IMPA, Coleção Projeto Euclides, Rio de Janeiro, 2012.
- [11] GALVIS, J., VERSIEUX, H. *Introdução à Aproximação Numérica de Equações Diferenciais Parciais Via o Método de Elementos Finitos*. 28<sup>o</sup> Colóquio Brasileiro de Matemática, IMPA, 2011.
- [12] HARARI, I., FRANCA, L. P., HULBERT, G. M. Finite element methods for Helmholtz equation in an exterior domain: Model problems. *Computer Methods in Applied Mechanics and Engineering* 87 (1991), 59–96.

- 
- [13] HARARI, I., TURKEL, E. Accurate finite difference methods for time-harmonic wave propagation. *Journal of Computational Physics* 119 (1995), 252–270.
- [14] HUGHES, T. J. R., FRANCA, L. P., HULBERT, G. M. A new finite element formulation for computational fluid dynamics: VIII. the Galerkin/least-squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering* 73 (1988), 173–189.
- [15] IHLENBURG, F. *Finite Element Analysis of Acoustic Scattering*. Springer-Verlag, New York, 1998.
- [16] IHLENBURG, F., BABUŠKA, I. Finite element solution of the Helmholtz equation with high wave number part i: The h-version of the fem. *Computer and Mathematics with Applications* (34, issue 1) (1995a), 9–37.
- [17] IHLENBURG, F., BABUŠKA, I. Dispersion analysis and error estimation of Galerkin finite element methods for the Helmholtz equation. *Computer & Mathematics with Applications* 38(22) (1995b), 3745–3774.
- [18] KIUSALAAS, J. *Numerical Methods in Engineering with MATLAB*, 2 edition ed. Cambridge University Press, 2009.
- [19] KREYSZIG, E. *Introductory Functional Analysis with Applications*. Wiley, 1989.
- [20] LAX, P. D., MILGRAM, A. N. Contributions to the theory of partial differential equations. *Annals of Mathematics Studies Princeton University Press*, 33 (1954), 167–190.
- [21] LIMA, E. L. *Curso de Análise*, vol. 1. Coleção Projeto Euclides, IMPA, 1976.
- [22] LIMA, E. L. *Curso de Análise*, vol. 2. Coleção Projeto Euclides, IMPA, 1981.
- [23] LIMA, E. L. *Espaços Métricos*, 4. ed. ed. IMPA, Coleção Projeto Euclides, Rio de Janeiro, 2011.
- [24] RANGEL, J. L., CERQUEIRA, R., CELES, W. *Introdução a estruturas de dados*. Campus, 2004.
- [25] SINGER, I., TURKEL, E. High-order finite difference methods for the Helmholtz equation. *Computer Methods in Applied Mechanics and Engineering* 163 (1998), 343–358.
- [26] STRANG, G., FIX, G. J. *An Analysis of Finite Element Method*. Prentice Hall, Englewoods Cliffs, NJ, 1973.
- [27] SZWARCFITER, J. L., MARKENZON, L. *Estruturas de dados e seus algoritmos*. Livros Técnicos e Científicos, Rio de Janeiro, 1994.
- [28] THOMPSON, L. L., PINSKY, P. M. A Galerkin least squares finite element method for the two-dimensional Helmholtz equation. *International Journal for Numerical Methods in Engineering* 73 (1995), 371–397.
- [29] TIMOSHENKO, S. P., GOODIER, J. N. *Theory of Elasticity*. Mcgraw-Hill Book Company, New York, Toronto, London, 1951.

# APÊNDICE A - Elementos de Análise Funcional

## A.1 Norma e produto interno

As noções de norma e produto interno tornam-se imprescindíveis quando é necessário comparar dois elementos de um conjunto. Desse modo, as definições A.1.1 e A.1.2 mostram como a norma e o produto interno são definidos em espaços vetoriais, respectivamente [23].

**Definição A.1.1.** Seja  $V$  um espaço vetorial sobre o corpo dos complexos  $\mathbb{C}$ ,  $V$  é chamado de espaço normado se existe uma aplicação  $\|\cdot\| : V \rightarrow \mathbb{R}$ , chamada de norma, com as seguintes propriedades:

1.  $\|v\| = 0 \Rightarrow v = 0, \quad \forall v \in V,$
2.  $\|\alpha v\| = |\alpha| \|v\|, \quad \forall v \in V, \alpha \in \mathbb{C},$
3.  $\|u + v\| \leq \|u\| + \|v\|, \quad \forall u, v \in V.$

**Definição A.1.2.** Um espaço  $V$  é equipado com produto interno se existe um mapeamento  $(\cdot, \cdot) : V \times V \rightarrow \mathbb{C}$  com as propriedades:

1.  $(v, v) \geq 0, (v, v) = 0 \Rightarrow v = 0, \quad \forall v \in V$
2.  $(\alpha u + v, w) = \alpha(u, w) + (v, w), \quad \forall u, v, w \in V, \alpha \in \mathbb{C},$
3.  $(u, v) = \overline{(v, u)}, \quad \forall u, v \in V,$

onde  $(u, v)$  é um número complexo,  $\overline{(u, v)}$  é seu conjugado complexo. Se um espaço vetorial  $V$  é equipado com produto interno, pode-se induzir, neste espaço, uma norma pelo produto interno por  $\|\cdot\|_V = (\cdot, \cdot)_V^{1/2}$  e que satisfaz todas as propriedades de norma, conforme apresentado na definição A.1.1.

## A.2 Espaços de Lebesgue

Restringe-se a atenção às funções  $f$  em  $\mathbb{R}^n$ , num domínio  $\Omega$ , sendo mensuráveis à Lebesgue e que

$$\int_{\Omega} f(x) dx \quad (\text{A.1})$$

defina a integral de Lebesgue para  $f$  ( $dx$  denota medida de Lebesgue) [6]. Para  $1 \leq p < \infty$ , seja

$$\|f\|_{L^p(\Omega)} := \left( \int_{\Omega} |f(x)|^p dx \right)^{1/p} \quad (\text{A.2})$$

a norma da função  $f$ . Portanto, pode-se definir o espaço de Lebesgue [2] como

$$L^p(\Omega) := \{f : \|f\|_{L^p(\Omega)} < \infty\}. \quad (\text{A.3})$$

Neste espaço, duas funções,  $f$  e  $g$ , serão iguais quando  $\|f - g\|_{L^p(\Omega)} = 0$ . Pode-se ilustrar, com um exemplo, sendo  $n = 1$ ,  $\Omega = [-1, 1]$  e duas funções

$$f(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{se } x < 0 \end{cases} \quad \text{e} \quad g(x) = \begin{cases} 1 & \text{se } x > 0 \\ 0 & \text{se } x \leq 0. \end{cases} \quad (\text{A.4})$$

As funções diferem-se em apenas um ponto, num conjunto de medida nula mais especificamente. Entretanto, sob o olhar do espaço  $L^1(\Omega)$  essas funções são idênticas, pois  $\|f - g\|_{L^1(\Omega)} = 0$ . Pode-se ainda pensar o espaço  $L^p(\Omega)$  como um conjunto de classes de equivalência de funções, respeitando essa identificação, segundo norma definida no espaço em questão.

## A.3 Espaços de Hilbert

Duas definições preliminares necessárias para que se defina espaços de Hilbert são as de seqüências de Cauchy e de espaço completo.

**Definição A.3.1.** Uma seqüência  $\{v_n\} \subset V$ , em um espaço vetorial normado  $V$ , é dita uma seqüência de Cauchy se

$$\sup_{m, n \geq k} \|v_n - v_m\|_V \rightarrow 0 \quad \text{para } k \rightarrow \infty$$

Uma seqüência de Cauchy não é necessariamente convergente, mas a recíproca é sem-

pre verdadeira. Dessa forma, temos uma definição através do fato da sequência de Cauchy convergir ou não, que é a definição A.3.2, sendo ela necessária para a construção de espaços por completamento.

**Definição A.3.2.** Um espaço vetorial normado  $V$  é dito completo se toda sequência de Cauchy  $\{v_n\} \subset V$  converge para um elemento  $v \in V$ , isto é, se existir um elemento  $v \in V$  tal que  $\lim_{n \rightarrow \infty} \|v - v_n\|_V = 0$ .

Dessa forma, através da definição de espaço completo, pode-se definir um espaço de Hilbert.

**Definição A.3.3.** Um espaço vetorial  $V$  é chamado espaço de Hilbert se ele é equipado com um produto interno  $(\cdot, \cdot)_V$  e é completo com respeito a norma induzida  $\|\cdot\|_V$ .

Considerando o intervalo  $(0, 1) \subset \mathbb{R}$ , define-se o espaço

$$L^2(0, 1) := \left\{ f : (0, 1) \rightarrow \mathbb{C}, \int_0^1 |f(x)|^2 dx < \infty \right\} \quad (\text{A.5})$$

das funções de quadrado integrável. A operação

$$(f, g) := \int_0^1 f(x) \overline{g(x)} dx \quad (\text{A.6})$$

define um produto interno neste mesmo espaço. De fato,  $L^2(0, 1)$  define um espaço de Hilbert [19] com a norma

$$\|f\|_2 = \left( \int_0^1 |f(x)|^2 dx \right)^{1/2}. \quad (\text{A.7})$$

Portanto, tem-se a definição do espaço  $L^2(\Omega)$  e que é usado em análises de convergência dos métodos numéricos tratados neste trabalho.

## A.4 Derivadas forte e fraca

Seguindo a referência [6], as definições de derivada variam de acordo com o contexto proposto. Em cálculo infinitesimal, estudado usualmente no início de carreiras em ciências exatas, a definição de derivada é

$$u'(x) = \lim_{h \rightarrow 0} \frac{u(x+h) - u(x)}{h}$$

onde  $u'(x)$  é um resultado local da função  $u$  em torno do ponto  $x$ . Em certos espaços de funções o valor pontual não é importante, assim é normal se pensar em derivadas nas quais isso também ocorra, caso este que foi mencionado nas funções descritas em (3.5). As derivadas, no sentido fraco, fornecem um aspecto global da derivada de uma função, aparecendo naturalmente em formulações variacionais.

Introduz-se o conceito de suporte de uma função em algum domínio  $\Omega$  definido em  $\mathbb{R}^n$ . Para uma função contínua  $u$ , o suporte é dado pelo fecho (definição em [21]) do conjunto  $\{x : u(x) \neq 0\}$ . Em outras palavras, é o fecho do conjunto de pontos onde a função  $u$  não se anula. Se esse conjunto for compacto então diz-se que  $u$  tem suporte compacto com respeito a  $\Omega$ .

Duas definições são essenciais para a existência de derivadas fracas. A primeira, como já foi mencionada, é a do suporte compacto de uma função. A segunda diz respeito a uma função que é localmente integrável, excluindo a ideia de que ela seja necessariamente integrável na fronteira.

**Definição A.4.1.** Seja  $\Omega$  um domínio em  $\mathbb{R}^n$ . Denote por  $C_0^\infty(\Omega)$  o conjunto das funções em  $C^\infty(\Omega)$ , isto é, infinitamente derivável, com suporte compacto em  $\Omega$ .

**Definição A.4.2.** Dado um domínio  $\Omega$ , o conjunto das funções localmente integráveis é denotado por

$$L_{loc}^1(\Omega) := \{f : f \in L^1(K) \quad \forall K \text{ compacto} \subset \text{int}(\Omega)\}.$$

Através das duas definições anteriores pode-se definir o que seria uma derivada no sentido fraco. Embora a função  $f$  na definição a seguir possa se comportar de forma irregular na fronteira, a função  $\phi$  dá esse controle do crescimento por ser de suporte compacto.

**Definição A.4.3.** Diz-se que uma função  $f \in L_{loc}^1(\Omega)$  tem uma derivada fraca,  $\partial^i f$  se existir uma função  $g \in L_{loc}^1(\Omega)$  tal que

$$\int_{\Omega} g(x)\phi(x)dx = (-1)^i \int_{\Omega} f(x)\phi^{(i)}dx \quad \forall \phi \in C_0^\infty(\Omega)$$

e definimos  $\partial^i f = g$ , sendo também  $\phi^{(i)}$  a derivada de ordem  $i$  da função  $\phi$ .

## A.5 Espaço $H^1$

Após as definições anteriores, principalmente a de derivadas fracas, tem-se um subespaço de grande importância. Esse subespaço é definido para inteiros  $m \geq 0$  como:

$$H^m(\Omega) := \{f \in L^2(\Omega) : \partial^i f \in L^2(0,1), i = 0, 1, \dots, m\},$$

Pode-se verificar que o produto escalar de  $H^m$  é definido por

$$(f, g)_{H^m} = \sum_{i=0}^m \int_{\Omega} \partial^i f(x) \overline{\partial^i g(x)} dx,$$

com a norma induzida  $\|f\|_{H^m} = (f, f)_{H^m}^{1/2}$ . Os subespaços  $H^m(\Omega) \subset L^2(\Omega)$  são também espaços de Hilbert. E eles também são espaços de Sobolev [2], especialmente para o caso  $p = 2$  dos espaços mais gerais de Sobolev  $W^{m,p}$ . Em particular, o espaço de Sobolev  $H^0(\Omega)$  é idêntico ao  $L^2(\Omega)$ .

Para  $f \in H^m(\Omega)$ , tem-se uma seminorma como

$$|f|_m := \left( \int_{\Omega} |\partial^m f(x)|^2 dx \right)^{1/2}. \quad (\text{A.8})$$

Ressalta-se aqui que o espaço  $H^1$  pode ser equipado com a seminorma (3.9) e  $H^0 = L^2$  com a norma (3.8). As seminormas satisfazem os dois últimos itens da definição 3.1, entretanto, o primeiro item não é necessariamente satisfeito.

## A.6 Construção de espaços por completamento

Esta seção mostra uma ferramenta que permite definir espaços de funções. Inicialmente, conceitua-se um espaço normado e completo e o que vem a ser o completamento de um espaço.

**Teorema A.6.1.** Seja  $(V, \|\cdot\|)$  um espaço vetorial normado, então existe um único espaço vetorial completo  $(W, \|\cdot\|)$  tal que

- $V \subset W$ ,
- Dado qualquer elemento  $v \in W$ , existe uma sequência  $\{v_n\} \subset V$  tal que

$$\lim_{n \rightarrow \infty} v_n = v$$

Neste caso, diz-se que  $V$  é denso em  $W$ , ou ainda que  $W$  é o fecho (ou completamento) de  $V$ . É observado também que a norma de ambos os espaços,  $V$  e  $W$ , é a mesma.

O espaço  $L^2(\Omega)$  é dado pelo fecho do espaço  $C(\Omega)$  com relação à norma de  $L^2$ . Os espaços  $H^1(\Omega)$  e  $H_0^1(\Omega)$  são definidos, respectivamente, como o completamento dos espaços  $C^\infty(\Omega)$  e  $C_0^\infty(\Omega)$  com relação à norma  $\|\cdot\|_{H^1}$ . O teorema e a definição anterior encontram-se em [11].

## A.7 Formas sesquilineares e operadores lineares

Pode-se escrever a formulação variacional (fraca) de um problema de valor de contorno da seguinte forma [15]:

$$\begin{cases} \text{Encontrar } u \in V_1 : \\ b(u, v) = f(v), \quad \forall v \in V_2, \end{cases} \quad (\text{A.9})$$

onde  $V_1$  e  $V_2$  são espaços vetoriais normados (espaço das funções teste),  $b$  é uma forma bilinear (ou sesquilinear)  $V_1 \times V_2 \rightarrow \mathbb{C}$ . O mapeamento  $b(\cdot, \cdot)$  é chamado de forma bilinear se os argumentos forem lineares um a um. Se a forma  $b(\cdot, \cdot)$  é linear no primeiro e anti-linear no segundo argumento, isto é

$$\begin{aligned} b(\alpha(u_1 + u_2), v) &= \alpha(b(u_1, v) + b(u_2, v)), \\ b(u, \alpha(v_1 + v_2)) &= \bar{\alpha}(b(u, v_1) + b(u, v_2)), \end{aligned}$$

então é chamado de forma sesquilinear. Uma forma sesquilinear  $b : V_1 \times V_2 \rightarrow \mathbb{C}$  é chamada limitada se existir uma constante  $M \in \mathbb{R}$  tal que

$$|b(u, v)| \leq M \|u\|_{V_1} \|v\|_{V_2}$$

para todo  $\{u, v\} \in V_1 \times V_2$ .

Considerando-se  $V_1, V_2$  espaços vetoriais normados, um mapeamento  $A : V_1 \rightarrow V_2$  é chamado operador linear se

$$A(\alpha u + \beta v) = \alpha A u + \beta A v, \quad \forall u, v \in V_1, \quad \alpha, \beta \in \mathbb{C}.$$

Os operadores lineares  $V_1 \rightarrow V_2$  formam o espaço vetorial  $\mathcal{L}(V_1, V_2)$ . O operador  $A$  é

limitado se existir uma constante  $M \in \mathbb{R}$  tal que

$$\|A(u)\|_{V_2} \leq M\|u\|_{V_1}$$

para todo  $u \in V_1$  [15].

## A.8 Existência e unicidade de soluções

Serão apresentados, brevemente, dois teoremas que garantem a existência e unicidade do problema variacional para certas condições de contorno. As demonstrações dos teoremas estão nas referências [19] e [3]. O teorema de Babuška é uma versão generalizada do Lax-Milgram, já que ele assume uma maior variedade de espaços de funções nos quais a função está definida.

O primeiro teorema a ser enunciado foi formulado por Peter Lax, matemático húngaro, e Arthur Norton Milgram, matemático norte-americano. A primeira publicação, ou a original, é datada de 1954 em um jornal de matemática da universidade de Princeton, que segue na referência [20].

**Teorema 3.1.(Lax-Milgram).** Assumindo a forma sesquilinear  $a : V \times V \rightarrow \mathbb{C}$ , definida em um espaço de Hilbert  $V$ , satisfazendo

1. Continuidade:

$$\exists M > 0 : |a(u, v)| \leq M\|u\|_V\|v\|_V, \forall u, v \in V.$$

2.  $V$ -Elipticidade:

$$\exists \alpha > 0 : \alpha\|u\|_V^2 \leq |a(u, u)|, \forall u \in V.$$

e seja  $f$  um funcional linear definido e limitado em  $V$ . Então deve existir um único elemento  $u_0 \in V$  tal que

$$a(v, u_0) = f(v), \quad \forall v \in V$$

Nota-se que os espaços de funções de  $u$  e  $v$  são idênticos, mas a princípio nada impede de buscar soluções em outros espaços. Além disso, não se sabe se para quaisquer condições de contorno o problema é resolvido.

**Teorema 3.2.(Babuška).** Seja a forma sesquilinear  $b : V_1 \times V_2 \rightarrow \mathbb{C}$  em espaços de Hilbert  $V_1$  e  $V_2$  satisfazendo

1. Continuidade:

$$\exists M > 0 : |b(u, v)| \leq M \|u\|_{V_1} \|v\|_{V_2}, \quad \forall u \in V_1, \forall v \in V_2.$$

2. Condição de inf-sup:

$$\exists \beta > 0 : \beta \leq \sup_{0 \neq v \in V_2} \frac{|b(u, v)|}{\|u\|_{V_1} \|v\|_{V_2}}, \quad \forall 0 \neq u \in V_1$$

3. Condição inf-sup Transposta:

$$\sup_{0 \neq v \in V_2} |b(u, v)| > 0, \quad \forall 0 \neq u \in V_1$$

e seja  $f : V_2 \rightarrow \mathbb{C}$  um funcional antilinear definido e limitado em  $V_2$ . Então deve existir um único elemento  $u_0 \in V_1$  tal que

$$b(u_0, v) = f(v), \quad \forall v \in V_2.$$

A solução  $u_0$  satisfaz a limitação

$$\|u_0\|_{V_1} \leq \frac{1}{\beta} \|f\|_{V_2^*},$$

considerando-se  $V^*$  como o espaço dos funcionais antilineares com a propriedade  $f(\alpha u) = \bar{\alpha} f(u)$  [15].

## APÊNDICE B - Códigos Computacionais

### B.1 Códigos dos métodos em uma dimensão

Código em Matlab para o método de diferenças finitas centradas de segunda ordem para o problema de Helmholtz 1D:

```

1  clc;clear all;close all;
2  %% -----Parametros de Entrada-----
3  k = 20;                % Numero de Onda
4  Xa = 0;  Xb = 1;      % Dominio [Xa,Xb]
5  N = 100;              % Numero de nos
6  h = (Xb-Xa)/(N-1);    % Distancia entre os nos
7  ContA = 0; ContB = -3; % Condiçao de Contorno Dirichlet
8  Diagonal = (h*k)^2-2; % Coeficientes da Diagonal Principal
9  %% -----Inicializacoes-----
10 K = spalloc(N,N,3*N); % Inicializacao da Matriz K
11 f = zeros(N,1); % Inicializacao do vetor f
12 u = zeros(N,1); % Inicializacao do vetor solucao u
13 %% -----Loop de atribuicao das Diagonais-----
14 for ii = 1:N-1
15     K(ii,ii) = Diagonal;          % Diagonal Principal
16     K(ii+1,ii) = 1;              % Diagonal Inferior
17     K(ii,ii+1) = 1;              % Diagonal Superior
18     f(ii,1) = -k^2*(ContA + (ii-1)*h); % Termo Fonte
19 end
20 f(N,1) = -k^2*(ContA + (ii-1)*h);
21 K(N,N)=Diagonal;
22 %% -----Condiçoes de Contorno no vetor solucao u-----
23 u(1) = ContA;
24 u(N) = ContB;
25 %% -----Resolucao do Sistema Linear-----
26 f_d = h^2*f - K*u; % Novo vetor f pelas condiçoes de contorno

```

```

27 int = 2:N-1;          % Indices do interior de K e f
28 K_int = K(int,int); % K interior
29 f_int = f_d(int);    % b interior
30 u_int = K_int\f_int; % Sistema K_int*u_aux = f_int
31 u_sol = u;           % Solucao do contorno
32 u_sol(int) = u_int; % Solucao com inclusao dos pontos interiores
33 %% -----Grafico Aproximada/Interpolante-----
34 xx = Xa:h:Xb;        % Dominio
35 yy=-2/(cos(k-pi/2)).*cos(k*xx-pi/2) - xx; % Solucao Exata com fonte
36 plot(xx,u_sol,'ro-',xx,yy,'bx-')
37 legend('MDFC','Interpolante');

```

Código em Matlab para o método de elementos finitos de Galerkin para o problema de Helmholtz 1D:

```

1 clc;close all;clear all;
2 %% -----Parametros de Entrada-----
3 Nel = 100;          % Numero de Elementos no Dominio
4 k = 20;            % Numero de Onda
5 Xa = 0; Xb = 1;    % Dominio [Xa,Xb]
6 ContA = 0; ContB = -3; % Condiacao de Contorno Dirichlet
7 %% -----PESOS DA QUADRATURA DE GAUSS [-1,1]; n = 3-----
8 omega_aux(1) = 5/9;
9 omega_aux(2) = 8/9;
10 omega_aux(3) = 5/9;
11 %% -----PONTOS DA QUADRATURA [-1,1]; n = 3 -----
12 p_aux(1) = -sqrt(15)/5;
13 p_aux(2) = 0;
14 p_aux(3) = sqrt(15)/5;
15 %% ----- Inicializacoes -----
16 x = linspace(Xa,Xb,Nel+1); % Posicao dos nos
17 H(1,9).xini=[];H(1,9).xfin=[];H(1,9).omega=[];H(1,9).x_q=[];
18 H(1,9).phi1=[];H(1,9).phi2=[];H(1,9).dphi1=[];H(1,9).dphi2=[];
19 H(1,9).dof=[];H(1,9).Klocal=[];H(1,9).flocal=[];
20 K = zeros(Nel+1,Nel+1); % Inicializacao da matriz K
21 f = zeros(Nel+1,1);     % Inicializacao do vetor f
22 u = zeros(Nel,1);      % Inicializacao do vetor solucao u
23 %% ----- Atribuicoes para cada elemento -----
24 for i=1:Nel
25 xini = x(i);           % Posicao inicial do elemento i
26 xfin = x(i+1);        % Posicao final do elemento i
27 H(i).xini = xini; % Atribuindo posicao do no inicial

```

```

28 H(i).xfin = xfin; % Atribuindo posicao do no final
29
30 omega = 0.5*(xfin-xini)*omega_aux; % Peso da Quadratura no no
31 x_q = xini + 0.5*(1 +p_aux)*(xfin-xini); % Ponto da Quadratura no no
32 H(i).omega = omega; % Atribuindo Peso da Quadratura
33 H(i).x_q = x_q; % Atribuindo Ponto da Quadratura
34
35 H(i).phi1 = (xfin-x_q)/(xfin-xini); % Funcao Base phi(i)
36 H(i).phi2 = (x_q-xini)/(xfin-xini); % Funcao Base phi(i+1)
37 H(i).dphi1 = [-1,-1,-1]/(xfin-xini); % Derivada da Funcao Base phi(i)
38 H(i).dphi2 = [1,1,1]/(xfin-xini); % Derivada da Funcao Base phi(i+1)
39 H(i).dof = [i,i+1]; % Indice dos nos em cada elemento
40 end
41 %% -----Calculo Matrices Locais-----
42 for i=1:Nel
43 phi1 = H(i).phi1; % Base phi(i)
44 phi2 = H(i).phi2; % Base phi(i+1)
45 dphi1 = H(i).dphi1; % Derivada da Funcao Base phi(i)
46 dphi2 = H(i).dphi2; % Derivada da Funcao Base phi(i+1)
47 w = H(i).omega; % Peso da Quadratura
48 x_q = H(i).x_q; % Pontos da Quadratura
49 %-----Matriz Local
50 k11 = sum(w.*dphi1.*dphi1-k^2*w.*phi1.*phi1); % Elemento K(1,1)
51 k12 = sum(w.*dphi1.*dphi2-k^2*w.*phi1.*phi2); % Elemento K(1,2)
52 k21 = sum(w.*dphi2.*dphi1-k^2*w.*phi2.*phi1); % Elemento K(2,1)
53 k22 = sum(w.*dphi2.*dphi2-k^2*w.*phi2.*phi2); % Elemento K(2,2)
54 %-----Termo Fonte
55 func=k^2*x_q;
56 f1 = sum(w.*func.*phi1);
57 f2 = sum(w.*func.*phi2);
58 %-----Atribuicoes
59 H(i).Klocal=[k11,k12;k21,k22]; % Atribuindo Matriz K Local
60 H(i).flocal = [f1;f2];
61 end
62 %% -----ASSEMBLY: LOCAL->GLOBAL-----
63 for i=1:Nel
64 Klocal = H(i).Klocal;
65 flocal = H(i).flocal;
66 dof = H(i).dof;
67 for ii=1:2
68 for jj=1:2
69 K(dof(ii),dof(jj)) = K(dof(ii),dof(jj))+Klocal(ii,jj);
70 end

```

```

71     f(dof(ii),1)=f(dof(ii),1)+flocal(ii);
72     end
73 end
74 %% -----Condições de Contorno no vetor solução u-----
75 u(1)     = ContA;
76 u(Nel+1)= ContB;
77 %% -----Resolução do Sistema Linear-----
78 b_d = f-K*u;           % Novo vetor f pelas condições de contorno
79 int = 2:Nel;          % Índices do interior de K e f
80 K_int = K(int,int);  % K interior
81 b_int = b_d(int);    % b interior
82 x_aux = K_int\b_int; % Sistema K_int*u_aux = f_int
83 x_sol = u;           % Solução do contorno
84 x_sol(int) = x_aux; % [Solução] com inclusão dos pontos interiores
85 %% -----Gráfico Aproximada/Interpolante-----
86 xx = linspace(Xa,Xb,Nel+1);           % Domínio
87 yy = -2/(cos(k-pi/2)).*cos(k*xx-pi/2) - xx; % Solução Exata
88 plot(xx,x_sol,'ro-',xx,yy,'bx-')
89 legend('Galerkin','Interpolante');

```

Código em Matlab para o método de elementos finitos de Galerkin mínimos quadrados (GLS) para o problema de Helmholtz 1D:

```

1  clc;close all;clear all;
2  %% -----Parâmetros de Entrada-----
3  Nel = 150;           % Número de Elementos no Domínio
4  k = 80;              % Número de Onda
5  h = 1/Nel;
6  Xa = 0; Xb = 1;     % Domínio [Xa,Xb]
7  ContA = 0; ContB = -3; % Condição de Contorno Dirichlet
8  %-----Parâmetro de Estabilização do GLS
9  tau = 1/k^2*(1 - 6*(1-cos(k*h))/(k^2*h^2*(2+cos(k*h))));
10 %% -----PESOS DA QUADRATURA DE GAUSS [-1,1]; n = 3-----
11 omega_aux(1) = 5/9;
12 omega_aux(2) = 8/9;
13 omega_aux(3) = 5/9;
14 %% -----PONTOS DA QUADRATURA [-1,1]; n = 3 -----
15 p_aux(1) = -sqrt(15)/5;
16 p_aux(2) = 0;
17 p_aux(3) = sqrt(15)/5;
18 %% -----Inicializações-----
19 H(1,9).xini=[];H(1,9).xfim=[];H(1,9).omega=[];H(1,9).x_q=[];

```

```

20 H(1,9).phi1=[];H(1,9).phi2=[];H(1,9).dphi1=[];H(1,9).dphi2=[];
21 H(1,9).dof=[];H(1,9).Klocal=[];H(1,9).flocal=[];
22 K = zeros(Nel+1,Nel+1); % Inicializacao da matriz K
23 f = zeros(Nel+1,1); % Inicializacao do vetor f
24 u = zeros(Nel,1); % Inicializacao do vetor solucao u
25 %% ----- Atribuicoes para cada elemento-----
26 for i=1:Nel
27 xini = (i-1)/Nel; %Posicao inicial do elemento i
28 xfin = i/Nel; %Posicao final do elemento i
29
30 H(i).xini = xini; %Atribuindo posicao do no inicial
31 H(i).xfin = xfin; %Atribuindo posicao do no final
32
33 omega = 0.5*(xfin-xini)*omega_aux; %Peso da Quadratura no no
34 x_q = xini + 0.5*(1 +p_aux)*(xfin-xini); %Ponto da Quadratura no no
35 H(i).omega = omega; %Atribuindo Peso da Quadratura
36 H(i).x_q = x_q; %Atribuindo Ponto da Quadratura
37
38 H(i).phi1 = (xfin-x_q)/(xfin-xini); %Funcao Base phi(i)
39 H(i).phi2 = (x_q-xini)/(xfin-xini); %Funcao Base phi(i+1)
40 H(i).dphi1 = [-1,-1,-1]/(xfin-xini); %Derivada da Funcao Base phi(i)
41 H(i).dphi2 = [1,1,1]/(xfin-xini); %Derivada da Funcao Base phi(i+1)
42 H(i).dof = [i,i+1]; %Indice do no em cada elemento
43 end
44 %% -----Calculo Matrices Locais-----
45 for i=1:Nel
46 phi1 = H(i).phi1; %Base phi(i)
47 phi2 = H(i).phi2; %Base phi(i+1)
48 dphi1 = H(i).dphi1; %Derivada da Funcao Base phi(i)
49 dphi2 = H(i).dphi2; %Derivada da Funcao Base phi(i+1)
50 w = H(i).omega; %Peso da Quadratura
51 x_q = H(i).x_q; %Pontos da Quadratura
52 %-----Matriz Local
53 k11 = sum(w.*dphi1.*dphi1-k^2*w.*phi1.*phi1 + tau*k^4*w.*phi1.*phi1);
54 k12 = sum(w.*dphi1.*dphi2-k^2*w.*phi1.*phi2 + tau*k^4*w.*phi1.*phi2);
55 k21 = sum(w.*dphi2.*dphi1-k^2*w.*phi2.*phi1 + tau*k^4*w.*phi2.*phi1);
56 k22 = sum(w.*dphi2.*dphi2-k^2*w.*phi2.*phi2 + tau*k^4*w.*phi2.*phi2);
57 %-----Termo Fonte
58 func=k^2*x_q;
59 f1 = sum(w.*func.*phi1 - tau*k^2*w.*phi1);
60 f2 = sum(w.*func.*phi2 - tau*k^2*w.*phi1);
61 %-----Atribuicoes
62 H(i).Klocal=[k11,k12;k21,k22]; %Atribuindo Matriz K Local

```

```

63 H(i).flocal = [f1;f2];
64
65 end
66 %% -----ASSEMBLY: LOCAL->GLOBAL-----
67 for i=1:Nel
68 Klocal = H(i).Klocal;
69 flocal = H(i).flocal;
70 dof = H(i).dof;
71     for ii=1:2
72         for jj=1:2
73             K(dof(ii),dof(jj)) = K(dof(ii),dof(jj))+Klocal(ii,jj);
74         end
75     f(dof(ii),1)=f(dof(ii),1)+flocal(ii);
76     end
77 end
78 %% -----Condiçoes de Contorno no vetor solucao u-----
79 u(1)      = ContA;
80 u(Nel+1)= ContB;
81 %% -----Resolucao do Sistema Linear-----
82 b_d = f-K*u;          % Novo vetor f pelas condiçoes de contorno
83 int = 2:Nel;         % Indices do interior de K e f
84 K_int = K(int,int); % K interior
85 b_int = b_d(int);   % b interior
86 x_aux = K_int\b_int; % Sistema K_int*u_aux = f_int
87 x_sol = u;          % Solucao do contorno
88 x_sol(int) = x_aux; % [Solucao] com inclusao dos pontos interiores
89 %% -----Grafico Aproximada/Interpolante-----
90 xx = linspace(Xa,Xb,Nel+1);          % Dominio
91 yy = -2/(cos(k-pi/2)).*cos(k*xx-pi/2) - xx; % Solucao Exata
92 plot(xx,x_sol,'ro-',xx,yy,'bx-')
93 legend('GLS','Interpolante');

```

## B.2 Códigos dos métodos em duas dimensões

Código em Matlab para o método de diferenças finitas centradas de segunda ordem para o problema de Helmholtz 2D:

```

1 clc;close all;clear all;
2 %% -----Parametros de Entrada-----
3 teta = [0;pi/8;3*(pi/16)]; % Angulos para direcao de ondas

```

```

4  kn = 10; % Numero de onda
5  %——Eixo x:
6  npx = 100; % Numero de pontos em x
7  Ax = 0; % Dominio em Ox = [Ax,Bx]
8  Bx = 1;
9  hx = (Bx - Ax)/(npx - 1); % Distancia entre nos em x
10 %——Eixo y:
11 npy = 100; % Numero de pontos em x
12 Ay = 0; % Dominio em Oy=[Ay,By]
13 By = 1;
14 hy = (By - Ay)/(npy - 1); % Distancia entre nos em y
15 %——Malha:
16 npt = npx*npy; % Numero total de pontos
17 npxi=npx-2; npyi=npy-2; % Numero de pontos internos em x ...
   e y
18 npti = (npxi)*(npyi); % Numero de pontos internos na malha
19 [X,Y] = meshgrid(Ax:hx:Bx,Ay:hy:By); % Malha das solucoes
20 %% ————— Inicializacoes —————
21 K = spalloc(npti,npti,5*npti); % Inicializacao da Matriz K
22 b = zeros((npxi)*(npyi),1); % Inicializacao do vetor b
23 sol_exata = zeros(npx,npy); % Inicializacao da solucao exata
24 %% —— Diagonais
25 Dp = -2*(hx^2 + hy^2)+ hx^2*hy^2*kn^2;
26 Ds = hy^2;
27 Di = hx^2;
28 Dsa = hx^2;
29 Dia = hy^2;
30 %% Diagonal Principal
31 for ii = 1:npti
32     K(ii,ii) = Dp;
33 end
34 %% Diagonal Superior
35 for ii = 1:npti-1
36     K(ii,ii+1) = Ds;
37 end
38 %% Diagonal Inferior
39 for ii = 2:npti
40     K(ii,ii-1) = Di;
41 end
42 %% Diagonal Superior Afastada
43 for ii = 1:(npy-3)*(npx-2)
44     K(ii,ii + npx - 2) = Dsa;
45 end

```

```

46 %% Diagonal Inferior Afastada
47 for ii = npx - 1:npti
48     K(ii,ii - npx + 2) = Dia;
49 end
50 %% -----Atribuicao das condicoes de Contorno-----
51 kk=0;
52 for jj=2:npv-1
53     for ii=2:npx-1
54         kk = kk+1;
55         %% -----Termo Fonte:
56         fonte = -1;
57         b(kk)=hx^2*hy^2*fonte;
58         for p=1:length(teta) % Loop para p ondas
59             %% -----Contorno Lateral Esquerdo:
60             x=Ax;
61             y=Ay+(jj-1)*hy;
62
63             if(ii==2)
64                 if(jj>2)
65                     K(kk, kk-1)=0;
66                 end
67                 b(kk)=b(kk)-hy^2*(cos(kn*(x*cos(teta(p))+y*sin(teta(p))))));
68             end
69             %% -----Contorno Lateral Direito:
70             x=Bx;
71             y=Ay+(jj-1)*hy;
72
73             if(ii==(npx-1))
74                 if(jj<(npv-1))
75                     K(kk, kk+1)=0;
76                 end
77                 b(kk)=b(kk)-hy^2*(cos(kn*(x*cos(teta(p))+y*sin(teta(p))))));
78             end
79             %% -----Contorno Inferior
80             x=Ax+(ii-1)*hx;
81             y=Ay;
82             if(jj==2)
83                 b(kk)=b(kk)-hx^2*(cos(kn*(x*cos(teta(p))+y*sin(teta(p))))));
84             end
85             %% -----Contorno Superior
86             x=Ax+(ii-1)*hx;
87             y=By;
88             if(jj==(npv-1))

```

```

89         b(kk)=b(kk)-hx^2*(cos(kn*(x*cos(teta(p))+y*sin(teta(p)))));
90     end
91 end
92 end
93 end
94 %% -----Resolucao do Sistema Linear-----
95 u=K\b;
96 %% -----Solucao Exata-----
97 for p=1:length(teta)
98     sol_exata = sol_exata + cos(kn*(X.*cos(teta(p))+Y.*sin(teta(p)))));
99 end
100 figure(1)
101 surf(X,Y,sol_exata)%::Solucao Exata
102 %% -----Solucao Aproximada-----
103 sol_aprox = sol_exata; %Para Inicializar e solucao no contorno
104 %-----Solucao no Interior:
105 for lin=1:npyi
106     for col=1:npxi
107         sol_aprox(lin+1,col+1) = u((lin-1)*npxi+col);
108     end
109 end
110 figure(2)
111 surf(X,Y,sol_aprox)%::Solucao Aproximada

```

Código em Matlab para o método de elementos finitos de Galerkin para o problema de Helmholtz 2D:

```

1 clear all;clc;close all;
2 %% -----Parametros de Entrada-----
3 teta=[0;pi/8;3*(pi/8)];
4 kn = 10; %Numero de Onda
5 %-----Eixo x
6 Nelx = 100; %Numero de Elementos em X e Y
7 Ax=0;
8 Bx=1;
9 hx=(Bx - Ax)/(Nelx);
10 %-----Eixo y
11 Nely = 90;
12 Ay=0;
13 By=1;
14 hy = (By - Ay)/(Nely);
15 %-----Malha do dominio: [Ax,Bx]x[Ay,By]

```

```

16 [X,Y] = meshgrid(Ax:hx:Bx,Ay:hy:By);
17 %% -----PESOS DA QUADRATURA DE GAUSS [-1,1]; n = 3-----
18 omega_aux(1) = 5/9;
19 omega_aux(2) = 8/9;
20 omega_aux(3) = 5/9;
21 %% -----PONTOS DA QUADRATURA [-1,1]; n = 3-----
22 p_aux(1) = -sqrt(15)/5;
23 p_aux(2) = 0;
24 p_aux(3) = sqrt(15)/5;
25 %% ----- Inicializacoes-----
26 pp=0; % Indice para o Loop      %%% Mapeamento dos nos Globais
27 H(1,900).xini=[];H(1,900).xfin=[];
28 H(1,900).yini=[];H(1,900).yfin=[];
29 H(1,900).omega=[];H(1,900).x_q=[];
30 H(1,900).omegax=[];H(1,900).omegay=[];
31 H(1,900).y_q=[];
32 H(1,900).phi1=[];H(1,900).phi2=[];
33 H(1,900).phi3=[];H(1,900).phi4=[];
34 H(1,900).dphi1x=[];H(1,900).dphi2x=[];
35 H(1,900).dphi3x=[];H(1,900).dphi4x=[];
36 H(1,900).dphi1y=[];H(1,900).dphi2y=[];
37 H(1,900).dphi3y=[];H(1,900).dphi4y=[];
38 H(1,900).dof=[];H(1,900).Klocal=[];
39 H(1,900).flocal=[];
40 u_b = sparse(Nelx+1,Nely+1);
41 K = sparse((Nelx+1)*(Nely+1),(Nelx+1)*(Nely+1)); %Inicializacao da ...
      matriz K
42 f = zeros((Nelx+1)*(Nely+1),1); %Inicializacao do vetor f
43 sol_exata = zeros(Nely+1,Nelx+1);
44 %% -----Atribuicoes de cada elemento-----
45 for kk=1:Nely
46     for ii=1:Nelx
47 %-----Coordenadas dos elementos:
48         xini = (ii-1)/Nelx; % Posicao Inicial em x
49         xfin = ii/Nelx;     % Posicao Final em x
50         yini = (kk-1)/Nely; % Posicao Inicial em y
51         yfin = (kk)/Nely;   % Posicao Final em y
52 %-----Atribuicao das coordenadas:
53         H(ii+pp).xini = xini;
54         H(ii+pp).xfin = xfin;
55         H(ii+pp).yini = yini;
56         H(ii+pp).yfin = yfin;
57 %-----Definicoes da quadratura de Gauss:

```

```

58     omegax = 0.5*(xfin-xini)*omega_aux; %Peso da Quadratura no X
59     omegay = 0.5*(yfin-yini)*omega_aux; %Peso da Quadratura no Y
60     x_q = xini + 0.5*(1 +p_aux)*(xfin-xini); %Ponto da Quadratura no ...
        no(x)
61     y_q = yini + 0.5*(1 +p_aux)*(yfin-yini); %Ponto da Quadratura no ...
        no(y)
62 %——Atribuicoes da quadratura de Gauss:
63     H(ii+pp).omegax = omegax;
64     H(ii+pp).omegay = omegay;
65     H(ii+pp).x_q = x_q;
66     H(ii+pp).y_q = y_q;
67 %——Vetores para quadratura:
68     e1 = [1 1 1];
69     e2 = [1;1;1];
70 %——Funcoes Base:
71     phi1 = (x_q'-xfin)*(y_q-yfin)/((xini-xfin)*(yini-yfin));
72     phi2 = (x_q'-xini)*(y_q-yfin)/((xfin-xini)*(yini-yfin));
73     phi3 = (x_q'-xini)*(y_q-yini)/((xfin-xini)*(yfin-yini));
74     phi4 = (x_q'-xfin)*(y_q-yini)/((xini-xfin)*(yfin-yini));
75 %——Derivadas em x das funcoes base:
76     dphi1x = (e2*y_q-yfin)/((xini-xfin)*(yini-yfin));
77     dphi2x = (e2*y_q-yfin)/((xfin-xini)*(yini-yfin));
78     dphi3x = (e2*y_q-yini)/((xfin-xini)*(yfin-yini));
79     dphi4x = (e2*y_q-yini)/((xini-xfin)*(yfin-yini));
80 %——Derivadas em y das funcoes base:
81     dphi1y = (x_q'*e1-xfin)/((xini-xfin)*(yini-yfin));
82     dphi2y = (x_q'*e1-xini)/((xfin-xini)*(yini-yfin));
83     dphi3y = (x_q'*e1-xini)/((xfin-xini)*(yfin-yini));
84     dphi4y = (x_q'*e1-xfin)/((xini-xfin)*(yfin-yini));
85 %——Atribuicoes das funcoes base:
86     H(ii+pp).phi1 = phi1;
87     H(ii+pp).phi2 = phi2;
88     H(ii+pp).phi3 = phi3;
89     H(ii+pp).phi4 = phi4;
90 %——Atribuicoes das derivadas em x das funcoes base:
91     H(ii+pp).dphi1x = dphi1x;
92     H(ii+pp).dphi2x = dphi2x;
93     H(ii+pp).dphi3x = dphi3x;
94     H(ii+pp).dphi4x = dphi4x;
95 %——Atribuicoes das derivadas em y das funcoes base:
96     H(ii+pp).dphi1y = dphi1y;
97     H(ii+pp).dphi2y = dphi2y;
98     H(ii+pp).dphi3y = dphi3y;

```

```

99     H(ii+pp).dphi4y = dphi4y;
100  %—Mapeamento das posicoes globais de cada no:
101     H(ii+pp).dof = [(1+Nelx)*(kk-1)+ii, (1+Nelx)*(kk-1)+ii+1, ...
102                    (1+Nelx)*(kk-1)+ii+Nelx+2, (1+Nelx)*(kk-1)+ii+Nelx+1];
103  end
104  pp=pp+Nelx;
105  end
106  %% —————MATRIZES LOCAIS—————
107  for i=1:Nelx*Nely
108     Klocal=zeros(4,4); % Iniciando Matriz de Rigidez
109  %— Funcoes base, suas derivadas e pesos da quadratura para o elemento:
110     phi1 = H(i).phi1; dphi1x = H(i).dphi1x; dphi1y = H(i).dphi1y;
111     phi2 = H(i).phi2; dphi2x = H(i).dphi2x; dphi2y = H(i).dphi2y;
112     phi3 = H(i).phi3; dphi3x = H(i).dphi3x; dphi3y = H(i).dphi3y;
113     phi4 = H(i).phi4; dphi4x = H(i).dphi4x; dphi4y = H(i).dphi4y;
114     wx = H(i).omegax; wy = H(i).omegay;
115  %— Matriz que carrega as combinacoes de peso da Quadratura:
116     Aw = omegax'*omegay;
117  %—Elementos da Matriz K Local para cada Elemento:
118     k11 = sum(sum(Aw.*(dphi1x.*dphi1x + dphi1y.*dphi1y - kn^2*phi1.*phi1)));
119     k12 = sum(sum(Aw.*(dphi1x.*dphi2x + dphi1y.*dphi2y - kn^2*phi1.*phi2)));
120     k13 = sum(sum(Aw.*(dphi1x.*dphi3x + dphi1y.*dphi3y - kn^2*phi1.*phi3)));
121     k14 = sum(sum(Aw.*(dphi1x.*dphi4x + dphi1y.*dphi4y - kn^2*phi1.*phi4)));
122     k21 = sum(sum(Aw.*(dphi2x.*dphi1x + dphi2y.*dphi1y - kn^2*phi2.*phi1)));
123     k22 = sum(sum(Aw.*(dphi2x.*dphi2x + dphi2y.*dphi2y - kn^2*phi2.*phi2)));
124     k23 = sum(sum(Aw.*(dphi2x.*dphi3x + dphi2y.*dphi3y - kn^2*phi2.*phi3)));
125     k24 = sum(sum(Aw.*(dphi2x.*dphi4x + dphi2y.*dphi4y - kn^2*phi2.*phi4)));
126     k31 = sum(sum(Aw.*(dphi3x.*dphi1x + dphi3y.*dphi1y - kn^2*phi3.*phi1)));
127     k32 = sum(sum(Aw.*(dphi3x.*dphi2x + dphi3y.*dphi2y - kn^2*phi3.*phi2)));
128     k33 = sum(sum(Aw.*(dphi3x.*dphi3x + dphi3y.*dphi3y - kn^2*phi3.*phi3)));
129     k34 = sum(sum(Aw.*(dphi3x.*dphi4x + dphi3y.*dphi4y - kn^2*phi3.*phi4)));
130     k41 = sum(sum(Aw.*(dphi4x.*dphi1x + dphi4y.*dphi1y - kn^2*phi4.*phi1)));
131     k42 = sum(sum(Aw.*(dphi4x.*dphi2x + dphi4y.*dphi2y - kn^2*phi4.*phi2)));
132     k43 = sum(sum(Aw.*(dphi4x.*dphi3x + dphi4y.*dphi3y - kn^2*phi4.*phi3)));
133     k44 = sum(sum(Aw.*(dphi4x.*dphi4x + dphi4y.*dphi4y - kn^2*phi4.*phi4)));
134  %—Matriz f local
135     f1 = 0;
136     f2 = 0;
137  %— Atribuicao da Matriz Local K e f:
138     H(i).Klocal = [k11,k12,k13,k14;k21,k22,k23,k24; ...
139                    k31,k32,k33,k34;k41,k42,k43,k44];
140     H(i).flocal = [f1;f2];
141  end

```

```

142 %% CONDICAOES DE CONTORNO
143 ind=1;indd=1;
144 ua=zeros((Nelx+1)*(Nely+1),1);
145 for ii=1:Nely+1
146     for jj=1:Nelx+1
147         %-----Condicao Inferior
148         if(ii==1)
149             x=Ax+(jj-1)*hx;y=Ay;
150             for p=1:length(teta)
151                 ua((ii-1)*(Nelx+1)+jj) = ua((ii-1)*(Nelx+1)+jj)+...
152                     cos(kn*(x*cos(teta(p))+y*sin(teta(p))));
153             end
154             conj(ind) = (ii-1)*(Nelx+1)+jj;
155             ind=ind+1;
156         %-----Condicao Superior
157         elseif(ii==Nely+1)
158             x=Ax+(jj-1)*hx;y=By;
159             for p=1:length(teta)
160                 ua((ii-1)*(Nelx+1)+jj) = ua((ii-1)*(Nelx+1)+jj)+...
161                     cos(kn*(x*cos(teta(p))+y*sin(teta(p))));
162             end
163             conj(ind) = (ii-1)*(Nelx+1)+jj;
164             ind=ind+1;
165         %-----Condicao Esquerda
166         elseif(jj==1)
167             x=Ax;y=Ay+(ii-1)*hy;
168             for p=1:length(teta)
169                 ua((ii-1)*(Nelx+1)+jj) = ua((ii-1)*(Nelx+1)+jj)+...
170                     cos(kn*(x*cos(teta(p))+y*sin(teta(p))));
171             end
172             conj(ind) = (ii-1)*(Nelx+1)+jj;
173             ind=ind+1;
174         %-----Condicao Direita
175         elseif(jj==Nelx+1)
176             x=Bx;y=Ay+(ii-1)*hy;
177             for p=1:length(teta)
178                 ua((ii-1)*(Nelx+1)+jj) = ua((ii-1)*(Nelx+1)+jj)+...
179                     cos(kn*(x*cos(teta(p))+y*sin(teta(p))));
180             end
181             conj(ind) = (ii-1)*(Nelx+1)+jj;
182             ind=ind+1;
183         end
184         %-----Interior

```

```

185         if(ii≠Nely+1&&ii≠1&&jj≠Nelx+1&&jj≠1)
186             interior(indd)= (ii-1)*(Nelx+1)+jj;
187             indd=indd+1;
188         end
189     end
190 end
191 %% -----Loops do Assembly-----
192 for i=1:Nelx*Nely
193     Klocal = H(i).Klocal;
194     flocal = H(i).flocal;
195     dof = H(i).dof;
196     for ii=1:4
197         for jj=1:4
198             K(dof(ii),dof(jj)) = K(dof(ii),dof(jj))+Klocal(ii,jj);
199         end
200     end
201     f(i,1)=0;
202 end
203 %% -----Resolucao do Sistema Linear-----
204 b_d = f-K*ua;
205 K_int = K(interior,interior);
206 b_int = b_d(interior);
207 x_aux = K_int\b_int;
208 %% -----Solucao Exata-----
209 for p=1:length(teta)
210     sol_exata = sol_exata + cos(kn*(X.*cos(teta(p))+Y.*sin(teta(p))));
211 end
212 figure(1)
213 surf(X,Y,sol_exata)%::Solucao Exata
214 %% -----Solucao Aproximada-----
215 sol_aprox = sol_exata;
216 for lin=1:Nely-1
217     for col=1:Nelx-1
218         sol_aprox(lin+1,col+1) = x_aux((lin-1)*(Nelx-1)+col);
219     end
220 end
221 figure(2)
222 surf(X,Y,sol_aprox)%::Solucao Aproximada

```

Código em Matlab para o método de elementos finitos de Galerkin mínimos quadrados (GLS) para o problema de Helmholtz 2D:

```

1 clear all;clc;close all;
2 %% -----Parametros de Entrada-----
3 teta=[0;pi/8;3*(pi/8)];
4 kn = 10; %Numero de Onda
5 %-----Eixo x
6 Nelx = 100; %Numero de Elementos em X e Y
7 Ax=0;
8 Bx=1;
9 hx=(Bx - Ax)/(Nelx);
10 %-----Eixo y
11 Nely = 100;
12 Ay=0;
13 By=1;
14 hy = (By - Ay)/(Nely);
15 %-----Malha do dominio: [Ax,Bx]x[Ay,By]
16 [X,Y] = meshgrid(Ax:hx:Bx,Ay:hy:By);
17 %-----Parametros do GLS:
18 h = hx;
19 s1 = kn*h*cos(pi/8);
20 s2 = kn*h*sin(pi/8);
21 tau = (1/kn^2)*(1-6*(4-cos(s1)-cos(s2)-2*cos(s1)*cos(s2))/...
22         ((2+cos(s1))*(2+cos(s2))*kn^2*h^2));
23 %% -----PESOS DA QUADRATURA DE GAUSS [-1,1]; n = 3-----
24 omega_aux(1) = 5/9;
25 omega_aux(2) = 8/9;
26 omega_aux(3) = 5/9;
27 %% -----PONTOS DA QUADRATURA [-1,1]; n = 3 -----
28 p_aux(1) = -sqrt(15)/5;
29 p_aux(2) = 0;
30 p_aux(3) = sqrt(15)/5;
31 %% ----- Inicializacoes-----
32 pp=0; % Indice para o Loop      %%% Mapeamento dos nos Globais
33 H(1,900).xini=[];H(1,900).xfin=[];
34 H(1,900).yini=[];H(1,900).yfin=[];
35 H(1,900).omega=[];H(1,900).x_q=[];
36 H(1,900).omegax=[];H(1,900).omegay=[];
37 H(1,900).y_q=[];
38 H(1,900).phi1=[];H(1,900).phi2=[];
39 H(1,900).phi3=[];H(1,900).phi4=[];
40 H(1,900).dphi1x=[];H(1,900).dphi2x=[];
41 H(1,900).dphi3x=[];H(1,900).dphi4x=[];
42 H(1,900).dphi1y=[];H(1,900).dphi2y=[];

```

```

43 H(1,900).dphi3y=[];H(1,900).dphi4y=[];
44 H(1,900).dof=[];H(1,900).Klocal=[];
45 H(1,900).flocal=[];
46 u_b = sparse(Nelx+1,Nely+1);
47 K = sparse((Nelx+1)*(Nely+1),(Nelx+1)*(Nely+1)); %Inicializacao da ...
    matriz K
48 f = zeros((Nelx+1)*(Nely+1),1); %Inicializacao do vetor f
49 sol_exata = zeros(Nely+1,Nelx+1);
50 %% -----Atribuicoes de cada elemento-----
51 for kk=1:Nely
52     for ii=1:Nelx
53 %-----Coordenadas dos elementos:
54     xini = (ii-1)/Nelx; % Posicao Inicial em x
55     xfin = ii/Nelx; % Posicao Final em x
56     yini = (kk-1)/Nely; % Posicao Inicial em y
57     yfin = (kk)/Nely; % Posicao Final em y
58 %-----Atribuicao das coordenadas:
59     H(ii+pp).xini = xini;
60     H(ii+pp).xfin = xfin;
61     H(ii+pp).yini = yini;
62     H(ii+pp).yfin = yfin;
63 %-----Definicoes da quadratura de Gauss:
64     omegax = 0.5*(xfin-xini)*omega_aux; %Peso da Quadratura no no X
65     omegay = 0.5*(yfin-yini)*omega_aux; %Peso da Quadratura no no Y
66     x_q = xini + 0.5*(1 +p_aux)*(xfin-xini); %Ponto da Quadratura no ...
        no(x)
67     y_q = yini + 0.5*(1 +p_aux)*(yfin-yini); %Ponto da Quadratura no ...
        no(y)
68 %-----Atribuicoes da quadratura de Gauss:
69     H(ii+pp).omegax = omegax;
70     H(ii+pp).omegay = omegay;
71     H(ii+pp).x_q = x_q;
72     H(ii+pp).y_q = y_q;
73 %-----Vetores para quadratura:
74     e1 = [1 1 1];
75     e2 = [1;1;1];
76 %-----Funcoes Base:
77     phi1 = (x_q'-xfin)*(y_q-yfin)/((xini-xfin)*(yini-yfin));
78     phi2 = (x_q'-xini)*(y_q-yfin)/((xfin-xini)*(yini-yfin));
79     phi3 = (x_q'-xini)*(y_q-yini)/((xfin-xini)*(yfin-yini));
80     phi4 = (x_q'-xfin)*(y_q-yini)/((xini-xfin)*(yfin-yini));
81 %-----Derivadas em x das funcoes base:
82     dphilx = (e2*y_q-yfin)/((xini-xfin)*(yini-yfin));

```

```

83     dphi2x = (e2*y_q-yfin)/((xfin-xini)*(yini-yfin));
84     dphi3x = (e2*y_q-yini)/((xfin-xini)*(yfin-yini));
85     dphi4x = (e2*y_q-yini)/((xini-xfin)*(yfin-yini));
86     %——Derivadas em y das funcoes base:
87     dphi1y = (x_q'*e1-xfin)/((xini-xfin)*(yini-yfin));
88     dphi2y = (x_q'*e1-xini)/((xfin-xini)*(yini-yfin));
89     dphi3y = (x_q'*e1-xini)/((xfin-xini)*(yfin-yini));
90     dphi4y = (x_q'*e1-xfin)/((xini-xfin)*(yfin-yini));
91     %——Atribuicoes das funcoes base:
92     H(ii+pp).phi1 = phi1;
93     H(ii+pp).phi2 = phi2;
94     H(ii+pp).phi3 = phi3;
95     H(ii+pp).phi4 = phi4;
96     %——Atribuicoes das derivadas em x das funcoes base:
97     H(ii+pp).dphi1x = dphi1x;
98     H(ii+pp).dphi2x = dphi2x;
99     H(ii+pp).dphi3x = dphi3x;
100    H(ii+pp).dphi4x = dphi4x;
101    %——Atribuicoes das derivadas em y das funcoes base:
102    H(ii+pp).dphi1y = dphi1y;
103    H(ii+pp).dphi2y = dphi2y;
104    H(ii+pp).dphi3y = dphi3y;
105    H(ii+pp).dphi4y = dphi4y;
106    %——Mapeamento das posicoes globais de cada no:
107    H(ii+pp).dof = [(1+Nelx)*(kk-1)+ii, (1+Nelx)*(kk-1)+ii+1, ...
108                  (1+Nelx)*(kk-1)+ii+Nelx+2, (1+Nelx)*(kk-1)+ii+Nelx+1];
109    end
110    pp=pp+Nelx;
111 end
112 %% -----MATRIZES LOCAIS-----
113 for i=1:Nelx*Nely
114     Klocal=zeros(4,4); % Iniciando Matriz de Rigidez
115     %—— Funcoes base, suas derivadas e pesos da quadratura para o elemento:
116     phi1 = H(i).phi1; dphi1x = H(i).dphi1x; dphi1y = H(i).dphi1y;
117     phi2 = H(i).phi2; dphi2x = H(i).dphi2x; dphi2y = H(i).dphi2y;
118     phi3 = H(i).phi3; dphi3x = H(i).dphi3x; dphi3y = H(i).dphi3y;
119     phi4 = H(i).phi4; dphi4x = H(i).dphi4x; dphi4y = H(i).dphi4y;
120     wx = H(i).omegax; wy = H(i).omegay;
121     %—— Matriz que carrega as combinacoes de peso da Quadratura:
122     Aw = omegax'*omegay;
123     %——Elementos da Matriz K Local para cada Elemento:
124     k11 = sum(sum(Aw.*(dphi1x.*dphi1x + dphi1y.*dphi1y - kn^2*phi1.*phi1 + ...
125                   tau*kn^4*phi1.*phi1)));

```

```

126 k12 = sum(sum(Aw.*(dphi1x.*dphi2x + dphi1y.*dphi2y - kn^2*phi1.*phi2 +...
127                 tau*kn^4*phi1.*phi2)));
128 k13 = sum(sum(Aw.*(dphi1x.*dphi3x + dphi1y.*dphi3y - kn^2*phi1.*phi3 +...
129                 tau*kn^4*phi1.*phi3)));
130 k14 = sum(sum(Aw.*(dphi1x.*dphi4x + dphi1y.*dphi4y - kn^2*phi1.*phi4 +...
131                 tau*kn^4*phi1.*phi4)));
132 k21 = sum(sum(Aw.*(dphi2x.*dphi1x + dphi2y.*dphi1y - kn^2*phi2.*phi1 +...
133                 tau*kn^4*phi2.*phi1)));
134 k22 = sum(sum(Aw.*(dphi2x.*dphi2x + dphi2y.*dphi2y - kn^2*phi2.*phi2 +...
135                 tau*kn^4*phi2.*phi2)));
136 k23 = sum(sum(Aw.*(dphi2x.*dphi3x + dphi2y.*dphi3y - kn^2*phi2.*phi3 +...
137                 tau*kn^4*phi2.*phi3)));
138 k24 = sum(sum(Aw.*(dphi2x.*dphi4x + dphi2y.*dphi4y - kn^2*phi2.*phi4 +...
139                 tau*kn^4*phi2.*phi4)));
140 k31 = sum(sum(Aw.*(dphi3x.*dphi1x + dphi3y.*dphi1y - kn^2*phi3.*phi1 +...
141                 tau*kn^4*phi3.*phi1)));
142 k32 = sum(sum(Aw.*(dphi3x.*dphi2x + dphi3y.*dphi2y - kn^2*phi3.*phi2 +...
143                 tau*kn^4*phi3.*phi2)));
144 k33 = sum(sum(Aw.*(dphi3x.*dphi3x + dphi3y.*dphi3y - kn^2*phi3.*phi3 +...
145                 tau*kn^4*phi3.*phi3)));
146 k34 = sum(sum(Aw.*(dphi3x.*dphi4x + dphi3y.*dphi4y - kn^2*phi3.*phi4 +...
147                 tau*kn^4*phi3.*phi4)));
148 k41 = sum(sum(Aw.*(dphi4x.*dphi1x + dphi4y.*dphi1y - kn^2*phi4.*phi1 +...
149                 tau*kn^4*phi4.*phi1)));
150 k42 = sum(sum(Aw.*(dphi4x.*dphi2x + dphi4y.*dphi2y - kn^2*phi4.*phi2 +...
151                 tau*kn^4*phi4.*phi2)));
152 k43 = sum(sum(Aw.*(dphi4x.*dphi3x + dphi4y.*dphi3y - kn^2*phi4.*phi3 +...
153                 tau*kn^4*phi4.*phi3)));
154 k44 = sum(sum(Aw.*(dphi4x.*dphi4x + dphi4y.*dphi4y - kn^2*phi4.*phi4 +...
155                 tau*kn^4*phi4.*phi4)));
156 %—Matriz f local
157 f1 = 0;
158 f2 = 0;
159 %— Atribuicao da Matriz Local K e f:
160 H(i).Klocal = [k11,k12,k13,k14;k21,k22,k23,k24;...
161               k31,k32,k33,k34;k41,k42,k43,k44];
162 H(i).flocal = [f1;f2];
163 end
164 %% CONDICAOES DE CONTORNO
165 ind=1;indd=1;
166 ua=zeros((Nelx+1)*(Nely+1),1);
167 for ii=1:Nely+1
168     for jj=1:Nelx+1

```

```

169 %-----Condicacao Inferior
170     if(ii==1)
171         x=Ax+(jj-1)*hx;y=Ay;
172         for p=1:length(teta)
173             ua((ii-1)*(Nelx+1)+jj) = ua((ii-1)*(Nelx+1)+jj)+...
174                                     cos(kn*(x*cos(teta(p))+y*sin(teta(p))));
175         end
176         conj(ind) = (ii-1)*(Nelx+1)+jj;
177         ind=ind+1;
178 %-----Condicacao Superior
179     elseif(ii==Nely+1)
180         x=Ax+(jj-1)*hx;y=By;
181         for p=1:length(teta)
182             ua((ii-1)*(Nelx+1)+jj) = ua((ii-1)*(Nelx+1)+jj)+...
183                                     cos(kn*(x*cos(teta(p))+y*sin(teta(p))));
184         end
185         conj(ind) = (ii-1)*(Nelx+1)+jj;
186         ind=ind+1;
187 %-----Condicacao Esquerda
188     elseif(jj==1)
189         x=Ax;y=Ay+(ii-1)*hy;
190         for p=1:length(teta)
191             ua((ii-1)*(Nelx+1)+jj) = ua((ii-1)*(Nelx+1)+jj)+...
192                                     cos(kn*(x*cos(teta(p))+y*sin(teta(p))));
193         end
194         conj(ind) = (ii-1)*(Nelx+1)+jj;
195         ind=ind+1;
196 %-----Condicacao Direita
197     elseif(jj==Nelx+1)
198         x=Bx;y=Ay+(ii-1)*hy;
199         for p=1:length(teta)
200             ua((ii-1)*(Nelx+1)+jj) = ua((ii-1)*(Nelx+1)+jj)+...
201                                     cos(kn*(x*cos(teta(p))+y*sin(teta(p))));
202         end
203         conj(ind) = (ii-1)*(Nelx+1)+jj;
204         ind=ind+1;
205     end
206 %-----Interior
207     if(ii≠Nely+1&&ii≠1&&jj≠Nelx+1&&jj≠1)
208         interior(indd) = (ii-1)*(Nelx+1)+jj;
209         indd=indd+1;
210     end
211 end

```

```

212 end
213 %% -----Loops do Assembly-----
214 for i=1:Nelx*Nely
215     Klocal = H(i).Klocal;
216     flocal = H(i).flocal;
217     dof = H(i).dof;
218     for ii=1:4
219         for jj=1:4
220             K(dof(ii),dof(jj)) = K(dof(ii),dof(jj))+Klocal(ii,jj);
221         end
222     end
223 f(i,1)=0;
224 end
225 %% -----Resolucao do Sistema Linear-----
226 b_d = f-K*ua;
227 K_int = K(interior,interior);
228 b_int = b_d(interior);
229 x_aux = K_int\b_int;
230 %% -----Solucao Exata-----
231 for p=1:length(teta)
232     sol_exata = sol_exata + cos(kn*(X.*cos(teta(p))+Y.*sin(teta(p))));
233 end
234 figure(1)
235 surf(X,Y,sol_exata)%::Solucao Exata
236 %% -----Solucao Aproximada-----
237 sol_aprox = sol_exata;
238 for lin=1:Nely-1
239     for col=1:Nelx-1
240         sol_aprox(lin+1,col+1) = x_aux((lin-1)*(Nelx-1)+col);
241     end
242 end
243 figure(2)
244 surf(X,Y,sol_aprox)%::Solucao Aproximada

```

Código em Matlab para o QSFEM para o problema de Helmholtz 2D:

```

1 clear all;clc;close all;
2 %% -----Parametros de Entrada-----
3 teta=[0;pi/8;3*(pi/8)];
4 kn = 10; %Numero de Onda
5 %-----Eixo x
6 Nelx = 50; %Numero de Elementos em X e Y

```

```

7 Ax=0;
8 Bx=1;
9 hx=(Bx - Ax)/(Nelx);
10 %——Eixo y
11 Nely = 50;
12 Ay=0;
13 By=1;
14 hy = (By - Ay)/(Nely);
15 %——Malha do dominio: [Ax,Bx]x[Ay,By]
16 [X,Y] = meshgrid(Ax:hx:Bx,Ay:hy:By);
17 %——Parimetros do QSFEM
18 h=hx;
19 c1=cos(kn*h*cos(pi/16));c2=cos(kn*h*cos(3*pi/16));
20 s1=cos(kn*h*sin(pi/16));s2=cos(kn*h*sin(3*pi/16));
21 A3=4;
22 A2=2*(c1*s1-c2*s2)/(c2*s2*(c1+s1)-c1*s1*(c2+s2));
23 A1=(c2+s2-c1-s1)/(c2*s2*(c1+s1)-c1*s1*(c2+s2));
24 %% —— Inicializacoes ——
25 pp=0; % Indice para o Loop      %%% Mapeamento dos nos Globais
26 H(1,900).dof=[];H(1,900).Klocal=[];
27 H(1,900).flocal=[];
28 u_b = sparse(Nelx+1,Nely+1);
29 K = sparse((Nelx+1)*(Nely+1),(Nelx+1)*(Nely+1));
30 f = zeros((Nelx+1)*(Nely+1),1);
31 sol_exata = zeros(Nely+1,Nelx+1);
32 %% —— Atribuicoes de cada elemento ——
33 for kk=1:Nely
34     for ii=1:Nelx
35 %——Mapeamento das posicoes globais de cada no:
36         H(ii+pp).dof = [(1+Nelx)*(kk-1)+ii,(1+Nelx)*(kk-1)+ii+1,...
37                         (1+Nelx)*(kk-1)+ii+Nelx+2,(1+Nelx)*(kk-1)+ii+Nelx+1];
38     end
39     pp=pp+Nelx;
40 end
41 %% —— MATRIZES LOCAIS ——
42 for i=1:Nelx*Nely
43 %——Matriz f local
44 f1 = 0;
45 f2 = 0;
46 %—— Atribuicao da Matriz Local K e f:
47 H(i).Klocal = [A3/4 A2/2 A1 A2/2;A2/2 A3/4 A2/2 A1;...
48                A1 A2/2 A3/4 A2/2;A2/2 A1 A2/2 A3/4];
49 H(i).flocal = [f1;f2];

```

```

50 end
51 %% CONDICAOES DE CONTORNO
52 ind=1; indd=1;
53 ua=zeros((Nelx+1)*(Nely+1),1);
54 for ii=1:Nely+1
55     for jj=1:Nelx+1
56         %-----Condicao Inferior
57         if(ii==1)
58             x=Ax+(jj-1)*hx;y=Ay;
59             for p=1:length(teta)
60                 ua((ii-1)*(Nelx+1)+jj) = ua((ii-1)*(Nelx+1)+jj)+...
61                                     cos(kn*(x*cos(teta(p))+y*sin(teta(p))));
62             end
63             conj(ind) = (ii-1)*(Nelx+1)+jj;
64             ind=ind+1;
65         %-----Condicao Superior
66         elseif(ii==Nely+1)
67             x=Ax+(jj-1)*hx;y=By;
68             for p=1:length(teta)
69                 ua((ii-1)*(Nelx+1)+jj) = ua((ii-1)*(Nelx+1)+jj)+...
70                                     cos(kn*(x*cos(teta(p))+y*sin(teta(p))));
71             end
72             conj(ind) = (ii-1)*(Nelx+1)+jj;
73             ind=ind+1;
74         %-----Condicao Esquerda
75         elseif(jj==1)
76             x=Ax;y=Ay+(ii-1)*hy;
77             for p=1:length(teta)
78                 ua((ii-1)*(Nelx+1)+jj) = ua((ii-1)*(Nelx+1)+jj)+...
79                                     cos(kn*(x*cos(teta(p))+y*sin(teta(p))));
80             end
81             conj(ind) = (ii-1)*(Nelx+1)+jj;
82             ind=ind+1;
83         %-----Condicao Direita
84         elseif(jj==Nelx+1)
85             x=Bx;y=Ay+(ii-1)*hy;
86             for p=1:length(teta)
87                 ua((ii-1)*(Nelx+1)+jj) = ua((ii-1)*(Nelx+1)+jj)+...
88                                     cos(kn*(x*cos(teta(p))+y*sin(teta(p))));
89             end
90             conj(ind) = (ii-1)*(Nelx+1)+jj;
91             ind=ind+1;
92         end

```

```

93  %-----Interior
94      if (ii≠Nely+1&&ii≠1&&jj≠Nelx+1&&jj≠1)
95          interior(indd) = (ii-1)*(Nelx+1)+jj;
96          indd=indd+1;
97      end
98  end
99  end
100 %% -----Loops do Assembly-----
101 for i=1:Nelx*Nely
102     Klocal = H(i).Klocal;
103     flocal = H(i).flocal;
104     dof = H(i).dof;
105     for ii=1:4
106         for jj=1:4
107             K(dof(ii),dof(jj)) = K(dof(ii),dof(jj))+Klocal(ii,jj);
108         end
109     end
110     f(i,1)=0;
111 end
112 %% -----Resolucao do Sistema Linear-----
113 b_d = f-K*ua;
114 K_int = K(interior,interior);
115 b_int = b_d(interior);
116 x_aux = K_int\b_int;
117 %% -----Solucao Exata-----
118 for p=1:length(teta)
119     sol_exata = sol_exata + cos(kn*(X.*cos(teta(p))+Y.*sin(teta(p)))));
120 end
121 figure(1)
122 surf(X,Y,sol_exata)%::Solucao Exata
123 %% -----Solucao Aproximada-----
124 sol_aprox = sol_exata;
125 for lin=1:Nely-1
126     for col=1:Nelx-1
127         sol_aprox(lin+1,col+1) = x_aux((lin-1)*(Nelx-1)+col);
128     end
129 end
130 figure(2)
131 surf(X,Y,sol_aprox)%::Solucao Aproximada

```