

THIAGO GIUSTI DEGASPARÉ

SIMULAÇÃO COMPUTACIONAL DE CAMPOS ACÚSTICOS  
GERADOS POR TRANSDUTORES ULTRASSÔNICOS EM  
EXCITAÇÃO CONTÍNUA

Monografia apresentada à Escola Politécnica  
da Universidade de São Paulo para obtenção  
do Título de Engenheiro Mecatrônico.

São Paulo  
2012

THIAGO GIUSTI DEGASPARE

**SIMULAÇÃO COMPUTACIONAL DE CAMPOS ACÚSTICOS  
GERADOS POR TRANSDUTORES ULTRASSÔNICOS EM  
EXCITAÇÃO CONTÍNUA**

Monografia apresentada à Escola Politécnica  
da Universidade de São Paulo para obtenção  
do Título de Engenheiro Mecatrônico.

Área de concentração:  
Engenharia Mecatrônica

Orientador:  
Prof. Dr. Flávio Buiochi

São Paulo  
2012

## FICHA CATALOGRÁFICA

Degaspare, Thiago Giusti

Simulação computacional de campos acústicos gerados por transdutores ultrassônicos em excitação contínua/ T. G. Degaspare. — São Paulo, 2012.

161 p.

Trabalho de Formatura — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1. Acústica (Simulação Computacional) 2. Transdutores eletromecânicos 3. Ultrassom I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos. II. t.

à minha Família

## AGRADECIMENTOS

Em primeiro lugar, ao meu pai Hermes e à minha mãe Celia, que me formaram enquanto pessoa e sempre me apoiaram e encorajaram de todas as formas possíveis. Também, à toda a minha família, que sempre esteve ao meu lado ao longo de todos esses anos.

Agradeço enormemente a meu orientador, Prof. Dr. Flávio Buiochi, que me apresentou o assunto do trabalho enquanto tema de iniciação científica, e posteriormente como trabalho de conclusão de curso, por toda a confiança, atenção, dedicação e auxílio fornecidos ao longo de seu desenvolvimento, e com o qual tive um grande aprendizado ao longo da minha formação acadêmica.

A todos os amigos e colegas com os quais trabalhei e convivi ao longo da minha participação no grupo PET Mecatrônica. Esse aprendizado contribuiu de maneira significativa não apenas para minha formação pessoal, mas também acadêmica e profissional. Em particular, ao Prof. Dr. Diolino José dos Santos Filho, pela atenção e dedicação dispensadas ao grupo enquanto tutor.

Aos colegas do laboratório de ultrassom e otimização, que me receberam com muita simpatia, pela companhia e pela disposição em ajudar e esclarecer dúvidas.

Finalmente, agradeço a todos os amigos e colegas que fiz ao longo de minha trajetória na Poli e durante o programa de Duplo Diploma. Todos os momentos compartilhados contribuíram de maneira direta ou indireta para minha formação, em todos os níveis, e consequentemente para a realização deste trabalho.

“Acoustic space has the basic character of a sphere whose focus or centre is simultaneously everywhere and whose margin is nowhere...”

(Marshall McLuhan)

# RESUMO

Sistemas ultrassônicos estão presentes em diversas aplicações médicas e industriais, o que lhes confere grande importância como domínio tecnológico. Exemplos são dados desde a formação de imagens para diagnóstico médico, até ensaios não destrutivos de componentes mecânicos. Ferramentas de simulação computacional são utilizadas com frequência para auxiliar na modelagem, projeto e seleção de tais sistemas. Uma questão de interesse é a determinação do campo acústico gerado por um transdutor ultrassônico. Este trabalho propõe o desenvolvimento de uma ferramenta gráfica para a modelagem e simulação do campo acústico gerado por três tipos de transdutores: circular plano, côncavo e *array* linear. O objetivo é obter simulações precisas de maneira rápida e interativa. Isso é obtido a partir da implementação do método da Matriz de Transferência Monocromática, um método numérico baseado na resposta em excitação contínua. A ferramenta gráfica é desenvolvida em *MATLAB*® e permite a realização de simulações envolvendo a propagação em meios simples e compostos, separados por uma interface de forma arbitrária. Primeiramente, os aspectos teóricos são abordados, como a teoria acústica e a demonstração do método numérico empregado. Apresentam-se aspectos de implementação, relacionados com as questões de modelagem dos transdutores e aspectos práticos de implementação dos algoritmos desenvolvidos. Finalmente, apresentam-se resultados comparando a simulação de testes de imersão com a ferramenta desenvolvida e um *software* comercial denominado *CIVA*®. Em particular, trata-se da questão de determinação das leis de atraso de um transdutor *array* comparando-se três métodos: dois propostos na literatura, e um desenvolvido no contexto deste trabalho.

**Palavras-chave:** Simulação computacional, transdutores ultrassônicos, campo acústico, modo de excitação contínuo, leis de atraso.

# ABSTRACT

Ultrasonic systems are widely used in many medical and industrial applications, and therefore have great importance as a field of technology. Examples are given from medical imaging applications for diagnosis, to nondestructive evaluation of mechanical components. Computer simulation is often used to aid in the modeling, design and selection of such systems. One important question is how to compute the acoustic field generated by an ultrasonic transducer. This work proposes the development of a graphic tool for modeling and simulating the acoustic field generated by three types of transducers: circular, both plane and focused, and linear array. The goal is to obtain precise, fast and interactive simulations. This is achieved by implementing the Monochromatic Transfer Matrix method, which is a continuous wave mode based numerical method. The graphic tool is developed in *MATLAB*® and allows simulations dealing with propagation in a simple medium or between two media, through arbitrary interfaces. Firstly, the theoretical aspects are presented, such as general concepts of acoustic theory and the derivation of the numerical method. Then, the implementation aspects are developed, discussing the general problem of modeling an ultrasonic system and the practical implementation aspects of the algorithms for acoustic field computation. Finally, the results are presented comparing immersion tests simulations for the developed tool and a comercial package, named *CIVA*®. Further attention is given to the determination of focal laws for array tranducers, comparing three different methods, two of them proposed in the litterature and one developed in the context of this work.

**Keywords:** Numerical simulation, ultrasonic transducers, acoustic field, continuous wave mode, focal laws.

# LISTA DE FIGURAS

1.1	Princípio de polarização em função da deformação no interior de materiais piezoelétricos, (CALLISTER, 2007) . . . . .	24
1.2	Transdutores circulares monoelementos de diferentes diâmetros, (ARNOLD; PELA, 2004) . . . . .	24
1.3	Exemplos de transdutores circulares monoelementos: (a) Transdutor circular plano, (b) Transdutor circular côncavo . . . . .	27
1.4	Formação de imagens para diagnóstico com ultrassom, (KASS; NELLIGAN; HENJES, 2006): (a) Imagem em modo B-scan, (b) Imagem em modo C-scan .	27
1.5	Geração de scans lineares e setoriais usando <i>arrays</i> com atraso de fase, (KASS; NELLIGAN; HENJES, 2006) . . . . .	28
1.6	Comparação entre sistemas de transdutores <i>arrays</i> , (KASS; NELLIGAN; HENJES, 2006): (a) Sistema industrial de <i>arrays</i> para inspeção de soldas, (b) Sistema para formação de imagens em modo C, utilizando <i>array</i> com 128 elementos . . . . .	29
1.7	Transdutores <i>arrays</i> em suas diferentes configurações, (DRINKWATER; WILCOX, 2006): (a) Arranjo linear multi-elementos, (b) Arranjo bidimensional, (c) Arranjo anelar multi-elementos . . . . .	29
1.8	Propagação do campo acústico através de uma interface, (IBÁÑEZ; VILAZÓN, 2006) . . . . .	31
2.1	Lugar geométrico dos pontos de fase constante, (KINSLER, 2000) . . . . .	38
2.2	Princípio de Huygens, (GOODMAN, 2005) . . . . .	39
2.3	Tensor de tensão. . . . .	41
2.4	Diferentes modos de propagação em um sólido, (CHEEKE, 2002) (a) Ondas de compressão, modo longitudinal (b) Ondas de cisalhamento, modo transversal, . . . . .	43
2.5	Aplicação do teorema de Green, (GOODMAN, 2005) . . . . .	45

2.6	Cálculo discreto de $U$ no ponto $P_0$ , (IBÁÑEZ; VILLAZÓN, 2006) . . . . .	47
2.7	Casos de propagação entre dois meios, colocando em evidência a existência de diferentes modos de propagação em sólidos, (CHEEKE, 2002) . . . . .	49
2.8	Energia transmitida e refletida para interface água-alumínio. . . . .	51
2.9	Aplicação do princípio de Fermat para o caso de propagação em meio composto com uma interface de separação $S$ entre os meios, (PARRILLA et al., 2008) . . . . .	53
3.1	Transdutor circular plano. . . . .	59
3.2	Modelagem de um transdutor circular plano: (a) Transdutor circular plano, $r = 10mm$ , (b) Transdutor circular plano discretizado, $r_{div} = 20$ . . . . .	59
3.3	Parâmetros de um transdutor circular côncavo: (a) Transdutor circular côncavo, (b) Sistema de coordenadas esférico. . . . .	60
3.4	Modelagem de um transdutor circular côncavo . . . . .	61
3.5	Modelagem de um transdutor circular côncavo: (a) Transdutor circular côncavo, $R = 10mm$ , $r = 6mm$ , $\phi = \pi rad$ , $\theta = \pi rad$ , (b) Transdutor circular côncavo discretizado, $\phi_{div} = 60$ , $\theta_{div} = 60$ . . . . .	61
3.6	Parâmetros de um transdutor <i>array</i> linear. . . . .	62
3.7	Modelagem de um transdutor <i>array</i> linear: (a) Transdutor <i>array</i> linear, $N = 4$ , $a = 2mm$ , $b = 10mm$ e $g = 1mm$ , (b) Transdutor <i>array</i> linear discretizado, $h_{div} = 4$ , $l_{div} = 10$ . . . . .	63
3.8	Discretização de uma região de interesse para o caso de propagação em um meio simples. . . . .	65
3.9	Modelagem de um meio de propagação composto com interface de separação do tipo <i>spline</i> (a) Visualização espacial, (b) Visualização no plano $(xz)$ . . . . .	66
3.10	Plano tridimensional traçado a partir de uma reta definida pelos pontos $(x_1, z_1) = (-20, 20)mm$ e $(x_2, z_2) = (20, 30)mm$ . . . . .	68
3.11	Etapas para implementação do método <i>MTM</i> . . . . .	76

3.12	Diagrama MVC. . . . .	77
3.13	Implementação do modelo do sistema. . . . .	78
3.14	Implementação da parte gráfica da interface. . . . .	80
3.15	Implementação do controlador. . . . .	81
3.16	Máquina de estados que modela o comportamento da interface. . . . .	81
4.1	Ferramenta para simulação do campo acústico, ambiente gráfico. . . . .	84
4.2	Modelagem de um teste de imersão em <i>CIVA</i> ®, transdutor circular plano: (a) Vista em perspectiva, (b) Vista superior. . . . .	86
4.3	Modelagem de um teste de imersão em <i>MATLAB</i> ®, transdutor circular plano: (b) Vista em perspectiva, (b) Vista superior. . . . .	86
4.4	Resultados para teste de imersão, transdutor circular plano em modo longitudinal: (a) Simulação em <i>CIVA</i> ®, (b) Simulação em <i>MATLAB</i> ®. . . . .	88
4.5	Resultados para teste de imersão, transdutor circular plano modo longitudinal sem conversão de modo: (a) Simulação em <i>CIVA</i> ®, (b) Simulação em <i>MATLAB</i> ®. . . . .	89
4.6	Resultados para teste de imersão, modo transversal: (a) Simulação em <i>CIVA</i> ®, (b) Simulação em <i>MATLAB</i> ®. . . . .	90
4.7	Modelagem de um teste de imersão em <i>CIVA</i> ®, transdutor circular côncavo: (a) Vista em perspectiva, (b) Vista superior. . . . .	91
4.8	Modelagem de um teste de imersão utilizando a ferramenta em <i>MATLAB</i> ®, transdutor circular côncavo: (b) Vista em perspectiva, (b) Vista superior. .	92
4.9	Resultados para teste de imersão, transdutor circular côncavo em modo longitudinal: (a) Simulação em <i>CIVA</i> ®, (b) Simulação em <i>MATLAB</i> ®. .	93
4.10	Campo acústico em três dimensões para um transdutor circular côncavo. .	94
4.11	Resultados para teste de imersão, transdutor circular côncavo em modo longitudinal: (a) Simulação em <i>CIVA</i> ®, (b) Simulação em <i>MATLAB</i> ®. .	95
4.12	Modelo do transdutor array para teste de imersão. . . . .	95

4.13 Determinação das leis de atraso em <i>MATLAB</i> <sup>®</sup> para transdutor <i>array</i> focalizado em $(x_f, y_f, z_f) = (0, 0, 100mm)$ : (a) Princípio de Fermat, (b) Método de Newton-Raphson, (c) Método de Nelder-Mead. . . . .	96
4.14 Determinação das leis de atraso em <i>CIVA</i> <sup>®</sup> para transdutor <i>array</i> focalizado em $(x_f, y_f, z_f) = (0, 0, 100mm)$ . . . . .	97
4.15 Resultados para teste de imersão, transdutor <i>array</i> em modo longitudinal focalizado em $(x_f, y_f, z_f) = (0, 0, 100mm)$ : (a) Simulação em <i>CIVA</i> <sup>®</sup> , (b) Simulação em <i>MATLAB</i> <sup>®</sup> . . . . .	98
4.16 Determinação das leis de atraso para transdutor <i>array</i> focalizado em $(x_f, y_f, z_f) = (5, 0, 100mm)$ : (a) Cálculo em <i>CIVA</i> <sup>®</sup> , (b) Cálculo em <i>MATLAB</i> <sup>®</sup> , princípio de Fermat. . . . .	99
4.17 Resultados para teste de imersão, transdutor <i>array</i> modo longitudinal focalizado em $(x_f, y_f, z_f) = (5, 0, 100mm)$ : (a) Simulação em <i>CIVA</i> <sup>®</sup> , (b) Simulação em <i>MATLAB</i> <sup>®</sup> . . . . .	100
4.18 Campo acústico em três dimensões para um transdutor <i>array</i> . . . . .	101
4.19 Resultados para teste de imersão, transdutor <i>array</i> em modo longitudinal focalizado em $(x_f, y_f, z_f) = (5, 0, 100mm)$ , plano transversal $z = 70mm$ : (a) Simulação em <i>CIVA</i> <sup>®</sup> , (b) Simulação em <i>MATLAB</i> <sup>®</sup> . . . . .	101
4.20 Determinação das leis de atraso em <i>MATLAB</i> <sup>®</sup> : (a) Modelo em <i>CIVA</i> <sup>®</sup> , (b) Modelo em <i>MATLAB</i> <sup>®</sup> , perspectiva, (c) Modelo em <i>MATLAB</i> <sup>®</sup> , vista superior. . . . .	102
4.21 Determinação das leis de atraso para transdutor <i>array</i> focalizado em $(x_f, y_f, z_f) = (-5, 0, 40)mm$ : (a) Cálculo em <i>CIVA</i> <sup>®</sup> , (b) Cálculo em <i>MATLAB</i> <sup>®</sup> , princípio de Fermat. . . . .	103
4.22 Resultados para teste de imersão, transdutor <i>array</i> em modo longitudinal focalizado em $(x_f, y_f, z_f) = (-5, 0, 40)mm$ , para interface curvilínea: (a) Simulação em <i>CIVA</i> <sup>®</sup> , (b) Simulação em <i>MATLAB</i> <sup>®</sup> . . . . .	104
4.23 Modelagem para avaliação dos algoritmos de lei de atraso de um transdutor <i>array</i> . . . . .	106

4.24 Determinação das leis de atraso, comparação entre métodos: (a) Erro médio em relação ao princípio de Fermat, (b) Tempo de cálculo. . . . .	107
A.1 Diagrama de estruturas MVC . . . . .	115
A.2 Estrutura interna de <i>Model</i> . . . . .	116
A.3 Classes associadas ao modelo do transdutor . . . . .	117
A.4 Classes associadas ao modelo do meio de propagação . . . . .	118
A.5 Classes associadas à interface gráfica . . . . .	119
A.6 Classes associadas à interface gráfica, atributos e métodos, parte 1 . . . . .	120
A.7 Classes associadas à interface gráfica, atributos e métodos, parte 2 . . . . .	121
A.8 Interface gráfica principal, métodos e atributos . . . . .	122
A.9 Interface gráfica principal, diagrama de componentes . . . . .	123
A.10 Classes associadas ao controlador: (a) Estrutura do controlador, (b) Atributos e métodos das classes associadas ao controlador . . . . .	124
A.11 Diagrama de estados associado ao controlador . . . . .	124
A.12 Exemplo de aplicação . . . . .	144
A.13 Ambiente de desenvolvimento . . . . .	145
A.14 Configuração de um meio de propagação simples . . . . .	146
A.15 Seleção de um meio de propagação simples . . . . .	146
A.16 Configuração de um transdutor circular plano . . . . .	147
A.17 Ambiente pós-configuração . . . . .	148
A.18 Campo acústico para transdutor circular plano imerso em água . . . . .	148
A.19 Ambiente de configuração de um meio composto . . . . .	149
A.20 Configuração de uma <i>spline</i> . . . . .	150
A.21 Geometria final do meio de propagação composto . . . . .	150
A.22 Ambiente de configuração de um transdutor <i>array</i> . . . . .	151
A.23 Ambiente de configuração das leis de atraso de um transdutor <i>array</i> . . . . .	152

A.24 Visualização das leis de atraso de um transdutor <i>array</i> . . . . .	152
A.25 Visualização de resultados para simulação com um transdutor <i>array</i> : (a) Habilitar focalização com posição do mouse, (b) Campo acústico em plano 2D . . . . .	153
A.26 Resultado para nova posição focal definida com o mouse para um transdutor <i>array</i> : (a) Campo acústico 2D, (b) Leis de atraso associadas . . . . .	153
A.27 Configuração de um plano em três dimensões . . . . .	154
A.28 Resultado para nova posição focal definida com o mouse para um transdutor <i>array</i> : (a) Vista em perspectiva, (b) Projeção no plano . . . . .	154
A.29 Ambiente de configuração de um transdutor circular côncavo . . . . .	155
A.30 Resultados para a simulação com um transdutor circular côncavo . . . . .	156
B.1 Ambiente de configuração da ferramenta <i>FOCUS</i> . . . . .	158
B.2 Resultados para teste de imersão, transdutor circular plano em modo longitudinal: (a) Simulação em <i>FOCUS</i> , (b) Simulação em <i>MATLAB</i> ® . . . . .	159
B.3 Resultados para teste de imersão, transdutor circular plano em modo longitudinal: (a) Simulação em <i>FOCUS</i> , (b) Simulação em <i>MATLAB</i> ® . . . . .	161

## LISTA DE TABELAS

3.1	Propriedades físicas de alguns meios de propagação de interesse, retirado de Cheeke (2002) . . . . .	64
3.2	Parâmetros de discretização de um meio simples, para a figura 3.8 . . . . .	64
3.3	Parâmetros de configuração para <i>spline</i> da figura 3.9. . . . .	66
4.1	Propriedades físicas dos meios de propagação para os testes de imersão . .	85
4.2	Configuração para teste de imersão, transdutor circular plano. . . . .	87
4.3	Configuração para teste de imersão com transdutor circular côncavo . . . .	92
4.4	Configuração para teste de imersão com transdutor <i>array</i> . . . . .	97
4.5	Propriedades físicas dos meios de propagação para avaliação do cálculo das leis de atraso de um transdutor <i>array</i> . . . . .	105
B.1	Parâmetros de configuração . . . . .	160
B.2	Parâmetros de configuração . . . . .	161

## LISTA DE ABREVIATURAS

**PZT** Titanato de Chumbo

**PVDF** Polivinilideno

**END** Ensaio Não Destrutivo

**CEA** Commissariat à l'énergie atomique et aux énergies alternatives

**DREAM** Discrete Representation Array Modelling

**FOCUS** Fast Object Oriented C++ Ultrasound Simulator

**FNM** Fast Nearfield Method

**MTM** Monochromatic Transfer Matrix

**MVC** Model-View-Controller

**UML** Unified Modeling Language

# LISTA DE SÍMBOLOS

---

Símbolo	Descrição
$\nabla$	operador diferencial
$\nabla^2$	operador laplaciano
$dA$	área de elemento infinitesimal
$dV$	volume de elemento infinitesimal
$u$	deslocamento
$\vec{r}$	vetor posição
$p$	pressão acústica
$\rho_0$	densidade
$S$	dilatação
$\chi$	compressibilidade de um fluido
$t$	tempo
$\phi$	potencial da pressão acústica
$\vec{k}$	módulo de elasticidade do material base
$j$	número complexo $j = \sqrt(-1)$
$\omega$	velocidade angular
$\lambda$	comprimento de onda
$c$	velocidade de propagação
$f$	frequência
$T$	período
$(r, \varphi, \theta)$	sistema de coordenadas esférico
$\sigma$	tensor de tensão
$\epsilon$	tensor deformação
$\mu, \lambda$	constantes de <i>Lamé</i>
$c_l$	velocidade de propagação em modo longitudinal
$c_s$	velocidade de propagação em modo transversal
$P_0$	ponto arbitrário do espaço para cálculo do campo acústico
$G$	função de Green do espaço livre
$U$	campo acústico em excitação contínua

Símbolo	Descrição
$\vec{n}$	vetor normal
$\mathbf{U}$	vetor campo acústico para solução do <i>MTM</i>
$\mathbf{T}$	matriz de transferência monocromática
$\mathbf{R}$	matriz de distâncias para implementação do <i>MTM</i>
$\mathbf{E}$	matriz de excitação do transdutor para implementação do <i>MTM</i> (descreve amplitude e fase)
$\alpha$	ângulo de incidência
$t_v$	tempo de propagação da onda para determinação das leis de atraso

# SUMÁRIO

1	INTRODUÇÃO	23
1.1	Ultrassom e suas aplicações	23
1.2	Formação de imagens por ultrassom	25
1.3	Definição do problema	30
1.4	Objetivos	31
1.5	Revisão bibliográfica	32
2	DETERMINAÇÃO NUMÉRICA DO CAMPO ACÚSTICO	35
2.1	Propagação em líquidos	36
2.1.1	Ondas planas em três dimensões	36
2.1.2	Ondas esféricas emitidas por fontes pontuais	39
2.2	Propagação de ondas em sólidos	41
2.3	Determinação do campo acústico	44
2.4	Método matricial para determinação do campo acústico	46
2.5	Casos de estudo	48
2.5.1	Meio de propagação simples	48
2.5.2	Meio de propagação composto	49
2.6	Determinação das leis de atraso para um transdutor array	52
2.6.1	Algoritmo de Newton-Raphson	54
2.6.2	Algoritmo de Nelder-Mead	56
3	IMPLEMENTAÇÃO COMPUTACIONAL	58
3.1	Modelagem	58

3.1.1	Transdutor Circular Plano . . . . .	58
3.1.2	Transdutor Circular Côncavo . . . . .	60
3.1.3	Transdutor Array . . . . .	61
3.1.4	Meio de propagação . . . . .	63
3.1.5	Análise tridimensional . . . . .	67
3.2	Implementação do <i>MTM</i> . . . . .	67
3.2.1	MTM em um meio de propagação simples . . . . .	68
3.2.2	MTM em um meio de propagação composto . . . . .	71
3.3	Projeto do software . . . . .	76
3.3.1	Implementação do modelo . . . . .	78
3.3.2	Implementação da parte gráfica . . . . .	79
3.3.3	Implementação do controlador . . . . .	79
3.3.4	Rotinas auxiliares . . . . .	79
4	RESULTADOS	83
4.1	Apresentação da Ferramenta de Simulação . . . . .	83
4.2	Testes de imersão . . . . .	84
4.2.1	Transdutor circular plano . . . . .	85
4.2.2	Transdutor circular côncavo . . . . .	91
4.2.3	Transdutor <i>array</i> . . . . .	94
4.2.4	Interface circular . . . . .	99
4.3	Métodos de determinação das leis de atraso . . . . .	105
4.4	Discussão dos resultados . . . . .	108
	CONCLUSÃO	110
	REFERÊNCIAS	112

ANEXO A – FERRAMENTA COMPUTACIONAL	114
A.1 Documentação . . . . .	114
A.1.1 Modelo . . . . .	114
A.1.2 Interface gráfica . . . . .	115
A.1.3 Controlador . . . . .	123
A.2 Listagem de rotinas auxiliares . . . . .	125
A.2.1 MTM meio simples . . . . .	125
A.2.2 MTM meio composto . . . . .	127
A.2.3 Apodização . . . . .	129
A.2.4 Leis de atraso, meio simples . . . . .	130
A.2.5 Leis de atraso meio composto, princípio de Fermat . . . . .	130
A.2.6 Leis de atraso meio composto, método de Newton-Raphson . . . . .	131
A.2.7 Método de Newton-Raphson . . . . .	133
A.2.8 Função objetivo, método de Newton-Rapshon . . . . .	134
A.2.9 Leis de atraso meio composto, método de Nelder-Mead . . . . .	134
A.2.10 Função objetivo, método de Nelder-Mead . . . . .	136
A.2.11 Ângulos de incidência . . . . .	136
A.2.12 Coeficientes de transmissão . . . . .	138
A.2.13 Conversão de modo . . . . .	141
A.2.14 Interface curvilínea . . . . .	141
A.2.15 Interface spline . . . . .	142
A.3 Tutorial . . . . .	144
A.3.1 Exemplo 1 . . . . .	145
A.3.2 Exemplo 2 . . . . .	149
A.3.3 Exemplo 3 . . . . .	155

A.4 Considerações finais . . . . .	155
ANEXO B – COMPARAÇÃO COM O TOOLBOX FOCUS	
B.1 Transdutor circular plano . . . . .	157
B.2 Transdutor array . . . . .	157
B.3 Discussão . . . . .	160

# 1 INTRODUÇÃO

## 1.1 Ultrassom e suas aplicações

Uma onda pode ser considerada como a propagação ou o movimento de uma perturbação, seja em um determinado meio físico, como no caso de ondas mecânicas, seja sem a presença do mesmo, como no caso de ondas eletromagnéticas, Serway e Jewett (2009).

A ciência que estuda a propagação de ondas mecânicas em líquidos, gases e sólidos é denominada *Acústica*. Ela compreende o estudo do som, infrassom e ultrassom, que diferem entre si de acordo com a frequência de vibração. O ouvido humano é sensível às frequências que variam no intervalo de 20Hz à 20kHz. Ondas abaixo dessa faixa são denominadas *infrassom* e ondas acima dessa faixa são denominadas *ultrassom*.

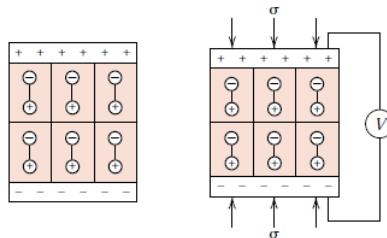
Intuitivamente, pode-se dizer que o ultrassom tem sua origem na vida marinha, Cheeke (2002). No ambiente terrestre, os cinco sentidos (olfato, audição, visão, paladar e tato), conferem ao ser-humano capacidades plenas de interação com o meio em curtas e longas distâncias. Porém, em ambientes marinhos, alguns sentidos, como a visão, possuem capacidades extremamente limitadas. As formas mais desenvolvidas da fauna marinha, como baleias e golfinhos, possuem sistemas de ultrassom extremamente sofisticados, que lhes conferem capacidades de interação e adaptação a esse ambiente.

Historicamente, o desenvolvimento tecnológico do ultrassom tem início na 1.<sup>a</sup> Guerra Mundial, com a criação de sonares rudimentares para a detecção de submarinos e minas. O físico francês, Paul Langevin, em parceria com Constantin Chilowski, seu compatriota, foram os responsáveis pelo desenvolvimento do princípio do sonar, em estudos realizados em Paris no ano de 1917, como descreve Zimmerman (2008).

O princípio do sensor desenvolvido por Langevin reside na utilização de transdutores utilizados como transmissores e receptores. De maneira geral, um transdutor é um dispositivo capaz de converter diferentes formas de energia. Em particular, a utilização da piezoelectricidade em dispositivos de ultrassom proporcionou um grande avanço no desenvolvimento de aplicações tecnológicas.

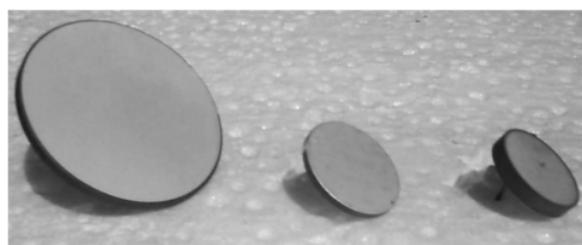
O efeito piezoelectrônico consiste no estabelecimento de um campo elétrico através de

uma amostra por meio da aplicação de forças externas, Callister (2007). É um processo reversível, no qual a reversão de um sinal de força externa inverte a direção do campo. O material é submetido a tensões elétricas, deforma-se e consequentemente vibra, emitindo ondas acústicas. Os materiais mais conhecidos que apresentam esse comportamento são o Titanatos de Chumbo (PZT), o cristal de quartzo e o polivinilideno (PVDF).



**Figura 1.1:** Princípio de polarização em função da deformação no interior de materiais piezoelétricos, (CALLISTER, 2007).

Atualmente, os sistemas baseados em ultrassom são aplicados em diversas áreas. Na medicina, ele é utilizado na geração de imagens e para fins terapêuticos, Fish (1990). Na indústria, sua principal aplicação é em ensaios não destrutivos (END) de componentes mecânicos e para identificação de defeitos, Drinkwater e Wilcox (2006). Além disso, é possível realizar a caracterização de líquidos a partir da medição de viscosidade Buiochi (1994). Outras aplicações são sistemas de navegação de veículos tripulados e não tripulados, como robôs móveis terrestres e submarinos, Cook (2011). A figura 1.2 exibe alguns transdutores cerâmicos monoelemento de geometria circular.



**Figura 1.2:** Transdutores circulares monoelementos de diferentes diâmetros, (ARNOLD; PELA, 2004).

A versatilidade dessas aplicações confere grande importância ao estudo de sistemas ultrassônicos, de maneira a se buscar constantemente seu aperfeiçoamento. Entre outros aspectos, é de fundamental importância o estudo dos níveis de pressão gerados por transdutores em um determinado meio. Esses níveis constituem o *campo acústico* de um transdutor.

O campo acústico é em grande parte função da geometria do transdutor e do meio de propagação. Transdutores de geometria simples, como os da figura 1.2, monoelementos de geometria circular, possuem solução analítica para o campo acústico, assunto amplamente explorada na literatura. No entanto, quando o cálculo envolve geometrias mais complexas, o uso de soluções analíticas é limitado e há a necessidade de se recorrer a métodos numéricos.

Métodos numéricos são ferramentas de extrema importância para a quase totalidade das áreas de engenharia. Problemas reais frequentemente envolvem geometrias complexas e não-linearidades que são dificilmente resolvidas utilizando-se soluções exatas. Nesse sentido, existe um grande esforço para o desenvolvimento de ferramentas computacionais eficientes para o estudo e análise de sistemas de engenharia.

Assim, a utilização de ferramentas computacionais para a simulação e o projeto de transdutores e sensores ultrassônicos é fundamental. Segundo o trabalho de Drinkwater e Wilcox (2006), a modelagem é uma etapa chave na seleção e otimização na realização de inspeções para aplicações industriais. Dada a grande variedade de arranjos existentes, a simulação computacional auxilia na investigação de configurações ótimas para uma dada aplicação. Nesse contexto, existe um grande esforço por parte da comunidade científica para o desenvolvimento de aplicações para auxiliar no dimensionamento, testes de sistemas e processamento de dados.

A seguir, pretende-se discutir as técnicas mais comuns utilizadas para a formação de imagens por ultrassom, de forma a apresentar os diferentes tipos de transdutores utilizados e introduzir o tipo de problemática que será abordada.

## 1.2 Formação de imagens por ultrassom

O trabalho de Kino (1987) descreve os diferentes tipos de técnicas e sistemas utilizados na formação de imagens por ultrassom. O sistema ultrassônico mais simples que pode ser empregado para a formação de imagens utiliza um transdutor monoelemento para transmitir ondas acústicas ao objeto a ser examinado. Em geral, o transdutor pode estar imerso em um meio, como por exemplo água, e a perturbação é transmitida para o objeto a partir desse meio. Outras aplicações podem utilizar o transdutor diretamente em contato com o objeto; nesse caso o contato é estabelecido a partir de uma fina camada acoplante de gel ou óleo.

O transdutor é excitado e parte das ondas serão refletidas devido a falhas ou descontinuidades na estrutura interna do objeto. O atraso entre emissão-recepção permite medir a profundidade das falhas. Já a amplitude do eco medido em recepção permite determinar o tamanho e a forma da falha. Essa técnica é denominada *amplitude scan*, mais comumente modo A ou A-Scan. Ela permite determinar a posição transversal das falhas e sua eficácia está fortemente relacionada com as dimensões do transdutor e parâmetros inerentes ao meio, como o comprimento de onda.

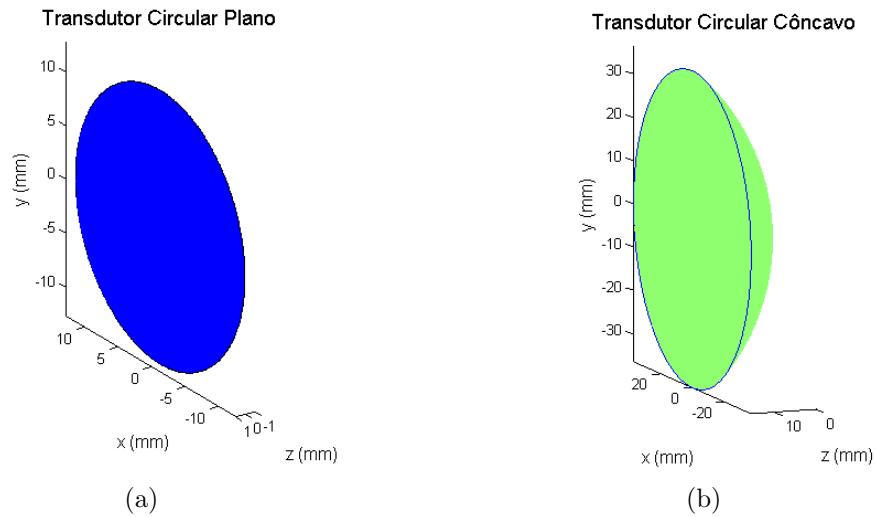
O A-scan emprega transdutores monoelementos simples, conforme ilustrado na figura 1.2. É uma técnica amplamente utilizada na indústria nuclear e aeronáutica para a detecção de falhas em materiais. Na medicina, é utilizada para diagnósticos em oftalmologia.

A desvantagem da realização do A-scan é de que se trata de um processo lento e tedioso, uma vez que apenas uma linha de amplitude pode ser analisada por vez e resulta em uma quantidade de informações excessiva para ser tratada por um operador humano. Como alternativa, utiliza-se uma técnica denominada modo B ou B-scan (*Brightness Scan*), na qual se modula a intensidade de um ponto no sistema de medição a partir da amplitude. O atraso é representado pela posição desse ponto no eixo das abscissas e a posição do transdutor é representada no eixo das ordenadas.

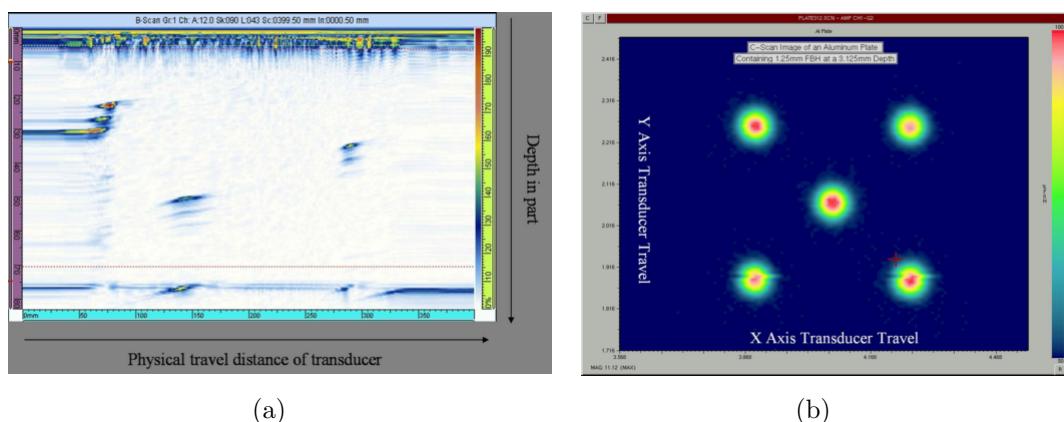
As figuras 1.3(a) e 1.3(b) resumem dois tipos básicos de transdutores monoelementos utilizados nas aplicações descritas acima. Tratam-se de transdutores circulares monoelementos em duas formas, plana e côncava. A vantagem de um transdutor circular côncavo é a possibilidade de focalização do feixe emitido no centro de curvatura do elemento.

Uma terceira técnica é denominada modo C ou C-scan. O objetivo é a formação de imagens no plano perpendicular à direção de propagação do feixe acústico. Ela mapeia a intensidade de amplitude do sinal medido em uma dada região em um gráfico bi-dimensional de posição no plano  $xy$ , onde a amplitude é representada em escala de cinza ou por cores.

As figuras 1.4(a) e 1.4(b) ilustram imagens nos modos B-scan e C-scan, respectivamente. Para a formação dessas imagens utilizam-se transdutores multielementos (matriciais ou *arrays*).



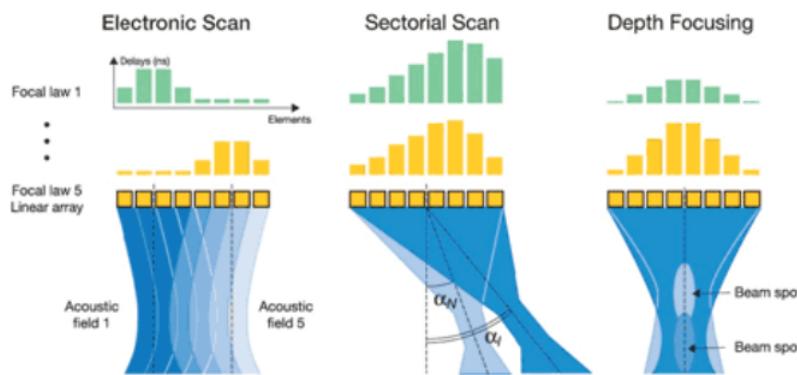
**Figura 1.3:** Exemplos de transdutores circulares monoelementos: (a) Transdutor circular plano, (b) Transdutor circular côncavo .



**Figura 1.4:** Formação de imagens para diagnóstico com ultrassom, (KASS; NELLIGAN; HENJES, 2006): (a) Imagem em modo B-scan, (b) Imagem em modo C-scan .

O trabalho de Kass, Nelligan e Henjes (2006) aponta os benefícios da utilização de transdutores *arrays* para aplicações de inspeção. Uma vantagem na utilização desses tipos de transdutores é a possibilidade de focalização do feixe acústico em diferentes posições do espaço de trabalho sem a necessidade de movimentar o conjunto de emissores.

O princípio de transdutores *arrays* com atraso de fase foi introduzido por James Sumner em 1968. A figura 1.5 ilustra um sistema de transdutores matriciais com seus respectivos sinais de emissão e recepção em varreduras lineares e setoriais.

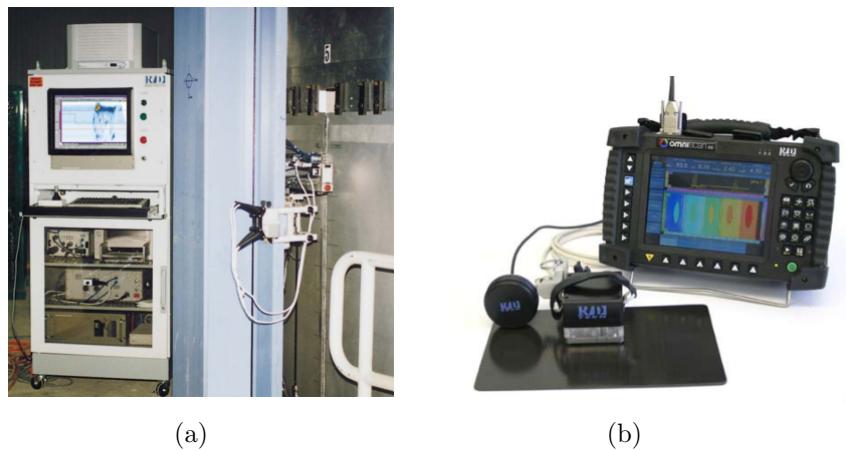


**Figura 1.5:** Geração de scans lineares e setoriais usando *arrays* com atraso de fase, (KASS; NELLIGAN; HENJES, 2006).

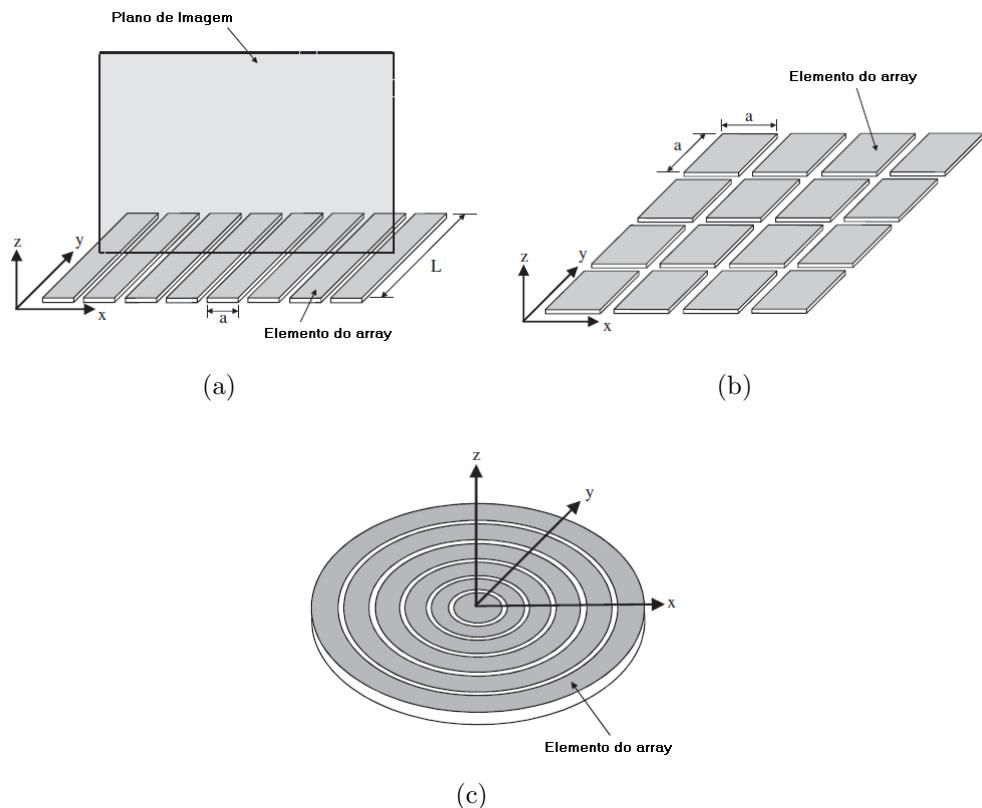
A partir dos anos 1970 essa tecnologia permitiu o desenvolvimento de sistemas de medição em tempo real de grande versatilidade, permitindo a variação de ângulo, posição e distância focal. As figuras 1.6(a) e 1.6(b) mostram a evolução dos sistemas de medição em termos de portabilidade. A primeira figura mostra os primeiros sistemas com transdutores *arrays*, os quais necessitavam transferir os dados para processamento e representação de imagens em computador. Trata-se de sistemas de grande porte. Na figura ao lado, mostra-se um sistema atual para formação de imagens em modo C-scan que conta com um *array* de 128 elementos.

Os transdutores *arrays* podem ser constituídos por elementos de diversas formas. A figura 1.7 apresenta diferentes geometrias utilizadas.

O trabalho desenvolvido por Drinkwater e Wilcox (2006) propõe um estado da arte da tecnologia de transdutores *array* utilizada para ensaios não destrutivos na indústria. O grande número de variáveis e parâmetros envolvidos torna o desenvolvimento desse tipo de sistema uma tarefa de engenharia complexa. Assim, ferramentas de simulação podem auxiliar no desenvolvimento de sistemas ultrassônicos.



**Figura 1.6:** Comparaçāo entre sistemas de transdutores *arrays*, (KASS; NELLIGAN; HENJES, 2006): (a) Sistema industrial de *arrays* para inspeção de soldas, (b) Sistema para formação de imagens em modo C, utilizando *array* com 128 elementos .



**Figura 1.7:** Transdutores *arrays* em suas diferentes configurações, (DRINKWATER; WILCOX, 2006): (a) Arranjo linear multi-elementos, (b) Arranjo bidimensional, (c) Arranjo anelar multi-elementos .

### 1.3 Definição do problema

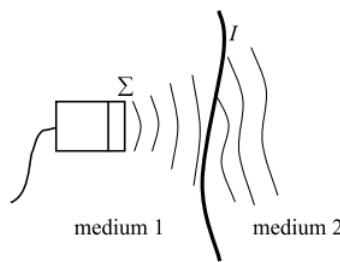
A tecnologia utilizada para inspeção por ultrassom varia segundo o tipo de aplicação. Porém, mesmo restringindo o problema, as escolhas entre os tipos de transdutores, número de elementos, modos de propagação e dimensionamento não são evidentes. Como em muitas outras áreas da engenharia, a simulação computacional surge como uma ferramenta de grande importância para auxiliar o desenvolvimento desses tipos de sistemas. Em particular, a determinação do campo acústico permite avaliar o comportamento e a resposta do sistema ultrassônico em função do meio de propagação e de seus parâmetros de configuração, o que está diretamente relacionado com seu desempenho.

A literatura propõe soluções comerciais e não comerciais para a simulação do campo acústico gerado por transdutores. Em geral, os métodos de cálculo empregados baseiam-se na resposta impulsional. A desvantagem desses métodos é que a capacidade de cálculo exigida cresce muito com o aumento da complexidade dos problemas tratados, por exemplo, para o caso da propagação em meios múltiplos. Outras implementações oferecem soluções viáveis, porém que não contemplam esse tipo de problema.

O presente trabalho aborda uma solução baseada no cálculo da resposta de transdutores em excitação contínua, denominado *método MTM (Monochromatic Transfer Matrix)*. Tal escolha tem como objetivo a realização de simulações com menor tempo de cálculo quando comparada com métodos baseados na resposta impulsiva, além da possibilidade de tratar o problema da propagação em diferentes meios separados por uma interface.

Deseja-se desenvolver uma ferramenta computacional que trate problemas próximos de aplicações práticas e que ao mesmo tempo seja simples para a compreensão dos fenômenos físicos envolvidos. Em aplicações médicas, por exemplo, os tecidos do corpo humano possuem densidades semelhantes entre si e próximas à da água. Dessa forma, pode-se considerar a aproximação de que as ondas propagam-se em um meio homogêneo e isotrópico, Cheeke (2002). Por outro lado, em aplicações industriais, como no caso de inspeções, observa-se a propagação entre meios distintos através de uma interface, como na figura 1.8.

Além disso, observa-se que as ferramentas abertas existentes muitas vezes limitam-se a fornecer bibliotecas utilizáveis apenas a partir de comandos em linha ou com interfaces



**Figura 1.8:** Propagação do campo acústico através de uma interface, (IBÁÑEZ; VILLAZÓN, 2006).

poluídas e limitadas.

Nesse sentido, a solução proposta visa ao tratamento de situações como as descritas anteriormente. Os programas desenvolvidos são reunidos em torno de uma ferramenta gráfica e a implementação realizada também tem por objetivo permitir que o usuário tenha acesso a um método eficaz e que aumente a dinâmica de trabalho na análise de problemas através simulações interativas.

Devido à sua grande versatilidade e facilidade para manipulação e cálculo utilizando matrizes, a implementação é realizada em *MATLAB*®. Atenta-se para o fato de que o método *MTM* é inteiramente baseado em operações matriciais. Além disso, o software oferece soluções eficientes para a integração dos modelos por meio de interfaces gráficas, ao contrário de outras linguagens de programação.

## 1.4 Objetivos

O presente trabalho aborda o estudo do campo acústico gerado por diferentes tipos de transdutores, com enfoque no desenvolvimento de uma ferramenta computacional que possibilite a simulação desses sistemas em diferentes configurações de interesse, das mais simples àquelas com um grau de complexidade maior, permitindo a análise da propagação do campo acústico em meio simples e compostos.

Diferentes tipos de transdutores são estudados: transdutor circular monoelemento, transdutor *array* linear e transdutor côncavo esférico. É dada ênfase em quatro situações de interesse: propagação em meios simples, propagação em meios compostos separados por uma interface de formato arbitrário, determinação de leis de atraso de transdutores *array* e determinação de diferentes modos de propagação de ondas em materiais sólidos,

contribuindo para a análise e compreensão dos fenômenos físicos envolvidos.

A simulação do campo acústico pode ser realizada através de diferentes métodos. Será considerada a solução obtida para o caso de excitações contínuas, implementado através do *MTM*. Em comparação com outros métodos utilizados, como diferenças finitas, elementos finitos e resposta impulsiva, essa solução representa uma maneira rápida e eficaz de determinar o campo acústico para problemas complexos, que muitas vezes envolvem a propagação em diferentes meios separados por uma interface.

Dessa forma, deseja-se uma ferramenta que seja de fácil utilização e que permita a determinação do campo acústico para diferentes tipos de transdutores de maneira rápida e confiável, para problemas que envolvam a propagação em meios simples ou meios compostos. A solução desenvolvida tem por objetivo oferecer ao usuário final não somente um método eficaz, mas também que permita a realização de simulações interativas, aumentando a dinâmica de trabalho e oferecendo análises consistentes.

## 1.5 Revisão bibliográfica

Em linhas gerais, a revisão bibliográfica está dividida em três grupos. Ela inclui trabalhos que enfatizam aspectos relativos às aplicações do ultrassom, principalmente nas áreas médicas e industrial, de caráter prático. Além disso, são citados trabalhos voltados para a teoria de acústica e que possibilitam o entendimento dos principais fenômenos físicos envolvidos. Finalmente, é realizado um breve estado da arte de métodos computacionais utilizados para a simulação do campo acústico, com enfoque nas técnicas utilizadas para implementação do *MTM* e resolução dos problemas propostos. Também são citados exemplos de aplicações desse método em específico, no desenvolvimento de sistemas ultrassônicos.

O trabalho de Fish (1990) discute de maneira sucinta e objetiva a base da tecnologia de ultrassom utilizada na área médica. Aspectos teóricos são abordados de maneira relativamente simples, dando enfoque a conceitos, aplicações e a constituição de sistemas de ultrassom para formação de imagens em modo A, B e C. Na área industrial, os trabalhos de Kass, Nelligan e Henjes (2006), Drinkwater e Wilcox (2006) dão enfoque a importância de transdutores *arrays* em aplicações voltadas para ensaios não destrutivos. O primeiro aponta os benefícios dessa tecnologia para a qualidade das atividades de inspeção e ressalta sua grande evolução ao longo dos últimos anos. O segundo trabalho realiza um estado de

arte das principais publicações referentes a transdutores array, abordando aspectos como dimensionamento, modelagem e processamento de sinais.

Os principais conceitos relacionados a teoria acústica são tratados em obras de três autores. Os trabalhos de Cheeke (2002) e Kuttruff (1991) apresentam a teoria acústica de maneira clara e simples, tratando inicialmente questões gerais, como teoria ondulatória e propagação de ondas mecânica e gradativamente introduz problemas mais complexos, cuja relação com aplicações de ultrassom são mais estreitas. As obras de Kinsler (2000) e Kino (1987) são textos mais densos, que tratam aspectos teóricos com maior nível de detalhe e abordam a modelagem de certas aplicações, fornecendo demonstrações detalhadas de algumas soluções analíticas existentes.

Em relação a métodos numéricos para a simulação de transdutores ultrassônicos, mencionam-se três trabalhos que em certa medida influenciaram o desenvolvimento do presente trabalho. Em primeiro plano, está o software comercial *CIVA*®, ao qual o autor teve acesso para a realização de simulações de comparação de resultados, desenvolvido pelo órgão francês *Commissariat à l'énergie atomique et aux énergies alternatives* (CEA) para a modelagem e simulação de campos acústicos e descrito por Calmon et al. (2006).

Em segundo plano, estão dois trabalhos de caráter aberto, o *toolbox DREAM* (*Discrete Representation Array Modelling*) e o *toolbox FOCUS* (*Fast Object Oriented C++ Ultrasound Simulator*), descritos respectivamente por Piwakowski e Sbai (1999) e Chen e McGough (2008), ambos desenvolvidos em *MATLAB*® e que possuem funcionalidades semelhantes. De maneira mais ampla, o trabalho de Huttunen et al. (2005) cita diversas obras que implementam outros métodos distintos, apontando suas vantagens e desvantagens.

Os métodos de cálculo propostos por Calmon et al. (2006) e Piwakowski e Sbai (1999) baseiam-se ambos no cálculo da resposta impulsional de transdutores, conforme apresentado por Robinson, Lees e Bess (1974), para um transdutor monoelemento circular. Já Chen e McGough (2008) implementam o cálculo do campo acústico a partir do método denominado *FNM* (*Fast Nearfield Method*), que utiliza técnicas baseadas transformada de Fourier.

De maneira geral, todos os trabalhos apresentam soluções viáveis para o problema de simulação computacional. Este trabalho pretende oferecer uma alternativa a esses métodos, seja visando a melhoria de certos aspectos, como tempo de cálculo, que pode

se tornar proibitivo em certos casos, seja abordando certos aspectos que não são tratados nas ferramentas abertas, como a propagação em meios múltiplos.

Isso é possível através da utilização do método *MTM*, cuja base teórica é inspirada do trabalho de Goodman (2005) e o desenvolvimento é realizado por Ibáñez e Villazón (2006) e Ibáñez et al. (2010). O método proposto é perfeitamente adaptado para o estudo da propagação de ondas em meios simples, múltiplos e o estudo da focalização de transdutores *arrays*. Esse último ponto é desenvolvido com base em técnicas de cálculo rápido das leis de atraso, propostos por Parrilla et al. (2008), a partir de métodos computacionalmente eficientes que permitem um tempo de cálculo compatível com uma aplicação em tempo real.

Por fim, citam-se trabalhos que aplicam e exemplificam o *MTM*. No primeiro, Andrade et al. (2009) compara a solução obtida a partir do *MTM* e do software comercial *CIVA*® para um mesmo problema de inspeção em componentes mecânicos, no contexto do estudo de correções na determinação das leis de atraso para focalização. A solução obtida com o *MTM* é da ordem de  $10h$  menor em tempo de cálculo. Em outro trabalho, Andrade et al. (2011) apresenta o desenvolvimento de uma técnica de levitação por ultrassom com base no *MTM*. Finalmente, o trabalho de Degaspare e Buiochi (2009) confronta resultados experimentais com simulações a partir do *MTM*, para o caso da propagação em meios simples, tratando o caso de um transdutor circular plano imerso em água.

## 2 DETERMINAÇÃO NUMÉRICA DO CAMPO ACÚSTICO

A literatura propõe diferentes métodos para a determinação do campo acústico de transdutores. Em geral, métodos convencionais baseados em diferenças finitas e elementos finitos requerem uma discretização espacial muito fina, exigindo uma capacidade de cálculo computacional proibitiva, Huttunen et al. (2005).

Sendo assim, métodos alternativos são utilizados, baseando-se na resposta impulsional, que leva em conta a existência de transientes e permite a determinação do campo acústico em um meio homogêneo e isotrópico em função do tempo, Calmon et al. (2006). Tais métodos tornam-se lentos quando o cálculo envolve a interação entre diferentes meios de propagação através de interfaces complexas. Nesses casos, métodos do tipo *ray-tracing* podem ser aplicados, porém, com relativa dificuldade e sem ganhos no tempo de cálculo exigido.

A simulação do campo acústico de maneira rápida e eficiente é obtida através da utilização de métodos matriciais baseados na matriz de transferência monocromática, Ibáñez e Villazón (2006) e Ibáñez et al. (2010). Em virtude de sua eficácia e versatilidade em comparação com outros métodos, esse método é preferido para a simulação do campo acústico que requer rapidez. Como exemplos de aplicação, citam-se Andrade et al. (2009) e Andrade et al. (2011). A seguir, descreve-se a base do método que será implementado na sequência do trabalho. A formulação teórica do problema é obtida a partir de Cheeke (2002), Kinsler (2000), Kino (1987), Kuttruff (1991) e Goodman (2005).

Discute-se a propagação de ondas em fluidos e sólidos. No caso de fluidos, considera-se um meio ideal: invíscido, isotrópico, homogêneo e perfeitamente elástico. Sólidos também serão considerados meios homogêneos e isotrópicos. Em seguida, apresenta-se o método que será empregado, assim como questões relacionadas à propagação em meios simples, compostos e a focalização de transdutores *arrays*.

## 2.1 Propagação em líquidos

A propagação de ondas em líquidos difere do caso da propagação de ondas em sólidos principalmente pela existência de um único modo de propagação. Em fluidos, observa-se apenas a propagação de ondas de compressão, o que corresponde ao modo de propagação longitudinal.

Dessa forma, o campo acústico será descrito em termos da pressão  $p$  no fluido. Em virtude da existência de apenas um modo de propagação, não há nenhum tipo de polarização e pode-se aplicar a teoria de ondas escalar.

### 2.1.1 Ondas planas em três dimensões

Seja um elemento infinitesimal cuja área é dada por  $dA$ , o volume por  $dV$  e o deslocamento por  $\vec{u}$ . Aplicando o teorema de Gauss, pode-se relacionar o deslocamento com a dilatação do elemento, dada por  $S(\vec{r})$ . A partir da equação 2.1, pode-se derivar a equação que relaciona essas duas grandezas, dada em 2.4.

$$\nabla^2 V = \oint_S \vec{u} \cdot \vec{n} dA = \int_V (\nabla \cdot \vec{u}) dV = \int_V S(\vec{r}) dV \quad (2.1)$$

onde  $\nabla$  corresponde ao operador diferencial dado em coordenadas cartesianas em 2.2:

$$\nabla = \hat{x} \frac{\partial}{\partial x} + \hat{y} \frac{\partial}{\partial y} + \hat{z} \frac{\partial}{\partial z} \quad (2.2)$$

e  $\nabla^2$  corresponde ao operador laplaciano dado em coordenadas cartesianas:

$$\nabla^2 = \frac{\partial}{\partial x^2} + \frac{\partial}{\partial y^2} + \frac{\partial}{\partial z^2} \quad (2.3)$$

A dilatação será dada por:

$$S(\vec{r}) = \nabla \cdot \vec{u} = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} \quad (2.4)$$

A lei de Newton aplicada ao elemento resulta em:

$$\rho_0 \frac{\partial^2 \vec{u}}{\partial t^2} = -\vec{\nabla} p \quad (2.5)$$

onde  $\rho_0$  é a densidade do meio de propagação.

Utilizando a relação 2.4, tem-se:

$$\rho_0 \frac{\partial^2 S}{\partial t^2} = -\nabla^2 p \quad (2.6)$$

A compressibilidade do fluido é definida por  $\chi$  e relaciona-se com a pressão e a dilatação:

$$p = -\frac{S}{\chi} \quad (2.7)$$

Aplicando 2.7 em 2.6, obtém-se a forma final da equação de onda:

$$\nabla^2 p = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} \quad (2.8)$$

onde  $c^2 = \frac{1}{\rho_0 \chi}$  corresponde à velocidade de propagação da onda no fluido isotrópico tal que  $c = \lambda f$ , onde  $\lambda$  é o comprimento de onda e  $f$  a frequência.

A velocidade  $\vec{v}$  da partícula associada a uma onda plana longitudinal em um meio homogêneo e isotrópico é obtida a partir de um potencial  $\phi$ , tal que  $\vec{v} = \nabla \phi$ . Essa aproximação é possível uma vez que se considera o descolamento das partículas como irrotacional,  $\vec{\nabla} \times \vec{v} = 0$ .

A pressão acústica  $p$  pode ser escrita em termos do potencial  $\phi$  a partir de 2.9:

$$p = -\rho_0 \frac{\partial \phi}{\partial t} \quad (2.9)$$

Utilizando uma notação complexa, podemos escrever a solução para a equação de onda:

$$p = A \exp j(\omega t - \vec{k} \cdot \vec{r} + \gamma) \quad (2.10)$$

onde  $A$  corresponde à amplitude,  $j$  indica o número complexo  $j = \sqrt{-1}$ ,  $\omega$  é a velocidade angular e  $\gamma$  uma fase inicial.

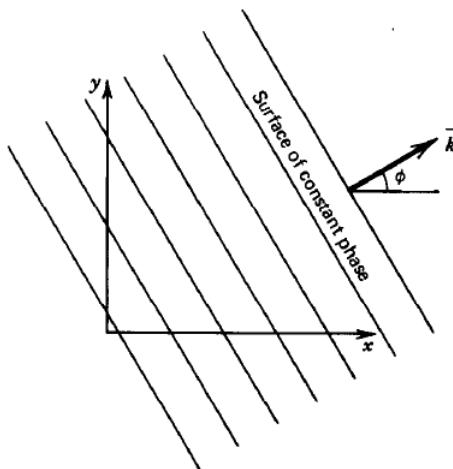
Comumente, adota-se a forma da solução com fase nula, dada em 2.11:

$$p = A \exp j(\omega t - [k_x x + k_y y + k_z z]) \quad (2.11)$$

A amplitude  $A$  é determinada a partir de uma condição de contorno. Já a amplitude de  $\vec{k}$  é dada por:

$$k = |\vec{k}| = \frac{2\pi}{\lambda} \quad (2.12)$$

A onda propaga-se na direção do vetor  $\vec{k} = (k_x, k_y, k_z)$ , onde  $k$  indica o número de onda em uma determinada direção. A frente de onda corresponde ao lugar geométrico dos pontos de fase constante em um instante  $t$ , o que corresponde a uma série de planos paralelos conforme figura 2.1.



**Figura 2.1:** Lugar geométrico dos pontos de fase constante, (KINSLER, 2000).

Seja  $\vec{r}$  o vetor posição traçado da origem do sistema de coordenadas até um ponto da frente de onda num dado instante  $t$ . Logo, a equação da frente de onda é dada por 2.13:

$$\vec{k} \cdot \vec{r} = constante \quad (2.13)$$

A equação 2.8 também pode ser escrita em termos do deslocamento. Ela é estendida

na forma:

$$\nabla^2 \vec{u} = \frac{1}{c^2} \frac{\partial^2 \vec{u}}{\partial t^2} \quad (2.14)$$

e cuja solução é da forma dada em 2.15

$$\vec{u} = \vec{u}_0 \exp j(\omega t - \vec{k} \cdot \vec{r}) \quad (2.15)$$

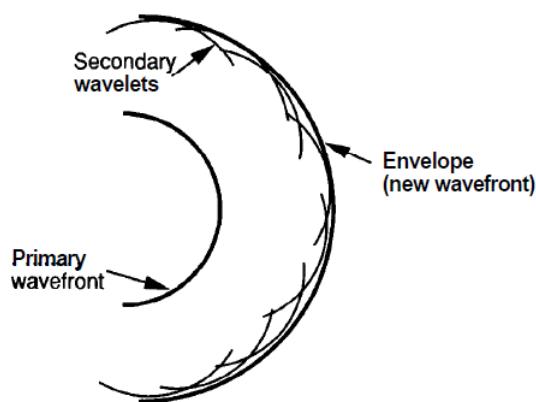
Outras relações recorrentes são dadas em 2.16

$$\begin{cases} f &= \omega/2\pi \\ T &= 1/f \\ c &= \omega/k \end{cases} \quad (2.16)$$

onde  $T$  é o período e  $c$  a velocidade de fase da onda.

### 2.1.2 Ondas esféricas emitidas por fontes pontuais

O método computacional implementado é baseado no princípio de Huygens, ilustrado na figura 2.2. A ideia intuitiva é que cada ponto da frente de onda é tratado como uma fonte pontual secundária, que emite ondas esféricas. Dessa forma, a frente de onda em um instante posterior pode ser calculada a partir do envelope de fontes secundárias.



**Figura 2.2:** Princípio de Huygens, (GOODMAN, 2005).

Em um primeiro momento, será tratado o caso de fontes pontuais, que posteriormente será expandido para o caso de pistões de diferentes formas geométricas.

Seja a equação de onda escrita em 2.8. O laplaciano em coordenadas esféricas é dado em 2.17:

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{2}{r} \frac{\partial}{\partial r} + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left( \sin \theta \frac{\partial}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2}{\partial \varphi^2} \quad (2.17)$$

As ondas possuem simetria radial, o que faz com que a pressão  $p$  seja apenas função da distância radial  $r$ . A equação 2.17 é simplificada para 2.18:

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{2}{r} \frac{\partial}{\partial r} \quad (2.18)$$

A equação da onda é reescrita na forma apresentada em 2.19

$$\frac{\partial^2 p}{\partial r^2} + \frac{2}{r} \frac{\partial p}{\partial r} - \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} = 0 \quad (2.19)$$

Realizando uma mudança de variáveis, como em Kinsler (2000), pode-se reescrever 2.19 em função de uma variável cuja amplitude independe de  $r$ , denotada por  $rp$ . O resultado é a equação da forma:

$$\frac{\partial^2 (rp)}{\partial r^2} - \frac{1}{c^2} \frac{\partial^2 (rp)}{\partial t^2} = 0 \quad (2.20)$$

cuja solução geral é da mesma forma que para uma onda plana:

$$p = \frac{1}{r} f_1(ct - r) + \frac{1}{r} f_2(ct + r) \quad (2.21)$$

para todo valor de  $r > 0$ . Ela consiste em uma combinação linear de uma solução divergente e outra convergente, respectivamente. A solução é singular para  $r = 0$ .

Serão consideradas as soluções de onda esférica divergentes e harmônica. Tem-se que:

$$p = \frac{A \exp j(\omega t - kr)}{r} \quad (2.22)$$

onde o sinal  $-$  indica que a onda propaga-se da origem da fonte pontual para fora (sentido divergente), na direção radial.

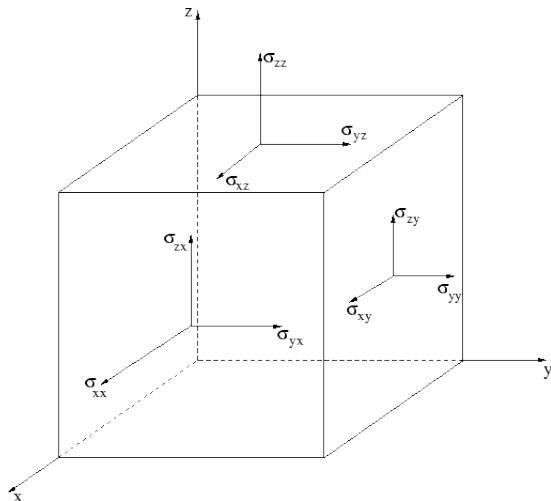
## 2.2 Propagação de ondas em sólidos

Sólidos suportam forças nas direções normal e tangencial. Isso ocorre devido à tendência dos mesmos em manter o seu volume e sua forma, ao contrário dos líquidos. As forças e deslocamentos são descritos a partir de tensores. A teoria é desenvolvida em detalhes em Cheeke (2002), Kuttruff (1991) e Kino (1987).

As forças exercidas são caracterizadas por um tensor de tensão, conforme equação 2.23:

$$\sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix} \quad (2.23)$$

As componentes desse tensor possuem dimensão de força por unidade de área. A figura 2.3 ilustra essas componentes em um elemento cúbico.



**Figura 2.3:** Tensor de tensão.

Da mesma maneira, defini-se o tensor de deformação do sólido, denotado por  $\varepsilon$ , conforme a equação 2.24:

$$\varepsilon = \begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{pmatrix} \quad (2.24)$$

As componentes do tensor de deformação correspondem a relações lineares entre as

derivadas de componentes do deslocamento  $\vec{u}$ . Ele pode ser escrito como:

$$\varepsilon = \begin{pmatrix} \frac{\partial u_x}{\partial x} & \frac{1}{2} \left( \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right) & \frac{1}{2} \left( \frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x} \right) \\ \frac{1}{2} \left( \frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y} \right) & \frac{\partial u_y}{\partial y} & \frac{1}{2} \left( \frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y} \right) \\ \frac{1}{2} \left( \frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z} \right) & \frac{1}{2} \left( \frac{\partial u_z}{\partial y} + \frac{\partial u_y}{\partial z} \right) & \frac{\partial u_z}{\partial z} \end{pmatrix} \quad (2.25)$$

Em sólidos, observam-se diferentes modos de propagação. Isso pode resultar em grande complexidade na análise do problema, principalmente em materiais anisotrópicos. Dessa forma, consideram-se os casos de sólidos homogêneos e isotrópicos.

As equações de onda associadas podem ser escritas como em 2.26. Uma apresentação mais detalhada da dedução dessas equações não engloba o escopo do presente trabalho. As referências citadas no início dessa seção apresentam em grande detalhe esse desenvolvimento.

$$\begin{cases} \mu \nabla^2 u_x + (\mu + \lambda) \frac{\partial \vartheta}{\partial x} = \rho_0 \frac{\partial^2 u_x}{\partial t^2} \\ \mu \nabla^2 u_y + (\mu + \lambda) \frac{\partial \vartheta}{\partial y} = \rho_0 \frac{\partial^2 u_y}{\partial t^2} \\ \mu \nabla^2 u_z + (\mu + \lambda) \frac{\partial \vartheta}{\partial z} = \rho_0 \frac{\partial^2 u_z}{\partial t^2} \end{cases} \quad (2.26)$$

onde  $\mu$  e  $\lambda$  são as constantes elásticas de Lamé, e  $\vartheta$  corresponde à expressão:

$$\vartheta = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} \quad (2.27)$$

Seja uma perturbação que se propaga como uma onda plana na direção  $x$ . As equações são simplificadas e reescritas da seguinte forma:

$$\begin{cases} (2\mu + \lambda) \frac{\partial^2 u_x}{\partial x^2} = \rho_0 \frac{\partial^2 u_x}{\partial t^2} \\ \mu \frac{\partial^2 u_y}{\partial x^2} = \rho_0 \frac{\partial^2 u_y}{\partial t^2} \\ \mu \frac{\partial^2 u_z}{\partial x^2} = \rho_0 \frac{\partial^2 u_z}{\partial t^2} \end{cases} \quad (2.28)$$

As relações em 2.28 indicam a existência de três modos de propagação perpendiculares. O primeiro deles corresponde ao modo longitudinal, descrito por ondas de compressão, como na figura 2.4(a).

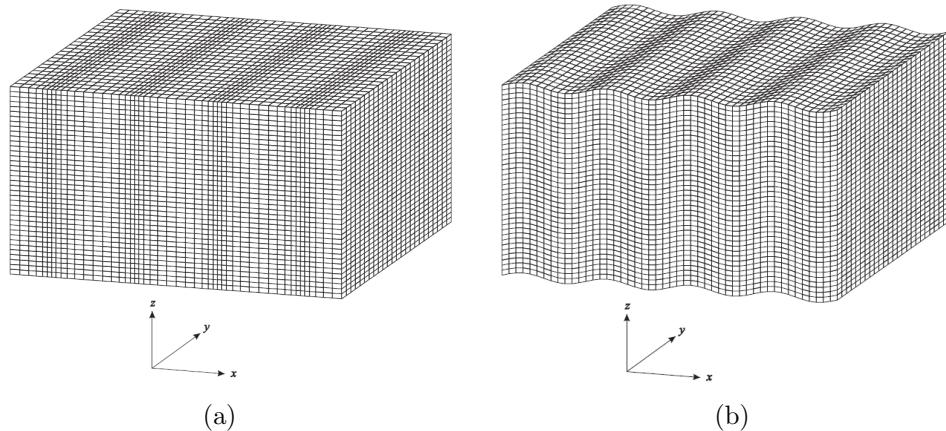
A velocidade de propagação para esse modo é dada por:

$$c_l^2 = \frac{2\mu + \lambda}{\rho_0} \quad (2.29)$$

Já os outros dois modos correspondem à propagação transversal das ondas, como indicado na figura 2.4(b). A direção de propagação é perpendicular ao deslocamento e a velocidade é dada por:

$$c_s^2 = \frac{\mu}{\rho_0} \quad (2.30)$$

Esse modo será denominado pelo índice  $s$ , em virtude do termo em inglês *shear wave*.



**Figura 2.4:** Diferentes modos de propagação em um sólido, (CHEEKE, 2002) (a) Ondas de compressão, modo longitudinal (b) Ondas de cisalhamento, modo transversal, .

Isso dá origem a diversos fenômenos de polarização observados em sólidos. Para o presente estudo, enfatiza-se que a descrição do modo de propagação longitudinal utilizado para um fluido também é adequada para o modo longitudinal de um sólido.

Assim, a solução da equação de onda utilizada para um fluido e um sólido é a mesma. Ela difere apenas no valor empregado da velocidade de propagação. Em função da análise do modo longitudinal ou transversal, utilizar-se-á respectivamente  $c_l$  ou  $c_s$  na determinação do campo acústico.

## 2.3 Determinação do campo acústico

Seja o caso mais geral de um campo escalar  $u$  (pressão acústica, potencial de velocidade, densidade, etc) que descreva o campo acústico gerado por um transdutor. A perturbação em um ponto  $P$  do espaço em um dado instante  $t$  é expressa pela função escalar  $u(P, t)$ .

Como descrito anteriormente, a solução geral que obedece a equação de onda é dada por 2.31:

$$u(P, t) = \operatorname{Re} \{U(P) \exp(-j\omega t)\} \quad (2.31)$$

onde se considera a parte real da solução dada em notação complexa e  $U(P)$  é uma função complexa da posição tal que:

$$U(P) = A(P) \exp(-j\gamma(P)) \quad (2.32)$$

A solução 2.31 verifica a equação de onda que descreve o problema:

$$\nabla^2 u - \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} = 0 \quad (2.33)$$

Substituindo 2.31 em 2.33, obtém-se a equação de Helmholtz:

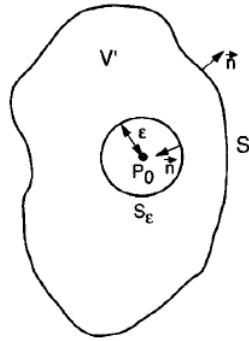
$$\nabla^2 U + k^2 U = 0 \quad (2.34)$$

onde  $k$  é o número de onda.

A função complexa  $U$  é independente do tempo e pode ser usada para descrever a perturbação em regime permanente. Como o método desenvolvido considera o caso de excitação contínua, ela é adequada para a descrição do campo acústico no espaço.

A função  $U$  é determinada a partir do teorema de Green, Goodman (2005). A figura 2.5 ilustra um caso de aplicação do teorema. Seja o ponto de observação  $P_0$ , uma superfície fechada arbitrária  $S$  ao redor de  $P_0$ , que delimita um volume  $V'$ . Deseja-se determinar a perturbação causada por  $P_0$  em termos da superfície  $S$ .

Seja  $G(P)$  uma função complexa auxiliar, que expressa o valor da perturbação



**Figura 2.5:** Aplicação do teorema de Green, (GOODMAN, 2005).

em um ponto arbitrário  $P_1$ . Supõe-se que  $P_0$  é uma fonte pontual esférica de amplitude unitária e raio  $\epsilon$ . O valor de  $G(P_1)$  é <sup>1</sup>:

$$G(P_1) = \frac{\exp(jkr_{01})}{r_{01}} \quad (2.35)$$

A função  $G$  é denominada função de Green do espaço livre (*Free space Green's function*).

O teorema de Green estabelece que, dadas duas funções  $U(P)$  e  $G(P)$  de classe  $C^2$  em uma superfície fechada  $S$  em torno de um volume  $V'$ , então

$$\int_{V'} (U \nabla^2 G - G \nabla^2 U) dv = \int_S \left( U \frac{\partial G}{\partial n} - G \frac{\partial U}{\partial n} \right) ds \quad (2.36)$$

onde  $\frac{\partial}{\partial n}$  denota a derivada na direção da normal externa à  $S$ .

A demonstração completa da aplicação do teorema de Green é dada em Kino (1987) e Goodman (2005). A solução obtida é dada em 2.37:

$$U(P_0) = \frac{1}{4\pi} \int_S \left( G \frac{\partial U}{\partial n} - U \frac{\partial G}{\partial n} \right) ds \quad (2.37)$$

Para que 2.37 seja válida, a condição de radiação de Sommerfield deve ser satisfeita: a medida que  $r \rightarrow \infty$  os dois termos da integral se cancelam e as contribuições da superfície vindas do infinito podem ser ignoradas.

A função  $U$ , dada por 2.37 e que descreve o campo acústico no espaço, será a base

<sup>1</sup>Para detalhes na escolha de  $G$ , recomendam-se as referências Kino (1987) e Goodman (2005)

para o desenvolvimento do método numérico desenvolvido na sequência.

## 2.4 Método matricial para determinação do campo acústico

O método para o cálculo numérico do campo acústico segue as referências Ibáñez e Villazón (2006) e Ibáñez et al. (2010). Fisicamente, a função  $U$  representa o potencial de velocidade  $\phi$ .

Dois tipos de condições de contorno são aplicadas à equação 2.37.

1. Condição de Dirichlet:  $U = 0$

2. Condição de Neumann:  $\frac{\partial U}{\partial n} = 0$

O problema consiste em determinar a pressão acústica emitida por um transdutor plano cercado por um defletor infinito em um meio de propagação homogêneo. Fazendo uma escolha apropriada da função de Green, Goodman (2005) deduz que, para as condições 1 e 2, o campo de pressão é caracterizado a partir da equação 2.38:

$$U(P_0) = \begin{cases} \frac{-jk}{2\pi} \int_{\Sigma} \frac{\exp(-jkr)}{r} \cdot \cos(\alpha) \cdot U(r) ds, & \text{condição de Dirichlet} \\ \frac{1}{2\pi} \int_{\Sigma} \frac{\exp(-jkr)}{r} \cdot \frac{\partial U(r)}{\partial n} ds, & \text{condição de Neumann} \end{cases} \quad (2.38)$$

onde  $\Sigma$  é a superfície de irradiação,  $\vec{r}$  é o vetor de  $P_0$  até um ponto de  $\Sigma$ ,  $\alpha$  é o ângulo definido entre  $\vec{r}$  e  $\vec{n}$  e  $U(r)$  é a função de excitação do transdutor.

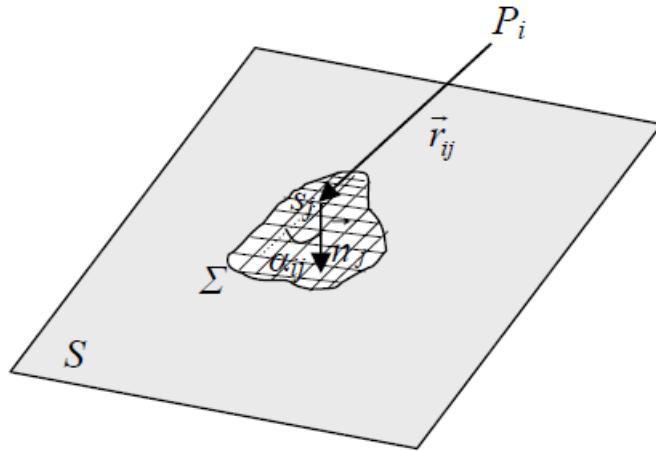
A interpretação física das condições de contorno corresponde às condições impostas ao defletor. A primeira condição corresponde ao problema do tipo *pressure release baffle*, que considera o potencial de velocidade nulo no defletor. Já a segunda condição trata o caso do defletor rígido, *rigid baffle*, que equivale à hipótese de um pistão plano rígido, cujo defletor possui velocidade nula na superfície.

Uma forma geral da equação 2.38 é dada em 2.39.

$$U(P_0) = \int_{\sigma} h(P_0, r) \cdot E(r) ds \quad (2.39)$$

onde  $E(r)$  representa a função de excitação.

Supõe-se que a superfície do transdutor é composta por  $n$  células de área  $s_j$ , suficientemente pequenas para que  $h(P_0, r) = h_j(P_0)$  e que  $E(r) = E_j$  seja constante para todos os elementos.



**Figura 2.6:** Cálculo discreto de  $U$  no ponto  $P_0$ , (IBÁÑEZ; VILLAZÓN, 2006).

Assim, a integral pode ser substituída pela soma das contribuições de todas as superfícies no ponto  $P_0$ :

$$U(P_0) = \sum_j h_j \cdot E_j \cdot s_j = \sum_j m_j(P_0) \cdot E_j \quad (2.40)$$

onde  $m_j(P_0) = m_{0j} = h_j \cdot s_j$  e o infinitésimo de área  $dS$  é substituído pela área  $s_j$  de cada elemento.

Seja um conjunto de  $m$  pontos  $P_i$  do espaço discretizado, onde se deseja determinar o campo acústico. Defini-se  $\mathbf{U}$  o vetor cujos valores representam a amplitude complexa do campo acústico em cada ponto desse conjunto. Igualmente, o vetor  $\mathbf{E}$  representa as amplitudes complexas de excitação de cada elemento  $j$  do transdutor definido por  $\Sigma$ . O módulo de um elemento de  $\mathbf{E}$  representa a amplitude de excitação e a fase corresponde à fase de excitação do elemento  $j$ .

Finalmente, o campo acústico é determinado pelo método denominado Matriz de Transferência Monocromática (*Monochromatic Transfer Matrix, MTM*). A solução é dada

a partir do sistema de equações escrito em notação matricial em 2.41:

$$\mathbf{U} = \mathbf{T}\mathbf{E} \quad (2.41)$$

onde os elementos da matriz de transferência  $\mathbf{T}$  de dimensão  $m \times n$  correspondem à:

$$m_{ij} = h_j(P_i) \cdot s_j = \begin{cases} \frac{s_j}{2\pi} \frac{\exp(-jkr_{ij})}{r_{ij}}, & \text{condição de Neumann} \\ \frac{-jks_j}{2\pi} \frac{\exp(-jkr_{ij})}{r_{ij}} \cos(\alpha_{ij}), & \text{condição de Dirichlet} \end{cases} \quad (2.42)$$

A equação 2.42 constitui a base do método computacional utilizado para determinação do campo acústico de transdutores em modo de excitação contínua. A forma explícita dessa equação é dada em 2.43.

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{m1} & m_{m2} & \cdots & m_{mn} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \quad (2.43)$$

## 2.5 Casos de estudo

O método descrito será implementado para o tratamento de dois casos de estudo. Em primeiro lugar, será tratado o caso de propagação de uma perturbação um meio simples. Em seguida, será tratado o caso da propagação em dois meios distintos separados por uma interface. Também é de interesse o estudo da focalização de transdutores *arrays*. Por fim, discutem-se métodos para o cálculo das leis de atraso em função do problema tratado.

### 2.5.1 Meio de propagação simples

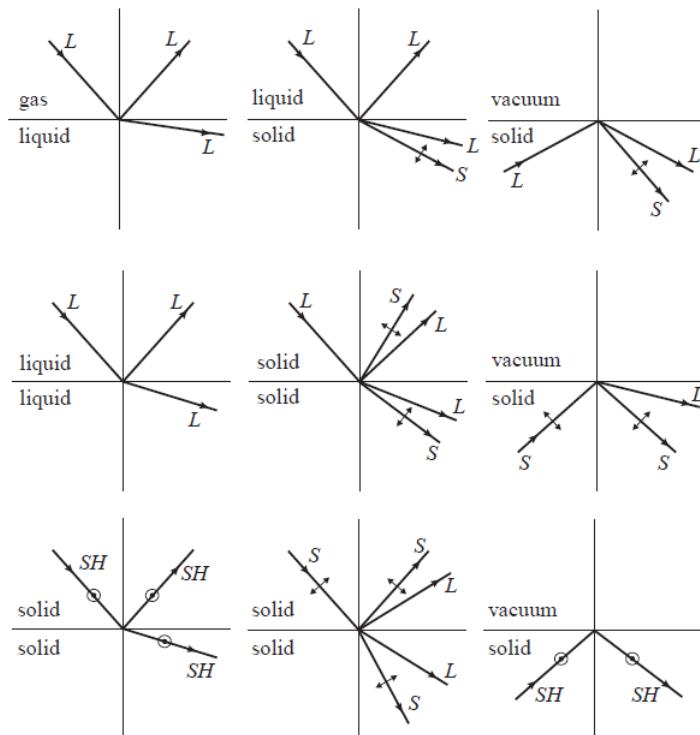
A formulação do MTM para o caso da propagação em meios simples corresponde a uma aplicação direta das formas dadas em 2.42 e 2.43. O estudo da propagação em um único meio permite explorar de maneira simples aspectos de implementação do método para que seja realizada sua extensão para casos mais complexos, seja envolvendo um meio composto, seja envolvendo o estudo de focalização de transdutores array e será rediscutido

na sequência do trabalho.

### 2.5.2 Meio de propagação composto

De acordo com Cheeke (2002), a propagação de ondas em meios compostos envolve diferentes casos que podem ser divididos em função dos meios envolvidos e do tipo de incidência, conforme ilustrado na figura 2.7. Os casos de maior importância são resumidos a seguir:

1. Fluido-fluido: caso mais simples de transmissão entre dois meios e extremamente recorrente em aplicações de diagnóstico médico;
2. Fluido-sólido: caso de extrema importância nas aplicações de ultrassom na área de testes não destrutivos;
3. Sólido-sólido, modos SH;
4. Sólido-vácuo: caso de importância no estudo de guias acústicas.



**Figura 2.7:** Casos de propagação entre dois meios, colocando em evidência a existência de diferentes modos de propagação em sólidos, (CHEEKE, 2002).

No presente trabalho, é dado ênfase ao caso 2. Conforme já discutido, chama-se a atenção para a existência de dois modos distintos de propagação, o modo transversal ( $s$ ) e o modo longitudinal ( $l$ ). A seguir, é descrita a extensão do *MTM* para esse caso. O estudo da propagação através de uma interface também é denominado conversão de modo.

Na notação utilizada, o índice  $i$  indica incidência,  $r$  reflexão,  $l$  modo de propagação longitudinal e  $s$  modo de propagação transversal. Os meios são designados pelos índices 1 e 2, onde o primeiro é um fluido e o segundo um sólido. Considerando o caso da figura 2.7, que é restrito ao plano  $xz$ , três condições de contorno devem ser verificadas na interface:

1. Continuidade de velocidades na direção normal
2. Continuidade da tensão na direção normal
3. Tensão na direção tangencial nula <sup>2</sup>

O detalhamento dos cálculos é dado nas obras de Kino (1987) e Cheeke (2002). As condições de contorno na interface levam à dedução da lei de Snell na forma apresentada em 2.44:

$$\frac{\sin \alpha_i}{c_1} = \frac{\sin \alpha_r}{c_1} = \frac{\sin \alpha_l}{c_{2l}} = \frac{\sin \alpha_s}{c_{2s}} \quad (2.44)$$

Dessa forma,  $\alpha_i = \alpha_r$  e os demais ângulos são determinados facilmente conhecendo-se os valores de  $\alpha_i$ , da velocidade de propagação do líquido  $c_1$  e das velocidades  $c_{2l}$  e  $c_{2s}$  para os diferentes modos de propagação em um sólido.

As condições de contorno também permitem a determinação dos coeficientes de transmissão e reflexão envolvidos:

$$R = \frac{Z_l \cos^2 2\alpha_s + Z_s \sin^2 2\alpha_s - Z_1}{Z_l \cos^2 2\alpha_s + Z_s \sin^2 2\alpha_s + Z_1} \quad (2.45)$$

$$T_l = \frac{\rho_1}{\rho_2} \frac{2Z_l \cos 2\alpha_s}{Z_l \cos^2 2\alpha_s + Z_s \sin^2 2\alpha_s + Z_1} \quad (2.46)$$

$$T_s = -\frac{\rho_1}{\rho_2} \frac{2Z_s \sin 2\alpha_s}{Z_l \cos^2 2\alpha_s + Z_s \sin^2 2\alpha_s + Z_1} \quad (2.47)$$

---

<sup>2</sup>Considera-se o fluido invíscido, o que implica em tensão tangencial nula, já que há apenas o modo de propagação longitudinal.

onde

$$Z_1 = \frac{\rho_1 c_1}{\cos \alpha_i}, \quad Z_l = \frac{\rho_2 c_2 l}{\cos \alpha_l}, \quad Z_s = \frac{\rho_2 c_2 s}{\cos \alpha_s} \quad (2.48)$$

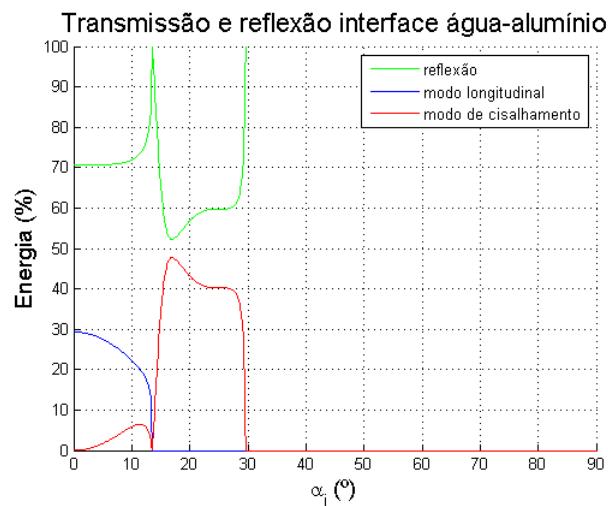
Os coeficientes podem ser reescritos em termos da porcentagem da energia transmitida e refletida, ou seja, em termos de coeficiente de potência. Sendo  $I$  a intensidade acústica total incidente na interface, temos, em função do ângulo de incidência  $\alpha$ :

$$\frac{I_r}{I} = |R(\alpha)|^2 \quad (2.49)$$

$$\frac{I_l}{I} = \frac{\rho_2 \tan \alpha}{\rho_1 \tan \alpha_l} |T_l(\alpha)|^2 \quad (2.50)$$

$$\frac{I_s}{I} = \frac{\rho_2 \tan \alpha}{\rho_1 \tan \alpha_s} |T_s(\alpha)|^2 \quad (2.51)$$

A figura 2.8 mostra a porcentagem de energia transmitida em função do ângulo de incidência para uma interface do tipo água-alumínio. Ela é obtida a partir da utilização das equações 2.49 a 2.51.



**Figura 2.8:** Energia transmitida e refletida para interface água-alumínio.

Observa-se claramente a existência de ângulos de incidência críticos,  $\alpha_{lc}$  e  $\alpha_{sc}$ , a partir dos quais há o fenômeno de reflexão total. Esses ângulos são determinados a partir da lei de Snell:

$$\alpha_{lc} = \sin^{-1} \frac{c_1}{c_{2l}} \quad (2.52)$$

$$\alpha_{sc} = \sin^{-1} \frac{c_1}{c_{2s}} \quad (2.53)$$

O campo acústico é apresentado graficamente em termos de amplitude. A intensidade acústica total é proporcional ao quadrado dessa grandeza. Assim, a amplitude transmitida relaciona-se com os coeficientes de potência da seguinte maneira:

$$\frac{I}{I_t} = \frac{A^2}{A_t^2} = t \Rightarrow A_t = \sqrt{t}A \quad (2.54)$$

onde  $t$  é um coeficiente de potência para transmissão e  $A$  é a amplitude.

## 2.6 Determinação das leis de atraso para um transdutor array

A focalização de um transdutor *array* é realizada a partir da introdução de diferentes fases de excitação em seus elementos. Em outras palavras, introduz-se um atraso da ordem de frações de segundos em cada elemento, de modo que seja permitida a criação de regiões do espaço onde a intensidade do campo acústico será superior do que nas demais. Entre outras vantagens, a utilização de leis de atraso em transdutores *arrays* permite acelerar a formação de imagens em aplicações de inspeção. Os métodos apresentados nesta seção, utilizados para a determinação das leis de atraso, decorrem do trabalho de Parrilla et al. (2008).

A determinação das leis de atraso é realizada com base em um princípio fundamental de geometria óptica, o princípio de Fermat. De um ponto de vista prático, esse princípio estabelece que o caminho percorrido entre dois pontos por um raio luminoso equivale ao caminho cuja duração é mínima.

Para o caso de um meio simples, homogêneo e isotrópico, a aplicação do princípio de Fermat é um método de aplicação relativamente simples e preciso.

Seja  $\bar{x}_f = (x_f, y_f, z_f)$  um ponto da região de interesse de determinação do campo acústico que corresponda ao foco do transdutor e seja  $\bar{x}_a = (x_a, y_a, z_a)$  o centro de um dos elementos do transdutor array. Determinam-se os tempos de propagação  $t_v$  para cada

elemento do transdutor a partir do cálculo:

$$t_v = \frac{|\bar{x}_a - \bar{x}_f|}{c} \quad (2.55)$$

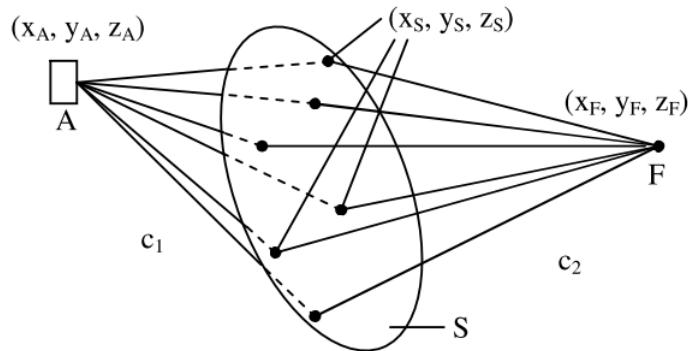
onde,  $c$  é a velocidade de propagação da onda no meio considerado.

Sendo  $T_v$  o vetor que representa os tempos de propagação para todos os elementos do array em relação ao ponto  $\bar{x}_f$ , os atrasos para um elemento  $A$  do array, para um dado foco  $F$ , são determinados pela equação 2.56:

$$\tau(A, F) = \max[T_v(A, F)] - T_v(A, F) \quad (2.56)$$

Isso corresponde simplesmente a atribuir o maior atraso ao elemento cujo tempo de propagação é menor em relação aos demais. Dessa forma, espera-se que as frentes de onda cheguem ao mesmo tempo no ponto definido por  $\bar{x}_f$ , estabelecendo uma concentração do campo acústico nessa região.

Quando se considera a existência de um meio de propagação composto separado por uma interface, a utilização do princípio de Fermat torna-se mais complexa. A existência de diferentes velocidades de propagação entre os dois meios faz com que seja necessário determinar-se a região de entrada da onda na interface, ou seja, deve-se determinar o tempo mínimo de propagação do elemento do transdutor à interface e em seguida da interface ao foco. A figura 2.9 ilustra a problemática.



**Figura 2.9:** Aplicação do princípio de Fermat para o caso de propagação em meio composto com uma interface de separação  $S$  entre os meios, (PARRILLA et al., 2008).

Diferentemente do caso anterior, diversos caminhos possíveis existem entre o ponto

$\bar{x}_a$  e  $\bar{x}_f$ . A aplicação do princípio de Fermat resulta em:

$$t_v = \frac{|\bar{x}_a - \bar{x}_s|}{c_1} + \frac{|\bar{x}_s - \bar{x}_f|}{c_2} \quad (2.57)$$

onde  $\bar{x}_s = (x_s, y_s, z_s)$  é um ponto da interface de separação entre os dois meios,  $c_1$  e  $c_2$  são as velocidades de propagação para os meios 1 e 2.

Computa-se o valor de  $t_v$  para todos os pontos  $\bar{x}_s$  e em seguida atribui-se a  $T_v$  o valor  $\min(t_v)$ , correspondente ao tempo mínimo de propagação segundo aplicação do princípio de Fermat. As leis de atraso são determinadas de maneira unívoca com base na equação 2.56.

No entanto, em função do problema tratado, a utilização desse método pode constituir uma solução lenta e exaustiva, fato que advém da necessidade de calcular o caminho de tempo mínimo entre  $\bar{x}_a$  e  $\bar{x}_f$  dentre todos os caminhos possíveis. No pior caso, para um transdutor linear de  $n$  elementos e uma interface  $S$  de  $m$  elementos, seriam necessárias  $n \times m$  iterações.

Para certas aplicações essa solução representa uma alternativa proibitiva, como é o caso da focalização dinâmica, onde se deseja determinar leis de atraso de maneira rápida para valores de  $\bar{x}_f$  que variam ao longo do tempo. As seguintes soluções alternativas são propostas com base em trabalhos disponíveis na literatura: Newton-Raphson e Nelder-Mead.

### 2.6.1 Algoritmo de Newton-Raphson

O trabalho de Parrilla et al. (2008) propõe um método de determinação das leis de atraso em presença de interfaces de formato arbitrário, cuja solução é obtida de maneira mais rápida e eficiente em comparação com a aplicação direta do princípio de Fermat.

O método considera a aproximação de um problema de geometria 2D, onde as leis de atraso são calculadas para a propagação no plano  $xz$  com  $y = 0$ . Ela é válida para o caso de transdutores array lineares e de interfaces sem mudanças abruptas de inclinação.

Trata-se o problema como um problema de otimização. Deseja-se determinar o ponto  $\bar{x} = (x, z)$ ,  $\bar{x} \in S$  tal que o tempo de propagação seja mínimo. A formulação é dada

pela equação 2.58.

$$\begin{cases} \min_{(x,z) \in S} T_v(x, z) \\ T_v(x, z) = \frac{\sqrt{(x-x_a)^2 + (z-z_a)^2}}{c_1} + \frac{\sqrt{(x-x_f)^2 + (z-z_f)^2}}{c_2} \end{cases} \quad (2.58)$$

À priori, a equação 2.58 representa um problema de otimização contínua bidimensional com restrições, constituindo um problema relativamente difícil, visto que se deseja minimizar o tempo de cálculo.

Por outro lado, uma vez que o espaço de trabalho é discreto, pode se reformular o problema de modo a transformá-lo em um problema de otimização discreta de uma dimensão. Tomando-se por  $k$  o índice do  $k$ -ésimo elemento da interface, escreve-se:

$$\begin{cases} x_s = X(k) = X_k \\ z_s = Z(k) = Z_k \\ T_v(k) = \frac{\sqrt{(X_k-x_a)^2 + (Z_k-z_a)^2}}{c_1} + \frac{\sqrt{(X_k-x_f)^2 + (Z_k-z_f)^2}}{c_2} \end{cases} \quad (2.59)$$

O índice  $k$  cresce ou decresce monotonicamente com o valor de  $x_s$  e varia no intervalo  $[1, m]$ , onde  $m$  é o número de pontos da interface.

O problema de otimização transforma-se em:

$$\min_{k \in [1, m]} T_v(k) \quad (2.60)$$

Derivando a equação 2.59 em função de  $k$ , obtém-se a função  $V_k$ :

$$V_k = \frac{1}{I_t c_1} \frac{(X_k - x_a) + M_k(Z_k - z_a)}{\sqrt{(X_k - x_a) + (Z_k - z_a)^2}} + \frac{1}{c_2} \frac{(X_k - x_f) + M_k(Z_k - z_f)}{\sqrt{(X_k - x_f) + (Z_k - z_f)^2}} \quad (2.61)$$

onde,

$$M_k = \frac{Z(k+1) - Z(k)}{X(k+1) - X(k)} \quad (2.62)$$

A expressão 2.62 aproxima a inclinação da interface. A solução do problema de otimização pode ser determinada quando a função 2.61 atinge valores próximos a zero. Para isso, utiliza-se o algoritmo de Newton-Raphson, descrito em detalhes em Chapra e

Canale (2005).

Atribui-se um valor  $k_0$  e uma tolerância  $tol$  para a função  $V_k$ . O algoritmo consiste em recalcular o valor de  $k$  tanto que  $V_{k+1} - V_k > tol$ .

O cálculo é realizado de maneira iterativa de modo que o valor de  $k_{i+1}$  é dado por:

$$k_{i+1} = k_i - \text{round} \left( \frac{V_k}{V_{k+1} - V_k} \right) \quad (2.63)$$

## 2.6.2 Algoritmo de Nelder-Mead

Alternativamente, podem ser utilizados outros tipos de algoritmos para determinar a solução do problema. Propõe-se a utilização do algoritmo de otimização de Nelder-Mead, cuja implementação é realizada em *Matlab* a partir da função *fminsearch*.

O algoritmo de Nelder-Mead é um algoritmo de otimização não linear sem restrições. É classificado como uma heurística e não faz uso de gradientes na busca pelo valor ótimo da função objetivo. Isso facilita sua utilização frente a problemas onde a função objetivo não possui uma forma fechada bem definida ou quando algoritmos de otimização contínua de funções não lineares são de difícil utilização.

Em geral, o método baseia-se na utilização de um simplex<sup>3</sup>. O algoritmo determina pontos de teste que são arranjados como um simplex. Então, novos pontos de teste são estabelecidos por extração da função objetivo. O algoritmo irá substituir alguns dos pontos existentes pelos novos pontos calculados, dando origem a um novo simplex. O cálculo é realizado de maneira iterativa. Os detalhes de implementação do algoritmo utilizado são dados por Lagarias et al. (1998).

Seja o problema de otimização dado em 2.60. A função objetivo, cujo valor deseja-se minimizar, é a função  $T_v$ , dada pela equação 2.59. Nota-se a partir das equações 2.58 e 2.60 de que se trata de um problema de otimização com restrições, onde o fato de que  $(x, z) \in S$  é traduzido por  $k \in [1, m], k \in \mathbb{Z}_+^*$ .

Assim, o algoritmo de Nelder-Mead deve impreterivelmente considerar que  $k \in [1, m], k \in \mathbb{Z}_+^*$ . Essa característica é importante na medida em que diferentes valores de  $k$  são avaliados de acordo com os passos do algoritmo. Eventualmente, o valor teste

---

<sup>3</sup>Polítopo de  $N + 1$  dimensões com  $N$  vértices

estipulado para  $k$  pode ser um número que não obedece as restrições do problema, como um número que não seja inteiro. Nesse caso, as funções  $X(k)$  e  $Z(k)$  não são definidas e o algoritmo não é capaz de determinar um resultado válido.

Essa alteração é realizada de forma simples, acrescentando as restrições diretamente na função objetivo. O valor de  $k$  será sempre fixado como um inteiro positivo no intervalo desejado. A função objetivo a ser avaliada é escrita na forma descrita em 2.64.

$$\begin{cases} k' = S^{at}(|\text{round}(k)|) \\ T_v(k') = \frac{\sqrt{(X_{k'} - x_a)^2 + (Z_{k'} - z_a)^2}}{c_1} + \frac{\sqrt{(X_{k'} - x_f)^2 + (Z_{k'} - z_f)^2}}{c_2} \end{cases} \quad (2.64)$$

onde, a função  $S^{at}(k) = k$  se  $k \in [1, m]$ ,  $S^{at}(k) = m$  se  $k > m$  e  $S^{at}(k) = 1$  se  $k < 1$ .

Dessa forma, a cada iteração, o algoritmo irá estipular um valor de  $k$  para teste e a função objetivo será avaliada. Utilizando a transformação proposta em 2.64, a função será avaliada para  $k'$ , que será necessariamente um valor válido para o problema.

O resultado prático dessa transformação é que o número de iterações realizadas pelo algoritmo torna-se excessivo, uma vez que pequenas variações de  $k$ , no intervalo  $[0, 1]$ , não são levadas em conta. A solução será calibrar o número de iterações máximas realizadas em um dado passo do algoritmo.

### 3 IMPLEMENTAÇÃO COMPUTACIONAL

Seguindo a proposta de Ibáñez e Villazón (2006), a simulação possui duas etapas: modelagem e cálculo. Primeiramente, apresentam-se aspectos da modelagem do problema. Ela é dividida em duas partes: modelagem do transdutor e do meio de propagação.

Mostra-se como os transdutores são parametrizados e discretizados. Em seguida, discute-se a modelagem do meio de propagação para os casos simples e composto. Aborda-se a questão da escolha dos meios e os diferentes parâmetros envolvidos, assim como a geração das malhas discretas em duas e três dimensões.

Na terceira parte, apresentam-se aspectos de implementação relacionados diretamente com o *MTM*. A determinação da matriz de transferência é explicitada para os diferentes casos abordados, incluindo coeficientes de propagação e leis de atraso.

Finalmente, discute-se o desenvolvimento da ferramenta computacional proposta, cujo objetivo é reunir as duas etapas acima em uma interface gráfica que permita ao usuário realizar simulações do campo acústico para diferentes casos de interesse. Dá-se ênfase na arquitetura utilizada e em sua composição. Todos os códigos são desenvolvidos a partir do software *MATLAB*®.

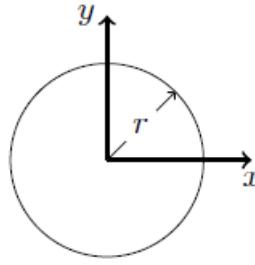
#### 3.1 Modelagem

Tratam-se três tipos de transdutores: circular plano, circular côncavo e *array linear*.

##### 3.1.1 Transdutor Circular Plano

Os parâmetros de configuração de um transdutor circular são o raio  $r$  e o número de divisões  $r_{div}$ , correspondendo ao número de divisões do diâmetro do transdutor. A figura 3.1 indica a referida geometria.

Deseja-se que após discretização os elementos possuam mesma área, de forma a facilitar a implementação do cálculo do campo acústico dada pelo *MTM*. Dessa forma,

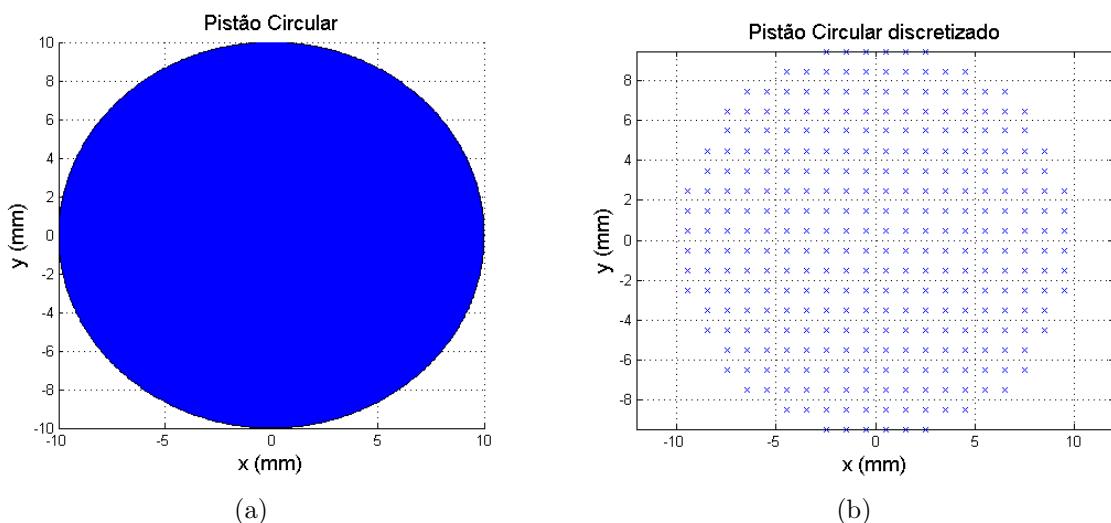


**Figura 3.1:** Transdutor circular plano.

adota-se a seguinte aproximação:

1. Constitui-se uma malha quadrada, de  $r_{div}^2$  elementos, centrada na origem do sistema de coordenadas e cujas dimensões totais são dadas por  $2r \times 2r$ ;
2. Cada ponto da malha indica o centro de um elemento quadrado de dimensões  $d_r \times d_r$ . A distância de cada centro em relação a origem, denominada  $d_0$ , é determinada e armazenada em um vetor;
3. Selecionam-se apenas os elementos cuja distância  $d_0$  verifica a relação  $d_0 \leq r$

A figura 3.2 mostra um transdutor circular plano de  $r = 10\text{mm}$  discretizado com  $r_{div} = 20$ .

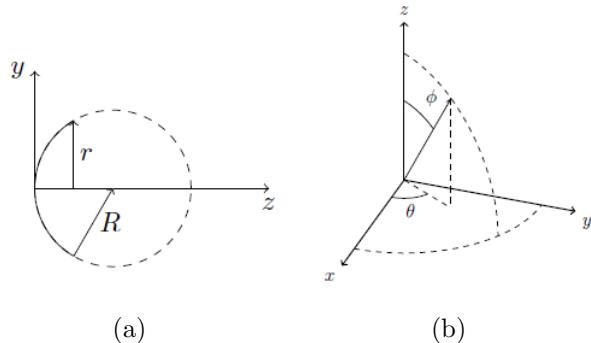


**Figura 3.2:** Modelagem de um transdutor circular plano: (a) Transdutor circular plano,  $r = 10\text{mm}$ , (b) Transdutor circular plano discretizado,  $r_{div} = 20$ .

A qualidade dos resultados da simulação será evidentemente influenciada pelo grau de discretização do transdutor. Ela é função do comprimento de onda dos meios de propagação. Em geral, mantém-se um passo da ordem de duas a três vezes menor que o comprimento de onda dos meios de propagação.

### 3.1.2 Transdutor Circular Côncavo

Os parâmetros de configuração do transdutor circular côncavo são dados na figura 3.3. O parâmetro  $R$  indica o raio de curvatura e  $r$  o raio do transdutor, como indicado em 3.3(a). Já  $\phi$  é o ângulo de elevação e  $\theta$  é o ângulo azimutal, segundo um sistema de coordenadas esféricas.



**Figura 3.3:** Parâmetros de um transdutor circular côncavo: (a) Transdutor circular côncavo, (b) Sistema de coordenadas esféricico.

Ao contrário do transdutor circular plano, o transdutor côncavo possuirá elementos em três dimensões. Devido ao seu formato esférico, é mais difícil discretizá-lo seguindo o esquema anterior, onde a circunferência é aproximada a partir dos elementos de uma malha quadrada. Isso explica a utilização do sistema de coordenadas esférico.

Como parâmetros adicionais têm-se o número de divisões dos ângulos de elevação e azimutal, dados por  $\phi_{div}$  e  $\theta_{div}$ .

A discretização é realizada da seguinte forma:

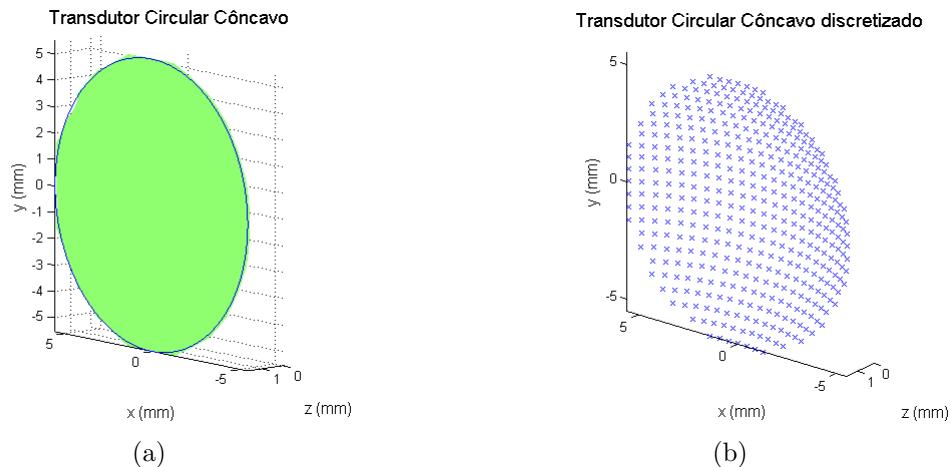
1. A discretização inicial é realizada em coordenadas esféricas. Cria-se uma malha nas dimensões  $(\theta, \phi)$ , conforme o número de divisões especificado. A cada ponto da malha é associado um raio  $R$ . Essa operação determina uma casca esférica.

2. Cada ponto corresponde ao centro de um elemento cuja área é aproximada por:

$$dS = R^2 \sin(\phi) d\theta d\phi \quad (3.1)$$

3. Os pontos da malha no sistema  $(R, \theta, \phi)$  são convertidos para coordenadas cartesianas em um sistema  $(x, y, z)$ . Nessa etapa o centro da esfera coincide com a origem do sistema de coordenadas.
4. A casca esférica é transladada de uma distância  $\sqrt{R^2 - r^2}$  na direção do eixo  $z$ . Em seguida determinam-se os pontos cuja distância  $d_0$  até a origem verifica  $d_0 \leq r$ . Dessa forma, secciona-se a casca esférica na região de interesse delimitada por  $r$ .
5. Todos os pontos são transladados de forma a que a casca esférica esteja posicionada conforme a figura 3.3(a).

A figura 3.5 mostra um transdutor circular côncavo de  $R = 10\text{mm}$  e  $r = 6\text{mm}$  discretizado segundo o método acima.

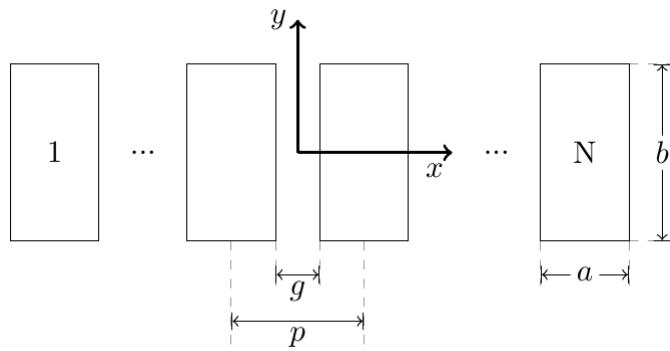


**Figura 3.4:** Modelagem de um transdutor circular côncavo

**Figura 3.5:** Modelagem de um transdutor circular côncavo: (a) Transdutor circular côncavo,  $R = 10\text{mm}$ ,  $r = 6\text{mm}$ ,  $\phi = \pi \text{ rad}$ ,  $\theta = \pi \text{ rad}$ , (b) Transdutor circular côncavo discretizado,  $\phi_{div} = 60$ ,  $\theta_{div} = 60$ .

### 3.1.3 Transdutor Array

O transdutor *array* segue uma extensão da modelagem discutida em 3.1.1. A figura 3.6 ilustra os parâmetros de configuração.



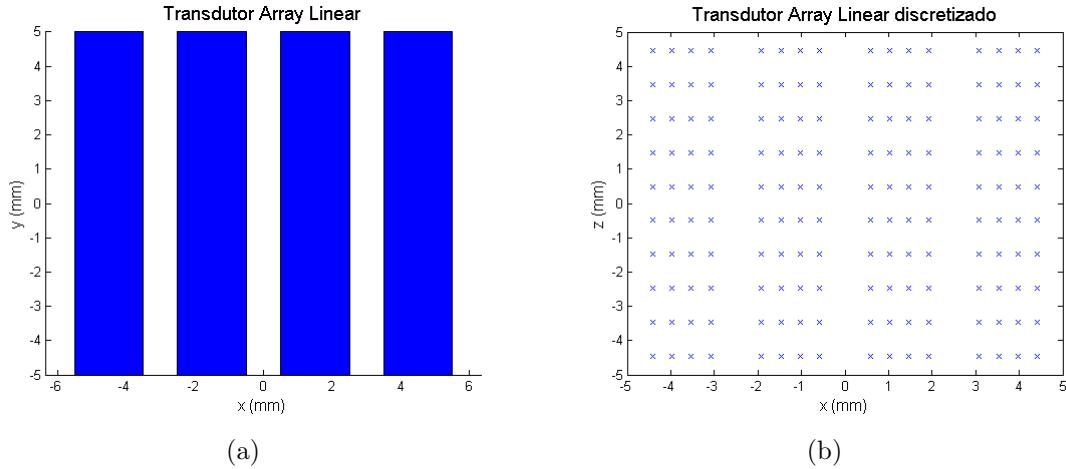
**Figura 3.6:** Parâmetros de um transdutor *array* linear.

São fixados os parâmetros geométricos de um elemento do *array*, considerado um retângulo de dimensões  $a \times b$ . O número de elementos do *array* é dado por  $N$ ,  $g$  corresponde ao espaçamento entre cada elemento e  $p$  é a distância entre os centros de cada elemento. O *array* é centrado em torno da origem do sistema de coordenadas.

A discretização é especificada em função do número de divisões nas direções horizontal e vertical para cada elemento, dado por  $h_{div}$  e  $l_{div}$  respectivamente. As seguintes etapas são realizadas para a discretização:

1. Calculam-se as dimensões totais do *array* para determinação dos centros de cada elemento, tendo em vista o espaçamento entre os mesmos;
2. A área de cada componente discreta de um elemento é constante e dada por  $dS = dx dy$ , onde  $dx$  e  $dy$  são calculados a partir de  $a$ ,  $b$ ,  $h_{div}$  e  $l_{div}$ ;
3. As coordenadas em  $y$  das componentes discretas serão comuns para todos os elementos; já as coordenadas em  $x$  são determinadas em função da posição do centro de um dado elemento e do passo da discretização;
4. As coordenadas das componentes de cada elemento são armazenadas em uma malha retangular. Cada ponto representa o centro de uma componente de dimensões  $dx$  e  $dy$ .

A figura 3.7 mostra um transdutor array de 4 elementos e dimensões  $a = 2mm$ ,  $b = 10mm$  e  $g = 1mm$  discretizado segundo o método acima.



**Figura 3.7:** Modelagem de um transdutor array linear: (a) Transdutor array linear,  $N = 4$ ,  $a = 2\text{mm}$ ,  $b = 10\text{mm}$  e  $g = 1\text{mm}$ , (b) Transdutor array linear discretizado,  $h_{div} = 4$ ,  $l_{div} = 10$ .

### 3.1.4 Meio de propagação

A configuração do meio é função do caso de propagação em um meio simples e em um meio composto.

#### 3.1.4.1 Escolha do meio de propagação

Primeiramente, escolhe-se o meio de propagação. São dadas opções de meios definidos à priori, com valores de velocidades e massa específica previamente configurados. Os valores podem ser alterados e novos meios podem ser configurados de acordo com o estudo realizado.

A tabela 3.1 indica os meios previamente configurados com os respectivos valores de suas propriedades, sendo  $\rho$  o valor da massa específica,  $c_l$  a velocidade de propagação em modo longitudinal e  $c_s$  a velocidade de propagação em modo transversal.

#### 3.1.4.2 Meio simples em duas dimensões

Em um primeiro momento, considera-se a propagação no plano de simetria  $xz$ . A região de interesse é delimitada por um plano descrito pela equação  $y = 0$  e pontos  $(x, z)$  tais que  $x \in [x_{min}, x_{max}]$  e  $z \in [z_{min}, z_{max}]$ . A discretização é realizada a partir dos parâmetros  $x_{div}$  e  $z_{div}$ , que indicam o número de divisões nas duas dimensões referidas.

**Tabela 3.1:** Propriedades físicas de alguns meios de propagação de interesse, retirado de Cheeke (2002).

Material	$\rho$ ( $kg/m^3$ )	$c_l$ ( $m/s$ )	$c_s$ ( $m/s$ )
Alumínio	2700	6420	3040
Aço	7900	5900	3200
Chumbo	24600	2160	700
Cobre	8930	5010	2270
Latão	8640	4700	2100
Quartzo	2200	5960	3750
Água	1000	1500	-

**Tabela 3.2:** Parâmetros de discretização de um meio simples, para a figura 3.8.

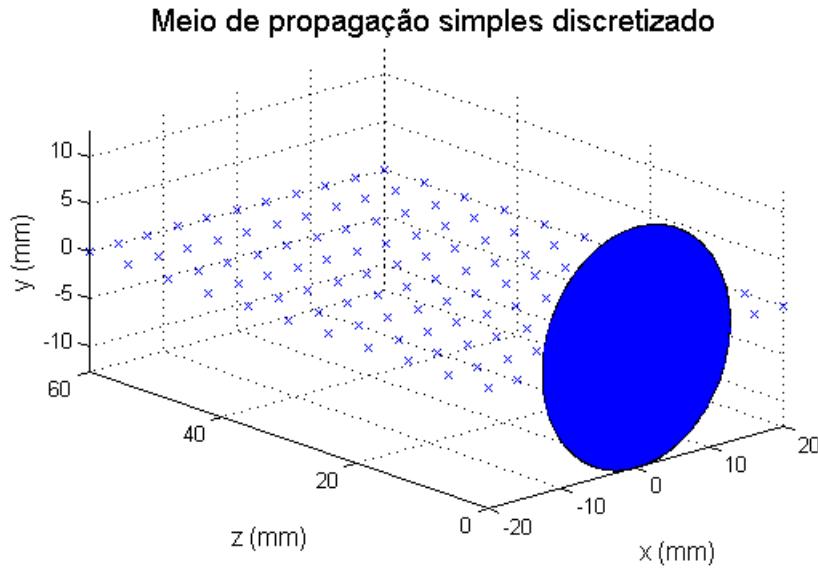
Parâmetros	Valores
$x_{min}$	-20mm
$x_{max}$	20mm
$z_{min}$	0mm
$z_{max}$	60mm
$x_{div}$	10
$z_{div}$	10

A figura 3.8 exemplifica a discretização para uma região do espaço para o caso de uma simulação com um transdutor circular plano. A região é descrita segundo os parâmetros da tabela 3.2.

### 3.1.4.3 Meio composto em duas dimensões

O caso para o meio composto assemelha-se ao discutido anteriormente. Além de definir uma região de interesse, deve-se delimitar uma interface entre os dois meios de propagação.

A interface é definida em três dimensões, no sistema de coordenadas  $(x, y, z)$ . A interface pode ser configurada de duas maneiras distintas, em função de sua forma.



**Figura 3.8:** Discretização de uma região de interesse para o caso de propagação em um meio simples.

No primeiro método, a interface é parametrizada como uma *spline* cúbica<sup>1</sup> no plano  $y = 0$ . Os parâmetros de configuração correspondem aos pontos de controle e ao grau de discretização indicado pelo número de divisões da curva.

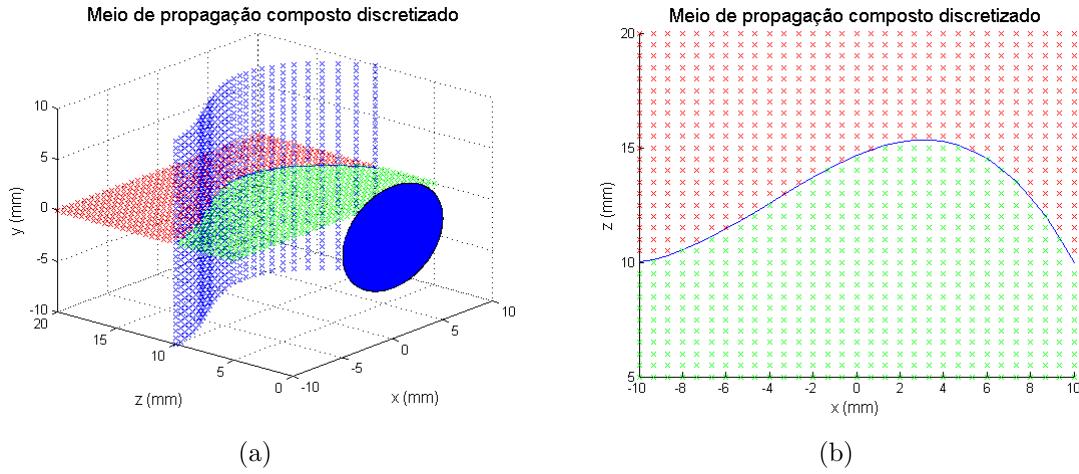
No segundo método, a interface é parametrizada como uma circunferência. Ela é definida a partir de dois pontos pertencentes à curva, o centro da circunferência e o grau de discretização, analogamente ao caso anterior.

A dimensão  $y$  é acrescida configurando-se os limites  $[y_{min}, y_{max}]$  e indicando o grau de discretização nessa dimensão; a curva definida no plano  $xz$  é então estendida nessa direção originando uma superfície tridimensional.

As figuras 3.9(a) e 3.9(b) exemplifica a modelagem descrita para uma interface de separação entre dois meios do tipo *spline*. Os parâmetros de configuração da interface são dados na tabela 3.3.

A configuração da região de interesse segue a mesma implementação da seção 3.1.4.2. Especificamente para o caso de um meio composto, discrimina-se na região de interesse quais pontos correspondem ao meio 1 e ao meio 2. Isso é demonstrado na figura

<sup>1</sup>Uma *spline* corresponde a uma função polinomial por partes definida a partir de seus pontos de controle.



**Figura 3.9:** Modelagem de um meio de propagação composto com interface de separação do tipo *spline* (a) Visualização espacial, (b) Visualização no plano  $(xz)$ .

**Tabela 3.3:** Parâmetros de configuração para *spline* da figura 3.9.

Parâmetro	Valor
tipo	spline
$y_{min}$	-10
$y_{max}$	10
$y_{div}$	30
$n_{div}$	30
$(x_1, z_1)$	$(-10, 10)$
$(x_2, z_2)$	$(-5, 12)$
$(x_3, z_3)$	$(5, 15)$
$(x_4, z_4)$	$(10, 10)$

3.9, com a distinção dos pontos em diferentes cores. O cálculo do campo acústico é realizado apenas na região correspondente ao meio 2.

### 3.1.5 Análise tridimensional

Uma vez determinado o campo acústico no plano  $xz$ , é possível refinar a análise determinando-se o campo acústico em uma região tridimensional do espaço. A ideia é que ao calcular o campo bidimensional, definam-se regiões no espaço para analisar, por exemplo, propriedades de interesse, como a abertura vertical do feixe acústico.

Dessa forma, em um segundo momento da análise, é possível definir um plano no espaço  $xyz$ , que delimitará a nova região de interesse. Procede-se da seguinte maneira:

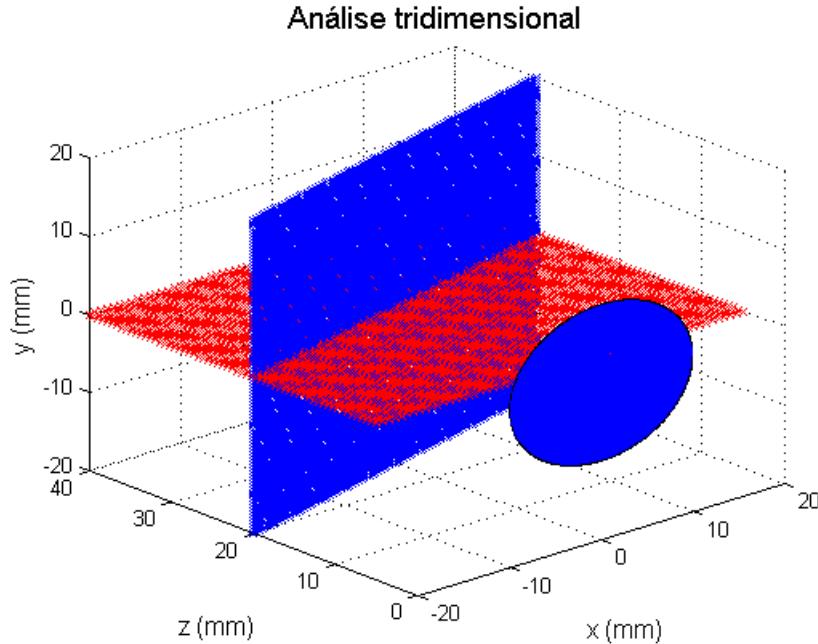
1. Defini-se uma reta entre dois pontos  $(x_1, 0, z_1)$  e  $(x_2, 0, z_2)$
2. Delimita-se a altura da região de interesse, definindo-se as coordenadas  $y_{min}$  e  $y_{max}$
3. Configuram-se os graus de discretização da reta em  $xz$  e da altura em  $y$ .

Um exemplo é dado na figura 3.10, para uma reta definida pelos pontos  $(x_1, z_1) = (-20, 20)$  e  $(x_2, z_2) = (20, 30)$  e altura definida por  $y_{min} = -20mm$  e  $y_{max} = 20mm$ .

## 3.2 Implementação do *MTM*

Uma das vantagens de utilização do software *MATLAB*® é a facilidade para manipulação e cálculo envolvendo matrizes. Para o caso tratado neste trabalho, isso permitirá a realização de cálculos sem a necessidade de introdução de laços do tipo *for* e *while*, construções comumente encontradas na implementação de métodos numéricos dos mais variados tipos. Isso contribui na medida em que não somente deixa o código mais sucinto, mas fundamentalmente porque diminui o custo computacional associado ao algoritmo em termos de tempo de cálculo.

Conforme a etapa de modelagem, a implementação do *MTM* também é dividida entre os casos de propagação em meio simples e propagação em meio composto. A justificativa são as diferenças de implementação associadas a cada caso, como discutido na seção 2.5.



**Figura 3.10:** Plano tridimensional traçado a partir de uma reta definida pelos pontos  $(x_1, z_1) = (-20, 20)mm$  e  $(x_2, z_2) = (20, 30)mm$ .

### 3.2.1 MTM em um meio de propagação simples

No cálculo da pressão acústica  $\mathbf{U}$ , retomam-se as equações básicas do método empregado, reescritas nas equações 3.2, 3.3 e 3.4, nas formas algébrica e em notação matricial, respectivamente. Os problemas tratados em toda a extensão do trabalho utilizam a condição de contorno de Neumann, que corresponde a hipótese de um pistão rígido:

$$\mathbf{U} = \mathbf{T}\mathbf{E} \quad (3.2)$$

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{m1} & m_{m2} & \cdots & m_{mn} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \quad (3.3)$$

onde  $U$  contém o valor da pressão acústica para os  $m$  pontos do meio de propagação e os elementos da matriz de transferência  $\mathbf{T}$  de dimensão  $m \times n$  correspondem à:

$$m_{ij} = \frac{s_j}{2\pi} \frac{\exp(-jkr_{ij})}{r_{ij}}, \quad \text{condição de Neumann} \quad (3.4)$$

O vetor  $\mathbf{E}$  corresponde aos termos de apodização e aos termos da lei de atraso. A dimensão desse vetor é  $n \times 1$ , que corresponde ao número de fontes pontuais do transdutor após discretização. Sua formulação geral é dada pela equação:

$$\mathbf{E} = \mathbf{A} \circ \mathbf{D} \quad (3.5)$$

onde  $\mathbf{A}_{n \times 1}$  é o vetor de apodização e  $\mathbf{D}_{n \times 1}$  o vetor de atrasos e o operador  $\circ$  indica a multiplicação termo a termo entre os vetores. As leis de apodização e de atraso são aplicáveis somente para o caso de um transdutor array.

A lei de apodização implementada tem a forma de uma distribuição normal, parametrizada por  $\mu$  e  $\sigma$ , conforme equação 3.6:

$$f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \quad (3.6)$$

onde  $x$  é a abscissa que dá a posição dos centros de cada elemento do transdutor.

Os fatores de apodização correspondem a uma atenuação da amplitude do emissor e são normalizados no intervalo  $[0, 1]$ . Na sequência do trabalho, esses termos serão considerados unitários e os resultados são apresentados sem consideração de termos de apodização.

As leis de atraso são determinadas conforme os princípios enunciados na seção 2.6. A forma geral dos termos de atraso é dada por:

$$g(x) = \exp(-j\Delta\theta) \quad (3.7)$$

onde  $\Delta\theta = \omega \cdot t_v$  corresponde à fase associada ao elemento na posição  $x$ .

Para a determinação da matriz  $\mathbf{T}$ , calcula-se primeiramente a matriz  $\mathbf{R}_{m \times n}$ , onde cada elemento  $r_{ij}$  corresponde à distância euclidiana entre um ponto da malha que caracteriza o meio de propagação e uma fonte pontual da malha do transdutor discretizado.

Sejam  $X_t$ ,  $Y_t$  e  $Z_t$  estruturas de dados de dimensão  $n \times 1$  associadas ao modelo de um transdutor. Elas armazenam cada ponto  $(x_t, y_t, z_t)$  da malha transdutor discretizado. Analogamente, sejam  $X_m$ ,  $Y_m$  e  $Z_m$  estruturas de dados de dimensão  $m \times 1$  associadas ao modelo do meio de propagação, que armazenam cada ponto do  $(x_m, y_m, z_m)$  da malha discretizada. Em geral, essas estruturas são obtidas aplicando o operador *meshgrid* da

biblioteca *MATLAB*®, seguido de uma operação de *reshape*, para a formação de uma matriz coluna.

Utilizando-se novamente a função *meshgrid*, é possível criar novas estruturas de dados associadas tanto ao transdutor, quanto ao meio de propagação, cuja dimensão seja  $m \times n$ . Por exemplo, para o vetor  $X_m$ , isso equivale a recopiá-lo em  $n$  vezes, como mostrado na equação:

$$X'_m = \left[ \begin{array}{c} X_m \cdots X_m \end{array} \right]_{m \times n} \quad (3.8)$$

onde o operador  $[\cdot]'$  indica a nova matriz de dimensão  $m \times n$  obtida após a transformação descrita.

Aplicando-se esse procedimento para todas as estruturas de dados, a matriz **R** é facilmente calculada pela expressão:

$$\mathbf{R} = \sqrt{(X_t - X_m)^2 + (Y_t - Y_m)^2 + (Z_t - Z_m)^2} \quad (3.9)$$

onde foi omitido o operador  $[\cdot]'$  por razões de clareza, e o operador  $[\cdot]^2$  calcula o quadrado de cada elemento da matriz resultante da operação algébrica de soma <sup>2</sup>.

Ressalta-se que a operação realizada em  $[\cdot]'$  só é possível, uma vez que todas as matrizes possuem a mesma dimensão. Isso permite explorar o potencial de cálculo matricial do software.

A matriz **T** é facilmente determinada por:

$$\mathbf{T} = \frac{1}{2\pi} \mathbf{s} \circ \exp \frac{-jk\mathbf{R}}{\mathbf{R}} \quad (3.10)$$

onde  $\mathbf{s}_{m \times n}$  é a matriz que contém a área de cada elemento do transdutor e todas as operações algébricas são aplicadas termo a termo.

A matriz **s** também é obtida aplicando-se a transformação descrita em 3.8, partindo-se do vetor que armazena as áreas de todos os elementos discretos do transdutor. Para os transdutores planos, todos os elementos resultantes da discretização possuem mesma área  $s_{ij}$ . Assim, esse termo é simplificado e não há a necessidade de utilização de uma

---

<sup>2</sup>De maneira geral, em *MATLAB*®, as operações precedidas por  $[\cdot]$  indicam uma operação matricial aplicada termo-a-termo

estrutura de dados matricial, reduzindo-se o custo computacional em memória.

O cálculo final é realizado a partir da equação:

$$\mathbf{U} = \left( \frac{1}{2\pi} \mathbf{s} \circ \exp \frac{-jk\mathbf{R}}{\mathbf{R}} \right) * [\mathbf{A} \circ \mathbf{D}] \quad (3.11)$$

onde  $*$  indica uma multiplicação matricial. O campo acústico é descrito por um vetor  $\mathbf{U}$  de dimensão  $m \times 1$ , que é reescrito na forma de uma malha para plotagem gráfica em função da malha do meio de propagação.

Analizando a expressão 3.11, nota-se que ela constitui uma aplicação direta do princípio de *Huygens*. A multiplicação matricial indica que o campo acústico em um ponto  $P_0$  em questão é resultado da contribuição de cada ponto associado ao transdutor, descritos ao longo das linhas do vetor  $\mathbf{E}$ , o que é enunciado na equação 2.40, reescrita a seguir:

$$U(P_0) = \sum_j m_j(P_0) \cdot E_j$$

Essa implementação constitui a forma básica do *MTM*. A seguir, descreve-se a extensão desse método para o caso da propagação entre dois meios através de uma interface.

### 3.2.2 MTM em um meio de propagação composto

As principais modificações no método em relação ao caso anterior dizem respeito aos efeitos provenientes da propagação através da interface. Para o caso de um meio composto, a formulação do *MTM* é alterada da seguinte forma. Em primeiro lugar, determina-se o campo acústico incidente na interface de separação entre os meios. Por sua vez, segundo o princípio de *Huygens*, cada ponto da interface irá atuar como uma fonte pontual, e o campo acústico resultante no segundo meio resulta da somatória das contribuições das diversas fontes da interface discretizada.

Então, a equação 3.2 é alterada para a expressão:

$$\mathbf{U} = \mathbf{T}_{12} * (\mathbf{T}_{11} * \mathbf{E}) \quad (3.12)$$

onde  $\mathbf{T}_{11}$  corresponde à matriz de transferência do transdutor posicionado no meio 1 para

a interface e  $\mathbf{T}_{\mathbf{I}2}$  corresponde à matriz de transferência da interface para o meio 2.

Por outro lado, nem toda a energia é transmitida da interface para o segundo meio, sendo uma parcela refletida e outra parcela transmitida, de acordo com os coeficientes de reflexão e transmissão correspondentes, conforme apontado na seção 2.5.2. Além disso, no caso do segundo meio se tratar de um sólido, há dois modos de propagação, um longitudinal e outro transversal. Esses modos serão descritos pelos coeficientes de potência associados e dados pelas matrizes  $\mathbf{T}^{tl}$  e  $\mathbf{T}^{ts}$ .

Essas modificações quando aplicadas à equação 3.12 resultam em:

$$\mathbf{U} = \mathbf{T}_{\mathbf{I}2} * ((\mathbf{T}^{tm} \circ \mathbf{T}_{\mathbf{II}}) * \mathbf{E}) \quad (3.13)$$

onde  $\mathbf{T}^{tm}$  indica a matriz de transmissão, cujo índice  $m$  corresponde ao modo longitudinal ( $l$ ) ou transversal ( $s$ ).

Há quatro matrizes a serem determinadas: duas matrizes de transferência,  $\mathbf{T}_{\mathbf{II}}$  e  $\mathbf{T}_{\mathbf{I}2}$ , a matriz de transmissão  $\mathbf{T}^{tm}$  e o vetor  $\mathbf{E}$ , que possui a mesma forma descrita anteriormente.

### 3.2.2.1 Determinação das matrizes de transferência

As matrizes de transferência são determinadas seguindo o mesmo procedimento utilizado no caso de um meio de propagação simples.

A matriz  $\mathbf{T}_{\mathbf{II}}$  é dada pela expressão:

$$\mathbf{T}_{\mathbf{II}} = \frac{1}{2\pi} \mathbf{s}_t \circ \exp \frac{-jk_1 \mathbf{R}_{\mathbf{II}}}{\mathbf{R}_{\mathbf{II}}} \quad (3.14)$$

onde  $\mathbf{s}_t$  é a matriz que agrupa os elementos de área do transdutor e  $\mathbf{R}_{\mathbf{II}}$  é uma matriz de dimensão  $(w \times n)$ , onde  $w$  corresponde ao número de pontos do primeiro meio de propagação e  $n$  ao número de pontos do transdutor discretizado, que fornece a distância euclidiana entre cada fonte pontual e cada ponto do meio de propagação. Essa distância é determinada a partir da equação 3.15:

$$\mathbf{R}_{\mathbf{II}} = \sqrt{(X_t - X_I)^2 + (Y_t - Y_I)^2 + (Z_t - Z_I)^2} \quad (3.15)$$

onde  $X_I$ ,  $Y_I$  e  $Z_I$  representam a interface e todas as matrizes possuem dimensão  $(w \times n)$ ,

obtidas a partir de uma transformação usando a função *meshgrid*, como em 3.2.1.

Procede-se da mesma forma para a determinação da matriz  $\mathbf{T}_{\mathbf{I2}}$ . Seja o meio 2 constituído por uma malha de  $m$  pontos de coordenadas  $(x_{m2}, y_{m2}, z_{m2})$ . Os elementos da interface fazem o papel de emissores, e a matriz  $\mathbf{T}_{\mathbf{I2}}$  é dada por:

$$\begin{cases} \mathbf{T}_{\mathbf{I2}} = \frac{1}{2\pi} \mathbf{S}_{\mathbf{I}} \circ \exp \frac{-jk_2 \mathbf{R}_{\mathbf{I2}}}{\mathbf{R}_{\mathbf{I2}}} \\ \mathbf{R}_{\mathbf{I2}} = \sqrt{(X_I - X_{m2})^2 + (Y_I - Y_{m2})^2 + (Z_I - Z_{m2})^2} \end{cases} \quad (3.16)$$

onde  $X_{m2}$ ,  $Y_{m2}$  e  $Z_{m2}$  representam o meio 2,  $s_I$  a matriz que agrupa os elementos de área da interface e todas as matrizes possuem dimensão  $(m \times w)$ .

### 3.2.2.2 Determinação da matriz de transmissão

Para a determinação da matriz de transmissão, necessita-se do conhecimento dos ângulos de incidência das ondas emitidas do transdutor para a interface e de uma função que relate os ângulos a um coeficiente de potência.

Seja  $\alpha_{ij}$  o ângulo de incidência da onda emitida pela fonte pontual  $i$  para o elemento  $j$  da interface, com  $i \in [1, n]$ ,  $j \in [1, m]$  e  $\alpha \in [0, \pi/2]$ . Seja  $tr(\alpha)$  uma função que calcula o coeficiente de transmissão  $tr_{ij}$  em função do ângulo de incidência. A determinação de  $tr_{ij}$  pode ser realizada da seguinte forma:

1. Cálculo dos ângulos de incidência, representados através de uma matriz  $I_\alpha$  de dimensão  $(m \times n)$ .
2. Cálculo da função  $tr(\alpha)$  a partir das relações dadas em 2.5.2.
3. Determinação dos  $tr_{ij}$  a partir de uma *lookup table*: associa-se um elemento de  $tr(\alpha)$  para cada  $\alpha_{ij}$

Esse procedimento pode ser facilmente realizado em *MATLAB*® a partir da função *interp1*, que implementa diretamente a *lookup table* para a função  $tr(\alpha)$ . Os ângulos de incidência são determinados de maneira aproximada, conforme procedimento descrito a seguir.

Seja a interface representada no plano  $xz$  e descrita pelos pontos  $(x_I, 0, z_I)$ . No caso mais geral, a interface é dada por uma *spline*, e não há uma função que a parametrize

analiticamente. Assim, o vetor tangente  $\vec{t}$  a um ponto da interface, de coordenadas  $(u, 0, v)$ , é aproximado por:

$$\vec{t} = \begin{bmatrix} u \\ 0 \\ v \end{bmatrix} = \begin{bmatrix} x_{I+1} - x_{I-1} \\ 0 \\ z_{I+1} - z_{I-1} \end{bmatrix} \quad (3.17)$$

Seja  $\vec{j}$  o versor normal ao plano  $xz$ . Pode-se determinar o vetor normal a um ponto da interface, dado por  $\vec{n}$ , a partir de um simples produto vetorial:

$$\vec{n} = \vec{j} \times \vec{t} \quad (3.18)$$

onde o operador  $\times$  indica o produto vetorial. Os vetores normais à interface calculados para os pontos no plano  $(xz)$  são os mesmos para os demais pontos de cota  $y_I$ , devido à maneira como a interface é discretizada, conforme seção 3.1.4.3.

Seja  $\vec{b}$  o vetor incidente entre um ponto do transdutor e um ponto da interface, determinado pela expressão 3.19:

$$\vec{b} = \begin{bmatrix} x_t - x_I \\ y_t - y_I \\ z_t - z_I \end{bmatrix} \quad (3.19)$$

O ângulo de incidência é então determinando através do produto interno entre  $\vec{n}$  e  $\vec{b}$ :

$$\alpha_{ij} = \cos^{-1} \left( \frac{\vec{n} \cdot \vec{b}}{|\vec{n}| |\vec{b}|} \right) \quad (3.20)$$

Uma vez calculada a matriz  $I_\alpha$ , que contém os valores de todos os ângulos de incidência  $\alpha_{ij}$ , os coeficientes são determinados por:

$$\mathbf{T}^{\text{tm}} = \text{interp1}(\text{tr}(\alpha), \alpha, I_\alpha) \quad (3.21)$$

onde  $\text{tr}(\alpha)$  e  $\alpha$  são vetores finamente discretizados que representam a função que fornece o coeficiente de transmissão em função de  $\alpha$  e  $\text{interp1}$  é a função *MATLAB*® que

implementa a *lookup table*.

Os coeficientes de transmissão são calculados conforme a abordagem proposta por Oliner (1969), a partir de modelos de linhas de transmissão, e desenvolvida em Franco et al. (2005).

### 3.2.2.3 Formulação final do MTM

A expressão do *MTM* para o caso de propagação em um meio composto pode, então, ser escrita da seguinte maneira:

$$\mathbf{U} = \left\{ \left( \frac{1}{2\pi} \mathbf{s}_I \circ \exp \frac{-jk_2 \mathbf{R}_{I2}}{\mathbf{R}_{I2}} \right) * \left[ \mathbf{T}^{tm} \circ \left( \frac{1}{2\pi} \mathbf{s}_t \circ \exp \frac{-jk_1 \mathbf{R}_{1I}}{\mathbf{R}_{1I}} \right) \right] \right\} * \mathbf{E} \quad (3.22)$$

Essa é forma final do *MTM*, implementada no cálculo do campo acústico para um meio de propagação composto. Assim como no caso para propagação em um meio simples, o método procura explorar ao máximo as operações matriciais, não havendo ocorrência de laços internos. Como anteriormente, o campo acústico será descrito por um vetor  $\mathbf{U}$  de dimensão  $m \times 1$ , equivalente ao campo em cada ponto da malha do meio de propagação.

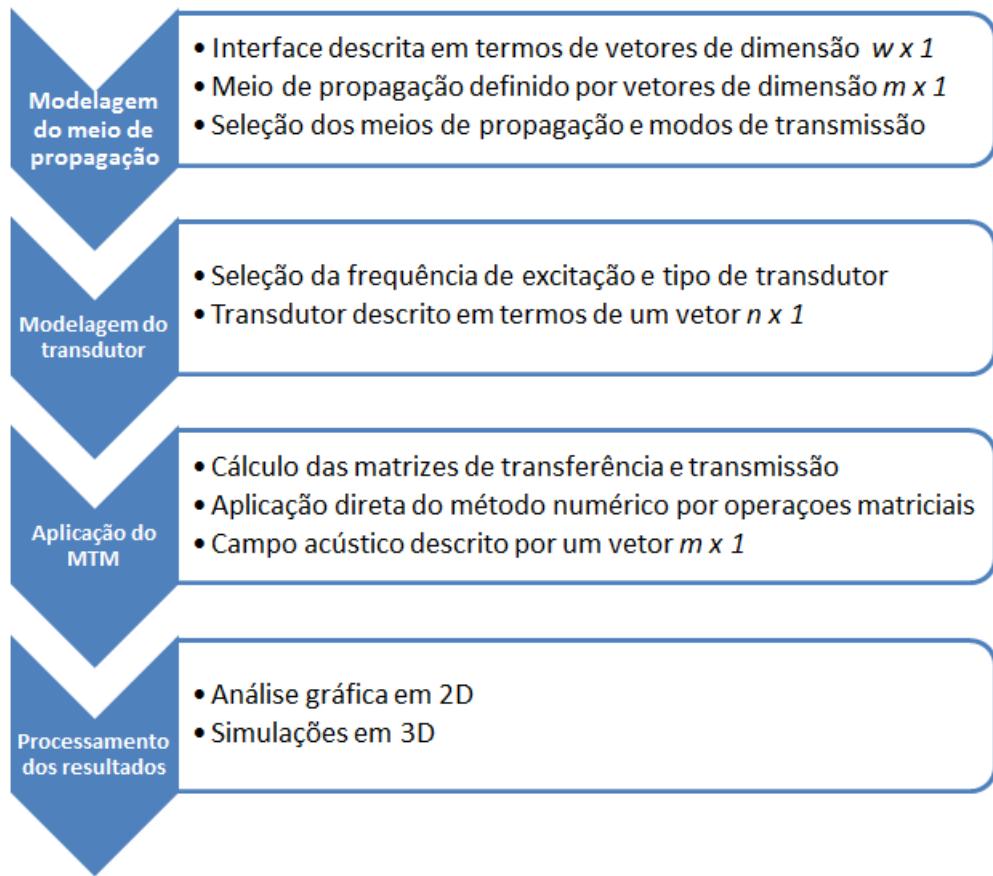
De maneira geral, a manutenção dessa característica é de grande interesse, pois permite a realização de estimativas complexas em um curto período de tempo. Por outro lado, existe uma desvantagem no que se refere à utilização de memória do sistema. Para interfaces com altos graus de discretização, pode faltar memória para a realização dos cálculos, devido à dimensão excessiva das matrizes de transferência. Isso é indicado através de uma mensagem de erro na linha de comando do software.

Em relação aos estudos realizados no contexto deste trabalho, essa característica não provocou maiores problemas. Esse aspecto será rediscutido ao final do trabalho.

Na prática, como o método *MTM* parte de uma solução esférica da equação de onda, observam-se singularidades, com descontinuidade do campo acústico calculado para distâncias muito próximas dos emissores. Dessa forma, consideram-se válidos os valores calculados para distâncias de no mínimo um comprimento de onda em relação aos mesmos.

O diagrama da figura 3.11 resume a implementação do método para o caso mais de geral de propagação em um meio composto, na ordem em que é executada no *software*

desenvolvido.



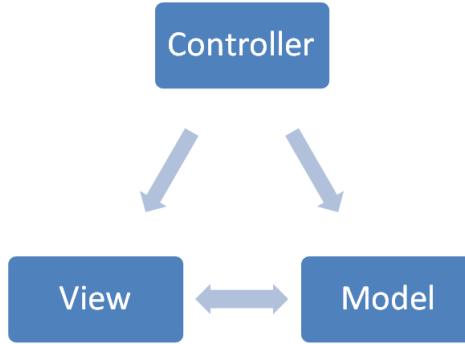
**Figura 3.11:** Etapas para implementação do método *MTM*.

### 3.3 Projeto do software

O desenvolvimento da ferramenta computacional é realizado a partir da utilização de um *design pattern*. Em programação orientada a objetos, um *design pattern* é uma solução de caráter geral para problemas recorrentes no desenvolvimento de sistemas. Em geral, a solução oferecida tem caráter reutilizável e deve ser completada de acordo com o problema tratado.

O *design pattern* utilizado é do tipo *Model-View-Controller* (MVC). Um esquema dessa arquitetura é apresentado na figura 3.12.

A vantagem da utilização dessa arquitetura é que se pode facilmente isolar as partes do sistema, promovendo a modularização do código. Assim, a parte *model* comprehende a etapa de modelagem, integrando classes que correspondem ao transdutor e ao meio de propagação. A parte *view* corresponde à interface gráfica, a partir do qual o usuário



**Figura 3.12:** Diagrama MVC.

pode alterar configurações do modelo, realizar as simulações e visualizar os resultados. Finalmente, a parte *controller* implementa ações de controle inerentes ao funcionamento correto do sistema.

Esse tipo de implementação também permite acrescentar funcionalidades de maneira relativamente simples, o que em muitos casos necessitaria um alto nível de compreensão do código utilizado. Nesse sentido, é importante ressaltar que a utilização da biblioteca que constitui a classe correspondente ao modelo é independente das demais no sentido de que o usuário tem a liberdade de utilizar as classes sem a necessidade de invocar o modo gráfico. Assim, novos métodos podem ser testados previamente à sua implementação na interface gráfica.

Um detalhe importante que diz respeito à programação orientada objetos em *MATLAB*® é utilização da classe *handle*. Através dela é possível realizar alocação dinâmica de memória, e assim, constituir objetos e alocar variáveis a partir de suas referências. Em geral, a maioria das classes implementadas derivam diretamente ou indiretamente dessa classe.

Esse tipo de implementação difere consideravelmente da implementação clássica realizada a partir do *toolbox MATLAB*® denominado *Guide*. A realização de múltiplas interfaces torna-se difícil e requer funções que realizem a comunicação entre as diferentes partes. O código torna-se prolixo e pouco modularizado.

No presente trabalho, o módulo gráfico do *toolbox* é utilizado apenas com o intuito de montar o *layout* das diferentes interfaces. Em seguida, são implementadas classes que agregam cada *layout*, deixando o código modularizado e facilitando a compreensão. A alteração de parâmetros é realizada diretamente pelas referências de cada objeto.

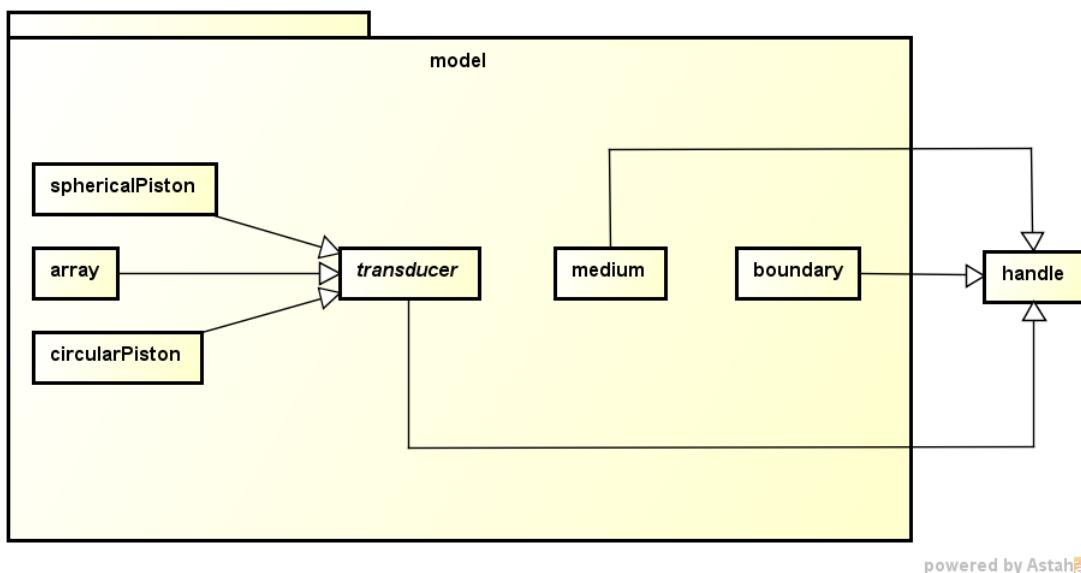
Assim, a utilização da programação orientada a objetos em *MATLAB*® permite a realização de interfaces com alto grau de complexidade e acrescenta inúmeras vantagens ao programador em relação a métodos clássicos.

O objetivo é que se possa abstrair o modelo e a implementação dos métodos de cálculo associados através da interface gráfica, oferecendo uma solução que possua utilização simples, não requerendo conhecimentos avançados de programação em *MATLAB*®.

A seguir, cada parte do *design pattern* é brevemente discutida a partir de diagramas *UML* (*Unified Modelling Language*) elaborados no software *ASTAH*®. A implementação detalhada das classes é dada no anexo A, onde cada uma é apresentada com seus respectivos atributos e métodos. Nesse capítulo, optou-se por omiti-los de forma a deixar os diagramas menos poluídos.

### 3.3.1 Implementação do modelo

A figura 3.13 mostra as classes do modelo que representa o sistema. Para os problemas tratados, o modelo é composto por um transdutor, meios de propagação e uma interface entre ambos. O transdutor constitui uma classe abstrata. Dela derivam três classes que representam os tipos de transdutores implementados: transdutor circular plano, côncavo e transdutor *array*.



**Figura 3.13:** Implementação do modelo do sistema.

### 3.3.2 Implementação da parte gráfica

A figura 3.14 mostra as classes associadas com a configuração de cada parte do modelo. Em geral, cada uma representa um ambiente de configuração distinto. A classe *main\_gui* reúne todos esses ambientes a partir de uma relação de composição. A classe *configGui* constitui uma classe abstrata que contém atributos e métodos comuns a uma classe do tipo “ambiente de configuração”.

### 3.3.3 Implementação do controlador

A figura 3.15 mostra a arquitetura do controlador. As ações de controle são relativamente simples, uma vez que uma grande parte da interface é dedicada a parte de modelagem e configuração de parâmetros associados ao modelo.

O controlador possui duas classes associadas. Uma, a classe *gui\_state*, possui tipos básicos de enumeração, que listam os estados possíveis para a interface. Nesse trabalho, a máquina de estados é reduzida a dois estados básicos: *idle* e *simulation*. No primeiro, a interface está a espera da entrada de configurações associadas a etapa de modelagem. Na segunda, as rotinas auxiliares são utilizadas de forma a realizar os cálculos associados à simulação. A figura 3.16 descreve esse comportamento.

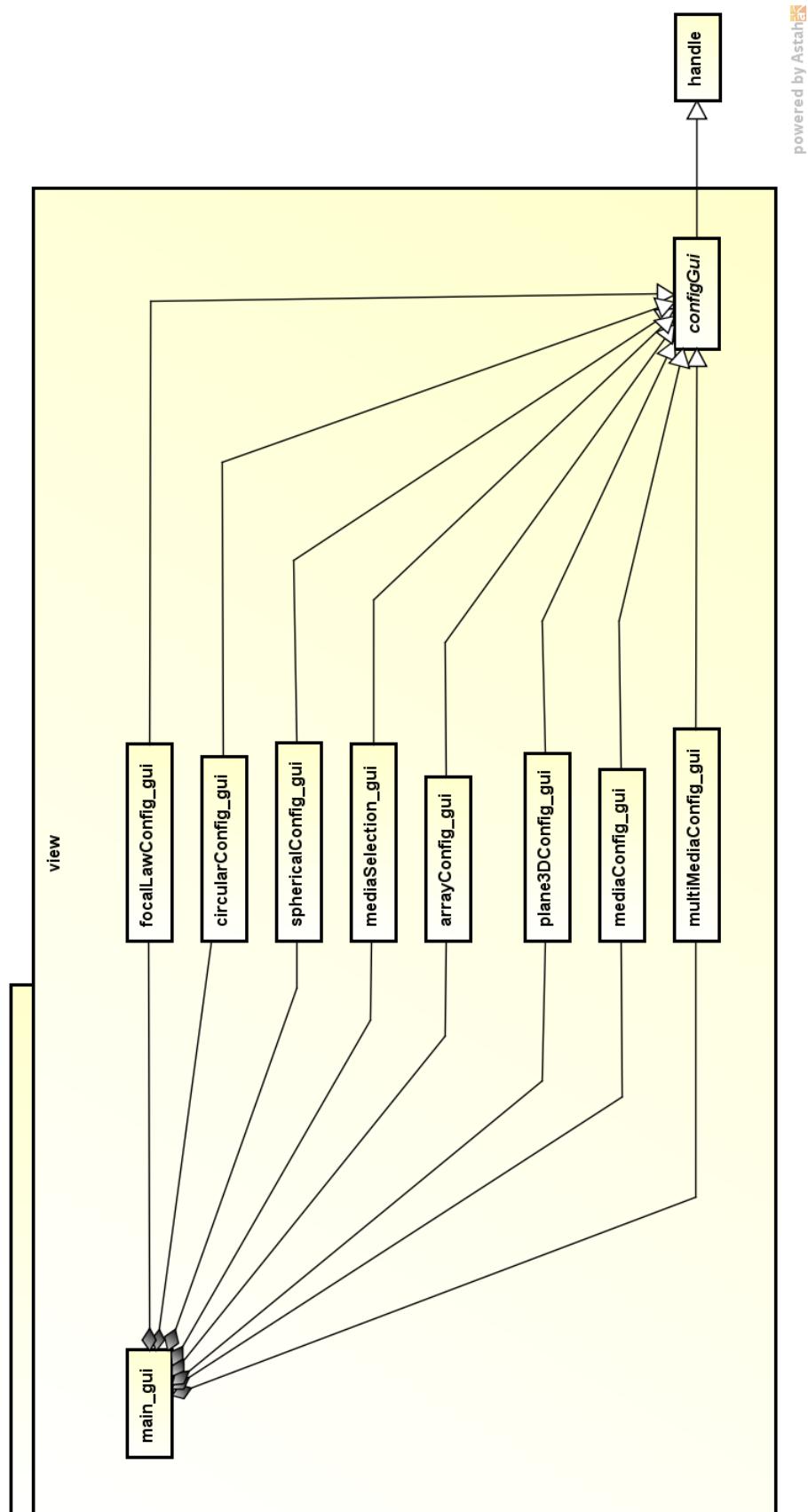
No estado *simulation*, todas as funcionalidades são temporariamente desabilitadas e o usuário deve aguardar o término dos cálculos.

### 3.3.4 Rotinas auxiliares

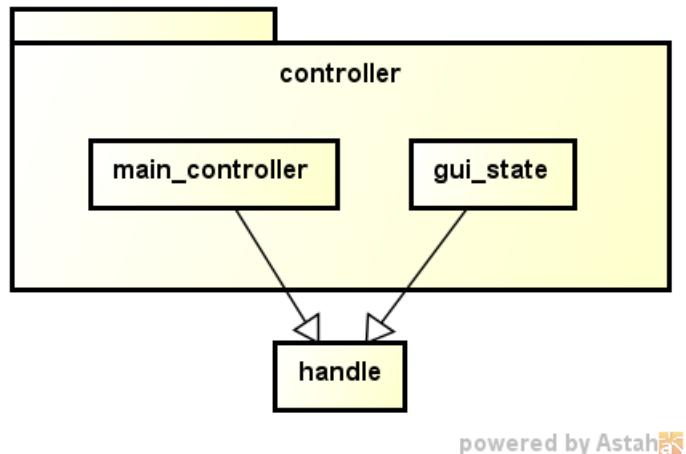
Um certo número de rotinas auxiliares foram desenvolvidas. Elas tratam em grande parte da implementação dos métodos numéricos descritos nos capítulos 2 e 3.

Em primeiro plano, têm-se as funções para determinação do campo acústico. Elas correspondem aos scripts *acousticFieldSimpleMedia* e *acousticFieldMultiMedia*. Tratam-se separadamente os casos de propagação em meio simples e em meio composto.

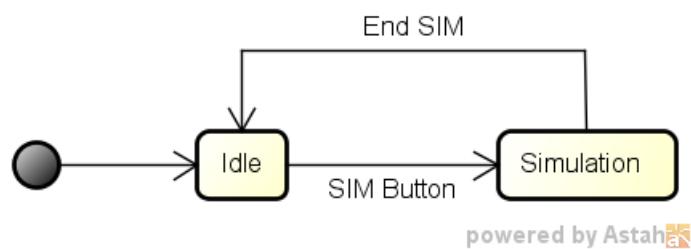
Em seguida, há as funções dedicadas à determinação das leis de atraso, de acordo com os três métodos descritos neste trabalho. A função *delay* realiza o cálculo exato das leis de atraso sem a presença de interfaces, para o caso de um meio de propagação simples. As funções *delay\_interface*, *delay\_newton* e *delay\_fmin*, realizam o cálculo para o caso onde



**Figura 3.14:** Implementação da parte gráfica da interface.



**Figura 3.15:** Implementação do controlador.



**Figura 3.16:** Máquina de estados que modela o comportamento da interface.

há a presença de uma interface de separação entre dois meios, implementando respectivamente a solução exata, o método de Newton-Raphson e o método de Nelder-Mead, de acordo com a seção 2.6. Finalmente, as funções *focal\_law* e *focal\_law2* implementam as funções objetivo dos algoritmos de Newton-Raphson e de Nelder-Mead.

O cálculo dos coeficientes de transmissão é realizado com base nas funções *boundary\_incidence*, que calcula os ângulos de incidência para cada emissor de um transdutor, e *solid\_propagation*, que implementa o método descrito em 2.5.2 para determinação dos coeficientes de potência.

Ainda, outras funções são dedicadas ao cálculo de coeficientes de apodização, definição das interfaces entre dois meios e visualização dos resultados.

## 4 RESULTADOS

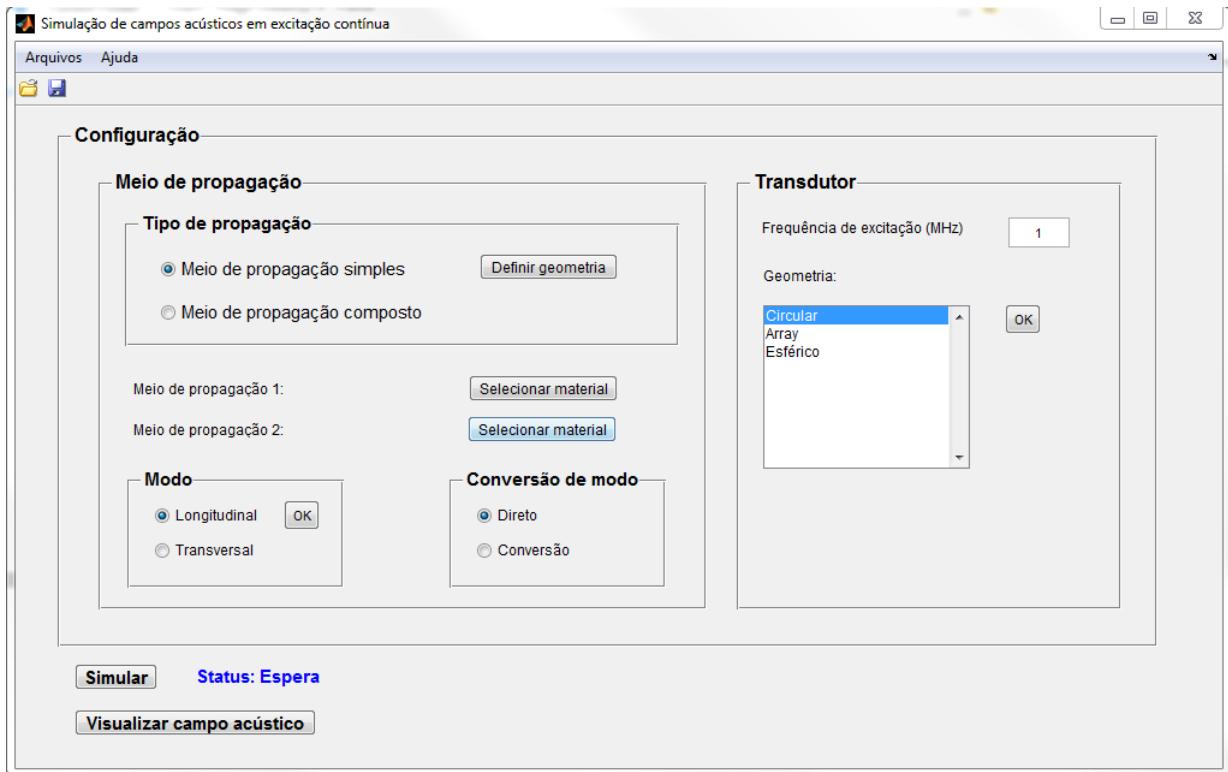
Neste capítulo é apresentada a ferramenta desenvolvida ao longo do capítulo 3, bem como resultados de simulações obtidos a partir da mesma. Os resultados são comparados com simulações análogas realizadas no software *CIVA*®, Calmon et al. (2006). Esse programa corresponde a uma ferramenta comercial para a simulação de sistemas de ultrassom, desenvolvida pelo órgão francês *CEA*, que realiza estudos na área energética, principalmente em energia nuclear e renovável. O interesse é comparar os resultados a partir de uma ferramenta comercial disponível no mercado, demonstrando que a solução desenvolvida possui interesse prático e é capaz de oferecer análises consistentes.

Considera-se o caso da propagação em um meio composto, que corresponde a uma aplicação recorrente na indústria. O caso da propagação em um meio simples, é apresentado no anexo B. Em geral, são realizados testes de imersão para as três configurações de transdutor implementadas. O transdutor encontra-se imerso em água e o campo acústico é traçado no interior de um componente metálico.

Além disso, realiza-se um estudo da determinação da lei de atraso de transdutores *arrays*. Para isso, os algoritmos de determinação das leis de atraso apresentados no capítulo 2 são comparados, levando-se em consideração o erro em relação à solução exata, dada pelo princípio de Fermat, e também o tempo de cálculo.

### 4.1 Apresentação da Ferramenta de Simulação

A ferramenta para simulação do campo acústico em excitação contínua é apresentada na figura 4.1. Entre outras funcionalidades, para um dado problema, ela permite realizar a seleção e modelagem dos meios de propagação e dos transdutores ultrassônicos. Cada etapa da modelagem possui um ambiente dedicado. A simulação pode ser realizada para análise de casos bi ou tridimensionais. A visualização dos resultados pode ser realizada em escala linear ou logarítmica, conforme especificação do usuário. A ferramenta também possui funcionalidades que permitem carregar e salvar dados das análises realizadas.



**Figura 4.1:** Ferramenta para simulação do campo acústico, ambiente gráfico.

Em particular, para um transdutor *array*, é possível escolher entre três métodos para determinação das leis de atraso. Além disso, o usuário é capaz de alterar a posição focal do transdutor *array* diretamente no gráfico de um campo acústico previamente traçado, permitindo análises mais rápidas e dinâmicas. O anexo A apresenta um tutorial de utilização da ferramenta.

## 4.2 Testes de imersão

Os resultados são apresentados para o caso de propagação em um meio composto no modo longitudinal. Considera-se um transdutor imerso em água e uma peça metálica de aço. A interface de separação corresponde à face do sólido, em contato com a água. São testados os casos para o transdutor circular plano, circular côncavo e *array*. Primeiramente, o sólido é considerado um cubo. Em seguida, analisa-se o caso de uma interface curvilínea, traçando-se o campo acústico no interior de um cilindro.

As propriedades dos meios de propagação são dadas na tabela 4.1. Elas correspondem as propriedades utilizadas no software *CIVA*®, que possui dois modos de trabalho *Beam Computation* e *Defect Response*. Para o presente trabalho, será utilizado o pri-

**Tabela 4.1:** Propriedades físicas dos meios de propagação para os testes de imersão

Meio	$c_l(m/s)$	$c_s(m/s)$	$\rho_0(kg/m^3)$
Água	1483	-	1000
Aço	5900	3230	7800

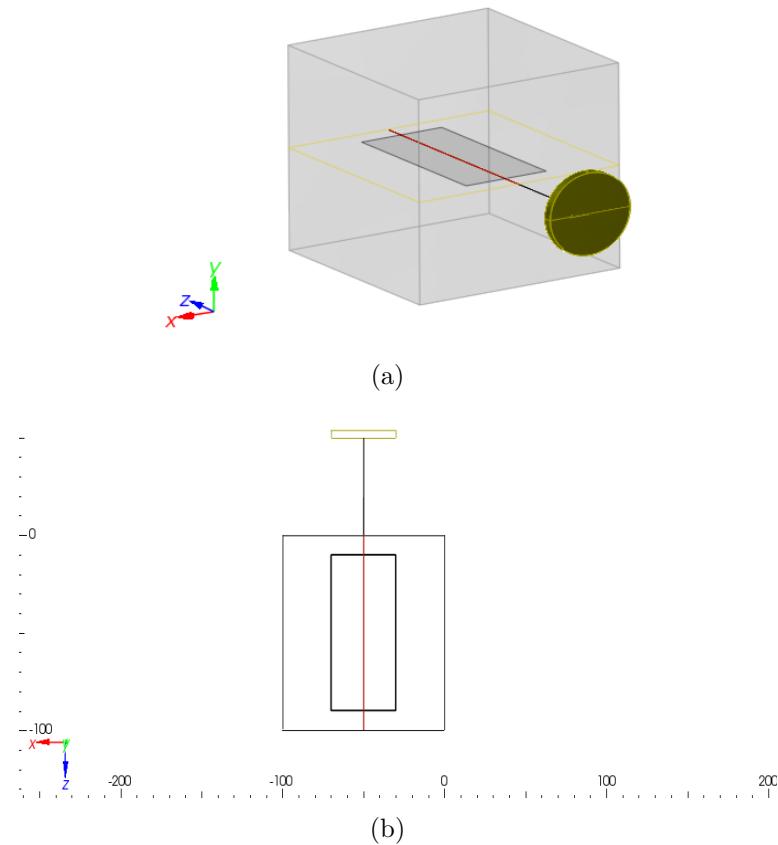
meiro modo. Os testes são configurados a partir das instruções fornecidas no tutorial do programa. Para as simulações, não são considerados efeitos de atenuação dos meios de propagação.

#### 4.2.1 Transdutor circular plano

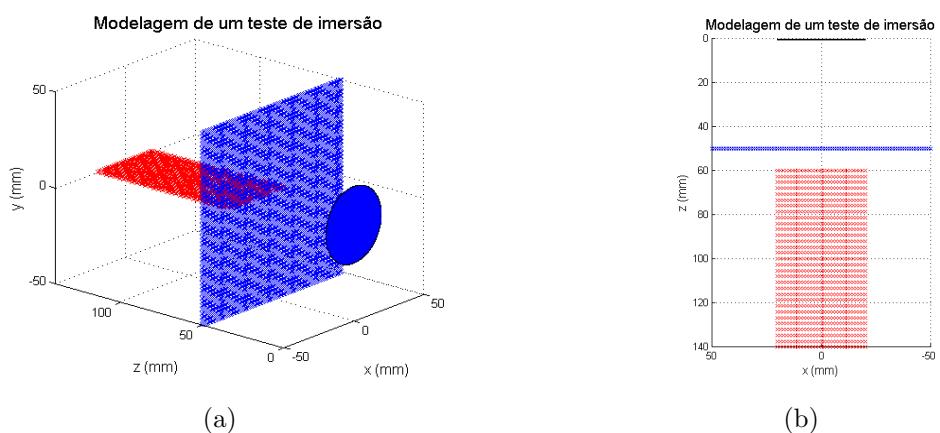
A figura 4.2 mostra a modelagem do problema utilizando o software *CIVA*®. A peça metálica consiste em um cubo de dimensão  $l = 100mm$ . O sólido é considerado um meio de propagação isotrópico e homogêneo. O transdutor consiste em um transdutor circular plano monoelemento de dimensão  $r = 20mm$ . A frequência de excitação em modo contínuo é de 2MHz. O transdutor encontra-se a uma distância de 50mm do sólido. O campo acústico é traçado no eixo acústico do transdutor, no plano  $xz$ , em uma região de dimensões  $40 \times 80mm$ , delimitada no interior do sólido. A figura 4.2(b) indica a configuração utilizada. A região de interesse é discretizada a partir do número de divisões horizontal e vertical. Adotaram-se 40 e 80 divisões respectivamente, o que corresponde a uma malha quadrada cujo passo é de 1mm.

De maneira análoga, o problema é modelado utilizando-se a ferramenta em *MATLAB*®, como ilustra a figura 4.3. Adicionalmente, em *MATLAB*®, o transdutor e a interface são discretizados conforme a tabela 4.2.

A discretização dos emissores, no caso o transdutor e a interface, será função do comprimento de onda nos meios de propagação. Para a água  $\lambda_1 = 0,75mm$  e para o aço  $\lambda_2 = 2,95mm$ . Em relação ao passo do transdutor,  $\lambda_1$  é da mesma ordem de grandeza, porém superior, e  $\lambda_2$  cerca de quatro vezes. Em relação à interface as ordens de grandeza são da ordem da unidade. Optou-se por não adotar uma discretização maior nesse caso, uma vez que isso provocaria um aumento desnecessário do tempo de cálculo sem proporcionar ganhos reais aos resultados.



**Figura 4.2:** Modelagem de um teste de imersão em *CIVA*®, transdutor circular plano:  
 (a) Vista em perspectiva, (b) Vista superior.



**Figura 4.3:** Modelagem de um teste de imersão em *MATLAB*®, transdutor circular plano: (b) Vista em perspectiva, (b) Vista superior.

**Tabela 4.2:** Configuração para teste de imersão, transdutor circular plano.

Parâmetro	Número de divisões	Discretização (mm)
$r(mm)$	60	0,66
$x_{int}$	100	1
$y_{int}$	100	1
$z_{int}$	100	1

Os gráficos de ambas as simulações são apresentados na figura 4.4. Ambos são dados em escala logarítmica, adotando um limiar em  $-19,4dB$ . O campo de pressão é normalizado, sendo  $0dB$  o valor correspondendo ao pico de amplitude na região delimitada.

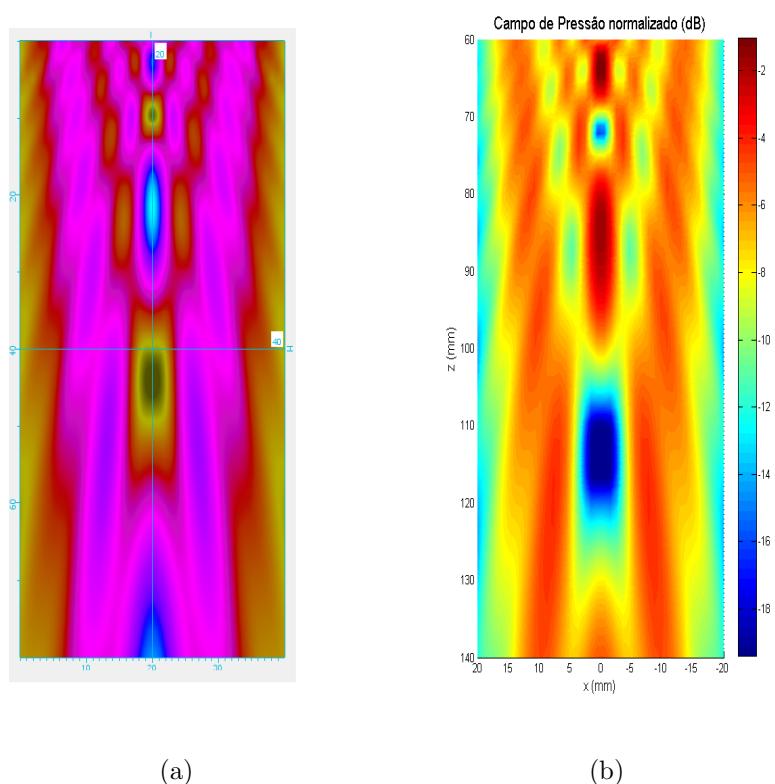
Atenta-se para o fato de que no programa *CIVA*<sup>®</sup> as cotas dos gráficos são dadas em coordenadas relativas à região delimitada para o cálculo, iniciando-se sempre em zero. Já em *MATLAB*<sup>®</sup> as coordenadas são dadas em relação ao centro do transdutor.

A partir da figura 4.4 nota-se que os lóbulos de intensidade máxima e mínima possuem formas semelhantes a exceto de uma defasagem entre eles. Por exemplo, na imagem 4.4(a), há um mínimo próximo a cota de  $z = 45mm$ . Na figura 4.4(b) o mesmo mínimo ocorre próximo a cota  $z = 115mm$ , o que seria equivalente a cota  $z = 55mm$  em 4.4(a).

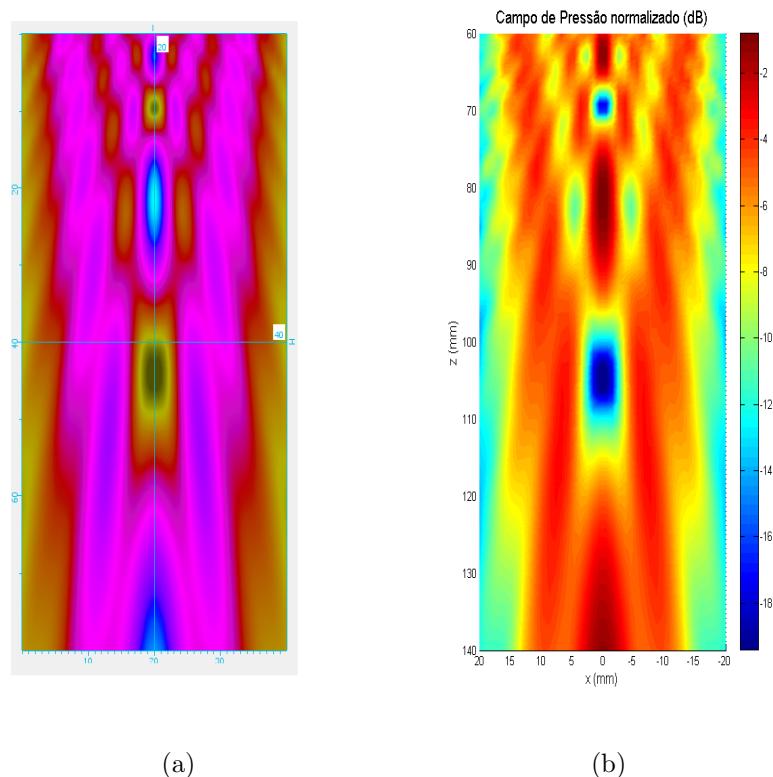
Analizando-se esse resultado, conclui-se que a defasagem é devida aos coeficientes de transmissão, introduzidos pela matriz  $\mathbf{T}^{tl}$ , conforme a implementação apresentada no capítulo 3. Isso é verificado realizando-se uma nova simulação e atribuindo valor unitário para todos os coeficientes de transmissão. Os resultados são apresentados na figura 4.5.

O software *CIVA*<sup>®</sup> possui uma opção para habilitar a conversão de modo. Porém, os resultados mantém-se idênticos aos observados na figura 4.4(a). Sendo assim, conclui-se que o programa não trabalha com o modelo implementado neste trabalho para a conversão de modo. Isso é corroborado quando comparam-se as respostas obtidas para o modo transversal, dadas na figura 4.6, onde o limiar da escala logarítmica é de  $-20dB$ .

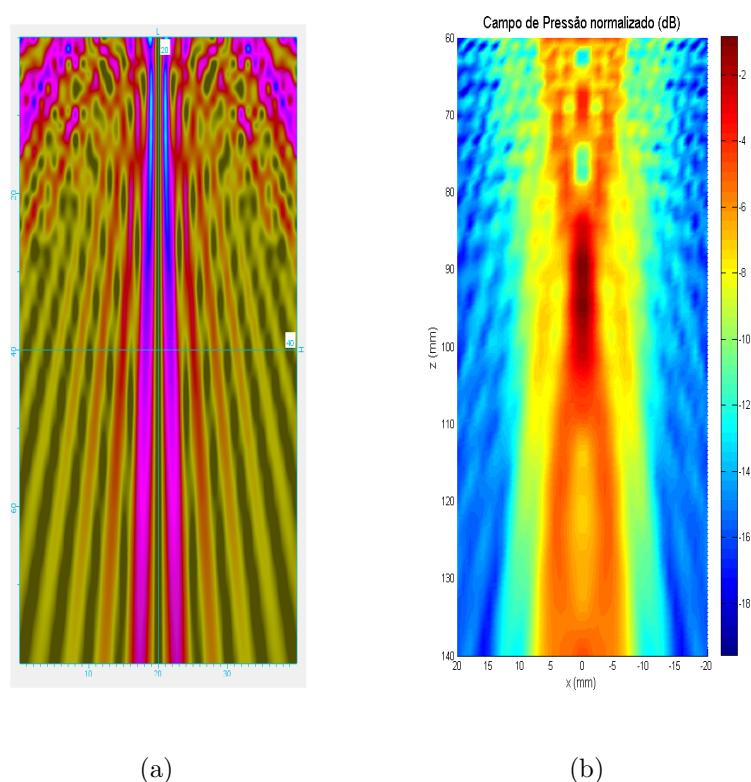
As diferenças nas curvas sugerem que o modelo para conversão de modo empregado em *CIVA*<sup>®</sup> difere do modelo empregado em *MATLAB*<sup>®</sup>. O software não indica qual



**Figura 4.4:** Resultados para teste de imersão, transdutor circular plano em modo longitudinal: (a) Simulação em *CIVA*®, (b) Simulação em *MATLAB*®.



**Figura 4.5:** Resultados para teste de imersão, transdutor circular plano modo longitudinal sem conversão de modo: (a) Simulação em *CIVA*<sup>®</sup>, (b) Simulação em *MATLAB*<sup>®</sup>.



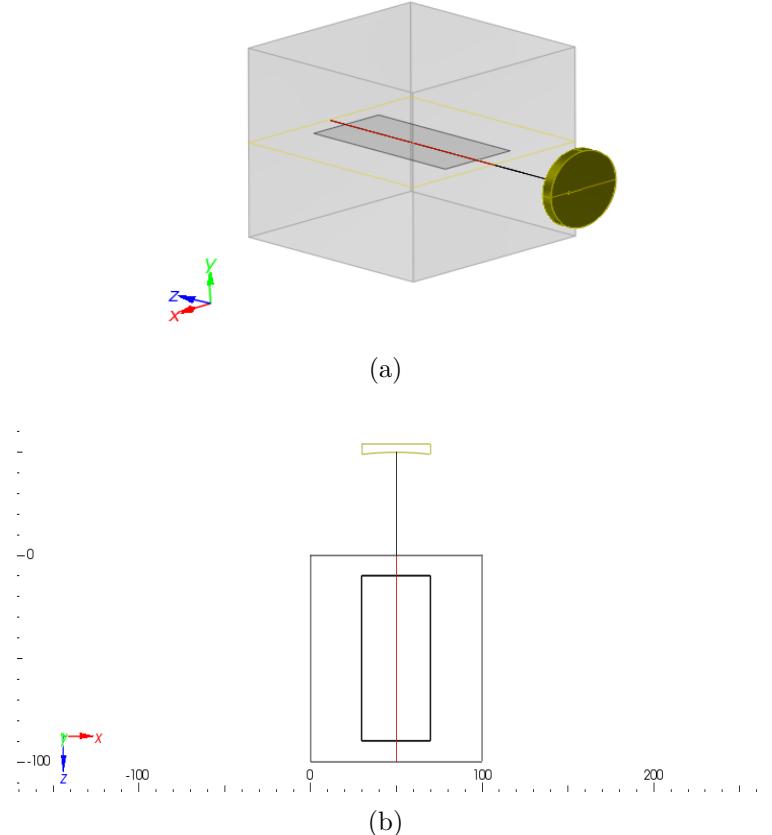
**Figura 4.6:** Resultados para teste de imersão, modo transversal: (a) Simulação em *CIVA*<sup>®</sup>, (b) Simulação em *MATLAB*<sup>®</sup>.

modelo é efetivamente empregado para a realização de novas comparações. Por outro lado, no modo longitudinal, observa-se que a solução obtida a partir do *MTM* para um transdutor circular plano é muito próxima da versão implementada comercialmente.

#### 4.2.2 Transdutor circular côncavo

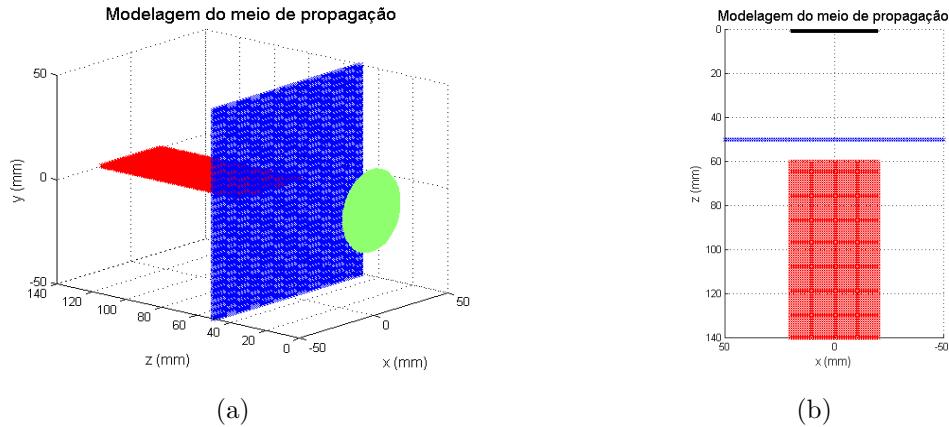
O mesmo tipo de simulação é realizada para o caso de um transdutor circular côncavo. A dimensão  $r = 20mm$  é mantida. O raio de curvatura do transdutor será de  $R = 200mm$ .

A modelagem dos meios de propagação é idêntica ao caso do transdutor circular plano. As figuras 4.7 e 4.8 mostram os resultados dessa etapa.



**Figura 4.7:** Modelagem de um teste de imersão em *CIVA*®, transdutor circular côncavo: (a) Vista em perspectiva, (b) Vista superior.

O transdutor é discretizado da forma apresentada na tabela 4.3. O grande número de divisões é devido ao raio de curvatura utilizado. Para  $R = 200mm$  e um passo de  $0,18^\circ$ , o comprimento de arco associado é da ordem de  $0,62mm$ , o que equivale a aproxi-



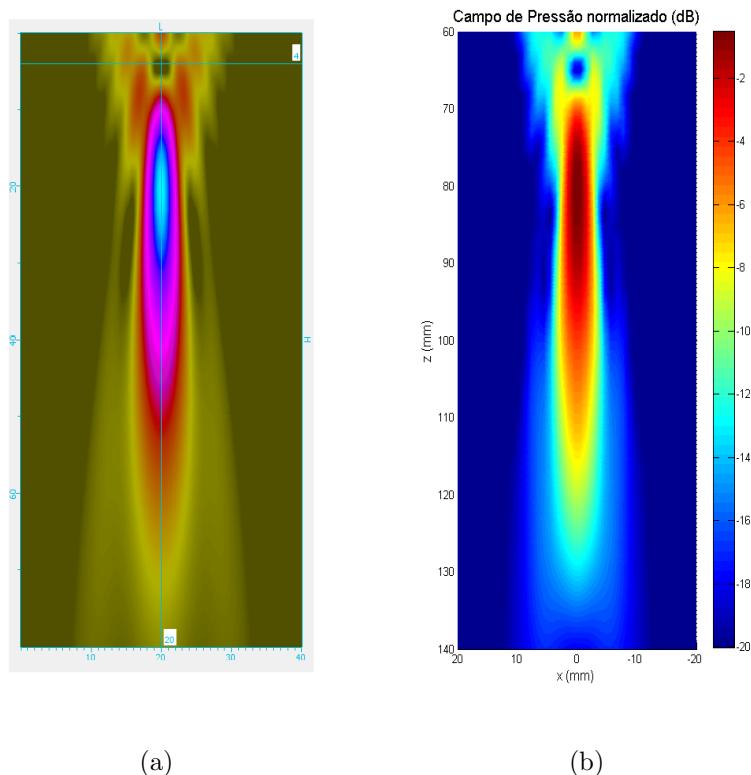
**Figura 4.8:** Modelagem de um teste de imersão utilizando a ferramenta em *MATLAB*<sup>®</sup>, transdutor circular côncavo: (b) Vista em perspectiva, (b) Vista superior.

**Tabela 4.3:** Configuração para teste de imersão com transdutor circular côncavo

Parâmetro	Ângulo (°)	Número de divisões	Passo (°)
$\varphi$	180	1000	0,18
$\theta$	180	1000	0,18

madamente um terço do comprimento de onda  $\lambda_2$ .

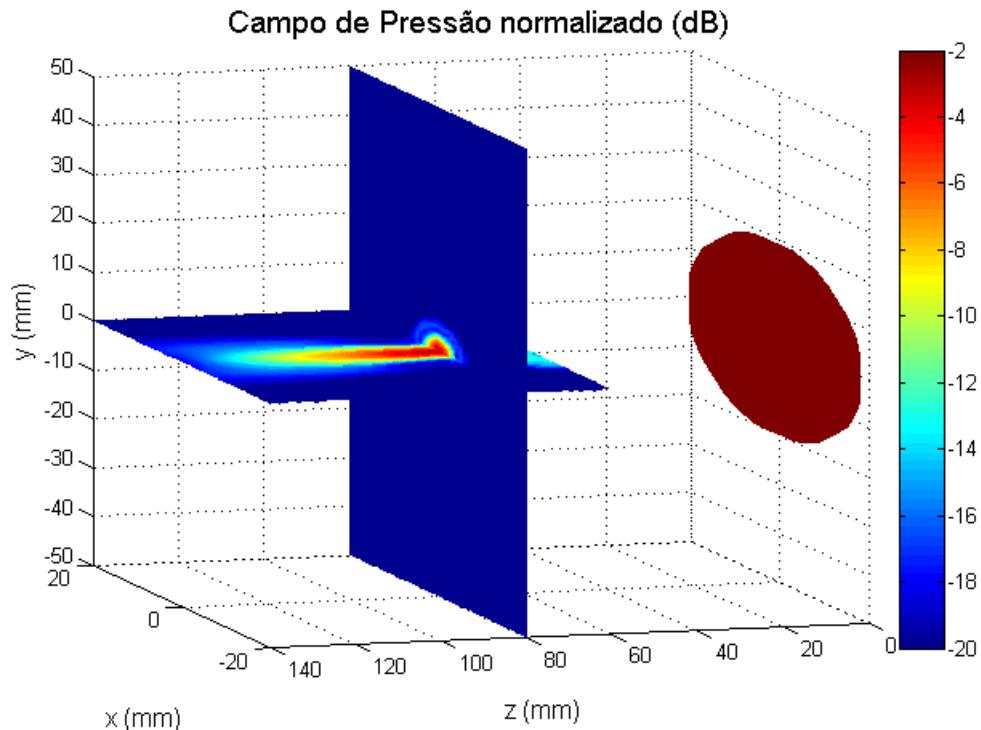
Os resultados da simulação são apresentados nas figuras dadas em 4.9. Ao contrário do transdutor circular, os coeficientes de transmissão tem menos influência no campo acústico resultante. Não há uma defasagem significativa entre as curvas apresentadas em 4.9(a) e 4.9(b).



**Figura 4.9:** Resultados para teste de imersão, transdutor circular côncavo em modo longitudinal: (a) Simulação em *CIVA*<sup>®</sup>, (b) Simulação em *MATLAB*<sup>®</sup>.

Alternativamente, o campo acústico pode ser traçado em um plano perpendicular ao plano de emissão. O interesse nesse tipo de análise é determinar a largura do feixe acústico emitido pelo transdutor, o que constitui um parâmetro de importância no dimensionamento de um sistema de ultrassom. A ferramenta em *MATLAB*<sup>®</sup> permite a análise tri-dimensional do problema, como apresentado na figura 4.10. O plano em *xy* é selecionado para a cota  $z = 80\text{mm}$  em relação ao sistema de coordenadas posicionado no centro do transdutor. O passo de discretização é mantido em  $1\text{mm}$ . O resultado obtido é comparado com uma simulação análoga realizada em *CIVA*<sup>®</sup>. Os resultados são comparados na figura 4.11, onde o limiar da curva logarítmica é escolhido em  $-40\text{dB}$ . As figuras mostram concordância entre os resultados obtidos. Tanto no caso em duas,

como em três dimensões os padrões observados para a formação de mínimos e máximos são muito próximos.



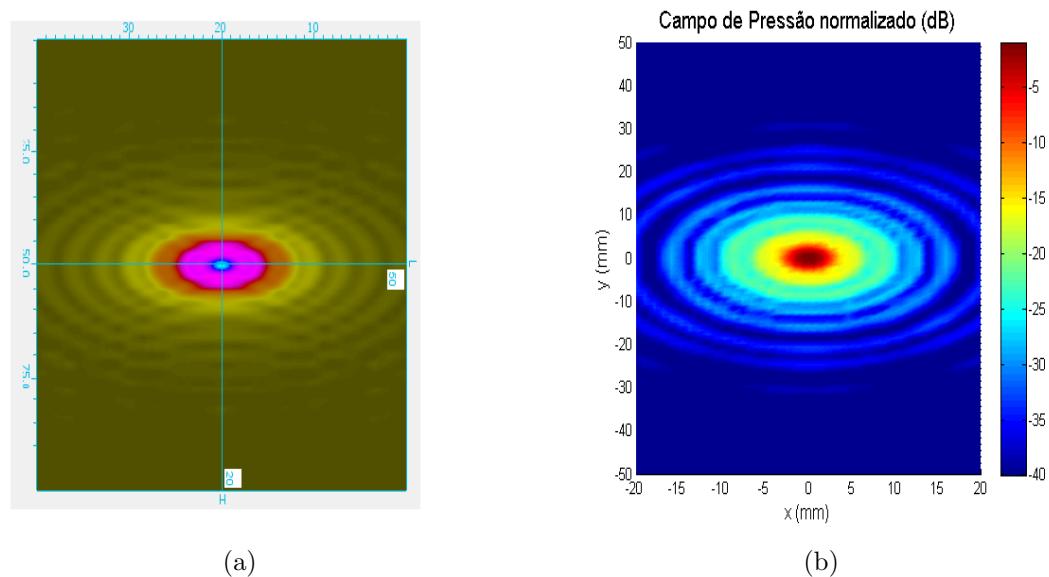
**Figura 4.10:** Campo acústico em três dimensões para um transdutor circular côncavo.

#### 4.2.3 Transdutor *array*

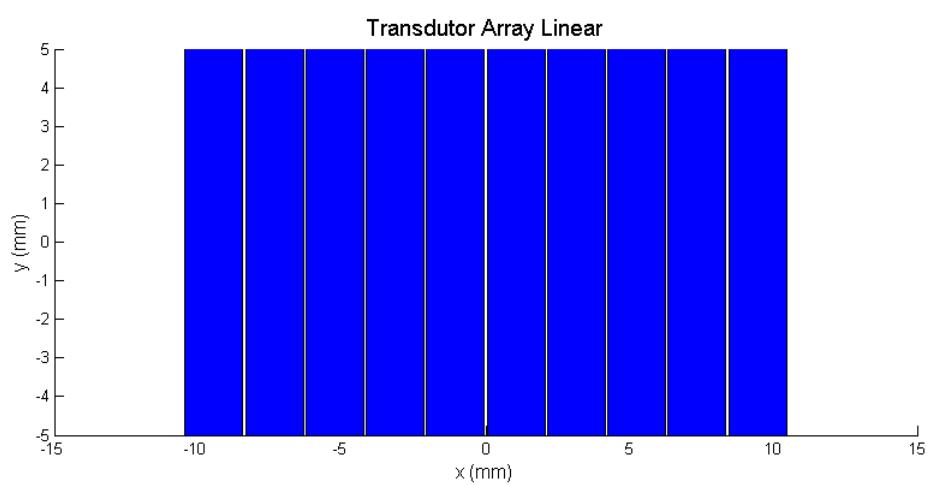
A análise é repetida novamente para o caso de um transdutor *array*. O transdutor é configurado com 10 elementos, onde o espaçamento entre cada elemento é de  $0,1mm$ , e a dimensão de cada elemento é de  $2 \times 10mm$ . A figura 4.12 mostra o modelo do transdutor utilizado.

Como anteriormente, o problema é modelado em *CIVA*® e em *MATLAB*®. Porém, ao contrário dos casos anteriores, o transdutor será configurado com uma lei de atraso calculada em função de um foco definido na região de interesse. Inicialmente, o foco é fixado no eixo de simetria do transdutor, ponto  $(x_f, y_f, z_f) = (0, 0, 100mm)$ , em relação ao sistema de coordenadas do mesmo.

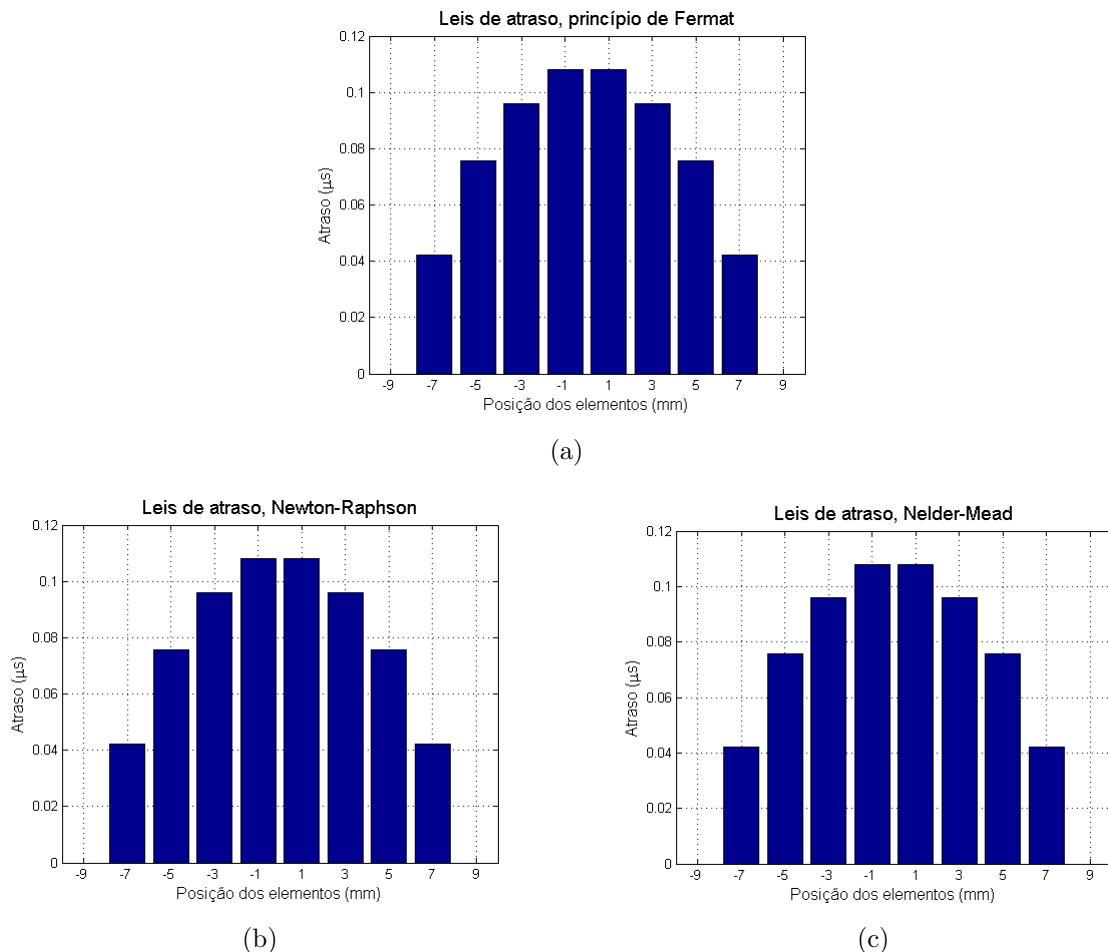
As leis de atraso são determinadas para o centro de cada transdutor, utilizando os três métodos descritos no capítulo 2. Os resultados são apresentados na figura 4.13. Uma discussão mais detalhada a respeito da determinação das leis de atraso será realizada na sequência dessa seção.



**Figura 4.11:** Resultados para teste de imersão, transdutor circular côncavo em modo longitudinal: (a) Simulação em *CIVA*®, (b) Simulação em *MATLAB*®.



**Figura 4.12:** Modelo do transdutor array para teste de imersão.

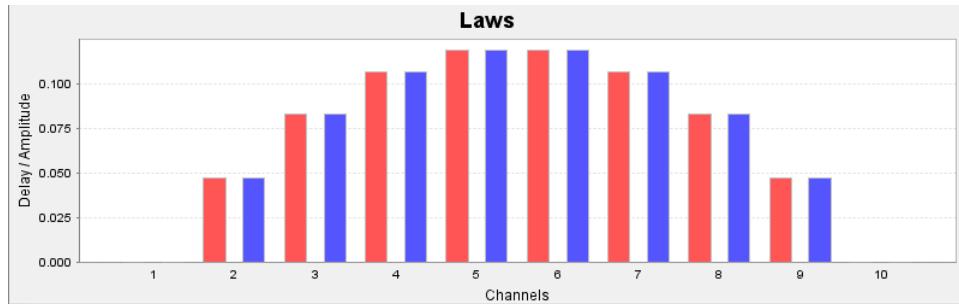


**Figura 4.13:** Determinação das leis de atraso em *MATLAB*<sup>®</sup> para transdutor *array* focalizado em  $(x_f, y_f, z_f) = (0, 0, 100\text{mm})$ : (a) Princípio de Fermat, (b) Método de Newton-Raphson, (c) Método de Nelder-Mead.

**Tabela 4.4:** Configuração para teste de imersão com transdutor *array*

Parâmetro	Número de divisões	Passo (mm)
$a$	8	0.25
$b$	40	0.25

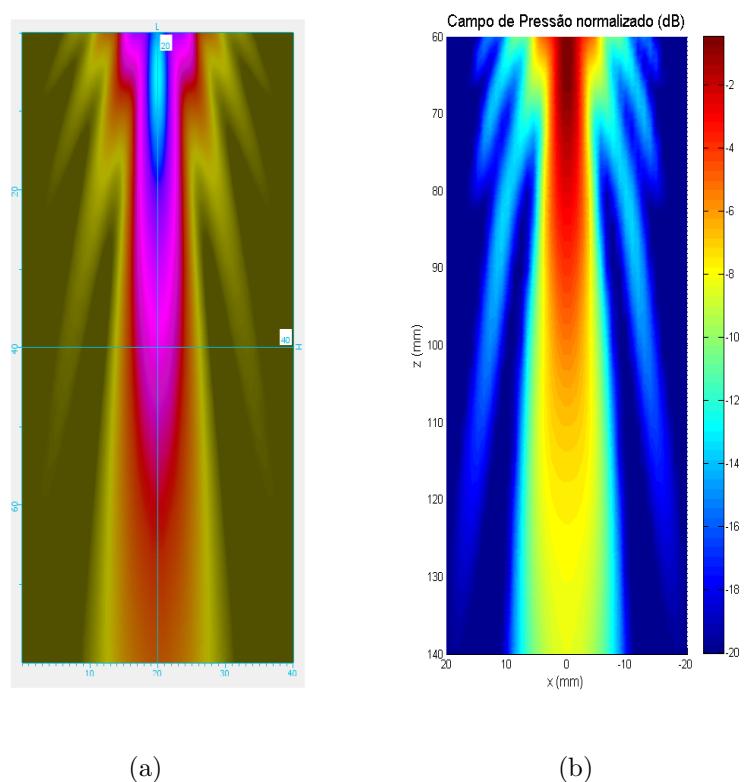
Observa-se que todos os algoritmos fornecem o mesmo resultado. Essas curvas podem ser comparadas com as leis de atraso calculadas em *CIVA*®. Os resultados são mostrados na figura 4.14. As leis de atraso determinadas são praticamente equivalentes para os dois programas.

**Figura 4.14:** Determinação das leis de atraso em *CIVA*® para transdutor *array* focalizado em  $(x_f, y_f, z_f) = (0, 0, 100mm)$ .

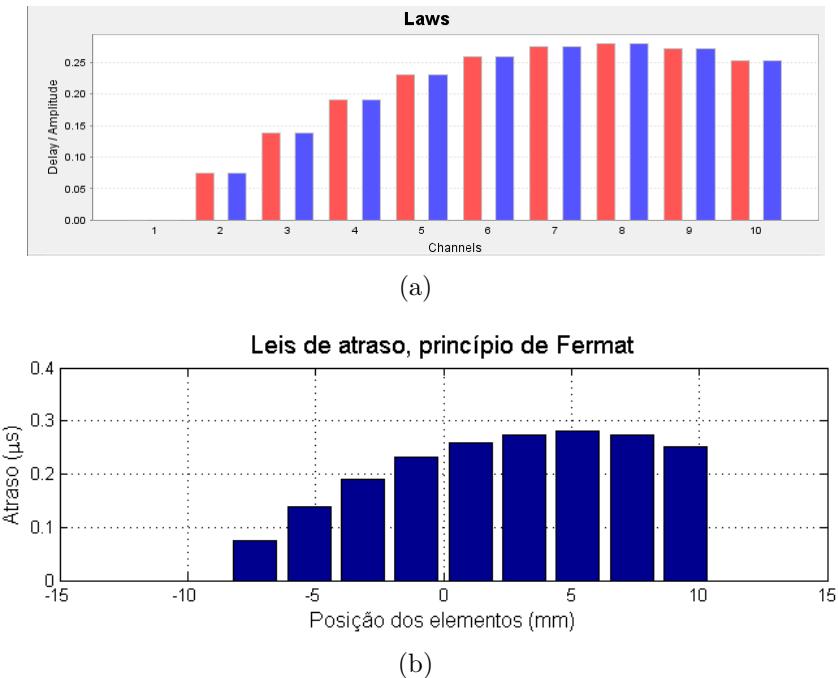
Uma vez determinadas as leis de atraso, o campo acústico pode ser calculado. A discretização do transdutor em *MATLAB*® é dada na tabela 4.4. A figura 4.15 mostra uma comparação entre as curvas obtidas nos dois *softwares*.

Em um segundo momento, o foco é posicionado em  $(x_f, y_f, z_f) = (5, 0, 100mm)$ . As leis de atraso são dadas na figura 4.16, onde utiliza-se a solução exata para o cálculo em *MATLAB*®. Os resultados são apresentados na figura 4.17. Para esse mesmo teste, o campo acústico pode ser observado em um plano perpendicular ao eixo de transmissão. A figura 4.18 ilustra o resultado no espaço tridimensional para um plano localizado em  $z = 70mm$ . O novo plano é discretizado com passo de  $1mm$ . Os resultados são apresentados em 4.19, onde as curvas são limiarizadas em  $-20dB$ .

As respostas obtidas nos casos bi e tridimensional são semelhantes à exceção de algumas deformações observadas para o caso onde o feixe acústico é defletido, com o foco localizado em  $(x_f, y_f, z_f) = (5, 0, 100mm)$ . Na figura 4.17, o feixe principal de emissão



**Figura 4.15:** Resultados para teste de imersão, transdutor *array* em modo longitudinal focalizado em  $(x_f, y_f, z_f) = (0, 0, 100\text{mm})$ : (a) Simulação em *CIVA*®, (b) Simulação em *MATLAB*®.



**Figura 4.16:** Determinação das leis de atraso para transdutor *array* focalizado em  $(x_f, y_f, z_f) = (5, 0, 100\text{mm})$ : (a) Cálculo em *CIVA*<sup>®</sup>, (b) Cálculo em *MATLAB*<sup>®</sup>, princípio de Fermat.

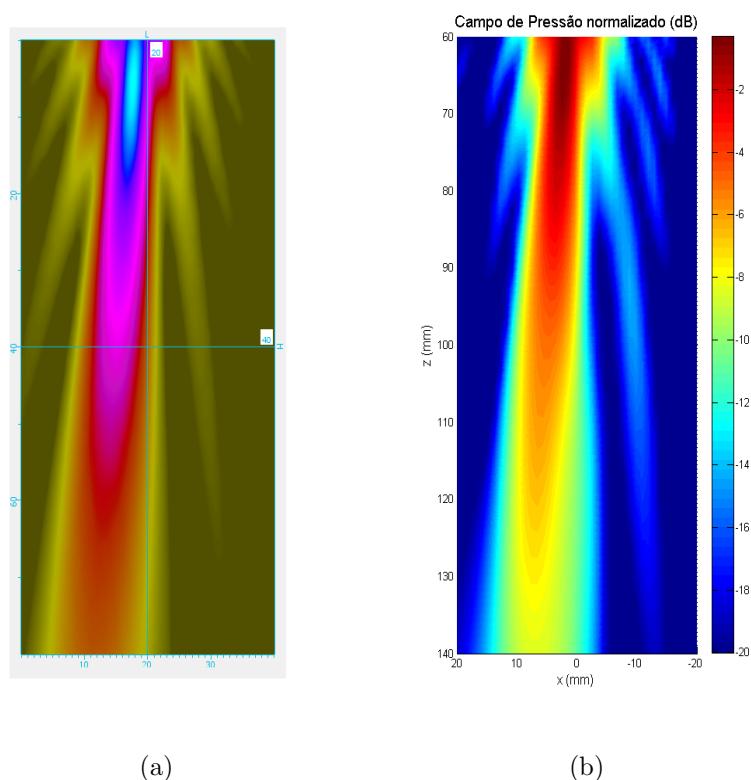
possui a mesma forma para ambas as figuras, porém os lóbulos laterais da figura 4.17(b) estão deformados em relação à figura 4.17(a). A mesma análise aplica-se para o caso da figura 4.19. Como apontado anteriormente, esse fato pode ser atribuído às diferenças nos algoritmos utilizados para realizar a conversão de modo.

#### 4.2.4 Interface circular

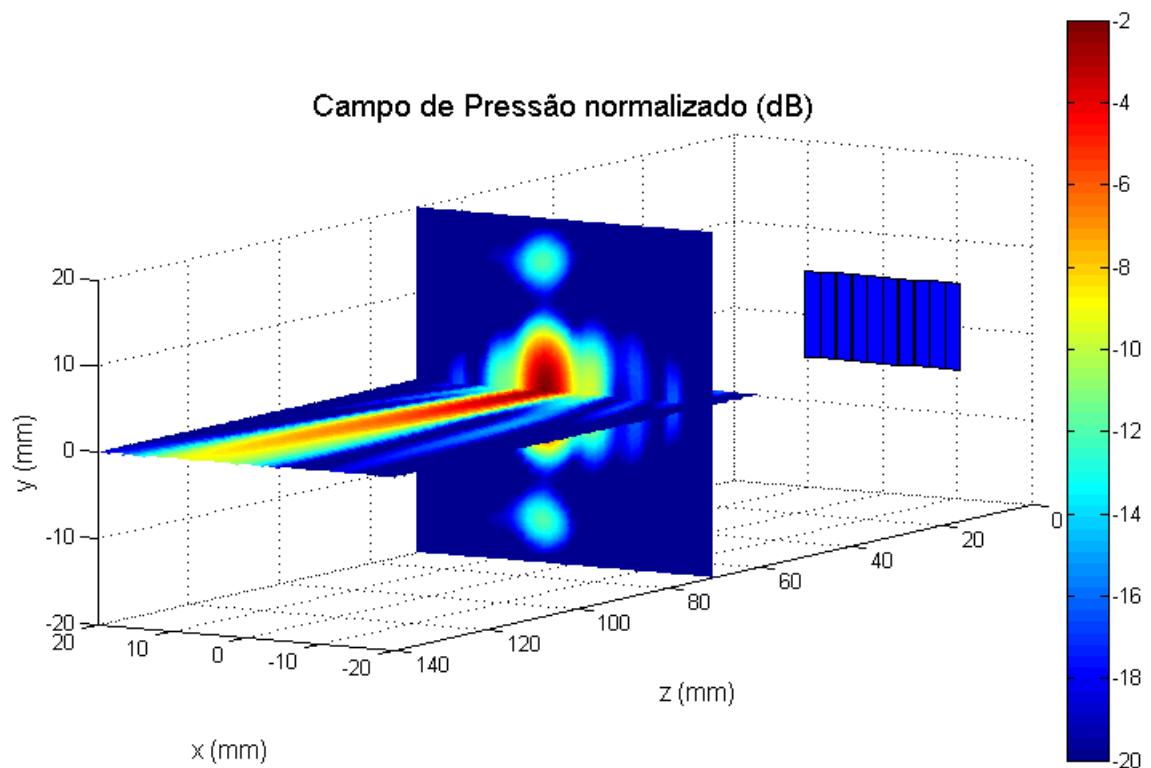
O último teste de imersão proposto consiste na avaliação do campo acústico no interior de um cilindro, simulando a situação de uma interface curvilínea. Considera-se o mesmo transdutor *array* utilizado anteriormente.

O cilindro consiste em um corpo maciço de raio  $r = 40\text{mm}$  e comprimento  $L = 100\text{mm}$ . A região de interesse é dada por um plano de dimensões  $20 \times 30\text{mm}$ , localizada no interior do cilindro. O problema é modelado conforme a figura 4.20. O passo de discretização tanto para a região de interesse, como para a interface é de  $1\text{mm}$ . O foco é posicionado em  $(x_f, y_f, z_f) = (-5, 0, 40\text{mm})$ . As leis de atraso calculadas em ambos os programas são dadas em 4.21.

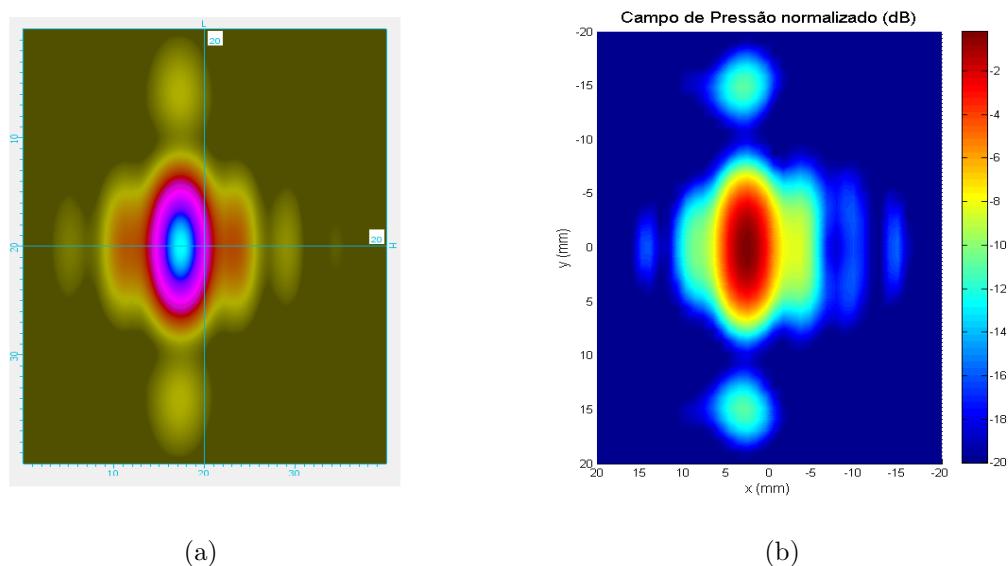
O campo acústico resultante é apresentado na figura 4.22.



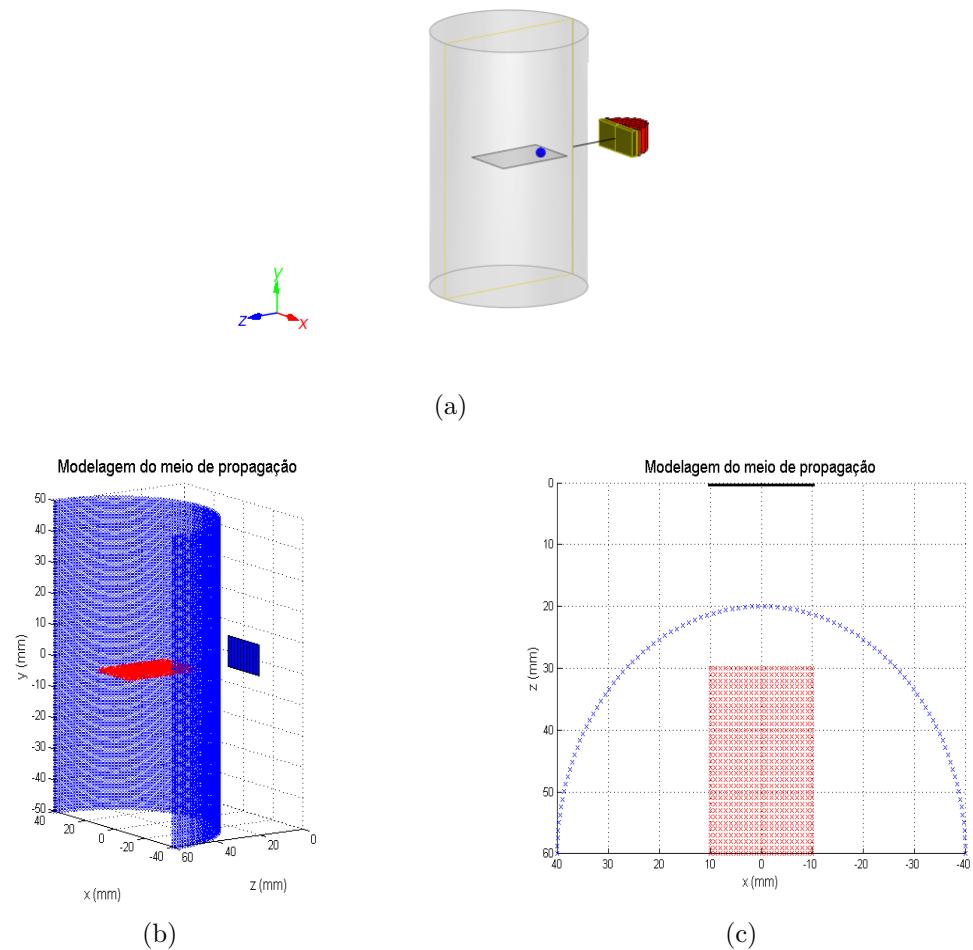
**Figura 4.17:** Resultados para teste de imersão, transdutor *array* modo longitudinal focalizado em  $(x_f, y_f, z_f) = (5, 0, 100\text{mm})$ : (a) Simulação em *CIVA*®, (b) Simulação em *MATLAB*®.



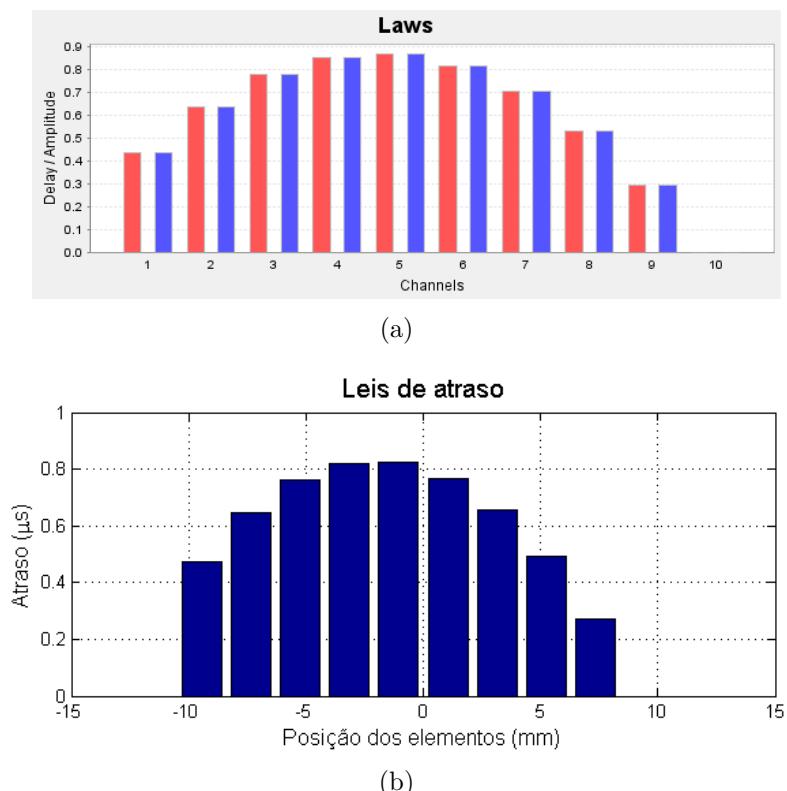
**Figura 4.18:** Campo acústico em três dimensões para um transdutor *array*



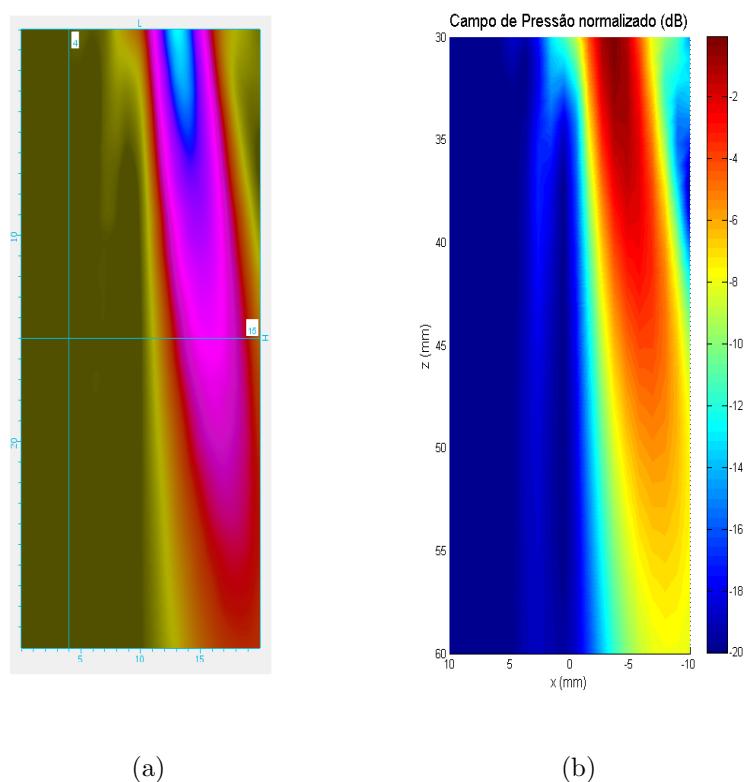
**Figura 4.19:** Resultados para teste de imersão, transdutor *array* em modo longitudinal focalizado em  $(x_f, y_f, z_f) = (5, 0, 100) \text{ mm}$ , plano transversal  $z = 70 \text{ mm}$ : (a) Simulação em *CIVA*®, (b) Simulação em *MATLAB*®.



**Figura 4.20:** Determinação das leis de atraso em *MATLAB*<sup>®</sup>: (a) Modelo em *CIVA*<sup>®</sup>, (b) Modelo em *MATLAB*<sup>®</sup>, perspectiva, (c) Modelo em *MATLAB*<sup>®</sup>, vista superior.



**Figura 4.21:** Determinação das leis de atraso para transdutor *array* focalizado em  $(x_f, y_f, z_f) = (-5, 0, 40)mm$ : (a) Cálculo em *CIVA*®, (b) Cálculo em *MATLAB*®, princípio de Fermat.



**Figura 4.22:** Resultados para teste de imersão, transdutor array em modo longitudinal focalizado em  $(x_f, y_f, z_f) = (-5, 0, 40)mm$ , para interface curvilínea: (a) Simulação em *CIVA*<sup>®</sup>, (b) Simulação em *MATLAB*<sup>®</sup>.

**Tabela 4.5:** Propriedades físicas dos meios de propagação para avaliação do cálculo das leis de atraso de um transdutor *array*

Meio	$c_l(m/s)$	$c_s(m/s)$	$\rho_0(kg/m^3)$
Água	1500	-	1000
Alumínio	6420	3040	7800

As leis de atraso dadas na figura 4.21 se equivalem. O campo acústico traçado em 4.22 também é muito próximo para as duas curvas. Como observado no caso de uma interface retilínea, os lóbulos laterais da figura 4.22(b) também são deformados em relação à figura 4.22(b). Nesse caso, o lóbulo lateral esquerdo é mais alongado.

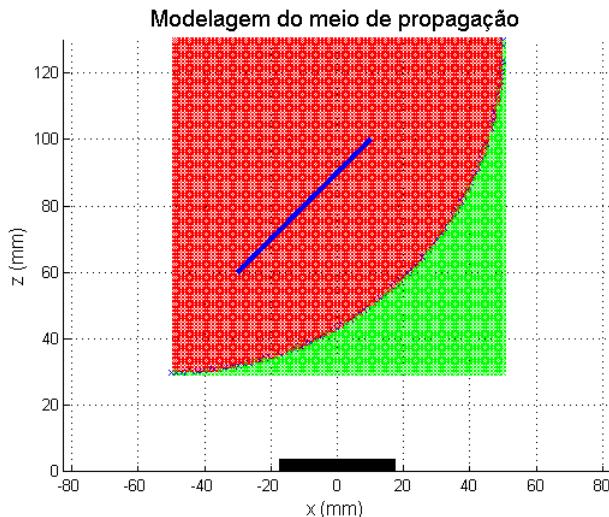
### 4.3 Métodos de determinação das leis de atraso

Utilizando uma abordagem análoga ao trabalho de Parrilla et al. (2008), pretende-se analisar a performance dos algoritmos propostos para a determinação das leis de atraso de um transdutor array.

Para isso, considera-se o problema dado na figura 4.23, onde os pontos em vermelho indicam o interior de um cilindro de alumínio, cujo raio é  $r = 100mm$ . A interface curva é discretizada em 100 pontos. As leis de atraso são determinadas para a propagação a partir de um transdutor imerso em água. As propriedades dos materiais empregados encontram-se na tabela 4.5. O transdutor consiste em um *array* de 32 elementos espaçados de  $0,1mm$ , onde cada elemento possui dimensões de  $1 \times 5mm$ .

A linha na figura 4.23 indica quais posições focais serão atribuídas. Ela é dada pelo lugar geométrico dos pontos tal que  $z = 90 + x$ , com  $x \in [-30, 10]$  e  $z \in [60, 100]$ , onde as unidades são dadas em  $mm$ . Cada vetor  $x$  e  $z$  contendo os pontos da reta é discretizado com passo  $1mm$ . A ideia do teste a ser realizado é calcular a lei de atraso associada a cada foco, a partir dos três métodos, e avaliá-los em função do erro médio, em relação à solução dada pelo princípio de *Fermat*, conforme equação 4.1, e do tempo de cálculo.

$$e_m = \frac{1}{N} \sum_{i=1}^N |d_i^f - \hat{d}_i| \quad (4.1)$$



**Figura 4.23:** Modelagem para avaliação dos algoritmos de lei de atraso de um transdutor *array*.

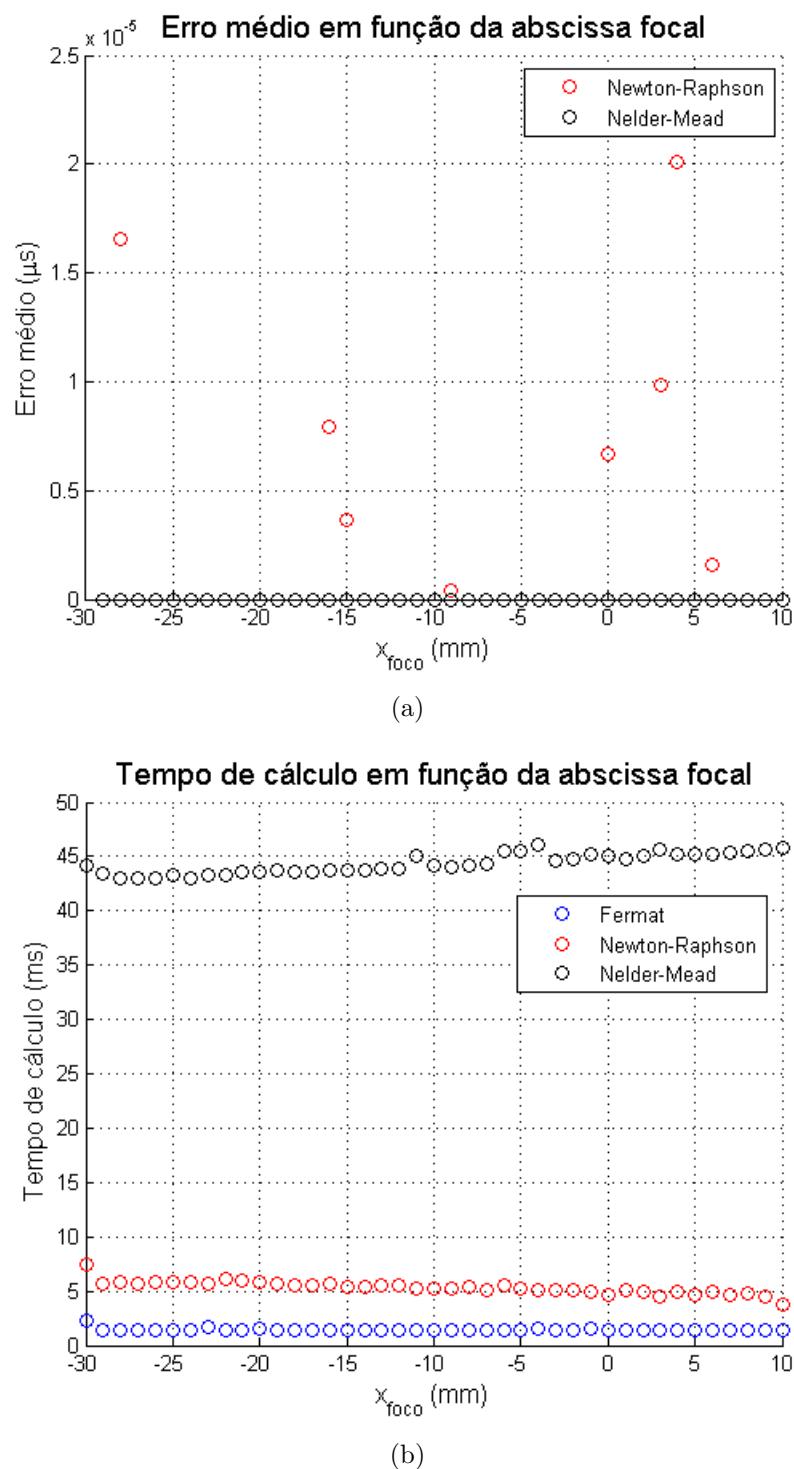
onde  $N$  é o número de elementos do transdutor,  $d_i^f$  é o atraso associado ao  $i$ -ésimo elemento, calculado a partir do princípio de *Fermat* e  $\hat{d}_i$  é o atraso associado ao  $i$ -ésimo elemento calculado a partir do método de *Newton-Raphson* ou *Nelder-Mead*.

A tolerância utilizada para o algoritmo de *Newton-Raphson* é  $10^{-8}$ . O limite de iterações em cada algoritmo aproximativo é fixado em 10. O chute inicial é o mesmo para ambos os casos e corresponde ao ponto médio da interface. Os testes fornecem os resultados observados na figura 4.24.

A figura 4.24(a) mostra que o algoritmo de *Nelder-Mead* tende a determinar a resposta ótima para o problema. As soluções obtidas são idênticas às soluções determinadas pelo princípio de *Fermat*. Para o algoritmo de *Newton-Raphson*, observa-se que algumas soluções possuem resíduos. Porém, de maneira geral, o erro obtido é da ordem de  $10^5$  menor do que a solução exata <sup>1</sup>.

A análise do tempo de cálculo é mais delicada e os resultados apresentados são apenas qualitativos; o método aplicado não permite inferir a respeito da eficiência dos algoritmos em relação a uma aplicação em tempo real, como é proposto em Parrilla et al. (2008). Por exemplo, a solução exata dada pelo princípio de *Fermat* possui menor tempo de cálculo dentre todos os métodos, o que pode não ser verificado na prática, a partir do que sugere a literatura.

<sup>1</sup>As leis de atraso são da ordem de  $\mu s$



**Figura 4.24:** Determinação das leis de atraso, comparação entre métodos: (a) Erro médio em relação ao princípio de Fermat, (b) Tempo de cálculo.

O algoritmo de *Newton-Raphson* possui um tempo de cálculo muito próximo da solução exata. Já o método de *Nelder-Mead* possui tempo de cálculo de cerca de 7 vezes o tempo gasto pela solução exata. Evidentemente, esses resultados tendem a variar em função dos tamanhos das amostras, do problema tratado, da maneira como o algoritmo é implementado e da plataforma de cálculo.

Levando em conta todos esses fatores, para o problema estudado, conclui-se que o algoritmo de *Newton-Raphson* possui um melhor compromisso entre tempo de cálculo e erro médio, uma vez que o mesmo é capaz de manter um baixo valor para o erro médio, a despeito de algumas oscilações, preservando um tempo de cálculo baixo. O algoritmo de *Nelder-Mead* fornece a solução ótima do problema com um tempo de cálculo excessivo, o que pode ser um fator negativo no caso de uma aplicação em tempo real, por exemplo.

## 4.4 Discussão dos resultados

A simulação de um teste de imersão, envolvendo a propagação em múltiplos meios, através de interfaces e incluindo a determinação das leis de atraso constitui um problema não trivial na área de ultrassom. Os resultados obtidos mostram que os padrões de emissão para todos os transdutores tratados, seja no caso de uma interface retilínea ou curva, são muito semelhantes. A resposta obtida a partir do *MTM* preserva as principais características do campo acústico obtido com *CIVA*®, mesmo com as diferenças observadas em virtude do modelo de conversão de modo empregado pelas duas abordagens.

Igualmente, as leis de atraso determinadas a partir dos três métodos propostos também fornecem soluções equivalentes em relação à versão comercial. Os resultados corroboram as conclusões obtidas por Parrilla et al. (2008). Efetivamente, o método de *Newton-Raphson* pode representar uma alternativa eficaz para a determinação de leis de atraso mantendo um compromisso entre tempo de cálculo e precisão. Alternativamente, a heurística utilizada no método de *Nelder-Mead* é capaz de determinar a solução ótima do problema, porém com maior custo computacional.

Procurou-se demonstrar que os resultados obtidos a partir da abordagem com o *MTM* podem fornecer análises tão coerentes quanto àquelas fornecidas através de um software comercial, como no caso do software *CIVA*®. O grande interesse da metodologia proposta, é contar com uma ferramenta de desenvolvimento aberta, que utiliza uma arquitetura estruturada e reprogramável, e que implementa um método capaz de determinar o

campo acústico em meios compostos de maneira rápida, o que não é realizado por outras ferramentas semelhantes, como os *toolboxes* *FOCUS*, Chen e McGough (2008) e *DREAM*, Piwakowski e Sbai (1999).

## CONCLUSÃO

Comparadas com outras áreas da engenharia, o desenvolvimento tecnológico do ultrassom é relativamente recente, datando do início do século XX. Não obstante, suas aplicações estenderam-se rapidamente em diversos meios, principalmente para a geração de imagens, tanto na medicina, como na indústria. Nesse contexto, há a necessidade de se projetar transdutores ultrassônicos. Dada a diversidade e a complexidade dos sistemas existentes, a utilização de ferramentas de simulação computacional é de grande importância para o projeto de tais transdutores.

O presente trabalho analisou uma das características de interesse no projeto desses sistemas, que corresponde ao campo acústico emitido pelos transdutores. Abordou-se a transmissão em modo de excitação contínua, em meios de propagação simples e compostos, separados por interfaces de formato arbitrário, empregando a conversão de modo, assim como o cálculo de leis de atraso de transdutores *arrays*.

A solução oferecida pelo *MTM* possui vantagens em relação a outros métodos de simulação, principalmente no que se refere ao compromisso entre consistência das soluções obtidas, custo computacional e tipos de problemas que podem ser tratados, como apontado no capítulo 2.

Os resultados obtidos mostraram-se satisfatórios em relação aos aspectos abordados. Mesmo com algumas diferenças, em razão de diferentes modelos aplicados para a conversão de modo, a comparação dos resultados com ferramentas similares, comerciais e abertas, mostra que a abordagem proposta permite análises consistentes do campo acústico para diversos casos de interesse prático.

Além disso, a implementação das etapas de modelagem e cálculo em torno de uma ferramenta gráfica confere grande flexibilidade para a solução proposta. No que se refere aos algoritmos utilizados e à programação em *MATLAB*®, o usuário necessita de pouco conhecimento prévio para sua utilização. Uma aplicação direta dessa solução é, por exemplo, como ferramenta didática. Também pode ser utilizada em análises de maior complexidade, como foi demonstrado ao longo dos capítulos 3 e 4.

De maneira geral, sua arquitetura modularizada permite que o código seja facil-

mente reutilizável, no sentido do acréscimo de novas funcionalidades e algoritmos. Isso faz com que possa ser modificada em função de novas aplicações, conferindo versatilidade.

Levando-se em conta essas características, como trabalhos futuros indicam-se duas frentes. Em primeiro lugar, a paralelização do método *MTM*. Eventualmente, o custo computacional do algoritmo pode ser elevado, quando se trabalha com matrizes de tamanho excessivo, próximo do limite da memória do sistema. A programação paralela é uma técnica recorrente em simulação computacional e com suporte em *MATLAB*®. Ela pode ser realizada tanto através de *clusters*, como em unidades de processamento gráfico (*Graphics Processing Unit, GPU*). O interesse reside na redução do tempo de cálculo, permitindo discretizações mais finas e o tratamento de problemas de maior complexidade. Também, sugere-se o estudo da validade dos modelos de conversão de modo, o que requer uma análise experimental detalhada fora do escopo deste trabalho.

Globalmente, o presente trabalho cumpriu os objetivos pretendidos, implementando uma solução que constitui uma alternativa flexível e rápida para a determinação do campo acústico para diversos tipos de problemas com diferentes graus de complexidade. A ferramenta gráfica que integra os algoritmos desenvolvidos permite a abstração dos problemas de maneira relativamente simples, tornando as análises dinâmicas e eficazes.

## REFERÊNCIAS

- ANDRADE, M.; ADAMOWSKI, J.; IBÁÑEZ, A.; PARRILLA, M.; FRITSCH, C. Optimization of complex shape part inspections with phased arrays. In: *Ultrasonics Symposium (IUS), 2009 IEEE International*. [S.l.: s.n.], 2009. p. 733 –736. ISSN 1948-5719.
- ANDRADE, M.; PEREZ, N.; BUIOCHI, F.; ADAMOWSKI, J. Matrix method for acoustic levitation simulation. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, v. 58, n. 8, p. 1674 –1683, august 2011. ISSN 0885-3010.
- ARNOLD, F.; PELA, C. Simulação computacional de campos ultra-sônicos. *Revista Brasileira de Ensino de Física*, Scielo, v. 26, p. 223 – 231, 00 2004. ISSN 1806-1117.
- BUIOCHI, F. Dissertação (Mestrado), *Análise dos métodos de medição de densidade de líquidos por ultra-som*. Escola Politécnica, Universidade de São Paulo. 1994.
- CALLISTER, W. *Materials Science And Engineering: An Introduction*. [S.l.]: John Wiley & Sons, 2007. ISBN 9780471736967.
- CALMON, P.; MAHAUT, S.; CHATILLON, S.; RAILLON, R. Civa: An expertise platform for simulation and processing ndt data. *Ultrasonics*, v. 44, Supplement, n. 0, p. e975 – e979, 2006. ISSN 0041-624X.
- CHAPRA, S.; CANALE, R. *Numerical methods for engineers*. [S.l.]: McGraw-Hill Higher Education, 2005. ISBN 9780072918731.
- CHEEKE, J. *Fundamentals and Applications of Ultrasonic Waves*. [S.l.]: Taylor & Francis, 2002. (CRC Series in Pure and Applied Physics). ISBN 9780849301308.
- CHEN, D.; MCGOUGH, R. J. A 2d fast near-field method for calculating near-field pressures generated by apodized rectangular pistons. *Journal of the Acoustical Society of America*, v. 124, n. 5, p. 1526–1537, 2008.
- COOK, G. *Mobile Robots: Navigation, Control and Remote Sensing*. [S.l.]: Wiley, 2011. ISBN 9780470630211.
- DEGASPERE, T.; BUIOCHI, F. Simulação computacional de campos gerados por transdutores ultra-sônicos em excitação contínua. *17º Simpósio de Iniciação Científica da USP*, n. 832, nov. 2009.
- DRINKWATER, B.; WILCOX, P. Ultrasonic arrays for non-destructive evaluation: A review. *NDT & E International*, v. 39, n. 7, p. 525–541, out. 2006. ISSN 09638695.
- FISH, P. *Physics and instrumentation of diagnostic medical ultrasound*. [S.l.]: Wiley, 1990. ISBN 9780471926511.
- FRANCO, E. E.; ANDRADE, M. A. B.; HIGUTI, R. T.; ADAMOWSKI, J. C.; BUIOCHI, F. Acoustic transmission with mode conversion phenomenon. In: . [S.l.: s.n.], 2005. v. 2, n. 11, p. 113 – 120.

- GOODMAN, J. *Introduction To Fourier Optics*. [S.l.]: Roberts & Company, 2005. (McGraw-Hill physical and quantum electronics series). ISBN 9780974707723.
- HUTTUNEN, T.; MALINEN, M.; KAIPIO, J.; WHITE, P.; HYNYNEN, K. A full-wave helmholtz model for continuous-wave ultrasound transmission. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, v. 52, n. 3, p. 397 –409, march 2005. ISSN 0885-3010.
- IBÁÑEZ, A.; FRITSCH, C.; PARRILLA, M.; VILLAZÓN, J. Monochromatic transfer matrix method for acoustic field simulation thorough media boundaries. *Physics Procedia*, v. 3, n. 1, p. 883 – 890, 2010. ISSN 1875-3892. International Congress on Ultrasonics, Santiago de Chile, January 2009.
- IBÁÑEZ, M. P. A.; VILLAZÓN, J. Simulation of continuous wave fields in homogeneous media with soft curved interfaces. *ENDT, Berlin*, Th. 3.3.1, 2006.
- KASS, D.; NELLIGAN, T.; HENJES, E. The Evolution and Benefits of Phased Array Technology for the Every Day Inspector. *ECNDT, Berlin*, p. 198, 2006.
- KINO, G. *Acoustic waves: devices, imaging, and analog signal processing*. [S.l.]: Prentice-Hall, 1987. (Prentice-Hall signal processing series). ISBN 9780130030474.
- KINSLER, L. *Fundamentals of acoustics*. [S.l.]: Wiley, 2000. ISBN 9780471847892.
- KUTTRUFF, H. *Ultrasonics fundamentals and applications*. [S.l.]: Elsevier Applied Science, 1991. (Black women in United States history). ISBN 9781851665532.
- LAGARIAS, J. C.; REEDS, J. A.; WRIGHT, M. H.; WRIGHT, P. E. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, v. 9, p. 112–147, 1998.
- OLINER, A. Microwave network methods for guided elastic waves. *Microwave Theory and Techniques, IEEE Transactions on*, v. 17, n. 11, p. 812 – 826, nov 1969. ISSN 0018-9480.
- PARRILLA, M.; BRIZUELA, J.; CAMACHO, J.; IBÁÑEZ, A.; NEVADO, P.; FRITSCH, C. Dynamic focusing through arbitrary geometry interfaces. In: *Ultrasonics Symposium, 2008. IUS 2008. IEEE*. [S.l.: s.n.], 2008. p. 1195 –1198.
- PIWAKOWSKI, B.; SBAI, K. A new approach to calculate the field radiated from arbitrarily structured transducer arrays. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, v. 46, n. 2, p. 422–440, march 1999. ISSN 0885-3010.
- ROBINSON, D.; LEES, S.; BESS, L. Near field transient radiation patterns for circular pistons. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, v. 22, n. 6, p. 395 – 403, dec 1974. ISSN 0096-3518.
- SERWAY, R.; JEWETT, J. *Physics for Scientists and Engineers*. [S.l.]: Cengage Learning, 2009. ISBN 9781439048382.
- ZIMMERMAN, D. Paul Langevin and the discovery of active Sonar or Asdic. *The Northern Mariner*, v. 12, n. 1, 2008.

# ANEXO A – FERRAMENTA COMPUTACIONAL

Neste anexo, a ferramenta computacional para simulação do campo acústico em excitação contínua é apresentada em detalhes. Primeiramente, apresenta-se a arquitetura utilizada para desenvolvimento do *software*. Em seguida, apresenta-se um tutorial de utilização com exemplos práticos.

## A.1 Documentação

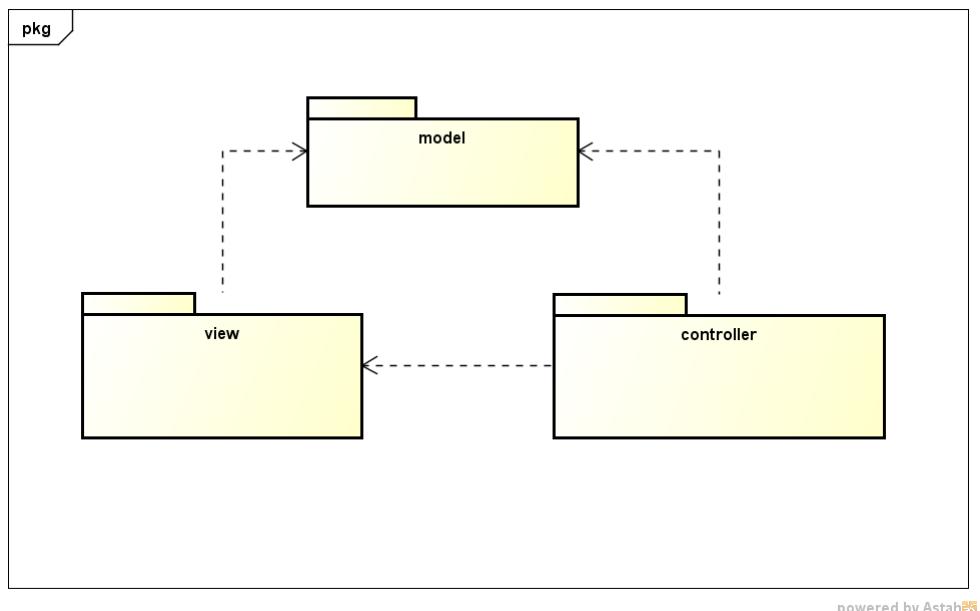
A implementação da ferramenta para simulação do campo acústico com o método *MTM* segue o *design pattern* denominado *Model-View-Controller* (MVC). O objetivo é não somente modularizar o código, como também prover classes especializadas de acordo com sua função no programa.

A figura A.1 mostra um diagrama de estrutura. Nele, notam-se três partes básicas. A parte *Model* comprehende classes relativas ao modelo do sistema, correspondente ao meio de propagação e transdutor. A parte *View* é responsável pela implementação de interfaces gráficas, através das quais o usuário realizará as modificações necessárias no modelo. A parte *Controller* arbitra determinadas ações relacionadas às outras duas partes e também é responsável pelo comportamento da parte *View*.

Todas as classes são descendentes da classe *handle*. Em *MATLAB*®, isso permite que as mesmas sejam alocadas dinamicamente, permitindo uma abordagem de programação orientada a objetos.

### A.1.1 Modelo

A figura A.2 mostra a estrutura do modelo associado ao sistema. Ele é composto por uma classe abstrata *transducer*, que estabelece as características básicas de um transdutor do ponto de vista do programa, independentemente do seu tipo. A figura A.3 mostra a estrutura interna dessa classe, com seus atributos, métodos associados e as classes especializadas que efetivamente implementam a classe abstrata. Da mesma forma, definem-se as classes relativas ao meio de propagação e à interface de separação entre dois meios, conforme figura A.4.



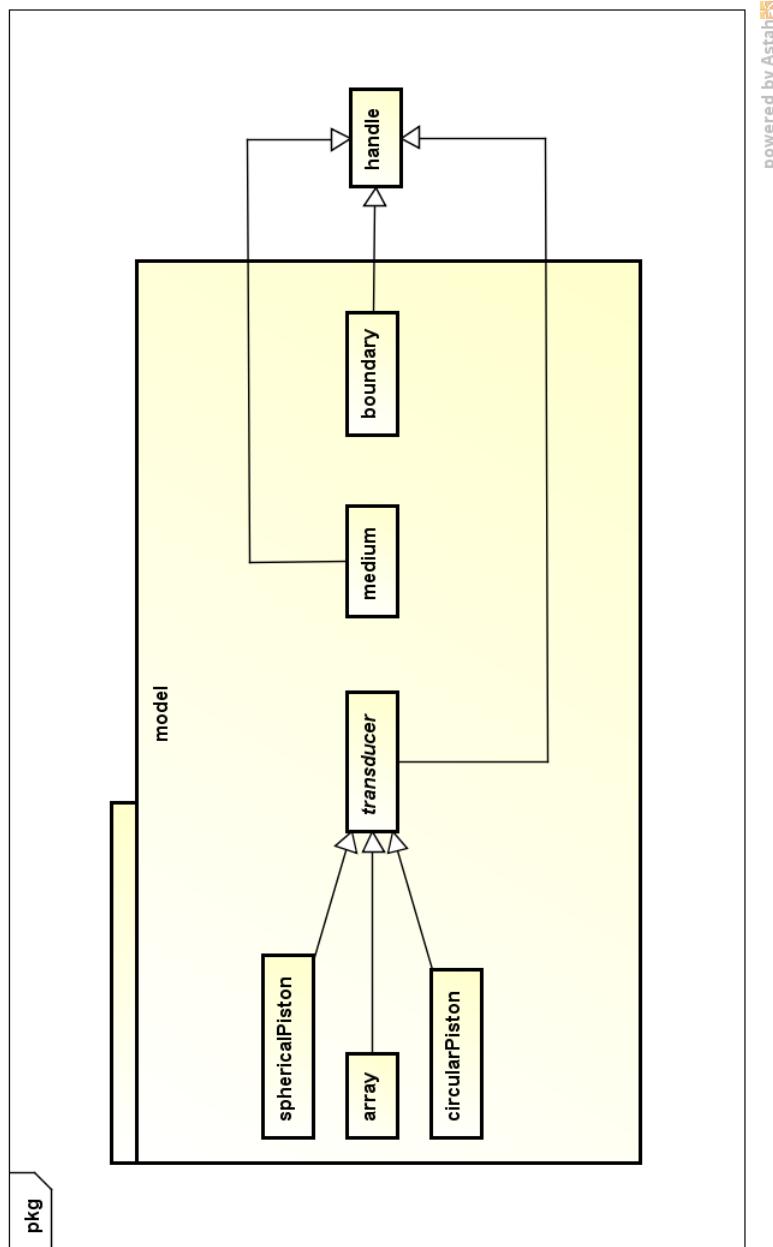
**Figura A.1:** Diagrama de estruturas MVC

### A.1.2 Interface gráfica

A figura A.5 mostra a estrutura da interface gráfica. A classe *main\_gui* implementa o ambiente de configuração principal e é composto por 8 interfaces auxiliares. Cada uma consiste em uma especialização da classe abstrata *configGui*, que descreve a estrutura geral de um ambiente de configuração. Em geral, essas classes são responsáveis por lidar com a configuração dos meios de propagação e transdutores. Os atributos e métodos de cada classe são explicitadas em A.6 e A.7.

## *Interface gráfica principal*

A classe principal do programa, que agrupa modelo, controlador e demais interfaces é a classe *main\_gui*. A figura A.8 mostra seu conteúdo, explicitando métodos e atributos. A figura A.9 mostra um diagrama de componentes simplificado, de modo a exemplificar sua arquitetura. De maneira geral, ela é composta pelo modelo do problema (transdutor, meios de propagação, interface) e por interfaces de configuração em um nível hierárquico abaixo da interface gráfica principal.



**Figura A.2:** Estrutura interna de *Model*

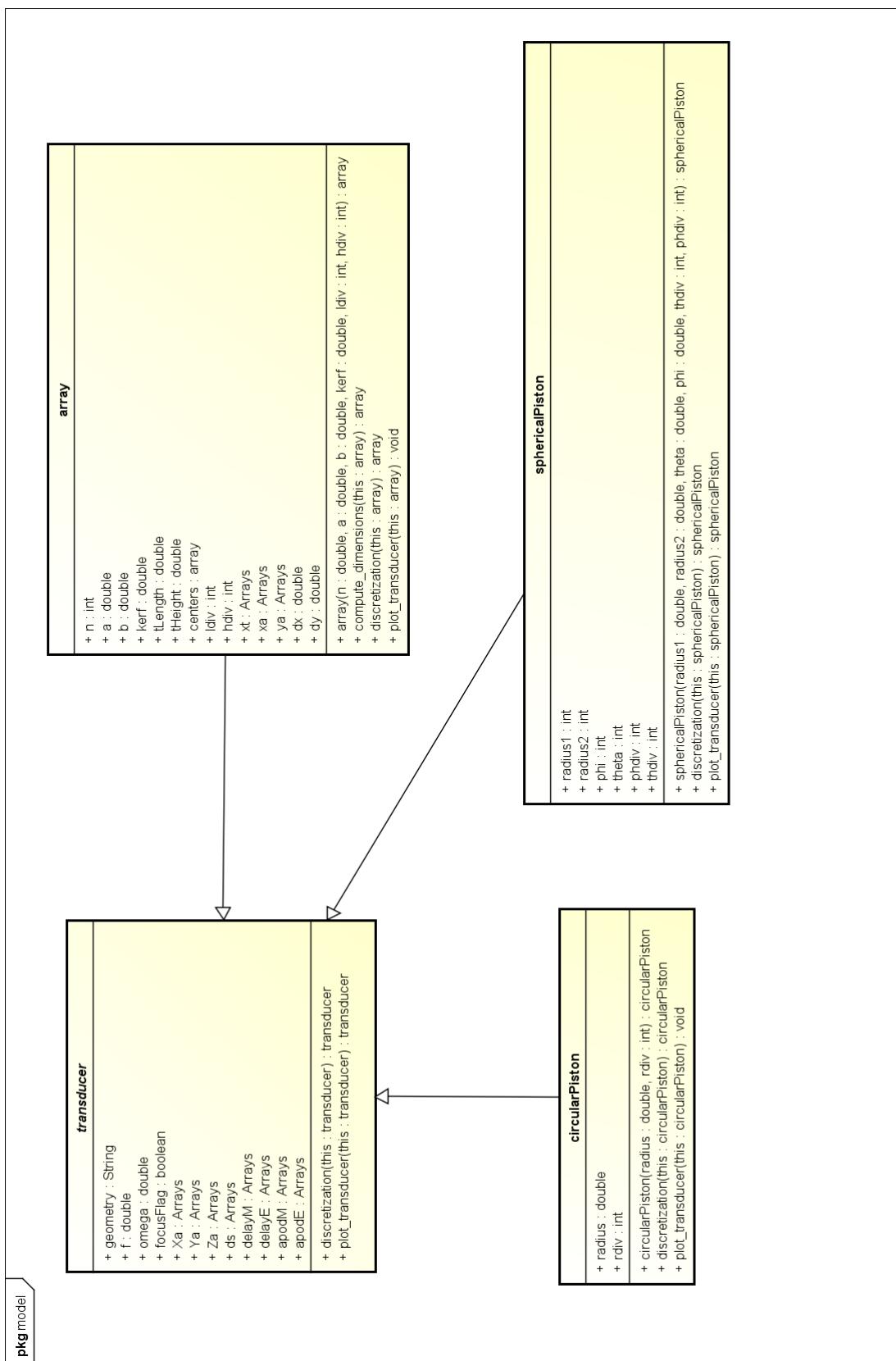
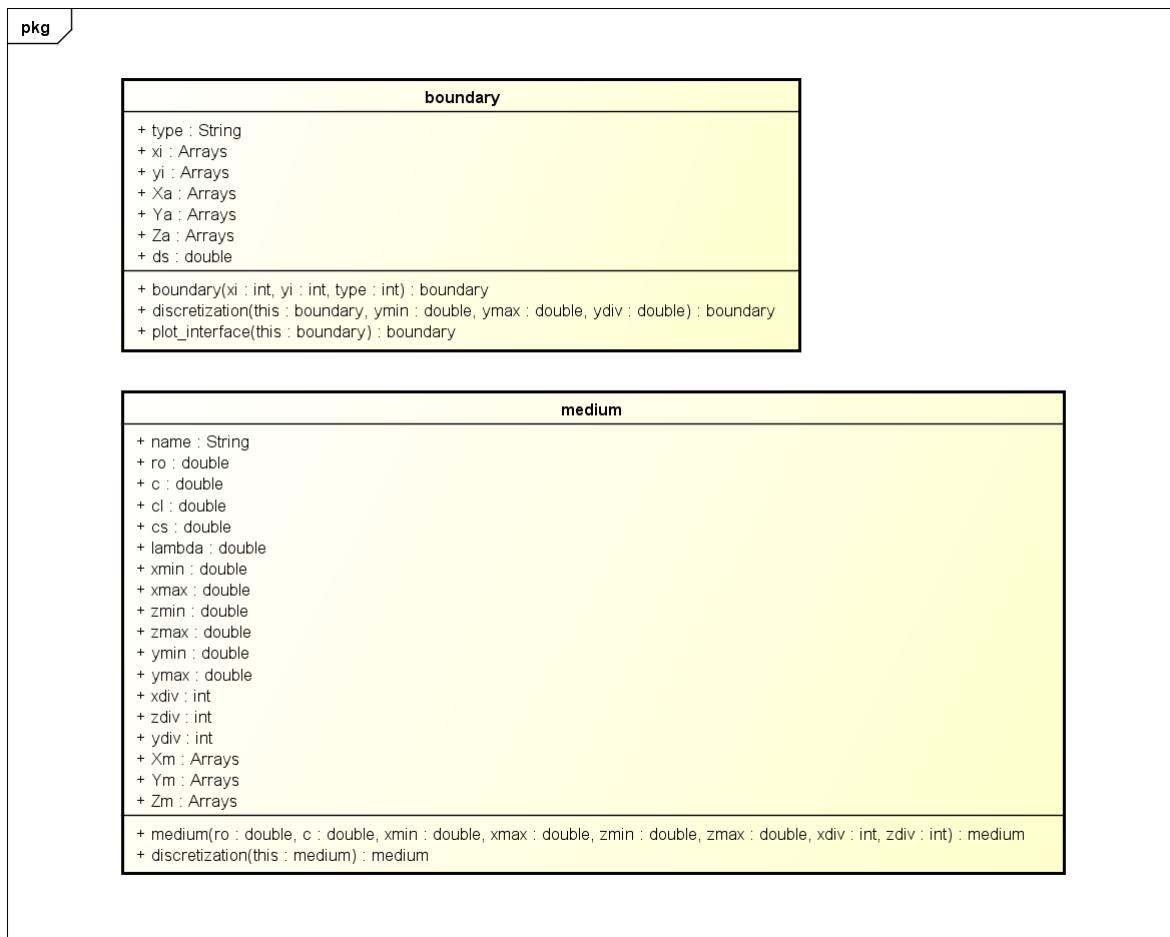
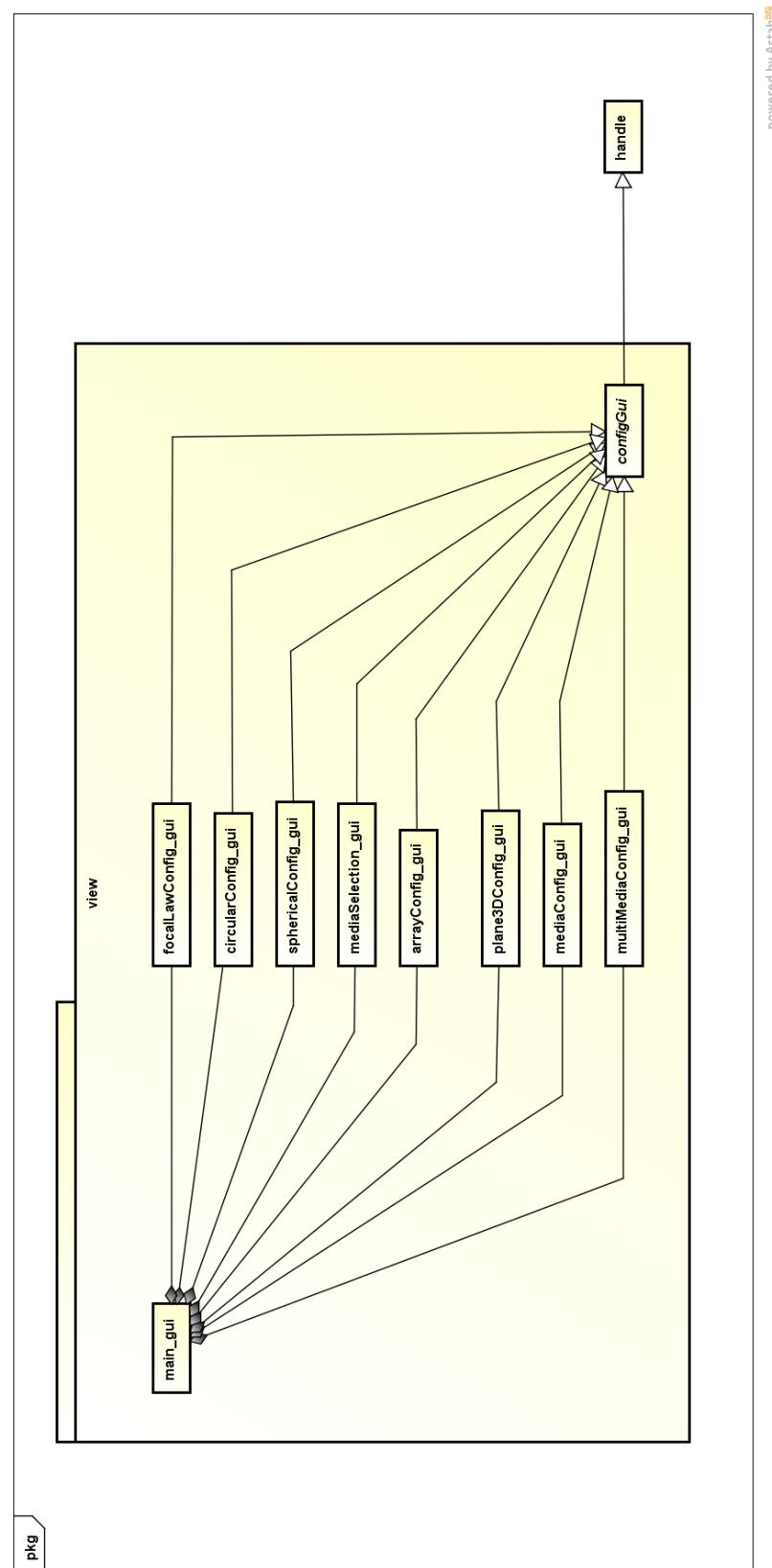


Figura A.3: Classes associadas ao modelo do transdutor



**Figura A.4:** Classes associadas ao modelo do meio de propagação



**Figura A.5:** Classes associadas à interface gráfica

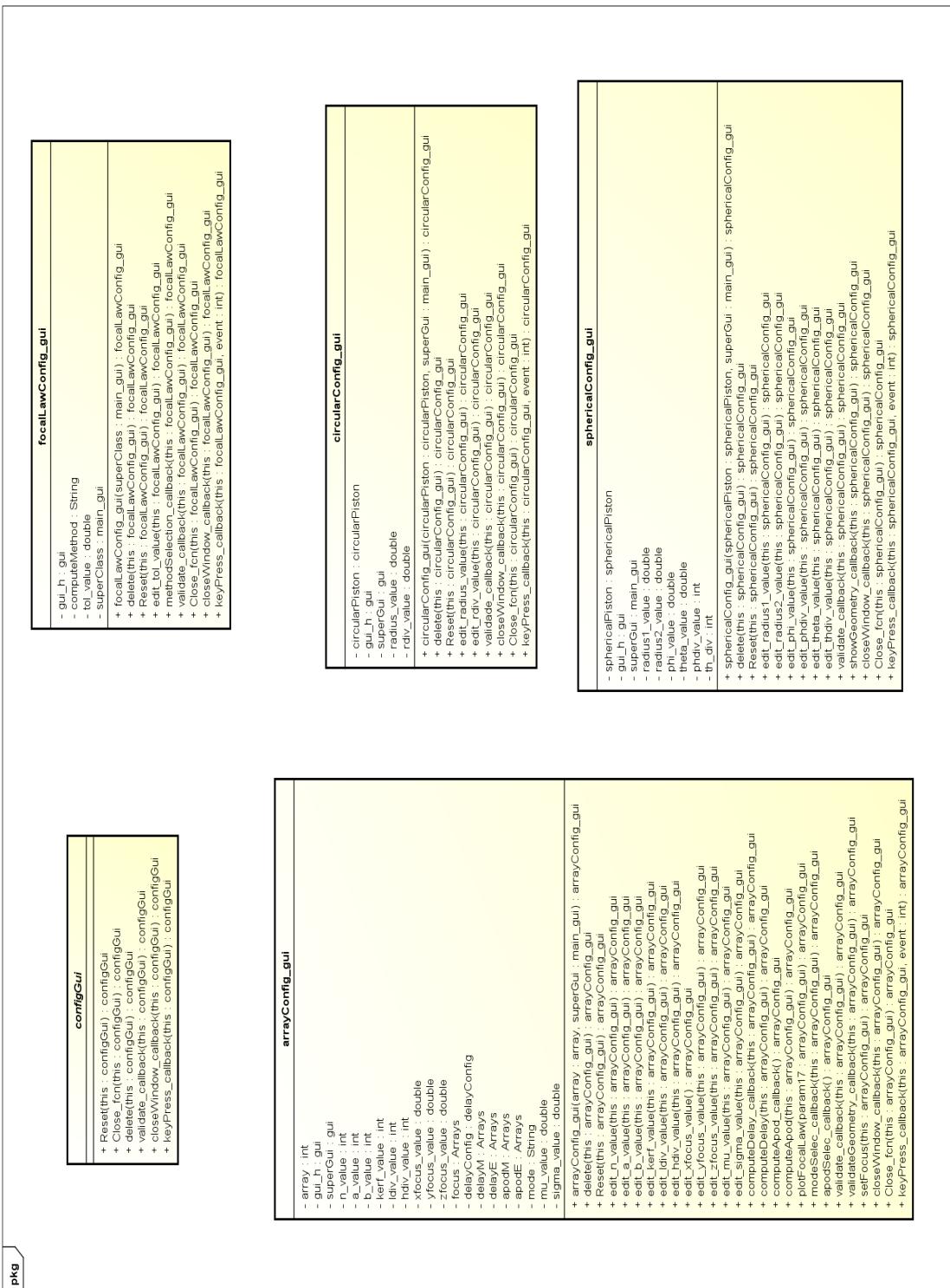
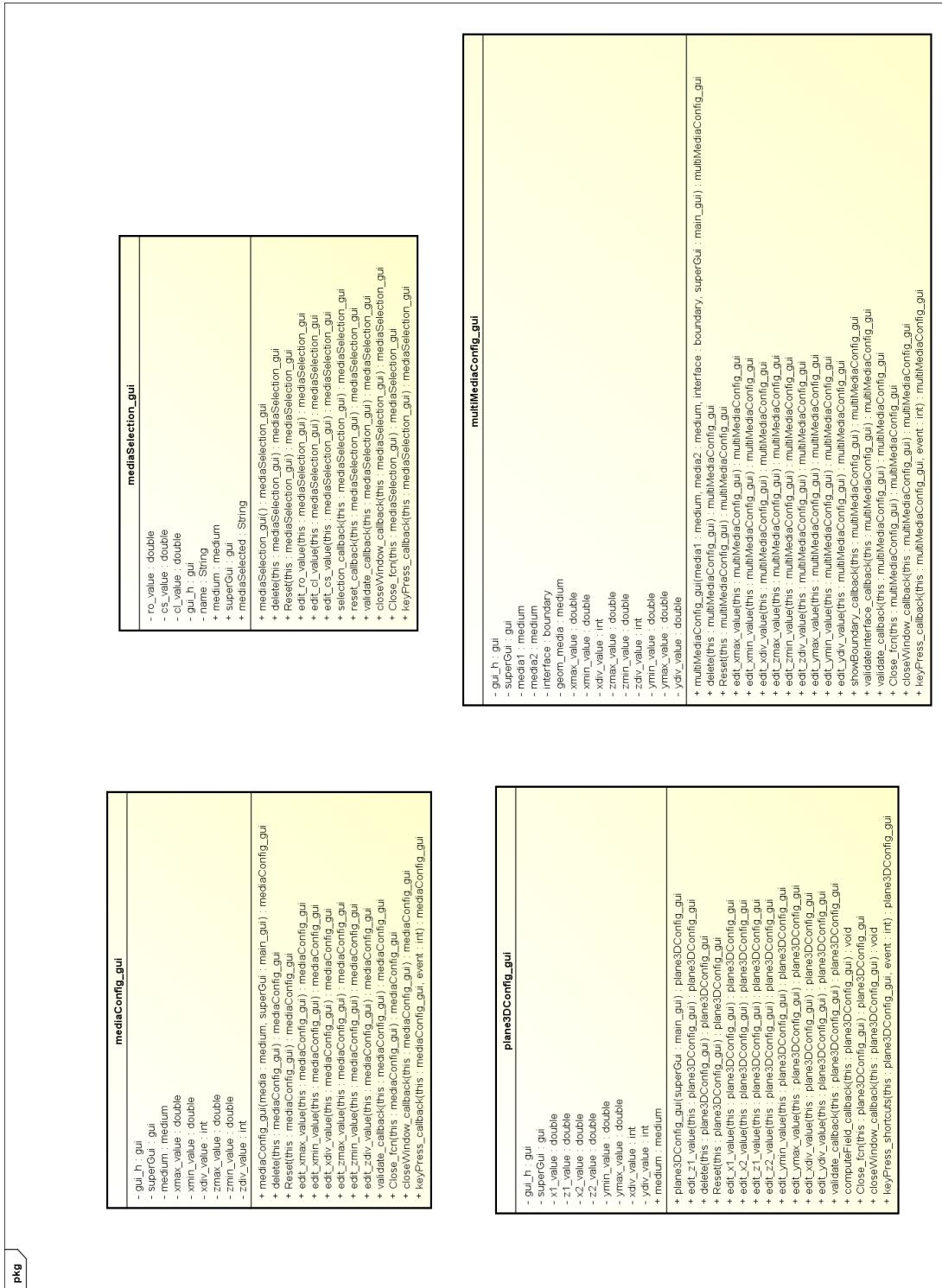


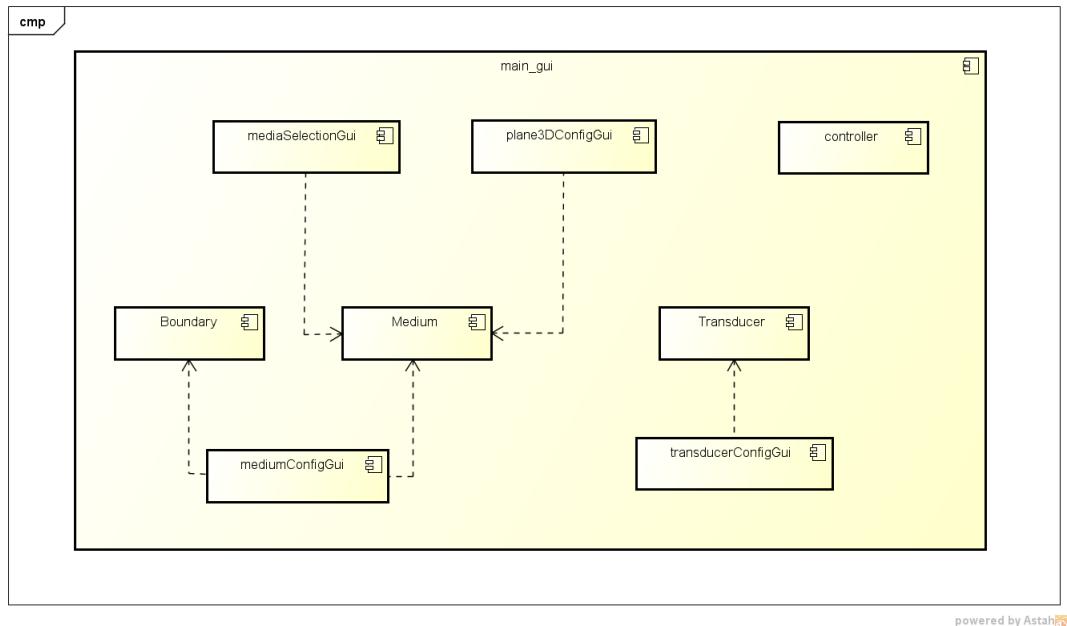
Figura A.6: Classes associadas à interface gráfica, atributos e métodos, parte 1



**Figura A.7:** Classes associadas à interface gráfica, atributos e métodos, parte 2



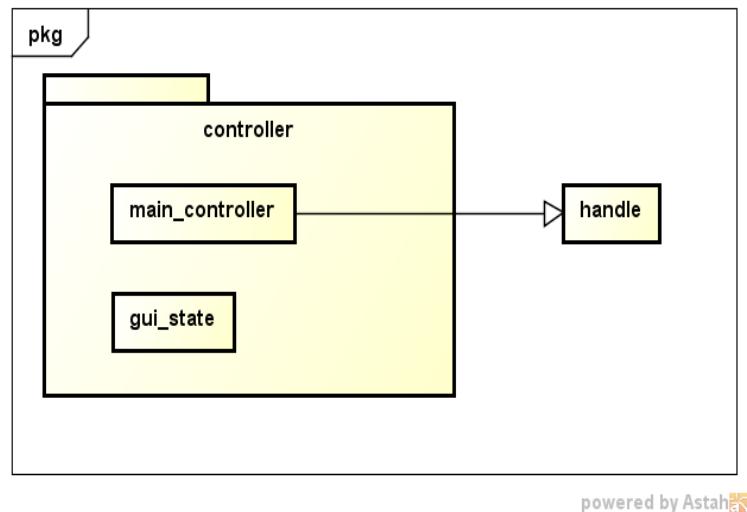
Figura A.8: Interface gráfica principal, métodos e atributos



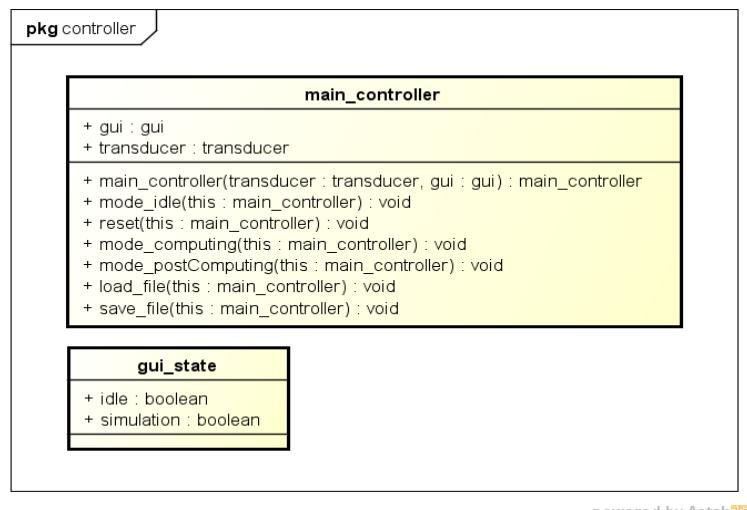
**Figura A.9:** Interface gráfica principal, diagrama de componentes

### A.1.3 Controlador

A parte *Controller* é formada somente por duas classes. Basicamente, o controlador é responsável pela mudança de estados da interface gráfica, alternando entre um estado de espera, em que o usuário realiza a etapa de modelagem, e o estado de simulação, no qual a etapa de cálculos é realizada. A figura A.10 mostra a estrutura e as classes associadas ao controlador. A figura A.11 mostra o diagrama de estados que controla o comportamento da interface gráfica principal. Outras ações associadas ao controlador são carregar e salvar arquivos.

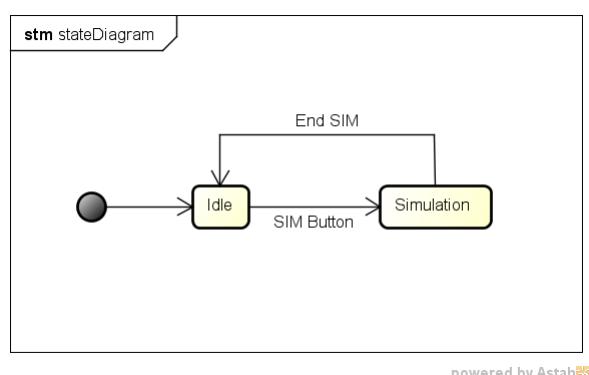


(a)



(b)

**Figura A.10:** Classes associadas ao controlador: (a) Estrutura do controlador, (b) Atributos e métodos das classes associadas ao controlador



**Figura A.11:** Diagrama de estados associado ao controlador

## A.2 Listagem de rotinas auxiliares

Um certo número de funções auxiliares foi desenvolvido para implementar os métodos numéricos descritos neste trabalho. Eles referem-se à implementação do método *MTM*, cálculo das leis de atraso, apodização, determinação das interfaces curvilíneas e *splines*, plotagem de resultados, etc. Entre todos os códigos desenvolvidos, os mais importantes são listados na sequência.

### A.2.1 MTM meio simples

```

1 function [ fieldP ,Rl ,Int ] = acousticFieldMultiMedia( transducer ,
2   boundary ,medium1 ,medium2 ,mode ,convMode )
3   %tstart = tic ;
4   k1 = transducer.omega/medium1.c ;
5   k2 = transducer.omega/medium2.c ;
6
7   omega = transducer.omega ;
8
9   dim = length( boundary.Xa ) ;
10
11  c1 = medium1.c ;
12  c2 = medium2.c ;
13
14  ro1 = medium1.ro ;
15  ro2 = medium2.ro ;
16
17  x1 = meshgrid( boundary.Xa , transducer.Xa ) ;
18  z1 = meshgrid( boundary.Za , transducer.Xa ) ;
19  y1 = meshgrid( boundary.Ya , transducer.Xa ) ;
20
21  Xa = meshgrid( transducer.Xa , ones( dim ,1 ) ) ;
22  Xa = Xa' ;
23  Ya = meshgrid( transducer.Ya , ones( dim ,1 ) ) ;
24  Ya = Ya' ;
25
26  delay_matrice = transducer.delayM ;

```

```

    apodization_matrice = transducer.apodM;

27
    if strcmp(transducer.geometry, 'spherical')
        Za = meshgrid(transducer.Za, ones(dim,1));
        Za = Za';
        Rl1 = sqrt((Xa-x1).^2 + (Ya-y1).^2 + (Za-z1).^2);
        ds = meshgrid(transducer.ds, ones(dim,1));
31
    else
        Rl1 = sqrt((Xa-x1).^2 + (Ya-y1).^2 + z1.^2);
        ds = transducer.ds;
    end
37
    dim2 = size(medium2.Xm,1);
39
    xm2 = meshgrid(medium2.Xm, boundary.Xa);
41    zm2 = meshgrid(medium2.Zm, boundary.Za);
    ym2 = meshgrid(medium2.Ym, boundary.Ya);
43
    Xint = meshgrid(boundary.Xa, ones(dim2,1));
45    Xint = Xint';
    Zint = meshgrid(boundary.Za, ones(dim2,1));
47    Zint = Zint';
    Yint = meshgrid(boundary.Ya, ones(dim2,1));
49    Yint = Yint';
    Rl2 = sqrt((Xint-xm2).^2 + (Zint-zm2).^2 + (Yint-ym2).^2);
51
    Rl2(Rl2<medium2.lambda) = inf;
53
    ds_b = meshgrid(boundary.ds, ones(dim2,1));
55
    if strcmp(convMode, 'conversion')
57
        if strcmp(transducer.geometry, 'spherical')
            theta_inc = boundary_incidence(transducer,
                boundary, x1, z1, y1, Xa, Ya, Za);

```

```

    else
        theta_inc = boundary_incidence(transducer,
                                         boundary, x1, z1, y1, Xa, Ya);
    end

63
    T = trans_coeff(mode, theta_inc, medium1, medium2);

65    else
        T = 1;
    end

67

69    if transducer.focusFlag
        %TO DO%
        71        Int = ((exp(-1j*(k2*Rl2')) ./ Rl2' .* ds_b) * ((exp(-1j*(k1*Rl1')) )
                    ./ Rl1' .* ds) .* T') .* (exp(-1j*delay_matrice) .* apodization_matrice) / (4*pi*pi);
        fieldP = abs(Int);

73    else
        %TO DO%
        75        Int = ((exp(-1j*(k2*Rl2')) ./ Rl2' .* ds_b) * (((exp(-1j*(k1*Rl1')) )
                    ./ Rl1' .* ds) .* T') ./ 4*pi*pi));
        fieldP = abs(sum(Int, 2));
    end

77    %telapsed = toc(tstart);
    79    %msgbox(['Tempo de simulação: ' num2str(telapsed)]);
end

```

### A.2.2 MTM meio composto

```

function [fieldP, Rl, Int] = acousticFieldSimpleMedia(transducer, medium
)
2    %dim = (medium.xdiv+1)*(medium.zdiv+1);
3    %tstart = tic;
4    k = transducer.omega/medium.c;
    omega = transducer.omega;
6
    ro = medium.ro;

```

```

8      dim = length(medium.Xm);
9      x = meshgrid(medium.Xm, transducer.Xa);
10     z = meshgrid(medium.Zm, transducer.Xa);
11     y = meshgrid(medium.Ym, transducer.Xa);

12
13     Xa = meshgrid(transducer.Xa, ones(dim,1));
14     Xa = Xa';
15     Ya = meshgrid(transducer.Ya, ones(dim,1));
16     Ya = Ya';

18     delay_matrice = transducer.delayM;
19     apodization_matrice = transducer.apodM;

20
21     if strcmp(transducer.geometry, 'spherical')
22         Za = meshgrid(transducer.Za, ones(dim,1));
23         Za = Za';
24         Rl = sqrt((Xa-x).^2 + (Ya-y).^2 + (Za-z).^2);
25         ds = meshgrid(transducer.ds, ones(dim,1));
26     else
27         Rl = sqrt((Xa-x).^2 + (Ya-y).^2 + z.^2);
28         ds = transducer.ds;
29     end

30
31     Rl(Rl<medium.lambda) = inf;

32
33     Int =(exp(-1j*(k*Rl'))./Rl')*(exp(-1j*delay_matrice).*.
34     apodization_matrice).*ds/(2*pi);

35
36     if transducer.focusFlag
37         fieldP = abs(ro*1j*omega*Int);
38     else
39         fieldP = abs(ro*1j*omega*sum(Int,2));
40     end

41     %telapsed = toc(tstart);
42     %msgbox(['Tempo de simulação: ', num2str(telapsed)]);

```

42 end

### A.2.3 Apodização

```

function [ apodization_matrice , apodization_element ] = apodization (
array , method , param )

2
apodization = ones( length( array . centers ) , 1 ) ;

4
if strcmp( method , ' gauss ' )

6
    mu = param( 1 ) ;
8
    sigma = param( 2 ) ;
    apodization = normpdf( array . centers , mu , sigma ) ;
10
    apodization = apodization ./ max( apodization ) ;

elseif strcmp( method , ' cos ' )

12
    distance = pi*( array . centers - min( array . centers ) ) / array . tLength
        - pi / 2 ;
    apodization = cos( distance ) ;

14
elseif strcmp( method , ' sin ' )

16
    distance = pi*( array . centers - min( array . centers ) ) / array . tLength
        - pi / 2 ;
    apodization = sin( distance ) . ^ 2 ;

18
end

20

22 for i = 1 : array . n
    x = array . xt ( : , i ) ' ;
24
    y = array . ya ;
    X = meshgrid( x , y ) ;
26
    apodization_matrice ( : , i ) = meshgrid( apodization ( i ) , X ) ;
end

28
apodization_element = apodization_matrice ( 1 , : ) ' ;

```

```
30 apodization_matrice = reshape(apodization_matrice, [array.ldiv*array.
    hdiv*array.n 1]);
```

#### A.2.4 Leis de atraso, meio simples

```
function [delay_matrice, delay_element] = delay(foci, array, c)
2
    omega = array.omega;
4
    delay_matrice = [];
6
    distance = sqrt((array.centers-foci.x).^2+(foci.y).^2+(foci.z)
        .^2);
8
    delay = omega*(max(distance)-distance)/c;
10
    for i=1:array.n
        x = array.xt(:, i)';
        y = array.ya;
12
        [X, Y] = meshgrid(x, y);
        delay_matrice(:, i) = meshgrid(delay(i), X);
14
    end
16
    delay_element = delay_matrice(1, :);
    delay_matrice = reshape(delay_matrice, [array.ldiv*array.hdiv*
        array.n 1]);
18
end
```

#### A.2.5 Leis de atraso meio composto, princípio de Fermat

```
function [delay_matrice, delay_element] = delay_interface(foci,
    interface, array, c1, c2)
2
    %fermat principle
4
    omega = array.omega;
    delay_matrice = [];
6
    tmin = zeros(length(array.centers), 1);
8
    for i=1:length(array.centers)
```

```

    %Calculo das distancias do i-ésimo centro aos pontos da
    interface
10    dist_array2int = sqrt((interface.xi-array.centers(i)).^2 +
                           interface.zi).^2);

12    %Calculo das distancias dos pontos da interface ao foco no
        meio 2
13    dist_int2focus = sqrt((interface.xi-foci.x).^2+foci.y.^2 +
                           interface.zi-foci.z).^2);

14    %Acha o caminho entre 1 e 2 com o tempo mínimo de propagação
15    travel_time = dist_array2int/c1 + dist_int2focus/c2;

16    tmin(i) = min(travel_time);

17    end

18
19    %Matriz base para o calculo dos atrasos
20    delay=omega*(max(tmin) - tmin);

21
22    for i=1:array.n
23        x = array.xt(:,i)';
24        y = array.ya;
25        X = meshgrid(x,y);
26        delay_matrice(:,i) = meshgrid(delay(i),X);
27    end

28
29    delay_element = delay_matrice(1,:);
30
31    delay_matrice = reshape(delay_matrice, [array.ldiv*array.hdiv*
32                                array.n 1]);
33
34    end

```

### A.2.6 Leis de atraso meio composto, método de Newton-Raphson

```

1 function [delay_matrice,delay_element] = delay_newton(foci,interface,
array,c1,c2,tol)

```

```

3 delay_matrice = [];
5 omega = array.omega;
7 tmin = zeros(length(array.centers),1);

9 k0 = round(length(interface.xi)/2);
for i=1:length(array.centers)
    k = newton(interface.xi, array.centers(i), interface.zi, 0, foci,
    c1, c2, k0, tol);

11 %Calculo das distancias do i-ésimo centro aos pontos da
    interface
13 dist_array2int = sqrt((interface.xi(k)-array.centers(i)).^2 +
    interface.zi(k)).^2;

15 %Calculo das distancias dos pontos da interface ao foco no
    meio 2
17 dist_int2focus = sqrt((interface.xi(k)-foci.x).^2+foci.y.^2 +
    interface.zi(k)-foci.z).^2;

19 %Acha o caminho entre 1 e 2 com o tempo mínimo de propagação
21 travel_time = dist_array2int/c1 + dist_int2focus/c2;

23 tmin(i) = travel_time;

25 %Matriz base para o calculo dos atrasos
27 delay=omega*(max(tmin) - tmin);

29 for i=1:array.n
    x = array.xt(:,i)';
    y = array.ya;
    X = meshgrid(x,y);
    delay_matrice(:,i) = meshgrid(delay(i),X);

```

```

33      end

35      delay_element = delay_matrice(1,:);
delay_matrice = reshape(delay_matrice, [array.ldiv*array.hdiv*
array.n 1]);
37  end

```

### A.2.7 Método de Newton-Raphson

```

1  function k = newton(xk,xa,zk,za,foci,c1,c2,k0,tolerance)
2      err = Inf;
3      k = k0;

5      kmin = 1;
6      kmax = length(zk)-2;
7      iterMax = 10;
8      iter = 0;

9
10     while (abs(err) > tolerance && iter < iterMax)
11         kPrev = k;

13     mkPrev = (zk(kPrev+1)-zk(kPrev))/(xk(kPrev+1)-xk(kPrev));
14     mk = (zk(kPrev+2)-zk(kPrev+1))/(xk(kPrev+2)-xk(kPrev+1));
15
16     fkPrev = focal_law(xk(kPrev),xa,zk(kPrev),za,foci.x,foci.z,mkPrev,
17                         c1,c2);
18     fk = focal_law(xk(kPrev+1),xa,zk(kPrev+1),za,foci.x,foci.z,mk,c1,
19                         c2);

20     k = kPrev - round(fkPrev/(fk-fkPrev));
21
22     if k > kmax
23         k = kmax;
24     elseif k < kmin
25         k = kmin;
26
27     end

```

```

27      m = (zk(k+1)-zk(k))/(xk(k+1)-xk(k));
% stop criterion: (f(x) - 0) < tolerance

29
30      err = focal_law(xk(k),xa,zk(k),za,foci.x,foci.z,m,c1,c2);
31      % stop criterion: change of x < tolerance
32      % err = x - xPrev;
33      iter = iter+1;
end

```

### A.2.8 Função objetivo, método de Newton-Rapshon

```

function [ V ] = focal_law(xk,xa,zk,za,xf,zf,mk,c1,c2)
2
V = ( 1/c1*( (xk-xa)+mk*(zk-za) )/( sqrt((xk-xa)^2+(zk-za).^2) )...
4
        +1/c2*( (xk-xf)+mk*(zk-zf) )/( sqrt((xk-xf)^2+(zk-zf).^2) ) );
6
end

```

### A.2.9 Leis de atraso meio composto, método de Nelder-Mead

```

function [ delay_matrice , delay_element ] = delay_fmin( foci , interface ,
array , c1 , c2 )
2
omega = array.omega;
4
delay_matrice = [];
tmin = zeros( length( array.centers ) , 1 );
6
kmin = 1;
8
kmax = length( interface.xi );
k0 = round( kmax/2 );
10
% options = optimset( 'disp' , 'iter' );
options = optimset( 'Maxiter' , 10 , 'Display' , 'off' );
12
for i=1:length( array.centers )

```

```

14     sol = fminsearch(@(k) focal_law2(k, interface.xi, interface.zi,
15                     array.centers(i), 0, foci.x, foci.z, c1, c2, kmin, kmax), k0,
16                     options);

18     sol = round(sol);
19     if sol > kmax
20         sol = kmax;
21     elseif sol < kmin
22         sol = kmin;
23     end

24     % Calculo das distancias do i-ésimo centro aos pontos da
25     % interface
26     dist_array2int = sqrt((interface.xi(sol)-array.centers(i))...
27                            .^2+(interface.zi(sol)).^2);

29     % Calculo das distancias dos pontos da interface ao foco no
30     % meio 2
31     dist_int2focus = sqrt((interface.xi(sol)-foci.x).^2+foci.y...
32                           .^2+(interface.zi(sol)-foci.z).^2);

33     % Acha o caminho entre 1 e 2 com o tempo mínimo de propagação
34     travel_time = dist_array2int/c1 + dist_int2focus/c2;

35     tmin(i) = travel_time;

36     % k0 = sol;

37     end

38     % Matriz base para o calculo dos atrasos
39     delay=omega*(max(tmin) - tmin);

40     for i=1:array.n
41         x = array.xt(:, i)';
42     end

```

```

y = array.ya;
44 X = meshgrid(x,y);
delay_matrice(:,i) = meshgrid(delay(i),X);
46 end

48 delay_element = delay_matrice(1,:);
delay_matrice = reshape(delay_matrice, [array.ldiv*array.hdiv*
array.n 1]);
50 end

```

### A.2.10 Função objetivo, método de Nelder-Mead

```

% function [ Tv ] = focal_law2(k, xs, zs, xa, za, xf, zf, c1, c2)
2 function [ Tv ] = focal_law2(k, xs, zs, xa, za, xf, zf, c1, c2, kmin, kmax)
k;
4 k = abs(round(k));

6 if k > kmax
    k = kmax;
8 elseif k < kmin
    k = kmin;
10 end

12 Tv = sqrt((xs(k)-xa).^2 + (zs(k)-za).^2)/c1 ...
    + sqrt((xs(k)-xf).^2 + (zs(k)-zf).^2)/c2;
14
end

```

### A.2.11 Ângulos de incidência

```

1 function theta = boundary_incidence(transducer, boundary, x1, z1, y1, Xa,
Ya, Za)

3 % t -> tangent vector (xz plane) approximation
% at each point of the boundary
5
u = boundary.xi(3:end)-boundary.xi(1:end-2);

```

```

7  v = boundary.zi(3:end)-boundary.zi(1:end-2);

9  u = [u(1);u;u(end)];      % approximation for extreme points
10 v = [v(1);v;v(end)];
11 h = zeros(length(u),1);  % component in the y direction

13 t = [u v h];

15 % normal vector for the xz plane
16 k = zeros(length(u),3);
17 k(:,3) = 1;

19 % n -> normal vector to the points of boundary
20 n = cross(k,t,2);
21 nx = n(:,1);
22 nz = n(:,2);
23 ny = n(:,3);

25 % normalization
26 normN = sqrt(nx.^2+ny.^2+nz.^2);
27 nx = nx./normN;
28 nz = nz./normN;
29 ny = ny./normN;

31 % correction of dimensions
32 dim1 = length(boundary.xi)*length(boundary.yi);
33 Nx = meshgrid(nx,boundary.yi);
34 Nz = meshgrid(nz,boundary.yi);
35 Ny = meshgrid(ny,boundary.yi);
36 Nx = reshape(Nx,[dim1 1]);
37 Nz = reshape(Nz,[dim1 1]);
38 Ny = reshape(Ny,[dim1 1]);

39 dim2 = length(Xa);
40 nx = meshgrid(Nx,transducer.Xa);

```

```

41 nz = meshgrid(Nz, transducer.Xa);
43 ny = meshgrid(Ny, transducer.Xa);

45 % incident vector and normalization
46 xinc = (x1-Xa);
47 yinc = (y1-Ya);

49 if strcmp(transducer.geometry, 'spherical')
50     zinc = (z1-Za);
51 else
52     zinc = z1;
53 end

55 normInc = sqrt(xinc.^2+zinc.^2+yinc.^2);
56 xinc = xinc./normInc;
57 yinc = yinc./normInc;
58 zinc = zinc./normInc;

59
60 % determination of incident angle
61 dotP = xinc.*nx+zinc.*nz+yinc.*ny;
62 cth = dotP;
63 theta = acos(cth);
64 theta = abs(theta);
65 theta(theta>pi/2) = pi/2;

67 end

```

### A.2.12 Coeficientes de transmissão

```

1 function [pTp, pTs, ti] = solid_propagation(medium1, medium2)

3 cp1 = medium1.cl/1e3;
4 cs1 = medium1.cs/1e3;
5 cp2 = medium2.cl/1e3;
6 cs2 = medium2.cs/1e3;
7 d1 = medium1.ro*1e9;

```

```

d2 = medium2.ro*1e9;

9
if cs1 < 1e-5
11    cs1 = 1e-10;
end

13
ti=0:(pi/2)/3500:pi/2; %discretizacion del angulo de incidencia (rad)
15 pRp=[]; pRs=[]; pTp=[]; pTs=[]; Rp=[]; Rs=[]; Tp=[]; Ts=[];

17 for m=1:length(ti)

19 ts1=conj(asin(cs1*sin(ti(m))/cp1)); % angulo refletido p/ shear wave.
tp2=conj(asin(cp2*sin(ti(m))/cp1)); % angulo refratado p/ onda longit
.

21 ts2=conj(asin(cs2*sin(ti(m))/cp1)); % angulo refratado p/ shear wave.
%Os1=[Os1 ts1];
23 %Op2=[Op2 tp2];
%Os2=[Os2 ts2];
25 Zp1=cp1*d1/cos(ti(m));
Zs1=cs1*d1/cos(ts1);
27 Zp2=cp2*d2/cos(tp2);
Zs2=cs2*d2/cos(ts2);
29 %w=1; % ya fue admitido frecuencia w=1 en las ecuaciones abajo,
% pues la solucion independe de la frecuencia.
31 kp1=1/cp1; %kp1=w/cp1;
kp1x=kp1*sin(ti(m));
33 kp1z=kp1*cos(ti(m));
ks1=1/cs1; %ks1=w/cs1;
35 ks1x=ks1*sin(ts1);
ks1z=ks1*cos(ts1);
37 kp2=1/cp2; %kp2=w/cp2;
kp2x=kp2*sin(tp2);
39 kp2z=kp2*cos(tp2);
ks2=1/cs2; %ks2=w/cs2;
41 ks2x=ks2*sin(ts2);

```

```

ks2z=ks2*cos( ts2 );

43 A=[kp1x/(kp1z*Zp1)           1/Zs1           -kp2x/(kp2z*Zp2)           1/
      Zs2;
      1/Zp1           -ks1x/(ks1z*Zs1)           1/Zp2
      ks2x/(ks2z*Zs2);
      -(ks1z^2-ks1x^2)/ks1^2   2*ks1x*ks1z/ks1^2   (ks2z^2-ks2x^2)/ks2^2
      2*ks2x*ks2z/ks2^2;
      2*kp1x*kp1z/ks1^2   -(ks1x^2-ks1z^2)/ks1^2   2*kp2x*kp2z/ks2^2   (
      ks2x^2-ks2z^2)/ks2^2];
      47 B = [-kp1x/(kp1z*Zp1); 1/Zp1; (ks1z^2-ks1x^2)/ks1^2; 2*kp1x*kp1z/
      ks1^2];
      X = A\B;
      49 % Calcula os coeficientes de amplitude
      Rp=[Rp X(1)]; % coef. de reflexao da onda p
      51 Rs=[Rs X(2)]; % Coef. de reflexao da onda shear
      Tp=[Tp X(3)]; % coef. de transmissao (onda p)
      53 Ts=[Ts X(4)]; % coef. de transmissao (shear wave)

      55 % Calculo dos coeficientes de reflexao de potencia (pRp y pRs) e
      % transmissoes de potencia (pTp e pTs)
      57 pRp = [pRp X(1)*conj(X(1))]; %coef. de potencia
      pRs = [pRs X(2)*conj(X(2))*real(conj(Zs1)^(-1))/real(conj(Zp1)^(-1))
      ]; %coef. de potencia
      59 pTp = [pTp X(3)*conj(X(3))*real(conj(Zp2)^(-1))/real(conj(Zp1)^(-1))
      ]; %coef. de potencia
      pTs = [pTs X(4)*conj(X(4))*real(conj(Zs2)^(-1))/real(conj(Zp1)^(-1))
      ]; %coef. de potencia

      61
      end
      63
      % tgrau=ti*180/pi;
      65 % figure(2)
      % plot(tgrau, pTp, tgrau, pTs)
      67 % title('coef de intensidade de transmissao (pTp e pTs)')
      % xlabel('ângulo de incidência (graus)')

```

69 % legend( 'pTp' , 'pTs' )

71 pTp = sqrt(pTp);

72 pTs = sqrt(pTs);

73

end

### A.2.13 Conversão de modo

function T = trans\_coeff(mode, theta, medium1, medium2)

2

[ Il , Is , th\_i ] = solid\_propagation (medium1,medium2);

4

switch mode

case 'longitudinal'

6

T = interp1 (th\_i , Il , theta);

case 'shear'

8

T = interp1 (th\_i , Is , theta);

otherwise

10

end

end

### A.2.14 Interface curvilínea

1 function [ xx , zz , yy ] = circ\_definition ( ymin , ymax , ydiv )

3

options . Resize = 'on' ;

options . WindowStyle = 'normal' ;

5

options . Interpreter = 'tex' ;

name = 'Definição de uma circuferência' ;

7

prompt = { 'Discretização : ' } ;

numlines = 1 ;

9

defaultanswer = { '0' } ;

answer = str2double ( cell2mat ( inputdlg ( prompt , name , numlines ,

defaultanswer , options ) ) ) ;

11

ndiv = answer ;

n = 3 ;

13

points = zeros ( n , 2 ) ;

for i = 1 : n

```

15      switch i
16          case 3
17              name=[ 'Centro : '];
18          otherwise
19              name=[ 'Ponto ' num2str(i) ': '];
20          end
21          prompt={ 'x ...
22              'z '};
23          numlines=1;
24          defaultanswer={ '0', '0' };
25          point = inputdlg(prompt, name, numlines, defaultanswer, options)
26              ;
27          points(i,:)= [ str2double( cell2mat( point(1))) str2double(
28              cell2mat( point(2)))];
29
30      end
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

```

### A.2.15 Interface spline

```
1 function [xx,zz,yy]=spline_definition(ymin,ymax,ydiv)
2     options.Resize='on';
3     options.WindowStyle='normal';
4     options.Interpreter='tex';
5
6     name='Definição de uma spline';
7     prompt={'Número de pontos:'};
8     numlines=1;
9     defaultanswer={'0'};
10    answer = str2double(cell2mat(inputdlg(prompt,name,numlines,
11        defaultanswer,options)));
12    n = answer;
13
14    prompt={'Discretização:'};
15    numlines=1;
16    defaultanswer={'0'};
17    answer = str2double(cell2mat(inputdlg(prompt,name,numlines,
18        defaultanswer,options)));
19    ndiv = answer;
20    points = zeros(n,2);
21
22    for i=1:n
23        name=['Ponto ' num2str(i) ':'];
24        prompt={'x',...
25            'z'};
26        numlines=1;
27        defaultanswer={'0','0'};
28        point = inputdlg(prompt,name,numlines,defaultanswer,options)
29        ;
30        points(i,:)= [str2double(cell2mat(point(1))) str2double(
31            cell2mat(point(2)))];
```

```

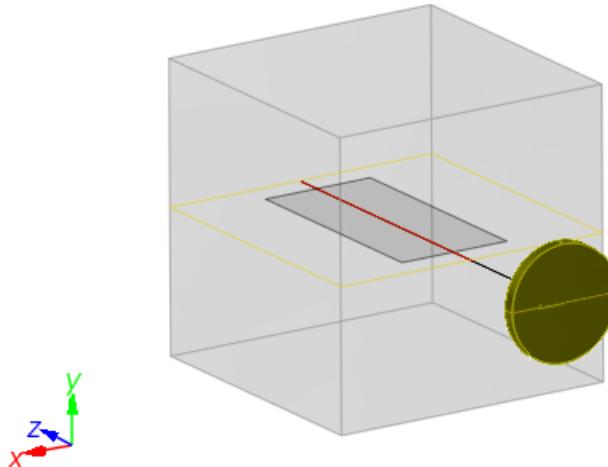
X = points (:,1)';
33 Z = points (:,2)';
boundary = spline (X,Z);

35
xx = linspace (min(X),max(X),ndiv+1)';
37 zz = ppval (boundary,xx);
yy = linspace (ymin,ymax,ydiv+1)';
39
end

```

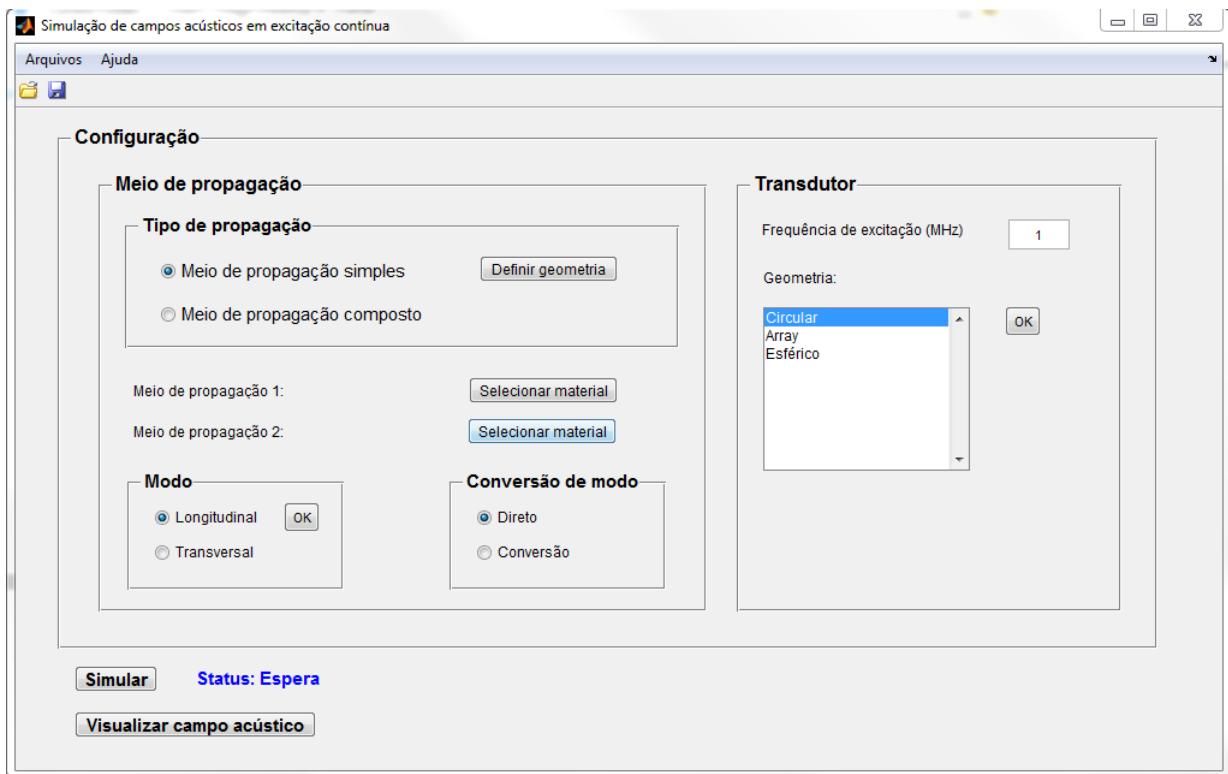
### A.3 Tutorial

Este tutorial pretende abordar os aspectos de utilização da ferramenta, tendo em vista exemplos práticos. Seja o exemplo da figura A.12. O problema consiste em determinar o campo acústico em uma região no interior do sólido, um cubo maciço de aço. O transdutor está imerso em água. A figura A.13 mostra o aspecto geral do ambiente de desenvolvimento.



**Figura A.12:** Exemplo de aplicação

Inicialmente o campo acústico será determinado na região que antecede a fronteira, ilustrando o caso de propagação em um meio simples. Então, ele será traçado no interior do sólido e outros aspectos da ferramenta serão explorados. As etapas para tratamento do problema são duas: modelagem e cálculo. Primeiramente, modela-se o meio de propagação e em seguida o transdutor. Então, o cálculo pode ser realizado. Para iniciar o programa



**Figura A.13:** Ambiente de desenvolvimento

deve-se executar o *script* *start\_gui*.

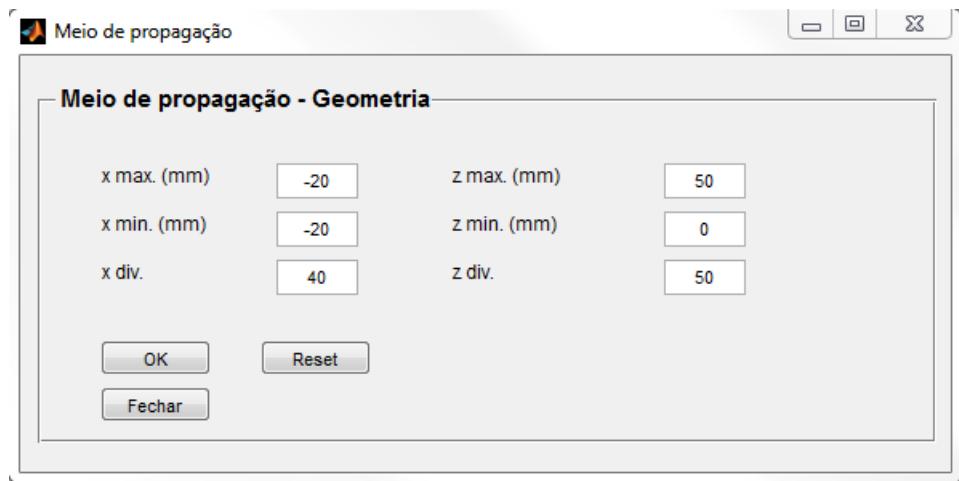
### A.3.1 Exemplo 1

#### A.3.1.1 Propagação em um meio simples

A propagação em um meio simples é definida a partir da caixa de seleção no canto superior esquerdo da interface. Em seguida o usuário deve clicar no botão *Definir geometria*. Isso habilita o ambiente para definição da região onde o campo acústico sera traçado.

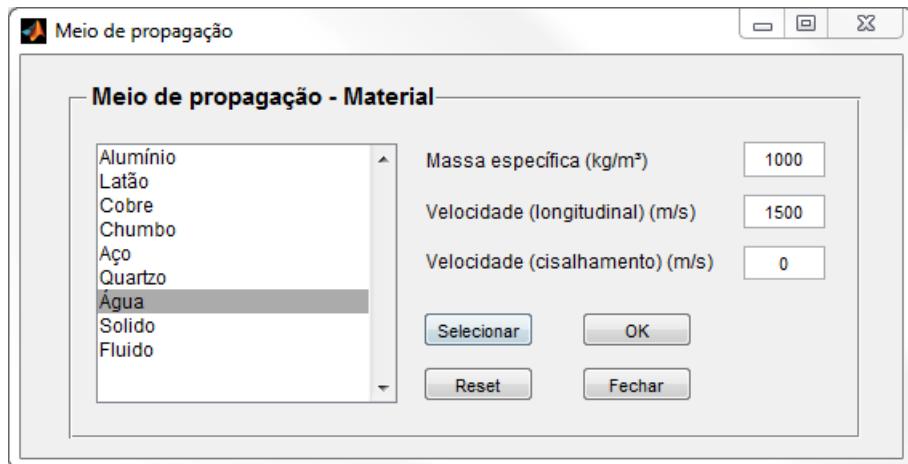
A figura A.14 ilustra as configurações que devem ser atribuídas. A região corresponde ao plano *xz* que antecede a interface fluido-sólido para o caso apresentado em A.12. De maneira geral, o usuário sempre deve clicar na opção *OK* após realizar alguma configuração.

Em seguida, escolhe-se o material correspondente. Novamente no ambiente principal, deve-se escolher a opção *Selecionar Material*. O ambiente da figura A.15 será habilitado. O usuário pode selecionar entre os meios disponíveis na lista ou configurar



**Figura A.14:** Configuração de um meio de propagação simples

meios genéricos a partir das entradas *Sólido* e *Fluido*. As configurações padrões de massa específica e velocidade também são modificáveis. Caso o usuário deseje retornar às configurações originais, o botão *Reset* deve ser selecionado. Uma vez escolhido o material, deve-se clicar em *Selecionar* e em seguida em *OK*, para validar a escolha.

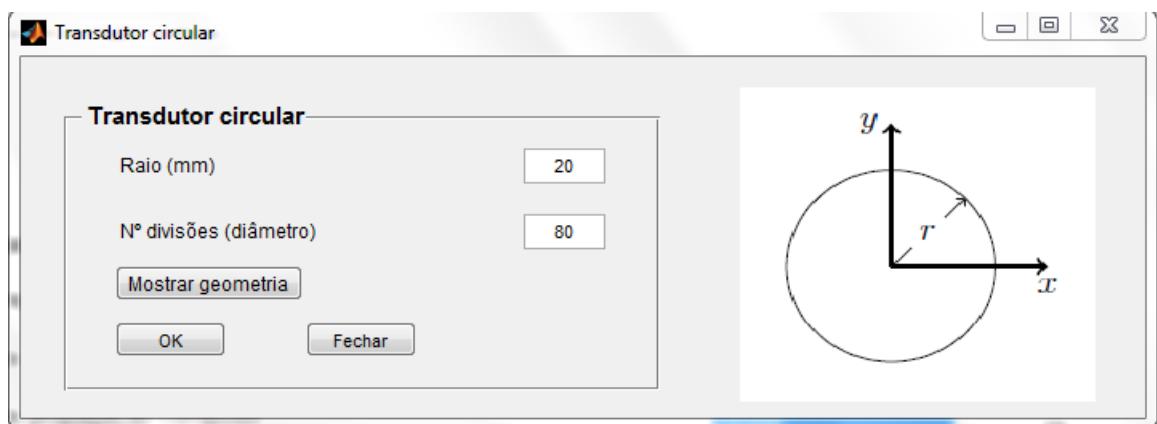


**Figura A.15:** Seleção de um meio de propagação simples

As opções *Modo* e *Conversão de Modo* configuram o modo de propagação e a aplicação do algoritmo de conversão de modo, que calcula os coeficientes de transmissão associados. Nesse caso, como não há propagação entre uma interface fluido-sólido, selecionam-se as opções *Longitudinal* e *Direto*. Assim, o meio de propagação está completamente configurado.

### A.3.1.2 Configuração do transdutor circular plano

Neste primeiro exemplo, será configurado um transdutor circular plano. A frequência de excitação atribuída é de  $2MHz$ , na caixa de edição no canto superior direito. Em seguida, seleciona-se a geometria *Circular* e clica-se no botão *OK*. O ambiente da figura A.16 é habilitado. O transdutor é configurado com raio  $20mm$  e 80 divisões na direção radial, ao longo do diâmetro do transdutor. O botão *Mostrar geometria* permite visualizar o transdutor no plano  $xy$ .



**Figura A.16:** Configuração de um transdutor circular plano

### A.3.1.3 Simulação

A figura A.17 mostra o ambiente final após configuração. A simulação pode ser realizada, pressionando-se o botão *Simular*. Os resultados são visualizados a partir do botão *Visualizar campo acústico*. A escala do campo acústico pode ser configurada como linear ou logarítmica. Nesse caso, será utilizada uma escala logarítmica com limiar em  $-20dB$ . Isso significa que o campo acústico é normalizado de forma que o ponto  $0dB$  corresponda ao máximo de amplitude e  $-10dB$  ao mínimo de amplitude. O campo resultante é mostrado na figura A.18.

O mesmo botão habilita a configuração do modo de simulação em três dimensões. Essa opção será explorada na sequência. Os resultados podem ser salvos em um arquivo de dados com extensão *.mat*. O arquivo correspondente a esse exemplo pode ser diretamente carregado a partir do arquivo *tutorial1.mat*, no diretório *tutorial*.

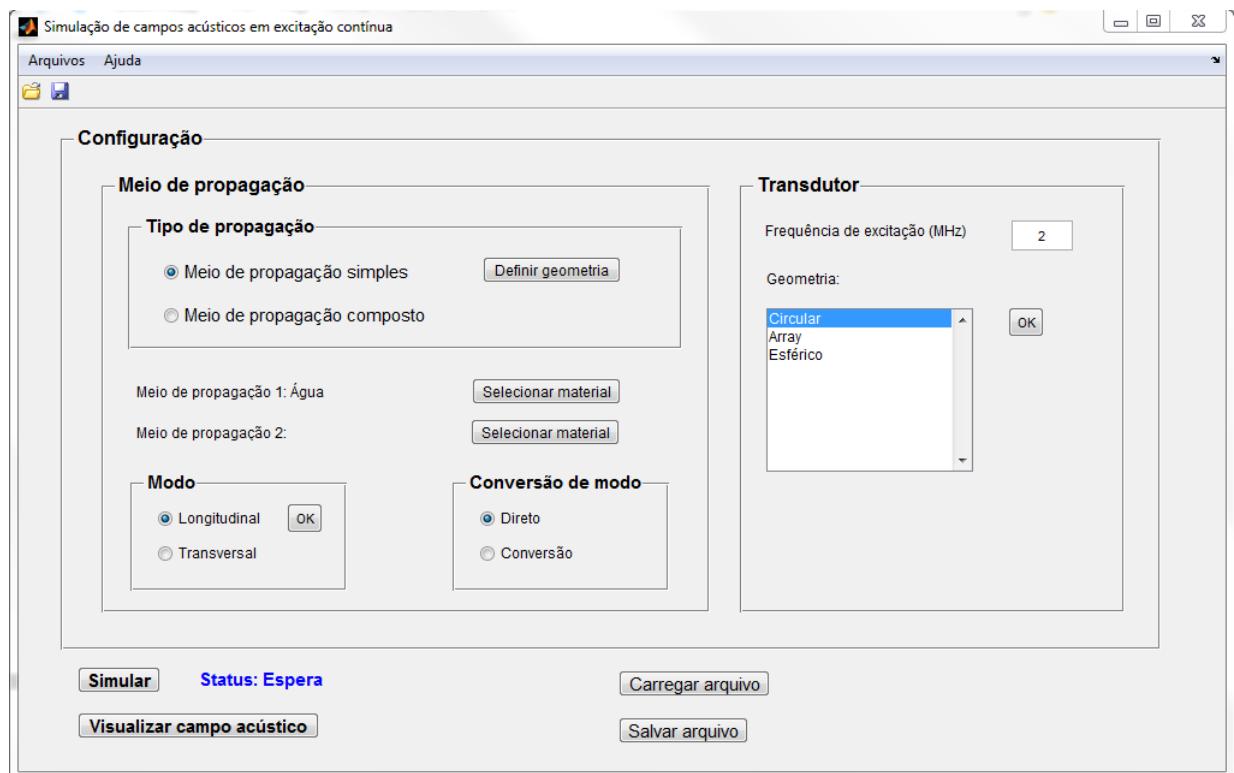


Figura A.17: Ambiente pós-configuração

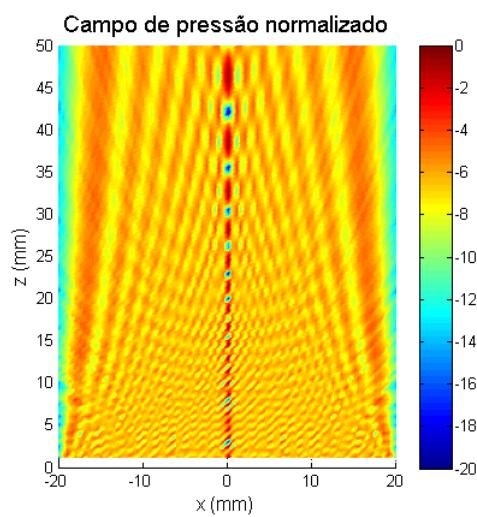
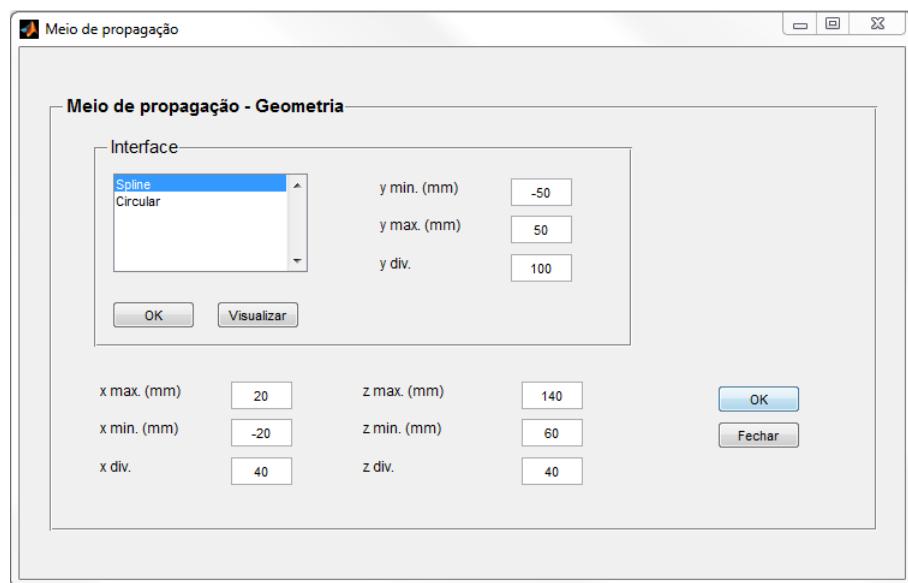


Figura A.18: Campo acústico para transdutor circular plano imerso em água

### A.3.2 Exemplo 2

#### A.3.2.1 Propagação em um meio composto

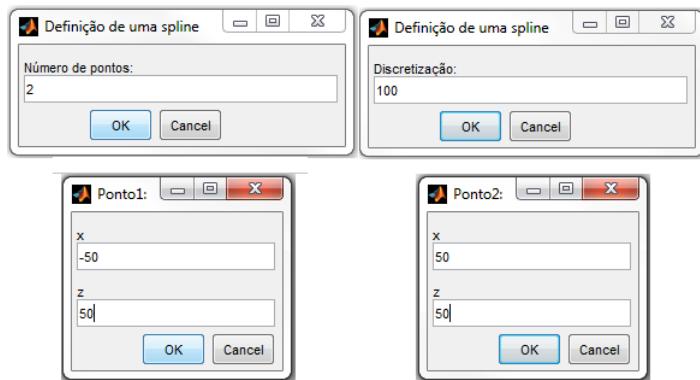
Neste exemplo será abordada a determinação do campo acústico no interior do sólido da figura A.12. O problema será complexificado, com a introdução de uma interface entre os meios e a utilização de um transdutor *array*. O primeiro passo será selecionar a opção *Meio de propagação composto* e em seguida *Definir geometria*. A nova geometria do problema será configurada conforme a figura A.19.



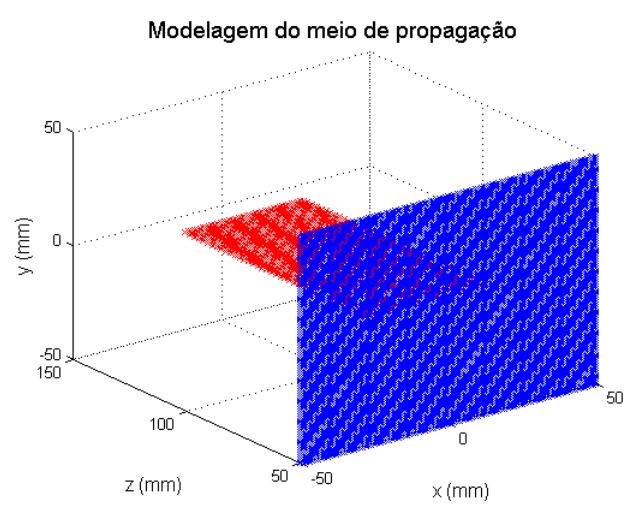
**Figura A.19:** Ambiente de configuração de um meio composto

Em primeiro lugar, configuram-se os limites em  $y$  da interface. Então, seleciona-se um tipo de interface, nesse caso uma *spline* e clica-se na opção *OK*. A *spline* de define o formato da interface e seus parâmetros de configuração variam em função do problema. Para o exemplo, como a interface é retangular, configuram-se 2 pontos, com discretização de 100 pontos. Os detalhes são dados em A.20. Em seguida, configuram-se os limites do plano  $xz$  no interior do sólido. Validando-se as configurações com *OK*, o programa exibe a configuração final em três dimensões, conforme figura A.21.

A seleção dos materiais é realizada da mesma forma que anteriormente. Como meio 1, seleciona-se a *Água* e como meio 2, o *Aço*. Para esse exemplo, será aplicado o modo de propagação longitudinal, com a opção de conversão de modo habilitada.



**Figura A.20:** Configuração de uma *spline*



**Figura A.21:** Geometria final do meio de propagação composto

### A.3.2.2 Configuração de um transdutor array

Neste exemplo, a frequência de  $2MHz$  é mantida. O transdutor utilizado é um *array*, configurado conforme as opções dadas no ambiente da figura A.22. As leis de atraso são configuradas no painel *Focalização*. Configuram-se as coordenadas do foco, no plano  $xz$ , e em seguida, clicando-se em *OK*, o ambiente de seleção dos métodos para determinação das leis de atraso é habilitado, como na figura A.23. Utiliza-se o método de *Fermat*. As leis de atraso são plotadas em um gráfico de barras, como em A.24. Adicionalmente, o usuário pode acrescentar leis de apodização ao transdutor. Neste exemplo, essa opção não será utilizada. Isso finaliza a etapa de modelagem do problema.

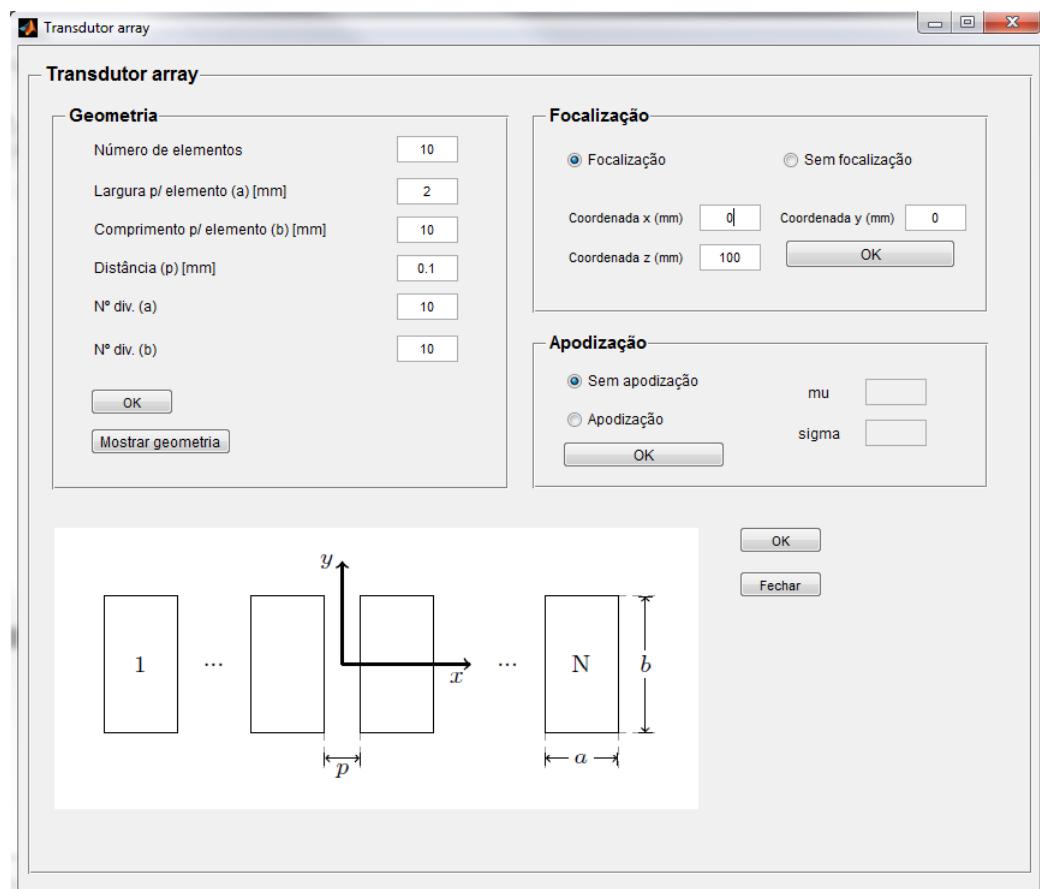
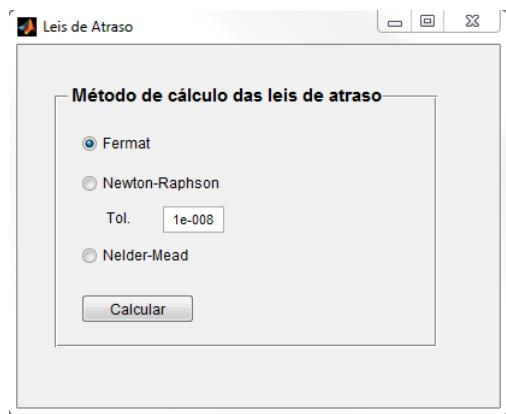


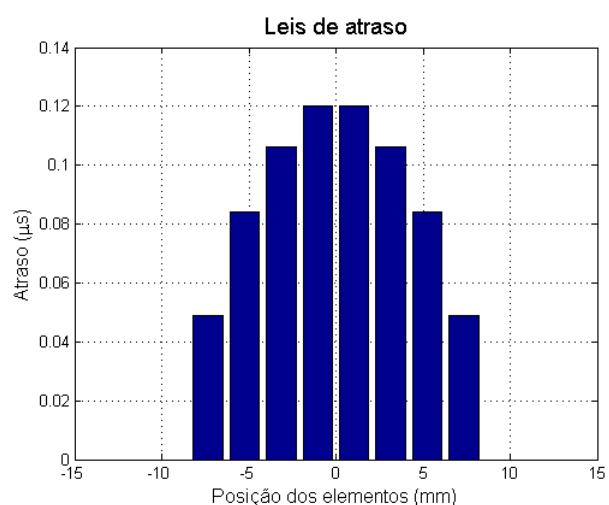
Figura A.22: Ambiente de configuração de um transdutor *array*

### A.3.2.3 Simulação

O campo acústico resultante da simulação é dado na figura A.25. Para o caso de um transdutor *array* o usuário pode habilitar o cursor do mouse para selecionar uma nova posição focal no próprio gráfico gerado, sem a necessidade de reconfigurar o programa. O

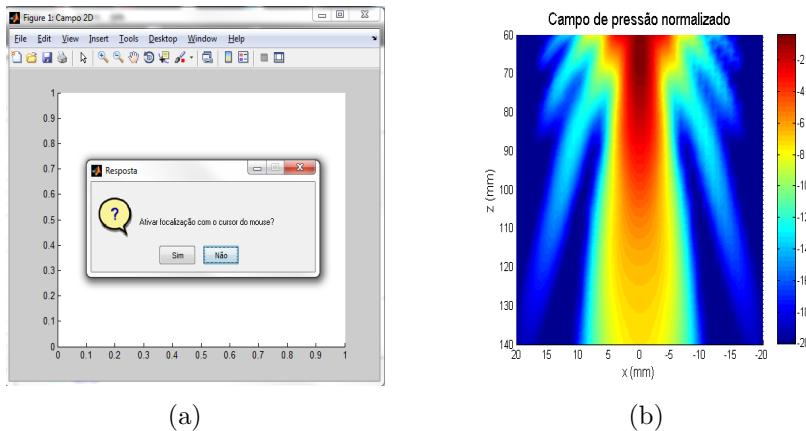


**Figura A.23:** Ambiente de configuração das leis de atraso de um transdutor *array*

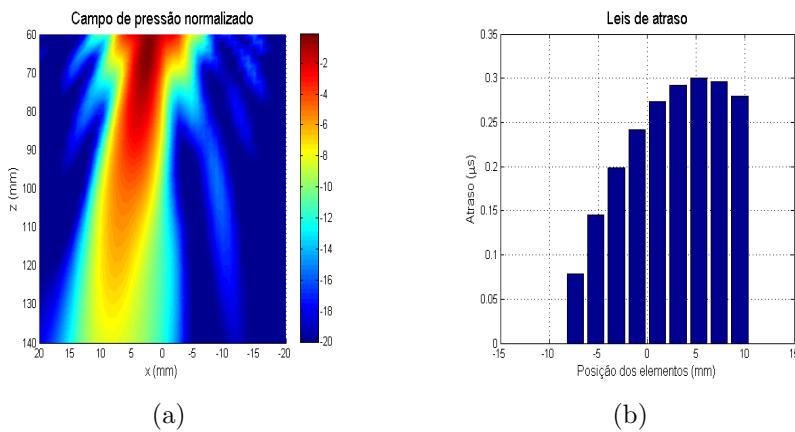


**Figura A.24:** Visualização das leis de atraso de um transdutor *array*

resultado é apresentado em A.26. A nova posição, assim como as leis de atraso associadas são plotadas em novas figuras. Nesse caso, o foco foi recalculado para a posição  $(x, y, z) = (5.55, 0, 100.35)$ .

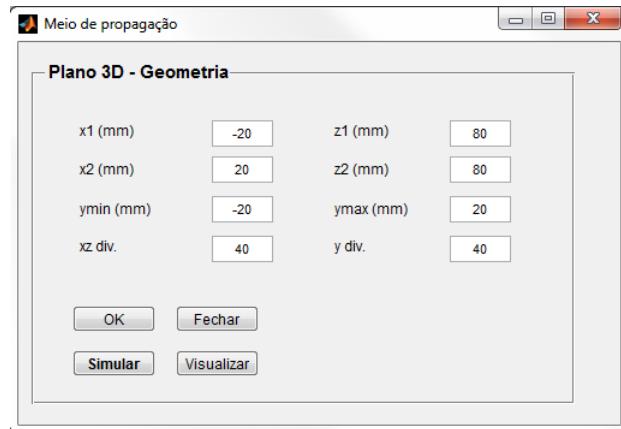


**Figura A.25:** Visualização de resultados para simulação com um transdutor *array*: (a) Habilitar focalização com posição do mouse, (b) Campo acústico em plano 2D

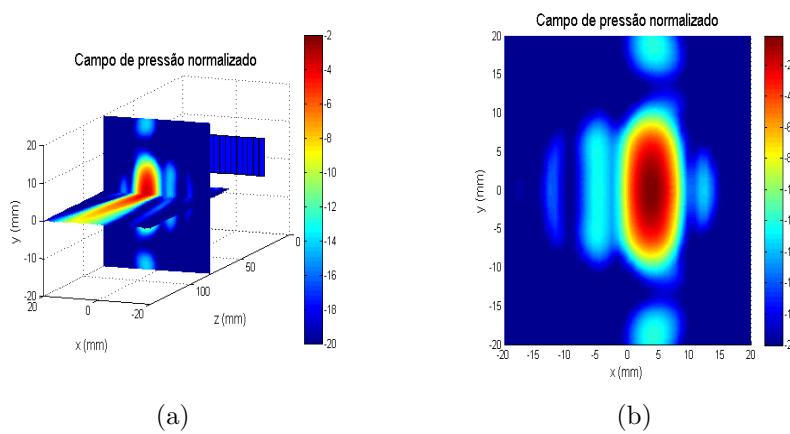


**Figura A.26:** Resultado para nova posição focal definida com o mouse para um transdutor *array*: (a) Campo acústico 2D, (b) Leis de atraso associadas

O usuário também pode realizar uma nova simulação em três dimensões. Para isso, deve configurar o ambiente A.27, que é exibido juntamente com o campo acústico em duas dimensões, clicando em *Visualizar campo acústico*. Nesse caso, o plano é configurado para a cota  $z = 80mm$ . As configuração são validades com o botão *OK* e uma nova simulação pode ser realizada. Os resultados são, em seguida, visualizados com o botão *Visualizar*. A figura A.28 mostra os dois gráficos resultantes, um em perspectiva e outro em duas dimensões. Os resultados são salvos no arquivo *tutorial2.mat*.



**Figura A.27:** Configuração de um plano em três dimensões



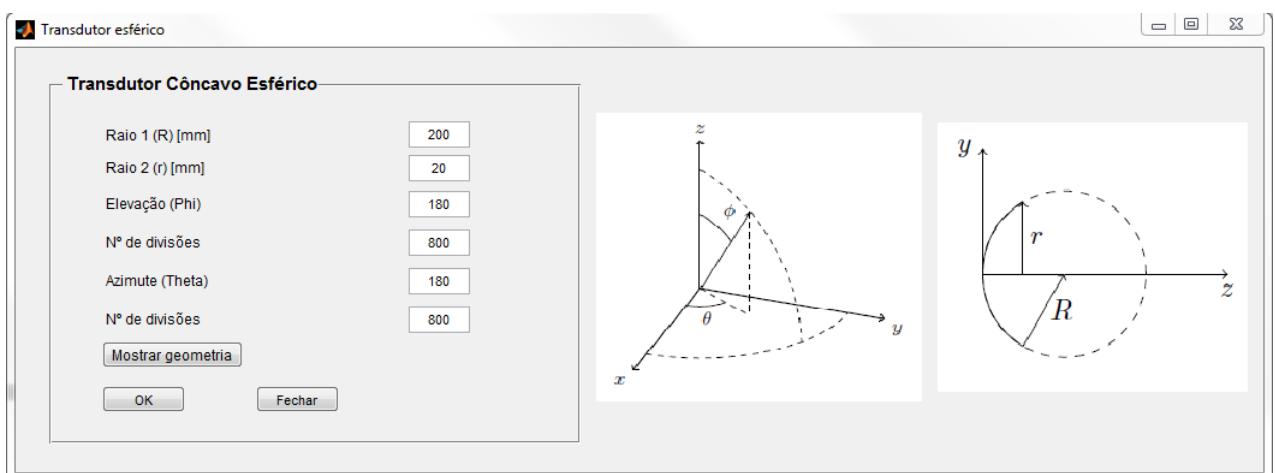
**Figura A.28:** Resultado para nova posição focal definida com o mouse para um transdutor *array*: (a) Vista em perspectiva, (b) Projeção no plano

### A.3.3 Exemplo 3

Neste exemplo, altera-se somente o tipo de transdutor utilizado. Será exemplificada a configuração de um transdutor circular côncavo, com frequência de excitação de  $2MHz$ .

#### A.3.3.1 Configuração de um transdutor circular côncavo

O usuário pode carregar o arquivo *tutorial2.mat* e selecionar a opção *Esférico* no painel de configuração do transdutor. O transdutor será configurado conforme as opções do ambiente dado na figura A.29. Como anteriormente, a geometria final do transdutor também pode ser visualizada.



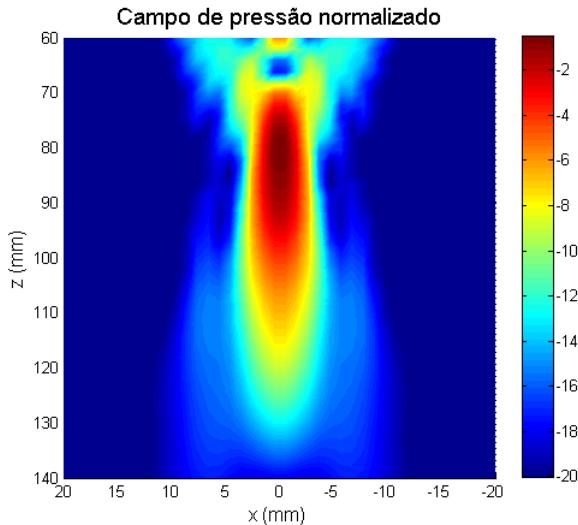
**Figura A.29:** Ambiente de configuração de um transdutor circular côncavo

#### A.3.3.2 Simulação

Os resultados da simulação são dados em A.30. Os desdobramentos da simulação são semelhantes aos casos anteriores. O usuário pode traçar o campo acústico em três dimensões, a partir do ambiente de configuração já mencionado, ou pode salvar os resultados e iniciar uma nova análise. Esse exemplo é salvo no arquivo *tutorial3.mat*.

## A.4 Considerações finais

Os exemplos acima tem por intuito fornecer um panorama geral de utilização da ferramenta. Evidentemente, o usuário é livre para testar outros tipos de configurações, como interfaces de diferentes formatos e materiais utilizados.



**Figura A.30:** Resultados para a simulação com um transdutor circular côncavo

A partir da forma *spline* o usuário é livre para modelar um grande número de problemas, tornando o formato da interface a ser configurada praticamente arbitrário. Além disso, existe a possibilidade de definir interfaces cilíndricas, a partir da opção *circular*, no mesmo ambiente. Os parâmetros de configuração são dois pontos pertencentes a uma circunferência traçada no plano *xz*, assim como seu centro. A orientação da interface pode variar em função da ordem de entrada dos parâmetros.

O objetivo da ferramenta gráfica é permitir tratar um certo número de problemas. Assim, caso o usuário deseje acrescentar uma nova funcionalidade ao programa, isso pode ser realizado mediante consulta da documentação das classes desenvolvidas.

## ANEXO B – COMPARAÇÃO COM O TOOLBOX FOCUS

Como mencionado no capítulo 1, existem algumas ferramentas abertas para a determinação do campo acústico que aplicam métodos numéricos diferentes dos implementados neste trabalho. O intuito desse capítulo é comparar os resultados obtidos com essas ferramentas e a solução proposta. Em particular, será utilizado o *toolbox FOCUS*, Chen e McGough (2008), que implementa uma solução baseada na transformada de Fourier, denominada *FNM*. A figura B.1 apresenta o aspecto geral do toolbox *FOCUS*.

A primeira diferença em relação à solução proposta neste trabalho é a arquitetura utilizada para o desenvolvimento da interface: toda a modelagem é concentrada no ambiente principal de configuração. Além disso, não é possível realizar simulações envolvendo a propagação entre diferentes meios, ou mesmo a determinação de leis de atraso para os transdutores *array*. Por outro lado, ela permite tratar um número maior de configurações de transdutores.

Serão considerados dois casos de propagação em um meio simples. No primeiro utiliza-se um transdutor circular plano e no segundo um transdutor *array*. Em ambos os casos o meio de propagação é a água. As configurações são listadas na tabela B.1. O campo acústico é calculado em uma região do plano *xz*.

### B.1 Transdutor circular plano

Utiliza-se um transdutor circular plano de raio  $r = 10mm$  e  $r_{div} = 40$ . O passo é de  $0,5mm$ , 6 vezes menor que o comprimento de onda, correspondente a  $\lambda = 3mm$ . Os resultados são dados em B.2. A escala linear é ajustada de maneira a considerar apenas o campo acústico calculado para a região  $z > \lambda$ . Como nota-se a partir das figuras em B.2, os resultados são idênticos.

### B.2 Transdutor array

As configurações utilizadas para o transdutor *array* são dadas na tabela B.2. Considera-se um *array* de 10 elementos espaçados em  $0.1mm$ . O passo de discretização é

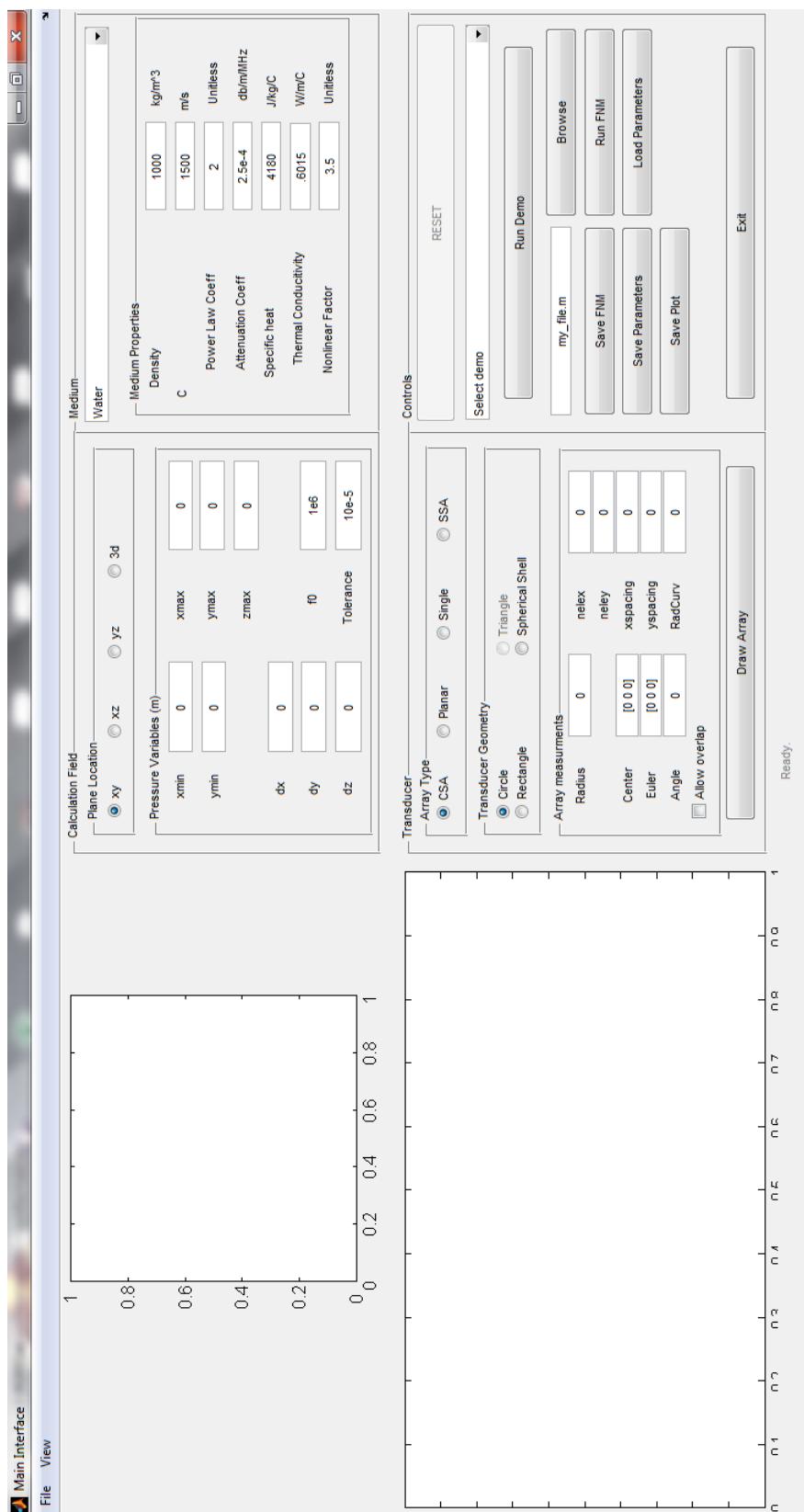
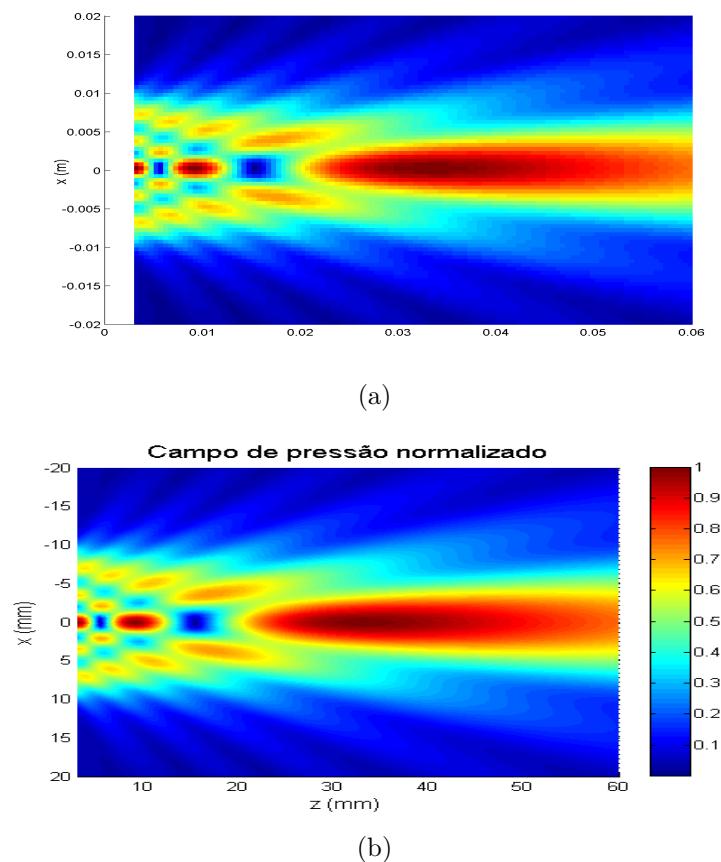


Figura B.1: Ambiente de configuração da ferramenta FOCUS



**Figura B.2:** Resultados para teste de imersão, transdutor circular plano em modo longitudinal: (a) Simulação em *FOCUS*, (b) Simulação em *MATLAB*<sup>®</sup>

**Tabela B.1:** Parâmetros de configuração

Parâmetro	Valor
$f$	$0.5MHz$
$x_{min}$	$-20mm$
$x_{max}$	$20mm$
$z_{min}$	$0mm$
$z_{max}$	$60mm$
$x_{div}$	$80mm$
$z_{div}$	$120mm$
Meio	Água
$c_l(m/s)$	$1500m/s$
$c_s(m/s)$	—
$\rho_0(kg/m^3)$	$1000kg/m^3$

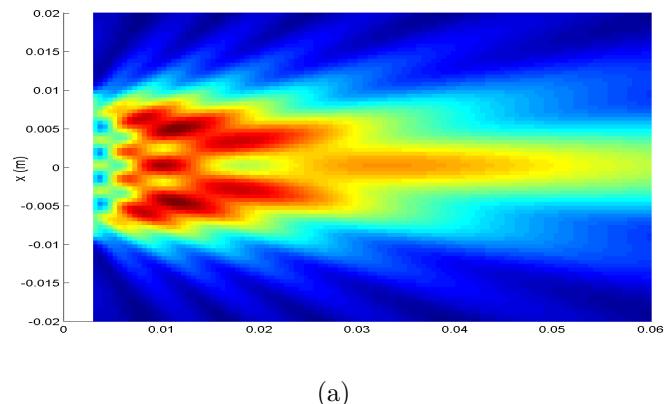
de  $0.5mm$  nas direções horizontal e vertical. Os resultados são apresentados em B.3. Utiliza-se uma escala linear. Novamente nota-se que os resultados são praticamente idênticos.

### B.3 Discussão

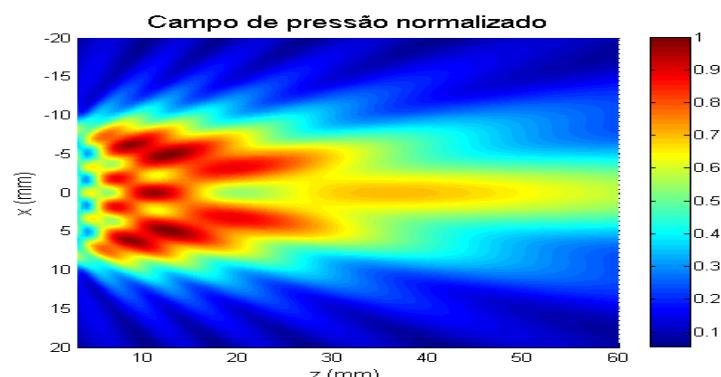
Os resultados apresentados neste capítulo, servem para validar as simulações realizadas para o caso de propagação em um meio simples, comparando-as com uma ferramenta aberta disponível na literatura e que emprega um método distinto do escolhido neste trabalho. Os resultados apresentados demonstram que as soluções obtidas com o método proposto neste trabalho são consistentes em relação àquelas dadas pelo *toolbox FOCUS*. Ressalta-se que essa ferramenta não permite a realização de simulações envolvendo múltiplos meios de propagação. Assim, em um primeiro momento, o método *MTM* mostra-se mais versátil que o método *FNM*, uma vez que ambos trabalham com a resposta de transdutores em excitação contínua.

**Tabela B.2:** Parâmetros de configuração

Parâmetro	Valor
$n$	10
$a$	$2mm$
$b$	$10mm$
$l_{div}$	5
$h_{div}$	21
$p$	$0.1mm$



(a)



(b)

**Figura B.3:** Resultados para teste de imersão, transdutor circular plano em modo longitudinal: (a) Simulação em *FOCUS*, (b) Simulação em *MATLAB*®