

RODRIGO MOLOGNI GONÇALVES DOS SANTOS

**VISUALIZAÇÃO DE FENÔMENOS DE PROPAGAÇÃO
DE ONDAS MECÂNICAS EM MEIO CONTÍNUO**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação *Stricto Sensu* em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação (Fec) da Universidade Estadual de Campinas (Unicamp), como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica, na área de concentração de Engenharia de Computação.

Orientador: Prof. Dr. José Mario De Martino

Campinas, SP

2011

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Sa59v Santos, Rodrigo Mologni Gonçalves dos
Visualização de fenômenos de propagação de ondas
mecânicas em meio contínuo / Rodrigo Mologni Gonçalves
dos Santos. -- Campinas, SP: [s.n.], 2011.

Orientador: José Mario De Martino.
Dissertação de Mestrado - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Visualização. 2. Ondas elásticas. 3. Mecânica dos
meios contínuos. 4. Mecânica dos sólidos. I. De Martino,
José Mario. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Elétrica e de Computação. III.
Título.

Título em Inglês: *Visualization of mechanical wave propagation phenomena in continuous medium*
Palavras-chave em Inglês: *Visualization, Elastic waves, Continuum mechanics, Solid mechanics*
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora: Renato Pavanello, Léo Pini Magalhães
Data da defesa: 04/08/2011
Programa de Pós-Graduação: Engenharia Elétrica

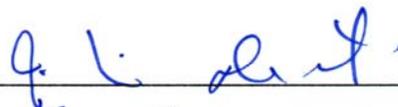
COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Rodrigo Mologni Gonçalves dos Santos

Data da Defesa: 04 de agosto de 2011.

Título da Tese: "Visualização de fenômenos de propagação de ondas mecânicas em meio contínuo".

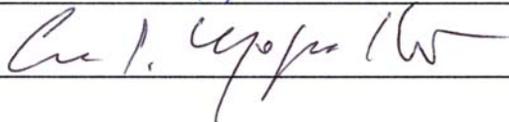
Prof. Dr. José Mario De Martino (Presidente):



Prof. Dr. Renato Pavanello:



Prof. Dr. Léo Pini Magalhães:



Esta dissertação de mestrado é dedicada aos meus familiares, que não tiveram a oportunidade de avançar nos estudos, pois precisaram trabalhar desde muito cedo; e especialmente à minha mãe, Sr.^a Zilda Mologni, que apesar das dificuldades concedeu-me a oportunidade de progredir nos estudos.

AGRADECIMENTOS

Agradeço aos meus ex-supervisores da Embrapa Informática Agropecuária, M.^a Adriana Delfino dos Santos, Dr. Leandro Henrique Mendonça de Oliveira e Dr. Leonardo Ribeiro Queiros, pelos quase dois anos de experiência na área de engenharia de *software* e por terem me incentivado fortemente a ingressar no curso de mestrado.

Agradeço ao meu ex-orientador do curso de graduação, Prof. Dr. Celmar Guimarães da Silva, da Faculdade de Tecnologia (FT) da Unicamp, por ter me introduzido na área de visualização de informação e incentivado a prosseguir nos estudos, ingressando no curso de mestrado.

Agradeço ao meu orientador, Prof. Dr. José Mario De Martino, por ter me aceito como um de seus orientados de mestrado e pela sua competente orientação e contribuições no desenvolvimento do trabalho apresentado.

Agradeço ao meu co-orientador, Prof. Dr. Euclides de Mesquita Neto, da Faculdade de Engenharia Mecânica (FEM) da Unicamp, pela sua co-orientação e contribuições no desenvolvimento do trabalho apresentado.

Agradeço aos docentes que ministraram as disciplinas cursadas por mim durante o curso de mestrado, Prof. Dr. Alexandre Xavier Falcão, Prof. Dr. Clésio Luis Tozzi, Prof. Dr. Roberto de Alencar Lotufo e Prof.^a Dr.^a Wu Shin-Ting, pelos seus ensinamentos, que foram de fundamental importância para a minha formação acadêmica e profissional.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes) pela bolsa de estudos concedida ao longo destes dois anos, pois sem este benefício financeiro eu não poderia ter me dedicado integralmente aos estudos.

Agradeço ao Prof. Dr. Pérsio Leister de Almeida Barros, da Faculdade de Engenharia Civil, Arquitetura e Urbanismo (FEC) da Unicamp, pela sua participação no processo de avaliação do ambiente de visualização desenvolvido, pelo conjunto de dados tridimensionais fornecidos para testes, pela sua dedicação e disposição em ajudar, e pelas suas sugestões e demais colaborações.

Agradeço ao Sr. Regis Terenzi Neuenschwander, líder do Grupo de Projetos do Laboratório Nacional de Luz Síncrotron (LNLS), por ter me recepcionado e apresentado as dependências do laboratório e pelas informações enriquecedoras compartilhadas.

“Uma imagem vale mais do que mil palavras”
(Fred R. Barnard)

RESUMO

Nas últimas décadas, o estudo dos fenômenos de propagação de ondas mecânicas e de suas consequências tem recebido crescente atenção de grupos de pesquisa de variados ramos da engenharia, sobretudo daqueles dedicados ao desenvolvimento de métodos computacionais para resolução de problemas complexos na área da Mecânica dos Meios Contínuos (MMC). Os métodos computacionais desenvolvidos por estes grupos geralmente são constituídos de modelagem matemática e simulação computacional de problemas físicos, que substituem a necessidade do uso de protótipos ou infraestrutura de laboratórios, mas que em contrapartida produzem uma volumosa quantidade de dados numéricos que pode dificultar ou inviabilizar uma análise mais eficiente dos resultados sem o suporte de técnicas específicas para o mapeamento destes dados em representações visuais. Esta dissertação apresenta um trabalho científico-tecnológico na área de Visualização Científica, cujas técnicas foram empregadas em MMC para a visualização dos fenômenos de propagação de ondas mecânicas, especificamente em meios materiais sólidos. Os objetivos principais do trabalho foram possibilitar que os especialistas pudessem validar mais rapidamente os seus métodos computacionais voltados ao estudo dos fenômenos de propagação de ondas mecânicas e, principalmente, que eles pudessem entender com mais facilidade os resultados gerados pela simulação computacional, ampliando assim a eficiência da análise de problemas complexos. Os estudos de casos realizados e apresentados no trabalho permitem afirmar que estes objetivos foram alcançados e os resultados foram satisfatórios. As principais contribuições deste trabalho são: identificação de um conjunto de informações derivadas dos dados brutos gerados pela simulação computacional que podem ser empregados para análise dos fenômenos de propagação de ondas mecânicas; metodologias e respectivas formulações matemáticas para cálculo destas informações; desenvolvimento de um protótipo de ambiente de visualização dos dados que descrevem o comportamento das ondas mecânicas; e estudo e definição de diretrizes de como as técnicas de visualização de dados escalares e vetoriais podem ser aplicadas para favorecer a análise das informações disponíveis. É importante observar que o trabalho é limitado à visualização de ondas mecânicas bi ou tridimensionais geradas sobre grades de dimensão bidimensional.

Palavras-chave: Visualização. Visualização de dados. Visualização científica. Ondas elásticas. Ondas mecânicas. Mecânica dos meios contínuos. Mecânica dos sólidos.

ABSTRACT

In recent decades, the study of the mechanical wave propagation phenomena and their consequences has been receiving increasing attention from research groups in engineering, principally those dedicated to the development of computational methods for solving complex problems in the area of Continuum Mechanics (CM). The computational methods developed by these groups usually consist of mathematical modeling and computer simulation of physical problems, which replace the need for prototypes or laboratory infrastructure. But in opposition, produces a large amount of numerical data that can hinder a more efficient analysis of the results without the support of specific techniques for mapping these data into visual representations. This master's dissertation presents a scientific-technological work in the area of Scientific Visualization, whose techniques were applied in the area of CM for visualization of the mechanical wave propagation phenomena, specifically in solid material media. The main objectives of the work were to allow MCM experts to validate more quickly their computational methods for the study of the mechanical wave propagation phenomena and, especially, that they could more easily understand the results generated by computer simulation, increasing the efficiency of analysis of complex problems. The case studies analyzed and presented in the dissertation allow to state that these objectives were attained and the results were satisfactory. The main contributions of the work are: identification of a set of implicit information, derived from raw data generated by computer simulation, that can be used to analyze the mechanical wave propagation phenomena; methodologies and their mathematical formulations to calculate these information; development of a visualization environment prototype of the data that describe the behavior of mechanical waves; and study and guideline definition of how to apply the visualization techniques of scalar and vector data for expand the analysis the available information. It is important to mention that the work is limited to visualization of two or three-dimensional mechanical waves generated on two-dimensional grids.

Keywords: *Visualization. Data visualization. Scientific visualization. Elastic waves. Mechanical waves. Continuum mechanics. Solid mechanics.*

LISTA DE FIGURAS

Figura 1.1	Procedimento convencional para resolução de problemas complexos em MMC	2
Figura 1.2	Procedimento proposto para resolução de problemas complexos em MMC	3
Figura 2.1	Representação dos seis principais tipos de células (baseado em Schroeder et al., 2006)	11
Figura 2.2	Exemplos dos quatro principais tipos de grades (baseado em Telea, 2007)	13
Figura 2.3	Fluxo de visualização convencional (baseado em Telea, 2007)	18
Figura 2.4	Mapeamento em cores pelos métodos baseados em textura e vértice	20
Figura 2.5	Exemplos de uso das três principais técnicas de visualização de dados escalares	21
Figura 2.6	Exemplos de uso das duas principais técnicas de visualização de dados vetoriais	22
Figura 2.7	Exemplo de uso da técnica de extração de subgrade	24
Figura 2.8	Exemplos de uso das duas principais técnicas de modelagem implícita	24
Figura 2.9	Exemplo de uso da visão estereoscópica por anaglifo	27
Figura 3.1	Representação de um problema geomecânico complexo em Mecânica dos Sólidos	30
Figura 3.2	Etapas da deformação de um material sólido gerada pela transmissão de ondas de corpo (baseado em Fowler, 2005)	33
Figura 3.3	Etapas da deformação de um material sólido gerada pela transmissão de ondas de superfície	36
Figura 3.4	Tipos de frente de onda	38
Figura 4.1	Disposição das amostras sobre uma grade estruturada 2D	42
Figura 4.2	Posição relativa de uma amostra após sofrer um deslocamento	45
Figura 4.3	Vértices da elipse que descreve a trajetória de deslocamento de uma amostra	48
Figura 4.4	Variação da relação de aspecto de uma elipse	49
Figura 4.5	Sentido do movimento de rotação da trajetória de uma partícula para o caso das ondas mecânicas 2D	52
Figura 4.6	Projeção da trajetória de uma partícula no plano do meio de propagação das ondas mecânicas para obtenção do sentido do movimento de rotação para o caso das ondas mecânicas 2D	53

Figura 4.7	Sentido do movimento de rotação da trajetória de uma partícula para o caso das ondas mecânicas 3D	54
Figura 4.8	Ângulo de inclinação da trajetória de uma amostra para o caso 2D	56
Figura 4.9	Sentido de propagação das ondas a partir de um corpo rígido	56
Figura 4.10	Ângulo de inclinação da trajetória de uma amostra para o caso 3D	58
Figura 4.11	Amplitudes das ondas sobre os eixos de coordenadas x e y	59
Figura 4.12	Ampliação da intensidade de oscilação de uma amostra com uso do fator de escala	60
Figura 4.13	Extração de dados por extração de subgrade	63
Figura 4.14	Extração de dados por recorte com um plano, um paralelepípedo e uma esfera	64
Figura 4.15	Extração de dados por fatiamento com um plano, uma paralelepípedo e uma esfera	64
Figura 5.1	Fluxo de visualização proposto implementado no MWPVis	75
Figura 5.2	Fluxo visualização padrão do MWPVis	75
Figura 5.3	Aplicação do algoritmo de <i>Extract Subset</i>	79
Figura 5.4	Aplicação do algoritmo de <i>Warp By Scalar</i>	80
Figura 5.5	Aplicação do algoritmo de <i>Reflect</i>	81
Figura 5.6	Aplicação do algoritmo de <i>Clip</i>	82
Figura 5.7	Aplicação do algoritmo de <i>Colormap</i>	84
Figura 5.8	Aplicação do algoritmo de <i>Contour</i>	85
Figura 5.9	Aplicação do algoritmo de <i>Glyph</i> para visualização das normais	86
Figura 5.10	Aplicação do algoritmo de <i>Glyph</i> para visualização dos atributos de dados vetoriais	86
Figura 5.11	Aplicação do algoritmo de <i>Outline</i>	87
Figura 5.12	Aplicação do algoritmo de <i>Slice</i>	88
Figura 5.13	Aplicação do algoritmo de <i>Stream Tracer</i>	90
Figura 5.14	Animação por controle da visibilidade dos atores	91
Figura 5.15	Aplicação de alguns <i>widgets</i> disponíveis no VTK (SCHROEDER; MARTIN; LORENSEN, 2006)	93
Figura 5.16	Leiaute simplificado da GUI do MWPVis	94
Figura 5.17	Organização e aplicação das classes de GUI	95
Figura 6.1	Problema geomecânico 2D com 50 camadas viscoelásticas horizontais	98
Figura 6.2	Visualização das componentes de deslocamento horizontal e vertical	99

Figura 6.3	Visualização da relação de aspecto da trajetória das partículas	100
Figura 6.4	Visualização da trajetória das partículas	101
Figura 6.5	Visualização do ângulo de inclinação das trajetórias das partículas	102
Figura 6.6	Visualização do sentido do movimento de rotação	103
Figura 6.7	Visualização do ângulo de fase das componentes de deslocamento horizontal e vertical	104
Figura 6.8	Visualização da amplitude das componentes de deslocamento horizontal e vertical	105
Figura 6.9	Problema geomecânico 3D de um bloco cilíndrico rígido em um meio isotrópico	106
Figura 6.10	Visualizações dos deslocamentos em x , y e z no instante de tempo $t = 0$	108
Figura 6.11	Visualizações da relação de aspecto das trajetórias elípticas das partículas	109
Figura 6.12	Visualizações das amplitudes das ondas ao longo dos eixos x , y e z	110
Figura 6.13	Visualização do ângulo de fase da componente horizontal u_x	111

LISTA DE QUADROS

Quadro 4.1 Relação de aplicação das técnicas de visualização com as informações enriquecidas 65

LISTA DE LISTAGENS

Listagem 5.1	Exemplo de dados salvos em um arquivo TXT.	76
Listagem 5.2	Exemplo de uso de uma instância da classe WanglayReader	77
Listagem 5.3	Exemplo de uso de uma instância da classe Wanglay2StructuredGridSeries	78
Listagem 5.4	Exemplo de uso de uma instância da classe vtkExtractGrid	79
Listagem 5.5	Exemplo de uso de uma instância da classe vtkWarpScalar	80
Listagem 5.6	Exemplo de uso de uma instância da classe vtkReflectionFilter	82
Listagem 5.7	Exemplo de uso de uma instância da classe vtkClipDataSet	83
Listagem 5.8	Exemplo de uso de uma instância da classe vtkColorTransferFunction	84
Listagem 5.9	Exemplo de uso de uma instância da classe vtkContourFilter	85
Listagem 5.10	Exemplo de uso de uma instância da classe vtkGlyph3D	87
Listagem 5.11	Exemplo de uso de uma instância a classe vtkOutlineFilter	88
Listagem 5.12	Exemplo de uso de uma instância da classe vtkCutter	89
Listagem 5.13	Exemplo de uso de uma instância da classe vtkStreamTracer	90
Listagem 5.14	Exemplo de animação pelo VTK	91
Listagem 5.15	Uso de visões múltiplas e interação com o usuário com VTK	92
Listagem 5.16	Uso de visão estereoscópica por anaglifo por intermédio do VTK	92

LISTA DE TABELAS

Tabela 3.1	Organização dos dados gerados por simulação computacional	31
Tabela 4.1	Ordem de variação crescente das coordenadas	43

LISTA DE SIGLAS

1D	unidimensional
2D	bidimensional
3D	tridimensional
ASC	<i>Advanced Simulation and Computing</i> (Computação e Simulação Avançadas)
CCW	<i>counter-counterwise</i> (anti-horário)
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
CSV	<i>Comma-Separated Values</i> (Valores Separados por Vírgula)
CW	<i>counterwise</i> (horário)
DGP	Diretório dos Grupos de Pesquisa no Brasil
DMC	Departamento de Mecânica Computacional
DOE	<i>Department of Energy</i> (Departamento de Energia)
EUA	Estados Unidos da América
FEM	Faculdade de Engenharia Mecânica
GCMCMC	Grupo de Métodos Computacionais em Mecânica do Contínuo
GPL	<i>General Public License</i> (Licença Pública Geral)
GPU	<i>Graphics Processing Unit</i> (Unidade de Processamento Gráfico)
GUI	<i>Graphical User Interface</i> (Interface Gráfica com o Usuário)
IDE	<i>Integrated Development Environment</i> (Ambiente de Desenvolvimento Integrado)
LGPL	<i>Lesser General Public License</i> (Licença Pública Geral Reduzida)
MMC	Mecânica dos Meios Contínuos
MSDNAA	<i>Microsoft Developer Network Academic Alliance</i> (Aliança Acadêmica da Rede de Desenvolvimento da Microsoft)
MSVC	Microsoft Visual C++
MWPV _{is}	<i>Mechanical Wave Propagation Visualizer</i> (Visualizador de Propagação de Ondas Mecânicas)
NA-MIC	<i>National Alliance for Medical Image Computing</i> (Aliança Nacional para Processamento de Imagens Médicas)
ND	multidimensional
NLM	<i>National Library of Medicine</i> (Biblioteca Nacional de Medicina)
NSF	<i>National Science Foundation</i> (Fundação Nacional da Ciência)

NSIS	<i>Nullsoft Scriptable Install System</i> (Sistema de Instalação por <i>Script</i> da Nullsoft)
SDK	<i>Software Development Kit</i> (Kit de Desenvolvimento de <i>Software</i>)
TXT	arquivo de texto sem formatação
Unicamp	Universidade Estadual de Campinas
VOI	<i>Volume of Interest</i> (Volume de Interesse)
VTK	<i>Visualization Toolkit</i> (Kit de Ferramentas de Visualização)
XML	<i>Extensible Markup Language</i> (Linguagem de Marcação Estendida)

SUMÁRIO

1 INTRODUÇÃO	1
1.1 CONTEXTUALIZAÇÃO	1
1.2 METODOLOGIA, LIMITAÇÕES E CONTRIBUIÇÕES	2
1.3 ORGANIZAÇÃO DA DISSERTAÇÃO	4
2 VISUALIZAÇÃO CIENTÍFICA	7
2.1 INTRODUÇÃO	7
2.2 REPRESENTAÇÃO DE DADOS	9
2.2.1 Conjunto de dados discretos	9
2.2.2 Tipos de células	10
2.2.2.1 Vértice	11
2.2.2.2 Linha	11
2.2.2.3 Triângulo	11
2.2.2.4 Quadrilátero	12
2.2.2.5 Tetraedro	12
2.2.2.6 Hexaedro	12
2.2.3 Tipos de grades	12
2.2.3.1 Grades uniformes	14
2.2.3.2 Grades retilíneas	14
2.2.3.3 Grades estruturadas	15
2.2.3.4 Grades não-estruturadas	15
2.2.4 Tipos de atributos de dados	16
2.2.4.1 Atributos escalares	16
2.2.4.2 Atributos vetoriais	16
2.2.4.3 Atributos tensoriais	17
2.3 FLUXO DE VISUALIZAÇÃO	17
2.3.1 Importação	17
2.3.2 Filtragem	18
2.3.3 Mapeamento	18
2.3.4 Renderização	18
2.4 VISUALIZAÇÃO DE DADOS ESCALARES	19

2.4.1 Mapeamento em cores	19
2.4.2 Contorno	20
2.4.3 Função de altura	22
2.5 VISUALIZAÇÃO DE DADOS VETORIAIS	22
2.5.1 Glifos de vetor	23
2.5.2 Linhas de fluxo	23
2.6 TÉCNICAS DE MODELAGEM	23
2.6.1 Extração de subgrade	24
2.6.2 Recorte	25
2.6.3 Fatiamento	25
2.7 TÉCNICAS DE COMPUTAÇÃO GRÁFICA	25
2.7.1 Animação	25
2.7.2 Interação de usuário	26
2.7.3 Renderização estereoscópica	26
2.8 CONSIDERAÇÕES FINAIS	27
3 SIMULAÇÃO E ANÁLISE DE ONDAS MECÂNICAS	29
3.1 SIMULAÇÃO DE ONDAS MECÂNICAS	29
3.1.1 Trabalhos relacionados	29
3.1.2 Simulação computacional	30
3.1.3 Dados brutos	31
3.2 ONDAS MECÂNICAS EM MEIOS SÓLIDOS	32
3.2.1 Introdução	32
3.2.2 Tipos de ondas	33
3.2.2.1 Ondas de corpo	34
3.2.2.1.1 Ondas <i>P</i>	34
3.2.2.1.2 Ondas <i>S</i>	35
3.2.2.2 Ondas de superfície	35
3.2.2.2.1 Ondas <i>R</i>	36
3.2.2.2.2 Ondas <i>L</i>	37
3.2.3 Frentes de onda	37
3.2.3.1 Frente de onda plana	37
3.2.3.2 Frente de onda cilíndrica	39

3.2.3.3 Frente de onda esférica	39
3.2.4 Fenômenos de propagação de ondas	39
3.2.4.1 Absorção	39
3.2.4.2 Reflexão	40
3.2.4.3 Refração	40
3.2.4.4 Difração	40
3.3 CONSIDERAÇÕES FINAIS	40
4 VISUALIZAÇÃO DE ONDAS MECÂNICAS	41
4.1 IMPORTAÇÃO	42
4.2 ENRIQUECIMENTO	44
4.2.1 Deslocamento	44
4.2.2 Posição relativa	45
4.2.3 Trajetória	46
4.2.4 Relação de aspecto	49
4.2.5 Sentido do movimento de rotação	50
4.2.5.1 Ondas mecânicas 2D	51
4.2.5.2 Ondas mecânicas 3D	52
4.2.6 Ângulo de inclinação	54
4.2.6.1 Ondas mecânicas 2D	55
4.2.6.2 Ondas mecânicas 3D	57
4.2.7 Amplitude das ondas	58
4.2.8 Ângulo de fase das ondas	59
4.2.9 Fator de escala	60
4.2.10 Sobreposição de ondas	61
4.2.11 Gradiente da amplitude das ondas	62
4.3 FILTRAGEM	62
4.4 MAPEAMENTO	65
4.5 RENDERIZAÇÃO	66
4.6 CONSIDERAÇÕES FINAIS	68
5 AMBIENTE DE VISUALIZAÇÃO DE ONDAS MECÂNICAS	69
5.1 INTRODUÇÃO	69
5.2 AMBIENTE DE DESENVOLVIMENTO	70

5.2.1 Linguagem de programação	71
5.2.2 Biblioteca de classes para visualização de dados científicos	72
5.2.3 Biblioteca de classes para construção e gerenciamento de GUI	73
5.3 FUNCIONAMENTO DO AMBIENTE DE VISUALIZAÇÃO	74
5.3.1 Importação	76
5.3.2 Enriquecimento	77
5.3.3 Filtragem	78
5.3.3.1 <i>Extract Subset</i>	78
5.3.3.2 <i>Warp By Scalar</i>	79
5.3.3.3 <i>Reflect</i>	81
5.3.3.4 <i>Clip</i>	82
5.3.4 Mapeamento	83
5.3.4.1 <i>Colormap</i>	83
5.3.4.2 <i>Contour</i>	84
5.3.4.3 <i>Glyph</i>	85
5.3.4.4 <i>Outline</i>	88
5.3.4.5 <i>Slice</i>	89
5.3.4.6 <i>Stream Tracer</i>	89
5.3.5 Renderização	90
5.3.6 Interface	92
5.4 CONSIDERAÇÕES FINAIS	96
6 ESTUDOS DE CASO	97
6.1 ONDAS MECÂNICAS 2D	97
6.1.1 Descrição do problema	97
6.1.2 Visualizações	98
6.1.2.1 Visualização do deslocamento	98
6.1.2.2 Visualização da relação de aspecto	99
6.1.2.3 Visualização da trajetória	100
6.1.2.4 Visualização do ângulo de inclinação	102
6.1.2.5 Visualização do sentido do movimento de rotação	103
6.1.2.6 Visualização do ângulo de fase	104
6.1.2.7 Visualização da amplitude	105

6.2 ONDAS MECÂNICAS 3D	105
6.2.1 Descrição do problema	105
6.2.2 Visualizações	107
6.2.2.1 Visualização do deslocamento	107
6.2.2.2 Visualização da relação de aspecto	109
6.2.2.3 Visualização da amplitude	110
6.2.2.4 Visualização do ângulo de fase	110
6.3 CONSIDERAÇÕES FINAIS	111
7 DISCUSSÃO E AVALIAÇÃO DAS VISUALIZAÇÕES	113
7.1 INFORMAÇÕES OBTIDAS PELOS ATRIBUTOS DE DADOS	113
7.1.1 Variação temporal	113
7.1.2 Fator de escala	113
7.1.3 Amplitude das ondas	114
7.1.4 Relação de aspecto	114
7.1.5 Deslocamentos	115
7.1.6 Ângulo de inclinação	115
7.1.7 Trajetória	116
7.1.8 Sentido do movimento de rotação	116
7.1.9 Ângulo de fase	117
7.2 EFICIÊNCIA DAS TÉCNICAS DE VISUALIZAÇÃO	117
7.2.1 Mapeamento em cores	117
7.2.2 Contorno	118
7.2.3 Função de altura	118
7.2.4 Glifos de vetor	118
7.2.5 Linhas de fluxo	119
7.2.6 Extração de subgrade, recorte e fatiamento	119
7.2.7 Trajetória de partícula	119
7.3 CONSIDERAÇÕES FINAIS	120
8 CONCLUSÕES	121
8.1 CONSIDERAÇÕES FINAIS	121
8.2 CONTRIBUIÇÕES	122
8.3 LIMITAÇÕES E TRABALHOS FUTUROS	122

REFERÊNCIAS	125
PUBLICAÇÕES	129
APÊNDICES	131
APÊNDICE A – Configuração do ambiente de desenvolvimento do MWPVis	133
A.1 Instalação dos pré-requisitos	133
A.2 Criação do produto VTK	136
A.3 Compilação e instalação do VTK	137
A.4 Configuração do IDE	139
A.5 Importação do MWPVis	142
APÊNDICE B – Instalação do MWPVis	145
B.1 Via executável de autoinstalação	145
B.2 Via código-fonte do programa	145
APÊNDICE C – Sugestões de melhorias do MWPVis	147
C.1 Funcionalidades de <i>software</i>	147
C.1.1 Gravar e recuperar estado da visualização	147
C.1.2 Manter endereços dos últimos estados de visualização	147
C.1.3 Alterar a resolução da imagem de tela	148
C.1.4 Alterar a resolução e a taxa de quadros do vídeo	148
C.1.5 Alterar as configurações para personalização do programa	148
C.1.6 Ver o manual de ajuda	148
C.1.7 Ocultar a janela de saída de erros gerada pelo VTK	149
C.2 Fluxo de visualização	149
C.2.1 Importação	149
C.2.2 Filtragem/enriquecimento	149
C.3 Técnicas de visualização	150
C.3.1 Mapeamento em cores	150
C.3.2 Contorno	150
C.3.3 Glifos de vetor e linhas de fluxo	150
C.3.4 Trajetória de partícula	151
C.3.5 Composição de técnicas	151
C.3.6 Gráficos de linhas	151
C.4 Técnicas de computação gráfica	152

C.4.1 Animação	152
C.4.2 Múltiplas visões	153
ANEXOS	154
ANEXO A – Questionário de Avaliação do MWPVis	154
ANEXO B – Manual do Usuário do MWPVis	154
ANEXO C – Formulário de Respostas do Questionário de Avaliação do MWPVis	154

1 INTRODUÇÃO

Esta dissertação de mestrado apresenta um trabalho científico-tecnológico na área de Visualização Científica, cujas técnicas foram aplicadas em Mecânica dos Meios Contínuos (MMC) para visualização dos fenômenos de propagação de ondas mecânicas, especificamente em meios materiais sólidos.

1.1 CONTEXTUALIZAÇÃO

Nas últimas décadas, o estudo dos fenômenos de propagação de ondas mecânicas e de suas consequências tem recebido crescente atenção de grupos de pesquisa da engenharia, especialmente daqueles que se dedicam ao desenvolvimento de métodos computacionais para resolução de problemas complexos em MMC. A principal motivação se dá por este estudo ser de fundamental importância para uma análise eficiente de problemas relacionados à interação dinâmica entre materiais rígidos e/ou deformáveis. Na engenharia mecânica, por exemplo, o estudo destes fenômenos contribui para o entendimento do comportamento de meios materiais sólidos quando excitados por cargas dinâmicas.

Os problemas afins não são exclusivos da engenharia mecânica, mas também de outros ramos da engenharia, como das engenharias de petróleo, de materiais e civil. Têm-se como exemplos de problemas práticos que se apoiam no estudo destes fenômenos: a busca pelo funcionamento eficiente de equipamentos industriais, o controle dos níveis de ruído e de vibração para um maior conforto ambiental, a estabilização mecânica do solo para suporte de equipamentos industriais ou laboratoriais etc. Contudo, em todos há um objetivo em comum, que é o de minimizar os efeitos danosos provocados pela propagação das ondas.

Os métodos computacionais produzidos pelos grupos de pesquisa atuantes neste tema frequentemente são fundamentados nos Métodos das Diferenças Finitas, dos Elementos Finitos e de Contorno, e consistem de modelagem matemática e simulação computacional de problemas físicos, que na maioria das vezes substituem ou reduzem a necessidade do uso de protótipos ou infraestrutura de laboratórios para ensaio dos mesmos. Entretanto, a simulação computacional produz uma volumosa quantidade de dados numéricos que pode dificultar ou inviabilizar uma análise

se mais eficiente dos resultados sem o suporte de técnicas específicas para mapeamento destes dados em representações visuais.

O procedimento convencional para resolução de problemas complexos em MMC, como para o estudo dos fenômenos de propagação de ondas mecânicas e de suas consequências, pode ser compreendido em três etapas: modelagem, simulação e análise (ver Figura 1.1). Em síntese, um problema complexo é matematicamente modelado e depois computacionalmente simulado. Posteriormente, o resultado da simulação é analisado pelo(s) especialista(s) com o objetivo de encontrar uma ou mais soluções para o problema inicial. Entretanto, este processo de análise se torna uma tarefa árdua e extremamente difícil, já que o resultado é representado por uma volumosa quantidade de dados numéricos.

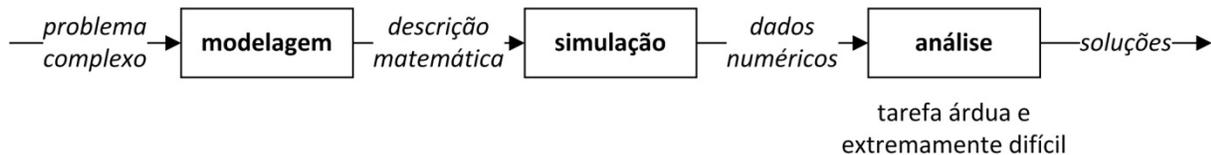


Figura 1.1 – Procedimento convencional para resolução de problemas complexos em MMC.

1.2 METODOLOGIA, LIMITAÇÕES E CONTRIBUIÇÕES

A Visualização é uma área da Computação dedicada ao estudo e desenvolvimento de técnicas para mapeamento de dados em representações visuais, com o objetivo de explorar as capacidades humanas de percepção e manipulação interativa para ampliação da cognição. Recentemente esta área é classificada em duas vertentes de pesquisa: a primeira e mais antiga é a Visualização Científica, que se dedica à visualização de dados científicos, frequentemente de origem natural; e a segunda e mais recente é a Visualização de Informação, que se dedica à visualização de dados abstratos, frequentemente de origem artificial (CARD; MACKINLAY; SHNEIDERMAN, 1999).

A Visualização Científica é uma ferramenta de apoio à ciência e engenharia que pode ser compreendida como sendo o uso de computadores ou técnicas para compreensão de dados ou obtenção de conhecimento a partir dos resultados de simulações, cálculos ou medições (McCORMICK; DE FANTI; BROWN, 1987). Na engenharia, ela possibilita oportunidades de pesquisa nos campos de Dinâmica de Fluidos Computacional, Análise de Elementos Finitos e afins

(DE FANTI; BROWN; MCCORMICK, 1989). Suas técnicas podem ser aplicadas, por exemplo, na simulação computacional de comportamentos para entendimento de fenômenos físicos (GALLAGHER, 1995).

A Visualização Científica também dispõe de um conjunto de técnicas fortemente consolidado para mapeamento de dados escalares, vetoriais, tensoriais ou volumétricos, tais como: mapeamento em cores, em conjunto de níveis, em função da altura, em glifos, entre outras; além de técnicas para modelagem e filtragem ou enriquecimento de conjunto de dados (SCHROEDER; MARTIN; LORENSE, 2006; TELEA, 2007).

Enfim, baseado nas afirmações anteriores citadas de forma sintética, ou seja, de que a Visualização Científica: (1) é capaz de ampliar a cognição, (2) serve de apoio ao campo de Análise de Elementos Finitos e afins, (3) pode ser empregada na simulação computacional de fenômenos físicos e (4) dispõe de um conjunto de técnicas fortemente consolidado para mapeamento de dados em representações visuais; então os conceitos desta área foram estudados e aplicados, de modo que o processo de visualização fosse incorporado ao procedimento convencional para resolução de problemas complexos em MMC (ver Figura 1.2).

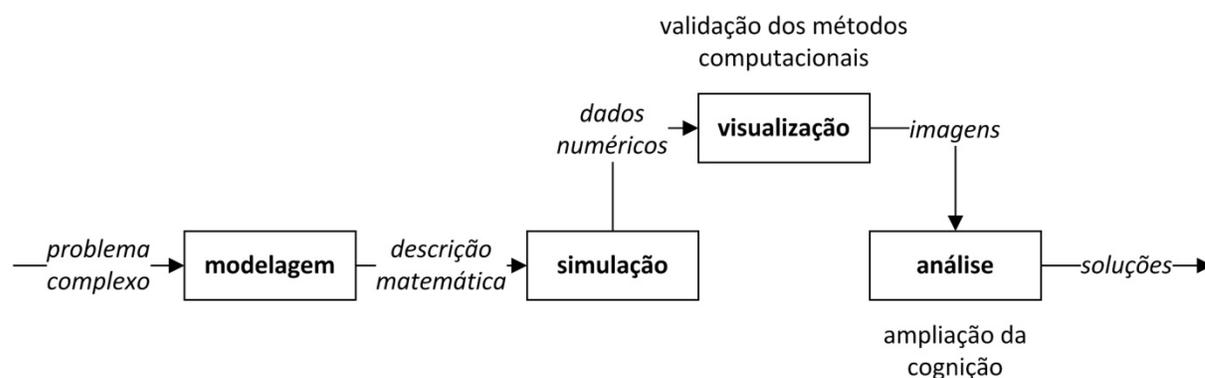


Figura 1.2 – Procedimento proposto para resolução de problemas complexos em MMC.

Esperava-se com a incorporação do processo de visualização que os especialistas pudessem validar mais rapidamente os seus métodos computacionais desenvolvidos, ou seja, as suas modelagens matemáticas e simulações computacionais de problemas direcionados para o estudo dos fenômenos de propagação de ondas mecânicas e de suas consequências. E principalmente, que eles pudessem compreender com mais facilidade os resultados gerados no processo de simulação computacional, o que aumentaria a eficiência da análise dos problemas complexos.

Vale ressaltar que este trabalho foi inicialmente concebido para a visualização dos fenômenos de propagação de ondas mecânicas bidimensionais (2D). Contudo, devido às sugestões de melhorias e demanda de contribuidores especialistas em Mecânica dos Sólidos, o trabalho foi posteriormente expandido para ondas mecânicas tridimensionais (3D). Ao final, este trabalho se limitou à visualização dos fenômenos de propagação de ondas mecânicas 2D ou 3D a partir de dados gerados sobre grades de dimensão 2D.

As contribuições deste trabalho podem ser resumidas em: (1) identificação de um conjunto de informações (dados escalares e vetoriais) derivadas dos dados brutos gerados pela simulação computacional que pode ser empregado para análise dos fenômenos de propagação de ondas mecânicas; (2) metodologia e respectiva formulação matemática para cálculo destas informações; (3) desenvolvimento de um protótipo de ambiente de visualização específico para dados que descrevem o comportamento de ondas mecânicas; e (4) estudo e definição de diretrizes de técnicas de visualização de dados escalares e vetoriais para visualização do conjunto de informações.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está organizada em sete capítulos além deste capítulo preliminar. Descrições sucintas destes capítulos são apresentadas a seguir:

Capítulo 2 – Visualização Científica

Neste capítulo são apresentados os fundamentos da área de Visualização Científica, cujos conceitos e técnicas foram aplicados neste trabalho para a visualização dos dados descritos no Capítulo 3.

Capítulo 3 – Simulação e Análise de Ondas Mecânicas

Neste capítulo são apresentados os conceitos teóricos essenciais para a análise dos estudos de caso apresentados no Capítulo 6, assim como o processo de simulação computacional de propagação de ondas mecânicas em meios contínuos sólidos e os dados gerados neste processo.

Capítulo 4 – Visualização de Ondas Mecânicas

Neste capítulo é apresentado o fluxo de visualização proposto para a visualização dos fenômenos de propagação de ondas mecânicas.

Capítulo 5 – Ambiente de Visualização de Ondas Mecânicas

Neste capítulo é apresentado o ambiente de visualização desenvolvido, no qual o fluxo de visualização proposto é implementado.

Capítulo 6 – Estudos de Caso

Neste capítulo são apresentados dois problemas geomecânicos para confirmação dos objetivos deste trabalho e validação do ambiente de visualização.

Capítulo 7 – Discussão e Avaliação das Visualizações

Neste capítulo é apresentada a avaliação das visualizações geradas pelo ambiente de visualização, assim como as informações que podem ser obtidas por meio destas visualizações.

Capítulo 8 – Conclusões

Neste capítulo são apresentadas as conclusões do trabalho desenvolvido, além das suas contribuições, limitações e sugestões de trabalhos futuros.

Além dos capítulos, três apêndices e três anexos também compõem a dissertação. Estes estão relacionados ao ambiente de visualização desenvolvido e descrito no Capítulo 5: o Visualizador de Propagação de Ondas Mecânicas (MWPVis – *Mechanical Wave Propagation Visualizer*).

Apêndice A – Configuração do Ambiente de Desenvolvimento do MWPVis

Este apêndice relata todos os procedimentos realizados para configurar o ambiente de desenvolvimento utilizado para a produção do MWPVis. Tem por objetivo ajudar os futuros desenvolvedores na preparação do ambiente para uso do código-fonte deste programa.

Apêndice B – Instalação do MWPVis

Este apêndice descreve os procedimentos para instalação do MWPVis a partir do executável de autoinstalação ou do código-fonte do mesmo.

Apêndice C – Sugestões de Melhorias do MWPVis

Este apêndice exibe um conjunto de sugestões de melhorias do MWPVis que pode ser implementado pelos futuros desenvolvedores.

Anexo A – Questionário de Avaliação do MWPVis

Este anexo contém o “Questionário de Avaliação do MWPVis” usado para a validação do programa desenvolvido.

Anexo B – Manual do Usuário do MWPVis

Este anexo contém o “Manual do Usuário do MWPVis” da versão 0.4.14 do programa.

Anexo C – Formulário de Respostas do Questionário de Avaliação do MWPVis

Este anexo contém o “Formulário de Respostas do Questionário de Avaliação do MWPVis”, no qual as respostas dos avaliadores foram registradas.

2 VISUALIZAÇÃO CIENTÍFICA

Neste capítulo são apresentados os fundamentos da área de visualização científica, cujas técnicas foram empregadas neste trabalho para a visualização dos dados produzidos por simulação computacional dos fenômenos de propagação de ondas mecânicas, apresentado no capítulo seguinte. Nas suas seções são apresentadas: definições da área de visualização de dados, com ênfase em visualização científica (Seção 2.1); técnicas para representação de conjunto de dados (Seção 2.2); fluxo de visualização científica convencional (Seção 2.3); técnicas para visualização de dados escalares e vetoriais (Seções 2.4 e 2.5); técnicas para modelagem de conjunto de dados (Seção 2.6); e técnicas de computação gráfica (Seção 2.7).

2.1 INTRODUÇÃO

O provérbio “uma imagem vale mais do que mil palavras”¹ ilustra perfeitamente como as imagens sintetizadas podem ser empregadas para a transmissão de conhecimento ou até mesmo para a produção do mesmo por meio da introspecção (CARD; MACKINLAY; SHNEIDERMAN, 1999; WARE, 2004). Ao longo da evolução humana, diversas abstrações visuais foram criadas para auxiliar o pensamento. Pinturas rupestres, mapas geográficos ou náuticos e diagramas matemáticos são exemplos de tais abstrações (WARE, 2004; WRIGHT, 2007). Estes exemplos mostram que a “transformação de conceitos abstratos em imagens reais ou mentalmente visíveis” – como é definida a palavra “visualização” no dicionário Michaelis (WEISZFLOG, 1998) –, de maneira geral, não é um fato historicamente recente.

Entretanto, a “visualização” a serviço da ciência ou da engenharia já pode ser considerada um fato moderno, tendo sido impulsionada pelo advento e uso do computador e de suas tecnologias, como principalmente a computação gráfica. A origem da visualização no campo da computação normalmente é atribuída ao trabalho intitulado de “*Visualization in Scientific Computing*” publicado pela Fundação Nacional da Ciência (NSF – *National Science Foundation*) estadunidense (MCCORMICK; DEFANTI; BROWN, 1987). A partir de então, diversas denominações

¹ De acordo com Mieder (2004), este provérbio foi originado do anúncio publicitário intitulado “*One Look is Worth a Thousand Words*”, criado por Fred R. Barnard e publicado no jornal estadunidense *Printer's Ink* em 8 de dezembro de 1921.

foram estabelecidas para visualização na computação, dentre as quais se destacam: **visualização de dados**, **visualização científica** e **visualização de informação** (SCHROEDER; MARTIN; LORENSEN, 2006). Por não ser foco deste trabalho, esta última não é tratada neste texto².

A **visualização de dados**, geralmente denominada apenas de visualização, busca explorar as capacidades humanas de percepção e cognição. Isso por que o sistema visual humano é um reconhecedor de padrões com extremo poder e acuidade, no qual os olhos e o córtex visual do cérebro formam uma poderosa central de processamento paralelo que possui um canal de comunicação direto e muito veloz com os processos cognitivos. Além do que o ser humano adquire mais informações por intermédio da visão do que pelos demais sentidos juntos, já que cerca de 40% do cérebro (mais de 20 bilhões de neurônios) são dedicados ao processamento visual, que é fundamental para a análise cognitiva (WARE, 2004).

Ware (2004) destaca algumas vantagens da visualização: supre a capacidade de entendimento de enormes quantidades de dados, já que as informações podem ser mapeadas dentro dos limites da imagem; possibilita a percepção de propriedades emergentes que não foram previstas, como detalhes na representação dos dados; problemas frequentes com os dados se tornam imediatamente perceptíveis, o que praticamente não seria possível por uma análise direta dos dados brutos; facilita o entendimento de aspectos de grande ou pequena proporção dos dados, já que é possível ter uma visão detalhada ou geral dos mesmos; e facilita a formulação de hipóteses, visando à resolução de problemas.

A **visualização científica** é uma ramificação da visualização de dados que se dedica à pesquisa e desenvolvimento de técnicas para o mapeamento de dados científicos (ou concretos) – normalmente de origem natural – em representações visuais interativas, com o objetivo de ampliar a cognição (CARD; MACKINLAY; SHNEIDERMAN, 1999). Apesar de tanto a computação gráfica quanto a visualização produzirem imagens, a primeira é apenas uma tecnologia usada pela segunda. Ou seja, a visualização é um processo de transformação de dados em imagens, no qual estas são produzidas com o uso das técnicas de síntese de imagens (ou “renderização”³) da computação gráfica. Outras tecnologias também podem ser agregadas à visualização, como processamento de imagens, visão computacional, desenho auxiliado por computador, processamento de sinais e metodologias de interfaces de usuário (MCCORMICK; DE FANTI; BROWN, 1987).

² Na Seção 1.2 consta uma prévia sobre visualização de informação, assim como uma referência bibliográfica para tal tema.

³ Anglicismo de *rendering* e seu aportuguesamento não oficial na ciência computacional.

A visualização científica pode ser empregada em diversas linhas de pesquisa das áreas da ciência, como modelagem molecular, imagens médicas, estrutura e função cerebral, matemática, geociências (meteorologia), exploração espacial e astrofísica; e da engenharia, como dinâmica de fluídos computacional e análise de elementos finitos (DE FANTI; BROWN; MCCORMICK, 1989). Na análise de elementos finitos, por exemplo, as técnicas de visualização podem ser aplicadas para que os especialistas possam estudar o comportamento de fenômenos naturais simulados por computador (GALLAGHER, 1995).

Enfim, Schroeder et. al (2006) e Telea (2007) apresentam um conjunto de técnicas fortemente consolidado da visualização científica, que pode ser empregado para a análise e exploração de dados produzidos por simulação computacional de fenômenos naturais, tais como os fenômenos de propagação de ondas mecânicas. Por isso, uma descrição sucinta dos conceitos e das técnicas que foram usadas neste trabalho é apresentada nas seções a seguir. Os conteúdos destas seções são apresentados com base em uma ou ambas as referências citadas.

2.2 REPRESENTAÇÃO DE DADOS

A visualização possui dois processos básicos: **transformação** e **representação**. A **transformação** é o processo de conversão dos dados até a forma de primitivas gráficas, que compõem as imagens a serem renderizadas. A **representação** abrange tanto a estrutura de dados usada para organizar os dados (formato computacional), quanto às primitivas gráficas usadas para apresentá-los (formato gráfico). Em outras palavras, pode-se definir visualização como um processo de transformação do formato computacional ao gráfico. Nas subseções que se seguem são apresentados a definição de conjunto de dados discretos e os tipos básicos de células, grades e atributos de dados que são usados para representação de dados.

2.2.1 Conjunto de dados discretos

A visualização científica atua na representação de dados de origem contínua, que inclusive, podem ser matematicamente modelados por funções. Contudo, as várias operações sobre estes dados, como filtragem, enriquecimento e renderização, não são eficientes e nem fáceis de serem

realizadas no âmbito computacional. Por isso, praticamente todos os métodos de visualização são aplicados na representação de dados discretos, amostrados de um domínio contínuo. Mas como a origem dos dados é contínua, então eles precisam ser posteriormente reconstruídos. Esta operação é realizada pela técnica de interpolação, que é muito explorada na representação de dados.

Os dados estruturalmente organizados e associados às suas informações (atributos de dados) formam os chamados **conjuntos de dados**. A estrutura de um conjunto de dados é composta segundo dois aspectos: **topologia** e **geometria**. A **topologia** pode ser entendida de maneira simplificada como sendo o conjunto de propriedades invariantes às transformações geométricas, tais como rotação, translação e mudança de escala. A **geometria** especifica a posição de algo no espaço. Uma superfície plana, por exemplo, possui topologia e geometria 2D. Contudo, uma superfície curva, possui topologia 2D e geometria 3D.

Para a representação de um conjunto de dados, quatro conceitos são importantes: **pontos**, **células**, **grades** e **atributos de dados**. Os **pontos** correspondem aos pontos de amostragem dos dados contínuos e especificam a geometria. As **células** são as bases para as operações de interpolação usadas para a reconstrução do domínio contínuo e especificam a topologia (Seção 2.2.2). As **grades**, também conhecidas como malhas, são compostas por conjuntos de células e representam o meio contínuo (Seção 2.2.3). E os **atributos de dados** são as informações amostradas associadas aos pontos (Seção 2.2.4).

2.2.2 Tipos de células

Uma **grade** é composta por um agrupamento de células, em que os vértices são os pontos desta grade. Uma vez que os dados são amostrados a partir destes pontos, então as **células** consistem nas funções usadas para interpolar os dados entre estes pontos. Ou seja, os pontos de amostragem são conectados para formar células, que por sua vez limitam o domínio a ser reconstruído. Dentre os tipos de células existentes, seis podem ser destacados: **vértice**, **linha**, **triângulo**, **quadrilátero**, **tetraedro** e **hexaedro** (ver Figura 2.1).

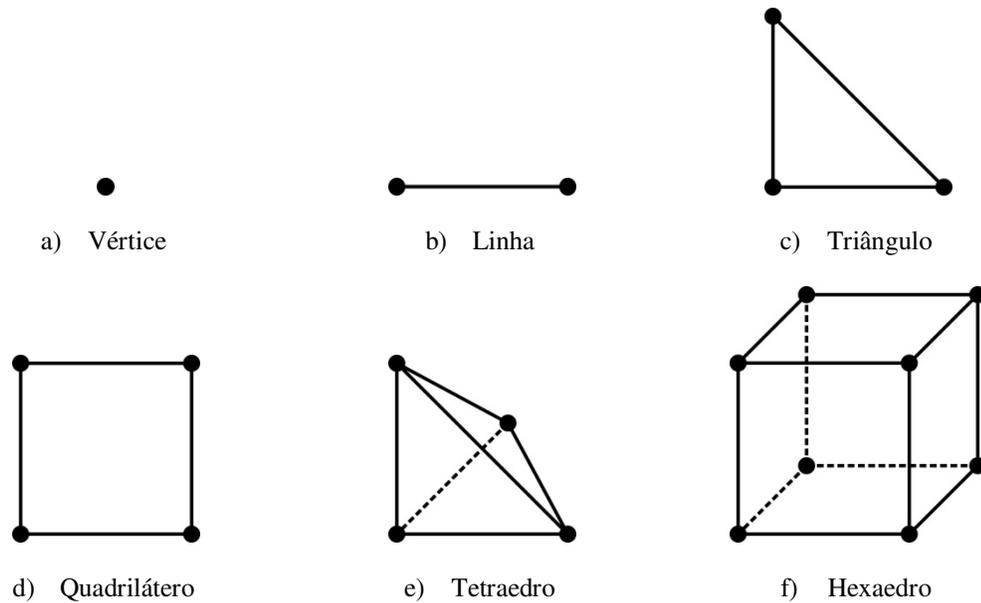


Figura 2.1 – Representação dos seis principais tipos de células (baseado em Schroeder et al., 2006).

2.2.2.1 Vértice

O **vértice** é uma célula adimensional que pode ser definida por apenas um ponto. Justamente por isso, os vértices são usados para representar os pontos de amostragem.

2.2.2.2 Linha

A **linha** é uma célula unidimensional (1D) e pode ser definida a partir de dois pontos. Os demais pontos de uma linha podem ser encontrados por meio de interpolação linear entre suas extremidades.

2.2.2.3 Triângulo

O **triângulo** é uma célula 2D que pode ser definida por três pontos. O sentido da normal da superfície desta célula é dependente da ordem de seus pontos (regra da mão direita). Os pontos

contidos na superfície desta célula podem ser encontrados por meio da interpolação linear entre seus três extremos.

2.2.2.4 Quadrilátero

O **quadrilátero** é uma célula 2D e pode ser definida por quatro pontos. Os pontos contidos na superfície desta célula podem ser encontrados por interpolação bilinear.

2.2.2.5 Tetraedro

O **tetraedro** é uma célula 3D que pode ser definida por quatro pontos não coplanares. Os pontos internos ao seu volume podem ser encontrados pela interpolação linear entre seus extremos.

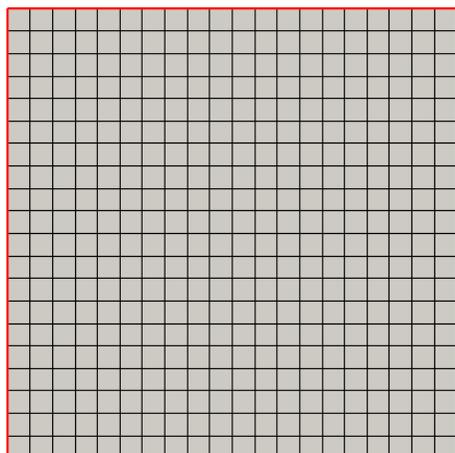
2.2.2.6 Hexaedro

O **hexaedro** é uma célula 3D que pode ser definida por oito vértices. Os pontos internos ao seu volume podem ser encontrados por interpolação trilinear.

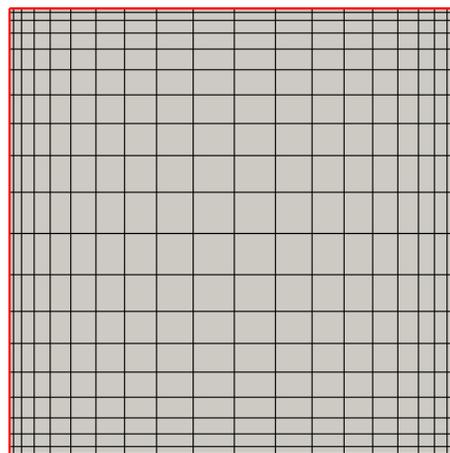
2.2.3 Tipos de grades

Os conjuntos de dados são caracterizados segundo suas estruturas topológicas e geométricas em **regulares** ou **irregulares**. Os pontos e/ou os relacionamentos entre as células que podem ser representados implicitamente são denominados de **regulares**. Quando estes podem ser representados apenas explicitamente, então são chamados de **irregulares**. Note que quando os pontos são regulares, então a geometria do conjunto de dados também é regular. E quando os relacionamentos entre as células são regulares, então a topologia do conjunto de dados também é regular.

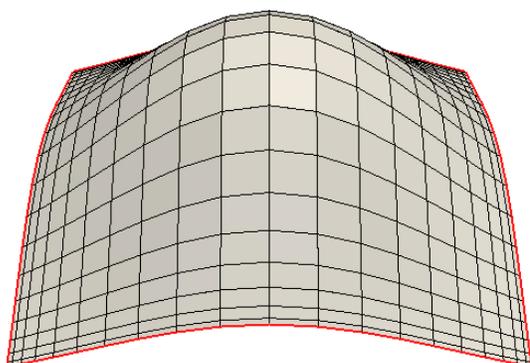
A partir das células apresentadas anteriormente é possível construir vários tipos de grades. Devido à grande quantidade de tipos de grades existentes, somente as mais usadas serão descritas: **grades uniformes**, **grades retilíneas**, **grades estruturadas** e **grades não-estruturadas**. A Figura 2.2 ilustra estes quatro tipos de grades.



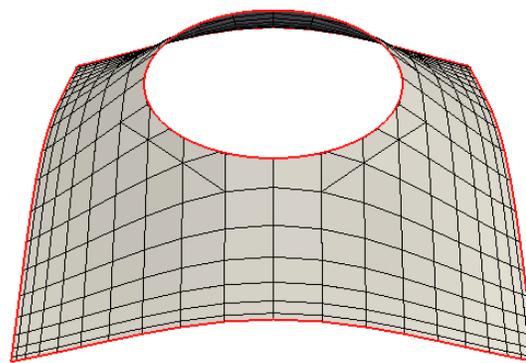
a) **Grade uniforme** de dimensões $21 \times 21 \times 1$.



b) **Grade retilínea** de dimensões igual ao da grade em a), mas com espaçamento variado das amostras ao longo dos eixos principais.



c) **Grade estruturada** gerada a partir de uma deformação da grade em b).



d) **Grade não-estruturada** gerada a partir de um recorte (Seção 2.6.2) aplicado sobre a grade em c) e uma esfera.

Figura 2.2 – Exemplos dos quatro principais tipos de grades (baseado em Telea, 2007).

2.2.3.1 Grades uniformes

A **grade uniforme** é o tipo mais simples de grade, cujas amostras são igualmente espaçadas ao longo dos eixos do domínio. Uma imagem é um exemplo de grade uniforme, no qual as células correspondem aos *pixels* ou *voxels*. Para a definição de uma grade uniforme são necessários apenas três parâmetros: dimensões da grade, posição da amostra de origem e espaçamento entre as amostras. As dimensões da grade correspondem ao número de amostras sobre cada um dos eixos de coordenadas. A posição da amostra de origem é a localização do referencial para o início da amostragem. E o espaçamento entre as amostras é a distância entre os pontos de amostragem sobre os eixos de coordenadas. Estas grades possuem topologia e geometria regulares, já que podem ser representadas implicitamente.

2.2.3.2 Grades retilíneas

Apesar de serem simples de implementar e computacionalmente eficientes, já que requerem pouquíssimo espaço para armazenamento, as grades uniformes têm limitado poder de modelagem. Primeiro, para a representação de domínios não-alinhados com os eixos de coordenadas, há necessidade da concepção de caixa delimitadora⁴ para que este domínio esteja contido em outro alinhado com os eixos. Com isso, ocorre desperdício de memória, uma vez que é realizada amostragem fora do domínio original. Segundo, para a representação precisa de uma função com taxa de variação não-uniforme é preciso usar uma alta densidade de amostragem sobre a grade uniforme ou usar uma grade com densidade de amostras não-uniformes. As grades retilíneas são a primeira tentativa de solução destas limitações.

Assim como nas grades uniformes, os domínios das grades retilíneas devem ser paralelos aos eixos de coordenadas globais, mas os espaçamentos entre as amostras podem ser variados. As **grades retilíneas** possuem topologia regular e geometria apenas parcialmente regular, já que suas células podem ser representadas implicitamente e seus pontos podem ser representados tanto de implicitamente quanto explicitamente.

⁴ Expressão traduzida do inglês *bounding box*.

2.2.3.3 Grades estruturadas

Apesar de serem úteis, as grades retilíneas não eliminam as restrições inerentes às grades uniformes. O domínio das amostras continua sendo paralelo aos eixos globais e não é possível ampliar a densidade dos pontos de amostragem sobre uma região específica da grade. Para isto, é preciso permitir que a amostragem seja livre. As grades estruturadas servem para este propósito.

As **grades estruturadas** têm topologia regular e geometria irregular. A topologia pode ser representada implicitamente pela especificação das suas dimensões. E a geometria pode ser representada explicitamente por um conjunto de pontos de localização independente. Intuitivamente, uma grade estruturada pode ser vista como a deformação livre de uma grade uniforme ou retilínea, em que os pontos podem assumir quaisquer posições espaciais, mas as células, ou seja, a topologia da grade, permanecem as mesmas.

2.2.3.4 Grades não-estruturadas

Existem formas que não podem ser eficientemente modeladas até mesmo pelas grades estruturadas. Por exemplo, um domínio retangular com um buraco circular no centro. Isto por que a topologia deste domínio é irregular, já que ocorre descontinuidade no mesmo. Outra limitação diz respeito à especificação das células. Para as grades uniformes, retilíneas e estruturadas, as células são especificadas implicitamente. Estes problemas são solucionados por intermédio das grades não-estruturadas.

A **grade não-estruturada** é o tipo mais genérico e flexível de grade. Nesta grade, tanto os pontos de amostragem quanto as células podem ser definidas explicitamente. Apesar dos benefícios, as grades não-estruturadas são utilizadas apenas em situações realmente necessárias. Isto por que este tipo de grade necessita de bastante espaço para armazenamento, além de muita memória e recursos computacionais para sua representação.

2.2.4 Tipos de atributos de dados

Os **atributos de dados** são as informações associadas aos dados. Estes atributos geralmente estão associados aos pontos de amostragem. Entretanto, em casos particulares, estes atributos também podem estar associados às células ou até mesmo às suas componentes, como arestas e faces. Os atributos de dados podem ser classificados quanto ao número de dimensões ou pela semântica dos dados que eles representam. Justamente por esta última, há diversos tipos de atributos de dados: **escalar**, **vetorial**, **tensorial**, de cor, de textura, não-numérico etc.

2.2.4.1 Atributos escalares

Os **atributos escalares** são 1D e permitem representar as grandezas escalares, tal como temperatura, pressão, concentração, densidade, elevação etc. Os atributos escalares são as formas mais simples e comuns de visualização de dados.

2.2.4.2 Atributos vetoriais

Os **atributos vetoriais** são 2D ou 3D e possibilitam representar as grandezas vetoriais, tal como velocidade, direção, força, gradiente de funções escalares etc. Os vetores possuem magnitude, direção e sentido. Quando apenas a direção é importante, um vetor pode ser reduzido para a sua forma normalizada, cuja sua magnitude é unitária. Quando estes vetores normalizados são perpendiculares a superfície, por exemplo, este é chamado de normal. Alguns autores, como Schroeder et al. (2006), consideram vetores e normais como sendo dois tipos diferentes de atributos de dados. Entretanto, na prática não há diferença estrutural entre ambos. É interessante observar que quando apenas a magnitude é relevante, um vetor pode ser reduzido para a forma de um escalar.

2.2.4.3 Atributos tensoriais

Os **atributos tensoriais** são multidimensionais (ND) e referem às grandezas tensoriais, tal como tensão e deformação. Os tensores são generalizações matemáticas complexas de vetores e matrizes. Um tensor de ordem zero é um escalar, de primeira ordem é um vetor, de segunda ordem é uma matriz e de terceira ordem é um conjunto retangular 3D. Os tensores de ordem elevada são conjuntos retangulares ND. Enfim, um tensor de ordem k pode ser considerado como uma tabela k -dimensional.

2.3 FLUXO DE VISUALIZAÇÃO

O processo de visualização consiste em uma sequência de etapas (ou operações) de transformação que são aplicadas sobre os dados para a formação de uma ou mais imagens. Esta sequência de etapas é chamada de **fluxo de visualização**⁵, no qual cada etapa é modelada para uma operação específica de transformação dos dados. Um fluxo de visualização convencional possui quatro estágios básicos: **importação**, **filtragem**, **mapeamento** e **renderização**. Estes estágios são ilustrados na Figura 2.3 e descritos nas seções subsequentes.

2.3.1 Importação

O primeiro estágio do fluxo de visualização é chamado de **importação**. Este é um processo de aquisição dos dados, no qual os dados brutos (dados de entrada) – obtidos por intermédio de dispositivos de medição (como vibrômetro, acelerômetro, frequencímetro ou tacômetro) ou simulações com o uso de métodos numéricos (como os métodos das diferenças finitas, dos elementos finitos ou de contorno) – são convertidos para um formato adequado para o processamento no sistema de visualização. Ou seja, para a forma de representação considerada mais apropriada para os dados tendo em vista os vários tipos de células, grades e atributos de dados.

⁵ Expressão traduzida do inglês *visualization pipeline*.

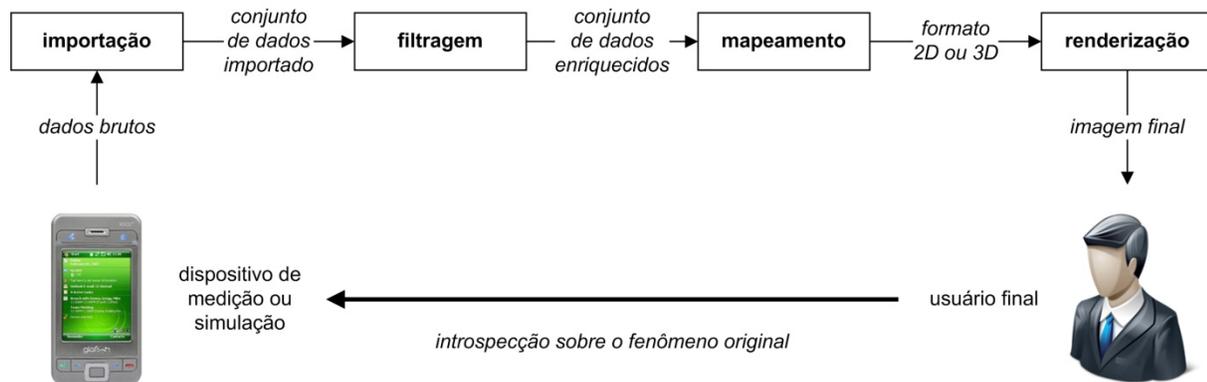


Figura 2.3 – Fluxo de visualização convencional (baseado em Telea, 2007).

2.3.2 Filtragem

A **filtragem**, ou **enriquecimento** dependendo do contexto, é a segunda etapa do fluxo de visualização. É neste estágio que são realizadas, por exemplo, as operações de subamostragem, de eliminação de informações desnecessárias, de redução do volume de dados, de adequação dos dados ao contexto ou aos algoritmos da aplicação. O principal objetivo da filtragem é eliminar os dados que não serão visualizados logo no início do fluxo para reduzir o tempo gasto com processamento.

2.3.3 Mapeamento

O penúltimo estágio do fluxo de visualização é chamado de **mapeamento**. Nesta etapa a estrutura organizacional para representação do conjunto de dados e seus atributos de dados são mapeados em representações visuais a partir de atributos gráficos. Em síntese, um conjunto de dados em seu formato computacional é transformado para o formato gráfico (ver Seção 2.2).

2.3.4 Renderização

A **renderização** é o último estágio do fluxo de visualização. Neste processo são realizadas basicamente as operações de aplicação das técnicas de computação gráfica, tais como trans-

formações de coordenadas, iluminação, mapeamento de textura e conversão por varredura (ou “rasterização”⁶).

2.4 VISUALIZAÇÃO DE DADOS ESCALARES

Nesta seção são apresentadas as principais técnicas de visualização de dados escalares: **mapeamento em cores** (Seção 2.4.1); **contorno** (Seção 2.4.2); e **função de altura** (Seção 2.4.3).

2.4.1 Mapeamento em cores

O mapeamento em cores é provavelmente o método mais comum de visualização de dados escalares. Em síntese, ele associa para cada valor escalar, uma cor correspondente. Neste método não há preocupação com a forma da grade, mas sim com a coloração dela. O **mapeamento em cores** pode ser considerado como uma função na qual a entrada é um valor escalar e a saída é uma cor. Ela permite que os valores escalares sejam identificados por cores. Existem várias maneiras de definir esta função, porém duas se destacam pelas suas simplicidades: **tabela de cores** e **função de transferência**.

A **tabela de cores** é uma função discreta, no qual para cada valor escalar discreto de entrada, há uma cor de saída correspondente pré-determinada. A **função de transferência** é uma função contínua, no qual as cores são geradas a partir de valores escalares contínuos. O mapeamento em cores apresentado na Figura 2.5a exemplifica a utilização de uma função de transferência denominada de “arco-íris”. De maneira simplificada, a tabela de cores pode ser considerada como uma discretização da função de transferência.

Existem dois métodos para realizar o mapeamento em cores: **método baseado em textura** e **método baseado em vértice**. No primeiro, a interpolação precede o mapeamento. Com isso, a interpolação é realizada sobre os valores escalares, o que faz com que os valores interpolados entre dois pontos sejam devidamente apresentados. Já no segundo, ocorre o oposto, a interpola-

⁶ Anglicismo de *rastering* e seu aportuguesamento não oficial na ciência computacional.

ção sucede o mapeamento. Com isso, a interpolação é realizada sobre as cores, o que pode provocar incoerências na apresentação destas regiões.

A Figura 2.4 ilustra ambos os métodos. Embora o método baseado em textura seja mais confiável no mapeamento dos dados, o método baseado em vértice é mais simples de ser implementado e exige menos recursos gráficos. Na prática, o segundo método ignora cores da tabela de cores quando os dados escalares variam rapidamente sobre o domínio do conjunto de dados, enquanto o primeiro método não.

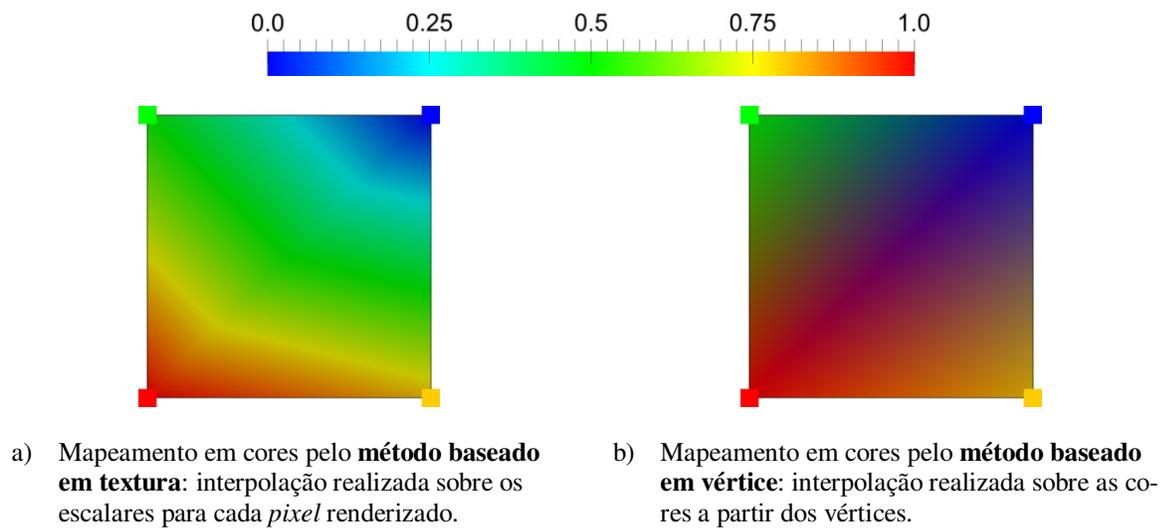
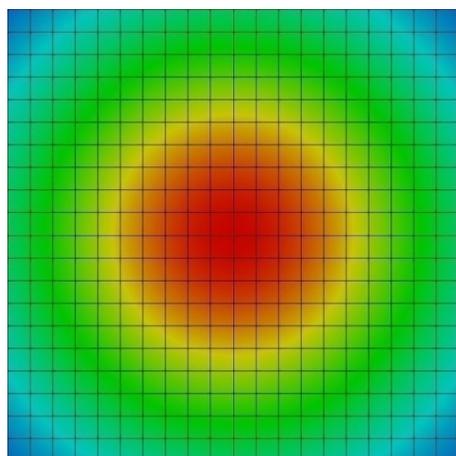
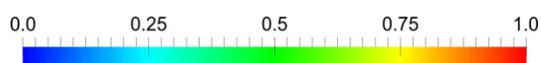


Figura 2.4 – Mapeamento em cores pelos métodos baseados em textura e vértice.

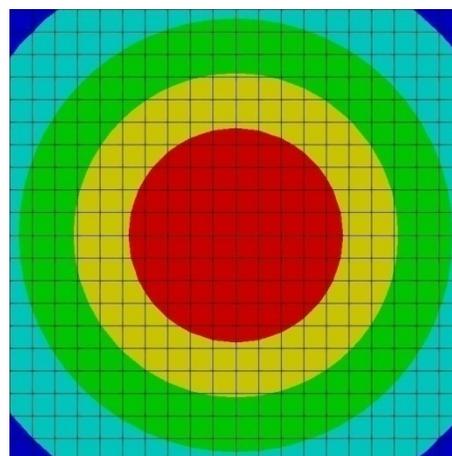
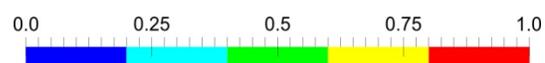
2.4.2 Contorno

Quando a técnica de mapeamento em cores é realizada com um número reduzido de cores, pode ocorrer o efeito denominado de **faixa de cores**. Por este efeito é possível identificar as regiões de fronteira entre intervalos de valores escalares, já que ocorre uma transição brusca entre as cores, tal como apresentado na Figura 2.5b. A técnica de **contorno** busca este mesmo efeito, só que sem o uso de cores. Quando esta técnica é realizada sobre uma superfície, ela é chamada de **isolinhas** (ver Figura 2.5c). E quando ela é realizada sobre um volume, ela é chamada de **isosuperfícies**. Ambas podem ser obtidas, por exemplo, pelos algoritmos de *marching squares* e *marching cubes*⁷, respectivamente.

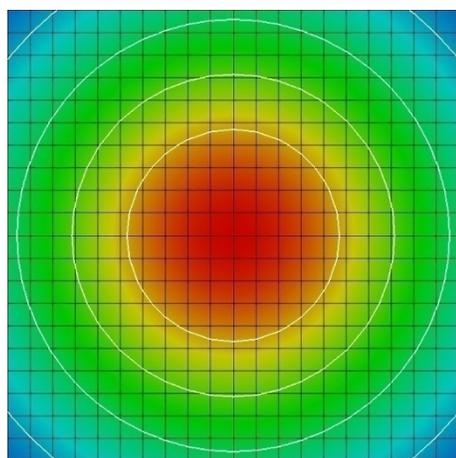
⁷ As definições de ambos os algoritmos são encontradas em Telea (2007).



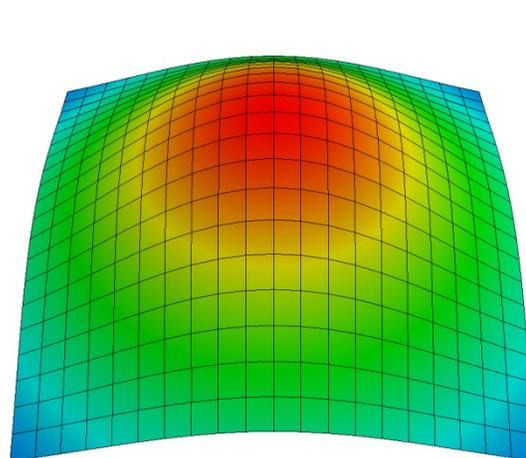
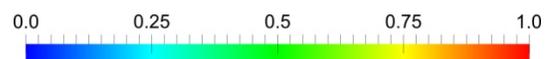
- a) **Mapeamento em cores** realizado por uma função de transferência “arco-íris” (cores variando do azul ao vermelho no modelo de cores HSV) e pelo método baseado em textura.



- b) **Mapeamento em cores** realizado por uma tabela de cores com apenas 5 cores, que foram obtidas a partir da função de transferência “arco-íris” aplicada em a) e que produz o efeito chamado “faixa de cores”.



- c) **Contorno** por isolinhas (em branco) de 5 intervalos de escalares sobrepostos sobre a grade apresentada em a).



- d) **Função de altura** com vetor direcional igual à normal da superfície da grade apresentada em a).

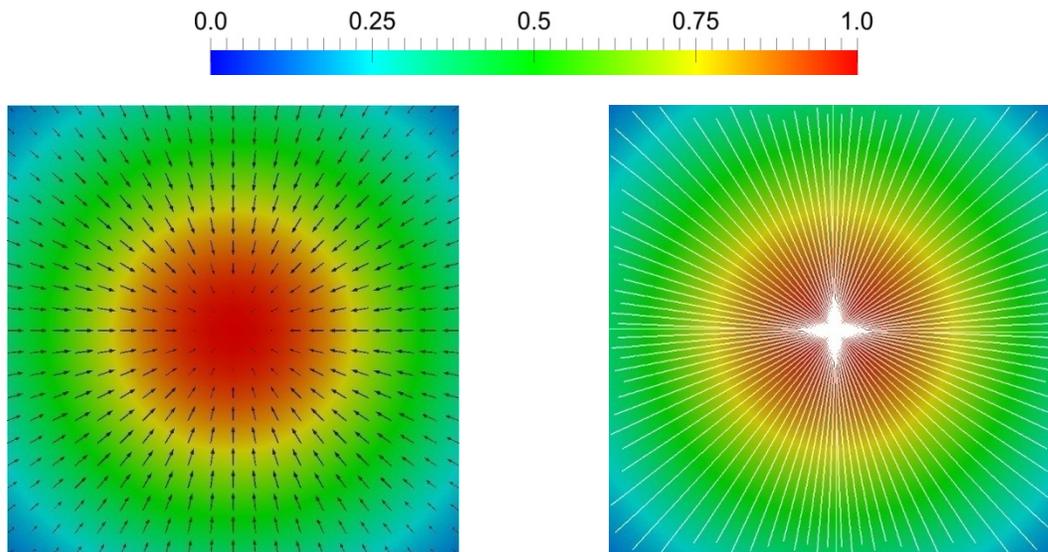
Figura 2.5 – Exemplos de uso das três principais técnicas de visualização de dados escalares.

2.4.3 Função de altura

A **função de altura**, também chamado de função de elevação, é uma técnica que provoca uma deformação na grade em função do valor dos dados escalares. Os vértices da grade sofrem um deslocamento ao longo de um vetor direcional – em geral, a normal da superfície –, cuja magnitude do deslocamento é dada pelo valor do atributo escalar a ser visualizado. A Figura 2.5d ilustra esta técnica.

2.5 VISUALIZAÇÃO DE DADOS VETORIAIS

Nesta seção são apresentadas as duas principais técnicas de visualização de dados vetoriais: **glifos de vetor** (Seção 2.5.1); e **linhas de fluxo** (Seção 2.5.2).



a) **Glifos de vetor** do gradiente ascendente dos escalares mapeados por cores.

b) **Linhas de fluxo** do campo vetorial representado em a).

Figura 2.6 – Exemplos de uso das duas principais técnicas de visualização de dados vetoriais.

2.5.1 Glifos de vetor

Os **glifos de vetor** são provavelmente a técnica mais comum para visualização de campos vetoriais. Esta técnica consiste em representar os atributos vetoriais por ícones, como linhas, setas, triângulos, tetraedros etc. Os glifos de vetor são representações gráficas que transmitem pela aparência as propriedades dos vetores, como direção, orientação e magnitude. A Figura 2.6a ilustra a representação do gradiente ascendente de um conjunto de escalares.

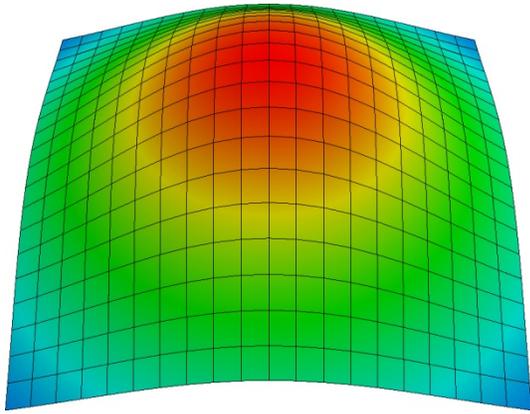
2.5.2 Linhas de fluxo

Para o estudo do fluxo de um campo vetorial representado por glifos de vetor é requisitada à habilidade de interpolação visual (ou mental) dos mesmos. Em situações nas quais as variações de direção e/ou sentido destes vetores são bruscas entre pontos vizinhos, a interpolação se torna difícil. Uma solução é a técnica de **linhas de fluxo**, que equivale ao “prolongamento” dos glifos de vetor por interpolação. Quando este conceito é empregado sobre a variação da posição dos pontos em função do tempo, ao invés dos atributos de dados, esta técnica é denominada **trajetória de partículas**. A Figura 2.6b ilustra as linhas de fluxo do campo vetorial apresentado na Figura 2.6a.

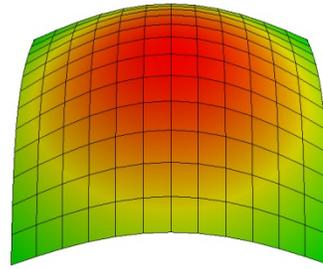
2.6 TÉCNICAS DE MODELAGEM

As técnicas de modelagem são o alicerce da maioria das técnicas de visualização. Todas possuem algo em comum: criam ou modificam a topologia e/ou a geometria dos conjuntos de dados. Nesta seção são descritas três técnicas de modelagem, sendo a primeira explícita e as outras implícitas: **extração de subgrade** (Seção 2.6.1), **recorte** (Seção 2.6.2) e “**fatiamento**”⁸ (Seção 2.6.3).

⁸ Anglicismo de *slicing* e seu aportuguesamento não oficial na ciência computacional.

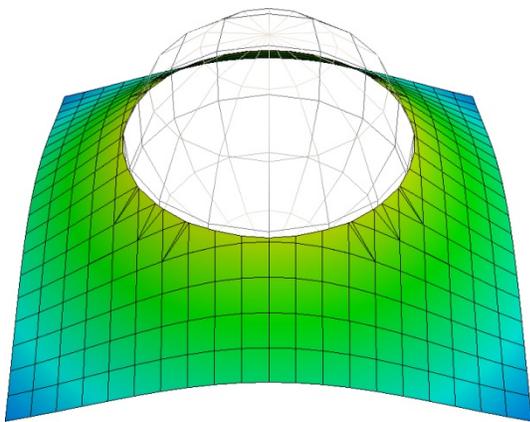
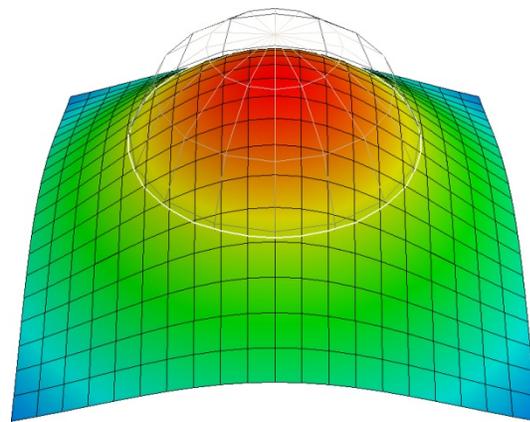


a) Uma grade estruturada qualquer.

b) Extração de uma região de interesse da grade pela técnica de **extração de subgrade**.**Figura 2.7** – Exemplo de uso da técnica de extração de subgrade.

2.6.1 Extração de subgrade

A técnica de **extração de subgrade** (ou subconjunto) consiste na extração de uma região de interesse de uma grade a partir de um conjunto de vértices vizinhos. A Figura 2.7 ilustra uma aplicação desta técnica para a extração de uma região de interesse localizada no centro da grade.

a) Resultado do **recorte** de uma grade com o volume de uma esfera localizada no centro da mesma (cavidade circular).b) Resultado do **fatiamento** de uma grade com a superfície de uma esfera localizada no centro da mesma (circunferência em branco).**Figura 2.8** – Exemplos de uso das duas principais técnicas de modelagem implícita.

2.6.2 Recorte

A técnica de **recorte** consiste na extração de um subconjunto de dados de uma grade, que representa um conjunto de dados. Esta extração é realizada pela operação de subtração entre uma grade, que representa um conjunto de dados, e uma figura geométrica descrita de forma implícita, como um plano, um paralelepípedo ou uma esfera. A Figura 2.8a ilustra o resultado de um recorte.

2.6.3 Fatiamento

A técnica de **fatiamento** é similar ao recorte. Com uma exceção, ao invés da operação ser uma subtração é uma intersecção. Com isso, o subconjunto de dados da grade extraído são os pontos que intersectam a grade e a superfície da figura geométrica. A Figura 2.8b ilustra o resultado de um fatiamento.

2.7 TÉCNICAS DE COMPUTAÇÃO GRÁFICA

Nesta seção são apresentadas algumas das principais técnicas de computação gráfica que são utilizadas na visualização de dados: **animação** (Seção 2.7.1); **interação de usuário** (Seção 2.7.2); e **renderização estereoscópica** (Seção 2.7.3).

2.7.1 Animação

A **animação** pode ser definida como o efeito causado pela rápida sucessão de uma sequência de imagens, que devem variar conforme as mudanças de algum parâmetro gráfico, como geometria, cor, iluminação, posição da câmera etc. A animação pode ser usada para apresentar variações de uma ou mais variáveis, como a variação do deslocamento dos vértices de uma grade ou a variação de um dado escalar mapeado em cores.

2.7.2 Interação de usuário

A **interação de usuário** é um recurso indispensável na visualização de dados. Isto por que o usuário necessita interagir com a visualização para explorá-la. Existem diversas técnicas de interação que permitem que o usuário interaja pelo *mouse* com a visualização, dentre elas: reposicionamento da câmera no espaço e ampliação/redução da escala. É pela interação que o usuário pode aplicar ou modificar as técnicas empregadas para visualização dos atributos de dados.

2.7.3 Renderização estereoscópica

A computação gráfica possui diversas técnicas para renderização de domínios 3D em dispositivos de exibição 2D, como projeção perspectiva, mudança de escala, iluminação, oclusão, sombreamento e animação. Todas elas possuem um objetivo em comum: reproduzir o efeito de profundidade dos objetos 3D. Entretanto, uma das técnicas mais eficientes para isto é a **paralaxe binocular**, que permite a produção de duas imagens ligeiramente deslocadas de um mesmo objeto ou cenário. Esta diferença é naturalmente interpretada pelo cérebro para obtenção da profundidade, o que produz o efeito 3D chamado de visão estereoscópica.

O processo de renderização por paralaxe binocular é chamado de **renderização estereoscópica**. Para apresentação das imagens estereoscópicas, existem diversos métodos, que podem ser classificados em duas categorias: **tempo multiplexado** e **tempo paralelo**. Nos métodos da primeira, o efeito 3D é produzido pela alternância na apresentação das imagens esquerda e direita. Já nos métodos da segunda, o efeito 3D é produzido pela apresentação das duas imagens combinadas e, posteriormente, filtradas por um processo de extração das visões esquerda e direita.

Um dos métodos mais simples e comuns de tempo paralelo é o **anaglifo**, no qual a estereoscopia é obtida pelo uso de um par de óculos para filtragem de componentes espectrais da luz. O processo de renderização envolve a geração de imagens formadas por duas camadas de cores sobrepostas – normalmente vermelho e ciano –, sendo que cada uma destas imagens representa uma perspectiva diferente. Os filtros dos óculos permitem apenas a passagem da cor correspondente ao filtro. O efeito de profundidade é gerado por este processo de filtragem. As vantagens desta técnica são que ela pode ser produzida em monitores, papel ou filmes e necessita de um par

de óculos de baixo custo. A Figura 2.9 ilustra o resultado da renderização estereoscópica por anaglifo.

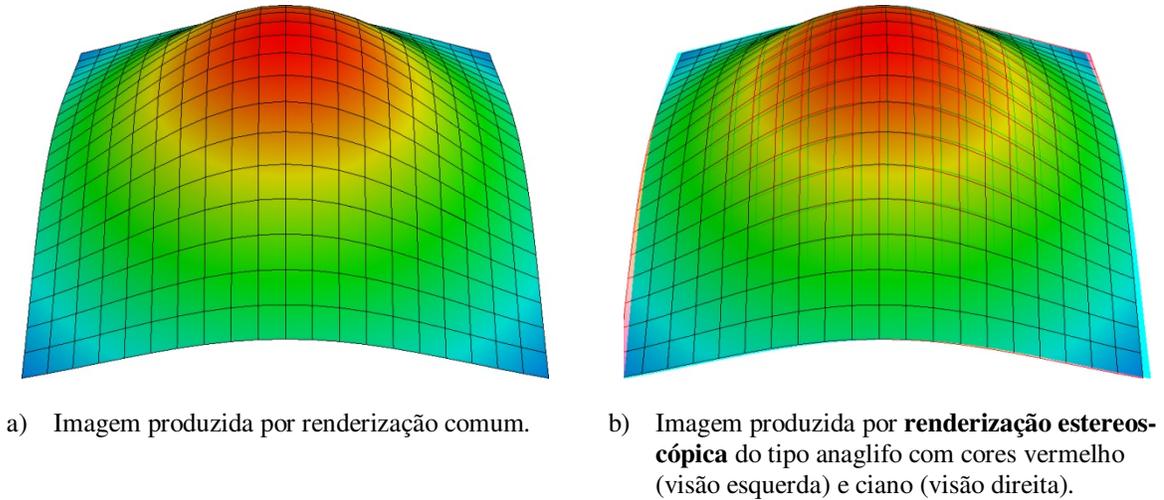


Figura 2.9 – Exemplo de uso da visão estereoscópica por anaglifo.

2.8 CONSIDERAÇÕES FINAIS

Como apresentado neste capítulo, a visualização científica é uma ramificação da área de visualização de dados que se dedica à pesquisa e desenvolvimento de técnicas para o mapeamento de dados científicos (geralmente de origem natural) em representações visuais interativas, com o objetivo de explorar as capacidades humana de percepção e cognição. Esta subárea possui um conjunto de técnicas fortemente consolidado que pode ser empregado para a análise e exploração de dados produzidos por simulação computacional de fenômenos naturais, tais como os fenômenos de propagação de ondas mecânicas – tema deste trabalho.

O conjunto de técnicas apresentado foi agrupado em: (1) representação de dados, que descreve a representação computacional e gráfica de conjunto de dados discretos, como células, grades e atributos de dados; (2) fluxo de visualização, que trata dos estágios de importação, filtração/enriquecimento, mapeamento e renderização de dados; (3) visualização de dados escalares e vetoriais, que descreve as técnicas de visualização dos tipos de dados explorados no contexto deste trabalho; (4) modelagem de conjunto de dados, o alicerce da maioria das técnicas de visualização de dados; e (5) técnicas de computação gráfica, uma das principais tecnologias usadas pela visualização de dados.

O conteúdo abordado neste capítulo é a base para os Capítulos 4 e 5. No próximo capítulo são tratados: o processo de simulação computacional dos fenômenos de propagação de ondas mecânicas em meios contínuos sólidos e os dados gerados neste processo, que são a entrada do fluxo de visualização; além dos conceitos teóricos fundamentais para a análise dos problemas geomecânicos complexos simulados por computador, ou da saída do fluxo de visualização.

3 SIMULAÇÃO E ANÁLISE DE ONDAS MECÂNICAS

Neste capítulo são tratados dois temas voltados aos fenômenos de propagação de ondas mecânicas em meios contínuos: o processo de simulação computacional e os dados gerados neste processo (Seção 3.1); e os conceitos teóricos fundamentais para o processo de análise e interpretação destes dados (Seção 3.2).

3.1 SIMULAÇÃO DE ONDAS MECÂNICAS

Considerando que não é propósito deste trabalho apresentar como o processo de simulação computacional dos fenômenos de propagação de ondas mecânicas é realizado, mas sim como as técnicas para mapeamento de dados em representações visuais da visualização científica podem ser aplicadas aos dados produzidos neste processo, esta seção aborda: os trabalhos relacionados ao tema (Seção 3.1.1); uma descrição sucinta da simulação computacional de um problema geomecânico (Seção 3.1.2); e uma descrição detalhada dos dados a serem visualizados, no contexto deste trabalho (Seção 3.1.3).

3.1.1 Trabalhos relacionados

Os fenômenos de propagação de ondas mecânicas compreendem diversos campos teóricos e de aplicações, que vão desde problemas geotécnicos à fratura em cristais. O recente interesse neste assunto é motivado pelas várias aplicações práticas na ciência e na engenharia. Na engenharia estrutural, por exemplo, o interesse é principalmente na reação das estruturas quando submetidas às cargas de impacto ou de explosão. Em cargas transientes, a teoria das ondas mecânicas pode ser suficiente para a previsão desta reação (GRAFF, 1991).

Com o desenvolvimento de formulações e de técnicas de aproximação numérica, uma série de problemas intratáveis pode ser solucionada. Este trabalho, contudo, se concentra na propagação de ondas em meios contínuos sólidos, particularmente em domínios ilimitados. As soluções numéricas para problemas complexos de propagação de ondas mecânicas 2D podem ser

encontradas nas obras de Rajapakse e Wang (1991; 1993), e Barros e Mesquita (1999). Os problemas 3D tem sido solucionados por Adolph et al. (2007) e Mesquita et al. (2009).

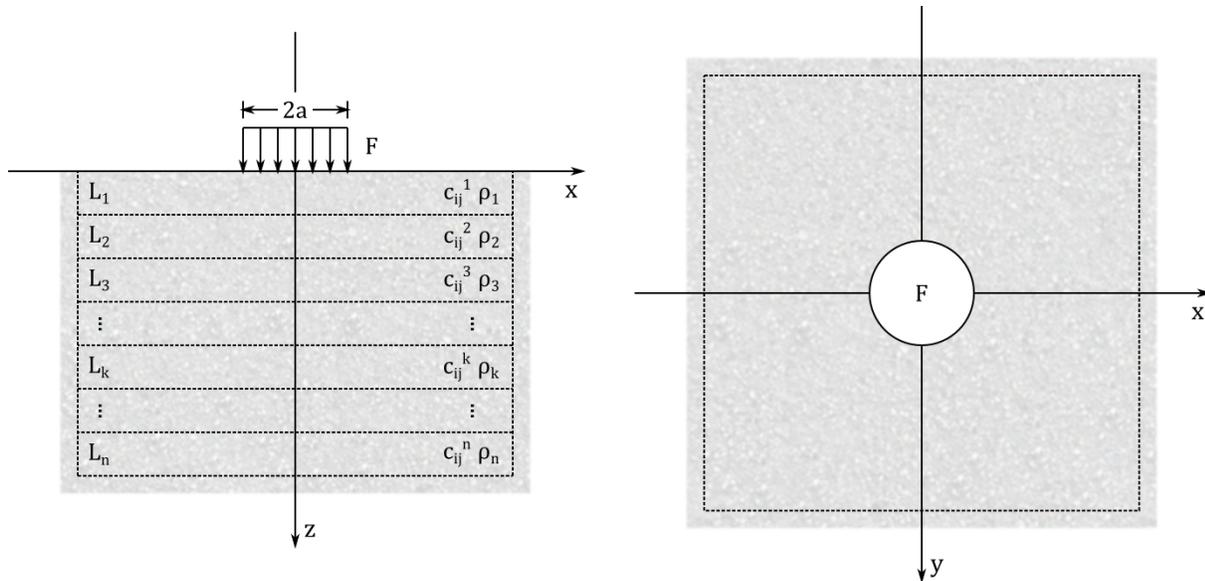


Figura 3.1 – Representação de um problema geomecânico complexo em Mecânica dos Sólidos.

3.1.2 Simulação computacional

Os dados utilizados para o desenvolvimento deste trabalho foram produzidos e fornecidos pelo Grupo de Métodos Computacionais em Mecânica do Contínuo (GMCMC)⁹. Nesta seção é brevemente descrito o processo de modelagem e simulação computacional para resolução de problemas complexos em Mecânica dos Sólidos especificamente estudados por este grupo.

Os problemas são geralmente modelados por uma série de camadas viscoelásticas horizontais excitadas harmonicamente por uma distribuição de tensão, que é a fonte geradora das ondas. As camadas estão contidas em um semiespaço transversalmente isotrópico, sendo que cada camada possui suas características de elasticidade, que influenciam na propagação das ondas.

⁹ Um grupo de pesquisa do Departamento de Mecânica Computacional (DMC) da Faculdade de Engenharia Mecânica (FEM) da Universidade Estadual de Campinas (Unicamp) dedicado ao desenvolvimento de métodos computacionais para resolução de problemas complexos em MMC, principalmente em Mecânica dos Sólidos. Para mais informações, acesse a página eletrônica do GMCMC no *website* do Diretório dos Grupos de Pesquisa no Brasil (DGP) do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), no endereço <http://dgp.cnpq.br/buscaoperacional/detalhegrupo.jsp?grupo=0079305BOOM6JX>.

A Figura 3.1 ilustra o modelo de um problema geomecânico complexo. À esquerda é apresentada a vista da profundidade do meio de propagação e à direita é apresentada a vista da superfície deste mesmo meio. O problema é descrito por n camadas L_k excitadas por uma distribuição de tensão superficial F_z . As camadas estão contidas em um semiespaço (em cinza) e as propriedades destas camadas são descritas pelos seus parâmetros constitutivos c_{ij}^k e ρ_k . As variações abruptas destes parâmetros ao longo do meio geram uma mudança de impedância com forte influência sobre o padrão de propagação das ondas. Estas alterações podem ser expressas em termos de componentes de deslocamento horizontais u_x e u_y e vertical u_z .

A distribuição de tensão não é sempre superficial, circular e perpendicular a um dos eixos de coordenadas, como foi ilustrado anteriormente. Ela também pode estar localizada no interior do meio, ser retangular e/ou inclinada. Estas alterações são realizadas de acordo com o problema modelado. A solução numérica de problemas mecânicos, isto é, a obtenção das componentes de deslocamento dos campos das ondas, é encontrada pela simulação computacional dos mesmos.

Tabela 3.1 – Organização dos dados gerados por simulação computacional.

x_0	y_0	z_0	$\Re(u_{x,0})$	$\Im(u_{x,0})$	$\Re(u_{y,0})$	$\Im(u_{y,0})$	$\Re(u_{z,0})$	$\Im(u_{z,0})$
x_1	y_1	z_1	$\Re(u_{x,1})$	$\Im(u_{x,1})$	$\Re(u_{y,1})$	$\Im(u_{y,1})$	$\Re(u_{z,1})$	$\Im(u_{z,1})$
x_2	y_2	z_2	$\Re(u_{x,2})$	$\Im(u_{x,2})$	$\Re(u_{y,2})$	$\Im(u_{y,2})$	$\Re(u_{z,2})$	$\Im(u_{z,2})$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
x_i	y_i	z_i	$\Re(u_{x,i})$	$\Im(u_{x,i})$	$\Re(u_{y,i})$	$\Im(u_{y,i})$	$\Re(u_{z,i})$	$\Im(u_{z,i})$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
x_{n-1}	y_{n-1}	z_{n-1}	$\Re(u_{x,n-1})$	$\Im(u_{x,n-1})$	$\Re(u_{y,n-1})$	$\Im(u_{y,n-1})$	$\Re(u_{z,n-1})$	$\Im(u_{z,n-1})$

3.1.3 Dados brutos

A simulação computacional gera uma volumosa quantidade de dados numéricos, os quais são compostos pelas coordenadas da posição de repouso ou equilíbrio dinâmico e pelas partes reais e imaginárias das componentes de deslocamento dos pontos de amostragem, normalmente dispostos sobre uma grade uniforme (ver Seção 2.2.3.1). A Tabela 3.1 ilustra a organização des-

tes dados para o caso 3D. Enquanto as linhas informam as n amostras i , as colunas representam seus atributos. As três primeiras colunas compõem as coordenadas x_i , y_i e z_i da posição de repouso e as demais correspondem, respectivamente, as partes reais (\Re) e imaginárias (\Im) das componentes de deslocamento $u_{x,i}$, $u_{y,i}$ e $u_{z,i}$.

3.2 ONDAS MECÂNICAS EM MEIOS SÓLIDOS

Esta seção visa introduzir os conceitos teóricos essenciais para a análise dos estudos de caso apresentados no Capítulo 6. Após a introdução, na Seção 3.2.1, são apresentadas as ondas de corpo e de superfície, duas classes de ondas que se propagam nos sólidos (Seção 3.2.2), as formas das frentes de ondas (Seção 3.2.3) e os fenômenos de propagação de ondas (Seção 3.2.4).

3.2.1 Introdução

As **ondas mecânicas**, também conhecidas como ondas elásticas ou de tensão, são perturbações que se propagam através dos meios materiais elásticos, entre sólidos, líquidos ou gasosos. Estas perturbações são propagadas pela transferência de energia mecânica, mas não de matéria. A energia mecânica faz com que as partículas de matéria que compõem o meio, como átomos ou moléculas, oscilem em torno de suas posições de repouso (centro de gravidade). As ondas sonoras, as ondas oceânicas e as ondas sísmicas são exemplos de ondas mecânicas, nos quais os meios de propagação são, respectivamente, o ar, a água e o solo (GARG; GHOSH; GUPTA, 2009; LINER, 2004).

Como as ondas mecânicas se comportam diferentemente em cada meio material, devido suas propriedades, e os dados utilizados para os estudos de caso são de problemas em meios sólidos, então nas seções seguintes são apresentadas as variações de ondas mecânicas, assim como seus fenômenos de propagação, para este tipo de meio.

3.2.2 Tipos de ondas

Diferentes tipos de ondas mecânicas são gerados quando, por exemplo, a região superficial de um meio material elástico sólido e espesso é perturbada por um carregamento dinâmico ou vibratório. Estas ondas podem ser classificadas em duas categorias, quanto às suas regiões de propagação no meio material: **ondas de corpo** ou **ondas de superfície** (NAIK; MALHOTRA; POPOVICS, 2004; FOWLER, 2005).

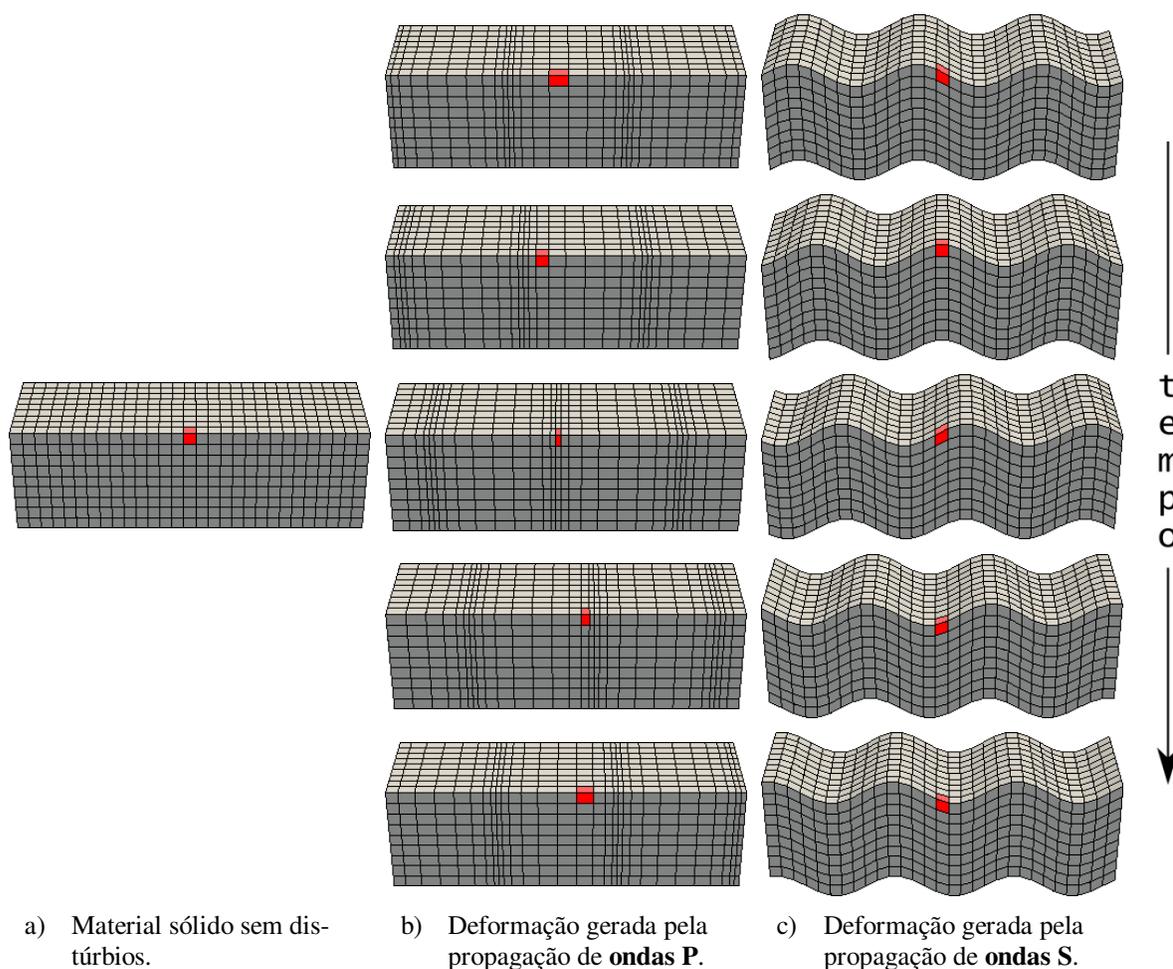


Figura 3.2 – Etapas da deformação de um material sólido gerada pela transmissão de ondas de corpo (baseado em Fowler, 2005).

3.2.2.1 Ondas de corpo

As **ondas de corpo**, como são chamadas na sismologia, são ondas sísmicas que se propagam através do corpo sólido da Terra. Pela capacidade que as ondas de corpo têm de se propagar por dentro do volume dos sólidos elásticos, elas também são chamadas de ondas de volume. A propagação das ondas de corpo é semelhante ao da luz: ocorre reflexão, refração ou absorção de energia nas interfaces onde ocorrem mudanças de velocidade. Nos sólidos elásticos isotrópicos existem dois tipos de ondas de corpo: **ondas P** e **ondas S**. Estas ondas são caracterizadas pela relação entre as direções do movimento das partículas do meio e da propagação do distúrbio ou da frente de onda (CARINO, 2004; LINER, 2004; FOWLER, 2005).

A velocidade das ondas P e S é determinada pela densidade e elasticidade do meio material através do qual elas se propagam. As ondas mecânicas, por exemplo, se propagam mais lentamente através de sólidos mais densos e mais rapidamente através de sólidos mais elásticos. A elasticidade é uma propriedade dos meios que podem ser deformados por uma força e que retornam às suas formas originais quando a força não está mais presente, tal como ocorrem nos sólidos (MONROE; WICANDER, 2009).

3.2.2.1.1 Ondas P

As **ondas P** são ondas de corpo que se propagam nos sólidos de forma semelhante às ondas sonoras no ar. Em comparação com as ondas S e as ondas de superfície, a velocidade de propagação das ondas P é superior. Por isso, elas são as primeiras a serem detectadas pelos sismógrafos na ocorrência de abalos sísmicos. As ondas P podem ser transmitidas através de qualquer meio material elástico. A sua propagação é dada por compressão e rarefação do meio material, o que provoca alteração no volume, mas não na forma do meio. Durante a passagem das ondas P, as partículas do meio são deslocadas paralelamente à direção de propagação das ondas. Pelas características citadas, as ondas P também são conhecidas como ondas primárias, ondas de compressão ou ondas longitudinais (LINER, 2004; FOWLER, 2005; MONROE; WICANDER, 2008; SEN, 2009). A Figura 3.2b apresenta as deformações de um material sólido ocasionadas pela transmissão de ondas P ao longo do tempo.

3.2.2.1.2 Ondas S

As **ondas S** são ondas de corpo capazes de se propagar apenas em meios materiais que suportam força de cisalhamento. Como os meios líquidos e gasosos não são rígidos, então as ondas S não conseguem se propagar por eles, apenas através dos sólidos. As ondas S são mais lentas do que as ondas P, entretanto mais rápidas do que as ondas de superfície. Nos sólidos, a velocidade das ondas S é aproximadamente equivalente a 60% da velocidade das ondas P. Entretanto, a amplitude das ondas S é superior à das ondas P. A propagação das ondas S é precedida por rotação e cisalhamento do material, o que provoca alteração na forma, mas não no volume do meio. Durante a passagem das ondas S, as partículas do material são deslocadas perpendicularmente em relação à direção de propagação das ondas. As ondas S podem ser classificadas quanto à direção das vibrações das partículas em **ondas S verticais** (S_v) ou **ondas S horizontais** (S_h). Pelas suas características, as ondas S também são conhecidas como ondas secundárias, ondas de cisalhamento ou ondas transversais (LOWRIE, 1997; LINER, 2004; NAIK; MALHOTRA; POPOVICS, 2004; FOWLER, 2005; MONROE; WICANDER, 2009). A Figura 3.2c apresenta as deformações de um meio material sólido provocadas pela transmissão das ondas S ao longo do tempo.

3.2.2.2 Ondas de superfície

Quando as ondas P e S chegam à superfície do material, ocorre uma combinação entre elas que resulta em outros tipos de ondas que se propagam apenas próximo da região superficial do meio, chamadas de **ondas de superfície**, dentre as quais se destacam as **ondas R** e as **ondas L**. Apesar de serem mais lentas do que as ondas de corpo, as ondas de superfície possuem maior amplitude. Nos abalos sísmicos, por exemplo, as ondas de superfície são as principais responsáveis pela destruição, isto por que elas possuem baixa frequência, longa duração e alta amplitude com decaimento muito lento (SEN, 2009).

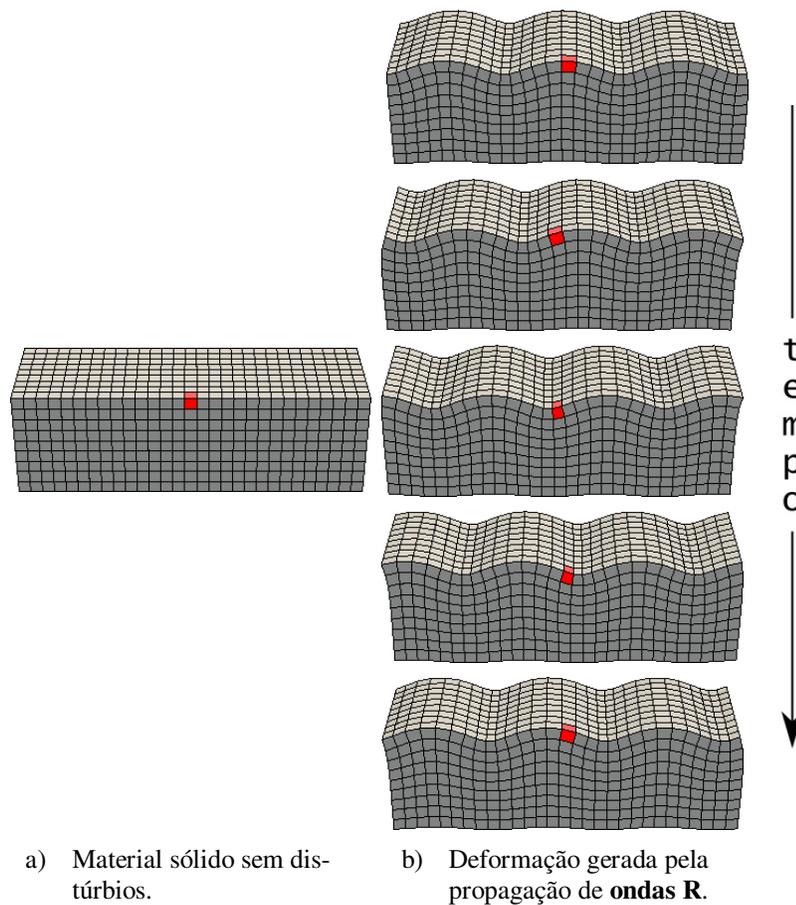


Figura 3.3 – Etapas da deformação de um material sólido gerada pela transmissão de ondas de superfície.

3.2.2.2.1 Ondas R

As **ondas R**, ou ondas de Rayleigh, são o resultado da interação entre as ondas P e S_v na superfície do meio. A propagação destas ondas é análoga à das ondas oceânicas e pode ser claramente observada em campos abertos durante a ocorrência de um terremoto. A trajetória de deslocamento das partículas na passagem das ondas R é elíptica sobre um plano vertical paralelo à direção de propagação das ondas. Nos meios sólidos, a velocidade de propagação das ondas R é aproximadamente equivalente a 55% da velocidade das ondas P e a trajetória das partículas é retrógrada – em oposição aos meios líquidos que é progressiva. As ondas R são 2D e podem ser decompostas em duas componentes: vertical e horizontal. Normalmente, as componentes verticais são 50% maiores do que as horizontais. A amplitude das ondas R decai exponencialmente com a profundidade do meio (GRAFF, 1991; NAIK; MALHOTRA; POPOVICS, 2004; FO-

WLER, 2005; SEN, 2009). A Figura 3.3b ilustra as deformações de um meio sólido ocasionadas pela transmissão das ondas R ao longo do tempo.

3.2.2.2 Ondas L

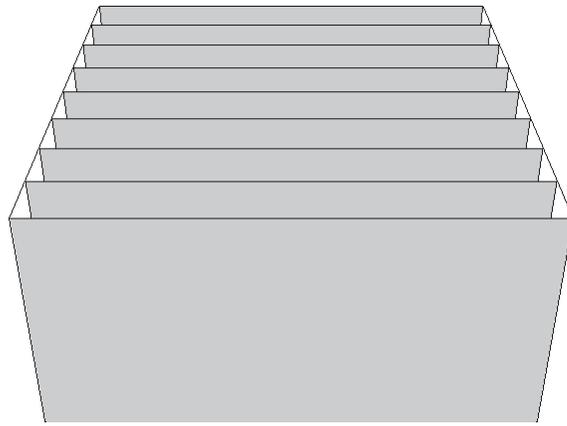
As **ondas L**, ou ondas de Love, são o resultado da interação entre as ondas P e S_h na superfície do meio e são mais lentas do que as ondas de corpo, porém mais rápidas do que as ondas R. A trajetória do movimento das partículas na passagem das ondas L é semelhante ao das ondas S, porém sobre um plano horizontal paralelo à direção de propagação das ondas (FOWLER, 2005; SEN, 2009).

3.2.3 Frentes de onda

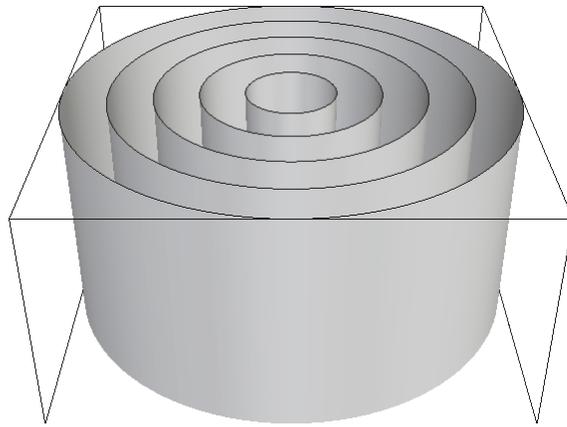
As frentes de onda definem as bordas dianteiras das ondas que se propagam pelo meio material. Há três tipos fundamentais de frentes de onda: **planas**, **cilíndricas** e **esféricas**. As formas das frentes de onda das ondas P, S, R e L dependem das características da fonte usada para gerar as ondas. Por exemplo, quando as ondas mecânicas são geradas pelo impacto de um ponto sobre a superfície de um sólido, as frentes de onda esféricas indicam a presença de ondas de corpo e as frentes de onda cilíndricas indicam a presença de ondas de superfície (CARINO, 2004). A Figura 3.4 apresenta os principais tipos de frentes de onda, cuja fonte de perturbação se encontra centralizada na parte superior do paralelogramo.

3.2.3.1 Frente de onda plana

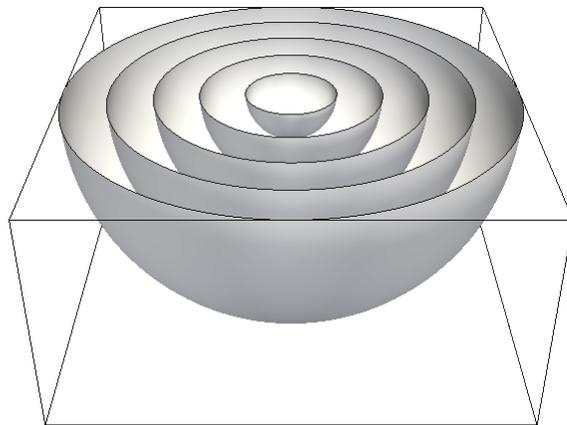
As **frentes de onda planas** são aquela cujas ondas assumem a forma de planos no espaço. Longitudinalmente e transversalmente estas frentes de ondas são retilíneas.



a) Frentes de onda planas



b) Frentes de onda cilíndricas



c) Frentes de onda esféricas

Figura 3.4 – Tipos de frente de onda.

3.2.3.2 Frente de onda cilíndrica

As **frentes de onda cilíndricas** são aquelas cujas ondas se propagam na forma de cilindros de mesma altura e de bases concêntricas. Estas frentes de onda possuem formas circunferenciais na longitude e retilíneas na transversal.

3.2.3.3 Frente de onda esférica

As **frentes de onda esféricas**, também chamadas de semiesféricas, são aquelas cujas ondas se propagam na forma de semiesferas concêntricas de centro localizado na fonte de perturbação. Longitudinalmente e transversalmente, estas frentes de onda possuem formas circunferenciais e semicircunferenciais, respectivamente.

3.2.4 Fenômenos de propagação de ondas

Durante a propagação das ondas mecânicas em um meio material elástico, elas podem encontrar interfaces de camadas com propriedades de elasticidade e/ou de densidade diferentes. Nesta situação, ocorrem os chamados fenômenos de propagação, dentre os quais se destacam: **absorção, reflexão, refração e difração** (CARINO, 2004).

3.2.4.1 Absorção

Na **absorção** ocorre transformação de energia mecânica em energia térmica. Com isso, há redução na velocidade de propagação das ondas.

3.2.4.2 Reflexão

A **reflexão** das ondas mecânicas é análoga ao da luz. Ela ocorre quando as ondas encontram interfaces de camadas reflexivas, que alteram a direção ou apenas o sentido de propagação das ondas.

3.2.4.3 Refração

A **refração** das ondas mecânicas ocorre a partir das interfaces das camadas cujas propriedades alteram a velocidade de propagação das ondas.

3.2.4.4 Difração

A **difração** ocorre quando as ondas mecânicas encontram um obstáculo no meio material, que faz com que as ondas sejam particionadas.

3.3 CONSIDERAÇÕES FINAIS

Neste capítulo foram abordados dois temas voltados ao estudo dos fenômenos de propagação de ondas mecânicas em meios contínuos: na Seção 3.1 foram apresentados os dados produzidos por simulação computacional, assim como o significado de seus atributos de dados, cujo entendimento é essencial para a fase de enriquecimento abordada no capítulo seguinte; e na Seção 3.2 foram apresentados os tipos de ondas que se propagam em meios sólidos, cujos conceitos são fundamentais para a análise dos estudos de casos apresentados no Capítulo 6.

4 VISUALIZAÇÃO DE ONDAS MECÂNICAS

Neste capítulo é apresentado o fluxo de visualização proposto para dados que descrevem o comportamento dos fenômenos de propagação de ondas mecânicas em meios contínuos sólidos gerados sobre domínios ilimitados e que admitam tratamento linear. Este fluxo, baseado no que foi apresentado na Seção 2.3, é composto por cinco estágios: importação, enriquecimento, filtragem, mapeamento e renderização.

Na **importação** (Seção 4.1) são apresentados os dados e como estes podem ser organizados e representados visualmente. No **enriquecimento** (Seção 4.2) são apresentadas as formulações matemáticas empregadas para o cálculo das informações, ou seja, dos novos atributos de dados. Na **filtragem** (Seção 4.3) é apresentada como as técnicas de modelagem de dados da Seção 2.6 podem ser empregadas sobre o conjunto de dados. No **mapeamento** (Seção 4.4) é apresentado como as técnicas para mapeamento de dados das Seções 2.4 e 2.5 podem ser empregadas para visualização dos atributos de dados. E por fim, na **renderização** (Seção 4.5) é apresentado como as técnicas de computação gráfica da Seção 2.7 podem ser empregadas para suporte à visualização.

Para simplificar as explicações apresentadas ao longo deste capítulo, os seguintes termos devem ser considerados: “dados” como sendo o mesmo produzido pelo GMCMC por intermédio de simulação computacional dos fenômenos de propagação de ondas mecânicas, tal como visto na Seção 3.1; e “ponto” ou “amostra” como sendo um dos vários pontos de amostragem que compõem estes dados. Nas equações a seguir, os índices separados por vírgula representam os atributos de dados.

As seguintes notações matemáticas também devem ser consideradas: n_x , n_y e n_z como sendo as dimensões de uma grade estruturada, ou seja, o número de amostras respectivamente nas direções das coordenadas x , y e z ; $n = n_x \times n_y \times n_z$ como sendo o número total de amostras presentes no conjunto de dados; i como sendo o índice de uma das n amostras, no qual $\{i \in \mathbb{N} \mid i < n\}$; $P_i = (x_i, y_i, z_i)$ como sendo a posição de repouso de uma amostra; $u_{x,i}$, $u_{y,i}$ e $u_{z,i}$ como sendo as componentes de deslocamento de uma amostra sobre os eixos de coordenadas; t como sendo o instante de tempo normalizado, no qual $\{t \in \mathbb{R} \mid 0 \leq t < 2\pi\}$; e \Re e \Im como sendo, respectivamente, as partes reais e imaginárias de um número complexo.

4.1 IMPORTAÇÃO

Como tratado na Seção 3.1.3, os dados são compostos pelas coordenadas da posição de repouso e pelas partes reais e imaginárias das componentes de deslocamento das amostras. A Tabela 3.1 ilustra a organização destes dados para o caso 3D. Enquanto as linhas representam as amostras, as colunas representam seus atributos. As três primeiras colunas compõem as coordenadas da posição de repouso P_i e as demais as partes reais e imaginárias das componentes de deslocamento $u_{x,i}$, $u_{y,i}$ e $u_{z,i}$, respectivamente.

Os dados produzidos são gerados sobre uma grade uniforme, ou seja, suas amostras são igualmente espaçadas e paralelas aos eixos de coordenadas (ver Seção 2.2.3.1). Ciente de que as partículas do meio de propagação oscilam em torno de suas posições de repouso e, por isso, provocam uma deformação do mesmo (ver Seção 3.2.1), têm-se então que estas amostras – que também são os vértices da grade – se deslocam livremente ao longo do tempo alterando as coordenadas de suas posições geométricas, mas preservando a topologia da grade sem gerar rupturas. Com isso, e considerando as informações apresentadas na Seção 2.2.3.3, é possível notar que o conjunto de dados pode ser organizado e representado por uma grade estruturada.

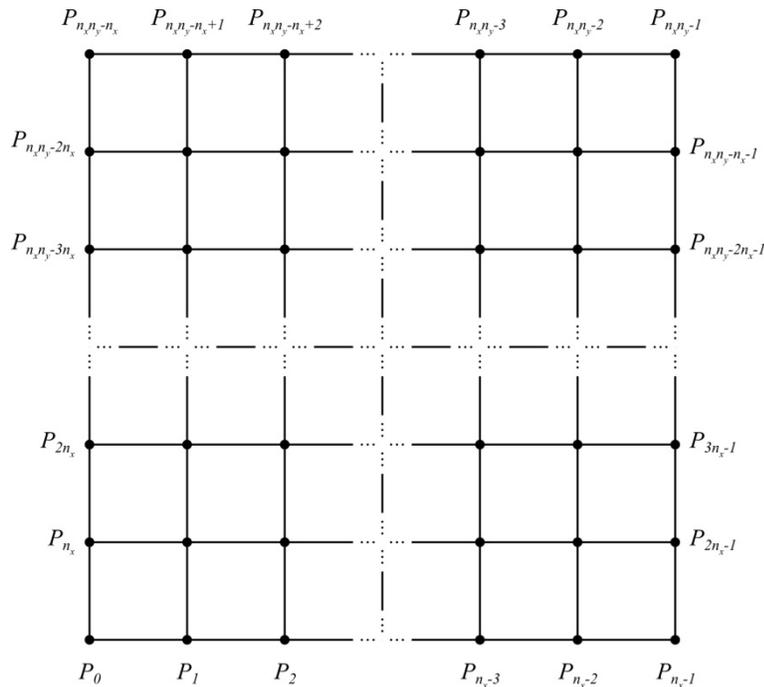


Figura 4.1 – Disposição das amostras sobre uma grade estruturada 2D.

Como visto na Seção 2.2.3.3, a topologia de uma grade estruturada é especificada por intermédio de suas dimensões n_x , n_y e n_z . Como o conjunto de dados não dispõe destas informações, então é necessário calculá-las. Estes valores podem ser facilmente identificados por uma análise das variações das coordenadas apresentadas na Tabela 3.1, que variam em ordem crescente respectivamente sobre os eixos x , y e z , assim como ilustrado na Tabela 4.1. A Figura 4.1 apresenta a disposição das amostras sobre uma grade estruturada 2D.

Tabela 4.1 – Ordem de variação crescente das coordenadas.

$x_{\text{mín}}$			
\vdots	$y_{\text{mín}}$		
$x_{\text{máx}}$			
$x_{\text{mín}}$			
\vdots	\vdots	$z_{\text{mín}}$	
$x_{\text{máx}}$			
$x_{\text{mín}}$			
\vdots	$y_{\text{máx}}$		
$x_{\text{máx}}$			
$x_{\text{mín}}$			
\vdots	$y_{\text{mín}}$		
$x_{\text{máx}}$			
$x_{\text{mín}}$			
\vdots	\vdots	\vdots	
$x_{\text{máx}}$			
$x_{\text{mín}}$			
\vdots	$y_{\text{máx}}$		
$x_{\text{máx}}$			
$x_{\text{mín}}$			
\vdots	$y_{\text{mín}}$		
$x_{\text{máx}}$			
$x_{\text{mín}}$			
\vdots	\vdots	$z_{\text{máx}}$	
$x_{\text{máx}}$			
$x_{\text{mín}}$			
\vdots	$y_{\text{máx}}$		
$x_{\text{máx}}$			

4.2 ENRIQUECIMENTO

Visto que é de interesse dos especialistas visualizar e explorar as características dos fenômenos descritos nos dados, tal como a amplitude e o ângulo de fase das ondas, e que os dados são compostos unicamente pelas coordenadas das posições de repouso e pelas componentes de deslocamento das amostras, então é necessário enriquecer estes dados com informações (novos atributos de dados) que possam ser posteriormente mapeadas para representações visuais. Sendo assim, nesta seção são descritas as formulações matemáticas usadas para o enriquecimento do conjunto de dados a partir dos seus atributos de dados iniciais.

4.2.1 Deslocamento

As partículas de um determinado meio de propagação, ao oscilarem, se deslocam provocando uma deformação do mesmo ao longo do tempo. Pela simulação computacional este deslocamento é descrito pelas componentes de deslocamento. Desta maneira, o deslocamento de uma partícula em um determinado instante de tempo t pode ser representado por um vetor $\vec{d}_{t,i}$, cuja magnitude informa o quanto uma amostra i se deslocou em relação a sua posição de repouso, como expresso pela Equação 4.1. Para seu cálculo são necessários apenas os valores constantes reais e imaginários das componentes de deslocamento da amostra.

$$\vec{d}_{t,i} = (d_{x,t,i}, d_{y,t,i}, d_{z,t,i}) \Rightarrow \begin{cases} d_{x,t,i} = \Re(u_{x,i}) \cos t - \Im(u_{x,i}) \sen t \\ d_{y,t,i} = \Re(u_{y,i}) \cos t - \Im(u_{y,i}) \sen t \\ d_{z,t,i} = \Re(u_{z,i}) \cos t - \Im(u_{z,i}) \sen t \end{cases} \quad (4.1)$$

Os deslocamentos $d_{x,t,i}$, $d_{y,t,i}$ e $d_{z,t,i}$ informam o quanto uma amostra i se deslocou no instante de tempo t em relação, respectivamente, aos eixos de coordenadas x , y e z . Estes deslocamentos podem ser aplicados, por exemplo, para a visualização do efeito de propagação das ondas e dos fenômenos de propagação, tal como transmissão e reflexão.

4.2.2 Posição relativa

A posição relativa de uma partícula é equivalente a sua posição atual em um determinado instante de tempo após sofrer um deslocamento. Sendo assim, a posição relativa $P_{t,i}$ de uma amostra i pode ser obtida pela soma da sua posição de repouso P_i com o seu deslocamento $\vec{d}_{t,i}$ no instante de tempo t , como formulado na Equação 4.2.

$$\vec{P}_{t,i} = \vec{P}_i + \vec{d}_{t,i} = (x_{t,i}, y_{t,i}, z_{t,i}) \Rightarrow \begin{cases} x_{t,i} = x_i + d_{x,t,i} \\ y_{t,i} = y_i + d_{y,t,i} \\ z_{t,i} = z_i + d_{z,t,i} \end{cases} \quad (4.2)$$

A Figura 4.2 ilustra a posição relativa de um ponto após sofrer um deslocamento sobre os eixos de coordenadas x e y (para o caso 2D). Com uso do recurso de animação, a posição relativa pode ser aplicada para a visualização do efeito de propagação das ondas.

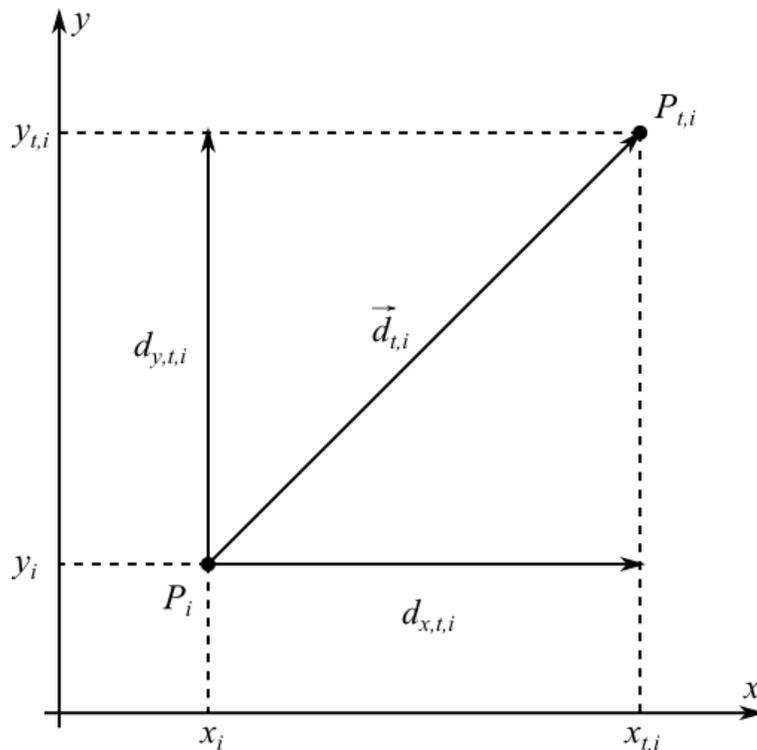


Figura 4.2 – Posição relativa de uma amostra após sofrer um deslocamento.

4.2.3 Trajetória

Quando o movimento de uma partícula é composto por uma sobreposição de movimentos harmônicos simples de mesma frequência e em direções perpendiculares entre si, mas com diferentes amplitudes, então o deslocamento desta partícula descreve uma trajetória elíptica (GARG; GHOSH; GUPTA, 2009). Como as simulações computacionais realizadas são deste tipo, então as trajetórias das amostras podem ser representadas por elipses centradas em suas posições de repouso.

As elipses podem ser obtidas a partir de suas extremidades (LEITHOLD, 1994), que são os quatro pontos extremos dos eixos maior e menor de uma elipse. Tratando-se das ondas mecânicas, as extremidades de uma elipse, que descrevem a trajetória de uma partícula, são as posições relativas desta partícula no momento em que as magnitudes dos seus vetores de deslocamento são máximas ou mínimas. Para o cálculo destas posições relativas, é necessário obter os valores de t que satisfaçam as condições apresentadas. Os passos necessários para se obter t são apresentados a seguir.

Para facilitar o processo para obtenção da formulação, considere: $g(t) = \vec{d}_{t,i}$, $a_x = \Re(u_{x,i})$, $a_y = \Re(u_{y,i})$, $a_z = \Re(u_{z,i})$, $b_x = \Im(u_{x,i})$, $b_y = \Im(u_{y,i})$ e $b_z = \Im(u_{z,i})$. A partir disso, a trajetória de uma partícula pode ser descrita pela função:

$$\begin{aligned}
 h(t) &= \|g(t)\|^2 \\
 &= x^2 + y^2 + z^2 \\
 &= (a_x \cos t - b_x \sin t)^2 + (a_y \cos t - b_y \sin t)^2 + (a_z \cos t - b_z \sin t)^2 \\
 &= (a_x^2 + a_y^2 + a_z^2) \cos^2 t + (b_x^2 + b_y^2 + b_z^2) \sin^2 t - 2(a_x b_x + a_y b_y + a_z b_z) \sin t \cos t \\
 &= (a_x^2 + a_y^2 + a_z^2) \cos^2 t + (b_x^2 + b_y^2 + b_z^2) \sin^2 t - (a_x b_x + a_y b_y + a_z b_z) \sin 2t
 \end{aligned} \tag{4.3}$$

Para obter os valores de t que satisfazem as condições apresentadas anteriormente é necessário calcular a derivada primeira de $h(t)$:

$$\begin{aligned}
h'(t) &= 2(b_x^2 + b_y^2 + b_z^2) \sin t \cos t - 2(a_x^2 + a_y^2 + a_z^2) \sin t \cos t - 2(a_x b_x + a_y b_y + a_z b_z) \cos 2t \\
&= 2[b_x^2 + b_y^2 + b_z^2 - (a_x^2 + a_y^2 + a_z^2)] \sin t \cos t - 2(a_x b_x + a_y b_y + a_z b_z) \cos 2t \\
&= [b_x^2 + b_y^2 + b_z^2 - (a_x^2 + a_y^2 + a_z^2)] \sin 2t - 2(a_x b_x + a_y b_y + a_z b_z) \cos 2t
\end{aligned} \tag{4.4}$$

Igualando-se a primeira derivada de $h(t)$ a zero, tem-se o valor de um instante de tempo t em que o deslocamento é máximo ou mínimo:

$$\begin{aligned}
h'(t) &= 0 \\
[b_x^2 + b_y^2 + b_z^2 - (a_x^2 + a_y^2 + a_z^2)] \sin 2t &= 2(a_x b_x + a_y b_y + a_z b_z) \cos 2t \\
\frac{\sin 2t}{\cos 2t} &= \frac{2(a_x b_x + a_y b_y + a_z b_z)}{b_x^2 + b_y^2 + b_z^2 - (a_x^2 + a_y^2 + a_z^2)} \\
\tan 2t &= \frac{2(a_x b_x + a_y b_y + a_z b_z)}{b_x^2 + b_y^2 + b_z^2 - (a_x^2 + a_y^2 + a_z^2)} \\
2t &= \tan^{-1} \left[\frac{2(a_x b_x + a_y b_y + a_z b_z)}{b_x^2 + b_y^2 + b_z^2 - (a_x^2 + a_y^2 + a_z^2)} \right] \\
t &= \frac{\tan^{-1} \left[\frac{2(a_x b_x + a_y b_y + a_z b_z)}{b_x^2 + b_y^2 + b_z^2 - (a_x^2 + a_y^2 + a_z^2)} \right]}{2}
\end{aligned} \tag{4.5}$$

Com isso, tem-se que as quatro soluções de t são:

$$\begin{aligned}
t_1 &= \frac{\tan^{-1} \left[\frac{2(a_x b_x + a_y b_y + a_z b_z)}{b_x^2 + b_y^2 + b_z^2 - (a_x^2 + a_y^2 + a_z^2)} \right]}{2} \\
t_2 &= t_1 + \frac{\pi}{2} \\
t_3 &= t_1 + \pi \\
t_4 &= t_1 + \frac{3\pi}{2}
\end{aligned} \tag{4.6}$$

Enfim, retornando às notações padrão, tem-se que a formulação para obtenção dos instantes de tempo $t_{k,i}$ para $k = \{1, 2, 3, 4\}$, em que o módulo dos vetores de deslocamento são máximos ou mínimos é dado pela expressão:

$$t_{k,i} = \frac{\tan^{-1} \left\{ \frac{2[\Re(u_{x,i}) \Im(u_{x,i}) + \Re(u_{y,i}) \Im(u_{y,i}) + \Re(u_{z,i}) \Im(u_{z,i})]}{\Im(u_{x,i})^2 + \Im(u_{y,i})^2 + \Im(u_{z,i})^2 - [\Re(u_{x,i})^2 + \Re(u_{y,i})^2 + \Re(u_{z,i})^2]} \right\} + (k-1)\pi}{2} \quad (4.7)$$

Com uso da Equação 4.7, tem-se, finalmente, que a trajetória de uma amostra i pode ser representada por uma elipse, cujas extremidades se encontram sob os pontos $P_{t_1,i}$, $P_{t_2,i}$, $P_{t_3,i}$ e $P_{t_4,i}$, tal como ilustrado na Figura 4.3. Note ainda que $\vec{d}_{t_1,i} = -\vec{d}_{t_3,i}$ e $\vec{d}_{t_2,i} = -\vec{d}_{t_4,i}$. Pela visualização da trajetória é possível verificar o quão elíptica é a trajetória das partículas e também a inclinação do eixo maior da elipse para o caso 2D ou do plano da elipse com relação ao plano do meio de propagação para o caso 3D.

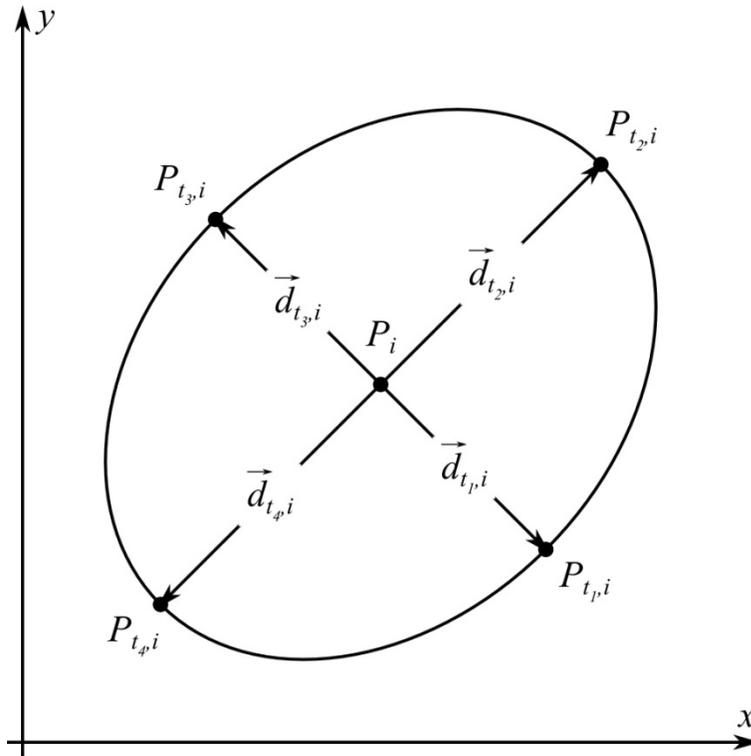


Figura 4.3 – Vértices da elipse que descreve a trajetória de deslocamento de uma amostra.

4.2.4 Relação de aspecto

Dado que o deslocamento de uma partícula descreve uma trajetória elíptica, então as características de uma elipse também podem ser usadas para enriquecimento do conjunto de dados. A relação de aspecto é uma característica que pode ser explorada. Isto por que ela pode ajudar na identificação do predomínio das ondas de corpo ou de superfície sobre uma região (ver Seção 3.2.2). As ondas de corpo, por exemplo, possuem uma relação de aspecto nula, já que a trajetória das partículas é retilínea.

A relação de aspecto de uma elipse é dada pela razão entre os seus eixos menor e maior (PIZLO, 2008). A Equação 4.8 apresenta esta formulação, no qual r é a relação de aspecto, tal que $\{r \in \mathbb{R} | 0 \leq r < 1\}$; e a e b são os valores dos eixos maior e menor da elipse, respectivamente.

$$r = \frac{b}{a} \quad (4.8)$$

Note que se $r = 0$, então a elipse assume a forma de um segmento de reta; e se $r = 1$, então a elipse assume a forma de uma circunferência (ver Figura 4.4). Ou seja, a relação de aspecto informa o quão achatada (ou arredondada) é uma elipse.

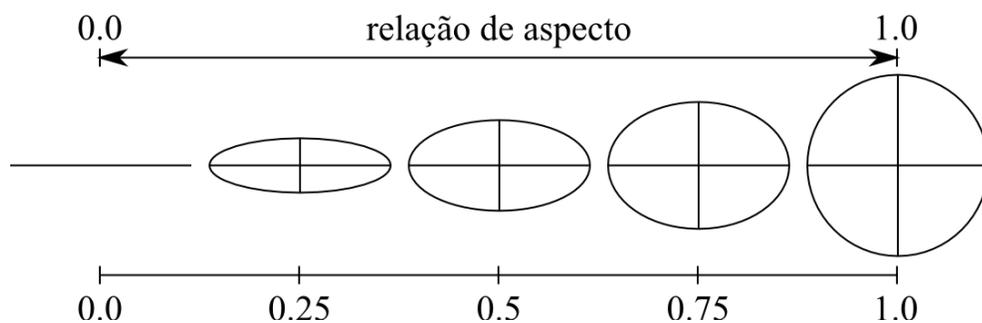


Figura 4.4 – Variação da relação de aspecto de uma elipse.

Desta maneira, a relação de aspecto de uma elipse que descreve a trajetória de uma partícula pode ser obtida de duas formas: a partir dos pontos relativos $P_{1,i}$, $P_{2,i}$, $P_{3,i}$ e $P_{4,i}$ ou por meio dos vetores de deslocamento $\vec{d}_{1,i}$ e $\vec{d}_{2,i}$.

Da primeira forma, os valores de a e b são equivalentes as distâncias entre os vértices da elipse, que são os pontos relativos $P_{t_1,i}$, $P_{t_2,i}$, $P_{t_3,i}$ e $P_{t_4,i}$ (ver Equação 4.9).

$$\begin{aligned}
 a &= \begin{cases} \|P_{t_3,i} - P_{t_1,i}\| & \text{se } \|P_{t_3,i} - P_{t_1,i}\| \geq \|P_{t_4,i} - P_{t_2,i}\| \\ \|P_{t_4,i} - P_{t_2,i}\| & \text{se } \|P_{t_3,i} - P_{t_1,i}\| < \|P_{t_4,i} - P_{t_2,i}\| \end{cases} \\
 b &= \begin{cases} \|P_{t_3,i} - P_{t_1,i}\| & \text{se } \|P_{t_3,i} - P_{t_1,i}\| < \|P_{t_4,i} - P_{t_2,i}\| \\ \|P_{t_4,i} - P_{t_2,i}\| & \text{se } \|P_{t_3,i} - P_{t_1,i}\| \geq \|P_{t_4,i} - P_{t_2,i}\| \end{cases}
 \end{aligned} \tag{4.9}$$

Da segunda forma, como $\vec{d}_{t_k,i} = -\vec{d}_{t_{k+2},i}$ para $k=1$ ou $k=2$, então os valores de a e b também podem ser obtidos por meio dos vetores de deslocamento $\vec{d}_{t_1,i}$ e $\vec{d}_{t_2,i}$ (ou $\vec{d}_{t_3,i}$ e $\vec{d}_{t_4,i}$), tal como apresentado na Equação 4.10.

$$\begin{aligned}
 a &= \begin{cases} 2\|\vec{d}_{t_1,i}\| & \text{se } 2\|\vec{d}_{t_1,i}\| \geq 2\|\vec{d}_{t_2,i}\| \\ 2\|\vec{d}_{t_2,i}\| & \text{se } 2\|\vec{d}_{t_1,i}\| < 2\|\vec{d}_{t_2,i}\| \end{cases} \\
 b &= \begin{cases} 2\|\vec{d}_{t_1,i}\| & \text{se } 2\|\vec{d}_{t_1,i}\| < 2\|\vec{d}_{t_2,i}\| \\ 2\|\vec{d}_{t_2,i}\| & \text{se } 2\|\vec{d}_{t_1,i}\| \geq 2\|\vec{d}_{t_2,i}\| \end{cases}
 \end{aligned} \tag{4.10}$$

4.2.5 Sentido do movimento de rotação

O sentido do movimento de rotação da trajetória de uma partícula também pode ajudar na identificação do tipo de ondas que predominam sobre uma região, como por exemplo, das ondas superficiais de Rayleigh, nas quais as partículas oscilam no sentido oposto ao de propagação das ondas. Esta informação pode ser representada por um valor escalar, no caso das ondas 2D, ou por um valor vetorial, no caso das ondas 3D.

4.2.5.1 Ondas mecânicas 2D

No caso das ondas 2D, como as partículas se movimentam somente sobre o plano do meio de propagação das ondas, então as trajetórias destas partículas estão neste plano. Sendo assim, o sentido do movimento de rotação da trajetória de uma partícula pode ser representado por um valor escalar (booleano), já que esta informação só pode assumir duas possibilidades: horário ou anti-horário.

O sentido do movimento de rotação R_i de uma amostra i pode ser obtido pelo produto escalar entre os vetores normais dos planos da trajetória da partícula n_i e do meio de propagação das ondas n_α . Desta maneira, se o resultado for negativo, então R_i é horário; caso contrário, se o resultado for positivo, então R_i é anti-horário (ver Equação 4.12). O vetor normal n_i é dado pelo versor do produto vetorial dos vetores de deslocamento $\vec{d}_{t_a,i}$ e $\vec{d}_{t_b,i}$, para $t_a < t_b < t_a + \pi$ (ver Equação 4.11). Note que como os planos da trajetória da partícula e do meio de propagação das ondas são coincidentes, então os vetores normais n_i e n_α são paralelos.

$$\vec{n}_i = \frac{\vec{d}_{t_a,i} \times \vec{d}_{t_b,i}}{\|\vec{d}_{t_a,i} \times \vec{d}_{t_b,i}\|} \quad (4.11)$$

$$R_i = \begin{cases} \text{horário} & \text{se } \vec{n}_i \cdot \vec{n}_\alpha < 0 \\ \text{anti-horário} & \text{se } \vec{n}_i \cdot \vec{n}_\alpha \geq 0 \end{cases} \quad (4.12)$$

A Figura 4.5 ilustra as duas possibilidades do sentido do movimento de rotação da trajetória de uma partícula (amostra P_i). Como os valores de t_1 e t_2 já foram encontrados anteriormente pela Equação 4.7 e estes satisfazem as condições para cálculo do vetor normal n_i (ou seja, $t_1 < t_2 < t_1 + \pi$), então os valores dos vetores de deslocamento de $\vec{d}_{t_1,i}$ e $\vec{d}_{t_2,i}$ foram usados e representados nesta ilustração. No primeiro caso (Figura 4.5a), os vetores normais n_i e n_α são contravetores¹⁰; e o produto escalar de ambos tem como resultado um valor negativo, que significa

¹⁰ Segundo Venturi (1949), dois vetores paralelos são **contravetores** se de sentidos contrários.

que o sentido é horário. Já no segundo caso (Figura 4.5b), os vetores normais \vec{n}_i e \vec{n}_α são equiversos¹¹; e o produto escalar de ambos tem como resultado um valor positivo, que significa que o sentido é anti-horário.

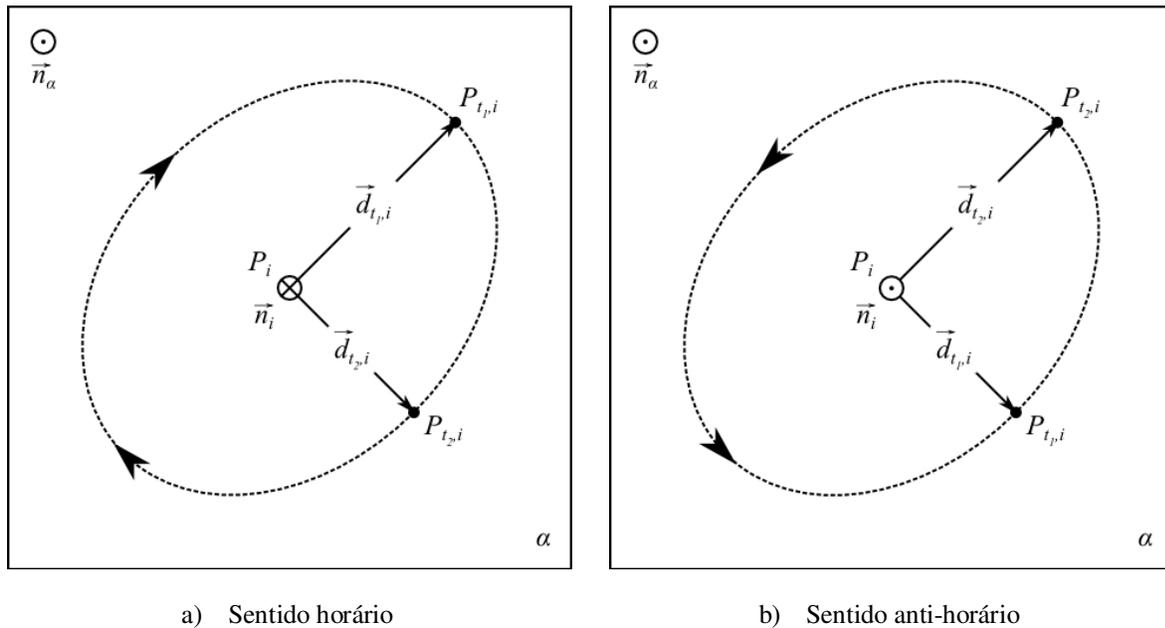


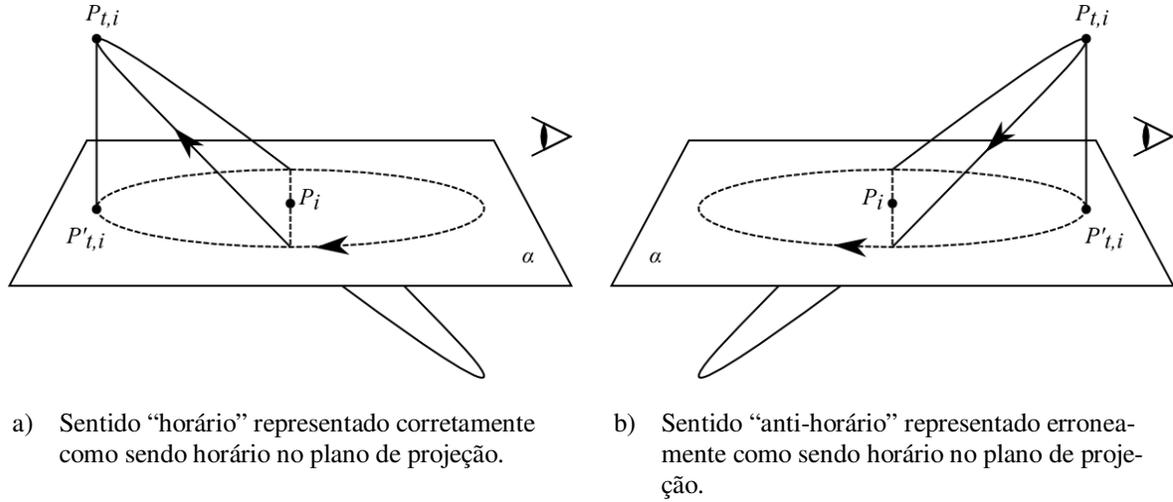
Figura 4.5 – Sentido do movimento de rotação da trajetória de uma partícula para o caso das ondas mecânicas 2D.

4.2.5.2 Ondas mecânicas 3D

No caso das ondas mecânicas 3D, as partículas não se movimentam sobre o plano do meio de propagação das ondas, já que as trajetórias destas partículas não estão neste plano. Uma alternativa seria projetar a trajetória das partículas no plano do meio de propagação das ondas mecânicas e, em seguida, realizar o procedimento descrito para o caso das ondas mecânicas 2D. Entretanto, por causa da projeção, uma partícula cujo sentido do movimento de rotação da trajetória seja “horário” pode ser representado erroneamente como sendo anti-horário no plano do meio de propagação das ondas mecânicas, ou vice-versa (ver Figura 4.6). Por isso, o sentido do movimento de rotação da trajetória de uma partícula não deve ser representado por um valor escalar, como no caso das ondas mecânicas 2D, mas pode ser expresso por um valor vetorial. Vale lembrar, que

¹¹ Segundo Venturi (1949), dois vetores paralelos são **equiversos** se de mesmo sentido.

a visualização não é restrita a apenas uma partícula e sim a um conjunto de partículas, o que pode causar confusão na visualização com uso da projeção.



- a) Sentido “horário” representado corretamente como sendo horário no plano de projeção.
- b) Sentido “anti-horário” representado erroneamente como sendo horário no plano de projeção.

Figura 4.6 – Projeção da trajetória de uma partícula no plano do meio de propagação das ondas mecânicas para obtenção do sentido do movimento de rotação para o caso das ondas mecânicas 2D.

O sentido do movimento de rotação \vec{r}_i de uma amostra i pode ser representado por um vetor, cuja origem se encontra em P_i e sua extremidade sobre um dos dois pontos (P_{t_α} ou $P_{t_\alpha+\pi}$) onde a trajetória desta amostra intersecta o plano do meio \vec{n}_α , tal como ilustrado na Figura 4.7. Estes dois pontos podem ser encontrados pela Equação 4.13, que expressa o instante de tempo t_α em que ocorre a intersecção neste plano. $u_{\vec{n}_\alpha}$ corresponde à componente de deslocamento de eixo paralelo à \vec{n}_α . Se o plano do meio coincide com o plano xy , por exemplo, então a normal deste plano é equiverso ao eixo z . E a componente de deslocamento $u_{\vec{n}_\alpha}$ é equivalente à componente sob este mesmo eixo, ou seja, u_z .

$$t_\alpha = \tan^{-1} \left[\frac{\Re(u_{\vec{n}_\alpha})}{\Im(u_{\vec{n}_\alpha})} \right] \quad (4.13)$$

$$\vec{r}_i = \begin{cases} \vec{d}_{t_\alpha, i} & \text{se } (\vec{d}_{t_\alpha, i} \times \vec{n}_\alpha) \cdot \vec{n}_i < 0 \\ \vec{d}_{t_\alpha+\pi, i} & \text{se } (\vec{d}_{t_\alpha, i} \times \vec{n}_\alpha) \cdot \vec{n}_i \geq 0 \end{cases} \quad (4.14)$$

O que determina qual dos dois pontos (P_{t_α} ou $P_{t_\alpha+\pi}$) será a extremidade de \vec{r}_i é o resultado do produto vetorial de $\vec{d}_{t_\alpha, i}$ com a normal do plano do meio n_α , seguido do produto escalar deste resultado com a normal do plano da trajetória da amostra n_i , assim como expresso na Equação 4.14. Desta maneira, se o resultado for negativo, então \vec{r}_i é o vetor obtido pela função $\vec{d}_{t_\alpha, i}$; caso contrário, então \vec{r}_i é o vetor obtido pela função $\vec{d}_{t_\alpha+\pi, i}$.

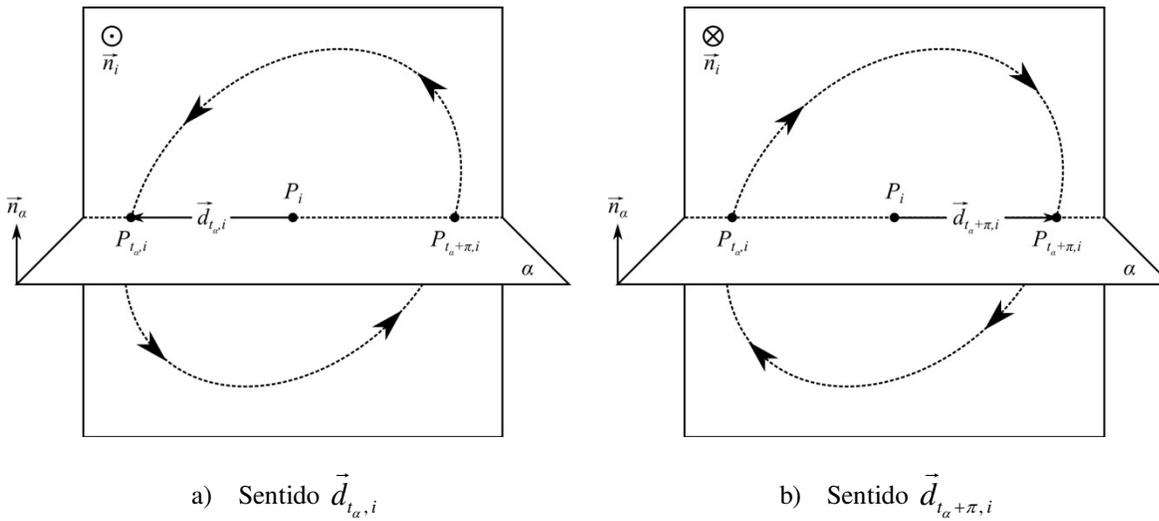


Figura 4.7 – Sentido do movimento de rotação da trajetória de uma partícula para o caso das ondas mecânicas 3D.

A Figura 4.7 representa o sentido do movimento de rotação da trajetória de uma partícula representado por um valor vetorial. Em ambos os casos (Figuras 4.7a e 4.7b) a trajetória de deslocamento da partícula é igual, porém o sentido do movimento de rotação é diferente. O que resulta nos vetores contrários $\vec{d}_{t_\alpha, i}$ e $\vec{d}_{t_\alpha+\pi, i}$ de mesma magnitude para representação do sentido do movimento de rotação no meio de propagação das ondas mecânicas. Enfim, \vec{r}_i indica o ponto da trajetória de deslocamento em que a partícula intersecta o plano α vindo do semiespaço positivo para o negativo.

4.2.6 Ângulo de inclinação

O ângulo de inclinação pode ser representado por um valor escalar, tanto para o caso das ondas mecânicas 2D quanto 3D. Entretanto, há duas maneiras de calculá-lo e interpretá-lo.

4.2.6.1 Ondas mecânicas 2D

Para o caso das ondas mecânicas 2D, como as partículas se deslocam apenas sobre o plano do meio de propagação, então o ângulo de inclinação de uma partícula pode ser representado pelo valor do ângulo agudo formado entre o eixo maior da elipse que descreve a trajetória desta partícula e um vetor planar que indica a direção de propagação das ondas mecânicas. Esta informação, juntamente com a relação de aspecto, pode indicar a ocorrência de sobreposição ou domínio dos tipos de ondas. Por exemplo, se o ângulo de inclinação e a relação de aspecto de uma partícula forem nulos, então é possível afirmar que há predominância de ondas P no sentido da propagação das ondas.

Desta maneira, dado que \vec{M}_i (ver Equação 4.16) é um vetor paralelo ao eixo maior da elipse e \vec{e}_i é um vetor planar que indica o sentido de propagação das ondas, então o ângulo de inclinação θ_i de uma amostra i é o ângulo agudo ($0 \leq \theta_i \leq \pi/2$) entre eles:

$$\theta_i = \begin{cases} \cos^{-1} \left(\frac{|\vec{M}_i \cdot \vec{e}_i|}{\|\vec{M}_i\| \cdot \|\vec{e}_i\|} \right) & \text{se } \vec{M}_i \cdot \vec{e}_i > 0 \\ \pi - \cos^{-1} \left(\frac{|\vec{M}_i \cdot \vec{e}_i|}{\|\vec{M}_i\| \cdot \|\vec{e}_i\|} \right) & \text{se } \vec{M}_i \cdot \vec{e}_i \leq 0 \end{cases} \quad (4.15)$$

$$\vec{M}_i = \begin{cases} \vec{d}_{t_1, i}, & \|\vec{d}_{t_1, i}\| > \|\vec{d}_{t_2, i}\| \\ \vec{d}_{t_2, i}, & \|\vec{d}_{t_1, i}\| \leq \|\vec{d}_{t_2, i}\| \end{cases} \quad (4.16)$$

A Figura 4.8 ilustra o ângulo de inclinação da trajetória de uma amostra i para o caso 2D. Neste caso, \vec{e}_i é paralelo ao eixo de coordenadas y .

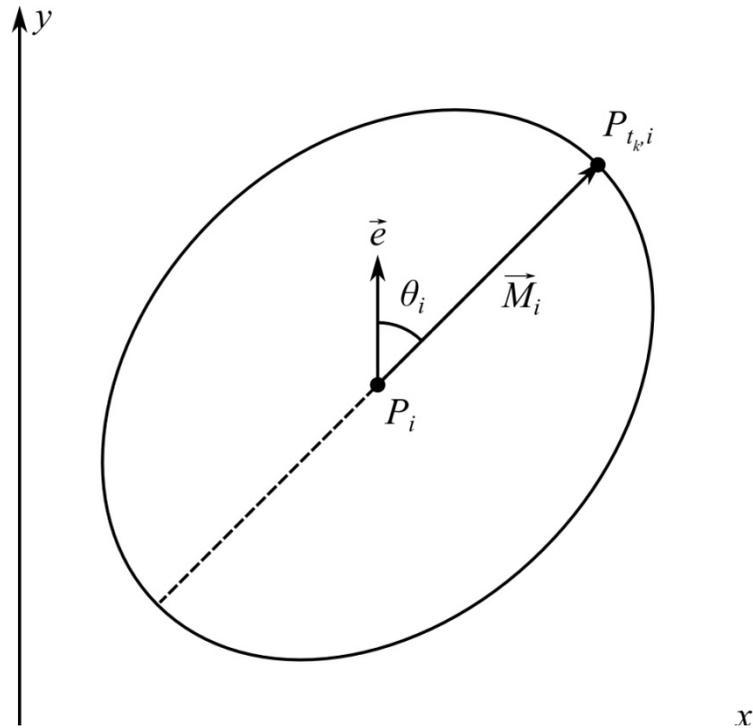


Figura 4.8 – Ângulo de inclinação da trajetória de uma amostra para o caso 2D.

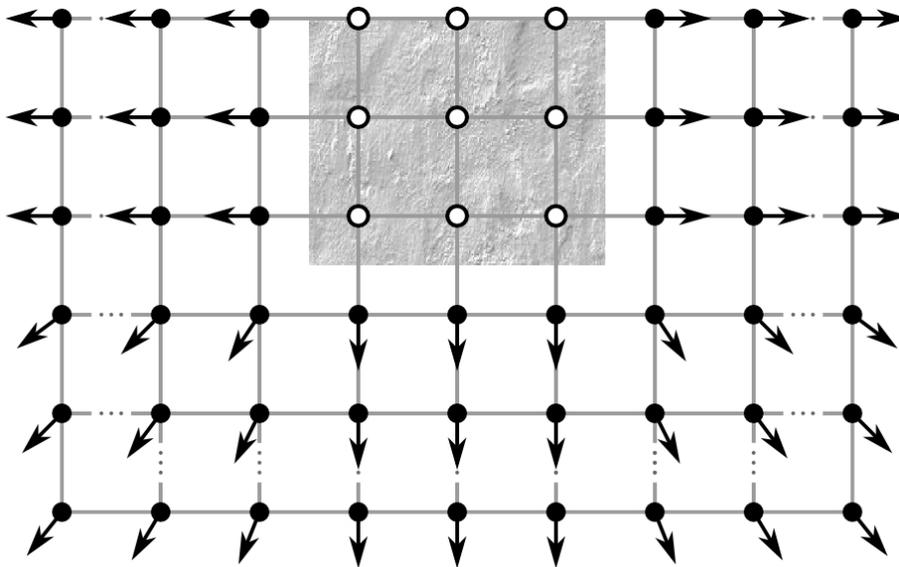


Figura 4.9 – Sentido de propagação das ondas a partir de um corpo rígido.

A Figura 4.9 ilustra os vetores planares \vec{e}_i que indicam o sentido de propagação das ondas a partir de um corpo rígido (região texturizada). Estes vetores podem ser calculados pela Equação

4.17, em que P_s é um dos pontos contidos no corpo rígido mais próximo de P_i . Para P_i contido no corpo rígido, o cálculo de \vec{e}_i deve ser menosprezado.

$$\vec{e}_i = \frac{P_i - P_s}{\|P_i - P_s\|} \quad (4.17)$$

4.2.6.2 Ondas mecânicas 3D

Para o caso das ondas mecânicas 3D, como as partículas não se deslocam apenas sobre o plano de propagação das ondas mecânicas, então o ângulo de inclinação de uma partícula pode ser representado pelo valor do ângulo formado entre os planos da trajetória das partículas e do meio de propagação. Esta informação pode indicar se o plano da elipse que descreve a trajetória de uma partícula se encontra perpendicular ao plano do meio de propagação ou paralelo ao plano do meio de propagação e, também, o sentido do movimento de rotação. Se o ângulo de inclinação for reto, então a elipse se encontra perpendicular; caso contrário, se o ângulo de inclinação for nulo ou igual a π , então a elipse se encontra paralela e o sentido do movimento de rotação é anti-horário quando este valor é nulo e horário quando este valor é igual a π .

Sendo assim, sabendo-se que \vec{n}_i e \vec{n}_α são, respectivamente, os vetores normais dos planos do meio de propagação das ondas mecânicas e da trajetória de uma partícula, então o ângulo de inclinação θ_i de uma amostra i é dado por:

$$\theta_i = \cos^{-1} \left(\frac{|\vec{n}_\alpha \cdot \vec{n}_i|}{\|\vec{n}_\alpha\| \cdot \|\vec{n}_i\|} \right) \quad (4.18)$$

A Figura 4.10 ilustra o ângulo de inclinação da trajetória de deslocamento de uma amostra i para o caso 3D.

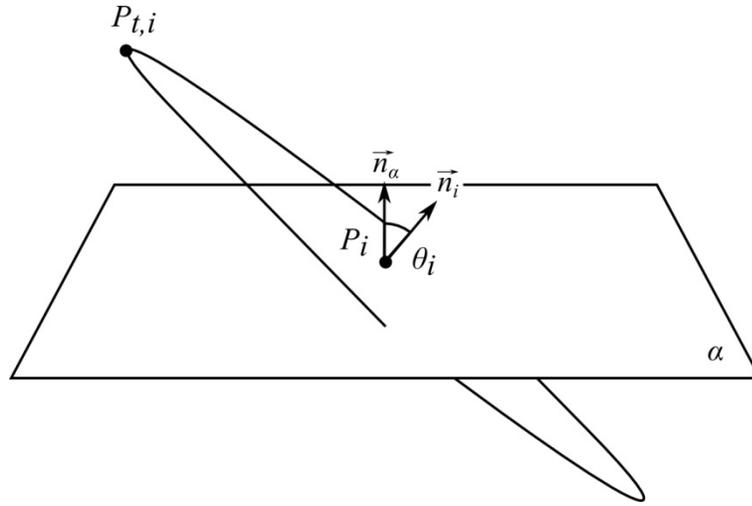


Figura 4.10 – Ângulo de inclinação da trajetória de uma amostra para o caso 3D.

4.2.7 Amplitude das ondas

As amplitudes das ondas que se propagam na direção dos eixos de coordenadas são equivalentes aos deslocamentos máximos das partículas, que correspondem ao módulo das componentes de deslocamento. A Equação 4.19 apresenta as amplitudes de uma amostra i sobre cada um dos eixos de coordenadas.

$$A_i = (A_{x,i}, A_{y,i}, A_{z,i}) \Rightarrow \begin{cases} A_{x,i} = \text{máx}(\|d_{x,t,i}\|) = |u_x| = \sqrt{\Re(u_x)^2 + \Im(u_x)^2} \geq 0 \\ A_{y,i} = \text{máx}(\|d_{y,t,i}\|) = |u_y| = \sqrt{\Re(u_y)^2 + \Im(u_y)^2} \geq 0 \\ A_{z,i} = \text{máx}(\|d_{z,t,i}\|) = |u_z| = \sqrt{\Re(u_z)^2 + \Im(u_z)^2} \geq 0 \end{cases} \quad (4.19)$$

A Figura 4.11 apresenta as amplitudes das ondas sobre os eixos de coordenadas x e y . Dado que $A_{x,i}$ e $A_{y,i}$ são as amplitudes da amostra i , tem-se então que o retângulo formado pelas retas que interceptam perpendicularmente os eixos de coordenadas em $x_{\text{mín},i}$, $x_{\text{máx},i}$, $y_{\text{mín},i}$ e $y_{\text{máx},i}$ é o delimitador da elipse.

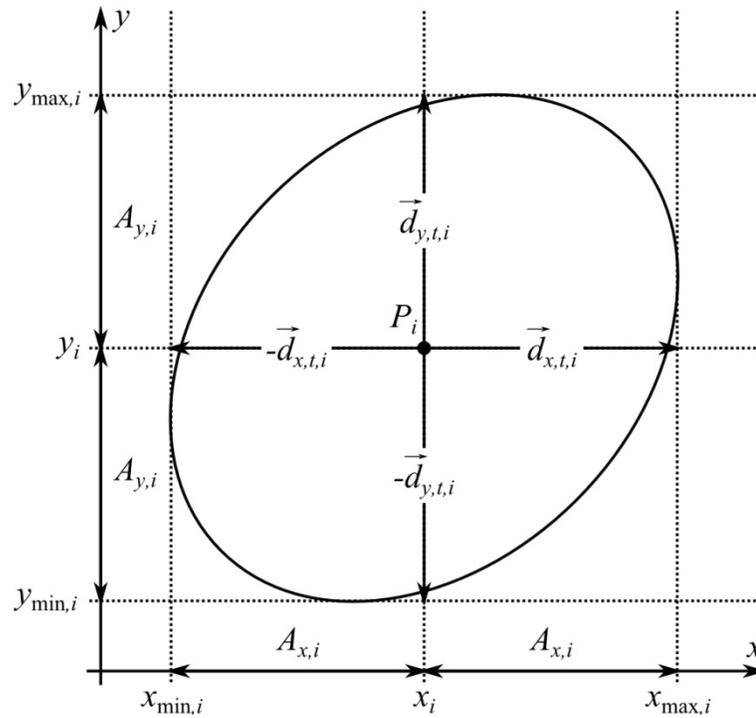


Figura 4.11 – Amplitudes das ondas sobre os eixos de coordenadas x e y .

4.2.8 Ângulo de fase das ondas

A defasagem entre as componentes de deslocamento é a responsável para que as trajetórias das partículas sejam elípticas. A Equação 4.20 apresenta as formulações para computo das fases de uma amostra i sobre cada um dos eixos.

$$\begin{aligned}
 \phi_{x,i} &= \tan^{-1} \left[\frac{\Im(u_{x,i})}{\Re(u_{x,i})} \right] & \text{para } -\pi \leq \phi_{x,i} \leq \pi \\
 \phi_{y,i} &= \tan^{-1} \left[\frac{\Im(u_{y,i})}{\Re(u_{y,i})} \right] & \text{para } -\pi \leq \phi_{y,i} \leq \pi \\
 \phi_{z,i} &= \tan^{-1} \left[\frac{\Im(u_{z,i})}{\Re(u_{z,i})} \right] & \text{para } -\pi \leq \phi_{z,i} \leq \pi
 \end{aligned} \tag{4.20}$$

4.2.9 Fator de escala

Quando as oscilações das partículas são de baixa intensidade fica difícil visualizar com clareza os fenômenos de propagação das ondas. Por isso, com uso de um valor escalar k , denominado de fator de escala, é possível amplificar esta intensidade, que conseqüentemente aumenta o grau de deformação do meio de propagação das ondas. Este escalar atua como um fator de multiplicação sobre o vetor de deslocamento $\vec{d}_{t,i}$, de tal maneira que $\{k \in \mathbb{R} | k \geq 1\}$, assim como apresentado na Equação 4.21.

$$P_{k,t,i} = P_i + k \vec{d}_{t,i} \quad (4.21)$$

A Figura 4.12 ilustra as trajetórias de uma amostra com e sem o uso do fator de escala. Note que a elipse que intersecta o ponto $P_{k,t,i}$ é maior do que a elipse que intersecta o ponto $P_{t,i}$ com uso de $k > 1$.

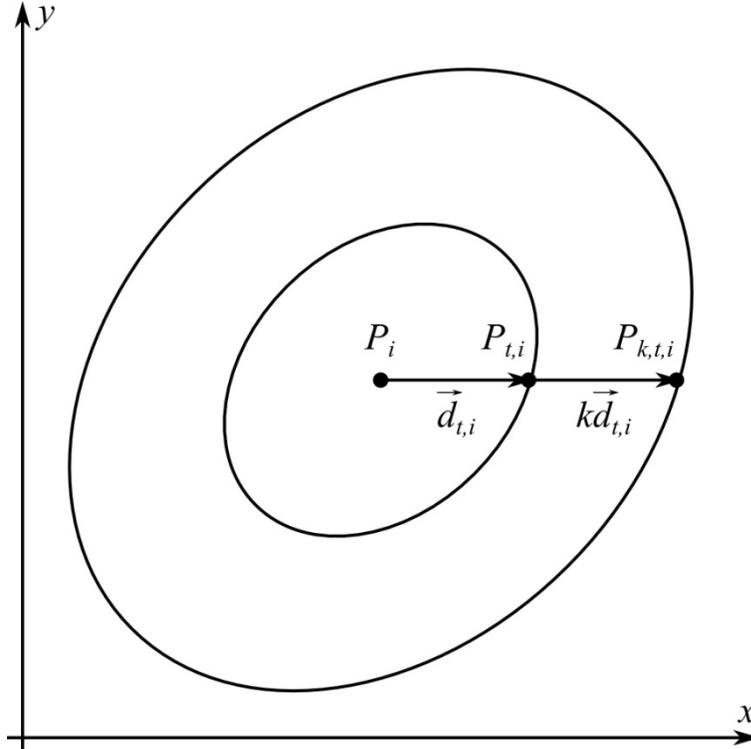


Figura 4.12 – Ampliação da intensidade de oscilação de uma amostra com uso do fator de escala.

4.2.10 Sobreposição de ondas

A simulação computacional é um processo relativamente custoso e demorado. O tempo requerido é dependente da capacidade de processamento dos computadores utilizados e dos valores dos parâmetros de cada simulação, como o número de camadas presente no meio de propagação, os valores dos parâmetros constitutivos das camadas elásticas e o número de pontos de amostragem. Em alguns casos muito complexos, que exigem excessivo tempo de processamento, as simulações podem ser, se aplicável, divididas em casos mais simples, que teoricamente requerem menos tempo de processamento, ou mesmo reaproveitadas. Quando uma simulação é equivalente a junção de outras simulações já realizadas, então esta pode ser poupada.

A junção de duas ou mais simulações é equivalente a sobreposição de ondas. Esta sobreposição é equivalente a soma dos valores das componentes de deslocamento. A Equação 4.22 apresenta uma junção de duas simulações (A e B) com mesmas dimensões ($n_x \times n_y \times n_z$). O resultado desta soma é armazenado em C, que é equivalente a simulação poupada.

$$\begin{aligned} u_{x,i}^C &= u_{x,i}^A + u_{x,i}^B \\ u_{y,i}^C &= u_{y,i}^A + u_{y,i}^B \\ u_{z,i}^C &= u_{z,i}^A + u_{z,i}^B \end{aligned} \quad (4.22)$$

Nos casos lineares, uma mesma simulação pode ser usada para sobreposição de ondas. Nestes casos, a sobreposição é realizada por meio de reflexão (ou espelhamento) da grade uniforme que representa o conjunto de dados. As Equações 4.23, 4.24 e 4.25 apresentam as formulações para espelhamento de uma grade em relação, respectivamente, aos eixos de coordenadas x , y e z .

$$i_x = \left(2 \frac{i}{n_x} + 1 \right) n_x - (i + 1) \quad (4.23)$$

$$i_y = \left(n_y - 2 \frac{i - \frac{i}{n_x n_y} n_x n_y}{n_x} - 1 \right) n_x + i \quad (4.24)$$

$$i_z = \left(n_z - 2 \frac{i}{n_x n_y} - 1 \right) n_x n_y + i \quad (4.25)$$

4.2.11 Gradiente da amplitude das ondas

Os vetores gradientes das amplitudes das ondas podem ser usados, por exemplo, para visualização das trajetórias das partículas livres ao longo do meio de propagação. A Equação 4.26 apresenta esta operação, que inclusive pode ser aplicada sobre outros atributos de dados escalares. Note que na equação não foram considerados os limites da grade. A função $d(\bullet)$ representa a distância entre as posições de repouso de dois pontos de amostragem.

$$G_A = (g_x, g_y, g_z) \Rightarrow \begin{cases} g_x = \frac{A_{x,i+1} - A_{x,i-1}}{d(P_{i+1}, P_{i-1})} \\ g_y = \frac{A_{x,i+n_x} - A_{x,i-n_x}}{d(P_{i+n_x}, P_{i-n_x})} \\ g_z = \frac{A_{x,i+n_x n_y} - A_{x,i-n_x n_y}}{d(P_{i+n_x n_y}, P_{i-n_x n_y})} \end{cases} \quad (4.26)$$

4.3 FILTRAGEM

O objetivo da filtragem de dados é a extração de informações relevantes de um conjunto de dados qualquer (ver Seção 2.3.2). Um dos conceitos empregados para esta finalidade é a **extração de dados**, cujos objetivos são a redução do tamanho de um conjunto de dados e a limita-

ção de uma região de interesse. A redução do tamanho de um conjunto de dados requer menos uso de processamento e memória, conseqüentemente, contribuindo com a melhora da interatividade, já que o tempo de resposta também é reduzido. E a limitação de uma região de interesse pode melhorar a eficácia da visualização, já que esta reduz o tamanho de um conjunto de dados e também torna mais rápido a navegação por eles.

A extração de dados pode ser classificada de duas formas: **extração geométrica** e **extração espacial**. A **extração geométrica** consiste na extração de um conjunto de pontos ou células com base em um ou mais intervalos de seus índices. A técnica de extração de subgrade é um exemplo desta forma de extração de dados (ver Seção 2.6.1). A **extração espacial** é a dada a partir da intersecção com uma figura geométrica. Suas técnicas consistem na seleção de pontos ou células que estejam contidas no interior ou exterior da região de intersecção da grade que representa o conjunto de dados com uma figura geométrica que possa ser descrita implicitamente. As técnicas de recorte e fatiamento das Seções 2.6.2 e 2.6.3 são exemplos desta forma de extração (SCHROEDER; MARTIN; LORENSEN, 2006).

A respeito do conjunto de dados, tal como o produzido pelo GMC MC, em alguns casos, o especialista está interessado em visualizar somente uma região específica da grade. Por exemplo, quando ele precisa analisar o comportamento das ondas sobre determinadas camadas do meio de propagação. Nestas situações, as técnicas de extração de subgrade e de recorte podem ser empregadas. A Figura 4.13 ilustra a aplicação da primeira técnica, no qual a área destacada em cinza representa a região de interesse do especialista.

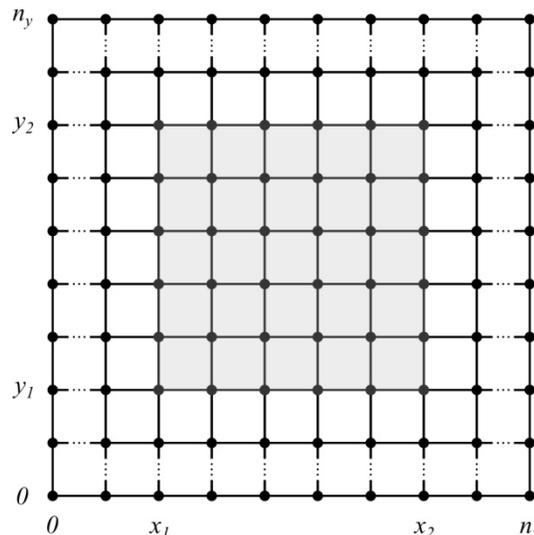


Figura 4.13 – Extração de dados por extração de subgrade.

A Figura 4.14 exemplifica o emprego da técnica de recorte. As áreas em cinza correspondem às regiões de interesse geradas, respectivamente, pela intersecção com um plano, um paralelepípedo e uma esfera (da esquerda para direita). As áreas externas à intersecção são apresentadas logo abaixo.

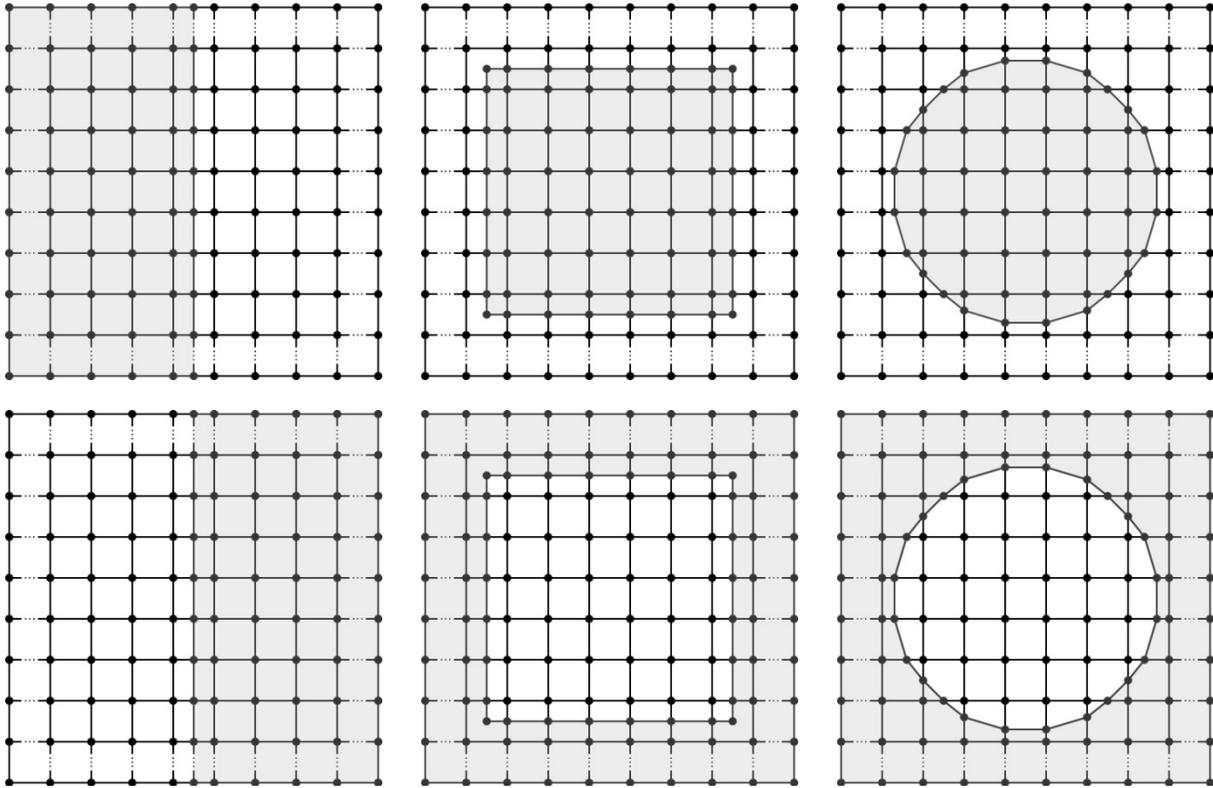


Figura 4.14 – Extração de dados por recorte com um plano, um paralelepípedo e uma esfera.

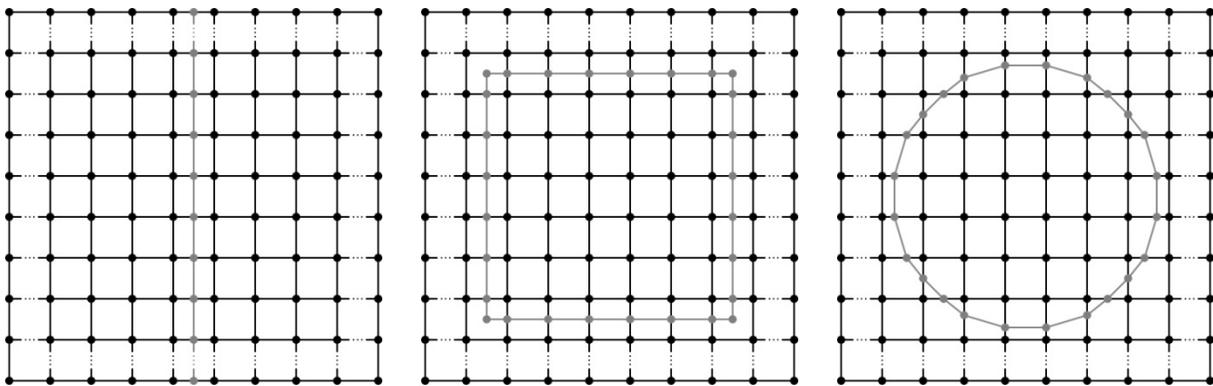


Figura 4.15 – Extração de dados por fatiamento com um plano, um paralelepípedo e uma esfera.

A técnica de fatiamento pode ser empregada, por exemplo, para análise do deslocamento de um subconjunto de amostras ao longo do tempo. A Figura 4.15 apresenta as linhas de contorno (em cinza) geradas, respectivamente, pela intersecção com um plano, um paralelepípedo e uma esfera (da esquerda para direita).

4.4 MAPEAMENTO

Na etapa de mapeamento de dados, as técnicas de visualização de dados escalares e vetoriais apresentadas nas Seções 2.4 e 2.5, respectivamente, podem ser empregadas para a visualização dos atributos de dados produzidos na fase de enriquecimento de dados descrito na Seção 4.2. O Quadro 4.1 representa quais as técnicas de visualização que podem (Sim) ou não podem (Não) ser aplicadas sob cada um dos atributos de dados enriquecidos. Nesta seção não há preocupação com a eficiência das técnicas para a análise das informações, apenas se são aplicáveis. Na Seção 7.2 são apresentadas quais destas técnicas de visualização são mais eficientes ou não com base nas informações extraídas dos estudos de caso realizado no Capítulo 6.

Quadro 4.1 – Relação de aplicação das técnicas de visualização com as informações enriquecidas.

Técnicas		Dados escalares			Dados vetoriais	
		Mapeamento em cores	Contorno	Função de altura	Glifos de vetor	Linhas de fluxo
Deslocamento		<i>Sim</i>	<i>Sim</i>	<i>Sim</i>	<i>Sim</i>	<i>Sim</i>
Trajetória		<i>Não</i>	<i>Não</i>	<i>Não</i>	<i>Não</i>	<i>Sim</i>
Relação de aspecto		<i>Sim</i>	<i>Sim</i>	<i>Sim</i>	<i>Não</i>	<i>Não</i>
Sentido do movimento de rotação	2D	<i>Sim</i>	<i>Sim</i>	<i>Sim</i>	<i>Não</i>	<i>Não</i>
	3D	<i>Não</i>	<i>Não</i>	<i>Não</i>	<i>Sim</i>	<i>Sim</i>
Ângulo de inclinação	2D	<i>Sim</i>	<i>Sim</i>	<i>Sim</i>	<i>Não</i>	<i>Não</i>
	3D	<i>Sim</i>	<i>Sim</i>	<i>Sim</i>	<i>Não</i>	<i>Não</i>
Amplitude das ondas		<i>Sim</i>	<i>Sim</i>	<i>Sim</i>	<i>Sim</i>	<i>Sim</i>
Ângulo de fase das ondas		<i>Sim</i>	<i>Sim</i>	<i>Sim</i>	<i>Não</i>	<i>Não</i>
Gradiente da amplitude das ondas		<i>Não</i>	<i>Não</i>	<i>Não</i>	<i>Sim</i>	<i>Sim</i>

O deslocamento pode ser visualizado pelas cinco técnicas de visualização, sendo que: os deslocamentos ao longo dos eixos de coordenadas ($\vec{d}_{x,t,i}$, $\vec{d}_{y,t,i}$ e $\vec{d}_{z,t,i}$) e os módulos do vetor de deslocamento ($\|\vec{d}_{t,i}\|$) ou das combinações de suas componentes, que são dados escalares, podem ser visualizados pelas técnicas destinadas aos dados escalares; e os vetores de deslocamento ($\vec{d}_{t,i}$), podem ser visualizados pelas técnicas destinadas aos dados vetoriais. A amplitude das ondas mantém a mesma lógica do deslocamento.

A trajetória é representada por uma figura geométrica que pode ser uma reta, uma elipse ou uma circunferência. A linhas de fluxo, ou mais precisamente, a técnica de trajetória de partículas (ver Seção 2.5.2), pela sua finalidade é a técnica que pode ser aplicada para visualização da trajetória das partículas. A relação de aspecto (r), por ser um dado escalar, pode ser visualizada pelas técnicas destinadas aos dados escalares.

O sentido do movimento de rotação pode ser representado e interpretado de duas maneiras. Para as ondas 2D, este é representado por um valor escalar, portanto as técnicas de visualização de dados escalares podem ser usadas para representá-lo. Para as ondas 3D, este é representado por um valor vetorial, portanto as técnicas de visualização de dados vetoriais podem ser usadas para representá-los.

O ângulo de inclinação também pode ser representado e interpretado de duas formas. Contudo, os dois casos são representados por um valor escalar e, portanto, as técnicas de visualização de dados escalares podem ser usadas para representá-lo. Os ângulos de fase das ondas são dados escalares e, portanto, podem ser visualizados por técnicas para dados escalares. E o gradiente da amplitude das ondas, por serem dados vetoriais, podem ser representados por técnicas para dados vetoriais.

4.5 RENDERIZAÇÃO

A seguir são apresentadas algumas técnicas e procedimentos que podem auxiliar os especialistas no processo de interpretação do conjunto de dados:

Interação de usuário

A interação de usuário possibilita que o especialista interaja em tempo real com a visualização. A interação pode ocorrer por reposicionamento do observador ou pela alteração das técnicas aplicadas para mapeamento dos atributos de dados.

Animação

As imagens (renderizações) dinâmicas ou as animações favorecem a percepção temporal. Podem ser aplicada para dar a alusão do efeito de propagação das ondas mecânicas e para facilitar a interpretação dos atributos que variam com o decorrer do tempo, tal como os deslocamentos e as posições relativas.

Reposicionamento do observador

O reposicionamento do observador permite que o investigador navegue pela grade. Esta navegação é dada pela capacidade de ampliação ou redução da grade e pelo deslocamento sobre ela. No caso das ondas mecânicas 3D é interessante que o investigador possa ver as visualizações produzidas em diferentes perspectivas.

Visões múltiplas

As visões múltiplas consistem na subdivisão da tela em mais de uma parte. Podem ser empregadas em dois casos: para ver uma mesma visualização em perspectivas diferentes ou para ver diferentes visualizações, cujos atributos sejam relacionados. Este último caso pode ser justificado pelo simples fato de que a visualização de diversos atributos em uma mesma tela pode prejudicar a interpretação.

Visão estereoscópica

A visão estereoscópica pode auxiliar a visualização de conjunto de dados com geometria 3D.

Medidas da grade

A apresentação das medidas da grade pode auxiliar os especialistas na localização de regiões de interesse.

Distância entre dois pontos geométricos

Por intermédio da distância entre dois pontos é possível obter a velocidade de propagação das ondas.

4.6 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentado o fluxo de visualização proposto para dados que descrevem o comportamento das ondas em meios contínuos, tais como os gerados pelo GMCMC por meio da simulação computacional destes fenômenos (Seção 3.1). Este fluxo de visualização foi implementado em um ambiente de visualização específico para os dados tratados neste trabalho. No capítulo seguinte, detalhes de desenvolvimento deste ambiente são descritos. O conteúdo apresentado na Seção 4.2 será constantemente exigido nos Capítulos 6 e 7.

5 AMBIENTE DE VISUALIZAÇÃO DE ONDAS MECÂNICAS

Neste capítulo é descrito o protótipo de ambiente de visualização desenvolvido para auxiliar os especialistas na etapa de análise dos dados gerados por simulação computacional de problemas geomecânicos (ver Seção 1.1). Neste ambiente é implementado o fluxo de visualização proposto para dados que descrevem o comportamento das ondas em meios contínuos, como informado no Capítulo 4. Ao longo deste capítulo são descritas as motivações para o desenvolvimento do ambiente de visualização, assim como as ferramentas usadas para sua produção. Também é apresentada uma descrição lógica do funcionamento do sistema, de maneira que sirva de documentação para futuros desenvolvedores.

5.1 INTRODUÇÃO

Antes de iniciar o desenvolvimento de um ambiente específico para a visualização e análise dos dados produzidos por simulação computacional dos fenômenos de propagação de ondas mecânicas, uma pesquisa por sistemas que pudessem ser usados pelos especialistas para a visualização de seus resultados foi realizada. Por se tratar de um problema específico, nenhum aplicativo para este caso em particular foi encontrado – o que de fato era esperado pelos relatos dos integrantes do GMCMC. A partir disto, a procura foi estendida para um aplicativo mais genérico, no qual um se destacou dos demais: o ParaView produzido pela Kitware¹².

O **ParaView** é um aplicativo de propósito geral dedicado à visualização e análise de dados científicos. É capaz de processar grandes conjuntos de dados em ambientes paralelos (assim como em sistemas de monoprocessador), além de apresentá-los em ambientes virtuais imersivos, como cavernas digitais ou mosaicos de monitores (ou de painéis para projeção). Além de ser multiplataforma, também é gratuito e com código-fonte de livre distribuição¹³ (AHRENS; GEVECI; LAW, 2005). Atualmente, o ParaView é patrocinado por conceituados laboratórios de pesquisas dos Estados Unidos da América (EUA): Computação e Simulação Avançadas (ASC – *Advanced Simulation and Computing*), Laboratório Nacional de Los Alamos (*Los Alamos National Lab*),

¹² Para mais informações, consulte o *website* da Kitware, no endereço <http://www.kitware.com>.

¹³ Expressão traduzida do inglês *open source*.

Laboratórios Nacionais de Sandia (*Sandia National Laboratories*) e Laboratório de Pesquisa do Exército (*Army Research Laboratory*)¹⁴.

Considerando que os especialistas precisam economizar o tempo gasto com as demandas de conhecimento em técnicas de visualização para que possam se concentrar no processo de análise dos resultados (ver Seção 1.1), assim como o fato do ParaView ser um ambiente de propósito geral, então foram identificadas as deficiências deste aplicativo para uso dos especialistas: no contexto deste trabalho, a implementação das formulações para enriquecimento dos dados (ver Seção 4.2) não é trivial e exigiria ao menos o desenvolvimento de um programa que realizasse esta etapa automaticamente; e há exigência de que o seu usuário tenha um conhecimento do funcionamento de um fluxo de visualização e das ligações e dependências entre as funcionalidades de visualização disponíveis no ParaView, chamadas de filtros.

A partir destas deficiências relatadas, foi decidido pela opção de se desenvolver um ambiente de visualização específico para o problema em estudo, o que ocasionou na implementação do Visualizador de Propagação de Ondas Mecânicas (**MWPVis** – *Mechanical Wave Propagation Visualizer*). Apesar disto, vale expor, que não é propósito deste trabalho fazer comparações entre os ambientes ParaView e MWPVis, já que ambos são de propósitos diferentes: respectivamente de propósitos geral e específico.

É importante informar que o ParaView foi usado ao longo do desenvolvimento deste projeto de pesquisa para testar e comparar os seus resultados, e a Interface Gráfica com o Usuário (GUI – *Graphical User Interface*) e as funcionalidades de visualização disponíveis no mesmo serviram de inspiração para a implementação do MWPVis. O MWPVis, inclusive, permite que o seu usuário converta os dados produzidos para um formato que possa ser interpretado pelo ParaView, caso ele já esteja familiarizado com esta ferramenta ou necessite futuramente explorá-los em um ambiente de visualização que é constantemente atualizado.

5.2 AMBIENTE DE DESENVOLVIMENTO

Para a implementação do MWPVis, foi pré-determinado que o seu ambiente de desenvolvimento (ou seja, as ferramentas e/ou tecnologias utilizadas pelos desenvolvedores) seria essencial-

¹⁴ PARAVIEW. **Participants**. Disponível em: <<http://www.paraview.org/paraview/project/parti.html>>. Acesso em: 19 jan. 2011.

almente composto por uma linguagem de programação de alto nível e por duas bibliotecas de classes, sendo uma para a visualização de dados científicos e outra para a construção das interfaces gráficas; e também que estes fossem preferencialmente orientados a objetos e multiplataformas, além de obrigatoriamente gratuitos.

Devido à variedade de linguagens de programação e de bibliotecas que poderiam ser aplicadas no ambiente de desenvolvimento, também foram estabelecidos pré-requisitos que deveriam ser considerados para as escolhas dos mesmos, que são: uma linguagem de programação que fosse de fácil entendimento e que poupasse o tempo dedicado à produção do aplicativo; uma biblioteca para visualização de dados científicos que fornecesse um extenso conjunto de técnicas de visualização; e uma biblioteca de construção de GUI compatível com a anterior, de maneira que as renderizações produzidas pela primeira biblioteca pudessem ser apresentadas através de janelas gerenciadas por esta.

Enfim, nas subseções seguintes (de 5.2.1 a 5.2.3) são apresentadas a linguagem de programação e as bibliotecas adotadas para o desenvolvimento. Mesmo com os pré-requisitos estabelecidos, o processo de configuração do ambiente de desenvolvimento não foi simples e rápido, dado que este era cíclico – pois foi constantemente necessário verificar (pesquisar e/ou testar) se as bibliotecas eram compatíveis entre si e também com a linguagem de programação pretendida.

5.2.1 Linguagem de programação

O **Python**¹⁵ é uma linguagem de programação de propósito geral. Por ser completamente orientado a objetos, seu modelo de classes suporta noções avançadas, como polimorfismo, sobrecarga de operadores e herança múltipla. Além de ser multiplataforma, também permite a integração com outras linguagens, tal como C, C++, Java ou Fortran. Dispõe de ferramentas de engenharia de *software* avançadas, normalmente encontradas nas linguagens compiladas, como tipagem dinâmica, gerenciamento de memória automático, suporte para programação em grande escala, tipos de objeto incorporados, ferramentas incorporadas, bibliotecas e utilitários de outros fornecedores etc. Por estar sob licença *Open Source*, seu uso e distribuição são gratuitos, até mesmo comercialmente (LUTZ, 2008).

¹⁵ Para mais informações, consulte o *website* do Python, no endereço <http://www.python.org>.

A partir de Lutz (2008), três características do Python foram consideradas como justificativas para sua adoção. A principal delas é o fato desta linguagem ser deliberadamente otimizada para agilidade no desenvolvimento – sua sintaxe simples, tipagem dinâmica, ausência de etapas de compilação e seu conjunto de ferramentas incorporado, permitem que os programadores criem programas em uma fração do tempo de desenvolvimento necessário para algumas outras ferramentas. O resultado é que o Python aumenta a produtividade do desenvolvedor muitas vezes além do que se consegue com as linguagens tradicionais.

A facilidade de aprendizado e uso do Python também foi um fator influenciador. Por não haver etapas de compilação e vinculações, como acontece nas linguagens compiladas, o Python executa imediatamente – os seus módulos são carregados dinamicamente, quando importados por um programa em execução. Entretanto, o ciclo de desenvolvimento é apenas um aspecto da facilidade de uso do Python. Ele também fornece uma sintaxe deliberadamente simples e legível, e um modelo de programação altamente coerente, além de poderosas ferramentas de alto nível incorporadas, o que também torna o Python uma linguagem fácil de aprender.

Outra justificativa à adoção do Python é a de que ele permite que o MWPVis seja posteriormente transcrito para uma linguagem compilada, caso futuramente haja exigências de desempenho ou necessidade de integração. Os sistemas podem ser implementados primeiro em Python para aumentar sua velocidade no desenvolvimento e, em seguida, transcritos para C ou outra linguagem compilada, como C++ ou Fortran.

5.2.2 Biblioteca de classes para visualização de dados científicos

O Kit de Ferramentas de Visualização (**VTK** – *Visualization Toolkit*) é um kit de ferramentas de código-fonte de livre distribuição destinado à computação gráfica, processamento de imagens e visualização de dados. É composto por uma biblioteca de classes desenvolvida em C++ e por diversas camadas de interfaces interpretadas (conhecidas como *wrapper*) para as linguagens de programação Tcl, Java e Python. Dispõe de uma extensa variedade de algoritmos para visualização de dados escalares, vetoriais, tensoriais e volumétricos; além de técnicas avançadas para modelagem, como modelagem implícita, redução de polígonos, suavização de malhas, recorte, contorno e triangulação de Delaunay. Suporta processamento paralelo e tem integração com

programas de construção de GUI, como Qt e Tk. Por ser multiplataforma, pode ser executado em ambientes Linux, Windows, Mac e Unix (KITWARE, 2006).

Como já informado (Seção 5.2.1), o ParaView serviu de inspiração para o desenvolvimento do MWPVis, e o fato dele usar o VTK foi um ponto a favor para a adoção desta ferramenta. Após um levantamento das vantagens de seu uso, as seguintes justificativas foram consideradas: é gratuito, é constantemente atualizado, possui *wrapper* para Python, é altamente recomendado em artigos científicos e livros da área de visualização de dados, e é utilizado e patrocinado por conceituados laboratórios de pesquisa dos EUA: Biblioteca Nacional de Medicina (NLM – *National Library of Medicine*), Aliança Nacional para Processamento de Imagens Médicas (NA-MIC – *National Alliance for Medical Image Computing*) e Departamento de Energia (DOE – *Department of Energy*), entre outros¹⁶.

5.2.3 Biblioteca de classes para construção e gerenciamento de GUI

O Qt é um arcabouço¹⁷ em C++ para desenvolvimento de aplicativos gráficos multiplataformas. A divisão Qt Development Frameworks¹⁸ da Nokia Corporation é a responsável pela atualização e distribuição oficial deste sistema, que fornece um rico conjunto de *widgets*¹⁹ (ou controles na terminologia Windows) que podem ser usados na criação de GUI para aplicativos, dentre outras funcionalidades gráficas (NOKIA, 2009). Entretanto, como o Qt não é distribuído para Python, foi necessário então usar uma biblioteca que fizesse está interface: o PyQt.

O **PyQt** é um *binding* do Python para Qt, ou seja, um conjunto de módulos que faz com que as bibliotecas de classes do Qt possam ser acessadas pelos programadores de Python. As plataformas atualmente suportadas incluem todas as versões do Windows, desde o Windows 98 ao Windows 7, e a maioria das variações e clones do Unix que executam X11, incluindo o Mac OS X, Linux, Solaris e HP-UX. Usando o PyQt, os desenvolvedores podem escrever aplicações com

¹⁶ VTK. **Participants**. Disponível em <<http://vtk.org/VTK/project/parti.html>>. Acesso em: 19 jan. 2011.

¹⁷ Termo traduzido do inglês *framework*.

¹⁸ Para mais informações, visite o *website* da Qt Development Frameworks, no endereço <http://qt.nokia.com>.

¹⁹ “*Widgets* são elementos visuais que podem ser combinados para criar interfaces gráficas. Botões, menus e barras de rolagem, além de janelas de caixas de mensagem e de aplicativos são exemplos de *widgets*.” (Nokia, 2009, p. 5, tradução nossa, grifo nosso).

*look and feel*²⁰ nativo em todas estas plataformas, sem ter que dedicar esforços para resolver problemas de incompatibilidade, uma vez que todos estes detalhes são tratados internamente pelo PyQt. A versão Licença Pública Geral (GPL – *General Public License*) do PyQt pode ser usada com ambas as versões Licença Pública Geral Reduzida (LGPL – *Lesser General Public License*) ou GPL do Qt (RIVERBANK, 2009).

Assim como no caso do VTK, o fato do ParaView ter trocado o KWWidgets (da própria desenvolvedora) pelo Qt para a construção de sua GUI serviu de ponto a favor para sua adoção. Entretanto, as seguintes vantagens também foram consideradas: pode ser gratuito (depende da licença adotada), possuiu um *binding* no Python, é atualizado com frequência e é usado por grandes organizações: Adobe, Boeing, Google, IBM, Motorola, NASA e Skype.

5.3 FUNCIONAMENTO DO AMBIENTE DE VISUALIZAÇÃO

A Figura 5.1 demonstra como a linguagem de programação e as bibliotecas foram empregadas em cada uma das etapas do fluxo de visualização proposto (ver Capítulo 4). As etapas de importação e enriquecimento são realizadas por classes escritas em Python; as etapas de filtragem, mapeamento e renderização são realizadas por meio do VTK; e, finalmente, a etapa de interface (meio pelo qual o usuário interage com o sistema, interferindo nas etapas anteriores) é realizada pelo Qt. Vale ressaltar que as duas primeiras etapas podem ser feitas pelo VTK, o que facilita a etapa de conversão do MWPVis para uma linguagem compilada (ver Seção 5.2.1). Por questões de facilidade de realização destes procedimentos no Python, foi optado por implementá-las.

O código-fonte do MWPVis contém 61 módulos (com 54 classes) organizados em três pacotes: `mwpvis.wanglay`, responsável pelas etapas de importação e enriquecimento; `mwpvis.vtk`, responsável pelas etapas de filtragem, mapeamento e renderização; e `mwpvis.qt`, responsável pela etapa de interface. Como o número de módulos é muito extenso e o objetivo desta seção é expor a lógica de funcionamento do sistema com ênfase na visualização, então apenas as classes consideradas de maior importância serão apresentadas.

²⁰ Em projeto (ou *design*) de *software*, *look and feel* compreende os aspectos da concepção de GUI, incluindo elementos como cores, formas, leiautes e fontes, assim como o comportamento dos elementos dinâmicos, como botões, caixas e menus.

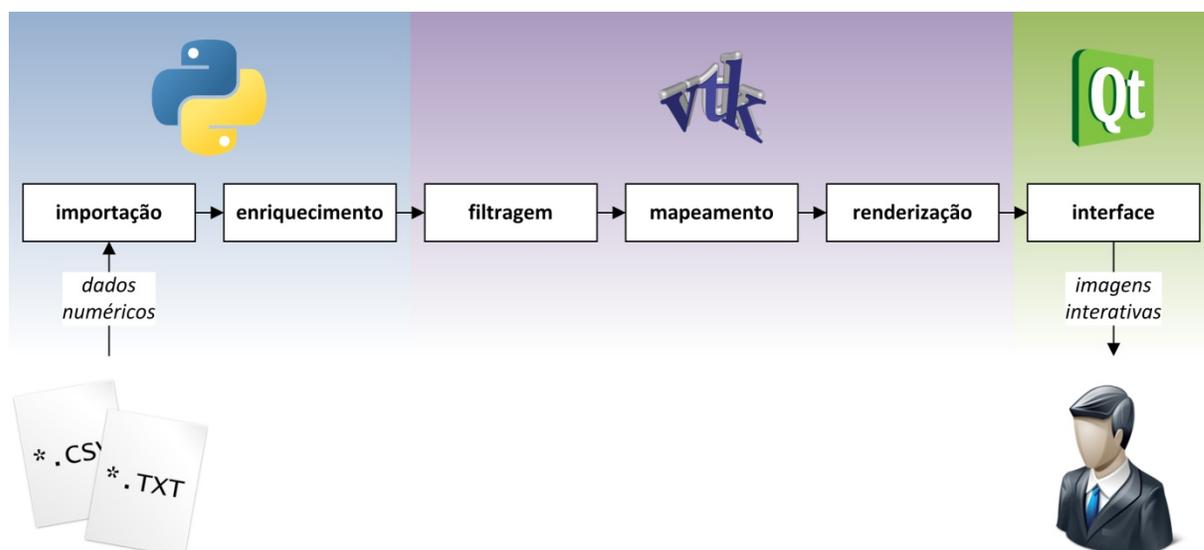


Figura 5.1 – Fluxo de visualização proposto implementado no MWPVis.

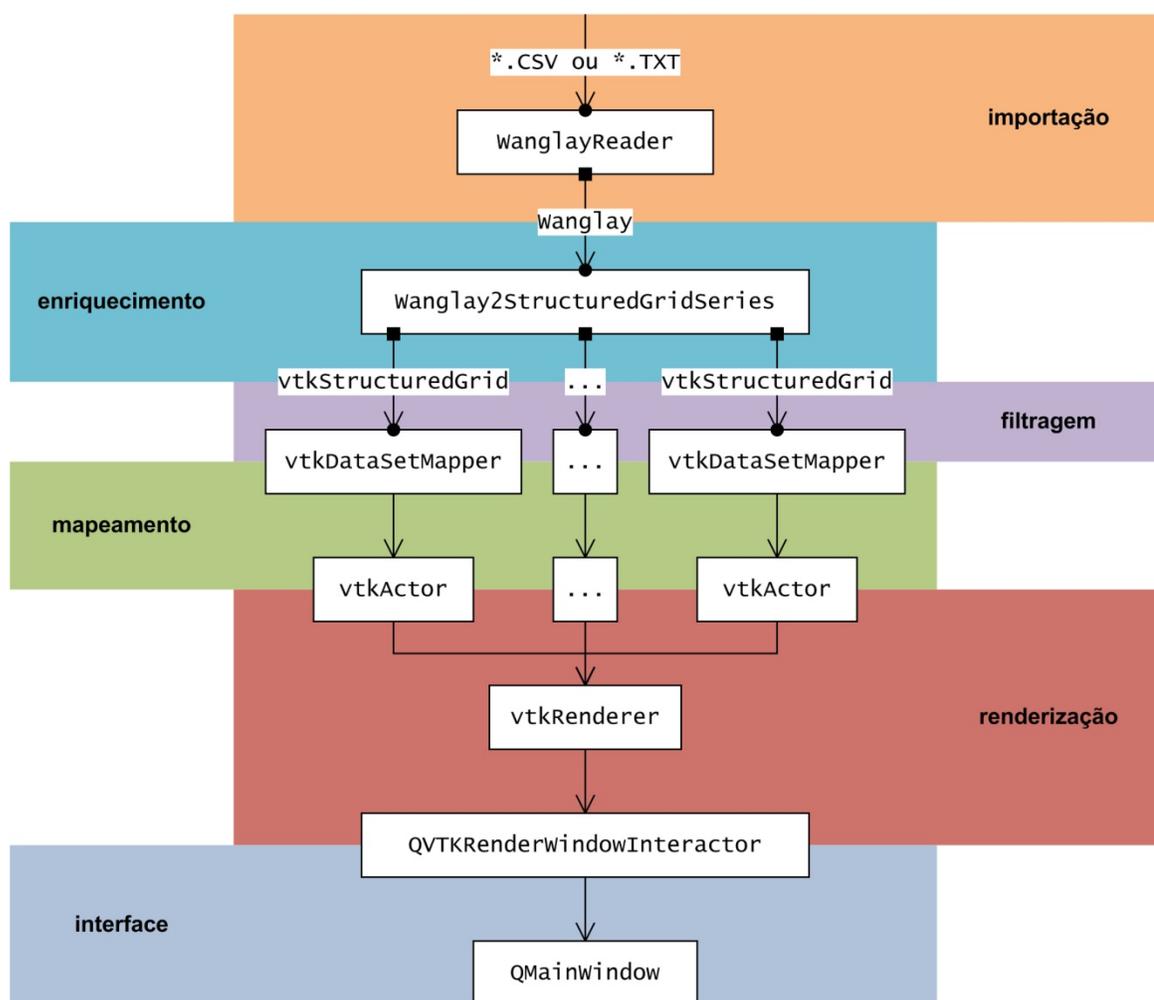


Figura 5.2 – Fluxo visualização padrão do MWPVis.

A Figura 5.2 apresenta o fluxo de visualização padrão criado pelo MWPVis, que é posteriormente incrementado conforme a interação e necessidade de visualização do usuário. Os retângulos brancos com contorno representam as classes e os sem contorno os dados. As linhas com setas representam a transferência de objetos instanciados ou dados. As entradas e saídas das classes são representadas, respectivamente, pelos círculos e quadrados pretos. As etapas do fluxo de visualização proposto (destacadas em colorido), assim como as funcionalidades de cada uma das classes são apresentadas nas subseções adiante. As classes usadas nas etapas de importação e enriquecimento foram criadas no contexto deste trabalho e as demais etapas são do VTK, com exceção da última que é do Qt.

5.3.1 Importação

A etapa de importação é realizada por uma instância da classe *WanglayReader* (ver Figura 5.2), que é responsável pela conversão dos dados presentes em arquivos de Valores Separados por Vírgula (CSV – *Comma-Separated Values*) ou texto sem formatação (TXT) para um objeto *Wanglay*, que armazena as coordenadas das posições de repouso e as componentes de deslocamento das amostras, além de informações adicionais: o número total de amostras, os intervalos de variação das coordenadas e as dimensões da grade estruturada (ver Seção 4.1).

x	z	RUx	IUx	RUz	IUz
-50.1000	0.0000	-.40613E-01	-.63326E-02	-.16594E-01	.55740E-01
-49.1000	0.0000	-.17026E-01	-.35387E-01	-.61315E-01	.15034E-01
-48.1000	0.0000	.19522E-01	-.29495E-01	-.47621E-01	-.46275E-01
-47.1000	0.0000	.31464E-01	.50445E-02	.13965E-01	-.65672E-01
-46.1000	0.0000	.58190E-02	.31513E-01	.62267E-01	-.21279E-01
-45.1000	0.0000	-.30884E-01	.20050E-01	.47154E-01	.43172E-01
-44.1000	0.0000	-.38537E-01	-.20333E-01	-.17278E-01	.61064E-01
-43.1000	0.0000	-.51283E-02	-.48961E-01	-.64395E-01	.11863E-01
-42.1000	0.0000	.38826E-01	-.34466E-01	-.42642E-01	-.54884E-01
.
.
.
49.9000	50.0000	.24981E-02	.26320E-02	-.78337E-02	-.26318E-02

Listagem 5.1 – Exemplo de dados salvos em um arquivo TXT.

Uma instância da classe `WanglayReader` é capaz de realizar a leitura de dois tipos de documentos de textos: com campos de largura fixa ou delimitados por vírgulas, no qual foram adotados, respectivamente, as extensões TXT e CSV para diferenciá-los. A Listagem 5.1 apresenta parte dos dados salvos em um arquivo TXT. Na primeira linha existe o cabeçalho e nas demais os valores numéricos. Em um arquivo CSV, cada bloco de espaços é substituído por uma vírgula.

```

1 reader = WanglayReader()
2 reader.setFilename("Example.csv")
3 reader.openFile()
4
5 print reader.getOutput()

```

```

Wanglay (43315648)
  Number Of Points: 5151
  X Range: [-50.1, 49.9]
  Y Range: [0.0, 0.0]
  Z Range: [0.0, 50.0]
  Dimensions: [101, 1, 51]

```

Listagem 5.2 – Exemplo de uso de uma instância da classe `WanglayReader`.

A Listagem 5.2 ilustra o uso de um objeto da classe `WanglayReader`. Há exigência de apenas um parâmetro para seu funcionamento, a localização do arquivo a ser lido (ver linha 2). Na parte inferior é apresentado o resultado da execução do código, que corresponde às informações adicionais relatadas no início desta seção.

5.3.2 Enriquecimento

A etapa de enriquecimento, como apresentado na Figura 5.2, é realizada por uma instância da classe `Wanglay2StructuredGridSeries`, que é responsável pela conversão de um objeto `Wanglay` em vários objetos `vtkStructuredGrid` do VTK, sendo que cada um representa as deformações de uma grade estruturada em cada instante de tempo. Estão implementadas na classe `Wanglay2StructuredGridSeries` as formulações matemáticas apresentadas na Seção 4.2.

A Listagem 5.3 ilustra o uso de um objeto da classe `Wanglay2StructuredGridSeries`. A entrada desta classe é a saída produzida pelo `WanglayReader` (um objeto `Wanglay`). Na parte inferi-

or é apresentado o resultado gerado pela execução do código, que informa que para cada instante há um objeto `vtkStructuredGrid` instanciado.

<pre> 1 series = Wanglay2StructuredGridSeries () 2 series.setInput (reader.getOutput ()) 3 series.setNumberOfTimeSteps (5) 4 series.setDeformationConstant (3.5) 5 series.movablePointsOn () 6 series.scalarArrayOfDisplacementOn () 7 series.vectorArrayOfDisplacementOn () 8 series.convert () 9 10 print series.getOutput () </pre>	<pre> Time Step: 0, vtkStructuredGrid ID: 52080272 Time Step: 1, vtkStructuredGrid ID: 52080248 Time Step: 2, vtkStructuredGrid ID: 52080296 Time Step: 3, vtkStructuredGrid ID: 52080320 Time Step: 4, vtkStructuredGrid ID: 52080344 </pre>
---	---

Listagem 5.3 – Exemplo de uso de uma instância da classe `Wanglay2StructuredGridSeries`.

5.3.3 Filtragem

A etapa de filtragem (ver Figura 5.2) consiste nas transformações aplicadas entre os objetos das classes `vtkStructuredGrid` e `vtkDataSetMapper`, que faz o mapeamento dos dados para o sistema gráfico. Estas transformações, chamadas de algoritmos no MWPVis, são: *Extract Subset*, *Warp By Scalar*, *Reflect* e *Clip*. As classes responsáveis por estes quatro procedimentos possuem a mesma denominação e foram implementadas no contexto deste trabalho.

5.3.3.1 *Extract Subset*

O algoritmo de *Extract Subset* consiste na inserção de uma instância da classe `vtkExtractGrid`, que implementa a técnica de extração geométrica (ver Seção 4.3), no fluxo de visualização padrão, como apresentado na Figura 5.3. Como o `vtkExtractGrid` exige que sua entrada seja um objeto da classe `vtkStructuredGrid`, então este é aplicado antes dos demais algoritmos. A classe `vtkExtractGrid` tem como parâmetro o Volume de Interesse (VOI – *Volume of Interest*), que é

definido pelos valores mínimos e máximos dos índices dos vértices da grade estruturada. A saída desta classe é um objeto `vtkStructuredGrid`, que corresponde a região de interesse selecionada pelo usuário.

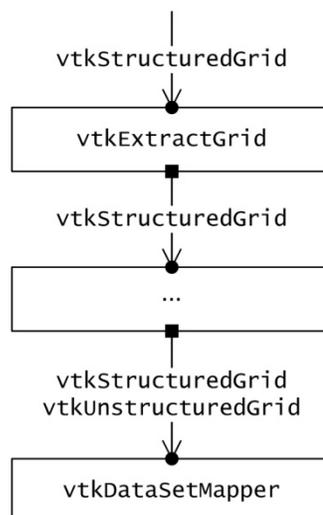


Figura 5.3 – Aplicação do algoritmo de *Extract Subset*.

A Listagem 5.4 exemplifica o uso de uma instância da classe `vtkExtractGrid`. O `vtkStructuredGrid` (variável `grid`) que antes era a entrada do `vtkDataSetMapper` (variável `mapper`), passa a ser a entrada do `vtkExtractGrid` (linha 2) e a saída deste, que também é um `vtkStructuredGrid`, passa a ser a entrada do `vtkDataSetMapper` (linha 6).

```

1  extract = vtk.vtkExtractGrid()
2  extract.SetInput(grid)
3  extract.SetVOI(2, 7, 1, 4, 0, 0)
4  extract.Update()
5
6  mapper.SetInput(extract.GetOutput())
7  mapper.Update()
  
```

Listagem 5.4 – Exemplo de uso de uma instância da classe `vtkExtractGrid`.

5.3.3.2 *Warp By Scalar*

O algoritmo de *Warp By Scalar* consiste na aplicação de uma instância da classe `vtkWarpScalar`, que implementa a técnica de mapeamento em função de altura (ver Seção 2.4.3), no

fluxo de visualização padrão, como ilustrado na Figura 5.4. A classe `vtkWarpScalar` exige como parâmetros os valores do fator de escala e da normal. O fator de escala é equivalente a um valor constante multiplicador que é aplicado sobre os escalares a serem mapeados. E a normal é a direção que será aplicada à deformação. Este algoritmo é o segundo na ordem de aplicação dos algoritmos.

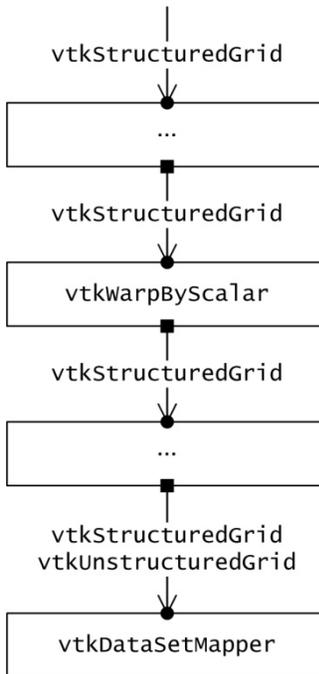


Figura 5.4 – Aplicação do algoritmo de *Warp By Scalar*.

```

1  grid.GetPointData().SetActiveScalars("Intensity")
2  grid.Update()
3
4  warp = vtk.vtkWarpScalar()
5  warp.SetInput(grid)
6  warp.SetScaleFactor(3.5)
7  warp.SetNormal(0, 0, 1)
8  warp.UseNormalOn()
9  warp.Update()
10
11 mapper.SetInput(warp.GetOutput())
12 mapper.Update()

```

Listagem 5.5 – Exemplo de uso de uma instância da classe `vtkWarpScalar`.

A Listagem 5.5 ilustra o uso de uma instância da classe `vtkWarpByScalar`. Na linha 1 é ativado o conjunto de escalares que serão mapeados, no caso um `vtkDoubleArray` (classe que ar-

mazena valores reais) chamado “Intensity”. Na linha 5 o objeto `vtkWarpScalar` recebe como entrada um objeto `vtkStructuredGrid`. Em seguida, nas linhas 6 e 7 são inicializados o fator de escala e a normal. Por fim, na linha 11, o mapeador (`vtkDataSetMapper`) recebe a saída produzida pelo `vtkWarpScalar`, um novo objeto `vtkStructuredGrid`.

5.3.3.3 *Reflect*

O algoritmo de *Reflect* consiste na aplicação de uma instância da classe `vtkReflectionFilter`, que implementa a técnica de reflexão ou espelhamento (ver Seção 4.2.10), no fluxo de visualização padrão, como apresentado na Figura 5.5. A classe `vtkReflectFilter` exige a inicialização de três parâmetros: plano, que informa em qual plano de coordenadas a grade será refletida; centro, o centro da reflexão; e cópia, se a grade refletida substituirá a grade original ou se fundirá a esta. Este algoritmo é o terceiro na ordem de aplicação dos algoritmos.

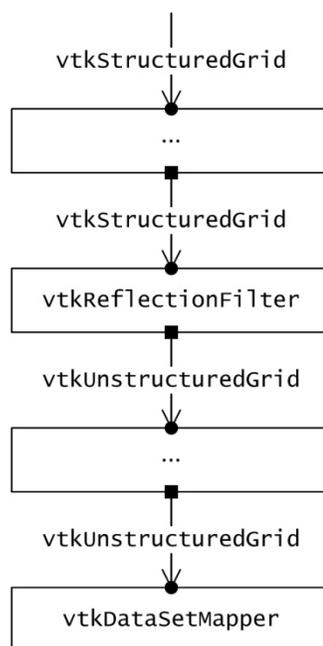


Figura 5.5 – Aplicação do algoritmo de *Reflect*.

A Listagem 5.6 ilustra o uso de uma instância da classe `vtkReflectionFilter`. Na linha 2 há entrada de um objeto `vtkStructuredGrid`. Na linha 3 é inicializado um plano de reflexão em xz , cujo deslocamento deste é de 5,2 em y (linha 4). Na linha 5 é mantido a grade original. Na saída é

produzido um objeto `vtkUnstructuredGrid` (grade não-estruturada), que é a entrada da classe `vtkDataSetMapper`.

```

1 reflect = vtk.vtkReflectionFilter()
2 reflect.SetInput(grid) # entrada de um objeto vtkStructuredGrid
3 reflect.SetPlaneToY() # plano de reflexão em XZ
4 reflect.SetCenter(5.2) # deslocamento do plano de reflexão em Y
5 reflect.CopyInputOn() # mantém a grade original
6 reflect.Update()
7
8 mapper.SetInput(reflect.GetOutput())
9 mapper.Update()

```

Listagem 5.6 – Exemplo de uso de uma instância da classe `vtkReflectionFilter`.

5.3.3.4 *Clip*

O algoritmo de *Clip* consiste na aplicação de uma instância da classe `vtkClipDataSet`, que implementa a técnica de recorte (ver Seção 2.6.2), no fluxo de visualização padrão, como apresentada na Figura 5.6. A classe `vtkClipDataSet` exige apenas um parâmetro de entrada: a função implícita de uma figura geométrica, que pode ser um plano (`vtkPlane`), um paralelepípedo (`vtkPlanes`) ou uma esfera (`vtkSphere`). Na ordem de aplicação dos algoritmos, este é o último deles.

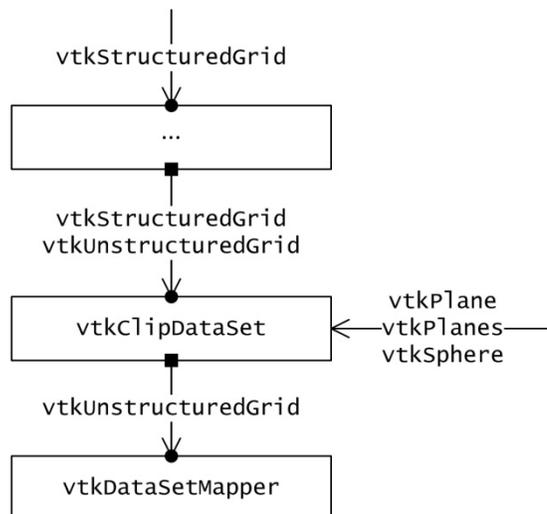


Figura 5.6 – Aplicação do algoritmo de *Clip*.

A Listagem 5.7 ilustra o uso de uma instância da classe `vtkClipDataSet`. Da linha 1 a 3 é declarado a função implícita de uma esfera, que é a figura geométrica de recorte (linha 7). A classe `vtkClipDataSet` produz um `vtkUnstructuredGrid`, que é a entrada da classe `vtkDataSetMapper` (linha 11).

```

1 sphere = vtk.vtkSphere()
2 sphere.SetCenter(15.2)
3 sphere.SetRadius(32.7)
4
5 clip = vtk.vtkClipDataSet()
6 clip.SetInput(grid)
7 clip.SetFunction(sphere)
8 clip.InsideOutOn()
9 clip.Update()
10
11 mapper.SetInput(clip.GetOutput())
12 mapper.Update()

```

Listagem 5.7 – Exemplo de uso de uma instância da classe `vtkClipDataSet`.

5.3.4 Mapeamento

A etapa de mapeamento consiste nas transformações aplicadas a partir dos objetos da classe `vtkDataSetMapper`, seja pela alteração destes objetos ou pela instanciação de novos mapeadores. As classes *Colormap*, *Contour*, *Glyph*, *Outline*, *Slice* e *Stream Tracer* possuem os procedimentos responsáveis por estas transformações.

5.3.4.1 *Colormap*

O algoritmo de *Colormap* consiste na aplicação de uma instância da classe `vtkColorTransferFunction` no fluxo de visualização padrão, assim como apresentado na Figura 5.7. A classe `vtkColorTransferFunction` implementa a função de transferência para mapeamento em cores (ver Seção 2.4.1). Esta classe exige apenas os parâmetros que definem a correspondência entre as intensidades dos atributos de dados escalares que serão mapeados e as cores que serão utilizadas.

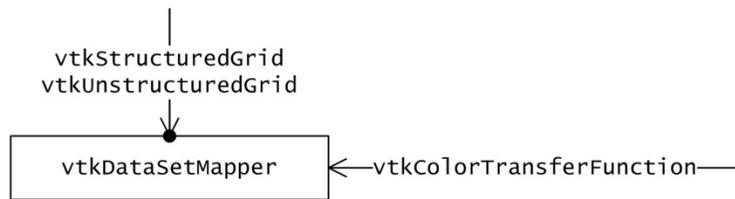


Figura 5.7 – Aplicação do algoritmo de *Colormap*.

A Listagem 5.8 ilustra o uso de uma instância da classe `vtkColorTransferFunction`. Da linha 1 a 5 é declarada uma função de transferência do tipo “arco-íris” para o mapeamento em cores. Na linha 7 é habilitado os atributos de dados escalares que serão mapeados em cores, na linha 8 é inicializado a tabela de cores com a função de transferência criada e nas demais linhas (de 9 a 11) são habilitadas algumas funções necessárias.

```

1  rainbow = vtk.vtkColorTransferFunction()
2  rainbow.SetColorSpaceToHSV()          # espaço de cor HSV
3  rainbow.AddRGBPoint(0.0, 0, 0, 1) # azul para intensidade 0.0
4  rainbow.AddRGBPoint(0.5, 0, 1, 0) # verde para intensidade 0.5
5  rainbow.AddRGBPoint(1.0, 1, 0, 0) # vermelho para intensidade 1.0
6
7  mapper.SelectColorArray("Aspect Ratio")
8  mapper.SetLookupTable(rainbow)
9  mapper.SetScalarModeToUsePointFieldData()
10 mapper.ScalarVisibilityOn()
11 mapper.InterpolateScalarsBeforeMappingOn()
12 mapper.Update()
  
```

Listagem 5.8 – Exemplo de uso de uma instância da classe `vtkColorTransferFunction`.

5.3.4.2 *Contour*

O algoritmo *Contour* consiste na inserção de uma instância da classe `vtkContourFilter` no fluxo de visualização padrão, como apresentado na Figura 5.8. A classe `vtkContourFilter` implementa a técnica de contorno apresentada na Seção 2.4.2. Esta classe depende apenas de dois parâmetros de entrada: do objeto `vtkStructuredGrid` (ou `vtkUnstructuredGrid`) fornecido à classe `vtkDataSetMapper` (ver Figura 5.2) e do número de intervalos para contorno. Como saída tem-se um objeto do tipo `vtkPolyData`, para representação de dados não estruturados, e por isso, a necessidade de um mapeador específico para este objeto (`vtkPolyDataMapper`).

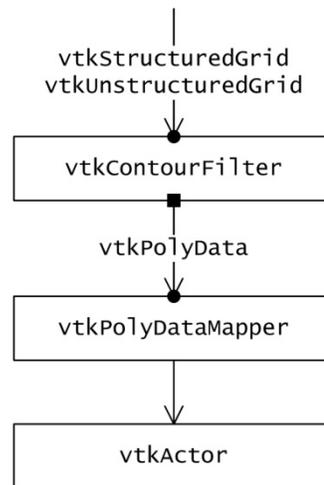


Figura 5.8 – Aplicação do algoritmo de *Contour*.

A Listagem 5.9 representa o uso de uma instância da classe `vtkContourFilter`. No primeiro bloco do código-fonte é instanciado um objeto desta classe, cujas entradas são: uma grade estruturada ou não-estruturada vinda do mapeador do fluxo de visualização padrão (variável `mapper`) e o número de contornos, no caso 5. Os demais blocos são necessários para a apresentação dos contornos sobre a estrutura original.

```

1  contour = vtk.vtkContourFilter()
2  contour.SetInput(mapper.GetInput())
3  contour.SetNumberOfContours(5)
4
5  mapper2 = vtk.vtkPolyDataMapper()
6  mapper2.SetInput(contour.GetOutput())
7  mapper2.SetScalarModeToUsePointFieldData()
8  mapper2.SetResolveCoincidentTopologyToPolygonOffset()
9
10 actor2 = vtk.vtkActor()
11 actor2.SetMapper(mapper2)
  
```

Listagem 5.9 – Exemplo de uso de uma instância da classe `vtkContourFilter`.

5.3.4.3 *Glyph*

O algoritmo de *Glyph* consiste na inserção de uma instância da classe `vtkGlyph3D`, que implementa a técnica de mapeamento em glifos de vetor da Seção 2.5.1, no fluxo de visualização padrão. As Figuras 5.9 e 5.10 representam o uso desta classe para a visualização das normais e

dos atributos de dados vetoriais, respectivamente. As classes à direita, em ambas as imagens, são as figuras geométricas 2D ou 3D que podem ser usadas para representação dos glifos, tais como: cone, cubo, cilindro, linha etc.

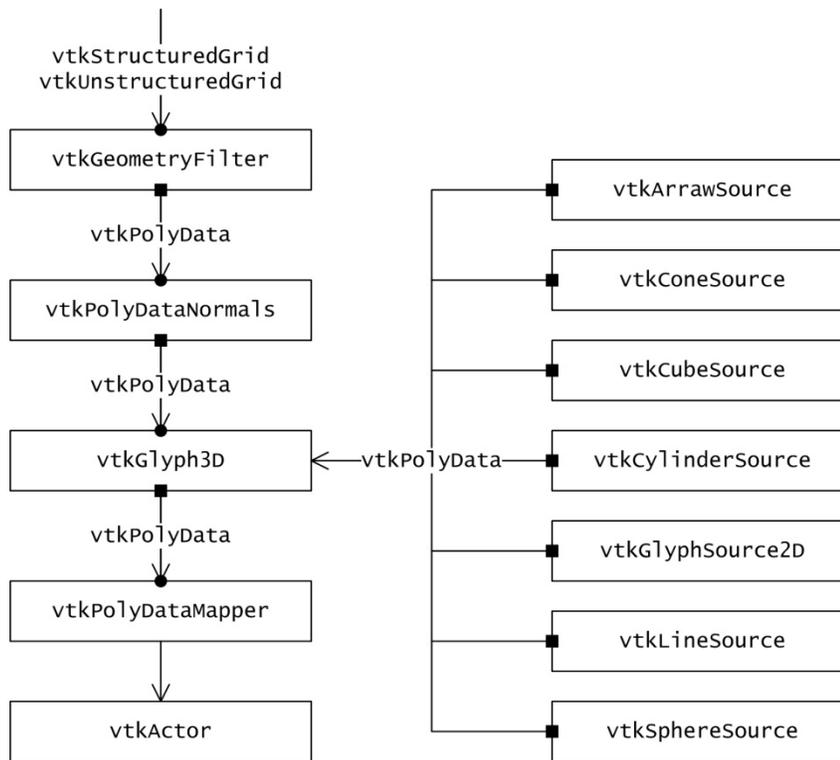


Figura 5.9 – Aplicação do algoritmo de *Glyph* para visualização das normais.

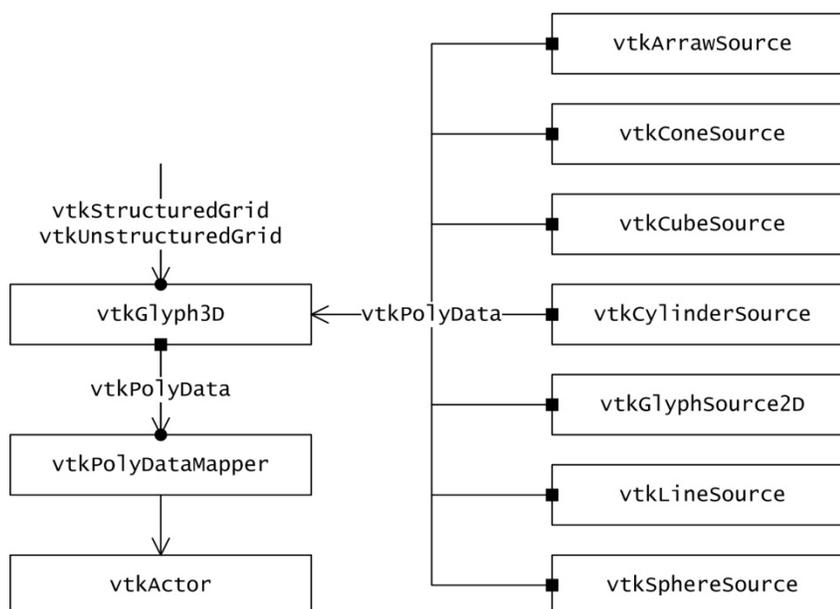


Figura 5.10 – Aplicação do algoritmo de *Glyph* para visualização dos atributos de dados vetoriais.

A Listagem 5.10 ilustra o uso de uma instância da classe `vtkGlyph3D` para visualização das normais. A entrada do mapeador do fluxo de visualização padrão também é a entrada do filtro (linha 1 e 2). A classe `vtkPolyDataNormals` calcula as normais a partir do objeto gerado pelo filtro e retorna um objeto da classe `vtkPolyData`, que contém as normais que são representadas por uma figura geométrica (`source`) por intermédio da classe `vtkGlyph3D`.

```

1  filter = vtk.vtkGeometryFilter()
2  filter.SetInput (mapper.GetInput ())
3
4  normals = vtk.vtkPolyDataNormals()
5  normals.SetInput (filter.GetOutput ())
6  normals.FlipNormalsOn()
7  normals.SplittingOff()
8
9  glyph = vtk.vtkGlyph3D()
10 glyph.SetInput (normals.GetOutput ())
11 glyph.SetSource (source)
12 glyph.SetScaleFactor (10)
13
14 mapper2 = vtk.vtkPolyDataMapper()
15 mapper2.SetInputConnection (glyph.GetOutputPort ())
16 mapper2.ScalarVisibilityOff()
17
18 actor2 = vtk.vtkActor()
19 actor2.SetMapper (mapper2)
20 actor2.VisibilityOff()

```

Listagem 5.10 – Exemplo de uso de uma instância da classe `vtkGlyph3D`.

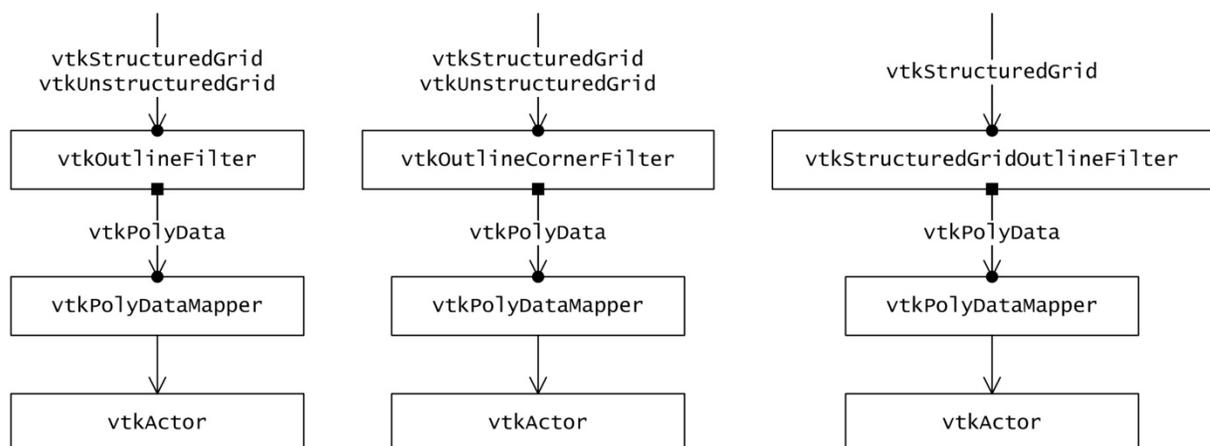


Figura 5.11 – Aplicação do algoritmo de *Outline*.

5.3.4.4 Outline

O algoritmo *Outline* consiste na aplicação de uma instância das classes de `vtkOutlineFilter`, `vtkOutlineCornerFilter` ou `vtkStructuredGridOutlineFilter` no fluxo de visualização padrão (ver Figura 5.11). Estas classes são variações da técnica de *bounding box*. A primeira apresenta um *bounding box* comum e a segunda apenas as quinas deste *bounding box*. Já a terceira segue os limites da grade, reduzindo o volume do *bounding box* ao mesmo da grade.

```

1  outline = vtk.vtkOutlineFilter()
2  outline.SetInput( mapper.GetInput() )
3
4  mapper2 = vtk.vtkPolyDataMapper()
5  mapper2.SetInputConnection( outline.GetOutputPort() )
6  mapper2.ScalarVisibilityOff()
7
8  actor2 = vtk.vtkActor()
9  actor2.SetMapper( mapper2 )
10 actor2.VisibilityOff()

```

Listagem 5.11 – Exemplo de uso de uma instância a classe `vtkOutlineFilter`.

A Listagem 5.11 ilustra o uso de uma instância da classe `vtkOutlineFilter`. Os demais filtros são aplicados de maneira semelhante, exceto o último que exige como entrada um objeto da classe `vtkStructuredGrid`.

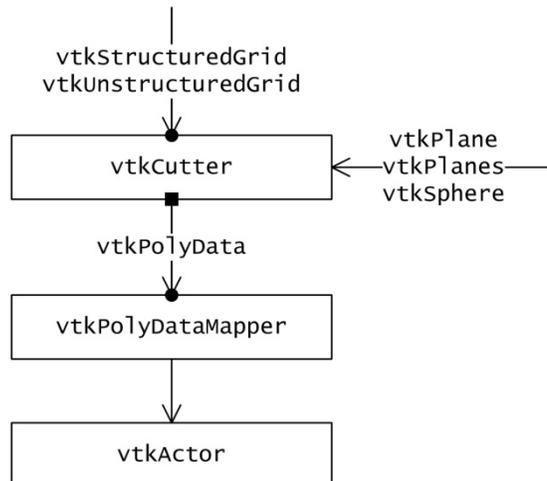


Figura 5.12 – Aplicação do algoritmo de *Slice*.

5.3.4.5 *Slice*

O algoritmo de *Slice* consiste na inserção de uma instância da classe `vtkCutter`, que implementa a técnica de fatiamento (ver Seção 2.6.3), no fluxo de visualização padrão, como apresentado na Figura 5.12. A classe `vtkCutter` exige apenas um parâmetro de entrada: a função implícita de uma figura geométrica, que pode ser um plano (`vtkPlane`), um paralelepípedo (`vtkPlanes`) ou uma esfera (`vtkSphere`).

A Listagem 5.12 representa o uso de uma instância da classe `vtkCutter`. Da linha 1 a 3 é declarada a função de uma esfera, que é a figura geométrica para fatiamento (linha 7). A classe `vtkCutter` produz um `vtkPolyData`, que é a entrada da classe `vtkPolyDataMapper`.

```

1 sphere = vtk.vtkSphere()
2 sphere.SetCenter(15.2)
3 sphere.SetRadius(32.7)
4
5 slice = vtk.vtkCutter()
6 slice.SetInput(mapper.GetInput())
7 slice.SetCutFunction(sphere)
8 slice.InsideOutOn()
9 slice.Update()
10
11 mapper2= vtk.vtkPolyDataMapper()
12 mapper2.SetInput(slice.GetOutput())
13 mapper2.SetResolveCoincidentTopologyToPolygonOffset()
14 mapper2.Update()
15
16 actor2= vtk.vtkActor()
17 actor2.SetMapper(mapper2)

```

Listagem 5.12 – Exemplo de uso de uma instância da classe `vtkCutter`.

5.3.4.6 *Stream Tracer*

O algoritmo de *Stream Tracer* consiste na aplicação de uma instância da classe `vtkStreamTracer` no fluxo de visualização padrão (ver Figura 5.13). Esta classe depende de três parâmetros: da grade, da distância percorrida a cada iteração (resolução das linhas) e do número de iterações (prolongamento das linhas).

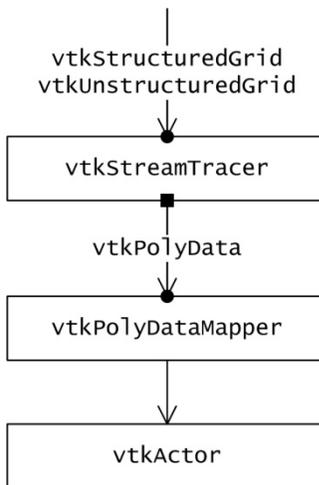


Figura 5.13 – Aplicação do algoritmo de *Stream Tracer*.

A Listagem 5.13 representa o uso de uma instância da classe *vtkStreamTracer*. No primeiro bloco é instanciado um objeto desta classe definido para gerar linhas de fluxo com no máximo cinco iterações com distância de 0,25 (entre vértices vizinhos) para cada uma delas.

```

1  tracer = vtk.vtkStreamTracer ()
2  tracer.SetSource (source)
3  tracer.SetInput (mapper.GetInput ())
4  tracer.SetInitialIntegrationStepUnitToLengthUnit ()
5  tracer.SetIntegratorTypeToRungeKutta2 ()
6  tracer.SetIntegrationDirectionToForward ()
7  tracer.SetInitialIntegrationStep (0.25)
8  tracer.SetMaximumPropagation (5)
9
10 mapper2 = vtk.vtkPolyDataMapper ()
11 mapper2.SetInput (tracer.GetOutput ())
12 mapper2.ScalarVisibilityOff ()
13 mapper2.SetResolveCoincidentTopologyToPolygonOffset ()
14
15 actor2 = vtk.vtkActor ()
16 actor2.SetMapper (mapper2)
17 actor2.VisibilityOff ()
  
```

Listagem 5.13 – Exemplo de uso de uma instância da classe *vtkStreamTracer*.

5.3.5 Renderização

A etapa de renderização no MWPVis implementa as técnicas descritas na Seção 2.7, são elas: animação, visões múltiplas e visão estereoscópica. A animação é realizada pela classe Ani-

mationTemporalActorSet. Consiste em habilitar ou desabilitar a exibição dos atores (vtkActor), que representam um objeto na cena. A Figura 5.14 representa esta lógica de funcionamento, no qual inicialmente todos os atores se encontram invisíveis, ou seja, não são renderizados. Posteriormente, cada um destes atores tem sua visibilidade habilitada ou desabilitada a cada quadro. Por exemplo, no Quadro 1, o ator de índice 0 está visível (fundo cinza) enquanto os demais estão invisíveis (fundo branco). No Quadro 2, a visibilidade do ator de índice 0 é desabilitada e a do ator de índice 1 é habilitada. Nos quadros subsequentes seguem a mesma lógica.

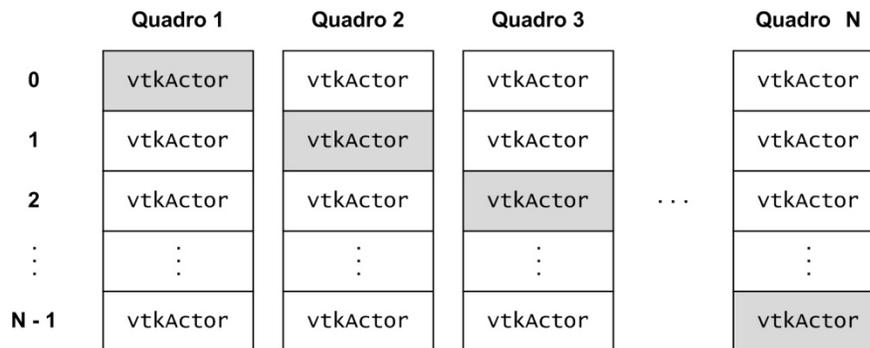


Figura 5.14 – Animação por controle da visibilidade dos atores.

A Listagem 5.14 ilustra este processo de animação com uso do VTK. O método tick é invocado pelo objeto iren, da classe QVTKRenderWindowInteractor (linha 9), a cada 100 milissegundos (linha 10). O método tick oculta um vtkActor de índice t contido em uma lista de atores (actors) e, em seguida, habilita a visibilidade do próximo na lista. Este procedimento continua até que o índice seja zerado e o processo reiniciado.

```

1  def tick(obj, e):
2      global actors, t, iren
3      actors.__getitem__(t).VisibilityOff()
4      t += 1
5      if t == NUMBER_OF_FRAMES: t = 0
6      actors.__getitem__(t).VisibilityOn()
7      iren.GetRenderWindow().Render()
8
9  iren._Iren.AddObserver("TimerEvent", self.tick)
10 iren._Iren.CreateRepeatingTimer(100)

```

Listagem 5.14 – Exemplo de animação pelo VTK.

A Listagem 5.15 ilustra como as visões múltiplas podem ser empregadas pelo VTK. Nas linhas 1 a 4 são especificadas os limites das regiões de quatro renderizadores. O resultado desta

execução é uma janela dividida em quatro partes independentes, uma para cada renderizador (render1, render2, render3 e render4).

```

1  render1.SetViewport (0.0, 0.5, 0.5, 1.0) # região superior esquerda
2  render2.SetViewport (0.5, 0.5, 1.0, 1.0) # região superior direita
3  render3.SetViewport (0.0, 0.0, 0.5, 0.5) # região inferior esquerda
4  render4.SetViewport (0.5, 0.0, 1.0, 0.5) # região inferior direita
5
6  irender = QVTKRenderWindowInteractor ()
7  irender.GetRenderWindow ().AddRenderer (render1)
8  irender.GetRenderWindow ().AddRenderer (render2)
9  irender.GetRenderWindow ().AddRenderer (render3)
10 irender.GetRenderWindow ().AddRenderer (render4)
11 irender.Initialize ()
12 irender.Start ()

```

Listagem 5.15 – Uso de visões múltiplas e interação com o usuário com VTK.

O VTK trabalha com visão estereoscópica por anaglifo (ver Seção 2.7.3). A Listagem 5.16 apresenta dois métodos, sendo um para ativação (useAnaglyphOn) e outro para desativação (useAnaglyphOff) desta visão.

```

1  # Habilita a visão estereoscópica por anaglifo.
2  def useAnaglyphOn (irender):
3      irender.GetRenderWindow ().SetStereoTypeToAnaglyph ()
4      irender.GetRenderWindow ().StereoRenderOn ()
5      irender.GetRenderWindow ().StereoUpdate ()
6      irender.GetRenderWindow ().Render ()
7
8  # Desabilita a visão estereoscópica por anaglifo.
9  def useAnaglyphOff (irender):
10     irender.GetRenderWindow ().StereoRenderOff ()
11     irender.GetRenderWindow ().SetStereoTypeToCrystalEyes ()
12     irender.GetRenderWindow ().StereoUpdate ()
13     irender.GetRenderWindow ().Render ()

```

Listagem 5.16 – Uso de visão estereoscópica por anaglifo por intermédio do VTK.

5.3.6 Interface

A etapa de interface será descrita sob duas perspectivas: interação com o usuário e organização da GUI, respectivamente realizadas por meio do VTK e do Qt. No contexto da visualização, a interação é um aspecto fundamental para que o usuário possa explorar seus dados. É uma

GUI bem estruturada pode agilizar este procedimento e os de transformação do fluxo de visualização, tal como para aplicação dos algoritmos.

O VTK dispõe de um conjunto de estilos de interação, que estabelecem as maneiras como o usuário pode interagir com o sistema para explorar seus dados, seja pelo teclado ou pelo *mouse* (KITWARE, 2006). A classe *vtkInteractorStyleRubberBandPick* foi utilizada no MWPVis. Esta possibilita ao usuário aplicar transformações de rotação, translação e escala em cenários 3D por meio de clique e arraste do ponteiro de *mouse*. Outra técnica de interação disponível no VTK e usada no MWPVis é a de seleção de regiões provida pela classe *vtkAreaPicker*, que permite ao usuário selecionar um conjunto de vértices contido dentro de uma região de interesse.

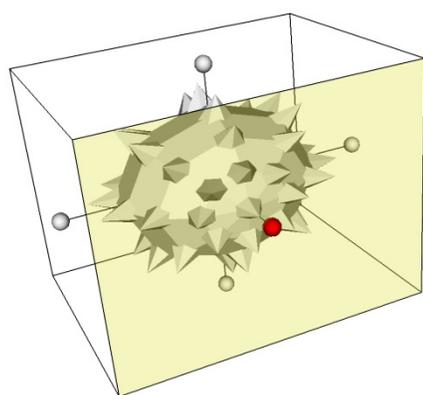
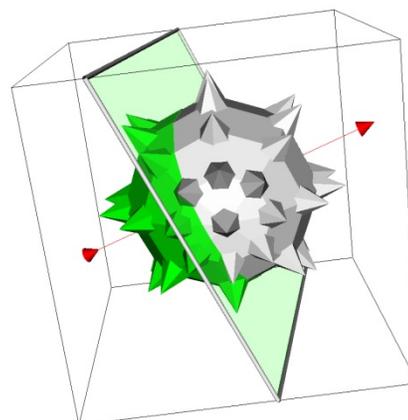
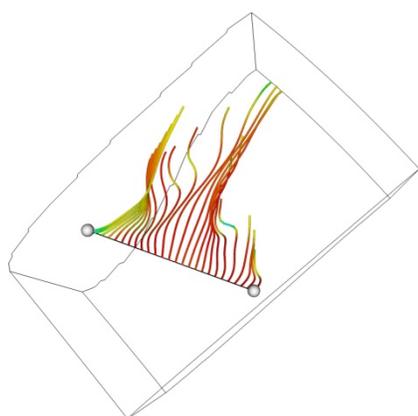
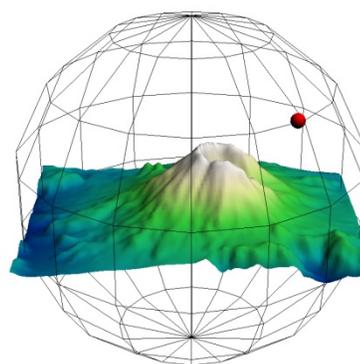
a) *vtkBoxWidget*b) *vtkImplicitPlaneWidget*c) *vtkLineWidget*d) *vtkSphereWidget*

Figura 5.15 – Aplicação de alguns *widgets* disponíveis no VTK (SCHROEDER; MARTIN; LORENSEN, 2006).

Os estilos de interação do VTK não são representados nas cenas, ou seja, eles não podem ser vistos pelo usuário enquanto ele interage com a visualização. Em alguns casos, muitas operações seriam facilitadas caso o usuário pudesse interagir diretamente com objetos em cena. Neste propósito, o VTK dispõe um conjunto de *widgets*, chamados de *widgets* 3D. A Figura 5.15 apresenta quatro destes *widgets*, que foram usados no MWPVis para posicionar as figuras geométricas (plano, paralelepípedo e esfera) sobre a grade estruturada para aplicação dos algoritmos de *Clip* e *Slice* e para obter a distância entre dois pontos.

A Figura 5.16 apresenta um leiaute simplificado da GUI do MWPVis. A janela do programa (QMainWindow) foi dividida em três regiões: região superior, que contém as barras de menus (QMenuBar) e de ferramentas (QToolBar); região central, que contém os painéis de navegação (QDockWidget); e região inferior, que contém as barras de estado (QStatusBar) e de progresso (QProgressBar). A região central, por sua vez, foi subdividida em duas regiões: região esquerda, que contém os painéis de navegação para controle dos algoritmos; e região direita, que contém o painel de navegação para renderização dos dados.

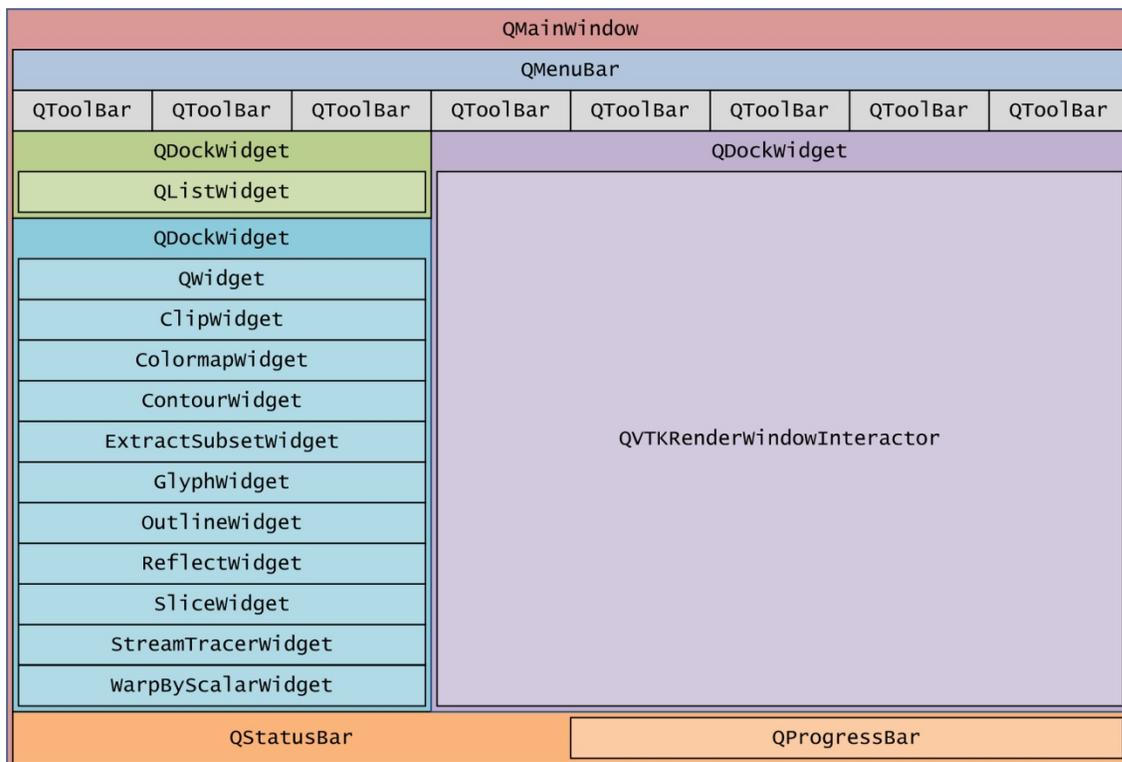


Figura 5.16 – Leiaute simplificado da GUI do MWPVis.

O painel de navegação superior da região central à esquerda contém uma lista (QListWidget) para exibir os algoritmos aplicados pelo usuário. No painel de navegação inferior a este, há vários *widgets*. O primeiro (QWidget) contém quatro botões: aplicar, editar, cancelar e excluir (não ilustradas), que são os comandos usados para aplicação dos algoritmos no fluxo de visualização. Os demais *widgets* (ClipWidget, ColormapWidget, ContourWidget etc.) são formulários para entrada dos valores dos parâmetros dos algoritmos. Estes estão inicialmente ocultos e se tornam visíveis conforme a interação do usuário.

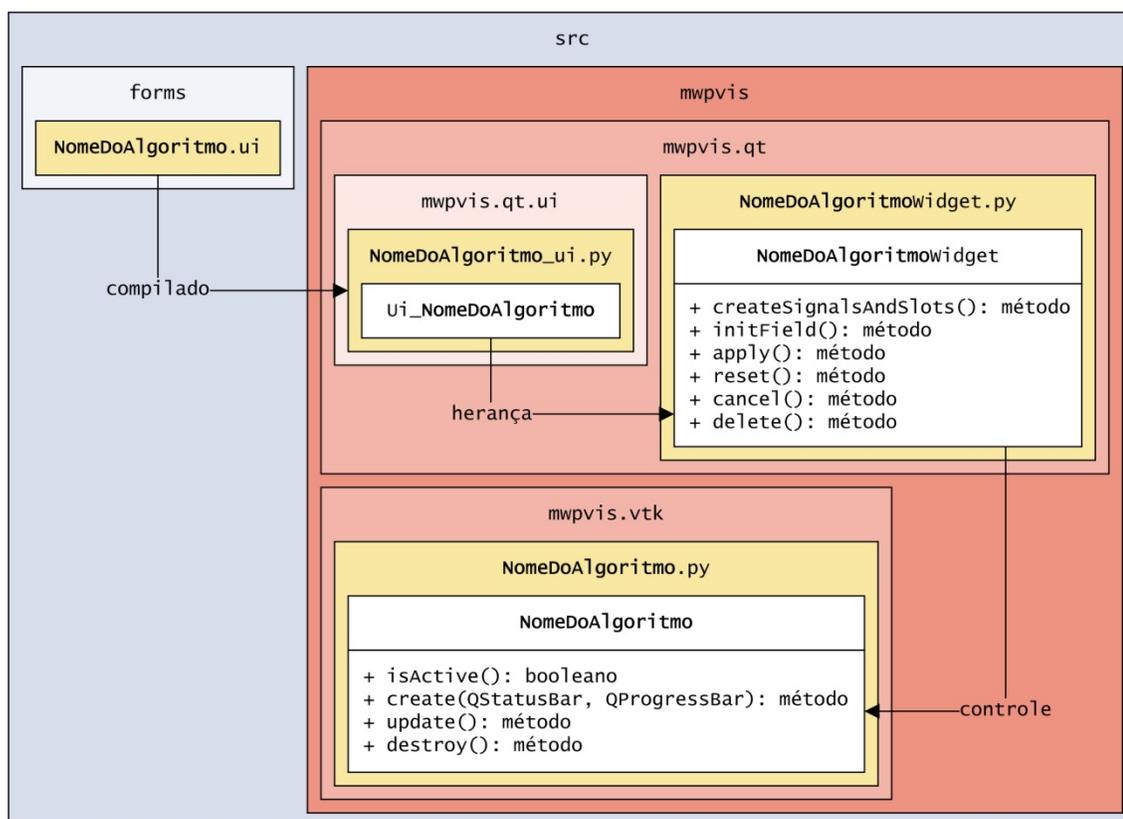


Figura 5.17 – Organização e aplicação das classes de GUI.

A seguir é apresentada a organização das classes da GUI e dos algoritmos (ver Figura 5.17). No diretório `src` do código-fonte do MWPVis há uma pasta chamada `forms` e outra chamada `mwpvis`. Na pasta `forms` há diversos arquivos de extensão UI (do Qt). Cada um deles descreve os layouts dos formulários dos algoritmos. Na pasta `mwpvis`, há os arquivos de extensão PY (do Python). Para cada arquivo UI (`NomeDoAlgoritmo.ui`) há um arquivo PY (`NomeDoAlgoritmo_ui.py`) gerado pela compilação do anterior. Estes arquivos PY contém classes (`Ui_Nome-`

DoAlgoritmo) que descrevem os leiautes dos formulários dos algoritmos que são herdados pelas classes NomeDoAlgoritmoWidget (em NomeDoAlgoritmoWidget.py). Estas classes são responsáveis pelo gerenciamento das classes NomeDoAlgoritmo (em NomeDoAlgoritmo.py), que empregam os respectivos algoritmos no fluxo de visualização.

5.4 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados detalhes de desenvolvimento do MWPVIs, o protótipo de ambiente de visualização produzido para auxiliar os especialistas no processo de análise dos problemas geomecânicos simulados. Neste ambiente foi implementado o fluxo de visualização proposto para o propósito anteriormente mencionado, tal como apresentado no Capítulo 4. Este capítulo serve de manual para os futuros desenvolvedores. As sugestões de melhorias são necessárias e estas são descritas no último capítulo.

6 ESTUDOS DE CASO

Neste capítulo são descritos dois estudos de caso para a verificação e validação das visualizações geradas pelo ambiente de visualização descrito no Capítulo 5. Nas Seções 6.1 e 6.2 são apresentados os estudos de caso, que correspondem à visualização de problemas de propagação de ondas mecânicas 2D e 3D, respectivamente. Para uma maior compreensão do conteúdo abordado é recomendada a leitura prévia do Capítulo 3 e da Seção 4.2.

6.1 ONDAS MECÂNICAS 2D

Nas Seções 6.1.1 e 6.1.2 são exibidas a descrição de um problema geomecânico 2D e as visualizações geradas pelo MWPVis, respectivamente.

6.1.1 Descrição do problema

A Figura 6.1 ilustra um problema geomecânico 2D composto por uma sequência de 50 camadas viscoelásticas horizontais (L_k), presentes em um semiespaço transversalmente isotrópico, que são harmonicamente excitadas por uma distribuição de tensão superficial (F_z), centrada na origem do sistema de coordenadas. As propriedades das camadas são descritas pelos seus parâmetros constitutivos: c_{11}^k , c_{13}^k , c_{33}^k , c_{44}^k e ρ_k . Em todas as camadas do meio, estes parâmetros possuem os mesmos valores, com exceção das camadas L_{25} a L_{27} , nas quais os quatro primeiros são menores. A mudança abrupta destes valores nestas camadas provoca uma forte descontinuidade de impedância que influencia no comportamento das ondas.

Para a simulação deste problema foram usados os seguintes valores: $a = 1m$ e $F_z = 1N$ para a tensão; $\rho_k = 1kg/m^3$ para todas as camadas; $c_{11} = 0,6N/m^2$, $c_{13} = 0,4N/m^2$, $c_{33} = 0,6N/m^2$ e $c_{44} = 0,1N/m^2$ para as camadas L_{25} a L_{27} ; e $c_{11} = 6N/m^2$, $c_{13} = 4N/m^2$, $c_{33} = 6N/m^2$ e $c_{44} = 1N/m^2$ para as demais camadas. Os dados foram gerados sobre uma grade

uniforme de espaçamento unitário e com dimensões 101×51 , totalizando 5.151 pontos de amostragem. O domínio do semiespaço está limitado entre os intervalos $-50 \leq x \leq 50$ e $0 \leq z \leq 50$.

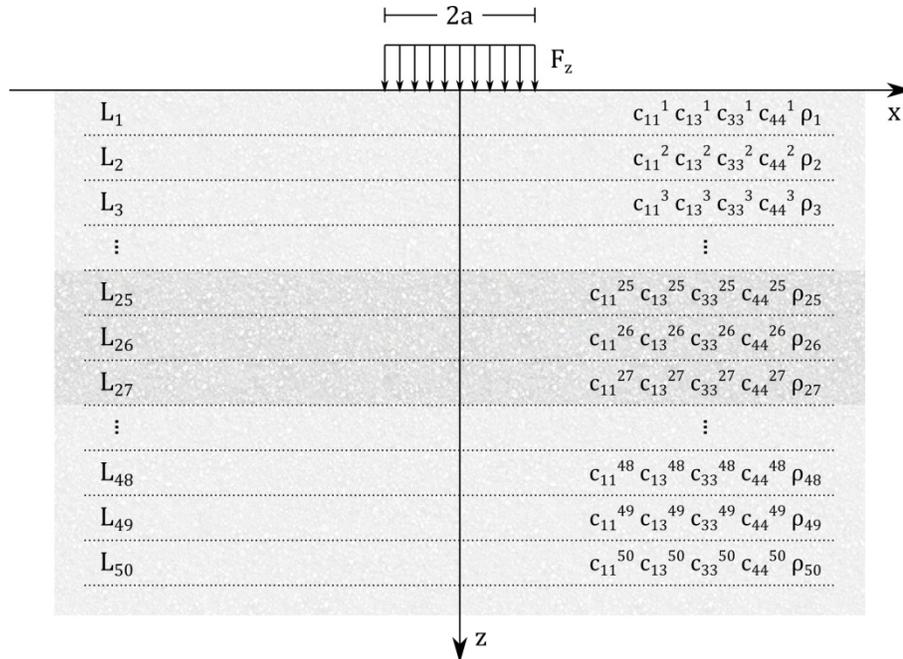


Figura 6.1 – Problema geomecânico 2D com 50 camadas viscoelásticas horizontais.

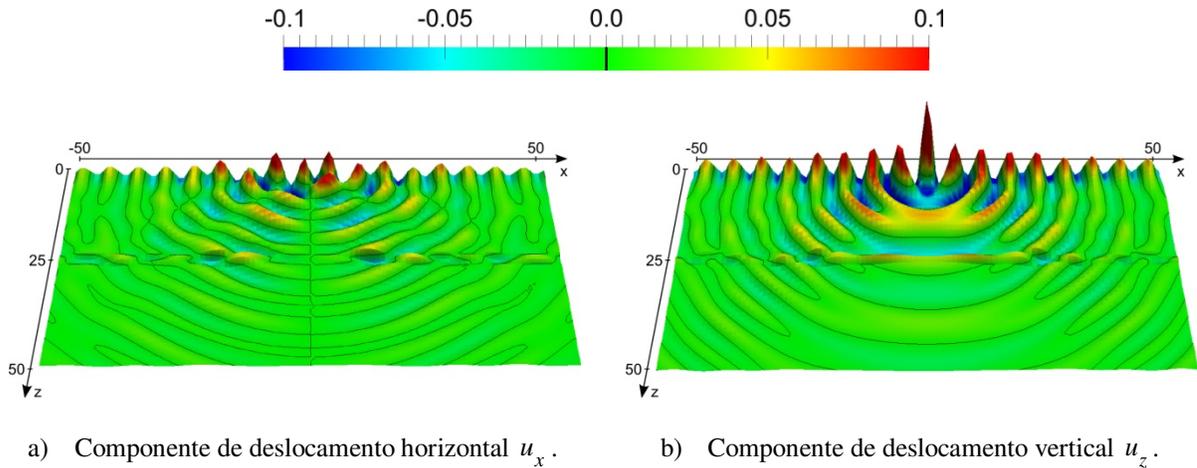
6.1.2 Visualizações

Para a análise do problema descrito anteriormente, foram geradas as visualizações apresentadas nas seções seguintes.

6.1.2.1 Visualização do deslocamento

A Figura 6.2 exibe as visualizações das componentes de deslocamento horizontal u_x e vertical u_z . A localização da excitação vertical F_z pode ser claramente identificada na Figura 6.2b devido ao paralelismo entre F_z e u_z . A partir desta região, ou seja, no intervalo $(-a \leq x \leq a, z = 0)$, um campo de deslocamento complexo é produzido e propagado através do meio. A “falha”, ou melhor, a descontinuidade de impedância produzida pela variação abrupta

dos parâmetros constitutivos das camadas L_{25} a L_{27} , pode ser claramente observada em ambas as imagens. Fica evidente que estas camadas provocam reflexão e transmissão de ondas, pois ocorre distorção no padrão de propagação. Entretanto, esta visualização não é suficiente para que o investigador possa identificar os padrões de ondas. As demais visualizações posteriormente apresentadas servirão para este propósito.



Parâmetros da visualização: instante de tempo normalizado $t = 0,19 \cdot 2\pi$, fator de escala do mapa de altura igual a 30, valor do escalar de contorno igual a 0 e intervalo dos valores escalares para normalização do mapeamento em cores igual a $[-0,1; 0,1]$.

Figura 6.2 – Visualização das componentes de deslocamento horizontal e vertical.

6.1.2.2 Visualização da relação de aspecto

Nos meios sólidos homogêneos se propagam duas classes de ondas: as ondas de corpo e as ondas de superfície. As ondas de corpo, como as ondas P e as ondas S, fazem com que as partículas do meio se desloquem em trajetórias retilíneas. As ondas de superfície, mais precisamente as ondas R (ou de Rayleigh), fazem com que as partículas do meio localizadas próximas à superfície se desloquem ao redor de suas posições de repouso em trajetórias elípticas retrógradas. Com base nestas características, a visualização da relação de aspecto pode auxiliar na identificação da presença ou predomínio de ondas de corpo ou de ondas de Rayleigh em regiões específicas do meio.

A Figura 6.3 apresenta a visualização da relação de aspecto da trajetória das partículas. As trajetórias retilíneas, que possuem relação de aspecto nula, indicam a presença de ondas de corpo puras. As regiões em azul exibem estas trajetórias, que são predominantes sobre o eixo de coordenadas z . As trajetórias elípticas com relação de aspecto entre 0,6 e 0,7, para os parâmetros constitutivos usados neste problema, indicam a presença de ondas de Rayleigh (GRAFF, 1991). As regiões de transição entre as cores verde e amarelo, também destacadas pelas isolinhas, exibem estas trajetórias, que são predominantes na região superficial do meio, ao longo do eixo de coordenadas x .

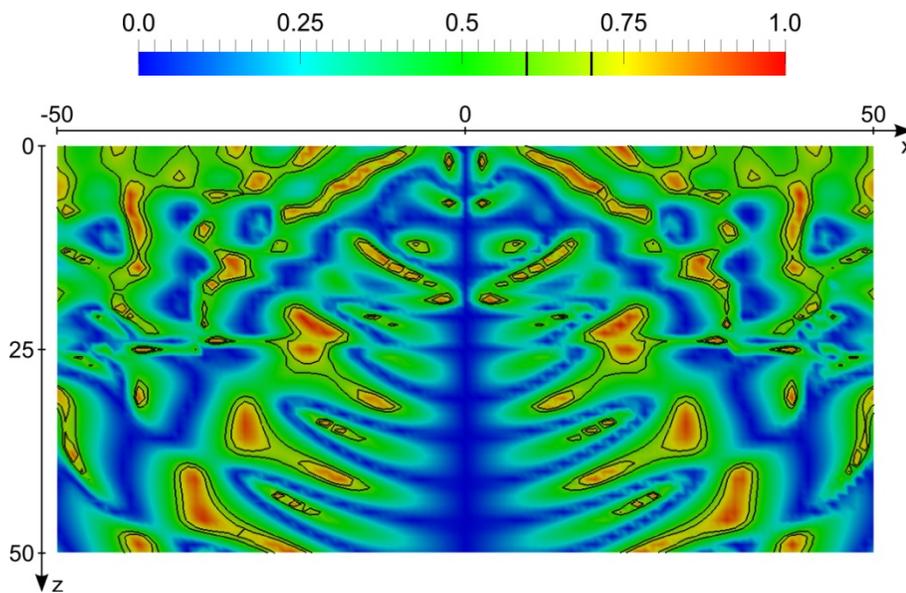
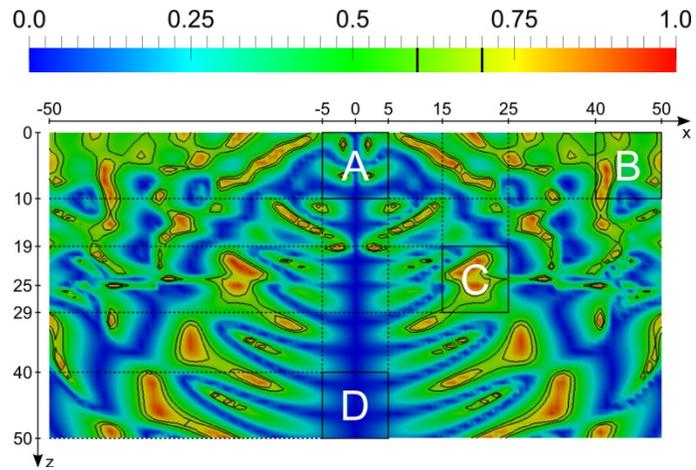


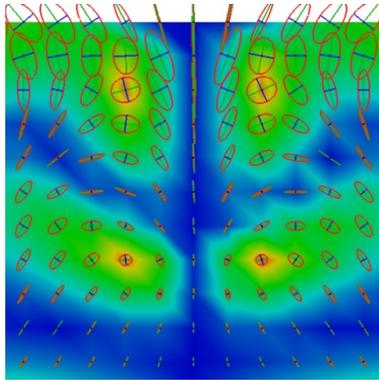
Figura 6.3 – Visualização da relação de aspecto da trajetória das partículas.

6.1.2.3 Visualização da trajetória

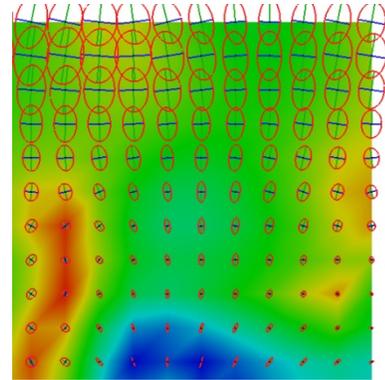
A Figura 6.4 apresenta a visualização da trajetória das partículas em quatro regiões da Figura 6.3, tal como ilustrado na Figura 6.4a. Como esperado, abaixo da região de excitação, devido à simetria horizontal das propriedades das camadas, as ondas de corpo são predominantes (ver Figura 6.4b). Na Figura 6.4c, os eixos maiores das trajetórias elípticas perpendicularmente alinhados com a superfície do material e o decaimento repentino da amplitude destas trajetórias indicam a presença de ondas de Rayleigh. A visualização do sentido do movimento de rotação poderá confirmar esta hipótese.



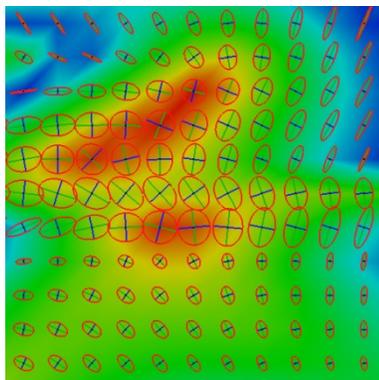
- a) Visão geral para localização das regiões de interesse A, B, C e D. Os parâmetros das dimensões da subgrade ($n_x = n_z = 11$) e do número de pontos de amostragem ($n = 121$) são equivalentes nas quatro regiões, com exceção da constante de deformação (k) e dos limites ($x_{\min} \leq x \leq x_{\max}$; $z_{\min} \leq z \leq z_{\max}$).



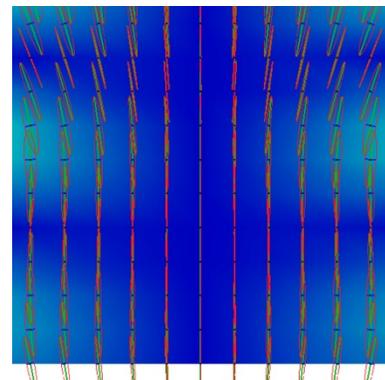
- b) Parâmetros da região A: $k = 2,5$ e $(-5 \leq x \leq 5; 0 \leq z \leq 10)$.



- c) Parâmetros da região B: $k = 10$ e $(40 \leq x \leq 50; 0 \leq z \leq 10)$.



- d) Parâmetros da região C: $k = 10$ e $(15 \leq x \leq 25; 19 \leq z \leq 29)$.



- e) Parâmetros da região D: $k = 50$ e $(-5 \leq x \leq 5; 40 \leq z \leq 50)$.

Figura 6.4 – Visualização da trajetória das partículas.

A trajetória das partículas situadas próximas à descontinuidade de impedância das camadas L_{25} a L_{27} pode ser observada na Figura 6.4d. As características retilínea, elíptica e circular das trajetórias, assim como as inclinações dos eixos, indicam a ocorrência de sobreposição de ondas de Rayleigh com ondas de corpo. A linearidade das trajetórias das partículas na região exibida na Figura 6.4e indica o predomínio das ondas de corpo. À medida que a profundidade aumenta, as ondas de corpo tendem a dominar o padrão de propagação das ondas.

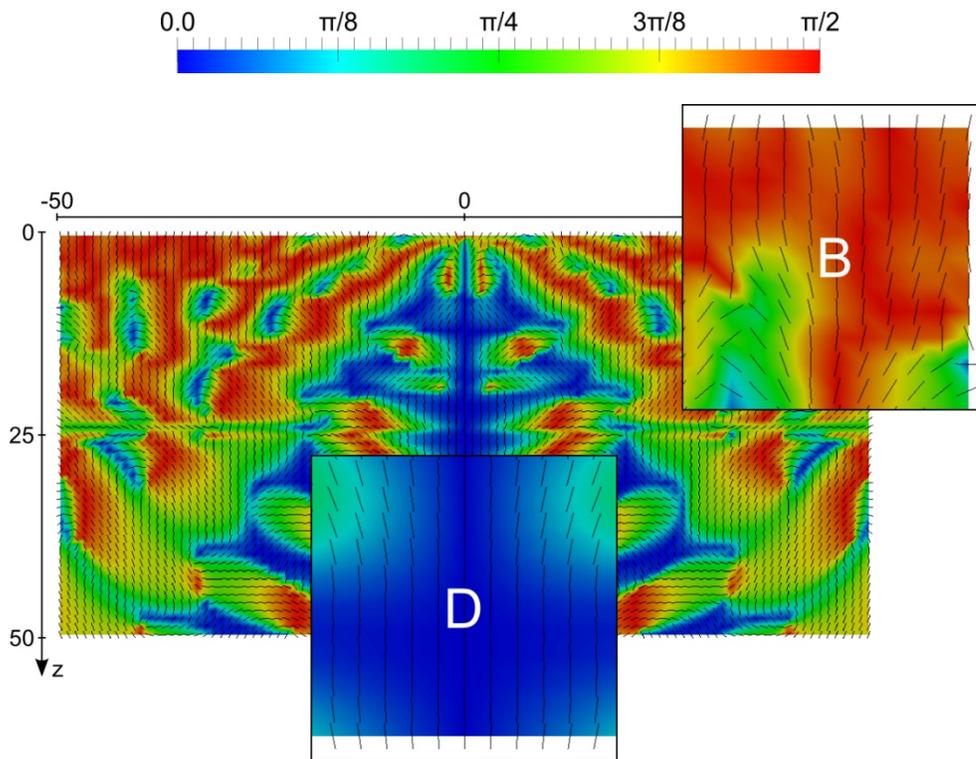


Figura 6.5 – Visualização do ângulo de inclinação das trajetórias das partículas.

6.1.2.4 Visualização do ângulo de inclinação

A Figura 6.5 apresenta a visualização do ângulo de inclinação das trajetórias das partículas. Ciente de que as ondas P e S puras formam, respectivamente, ângulos iguais a 0° e 90° entre as direções de deslocamento das partículas e de propagação das ondas e que estas ondas descrevem trajetórias retilíneas, então é possível identificar a partir das Figuras 6.3 e 6.5 regiões que evidenciam a presença destes tipos de ondas. Na região D (ver Figura 6.4), por exemplo, é possível afirmar que ocorre predomínio das ondas P, já que a relação de aspecto e o ângulo de inclina-

ção são nulos. Este mesmo artifício pode ser usado para a identificação das ondas de Rayleigh, cujos eixos maiores das elipses que descrevem as trajetórias formam ângulo de 90° com a direção de propagação das ondas. Nas regiões próximas aos cantos superiores das Figuras 6.3 e 6.5 é possível identificar a presença destas ondas.

6.1.2.5 Visualização do sentido do movimento de rotação

A Figura 6.6 apresenta a visualização do sentido do movimento de rotação das partículas, que pode ser horário (CW) ou anti-horário (CCW). Pela Figura 6.3 foi possível verificar que a relação de aspecto das trajetórias das partículas próximas à superfície varia ao redor de 0,65. Pela Figura 6.4c foi possível confirmar a trajetória elíptica das partículas e verificar que os eixos maiores das elipses estão visualmente alinhados com a superfície do meio e que a amplitude destas trajetórias decresce rapidamente com a profundidade. E pela Figura 6.5 foi possível confirmar a perpendicularidade dos eixos maiores com a direção de propagação das ondas. Pela Figura 6.6 é possível confirmar a previsão das imagens anteriores. Como na região B (ver Figura 6.4) o sentido do movimento de rotação é horário, ou seja, retrógrado à propagação das ondas, então é possível afirmar que há predominância de ondas de Rayleigh nesta região.

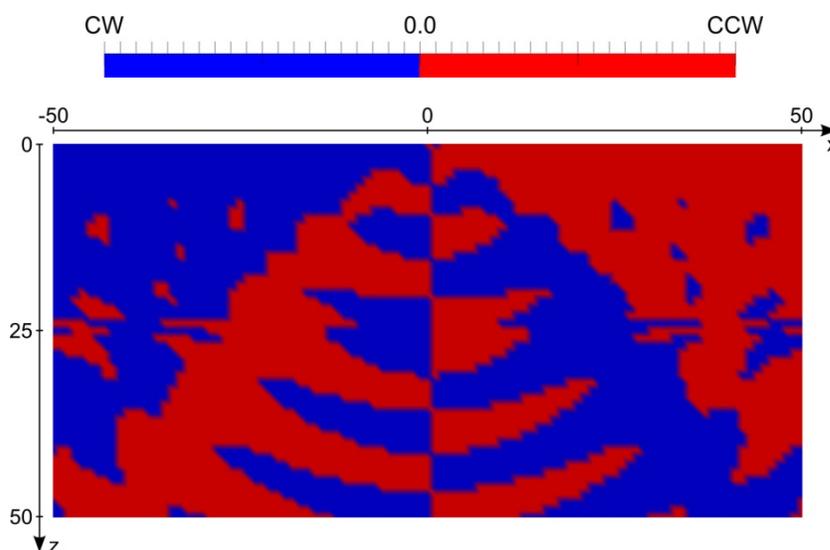


Figura 6.6 – Visualização do sentido do movimento de rotação.

6.1.2.6 Visualização do ângulo de fase

Por relacionar as componentes reais e imaginárias dos campos de deslocamento estacionários, o ângulo de fase expressa uma informação útil sobre o processo de propagação das ondas. A Figura 6.7 apresenta a visualização do ângulo de fase das componentes de deslocamento horizontal u_x e vertical u_z . Como as ondas de frente tendem a apresentar o mesmo ângulo de fase, então por uma análise da Figura 6.7b é possível notar a presença de dois tipos de frentes de ondas: retilínea (plana em 2D) e circunferencial (esférica em 2D).

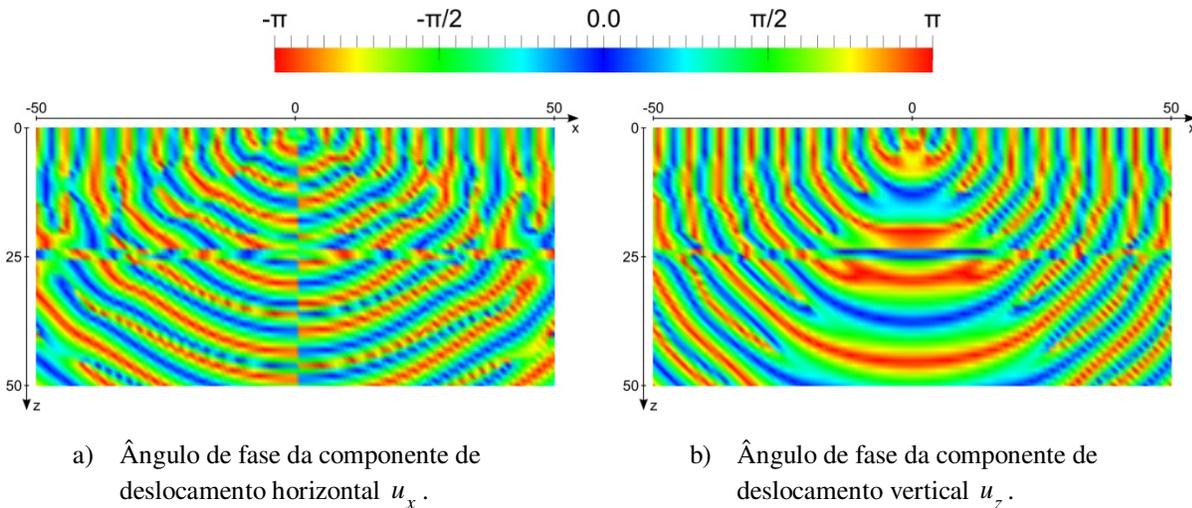


Figura 6.7 – Visualização do ângulo de fase das componentes de deslocamento horizontal e vertical.

Próximo à superfície e se deslocando para o fim do domínio, as frentes de ondas são retilíneas e normalmente indicam a presença de ondas de Rayleigh. Esta imagem ajuda a confirmar que as ondas de Rayleigh são predominantes na superfície. Através do meio, as frentes de ondas são circunferenciais, uma característica das ondas de corpo. As frentes de ondas também podem apresentar as discontinuidades de impedância. Estas informações podem ser obtidas da Figura 6.7a, porém com maior dificuldade.

6.1.2.7 Visualização da amplitude

A Figura 6.8 exibe a visualização da amplitude das componentes de deslocamento horizontal e vertical. O decaimento repentino da amplitude na região B (ver Figura 6.3) foi comprovado pela visualização da trajetória das partículas. Outra maneira é pela amplitude das componentes, como apresentado na Figura 6.10. Preferencialmente, neste caso, pela Figura 6.8b, cuja componente é vertical. As linhas de fluxo (em branco) indicam os gradientes das amplitudes e podem ser interpretados como sendo as trajetórias das partículas livres no meio.

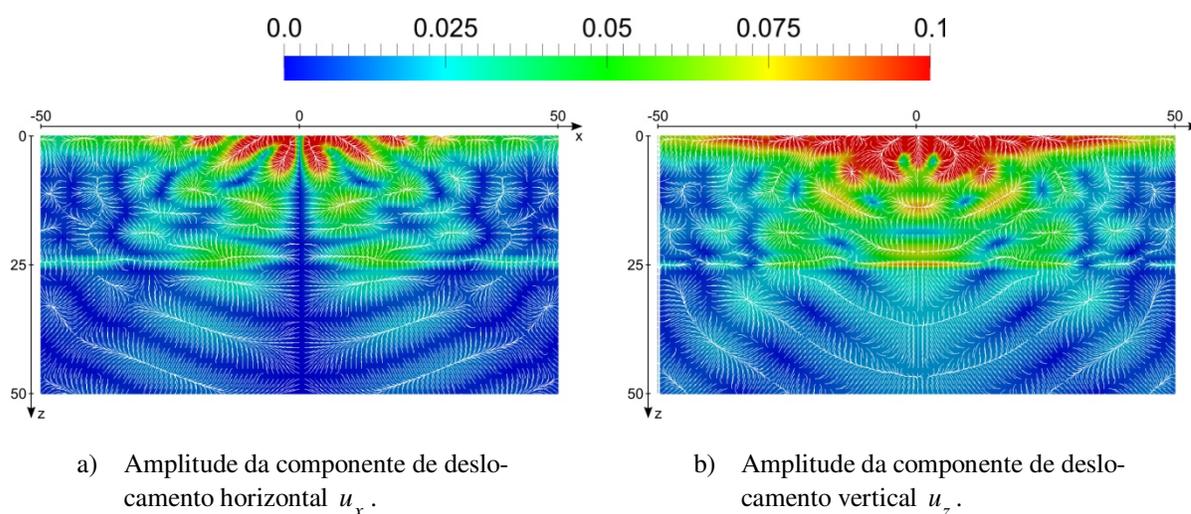


Figura 6.8 – Visualização da amplitude das componentes de deslocamento horizontal e vertical.

6.2 ONDAS MECÂNICAS 3D

Nas Seções 6.2.1 e 6.2.2 são exibidas a descrição de um problema geomecânico 3D e as visualizações geradas pelo MWPVis, respectivamente.

6.2.1 Descrição do problema

A Figura 6.9 ilustra um problema geomecânico 3D composto por um sistema de cargas dinâmicas de uma fundação rígida que interage com um semiespaço viscoelástico. O estudo deste

problema é fundamental para o projeto de fundações de máquinas e a análise sísmica de estruturas, por exemplo. Para a simulação deste problema foram usados os seguintes parâmetros: força dinâmica horizontal $F_x = F_0 \cos \omega t$, com $F_0 = 10 \text{ kN}$ e frequência circular $\omega = 63,25 \text{ rad/s}$, ou frequência $f = 10 \text{ Hz}$; bloco cilíndrico rígido com altura $h = 2 \text{ m}$ e raio $a = 1 \text{ m}$; e meio viscoelástico homogêneo e isotrópico de parâmetros constitutivos: densidade $\rho = 2 \cdot 10^3 \text{ kg/m}^3$, módulo de cisalhamento $G = 8 \cdot 10^6 \text{ N/m}^2$ e coeficiente de Poisson $\nu = 0,25$. Como o meio é isotrópico, então os parâmetros c_{ij} podem ser desprezados neste caso²¹. Os dados foram gerados sobre uma grade uniforme de espaçamento igual a $0,2 \text{ m}$ e com dimensões $71 \times 71 \times 51$, totalizando 257.091 pontos de amostragem. O domínio do semiespaço está limitado entre os intervalos $-7 \leq x \leq 7$, $-7 \leq y \leq 7$ e $0 \leq z \leq 10$.

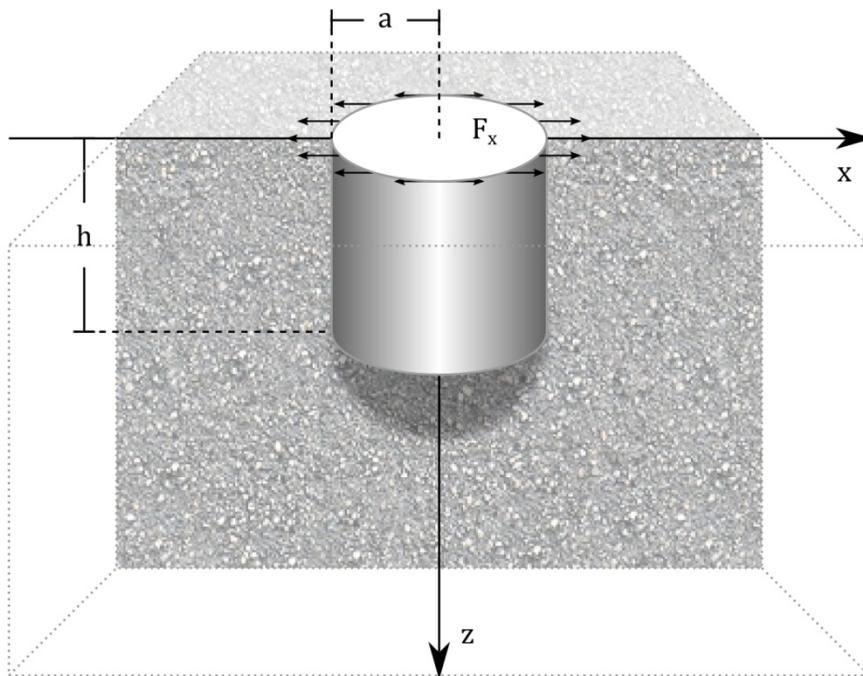


Figura 6.9 – Problema geomecânico 3D de um bloco cilíndrico rígido em um meio isotrópico.

²¹ Barros (2006) exibe informações mais detalhadas da modelagem matemática deste problema geomecânico 3D.

6.2.2 Visualizações

Para a análise do problema descrito anteriormente, foram geradas as visualizações apresentadas nas seções seguintes. Ao contrário do que foi apresentado na Seção 6.1.2, cujo objetivo era extrair informações a partir das visualizações para ampliar a compreensão dos fenômenos, neste caso o objetivo principal é validar a modelagem do problema.

6.2.2.1 Visualização do deslocamento

A Figura 6.10 apresenta as visualizações dos deslocamentos horizontais u_x e u_y , e vertical u_z das partículas. Nos conjuntos de imagens {a, d, g, j}, {b, e, h, k} e {c, f, i, l} são exibidos, respectivamente, os deslocamentos ao longo dos eixos x , y e z . Nos conjuntos de imagens {a, b, c}, {d, e, f}, {g, h, i} e {j, k, l} são apresentados os diferentes planos de corte sobre os eixos: π_x , π_y e π_z . Para as visualizações foram usados os seguintes parâmetros: instante de tempo $t = 0$, constante de deformação $k = 20$, mapeamento em cores “arco-íris” variando de $-0,025$ a $0,025$ e isolinhas nos deslocamentos nulos.

Pelo conjunto de imagens {a, d, g, j}, em comparação com as demais imagens respectivamente à direita, pode-se verificar que os deslocamentos ao longo do eixo x têm maior intensidade. Isto confirma o esperado, dado que a tensão superficial F_x é aplicada na mesma direção deste eixo. Como os deslocamentos são de baixa intensidade, pelo uso da constante de deformação é possível ampliar estes valores de maneira que a deformação do meio causada pelas ondas e a identificação do bloco rígido seja facilitada. Estes podem ser vistos em todas as imagens. Com a animação dos deslocamentos fica evidente o efeito de propagação das ondas, tanto pela deformação da grade quanto pela variação das cores.

Em todas as imagens é possível verificar que não ocorre reflexão de ondas nos limites do domínio, como é de fato dado que o domínio é ilimitado. Pelo conjunto de imagens {c, f, i, l} é possível verificar que os deslocamentos verticais são ínfimos e se concentra na região superficial em torno do bloco rígido. Por elas, ainda é possível verificar que o deslizamento da fundação com o solo foi desconsiderado na simulação. Pelo conjunto de imagens {e, h, k} pode-se verificar

que não ocorrem deslocamentos ao longo do eixo y nas regiões de fatia sobre do bloco rígido. Isto confirma o esperado, devido à simetria horizontal das propriedades elásticas do meio.

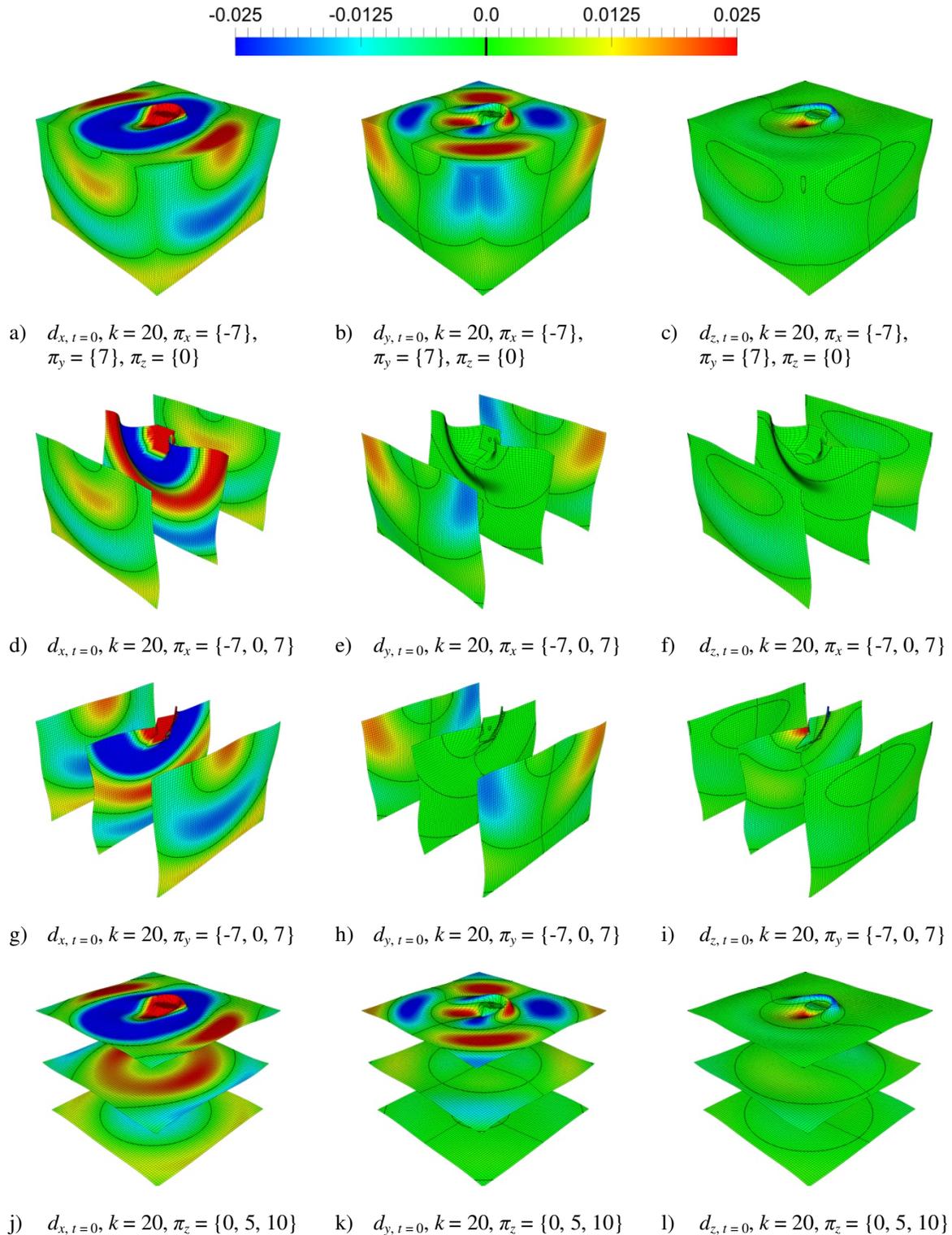


Figura 6.10 – Visualizações dos deslocamentos em x , y e z no instante de tempo $t = 0$.

6.2.2.2 Visualização da relação de aspecto

A Figura 6.11 exibe as visualizações da relação de aspecto das trajetórias elípticas das partículas. Pelas Figuras 6.11b e 6.11c é possível verificar que a trajetória das partículas é predominantemente linear abaixo do corpo rígido, como esperado pela simetria horizontal do domínio e pela direção da força dinâmica. Isto caracteriza a propagação de ondas de corpo nestas regiões. Pela Figura 6.11d é possível verificar que com o aumento da profundidade as ondas de corpo tendem a dominar o padrão de ondas. As regiões com cores diferentes de azul indicam a ocorrência de sobreposição entre ondas de corpo e de superfície.

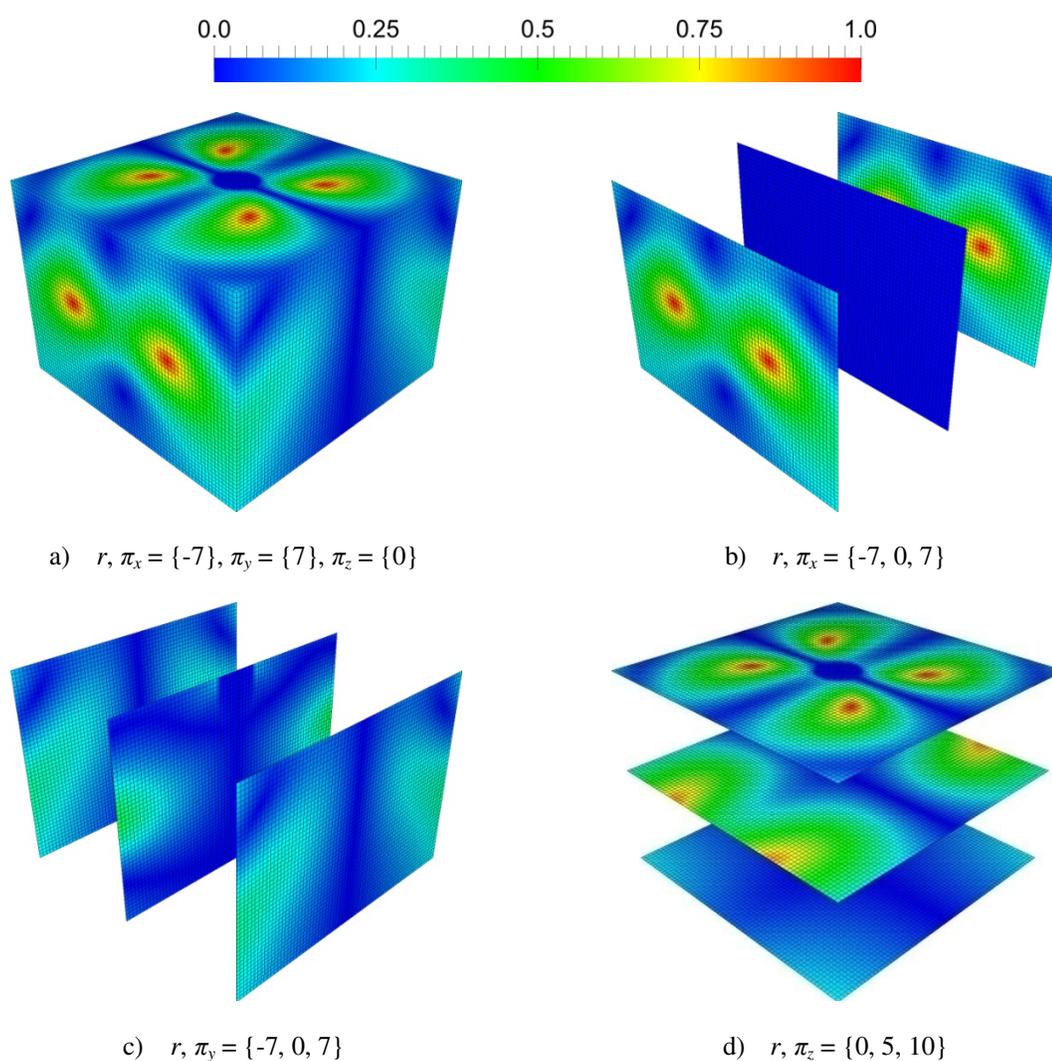


Figura 6.11 – Visualizações da relação de aspecto das trajetórias elípticas das partículas.

6.2.2.3 Visualização da amplitude

Na Figura 6.12 são apresentadas as visualizações das amplitudes das componentes verticais e horizontal. Nestes casos foram usados os seguintes parâmetros: mapeamento em cores “arco-íris” variando de 0 a 0,05 e isolinhas nas intensidades iguais a 0,025. Por estas imagens é possível confirmar as informações extraídas pela visualização dos deslocamentos. Porém, sem a necessidade de animação e deformação da grade.

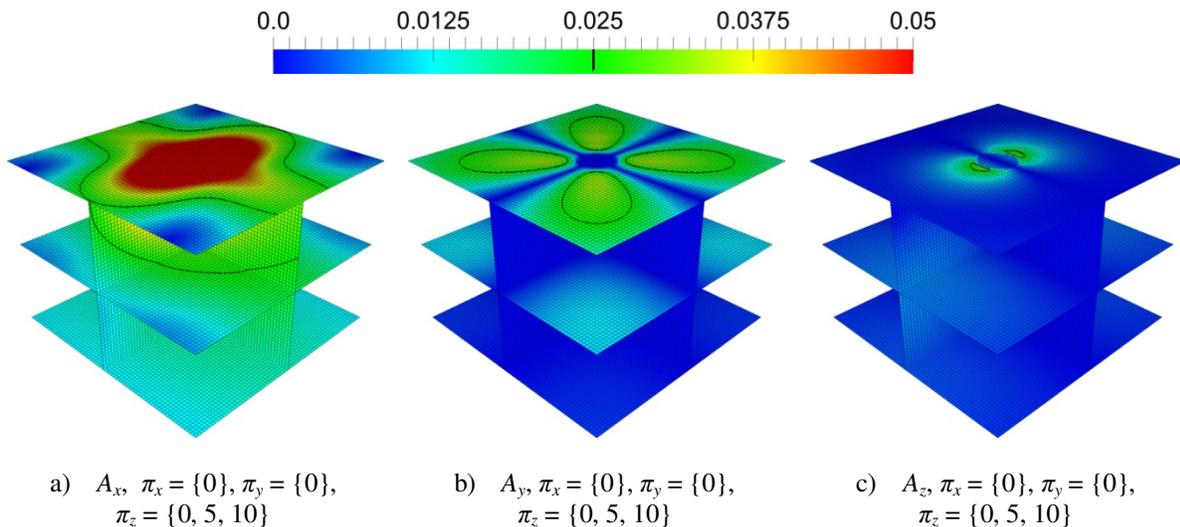


Figura 6.12 – Visualizações das amplitudes das ondas ao longo dos eixos x , y e z .

A Figura 6.12a mostra que os deslocamentos das partículas são mais intensos ao longo do eixo x , de mesma direção da força, e se estende ao interior do meio. Pela Figura 6.12b pode-se verificar que os deslocamentos em y se concentram nas diagonais do domínio e reduzem rapidamente com o aumento da profundidade. Por fim, a imagem Figura 6.12c mostra que os deslocamentos verticais se concentram ao redor da região superficial do bloco rígido.

6.2.2.4 Visualização do ângulo de fase

A Figura 6.13 apresenta visualização do ângulo de fase da componente horizontal u_x nos planos $\pi_y = \{-7, 0, 7\}$. Pela imagem é possível verificar a ocorrência de frentes de ondas planas

nas regiões próximas às extremidades superiores dos planos frontal e traseiro. Esta informação é importante para a construção de trincheiras, pois ajudam na determinação das profundidades mínimas necessárias para que se possa minimizar ou impedir a propagação das ondas de superfícies para as regiões de vizinhança. Vale lembrar que estas ondas são as responsáveis pelos efeitos danosos provocados pela perturbação do meio, tais como vibrações e ruídos. As trincheiras com profundidade abaixo do necessário não impedem a propagação das ondas de superfície. As trincheiras com profundidade acima do necessário dificultam ou inviabilizam as suas construções.

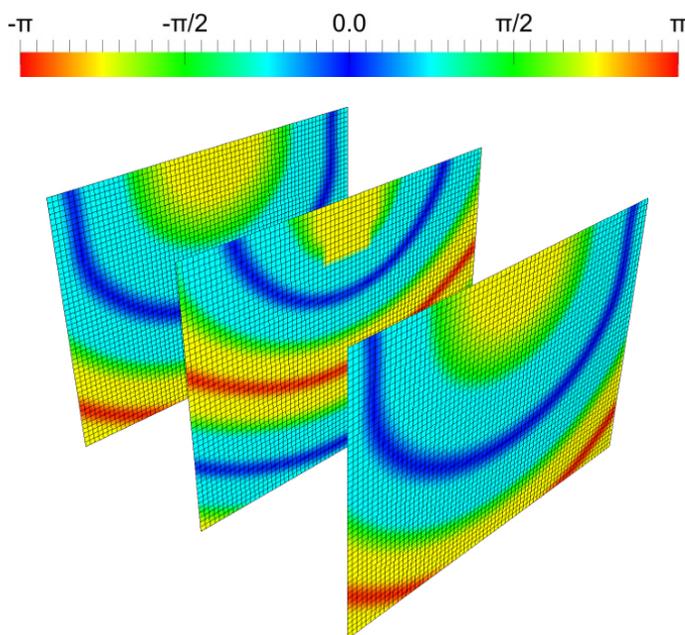


Figura 6.13 – Visualização do ângulo de fase da componente horizontal u_x .

6.3 CONSIDERAÇÕES FINAIS

Neste capítulo foram descritos dois estudos de casos para verificação e validação das visualizações geradas pelo programa descrito no capítulo anterior. Ambos são problemas geomecânicos estudados pelo GMCMC. Os dados utilizados foram cedidos pelo grupo e as validações foram realizadas por dois de seus integrantes. No próximo capítulo são apresentadas as avaliações das visualizações produzidas pelo ambiente descrito no capítulo anterior. O conteúdo deste capítulo e da Seção 4.2, sobre enriquecimento de dados, foi útil para o desenvolvimento do Capítulo 7.

7 DISCUSSÃO E AVALIAÇÃO DAS VISUALIZAÇÕES

Neste capítulo é apresentada a avaliação das visualizações geradas pelo ambiente de visualização desenvolvido. Nas Seções 7.1 e 7.2 são expressos os relacionamentos entre os atributos de dados, as análises para obtenção de informações e as técnicas de visualização mais eficientes.

7.1 INFORMAÇÕES OBTIDAS PELOS ATRIBUTOS DE DADOS

Nesta seção são apresentadas as informações que podem ser extraídas a partir da visualização dos atributos de dados descritos na etapa de enriquecimento de dados (ver Seção 4.2). Estas informações foram obtidas por intermédio das análises realizadas por especialistas da área de mecânica computacional, nos estudos de caso apresentados no Capítulo 6.

7.1.1 Variação temporal

Como já visto, as partículas do meio de propagação variam de posição em função do tempo e estas mudanças são ocasionadas pelas passagens das ondas. Sendo assim, com a animação da transição destas posições ao longo do tempo é possível visualizar o efeito de propagação das ondas no meio. As Figuras 3.2 e 3.3 demonstram como uma série de imagens, mesmo pequena, é capaz de transmitir este efeito. Entretanto, a propagação não é evidenciada apenas pelos deslocamentos dos pontos em si, mas também pelos atributos de dados que são dependentes do valor temporal, como os deslocamentos. Pela visualização das componentes de deslocamento é possível notar o efeito de propagação das ondas.

7.1.2 Fator de escala

Em casos nos quais os deslocamentos das partículas são ínfimos, a visualização é dificultosa. Com o fator de escala é possível aumentar estes deslocamentos de tal maneira que a visuali-

zação seja facilitada. Na Seção 6.2.2.1 foi apresentada uma necessidade de uso deste fator. Contudo, duas condições devem ser consideradas quanto ao uso do fator de escala: (1) a visualização não é real, já que ocorre ampliação/redução dos deslocamentos; e (2) podem ocorrer falhas na representação visual do conjunto de dados, já que os vértices da grade podem atravessar as faces de seus vértices vizinhos.

7.1.3 Amplitude das ondas

A amplitude é equivalente ao deslocamento de maior magnitude das partículas sobre os eixos de coordenadas. Pela visualização das amplitudes máxima e mínima global, os especialistas podem identificar regiões de forte impacto de ondas, a posição da fonte de perturbação e de regiões de descontinuidade de impedância. Em casos simples, como os ilustrados na Figura 3.2, as amplitudes das componentes de deslocamento horizontal e vertical podem ser usadas para identificação do tipo de onda de corpo que se propaga no meio material. Se as amplitudes das componentes horizontal e vertical forem, respectivamente, positiva e nula, então ocorre propagação de ondas P (ver Figura 3.2b). Se os valores forem opostos, então ocorre propagação de ondas S (ver Figura 3.2c).

Em casos mais complexos, as informações abstraídas na análise anterior já não são possíveis apenas pela amplitude das componentes, sendo necessário o uso da visualização de outros atributos complementares. Por exemplo, a amplitude pode ser usada, como foi na Seção 6.1.2.7, para ajudar na confirmação da propagação de ondas de Rayleigh em regiões superficiais específicas, já que uma das características destas ondas é o decaimento repentino da amplitude com o aumento da profundidade.

7.1.4 Relação de aspecto

A relação de aspecto expressa a razão entre os eixos menor e maior das elipses que descrevem as trajetórias das partículas. Ela pode ser empregada para identificação do formato das trajetórias, que variam entre retilínea e circular. A relação de aspecto nula indica a ocorrência de ondas de corpo, já que as trajetórias destas ondas são retilíneas. As trajetórias elípticas e circula-

res podem representar duas situações, ocorrência de sobreposição de ondas ou propagação de ondas de Rayleigh, cujos parâmetros constitutivos das camadas do meio são determinantes para obtenção do valor da relação de aspecto. A excentricidade de uma elipse é uma propriedade que também pode ser usada para obtenção destas mesmas informações.

7.1.5 Deslocamentos

Os deslocamentos, que variam em função do tempo, podem ser empregados para visualização do efeito de propagação das ondas (ver Seção 6.2.2.1) e dos fenômenos de propagação, tal como transmissão e reflexão. Os deslocamentos direcionados com o vetor direcional da propagação das ondas evidenciam mais claramente estas informações. Tal como pela amplitude, os deslocamentos também possibilitam a identificação da localização da fonte de perturbação e das regiões de forte impacto de ondas e de descontinuidade de impedância. Contudo, para isto é necessário uma visualização da animação de uma série de deslocamentos.

7.1.6 Ângulo de inclinação

Como informado, as ondas de corpo fazem com que as partículas do meio oscilem em trajetórias retilíneas. O que diferencia as ondas P das S, ambas ondas de corpo, é direção dos deslocamentos em relação à direção de propagação das ondas. O ângulo de inclinação expõe justamente esta relação. Como isso, quando a relação de aspecto é nula (trajetória retilínea) e o ângulo de inclinação também é nulo, pode-se notar a presença de ondas P. Em uma mesma situação, mas com ângulo de inclinação igual a 90° , pode-se notar a presença de ondas S.

Ao menos para o caso 2D, o ângulo de inclinação também pode ser aplicado para identificação das ondas de Rayleigh. Como estas ondas, quando puras, têm trajetórias elípticas com o eixo principal perpendicularmente alinhado com a direção de propagação das ondas, então se pode notar a presença delas em situações nas quais a relação de aspecto é compatível e o ângulo de inclinação é igual a 90° .

7.1.7 Trajetória

A trajetória é uma maneira de adquirir as mesmas informações fornecidas pela relação de aspecto e pelo ângulo de inclinação, porém por estimativa visual²², que é imprecisa. Pela trajetória de deslocamento é possível verificar a forma da trajetória das partículas (relação de aspecto) e a inclinação do eixo principal em comparação à direção de propagação das ondas (ângulo de inclinação). Porém, como as informações são obtidas por estimativa visual, em situações complexas, a análise é dificultada.

Três problemas podem ser destacados para estes casos: (1) os deslocamentos de baixa amplitude são visualmente imperceptíveis; (2) com o uso do fator de escala, na tentativa de solucionar este problema, os deslocamentos de baixa amplitude se tornam mais visíveis, contudo as trajetórias das partículas de alta amplitude se sobrepõem de modo que a visualização é comprometida sobre estas regiões; e (3) as partículas distantes e/ou não alinhadas com a fonte de perturbação dificultam a visualização do grau de inclinação de suas trajetórias, já que para isso é necessário uma estimativa visual da direção de propagação das ondas.

Para os problemas 3D, como o apresentado na Seção 6.2, um quarto problema pode surgir: como a trajetória das partículas é representada por linhas no espaço, sua visualização é confusa, principalmente em grades com elevado número de pontos. Neste caso, o mais indicado é a relação de aspecto ou a visualização das trajetórias projetadas sobre os planos.

7.1.8 Sentido do movimento de rotação

O sentido do movimento de rotação das trajetórias das partículas pode ser usado para a identificação das ondas de Rayleigh, que são retrógradas em relação ao sentido de propagação das ondas. Há relação entre as defasagens das componentes horizontal e vertical, para o caso 2D, e os sentidos de rotação das partículas²³. O sentido do movimento de rotação também pode ser obtido pela trajetória. Porém, para este caso também é necessário o emprego da variação temporal para reposicionamento das partículas ao longo do tempo. A obtenção do sentido fica limitada para

²² Estimativa numérica de proporção ou percentagem baseada na capacidade visual humana (HEALEY; BOOTH; ENNS, 1996).

²³ Informações complementares disponíveis em Pavlic (2009).

uma ou poucas partículas vizinhas, já que os dois primeiros problemas da trajetória podem ocorrer.

7.1.9 Ângulo de fase

Pelo ângulo de fase das ondas é possível identificar os tipos de frente de onda que ocorrem no meio. As frentes de ondas, por sua vez, ajudam no reconhecimento dos tipos de ondas, de corpo ou de superfície, que se propagam pelo meio. Nas Seções 6.1.2.6 e 6.2.2.4, as visualizações dos ângulos de fase ajudaram na identificação das ondas de Rayleigh, que são altamente destrutivas.

7.2 EFICIÊNCIA DAS TÉCNICAS DE VISUALIZAÇÃO

Nesta seção são apresentadas as técnicas mais eficientes para a interpretação dos atributos de dados gerados na etapa de enriquecimento de dados descrita na Seção 4.2. Os resultados foram obtidos por meio de uma avaliação realizada com os especialistas e de seus relatos durante os testes realizados. O formulário de avaliação desenvolvido no contexto deste trabalho é apresentado no Anexo A²⁴.

7.2.1 Mapeamento em cores

A técnica de mapeamento em cores se demonstrou como sendo a mais eficaz para a visualização dos diversos tipos de atributos escalares, desde os invariantes no tempo, como a relação de aspecto, quanto os variantes no tempo, como os deslocamentos. Isto pode ser explicado pelo efeito de transição contínua dos atributos escalares e pela alta sensibilidade do sistema visual às

²⁴ No Anexo A consta o “Questionário de Avaliação do MWPVis” que foi aplicado para avaliar o programa. Algumas informações estão desatualizadas em relação à versão final do aplicativo, pois este foi atualizado com as sugestões obtidas ao longo do processo de avaliação do mesmo. Nos Anexos B e C são apresentados o “Manual do Usuário do MWPVis” da versão 0.4.14 do programa e o “Formulário de Respostas do Questionário de Avaliação do MWPVis”, respectivamente.

cores, que favorece a interpretação dos dados (WARE, 2004). Pelas imagens apresentadas nos estudos de caso (ver Capítulo 6), pode-se verificar que o mapeamento em cores foi aplicado em todas as visualizações.

7.2.2 Contorno

Individualmente, a técnica de contorno não se provou eficaz para a visualização de atributos escalares não variantes no tempo, uma vez que as isolinhas informam apenas o local de transição e não qual das regiões delimitadas se encontra abaixo ou acima do valor de contorno. Com isso, não ocorre o efeito de variação gradativa dos valores escalares. Para os atributos escalares variantes no tempo, apesar das mesmas dificuldades, é possível ao menos visualizar o efeito de propagação das ondas. Contudo, quando juntamente aplicada com o mapeamento em cores, a técnica de contorno se demonstrou eficiente para a delimitação das regiões de interesse sem que haja perda do efeito de continuidade dos valores escalares, o que favorece a interpretação dos dados.

7.2.3 Função de altura

A função de altura se demonstrou eficaz para a análise dos atributos escalares, principalmente para os problemas 2D e quando empregado juntamente com a técnica de mapeamento em cores. Com uma ressalva, quando os vértices da grade são móveis, ou seja, se deslocam com o decorrer do tempo, a visualização pela função de altura é prejudicada, pois ocorrem duas deformações: uma gerada pela variação do tempo e outra pela variação da altura.

7.2.4 Glifos de vetor

Apesar de nenhuma imagem ter sido usada para exemplificar o uso dos glifos de vetor nos estudos de caso (ver Capítulo 6), pelas avaliações dos especialistas, esta técnica se demonstrou bastante eficaz para a visualização dos atributos vetoriais, uma vez que as grandezas que com-

põem estes, ou seja, a magnitude, a direção e o sentido, são preservadas quando visualmente apresentadas. Uma ressalva é que como as magnitudes dos campos vetoriais decaem rapidamente a partir da fonte de perturbação, a visualização é prejudicada pela ocorrência de sobreposição de glifos. Para isto, em situações nas quais a magnitude pode ser desprezada, os glifos podem ser representados na forma normalizada.

7.2.5 Linhas de fluxo

Em situações nas quais se pretende visualizar o “prolongamento” dos glifos, como para os atributos de gradiente da amplitude ou do eixo maior das trajetórias elípticas, a técnica de linhas de fluxo se demonstrou eficaz. Entretanto, esta técnica exige alto processamento e consumo de memória.

7.2.6 Extração de subgrade, recorte e fatiamento

As técnicas de recorte e de extração de subgrade podem ser empregadas para reduzir a região de interesse a ser visualizada. A técnica de fatiamento pode ser usada, por exemplo, para a visualização dos deslocamentos das partículas sobre as linhas geradas pela técnica.

7.2.7 Trajetória de partícula

A técnica de trajetória de partícula aplicada para a visualização das trajetórias das partículas é eficiente para a representação dos mesmos. Contudo, ela é dependente das novas coordenadas dos vértices da grade a cada variação de tempo e sua qualidade de representação é determinada pelo número de intervalos de tempos (quadros de animação) usados.

7.3 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentadas as avaliações das visualizações produzidas pelo ambiente de visualização desenvolvido descrito no Capítulo 5. As avaliações foram baseadas nos estudos de caso apresentados no Capítulo 6 e nos relatos dos especialistas consultados. No capítulo a seguir, são relatadas as conclusões deste trabalho.

8 CONCLUSÕES

Neste capítulo final são apresentadas as conclusões deste trabalho, assim como as contribuições, as limitações e as sugestões de trabalhos futuros.

8.1 CONSIDERAÇÕES FINAIS

O estudo do comportamento dos fenômenos de propagação de ondas mecânicas e de suas consequências frequentemente requer o desenvolvimento de métodos computacionais para resolução de problemas complexos em MMC. Estes métodos computacionais, em geral, são baseados nos Métodos das Diferenças Finitas, dos Elementos Finitos e de Contorno e consistem de modelagem matemática e simulação computacional de problemas físicos, que na maioria das vezes substituem ou reduzem a necessidade do uso de protótipos ou infraestrutura de laboratórios para ensaio dos mesmos. Entretanto, o processo de simulação computacional produz uma volumosa quantidade de dados numéricos que torna o procedimento de análise dos resultados uma tarefa árdua e extremamente difícil sem o suporte de técnicas para mapeamento destes dados em representações visuais.

Ciente de que a visualização de dados é uma área de estudo da computação que visa ampliar a cognição e que possui um conjunto de técnicas fortemente consolidado para representação visual de dados numéricos escalares e vetoriais, por exemplo, estes conhecimentos foram empregados para auxiliar os especialistas do domínio da aplicação a: (1) validar mais rapidamente os seus métodos computacionais produzidos; e (2) analisar com mais facilidade os resultados de suas simulações computacionais, o que permitiria um estudo mais eficiente do problema a ser solucionado.

Como os dados gerados por simulação computacional, em sua forma bruta, não possuem informações suficientes para uma forte exploração dos fenômenos de propagação de ondas mecânicas, foi necessário realizar um procedimento de enriquecimento de dados para agregar informação aos mesmos. Este procedimento foi descrito na Seção 4.2. Para apresentar os resultados do trabalho, um protótipo de ambiente de visualização foi desenvolvido e seus fundamentos de im-

plementação apresentados no Capítulo 5. Por fim, no Capítulo 6, foram realizados dois estudos de caso que comprovaram que os objetivos do presente trabalho atingidos.

8.2 CONTRIBUIÇÕES

As seguintes contribuições podem ser consideradas como resultados finais deste trabalho: (1) identificação de um conjunto de informações (ou atributos de dados escalares e vetoriais) derivadas dos dados brutos gerados por simulação computacional que pode ser aplicado para análise dos fenômenos de propagação de ondas mecânicas: deslocamento, posição relativa, trajetória, relação de aspecto, sentido do movimento de rotação, ângulo de inclinação, amplitude das ondas, ângulo de fase das ondas, fator de escala, sobreposição das ondas e gradiente da amplitude das ondas; (2) metodologia e respectiva formulação matemática para cálculo destas informações; (3) desenvolvimento de um protótipo de ambiente de visualização específico para dados de problemas geomecânicos que descrevem o comportamento das ondas; e (4) estudo e definição de diretrizes de técnicas de visualização de dados escalares e vetoriais para visualização do conjunto de informações.

8.3 LIMITAÇÕES E TRABALHOS FUTUROS

Este trabalho foi inicialmente desenvolvido para a visualização dos fenômenos de propagação de ondas mecânicas 2D. Contudo, devido à demanda de visualização de casos 3D, o mesmo foi posteriormente adaptado para atender a ambos os casos. Apesar disto, como nenhuma técnica específica para visualização de dados volumétricos foi utilizada neste trabalho, o mesmo ficou limitado à representação de dados gerados sobre grades de dimensão 2D para ambos os tipos de ondas. Considerando isto, segue como primeira sugestão de trabalhos futuros expandir o trabalho para a visualização de ondas mecânicas 3D e cujas grades também sejam de dimensão 3D. O que exigiria a aplicação das técnicas de visualização volumétrica e possivelmente de programação em Unidade de Processamento Gráfico (GPU – *Graphics Processing Unit*).

Com informado na Seção 2.1, a visualização científica agrega várias tecnologias, dentre as quais a visão computacional. A técnica de fluxo ótico²⁵ deste campo de estudo pode ser empregada nas imagens renderizadas que correspondem à visualização de dados escalares que variam no tempo, como os deslocamentos (ver Seção 4.2.1). A aplicação desta segunda sugestão exibiria as trajetórias de partículas livres que acompanhariam as frentes de ondas. O resultado desta aplicação tende a ser similar à visualização dos eixos maiores das elipses que descrevem a trajetória das partículas pela técnica de linhas de fluxo.

Tratando-se do ambiente de visualização desenvolvido, no Apêndice C é descrito um conjunto de sugestões de melhorias do MWPVis que pode ser implementado por futuros desenvolvedores.

²⁵ Informações sobre esta técnica em Horn (1986).

REFERÊNCIAS

- ADOLPH, M.; MESQUITA, E.; CARVALHO, E. R.; ROMANINI, E. Numerically evaluated displacement and stress solutions for a 3D viscoelastic half space subjected to a vertical distributed surface stress loading using the Radon and Fourier transforms. **Numerical Methods in Engineering**, v. 23, n. 8, p. 787-804, ago. 2007. DOI: <10.1002/cnm.932>.
- AHRENS, J.; GEVECI, B.; LAW, C. ParaView: an end-user tool for large-data visualization. In: HANSEN, C. D.; JOHNSON, C. R. (Eds.). **The visualization handbook**. Burlington, MA, USA: Elsevier Butterworth-Heinemann, 2005. cap. 26, p. 717-731. ISBN: 0-12-387582-X.
- BARROS, P. L. A.; MESQUITA, E. Elastodynamic Green's functions for orthotropic plane strain continua with inclined axes of symmetry. **International Journal of Solids and Structures**, v. 36, n. 31-32, p. 4764-4788, nov. 1999. DOI: <10.1016/S0020-7683(98)00264-9>.
- BARROS, P. L. A. Impedances of rigid cylindrical foundations embedded in transversely isotropic soils. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 30, n. 7, p. 683-702, jan. 2006. DOI: <10.1002/nag.496>.
- CARD, S. K.; MACKINLAY, J. D.; SHNEIDERMAN, B. (Eds.). **Readings in information visualization: using vision to think**. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1999. (The Morgan Kaufmann Series in Interactive Technologies). ISBN: 1-55860-533-9.
- CARINO, N. J. Stress wave propagation methods. In: MALHOTRA, V. M.; CARINO, N. J. (Eds.). **Handbook on nondestructive testing of concrete**. 2. ed. Boca Raton, FL, USA: CRC Press, 2004. cap. 14. ISBN: 0-8031-2099-0.
- DEFANTI, T. A.; BROWN, M. D.; MCCORMICK, B. H. Visualization: expanding scientific and engineering research opportunities. **IEEE Computer**, v. 22, n. 8, p. 12-25, ago. 1989. DOI: <10.1109/2.35195>.
- FOWLER, C. M. R. **The solid earth: an introduction to global geophysics**. 2. ed. Cambridge, UK: Cambridge University Press, 2005. ISBN: 0-521-89307-0.
- GALLAGHER, R. S (Ed.). **Computer visualization: graphics techniques for scientific and engineering analysis**. Boca Raton, FL, USA: CRC Press, 1995. ISBN: 0-8493-9050-8.
- GARG, S.; GHOSH, C. K.; GUPTA, S. **Oscillations and waves**. [S.l.]: PHI Learning, 2009. ISBN: 978-81-203-3921-7.
- GRAFF, K. F. **Wave motion in elastic solids**. Mineola, NY, USA: Dover Publications, 1991, c1975. ISBN: 0-486-66745-6.
- HEALEY, C. G.; BOOTH, K. S.; ENNS, J. T. High-speed visual estimation using preattentive processing. **ACM Transactions on Computer-Human Interaction**, v. 3, n. 2, p. 107-135, jun. 1996. DOI: <10.1145/230562.230563>.

HORN, B. K. P. **Robotic vision**. Cambridge, MA, USA: MIT Press, 1986. (MIT Electrical Engineering and Computer Science Series). ISBN: 0-262-08159-8.

KITWARE. **The VTK user's guide**: updated for VTK version 5. [S.l.]: Kitware, 2006. ISBN: 1-930934-18-1.

LEITHOLD, L. **O cálculo com geometria analítica**. Tradução: Cyro de Carvalho Patarra. 3. ed. São Paulo, SP: Harbra, v. 1, 1994.

LINER, C. **Elements of 3D seismology**. 2. ed. [S.l.]: PennWell Books, 2004. ISBN: 1-59370-015-6.

LOWRIE, W. **Fundamentals of geophysics**. Cambridge, UK: Cambridge University Press, 1997. ISBN: 0-521-46728-4.

LUTZ, M. **Learning Python**. 3. ed. Sebastopol, CA, USA: O'Reilly Media, 2008. ISBN: 978-0-596-51398-6.

MCCORMICK, B. H.; DEFANTI, T. A.; BROWN, M. D. Visualization in scientific computing. **IEEE Computer Graphics and Applications**, v. 7, n. 7, p. 63-72, jul. 1987. DOI: <10.1109/MCG.1987.277014>.

MESQUITA, E.; ADOLPH, M.; CARVALHO, E. R.; ROMANINI, E. Dynamic displacement and stress solutions for viscoelastic half-spaces subjected to harmonic concentrated loads using the Radon and Fourier transforms. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 33, n. 18, p. 1933-1952, dez. 2009. DOI: <10.1002/nag.802>.

MIEDER, W. **Proverbs**: a handbook. Westport, CT, USA: Greenwood Press, 2004. ISBN: 0-313-32698-3.

MONROE, J. S.; WICANDER, R. **The changing earth**: exploring geology and evolution. Belmont, CA, USA: Brooks/Cole, Cengage Learning, 2009. ISBN: 978-0-495-55480-6.

NAIK, T. T.; MALHOTRA, V. M.; POPOVICS, J. S. The ultrasonic pulse velocity method. In: MALHOTRA, V. M.; CARINO, N. J. (Eds.). **Handbook on nondestructive testing of concrete**. 2. ed. Boca Raton, FL, USA: CRC Press, 2004. cap. 8. ISBN: 0-8031-2099-0.

NOKIA, C. **Qt 4.6 whitepaper**. [S.l.]: Nokia Corporation, 2009. Disponível em: <<http://qt.nokia.com/files/pdf/qt-4.6-whitepaper>>. Acesso em: 19 jan. 2011.

PAVLIC, T. P. **Lissajous figures**. [S.l.: s.n.], 2009. Disponível em: <http://www.tedpavlic.com/teaching/osu/ece209/lab1_intro/lab1_intro_lissajous.pdf>. Acesso em: 10 nov. 2010.

PIZLO, Z. **3D shape**: its unique place in visual perception. Cambridge, MA, USA: MIT Press, 2008. ISBN: 978-0-262-16251-7.

RAJAPAKSE, R. K. N. D.; WANG, Y. Elastodynamic Green's functions of orthotropic half plane. **Journal of Engineering Mechanics**, v. 117, n. 3, p. 588-604, mar. 1991. DOI: <10.1061/(ASCE)0733-9399(1991)117:3(588)>.

RAJAPAKSE, R. K. N. D.; WANG, Y. Green's functions for transversely isotropic elastic half space. **Journal of Engineering Mechanics**, v. 119, n. 9, p. 1724-1746, set. 1993. DOI: <10.1061/(ASCE)0733-9399(1993)119:9(1724)>.

RIVERBANK, C. L. **PyQt whitepaper**. [S.l.]: Riverbank Computing Limited, 2009. Disponível em: <<http://www.riverbankcomputing.co.uk/software/pyqt/whitepaper>>. Acesso em: 18 jan. 2011.

SCHROEDER, W.; MARTIN, K.; LORENSEN, B. **The visualization toolkit**: an object-oriented approach to 3D graphics. 4. ed. Clifton Park, NY, USA: Kitware, 2006. ISBN: 1-930934-19-X.

SEN, T. K. **Fundamentals of seismic loading on structures**. Chichester, UK: John Wiley and Sons, 2009. ISBN: 978-0-470-01755-5.

TELEA, A. C. **Data visualization**: principles and practice. Natick, MA, USA: A K Peters, 2007. ISBN: 978-1-56881-306-6.

VENTURI, J. J. **Álgebra vetorial e geometria analítica**. 9. ed. Curitiba, PR: [s.n.], 1949. ISBN: 85-85132-48-5. Disponível em: <<http://www.geometriaanalitica.com.br/av/av.pdf>>. Acesso em: 11 jan. 2011.

WARE, C. **Information visualization**: perception for design. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2004. (The Morgan Kaufmann Series in Interactive Technologies). ISBN: 1-55860-819-2.

WEISZFLOG, W. (Ed.). **Michaelis**: moderno dicionário da língua portuguesa. São Paulo: Melhoramentos, 1998. ISBN 978-85-06-02759-2.

WRIGHT, H. **Introduction to scientific visualization**. [S.l.]: Springer, 2007. ISBN: 978-1-84628-494-6.

PUBLICAÇÕES

SANTOS, R. M. G.; DE MARTINO, J. M.; MESQUITA, E. Computer simulation and visualization of mechanical wave propagation phenomena in continuous medium. In: INTERNATIONAL CONFERENCE IN CENTRAL EUROPE ON COMPUTER GRAPHICS, VISUALIZATION AND COMPUTER VISION (WSCG), 19., 2011, Pilsen. SKALA, V. (Ed.). **WSCG'2011 poster papers proceedings**. Pilsen: Union Agency, 2011, p. 1-4. ISBN: 978-80-86943-81-7. Disponível em: <http://wscg.zcu.cz/WSCG2011/!_2011_WS-CG_Poster_Papers.pdf>. Acesso em: 22 fev. 2011.

APÊNDICES

APÊNDICE A – Configuração do ambiente de desenvolvimento do MWPVis

Este apêndice relata todos os procedimentos realizados para configurar o ambiente de desenvolvimento utilizado para a produção do MWPVis (ver Capítulo 5). Tem por objetivo auxiliar os desenvolvedores na preparação do ambiente de desenvolvimento para uso do código-fonte do programa. Todos os procedimentos estão descritos para o sistema operacional Windows Vista.

O MWPVis foi implementado em Python e é dependente dos pacotes PyQt e VTK. Como apenas o instalador do VTK não é distribuído para Python, então foi necessário compilar o seu código-fonte. Pela necessidade, a descrição deste processo também foi incluída neste apêndice.

O procedimento de configuração do ambiente de desenvolvimento foi dividido em cinco etapas: instalação dos pré-requisitos, criação do produto VTK, compilação e instalação do VTK, configuração do IDE e importação do MWPVis. Sendo que as três primeiras etapas estão relacionadas com o processo de compilação e instalação do VTK. E visando eliminar a repetição deste processo pelos desenvolvedores, foi criado um instalador que contém o VTK compilado com empacotador para Python e suporte de GUI para Qt, chamado “VTK for Python”. Este é automaticamente instalado e configurado no Python.

Todos os instaladores ou arquivos compactados dos programas utilizados neste trabalho estão contidos no DVD que acompanha esta dissertação. Dentre eles, a versão 2.8.2 do CMake, a versão 3.6 do Eclipse SDK, a versão 0.5.7 do MWPVis, a versão 1.5.9 do PyDev, a versão 4.7.3 do PyQt, a versão 2.6.5 do Python, a versão 4.6.3 do Qt, a versão 8.5.8 do Tcl, a versão 5.4.2 do VTK.

É aconselhável que todos os programas referenciados ao longo deste apêndice sejam instalados nos endereços e nas configurações recomendadas pelos seus respectivos instaladores ao longo do processo de instalação. O endereço C:\ será considerado como a raiz da estrutura de diretórios da unidade de disco rígido principal.

A.1 INSTALAÇÃO DOS PRÉ-REQUISITOS

A primeira etapa do processo de configuração do ambiente de desenvolvimento é a instalação dos pré-requisitos, que são os programas necessários para gerar os projetos usados para a

compilação e instalação do VTK com empacotador para Python e suporte de GUI para Qt. Dão-se como pré-requisitos o compilador do Qt e os interpretadores do Python e do Tcl. Sendo este último opcional e os demais obrigatórios.

Para instalar o Qt, basta executar o arquivo `qt-win-opensource-4.6.3-vs2008.exe` e seguir os procedimentos de instalação padrão do programa. Este será instalado no diretório `C:\Qt\4.6.3`. Uma versão mais recente do Qt pode ser obtida em seu *website*, no endereço <http://qt.nokia.com>.

Para instalar o Python, basta executar o arquivo `python-2.6.5.msi` e seguir os procedimentos de instalação padrão do programa. Este será instalado no diretório `C:\Python26`. Uma versão mais recente do Python pode ser obtida em seu *website*, no endereço <http://www.python.org>.

A instalação padrão do Python, como recomendada, inclui uma interface para o conjunto de ferramentas de GUI Tk, chamada Tkinter. E para que o VTK também tenha suporte de GUI para Tk é necessário instalar o Tcl. Para isto, basta executar o arquivo `ActiveTcl8.5.8.2.292682-win32-ix86-threaded.exe` e seguir os procedimentos de instalação padrão do programa. Este será instalado no diretório `C:\Tcl`. Uma versão mais recente do Tcl pode ser obtida em seu *website*, no endereço <http://www.tcl.tk>.

Para compilar o código-fonte do VTK é necessário um compilador C++. Como a Unicamp é parceira do programa Aliança Acadêmica da Rede de Desenvolvimento da Microsoft (MSDNAA – *Microsoft Developer Network Academic Alliance*), produtos para desenvolvimento de *software* da Microsoft podem ser utilizados em projetos de pesquisa sem nenhum custo aos alunos. Por isso, optou-se pelo compilador do Microsoft Visual C++ (MSVC) 2008. Além do que o sistema operacional usado também era um produto da Microsoft e isso favoreceria na compatibilidade do ambiente.

Note que o Qt instalado foi compilado no mesmo IDE adotado neste trabalho, o MSVC 2008. E em caso de uso de outro compilador C++, também será necessário compilar o código-fonte do Qt para que não ocorram erros durante o processo de compilação do VTK. O código-fonte do Qt está disponível em seu *website*.

Por fim, para que o interpretador do Python e o compilador do Qt sejam detectados pelo sistema operacional, é necessário incluir suas localizações na variável de ambiente Path do sistema. Ambos estão localizados, respectivamente, nos endereços `C:\Python26` e `C:\Qt\4.6.3\bin`. Note que não há necessidade de incluir o endereço do interpretador do Tcl, pois este é automaticamente adicionado durante sua instalação.

Para incluir endereços na variável de ambiente Path do Windows Vista, faça:

- 1) Execute o recurso Configurações Avançadas do Sistema (localizado no endereço Painel de Controle\Sistema e Manutenção\Sistema) e na aba Avançado, clique no botão Variáveis de Ambiente (ver Figura A.1);
- 2) No diálogo Variáveis de Ambiente, selecione a variável Path na lista de variáveis do sistema e clique em Editar;
- 3) No campo Valor da variável do diálogo Editar Variável de Sistema, inclua os endereços separando-os por ponto-e-vírgula. Ao término clique em OK para fechar o diálogo;
- 4) Clique em OK para encerrar o diálogo Variáveis de Ambiente e, novamente, em OK para encerrar a janela Configurações Avançadas do Sistema.

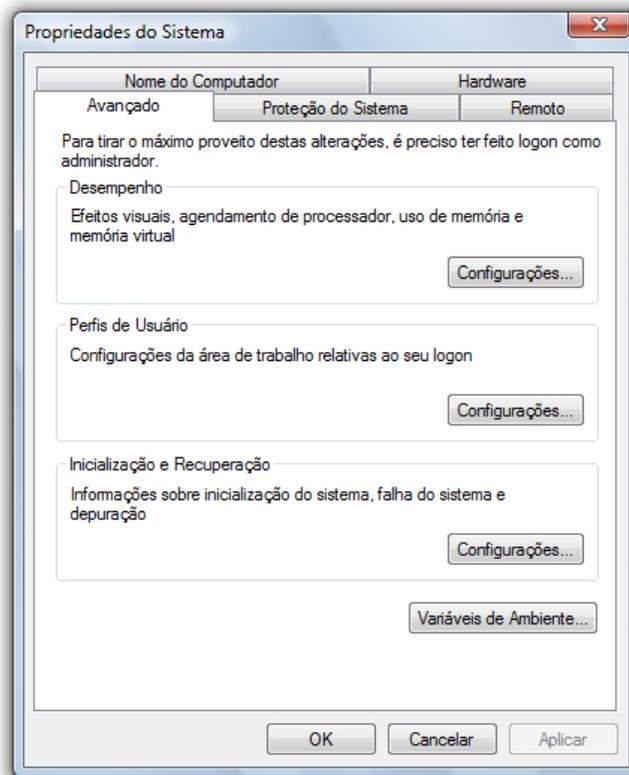


Figura A.1 – Janela de Configurações Avançadas do Sistema do Windows Vista.

Ao término desta etapa, todos os pré-requisitos necessários para gerar os projetos usados para a compilação e instalação do VTK já estão devidamente instalados e prontos para serem usados.

A.2 CRIAÇÃO DO PRODUTO VTK

A segunda etapa do processo de configuração do ambiente de desenvolvimento é a criação do produto VTK. Ou seja, do conjunto de projetos necessários para a compilação e instalação do VTK. Este conjunto de projetos é gerado com uso do CMake. Mas antes de usá-lo, é necessário preparar o local onde os projetos serão criados. Para isto, faça:

- 1) Crie uma pasta chamada VTK54, preferencialmente, no diretório C:\;
- 2) Crie três pastas chamadas build, install e src no diretório C:\VTK54;
- 3) Extraia todo o conteúdo existente no arquivo compactado vtk-5.4.2.zip no diretório C:\VTK54\src. Este arquivo contém o código-fonte do VTK. Uma versão mais recente pode ser obtida em seu *website*, no endereço <http://www.vtk.org>.

Para instalar o CMake, basta executar o arquivo cmake-2.8.2-win32-x86.exe e seguir os procedimentos de instalação padrão do programa. Este será instalado no diretório C:\Program Files\CMake 2.8. Para obter uma versão mais recente do CMake, acesse o seu *website*, no endereço <http://www.cmake.org>. Concluída a instalação, faça:

- 1) Execute o CMake e preencha os campos Where is the source code e Where to build the binaries, respectivamente, com os endereços C:\VTK54\src e C:\VTK54\build. Em seguida, clique em Configure;
- 2) No diálogo em aberto, selecione a opção Visual Studio 9 2008 no campo Specify the generator for this project. Clique em Finish e aguarde pela atualização da lista de variáveis (referida nas próximas etapas apenas por atualização);
- 3) Habilite ao menos as variáveis BUILD_SHARED_LIBS, VTK_WRAP_PYTHON e VTK_USE_GUISUPPORT (necessário habilitar a opção Advanced para visualizar esta última variável). Insira o endereço C:\VTK54\install no campo da variável CMAKE_INSTALL_PREFIX. Em seguida, clique em Configure e aguarde pela atualização. Caso não tenha optado pela instalação do Tcl, recomendada anteriormente, será emitido uma mensagem de erro informando que a variável VTK_USE_TK é dependente do Tcl. Desabilite-a ou instale o Tcl e refaça esta etapa;
- 4) Habilite a variável VTK_USE_QVTK, clique em Configure e aguarde pela atualização;

- 5) Digite o valor “4” no campo da variável DESIRED_QT_VERSION e habilite a variável VTK_USE_QVTK_QTOPENGL (necessário habilitar a opção Advanced para visualizar está variável). Clique em Configure e aguarde pela atualização;
- 6) Clique em Configure e, após a atualização, clique em Generate para gerar os projetos necessários para a compilação do VTK (ver Figura A.2). Por fim, finalize o CMake.

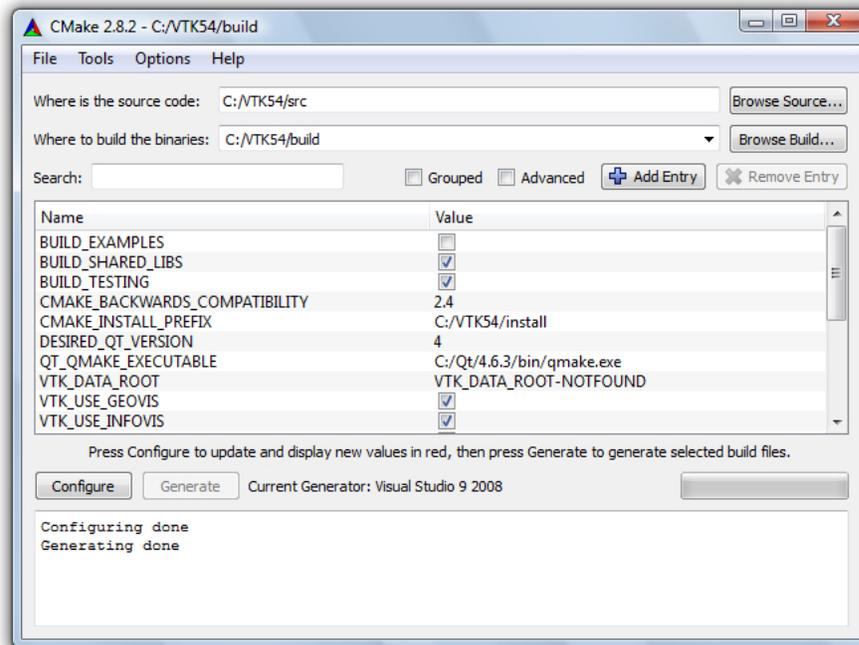


Figura A.2 – Uso do CMake para gerar os projetos necessários para a compilação do VTK.

Ao término desta etapa, o conjunto de projetos necessário para a compilação e instalação do VTK foi gerado no diretório C:\VTK54\build.

A.3 COMPILAÇÃO E INSTALAÇÃO DO VTK

A terceira etapa do processo de configuração do ambiente de desenvolvimento é a compilação e instalação do VTK. Para realizá-la, faça:

- 1) Execute o MSVC (preferencialmente com privilégios de administrador) e abra o arquivo VTK.sln localizado em C:\VTK54\build. Aguarde até o término da leitura do arquivo;

- 2) Acesse Configuration Manager (localizado em Build), selecione a opção Release do campo Active solution configuration e clique em Close. Aguarde até o término das alterações;
- 3) Selecione ALL_BUILD em Solution Explorer e, em seguida, clique em Build ALL_BUILD, localizado em Build (ver Figura A.3). O processo de compilação será iniciado, aguarde até o término (aproximadamente uma hora de duração);
- 4) Concluída a compilação, já é possível instalar o VTK. Para isto, selecione INSTALL em Solution Explorer e, em seguida, clique em Build Only INSTALL (localizado sucessivamente em Build e Project Only). O processo de instalação será iniciado, aguarde até o término (alguns minutos de duração) e encerre o MSVC.

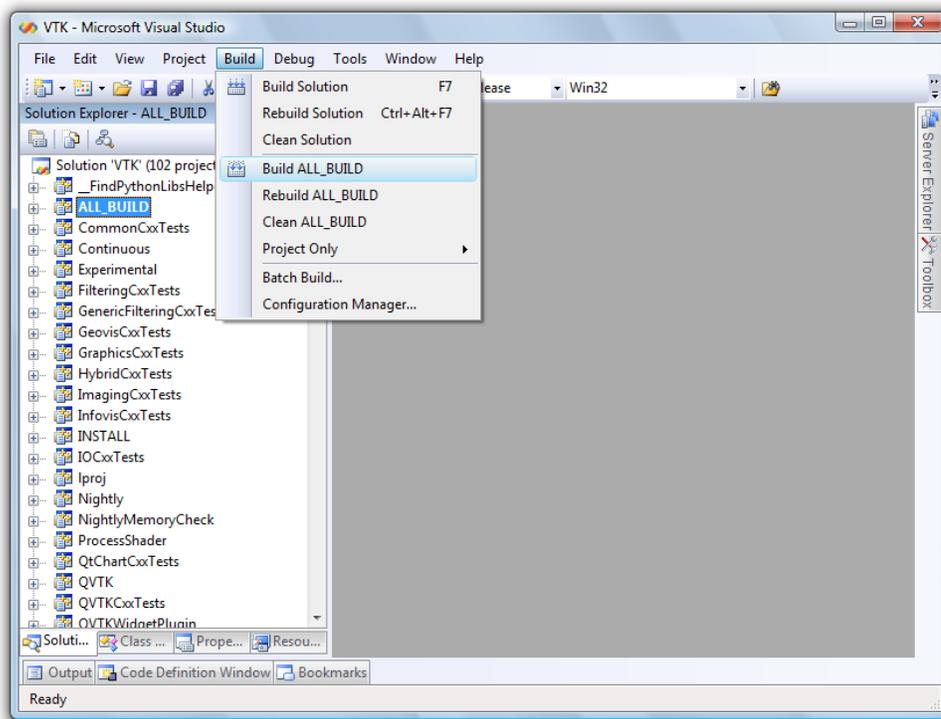


Figura A.3 – Início da compilação do VTK com uso do MSVC 2008.

Ao término desta etapa, o VTK foi compilado e instalado, respectivamente, nos diretórios C:\VTK54\build\bin\Release e C:\VTK54\install. Agora, resta apenas instalar o pacote do empacotador do VTK para Python. Para isto, faça:

- 1) Copie todo o conteúdo existente no diretório C:\VTK54\install\lib\site-packages para C:\Python26\Lib\site-packages;

- 2) Inclua o endereço C:\VTK54\install\bin na variável Path do sistema (ver procedimento na Seção A.1).

A.4 CONFIGURAÇÃO DO IDE

A quarta etapa do processo de configuração do ambiente de desenvolvimento é a configuração do IDE. Neste caso, como foi adotado o Eclipse, uma plataforma de *software* livre, será ensinado como instalá-lo e configurá-lo para que desenvolvedores possam usar o código-fonte do MWPVis. Entretanto, para isto não é exigido nenhum IDE específico.

Para instalar o Eclipse, basta descompactar o arquivo compactado eclipse-SDK-3.6-win32.zip, preferencialmente, no diretório C:\. Uma versão mais recente pode ser encontrada em seu *website*, no endereço <http://www.eclipse.org>. Em caso de *download*, sugere-se a versão clássica. Para usá-lo, basta executar o arquivo eclipse.exe existente no diretório C:\Eclipse.

O Eclipse não possui suporte ao Python embutido. Deste modo, será necessário instalar o PyDev, um excelente *plugin* para Eclipse, que torna este um ambiente favorável para desenvolvimento de aplicações em Python. Informações adicionais sobre o *plugin* podem ser encontradas no *website* do PyDev, no endereço <http://pydev.org>.

Há duas maneiras de instalar o PyDev: manualmente, por meio de descompactação do *plugin* (distribuído em arquivo compactado) em diretórios específicos do Eclipse; ou pelo próprio Eclipse, mediante a funcionalidade de instalação de novos programas. Neste documento será ensinada apenas a segunda forma de instalação. Para informações sobre o processo de instalação manual, acesse o manual de instruções disponível no *website* do PyDev.

Para instalar o PyDev pela funcionalidade de instalação de novos programas do Eclipse, siga os seguintes procedimentos:

- 1) Execute o Eclipse e acesse Install New Software (localizado em Help) para exibir a janela de instalação de novos programas (denominada Install);
- 2) Clique em Add e digite nos campos Name e Location do diálogo Add Repository, respectivamente, “PyDev” e “<http://pydev.org/updates>” (ver Figura A.4). Em seguida, clique em OK e aguarde até a conclusão da busca pelo *plugin* no repositório;

- 3) Selecione a opção PyDev na lista de programas apresentados e clique em Next para iniciar a instalação (ver Figura A.4);
- 4) Será exibida a versão do PyDev for Eclipse que será instalada. No caso a versão mais recente do programa disponível no repositório. Dê continuidade clicando em Next;
- 5) Aceite o contrato de licença selecionando a opção I accept the terms of the license agreement e clique em Finish para iniciar o processo de instalação do *plugin*;
- 6) Durante a instalação será solicitado que o usuário selecione os certificados desejados. Clique em Select All para selecionar todos os certificados e, em seguida, clique em OK. Vale ressaltar que é necessário selecionar ao menos um certificado para que a instalação não seja interrompida;
- 7) Para concluir a instalação, será necessário reiniciar o Eclipse. Para isto, clique em Restart Now no diálogo em aberto.

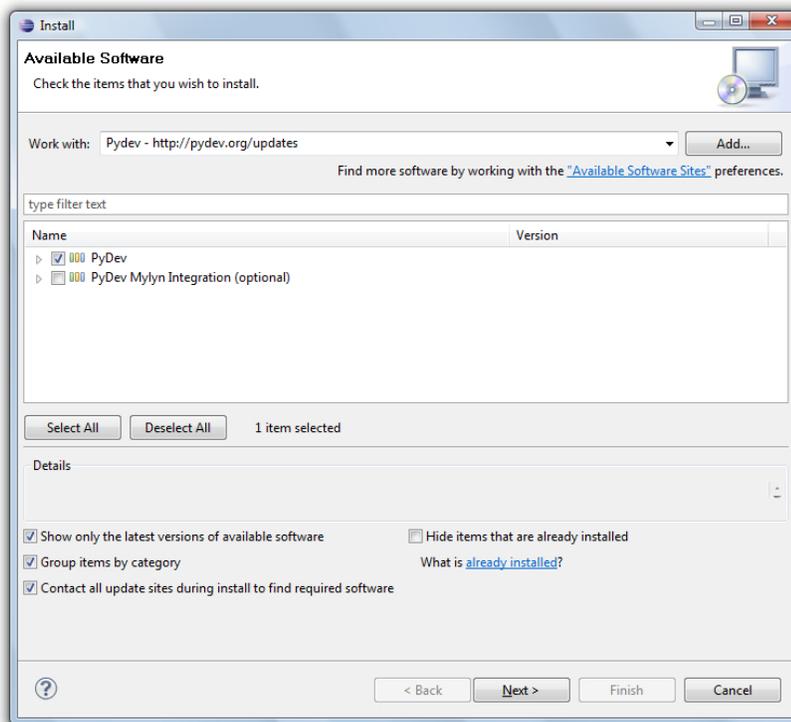


Figura A.4 – Instalação do PyDev pela funcionalidade de instalação de novos programas do Eclipse.

Após a instalação, será necessário configurar o PyDev com a localização do interpretador do Python. Para isto, faça:

- 1) Acesse Preferences (localizado em Window) e na janela em aberto acesse Interpreter - Python em Pydev;
- 2) Clique em Auto Config para que o PyDev detecte automaticamente a localização do interpretador Python instalado no computador. Vale ressaltar que para que isto ocorra é necessário que o caminho C:\Python26 esteja incluso na variável Path em Variáveis de Ambiente, como sugerido anteriormente;
- 3) Na janela em aberto, clique em OK para confirmar os diretórios do interpretador Python localizado;
- 4) Clique em Apply e aguarde até que o PyDev conclua a configuração. Terminado, clique em OK para concluir o processo (ver Figura A.5).

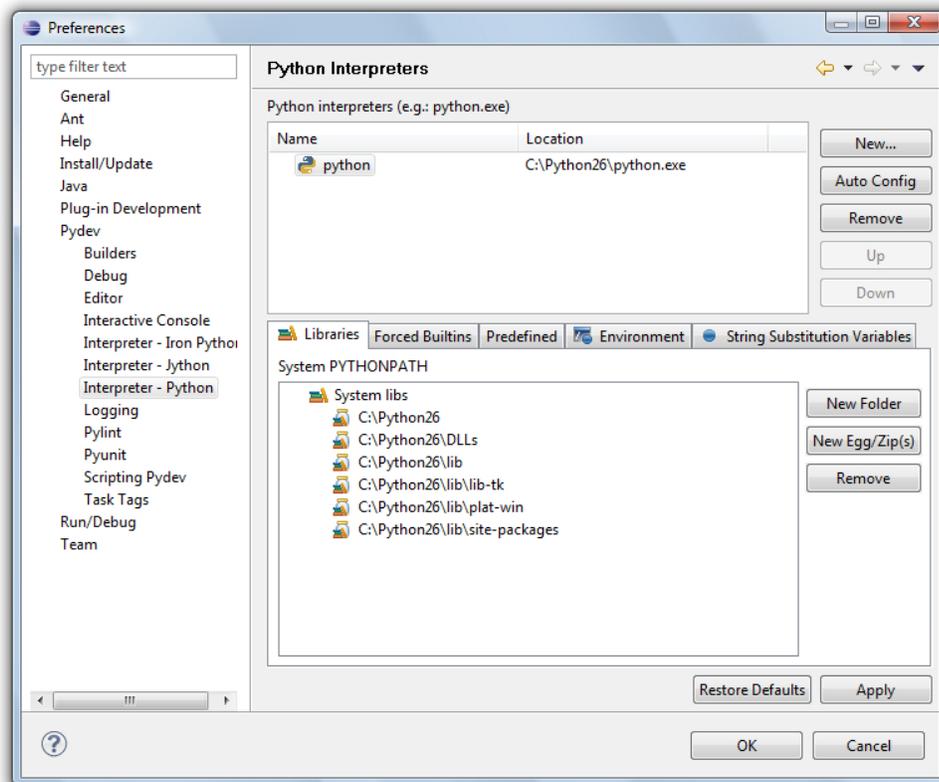


Figura A.5 – Localização do interpretador do Python.

Ao término desta etapa, o Eclipse está apto para interpretar o código-fonte do MWPVis.

A.5 IMPORTAÇÃO DO MWPVIS

A quinta e última etapa do processo de configuração do ambiente de desenvolvimento é a importação do MWPVis, mais especificamente, do código-fonte do MWPVis pelo Eclipse. Como já foi dito anteriormente, o MWPVis é dependente dos pacotes PyQt e VTK para Python. Portanto, será necessário instalá-los.

Para instalar o PyQt, basta executar o arquivo `PyQt-Py2.6-gpl-4.7.3-2.exe` e seguir os procedimentos de instalação padrão do programa. Este será instalado no diretório `C:\Python26\Lib\site-packages\PyQt`. Uma versão mais recente do PyQt pode ser obtida em seu *website*, no endereço <http://www.riverbankcomputing.com/software/pyqt>.

Se as três primeiras etapas do processo de configuração do ambiente de desenvolvimento foram realizadas, o VTK já deve estar instalado. Caso contrário, será necessário instalá-lo. Para isto, basta executar o arquivo `vtk54-py26-win32.exe` e seguir os procedimentos de instalação padrão do programa. Este será instalado no diretório `C:\Python26\Lib\site-packages\vtk`. Este instalador foi criado para este trabalho e não está disponível para *download* na internet.

Concluída as instalações dos pacotes, o código-fonte do MWPVis já pode ser importado pelo Eclipse. Para isto, faça:

- 1) Execute o Eclipse e pressione `Ctrl+N` para criar um novo projeto;
- 2) Na janela em aberto, selecione sucessivamente `Pydev`, `Pydev Project` e clique em `Next`;
- 3) No campo `Project name` digite “`mwpvis-0.5.7`” e, posteriormente, clique em `Finish` para concluir a criação do projeto. Caso a perspectiva atual do Eclipse não seja a apropriada para o projeto criado, um diálogo será exibido sugerindo a alteração para a perspectiva do `PyDev`. Clique em `Yes` para confirmar a alteração;
- 4) Selecione o projeto `mwpvis-0.5.7` em `Pydev Package Explorer` e acesse `Import` (localizado em `File`) para iniciar a importação do código-fonte do MWPVis;
- 5) Selecione `Archive File` em `General` e clique em `Next`;
- 6) Clique em `Browse`, localize o arquivo compactado `mwpvis-0.5.7-src` e, em seguida, clique em `Finish`. Aguarde até a conclusão da importação.

Para executar o programa, basta selecionar o arquivo `mwpvis.pyw` existente na pasta `src` do projeto criado, pressionar `Ctrl+F11` e selecionar a opção `Python Run` na janela em aberto.

APÊNDICE B – Instalação do MWPVis

Este apêndice descreve duas formas de instalação do MWPVis: pelo executável de autoinstalação, que contém apenas os arquivos binários do programa, tal como o executável e suas bibliotecas dependentes (Seção B.1); e pelo código-fonte do programa, que contém além do projeto com os códigos-fonte, os *scripts* para compilação e instalação do mesmo (Seção B.2).

B.1 VIA EXECUTÁVEL DE AUTOINSTALAÇÃO

Pela primeira forma de instalação do MWPVis, via executável de autoinstalação, basta executar o arquivo `mwpvis-0.5.7-win32.exe` e seguir os procedimentos de instalação padrão do programa. Este será instalado no diretório `C:\Program Files\MWPVis\0.5.7`. Um atalho para o programa será criado no menu Iniciar.

B.2 VIA CÓDIGO-FONTE DO PROGRAMA

Pela segunda forma de instalação do MWPVis, via código-fonte do programa, é necessário ter instalado previamente o interpretador do Python e os pacotes PyQt e VTK, dos quais o MWPVis é dependente. Para gerar os arquivos binários do MWPVis é usado o `py2exe`, um pacote que converte *scripts* escritos em Python para programas executáveis do Windows. Para instalá-lo, basta executar o arquivo `py2exe-0.6.9.win32-py2.6.exe` e seguir os procedimentos de instalação padrão do programa. Esse será instalado no diretório `C:\Python26\Lib\site-packages\py2exe`. Uma versão mais recente do pacote `py2exe` pode ser encontrada em seu *website*, no endereço <http://www.py2exe.org>. Tendo por instalados o interpretador do Python e os pacotes PyQt, VTK e `py2exe`, já é possível converter os códigos-fonte do projeto MWPVis, escritos em Python, para um arquivo executável. Para isto, faça:

- 1) Crie uma pasta chamada MWPVis, preferencialmente, em `C:\`;
- 2) Descompacte todo o conteúdo existente no arquivo compactado `mwpvis-0.5.7-win32-src.zip` no diretório `C:\MWPVis`;

- 3) Execute o arquivo `bdist.bat` existente no diretório `C:\MWPVis\src` para iniciar a conversão dos *scripts* contidos na pasta `mwpvis` para um arquivo executável. Este se chama `mwpvis.exe` e será criado no diretório `C:\MWPVis\bin`, juntamente com suas bibliotecas dependentes.

Para gerar o executável de autoinstalação do MWPVis, precisa-se executar o arquivo `setup.bat` existente no diretório `C:\MWPVis\setup`. Este é gerado com uso do compilador Sistema de Instalação por *Scripts* da Nullsoft (NSIS – *Nullsoft Scriptable Install System*). Para instalá-lo, basta executar o arquivo `nsis-2.46-setup.exe` e seguir os procedimentos de instalação padrão do programa. Este será instalado no diretório `C:\Program Files\NSIS`. Uma versão mais recente do NSIS pode ser encontrada em seu *website*, no endereço <http://nsis.sourceforge.net>.

Como o endereço do diretório de instalação do NSIS, onde se encontra o seu compilador, não é automaticamente incluso na variável de ambiente `Path` pelo instalador do programa, será necessário incluí-lo manualmente. No Apêndice A, constam informações sobre como incluir um ou mais endereços na variável de ambiente `Path` do Windows Vista.

Todos os instaladores dos programas referenciados neste tópico estão contidos no DVD que acompanha esta dissertação. Dentre eles, a versão 2.46 do NSIS, a versão 0.5.7 do MWPVis, a versão 0.6.9 do `py2exe`, a versão 4.7.3 do `PyQt`, a versão 2.6.5 do `Python` e a versão 5.4.2 do `VTK`. É aconselhável que estes programas sejam instalados nos endereços e nas configurações recomendadas pelos seus respectivos instaladores durante o processo de instalação.

APÊNDICE C – Sugestões de melhorias do MWPVis

Este trabalho foi inicialmente desenvolvido para a visualização dos fenômenos de propagação de ondas mecânicas 2D. Para a apresentação dos resultados obtidos, o ambiente de visualização MWPVis foi desenvolvido. Como uma forma de avaliar e validar estes resultados, um processo de avaliação das visualizações produzidas pelo programa foi realizado. Após este processo, por necessidade de um dos avaliadores, o trabalho foi expandido também para a visualização de ondas 3D. Contudo, nem todas as sugestões dadas foram implementadas. Por isso, neste apêndice são apresentadas as sugestões de melhorias.

C.1 FUNCIONALIDADES DE *SOFTWARE*

Nesta seção são apresentadas as sugestões de funcionalidades de *software* que não foram concluídas:

C.1.1 Gravar e recuperar estado da visualização

Para evitar que o usuário tenha que reconstruir as mesmas visualizações após cada encerramento de programa, o ideal é que houvesse uma forma de gravar em arquivo o estado da visualização para posterior recuperação. Como o VTK é orientado a objetos, a alternativa mais evidente seria o uso da serialização de objetos. Contudo, o VTK não permite que este procedimento seja realizado pelo Python. Uma forma possível seria com uso de XML.

C.1.2 Manter endereços dos últimos estados de visualização

A partir da funcionalidade anterior, um procedimento que poderia agilizar a busca dos arquivos a serem recuperados é a manutenção do histórico dos endereços dos últimos arquivos salvos/recuperados. Isto é muito comum nos programas atuais.

C.1.3 Alterar a resolução da imagem de tela

Para que o usuário possa armazenar imagens das visualizações produzidas, há a funcionalidade de captura de tela. Contudo, a qualidade da imagem está limitada ao tamanho da área da tela do monitor usado. Sendo assim, a funcionalidade de alteração da resolução seria importante.

C.1.4 Alterar a resolução e a taxa de quadros do vídeo

A mesma justificativa da funcionalidade anterior é válida para esta. Contudo, nos vídeos, além da qualidade de resolução, existe também a taxa de variação dos quadros para controle da velocidade da transição entre eles (ou animação).

C.1.5 Alterar as configurações para personalização do programa

Opções para que o usuário possa alterar as configurações para personalização do programa são recomendadas. Por exemplo, para alteração da cores dos vértices ou linhas da grade que representa os dados, da cor de fundo da área de visualização etc.

C.1.6 Ver o manual de ajuda

Essencial para ajudar o usuário de qualquer programa. No Anexo B é apresentado o “Manual do Usuário do MWPVis” da versão 0.4.14 do programa. Este documento acompanhava o “Questionário de Avaliação do MWPVis” (ver Anexo A) no processo de avaliação do mesmo. Se atualizado, este mesmo documento pode ser usado para a criação do manual de ajuda do programa.

C.1.7 Ocultar a janela de saída de erros gerada pelo VTK

Há situações em que uma janela para emissão de erros do VTK é exibida. Apesar desta ação não interferir no funcionamento do MWPVIs, para evitar que o usuário interprete isto como um erro do programa, o ideal é que esta funcionalidade do VTK fosse omitida.

C.2 FLUXO DE VISUALIZAÇÃO

Nesta seção são apresentadas as sugestões de melhorias relacionadas ao fluxo de visualização proposto.

C.2.1 Importação

Para os dados de dimensão 3D, apenas as superfícies do volume são apresentadas na visualização. Contudo, mesmo assim, é necessário que todos os pontos sejam carregados e convertidos. Isto exige alto tempo de processamento e consumo desnecessário de memória, ao menos quando as regiões de interesse estão limitadas à superfície. Por isso, seria ideal que as amostras internas fossem ignoradas durante a importação de dados ou, para casos nas quais as regiões de interesse são internas ao volume, permitir que usuário opte antecipadamente quais os planos (fatias) serão visualizados.

C.2.2 Filtragem/enriquecimento

Um recurso que melhoraria a qualidade visual é a reamostragem por interpolação dos vértices da grade. Isto em contrapartida exigiria maior consumo de memória e tempo de processamento. Por isso, a reamostragem (superamostragem) poderia ser aplicada ao menos para a geração das imagens ou dos vídeos a serem gerados pelo usuário.

C.3 TÉCNICAS DE VISUALIZAÇÃO

Nesta seção são apresentadas as sugestões de melhorias relacionadas às técnicas de visualização.

C.3.1 Mapeamento em cores

Como apresentado na Seção 2.4.1, a técnica de mapeamento em cores pode ser empregada para a visualização das regiões de fronteiras entre intervalos de valores escalares, assim como fornecida pela técnica de contorno. Para este efeito, chamado de faixa de cores, deve-se permitir a redução do número de cores usado para o mapeamento de tal maneira que a distinção entre as cores seja visivelmente perceptível.

C.3.2 Contorno

Como visto na Seção 6.2.2, o mapeamento em cores favorece a visualização dos escalares por contorno. Por isso, o mapeamento em cores pode ser aplicado também sobre as linhas e não somente sobre a superfície da grade, como foi implementado.

C.3.3 Glifos de vetor e linhas de fluxo

Em casos no qual existe relação entre atributos escalares e vetoriais, como os deslocamentos, que podem ser representados de ambas as formas, o mapeamento em cores pode ser aplicado não somente sobre a grade, mas também sobre os glifos de vetor e/ou linhas de fluxo.

C.3.4 Trajetória de partícula

Dado que a trajetória das partículas pode ser construída a partir de duas posições (ver Seção 4.2.3), não haveria necessidade de construir a visualização das trajetórias a partir da obtenção de todas as posições intermediárias como foi implementado.

C.3.5 Composição de técnicas

O MWPVis permite que o usuário use mais de uma técnica de visualização em uma mesma imagem, tanto para analisar as mesmas informações, quanto para analisar situações diferentes. Um exemplo é o uso das técnicas de mapeamento em cores e de contorno em uma mesma imagem. Contudo, não é possível usar uma mesma técnica para representar duas ou mais informações em uma mesma imagem. Por exemplo, se o investigador quer analisar a relação entre a visualização da relação de aspecto e a visualização do ângulo de inclinação por mapeamento em cores, ele precisa gerar duas imagens. Para estes casos, as técnicas de processamento de imagens digitais podem ser usadas para mesclar as duas imagens, de maneira que a relação entre as duas informações possam ser analisadas em apenas uma imagem, ao invés de duas.

C.3.6 Gráficos de linhas

Para ajudar a interpretação numérica dos dados a partir das visualizações geradas, dois tipos de gráficos de linhas podem ser desenvolvidos: (1) gráfico dos valores escalares em função dos índices de um conjunto de pontos de amostragem pré-estabelecido e (2) gráfico dos valores escalares de um único ponto de amostragem em função do tempo normalizado. A Figura C.1 apresenta e exemplifica estes dois tipos de gráficos. No primeiro é apresentada a variação dos valores ($v_{\min} \leq v \leq v_{\max}$) dos escalares 1, 2 e 3 em função do conjunto de pontos ($P_{0 \leq i \leq 13}$). No segundo é apresentado a variação dos valores dos escalares 1, 2 e 3 de um único ponto em função do tempo ($t_{0 \leq i \leq 13}$).

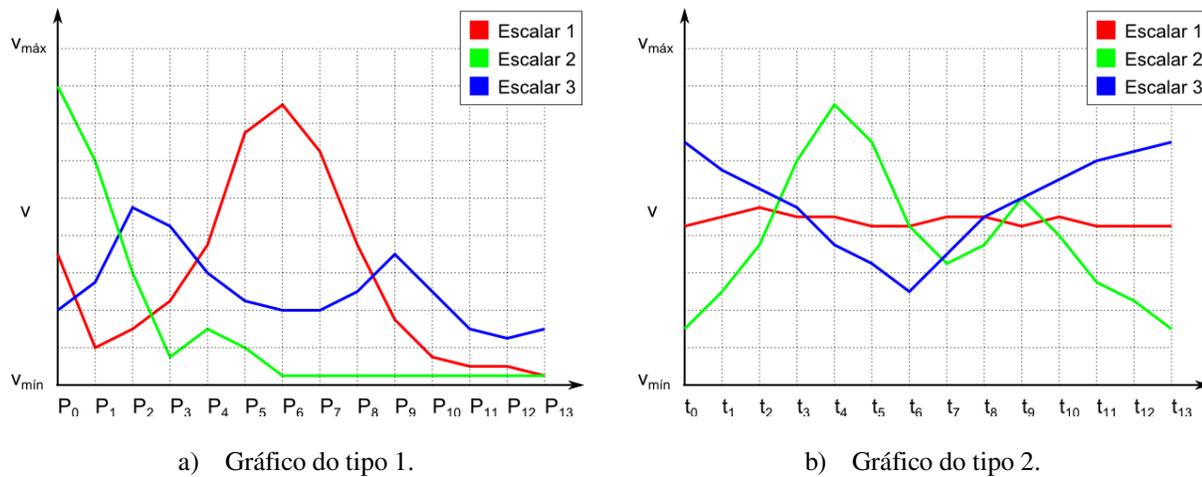


Figura C.1 – Gráficos de linhas para análise dos atributos de dados escalares.

O primeiro gráfico se encontra disponível no MWPVis. Entretanto, a biblioteca usada para gerar o gráfico (VTK) não oferece muitos recursos. O Qt disponibiliza métodos para criação de gráficos de funções. Estes podem ser utilizados para melhoria deste gráfico e criação do segundo sugerido.

C.4 TÉCNICAS DE COMPUTAÇÃO GRÁFICA

Nesta seção são apresentadas as sugestões de melhorias relacionadas às técnicas de computação gráfica aplicadas no programa.

C.4.1 Animação

Para evitar o reprocessamento das posições relativas dos vértices da grade a cada intervalo de tempo de animação, várias grades foram criadas, sendo cada uma para uma para representar um instante de tempo específico, como descrito no Capítulo 5. Em contrapartida, há alto consumo de memória para dados com grande quantidade de pontos de amostragem.

Quando o número de quadros escolhido pelo usuário é reduzido, a velocidade da animação é extremamente alta. Isto dificulta a análise dos atributos de dados variantes no tempo, como

o deslocamento. Um procedimento para resolver este problema seria permitir que o usuário controlasse a taxa de quadros por segundo da animação.

C.4.2 Múltiplas visões

Na versão apresentada, as múltiplas visões permitem que a tela seja subdividida para apresentação de uma mesma visualização em perspectivas diferentes: nos planos xy , yz , zx e outra personalizada pelo usuário. Outra forma sugerida é que as subdivisões de tela permitissem múltiplas visualizações, como de atributos diferentes em cada uma delas. Isto facilitaria a comparação de atributos relacionados, mas que não podem (por motivos de *software* e/ou de visualização) ser representado em uma mesma visualização (imagem).

ANEXOS

ANEXO A – Questionário de Avaliação do MWPVis

Questionário de Avaliação do MWPVis

Campinas, 11 de agosto de 2010.

Desenvolvedores do MWPVis: Rodrigo Mogni Gonçalves dos Santos e José Mario De Martino
Universidade Estadual de Campinas – Unicamp
Faculdade de Engenharia Elétrica e de Computação – Feec
Departamento de Engenharia de Computação e Automação Industrial – DCA

Prezado(a) Senhor(a):

Em primeiro lugar, gostaríamos de agradecer-lhe por ter aceitado nosso convite para ser um dos avaliadores do MWPVis. Esperamos, com sua contribuição, obter informações que nos ajudarão a avaliar com maior exatidão a qualidade do projeto de dissertação de mestrado do estudante Rodrigo Mogni Gonçalves dos Santos, do curso de Mestrado em Engenharia Elétrica, e de seu orientador Prof. Dr. José Mario De Martino. Daqui adiante, ao longo deste texto, nós estaremos instruindo-lhe dos procedimentos de avaliação do MWPVis.

Este é o **Questionário de Avaliação do MWPVis**. Juntamente com ele, o(a) senhor(a) deverá ter recebido um CD com uma cópia do **Pacote de Avaliação do MWPVis**, que contém, além deste documento, o executável de auto-instalação do programa, um conjunto de arquivos de exemplos, o **Manual do Usuário do MWPVis** e o **Formulário de Respostas do Questionário de Avaliação do MWPVis**. Caso não tenha recebido o CD, por favor, acesse o site na *Web* do MWPVis, no endereço <http://www.dca.fee.unicamp.br/~mogni/mwpvis>, e efetue o *download* do **Pacote de Avaliação do MWPVis**.

É realmente imprescindível que o(a) senhor(a) familiarize-se previamente com a metodologia do ambiente de trabalho do programa antes de responder as questões do **Questionário de Avaliação do MWPVis**. Como o tempo necessário para isso é relativo ao grau de instrução do usuário, nós não especificamos um período de adaptação. Mas como sugestão, para agilizar este processo, é recomendado que o(a) senhor(a) leia o **Manual do Usuário do MWPVis**, que contém todas as instruções de instalação e uso do programa, assim como exemplos de uso e detalhes de cada uma das funcionalidades.

As suas respostas deverão ser anotadas no **Formulário de Respostas do Questionário de Avaliação do MWPVis** e, posteriormente, este deverá ser enviado para o endereço de e-mail mogni@dca.fee.unicamp.br, com o assunto “Minha avaliação do MWPVis”. Como o MWPVis ainda encontra-se em fase de desenvolvimento, é possível que falhas venham a ocorrer durante sua instalação ou execução, por isso, elas também devem ser relatadas.

Cordialmente,

Desenvolvedores do MWPVis.

INSTRUÇÕES DE AVALIAÇÃO

Por favor, siga exatamente cada uma das instruções a seguir:

- 1) Execute o executável de auto-instalação do MWPVis e siga os procedimentos de instalação padrão para instalá-lo em seu computador.
- 2) Após a instalação do programa, familiarize-se previamente com a metodologia de seu ambiente de trabalho antes de responder as questões do **Questionário de Avaliação do MWPVis**.
- 3) Use obrigatoriamente um dos arquivos de exemplos para testar o programa e responder as questões do **Questionário de Avaliação do MWPVis**.
- 4) Preencha o **Formulário de Respostas do Questionário de Avaliação do MWPVis** com as suas respostas e, posteriormente, envie uma cópia deste formulário para o endereço de e-mail mogni@dca.fee.unicamp.br, com o assunto “Minha avaliação do MWPVis”.

1. PERFIL DO AVALIADOR

Nas questões 1.1 a 1.5 serão perguntadas algumas informações pessoais para que conheçamos quem é você.

1.1 Qual é o seu sexo?

- (A) Masculino.
- (B) Feminino.

1.2 Qual é a sua faixa etária?

- (A) Até 29 anos.
- (B) De 30 a 39 anos.
- (C) De 40 a 49 anos.
- (D) De 50 a 59 anos.
- (E) 60 anos ou mais.

1.3 Qual é o seu vínculo com a Unicamp?

- (A) Discente de graduação. *(Por favor, avance para a questão 1.5)*
- (B) Discente de pós-graduação.
- (C) Docente. *(Por favor, avance para a questão 1.5)*
- (D) Pesquisador. *(Por favor, avance para a questão 1.5)*
- (E) Outro: _____ . *(Por favor, avance para a questão 1.5)*

1.4 Se você for discente de pós-graduação, qual é a modalidade do seu curso?

- (A) Especialização.
- (B) Mestrado Profissional.
- (C) Mestrado.
- (D) Doutorado.
- (E) Pós-doutorado.

1.5 Qual é o nome da unidade de ensino que você atua?

- (A) Faculdade de Engenharia Civil, Arquitetura e Urbanismo – FEC.
- (B) Faculdade de Engenharia Elétrica e de Computação – Feec.
- (C) Faculdade de Engenharia Mecânica – FEM.
- (D) Outra: _____.

2. CONHECIMENTOS E NECESSIDADES DO AVALIADOR

Nas questões 2.1 a 2.8 serão perguntadas algumas informações sobre seus conhecimentos e suas necessidades.

2.1 Qual é o seu nível de instrução em informática?

- (A) Básico.
- (B) Básico-intermediário.
- (C) Intermediário.
- (D) Intermediário-avançado.
- (E) Avançado.

2.2 Você atua na área de propagação de ondas mecânicas?

- (A) Sim.
- (B) Não. *(Por favor, avance para a questão 2.4)*

2.3 Se você atua na área de propagação de ondas mecânicas, quanto tempo faz?

- (A) Menos de 2 anos.
- (B) Entre 2 e 4 anos.
- (C) Entre 4 e 10 anos.
- (D) Mais de 10 anos.

- 2.4 Qual das seguintes alternativas melhor descreve sua necessidade de visualizar dados que descrevem o comportamento de ondas mecânicas?
- (A) Não tenho necessidade de visualizar estes dados.
 - (B) Tenho pouca necessidade de visualizar estes dados.
 - (C) Tenho razoável necessidade de visualizar estes dados.
 - (D) Tenho muita necessidade de visualizar estes dados.
 - (E) Tenho extrema necessidade de visualizar estes dados.
- 2.5 Você usa algum programa para visualizar dados que descrevem o comportamento de ondas mecânicas?
- (A) Sim.
 - (B) Não. *(Por favor, avance para a seção 3)*
- 2.6 Se você usa algum programa para visualizar dados que descrevem o comportamento de ondas mecânicas, qual é o nome deste programa?
- (A) MATLAB.
 - (B) ParaView.
 - (C) Wolfram Mathematica.
 - (D) Outro: _____.
- 2.7 Se você usa algum programa para visualizar dados que descrevem o comportamento de ondas mecânicas, assinale a alternativa que melhor descreve o seu nível de satisfação com este programa?
- (A) Não estou satisfeito com este programa.
 - (B) Estou pouco satisfeito com este programa.
 - (C) Estou razoavelmente satisfeito com este programa.
 - (D) Estou muito satisfeito com este programa. *(Por favor, avance para a seção 3)*
 - (E) Estou extremamente satisfeito com este programa. *(Por favor, avance para a seção 3)*
- 2.8 Se você usa algum programa para visualizar dados que descrevem o comportamento de ondas mecânicas e tem um nível de satisfação inferior ou igual a razoável com este programa, relate os motivos.

3. ALGORITMOS DO MWPVIS

Das questões 3.1 a 3.11, você avaliará a eficácia de cada um dos algoritmos disponíveis no MWPVis.

3.1 Do item 3.1.1 ao 3.1.6, assinale a alternativa que melhor descreve o grau de visualização de cada um dos conjuntos de dados escalares representados pelo algoritmo *Colormap*. Interprete a palavra visualização como sendo a capacidade de ver e compreender os dados representados.

	Não consegui visualizar.	Consegui visualizar com pouca clareza.	Consegui visualizar com razoável clareza.	Consegui visualizar com muita clareza.	Consegui visualizar com extrema clareza.
3.1.1	(A)	(B)	(C)	(D)	(E)
3.1.2	(A)	(B)	(C)	(D)	(E)
3.1.3	(A)	(B)	(C)	(D)	(E)
3.1.4	(A)	(B)	(C)	(D)	(E)
3.1.5	(A)	(B)	(C)	(D)	(E)
3.1.6	(A)	(B)	(C)	(D)	(E)

3.2 Do item 3.2.1 ao 3.2.6, assinale a alternativa que melhor descreve o grau de visualização de cada um dos conjuntos de dados escalares representados pelo algoritmo *Contour*. Interprete a palavra visualização como sendo a capacidade de ver e compreender os dados representados.

	Não consegui visualizar.	Consegui visualizar com pouca clareza.	Consegui visualizar com razoável clareza.	Consegui visualizar com muita clareza.	Consegui visualizar com extrema clareza.
3.2.1	(A)	(B)	(C)	(D)	(E)
3.2.2	(A)	(B)	(C)	(D)	(E)
3.2.3	(A)	(B)	(C)	(D)	(E)
3.2.4	(A)	(B)	(C)	(D)	(E)
3.2.5	(A)	(B)	(C)	(D)	(E)
3.2.6	(A)	(B)	(C)	(D)	(E)

- 3.3 Do item 3.3.1 ao 3.3.6, assinale a alternativa que melhor descreve o grau de visualização de cada um dos conjuntos de dados escalares representados pelo algoritmo *Warp By Scalar*. Interprete a palavra visualização como sendo a capacidade de ver e compreender os dados representados.

		Não consegui visualizar.	Consegui visualizar com pouca clareza.	Consegui visualizar com razoável clareza.	Consegui visualizar com muita clareza.	Consegui visualizar com extrema clareza.
3.3.1	X-Amplitude Y-Amplitude Z-Amplitude	(A)	(B)	(C)	(D)	(E)
3.3.2	Distance	(A)	(B)	(C)	(D)	(E)
3.3.3	Maximum X-Amplitude Maximum Y-Amplitude Maximum Z-Amplitude	(A)	(B)	(C)	(D)	(E)
3.3.4	Maximum Distance	(A)	(B)	(C)	(D)	(E)
3.3.5	XY-Amplitude Magnitude YZ-Amplitude Magnitude ZX-Amplitude Magnitude XYZ-Amplitude Magnitude	(A)	(B)	(C)	(D)	(E)
3.3.6	Movement Direction	(A)	(B)	(C)	(D)	(E)

- 3.4 Do item 3.4.1 ao 3.4.5, assinale a alternativa que melhor descreve o grau de visualização de cada um dos conjuntos de dados vetoriais representados pelo algoritmo *Glyph*. Interprete a palavra visualização como sendo a capacidade de ver e compreender os dados representados.

		Não consegui visualizar.	Consegui visualizar com pouca clareza.	Consegui visualizar com razoável clareza.	Consegui visualizar com muita clareza.	Consegui visualizar com extrema clareza.
3.4.1	Normals	(A)	(B)	(C)	(D)	(E)
3.4.2	XY-Amplitude YZ-Amplitude ZX-Amplitude XYZ-Amplitude	(A)	(B)	(C)	(D)	(E)
3.4.3	Maximum X-Amplitude Gradient Maximum Y-Amplitude Gradient Maximum Z-Amplitude Gradient	(A)	(B)	(C)	(D)	(E)
3.4.4	Maximum Distance Gradient	(A)	(B)	(C)	(D)	(E)
3.4.5	Movement Sense	(A)	(B)	(C)	(D)	(E)

3.5 Do item 3.5.1 ao 3.5.4, assinale a alternativa que melhor descreve o grau de visualização de cada um dos conjuntos de dados vetoriais representados pelo algoritmo *Stream Tracer*. Interprete a palavra visualização como sendo a capacidade de ver e compreender os dados representados.

	Não consegui visualizar.	Consegui visualizar com pouca clareza.	Consegui visualizar com razoável clareza.	Consegui visualizar com muita clareza.	Consegui visualizar com extrema clareza.
	1	2	3	4	5
3.5.1					
	(A)	(B)	(C)	(D)	(E)
3.5.2					
	(A)	(B)	(C)	(D)	(E)
3.5.3	(A)	(B)	(C)	(D)	(E)
3.5.4	(A)	(B)	(C)	(D)	(E)

3.6 Assinale a alternativa que melhor descreve o grau de utilidade do algoritmo *Extract Subset*. Interprete a palavra utilidade como sendo a capacidade de aplicação para solução de um problema.

- (A) Não tem utilidade.
- (B) Tem pouca utilidade.
- (C) Tem razoável utilidade.
- (D) Tem muita utilidade.
- (E) Tem extrema utilidade.

3.7 Assinale a alternativa que melhor descreve o grau de utilidade do algoritmo *Outline*. Interprete a palavra utilidade como sendo a capacidade de aplicação para solução de um problema.

- (A) Não tem utilidade.
- (B) Tem pouca utilidade.
- (C) Tem razoável utilidade.
- (D) Tem muita utilidade.
- (E) Tem extrema utilidade.

3.8 Assinale a alternativa que melhor descreve o grau de utilidade do algoritmo *Particle Pathlines*. Interprete a palavra utilidade como sendo a capacidade de aplicação para solução de um problema.

- (A) Não tem utilidade.
- (B) Tem pouca utilidade.
- (C) Tem razoável utilidade.
- (D) Tem muita utilidade.
- (E) Tem extrema utilidade.

3.9 Assinale a alternativa que melhor descreve o grau de utilidade do algoritmo *Reflect*. Interprete a palavra utilidade como sendo a capacidade de aplicação para solução de um problema.

- (A) Não tem utilidade.
- (B) Tem pouca utilidade.
- (C) Tem razoável utilidade.
- (D) Tem muita utilidade.
- (E) Tem extrema utilidade.

4.2 Se o gráfico *Plot Over Cut Plane* tem utilidade, assinale a alternativa que melhor descreve o seu grau de satisfação com o uso deste recurso.

- (A) Não estou satisfeito.
- (B) Estou pouco satisfeito.
- (C) Estou razoavelmente satisfeito.
- (D) Estou muito satisfeito. *(Por favor, avance para a questão 4.4)*
- (E) Estou extremamente satisfeito. *(Por favor, avance para a questão 4.4)*

4.3 Se o seu grau de satisfação com o gráfico *Plot Over Cut Plane* é inferior ou igual a razoável, relate os motivos.

4.4 Assinale a alternativa que melhor descreve o grau de utilidade do gráfico *Plot Selection Over Time*. Interprete a palavra utilidade como sendo a capacidade de aplicação para solução de um problema.

- (A) Não tem utilidade. *(Por favor, avance para a questão 5.1)*
- (B) Tem pouca utilidade.
- (C) Tem razoável utilidade.
- (D) Tem muita utilidade.
- (E) Tem extrema utilidade.

4.5 Se o gráfico *Plot Selection Over Time* tem utilidade, assinale a alternativa que melhor descreve o seu grau de satisfação com o uso deste recurso.

- (A) Não estou satisfeito.
- (B) Estou pouco satisfeito.
- (C) Estou razoavelmente satisfeito.
- (D) Estou muito satisfeito. *(Por favor, avance para a questão 5.1)*
- (E) Estou extremamente satisfeito. *(Por favor, avance para a questão 5.1)*

4.6 Se o seu grau de satisfação com o gráfico *Plot Selection Over Time* é inferior ou igual a razoável, relate os motivos.

5. FUNCIONALIDADES DO MWPVIS

Nas questões 5.1 a 5.5, você avaliará a eficácia das demais funcionalidades disponíveis no MWPVis.

- 5.1 Assinale a alternativa que melhor descreve o grau de utilidade da funcionalidade *Show Color Legend*. Interprete a palavra utilidade como sendo a capacidade de aplicação para solução de um problema.
- (A) Não tem utilidade.
 - (B) Tem pouca utilidade.
 - (C) Tem razoável utilidade.
 - (D) Tem muita utilidade.
 - (E) Tem extrema utilidade.
- 5.2 Assinale a alternativa que melhor descreve o grau de utilidade da funcionalidade *Show Measures*. Interprete a palavra utilidade como sendo a capacidade de aplicação para solução de um problema.
- (A) Não tem utilidade.
 - (B) Tem pouca utilidade.
 - (C) Tem razoável utilidade.
 - (D) Tem muita utilidade.
 - (E) Tem extrema utilidade.
- 5.3 Assinale a alternativa que melhor descreve o grau de utilidade da funcionalidade *Stereo Vision*. Interprete a palavra utilidade como sendo a capacidade de aplicação para solução de um problema.
- (A) Não tem utilidade.
 - (B) Tem pouca utilidade.
 - (C) Tem razoável utilidade.
 - (D) Tem muita utilidade.
 - (E) Tem extrema utilidade.
- 5.4 Assinale a alternativa que melhor descreve o grau de utilidade da funcionalidade *Comparative*. Interprete a palavra utilidade como sendo a capacidade de aplicação para solução de um problema.
- (A) Não tem utilidade.
 - (B) Tem pouca utilidade.
 - (C) Tem razoável utilidade.
 - (D) Tem muita utilidade.
 - (E) Tem extrema utilidade.
- 5.5 Assinale a alternativa que melhor descreve o grau de utilidade da funcionalidade *Distance*. Interprete a palavra utilidade como sendo a capacidade de aplicação para solução de um problema.
- (A) Não tem utilidade.
 - (B) Tem pouca utilidade.
 - (C) Tem razoável utilidade.
 - (D) Tem muita utilidade.
 - (E) Tem extrema utilidade.

6. INTERFACE GRÁFICA DO MWPVIS

Das questões 6.1 a 6.6, você avaliará a interface gráfica do MWPVis, segundo os seguintes critérios de qualidade: agradabilidade, usabilidade, interatividade, intuitividade e produtividade.

- 6.1 Assinale a alternativa que melhor descreve o nível de agradabilidade da interface gráfica do MWPVis. Ou seja, o quão agradável é a estética do programa.
- (A) Não é agradável.
 - (B) É pouco agradável.
 - (C) É razoavelmente agradável.
 - (D) É muito agradável.
 - (E) É extremamente agradável.

- 6.2 Assinale a alternativa que melhor descreve o nível de usabilidade da interface gráfica do MWPVis. Ou seja, o quão fácil de usar é o programa.
- (A) Não é fácil de usar.
 - (B) É pouco fácil de usar.
 - (C) É razoavelmente fácil de usar.
 - (D) É muito fácil de usar.
 - (E) É extremamente fácil de usar.
- 6.3 Assinale a alternativa que melhor descreve o nível de interatividade da interface gráfica do MWPVis. Ou seja, o quanto o programa permite que o usuário realize suas atividades de forma interativa.
- (A) Não é interativo.
 - (B) É pouco interativo.
 - (C) É razoavelmente interativo.
 - (D) É muito interativo.
 - (E) É extremamente interativo.
- 6.4 Assinale a alternativa que melhor descreve o nível de intuitividade da interface gráfica do MWPVis. Ou seja, o quão facilmente dedutível é a realização das funcionalidades oferecidas pelo programa.
- (A) Não é intuitivo.
 - (B) É pouco intuitivo.
 - (C) É razoavelmente intuitivo.
 - (D) É muito intuitivo.
 - (E) É extremamente intuitivo.
- 6.5 Assinale a alternativa que melhor descreve o nível de produtividade da interface gráfica do MWPVis. Ou seja, o quanto a metodologia do ambiente de trabalho do programa favorece nas realizações de suas tarefas.
- (A) Não é produtivo.
 - (B) É pouco produtivo.
 - (C) É razoavelmente produtivo.
 - (D) É muito produtivo.
 - (E) É extremamente produtivo.

6.6 Para cada um dos critérios de qualidade negativamente avaliados, ou seja, com nível inferior ou igual a razoável, relate os respectivos motivos.

7. ACEITAÇÃO DO MWPVIS

- 7.1 De 0 a 100, no geral, qual nota você dá para o MWPVis?
- (A) Entre 0 a 20.
 - (B) Entre 20 a 40.
 - (C) Entre 40 a 60.
 - (D) Entre 60 a 80.
 - (E) Entre 80 a 100.
- 7.2 Você usaria o MWPVis para visualizar seus dados?
- (A) Sim. *(Por favor, avance para a questão 7.4)*
 - (B) Não.

7.3 Se você não usaria este programa, diga os motivos.

7.4 Se você já usa algum programa para visualizar seus dados, você deixaria de usá-lo para usar o MWPVis?

- (A) Sim. *(Por favor, avance para a questão 7.6)*
- (B) Não.

7.5 Se você não deixaria de usar o seu programa para visualizar seus dados, relate os motivos.

7.6 Se você tem alguma crítica, sugestão ou comentário sobre o MWPVis. Por favor, relate-as.

ANEXO B – Manual do Usuário do MWPVis

Manual do Usuário do MWPVis

1. INTRODUÇÃO

O MWPVis – acrônimo do inglês *Mechanical Wave Propagation Visualization* – é um programa de computador destinado à visualização de propagação de ondas mecânicas. Foi desenvolvido no Departamento de Engenharia de Computação e Automação Industrial da Faculdade de Engenharia Elétrica e de Computação, pelo estudante Rodrigo Mogni Gonçalves dos Santos, do curso de Mestrado em Engenharia Elétrica, e seu orientador Prof. Dr. José Mario De Martino. Surgiu como uma tentativa de sanar uma carência do Departamento de Mecânica Computacional da Faculdade de Engenharia Mecânica: a necessidade da visualização e exploração interativa dos dados gerados pelo Wanglay (um simulador de propagação de ondas mecânicas).

1.1 LICENÇA

O MWPVis é um programa de computador proprietário e não comercial. Todos os direitos são reservados aos seus desenvolvedores. Por ser de uso restrito, nenhuma cópia, modificação ou redistribuição pode ser realizada sem consentimento de seus desenvolvedores.

1.2 REQUISITOS

Como nenhuma análise específica para obtenção dos requisitos mínimos necessários para instalação e execução do MWPVis foi realizada, será relatada então a configuração do computador usado para o desenvolvimento do programa: processador Intel Core 2 Duo E8400 de 3 GHz, sistema operacional Windows Vista de 32 bits, memória RAM de 4 GB, disco rígido de 250 GB, placa de vídeo de 256 MB, unidade de DVD-ROM e monitor colorido com resolução de 1024x768 pixels. Entretanto é de conhecimento que o programa necessita de no mínimo 84 MB de espaço disponível em disco rígido para instalação.

1.3 INSTALAÇÃO

Há duas maneiras de instalar o MWPVis: por meio do executável de auto-instalação, que contém apenas os arquivos binários do programa, tal como um executável e suas bibliotecas dependentes; ou pelo código-fonte do programa, que contém além do projeto com os códigos-fontes, os *scripts* de compilação e instalação.

Pela primeira maneira de instalação, via executável de auto-instalação, basta executar o arquivo *mwpvis-0.4.14-win32.exe* e seguir os procedimentos de instalação padrão do programa. Esse será instalado no diretório *C:\Program Files\MWPVis\0.4.14*. Um atalho para o programa será criado no menu Iniciar. Já pela segunda maneira de instalação, via código-fonte do programa, é necessário previamente ter instalado o interpretador do Python e os pacotes PyQt, VTK e NumPy, dos quais o MWPVis é dependente.

Para gerar os arquivos binários do MWPVis é usado o Py2exe, um pacote que converte *scripts* escritos em Python para programas executáveis do Windows. Para instalá-lo, basta executar o arquivo *py2exe-0.6.9.win32-py2.6.exe* e seguir os procedimentos de instalação padrão do programa. Esse será instalado no diretório *C:\Python26\Lib\site-packages\py2exe*. Uma versão mais recente do Py2exe pode ser encontrada em seu site na Web, no endereço <http://www.py2exe.org>.

Tendo por instalados o interpretador do Python e os pacotes PyQt, VTK, NumPy e Py2exe, já é possível converter os códigos-fontes do projeto MWPVis, escritos em Python, para um arquivo executável. Para isto, faça:

- 1) Crie uma pasta chamada *MWPVis*, preferencialmente, em *C:*.
- 2) Descompacte todo o conteúdo existente no arquivo compactado *mwpvis-0.4.14-win32-src.zip* no diretório *C:\MWPVis*.
- 3) Execute o arquivo *bdist.bat* existente no diretório *C:\MWPVis\src* para iniciar a conversão dos *scripts* contidos na pasta *mwpvis* para um arquivo executável. Este chama-se *mwpvis.exe* e será criado no diretório *C:\MWPVis\bin*, juntamente com suas bibliotecas dependentes.

Para gerar o executável de auto-instalação do MWPVis, deve-se executar o arquivo *setup.bat* existente no diretório *C:\MWPVis\setup*. Esse é gerado com uso do compilador do NSIS (*Nullsoft Scriptable Install System* – sistema de instalação por *scripts* da Nullsoft). Para instalá-lo, basta executar o arquivo *nsis-2.46-setup.exe* e seguir os

procedimentos de instalação padrão do programa. Esse será instalado no diretório *C:\Program Files\NSIS*. Uma versão mais recente do NSIS pode ser encontrada em seu site na Web, no endereço <http://nsis.sourceforge.net>.

Como o endereço do diretório de instalação do NSIS, onde encontra-se o seu compilador, não é automaticamente incluído na variável de ambiente *Path* pelo instalador do programa, será necessário incluí-lo manualmente. No Apêndice A, constam informações sobre como incluir um ou mais endereços na variável de ambiente *Path* do Windows Vista.

Todos os instaladores dos programas referenciados neste tópico estão contidos no DVD que acompanha esta dissertação. Dentre eles, a versão 2.46 do NSIS, a versão 0.4.14 do MWPVis, a versão 1.4.1 do NumPy, a versão 0.6.9 do Py2exe, a versão 4.7.3 do PyQt, a versão 2.6.5 do Python e a versão 5.4.2 do VTK. É aconselhável que estes programas sejam instalados nos endereços e nas configurações recomendadas pelos seus respectivos instaladores durante o processo de instalação.

2. INTERFACE GRÁFICA

A interação entre o usuário e o programa é provida por meio de uma interface gráfica agradável e intuitiva, composta pelos seguintes componentes: barra de menus, barras de ferramentas, painéis de navegação, barra de estado e painel de visualização. A janela principal do programa é apresentada na Figura 1.

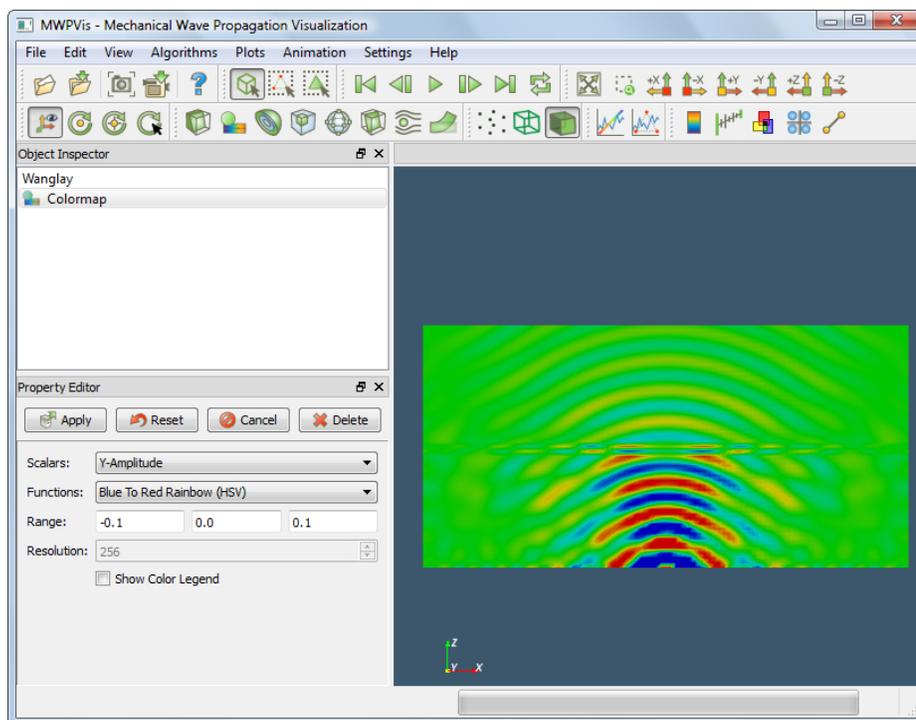


Figura 1 – Janela principal do MWPVis.

A barra de menus, localizada no topo da janela principal do programa, contém os seguintes menus: *File*, *Edit*, *View*, *Algorithms*, *Plots*, *Animation*, *Settings* e *Help*. Os comandos existentes em cada um destes menus são apresentados na Tabela 1.

Tabela 1 – Itens de menu da barra de menus do MWPVis.

<i>File</i>	Agrupa os seguintes comandos: <i>Open...</i> , <i>Recent Files</i> , <i>Save</i> , <i>Save As...</i> , <i>Save As Screenshot</i> , <i>Save As Animation</i> , <i>Import...</i> , <i>Export...</i> e <i>Exit</i> .
<i>Edit</i>	Agrupa os seguintes comandos: <i>Interact</i> , <i>Select Surface Points</i> e <i>Select Surface Cells</i> .
<i>View</i>	Agrupa os seguintes comandos: <i>Reset Camera</i> , <i>Zoom To Selection</i> , <i>+X</i> , <i>-X</i> , <i>+Y</i> , <i>-Y</i> , <i>+Z</i> , <i>-Z</i> , <i>Show Orientation Axes</i> , <i>Show Center Axes</i> , <i>Reset Center Axes</i> , <i>Pick Center Axes</i> , <i>Points</i> , <i>Wireframe</i> , <i>Surface</i> , <i>Show</i>

	<i>Color Legend, Show Measures, Stereo Vision, Comparative e Distance.</i>
<i>Algorithms</i>	Agrupa os seguintes comandos: <i>Clip, Colormap, Contour, Extract Subset, Glyph, Slice, Stream Tracer, Warp By Scalar, Outline, Particle Pathlines e Reflect.</i>
<i>Plots</i>	Agrupa os seguintes comandos: <i>Plot Over Cut Plane e Plot Selection Over Time.</i>
<i>Animation</i>	Agrupa os seguintes comandos: <i>First, Back, Play, Next, Last e Loop.</i>
<i>Settings</i>	Agrupa os seguintes comandos: .
<i>Help</i>	Agrupa os seguintes comandos: <i>Help, About MWPVis, About PyQt, About Python e About VTK.</i>

As barras de ferramentas estão localizadas logo abaixo da barra de menus. Cada uma delas agrupa comandos com funcionalidades similares. Tem por objetivo reduzir o número de ações necessárias para execução de comandos mais comumente usados. A Tabela 2 apresenta cada uma das barras de ferramentas e seus respectivos agrupamentos de comandos. No submenu *Toolbars*, localizado no menu *View*, é possível controlar a visibilidade de cada uma dessas barras de ferramentas.

Tabela 2 – Barras de ferramentas do MWPVis.

<i>Main</i>	Agrupa os seguintes comandos: <i>Open..., Save, Save As Screenshot, Save As Animation e Help.</i>
<i>Selection</i>	Agrupa os seguintes comandos: <i>Interact, Select Surface Points e Select Surface Cells.</i>
<i>Animation</i>	Agrupa os seguintes comandos: <i>First, Back, Play, Next, Last e Loop.</i>
<i>Camera</i>	Agrupa os seguintes comandos: <i>Reset Camera, Zoom To Selection, +X, -X, +Y, -Y, +Z e -Z.</i>
<i>Axes</i>	Agrupa os seguintes comandos: <i>Show Orientation Axes, Show Center Axes, Reset Center Axes e Pick Center Axes.</i>
<i>Algorithms</i>	Agrupa os seguintes comandos: <i>Clip, Colormap, Contour, Extract Subset, Glyph, Slice, Stream Tracer, Warp By Scalar, Outline, Particle Pathlines e Reflect.</i>
<i>Representations</i>	Agrupa os seguintes comandos: <i>Points, Wireframe e Surface.</i>
<i>Plots</i>	Agrupa os seguintes comandos: <i>Plot Over Cut Plane e Plot Selection Over Time.</i>
<i>Others</i>	Agrupa os seguintes comandos: <i>Show Color Legend, Show Measures, Stereo Vision, Comparative e Distance.</i>

A Tabela 3 descreve cada um dos painéis de navegação do programa. No submenu *Navigation Panels*, localizado no menu *View*, é possível controlar a visibilidade de cada um desses painéis.

Tabela 3 – Descrição dos painéis de navegação do MWPVis.

<i>Object Inspector</i>	Contém uma lista com os algoritmos e gráficos usados pelo usuário.
<i>Property Editor</i>	Permite alterar as propriedades dos algoritmos e gráficos usados pelo usuário.
<i>XY Plot</i>	Área de exibição dos gráficos usados pelo usuário.
<i>Statistics View</i>	Tabela com as descrições de cada um dos algoritmos e gráficos usados pelo usuário.

A barra de estado encontra-se na parte inferior da janela principal do programa. Nela é apresentada informações sobre o progresso dos comandos executados pelo usuário. O painel de navegação, localizado no centro da janela principal, é a área destinada para renderização da visualização. Cada um dos comandos do programa são brevemente descritos na Tabela 4.

Tabela 4 – Descrição dos comandos do MWPVis.

	<i>About MWPVis</i>	Exibe um diálogo com uma breve descrição do MWPVis.
	<i>About PyQt</i>	Exibe um diálogo com uma breve descrição do PyQt.
	<i>About Python</i>	Exibe um diálogo com uma breve descrição do Python.

	<i>About VTK</i>	Exibe um diálogo com uma breve descrição do VTK.
	<i>Back</i>	Força a renderização do quadro anterior da animação.
	<i>Clip</i>	Corresponde ao algoritmo de recorte.
	<i>Colormap</i>	Corresponde ao algoritmo de mapeamento em cores.
	<i>Comparative</i>	Subdivide a área de visualização em quatro partes iguais. No qual cada uma é representada por um plano de coordenadas.
	<i>Contour</i>	Corresponde ao algoritmo de contorno sobre escalares.
	<i>Distance</i>	Permite que o usuário obtenha o valor da distância entre dois pontos.
	<i>Exit</i>	Encerra a execução do programa. Caso o usuário tenha feito alterações no arquivo de leitura, uma mensagem será exibida para alertá-lo de que as modificações podem ser perdidas se não forem previamente salvas.
	<i>Export...</i>	Exibe um diálogo que permite ao usuário exportar para um conjunto de arquivos de extensão VTK. Cada arquivo corresponde a um quadro da animação.
	<i>Extract Subset</i>	Corresponde ao algoritmo de redução da grade estruturada.
	<i>First</i>	Força a renderização do primeiro quadro da animação.
	<i>Glyph</i>	Corresponde ao algoritmo de glifos de vetores.
	<i>Help</i>	Exibe este Manual do Usuário do MWPVis.
	<i>Import...</i>	Exibe um diálogo que permite ao usuário importar arquivos de extensões TXT ou CSV.
	<i>Interact</i>	Permite que o usuário interaja por intermédio do mouse com a visualização. Sendo o botão esquerdo usado para rotacionar, o do meio para mover e o direito para ampliar ou reduzir.
	<i>Last</i>	Força a renderização do último quadro da animação.
	<i>Loop</i>	Ativa ou desativa a repetição automática da animação.
	<i>Next</i>	Força a renderização do próximo quadro da animação.
	<i>Open...</i>	Exibe um diálogo que permite ao usuário selecionar um arquivo de extensão MWPV para leitura.
	<i>Outline</i>	Corresponde ao algoritmo de contorno da grade estruturada.
	<i>Particle Pathlines</i>	Corresponde ao algoritmo de trajetória dos vértices da grade estruturada.
	<i>Pick Center Axes</i>	Permite que o usuário desloque as três linhas que indicam o ponto de foco da câmera.
	<i>Play</i>	Inicia ou interrompe a animação.
	<i>Plot Over Cut Plane</i>	Gera um gráfico de linhas da variação de um ou mais escalares de um conjunto de pontos em um determinado instante de tempo, ou seja, o quadro atual. O conjunto de pontos é obtido pela intersecção da grade estruturada com um plano.
	<i>Plot Selection Over Time</i>	Gera um gráfico de linhas da variação de um ou mais escalares de um ponto ao longo do tempo, ou seja, da sequência de quadros. O ponto é selecionado com uso do comando <i>Select Surface Points</i> .
	<i>Points</i>	Os vértices da grade estruturada são representados por pontos.
	<i>Reflect</i>	Corresponde ao algoritmo de reflexão da grade estruturada.
	<i>Reset Camera</i>	Centraliza automaticamente a grade estruturada.
	<i>Reset Center Axes</i>	Reposiciona as três linhas que indicam o ponto de foco da câmera para o centro da visualização.
	<i>Save</i>	Permite que o usuário salve as alterações realizadas no arquivo de leitura.

	<i>Save As Animation</i>	Exibe um diálogo que permite ao usuário salvar, em um arquivo de vídeo de extensão AVI, a animação da visualização.
	<i>Save As Screenshot</i>	Exibe um diálogo que permite ao usuário salvar, em um arquivo de imagem de extensão PNG, a renderização atual da visualização.
	<i>Save As...</i>	Exibe um diálogo que permite ao usuário salvar, em um novo arquivo de extensão MWPV, as alterações realizadas no arquivo de leitura.
	<i>Select Surface Cells</i>	Permite que o usuário selecione uma ou mais células vizinhas da grade estruturada.
	<i>Select Surface Points</i>	Permite que o usuário selecione um ou mais vértices vizinhos da grade estruturada.
	<i>Show Center Axes</i>	Exibe no centro da visualização, três linhas que indicam o ponto de foco da câmera.
	<i>Show Color Legend</i>	Exibe no lado direito da área de visualização, uma legenda da tabela de cores.
	<i>Show Measures</i>	Exibe sobre a grade estruturada as suas medidas.
	<i>Show Orientation Axes</i>	Exibe no canto inferior esquerdo da área de visualização, três vetores que representam os eixos das coordenadas X, Y e Z.
	<i>Slice</i>	Corresponde ao algoritmo de fatiamento.
	<i>Stereo Vision</i>	Ativa ou desativa a visão estereoscópica por anaglifo.
	<i>Stream Tracer</i>	Corresponde ao algoritmo de fluxo de linhas.
	<i>Surface</i>	A grade estruturada é representada por superfícies.
	<i>Warp By Scalar</i>	Corresponde ao algoritmo de deformação sobre escalares.
	<i>Wireframe</i>	A grade estruturada é representada por linhas que interligam vértices vizinhos.
	<i>Zoom To Selection</i>	Permite que o usuário aproxime a câmera.
	+X	Posiciona a câmera sobre a parte positiva do eixo de coordenadas X.
	+Y	Posiciona a câmera sobre a parte positiva do eixo de coordenadas Y.
	+Z	Posiciona a câmera sobre a parte positiva do eixo de coordenadas Z.
	-X	Posiciona a câmera sobre a parte negativa do eixo de coordenadas X.
	-Y	Posiciona a câmera sobre a parte negativa do eixo de coordenadas Y.
	-Z	Posiciona a câmera sobre a parte negativa do eixo de coordenadas Z.

3. ARQUIVOS

O MWPVis suporta o carregamento de dados armazenados em dois tipos de arquivos de texto: delimitado por vírgula ou campos de largura fixa. Os formatos CSV e TXT são os padrões respectivamente usados para diferenciar cada um destes tipos de arquivos de texto.

A Tabela 5 contém um pequeno trecho de um conjunto de dados armazenado em um arquivo de texto com campos de largura fixa (formato TXT). Para armazenar este mesmo conjunto de dados em um arquivo de texto delimitado por vírgula (formato CSV), basta substituir cada bloco de espaços por uma vírgula.

Tabela 5 – Exemplo de arquivo de texto com campos de largura fixa.

1	x	z	RUx	IUx	RUz	IUz	RWz	IWz	
2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.05857	-0.10611
3	0.20000	0.00000	-0.00658	0.00159	0.00005	-0.00001	0.05849	-0.10552	
4	0.40000	0.00000	-0.01298	0.00313	0.00020	-0.00004	0.05781	-0.10410	
5	0.60000	0.00000	-0.01898	0.00452	0.00042	-0.00011	0.05666	-0.10176	
6	0.80000	0.00000	-0.02425	0.00552	0.00066	-0.00023	0.05498	-0.09850	

7	1.00000	0.00000	-0.02778	0.00529	0.00078	-0.00013	0.05287	-0.09409
8	1.20000	0.00000	-0.02905	0.00695	0.01296	-0.00176	0.04230	-0.09342
9	1.40000	0.00000	-0.02735	0.00894	0.02127	-0.00509	0.03067	-0.09108
10	1.60000	0.00000	-0.02414	0.01050	0.02606	-0.00904	0.01856	-0.08702
.
.
.
2602	10.00000	10.00000	-0.00325	0.00158	0.00164	0.00240	0.00421	-0.01082

Independente do tipo de arquivo de texto, sempre na primeira linha deve constar o cabeçalho e nas demais as coordenadas e os vetores complexos de deslocamento de cada uma das amostras. É importante que os valores das coordenadas variem, respectivamente, na ordem X, Y e Z.

3.1 IMPORT...

O arquivo de texto apresentado na Tabela 5 foi gerado por intermédio do Wanglay (sendo que o cabeçalho foi posteriormente inserido de maneira manual). Para carregar o conjunto de dados armazenados neste arquivo, por exemplo, é necessário executar o comando *Import...* (localizado no menu *File*). Para visualizar estes dados, é necessário convertê-los para uma grade estruturada. A Tabela 6 apresenta os métodos disponíveis no programa para conversão dos dados.

Tabela 6 – Descrição dos campos do formulário do Wanglay.

<i>File Name</i>	Exibe a localização do arquivo de entrada escolhido pelo usuário.
<i>Horizontal</i>	Gera sobreposição de ondas mecânicas no sentido horizontal.
<i>Vertical</i>	Gera sobreposição de ondas mecânicas no sentido vertical.
<i>Number of Time Steps</i>	Define o número de quadros da animação do efeito de propagação das ondas mecânicas. O grau de precisão dos conjuntos de dados escalares e vetoriais gerados é proporcional ao número de quadros.
<i>Movable Points</i>	Define se os vértices da grade estruturada, que representam os pontos de amostragem, serão móveis ou estáticos.
<i>X-Amplitude</i>	Gera um conjunto de dados escalares que representa a amplitude sobre o eixo X de cada um dos pontos de amostragem. As intensidades deste conjunto de dados escalares variam entre a menor e a maior amplitude sobre o eixo X local.
<i>Y-Amplitude</i>	Gera um conjunto de dados escalares que representa a amplitude sobre o eixo Y de cada um dos pontos de amostragem. As intensidades deste conjunto de dados escalares variam entre a menor e a maior amplitude sobre o eixo Y local.
<i>Z-Amplitude</i>	Gera um conjunto de dados escalares que representa a amplitude sobre o eixo Z de cada um dos pontos de amostragem. As intensidades deste conjunto de dados escalares variam entre a menor e a maior amplitude sobre o eixo Z local.
<i>Distance</i>	Gera um conjunto de dados escalares que representa a distância de cada um dos pontos de amostragem. As intensidades deste conjunto de dados escalares variam entre a menor e a maior distância local.
<i>Maximum X-Amplitude</i>	Gera um conjunto de dados escalares que representa a amplitude máxima sobre o eixo X de cada um dos pontos de amostragem. As intensidades deste conjunto de dados escalares variam entre a menor e a maior amplitude sobre o eixo X global.
<i>Maximum Y-Amplitude</i>	Gera um conjunto de dados escalares que representa a amplitude máxima sobre o eixo Y de cada um dos pontos de amostragem. As intensidades deste conjunto de dados escalares variam entre a menor e a maior amplitude sobre o eixo Y global.
<i>Maximum Z-Amplitude</i>	Gera um conjunto de dados escalares que representa a amplitude máxima sobre o eixo Z de cada um dos pontos de amostragem. As intensidades deste conjunto de dados escalares variam entre a menor e a maior amplitude sobre o eixo Z global.
<i>Maximum Distance</i>	Gera um conjunto de dados escalares que representa a distância

	máxima de cada um dos pontos de amostragem. As intensidades deste conjunto de dados escalares variam entre a menor e a maior distância global.
<i>XY-Amplitude Magnitude</i>	Gera um conjunto de dados escalares que representa a magnitude das amplitudes sobre os eixos X e Y de cada um dos pontos de amostragem. As intensidades deste conjunto de dados escalares variam entre a menor e a maior magnitude das amplitudes sobre os eixos X e Y local.
<i>YZ-Amplitude Magnitude</i>	Gera um conjunto de dados escalares que representa a magnitude das amplitudes sobre os eixos Y e Z de cada um dos pontos de amostragem. As intensidades deste conjunto de dados escalares variam entre a menor e a maior magnitude das amplitudes sobre os eixos Y e Z local.
<i>ZX-Amplitude Magnitude</i>	Gera um conjunto de dados escalares que representa a magnitude das amplitudes sobre os eixos Z e X de cada um dos pontos de amostragem. As intensidades deste conjunto de dados escalares variam entre a menor e a maior magnitude das amplitudes sobre os eixos Z e X local.
<i>XYZ-Amplitude Magnitude</i>	Gera um conjunto de dados escalares que representa a magnitude das amplitudes sobre os eixos X, Y e Z de cada um dos pontos de amostragem. As intensidades deste conjunto de dados escalares variam entre a menor e a maior magnitude das amplitudes sobre os eixos X, Y e Z local.
<i>Movement Direction</i>	Gera um conjunto de dados escalares que representa a direção do movimento de oscilação de cada um dos pontos de amostragem. As intensidades deste conjunto de dados escalares variam entre 0 e 180.
<i>XY-Amplitude</i>	Gera um conjunto de dados vetoriais que representa as amplitudes sobre os eixos X e Y de cada um dos pontos de amostragem.
<i>YZ-Amplitude</i>	Gera um conjunto de dados vetoriais que representa as amplitudes sobre os eixos Y e Z de cada um dos pontos de amostragem.
<i>ZX-Amplitude</i>	Gera um conjunto de dados vetoriais que representa as amplitudes sobre os eixos Z e X de cada um dos pontos de amostragem.
<i>XYZ-Amplitude</i>	Gera um conjunto de dados vetoriais que representa as amplitudes sobre os eixos X, Y e Z de cada um dos pontos de amostragem.
<i>Maximum X-Amplitude Gradient</i>	Gera um conjunto de dados vetoriais que representa o gradiente da amplitude máxima sobre o eixo X de cada um dos pontos de amostragem.
<i>Maximum Y-Amplitude Gradient</i>	Gera um conjunto de dados vetoriais que representa o gradiente da amplitude máxima sobre o eixo Y de cada um dos pontos de amostragem.
<i>Maximum Z-Amplitude Gradient</i>	Gera um conjunto de dados vetoriais que representa o gradiente da amplitude máxima sobre o eixo Z de cada um dos pontos de amostragem.
<i>Maximum Distance Gradient</i>	Gera um conjunto de dados vetoriais que representa o gradiente da distância máxima de cada um dos pontos de amostragem.
<i>Movement Sense</i>	Gera um conjunto de dados vetoriais que representa o sentido de deslocamento de cada um dos pontos de amostragem.

3.2 EXPORT...

O comando *Export...* (localizado no menu *File*) permite salvar a grade estruturada em arquivos VTK sequenciados, que posteriormente podem ser lidos, por exemplo, pelo ParaView.

4. ALGORITMOS

O menu *Algorithms* da barra de ferramentas reúne uma coleção de algoritmos – técnicas de visualização de dados – que podem ser aplicados para visualização dos conjuntos de dados escalares ou vetoriais pré-calculados durante a

geração da grade estruturada, que representa o meio de propagação das ondas mecânicas. Nas subseções a seguir é brevemente descrito cada um desses algoritmos.

4.1 CLIP

O algoritmo *Clip* realiza uma operação de recorte entre duas figuras geométricas: a grade estruturada e uma outra qualquer que possa ser descrita por uma função implícita. Por padrão, o algoritmo dispõe de três opções de funções implícitas: plano, paralelepípedo e esfera; além da opção de recorte por escalar. A Tabela 7 apresenta uma breve descrição de cada um dos campos do formulário desse algoritmo.

Tabela 7 – Campos do formulário do algoritmo Clip.

<i>Function</i>	Funções implícitas: plano, paralelepípedo e esfera. Além da opção de recorte por escalar.
<i>Inside Out</i>	Parte que será eliminada com o recorte.
<i>Show Widget</i>	Exibe a figura geométrica na área de visualização.
<i>Scalars</i>	Conjuntos de dados escalares.
<i>Value</i>	Intensidade do escalar de recorte.
<i>Origin</i>	Coordenadas da origem do plano de corte.
<i>Normal</i>	Vetor normal do plano de corte.
<i>Translate</i>	Coordenadas de translação do paralelepípedo.
<i>Rotate</i>	Coordenadas de rotação do paralelepípedo.
<i>Scale</i>	Coordenadas de escala do paralelepípedo.
<i>Center</i>	Coordenadas do centro da esfera.
<i>Radius</i>	Valor do raio da esfera.

4.2 COLORMAP

O algoritmo *Colormap* é capaz de mapear um conjunto de dados escalares em cores. As cores são geradas a partir das funções de transferências descritas na Tabela 8.

Tabela 8 – Descrição das funções de transferências do algoritmo Colormap.

<i>Cool To Warm (Diverging)</i>	O conjunto de dados escalares é mapeado por uma função de transferência que varia entre as cores azul e vermelha no modelo de cores Divergente.
<i>Blue To Red Rainbow (HSV)</i>	O conjunto de dados escalares é mapeado por uma função de transferência que varia entre as cores azul e vermelha no modelo de cores HSV.
<i>Red To Blue Rainbow (HSV)</i>	O conjunto de dados escalares é mapeado por uma função de transferência que varia entre as cores vermelha e azul no modelo de cores HSV.
<i>Grayscale (RGB)</i>	O conjunto de dados escalares é mapeado por uma função de transferência que varia entre as cores preta e branca no modelo de cores RGB.
<i>Blue To Yellow (RGB)</i>	O conjunto de dados escalares é mapeado por uma função de transferência que varia entre as cores preta e branca no modelo de cores RGB.
<i>Black-Body Radiation (RGB)</i>	O conjunto de dados escalares é mapeado por uma função de transferência que varia entre as cores preta, vermelha, amarela e branca no modelo de cores RGB.
<i>Blue To Red (CIELAB)</i>	O conjunto de dados escalares é mapeado por uma função de transferência que varia entre as cores azul e vermelha no modelo de cores CIELAB.

A Tabela 9 apresenta uma breve descrição de cada um dos campos do formulário do algoritmo *Colormap*.

Tabela 9 – Campos do formulário do algoritmo Colormap.

<i>Scalars</i>	Conjuntos de dados escalares.
----------------	-------------------------------

<i>Functions</i>	Sete diferentes tipos de funções de transferências para mapeamento de um determinado conjunto de dados escalares em cores.
<i>Range</i>	Intervalo de valores de um conjunto de dados.
<i>Resolution</i>	Número de cores da tabela de cores usada para o mapeamento de um determinado conjunto de dados escalares.

4.3 CONTOUR

O algoritmo *Contour* pode ser usado para gerar isolinhas ou isosuperfícies a partir de uma ou mais intensidades de um conjunto de dados escalares. A Tabela 10 apresenta uma breve descrição de cada um dos campos do formulário desse algoritmo.

Tabela 10 – Campos do formulário do algoritmo Contour.

<i>Scalars</i>	Conjuntos de dados escalares.
<i>Steps</i>	Intensidades dos escalares.
<i>Color</i>	Cor da linha de contorno.
<i>Line Width</i>	Largura da linha de contorno.

4.4 EXTRACT SUBSET

O algoritmo *Extract Subset* reduz a extensão da grade estruturada. A Tabela 11 apresenta uma breve descrição de cada um dos campos do formulário desse algoritmo.

Tabela 11 – Campos do formulário do algoritmo Extract Subset.

<i>X Range</i>	Extensão da grade estruturada ao longo do eixo X.
<i>Y Range</i>	Extensão da grade estruturada ao longo do eixo Y.
<i>Z Range</i>	Extensão da grade estruturada ao longo do eixo Z.

4.5 GLYPH

O algoritmo *Glyph* é usado para representar um conjunto de dados vetoriais. Para isto são utilizados glifos – símbolos geométricos – que podem variar entre setas bidimensionais ou tridimensionais, linhas, cones, cubos, cilindros e esferas. A Tabela 12 apresenta uma breve descrição de cada um dos campos do formulário desse algoritmo.

Tabela 12 – Campos do formulário do algoritmo Glyph.

<i>Vectors</i>	Conjuntos de dados vetoriais.
<i>Glyph Type</i>	Tipo de representação de glifos: setas bidimensionais ou tridimensionais, linhas, cones, cubos, cilindros e esferas.
<i>Glyph Color</i>	Cor do glifo.
<i>Scale Factor</i>	Intensidade do fator de escala.
<i>Normalize</i>	Normaliza o conjunto de dados vetoriais.

4.6 SLICE

O algoritmo *Slice* realiza uma operação de intersecção entre duas figuras geométricas: a grade estruturada e uma outra qualquer que possa ser descrita por uma função implícita. Por padrão, o algoritmo dispõe de três opções de funções implícitas: plano, paralelepípedo e esfera. A Tabela 13 apresenta uma breve descrição de cada um dos campos do formulário desse algoritmo.

Tabela 13 – Campos do formulário do algoritmo Slice.

<i>Function</i>	Funções implícitas: plano, paralelepípedo e esfera.
<i>Show Widget</i>	Exibe a figura geométrica na área de visualização.
<i>Origin</i>	Coordenadas da origem do plano de corte.
<i>Normal</i>	Vetor normal do plano de corte.
<i>Translate</i>	Coordenadas de translação do paralelepípedo.
<i>Rotate</i>	Coordenadas de rotação do paralelepípedo.
<i>Scale</i>	Coordenadas de escala do paralelepípedo.

<i>Center</i>	Coordenadas do centro da esfera.
<i>Radius</i>	Valor do raio da esfera.

4.7 STREAM TRACER

O algoritmo *Stream Tracer* gera linhas de fluxo de um conjunto de dados vetoriais. Para usá-lo é necessário que os vértices da grade estruturada sejam pontos estáticos. A Tabela 14 apresenta uma breve descrição de cada um dos campos do formulário desse algoritmo.

Tabela 14 – Campos do formulário do algoritmo Stream Tracer.

<i>Vectors</i>	Conjuntos de dados vetoriais.
<i>Initial Integration Step</i>	Distância de cada parada.
<i>Maximum Propagation</i>	Número de interações.
<i>Color</i>	Cor das linhas de fluxo.
<i>Line Width</i>	Largura das linhas de fluxo.

4.8 WARP BY SCALAR

O algoritmo *Warp By Scalar* gera uma deformação na grade estruturada em função da intensidade de um conjunto de dados escalares.

Tabela 15 – Campos do formulário do algoritmo Warp By Scalar.

<i>Scalars</i>	Lista com conjuntos de escalares pré-calculados.
<i>Scale Factor</i>	Fator multiplicador da escala. O valor deve ser maior ou igual a zero.
<i>Normal</i>	Coordenadas do vetor normal.

4.9 OUTLINE

O algoritmo *Outline* gera uma caixa delimitadora sob a grade estruturada. Esta caixa delimitadora pode ser representada de três maneiras diferentes: apenas pelos seus vértices, pelas suas arestas ou pelo contorno das extremidades da grade estruturada. A Tabela 16 apresenta uma breve descrição de cada um dos campos do formulário desse algoritmo.

Tabela 16 – Campos do formulário do algoritmo Outline.

<i>Type</i>	Tipo de caixa delimitadora.
<i>Color</i>	Cor da linhas de contorno da caixa demilitadora.
<i>Line Width</i>	Largura da linha de contorno da caixa delimitadora.

4.10 PARTICLE PATHLINES

O algoritmo *Particle Pathlines* descreve a trajetória de deslocamento de cada um dos vértices da grade estruturada ao longo da sequência de quadros da animação. Para usá-lo é necessário que estes vértices sejam pré-definidos como sendo móveis e que o número de quadros seja superior a um. Como o grau de precisão da trajetória de deslocamento é proporcional ao número de quadros da animação, quanto maior o número de quadros, maior será o grau de precisão. A Tabela 17 apresenta cada um dos campos do formulário desse algoritmo.

Tabela 17 – Campos do formulário do algoritmo Particle Pathlines.

<i>IDs</i>	Identificação de cada um dos vértices selecionados na grade estruturada.
<i>Color</i>	Cor das linhas que representam a trajetória de deslocamento.
<i>Line Width</i>	Largura das linhas que representam a trajetória de deslocamento.

4.11 REFLECT

O algoritmo *Reflect* gera uma cópia da grade estruturada por reflexão. A Tabela 18 apresenta cada um dos campos do formulário desse algoritmo.

Tabela 18 – Campos do formulário do algoritmo Reflect.

<i>Plane</i>	Dispõe de nove diferentes tipos de planos de reflexão.
<i>Center</i>	Centro do plano de reflexão.
<i>Copy Input</i>	Cria uma cópia da grade estruturada original.

5. GRÁFICOS

No menu *Plots* da barra de ferramentas há dois métodos para geração de gráficos. Estes podem ser usados para análise da variação de um ou mais conjuntos de dados escalares, pré-calculados durante a geração da grade estruturada. Cada um desses métodos é brevemente descrito a seguir.

5.1 PLOT OVER CUT PLANE

O método *Plot Over Cut Plane* permite gerar um gráfico de linhas que relaciona as intensidades de um ou mais conjuntos de dados escalares com um agrupamento de vértices da grade estruturada em um determinado quadro. Este agrupamento de vértices é obtido pela intersecção da grade estruturada com um plano. Para visualizar o gráfico gerado é necessário habilitar o painel de navegação *XY Plot* que por padrão vem oculto.

5.2 PLOT SELECTION OVER TIME

O método *Plot Selection Over Time* permite gerar um gráfico de linhas que relaciona as intensidades de um ou mais conjuntos de dados escalares de um único vértice da grade estruturada ao longo da sequência de quadros da animação. Este vértice é obtido pelo comando de seleção de pontos superficiais: *Select Surface Points*. Para visualizar o gráfico gerado é necessário habilitar o painel de navegação *XY Plot* que por padrão vem oculto.

ANEXO C – Formulário de Respostas do Questionário de Avaliação do MWPVis

Formulário de Respostas do Questionário de Avaliação do MWPVis

Anote neste **Formulário de Respostas do Questionário de Avaliação do MWPVis** as suas respostas do **Questionário de Avaliação do MWPVis** e, posteriormente, envie-o para o endereço de e-mail *mologni@dca.fee.unicamp.br*, com o assunto “Minha avaliação do MWPVis”.

QUESTÕES ALTERNATIVAS

	A	B	C	D	E
1.					
1.1					
1.2					
1.3					
1.4					
1.5					
2.					
2.1					
2.2					
2.3					
2.4					
2.5					
2.6					
2.7					
2.8					
3.					
3.1.					
3.1.1					
3.1.2					
3.1.3					
3.1.4					
3.1.5					
3.1.6					
3.2					
3.2.1					
3.2.2					
3.2.3					
3.2.4					
3.2.5					
3.2.6					

	A	B	C	D	E
3.3					
3.3.1					
3.3.2					
3.3.3					
3.3.4					
3.3.5					
3.3.6					
3.4					
3.4.1					
3.4.2					
3.4.3					
3.4.4					
3.4.5					
3.5					
3.5.1					
3.5.2					
3.5.3					
3.5.4					
3.6					
3.7					
3.8					
3.9					
3.10					
3.10.1					
3.10.2					
3.10.3					
3.10.4					
3.11					
3.11.1					
3.11.2					

	A	B	C	D	E
3.11.3					
3.12					
3.13					
4.					
4.1					
4.2					
4.3					
4.4					
4.5					
4.6					
5.					
5.1					
5.2					
5.3					
5.4					
5.5					
6.					
6.1					
6.2					
6.3					
6.4					
6.5					
6.6					
7.					
7.1					
7.2					
7.3					
7.4					
7.5					
7.6					

QUESTÕES DISSERTATIVAS

1.3

7.6
