

**Alexandre Magno Milagre**

**Condições de Fronteiras de Absorção  
no Método FDTD**

Dissertação apresentada à Escola  
Politécnica da Universidade de São  
Paulo para obtenção do Título de  
Mestre em Engenharia Elétrica.

São Paulo 2007

**Alexandre Magno Milagre**

**Condições de Fronteiras de Absorção  
no Método FDTD**

Dissertação apresentada à Escola  
Politécnica da Universidade de São  
Paulo para obtenção do Título de  
Mestre em Engenharia Elétrica.

Área de Concentração:  
Sistemas Eletrônicos

Orientador:  
Prof. Dr. Silvio Ernesto Barbin

São Paulo 2007

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 13 de Agosto de 2007

Assinatura do autor : \_\_\_\_\_

Assinatura do orientador : \_\_\_\_\_

## FICHA CATALOGRÁFICA

**Milagre, Alexandre Magno**  
**Condições de Fronteiras de Absorção no Método FDTD / A.**  
**M. Milagre. -- São Paulo, 2007.**  
115 p.

**Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Telecomunicações e Controle.**

**1.Eletromagnetismo 2.Métodos numéricos 3.Diferenças finitas I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Telecomunicações e Controle II.t.**

Dedico este trabalho ao meu pai, a minha mãe e ao meu irmão, que sempre farão parte da minha vida.

# **AGRADECIMENTOS**

Gostaria de agradecer a todos que de uma forma ou de outra contribuíram para o desenvolvimento desta dissertação, em especial ao meu orientador Professor Dr. Silvio Ernesto Barbin, por sua ajuda, dedicação e paciência durante todo o desenvolvimento e conclusão deste trabalho. À CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, pelo financiamento de meu mestrado. Aos meus colegas de mestrado e do LCS – Laboratório de Comunicações e Sinais do Departamento de Engenharia de Telecomunicações e Controle da Escola Politécnica da USP, pelo apoio e incentivo. E por fim à minha mãe, que sempre me apoiou em várias etapas da minha vida.

# RESUMO

Em muitas simulações eletromagnéticas utilizando o método *FDTD*, é desejado que os campos radiados pelas estruturas em análise sejam transmitidos para fora do domínio computacional. Infelizmente isto não é possível de ser realizado através do método *FDTD* em sua forma original. Para resolver este problema, deve-se implementar, nas superfícies limítrofes dos domínios computacionais, condições especiais denominadas na literatura técnica de Condições de Fronteiras de Absorção, ou, em inglês, “Absorbing Boundary Conditions” (*ABC*’s).

Essas Condições de Fronteiras de Absorção impedem que os campos radiados sejam refletidos nas superfícies limítrofes dos domínios computacionais, retornando para o interior do domínio e interferindo no resultado final das simulações. Não existe uma técnica de absorção ideal, ou seja, que elimine totalmente a reflexão. As técnicas atualmente existentes possuem vantagens e desvantagens, podendo ser mais ou menos eficientes, o que faz com que esse tema ainda seja motivo de extensivos estudos.

O objetivo deste trabalho consiste no estudo, implementação e comparação de Condições de Fronteiras de Absorção e na indicação de uma possível melhoria nessa área. São realizadas simulações em domínios bidimensionais e tridimensionais para se determinar vantagens e desvantagens de cada técnica de absorção. A análise dos resultados das simulações está focalizada no grau de atenuação que as *ABC*s possuem e na carga computacional despendidas por elas.

Este trabalho é concluído com simulações empregando as condições de fronteiras analisadas para três estruturas clássicas. As vantagens e desvantagens de cada *ABC* são apresentadas e uma melhoria proposta na técnica de Auto Teleportação de Campos, ou, em inglês, “Self Teleportation of Fields” é validada. As estruturas analisadas são uma microlinha de transmissão, um filtro planar e um cilindro metálico iluminado por uma onda plana uniforme.

# ABSTRACT

In many electromagnetic computational simulations using the *FDTD* method, it is desired that the electromagnetic fields radiated by the structures under analysis can be transmitted outwards the computational domain. Unfortunately, this is impossible to be done by the *FDTD* method in its original form. To mitigate this problem, one must apply special conditions to the computational domain boundaries, known in the technical literature as Absorbing Boundary Conditions (*ABCs*)

These Absorbing Boundaries Conditions prevent the radiated fields to be reflected by boundaries back into the computational domain. Without them, these fields would interfere with the final simulation results. However, there is no ideal technique that completely eliminates the reflections. The existing techniques have advantages and disadvantages, which make them more or less efficient, still making this subject a theme of extensive studies.

This work is aimed at studying, implementing and comparing these Absorbing Boundary Conditions and at indicating a possible improvement in this field. Simulations in bi-dimensional and three-dimensional domains were made to evaluate advantages and disadvantages of each absorption technique. The analysis of the simulation results was focused in the attenuation degree of the *ABCs* and their computational burden.

The work is concluded with simulations using the analyzed *ABCs* for three classic structures. The advantages and disadvantages of each *ABC* are presented and a proposed improvement on the “Self Teleportation of Fields” technique is validated. The analyzed structures are a microstrip line, a planar filter and a metallic cylinder illuminated by a uniform plane wave.

# SUMÁRIO

1	Introdução.....	1
1.1	Histórico.....	1
1.2	Objetivo e Metodologia.....	2
1.3	Justificativa.....	3
1.4	Conteúdo.....	3
2	Fronteiras de Re-Radiação.....	5
2.1	Introdução.....	5
2.2	Equação de Onda de Sentido Único.....	6
2.3	Aproximações de Mur.....	9
2.3.1	Resultados da Primeira e Segunda Aproximações de Mur.....	12
2.4	Aproximação por Operador de Higdon.....	18
2.4.1	Resultados da Aproximação por Operador de Higdon.....	20
2.5	Conclusão.....	23
3	Camadas Perfeitamente Casadas.....	24
3.1	Introdução.....	24
3.2	Equacionamento de Berenger.....	24
3.2.1	Propagação em um Meio PML.....	26
3.3	Equacionamento de Gedney.....	30
3.3.1	Implementação em FDTD.....	33
3.4	Resultados das Simulações.....	35
3.5	Conclusão.....	42
4	Auto Teleportação de Campos.....	44
4.1	Introdução.....	44
4.2	Implementação da Auto Teleportação no Método <i>FDTD</i> .....	45
4.2.1	Resultados da Auto Teleportação de Campos.....	47
4.3	Compensação Espacial.....	54
4.3.1	Resultados da Compensação Espacial.....	55
4.3.2	Desvantagem da Compensação Espacial.....	57
4.4	Compensação Temporal.....	58
4.4.1	Resultados da Compensação Temporal.....	64
4.4.2	Desvantagem da Compensação Temporal.....	69
4.5	Compensação por Equação de Onda.....	69
4.5.1	Implementação Utilizando a Primeira Aproximação de Mur.....	70
4.5.2	Implementação Utilizando a Aproximação de Higdon.....	72
4.5.3	Resultados da Compensação por Equação de Onda.....	72
4.6	Conclusão.....	74
5	Resultados Numéricos.....	76
5.1	Introdução.....	76
5.2	Estudo da <i>STL</i> em Domínios Bidimensionais.....	76
5.2.1	<i>STL</i> sem Compensação.....	76
5.2.2	<i>STL</i> com Compensações Temporais Diferentes.....	78
5.2.3	<i>STL</i> com Compensações por Equação de Onda.....	79
5.2.4	<i>STL</i> Finalizada por Fronteira de Re-Radiação.....	80
5.3	Estudo da <i>STL</i> em Domínios Tridimensionais.....	82



5.3.1	<i>STL</i> sem Compensação .....	83
5.3.2	<i>STL</i> com Compensação Temporal .....	84
5.3.3	<i>STL</i> com Compensação por Equação de Onda .....	85
5.3.4	<i>STL</i> Finalizada por Fronteira de Re-Radiação .....	86
5.4	Comparação entre Técnicas de Absorção .....	86
5.4.1	Comparação em Domínios Bidimensionais .....	87
5.4.2	Comparação em Domínios Tridimensionais .....	87
5.4.3	Carga Computacional .....	88
5.5	Exemplos Práticos .....	91
5.5.1	Microlinha de Transmissão .....	91
5.5.2	Filtro Planar .....	93
5.5.3	Cilindro Iluminado por uma Onda Plana .....	95
5.6	Conclusão .....	98
6	Considerações Finais .....	99
	Anexo A .....	101
	Anexo B .....	110
	Anexo C .....	113
	Anexo D .....	116
	Referências .....	159

# LISTA DE FIGURAS

Figura 2.1 Utilização das diferenças centrais em um domínio computacional. ....	5
Figura 2.2 Falha da técnica das diferenças centrais na fronteira do domínio.....	5
Figura 2.3 Discretização proposta por Mur. ....	10
Figura 2.4 Implementação da segunda aproximação de Mur. ....	12
Figura 2.5 Falha da segunda aproximação em um vértice. ....	12
Figura 2.6 Interface entre metal e dielétrico em contato com a fronteira.....	12
Figura 2.7 Descontinuidade do campo elétrico na interface. ....	12
Figura 2.8 Ilustração do domínio computacional. $\theta$ é o ângulo de incidência na fronteira. .....	13
Figura 2.9 Intensidade relativa do campo $E_z$ para $\epsilon = \epsilon_0$ . ....	14
Figura 2.10 Intensidade relativa do campo $E_z$ para $\epsilon = 2\epsilon_0$ . ....	14
Figura 2.11 Intensidade relativa do campo $E_z$ para $\epsilon = 5\epsilon_0$ . ....	14
Figura 2.12 Intensidade relativa do campo $E_z$ para $\epsilon = 10\epsilon_0$ . ....	14
Figura 2.13 Intensidade relativa do campo $E_z$ para $\epsilon = \epsilon_0$ . ....	15
Figura 2.14 Intensidade relativa do campo $E_z$ para $\epsilon = 2\epsilon_0$ . ....	15
Figura 2.15 Intensidade relativa do campo $E_z$ para $\epsilon = 5\epsilon_0$ . ....	15
Figura 2.16 Intensidade relativa do campo $E_z$ para $\epsilon = 10\epsilon_0$ . ....	15
Figura 2.17 Domínio computacional para o cálculo do erro e da energia.....	16
Figura 2.18 Erro quadrático global do campo elétrico para a primeira e segunda aproximações de Mur. ....	17
Figura 2.19 Energia no domínio computacional. ....	18
Figura 2.20 Intensidade relativa do campo elétrico para $\alpha = 0^\circ$ . ....	21
Figura 2.21 Intensidade relativa do campo elétrico para $\alpha = 30^\circ$ . ....	21
Figura 2.22 Intensidade relativa do campo elétrico para $\alpha = 45^\circ$ . ....	21
Figura 2.23 Intensidade relativa do campo elétrico para $\alpha = 50^\circ$ . ....	21
Figura 2.24 Erro global do campo elétrico para as aproximações de Higdon.....	22
Figura 2.25 Energia no domínio computacional para os ângulos $\alpha$ simulados anteriormente. ....	23
Figura 3.1 Propagação de uma onda no modo <i>TE</i> . ....	27
Figura 3.2 Estrutura de implementação de um <i>PML</i> em duas dimensões para o modo <i>TE</i> . ....	29
Figura 3.3 Onda incidindo em uma camada <i>PML</i> . ....	30
Figura 3.4 Ilustração do domínio computacional. $\theta$ é o ângulo de incidência no <i>PML</i> . ..	36
Figura 3.5 Intensidade relativa do campo $E_z$ para $L = 4$ e $m = 2$ . ....	37
Figura 3.6 Intensidade relativa do campo $E_z$ para $L = 6$ e $m = 2$ . ....	37
Figura 3.7 Intensidade relativa do campo $E_z$ para $L = 8$ e $m = 2$ . ....	37
Figura 3.8 Intensidade relativa do campo $E_z$ para $L = 10$ e $m = 2$ . ....	37
Figura 3.9 Intensidade relativa do campo $E_z$ para $L = 4$ e $m = 4$ . ....	37
Figura 3.10 Intensidade relativa do campo $E_z$ para $L = 6$ e $m = 4$ . ....	37
Figura 3.11 Intensidade relativa do campo $E_z$ para $L = 8$ e $m = 4$ . ....	38
Figura 3.12 Intensidade relativa do campo $E_z$ para $L = 10$ e $m = 4$ . ....	38
Figura 3.13 Intensidade relativa do campo $E_z$ para $\epsilon = 1\epsilon_0$ e $\sigma = 10.6103$ S/m. ....	39
Figura 3.14 Intensidade relativa do campo $E_z$ para $\epsilon = 2\epsilon_0$ e $\sigma = 10.6103$ S/m. ....	39
Figura 3.15 Intensidade relativa do campo $E_z$ para $\epsilon = 5\epsilon_0$ e $\sigma = 10.6103$ S/m. ....	39

Figura 3.16 Intensidade relativa do campo $E_z$ para $\epsilon = 10\epsilon_0$ e $\sigma = 10.6103$ S/m. ....	39
Figura 3.17 Intensidade relativa do campo $E_z$ para $\epsilon = 1\epsilon_0$ e $\sigma = 3.3553$ S/m. ....	39
Figura 3.18 Intensidade relativa do campo $E_z$ para $\epsilon = 2\epsilon_0$ e $\sigma = 3.3553$ S/m. ....	39
Figura 3.19 Intensidade relativa do campo $E_z$ para $\epsilon = 5\epsilon_0$ e $\sigma = 3.3553$ S/m. ....	40
Figura 3.20 Intensidade relativa do campo $E_z$ para $\epsilon = 10\epsilon_0$ e $\sigma = 3.3553$ S/m. ....	40
Figura 3.21 Ilustração do domínio computacional utilizado para o cálculo do erro e da energia. ....	40
Figura 3.22 Erro global de algumas camadas <i>PML</i> . ....	41
Figura 3.23 Energia no domínio computacional de algumas camadas <i>PML</i> . ....	42
Figura 4.1 Ilustração da técnica da Auto Teleportação de Campos. ....	44
Figura 4.2 Fluxograma da implementação da Auto Teleportação de Campos no método <i>FDTD</i> . ....	46
Figura 4.3 Domínio computacional utilizado. $\theta$ é o ângulo de incidência da onda na fronteira de Auto Teleportação. ....	47
Figura 4.4 Intensidade relativa do campo $E_z$ no ponto A. ....	48
Figura 4.5 Domínio computacional utilizado. $\theta$ é o ângulo de incidência da onda na fronteira de Auto Teleportação. ....	48
Figura 4.6 Intensidade relativa do campo $E_z$ no ponto B. ....	49
Figura 4.7 Comportamento do campo elétrico ao atravessar uma fronteira de Auto Teleportação. ....	50
Figura 4.8 Comportamento do campo elétrico ao atravessar uma fronteira de Auto Teleportação e ser refletida por um <i>PEC</i> . ....	51
Figura 4.9 Ilustração do domínio computacional utilizado para o cálculo do erro e da energia. ....	52
Figura 4.10 Erro global para o domínio computacional. ....	53
Figura 4.11 Energia no domínio computacional. ....	53
Figura 4.12 Diferença entre uma grade uniforme de uma grade não uniforme. ....	55
Figura 4.13 Intensidade relativa do campo $E_z$ ....	56
Figura 4.14 Intensidade relativa do campo $E_z$ ....	56
Figura 4.15 Intensidade relativa do campo $E_z$ ....	56
Figura 4.16 Intensidade relativa do campo $E_z$ ....	56
Figura 4.17 Intensidade relativa do campo $E_z$ ....	57
Figura 4.18 Intensidade relativa do campo $E_z$ ....	57
Figura 4.19 Intensidade relativa do campo $E_z$ ....	57
Figura 4.20 Intensidade relativa do campo $E_z$ ....	57
Figura 4.21 Diagrama das ondas copiadas e originais. ....	59
Figura 4.22 Atenuação teórica para uma frente de onda plana. ....	60
Figura 4.23 Diagrama para o cálculo das relações $(r_1 - r_2)$ e $\left(\frac{r_1}{r_2}\right)$ ....	62
Figura 4.24 Comparação entre a atenuação teórica e simulada para uma frente de onda cilíndrica com três atrasos diferentes. ....	63
Figura 4.25 Ilustração de um domínio computacional unidimensional. ....	64
Figura 4.26 Intensidade relativa do campo $E_z$ para $a = 1,4142$ . ....	65
Figura 4.27 Intensidade relativa do campo $E_z$ para $a = 1,2$ . ....	65
Figura 4.28 Intensidade relativa do campo $E_z$ para $a = 1,0$ . ....	65
Figura 4.29 Intensidade relativa do campo $E_z$ para $a = 0,5$ . ....	65
Figura 4.30 Intensidade relativa do campo $E_z$ para $\epsilon = \epsilon_0$ . ....	66
Figura 4.31 Intensidade relativa do campo $E_z$ para $\epsilon = 2\epsilon_0$ . ....	66

Figura 4.32 Intensidade relativa do campo $E_z$ para $\varepsilon = 5\varepsilon_0$ .	66
Figura 4.33 Intensidade relativa do campo $E_z$ para $\varepsilon = 10\varepsilon_0$ .	66
Figura 4.34 Curvas de atraso em função do ângulo de incidência para vários valores de $\varepsilon$ .	67
Figura 4.35 Intensidade relativa do campo $E_z$ para $\varepsilon = \varepsilon_0$ .	67
Figura 4.36 Intensidade relativa do campo $E_z$ para $\varepsilon = 2\varepsilon_0$ .	67
Figura 4.37 Intensidade relativa do campo $E_z$ para $\varepsilon = 5\varepsilon_0$ .	68
Figura 4.38 Intensidade relativa do campo $E_z$ para $\varepsilon = 10\varepsilon_0$ .	68
Figura 4.39 Erro global.	68
Figura 4.40 Energia no domínio computacional	68
Figura 4.41 Esquema de implementação da compensação por equação de onda.	70
Figura 4.42 Posicionamento do campo elétrico $E_z$ (em vermelho) e do campo magnético $H_x$ (em azul) no domínio computacional.	71
Figura 4.43 Intensidade relativa do campo $E_z$	73
Figura 4.44 Intensidade relativa do campo $E_z$	73
Figura 4.45 Intensidade relativa do campo $E_z$	73
Figura 4.46 Intensidade relativa do campo $E_z$	73
Figura 4.47 Erro global para $\alpha = 0^\circ$ .	74
Figura 4.48 Energia no domínio computacional para $\alpha = 0^\circ$ .	74
Figura 4.49 Erro global para $\alpha = 45^\circ$ .	74
Figura 4.50 Energia no domínio computacional para $\alpha = 45^\circ$ .	74
Figura 5.1 Erro global.	77
Figura 5.2 Energia no domínio computacional.	77
Figura 5.3 Intensidade relativa do campo $E_z$ para $\lambda/\Delta = 10$ .	77
Figura 5.4 Intensidade relativa do campo $E_z$ para $\lambda/\Delta = 20$ .	77
Figura 5.5 Intensidade relativa do campo $E_z$ para $\lambda/\Delta = 30$ .	77
Figura 5.6 Intensidade relativa do campo $E_z$ para $\lambda/\Delta = 40$ .	77
Figura 5.7 Erro global.	78
Figura 5.8 Energia no domínio computacional.	78
Figura 5.9 Intensidade relativa do campo $E_z$ .	78
Figura 5.10 Erro global.	79
Figura 5.11 Energia no domínio computacional.	79
Figura 5.12 Intensidade relativa do campo $E_z$ para $\lambda/\Delta = 10$ .	80
Figura 5.13 Intensidade relativa do campo $E_z$ para $\lambda/\Delta = 20$ .	80
Figura 5.14 Intensidade relativa do campo $E_z$ para $\lambda/\Delta = 30$ .	80
Figura 5.15 Intensidade relativa do campo $E_z$ para $\lambda/\Delta = 40$ .	80
Figura 5.16 Erro global.	81
Figura 5.17 Energia no domínio computacional.	81
Figura 5.18 Intensidade relativa do campo $E_z$ para $\lambda/\Delta = 10$ .	81
Figura 5.19 Intensidade relativa do campo $E_z$ para $\lambda/\Delta = 20$ .	81
Figura 5.20 Intensidade relativa do campo $E_z$ para $\lambda/\Delta = 30$ .	82
Figura 5.21 Intensidade relativa do campo $E_z$ para $\lambda/\Delta = 40$ .	82
Figura 5.22 Ilustração dos domínios computacionais tridimensionais.	83
Figura 5.23 Erro global.	83
Figura 5.24 Energia no domínio computacional.	83
Figura 5.25 Curvas de atraso em função do ângulo de incidência para um domínio tridimensional.	84

Figura 5.26 Erro global.....	85
Figura 5.27 Energia no domínio computacional. ....	85
Figura 5.28 Erro global.....	85
Figura 5.29 Energia no domínio computacional. ....	85
Figura 5.30 Erro global.....	86
Figura 5.31 Energia no domínio computacional. ....	86
Figura 5.32 Erro global.....	87
Figura 5.33 Energia no domínio computacional. ....	87
Figura 5.34 Erro global.....	88
Figura 5.35 Energia no domínio computacional. ....	88
Figura 5.36 Domínio computacional hexaédrico. ....	89
Figura 5.37 Levantamento dos tempos de simulação entre as técnicas <i>PML</i> e <i>STL</i> . ....	90
Figura 5.38 Desenho esquemático da microlinha de transmissão. ....	92
Figura 5.39 S11 e S21 da microlinha truncada pela 1ª aproximação de Mur.....	92
Figura 5.40 S11 e S21 da microlinha truncada pela técnica <i>PML</i> .....	92
Figura 5.41 S11 e S21 da microlinha truncada pela técnica <i>STL</i> sem compensação. ....	92
Figura 5.42 S11 e S21 da microlinha truncada pela técnica <i>STL</i> com compensação por equação de onda.....	92
Figura 5.43 Esquemático do filtro planar. ....	93
Figura 5.44 S11 e S21 do filtro truncado pela 1ª aproximação de Mur. ....	94
Figura 5.45 S11 e S21 do filtro truncado pela técnica <i>PML</i> ( $L = 10$ , $m = 4$ , $\sigma_{\max} = 16,8993$ ).....	94
Figura 5.46 S11 e S21 do filtro truncado com uma <i>STL</i> finalizada com a 1ª aproximação de Mur.....	94
Figura 5.47 S11 e S21 do filtro truncado com uma <i>STL</i> com compensação por equação de onda e finalizada com a 1ª aproximação de Mur. ....	94
Figura 5.48 Domínio utilizado no problema do cilindro iluminado.....	95
Figura 5.49 Erro percentual entre os domínios ideal e truncados, (a) campo elétrico $E_z$ no domínio ideal, (b) erro para a 1ª aproximação de Mur, (c) erro para uma <i>PML</i> , (d) erro para uma <i>STL</i> . ....	96
Figura 5.50 Erro médio percentual entre os domínios simulados para alguns comprimentos de onda. ....	97

## LISTA DE TABELAS

Tabela 5.1 Operações por célula na primeira aproximação de Mur.....	89
Tabela 5.2 Operações por célula na segunda aproximação de Mur. ....	89
Tabela 5.3 Operações por célula em uma camada <i>PML</i> .....	89
Tabela 5.4 Operações por célula na <i>STL</i> sem compensação. ....	89
Tabela 5.5 Operações por célula na <i>STL</i> com compensação temporal.....	90
Tabela 5.6 Operações por célula na <i>STL</i> com compensação por equação de onda.....	90

# LISTA DE SIGLAS E SÍMBOLOS

<i>ABC</i>	Absorbing Boundary Condition
<i>FDTD</i>	Finite Difference Time Domain
<i>PEC</i>	Perfect Electric Conductor
<i>PMC</i>	Perfect Magnetic Conductor
<i>PML</i>	Perfectly Matched Layer
<i>RF</i>	Radio Frequency
<i>STF</i>	Self – Teleportation of Fields
----	----
<i>E</i>	Campo elétrico
<i>H</i>	Campo magnético
<i>D</i>	Vetor Deslocamento
<i>B</i>	Vetor Indução Magnética
$\epsilon_0$	Permissividade elétrica do vácuo
$\mu_0$	Permeabilidade magnética do vácuo
$\sigma$	Condutividade elétrica
$\rho^*$	Resistividade magnética
<i>v</i>	Velocidade de propagação da onda
$\lambda$	Comprimento de onda
<i>W</i>	Campo escalar genérico
$\Xi$	Operador diferencial para a equação de onda
$\mathcal{D}_x$	Derivada espacial em relação ao eixo <i>x</i>
$\mathcal{D}_y$	Derivada espacial em relação ao eixo <i>y</i>
$\mathcal{D}_z$	Derivada espacial em relação ao eixo <i>z</i>
$\mathcal{D}_t$	Derivada temporal
$\Delta x$	Discretização espacial no eixo <i>x</i>
$\Delta y$	Discretização espacial no eixo <i>y</i>
$\Delta z$	Discretização espacial no eixo <i>z</i>
$\Delta$	Discretização espacial genérica
$\Delta t$	Discretização temporal

$a$	Parâmetro de compensação temporal
$m$	Fator de redução de células não uniformes
$R$	Fator de amortecimento na Auto Teleportação de Campos
$\alpha$	Ângulo de máxima atenuação para o operador de Higdon
$\theta$	Ângulo de incidência da onda na fronteira de absorção
$N_f$	Número de fronteiras de Auto Teleportação de Campos
$\Omega_T$	Domínio computacional truncado
$\Omega_I$	Domínio computacional ideal
$L_I$	Comprimento do domínio computacional ideal
$W_I$	Largura do domínio computacional ideal
$H_I$	Altura do domínio computacional ideal
$L_T$	Comprimento do domínio computacional truncado
$W_T$	Largura do domínio computacional truncado
$H_T$	Altura do domínio computacional truncado
$e_{\text{global}}$	Erro global no domínio computacional
$E_G$	Energia no domínio computacional



# 1 Introdução

## 1.1 Histórico

Em 1966, K. S. Yee [1] apresentou um algoritmo numérico para resolver as equações de Maxwell que foi denominado Método das Diferenças Finitas no Domínio do Tempo, em inglês “*Finite Difference Time Domain*” (*FDTD*). Este algoritmo opera com o espaço e o tempo discretizados, sendo os campos elétrico e magnético obtidos de forma iterativa, conforme detalhado no Anexo A.

Contudo, esse método não recebeu muita atenção por quase duas décadas, pois a maioria das pesquisas em eletromagnetismo estavam concentradas em métodos no domínio da frequência, tais como o Método dos Elementos Finitos, em inglês, “*Finite Element Method*” (*FEM*) e o Método dos Momentos, em inglês, “*Method of Moments*” (*MoM*). A maior razão para esse abandono deve-se a grande quantidade de memória computacional geralmente requerida pelo método *FDTD*, que não era disponível nos computadores convencionais da época. Porém, com o avanço da microeletrônica, os computadores passaram a ter cada vez mais capacidade de memória e maiores velocidades de processamento. Isso despertou novamente o interesse dos pesquisadores pelo método *FDTD*, proporcionando um significativo avanço nas pesquisas voltadas para esse método.

Um outro fator que contribuiu para o progresso do método *FDTD*, foi o desenvolvimento de Condições de Fronteiras de Absorção, em inglês “*Absorbing Boundary Conditions*” (*ABCs*). Essas condições permitem a simulação de estruturas que operam em regiões definidas como abertas, ou seja, em um domínio espacial ilimitado. Certamente não há computadores com memória suficiente para tratar um espaço infinito. Portanto, o domínio computacional deve ser limitado a um tamanho suficiente para abrigar a estrutura de interesse com alguma reserva e sua fronteira deve ser construída de tal forma a simular um espaço aberto para os campos existentes no interior desse domínio.

Um trabalho pioneiro nessa área foi realizado por Engquist e Majda [2] e aperfeiçoado por Mur [3], cujas aproximações de primeira e segunda ordem para as *ABCs* têm sido amplamente utilizadas até hoje. Posteriormente, uma nova categoria de *ABCs* foi desenvolvida por Berenger [4] e aperfeiçoada por Gedney [5]. Ela foi

denominada Camadas Perfeitamente Casadas, em inglês, “Perfectly Matched Layers” (*PML*). As *PMLs* representaram um novo avanço no modelamento de fronteiras de absorção, não somente para o método *FDTD*, mas, também, para outros métodos numéricos, tal como o método das Linhas de Transmissão, em inglês, “*Transmission Line Method*” (*TLM*) [6].

A principal diferença entre as técnicas de absorção mencionadas anteriormente, é que, na técnica de Mur, o domínio computacional é limitado por uma fronteira de re-radiação e, na técnica *PML*, por uma fronteira com dissipação de energia. Enquanto na técnica de Mur o cancelamento das ondas incidentes na fronteira se dá por re-radiação, na técnica *PML* a absorção baseia-se no modelo das câmaras anecóicas. No caso da *PML* a questão fundamental é a modelagem de um material fictício a ser empregado na fronteira.

Com o passar dos anos, diversos pesquisadores trabalharam para melhorar as técnicas já existentes. Recentemente, uma outra técnica de absorção, baseada no modelo de fronteira de re-radiação, foi desenvolvida por Diaz e Scherbatko [7 - 9], tendo sido denominada de Auto Teleportação de Campos, em inglês “Self Teleportation of Fields” (*STF*). Nesta técnica o cancelamento de uma onda incidente na fronteira ocorre por simples adição dos campos da onda original com uma cópia negativa dos mesmos, como será explicado mais adiante neste trabalho. Resultados de simulação como os obtidos por Milagre e Barbin [10], serão, dentre outros, apresentados no texto. Inovações que melhoram o desempenho desta técnica, como as introduzidas por Milagre e Barbin [11] encontram-se incluídas, dentre outras que são apresentadas em detalhe. Será também mostrado que esta técnica resulta em redução da carga computacional [12] para um dado objetivo de desempenho em simulações empregando *FDTD*.

## 1.2 Objetivo e Metodologia

Esta dissertação tem como objetivo o estudo e a implementação de fronteiras de absorção utilizadas no método *FDTD* e a apresentação de uma melhoria na técnica de Auto Teleportação. Foram realizadas diversas simulações para cada técnica de absorção apresentada acima, visando determinar suas vantagens e desvantagens através de uma comparação quantitativa entre elas.

Para o estudo dessas técnicas, foram analisadas estruturas empregando domínios em duas e três dimensões. Nas simulações foram avaliados os erros que os truncamentos dos domínios introduzem em relação a um domínio infinito. Um domínio infinito é simulado fazendo-se o tempo de simulação tal que as ondas geradas pelas fontes no seu interior ainda não tenham sido refletidas pela fronteira do domínio no instante em que os campos são observados. Para todas as técnicas, os parâmetros de absorção de energia e o perfil de atenuação em função do ângulo de incidência foram levantados.

### 1.3 Justificativa

Em simulações eletromagnéticas de estruturas abertas, deseja-se obter os campos radiados tanto no interior como fora do domínio computacional. Infelizmente pela própria natureza do método *FDTD*, os campos fora dos domínios computacionais são assumidos como sendo nulos. Sendo assim, os campos radiados pela estrutura em análise serão refletidos pelas fronteiras limítrofes dos domínios computacionais e com isso interferem no resultado final da simulação. Para contornarmos esse problema, é necessário a implementação de fronteiras absorvedoras nos limites dos domínios computacionais.

Por esse motivo, faz-se necessário o estudo de técnicas que melhorem o desempenho de fronteiras absorvedoras. Esse estudo deve focar principalmente dois tópicos: o grau de atenuação de uma onda incidente e a carga computacional necessária. Uma boa fronteira de absorção deve possuir um grau de atenuação elevado e necessitar de pouca capacidade de processamento para que as simulações sejam realizadas da forma mais rápida possível.

### 1.4 Conteúdo

Este trabalho está organizado de forma a apresentar as principais técnicas utilizadas em fronteiras de absorção no método *FDTD* de forma clara e de fácil entendimento.

O segundo capítulo apresenta a descrição e implementação das primeiras técnicas bem sucedidas utilizadas como fronteiras de absorção em *FDTD*. Elas são conhecidas na literatura técnica como fronteiras de re-radiação, das quais fazem parte as aproximações de Mur e de Higdon.

O terceiro capítulo aborda um segundo avanço nas técnicas utilizadas como fronteira de absorção. A nova técnica utiliza o conceito de materiais absorvedores, tais como utilizados em câmaras anecóicas, para implementar o truncamento de domínios computacionais. Ela ficou conhecida na literatura técnica como Camadas Perfeitamente Casadas.

O quarto capítulo descreve um novo tipo de fronteira de re-radiação que foi recentemente formulado por Diaz e Scherbatko. Este capítulo apresenta ainda três melhorias desta técnica que foram desenvolvidas durante a realização deste trabalho.

O quinto capítulo apresenta análises comparativas entre as técnicas descritas nos três capítulos anteriores. Essas análises estão concentradas na técnica de Auto Teleportação de Campos por ser uma nova técnica e cuja eficácia ainda não foi suficientemente explorada. Ao longo deste capítulo são apresentadas simulações em duas e três dimensões, tabelas comparativas entre as técnicas de absorção com relação ao número de operações matemáticas e exemplos de simulações de estruturas clássicas.

O sexto capítulo apresenta conclusões obtidas com base nos resultados apresentados nesta dissertação e propostas para futuros trabalhos de pesquisa.

# 2 Fronteiras de Re-Radiação

## 2.1 Introdução

A determinação dos campos eletromagnéticos empregando o método *FDTD* exige um domínio computacional finito, isto é, somente pode ser realizada para uma região delimitada do espaço. No caso de estruturas fechadas, o domínio computacional fica definido geralmente pela superfície limítrofe da estrutura. Para estruturas abertas, como no caso de antenas por exemplo, faz-se necessário limitar a região onde os campos serão determinados, truncando-se assim o domínio ilimitado no qual a estrutura está imersa. Isso é feito impondo-se uma fronteira a esse domínio.

Para que as ondas radiadas não sejam refletidas na fronteira do domínio computacional, é necessário impor certas condições matemáticas a esta fronteira. Essas condições são chamadas de Condições de Fronteiras de Absorção, em inglês “Absorbing Boundary Condition” (*ABC*). Infelizmente, elas não podem ser criadas diretamente do método *FDTD*, pois ele utiliza a técnica de diferenças centrais para a discretização das equações de Maxwell na forma diferencial.

Para o cálculo do valor de um campo na posição ( $p$ ) e no instante ( $n + 1$ ) no interior do domínio computacional, utilizando-se diferenças centrais, é necessário ter o conhecimento prévio dos valores do campo ao redor de ( $p$ ) no instante ( $n$ ), como mostrado na Figura 2.1. Para o cálculo do valor do campo numa posição pertencente a uma fronteira, utilizando-se diferenças centrais, surgirá um problema de indeterminação, pois os valores do campo, se existirem, fora desse domínio não são conhecidos, como mostrado na Figura 2.2.

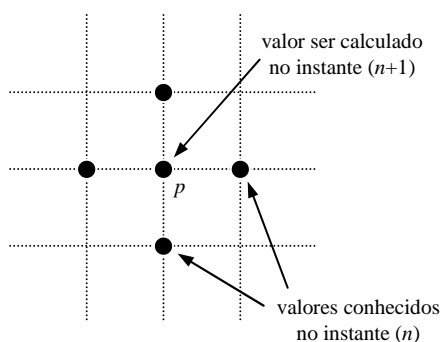


Figura 2.1 Utilização das diferenças centrais em um domínio computacional.

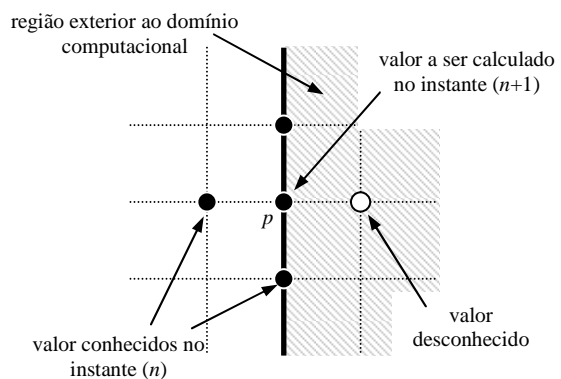


Figura 2.2 Falha da técnica das diferenças centrais na fronteira do domínio.

Uma forma de contornar este problema é utilizar a técnica das diferenças regressivas, em inglês “Backward Finite Difference” ou das diferenças progressivas, em inglês “Forward Finite Difference”. Entretanto, essas técnicas não são utilizadas na maioria dos algoritmos *FDTD* por possuírem menor precisão que a de diferenças centrais.

A forma usual de contornar essa limitação é utilizar a equação de onda para estimar o valor dos campos no exterior do domínio computacional. Essa técnica é chamada de condição de fronteira de re-radiação e tem sido amplamente utilizada no truncamento de domínios computacionais. Este capítulo explora os fundamentos das fronteiras de re-radiação utilizadas para truncar domínios *FDTD*.

## 2.2 Equação de Onda de Sentido Único

A equação diferencial parcial que permite descrever a propagação de ondas em apenas um sentido é denominada “Equação de Onda de Sentido Único”, em inglês “One – Way Wave Equation” e foi desenvolvida por Engquist e Majda [2]. Esta equação pode ser utilizada para estimar o valor dos campos de uma onda fora do domínio computacional.

Em seu trabalho, Engquist e Majda derivaram a equação de onda de sentido único para ser utilizada nas condições de contorno para equações diferenciais. A equação de onda de sentido único pode ser obtida por fatoração do operador diferencial de segunda ordem ( $\Xi$ ), definido em coordenadas cartesianas por

$$\Xi \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} - \frac{1}{v^2} \frac{\partial^2}{\partial t^2} \equiv \mathcal{D}_x^2 + \mathcal{D}_y^2 - \frac{1}{v^2} \mathcal{D}_t^2, \quad (2.1)$$

sendo  $v$  a velocidade da onda.

A equação de onda em duas dimensões e em coordenadas cartesianas é dada por

$$\frac{\partial^2 W}{\partial x^2} + \frac{\partial^2 W}{\partial y^2} - \frac{1}{v^2} \frac{\partial^2 W}{\partial t^2} = 0, \quad (2.2)$$

sendo  $W$  um campo escalar.

Esta equação pode ser reescrita de forma compacta como

$$\Xi W = 0. \quad (2.3)$$

O operador  $\Xi$ , pode ser decomposto em dois operadores diferenciais de primeira ordem  $\Xi^+$  e  $\Xi^-$  definidos por

$$\Xi^+ = \mathcal{D}_x + \frac{\mathcal{D}_t}{v} \sqrt{1 - S^2}, \quad (2.4a)$$

$$\Xi^- = \mathcal{D}_x - \frac{\mathcal{D}_t}{v} \sqrt{1 - S^2}, \quad (2.4b)$$

onde  $S$  é o operador dado por

$$S \equiv \frac{\mathcal{D}_y}{(1/v) \cdot \mathcal{D}_t}. \quad (2.4c)$$

Sendo assim, (2.3) pode ser reescrita na forma

$$[\Xi^+ \cdot \Xi^-] W = 0; \quad (2.5a)$$

$$\Xi^- W = 0; \quad (2.5b)$$

$$\Xi^+ W = 0. \quad (2.5c)$$

A equação (2.5a) descreve uma onda se propagando tanto no sentido progressivo como regressivo. Já as equações (2.5b) ou (2.5c) descrevem uma onda se propagando em apenas um sentido. Engquist e Majda mostram em seu trabalho que para absorver uma onda incidente em uma fronteira, posicionada em  $x = x_{\min}$ , deve-se aplicar (2.5b) e, para uma fronteira posicionada em  $x = x_{\max}$ , deve-se utilizar (2.5c).

Infelizmente a operação raiz quadrada presente em  $\Xi^+$  e  $\Xi^-$  faz com que eles sejam operadores pseudo-diferenciais. Esta característica impede que estes operadores possam ser diretamente implementados nos algoritmos *FDTD*. Para contornar essa dificuldade, as equações obtidas através desses operadores devem ser primeiramente transformadas em equações diferenciais parciais lineares. Uma forma simples de se fazer isso, é através de aproximações de  $\sqrt{1 - S^2}$  por série de Taylor, como por exemplo, mostrado na equação

$$\sqrt{1 - S^2} \cong 1. \quad (2.6)$$

onde é apresentado apenas o primeiro termo da expansão.

Porém, isso cria uma solução aproximada da função original e, por conseguinte, a absorção da onda não é perfeita.

Substituindo-se (2.6) em (2.4b), obtém-se uma aproximação de primeira ordem para o operador  $\Xi^-$ , dada por

$$\Xi^- = \mathcal{D}_x - \frac{\mathcal{D}_t}{v}; \quad (2.7)$$

Assim, a equação de onda (2.5b) pode ser reescrita na forma

$$\frac{\partial W}{\partial x} - \frac{1}{v} \frac{\partial W}{\partial t} = 0. \quad (2.8)$$

Na equação (2.8), é assumido que a derivada parcial em relação a  $y$  é muito pequena em comparação com a derivada parcial em relação a  $t$ . Esta primeira aproximação é válida para pequenos ângulos de incidência, ou seja, para ângulos próximos de zero. Expandindo-se  $\sqrt{1-S^2}$  em série de Taylor até termos de segunda ordem, obtém-se

$$\sqrt{1-S^2} \cong 1 - \frac{1}{2} S^2. \quad (2.9)$$

Substituindo-se (2.9) em (2.4b), obtém-se uma aproximação de segunda ordem para o operador  $\Xi^-$ , dada por

$$\Xi^- = \mathcal{D}_x - \frac{\mathcal{D}_t}{v} + \frac{v}{2} \frac{\mathcal{D}_y^2}{\mathcal{D}_t}. \quad (2.10)$$

Substituindo-se (2.10) em (2.5b) e multiplicando-se os dois lados da igualdade por  $\mathcal{D}_t$ , obtém-se a seguinte equação diferencial parcial:

$$\frac{\partial^2 W}{\partial x \partial t} - \frac{1}{v} \frac{\partial^2 W}{\partial t^2} + \frac{v}{2} \frac{\partial^2 W}{\partial y^2} = 0. \quad (2.11a)$$

As equações acima foram desenvolvidas para uma fronteira posicionada em  $x = x_{\max}$ . Analogamente, pode-se repetir este mesmo procedimento para as fronteiras posicionadas em  $x = x_{\min}$ ,  $y = y_{\max}$  e  $y = y_{\min}$ , obtendo-se as equações

$$\frac{\partial^2 W}{\partial x \partial t} + \frac{1}{v} \frac{\partial^2 W}{\partial t^2} - \frac{v}{2} \frac{\partial^2 W}{\partial y^2} = 0 \quad \text{para } x = x_{\max}, \quad (2.11b)$$

$$\frac{\partial^2 W}{\partial y \partial t} - \frac{1}{v} \frac{\partial^2 W}{\partial t^2} + \frac{v}{2} \frac{\partial^2 W}{\partial x^2} = 0 \quad \text{para } y = y_{\min}, \quad (2.11c)$$

$$\frac{\partial^2 W}{\partial y \partial t} + \frac{1}{v} \frac{\partial^2 W}{\partial t^2} - \frac{v}{2} \frac{\partial^2 W}{\partial x^2} = 0 \quad \text{para } y = y_{\max}, \quad (2.11d)$$

Para domínios computacionais tridimensionais o mesmo procedimento deve ser seguido, porém, utilizando-se o operador

$$\Xi \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} - \frac{1}{v^2} \frac{\partial^2}{\partial t^2} \equiv \mathcal{D}_x^2 + \mathcal{D}_y^2 + \mathcal{D}_z^2 - \frac{1}{v^2} \mathcal{D}_t^2, \quad (2.12a)$$



o qual pode ser decomposto da forma análoga à apresentada em (2.4a – 2.4b), com  $S$  dado por

$$S^2 \equiv \frac{(\mathcal{D}_y^2 + \mathcal{D}_z^2)}{((1/\nu) \cdot \mathcal{D}_t)^2}. \quad (2.12b)$$

A aproximação de primeira ordem para (2.12b) é a mesma dada por (2.6) e a equação a ser implementada em um domínio tridimensional é obtida de (2.8). A aproximação de segunda ordem é então dada pelas equações

$$\frac{\partial^2 W}{\partial x \partial t} - \frac{1}{\nu} \frac{\partial^2 W}{\partial t^2} + \frac{\nu}{2} \frac{\partial^2 W}{\partial y^2} + \frac{\nu}{2} \frac{\partial^2 W}{\partial z^2} = 0 \quad \text{para } x = x_{\min}, \quad (2.13a)$$

$$\frac{\partial^2 W}{\partial x \partial t} + \frac{1}{\nu} \frac{\partial^2 W}{\partial t^2} - \frac{\nu}{2} \frac{\partial^2 W}{\partial y^2} - \frac{\nu}{2} \frac{\partial^2 W}{\partial z^2} = 0 \quad \text{para } x = x_{\max}, \quad (2.13b)$$

$$\frac{\partial^2 W}{\partial y \partial t} - \frac{1}{\nu} \frac{\partial^2 W}{\partial t^2} + \frac{\nu}{2} \frac{\partial^2 W}{\partial x^2} + \frac{\nu}{2} \frac{\partial^2 W}{\partial z^2} = 0 \quad \text{para } y = y_{\min}, \quad (2.13c)$$

$$\frac{\partial^2 W}{\partial y \partial t} + \frac{1}{\nu} \frac{\partial^2 W}{\partial t^2} - \frac{\nu}{2} \frac{\partial^2 W}{\partial x^2} - \frac{\nu}{2} \frac{\partial^2 W}{\partial z^2} = 0 \quad \text{para } y = y_{\max}, \quad (2.13d)$$

$$\frac{\partial^2 W}{\partial z \partial t} - \frac{1}{\nu} \frac{\partial^2 W}{\partial t^2} + \frac{\nu}{2} \frac{\partial^2 W}{\partial x^2} + \frac{\nu}{2} \frac{\partial^2 W}{\partial y^2} = 0 \quad \text{para } z = z_{\min}, \quad (2.13e)$$

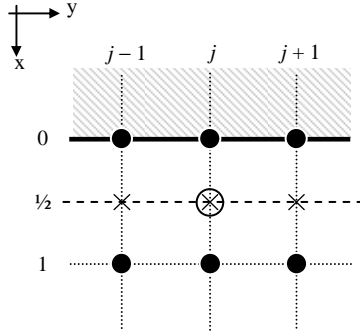
$$\frac{\partial^2 W}{\partial z \partial t} + \frac{1}{\nu} \frac{\partial^2 W}{\partial t^2} - \frac{\nu}{2} \frac{\partial^2 W}{\partial x^2} - \frac{\nu}{2} \frac{\partial^2 W}{\partial y^2} = 0 \quad \text{para } z = z_{\max}, \quad (2.13f)$$

As formulações apresentadas tanto no caso bidimensional quanto no caso tridimensional empregam aproximações de primeira ou segunda ordem. Aproximações de ordem superiores podem ser utilizadas. Porém, simulações realizadas em [2] e [3] mostram que elas não produzem melhora significativa nos resultados.

### 2.3 Aproximações de Mur

A seção 2.2 apresentou uma forma de se obter o valor do campo elétrico ou magnético em uma posição fora do domínio computacional, utilizando-se a equação de onda de sentido único. Nessa seção, foi apresentada uma forma de transformar as equações pseudo-diferenciais em equações diferenciais parciais lineares. Para se utilizar estas equações como condições de fronteiras de absorção, deve-se implementá-las na forma de diferenças finitas.

Uma possível implementação está descrita no trabalho de Mur [3]. Neste trabalho, as equações diferenciais parciais são reescritas na forma de diferenças centrais expandidas. O equacionamento a seguir supõe uma fronteira posicionada em  $x = 0$ . A expansão é realizada em torno do ponto  $(1/2, j)$  mostrado na Figura 2.3.



**Figura 2.3** Discretização proposta por Mur.

Na discretização proposta por Mur, os componentes da equação (2.8) são reescritos na forma

$$\frac{\partial W}{\partial x} \Big|_{1/2,j}^{n+1/2} = \frac{W|_{(1,j)}^{n+1/2} - W|_{0,j}^{n+1/2}}{\Delta x} = \frac{1}{\Delta x} \left[ \frac{W|_{(1,j)}^{n+1} + W|_{1,j}^n}{2} - \frac{W|_{0,j}^{n+1} + W|_{0,j}^n}{2} \right], \quad (2.14a)$$

$$\frac{\partial W}{\partial t} \Big|_{1/2,j}^{n+1/2} = \frac{W|_{1/2,j}^{n+1} - W|_{1/2,j}^n}{\Delta t} = \frac{1}{\Delta t} \left[ \frac{W|_{1,j}^{n+1} + W|_{0,j}^{n+1}}{2} - \frac{W|_{1,j}^n + W|_{0,j}^n}{2} \right]. \quad (2.14b)$$

Reagrupando-se estas equações em (2.8), obtém-se o valor do campo na fronteira  $x = x_{\min}$ , dado por

$$W|_{0,j,k}^{n+1} = W|_{1,j,k}^n + \frac{v\Delta t - \Delta x}{v\Delta t + \Delta x} \left[ W|_{1,j,k}^{n+1} - W|_{0,j,k}^n \right]. \quad (2.15)$$

Para as fronteiras posicionadas em  $x = x_{\max}$ ,  $y = y_{\min}$  e  $y = y_{\max}$ , basta repetir o mesmo procedimento apresentado.

Para a implementação da aproximação de segunda ordem em *FDTD*, Mur apresentou as discretizações

$$\begin{aligned} \frac{\partial^2 W}{\partial x \partial t} \Big|_{1/2,j}^n &= \frac{1}{2\Delta t} \left( \frac{\partial W}{\partial x} \Big|_{1/2,j}^{n+1} - \frac{\partial W}{\partial x} \Big|_{1/2,j}^{n-1} \right), \\ &= \frac{1}{2\Delta t} \left[ \left( \frac{W|_{1,j}^{n+1} - W|_{0,j}^{n+1}}{\Delta x} \right) - \left( \frac{W|_{1,j}^{n-1} - W|_{0,j}^{n-1}}{\Delta x} \right) \right], \end{aligned} \quad (2.16a)$$

$$\begin{aligned}\frac{\partial^2 W}{\partial t^2}\Big|_{1/2,j}^n &= \frac{1}{2} \left( \frac{\partial^2 W}{\partial t^2}\Big|_{0,j}^n + \frac{\partial^2 W}{\partial t^2}\Big|_{1,j}^n \right), \\ &= \frac{1}{2} \left[ \left( \frac{W_{0,j}^{n+1} - 2W_{0,j}^n + W_{0,j}^{n-1}}{(\Delta t)^2} \right) - \left( \frac{W_{1,j}^{n+1} - 2W_{1,j}^n + W_{1,j}^{n-1}}{(\Delta t)^2} \right) \right]\end{aligned}\quad (2.16b)$$

$$\begin{aligned}\frac{\partial^2 W}{\partial y^2}\Big|_{1/2,j}^n &= \frac{1}{2} \left( \frac{\partial^2 W}{\partial y^2}\Big|_{0,j}^n + \frac{\partial^2 W}{\partial y^2}\Big|_{1,j}^n \right), \\ &= \frac{1}{2} \left[ \left( \frac{W_{0,j+1}^n - 2W_{0,j}^n + W_{0,j-1}^n}{(\Delta y)^2} \right) + \left( \frac{W_{1,j+1}^n - 2W_{1,j}^n + W_{1,j-1}^n}{(\Delta y)^2} \right) \right]\end{aligned}\quad (2.16c)$$

para os termos de (2.11a). Reagrupando-se estes termos e fazendo-se as devidas manipulações algébricas, obtém-se

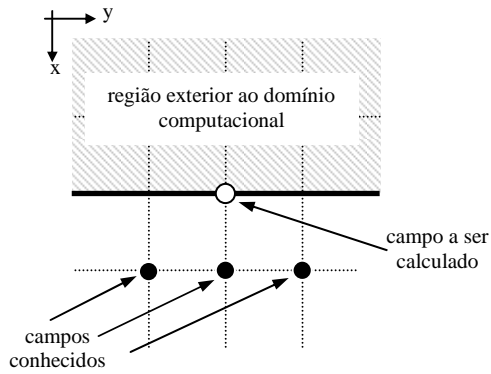
$$\begin{aligned}W_{0,j}^{n+1} &= -W_{1,j}^{n-1} + \frac{v\Delta t - \Delta x}{v\Delta t + \Delta x} [W_{1,j}^{n+1} - W_{0,j}^{n-1}] + \frac{2\Delta x}{v\Delta t + \Delta x} [W_{0,j}^n - W_{1,j}^n] + \\ &+ \frac{\Delta x(v\Delta t)^2}{2(\Delta y)^2(v\Delta t + \Delta x)} [W_{0,j+1}^n - 2W_{0,j}^n + W_{0,j-1}^n + \\ &+ W_{1,j+1}^n - 2W_{1,j}^n + W_{1,j-1}^n]\end{aligned}\quad (2.17)$$

Para domínios tridimensionais, equações (2.13a – 2.13f), basta seguir o mesmo procedimento realizado acima. A equação

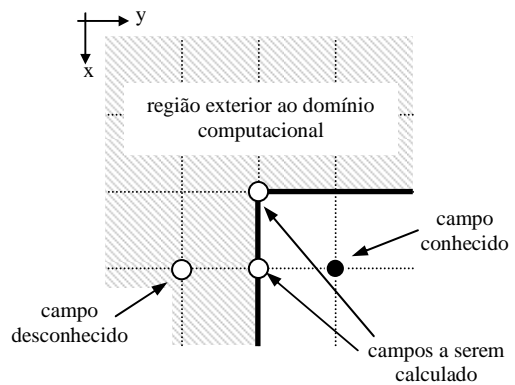
$$\begin{aligned}W_{0,j,k}^{n+1} &= -W_{1,j,k}^{n-1} + \frac{v\Delta t - \Delta x}{v\Delta t + \Delta x} [W_{1,j,k}^{n+1} - W_{0,j,k}^{n-1}] + \frac{2\Delta x}{v\Delta t + \Delta x} [W_{0,j,k}^n - W_{1,j,k}^n] + \\ &+ \frac{\Delta x(v\Delta t)^2}{2(\Delta y)^2(v\Delta t + \Delta x)} [W_{0,j+1,k}^n - 2W_{0,j,k}^n + W_{0,j-1,k}^n + \\ &+ W_{1,j+1,k}^n - 2W_{1,j,k}^n + W_{1,j-1,k}^n] + \\ &+ \frac{\Delta x(v\Delta t)^2}{2(\Delta z)^2(v\Delta t + \Delta x)} [W_{0,j,k+1}^n - 2W_{0,j,k}^n + W_{0,j,k-1}^n + \\ &+ W_{1,j,k+1}^n - 2W_{1,j,k}^n + W_{1,j,k-1}^n]\end{aligned}\quad (2.18)$$

é a discretização de (2.13a).

Uma consideração importante deve ser feita quanto à segunda aproximação de Mur: ela não pode ser aplicada nos vértices do domínio computacional. O motivo é que como são utilizadas segundas derivadas espaciais, alguns valores necessários para a atualização das equações se encontram fora do domínio computacional e são portanto desconhecidos, como ilustrado nas Figuras 2.4 e 2.5.

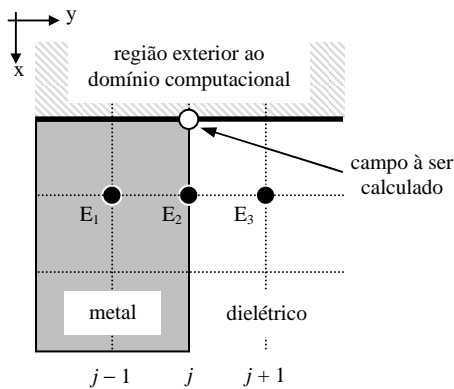


**Figura 2.4 Implementação da segunda aproximação de Mur.**

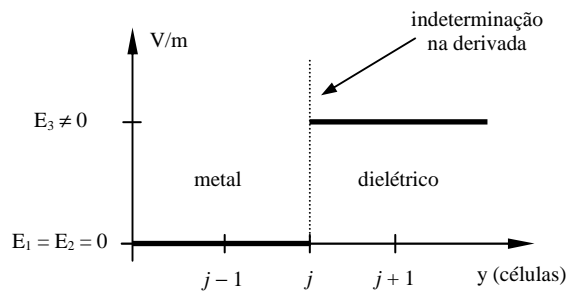


**Figura 2.5 Falha da segunda aproximação em um vértice.**

A segunda aproximação de Mur também não pode ser aplicada nas interfaces dielétrico – metal e dielétrico – dielétrico que estejam em contato com a fronteira de absorção. Como o valor dos campos incidentes antes e depois dessas interfaces não são contínuos, as derivadas parciais em torno deste ponto tornam-se indefinidas. As Figuras 2.6 e 2.7 ilustram o que ocorre com o campo elétrico incidente em uma fronteira dielétrico – metal. Para resolver este problema, é necessário utilizar a primeira aproximação de Mur nos pontos onde estas interfaces estão em contato com as fronteiras de absorção.



**Figura 2.6 Interface entre metal e dielétrico em contato com a fronteira.**



**Figura 2.7 Descontinuidade do campo elétrico na interface.**

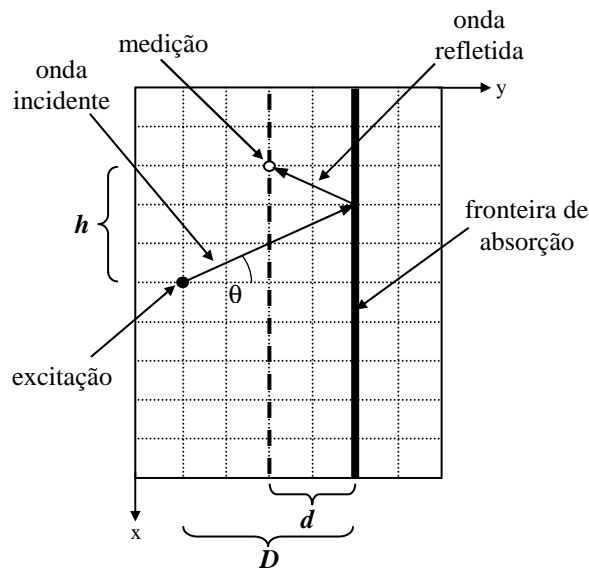
### 2.3.1 Resultados da Primeira e Segunda Aproximações de Mur

As simulações a seguir apresentam o perfil de atenuação de uma fronteira de absorção implementada através da técnica de Mur. É utilizado um domínio

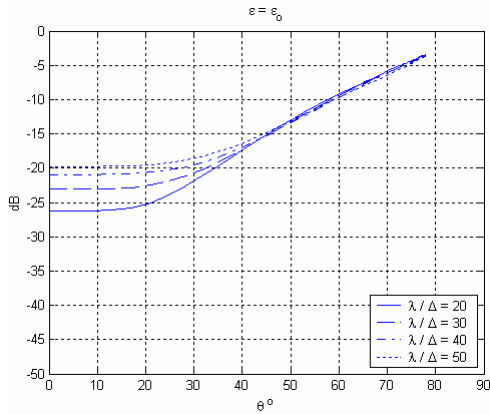
computacional de  $\{200 \times 150\}$  células, um pulso gaussiano, como fonte de excitação, posicionado em  $x_o = 100$  células e  $y_o = 75$  células e uma fronteira de absorção posicionada em  $y_{ABC} = 85$  células. A Figura 2.8 ilustra o domínio computacional. Sendo  $D$  a distância entre o ponto de excitação e a fronteira de absorção,  $d$  é a distância entre esta fronteira e o ponto de medição e  $h$  é a distância entre os pontos de excitação e de medição. Nesta simulação é utilizado  $D = 10$  células e  $d = 3$  células.

### Primeira Aproximação de Mur

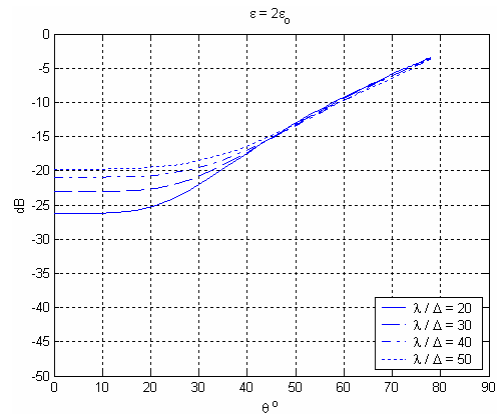
As Figuras 2.9 – 2.12 apresentam a intensidade relativa do campo elétrico de uma onda refletida pela fronteira de absorção em relação ao campo elétrico de uma onda sem a presença dessa fronteira. Essas intensidades são mostradas para alguns valores comprimentos de onda  $\lambda$  normalizados em relação a dimensão  $\Delta$  de uma célula. As simulações são realizadas para diversos valores da constante dielétrica  $\epsilon$  do meio e a discretização temporal  $\Delta t$  é mantida constante.



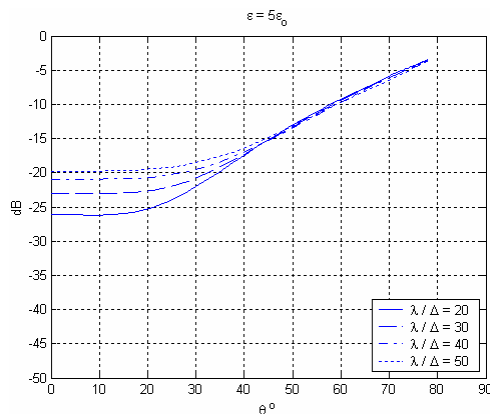
**Figura 2.8 Ilustração do domínio computacional.  $\theta$  é o ângulo de incidência na fronteira.**



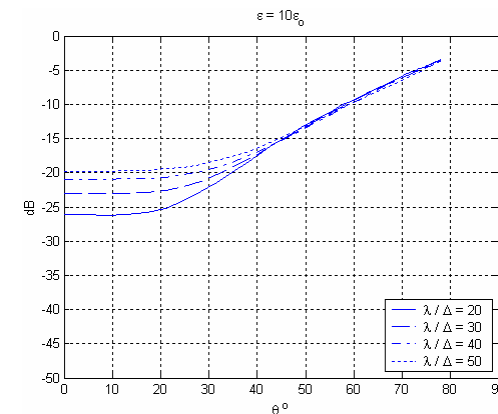
**Figura 2.9** Intensidade relativa do campo  $E_z$  para  $\epsilon = \epsilon_0$ .



**Figura 2.10** Intensidade relativa do campo  $E_z$  para  $\epsilon = 2\epsilon_0$ .



**Figura 2.11** Intensidade relativa do campo  $E_z$  para  $\epsilon = 5\epsilon_0$ .

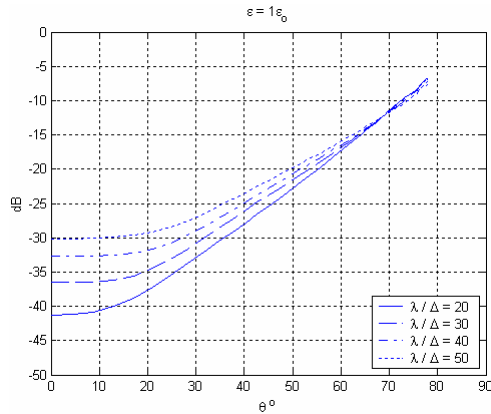


**Figura 2.12** Intensidade relativa do campo  $E_z$  para  $\epsilon = 10\epsilon_0$ .

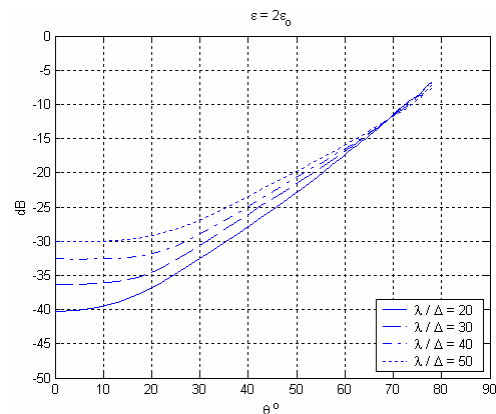
Dos gráficos apresentados, nota-se que a atenuação resultante para esse tipo de fronteira é praticamente independente da constante dielétrica do meio. Porém, ela é altamente sensível ao ângulo de incidência da onda.

### Segunda Aproximação de Mur

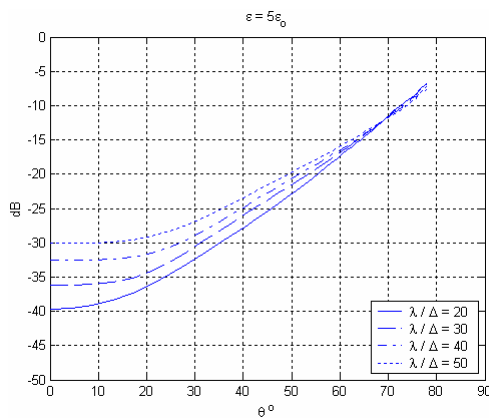
As Figuras 2.13 – 2.16 apresentam a intensidade relativa do campo elétrico de uma onda refletida pela fronteira de absorção, para a segunda aproximação de Mur de forma análoga ao que foi feito para a primeira.



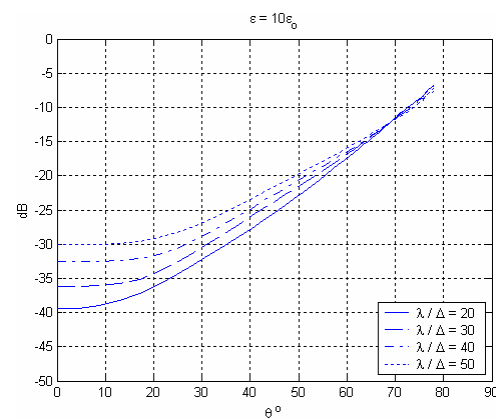
**Figura 2.13** Intensidade relativa do campo  $E_z$  para  $\epsilon = \epsilon_0$ .



**Figura 2.14** Intensidade relativa do campo  $E_z$  para  $\epsilon = 2\epsilon_0$ .



**Figura 2.15** Intensidade relativa do campo  $E_z$  para  $\epsilon = 5\epsilon_0$ .



**Figura 2.16** Intensidade relativa do campo  $E_z$  para  $\epsilon = 10\epsilon_0$ .

Assim como no caso anterior, a absorção da segunda aproximação de Mur também se mostra pouco sensível à variação da constante dielétrica do meio e altamente sensível ao ângulo de incidência da onda. Porém, a segunda aproximação de Mur apresenta em média, atenuações da ordem de 20 dB maiores que a primeira aproximação para ângulos de incidência não rasantes.

## Energia e Erro

Além do perfil de atenuação, deve-se também analisar a absorção de energia e o erro que um domínio truncado pela fronteira de absorção apresenta em relação a um domínio ideal, ou seja, sem fronteiras. Este erro indica quanto o domínio truncado ( $\Omega_T$ ) difere de um domínio ideal ( $\Omega_I$ ).

A Figura 2.17 ilustra os domínios que são utilizados na obtenção do erro e da energia absorvida pela fronteira de re-radiação. Para cada um desses domínios,  $\Omega_T$  e  $\Omega_I$  é executado uma simulação independente no algoritmo *FDTD*. Por construção, esses domínios têm seus centros espacialmente coincidentes e possuem a mesma discretização espacial e temporal. Isso faz com que cada componente do campo eletromagnético em  $\Omega_T$  seja correspondente aos componentes do campo eletromagnético em  $\Omega_I$ . O tamanho do domínio  $\Omega_I$  é escolhido de tal forma que seus limites fiquem muito afastados do domínio  $\Omega_T$ . Isso faz com que o domínio  $\Omega_T$  fique isolado durante o tempo necessário de simulação. Este isolamento é necessário para que as ondas refletidas em  $\Omega_I$  não interfiram em  $\Omega_T$ . Desta forma, o domínio  $\Omega_I$  simula exatamente um domínio infinito em relação  $\Omega_T$ , durante o tempo de simulação.

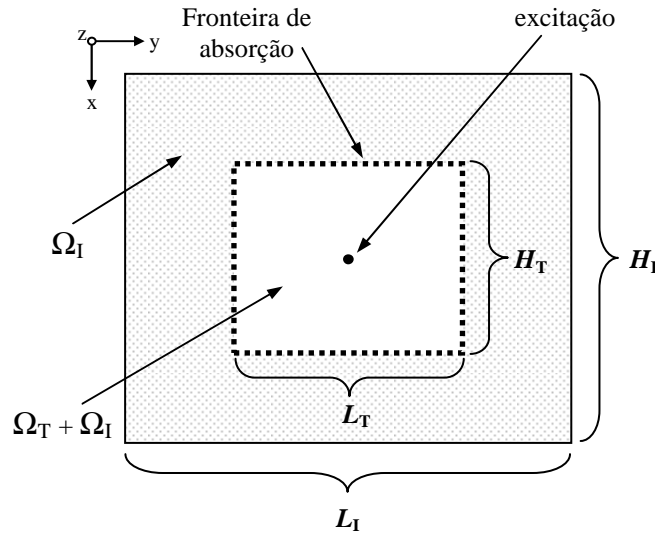


Figura 2.17 Domínio computacional para o cálculo do erro e da energia.

Para a realização destas simulações um pulso descrito por

$$E_z(n) = \begin{cases} \text{sen}\left(\frac{2\pi}{60}n\right) & ; \text{ para } n \leq 60 \\ 0 & ; \text{ para } n > 60 \end{cases} \quad (2.19)$$

é posicionado no centro dos domínios.

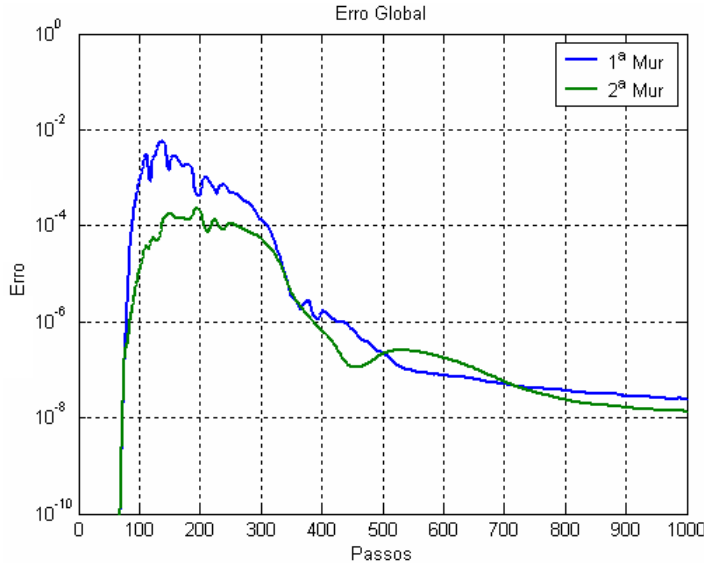
Este pulso, além de possuir transições suaves entre seus valores, não possui a componente de frequência zero. Para medir em cada instante de tempo quanto o



domínio truncado difere do domínio ideal, uma função erro é definida por Taflove [13] como sendo

$$e_{global}^n = \sum_i \sum_j \left| E_{z,T}^n|_{(i,j)} - E_{z,I}^n|_{(i,j)} \right|^2, \quad \text{para } i, j \in \Omega_T \quad (2.20)$$

Segundo Taflove, esta função erro é proporcional a densidade de potência refletida pelo truncamento do domínio ideal. Os gráficos da Figura 2.18 apresentam o erro quadrático global para a primeira aproximação de Mur (azul) e para a segunda aproximação de Mur (verde). O erro destes domínios é zero até o passo 80, pois a onda ainda não atingiu as fronteiras do domínio truncado. Após isso, nota-se que o erro causado pelo truncamento através da primeira aproximação de Mur é, em geral, maior que o erro causado pela segunda aproximação de Mur.



**Figura 2.18 Erro quadrático global do campo elétrico para a primeira e segunda aproximações de Mur.**

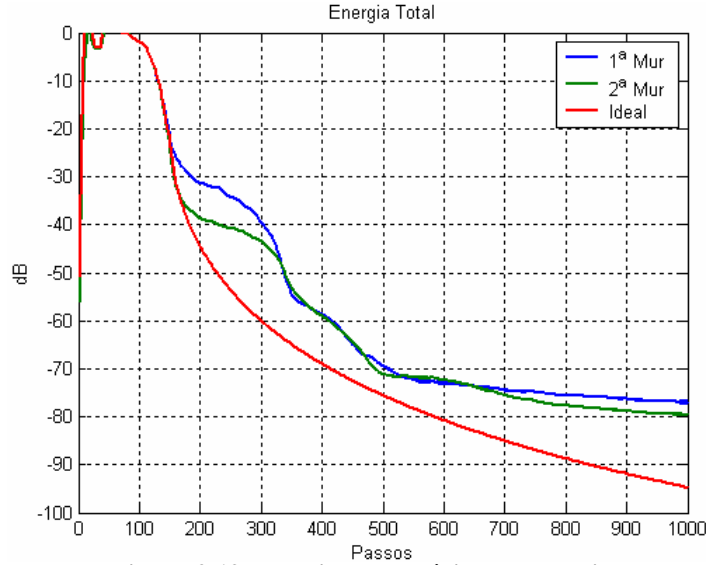
O perfil da energia absorvida representa a quantidade de energia eletromagnética que é absorvida pela fronteira. Essa quantidade de energia é obtida por.

$$E_G^n = \sum_i \sum_j \left( \frac{\epsilon_0}{2} E_z^2|_{(i,j)}^n + \frac{\mu_0}{2} H_x^2|_{(i,j)}^n + \frac{\mu_0}{2} H_y^2|_{(i,j)}^n \right) \cdot \Delta^2; \quad (2.21)$$

sendo,  $\Delta$  a discretização espacial do domínio computacional.

Os gráficos da Figura 2.19 apresentam a energia eletromagnética no domínio computacional simulado. A curva vermelha representa a energia no domínio ideal, ou

seja, sem fronteiras. A curva azul representa a energia no domínio truncado com uma fronteira de absorção implementada com a primeira aproximação de Mur. Já a curva verde representa a energia no domínio truncado com a segunda aproximação de Mur. Os gráficos estão normalizados pelo valor máximo, comum a todos.



**Figura 2.19 Energia no domínio computacional.**

Apesar da segunda aproximação de Mur apresentar melhor resultado que a primeira aproximação, para energias muito baixas, elas apresentam praticamente o mesmo resultado. Isto é explicado porque estes níveis de energia possuem alta frequência devido a erros numéricos inerentes ao método *FDTD* ; e nestas condições as aproximações possuem pouca eficiência de absorção.

## 2.4 Aproximação por Operador de Higdon

Considere uma combinação linear de frentes de onda se propagando com velocidade  $v$  em direção a uma fronteira, posicionada em  $x = 0$ , de um domínio computacional cartesiano bidimensional. Essa ondas se propagam com ângulos de incidência  $\pm\alpha_1, \dots, \pm\alpha_p$  em relação ao eixo  $x$ . O campo resultante dessa combinação linear de ondas é dado analiticamente por

$$\begin{aligned}
 W(x, y, t) &= \sum_{j=1}^p f_j(v \cdot t + \vec{k}_j \cdot \vec{r}) + \sum_{j=1}^p g_j(v \cdot t + \vec{k}'_j \cdot \vec{r}) \\
 &= f_1(v \cdot t + x \cdot \cos \alpha_1 + y \cdot \text{sen} \alpha_1) + \dots + f_p(v \cdot t + x \cdot \cos \alpha_p + y \cdot \text{sen} \alpha_p), \\
 &+ g_1(v \cdot t + x \cdot \cos \alpha_1 - y \cdot \text{sen} \alpha_1) + \dots + g_p(v \cdot t + x \cdot \cos \alpha_p - y \cdot \text{sen} \alpha_p)
 \end{aligned} \tag{2.22}$$

sendo  $-\pi/2 \leq \alpha_j \leq +\pi/2$

Para esta combinação linear, Higdon [14] propôs o seguinte operador

$$\Xi = \prod_{j=1}^p \left( (\cos \alpha_j) \frac{\partial}{\partial t} \pm v \frac{\partial}{\partial x} \right); \quad (2.23)$$

onde  $p$  é denominada ordem do operador.

Agora, ao escrever (2.23) em (2.22) é obtido

$$\Xi \cdot W = 0. \quad (2.24)$$

Em seu trabalho, Higdon demonstrou que o operador  $\Xi$  satisfaz às seguintes propriedades:

- 1) para qualquer combinação linear de ondas, como descrito por (2.22) haverá absorção total para uma fronteira posicionada em  $x = 0$ ;
- 2) para qualquer ordem  $p$  do operador, um ângulo  $\alpha_j$  pode ser escolhido para otimizar a absorção de onda;
- 3) o operador é uma representação generalizada das fronteiras de re-radiação desenvolvidas por Engquist e Majda.

O operador de Higdon mais simples é obtido para  $p = 1$ , resultando em

$$\cos \alpha_1 \cdot \frac{\partial W}{\partial t} - v \frac{\partial W}{\partial x} = 0. \quad (2.25)$$

Esta equação absorve completamente as ondas que se propagam com um ângulo  $\pm\alpha_1$  em relação ao eixo  $x$ . Se  $\alpha_1 = 0^\circ$ , este operador reduz-se à equação (2.8) proposta por Engquist e Majda. Isso ocorre quando a derivada parcial da onda incidente em relação a  $y$  é desprezível em relação às outras derivadas parciais. Esta particularização mostra que o operador de Higdon de primeira ordem é equivalente a aproximação por série de Taylor de ordem 1 para a função  $\sqrt{1 - S^2}$  como em (2.6).

Impondo-se  $p = 2$  obtém-se a equação

$$\cos \alpha_1 \cdot \cos \alpha_2 \cdot \frac{\partial^2 W}{\partial t^2} - v(\cos \alpha_1 + \cos \alpha_2) \frac{\partial^2 W}{\partial x \partial t} + v^2 \frac{\partial^2 W}{\partial x^2} = 0. \quad (2.26)$$

Esta equação representa a combinação linear de duas frentes de onda se propagando com ângulos  $\pm\alpha_1$  e  $\pm\alpha_2$  em relação ao eixo  $x$ .

Infelizmente esta equação utiliza a segunda derivada parcial em relação ao eixo  $x$ . Isto torna impossível sua implementação utilizando as diferenças finitas, pois há

necessidade de se conhecer os valores da onda em uma posição fora do domínio computacional. Porém, é possível simplificar esta equação utilizando a equação de onda

$$\frac{1}{v^2} \frac{\partial^2 W}{\partial t^2} = \frac{\partial^2 W}{\partial x^2} + \frac{\partial^2 W}{\partial y^2} \quad (2.27)$$

para eliminar a segunda derivada parcial em relação a  $x$ . Inserindo (2.27) em (2.26) e fazendo-se as devidas manipulações algébricas, obtém-se a equação

$$\frac{\partial^2 W}{\partial x \partial t} - \frac{1 + \cos \alpha_1 \cos \alpha_2}{v(\cos \alpha_1 + \cos \alpha_2)} \cdot \frac{\partial^2 W}{\partial t^2} + \frac{v}{(\cos \alpha_1 + \cos \alpha_2)} \frac{\partial^2 W}{\partial y^2} = 0. \quad (2.28)$$

Esta nova equação absorve as ondas que se propagam com ângulos  $\pm\alpha_1$  e  $\pm\alpha_2$  em relação ao eixo  $x$ . Impondo  $\alpha_1 = \alpha_2 = 0^\circ$ , esta equação se reduz à equação (2.11a), que é a aproximação por série de Talyor de ordem 2 para a função  $\sqrt{1 - S^2}$ . Novamente o operador de Higdon mostra-se como uma generalização das fronteiras de re-radiação.

Apesar da grande generalidade e facilidade na implementação das fronteiras de absorção usando o operador de Higdon, ainda existe um problema fundamental. A velocidade de onda utilizada nas equações é assumida constante em relação ao ângulo de incidência da onda. Porém no algoritmo *FDTD*, o valor numérico da velocidade de fase varia conforme o ângulo de incidência [13].

Para resolver este problema é necessário fazer com que a velocidade de onda seja uma função do ângulo de incidência. Uma vez definido um ângulo de absorção no qual a fronteira ira trabalhar, basta impor uma velocidade de onda referente à este ângulo através da equação

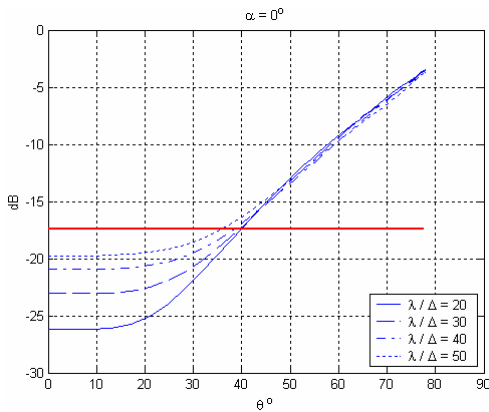
$$\left[ \prod_{j=1}^p \left( (\cos \alpha_j) \frac{\partial}{\partial t} - v_p(\alpha_j) \frac{\partial}{\partial x} \right) \right] W = 0; \quad (2.29)$$

sendo,  $v_p$  a velocidade de onda definida para o ângulo de incidência  $\alpha_j$ .

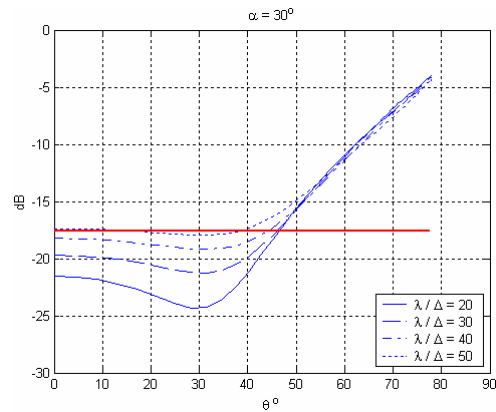
#### 2.4.1 Resultados da Aproximação por Operador de Higdon

As simulações a seguir apresentam o perfil de atenuação de uma fronteira de absorção implementada através do operado de Higdon de primeira ordem ( $p = 1$ ). Para isso, são utilizados um domínio computacional de  $\{200 \times 150\}$  células, um pulso gaussiano como fonte de excitação posicionado em  $x_o = 100$  células e  $y_o = 75$  células e uma fronteira absorção posicionada em  $y_{ABC} = 85$  células, como mostra a Figura 2.8. As

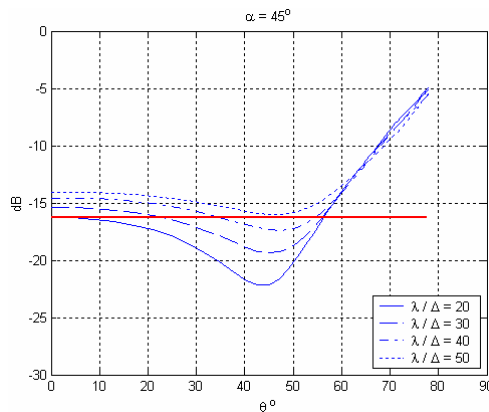
simulações apresentam o perfil de atenuação da onda em função do ângulo  $\alpha_j$ , equação (2.23).



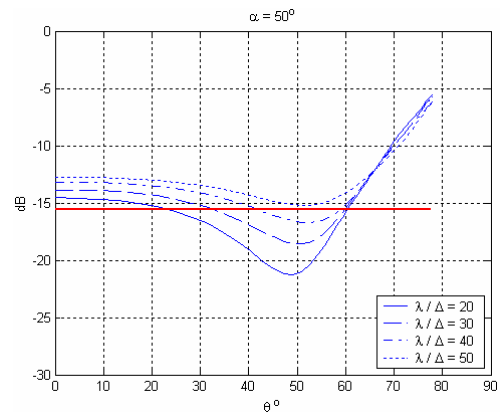
**Figura 2.20** Intensidade relativa do campo elétrico para  $\alpha = 0^\circ$ .



**Figura 2.21** Intensidade relativa do campo elétrico para  $\alpha = 30^\circ$ .



**Figura 2.22** Intensidade relativa do campo elétrico para  $\alpha = 45^\circ$ .



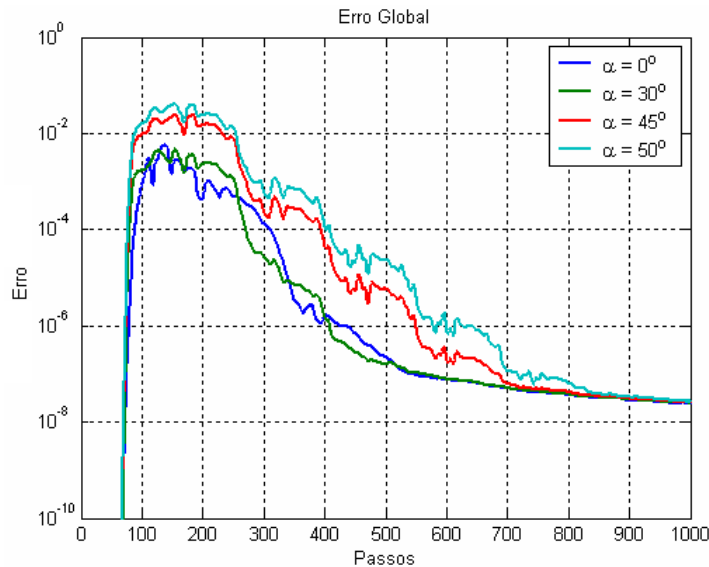
**Figura 2.23** Intensidade relativa do campo elétrico para  $\alpha = 50^\circ$ .

As Figuras 2.20 – 2.23 apresentam o perfil de atenuação de uma fronteira de absorção utilizando o operador de Higdon de primeira ordem. As curvas em azul foram levantadas para as os comprimentos de onda normalizados  $\lambda/\Delta = 20$ ,  $\lambda/\Delta = 30$ ,  $\lambda/\Delta = 40$  e  $\lambda/\Delta = 50$ . A linha horizontal representa o valor médio da atenuação para o comprimento de onda  $\lambda/\Delta = 20$ . O ângulo  $\alpha$  pode ser escolhido para termos uma máxima eficiência de absorção para cada tipo de problema.

## Energia e Erro

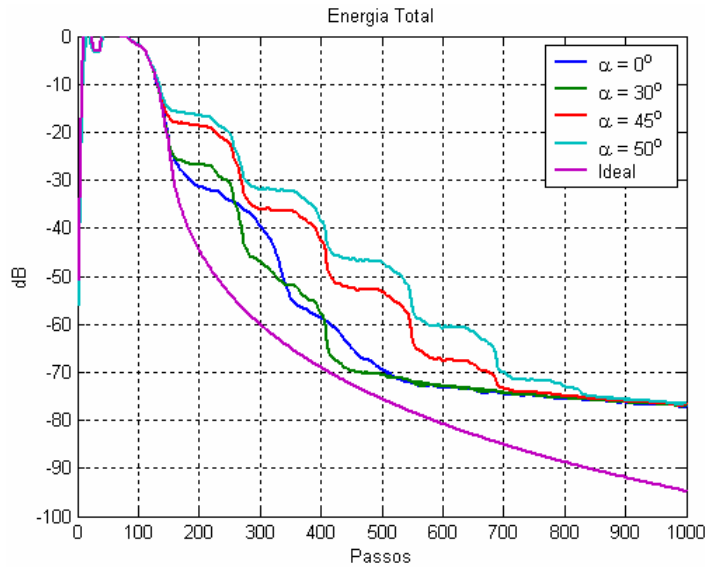
Analogamente como realizado nas simulações anteriores, nas aproximações de Higdon também é útil analisar o comportamento da absorção da energia eletromagnética e o erro global que um domínio truncado apresenta em relação a um domínio ideal.

Os gráficos da Figura 2.24 apresentam o erro quadrático global para os seguintes ângulos  $\alpha = 0^\circ$ ,  $\alpha = 30^\circ$ ,  $\alpha = 45^\circ$  e  $\alpha = 50^\circ$ . O erro deste domínio é zero até o passo 80, pois a onda ainda não atingiu os limites do domínio truncado. Nota-se que o valor do ângulo  $\alpha$  resulta em um erro diferente no domínio truncado, sendo os maiores erros pertencentes aos maiores ângulos.



**Figura 2.24 Erro global do campo elétrico para as aproximações de Higdon.**

A Figura 2.25 apresenta a energia eletromagnética presente no domínio computacional para os ângulos  $\alpha = 0^\circ$ ,  $\alpha = 30^\circ$ ,  $\alpha = 45^\circ$  e  $\alpha = 50^\circ$ . Como anteriormente, nota-se que, em geral, os melhores ângulos para a absorção de ondas estão entre os ângulos de  $\alpha = 0^\circ$  a  $\alpha = 30^\circ$ .



**Figura 2.25** Energia no domínio computacional para os ângulos  $\alpha$  simulados anteriormente.

## 2.5 Conclusão

As fronteiras de re-radiação foram as primeiras técnicas de absorção que deram crédito ao método *FDTD* [13], pois antes não era possível sua aplicação a regiões abertas. Essas técnicas se mostraram bastante simples de serem implementadas e ainda hoje são bastante utilizadas em *softwares* comerciais baseados no método *FDTD*. As técnicas mais utilizadas nesses *softwares* são as aproximações de Mur. Porém a segunda aproximação de Mur requer um cuidado especial nos cantos do domínio e nas interfaces entre os diferentes meios, como mencionado anteriormente.

Outra técnica que pode ser utilizada é o operador de Higdon. Em seu trabalho, Higdon apresentou um operador que é uma generalização do operador desenvolvido por Engquist e Majda. Com este operador é possível projetar uma fronteira de absorção para maximizar a atenuação da onda para cada tipo de problema.

Apesar desses tipos de fronteiras serem bastante utilizados, eles apresentam limitações na absorção das ondas. A absorção das ondas nesses casos atingem no máximo 20dB para a primeira aproximação de Mur e 40dB para a segunda aproximação de Mur. Com os operadores de Higdon, atingem-se aproximadamente esses mesmos valores. Operadores com ordens maiores podem ser implementados, porém, além de aumentar a carga computacional, eles não apresentam uma melhora significativa na absorção das ondas, como apresentado em [3] e [14].

# 3 Camadas Perfeitamente Casadas

## 3.1 Introdução

As medições realizadas na prática para determinar os parâmetros de antenas, de objetos iluminados por radares e outras são feitas dentro de câmaras anecóicas, ou seja, câmaras construídas com paredes de materiais absorvedores. Essas câmaras são construídas de forma a promover uma atenuação da ordem de 70dB nas ondas incidentes em suas paredes.

Fronteiras absorvedoras baseadas no princípio de funcionamento de câmaras anecóicas podem ser implementadas no método *FDTD* para truncamento de domínios computacionais. A técnica utilizada é denominada de Camadas Perfeitamente Casadas. Essa técnica, que em inglês é conhecida “Perfectly Matched Layer” (*PML*), foi desenvolvida por Berenger [4] e mais tarde, aperfeiçoada por Gedney [5]. Ela baseia-se no modelamento de materiais absorvedores com propriedades não refletoras que são usados nas fronteiras do domínio computacional.

É necessário impor que a impedância do material das fronteiras seja igual à impedância do meio com o qual está em contato. Além disso, o material deve apresentar perdas elétricas e magnéticas para que as ondas incidentes sejam absorvidas. A seguir são apresentados os equacionamentos propostos por Berenger e por Gedney para a modelagem desse tipo de material. Resultados e simulações baseados nesse equacionamento são também apresentados neste capítulo.

## 3.2 Equacionamento de Berenger

Na técnica *PML* proposta por Berenger os componentes dos campos elétrico e magnético no sentido de propagação da onda são tratados separadamente no meio absorvedor, de forma que a energia associada a cada um deles seja dissipada individualmente. Um meio com essas propriedades físicas não existe, porém do ponto de vista computacional ele é perfeitamente implementável. O procedimento apresentado a seguir mostra o equacionamento proposto para a separação desses campos no modo de propagação *TE* (Transversal Elétrico) em um domínio bidimensional. Para este modo as equações de Maxwell ficam reduzidas à seguinte forma:



$$\varepsilon_0 \frac{\partial E_x}{\partial t} + \sigma E_x = \frac{\partial H_z}{\partial y}, \quad (3.1)$$

$$\varepsilon_0 \frac{\partial E_y}{\partial t} + \sigma E_y = -\frac{\partial H_z}{\partial x}, \quad (3.2)$$

$$\mu_0 \frac{\partial H_z}{\partial t} + \rho^* H_z = \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x}, \quad (3.3)$$

sendo,  $\varepsilon_0$  a permissividade do vácuo,  $\mu_0$  a permeabilidade do vácuo,  $\sigma$  a condutividade elétrica do meio e  $\rho^*$  a resistividade magnética do meio.

Tomando-se o vácuo como o meio no qual os campos serão determinados e impondo-se

$$\frac{\sigma}{\varepsilon_0} = \frac{\rho^*}{\mu_0}, \quad (3.4)$$

a impedância intrínseca do meio será igual a impedância intrínseca do vácuo. Nesta condição não ocorrerão reflexões na interface *PML* – vácuo.

Como no modo *TE* o componente  $H_z$  se propaga nos sentidos dos eixos  $x$  e  $y$ , Berenger dividiu-o em dois outros componentes  $H_{zx}$  e  $H_{zy}$ . Com isso, as equações de Maxwell ficam reescritas na forma

$$\varepsilon_0 \frac{\partial E_x}{\partial t} + \sigma_y E_x = \frac{\partial (H_{zx} + H_{zy})}{\partial y}; \quad (3.5)$$

$$\varepsilon_0 \frac{\partial E_y}{\partial t} + \sigma_x E_y = -\frac{\partial (H_{zx} + H_{zy})}{\partial x}; \quad (3.6)$$

$$\mu_0 \frac{\partial H_{zx}}{\partial t} + \rho_x^* H_{zx} = -\frac{\partial E_y}{\partial x}; \quad (3.7)$$

$$\mu_0 \frac{\partial H_{zy}}{\partial t} + \rho_y^* H_{zy} = \frac{\partial E_x}{\partial y}; \quad (3.8)$$

sendo,  $\sigma_x$  e  $\sigma_y$  as condutividades elétricas nas direções dos eixos  $x$  e  $y$  e  $\rho_x^*$  e  $\rho_y^*$  as resistividades magnéticas nas direções dos eixos  $x$  e  $y$ .

Isto introduziu um novo grau de liberdade nas equações de Maxwell, pois permite definirmos de forma individual as impedâncias e as perdas nas direções dos eixos  $x$  e  $y$ .

Seguindo a notação de Berenger um meio *PML* é descrito pela quadra  $(\sigma_x, \rho_x^*, \sigma_y, \rho_y^*)$ . Impondo-se  $(0, 0, 0, 0)$ , as equações (3.5 – 3.8) reduzem-se as

equações de Maxwell no vácuo. Impondo-se  $(\sigma, 0, \sigma, 0)$ , com  $\sigma \neq 0$ , as equações descrevem um meio com condutividade elétrica. Finalmente impondo-se  $(\sigma, \rho^*, \sigma, \rho^*)$  as equações descrevem o comportamento de um meio absorvedor descrito por (3.1 – 3.3).

Outras possibilidades podem ser definidas através desses parâmetros. A condição  $(\sigma_x, \rho_x^*, 0, 0)$  faz com que o meio seja capaz de absorver uma frente de onda se propagando na direção do eixo  $x$ , mas não uma frente de onda se propagando na direção do eixo  $y$ . Agora, Impondo-se a condição  $(0, 0, \sigma_y, \rho_y^*)$  teremos o efeito contrário.

Assim um meio *PML* pode ser construído de forma independente nas fronteiras definidas por  $x = x_{\min}$ ,  $y = y_{\min}$ ,  $x = x_{\max}$  e  $y = y_{\max}$ . Isto é feito definindo-se apenas os parâmetros de cada fronteira,  $(\sigma_x, \rho_x^*, 0, 0)$  para a fronteira posicionada no eixo  $x$  e  $(0, 0, \sigma_y, \rho_y^*)$  para a fronteira posicionadas no eixo  $y$ .

### 3.2.1 Propagação em um Meio PML

Considerando-se uma onda plana na qual a direção do campo elétrico forma um ângulo  $\phi$  com o eixo  $y$ , como mostrado na Figura 3.1, os componentes dos campos se propagam em um meio *PML* podem ser expressos por

$$E_x = -E_0 \sin(\phi) \exp(j\omega \cdot (t - \alpha x - \beta y)); \quad (3.9)$$

$$E_y = -E_0 \cos(\phi) \exp(j\omega \cdot (t - \alpha x - \beta y)); \quad (3.10)$$

$$H_{zx} = -H_{zx0} \exp(j\omega \cdot (t - \alpha x - \beta y)); \quad (3.11)$$

$$H_{zy} = -H_{zy0} \exp(j\omega \cdot (t - \alpha x - \beta y)); \quad (3.12)$$

sendo,  $\omega$  a frequência angular da onda,  $t$  o tempo,  $\alpha$  e  $\beta$  as frequências espaciais da onda,  $E_0$  a intensidade do campo elétrico e  $H_{zx0}$  e  $H_{zy0}$  as intensidades dos componentes  $H_{zx}$  e  $H_{zy}$  do campo magnético.

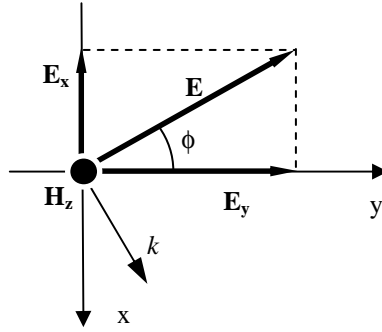


Figura 3.1 Propagação de uma onda no modo TE.

Das equações (3.9 – 3.12) é necessário determinar  $\alpha$ ,  $\beta$ ,  $H_{zx0}$  e  $H_{zy0}$ , visto que  $E_0$  é dado. Substituindo-se (3.9 – 3.12) em (3.5 – 3.8), obtém-se as equações

$$\epsilon_0 E_0 \text{sen}(\phi) - j \frac{\sigma_y}{\omega} E_0 \text{sen}(\phi) = \beta (H_{zx0} + H_{zy0}); \quad (3.13)$$

$$\epsilon_0 E_0 \cos(\phi) - j \frac{\sigma_x}{\omega} E_0 \cos(\phi) = \alpha (H_{zx0} + H_{zy0}); \quad (3.14)$$

$$\mu_0 H_{zx0} - j \frac{\rho_x^*}{\omega} H_{zx0} = \alpha E_0 \cos(\phi); \quad (3.15)$$

$$\mu_0 H_{zy0} - j \frac{\rho_y^*}{\omega} H_{zy0} = \beta E_0 \text{sen}(\phi); \quad (3.16)$$

Substituindo-se  $H_{zx0}$  e  $H_{zy0}$  das equações (3.15) e (3.16), respectivamente, nas equações (3.13) e (3.14), obtém-se as seguintes relações

$$\mu_0 \epsilon_0 \left( 1 - j \frac{\sigma_x}{\omega \epsilon_0} \right) \cos(\phi) = \alpha \left( \frac{\alpha \cos(\phi)}{1 - j \rho_x^* / \mu_0 \omega} + \frac{\beta \text{sen}(\phi)}{1 - j \rho_y^* / \mu_0 \omega} \right); \quad (3.17)$$

$$\mu_0 \epsilon_0 \left( 1 - j \frac{\sigma_y}{\omega \epsilon_0} \right) \text{sen}(\phi) = \beta \left( \frac{\alpha \cos(\phi)}{1 - j \rho_x^* / \mu_0 \omega} + \frac{\beta \text{sen}(\phi)}{1 - j \rho_y^* / \mu_0 \omega} \right); \quad (3.18)$$

Resolvendo-se o sistema formado por (3.17) e (3.18), obtém-se

$$\alpha = \frac{\sqrt{\mu_0 \epsilon_0}}{G} \left( 1 - j \frac{\sigma_x}{\omega \epsilon_0} \right) \cos(\phi); \quad (3.19)$$

$$\beta = \frac{\sqrt{\mu_0 \epsilon_0}}{G} \left( 1 - j \frac{\sigma_y}{\omega \epsilon_0} \right) \text{sen}(\phi); \quad (3.20)$$

sendo:

$$G = \sqrt{w_x \cos^2(\phi) + w_y \text{sen}^2(\phi)}; \quad (3.21)$$

$$w_x = \frac{1 - j\sigma_x/\varepsilon_0\omega}{1 - j\rho_x^*/\mu_0\omega}; \quad (3.22)$$

$$w_y = \frac{1 - j\sigma_y/\varepsilon_0\omega}{1 - j\rho_y^*/\mu_0\omega}; \quad (3.23)$$

Sendo  $\psi$  um componente qualquer dos campos,  $\psi_0$  a sua intensidade e  $c$  a velocidade da luz no vácuo, de (3.9 – 3.12) e (3.21 – 3.23) obtém-se

$$\psi = \psi_0 \exp\left(j\omega\left(t - \frac{x\cos(\phi) + y\sin(\phi)}{cG}\right)\right) \cdot \exp\left(-\frac{\sigma_x\cos(\phi)}{\varepsilon_0cG}x\right) \cdot \exp\left(-\frac{\sigma_y\sin(\phi)}{\varepsilon_0cG}y\right); \quad (3.24)$$

Os valores de  $H_{zx0}$  e  $H_{zy0}$  podem ser determinados pela substituição de  $\alpha$  e  $\beta$  nas equações (3.15 – 3.16)

$$H_{zx0} = E_0 \sqrt{\frac{\varepsilon_0}{\mu_0}} \cdot \frac{1}{G} \cdot w_x \cos^2(\phi); \quad (3.25)$$

$$H_{zy0} = E_0 \sqrt{\frac{\varepsilon_0}{\mu_0}} \cdot \frac{1}{G} \cdot w_y \sin^2(\phi); \quad (3.26)$$

O campo  $H_0$  é obtido pela soma de  $H_{zx0}$  e  $H_{zy0}$ , resultando resultado em

$$H_0 = E_0 \sqrt{\frac{\varepsilon_0}{\mu_0}} \cdot G; \quad (3.27)$$

a impedância ( $Z$ ) do meio é dado por

$$Z = \sqrt{\frac{\mu_0}{\varepsilon_0}} \cdot \frac{1}{G}; \quad (3.28)$$

Considerando-se a possibilidade de cada par  $(\sigma_x, \rho_x^*)$  e  $(\sigma_y, \rho_y^*)$  satisfazer a (3.4) então as constantes  $w_x$ ,  $w_y$  e  $G$  serão iguais a 1 para qualquer frequência. Com isso, a equação de onda (3.27) e a impedância  $Z$  (3.28) podem ser reescritas respectivamente nas formas

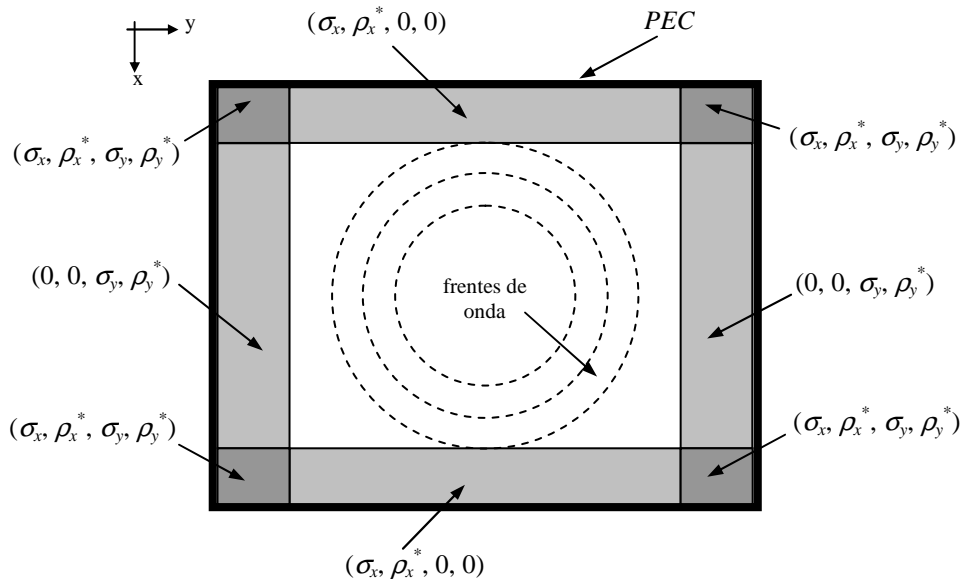
$$\psi = \psi_0 \exp\left(j\omega\left(t - \frac{x\cos(\phi) + y\sin(\phi)}{c}\right)\right) \cdot \exp\left(-\frac{\sigma_x\cos(\phi)}{\varepsilon_0c}x\right) \cdot \exp\left(-\frac{\sigma_y\sin(\phi)}{\varepsilon_0c}y\right); \quad (3.29)$$

$$Z = \sqrt{\frac{\mu_0}{\varepsilon_0}}; \quad (3.30)$$

A primeira exponencial em (3.29) mostra que a onda no meio *PML* se propaga com velocidade igual a da luz no vácuo. As duas exponenciais restantes indicam um decaimento da intensidade da onda na direção dos eixos  $x$  e  $y$ . A operação (3.30) mostra

que a impedância do meio *PML* é igual a do vácuo para qualquer ângulo de propagação, indicando portanto um casamento de impedância perfeita entre esses dois meios.

A implementação em duas dimensões para o modo *TE* de propagação proposto por Berenger está ilustrada na Figura 3.2. Nas posições  $x = x_{\min}$  e  $x = x_{\max}$ , a *PML* tem o par  $(\sigma_x, \rho_x^*)$  casado com relação a condição (3.4), nestas posições  $(\sigma_y, \rho_y^*) = (0, 0)$ . Já nas posições  $y = y_{\min}$  e  $y = y_{\max}$ , a *PML* tem o par  $(\sigma_y, \rho_y^*)$  casado com relação a condição (3.4), nestas posições  $(\sigma_x, \rho_x^*) = (0, 0)$ . Nos quatro cantos do domínio existe uma sobreposição das camadas *PML*, portanto os pares  $(\sigma_x, \rho_x^*)$  e  $(\sigma_y, \rho_y^*)$  devem estar todos casados de acordo com a condição (3.4).



**Figura 3.2** Estrutura de implementação de um *PML* em duas dimensões para o modo *TE*.

Berenger também propôs que a condutividade das camadas *PML* deva ser variada de forma suave, como mostra a equação

$$\sigma(r) = \sigma_{\max} \left( \frac{r}{\delta} \right)^n ; \quad (3.31)$$

sendo,  $r$  a posição dentro da *PML* a partir de uma posição inicial,  $\delta$  a espessura da camada *PML*,  $\sigma_{\max}$  o valor da condutividade máxima e  $n$  um expoente polinomial que indica quanto a condutividade cresce em função da posição.

### 3.3 Equacionamento de Gedney

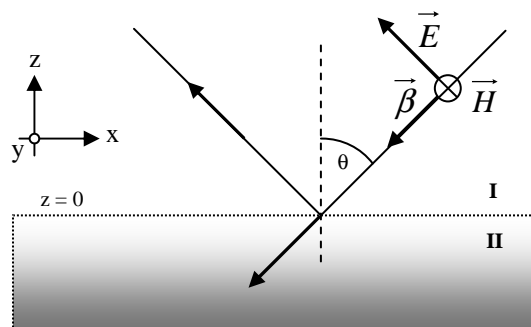
As equações propostas por Berenger baseiam-se na divisão dos componentes dos campos elétrico e magnético em um sistema de coordenadas cartesianas. Essa divisão faz com que as camadas *PML* sejam governados por equações não “Maxwellianas”, ou seja, as equações que descrevem a *PML* não estão na forma padrão proposta por Maxwell. Além do mais, essa divisão faz com que o número de operações matemáticas seja o dobro em relação ao algoritmo *FDTD* convencional. No trabalho de Gedney [5] é proposto um novo tipo de equacionamento para descrever as camadas *PML*. Neste equacionamento, a *PML* é descrita como um meio uniaxial dispersivo e perfeitamente casado com um meio isotrópico. Além do mais, a *PML* proposta por Gedney possui as seguintes características:

- 1) é descrita pelas equações de Maxwell.
- 2) é computacionalmente mais eficiente que a proposta por Berenger.
- 3) pode facilmente ser adaptada para técnicas de Grades Não – Uniformes.

Para descrever essa *PML*, Gedney propôs o seguinte procedimento: Um plano de onda descrito por

$$\vec{H}^{inc} = \vec{H}_0 \cdot \exp(j\omega t - j\beta_x^i x - j\beta_z^i z); \quad (3.32)$$

sendo o índice *i* a indicação de um meio isotrópico, com vetor de onda dado por  $\beta$ , se propaga em um meio isotrópico ( **I** ) e incide em um meio uniaxialmente anisotrópico ( **II** ) em  $z = 0$ , como mostrado na Figura 3.3.



**Figura 3.3** Onda incidindo em uma camada *PML*.

Os campos existentes no meio anisotrópico ( **II** ) devem satisfazer as equações de Maxwell e, no plano de propagação da frente onda, elas são escritas na forma

$$\vec{\beta}^a \times \vec{E} = \omega \mu \overline{\mu} \vec{H}; \quad (3.33)$$

$$\vec{\beta}^a \times \vec{H} = -\omega \varepsilon \overline{\varepsilon} \vec{E}; \quad (3.34)$$

$$\vec{\beta}^a = \beta_x^i \vec{x} + \beta_z^a \vec{z}; \quad (3.35)$$

sendo,  $\varepsilon$  e  $\mu$  respectivamente a permissividade e a permeabilidade do meio isotrópico e o índice  $a$  a indicação de um meio anisotrópico. Os tensores  $\overline{\varepsilon}$  e  $\overline{\mu}$  para um meio rotacionalmente simétrico em torno do eixo  $z$  são dados por

$$\overline{\varepsilon} = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & b \end{bmatrix}; \quad (3.36)$$

$$\overline{\mu} = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & d \end{bmatrix}. \quad (3.37)$$

Através de (3.33) e (3.34) obtém-se a seguinte equação de onda

$$\vec{\beta}^a \times \left( \overline{\varepsilon}^{-1} \cdot \left( \vec{\beta}^a \times \vec{H} \right) \right) + k^2 \overline{\mu} \vec{H} = 0; \quad (3.38)$$

sendo,  $k^2 = \omega^2 \mu \varepsilon$ .

Reescrevendo (3.38) na forma matricial, obtém-se

$$\begin{bmatrix} k^2 c - a^{-1} (\beta_z^a)^2 & 0 & a^{-1} \beta_x^i \beta_z^a \\ 0 & k^2 c - a^{-1} (\beta_z^a)^2 - b^{-1} (\beta_x^i)^2 & 0 \\ a^{-1} \beta_x^i \beta_z^a & 0 & k^2 d - a^{-1} (\beta_x^i)^2 \end{bmatrix} \cdot \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \quad (3.39)$$

A relação de dispersão para um meio uniaxial é obtido através do determinante da matriz em (3.39). Igualando-se esse determinante a zero, obtém-se quatro autovalores, que podem ser decompostos nos dois modos de propagação, um modo  $TE_y$  e outro  $TM_y$ , para as ondas que se propagam no sentido positivo de  $z$  e para as ondas que se propagam no sentido negativo de  $z$ . Estes modos estão expressos por

$$\text{modo } TE_y: \quad k^2 c - a^{-1} (\beta_z^a)^2 - b^{-1} (\beta_x^i)^2 = 0; \quad (3.40)$$

$$\text{modo } TM_y: \quad k^2 a - c^{-1} (\beta_z^a)^2 - d^{-1} (\beta_x^i)^2 = 0; \quad (3.41)$$

O coeficiente de reflexão na interface dos dois meios é determinado da seguinte forma: assumindo-se o modo  $TE_y$ , para o meio isotrópico ( **I** ) os campos são expressos como uma superposição dos campos incidente e refletido. As equações a seguir indicam respectivamente o campo magnético total e o campo elétrico total no meio ( **I** ).

$$\vec{H}^{(I)} = (1 + \Gamma e^{2j\beta_z^i z}) H_o \exp(-j\beta_x^i x - j\beta_z^i z) \hat{y}; \quad (3.42)$$

$$\vec{E}^{(I)} = \left[ \frac{\beta_z^i}{\omega\epsilon} (1 - \Gamma \exp(2j\beta_z^i z)) \hat{x} - \frac{\beta_x^i}{\omega\epsilon} (1 + \Gamma \exp(2j\beta_z^i z)) \hat{z} \right] H_o \exp(-j\beta_x^i x - j\beta_z^i z); \quad (3.43)$$

O campo transmitido para o meio anisotrópico ( **II** ) também terá o modo de propagação  $TE_y$ , e será governado pelas equações

$$\vec{H}^{(II)} = \tau H_o \exp(-j\beta_x^i x - j\beta_z^a z) \hat{y}; \quad (3.44)$$

$$\vec{E}^{(II)} = \left[ \frac{\beta_z^i a^{-1}}{\omega\epsilon} \hat{x} - \frac{\beta_x^i b^{-1}}{\omega\epsilon} \hat{z} \right] \tau H_o \exp(-j\beta_x^i x - j\beta_z^a z); \quad (3.45)$$

sendo,  $\tau$  o coeficiente de transmissão e  $\Gamma$  o coeficiente de reflexão.

Na fronteira entre os dois meios a condição de contorno

$$\begin{cases} \vec{E}_{t1} - \vec{E}_{t2} = 0 \\ \vec{H}_{t1} - \vec{H}_{t2} = 0 \end{cases}; \quad (3.46)$$

é imposta, com isso são obtidos os coeficientes de reflexão e de transmissão, dados respectivamente por

$$\Gamma = \frac{\beta_z^i - \beta_z^a a^{-1}}{\beta_z^i + \beta_z^a a^{-1}}; \quad (3.47)$$

$$\tau = \frac{2\beta_z^i}{\beta_z^i + \beta_z^a a^{-1}}; \quad (3.48)$$

Por (3.47) é observado que se a condição  $\beta_z^i = \beta_z^a a^{-1}$  for imposta, então não haverá reflexão na interface entre os meios ( **I** ) e ( **II** ). Impondo essa condição em (3.40), obtém-se a relação

$$(\beta_z^i)^2 = k^2 c a^{-1} - b^{-1} a^{-1} (\beta_x^i)^2; \quad (3.49)$$

sendo  $k^2 = (\beta_z^i)^2 + (\beta_x^i)^2$ .

A equação (3.49) é válida somente se  $c = a$  e  $b = a^{-1}$ . Aplicando-se o mesmo procedimento para o modo de propagação  $TM_y$ , equação (3.41), obtém-se  $c = a$  e



$d = c^{-1}$ . Portanto, se a condição  $a = c = b^{-1} = d^{-1}$  for respeitada, então um plano de onda será totalmente transmitido para o meio ( **II** ) independente do ângulo de incidência, da polarização e frequência da onda incidente.

### 3.3.1 Implementação em FDTD

Para aplicações em *FDTD*, é extremamente importante que o meio *PML* tenha uma alta capacidade de atenuação. Desse modo quando o meio *PML* termina no limite do domínio computacional, as reflexões que ocorrerão devido a espessura finita do *PML*, ficam pequenas. Contudo, se esse meio possuir alta atenuação, essas reflexões podem ser muito reduzidas. Para um meio uniaxial, uma escolha óbvia para os parâmetros constituintes dos tensores  $\overset{=}{\varepsilon}$  e  $\overset{=}{\mu}$  é  $a = 1 + \sigma / j\omega\varepsilon_0$ . Com isso os tensores ficam na forma

$$\overset{=}{\varepsilon} = \overset{=}{\mu} = \begin{bmatrix} 1 + \frac{\sigma}{j\omega\varepsilon_0} & 0 & 0 \\ 0 & 1 + \frac{\sigma}{j\omega\varepsilon_0} & 0 \\ 0 & 0 & \frac{1}{1 + \frac{\sigma}{j\omega\varepsilon_0}} \end{bmatrix}. \quad (3.50)$$

Para a implementação em *FDTD* é necessário escrever as equações de Maxwell no domínio do tempo na forma discretizada. Porém, as deduções feitas até agora usaram as equações de Maxwell no domínio da frequência. Para passar estas equações do domínio da frequência para o domínio do tempo Gedney propôs, usando a lei de Ampère como exemplo, o seguinte procedimento

$$\begin{bmatrix} \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \\ \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \end{bmatrix} = j\omega\varepsilon_0\varepsilon_r \cdot \begin{bmatrix} 1 + \frac{\sigma}{j\omega\varepsilon_0} & 0 & 0 \\ 0 & 1 + \frac{\sigma}{j\omega\varepsilon_0} & 0 \\ 0 & 0 & \frac{1}{1 + \frac{\sigma}{j\omega\varepsilon_0}} \end{bmatrix} \cdot \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}. \quad (3.51)$$

Para implementar esta técnica em *FDTD*, deve-se aplicar o método das diferenças centrais na equação matricial (3.51). As duas primeiras equações que correspondente respectivamente a  $E_x$  e  $E_y$ , são idênticas ao método *FDTD* original, bastando-se apenas usar a relação  $j\omega \rightarrow \partial/\partial t$ .

$$E_x \Big|_{i+1/2,j,k}^{n+1/2} = \frac{2\varepsilon_0 - \Delta t\sigma}{2\varepsilon_0 + \Delta t\sigma} E_x \Big|_{i+1/2,j,k}^{n-1/2} + \frac{2\Delta t/\varepsilon_r}{2\varepsilon_0 + \Delta t\sigma} \cdot \left( \frac{H_z \Big|_{i+1/2,j+1/2,k}^n - H_z \Big|_{i+1/2,j-1/2,k}^n}{\Delta y} - \frac{H_y \Big|_{i+1/2,j,k+1/2}^n - H_y \Big|_{i+1/2,j,k-1/2}^n}{\Delta z} \right); \quad (3.52)$$

A última equação de (3.51), componente  $E_z$ , deve ser tratada de forma especial pois existe uma dependência não linear com a frequência. Observando (3.50) verifica-se que a posição  $(\varepsilon_{33})$  possui um termo imaginário positivo. Isso implica uma condutividade negativa ao longo do eixo  $z$ . O componente  $E_z$  pode ser implementado de inúmeras formas. Porém, a mais simples é decompor essa equação de forma a introduzir o vetor deslocamento  $D_z$ , para o campo elétrico.

$$D_z = \frac{\varepsilon_o \varepsilon_r}{1 + \frac{\sigma}{j\omega \varepsilon_o}} E_z; \quad (3.53)$$

$$\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = \frac{\partial D_z}{\partial t}; \quad (3.54)$$

com isso, o vetor  $D_z$  é atualizado utilizando o método *FDTD* convencional. Como mostrado na equação

$$D_z \Big|_{i,j,k+1/2}^{n+1/2} = D_z \Big|_{i,j,k+1/2}^{n-1/2} + \Delta t \cdot \left( \frac{H_y \Big|_{i+1/2,j,k+1/2}^n - H_y \Big|_{i-1/2,j,k+1/2}^n}{\Delta x} - \frac{H_x \Big|_{i,j+1/2,k+1/2}^n - H_x \Big|_{i,j-1/2,k+1/2}^n}{\Delta y} \right); \quad (3.55)$$

e o campo  $E_z$  é calculado através de  $D_z$  pelo seguinte procedimento

$$\left( 1 + \frac{\sigma}{j\omega \varepsilon_o} \right) D_z = \varepsilon_o \varepsilon_r E_z \quad \rightarrow \quad j\omega D_z + \frac{\sigma}{\varepsilon_o} D_z = j\omega \varepsilon_o \varepsilon_r E_z \quad (3.56)$$

$$E_z^{n+1/2} = E_z^{n-1/2} + \frac{1}{\varepsilon_o \varepsilon_r} \left[ \left( 1 + \frac{\sigma \Delta t}{2\varepsilon_o} \right) D_z^{n+1/2} - \left( 1 - \frac{\sigma \Delta t}{2\varepsilon_o} \right) D_z^{n-1/2} \right]; \quad (3.57)$$

Nos cantos onde há a interseção entre camadas *PML*, Gedney propôs uma relação constitutiva mais geral. Com isso, os tensores  $\bar{\epsilon}$  e  $\bar{\mu}$  ficam na forma

$$\bar{\epsilon} = \bar{\mu} = \begin{bmatrix} \frac{s_y s_z}{s_x} & 0 & 0 \\ 0 & \frac{s_x s_z}{s_y} & 0 \\ 0 & 0 & \frac{s_x s_y}{s_z} \end{bmatrix}; \quad (3.58)$$

sendo,  $s_x = 1 + \frac{\sigma_x}{j\omega\epsilon_0}$ ,  $s_y = 1 + \frac{\sigma_y}{j\omega\epsilon_0}$ ,  $s_z = 1 + \frac{\sigma_z}{j\omega\epsilon_0}$ . Nestes cantos, os campos são atualizados através da mesma técnica apresentada anteriormente.

O perfil de condutividade  $\sigma$  deve estar de acordo com a posição dos componentes dos campos, ou seja, ela deve ser ajustada levando em conta a defasagem espacial desses componentes. Para um *PML* construído no eixo  $y$ , o perfil de condutividade para os componentes posicionados em  $(j)$  é dado por

$$\sigma|_j = \sigma_{\max} \cdot \left( \frac{j - j_o}{L} \right)^m; \quad (3.59)$$

e para os componentes posicionados em  $(j - 1/2)$  este perfil é dado por

$$\sigma|_{j-1/2} = \sigma_{\max} \cdot \left( \frac{(j - 1/2) - j_o}{L} \right)^m; \quad (3.60)$$

sendo,  $j_o$  e  $L$  respectivamente a posição inicial e a espessura da *PML* em células,  $m$  um expoente polinomial e  $\sigma_{\max}$  a condutividade máxima desejada para essa *PML*.

Estes parâmetros podem assumir qualquer valor, porém Gedney propôs um valor otimizado para cada tipo de problema dado por

$$\sigma_{\max} = \frac{m + 1}{150\pi \Delta \sqrt{\epsilon_r}}; \quad (3.61)$$

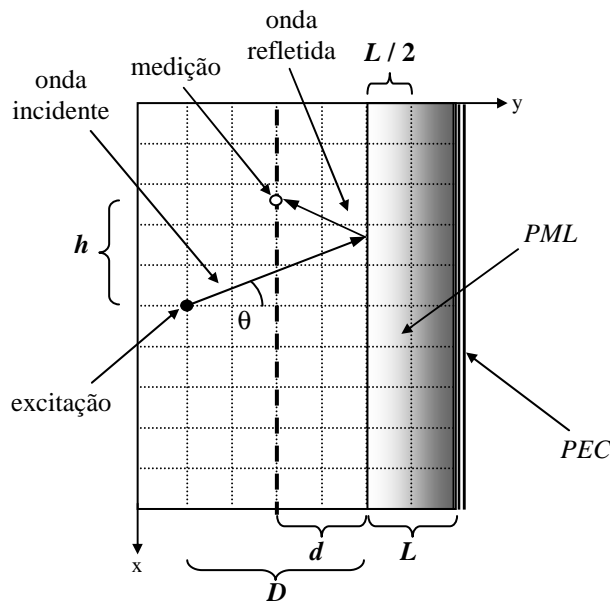
sendo,  $m$  o expoente polinomial,  $\Delta$  a discretização espacial no sentido de crescimento da condutividade e  $\epsilon_r$  a permissividade relativa do meio no qual a *PML* está inserida.

### 3.4 Resultados das Simulações

A seguir é apresentado o perfil de atenuação de uma fronteira de absorção obtido através da técnica *PML*. Para isso são utilizados um domínio computacional de

{200x150} células, um pulso gaussiano como fonte de excitação posicionado em  $x_0 = 100$  células e  $y_0 = 75$  células e o início das camadas *PML* posicionadas em  $y_{ABC} = 85$  células.

A Figura 3.4 ilustra o domínio computacional utilizado. Nesta figura,  $D$  é a distância entre o ponto de excitação e o início da *PML,  $d$  é a distância entre o início da *PML* e o ponto de medição,  $h$  é a distância entre os pontos de excitação e medição medida no eixo  $x$  e  $L$  é a espessura da camada *PML*. Para esta simulação é utilizado os parâmetros  $D = 10$  células e  $d = 3$  células.*



**Figura 3.4 Ilustração do domínio computacional.  $\theta$  é o ângulo de incidência no *PML*.**

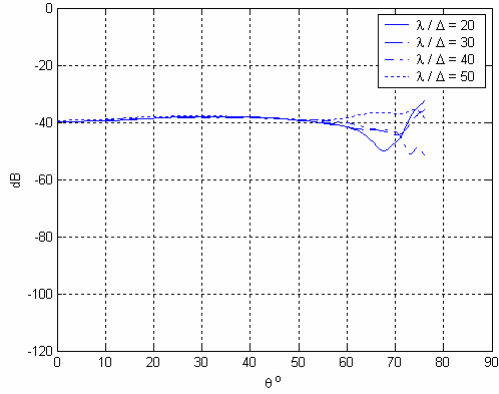
As simulações a seguir apresentam, para vários comprimentos de onda, a intensidade relativa do campo elétrico de uma onda refletida pelas camadas *PML* em relação aos campos de uma onda sem a presença da mesma.

Estas simulações foram divididas em duas partes. A primeira parte apresenta o comportamento da *PML* para dois valores do expoente polinomial  $m$ , que são  $m = 2$  e  $m = 4$ . A segunda parte apresenta o comportamento da *PML* em função da constante dielétrica do meio o qual ela está imersa.

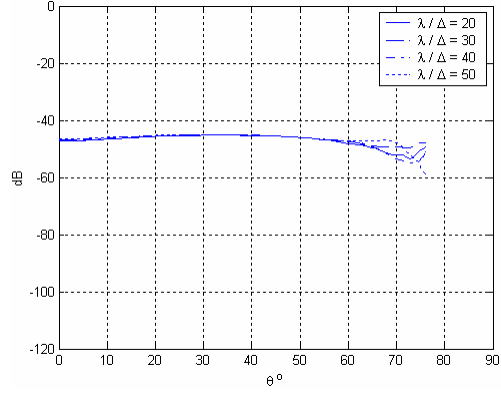
### 1ª Parte: Variação da Constante Polinomial

As Figuras 3.5 – 3.8 apresentam o comportamento da *PML* para um expoente polinomial  $m = 2$  e as Figuras 3.9 – 3.12 apresentam o comportamento da *PML* para um

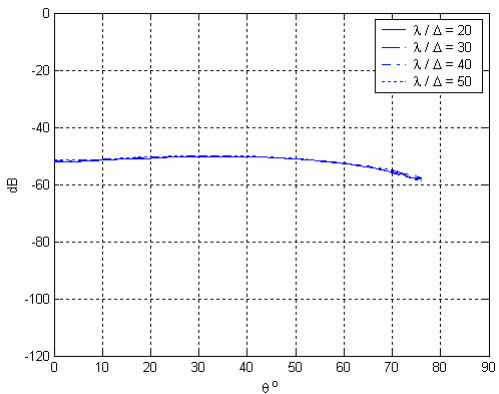
exponente polinomial  $m = 4$ . Este comportamento é levantado em função de quatro valores para a espessura da *PML*. Nestas simulações  $\varepsilon$  é mantido constante em  $\varepsilon = \varepsilon_0$  e a condutividade máxima é obtida por (3.61).



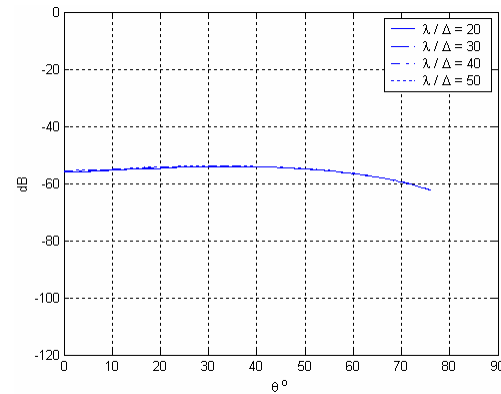
**Figura 3.5** Intensidade relativa do campo  $E_z$  para  $L = 4$  e  $m = 2$ .



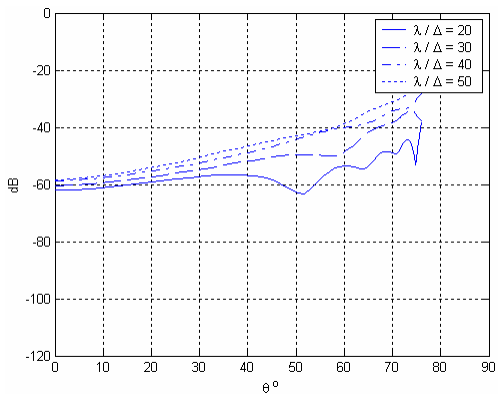
**Figura 3.6** Intensidade relativa do campo  $E_z$  para  $L = 6$  e  $m = 2$ .



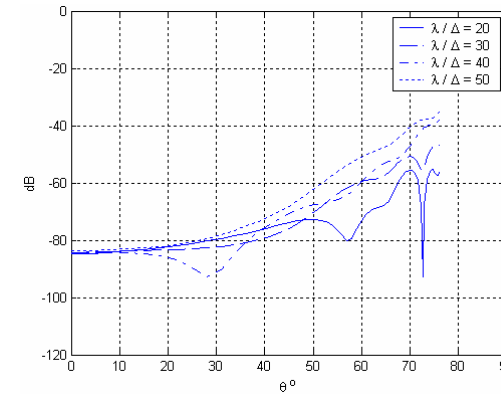
**Figura 3.7** Intensidade relativa do campo  $E_z$  para  $L = 8$  e  $m = 2$ .



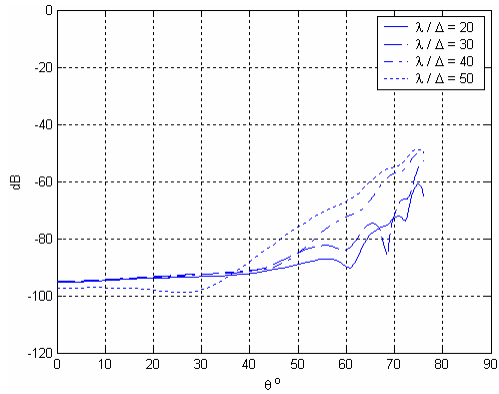
**Figura 3.8** Intensidade relativa do campo  $E_z$  para  $L = 10$  e  $m = 2$ .



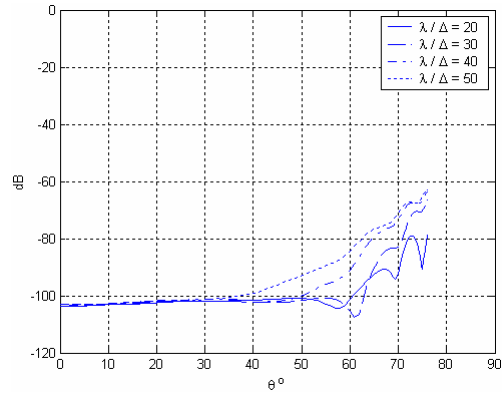
**Figura 3.9** Intensidade relativa do campo  $E_z$  para  $L = 4$  e  $m = 4$ .



**Figura 3.10** Intensidade relativa do campo  $E_z$  para  $L = 6$  e  $m = 4$ .



**Figura 3.11** Intensidade relativa do campo  $E_z$  para  $L = 8$  e  $m = 4$ .



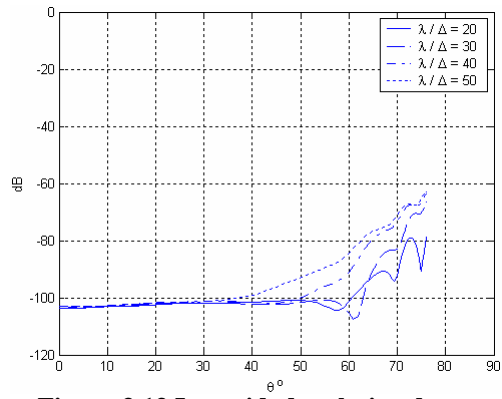
**Figura 3.12** Intensidade relativa do campo  $E_z$  para  $L = 10$  e  $m = 4$ .

Estes gráficos mostram como a atenuação de uma onda incidente é influenciada pelos parâmetros de construção da *PML*. Quanto maior é a espessura da *PML*, maior é a atenuação. Além disso, nota-se que a atenuação é maior quando a constante polinomial possui um valor elevado. Na maioria dos trabalhos realizados na literatura, este parâmetro é geralmente escolhido entre  $m = 3$  e  $m = 4$ .

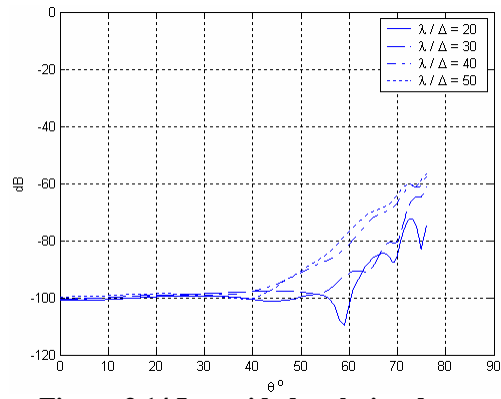
## 2ª Parte: Variação da Constante Dielétrica

Esta parte da análise apresenta o comportamento da *PML* em função da constante dielétrica do meio na qual ela está imersa. Nestas simulações a espessura da *PML* é mantida em  $L = 10$  células e o expoente polinomial é mantido em  $m = 4$ .

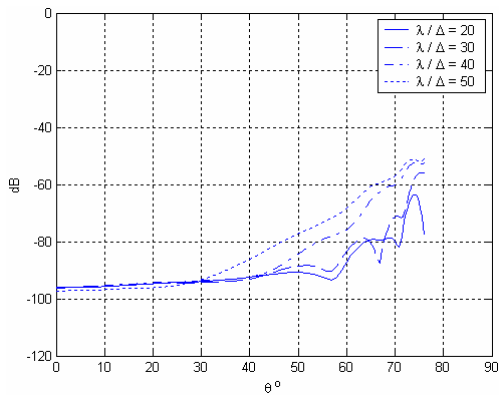
As Figuras 3.13 – 3.16 apresentam o comportamento da *PML* quando é escolhido uma condutividade máxima de  $\sigma_{\max} = 10,6103$  S/m, este valor é obtido através de (3.61) e é adequado para um meio com constante dielétrica  $\epsilon = \epsilon_0$ . Já as Figuras 3.17 – 3.20 apresenta o comportamento da *PML* quando a condutividade máxima é  $\sigma_{\max} = 3,3553$  S/m, este valor é adequado para um meio com constante dielétrica  $\epsilon = 10\epsilon_0$ .



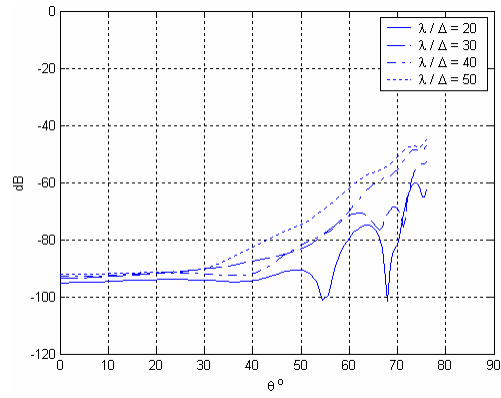
**Figura 3.13** Intensidade relativa do campo  $E_z$  para  $\epsilon = 1\epsilon_0$  e  $\sigma = 10.6103$  S/m.



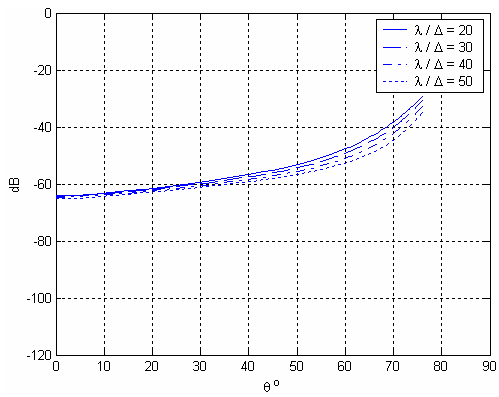
**Figura 3.14** Intensidade relativa do campo  $E_z$  para  $\epsilon = 2\epsilon_0$  e  $\sigma = 10.6103$  S/m.



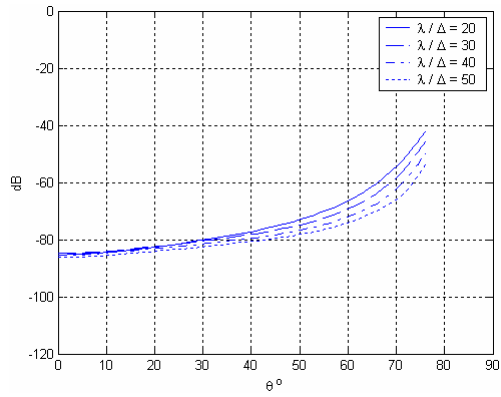
**Figura 3.15** Intensidade relativa do campo  $E_z$  para  $\epsilon = 5\epsilon_0$  e  $\sigma = 10.6103$  S/m.



**Figura 3.16** Intensidade relativa do campo  $E_z$  para  $\epsilon = 10\epsilon_0$  e  $\sigma = 10.6103$  S/m.



**Figura 3.17** Intensidade relativa do campo  $E_z$  para  $\epsilon = 1\epsilon_0$  e  $\sigma = 3.3553$  S/m.



**Figura 3.18** Intensidade relativa do campo  $E_z$  para  $\epsilon = 2\epsilon_0$  e  $\sigma = 3.3553$  S/m.

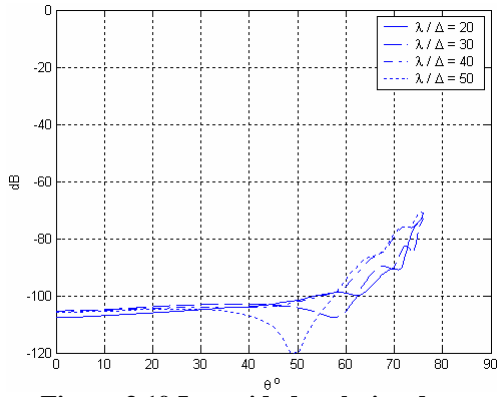


Figura 3.19 Intensidade relativa do campo  $E_z$  para  $\epsilon = 5\epsilon_0$  e  $\sigma = 3.3553$  S/m.

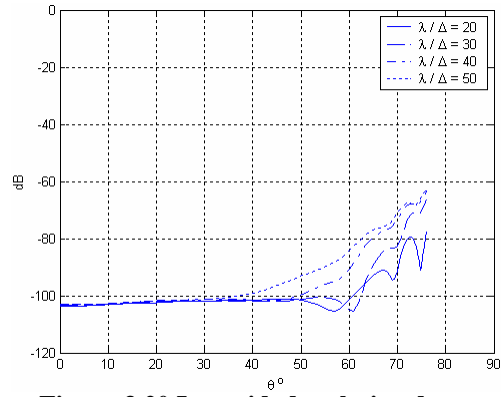


Figura 3.20 Intensidade relativa do campo  $E_z$  para  $\epsilon = 10\epsilon_0$  e  $\sigma = 3.3553$  S/m.

Por estes resultados é possível observar que a atenuação da onda é sensível à constante dielétrica do meio, o que já era previsto por (3.61) proposta por Gedney. Essa característica é indesejada quando é necessário simular meios com constantes dielétricas muito diferentes entre si. Nesse caso, deve-se escolher um valor intermediário para  $\sigma_{\max}$ .

### Energia e Erro

Além do perfil de atenuação, deve-se analisar a absorção de energia e o erro que um domínio truncado por uma fronteira de absorção apresenta em relação a um domínio ideal, ou seja, sem fronteiras. Este erro mostra o quanto o domínio truncado ( $\Omega_T$ ) se difere de um domínio ideal ( $\Omega_I$ ). Na Figura 3.21 está exemplificado o esquema utilizado nas próximas simulações.

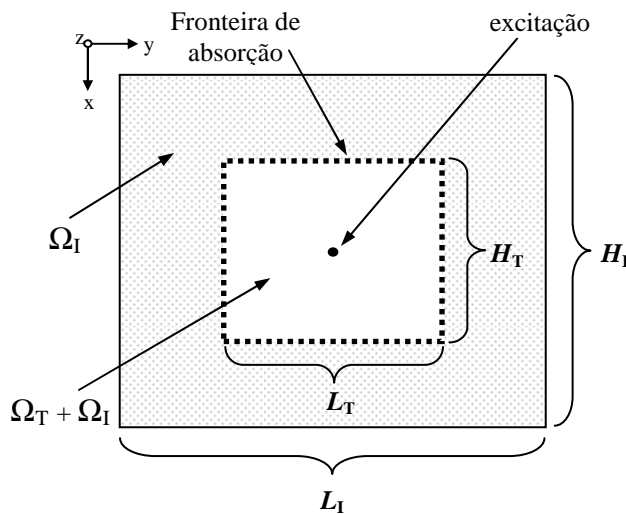


Figura 3.21 Ilustração do domínio computacional utilizado para o cálculo do erro e da energia.



Para a realização destas simulações, uma excitação descrita pela equação a seguir é inserida no centro dos domínios  $\Omega_T$  e  $\Omega_I$ .

$$E_z(n) = \begin{cases} \text{sen}\left(\frac{2\pi}{60}n\right) & ; \text{ para } n \leq 60 \\ 0 & ; \text{ para } n > 60 \end{cases} \quad (3.62)$$

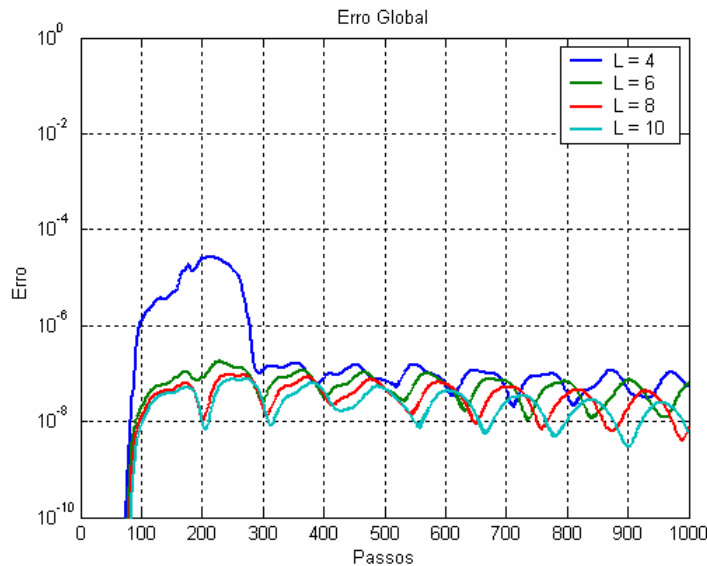
A função erro segundo Taflove [13] e a energia armazenada são calculadas, em cada passo de simulação, respectivamente pelas equações

$$e_{global}|^n = \sum_i \sum_j \left| E_{z,T}|_{(i,j)}^n - E_{z,I}|_{(i,j)}^n \right|^2 ; \quad \text{para } i, j \in \Omega_T \quad (3.63)$$

$$E_G|^n = \sum_i \sum_j \left( \frac{\epsilon_0}{2} E_z^2|_{(i,j)}^n + \frac{\mu_0}{2} H_x^2|_{(i,j)}^n + \frac{\mu_0}{2} H_y^2|_{(i,j)}^n \right) \cdot \Delta^2 ; \quad (3.64)$$

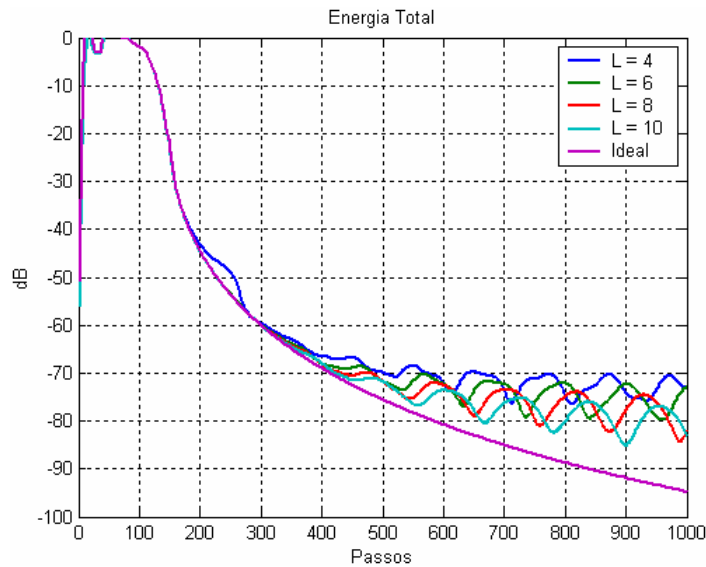
sendo,  $\Delta$  a discretização espacial do domínio simulado.

Os gráficos da Figura 3.22 apresentam o erro quadrático global de alguns domínios truncados pelas camadas *PML* com as seguintes espessuras:  $L = 4$ ,  $L = 6$ ,  $L = 8$  e  $L = 10$  células. Nestes domínios foram mantidas as seguintes propriedades:  $\epsilon = \epsilon_0$ ,  $m = 4$ . É observado nesses gráficos que quanto maior a camada *PML* menor é o erro entre os domínios.



**Figura 3.22 Erro global de algumas camadas *PML*.**

Os gráficos da Figura 3.23 apresentam a energia eletromagnética presente em alguns domínios simulado para as mesmas camadas *PML* simuladas anteriormente ( $L = 4$ ,  $L = 6$ ,  $L = 8$  e  $L = 10$  células). Todos os gráficos estão normalizados pelo seu valor máximo.



**Figura 3.23** Energia no domínio computacional de algumas camadas *PML*.

Os gráficos das Figura 3.22 e 3.23 têm a mesma finalidade dos gráficos anteriores, porém estes apresentam um comportamento global de um domínio truncado pelas camadas *PML*.

### 3.5 Conclusão

As técnicas de truncamento utilizando expressões analíticas, tais como Mur ou Higdon, propiciam um truncamento eficiente do domínio computacional para a maioria dos problemas encontrados na prática. Porém estas técnicas apresentam um limite de absorção da ordem de 40 dB. Além disso, elas não apresentam uma faixa de absorção ampla com relação ao ângulo de incidência das ondas, fazendo com que a absorção varie muito.

Já as técnicas utilizando a *PML* podem apresentar atenuações muito maiores que 40 dB, a custa de um aumento do número de células. Quanto maior for a camada *PML*

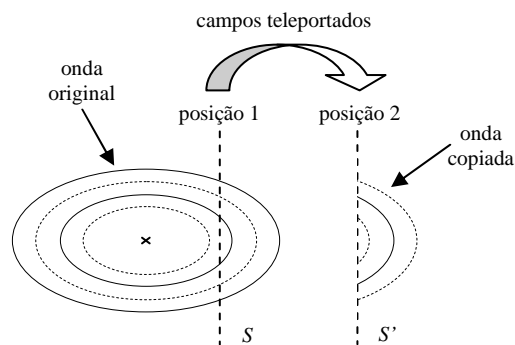
maior será a absorção das ondas. Estas células extras aumentam a quantidade de memória necessária e o número de operações matemáticas requerido.

Uma outra desvantagem desse tipo de técnica é a de não haver uma construção genérica de camadas *PML* que se adapte bem a todos os problemas. A equação proposta por Gedney (3.61) apresenta um contorno para essa dificuldade. Mesmo assim, ela é ajustada para apenas uma constante dielétrica presente no domínio.

# 4 Auto Teleportação de Campos

## 4.1 Introdução

Na técnica de Auto Teleportação de Campos, os campos eletromagnéticos de uma onda incidente em uma superfície localizada na posição 1 do domínio são copiados para outra superfície localizada na posição 2, como ilustrado na Figura 4.1. Os campos copiados são introduzidos no método *FDTD* com sinal contrário.



**Figura 4.1 Ilustração da técnica da Auto Teleportação de Campos.**

Se as superfícies  $S$  e  $S'$  fossem espacialmente coincidentes, uma onda incidente em  $S$  sofreria um cancelamento perfeito. A condição de absorção criada em  $S'$  através desse efeito pode ser utilizada para o truncamento do domínio computacional em algoritmos *FDTD*.

Entretanto, não é possível implementar  $S'$  coincidente com  $S$  no método *FDTD*, pois como os campos em  $S$  são utilizados para atualizar os campos em  $S'$ , se  $S$  e  $S'$  fossem coincidentes haveria uma realimentação positiva no processo de iteração, o que faria o algoritmo divergir. Posicionando-se  $S$  e  $S'$  afastadas de pelo menos uma célula, uma da outra e, tomando-se os campos copiados atrasados de um passo de tempo em relação aos campos originais, evita-se a realimentação.

Nessas condições, o cancelamento da onda será perfeito para problemas unidimensionais em domínios homogêneos e não dissipativos. Para problemas bidimensionais e tridimensionais, o cancelamento não será perfeito devido a diferenças de amplitude e fase entre os campos original e copiado. Apesar disso, essa fronteira de absorção não ideal pode ainda ser utilizada eficientemente no truncamento do domínio computacional, como será mostrado mais adiante.

## 4.2 Implementação da Auto Teleportação no Método *FDTD*

A implementação desta técnica utiliza uma forma discretizada do Teorema da Equivalência de Schelkunoff. Este teorema diz que os campos exteriores a uma fonte envolvida por uma superfície imaginária fechada  $S$  podem ser obtidos através da densidade de corrente superficial elétrica  $K_e$  e da densidade de corrente superficial magnética  $K_m$  induzidas sobre essa superfície, eliminado-se a fonte original. Estas densidades de corrente superficiais são escolhidas de tal forma que os campos no interior de  $S$  sejam nulos e os campos no exterior de  $S$  sejam iguais aos campos originais criados pela fonte. Assim, as densidades  $K_e$  e  $K_m$  são obtidas por

$$\vec{K}_e = \hat{n} \times \vec{H}_S; \quad (4.1)$$

$$\vec{K}_m = -\hat{n} \times \vec{E}_S; \quad (4.2)$$

sendo,  $\hat{n}$  o vetor normal à superfície  $S$  com orientação para fora do domínio e  $\vec{H}_S$  e  $\vec{E}_S$  os campos tangentes à superfície  $S$ .

A implementação dessas equações no método *FDTD* exige a criação de um plano imaginário  $S$  onde os campos elétrico e magnético de uma onda incidente são utilizados para gerar as densidades de corrente  $K_e$  e  $K_m$  em um plano imaginário  $S'$ , como ilustrado na Figura 4.1. Essas densidades de corrente são introduzidas no método *FDTD* através das equações

$$\mu \frac{\partial \vec{H}}{\partial t} = -\vec{\nabla} \times \vec{E} - \frac{\vec{K}_m}{\Delta}; \quad (4.3)$$

$$\varepsilon \frac{\partial \vec{E}}{\partial t} = \vec{\nabla} \times \vec{H} - \frac{\vec{K}_e}{\Delta}; \quad (4.4)$$

onde, devido aos sinais de  $K_m$  e  $K_e$  os campos copiados são introduzidos com sinal invertido e  $\Delta$  é o comprimento da célula.

Ao se aplicar a técnica das diferenças centrais nas equações (4.3) e (4.4), obtém-se as seguintes equações discretizadas:

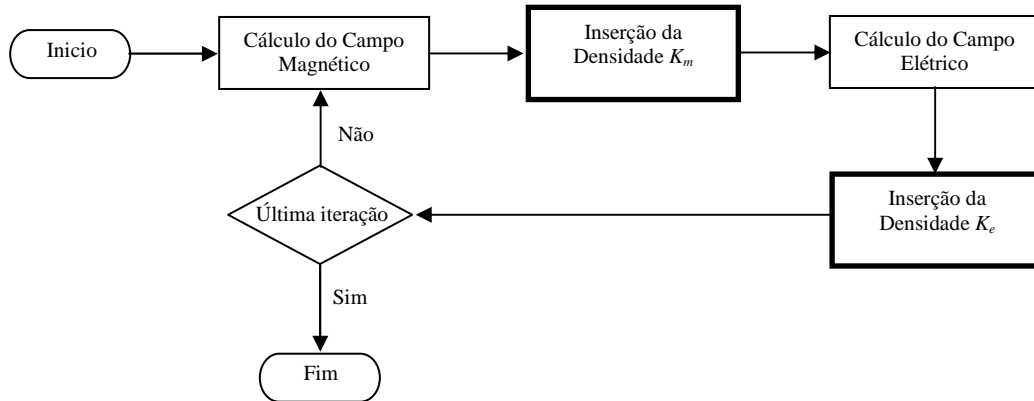
$$\vec{H}^{n+1/2} = \left[ \frac{1 - \frac{\rho^* \Delta t}{2\mu}}{1 + \frac{\rho^* \Delta t}{2\mu}} \vec{H}^{n-1/2} - \frac{\Delta t}{\mu + \rho^* \Delta t / 2} (\vec{\nabla} \times \vec{E})^n \right] - \frac{\Delta t}{(\mu + \rho^* \Delta t / 2) \Delta} \vec{K}_m; \quad (4.5)$$

$$\vec{E}^{n+1} = \left[ \frac{1 - \frac{\sigma \Delta t}{2\varepsilon}}{1 + \frac{\sigma \Delta t}{2\varepsilon}} \vec{E}^n + \frac{\Delta t}{\varepsilon + \sigma \Delta t / 2} (\vec{\nabla} \times \vec{H})^{n+1/2} \right] - \frac{\Delta t}{(\varepsilon + \sigma \Delta t / 2) \Delta} \vec{K}_e. \quad (4.6)$$

Nota-se que os termos dentro do contorno são os mesmos encontrados nas equações convencionais do método *FDTD*. Com isso (4.5) e (4.6) podem, seguindo o algoritmo apresentado na Figura 4.2, ser reescritas nas formas

$$\vec{H}^{n+1/2} = \left[ \vec{H}^{n+1/2} \right] - \frac{R \cdot \Delta t}{(\mu + \rho^* \Delta t / 2) \Delta} \vec{K}_m; \quad (4.7)$$

$$\vec{E}^{n+1} = \left[ \vec{E}^{n+1} \right] - \frac{R \cdot \Delta t}{(\varepsilon + \sigma \Delta t / 2) \Delta} \vec{K}_e; \quad (4.8)$$



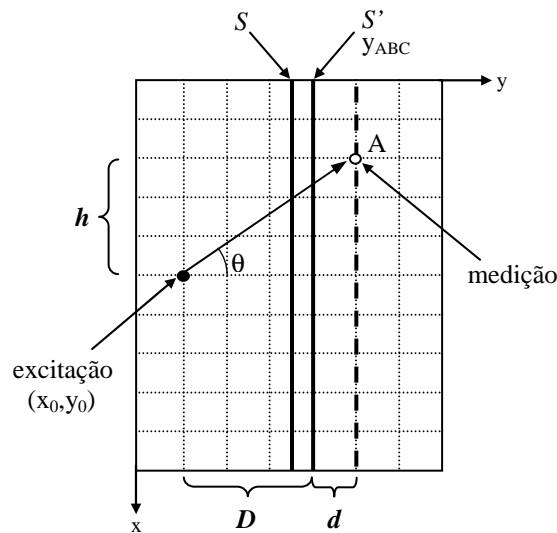
**Figura 4.2 Fluxograma da implementação da Auto Teleportação de Campos no método *FDTD*.**

A constante  $R$ , acrescentada em (4.7) e (4.8), é chamada de fator de amortecimento. Esse fator de amortecimento é utilizado para garantir a estabilidade do algoritmo para simulações de longa duração. Como reportado em [7 – 9], um valor de  $R = 0,99$  garante a estabilidade do algoritmo.

### 4.2.1 Resultados da Auto Teleportação de Campos

As simulações a seguir apresentam a atenuação de uma fronteira de Auto Teleportação para uma onda incidente. Nessas simulações, é utilizado um domínio computacional de  $\{200 \times 150\}$  células, com um pulso gaussiano posicionado em  $x_0 = 100$  células e  $y_0 = 75$  células como fonte de excitação e uma fronteira de Auto Teleportação posicionada em  $y_{ABC} = 85$  células.

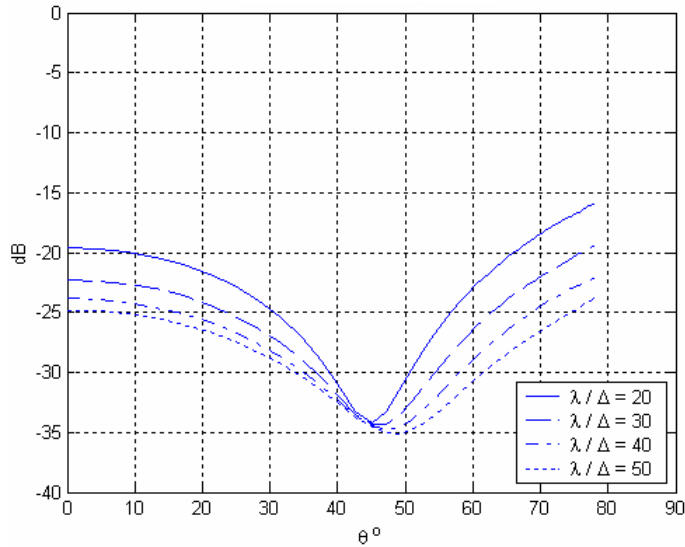
As simulações realizadas são apresentadas em duas partes. A primeira visa analisar o comportamento da onda ao atravessar uma fronteira de Auto Teleportação. A segunda visa analisar o comportamento da onda ao atravessar uma fronteira de Auto Teleportação com reflexão pela superfície limítrofe do domínio computacional, representada por um *PEC*.



**Figura 4.3** Domínio computacional utilizado.  $\theta$  é o ângulo de incidência da onda na fronteira de Auto Teleportação.

A Figura 4.3 ilustra o domínio computacional utilizado.  $D$  é a distância na direção  $y$  entre o ponto de excitação e a superfície  $S'$ ,  $d$  é a distância na direção  $y$  entre a superfície  $S'$  e o ponto de medição dos campos e  $h$  é a distância no eixo  $x$  entre os pontos de excitação e medição. Na simulação são utilizados  $D = 10$  células e  $d = 3$  células.

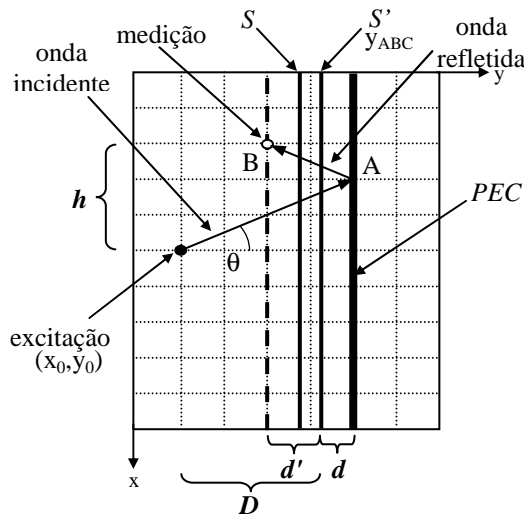
O gráfico da Figura 4.4 mostra a intensidade do campo elétrico no ponto A de uma onda transmitida através da fronteira de Auto Teleportação em relação a uma onda sem a presença dessa fronteira.



**Figura 4.4 Intensidade relativa do campo  $E_z$  no ponto A.**

As fronteiras de Auto Teleportação atenuam as ondas que a atravessam sem causar reflexão. Para se utilizar essa técnica para truncar domínios computacionais, é necessário analisar o seu comportamento em conjunto com as superfícies limítrofes do domínio, que ocorre após ela. Este comportamento conjunto é apresentado na próxima simulação em que um *PEC* é utilizado para truncar o domínio computacional.

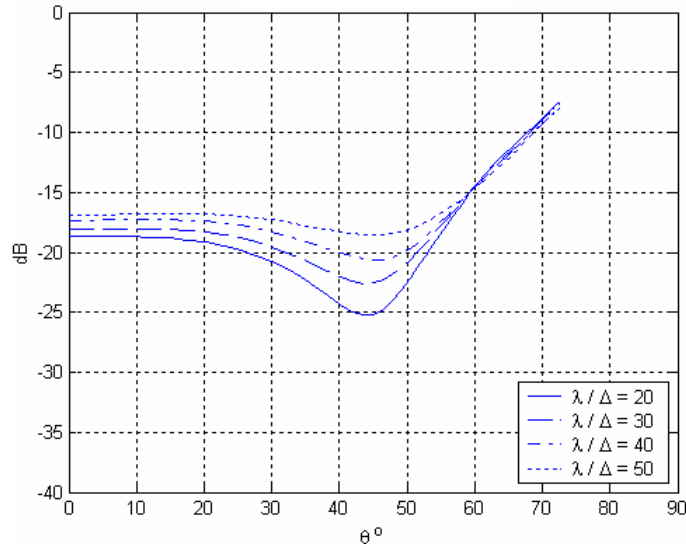
A Figura 4.5 ilustra esta nova condição. Nela  $D$  é a distância na direção  $y$  entre o ponto de excitação e a superfície  $S'$ ,  $d'$  é a distância na direção  $y$  entre a superfície  $S'$  e o ponto de medição dos campos,  $d$  é a distância no eixo  $y$  entre  $S'$  e o *PEC* e  $h$  é a distância no eixo  $x$  entre os pontos de excitação e medição. Para esta simulação são utilizados  $D = 10$  células,  $d = 3$  células e  $d' = 3$  células.



**Figura 4.5 Domínio computacional utilizado.  $\theta$  é o ângulo de incidência da onda na fronteira de Auto Teleportação.**



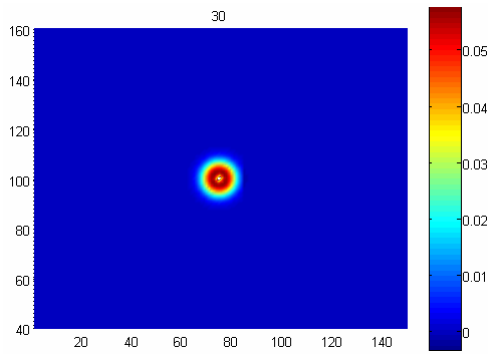
O gráfico da Figura 4.6 mostra a intensidade relativa do campo elétrico de uma onda no ponto  $B$  após atravessar a fronteira de Auto Teleportação e ser refletida pelo  $PEC$ , em relação a uma onda sem a presença dessa fronteira e do  $PEC$ .



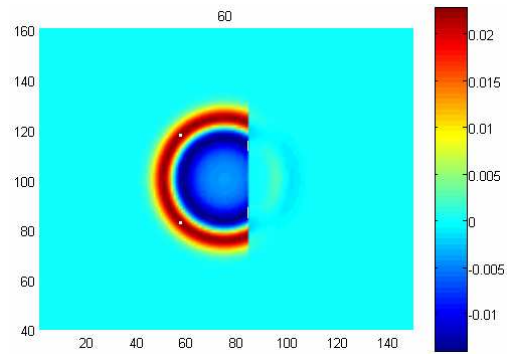
**Figura 4.6 Intensidade relativa do campo  $E_z$  no ponto B.**

Este gráfico apresenta o comportamento de um domínio truncado pela técnica da Auto Teleportação. Nota-se que esse comportamento difere do apresentado na Figura 4.4 porque uma parcela da onda que foi transmitida pela fronteira de Auto Teleportação é refletida pelo  $PEC$  e volta a incidir sobre a fronteira. Isso cria uma re-inserção da onda no domínio computacional e por conseguinte o aparecimento de uma onda refletida.

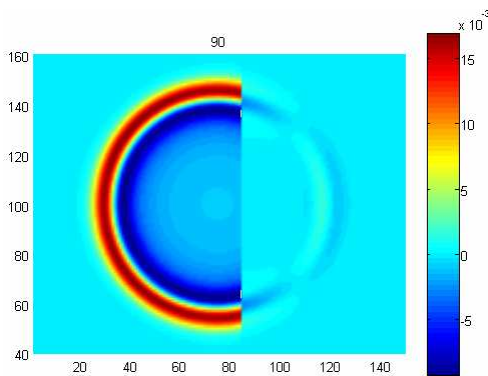
Para se entender o que ocorre nas duas situações apresentadas anteriormente, são apresentadas duas seqüências de imagem que ilustram o comportamento do campo elétrico no domínio computacional. A seqüência apresentada nas Figuras 4.7(a – d) ilustram o comportamento do campo elétrico ao atravessar uma fronteira de Auto Teleportação. Já a seqüência das Figuras 4.8(a – d) ilustram o comportamento do campo elétrico ao atravessar uma fronteira de Auto Teleportação e ser refletida pelo  $PEC$ .



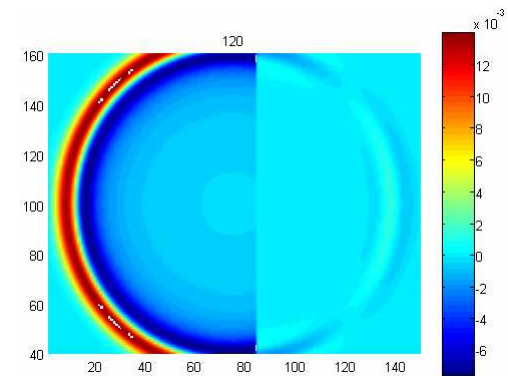
(a) Campo  $E_z$  no passo 30



(b) Campo  $E_z$  no passo 60



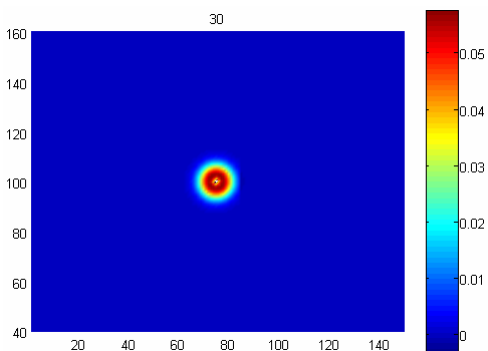
(c) Campo  $E_z$  no passo 90



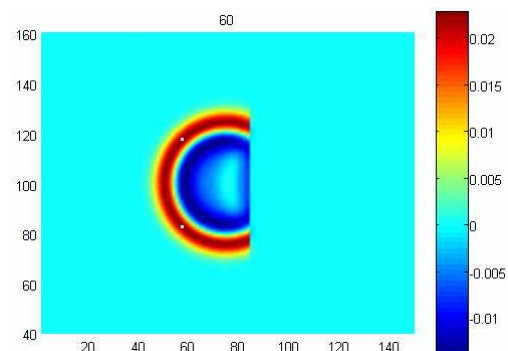
(d) Campo  $E_z$  no passo 120

**Figura 4.7** Comportamento do campo elétrico ao atravessar uma fronteira de Auto Teleportação.

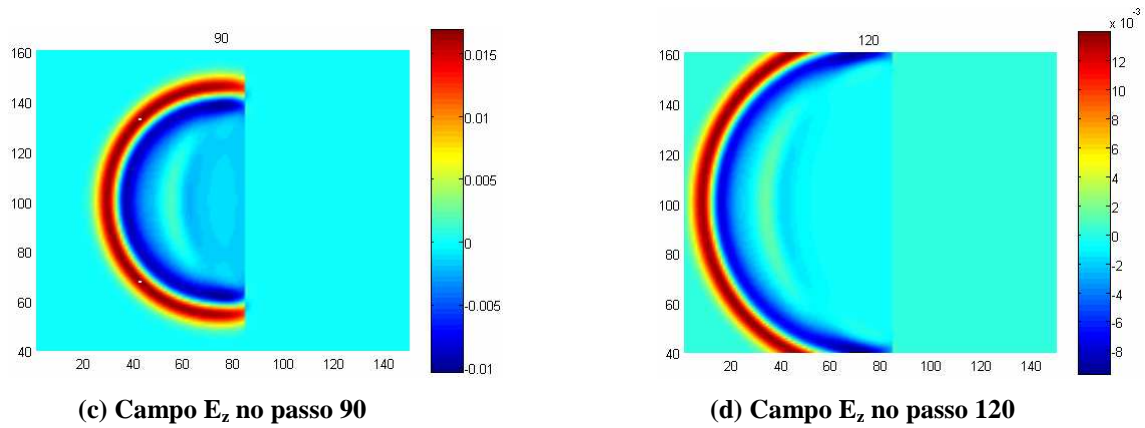
Por essa seqüência de passos é observado que uma onda ao se propagar através de uma fronteira de Auto Teleportação sofre atenuação. Essa atenuação é mais pronunciada em  $45^\circ$ , como pode ser observado pela Figura 4.4. Por conveniência a escala de cores é modificada, em cada imagem, para melhor observação do processo.



(a) Campo  $E_z$  no passo 30



(b) Campo  $E_z$  no passo 60



**Figura 4.8** Comportamento do campo elétrico ao atravessar uma fronteira de Auto Teleportação e ser refletida por um *PEC*.

Dessa seqüência de imagens é observado que parte da onda é refletida pelo conjunto fronteira de Auto Teleportação e *PEC*. Essa parcela da onda possui uma amplitude maior do que a onda transmitida devido a re-inserção da onda no domínio computacional. Novamente, por conveniência, a escala de cores é alterada para melhor visualização do processo.

### Energia e Erro

Além do perfil de atenuação, deve-se também analisar a absorção de energia e o erro que um domínio truncado por uma fronteira de Auto Teleportação apresenta em relação a um domínio ideal, ou seja, sem fronteiras. Este erro mostra quanto o domínio truncado ( $\Omega_T$ ) difere do domínio ideal ( $\Omega_I$ ). A Figura 4.9 apresenta o esquema utilizado nestas simulações.

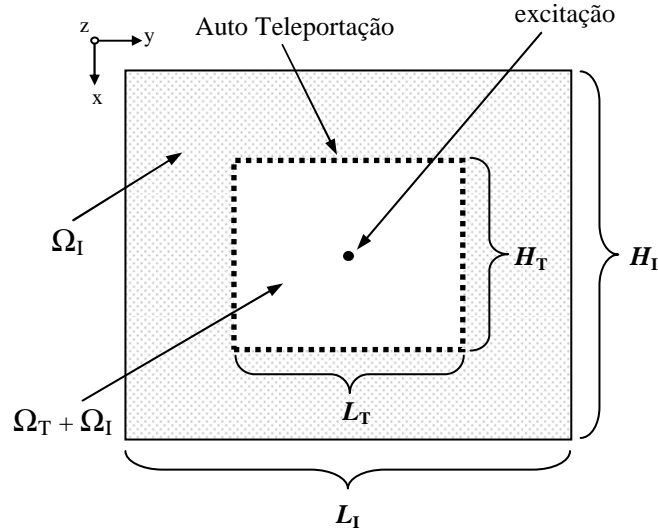


Figura 4.9 Ilustração do domínio computacional utilizado para o cálculo do erro e da energia.

Nestas simulações é utilizada uma fonte de excitação descrita por

$$E_z(n) = \begin{cases} \text{sen}\left(\frac{2\pi}{60}n\right) & ; \text{ para } n \leq 60 \\ 0 & ; \text{ para } n > 60 \end{cases}, \quad (4.9)$$

que está posicionada no centro dos domínios  $\Omega_T$  e  $\Omega_I$ . A função erro é calculado em cada instante, seguindo a formulação de Taflove [13], através da equação

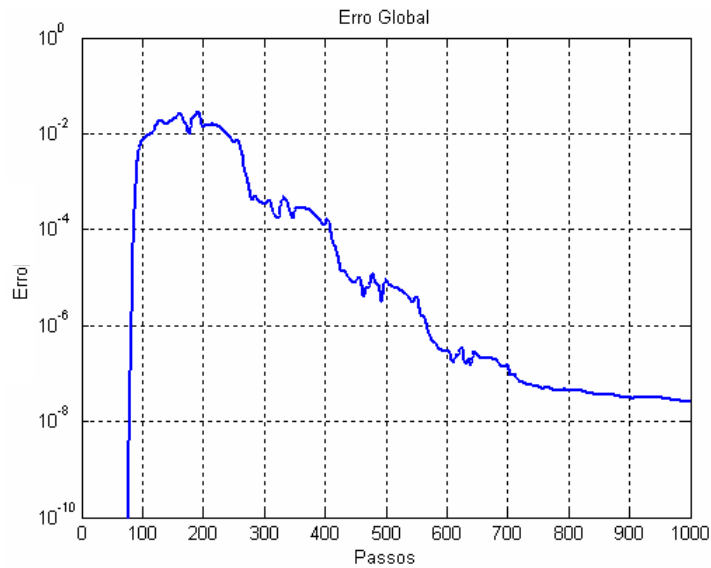
$$e_{global}|^n = \sum_i \sum_j \left| E_{z,T}|_{(i,j)}^n - E_{z,I}|_{(i,j)}^n \right|^2; \quad \text{para } i, j \in \Omega_T \quad (4.10)$$

e a energia por unidade de comprimento no domínio computacional é calculada através da equação

$$E_G|^n = \sum_i \sum_j \left( \frac{\epsilon_0}{2} E_z^2|_{(i,j)}^n + \frac{\mu_0}{2} H_x^2|_{(i,j)}^n + \frac{\mu_0}{2} H_y^2|_{(i,j)}^n \right) \cdot \Delta^2 \quad (4.11)$$

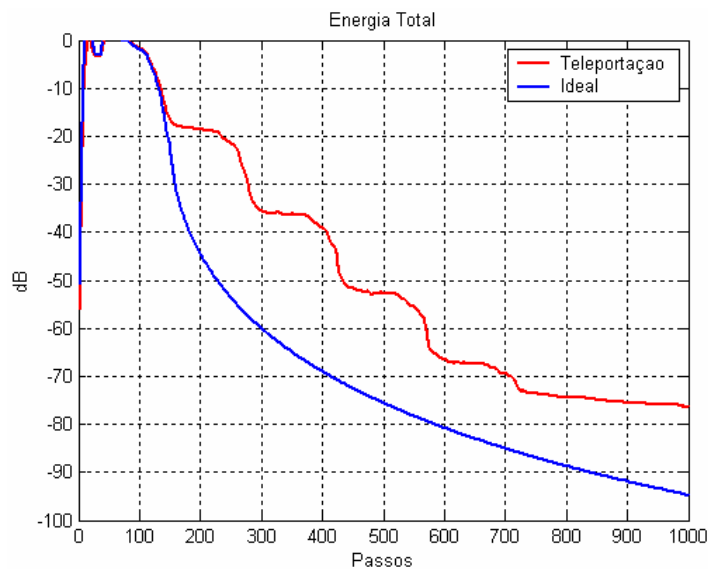
sendo,  $\Delta$  a discretização espacial do domínio simulado.

O gráfico da Figura 4.10 apresenta o erro quadrático global de um domínio computacional truncado empregando-se a técnica Fronteira de Auto Teleportação de Campos.



**Figura 4.10 Erro global para o domínio computacional.**

Nota-se que este gráfico apresenta patamares em virtude das ondas incidentes nas fronteiras não serem totalmente absorvidas e voltarem para o domínio computacional. Este processo ocorre até que a intensidade das ondas refletidas seja tão baixa que as fronteiras de Auto Teleportação não consigam mais absorvê-las. Devido ao surgimento de componentes de alta frequência, criados por erros numéricos. Esses componentes não atendem à condição de estabilidade do método *FDTD* descrita no item A.4 do Anexo A.



**Figura 4.11 Energia no domínio computacional.**

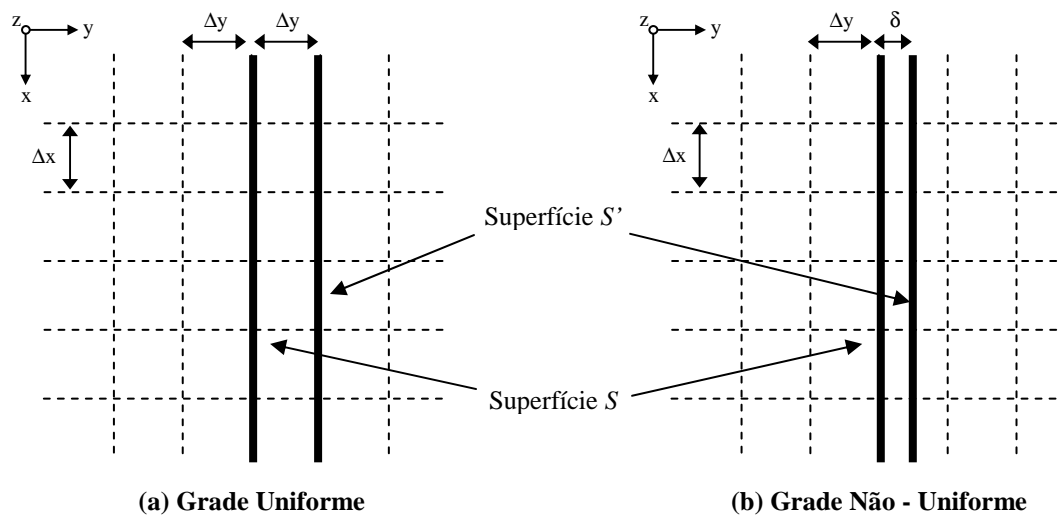
As curvas da Figura 4.11 apresentam a energia eletromagnética presente no domínio computacional simulado. A curva inferior representa a energia no domínio ideal e a curva superior representa a energia no domínio truncado. Ambas as curvas estão normalizadas pelo seu valor máximo. A curva da energia no domínio truncado possui uma forma irregular. Devido à re-inserção das ondas não absorvidas no domínio computacional. Os patamares nesse gráfico indicam que a onda está no domínio computacional e as inclinações indicam que a onda está sendo absorvida pelas fronteiras de Auto Teleportação.

A re-inserção da onda é um efeito indesejado porque faz com que o domínio truncado apresente um erro em relação a um domínio ideal. Este problema pode ser reduzido pela adição de uma outra fronteira de re-radiação, tal como Mur, Higdon, Huygens ou mesmo aumentar o número de fronteiras de Auto Teleportação. Outra forma de minimizar este problema é modificar o algoritmo original proposto por Diaz e Scherbatko através de compensações. A seguir estão descritas três formas de compensação: Compensação Espacial, Compensação Temporal e Compensação por Equação de Onda.

### 4.3 Compensação Espacial

Como a superfície dos campos copiados e a superfície dos campos originais não podem ocupar a mesma posição na grade *FDTD*, é preciso deixá-las o mais próximo possível uma da outra. No algoritmo original, estas superfícies estão posicionadas uma célula de distância uma da outra.

Porém, é possível aproximar ainda mais essas superfícies através de uma técnica comumente utilizada para aumentar a precisão dos cálculos numéricos em apenas certas regiões do domínio computacional. Esta técnica é conhecida como Grade Não – Uniforme [16 – 19], e está apresentada em detalhes no Anexo B. A idéia é “comprimir” uma ou mais células do domínio computacional, de forma que os componentes dos campos elétrico e magnético fiquem o mais próximo possível uns dos outros. Esta “compressão” torna possível aproximar as fronteiras  $S$  e  $S'$ . Na Figura 4.12 estão ilustradas uma grade uniforme e uma grade não uniforme no eixo  $y$ .



**Figura 4.12** Diferença entre uma grade uniforme de uma grade não uniforme.

No domínio formado pelas células não uniformes as superfícies  $S$  e  $S'$  estão mais próximas uma da outra, através de células de comprimento  $\delta$  em relação ao eixo  $y$ .

### 4.3.1 Resultados da Compensação Espacial

As simulações a seguir apresentam o comportamento das fronteiras de Auto Teleportação modificadas pela compensação espacial. Nestas simulações é levantado o perfil de atenuação de uma onda em relação ao comprimento das células não uniformes ( $\delta$ ), como ilustrado na Figura 4.12. Esse comprimento é obtido através da relação  $\delta = \Delta y / m$ , sendo  $m$  um fator de redução de comprimento. O domínio computacional utilizado nestas simulações está ilustrado na Figura 4.3.

Os gráficos das Figuras 4.13 – 4.16 mostram a intensidade relativa do campo elétrico de uma onda transmitida através da fronteira de Auto Teleportação modificada pela compensação espacial, em relação a uma onda sem a presença dessa fronteira.

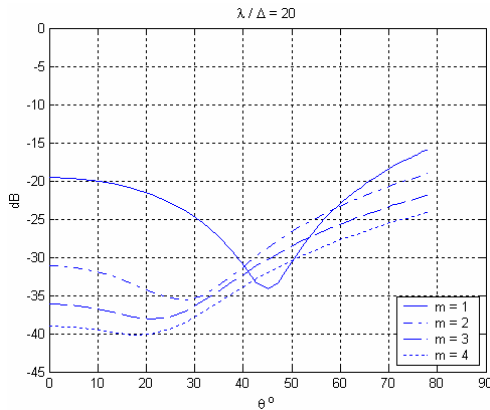


Figura 4.13 Intensidade relativa do campo  $E_z$

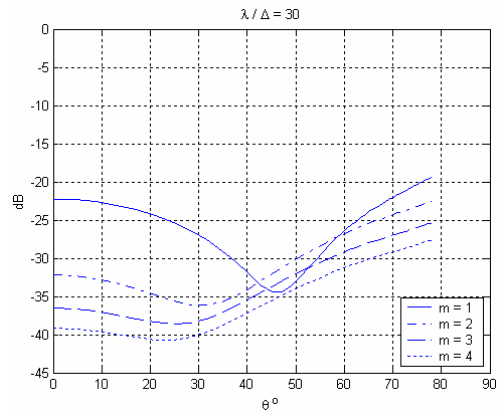


Figura 4.14 Intensidade relativa do campo  $E_z$

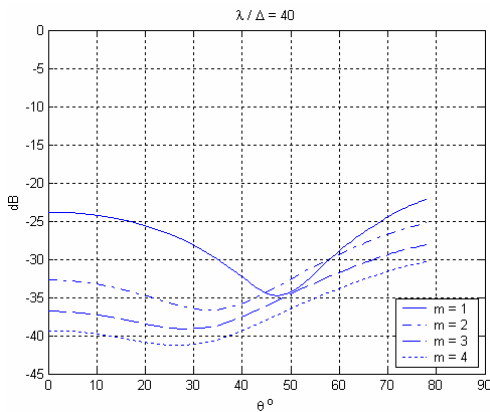


Figura 4.15 Intensidade relativa do campo  $E_z$

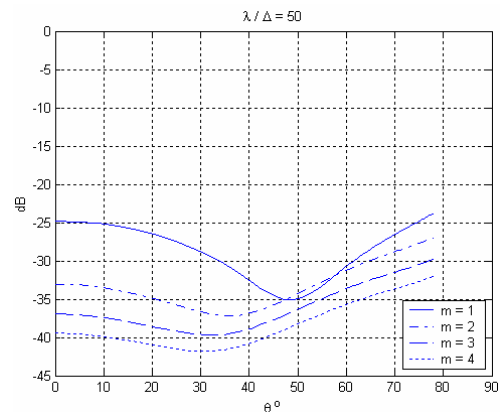


Figura 4.16 Intensidade relativa do campo  $E_z$

Por estes gráficos é possível observar a influência do tamanho das células não uniformes na atenuação da onda incidente. Pode se notar que a maior diferença ocorre quando as células possuem a metade do tamanho original. Para tamanhos menores, essa atenuação não sofre uma melhora pronunciada. Também pode ser observado que o ângulo de mínimo é alterado pela proximidade das superfícies  $S$  e  $S'$ . O que indica que a discretização espacial tem influência na atenuação da onda.

Novamente, é necessário analisar o comportamento desta técnica quando ela é utilizada para truncar domínios computacionais. As simulações a seguir apresentam as características de atenuação para esta condição. O domínio computacional utilizado nestas simulações está ilustrado na Figura 4.5. Os campos elétricos foram medidos a duas células de distância antes da fronteira de Auto Teleportação e o domínio computacional é truncado à 3 células de distância após essa fronteira. Os gráficos das



Figuras 4.17 – 4.20 mostram a intensidade relativa da onda refletida na fronteira do domínio computacional em relação a uma onda sem a presença dessa fronteira.

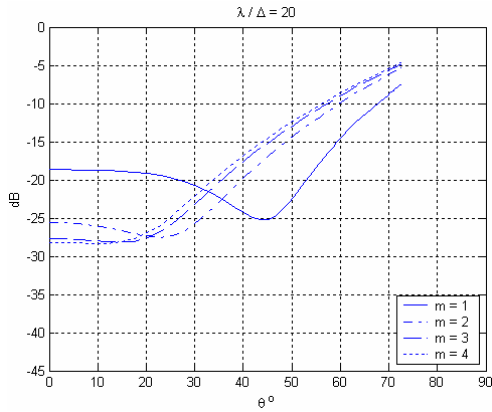


Figura 4.17 Intensidade relativa do campo  $E_z$

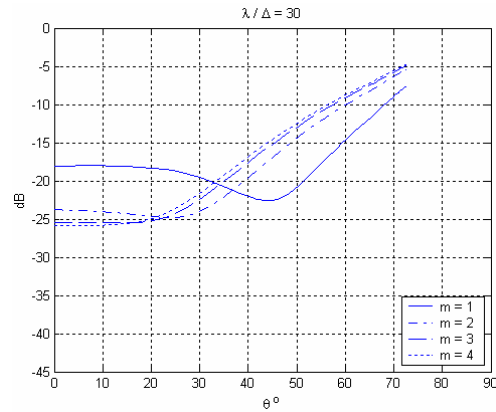


Figura 4.18 Intensidade relativa do campo  $E_z$

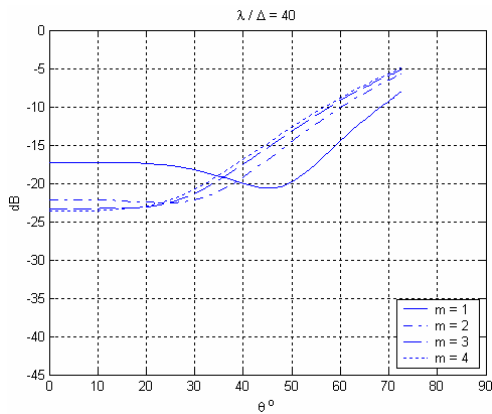


Figura 4.19 Intensidade relativa do campo  $E_z$

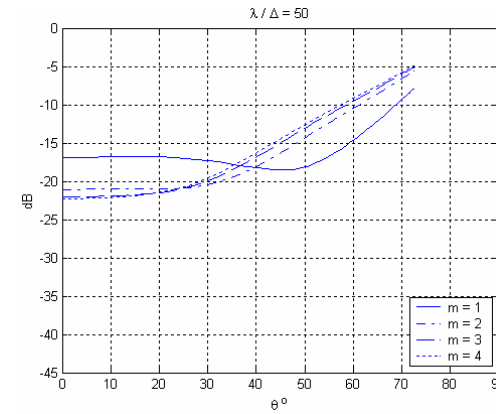


Figura 4.20 Intensidade relativa do campo  $E_z$

Novamente é observada a influência do tamanho das células na atenuação da onda. Assim como nos gráficos apresentados anteriormente, a maior diferença ocorre quando se tem uma célula com a metade do tamanho das células originais. Células menores não apresentam vantagem com relação a atenuação da onda, além de resultarem em um aumento da carga computacional.

### 4.3.2 Desvantagem da Compensação Espacial

As fronteiras de Auto Teleportação com compensação espacial apresentam, em uma certa faixa de ângulos de incidência, uma melhora na absorção da onda quando comparada com uma fronteira de Auto Teleportação sem compensação. Essa

característica pode prejudicar a absorção da onda quando há a interseção de duas dessas fronteiras. Como a compensação espacial apresenta uma boa atenuação em ângulos próximos à incidência normal, ela é eficientemente aplicável a guias de onda, nos quais os campos apresentam um comportamento bem definido.

Outra característica deste tipo de compensação é o aumento da carga computacional devido ao critério de estabilidade.

Para evitar a divergência do algoritmo *FDTD*, deve-se aplicar a condição de estabilidade de Courant, utilizando as menores dimensões das células do domínio computacional, dada por.

$$c \cdot \Delta t \leq \left[ \left( \frac{1}{\Delta x_{\min}} \right)^2 + \left( \frac{1}{\Delta y_{\min}} \right)^2 \right]^{-1/2}. \quad (4.12)$$

Como a compensação espacial diminui algumas células do domínio, estas dimensões devem ser computadas por esta condição. Isso faz com que a discretização temporal ( $\Delta t$ ) seja menor quando comparada com um domínio sem compensação espacial. Para manter o mesmo tempo real de análise do problema ( $T_{real} = N_{\max} \cdot \Delta t$ ), torna-se necessário aumentar o número máximo de iterações, o que leva a um aumento do tempo de simulação.

Nas simulações anteriores são utilizadas as seguintes discretizações espaciais  $\Delta x = \Delta$  e  $\Delta y = \Delta / m$ . A equação que descreve quanto deve ser o acréscimo no número máximo de iterações em função do fator de redução  $m$ , é

$$N'_{\max} = N_{\max} \cdot \sqrt{\frac{1+m^2}{2}}, \quad (4.13)$$

sendo,  $N'_{\max}$  o número máximo de iterações com compensação espacial e  $N_{\max}$  o número máximo de iterações sem compensação espacial.

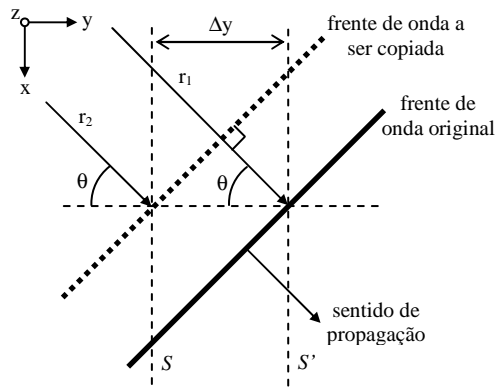
Nota-se que esta equação não é geral, mas descreve os tempos computacionais para os problemas simulados nesta seção.

## 4.4 Compensação Temporal

No algoritmo da Auto Teleportação, os campos copiado e original devem apresentar um atraso de um passo de simulação entre eles. Porém, isso cria um erro de fase entre estes campos. Para contornar este problema, é necessário aplicar uma

compensação temporal nesses campos. Este tipo de compensação pode ser realizado através do seguinte procedimento.

Para que uma fronteira de Auto Teleportação cause cancelamento perfeito de uma onda, é necessário que a fase da onda copiada em  $S$  seja idêntica a fase da onda em  $S'$ . As deduções a seguir são feitas com base em uma frente de onda plana se propagando nos sentidos positivos de  $x$  e  $y$ , como ilustra a Figura 4.21. Nesta figura a variável  $\Delta y$  é a discretização espacial no eixo  $y$  e  $\theta$  é o ângulo de incidência da onda nas superfícies  $S$  e  $S'$  com relação a sua normal.



**Figura 4.21 Diagrama das ondas copiadas e originais.**

Uma onda plana se propagando nos sentidos positivos de  $x$  e  $y$  pode ser descrita através das seguinte equações

$$f(r, t) = e^{j(k \cdot r - \omega \cdot t)}, \quad (4.14)$$

$$f(r_2, t_2) = e^{j(k \cdot r_2 - \omega \cdot \Delta t \cdot (n-a))}, \quad (4.15)$$

$$f(r_1, t_1) = e^{j(k \cdot r_1 - \omega \cdot \Delta t \cdot n)}. \quad (4.16)$$

A equação geral é dada por (4.14) e as equações (4.15) e (4.16) descrevem a onda nas posições  $S$  e  $S'$ , respectivamente.

Nestas equações  $r_1$  e  $r_2$  são as distâncias entre a origem do sistema de coordenadas e as superfícies  $S'$  e  $S$  respectivamente,  $n$  é o instante de tempo computacional,  $\Delta t$  é a discretização temporal e  $a$  é o atraso computacional da onda a ser copiada. Desse modo,  $t_1 = \Delta t \cdot n$  e  $t_2 = \Delta t \cdot (n - a)$ .

Para que haja um cancelamento perfeito entre as ondas copiada e original, a igualdade

$$f(r_1, t_1) = f(r_2, t_2) \quad (4.17)$$

deve ser imposta. Substituindo as equações (4.15) e (4.16) em (4.17) e fazendo-se as devidas manipulações algébricas, obtém-se o atraso que deverá ser imposto para se obter o cancelamento perfeito da onda. Este atraso é dado por

$$a = \frac{k \cdot (r_1 - r_2)}{\omega \cdot \Delta t}; \quad (4.18)$$

sendo,  $k = \frac{2\pi}{\lambda}$  a frequência espacial,  $\omega = 2\pi \cdot \frac{v}{\lambda}$  a frequência temporal e

$$(r_1 - r_2) = \Delta y \cdot \cos(\theta).$$

Ao substituir  $k$ ,  $\omega$  e  $(r_1 - r_2)$  em (4.18), obtém-se a expressão do atraso  $a$  em função do ângulo de incidência da onda:

$$a = \frac{\Delta y}{v \cdot \Delta t} \cdot \cos(\theta); \quad (4.19)$$

sendo  $v$  a velocidade de onda na fronteira de absorção.

Com isso, a atenuação teórica da onda que atravessa a fronteira de Auto Teleportação pode ser descrita pela equação

$$S_{21} = \frac{f(r_1, t_1) - f(r_2, t_2)}{f(r_1, t_1)} = 1 - \exp\left(-j \frac{2\pi}{\lambda} [\Delta y \cdot \cos(\theta) - v \cdot \Delta t \cdot a']\right); \quad (4.20)$$

sendo  $a'$  um atraso pré-determinado obtido por (4.19). Os gráficos da Figura 4.22 apresentam a atenuação teórica obtida por (4.20) para três valores diferentes de atraso.

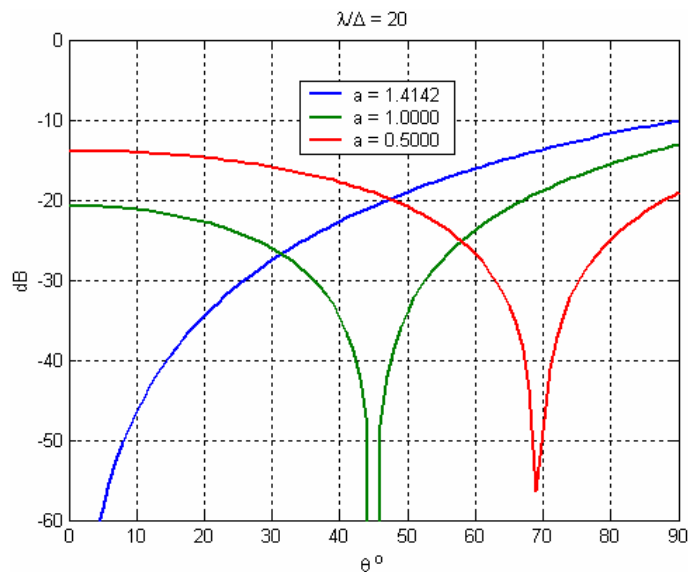


Figura 4.22 Atenuação teórica para uma frente de onda plana.

Estes gráficos mostram como a atenuação da onda incidente é influenciada pelo valor do atraso  $a$ . Além do mais, eles também mostram que o ponto de máxima atenuação está relacionado com o atraso da onda copiada.

Da (4.19), nota-se que o atraso  $a$  pode assumir valor não inteiro, porém no método *FDTD* só dispomos de valores inteiros para as discretizações espacial e temporal. Para obter valores inteiros a partir de valores não inteiros é necessário utilizar a seguinte interpolação linear

$$0 < a \leq 1: \quad g(n-a) = (1-a) \cdot g(n-0) + (a-0) \cdot g(n-1); \quad (4.21)$$

$$1 < a \leq 2: \quad g(n-a) = (2-a) \cdot g(n-1) + (a-1) \cdot g(n-2); \quad (4.22)$$

⋮

$$m < a \leq m+1: \quad g(n-a) = ((m+1)-a) \cdot g(n-m) + (a-m) \cdot g(n-(m+1)); \quad (4.23)$$

sendo  $g(\bullet)$  uma função genérica,  $n$  o instante de tempo computacional discretizado e  $m = \lfloor a \rfloor$ .

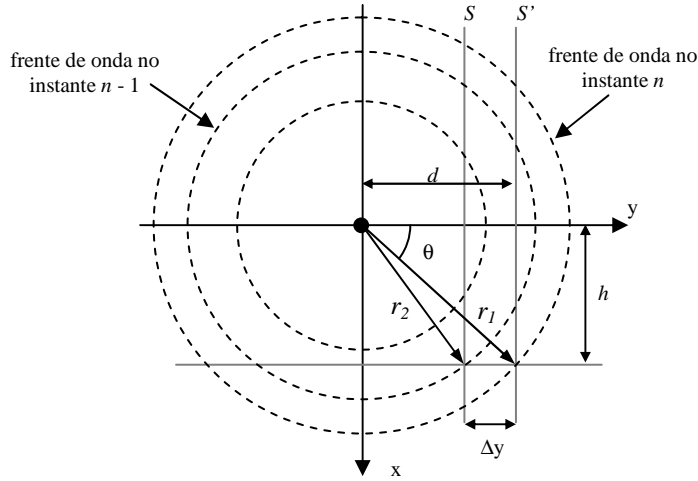
De posse dessa interpolação, é possível reescrever (4.20) para representar a atenuação teórica utilizando-se valores inteiros do atraso  $a$ . Esta nova equação é dada por

$$S_{21} = 1 - \left( (m+1-a) + (a-m) \cdot \exp\left(j \frac{2\pi}{\lambda} [v \cdot \Delta t]\right) \right) \cdot \exp\left(-j \frac{2\pi}{\lambda} [\Delta y \cdot \cos(\theta) - v \cdot \Delta t \cdot m]\right); \quad (4.24)$$

Nas simulações apresentadas neste trabalho é utilizada uma frente de onda cilíndrica. Por isso é necessário descrever a atenuação teórica com base nessa frente de onda, utilizando-se valores discretizados. A nova atenuação pode ser obtida por

$$S_{21} = 1 - A \cdot \left( (m+1-a) + (a-m) \cdot \exp\left(j \frac{2\pi}{\lambda} [v \cdot \Delta t]\right) \right) \cdot \exp\left(-j \frac{2\pi}{\lambda} [(r_1 - r_2) - v \cdot \Delta t \cdot m]\right); \quad (4.25)$$

sendo  $A = \left(\frac{r_1}{r_2}\right)^{0,4}$  um fator de correção obtido experimentalmente para a frente de onda simulada. Com o auxílio da Figura 4.23 são obtidas as relações  $(r_1 - r_2)$  e  $\left(\frac{r_1}{r_2}\right)$ .



**Figura 4.23** Diagrama para o cálculo das relações  $(r_1 - r_2)$  e  $\left(\frac{r_1}{r_2}\right)$

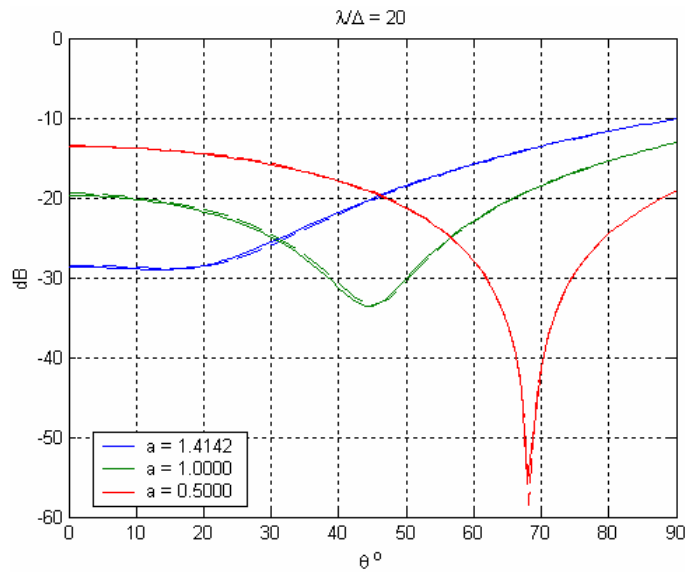
$$\begin{cases} r_1^2 = d^2 + h^2 \\ r_2^2 = (d - \Delta y)^2 + h^2 ; \\ \operatorname{tg}(\theta) = h/d \end{cases} \quad (4.26)$$

Fazendo-se as devidas manipulações algébricas nas relações (4.26), as seguintes relações são obtidas

$$\left(\frac{r_1}{r_2}\right) = \sec(\theta) \cdot \left[ \sec^2(\theta) + \left(\frac{\Delta y}{d}\right)^2 - \frac{2\Delta y}{d} \right]^{-1/2}; \quad (4.27)$$

$$(r_1 - r_2) = d \cdot \left[ \sec(\theta) - \left( \sec^2(\theta) + \left(\frac{\Delta y}{d}\right)^2 - \frac{2\Delta y}{d} \right)^{1/2} \right]; \quad (4.28)$$

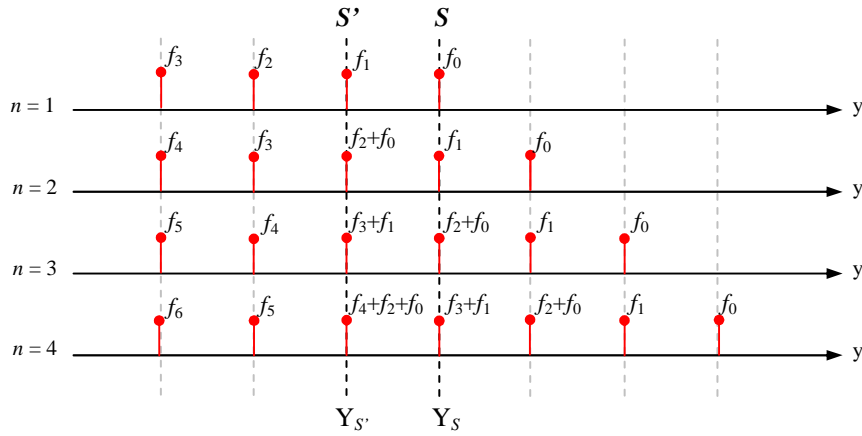
Substituindo-se (4.27) e (4.28) em (4.25), obtém-se a equação que descreve a atenuação teórica para a frente de onda simulada. A Figura 4.24 apresenta a comparação entre a atenuação obtida experimentalmente (linha tracejada) e a atenuação calculada (linha cheia) por (4.25) para três valores do atraso ( $a$ ).



**Figura 4.24** Comparação entre a atenuação teórica e simulada para uma frente de onda cilíndrica com três atrasos diferentes.

As equações apresentadas anteriormente apenas descrevem o comportamento das ondas que atravessam uma fronteira de Auto Teleportação no sentido direto. Porém como essas ondas são refletidas pelos limites do domínio computacional, elas voltam a reincidir na fronteira de Auto Teleportação no sentido inverso. Portanto, é importante descrever o comportamento destas ondas refletidas. Infelizmente não existe uma forma fechada para descreve-las, estas ondas só podem ser obtidas de forma iterativa.

Para entender o que ocorre com as ondas refletidas, a seguir é descrito seu comportamento em um domínio unidimensional. A Figura 4.25 ilustra uma onda se propagando no sentido positivo do eixo  $y$  e incidindo em um fronteira de Auto Teleportação. Essa fronteira possui suas superfícies  $S$  e  $S'$  posicionadas respectivamente em  $Y_S$  e  $Y_{S'}$ .



**Figura 4.25 Ilustração de um domínio computacional unidimensional.**

Pela Figura 4.25 é possível observar que o valor da onda na posição  $Y_{S'}$  pode ser descrito na forma

$$y(n) = y(n-2) + x(n); \quad (4.29)$$

onde  $y(\bullet)$  é novo valor da onda na posição  $Y_{S'}$  e  $x(\bullet)$  é o valor antigo da onda na mesma posição  $Y_{S'}$ . Com isso, a equação

$$H(e^{j\Omega}) = \frac{1}{1 - e^{-2j\Omega}} \quad (4.30)$$

descreve a resposta em frequência desse sistema formado pela Fronteira de Auto Teleportação, sendo  $\Omega = \frac{\omega \cdot \Delta t}{2}$ .

As equações anteriores foram obtidas para com um atraso de um passo de simulação ( $a = 1$ ). Para um atraso ajustado pela velocidade de onda, as equações anteriores pode ser reescritas na forma

$$y(n) = y(n - (a+1)) + x(n); \quad (4.32)$$

$$H(e^{j\Omega}) = \frac{1}{1 - e^{-j(a+1)\Omega}}; \quad (4.33)$$

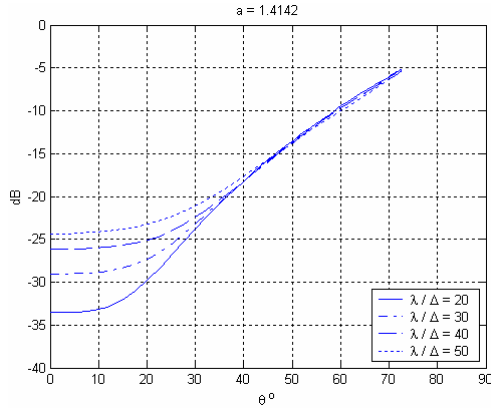
Na equação (4.32) deve se usar a interpolação linear descrita por (4.23).

#### 4.4.1 Resultados da Compensação Temporal

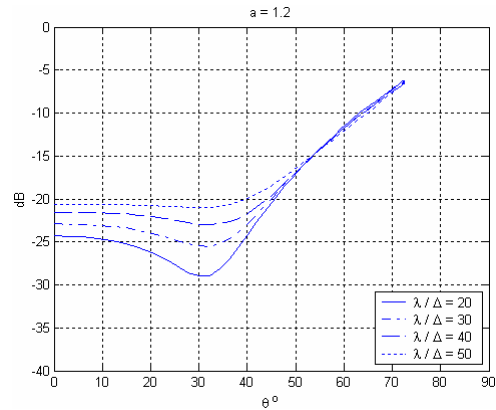
As simulações a seguir apresentam a característica de atenuação que um domínio truncado por uma fronteira de Auto Teleportação com compensação temporal provoca em uma onda nela incidente. O domínio computacional simulado está ilustrado na



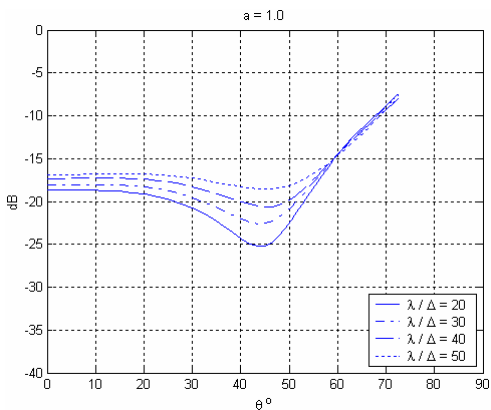
Figura 4.5. As Figuras 4.26 – 4.29 apresentam o perfil de atenuação para os atrasos  $a = 1,4142$ ,  $a = 1,2$ ,  $a = 1,0$  e  $a = 0,5$ . Estes valores são escolhidos de forma a cobrir os ângulos baixos, médios e rasantes de incidência da onda. Nestas figuras é observado que o ângulo de máxima atenuação é alterado conforme o valor do atraso.



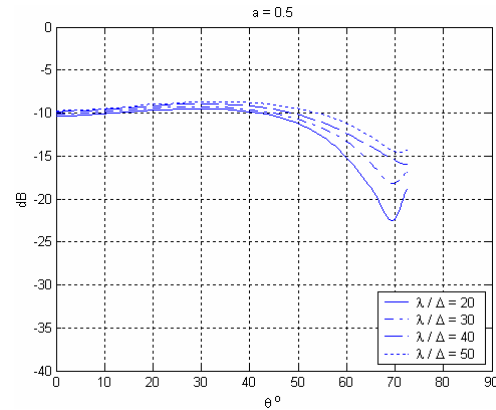
**Figura 4.26** Intensidade relativa do campo  $E_z$  para  $a = 1,4142$ .



**Figura 4.27** Intensidade relativa do campo  $E_z$  para  $a = 1,2$ .



**Figura 4.28** Intensidade relativa do campo  $E_z$  para  $a = 1,0$ .

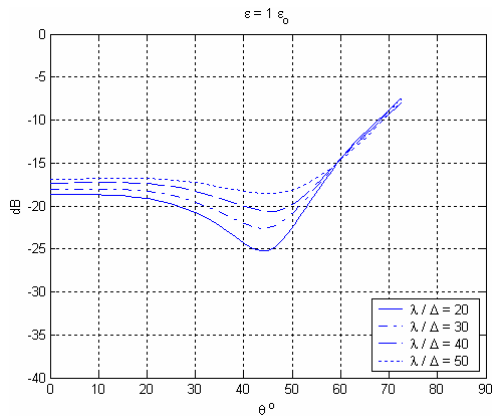


**Figura 4.29** Intensidade relativa do campo  $E_z$  para  $a = 0,5$ .

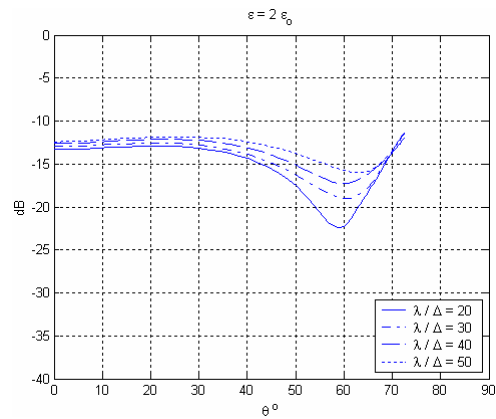
As próximas simulações apresentam o efeito da compensação temporal quando o domínio simulado possui uma constante dielétrica  $\epsilon$  e a discretização temporal  $\Delta t$  é definida como se o domínio tivesse uma constante dielétrica  $\epsilon_0$ . Isso geralmente acontece quando se tem dois ou mais meios diferentes no mesmo domínio, neste caso o  $\Delta t$  é definido para a maior velocidade de onda, ou seja, para o menor  $\epsilon$ .

Nestas simulações, a constante dielétrica  $\epsilon$  do meio é variada porém o atraso entre os campos é mantido igual ao atraso do algoritmo original ( $a = 1,0$ ). Nos gráficos

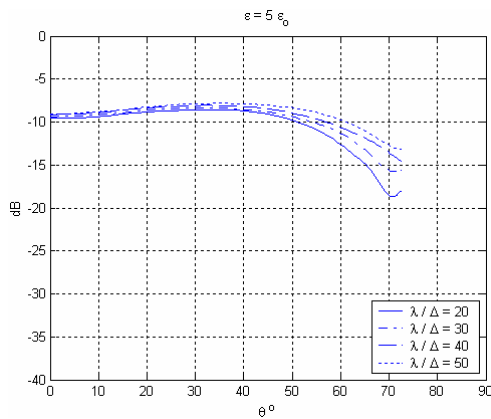
das Figuras 4.30 – 4.33 é observado que quanto maior o valor de  $\epsilon$  menor é a eficiência da absorção de onda nas fronteiras de Auto Teleportação.



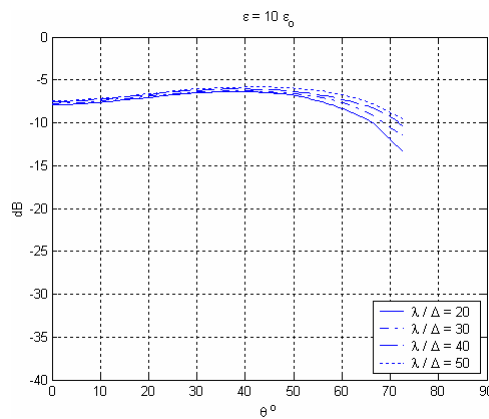
**Figura 4.30** Intensidade relativa do campo  $E_z$  para  $\epsilon = \epsilon_0$ .



**Figura 4.31** Intensidade relativa do campo  $E_z$  para  $\epsilon = 2\epsilon_0$ .

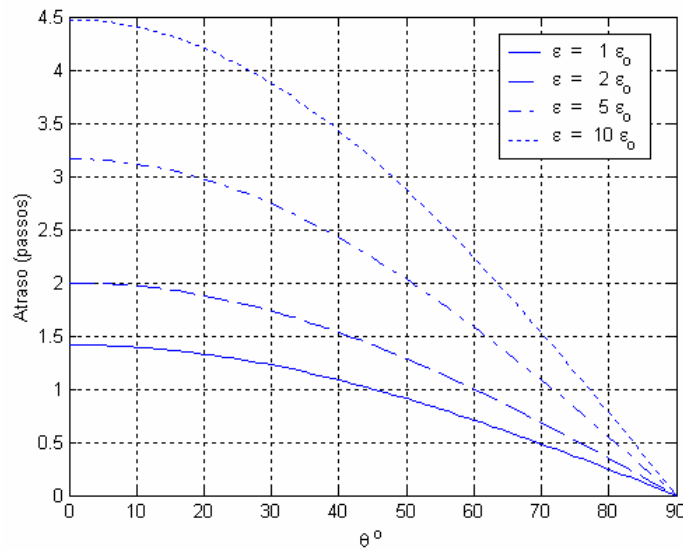


**Figura 4.32** Intensidade relativa do campo  $E_z$  para  $\epsilon = 5\epsilon_0$ .



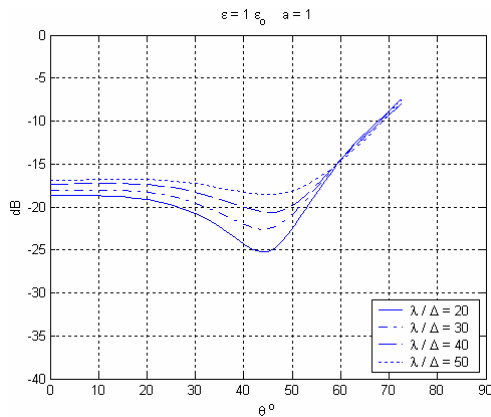
**Figura 4.33** Intensidade relativa do campo  $E_z$  para  $\epsilon = 10\epsilon_0$ .

Nestes gráficos não só é observada a perda de eficiência da absorção em função da constante dielétrica, como também observa-se a variação do ângulo de máxima absorção devida a mudança na constante dielétrica. Este ângulo pode ser previsto pelo uso do gráfico da Figura 4.34. Para  $a = 1,0$  são obtidos os valores ângulos  $45^\circ$ ,  $60^\circ$ ,  $71^\circ$  e  $78^\circ$  para esse ângulo. Este último valor não é aparece nos gráficos devido à limitação da escala de medição dos ângulos em  $72^\circ$ .

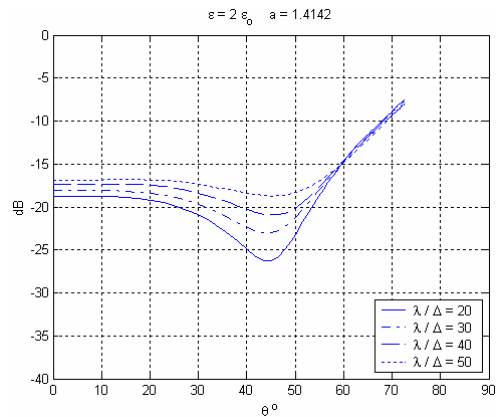


**Figura 4.34** Curvas de atraso em função do ângulo de incidência para vários valores de  $\epsilon$ .

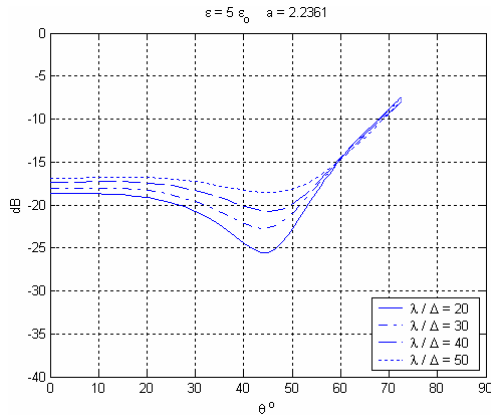
No próximo grupo de simulações, Figuras 4.35 – 4.38, o atraso entre os campos é modificado conforme o valor da constante dielétrica do meio. Isto é feito de forma a se obter um máximo de atenuação em  $45^\circ$ , ângulo também obtido pelo algoritmo original, sem compensação. O valor do atraso  $a$  é obtido através das curvas da Figura 4.26.



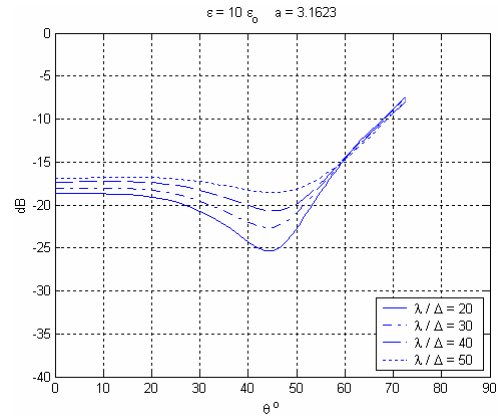
**Figura 4.35** Intensidade relativa do campo  $E_z$  para  $\epsilon = \epsilon_0$ .



**Figura 4.36** Intensidade relativa do campo  $E_z$  para  $\epsilon = 2\epsilon_0$ .



**Figura 4.37** Intensidade relativa do campo  $E_z$  para  $\epsilon = 5\epsilon_0$ .

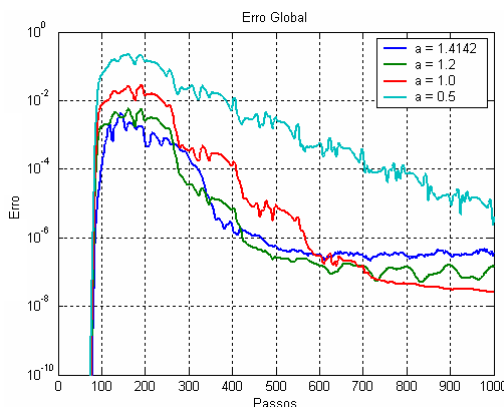


**Figura 4.38** Intensidade relativa do campo  $E_z$  para  $\epsilon = 10\epsilon_0$ .

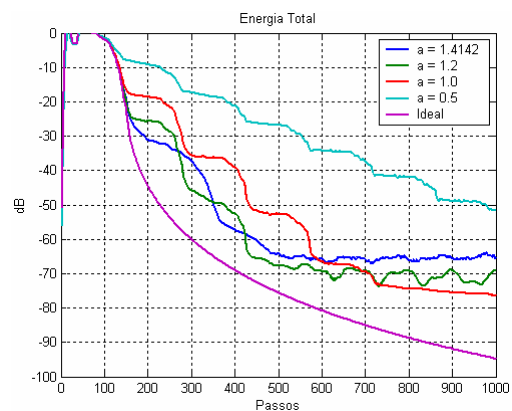
Nos gráficos das Figuras 4.35 – 4.38 é possível observar a diferença entre as fronteiras sem compensação, primeiro grupo, e as fronteiras com compensação, segundo grupo. Com a compensação temporal as fronteiras de Auto Teleportação possuem praticamente a mesma atenuação para qualquer constante dielétrica. Isto mostra que a compensação temporal adapta melhor as fronteiras de Auto Teleportação em relação ao algoritmo original.

## Energia e Erro

Como feito na seção 4.2.1, é possível analisar a absorção de energia e o erro que um domínio truncado apresenta em relação a um domínio ideal. Nestas simulações são utilizados os seguintes parâmetros:  $a = 1,4142$ ,  $a = 1,2$ ,  $a = 1,0$  e  $a = 0,5$ . O domínio computacional possui uma constante dielétrica  $\epsilon = \epsilon_0$ .



**Figura 4.39** Erro global.



**Figura 4.40** Energia no domínio computacional

Nota-se que o erro para  $a = 0$  vai aumentando lentamente a partir do passo 600, o que faz com que o algoritmo divirja para simulações muito longas. Uma forma de contornar este problema é impor um fator de amortecimento  $R \leq 0,95$ , o que, por outro lado, prejudica o desempenho com relação à absorção da onda. Outra possível solução é acrescentar uma fronteira de Mur nos limites do domínio, o que será tratado no capítulo de Simulações Numéricas.

#### 4.4.2 Desvantagem da Compensação Temporal

Nas fronteiras de Auto Teleportação modificadas pela compensação temporal é possível mudar o ângulo de máxima absorção para se adequar a um determinado problema. Isto faz com que este tipo de compensação apresente uma vantagem em relação ao algoritmo original: torna-se possível ajustar o atraso entre os campos copiado e original de acordo com a velocidade de propagação da onda no meio.

Entretanto, há duas desvantagens nesse tipo de compensação. A primeira é que o tamanho do *buffer* utilizado para armazenar os campos copiados de  $S$  para  $S'$  aumenta proporcionalmente com o aumento do valor do atraso  $a$ . Isto provoca um aumento de memória indesejável, principalmente quando se utiliza meios com constantes dielétricas muito diferentes entre si. A segunda é que não se pode utilizar diferentes valores de atraso em uma mesma fronteira de Auto Teleportação para adequá-la a meios diferentes. É então necessário utilizar-se um valor médio para cada meio ou utilizar varias fronteiras de Auto Teleportação, cada uma ajustada para um meio diferente.

### 4.5 Compensação por Equação de Onda

No algoritmo original os campos presentes na posição da superfície  $S$  são inseridos diretamente na posição da superfície  $S'$ , mas como é explicado nas seções anteriores, isso causa um erro no cancelamento da onda. Isto se deve ao fato de que, ao se propagar, a onda apresenta diferentes valores em módulo ao longo de seu trajeto. Este erro pode ser minimizado se soubermos de ante mão qual o valor dos campos da onda que devem ser inseridos na posição da superfície  $S'$ .

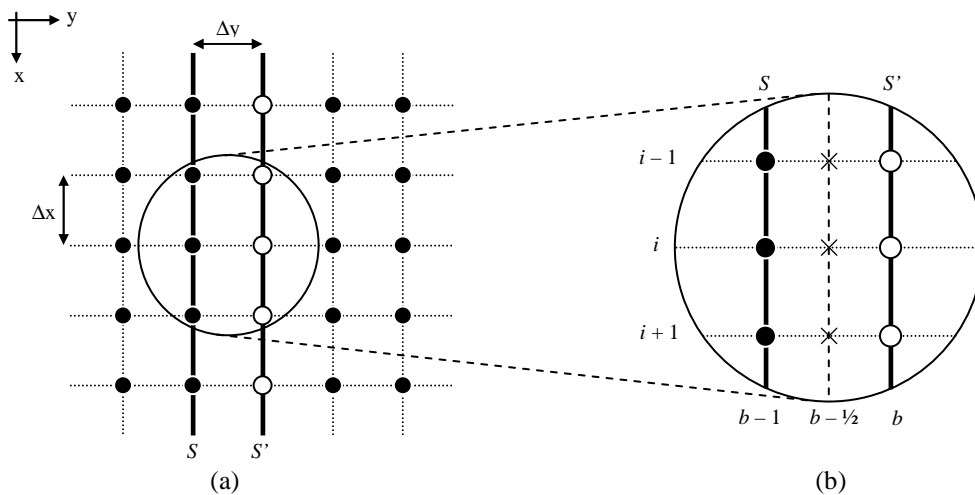
Essa determinação dos campos pode ser obtida através da utilização da equação de onda, mais precisamente através da “Equação de Onda de Sentido Único” (One Way Wave Equation) explicada no Capítulo 2. Esta equação de onda é implementada no

método *FDTD* através das aproximações de Mur ou, para um caso geral, através das aproximações pelo operador de Higdon.

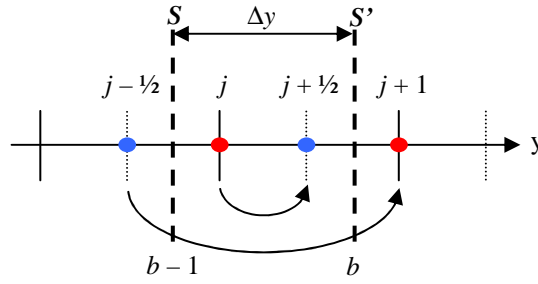
A vantagem de se utilizar este tipo de compensação é que não é necessário termos o tamanho do buffer atrelado ao atraso dos campos copiados. Outra vantagem deste tipo de compensação é que a fronteira de Auto Teleportação se torna adaptada ao meio o qual esta imersa, ou seja, ela apresenta a mesma eficiência quando se tem dois ou mais meios diferentes no mesmo domínio.

#### 4.5.1 Implementação Utilizando a Primeira Aproximação de Mur

A compensação por equação de onda consiste em determinar o valor dos campos de uma onda na posição da superfície  $S'$  (campos indeterminados) através do valor dos campos na posição da superfície  $S$  (campos conhecidos). A Figura 4.41 mostra uma parte do domínio computacional, os círculos vazios representam os campos indeterminados e os círculos cheios representam os campos conhecidos. A superfície  $S$  esta na posição  $b - 1$ , a superfície  $S'$  esta na posição  $b$  e a posição  $b - \frac{1}{2}$  é uma posição auxiliar utilizada no cálculo dos campos como mencionada no capítulo 2.



**Figura 4.41** Esquema de implementação da compensação por equação de onda.



**Figura 4.42** Posicionamento do campo elétrico  $E_z$  (em vermelho) e do campo magnético  $H_x$  (em azul) no domínio computacional.

Os campos elétrico e magnético de uma onda não estão na mesma posição na grade *FDTD*, como mostrado na Figura 4.42, eles estão defasados espacialmente de  $\frac{1}{2}$  passo. Porém, para a implementação desse tipo de compensação é admitido que estes campos estão na mesma posição. Posição  $b - 1$  para os campos tangente a superfície  $S$  e posição  $b$  para os campos tangente a superfície  $S'$ . Com isso é possível aplicar a primeira aproximação de Mur para determinar os campos na superfície  $S'$ . Estes campos são obtidos pelas equações

$$E_z|_{i,b}^{n+1} = E_z|_{i,b-1}^n + m \cdot (E_z|_{i,b-1}^{n+1} - E_z|_{i,b}^n); \quad (4.34)$$

$$H_x|_{i,b}^{n+1} = H_x|_{i,b-1}^n + m \cdot (H_x|_{i,b-1}^{n+1} - H_x|_{i,b}^n); \quad (4.35)$$

sendo  $m = \left( \frac{v \cdot \Delta t - \Delta y}{v \cdot \Delta t + \Delta y} \right)$ .

Como  $E_z|_{i,b}^{n+1}$  depende de  $E_z|_{i,b}^n$  e o valor destes campos na posição  $b$  são indeterminados, como dito anteriormente, é necessário que haja uma recursão nas equações (4.34) e (4.35). Isto faz com que seja necessário a criação de um buffer para o armazenamento dos campos calculados. Sendo assim, estas equações podem ser reescritas na forma

$$E_{buff}|_i = E_z|_{i,b-1}^n + m \cdot (E_z|_{i,b-1}^{n+1} - E_{buff}|_i); \quad (4.36)$$

$$H_{buff}|_i = H_x|_{i,b-1}^n + m \cdot (H_x|_{i,b-1}^{n+1} - H_{buff}|_i); \quad (4.37)$$

e os valores obtidos de  $E_{buff}$  e  $H_{buff}$  são utilizados para gerar as correntes superficiais  $K_e$  e  $K_m$  da equações (4.7) e (4.8).

Novamente é necessário introduzir um buffer, mas diferentemente do buffer da compensação temporal, o tamanho desse buffer não está relacionado com o atraso da onda incidente.

#### 4.5.2 Implementação Utilizando a Aproximação de Higdon

A aproximação de Higdon é uma generalização das Aproximações de Mur. Neste tipo de aproximação é possível definir um ângulo de máxima atenuação. Higdon propôs que uma onda pode ser decomposta por uma seqüência linear de planos de onda com diferentes ângulos de incidência  $\alpha_j$ ,

$$\left[ \prod_{j=1}^p \left( (\cos \alpha_j) \frac{\partial}{\partial t} \pm c \frac{\partial}{\partial y} \right) \right] W = 0; \quad (4.38)$$

sendo,  $p$  o número de planos de onda..

Trabalhos feitos por Higdon mostram que não há muita diferença na absorção da onda quando se utiliza a equação (4.38) com  $p > 2$ . Por isso, neste tipo de compensação é utilizado  $p = 1$ . Nota-se que se  $p = 1$  e  $\alpha_j = 0$ , (4.38) assume a forma da primeira aproximação de Mur.

Para implementarmos (4.38) basta utilizar as mesmas equações (4.36) e (4.37) desenvolvidas anteriormente porém a constante  $m$  deve ser reescrita na forma

$$m = \left( \frac{v \cdot \Delta t - \Delta y \cdot \cos(\alpha)}{v \cdot \Delta t + \Delta y \cdot \cos(\alpha)} \right); \quad (4.39)$$

sendo,  $\alpha$  um ângulo pré-determinado para máxima absorção.

#### 4.5.3 Resultados da Compensação por Equação de Onda

As simulações a seguir mostram os resultados obtidos com compensação por aproximação de Higdon, pois é mais genérica. Nelas são utilizados dois valores para o ângulo  $\alpha$ . O primeiro ângulo utilizado é  $0^\circ$ , quando a equação de Higdon se torna igual a de Mur. O segundo ângulo utilizado é  $45^\circ$ , que representa o ângulo obtido pela técnica original sem compensação. As Figuras 4.43 e 4.44 apresentam a atenuação obtida em um domínio com constante dielétrica  $\epsilon_0$ . As Figuras 4.45 e 4.46 apresentam a atenuação obtida para um domínio com constante dielétrica  $\epsilon = 5\epsilon_0$ .



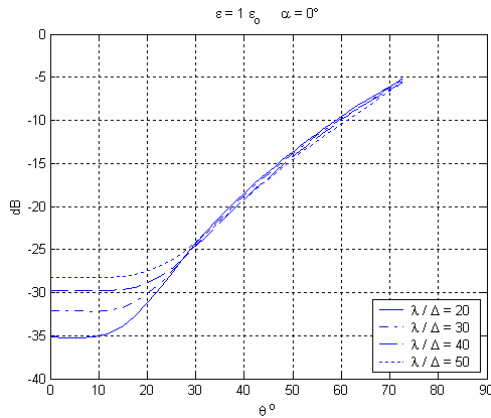


Figura 4.43 Intensidade relativa do campo  $E_z$

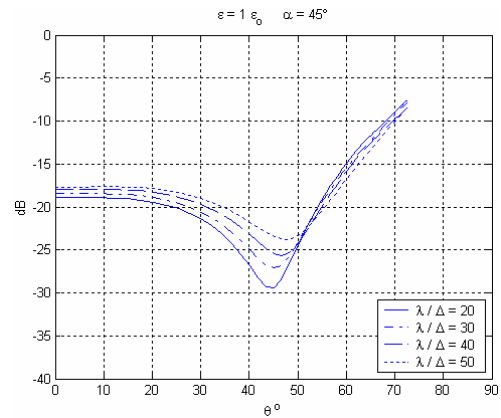


Figura 4.44 Intensidade relativa do campo  $E_z$

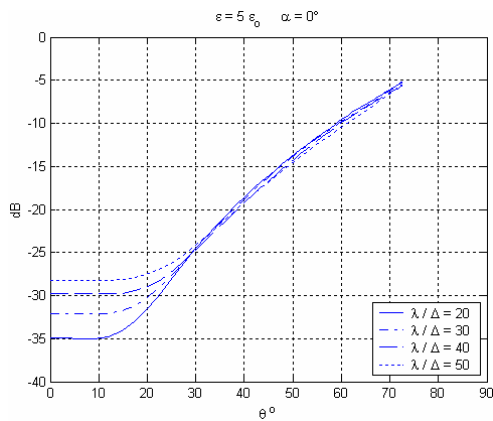


Figura 4.45 Intensidade relativa do campo  $E_z$

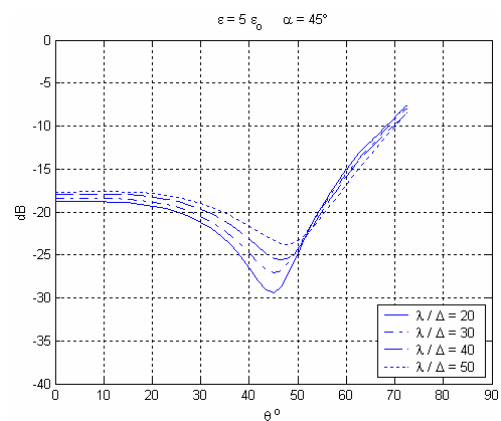
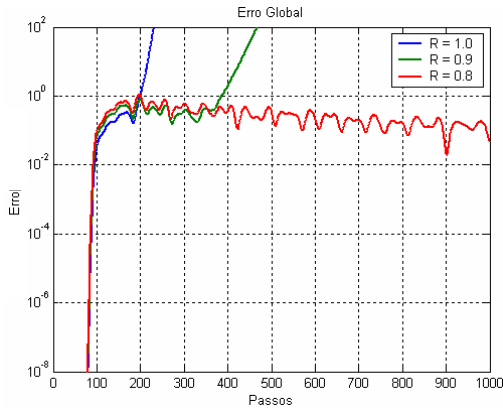


Figura 4.46 Intensidade relativa do campo  $E_z$

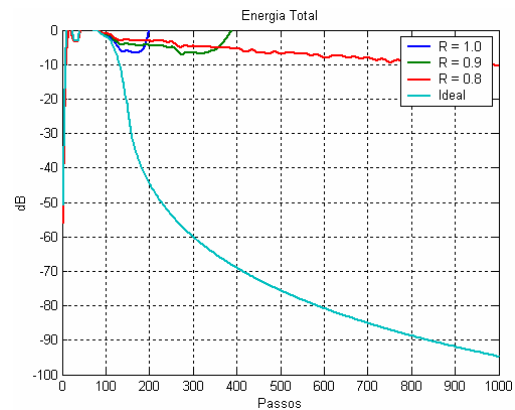
Estas figuras mostram a eficiência deste tipo de compensação para qualquer tipo de domínio simulado. Esta compensação se adapta a cada tipo de problema, o que é diferente da compensação temporal que necessita de um ajuste prévio para cada tipo de domínio simulado.

## Energia e Erro

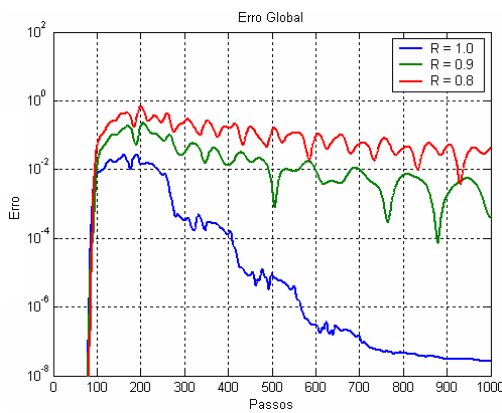
Como feito na seção 4.2.1, é possível analisar a absorção de energia e o erro que um domínio truncado apresenta em relação a um domínio ideal. A simulações a seguir apresentam os erros globais e as energias absorvidas para uma compensação utilizando a primeira aproximação de Higdon para  $\alpha = 0^\circ$  e  $\alpha = 45^\circ$ . Nestas simulações é utilizado um domínio com um constante dielétrica  $\epsilon = \epsilon_0$ .



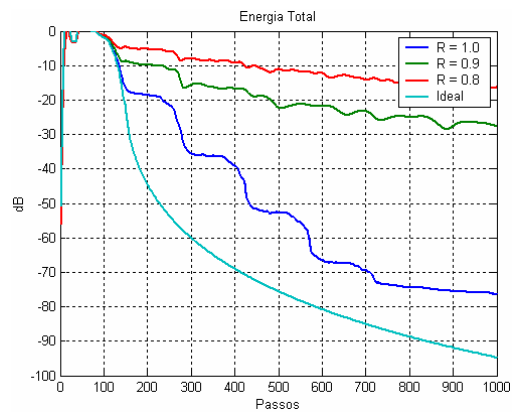
**Figura 4.47 Erro global para  $\alpha = 0^\circ$ .**



**Figura 4.48 Energia no domínio computacional para  $\alpha = 0^\circ$ .**



**Figura 4.49 Erro global para  $\alpha = 45^\circ$ .**



**Figura 4.50 Energia no domínio computacional para  $\alpha = 45^\circ$ .**

A Figura 4.47 apresenta o erro global para  $\alpha = 0^\circ$ . Nota-se que a estabilidade numérica é obtida com um fator de amortecimento máximo de  $R = 0,80$  e que acima desse valor, o algoritmo diverge. Esta divergência ocorre nos cantos do domínio computacional, ou seja, na junção entre duas fronteiras de Auto Teleportação. Este problema pode ser minimizado com a inserção de uma fronteira de Mur nas bordas do domínio computacional, como será comentado no Capítulo 5. Para um ângulo  $\alpha = 45^\circ$ , conforme a Figura 4.49, a estabilidade numérica é conseguida para um fator de amortecimento  $R = 1,0$ .

## 4.6 Conclusão

A técnica da Auto Teleportação de Campos permite a implementação de um novo tipo de condição de fronteira de absorção que não exige uma carga computacional muito elevada, como é o caso da técnica *PML*.

Devido à sensibilidade dessa técnica às discretizações espaciais ( $\Delta x$ ,  $\Delta y$  e  $\Delta z$ ), à discretização temporal ( $\Delta t$ ) e à velocidade da onda ( $v$ ), é necessário introduzir-se compensações para melhorar a absorção das ondas incidentes.

A compensação espacial surge da necessidade de se aproximar as superfícies  $S$  e  $S'$  e a compensação temporal da necessidade de sincronizar as fases das ondas copiada e original. Cada compensação possui vantagens e desvantagens. A compensação por equação de onda é a mais genérica de todas, pois ela faz ao mesmo tempo uma compensação espacial e temporal das ondas copiada e original. Isso cria a vantagem de se adaptar ao domínio simulado, utilizando para isso apenas algumas operações adicionais no algoritmo *FDTD*.

# 5 Resultados Numéricos

## 5.1 Introdução

Neste Capítulo, inicialmente será dada continuidade ao estudo iniciado no Capítulo 4 sobre domínios truncados pelas técnicas de Auto Teleportação. Esse tipo de truncamento pode ser entendido como Camadas de Auto Teleportação, ou em inglês, “Self Teleportation Layer” (*STL*).

A seguir são apresentadas comparações entre as *STL* e as demais técnicas de truncamento. Essas comparações são realizadas com base nos erros globais e nas energias presentes em cada domínio computacional. São também apresentadas tabelas do número de operações matemáticas necessárias para a execução de cada algoritmo de truncamento. Completando as comparações, são apresentados resultados obtidos para dois circuitos de *RF*: uma microlinha de transmissão e um filtro planar e resultados obtidos para uma simulação de um cilindro iluminado por uma onda plana.

## 5.2 Estudo da *STL* em Domínios Bidimensionais

Este item apresenta o comportamento de uma onda refletida nas fronteiras de um domínio computacional truncado pela técnica *STL*, com e sem compensação. Para isso, é levantado o erro global, a energia no domínio computacional e o perfil de atenuação das ondas incidentes nas *STL*. Nessas simulações a constante dielétrica do meio é mantida em  $\epsilon = \epsilon_0$ .

### 5.2.1 *STL* sem Compensação

As análises apresentadas neste item utilizam a técnica de Auto Teleportação sem compensação. Estas análises apresentam a influência do número de fronteiras de Auto Teleportação na absorção de uma onda incidente. As Figuras 5.1 e 5.2 apresentam respectivamente o erro global e a energia existente nos domínios computacionais. As Figuras 5.3 – 5.6 apresentam a atenuação de uma onda incidente para no máximo três fronteiras de Auto Teleportação. O número de fronteiras de Auto Teleportação está representado por  $N_f$ .

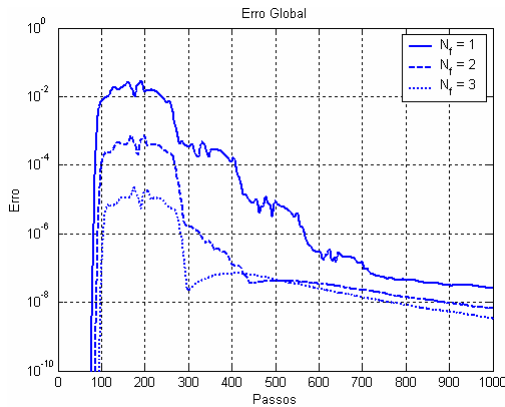


Figura 5.1 Erro global.

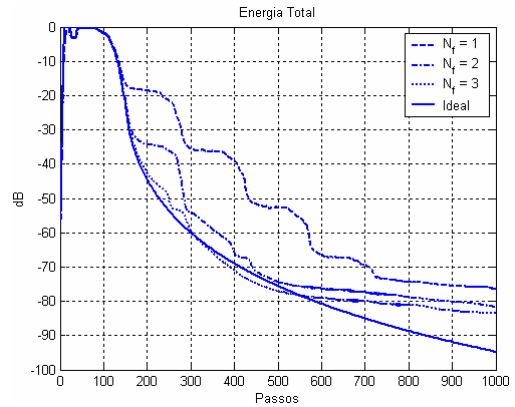


Figura 5.2 Energia no domínio computacional.

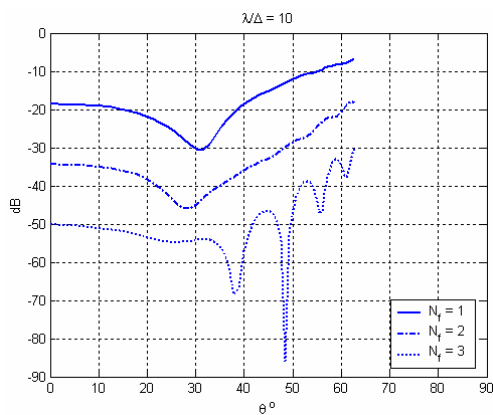


Figura 5.3 Intensidade relativa do campo  $E_z$  para  $\lambda/\Delta = 10$ .

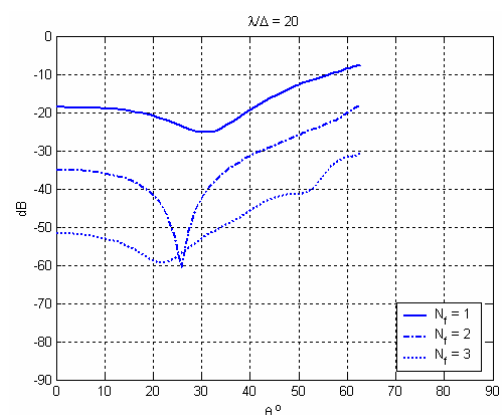


Figura 5.4 Intensidade relativa do campo  $E_z$  para  $\lambda/\Delta = 20$ .

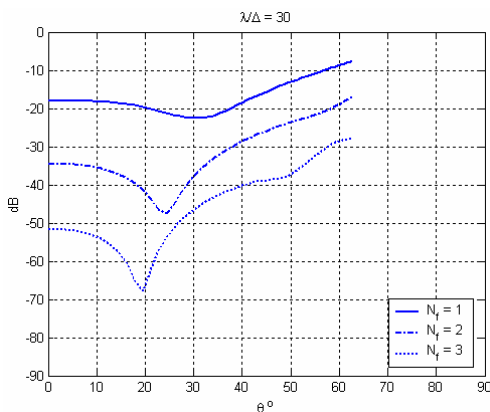


Figura 5.5 Intensidade relativa do campo  $E_z$  para  $\lambda/\Delta = 30$ .

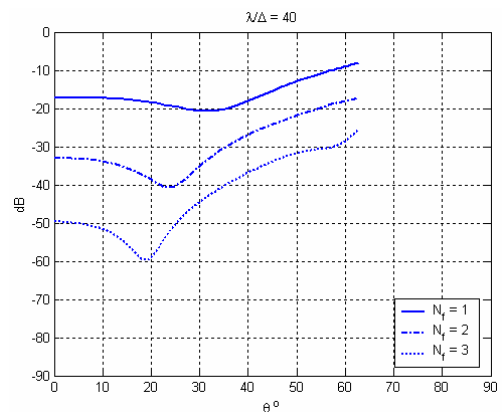


Figura 5.6 Intensidade relativa do campo  $E_z$  para  $\lambda/\Delta = 40$ .

Observa-se que quanto mais camadas *STL* são inseridas no domínio, maior é a atenuação da onda incidente. Observa-se também que, em geral, essa técnica perde eficiência com o aumento do comprimento de onda. O fator de amortecimento  $R = 1,0$  é suficiente para garantir a estabilidade do algoritmo nessas simulações.

## 5.2.2 STL com Compensações Temporais Diferentes

No Capítulo 4, item 4.4, foi apresentada a influência da compensação temporal nas ondas refletidas por uma STL. Pelos resultados apresentados, é possível observar a mudança do ângulo de máxima absorção.

Este item possui a finalidade de apresentar a influência da combinação de compensações temporais nas ondas refletidas por uma STL, implementada com três fronteiras com compensações temporais diferentes. Para isso, são utilizados os atrasos  $a = 1,4142$ ,  $a = 1,0$  e  $a = 0,5$ , escolhidos para causar um máximo de atenuação nos ângulos  $0^\circ$ ,  $45^\circ$  e  $70^\circ$ , cobrindo, assim, uma ampla faixa de ângulos de incidência de onda.

A Figura 5.7 apresenta o erro global entre o domínio truncado e um domínio ideal. A Figura 5.8 apresenta a energia existente nos domínios computacionais e a Figura 5.9 apresenta o perfil de atenuação para essa STL.

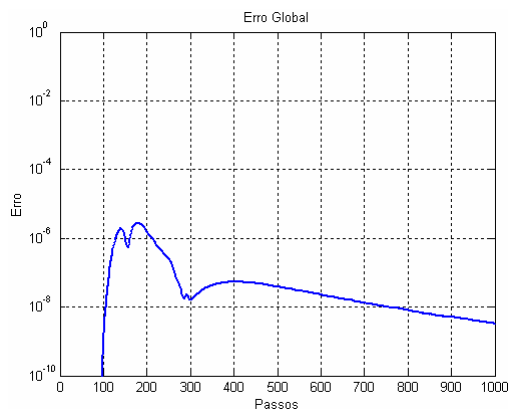


Figura 5.7 Erro global.

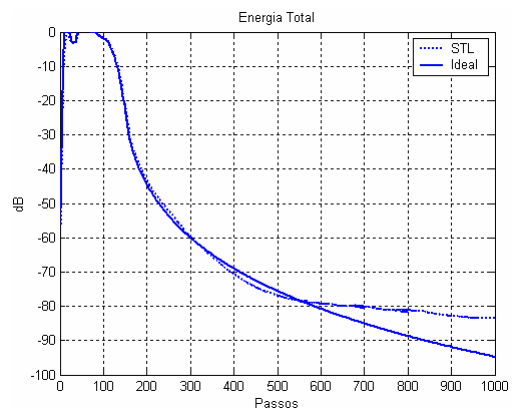


Figura 5.8 Energia no domínio computacional.

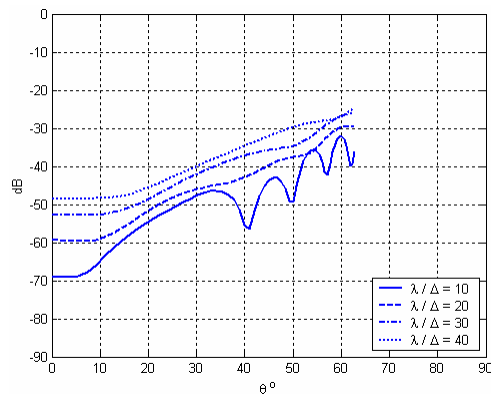


Figura 5.9 Intensidade relativa do campo  $E_z$ .

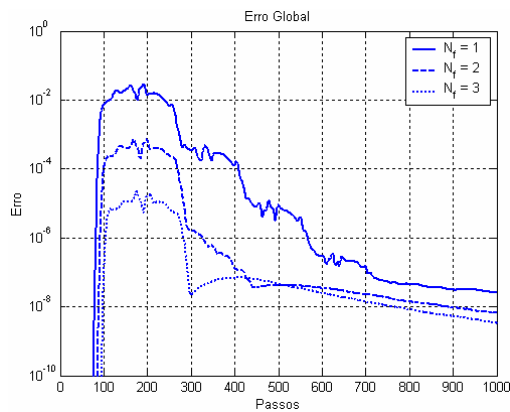
Comparando-se os três últimos gráficos com os gráficos das Figuras 5.1 – 5.6, para  $N_f = 3$ , é observado que é possível melhorar a absorção de uma onda apenas ajustando-se de forma adequada as compensações temporais. Estas devem ser ajustadas em função da constante dielétrica do meio, da velocidade de onda e dos parâmetros de discretizações temporal e espacial, como demonstrado em (4.19).

### 5.2.3 STL com Compensações por Equação de Onda

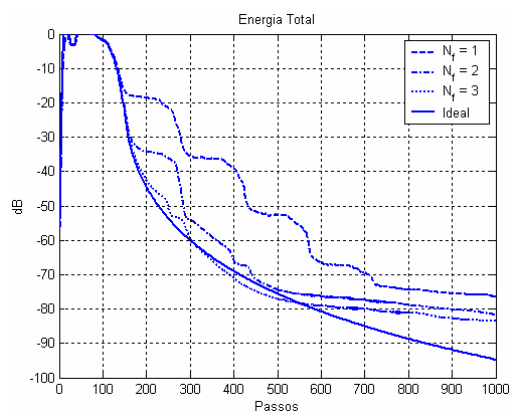
A próxima análise apresenta o comportamento de uma STL com compensação por equação de onda. Nas simulações são comparadas STLs implementadas com 1, 2 e 3 fronteiras.

Como foi visto no Capítulo 4, item 4.5, este tipo de compensação é altamente instável para  $\alpha < 45^\circ$  sendo necessário impor  $R < 1,0$  para estabilizar as simulações. Por outro lado, isto reduz a eficiência de absorção da STL. Portanto as simulações foram limitadas em  $\alpha = 45^\circ$  e  $R = 1,0$ . As Figuras 5.10 e 5.11 apresentam respectivamente o erro global e a energia presente nos domínios computacionais.

As Figuras 5.12 – 5.15 apresentam o perfil de atenuação da onda para essa condição. O número de fronteiras STLs está representado por  $N_f$ .



**Figura 5.10 Erro global.**



**Figura 5.11 Energia no domínio computacional.**

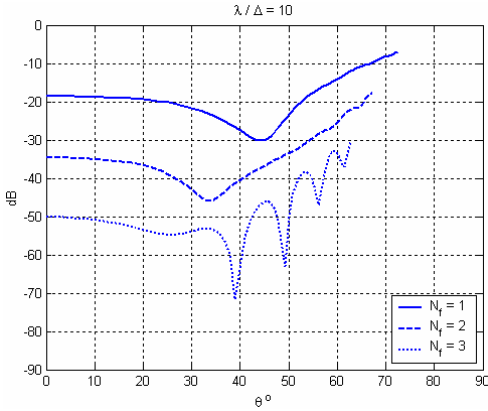


Figura 5.12 Intensidade relativa do campo  $E_z$  para  $\lambda/\Delta = 10$ .

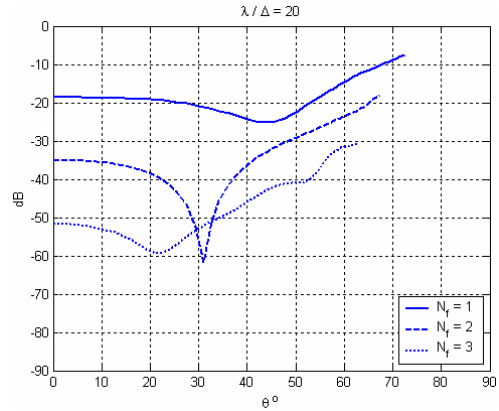


Figura 5.13 Intensidade relativa do campo  $E_z$  para  $\lambda/\Delta = 20$ .

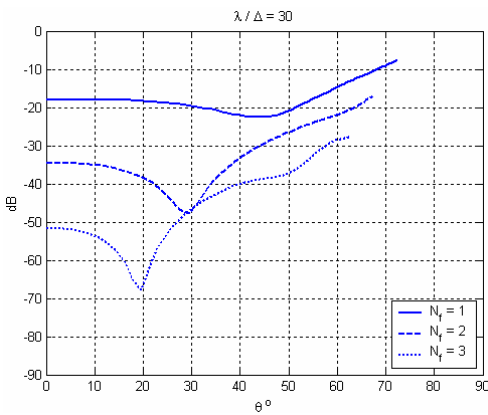


Figura 5.14 Intensidade relativa do campo  $E_z$  para  $\lambda/\Delta = 30$ .

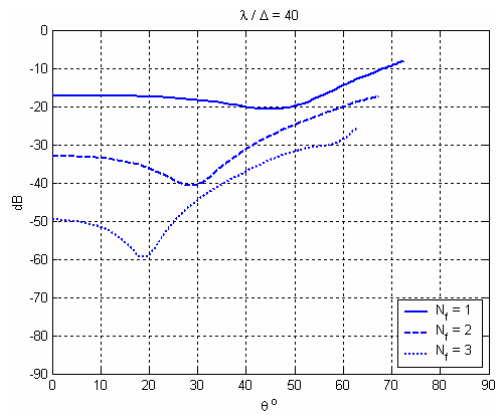


Figura 5.15 Intensidade relativa do campo  $E_z$  para  $\lambda/\Delta = 40$ .

É observado que as *STLs* com compensação por equação de onda apresentam aproximadamente o mesmo comportamento das *STLs* sem compensação se o meio for homogêneo. Porém este tipo de compensação é independente dos parâmetros de discretização espacial, temporal e da velocidade de onda como mostrado no Capítulo 4. Isso representa uma significativa vantagem no caso de meios não homogêneos e grades não uniformes.

#### 5.2.4 *STL* Finalizada por Fronteira de Re-Radiação

As *STLs* apresentadas anteriormente são terminadas simplesmente pelas fronteiras do domínio computacional, ou seja, os campos são nulos no exterior do domínio. Porém, pode-se terminá-las com um outro tipo de fronteira de re-radiação, tais



como Mur ou Higdon. Permitindo um maior grau de atenuação sem adicionar mais células.

As simulações a seguir apresentam o comportamento de uma *STL* sem compensação, finalizada por uma fronteira realizada pela primeira aproximação de Mur. As Figuras 5.16 e 5.17 apresentam respectivamente o erro global e a energia presente nos dois domínios computacionais. As Figuras 5.18 – 5.21 apresentam o perfil de atenuação da onda.

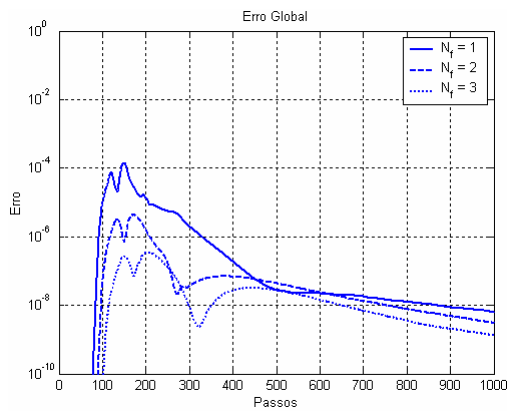


Figura 5.16 Erro global.

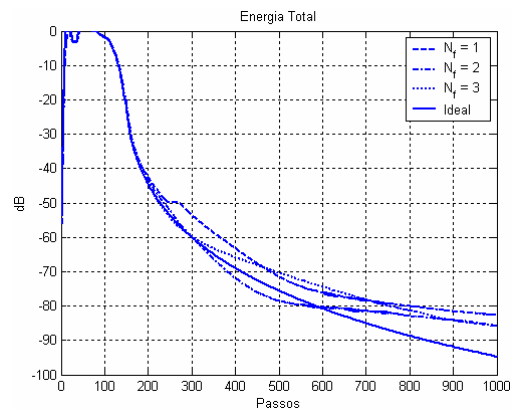


Figura 5.17 Energia no domínio computacional.

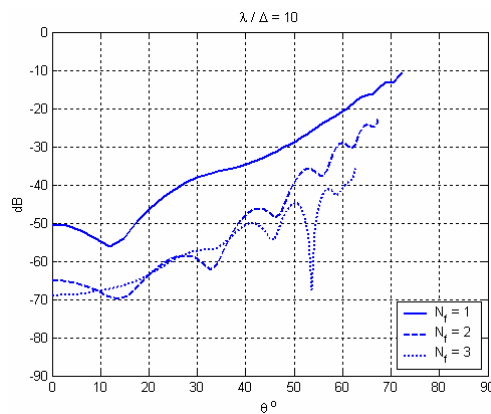


Figura 5.18 Intensidade relativa do campo  $E_z$  para  $\lambda/\Delta = 10$ .

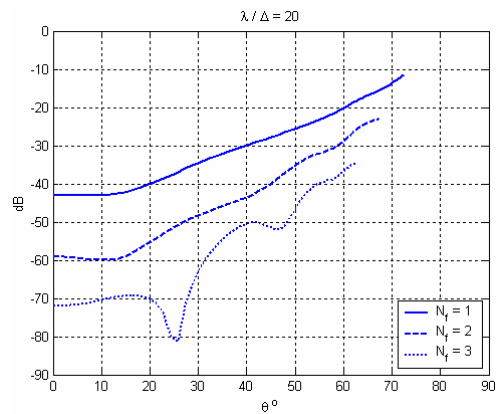
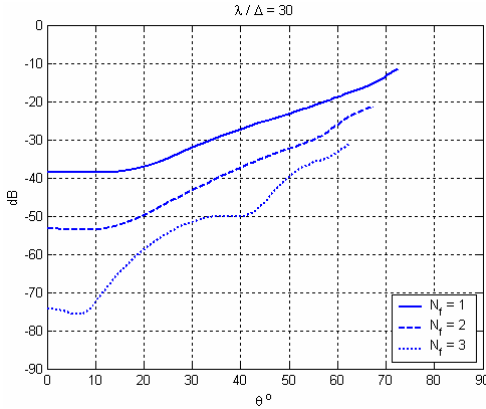
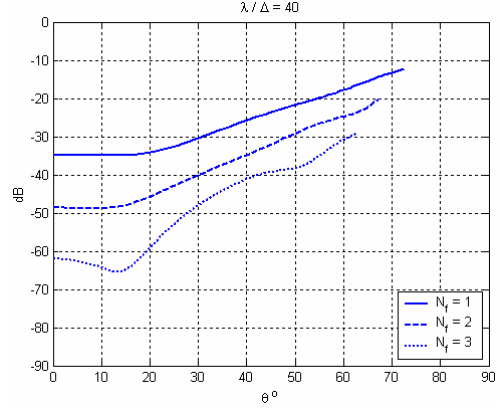


Figura 5.19 Intensidade relativa do campo  $E_z$  para  $\lambda/\Delta = 20$ .



**Figura 5.20** Intensidade relativa do campo  $E_z$  para  $\lambda/\Delta = 30$ .



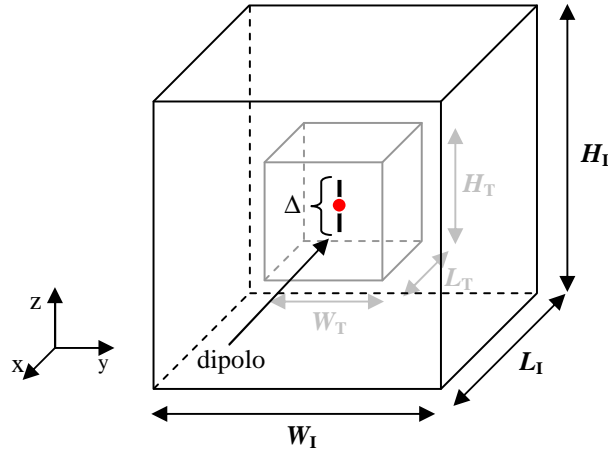
**Figura 5.21** Intensidade relativa do campo  $E_z$  para  $\lambda/\Delta = 40$ .

Dessas figuras é possível observar que a finalização das *STLs* pela primeira aproximação de Mur resulta em maior atenuação quando comparada com as *STLs* sem essa finalização. A atenuação extra é conseguida pois a finalização por uma fronteira de re-radiação elimina boa parte das ondas que seriam re-inseridas ao domínio no sentido inverso de propagação nas fronteiras de Auto Teleportação.

### 5.3 Estudo da *STL* em Domínios Tridimensionais

As simulações até agora apresentadas foram realizadas em domínios bidimensionais. Como a maioria das aplicações práticas requerem domínios tridimensionais, é necessário verificar o comportamento das *STLs* nesses domínios.

Como nas simulações em duas dimensões, o domínio tridimensional truncado ( $\Omega_T$ ) é comparado com um domínio tridimensional ideal ( $\Omega_I$ ). Para a realização das simulações são construídos um domínio ideal ( $\Omega_I$ ), com dimensões  $\{300 \times 300 \times 300\}$  células e um domínio truncado ( $\Omega_T$ ) com dimensões  $\{50 \times 50 \times 50\}$  células, ambos com discretização espacial  $\Delta x = \Delta y = \Delta z = \Delta$ . A fonte de excitação utilizada é um dipolo elementar de comprimento  $\Delta$ , como apresentado na Figura 5.22. A forma de onda de excitação é descrita por (5.1).



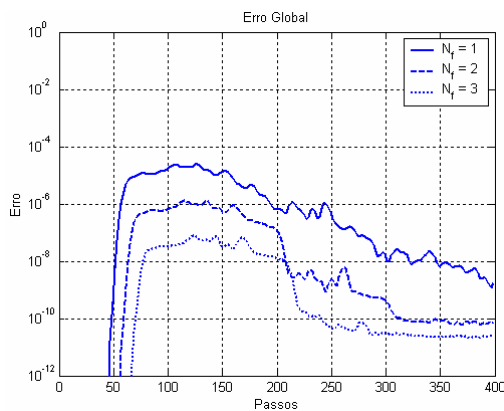
**Figura 5.22 Ilustração dos domínios computacionais tridimensionais.**

$$E_z(n) = \begin{cases} \text{sen}\left(\frac{2\pi}{60}n\right) & ; \text{ para } 0 \leq n \leq 60 \\ 0 & ; \text{ para } n > 60 \end{cases}, \quad (5.1)$$

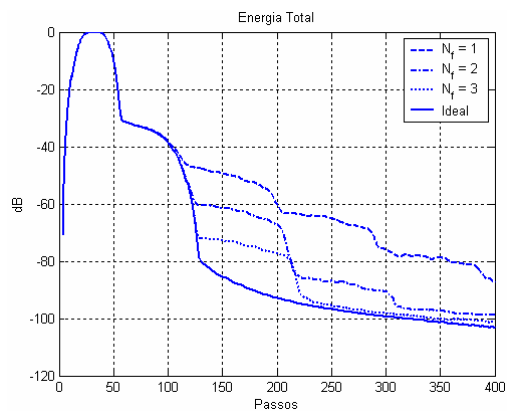
sendo,  $n$  o passo temporal.

### 5.3.1 STL sem Compensação

Como realizado anteriormente, esse item apresenta o comportamento da *STL* implementada sem compensações. As Figuras 5.23 e 5.24 apresentam respectivamente o erro global e a energia presente nos domínios computacionais, para uma *STL* implementada com 1, 2 e 3 fronteiras de Auto Teleportação. O número de fronteiras de Auto Teleportação está representado por  $N_f$ .



**Figura 5.23 Erro global.**

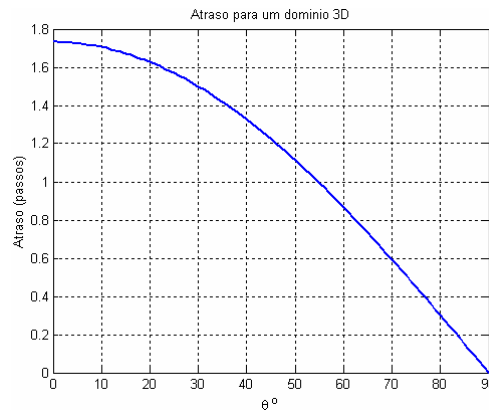


**Figura 5.24 Energia no domínio computacional.**

Novamente é observada a influência do número de fronteiras de Auto Teleportação na atenuação de uma incidente.

### 5.3.2 STL com Compensação Temporal

Este item apresenta o comportamento de uma STL com compensação temporal. Como visto no Capítulo 4, a compensação temporal ( $a$ ) é dependente da dimensão do domínio computacional através da discretização  $\Delta t$ . Para a realização de simulações em três dimensões, os valores de  $a$  podem ser retirados do gráfico apresentado na Figura 5.25. Este gráfico é levantado para um domínio tridimensional com constante dielétrica  $\epsilon = \epsilon_0$  através de (4.19).



**Figura 5.25** Curvas de atraso em função do ângulo de incidência para um domínio tridimensional.

Para se analisar o comportamento desse tipo de compensação, são escolhidos os valores  $a = 1,0$ , que representa o algoritmo sem compensação;  $a = 1,7320$ , que resulta em um máximo da atenuação em  $\theta = 0$  e  $a = 1,4142$ , que resulta em um máximo de atenuação para  $\theta = 0$  em domínios bidimensionais. As Figuras 5.26 e 5.27 apresentam respectivamente o erro global e a energia presente nos domínios computacionais. Nessas simulações é utilizado o fator de amortecimento  $R = 1,0$ .

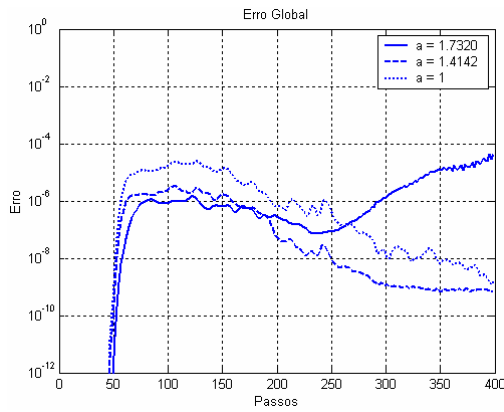


Figura 5.26 Erro global.

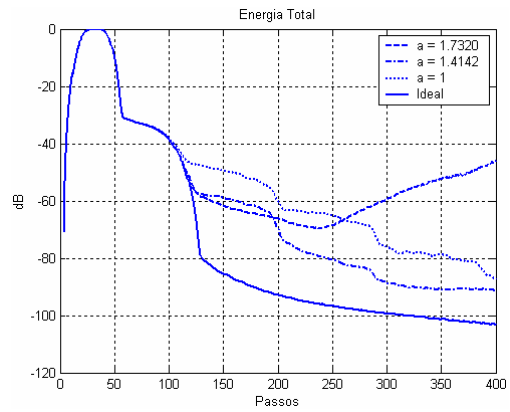


Figura 5.27 Energia no domínio computacional.

Assim como em duas dimensões, o algoritmo diverge quando é imposta uma compensação temporal ajustada para  $\theta = 0$ . O fator de amortecimento  $R = 1,0$  garante a estabilidade do sistema para os outros atrasos, exceto para o atraso  $a = 1,7320$ . Novamente, este problema é solucionado impondo  $R < 1,0$ , apesar da redução da eficiência de absorção.

### 5.3.3 STL com Compensação por Equação de Onda

Os resultados a seguir apresentam o comportamento da STL com compensação por equação de onda. Neles são apresentados o erro global e a energia no domínio computacional em relação ao número de fronteiras de Auto Teleportação. Essas fronteiras são implementadas com os parâmetros  $\alpha = 45^\circ$  e  $R = 0,99$ , escolhidos de forma a se ter uma eficiência máxima de absorção da onda. As Figuras 5.41 e 5.42 apresentam respectivamente o erro e a energia nos domínios.

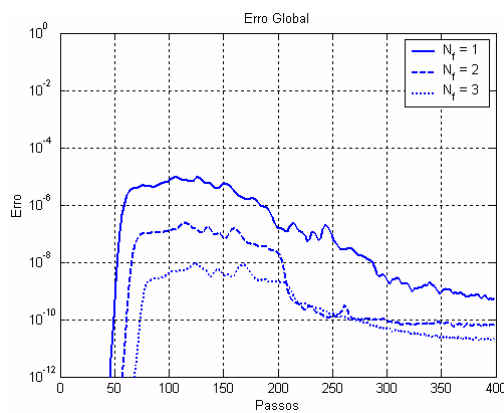


Figura 5.28 Erro global.

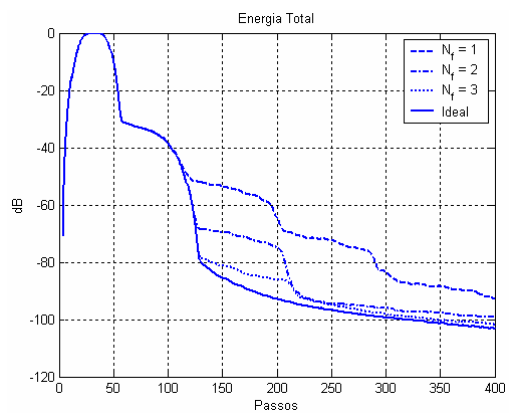


Figura 5.29 Energia no domínio computacional.

Dos gráficos é observado que três fronteiras de Auto Teleportação apresentam um resultado pior que duas fronteiras. Isso é causado por estimativas erradas na compensação por equação de onda. Novamente esse problema é solucionado pela redução do fator de amortecimento  $R$ .

### 5.3.4 STL Finalizada por Fronteira de Re-Radiação

Estas análises apresentam o comportamento de um domínio truncado por um conjunto de fronteiras de Auto Teleportação finalizado por uma fronteira de re-radiação, implementada com a primeira aproximação de Mur. As Figuras 5.30 e 5.31 apresentam, respectivamente, o erro global e a energia no domínio computacional.

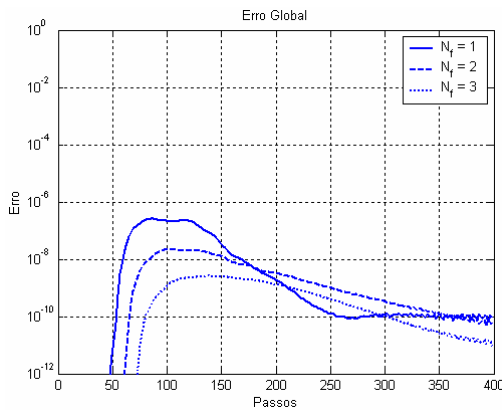


Figura 5.30 Erro global.

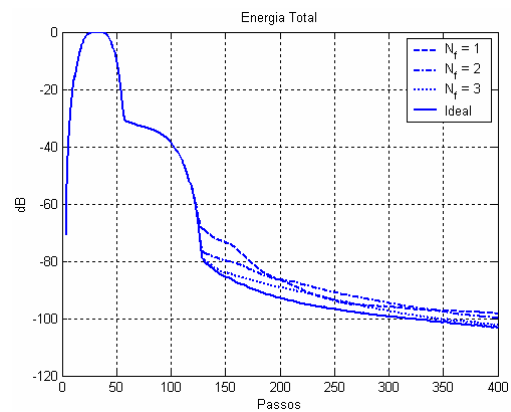


Figura 5.31 Energia no domínio computacional.

Como ocorre para domínios bidimensionais, a finalização pela fronteira de re-radiação resulta em maior atenuação devido a eliminação de parte das ondas que seriam re-inseridas no domínio computacional.

## 5.4 Comparação entre Técnicas de Absorção

Este item apresenta comparações realizadas entre os três tipos de técnicas de absorção descritas nos capítulos anteriores. Estas comparações têm por finalidade avaliar a atenuação, o erro global e a carga computacional exigida por estas técnicas.

### 5.4.1 Comparação em Domínios Bidimensionais

Os gráficos das Figuras 5.32 e 5.33 apresentam respectivamente o erro global e a energia presente nos domínios computacionais. Estes domínios são truncados pelas técnicas de Mur (primeira e segunda aproximação), *PML* (10 células), e *STL* com compensação por equação de onda e finalização por fronteira de re-radiação. O número de fronteiras *STL* é escolhido de forma a ocupar o mesmo número de células que a técnica *PML*. A *STL* utilizada é implementada com os parâmetros  $\alpha = 45^\circ$  e  $R = 0,99$ . Esses dois parâmetros foram obtidos através de simulações para apresentar máxima eficiência de atenuação da onda.

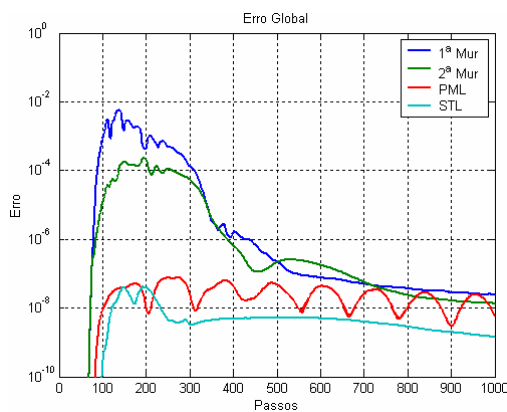


Figura 5.32 Erro global.

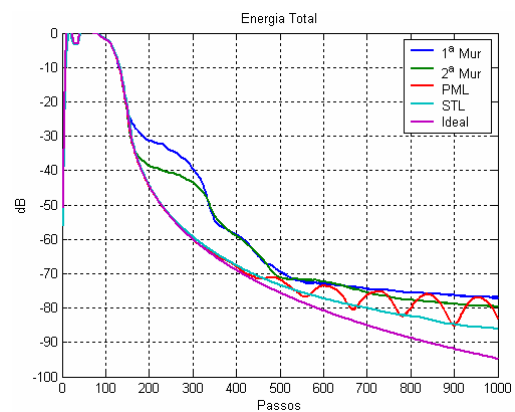


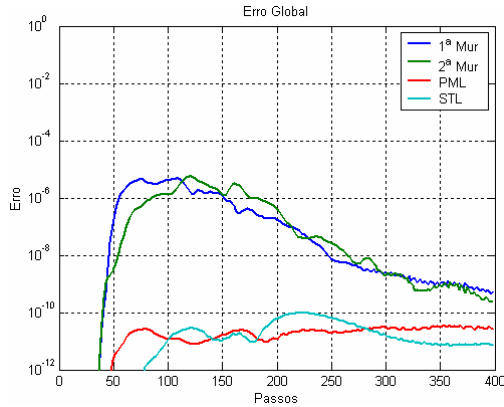
Figura 5.33 Energia no domínio computacional.

Dos gráficos é possível observar que um domínio truncado com técnicas *PML* e *STL* possuem erro menor do que um domínio truncado com as técnicas de Mur. Além disso a técnica *STL* possui atenuação equivalente a técnica *PML*. A técnica *PML* pode apresentar atenuações maiores, porém é preciso aumentar a largura (células) do material absorvedor, o que por consequência leva a um aumento na carga computacional.

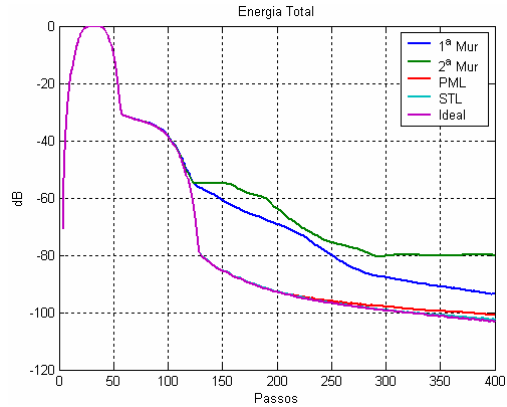
### 5.4.2 Comparação em Domínios Tridimensionais

Neste item é repetido o mesmo procedimento do item 5.4.1, porém para um domínio tridimensional. As Figuras 5.34 e 5.35 apresentam o erro global e a energia presente nos domínios computacionais. Estes domínios foram truncados pelas técnicas de Mur (primeira e segunda aproximação), *PML* (10 células) e *STL* com compensação

por equação de onda e finalizada por fronteira de re-radiação. A *STL* é implementada com os parâmetros  $\alpha = 45^\circ$  e  $R = 0,95$ . Novamente esses parâmetros foram obtidos por simulação para apresentar a máxima eficiência de absorção da onda.



**Figura 5.34 Erro global.**



**Figura 5.35 Energia no domínio computacional.**

Novamente é notado que a técnica *STL* possui uma atenuação equivalente à técnica *PML*.

### 5.4.3 Carga Computacional

Um aspecto importante de comparação entre as técnicas de absorção, além da capacidade de atenuação, é o número de operações matemáticas necessárias efetuadas por elas. Se um algoritmo necessitar de muitas operações matemáticas, o seu tempo de processamento poderá ser muito grande. Este tempo é diretamente proporcional a quantidade de células presentes no domínio computacional.

Neste item são apresentadas comparações entre os números de operações matemáticas necessárias para cada fronteira de absorção. Estas fronteiras são construídas em um domínio tridimensional hexaédrico, como mostrado na Figura 5.36. As fronteiras desse domínio estão posicionadas nas faces do hexaedro, as arestas são as interseções de duas fronteiras e os vértices são a interseções de três fronteiras. Dependendo da técnica de absorção utilizada torna-se necessário tratar essas interseções, como nas técnicas *PML* e segunda aproximação de Mur. As Tabela 5.1 – 5.6 apresentam o número de operações matemáticas necessárias para cada técnica.



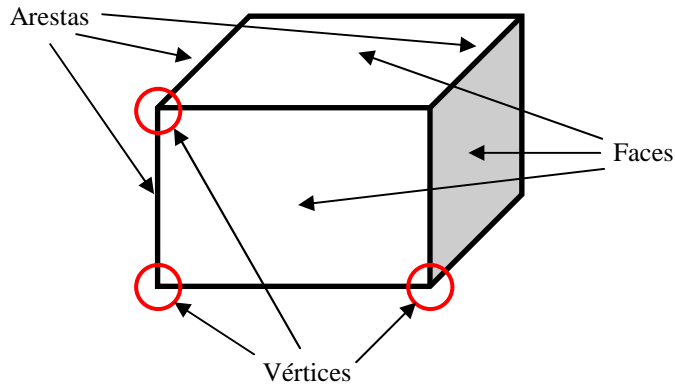


Figura 5.36 Domínio computacional hexaédrico.

Tabela 5.1 Operações por célula na primeira aproximação de Mur.

Primeira Aproximação de Mur	número de operações do tipo ( $\times$ , $\div$ )	número de operações do tipo ( $+$ , $-$ )
6 Faces	12	24

Tabela 5.2 Operações por célula na segunda aproximação de Mur.

Segunda Aproximação de Mur	número de operações do tipo ( $\times$ , $\div$ )	número de operações do tipo ( $+$ , $-$ )
6 Faces	96	204
12 Arestas	36	72
Total	132	276

Tabela 5.3 Operações por célula em uma camada PML.

PML (1 camada)	número de operações do tipo ( $\times$ , $\div$ )	Número de operações do tipo ( $+$ , $-$ )
6 Faces	120	168
12 Arestas	360	432
8 Vértices	144	144
Total	624	744

Tabela 5.4 Operações por célula na STL sem compensação.

Auto Teleportação de Campos (1 fronteira)	número de operações do tipo ( $\times$ , $\div$ )	número de operações do tipo ( $+$ , $-$ )
6 Faces	24	24

**Tabela 5.5 Operações por célula na *STL* com compensação temporal.**

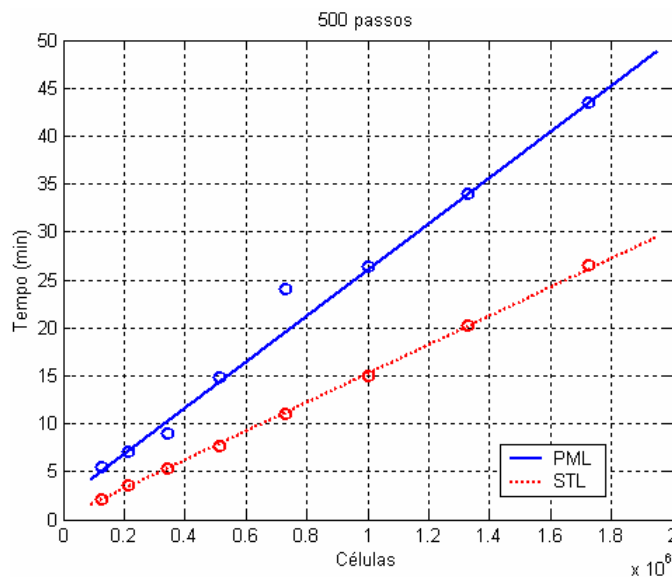
Auto Teleportação de Campos (1 fronteira)	número de operações do tipo ( $\times$ , $\div$ )	Número de operações do tipo ( $+$ , $-$ )
6 Faces	72	96

**Tabela 5.6 Operações por célula na *STL* com compensação por equação de onda.**

Auto Teleportação de Campos (1 fronteira)	número de operações do tipo ( $\times$ , $\div$ )	Número de operações do tipo ( $+$ , $-$ )
6 Faces	48	72

O número de operações matemáticas necessárias na *STL* com compensação espacial é a mesma que na *STL* sem compensação. Porém, a carga computacional dependerá do tamanho das células não uniformes, ou seja, quanto menor o tamanho delas, maior será o tempo de simulação de acordo com a condição de estabilidade de Courant.

Para ilustrar quanto estas operações influem no tempo de simulação, na Figura 5.37 são apresentados gráficos dos tempos de simulação das técnicas *PML* e *STL*. Estes gráficos relacionam o tempo de simulação despendido em relação ao número de células do domínio computacional. Os tempos foram obtidos utilizando-se um domínio tridimensional de formato cúbico. Nesse domínio as duas técnicas foram simuladas com 500 passos de simulação.



**Figura 5.37 Levantamento dos tempos de simulação entre as técnicas *PML* e *STL*.**

Dos gráficos apresentados, observa-se que a técnica *PML* despende um tempo maior que a técnica *STL*. Esta diferença torna-se cada vez mais pronunciada com o aumento do tamanho do domínio a ser simulado. Os pontos obtidos não se ajustam perfeitamente à reta interpoladora devido a flutuações nas operações do sistema operacional utilizado. Esses gráficos foram levantados em um sistema operacional “Windows XP”, com um computador com processador “Sempron” com *clock* de 1.63 GHz e 1GB de memória RAM e compilado em ambiente MatLab.

Outro aspecto que a ser analisado, é a quantidade de memória utilizada no processamento das fronteiras de absorção. As técnicas que utilizam menos memória são as aproximações de Mur e a *STL* sem compensação, pois elas não dependem de memória extra para sua execução. Na técnica *PML* é necessário definir os campos *D* e *B*, o que faz com que esta técnica necessite de mais memória que as outras. Já na técnica *STL* com compensação temporal a quantidade de memória extra dependerá do valor da compensação temporal, ou seja, o meio a ser simulado definirá o tamanho do buffer a ser utilizado. Por fim, a quantidade de memória utilizada na técnica *STL* com compensação por equação de onda é o dobro da quantidade de memória utilizada pela técnica de Mur. Estas informações são obtidas através dos desenvolvimentos apresentados nos capítulos anteriores.

## 5.5 Exemplos Práticos

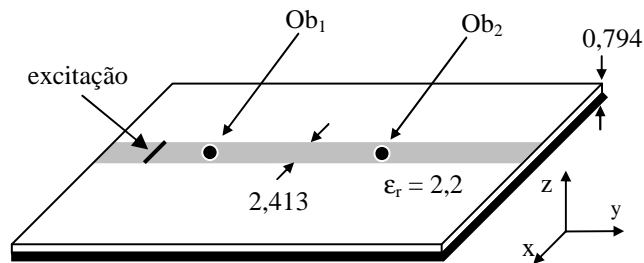
As simulações a seguir apresentam o comportamento das fronteiras de absorção em três estruturas clássicas.

### 5.5.1 Microlinha de Transmissão

A primeira estrutura utilizada é uma microlinha de transmissão. Esse tipo de circuito planar de microondas serve de interface entre um circuito de *RF* e dispositivos de entrada/saída ou entre dois circuitos de *RF*. Nas simulações *FDTD* de outros circuitos planares, elas são utilizadas para auxiliar na determinação dos parâmetros *S* (Anexo C). A Figura 5.38 ilustra a microlinha de transmissão utilizada nas simulações. As dimensões estão em milímetros.

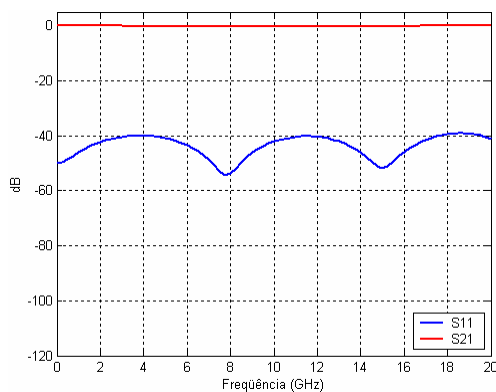
As simulações apresentam uma comparação entre os resultados obtidos para os parâmetros  $S_{11}$  e  $S_{21}$  de microlinhas de transmissão truncadas por diferentes técnicas de

absorção. Se estas estruturas fossem truncadas por fronteiras absorvedoras perfeitas, seus parâmetros  $S_{11}$  e  $S_{21}$  seriam respectivamente infinito e zero, não importando o valor de sua impedância. Como o truncamento do domínio é imperfeito, outros valores são obtidos para esses parâmetros, que nas simulações foram medidos nos pontos  $Ob_1$  e  $Ob_2$ .

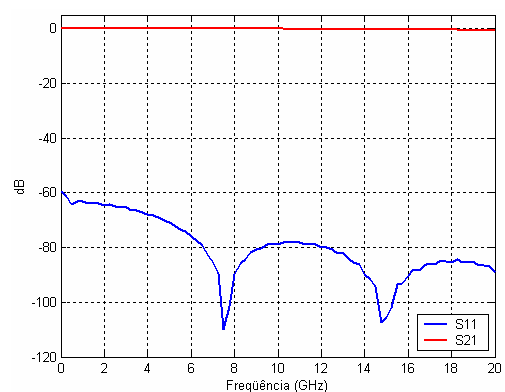


**Figura 5.38** Desenho esquemático da microlinha de transmissão.

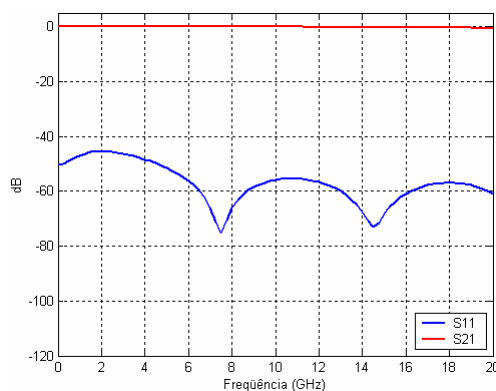
As Figuras 5.39 – 5.42 apresentam respectivamente as simulações obtidas empregando a técnica de truncamento da 1ª aproximação de Mur, a técnica PML, a técnica *STL* sem compensação e a técnica *STL* com compensação por equação de onda.



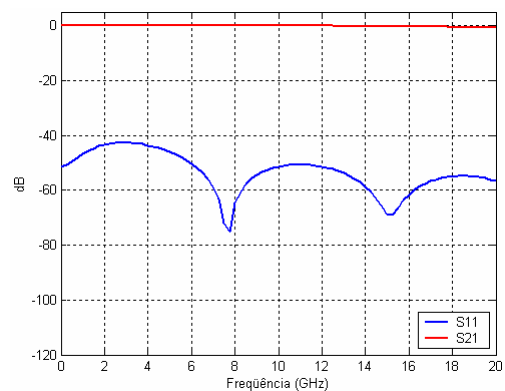
**Figura 5.39**  $S_{11}$  e  $S_{21}$  da microlinha truncada pela 1ª aproximação de Mur.



**Figura 5.40**  $S_{11}$  e  $S_{21}$  da microlinha truncada pela técnica PML.



**Figura 5.41**  $S_{11}$  e  $S_{21}$  da microlinha truncada pela técnica *STL* sem compensação.



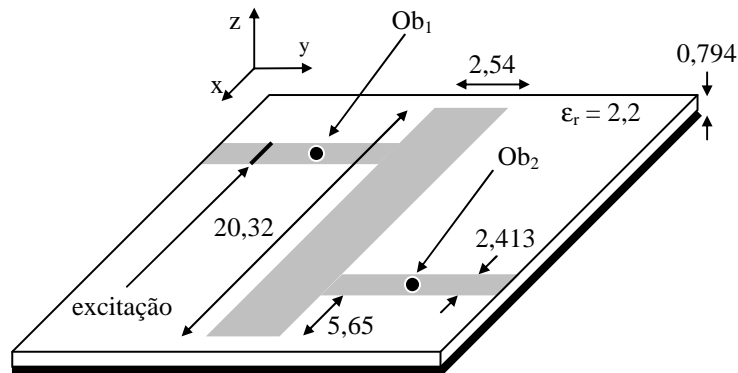
**Figura 5.42**  $S_{11}$  e  $S_{21}$  da microlinha truncada pela técnica *STL* com compensação por equação de onda.

Embora as simulações anteriores mostrarem que as técnicas *PML* e *STL* possuem resultados equivalentes, isso difere um pouco neste tipo de simulação. A técnica *STL* não apresenta a mesma atenuação que a técnica *PML* pois nesta técnica, o ângulo de máxima absorção é em  $45^\circ$  e a microlinha força a incidência da onda em  $0^\circ$ . Este exemplo reforça o fato de que a técnica *STL* é altamente dependente do ângulo de incidência da onda.

### 5.5.2 Filtro Planar

A Figura 5.43 mostra um filtro de *RF* passa-baixa, planar, construído em técnica de micro fita. Essa estrutura bem como as simulações em *FDTD* utilizando fronteiras de absorção, estão reportados em [15]. O desempenho das três técnicas de fronteiras estudadas é apresentado e como elas afetam o resultado final de simulação.

As dimensões que aparecem na Figura 5.43 estão em milímetros e as discretizações espaciais utilizadas são  $\Delta x = 0,4064 \text{ mm}$ ,  $\Delta y = 0,4233 \text{ mm}$ ,  $\Delta z = 0,265 \text{ mm}$ . Um pulso gaussiano de  $15 \text{ ps}$  de duração posicionado a 20 células de distância da fronteira do ressoador é utilizado como fonte de excitação. Os dois pontos de medição  $Ob_1$  e  $Ob_2$  estão posicionados à 10 células de distância da fronteira do ressoador.



**Figura 5.43** Esquemático do filtro planar.

As Figuras 5.44 – 5.47 apresentam comparações entre os parâmetros  $S_{11}$  e  $S_{21}$  simulados e medido do filtro real. Os parâmetros  $S$  usados nestas comparações são calculados utilizando-se a formulação descrita no Anexo C. Os domínios computacionais são truncados pelas seguintes técnicas: primeira aproximação de Mur,

*PML* com  $L = 10$  células,  $m = 4$ ,  $\sigma_{\max} = 16,8993$ , *STL* sem compensação e *STL* com compensação por equação de onda. As *STL*s são finalizadas por uma fronteira de re-radiação implementada com a primeira aproximação de Mur. O parâmetro  $\sigma_{\max}$  utilizado na técnica *PML* é obtido por

$$\sigma_{\max} = \frac{m+1}{150\pi \Delta \sqrt{\epsilon_{\text{eff}}}}, \quad (5.2)$$

sendo,  $\Delta$  a discretização espacial das células ao longo do eixo das camadas *PML* e  $\epsilon_r$  é constante dielétrica efetiva da microlinha.

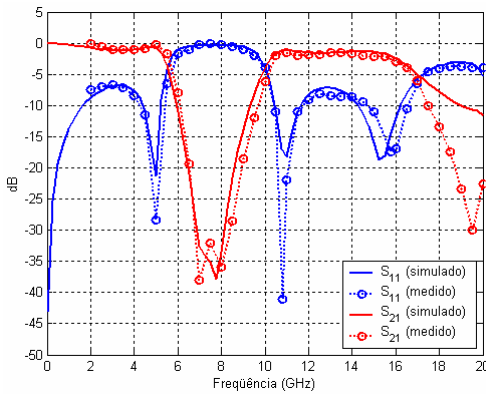


Figura 5.44 S11 e S21 do filtro truncado pela 1ª aproximação de Mur.

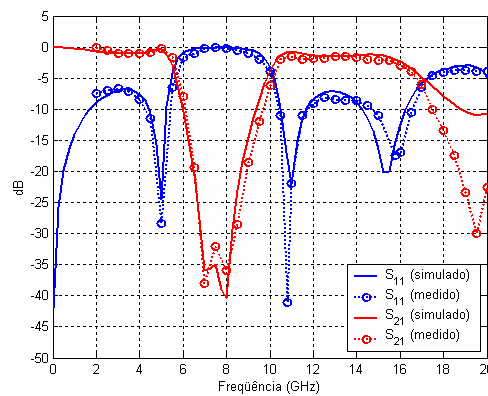


Figura 5.45 S11 e S21 do filtro truncado pela técnica *PML* ( $L = 10$ ,  $m = 4$ ,  $\sigma_{\max} = 16,8993$ ).

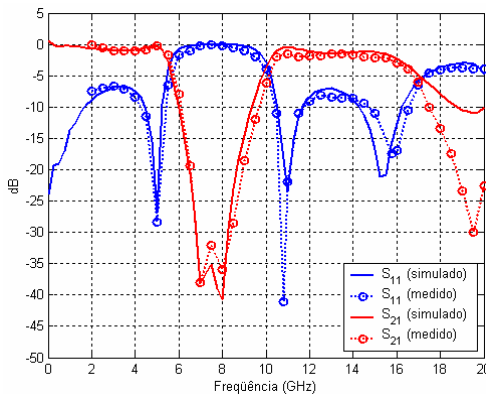


Figura 5.46 S11 e S21 do filtro truncado com uma *STL* finalizada com a 1ª aproximação de Mur.

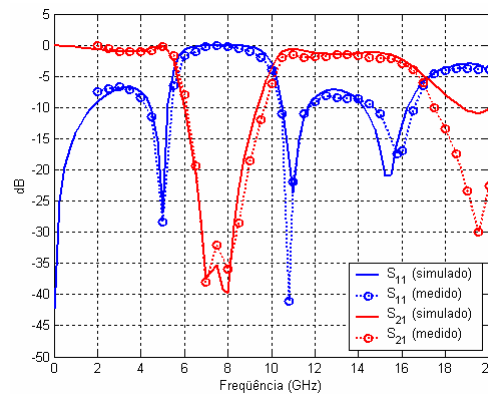


Figura 5.47 S11 e S21 do filtro truncado com uma *STL* com compensação por equação de onda e finalizada com a 1ª aproximação de Mur.

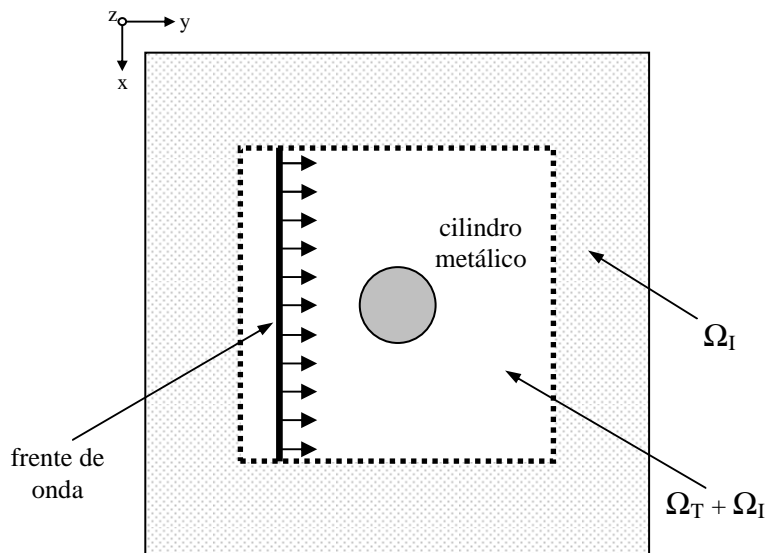
Os gráficos acima mostram excelente conformidade entre os resultados simulados e os obtidos por medição. Nestes gráficos nota-se que todas as técnicas apresentam um erro maior dos parâmetros  $S$  a partir de 17 GHz. Possivelmente, isto é

causado por erros numéricos no algoritmo *FDTD*. Observa-se também que não há uma diferença pronunciada entre as técnicas de Auto Teleportação e a técnica *PML*.

### 5.5.3 Cilindro Iluminado por uma Onda Plana

A terceira estrutura clássica analisada é a de um cilindro metálico iluminado por uma onda plana uniforme. A solução analítica deste problema foi formulada por Mie e constitui um dos problemas clássicos em Eletromagnetismo [20].

Para a onda plana, foi escolhido o modo TM e incidência perpendicular no cilindro metálico, considerado condutor perfeito e de comprimento infinito. Foi utilizado um domínio truncado  $\Omega_T$  imerso em um domínio  $\Omega_I$  considerado ideal, como apresentado na Figura 5.48.



**Figura 5.48 Domínio utilizado no problema do cilindro iluminado**

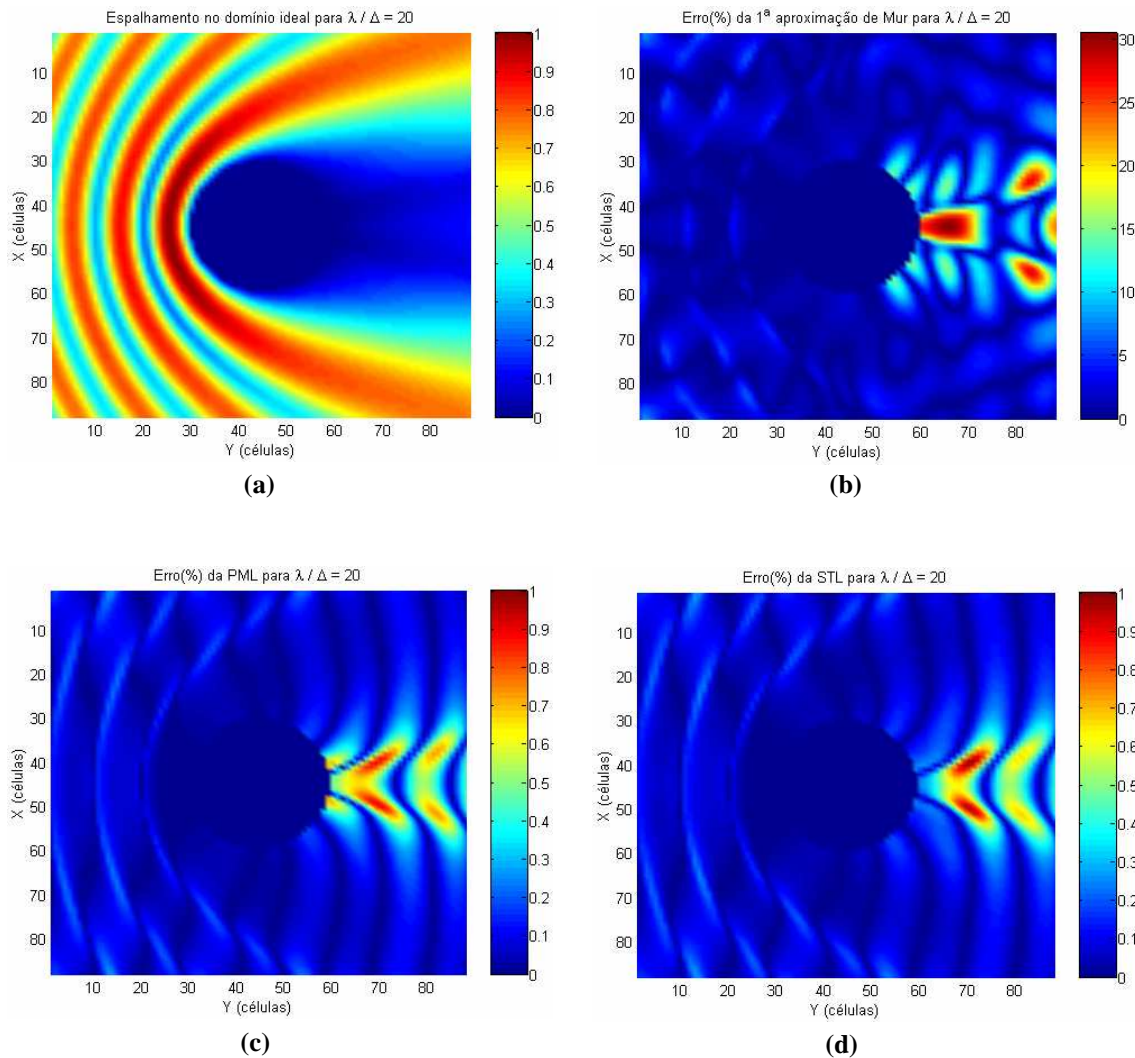
Como em todo problema numérico, existe um erro intrínseco entre a solução analítica e a do modelo computacional. No método *FDTD*, esse erro é causado basicamente pela resolução do domínio computacional e pelo tipo de truncamento empregado para o domínio.

Neste trabalho, é avaliado o erro provocado pelas condições de fronteiras de absorção, comparando-se os resultados obtidos para o domínio truncado e para o domínio ideal, mantendo-se a mesma discretização em ambos os casos.

Para as simulações, são utilizados domínios bi-dimensionais com permissividade  $\epsilon_0$  e permeabilidade  $\mu_0$ , discretizados com  $\Delta x = \Delta y = \Delta$  e um cilindro de condutividade infinita com raio  $r = 14,5 \Delta$ . O erro percentual entre os domínios ideal e truncado é dado por

$$erro|_{i,j} = 100 \cdot \frac{\left| |E_I|_{i,j} - |E_T|_{i,j} \right|}{\left| |E_I|_{i,j} \right|}, \quad \text{para } i, j \in \Omega_T \quad (5.3)$$

sendo  $E_I$  e  $E_T$  respectivamente os campos elétricos dos domínios ideal e truncado. Os erros obtidos estão apresentados nas Figuras 5.49 (a) – (d).



**Figura 5.49** Erro percentual entre os domínios ideal e truncados, (a) campo elétrico  $E_z$  no domínio ideal, (b) erro para a 1ª aproximação de Mur, (c) erro para uma PML, (d) erro para uma STL.



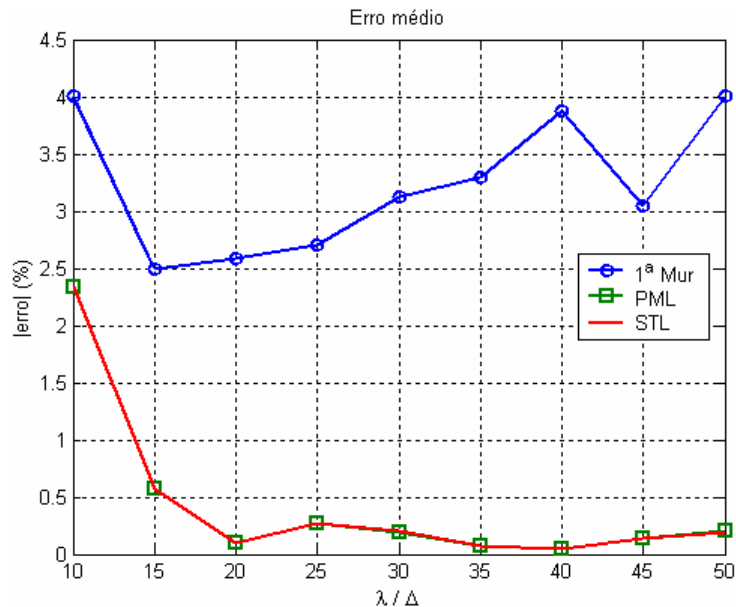
A Figura 5.49(a) apresenta o comportamento do campo elétrico  $E_z$ , no domínio ideal, para um comprimento de onda  $\lambda = 20\Delta$ . As Figuras 5.49 (b) – (c) apresentam, respectivamente, os erros percentuais obtidos para a 1ª aproximação de Mur, para uma *PML* com 10 células e para uma *STL* com 3 fronteiras de absorção com compensação por equação de onda, finalizada por uma fronteira de Mur.

Nessas figuras, nota-se que o domínio truncado pela 1ª aproximação de Mur apresenta erros da ordem de 30%. Já os outros domínios apresentam erros da ordem de 1%. Além disso, é observado que o domínio truncado pela *PML* apresenta mais erros que o domínio truncado pela *STL*.

É possível também definir um erro médio global, referente a todo domínio computacional, dado por

$$\overline{\text{erro}} = \frac{\sum_{i=1}^M \sum_{j=1}^N \text{erro}|_{i,j}}{(M + N)}, \quad \text{para } i, j \in \Omega_T \quad (5.4)$$

sendo  $(M + N)$  a soma de todos os pontos do domínio. Os erros médios globais para as fronteiras de absorção analisadas são apresentados na Figura 5.50, em função do comprimento de onda.



**Figura 5.50** Erro médio percentual entre os domínios simulados para alguns comprimentos de onda.

Das curvas dos erros é possível observar que o erro médio global decorrente da 1ª aproximação de Mur é muito maior que os erros médios globais apresentados pelas

outras duas técnicas de truncamento. Já para as técnicas *PML* e *STL*, os erros médios globais estão muito próximos entre si.

Esses erros estão abaixo de 0,5%, para comprimentos de onda acima de  $15\Delta$ . Isso ocorre porque o método *FDTD* começa a perder eficiência quando são utilizados comprimentos de onda abaixo de  $20\Delta$ .

## 5.6 Conclusão

Os resultados apresentados neste capítulo mostram que a técnica *STL* resulta em atenuações melhores que as obtidas pelas aproximações de Mur mas equivalentes às da técnica *PML* com 10 células de espessura. Apesar disso a técnica *STL* possui a vantagem de necessitar muito menos carga computacional que a *PML*, como observado nas Tabelas 5.1 – 5.6.

Nas simulações do filtro passa baixa, é observado que não há muita diferença nos resultados entre os tipos de fronteiras de absorção utilizadas. Na maioria dos problemas encontrados na prática, é mais importante ter se um algoritmo rápido e que gaste menos carga computacional, que um algoritmo que possua uma absorção extremamente elevada.

## 6 Considerações Finais

Este trabalho apresentou um estudo de fronteiras de absorção que possuem um papel importante nas simulações eletromagnéticas utilizando-se o método *FDTD*. Os resultados obtidos mostram que cada uma das técnicas de fronteiras de absorção apresentam vantagens e desvantagens.

As fronteiras de re-radiação de Mur e Higdon são as técnicas mais simples de serem implementadas. Porém, elas apresentam uma absorção de aproximadamente 20 dB, para aproximações de primeira ordem, e 40 dB para aproximações de segunda ordem. Aproximações de ordem maior ou igual a três podem ser implementadas, mas elas não resultam em melhora significativa na absorção de onda. Outro ponto desfavorável deste tipo de fronteira é a sensibilidade da atenuação com o ângulo de incidência da onda.

A técnica usando camadas absorvedoras (*PML*) pode apresentar atenuações muito maiores que 40 dB e tem pouca sensibilidade ao ângulo de incidência da onda. Isso é conseguido através do aumento da espessura de material absorvedor e da seleção de parâmetros adequados deste material. Porém, a desvantagem deste tipo de fronteira é que ela exige um domínio computacional maior que o anterior e tem uma maior carga computacional.

A terceira técnica apresentada neste trabalho, a técnica de Auto Teleportação de Campos, pode ser entendida como um tipo de fronteira de re-radiação que permite a associação de várias fronteiras. Assim, sua atenuação é aumentada em função do número de fronteiras implementadas. Simulações mostraram que para um número de seis fronteiras, o algoritmo *FDTD* diverge.

A maioria das simulações realizadas mostram que apenas duas ou três fronteiras de Auto Teleportação finalizadas com uma fronteira de Mur, ou Higdon, são suficientes para se ter uma atenuação compatível com a técnica *PML*. Outro atrativo da técnica de Auto Teleportação de Campos é que não há necessidade de tratar interseções entre duas fronteiras de Auto Teleportação, como no caso da técnica *PML*, e nem a necessidade de tratamentos especiais nos cantos, bordas e descontinuidades do domínio, como no caso da segunda aproximação de Mur.

Esta técnica pode ser ainda melhorada através de compensações espacial, temporal, e espaço-temporal que foi denominada neste trabalho compensação por equação de onda. Estas compensações adaptam a técnica de Auto Teleportação ao domínio simulado, fornecendo melhores resultados. A compensação por equação de onda é a mais genérica e eficiente delas.

Os resultados obtidos durante a realização deste trabalho foram bastante satisfatórios. Para futuras atividades de pesquisa, há ainda alguns detalhes sobre a técnica de Auto Teleportação de Campos que precisam ser melhor investigados. Um tópico importante de estudo seria avaliar como se pode aproximar ao máximo as superfícies  $S$  e  $S'$ , mantendo-se a realimentação entre os campos nestas duas superfícies, no menor nível possível. Outro tópico a ser investigado seria o de reescrever as equações de tal forma que a Auto Teleportação possa ser utilizada em materiais que variem suas propriedades na direção normal as superfícies  $S$  e  $S'$ .

Outros trabalhos sobre Condições de Fronteiras de Absorção podem ser encontrados nas referências [21 – 30].

# Anexo A

## Diferenças Finitas no Domínio do Tempo “*Finite Difference Time Domain*” (*FDTD*)

### A.1 Introdução

Em 1966, K. S. Yee [1] apresentou um novo algoritmo numérico para resolver as equações de Maxwell no domínio do tempo. Esse algoritmo foi chamado Método das Diferenças Finitas no Domínio do Tempo, em inglês “*Finite Difference Time Domain*” (*FDTD*). Ele emprega a discretização do tempo e do espaço em análise de tal forma que os componentes dos campos elétricos e magnéticos contidos nesse espaço possam ser obtidos de forma iterativa. As equações de Maxwell que descrevem um meio linear, homogêneo, isotrópico, não dispersivo e livre de fontes são dadas por

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (\text{A.1})$$

$$\nabla \times \vec{H} = \frac{\partial \vec{D}}{\partial t} \quad (\text{A.2})$$

$$\nabla \cdot \vec{B} = 0 \quad (\text{A.3})$$

$$\nabla \cdot \vec{D} = 0 \quad (\text{A.4})$$

e as relações constitutivas do meio por

$$\vec{B} = \mu \vec{H} \quad (\text{A.5})$$

$$\vec{D} = \varepsilon \vec{E} \quad (\text{A.6})$$

sendo,  $\mu$  e  $\varepsilon$ , respectivamente, a permeabilidade e permissividade do meio em análise.

### A.2 Base Matemática

A base matemática para o método *FDTD* é uma técnica chamada de Diferenças Centrais. A idéia consiste em substituir as derivadas parciais de uma equação diferencial, por uma aproximação em forma de diferenças. Essas diferenças são obtidas através da expansão em série de Taylor da função, na região de interesse, da seguinte forma:

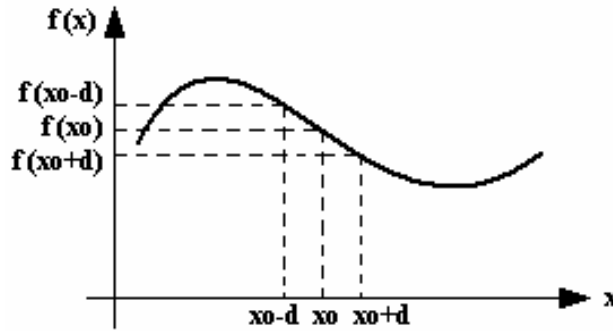


Figura A.1. Ilustração da expansão de Taylor para uma função genérica

Dada uma função  $f(x)$  qualquer, Figura A.1, sua expansão em série de Taylor no ponto  $x_0$  é obtida através da expressão

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + f''(x_0) \frac{(x - x_0)^2}{2!} + \dots \quad (\text{A.7})$$

Para um deslocamento de  $\pm d$  em torno do ponto  $x_0$ , as expansões em série de Taylor são dadas por

$$f(x_0 - d) = f(x_0) - f'(x_0)(d) + f''(x_0) \frac{(d)^2}{2!} - f'''(x_0) \frac{(d)^3}{3!} + \dots \quad (\text{A.8})$$

$$f(x_0 + d) = f(x_0) + f'(x_0)(d) + f''(x_0) \frac{(d)^2}{2!} + f'''(x_0) \frac{(d)^3}{3!} + \dots \quad (\text{A.9})$$

Subtraindo-se a expressão (A.9) de (A.8) e isolando o termo  $f'(x_0)$ , obtém-se

$$f'(x_0) = \frac{f(x_0 + d) - f(x_0 - d)}{2d} - \left[ f'''(x_0) \frac{d^2}{3!} + f^{(5)}(x_0) \frac{d^4}{4!} + \dots \right] \quad (\text{A.10})$$

Essa equação representa a aproximação da derivada parcial da função  $f(x)$  em torno do ponto  $x_0$ .

Em (A.10), os termos não lineares de  $d$  podem ser agrupados de forma a representarem o erro da função  $f'$  no ponto  $x_0$ . Para um valor  $d < 1$  esse erro é proporcional a  $d^2$ , pois os termos de maior ordem tornam-se desprezíveis rapidamente. Portanto, para um intervalo  $h = 2d$  a expressão (A.10) pode ser reescrita na forma

$$f'(x_0) = \frac{f(x_0 + h/2) - f(x_0 - h/2)}{h} + O(h^2) \quad (\text{A.11})$$

sendo,  $O(h^2)$  a função erro, a qual pode ser desprezada quando  $h \ll 1$ .

Essa nova expressão será utilizada para substituir as equações diferenciais, como dito anteriormente.

### A.3 O Método *FDTD*

O ponto de partida do método *FDTD* são as duas equações de Maxwell (A.1) e (A.2) e as duas relações constitutivas (A.5) e (A.6). Portanto, as equações utilizadas nesse método são

$$\nabla \times \vec{E} = -\mu \frac{\partial \vec{H}}{\partial t} \quad (\text{A.12})$$

$$\nabla \times \vec{H} = \varepsilon \frac{\partial \vec{E}}{\partial t} \quad (\text{A.13})$$

Em um sistema de coordenadas retangulares, essas equações são expandidas em seis equações diferenciais parciais, a saber:

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu} \left( \frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \right) \quad (\text{A.14}) \quad \frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon} \left( \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right) \quad (\text{A.17})$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu} \left( \frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \right) \quad (\text{A.15}) \quad \frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon} \left( \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right) \quad (\text{A.18})$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu} \left( \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right) \quad (\text{A.16}) \quad \frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon} \left( \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) \quad (\text{A.19})$$

sendo,  $E_x$ ,  $E_y$ ,  $E_z$  os componentes do campo elétrico e  $H_x$ ,  $H_y$ ,  $H_z$  os componentes do campo magnético, em coordenadas cartesianas.

Para resolver essas equações utilizando-se a técnica das diferenças centrais, apresentada no item A.2, deve-se primeiro discretizar o espaço a ser analisado. Esta discretização é feita dividindo-se o espaço em pequenas partes chamadas células, que, na literatura técnica, são chamadas de células de Yee. Na Figura A.2 está apresentada uma possível distribuição dos componentes dos campos elétrico e magnético em uma célula de Yee. Posteriormente, deve-se discretizar o tempo escolhendo-se, adequadamente, um passo temporal ( $\Delta t$ ).

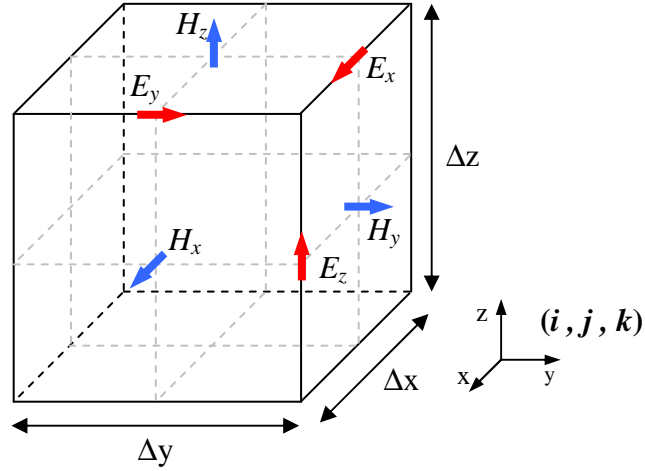


Figura A.2. Célula de Yee na posição  $(i, j, k)$ .

Utilizando-se esta alocação dos componentes dos campos na célula de Yee na posição  $(i, j, k)$  e aplicando a técnica de diferenças centrais nas equações (A.14 – A.19), obtém-se as equações iterativas

$$H_x|_{i,j,k}^{n+1/2} = H_x|_{i,j,k}^{n-1/2} - \frac{\Delta t}{\mu} \left[ \frac{E_z|_{i,j,k}^n - E_z|_{i,j-1,k}^n}{\Delta y} - \frac{E_y|_{i,j,k}^n - E_y|_{i,j,k-1}^n}{\Delta z} \right], \quad (\text{A.20})$$

$$H_y|_{i,j,k}^{n+1/2} = H_y|_{i,j,k}^{n-1/2} - \frac{\Delta t}{\mu} \left[ \frac{E_x|_{i,j,k}^n - E_x|_{i,j,k-1}^n}{\Delta z} - \frac{E_z|_{i,j,k}^n - E_z|_{i-1,j,k}^n}{\Delta x} \right], \quad (\text{A.21})$$

$$H_z|_{i,j,k}^{n+1/2} = H_z|_{i,j,k}^{n-1/2} - \frac{\Delta t}{\mu} \left[ \frac{E_y|_{i,j,k}^n - E_y|_{i-1,j,k}^n}{\Delta x} - \frac{E_x|_{i,j,k}^n - E_x|_{i,j-1,k}^n}{\Delta y} \right], \quad (\text{A.22})$$

$$E_x|_{i,j,k}^{n+1} = E_x|_{i,j,k}^n + \frac{\Delta t}{\varepsilon} \left[ \frac{H_z|_{i,j+1,k}^{n+1/2} - H_z|_{i,j,k}^{n+1/2}}{\Delta y} - \frac{H_y|_{i,j,k+1}^{n+1/2} - H_y|_{i,j,k}^{n+1/2}}{\Delta z} \right], \quad (\text{A.23})$$

$$E_y|_{i,j,k}^{n+1} = E_y|_{i,j,k}^n + \frac{\Delta t}{\varepsilon} \left[ \frac{H_x|_{i,j,k+1}^{n+1/2} - H_x|_{i,j,k}^{n+1/2}}{\Delta z} - \frac{H_z|_{i+1,j,k}^{n+1/2} - H_z|_{i,j,k}^{n+1/2}}{\Delta x} \right], \quad (\text{A.24})$$

$$E_z|_{i,j,k}^{n+1} = E_z|_{i,j,k}^n + \frac{\Delta t}{\varepsilon} \left[ \frac{H_y|_{i+1,j,k}^{n+1/2} - H_y|_{i,j,k}^{n+1/2}}{\Delta x} - \frac{H_x|_{i,j+1,k}^{n+1/2} - H_x|_{i,j,k}^{n+1/2}}{\Delta y} \right], \quad (\text{A.25})$$

$$H_C|_{i,j,k}^n = \frac{H_C|_{i,j,k}^{n+1/2} - H_C|_{i,j,k}^{n-1/2}}{2}, \quad (\text{A.26})$$

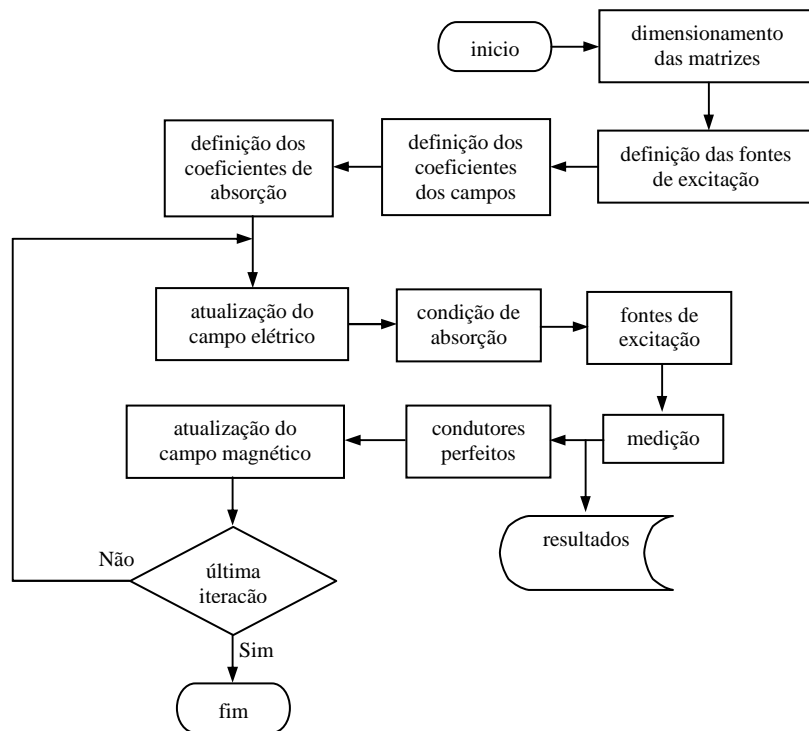


$$E_C|_{i,j,k}^{n+1/2} = \frac{E_C|_{i,j,k}^{n+1} - E_C|_{i,j,k}^n}{2}, \quad (\text{A.27})$$

que servirão para implementar o algoritmo *FDTD*.

Destas equações, três são para os componentes do campos magnético (A.20 – A.22) e três são para os componentes do campo elétrico (A.23 – A.25). Nota-se que é necessário o uso das equações semi-implícitas (A.26) e (A.27), sendo que o sub-índice *C* representa as direções *x*, *y* e *z* dos componentes dos campos.

Para se obter os valores dos campos eletromagnéticos através destas equações, deve-se calcular de forma iterativa o valor dos campos em um instante *n+1* utilizando-se com isso o valor dos campos no instante *n*. A Figura A.3 apresenta o fluxograma de uma possível implementação do algoritmo *FDTD*.



**Figura A.3. Fluxograma de uma possível implementação de um algoritmo *FDTD*.**

## A.4 Critérios de Estabilidade

Para se obter uma resolução espacial adequada para o domínio computacional, as células de Yee devem ser pequenas o suficiente. Tipicamente este valor é escolhido entre 1/10 e 1/20 do comprimento de onda da maior frequência de interesse. Valores

maiores podem ser escolhidos; contudo, haverá uma maior dispersão numérica do método.

A discretização temporal  $\Delta t$  também deve ser escolhida de forma apropriada para garantir a estabilidade numérica por todo o tempo de simulação. Este valor é obtido através da condição de estabilidade de Courant. Para um domínio computacional tridimensional essa condição é dada pela expressão

$$v_{\max} \cdot \Delta t \leq \left[ \left( \frac{1}{\Delta x} \right)^2 + \left( \frac{1}{\Delta y} \right)^2 + \left( \frac{1}{\Delta z} \right)^2 \right]^{-1/2} \quad (\text{A.28})$$

sendo,  $v_{\max}$  a maior velocidade de propagação de onda.

Para simulações de antenas e circuitos passivos, é prática usual fazer com que o passo  $\Delta t$  seja o maior valor permitido pela condição de Courant, de forma que o número de iterações seja o menor possível. Contudo, para simulações muito longas, utilizando materiais que apresentam perdas e circuitos não lineares, deve-se escolher um passo  $\Delta t$  menor que o valor máximo permitido, de forma a se evitar que o algoritmo divirja.

## A.5 Modelamento de Materiais

Uma das vantagens do algoritmo *FDTD* é que se pode modelar uma grande quantidade de estruturas complexas da mesma forma que estruturas simples. Existem três tipos básicos de estruturas que se pode construir, das quais são derivadas todas as outras. Essas estruturas básicas são os condutores elétricos perfeitos, os condutores magnéticos perfeitos e os dielétricos.

### A.5.1 Condutor Elétrico Perfeito

Com um Condutor Elétrico Perfeito, ou em inglês, “Perfect Electric Conductor” (*PEC*) pode-se modelar estruturas com condutividade infinita ( $\sigma = \infty$ ). Sobre a superfície de um *PEC*, os componentes tangenciais do campo elétrico e os componentes normais do campo magnético devem ser nulos. Referindo-se as alocações dos componentes dos campos em uma célula de Yee, Figura A.2, um *PEC* posicionado na face superior de uma célula na posição  $(i, j, k)$  pode ser modelado pela simples imposição dos campos

$$E_x|_{i,j,k} = E_x|_{i,j-1,k} = E_y|_{i,j,k} = E_y|_{i-1,j,k} = H_z|_{i,j,k} \equiv 0 \quad (\text{A.29})$$

em todos os passos de simulação.

Naturalmente, não é preciso impor esta condição para o componente  $H_z$  pois, como ele é calculado pela circuitação dos componentes  $E_x$  e  $E_y$  e estes assumem o valor zero,  $H_z$  também assumirá o valor zero. Similarmente, se quisermos definir esta célula como um bloco de *PEC*, deve-se impor o valor zero para os doze componentes dos campos elétrico e magnético.

### A.5.2 Condutor Magnético Perfeito

Com um Condutor Magnético Perfeito, ou em inglês “Perfect Magnetic Conductor” (*PMC*), pode-se modelar estruturas com resistividade magnética infinita ( $\rho^* = \infty$ ). Sobre a superfície de um *PMC*, os componentes tangenciais do campo magnético e os componentes normais do campo elétrico devem ter valor zero. Referindo-se novamente a Figura A.2, nota-se que os *PMC* só podem ser construídos no meio das células de Yee, devido a alocação dos componentes do campo magnético. O *PMC* é implementado pela imposição de

$$H_x|_{i,j,k} = H_x|_{i-1,j,k} = H_y|_{i,j,k} = H_y|_{i,j-1,k} \equiv 0 \quad (\text{A.30})$$

em todos os passos de simulação.

### A.5.3 Interface meios dielétricos

Nas interfaces de dois meios dielétricos diferentes, como, por exemplo, a interface ar – dielétrico, as equações do algoritmo *FDTD* precisam ser modificadas de forma apropriada. Usualmente o espaço é discretizado de tal forma que as interfaces entre meios diferentes possam ser alocadas em uma camada de células. Para a interface entre dois meios dielétricos  $\epsilon_1$  e  $\epsilon_2$ , pode-se definir os campos da seguinte forma

$$\frac{\epsilon_1 + \epsilon_2}{2} \cdot \frac{\partial E_x}{\partial t} = \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \quad (\text{A.31})$$

$$\frac{\epsilon_1 + \epsilon_2}{2} \cdot \frac{\partial E_y}{\partial t} = \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \quad (\text{A.32})$$

## A.6 Fontes de Excitação

A escolha apropriada de uma fonte de excitação é um dos fatores críticos para uma boa simulação em *FDTD*. Para a maioria dessas simulações, tais como em estruturas de microondas, é prática comum usar-se pulsos e excitar um componente específico dos campos em uma determinada localização dentro da grade *FDTD*. Em particular, pulsos Gaussianos, modulados ou não, têm sido muito utilizados na maioria das simulações, porque eles têm uma forma e um espectro bem definidos. Um pulso Gaussiano em banda-base pode ser expresso por

$$g(t) = \exp\left(-\left(\frac{t-t_0}{T}\right)^2\right) \quad (\text{A.33})$$

e um pulso modulado expresso por

$$g(t) = \exp\left(-\left(\frac{t-t_0}{T}\right)^2\right) \cdot \text{sen}(2\pi \cdot f_0(t-t_0)), \quad (\text{A.34})$$

sendo,  $t_0$  o centro do pulso,  $T$  a largura do pulso e  $f_0$  a frequência de modulação.

### A.6.1 Fontes “Hard” e “Soft”

Um outro ponto importante é a forma de inserir estes pulsos na grade *FDTD*. Os tipos mais comuns de inserção são chamadas de “Hard” e “Soft”. As fontes de excitação do tipo “Hard” injetam o sinal na grade *FDTD* de modo direto, ou seja, a função  $g(t)$  impõe seu valor num componente do campo, como por exemplo

$$E_z|_{i_e, j_e, k_e}^n = g(n \cdot \Delta t) \quad (\text{A.35})$$

Nas fontes do tipo “Soft”, o valor da função é somado ao valor do campo calculado pelas equações iterativas. A equação que exemplifica a implementação do tipo “Soft” é dada por

$$E_z|_{i_e, j_e, k_e}^{n+1} = g(n \cdot \Delta t) + E_z|_{i_e, j_e, k_e}^n + \frac{\Delta t}{\epsilon} \cdot \left[ \frac{H_y|_{i_e+1, j_e, k_e}^{n+1/2} - H_y|_{i_e, j_e, k_e}^{n+1/2}}{\Delta x} - \frac{H_x|_{i_e, j_e+1, k_e}^{n+1/2} - H_x|_{i_e, j_e, k_e}^{n+1/2}}{\Delta y} \right] \quad (\text{A.36})$$

sendo  $(i_e, j_e, k_e)$  as coordenadas da célula aonde a fonte é inserida.

A diferença básica entre esses dois tipos de inserção, está no fato de que o tipo “Hard” cria um *PEC* não físico no local onde a fonte é inserida. Isso causa, na maioria dos problemas, reflexões indesejáveis devido a imposição forçada do campo  $E_z$ . Já as fontes do tipo “Soft” inserem pulsos de forma natural, não causando nenhum tipo de reflexão. Como o primeiro termo à direita da equação (A.36) vai para zero após alguns passos de simulação, esta equação assumirá a forma padrão das equações iterativas do algoritmo *FDTD*.

# Anexo B

## Grades Não – Uniformes

### B.1 Introdução

Para se obter um bom resultado numérico nas simulações utilizando-se o método *FDTD*, deve-se impor uma alta resolução de discretização nos domínios computacionais. Ela deve ser elevada o suficiente para que se possa modelar de forma correta os fenômenos eletromagnéticos em determinadas regiões do espaço, tais como bordas, fendas, cantos e outras discontinuidades em geral.

Utilizando-se a discretização padrão do algoritmo *FDTD*, nota-se que a densidade de células aumenta em todo o domínio computacional. Isso leva a um aumento desnecessário da carga computacional, pois, na maioria dos problemas, esse refinamento elevado não é necessário em todo o domínio, mas somente em certas regiões. Esse refinamento localizado é possível através do uso da técnica de Sub-Grades [9, 12, 13]. Contudo, há também necessidade de se modificar o tamanho de um dos lados das células, o que é conseguido através da técnica de Grades Não – Uniformes [9, 14, 15]. As Figuras B.1(a) e B.1(b) ilustram uma grade uniforme e uma grade não-uniforme .

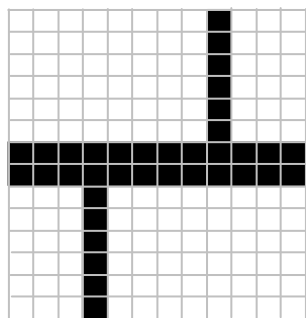


Figura B.1(a) Domínio computacional utilizando uma grade uniforme.

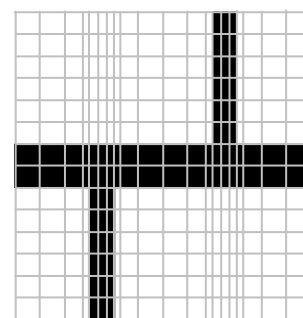


Figura B.1(b) Domínio computacional utilizando uma grade não uniforme.

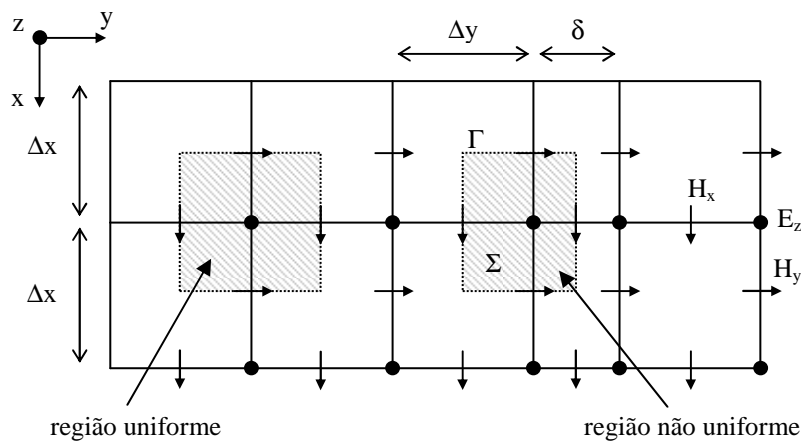
### B.2 Implementação

O algoritmo de Grades Não – Uniformes pode ser implementado através da discretização das equações de Maxwell em sua forma integral, mais especificamente, das leis de Ampère e de Faraday, respectivamente, dadas por

$$\oint_{\Gamma} \vec{H} \cdot d\vec{l} = \frac{\partial}{\partial t} \left( \iint_{\Sigma} \epsilon \vec{E} \cdot d\vec{s} \right) \quad (\text{B.1})$$

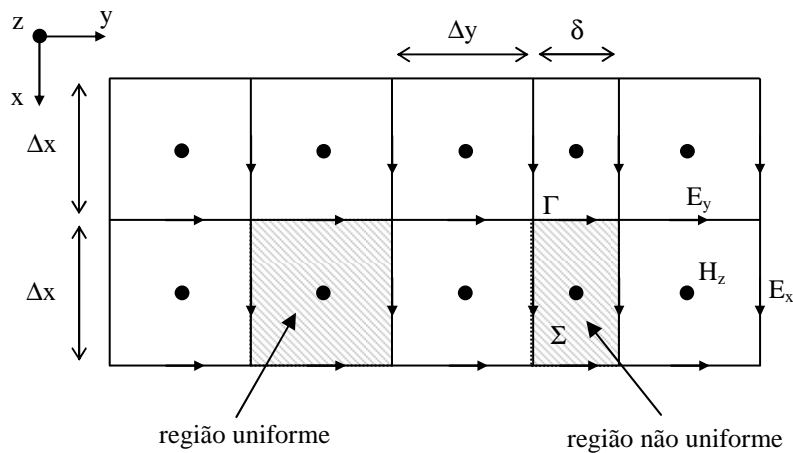
$$\oint_{\Gamma} \vec{E} \cdot d\vec{l} = - \frac{\partial}{\partial t} \left( \iint_{\Sigma} \mu \vec{H} \cdot d\vec{s} \right). \quad (\text{B.2})$$

As integrais descritas por (B.1) podem ser dispostas no reticulado através do esquema mostrado na Figura B.2. Nesta figura, estão esquematizados os contornos feitos em uma área uniforme e em uma área não uniforme. Os círculos representam os campos elétricos e as setas representam os campos magnéticos.



**Figura B.2. Ilustração da aplicação da lei de Ampère.**

Já as integrais descritas por (B.2) podem ser dispostas no reticulado através do esquema mostrado na Figura B.3. Nessa figura, estão esquematizados contornos feitos em uma área uniforme e em uma área não uniforme. Os círculos representam os campos magnéticos e as setas representam os campos elétricos.



**Figura B.3. Ilustração da aplicação da lei de Faraday.**

Para o cálculo das equações de atualização dos campos elétricos, deve-se utilizar a equação (B.1) e os contornos ilustrados na Figura B.2. De posse disso, e utilizando-se a técnica de diferenças centrais, obtém-se a expressão

$$\epsilon \cdot \frac{E_z|_{i,j}^{n+1} - E_z|_{i,j}^{n-1}}{\Delta t} = \frac{\left(H_x|_{i,j}^{n+1/2} - H_x|_{i,j+1}^{n+1/2}\right)}{\left(\frac{\Delta y + \delta}{2}\right)} + \frac{\left(H_y|_{i+1,j}^{n+1/2} - H_y|_{i,j}^{n+1/2}\right)}{\Delta x}. \quad (\text{B.3})$$

O cálculo dos campos magnéticos é feito de forma análoga, através do uso da equação (B.2) e os contornos mostrados na Figura B.3. Com isso, obtém-se a expressão

$$-\mu \cdot \frac{H_z|_{i,j}^{n+1/2} - H_z|_{i,j}^{n-1/2}}{\Delta t} = \frac{\left(E_x|_{i,j-1}^n - E_x|_{i,j}^n\right)}{\delta} + \frac{\left(E_y|_{i,j}^n - E_y|_{i-1,j}^n\right)}{\Delta x}. \quad (\text{B.4})$$

O cálculo dos outros componentes dos campos é feito de maneira análoga, tomando-se o cuidado de escolher as superfícies e os contornos de integração de forma adequada.

Para se evitar a divergência do método *FDTD*, é necessário aplicar a condição de estabilidade de Courant dada por

$$v \cdot \Delta t \leq \left[ \left(\frac{1}{\Delta x_{\min}}\right)^2 + \left(\frac{1}{\Delta y_{\min}}\right)^2 + \left(\frac{1}{\Delta z_{\min}}\right)^2 \right]^{-1/2} \quad (\text{B.5})$$

utilizando-se as dimensões mínimas das células nas direções  $x$ ,  $y$  e  $z$ .

A técnica de grades não uniformes é extremamente útil para a análise de circuitos planares de microondas. Os detalhes geométricos destes circuitos, tais como cantos e bordas, acarretam pequenas dimensões das células do domínio computacional. Além disso, estes circuitos estão imersos em um meio sem fronteiras. Isso faz com que as fronteiras de absorção tenham que ser colocadas à uma distância suficientemente grande de tais circuitos, de forma a evitar reflexões não-físicas.

Nas grades uniformes, essas duas características acarretam em um domínio computacional extremamente grande. Com as grades não uniformes, as dimensões das células próximas a estas singularidades da geometria podem ser pequenas o bastante para modelar corretamente os campos eletromagnéticos. As células com dimensões maiores são utilizadas em regiões onde os campos são bem comportados. Nota-se que as dimensões dessas células ainda devem ser mantidas menores que  $\lambda_{\min}/20$  ou no máximo  $\lambda_{\min}/10$ , sendo  $\lambda_{\min}$  o menor comprimento de onda presente no domínio.



# Anexo C

## Determinação de Parâmetros S

Os parâmetros  $S$  são frequentemente usados para a caracterização do desempenho de circuitos de microondas. A literatura sobre esses parâmetros é vasta e seu desenvolvimento é explorado em muitos livros texto. Neste apêndice optou-se por seguir a formulação apresentada em [30].

Para um circuito que possui  $N$  portas, os parâmetros  $S$  podem ser expressos por uma matriz  $\{N \times N\}$  na qual cada elemento é calculado pela expressão

$$S_{mn} = \frac{\mathfrak{S}[V_m(t)]}{\mathfrak{S}[V_n(t)]} \cdot \sqrt{\frac{Z_{0n}}{Z_{0m}}} \quad (\text{C.1})$$

sendo  $\mathfrak{S}$  a notação para a transformação de Fourier,  $m$  e  $n$  o número de portas do circuito,  $V_m(t)$  e  $V_n(t)$  as tensões no domínio do tempo nas entradas  $m$  e  $n$  respectivamente e  $Z_{0m}$  e  $Z_{0n}$  são as impedâncias características das linhas de transmissão conectadas as entradas  $m$  e  $n$  respectivamente. Na maioria dos circuitos encontrados na prática, estas impedâncias são usualmente iguais entre si e iguais a impedância de intrínseca do gerador (normalmente  $50 \Omega$ ).

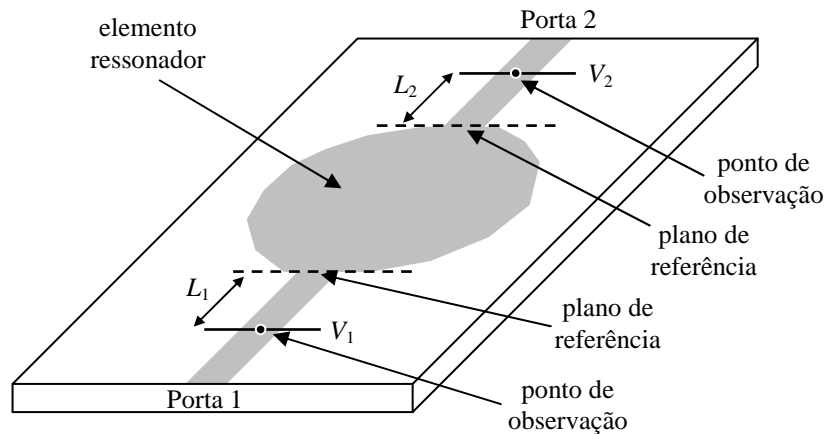


Figura C.1. Circuito de microondas de duas portas genérico.

Para um circuito planar genérico de duas portas, Figura C.1, os parâmetros  $S$  são dados por

$$[S] = \begin{bmatrix} \frac{\mathfrak{S}[V_1(t)]_R}{\mathfrak{S}[V_1(t)]_I} & \frac{\mathfrak{S}[V_1(t)]_T}{\mathfrak{S}[V_2(t)]_I} \\ \frac{\mathfrak{S}[V_2(t)]_T}{\mathfrak{S}[V_1(t)]_I} & \frac{\mathfrak{S}[V_2(t)]_R}{\mathfrak{S}[V_2(t)]_I} \end{bmatrix} \quad (C.2)$$

sendo  $\mathfrak{S}[V_i(t)]_I$  e  $\mathfrak{S}[V_i(t)]_R$  respectivamente as tensões incidentes e refletidas em cada porta e  $\mathfrak{S}[V_i(t)]_T$  é a tensão transmitida na porta  $i$  com relação ao sinal incidente em outra porta.

Nas simulações em *FDTD* é prática comum escolher-se os pontos de observação à uma certa distância dos planos de referência, Figura C.1, de forma a evitar possíveis efeitos causados por modos de propagação de alta ordem devidos a descontinuidades nos circuitos ressoadores. Em vista disso, é necessário transformar as fases dos parâmetros  $S$  obtidos nesses pontos de observação para posicionar os planos de referência como indicado na Figura C.1. Com isso, os elementos da matriz (C.2) ficam modificados nas formas

$$S_{11} = \frac{\mathfrak{S}[V_1(t)]_R}{\mathfrak{S}[V_1(t)]_I} \cdot \exp(2\gamma_1 L_1) \quad (C.3)$$

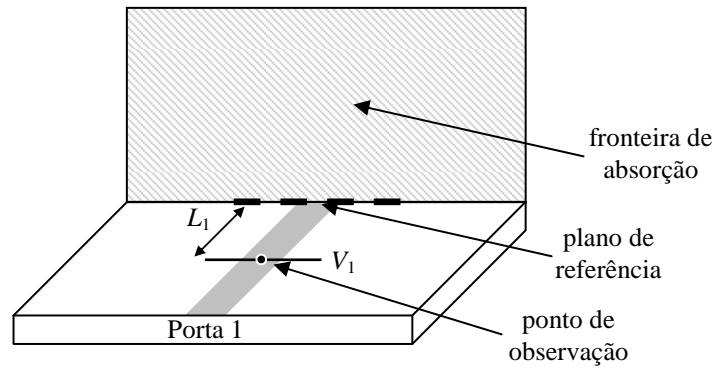
$$S_{12} = \frac{\mathfrak{S}[V_1(t)]_T}{\mathfrak{S}[V_2(t)]_I} \cdot \exp(\gamma_1 L_1 + \gamma_2 L_2) \quad (C.4)$$

$$S_{21} = \frac{\mathfrak{S}[V_2(t)]_T}{\mathfrak{S}[V_1(t)]_I} \cdot \exp(\gamma_1 L_1 + \gamma_2 L_2) \quad (C.5)$$

$$S_{22} = \frac{\mathfrak{S}[V_2(t)]_R}{\mathfrak{S}[V_2(t)]_I} \cdot \exp(2\gamma_2 L_2) \quad (C.6)$$

sendo  $L_1$  e  $L_2$  as distâncias entre os pontos de observação e os planos de referência e  $\gamma_1$  e  $\gamma_2$  as constantes de propagação ( $\gamma = \alpha + j\beta$ ) das linhas de transmissão conectadas nas portas 1 e 2, respectivamente.

Para se determinar as ondas incidentes, deve-se simular apenas o trecho da linha de transmissão que insere a onda no elemento ressoador. A Figura C.2 ilustra a estrutura a ser simulada para a determinação de uma onda incidente inserida na porta 1.



**Figura C.2. Determinação da onda incidente.**

Com isso, é obtido um resultado simulando uma linha de transmissão de comprimento infinito.

# Anexo D

## Algoritmos Utilizados

Neste apêndice é apresentado os principais códigos – fonte utilizados para a realização deste trabalho. Esses códigos estão implementados em linguagem MatLab.

### (1) Campo\_Ez\_1mur.m

```
%---- propriedades do vacuo ----%
eo = 8.854e-12;
uo = 4*pi*1e-7;
%-----%

%---- tamanho discretizacao do dominio ----%
xmax = 100;
ymax = 100;

nmax = 200;

dx = 1e-3;          % discretizacao espacial no eixo x
dy = 1e-3;          % discretizacao espacial no eixo y
%-----%

%---- matrizes dos campos ----%
Ez(1:xmax,1:ymax,1:2) = 0;
Hx(1:xmax,1:ymax,1:2) = 0;
Hy(1:xmax,1:ymax,1:2) = 0;

M(1:xmax,1:ymax,1:nmax) = 0;          % matriz para salvar os resultados
%-----%

%---- propriedades do meio ----%
e(1:xmax,1:ymax) = eo;
u(1:xmax,1:ymax) = uo;
s(1:xmax,1:ymax) = 0;
pm(1:xmax,1:ymax) = 0;
%-----%

%--- descretizacao temporal ---%
v = 1./sqrt(uo*eo);
dt = (1/sqrt((1/dx)^2+(1/dy)^2))/v;
%-----%

%---- excitacao ----%
n = 1:nmax;
t = dt*n;
Lmax = max(10.*[dx dy]);
T = Lmax/v;
Vz(n) = exp(-(((dt*n)-(3*T))./(T/3)).^2);
xexc = round(xmax/2);
yexc = round(ymax/2);
%-----%

%---- coef. dos campos ----%
e1 = (1-s.*dt./(2.*e))./(1+s.*dt./(2.*e));
e2 = (dt./e)./(1+s.*dt./(2.*e));
h1 = (1-pm.*dt./(2.*u))./(1+pm.*dt./(2.*u));
h2 = (dt./u)./(1+pm.*dt./(2.*u));
%-----%
```

```

%---- posicionamento das fronteiras ----%
    bx1 = 0+1;          % xmin+1
    by1 = 0+1;          % ymin+1
    bx2 = xmax-1;      % xmax-1
    by2 = ymax-1;      % ymax-1
%-----%

%---- coef. lo Mur ----%
    a = 0;              % Higdon para a > 0
    v2 = 1./sqrt(u.*e);
    mur1x = ((v2.*dt-cos((pi/180)*a)*dx)./(v2.*dt+cos((pi/180)*a)*dx));
    mur1y = ((v2.*dt-cos((pi/180)*a)*dy)./(v2.*dt+cos((pi/180)*a)*dy));
%-----%

%---- vetores para varredura ----%
    i = 2:xmax-1;
    j = 2:ymax-1;
%-----%

%----- inicio do calculo -----%
for n=2:nmax

    disp(n);

    %---- excitacao ----%
        Ez(xexc,yexc,1)=Ez(xexc,yexc,1)+Vz(n);
    %-----%

    %---- campo magnetico ----%
        Hx(i,j,2)=h1(i,j).*Hx(i,j,1)...
            -h2(i,j).*(Ez(i,j,1)-Ez(i,j-1,1))./dy;

        Hy(i,j,2)=h1(i,j).*Hy(i,j,1)...
            -h2(i,j).*(-(Ez(i,j,1)-Ez(i-1,j,1))./dx);
    %-----%

    %---- campo eletrico ----%
        Ez(i,j,2)=e1(i,j).*Ez(i,j,1)...
            +e2(i,j).*(Hy(i+1,j,2)-Hy(i,j,2))./dx-(Hx(i,j+1,2)-Hx(i,j,2))./dy);
    %-----%

    %---- condicao de absorcao ----%
        % xmin+1
        Ez(bx1,:,2)=Ez(bx1+1,:,1)+mur1x(bx1,:).*(Ez(bx1+1,:,2)-Ez(bx1,:,1));
        % xmax-1
        Ez(bx2,:,2)=Ez(bx2-1,:,1)+mur1x(bx2,:).*(Ez(bx2-1,:,2)-Ez(bx2,:,1));
        % ymin+1
        Ez(:,by1,2)=Ez(:,by1+1,1)+mur1y(:,by1).*(Ez(:,by1+1,2)-Ez(:,by1,1));
        % ymax-1
        Ez(:,by2,2)=Ez(:,by2-1,1)+mur1y(:,by2).*(Ez(:,by2-1,2)-Ez(:,by2,1));
    %-----%

        M(:, :, n) = Ez(:, :, 2);          % salva resultados do campo Ez

    %---- restauracao das matrizes ----%
        Ez(:, :, 1)=Ez(:, :, 2);  Hx(:, :, 1)=Hx(:, :, 2);  Hy(:, :, 1)=Hy(:, :, 2);
    %-----%

end
%-----%

%---- apresenta resultados na tela ----%
figure(1);
hh=uicontrol('Style','slider','Min',0,'Max',nmax-1,'SliderStep',[1/nmax 10/nmax],...
    'Callback',['vi=get(gca, 'View');'...
        'n=round(get(hh, 'Value'))+1;'...
        'surf(M(:, :, n)); axis ij;'...
        'shading interp;'...
        'set(gca, 'View', vi);'...
        'title(n);'...
    ]);
set(gcf, 'toolbar', 'figure');
%-----%

```

## (2) Campo\_Ez\_2mur.m

```

%---- propriedades do vacuo ----%
eo = 8.854e-12;
uo = 4*pi*1e-7;
%-----%

%---- tamanho discretizacao do dominio ----%
xmax = 100;
ymax = 100;

nmax = 200;

dx = 1e-3;          % discretizacao espacial no eixo x
dy = 1e-3;          % discretizacao espacial no eixo y
%-----%

%---- matrizes dos campos ----%
Ez(1:xmax,1:ymax,1:2) = 0;
Hx(1:xmax,1:ymax,1:2) = 0;
Hy(1:xmax,1:ymax,1:2) = 0;

M(1:xmax,1:ymax,1:nmax) = 0;          % matriz para salvar os resultados
%-----%

%---- propriedades do meio ----%
e(1:xmax,1:ymax) = eo;
u(1:xmax,1:ymax) = uo;
s(1:xmax,1:ymax) = 0;
pm(1:xmax,1:ymax) = 0;
%-----%

%--- descretizacao temporal ---%
v = 1./sqrt(uo*eo);
dt = (1/sqrt((1/dx)^2+(1/dy)^2))/v;
%-----%

%---- excitacao ----%
n = 1:nmax;
t = dt*n;
Lmax = max(10.*[dx dy]);
T = Lmax/v;
Vz(n) = exp(-(((dt*n)-(3*T))./(T/3)).^2);
xexc = round(xmax/2);
yexc = round(ymax/2);
%-----%

%---- coef. dos campos ----%
e1 = (1-s.*dt./(2.*e))./(1+s.*dt./(2.*e));
e2 = (dt./e)./(1+s.*dt./(2.*e));
h1 = (1-pm.*dt./(2.*u))./(1+pm.*dt./(2.*u));
h2 = (dt./u)./(1+pm.*dt./(2.*u));
%-----%

%---- posicionamento das fronteiras ----%
bx1 = 0+1;          % xmin+1
by1 = 0+1;          % ymin+1
bx2 = xmax-1;      % xmax-1
by2 = ymax-1;      % ymax-1
%-----%

%---- coef. 2o Mur ----%
v2 = 1./sqrt(u.*e);
mur1x = ((v2.*dt-dx)./(v2.*dt+dx));
mur2x = (2*dx)./(v2.*dt+dx);
mur3x = (dx.*(v2.*dt).^2)./(2*dy^2.*(v2.*dt+dx));
%---%
mur1y = ((v2.*dt-dy)./(v2.*dt+dy));
mur2y = (2*dy)./(v2.*dt+dy);
mur3y = (dy.*(v2.*dt).^2)./(2*dx^2.*(v2.*dt+dy));
%-----%

```

```

%---- vetores para varredura ----%
i = 2:xmax-1;
j = 2:ymax-1;
%-----%

%----- inicio do calculo -----%
for n=2:nmax

disp(n);

%---- excitacao ----%
Ez(xexc,yexc,2)=Ez(xexc,yexc,2)+Vz(n);
%-----%

%---- campo magnetico ----%
Hx(i,j,3)=h1(i,j).*Hx(i,j,2)...
-h2(i,j).*(Ez(i,j,2)-Ez(i,j-1,2))./dy;

Hy(i,j,3)=h1(i,j).*Hy(i,j,2)...
-h2(i,j).*(-(Ez(i,j,2)-Ez(i-1,j,2))./dx);
%-----%

%---- campo eletrico ----%
Ez(i,j,3)=e1(i,j).*Ez(i,j,2)...
+e2(i,j).*(Hy(i+1,j,3)-Hy(i,j,3))./dx-(Hx(i,j+1,3)-Hx(i,j,3))./dy);
%-----%

%---- condicao de absorcao ----%
%---- 2Mur ----%
% xmin+1
Ez(bx1,j,3)=-Ez(bx1+1,j,1)...
+mur1x(bx1,j).*(Ez(bx1+1,j,3)+Ez(bx1,j,1))...
+mur2x(bx1,j).*(Ez(bx1,j,2)+Ez(bx1+1,j,2))...
+mur3x(bx1,j).*(Ez(bx1,j+1,2)-2.*Ez(bx1,j,2)+Ez(bx1,j-1,2)+...
Ez(bx1+1,j+1,2)-2.*Ez(bx1+1,j,2)+Ez(bx1+1,j-1,2));

% xmax-1
Ez(bx2,j,3)=-Ez(bx2-1,j,1)...
+mur1x(bx2,j).*(Ez(bx2-1,j,3)+Ez(bx2,j,1))...
+mur2x(bx2,j).*(Ez(bx2,j,2)+Ez(bx2-1,j,2))...
+mur3x(bx2,j).*(Ez(bx2,j+1,2)-2.*Ez(bx2,j,2)+Ez(bx2,j-1,2)+...
Ez(bx2-1,j+1,2)-2.*Ez(bx2-1,j,2)+Ez(bx2-1,j-1,2));

% ymin+1
Ez(i,by1,3)=-Ez(i,by1+1,1)...
+mur1y(i,by1).*(Ez(i,by1+1,3)+Ez(i,by1,1))...
+mur2y(i,by1).*(Ez(i,by1,2)+Ez(i,by1+1,2))...
+mur3y(i,by1).*(Ez(i+1,by1,2)-2.*Ez(i,by1,2)+Ez(i-1,by1,2)+...
Ez(i+1,by1+1,2)-2.*Ez(i,by1+1,2)+Ez(i-1,by1+1,2));

% ymax-1
Ez(i,by2,3)=-Ez(i,by2-1,1)...
+mur1y(i,by2).*(Ez(i,by2-1,3)+Ez(i,by2,1))...
+mur2y(i,by2).*(Ez(i,by2,2)+Ez(i,by2-1,2))...
+mur3y(i,by2).*(Ez(i+1,by2,2)-2.*Ez(i,by2,2)+Ez(i-1,by2,2)+...
Ez(i+1,by2-1,2)-2.*Ez(i,by2-1,2)+Ez(i-1,by2-1,2));

%-----%
%---- 1Mur ----%
Ez(bx1,by1,3)=Ez(bx1+1,by1,2)+mur1x(bx1,by1).*(Ez(bx1+1,by1,3)-Ez(bx1,by1,2));
Ez(bx2,by1,3)=Ez(bx2-1,by1,2)+mur1x(bx2,by1).*(Ez(bx2-1,by1,3)-Ez(bx2,by1,2));
Ez(bx1,by2,3)=Ez(bx1+1,by2,2)+mur1x(bx1,by2).*(Ez(bx1+1,by2,3)-Ez(bx1,by2,2));
Ez(bx2,by2,3)=Ez(bx2-1,by2,2)+mur1x(bx2,by2).*(Ez(bx2-1,by2,3)-Ez(bx2,by2,2));
%-----%

M(:,n)=Ez(:,n,3); % salva resultados do campo Ez

%---- restauracao das matrizes ----%
Ez(:,n,1)=Ez(:,n,2); Hx(:,n,1)=Hx(:,n,2); Hy(:,n,1)=Hy(:,n,2);
Ez(:,n,2)=Ez(:,n,3); Hx(:,n,2)=Hx(:,n,3); Hy(:,n,2)=Hy(:,n,3);
%-----%

end
%-----%

```

```

%---- apresenta resultados na tela ----%
figure(1);
hh=icontrol('Style','slider','Min',0,'Max',nmax-1,'SliderStep',[1/nmax 10/nmax],...
    'Callback',['vi=get(gca,'View');'...'
    'n=round(get(hh,'Value'))+1;'...'
    'surf(M(3:xmax-2,3:yymax-2,n)); axis ij;'...'
    'shading interp;'...'
    'set(gca,'View',vi);'...'
    'title(n);'...'
    ]);
set(gcf,'toolbar','figure');
%-----%

```



### (3) Campo\_Ez\_PML.m

```

%---- propriedades do vacuo ----%
eo = 8.854e-12;
uo = 4*pi*1e-7;
%-----%

%---- tamanho discretizacao do dominio ----%
xmax = 100;
ymax = 100;
nmax = 200;

dx = 1e-3;          % discretizacao espacial no eixo x
dy = 1e-3;          % discretizacao espacial no eixo y
%-----%

%--- descretizacao do tempo ---%
v = 1./sqrt(uo*eo);
dt = (1/sqrt((1/dx)^2+(1/dy)^2))/v;
%-----%

%---- matrizes dos campos ----%
Ez(1:xmax,1:ymax,1:2) = 0;
Hx(1:xmax,1:ymax,1:2) = 0;
Hy(1:xmax,1:ymax,1:2) = 0;

Dz(1:xmax,1:ymax,1:2) = 0;
Bx(1:xmax,1:ymax,1:2) = 0;
By(1:xmax,1:ymax,1:2) = 0;

M(1:xmax,1:ymax,1:nmax) = 0;          % matriz para salvar os resultados
%-----%

%---- propriedades do meio ----%
e(1:xmax,1:ymax) = eo;
u(1:xmax,1:ymax) = uo;
s(1:xmax,1:ymax) = 0;
pm(1:xmax,1:ymax) = 0;
%-----%

%-----PML-----%
er = max(max(e))/eo;          % obtem o er do meio
L = 10;          % espessura da camada PML
m = 4;          % fator polinomial
smax = (m+1)/(150*pi*dy*sqrt(er)); % condutividade do PML

txa = L;  txb = L;          % largura da PML no eixo x
tya = L;  tyb = L;          % largura da PML no eixo y

bx1 = 2+txa;  bx2 = (xmax-1)-txb; % posicao inicial e final da PML no eixo x
by1 = 2+tya;  by2 = (ymax-1)-tyb; % posicao inicial e final da PML no eixo y

ip1 = 2:bx1;  ip2=bx2:xmax-1;     % varredura da PML no eixo x
jp1 = 2:by1;  jp2=by2:ymax-1;     % varredura da PML no eixo y

sxpi = zeros(xmax,1,1);  sxpm = zeros(xmax,1,1); % vetores de condutividade
syphi = zeros(1,ymax,1);  syphi = zeros(1,ymax,1); % vetores de condutividade

sxpi(ip1) = smax*((bx1-ip1)./txa).^m; % condutividade no eixo x (posicoes "i")
sxpi(ip2) = smax*((ip2+0.5-bx2)./txb).^m; % condutividade no eixo x (posicoes "i")

sxpm(ip1) = smax*((bx1-ip1+0.5)./txb).^m; % condutividade no eixo x (posicoes "i-1/2")
sxpm(ip2) = smax*((ip2-bx2)./txb).^m; % condutividade no eixo x (posicoes "i-1/2")

syphi(jp1) = smax*((by1-jp1)./tya).^m; % condutividade no eixo y (posicoes "j")
syphi(jp2) = smax*((jp2+0.5-by2)./tyb).^m; % condutividade no eixo y (posicoes "j")

syphi(jp1) = smax*((by1-jp1+0.5)./tyb).^m; % condutividade no eixo y (posicoes "j-1/2")
syphi(jp2) = smax*((jp2-by2)./tyb).^m; % condutividade no eixo y (posicoes "j-1/2")

sxi = s+repmat(sxpi,[1 ymax]); % PML no eixo x + condutividade original do meio
sxm = s+repmat(sxpm,[1 ymax]); % PML no eixo x + condutividade original do meio

```

```

syi = s+repmat(sypi,[xmax 1]); % PML no eixo y + condutividade original do meio
sym = s+repmat(sypm,[xmax 1]); % PML no eixo y + condutividade original do meio

%-- coef. do PML --%
PML1xi = (1+sxi.*dt./2./eo);
PML2xi = (1-sxi.*dt./2./eo);
PML1xm = (1+sxm.*dt./2./eo);
PML2xm = (1-sxm.*dt./2./eo);

PML1yi = (1+syi.*dt./2./eo);
PML2yi = (1-syi.*dt./2./eo);
PML1ym = (1+sym.*dt./2./eo);
PML2ym = (1-sym.*dt./2./eo);
%-----%
%-----%

%---- coef. dos campos ----%
e1 = (1-s.*dt./(2.*e))./(1+s.*dt./(2.*e));
e2 = (dt./e)./(1+s.*dt./(2.*e));
h1 = (1-pm.*dt./(2.*u))./(1+pm.*dt./(2.*u));
h2 = (dt./u)./(1+pm.*dt./(2.*u));
%-----%

%---- excitacao ----%
n=1:nmax;
t=dt*n;
Lmax=max(20.*[dx dy]);
T=Lmax/v;
Vz(n) = exp(-((dt*n)-(3*T))./(T/3)).^2);

xexc=round(xmax/2);
yexc=round(ymax/2);
%-----%

%---- vetores para varredura ----%
i = 2:xmax-1;
j = 2:ymax-1;
%-----%

%----- inicio do calculo -----%
for n=2:nmax

    disp(n);

    p1 = mod(n+1,2)+1;
    p2 = mod(n,2)+1;

    %---- excitacao ----%
    Ez(xexc,yexc,p1)=Ez(xexc,yexc,p1)+Vz(n);
    %-----%

% algoritmo FDTD + PML
%-----%
Bx(i,j,p2)=(PML2ym(i,j)./PML1ym(i,j)).*Bx(i,j,p1)...
-(dt./PML1ym(i,j)).*(Ez(i,j,p1)-Ez(i,j-1,p1))./dy);

Hx(i,j,p2)=Hx(i,j,p1)...
+(1./u(i,j)).*(PML1xi(i,j).*Bx(i,j,p2)...
-PML2xi(i,j).*Bx(i,j,p1));
%-----%

%-----%
By(i,j,p2)=(PML2xm(i,j)./PML1xm(i,j)).*By(i,j,p1)...
-(dt./PML1xm(i,j)).*(-Ez(i,j,p1)-Ez(i-1,j,p1))./dx);

Hy(i,j,p2)=Hy(i,j,p1)...
+(1./u(i,j)).*(PML1yi(i,j).*By(i,j,p2)...
-PML2yi(i,j).*By(i,j,p1));
%-----%

```

```

%-----%
Dz(i,j,p2)=(PML2yi(i,j)./PML1yi(i,j)).*Dz(i,j,p1)...
            +(dt./PML1yi(i,j)).*(Hy(i+1,j,p2)-Hy(i,j,p2))./dx...
            -(Hx(i,j+1,p2)-Hx(i,j,p2))./dy);

Ez(i,j,p2)=(PML2xi(i,j)./PML1xi(i,j)).*Ez(i,j,p1)...
            +(1./e(i,j)./PML1xi(i,j)).*(Dz(i,j,p2)...
            -Dz(i,j,p1));
%-----%

%---- PEC's ----%
Ez(2,:,p2) = 0; Ez(xmax-1,:,p2) = 0;
Ez(:,2,p2) = 0; Ez(:,ymax-1,p2) = 0;
%-----%

M(:,:,n)=Ez(:,:,p2);           % salva resultados do campo Ez

end
%-----%

%---- apresenta resultados na tela ----%
figure;
hh=icontrol('Style','slider','Min',0,'Max',nmax-1,'SliderStep',[1/nmax 10/nmax],...
            'Callback',['vi=get(gca,'View');'...
                        'n=round(get(hh,'Value'))+1;'...
                        'surf(M(:,:,n)); axis ij;'...
                        'shading interp;'...
                        'set(gca,'View',vi);'...
                        'title(n);'...
                        ']);
set(gcf,'toolbar','figure');
%-----%

```

## (4) Campo\_Ez\_Teleport.m

```
% para mais fronteiras de Auto-Teleportação basta apenas
% copiar o núcleo "ABC Teleport field"

%---- propriedades do vacuo ----%
eo = 8.854e-12;
uo = 4*pi*1e-7;
%-----%

%---- tamanho discretizacao do dominio ----%
xmax = 100;
ymax = 100;

nmax = 200;

dx = 1e-3;          % discretizacao espacial no eixo x
dy = 1e-3;          % discretizacao espacial no eixo y
%-----%

%---- matrizes dos campos ----%
Ez(1:xmax,1:ymax,1:3) = 0;
Hx(1:xmax,1:ymax,1:3) = 0;
Hy(1:xmax,1:ymax,1:3) = 0;

M(1:xmax,1:ymax,1:nmax) = 0;      % matriz para salvar os resultados
%-----%

%---- propriedades do meio ----%
e(1:xmax,1:ymax) = eo;
u(1:xmax,1:ymax) = uo;
s(1:xmax,1:ymax) = 0;
pm(1:xmax,1:ymax) = 0;
%-----%

%--- descretizacao temporal ---%
v = 1./sqrt(uo*eo);
dt = (1/sqrt((1/dx)^2+(1/dy)^2))/v;
%-----%

%---- excitacao ----%
n = 1:nmax;
t = dt*n;
Lmax = max(10.*[dx dy]);
T = Lmax/v;
Vz(n) = exp(-((dt*n)-(3*T))./(T/3)).^2);

xexc = round(xmax/2);
yexc = round(ymax/2);
%-----%

%---- coef. dos campos ----%
e1 = (1-s.*dt./(2.*e))./(1+s.*dt./(2.*e));
e2 = (dt./e)./(1+s.*dt./(2.*e));
h1 = (1-pm.*dt./(2.*u))./(1+pm.*dt./(2.*u));
h2 = (dt./u)./(1+pm.*dt./(2.*u));
%-----%

%---- posicionamento das fronteiras ----%
bx1 = 1+3;          % xmin+3
by1 = 1+3;          % ymin+3
bx2 = xmax-3;      % xmax-3
by2 = ymax-3;      % ymax-3

R = 0.99; % coef. de amortecimento
%-----%

%---- vetores para varredura ----%
i = 2:xmax-1;
j = 2:ymax-1;
%-----%
```

```

%----- inicio do calculo -----%
for n=2:nmax

disp(n);

%---- campo magnetico ----%
Hx(i,j,3)=h1(i,j).*Hx(i,j,2)...
-h2(i,j).*(Ez(i,j,2)-Ez(i,j-1,2))./dy;

Hy(i,j,3)=h1(i,j).*Hy(i,j,2)...
-h2(i,j).*(-(Ez(i,j,2)-Ez(i-1,j,2))./dx);
%-----%

%---- ABC teleport field ----%
% xmin+3
Hy(bx1+1,j,3)=Hy(bx1+1,j,3)-R*(h2(bx1+1,j)./dx).*Ez(bx1+1,j,1);
% xmax-3
Hy(bx2,j,3)=Hy(bx2,j,3)+R*(h2(bx2-1,j)./dx).*Ez(bx2-1,j,1);
% ymin+3
Hx(i,by1+1,3)=Hx(i,by1+1,3)+R*(h2(i,by1+1)./dy).*Ez(i,by1+1,1);
% ymax-3
Hx(i,by2,3)=Hx(i,by2,3)-R*(h2(i,by2-1)./dy).*Ez(i,by2-1,1);
%-----%

%---- campo eletrico ----%
Ez(i,j,3)=e1(i,j).*Ez(i,j,2)...
+e2(i,j).*((Hy(i+1,j,3)-Hy(i,j,3))./dx-(Hx(i,j+1,3)-Hx(i,j,3))./dy);
%-----%

%---- exciacao ----%
Ez(xexc,yexc,3) = Ez(xexc,yexc,3) + Vz(n);
%-----%

%---- ABC teleport field ----%
% xmin+3
Ez(bx1,j,3)=Ez(bx1,j,3)-R*(e2(bx1+2,j)./dx).*Hy(bx1+2,j,2);
% xmax-3
Ez(bx2,j,3)=Ez(bx2,j,3)+R*(e2(bx2-1,j)./dx).*Hy(bx2-1,j,2);
% ymin+3
Ez(i,by1,3)=Ez(i,by1,3)+R*(e2(i,by1+2)./dy).*Hx(i,by1+2,2);
% ymax-3
Ez(i,by2,3)=Ez(i,by2,3)-R*(e2(i,by2-1)./dy).*Hx(i,by2-1,2);
%-----%

%---- PEC's ----%
Ez(2,:,3)=0; Ez(xmax-1,:,3)=0;
Ez(:,2,3)=0; Ez(:,ymax-1,3)=0;
%-----%

M(:,n)=Ez(:,n); % salva resultados do campo Ez

%---- restauracao das matrizes ----%
Ez(:,1)=Ez(:,2); Hx(:,1)=Hx(:,2); Hy(:,1)=Hy(:,2);
Ez(:,2)=Ez(:,3); Hx(:,2)=Hx(:,3); Hy(:,2)=Hy(:,3);
%-----%

end
%-----%

%---- apresenta resultados na tela ----%
figure(1);
hh=uicontrol('Style','slider','Min',0,'Max',nmax-1,'SliderStep',[1/nmax 10/nmax],...
'Callback',['vi=get(gca,'View');'...
'n=round(get(hh,'Value'))+1;'...
'surf(M(3:xmax-2,3:ymax-2,n)); axis ij;'...
'shading interp;'...
'set(gca,'View',vi);'...
'title(n);'...
]);
set(gcf,'toolbar','figure');
%-----%

```

## (5) Campo\_Ez\_Teleport\_atraso.m

```

%---- propriedades do vacuo ----%
eo = 8.854e-12;
uo = 4*pi*1e-7;
%-----%

%---- tamanho discretizacao do dominio ----%
xmax = 100;
ymax = 100;

nmax = 200;

dx = 1e-3;          % discretizacao espacial no eixo x
dy = 1e-3;          % discretizacao espacial no eixo y
%-----%

%---- matrizes dos campos ----%
Ez(1:xmax,1:ymax,1:3) = 0;
Hx(1:xmax,1:ymax,1:3) = 0;
Hy(1:xmax,1:ymax,1:3) = 0;

M(1:xmax,1:ymax,1:nmax) = 0;          % matriz para salvar os resultados
%-----%

%---- propriedades do meio ----%
e(1:xmax,1:ymax) = eo;
u(1:xmax,1:ymax) = uo;
s(1:xmax,1:ymax) = 0;
pm(1:xmax,1:ymax) = 0;
%-----%

%--- descretizacao temporal ---%
v = 1./sqrt(uo*eo);
dt = (1/sqrt((1/dx)^2+(1/dy)^2))/v;
%-----%

%---- excitacao ----%
n = 1:nmax;
t = dt*n;
Lmax = max(10.*[dx dy]);
T = Lmax/v;
Vz(n) = exp(-(((dt*n)-(3*T))./(T/3)).^2);

xexc = round(xmax/2);
yexc = round(ymax/2);
%-----%

%---- coef. dos campos ----%
e1 = (1-s.*dt./(2.*e))./(1+s.*dt./(2.*e));
e2 = (dt./e)./(1+s.*dt./(2.*e));
h1 = (1-pm.*dt./(2.*u))./(1+pm.*dt./(2.*u));
h2 = (dt./u)./(1+pm.*dt./(2.*u));
%-----%

%---- posicionamento das fronteiras ----%
bx1 = 1+3;          % xmin+3
by1 = 1+3;          % ymin+3
bx2 = xmax-3;      % xmax-3
by2 = ymax-3;      % ymax-3

R = 0.99;          % coef. de amortecimento;
a = sqrt(2);      % atraso temporal (muda esse valor para se adequar ao problema)
m = floor(a);
%-----%

%---- vetores para varredura ----%
i = 2:xmax-1;
j = 2:ymax-1;
%-----%

```

```

%----- inicio do calculo -----%
for n=2:nmax

    disp(n);

%---- exciacao ----%
    Ez(xexc,yexc,3)=Ez(xexc,yexc,3)+Vz(n);
%-----%

%---- campo magnetico ----%
    Hx(i,j,4)=h1(i,j).*Hx(i,j,3)...
        -h2(i,j).*(Ez(i,j,3)-Ez(i,j-1,3))./dy;

    Hy(i,j,4)=h1(i,j).*Hy(i,j,3)...
        -h2(i,j).*(-(Ez(i,j,3)-Ez(i-1,j,3))./dx);
%-----%

%---- ABC teleport field ----%
    % xmin+3
    Hy(bx1+1,j,4)=Hy(bx1+1,j,4)-R*(h2(bx1+1,j)./dx).*((m+1-a)*Ez(bx1+1,j,3-m)+(a-
m)*Ez(bx1+1,j,3-(m+1)));

    % xmax-3
    Hy(bx2,j,4)=Hy(bx2,j,4)+R*(h2(bx2-1,j)./dx).*((m+1-a)*Ez(bx2-1,j,3-m)+(a-m)*Ez(bx2-
1,j,3-(m+1)));

    % ymin+3
    Hx(i,by1+1,4)=Hx(i,by1+1,4)+R*(h2(i,by1+1)./dy).*((m+1-a)*Ez(i,by1+1,3-m)+(a-
m)*Ez(i,by1+1,3-(m+1)));

    % ymax-3
    Hx(i,by2,4)=Hx(i,by2,4)-R*(h2(i,by2-1)./dy).*((m+1-a)*Ez(i,by2-1,3-m)+(a-
m)*Ez(i,by2-1,3-(m+1)));
%-----%

%---- campo eletrico ----%
    Ez(i,j,4)=e1(i,j).*Ez(i,j,3)...
        +e2(i,j).*(Hy(i+1,j,4)-Hy(i,j,4))./dx-(Hx(i,j+1,4)-Hx(i,j,4))./dy);
%-----%

%---- ABC teleport field ----%
    % xmin+3
    Ez(bx1 ,j,4)=Ez(bx1 ,j,4)-R*(e2(bx1+2,j)./dx).*((m+1-a)*Hy(bx1+2,j,4-m)+(a-
m)*Hy(bx1+2,j,4-(m+1)));

    % xmax-3
    Ez(bx2,j,4)=Ez(bx2,j,4)+R*(e2(bx2-1,j)./dx).*((m+1-a)*Hy(bx2-1,j,4-m)+(a-m)*Hy(bx2-
1,j,4-(m+1)));

    % ymin-3
    Ez(i,by1 ,4)=Ez(i,by1 ,4)+R*(e2(i,by1+2)./dy).*((m+1-a)*Hx(i,by1+2,4-m)+(a-
m)*Hx(i,by1+2,4-(m+1)));

    % ymax-3
    Ez(i,by2,4)=Ez(i,by2,4)-R*(e2(i,by2-1)./dy).*((m+1-a)*Hx(i,by2-1,4-m)+(a-
m)*Hx(i,by2-1,4-(m+1)));
%-----%

%---- PEC's ----%
    Ez(2,:,4)=0; Ez(xmax-1,:,4)=0;
    Ez(:,2,4)=0; Ez(:,ymax-1,4)=0;
%-----%

    M(:, :, n)=Ez(:, :, 4); % salva resultados do campo Ez

%---- restauracao das matrizes ----%
    Ez(:, :, 1)=Ez(:, :, 2); Hx(:, :, 1)=Hx(:, :, 2); Hy(:, :, 1)=Hy(:, :, 2);
    Ez(:, :, 2)=Ez(:, :, 3); Hx(:, :, 2)=Hx(:, :, 3); Hy(:, :, 2)=Hy(:, :, 3);
    Ez(:, :, 3)=Ez(:, :, 4); Hx(:, :, 3)=Hx(:, :, 4); Hy(:, :, 3)=Hy(:, :, 4);
%-----%

end
%-----%

```

```

%---- apresenta resultados na tela ----%
figure;
hh=uicontrol('Style','slider','Min',0,'Max',nmax-1,'SliderStep',[1/nmax 10/nmax],...
    'Callback',['vi=get(gca,'View');'...
        'n=round(get(hh,'Value'))+1;'...
        'surf(M(3:xmax-2,3:ymax-2,n)); axis ij;'...
        'shading interp;'...
        'set(gca,'View',vi);'...
        'title(n);'...
    ]);
set(gcf,'toolbar','figure');
%-----%

```



## (6) Campo\_Ez\_Teleport\_subgrade.m

```

%---- propriedades do vacuo ----%
eo = 8.854e-12;
uo = 4*pi*1e-7;
%-----%

%---- tamanho e discretização do dominio ----%
xmax = 100;
ymax = 100;

nmax = 400;

dx = 1e-3;          % discretizacao espacial no eixo x
dy = 1e-3;          % discretizacao espacial no eixo y
%-----%

%---- matrizes dos campos ----%
Ez(1:xmax,1:ymax,1:3) = 0;
Hx(1:xmax,1:ymax,1:3) = 0;
Hy(1:xmax,1:ymax,1:3) = 0;

M(1:xmax,1:ymax,1:nmax) = 0;          % matriz para salvar os resultados
%-----%

%---- propriedades do meio ----%
e(1:xmax,1:ymax) = eo;
u(1:xmax,1:ymax) = uo;
s(1:xmax,1:ymax) = 0;
pm(1:xmax,1:ymax) = 0;
%-----%

%---- posicionamento das fronteiras ----%
bx1 = 1+3;          % xmin+3
by1 = 1+3;          % ymin+3
bx2 = xmax-3;      % xmax-3
by2 = ymax-3;      % ymax-3

R = 0.95; % coef. de amortecimento;
%-----%

%---- montagem das celulas não uniformes ----%
d(1:xmax,1:ymax)=dy;% matrix com discretizações espaciais

m = 2;          % fator de redução das celulas não uniformes

d(bx1,:) = dy/m; d(bx2,:)=dy/m;
d(:,by1) = dy/m; d(:,by2)=dy/m;

d(bx1+1,:) = dy/m; d(bx2-1,:)=dy/m;
d(:,by1+1) = dy/m; d(:,by2-1)=dy/m;
%-----%

%--- descretizacao do tempo ---%
v_onda=min(min(1./sqrt(u.*e)));
v=1./sqrt(uo*eo);
dt=(1/sqrt((1/min(min(d))^2+(1/min(min(d))^2)))/v);
%-----%

%---- exciacao ----%
n = 1:nmax;
t = dt*n;
Lmax = max(10.*[dx dy]);
T = Lmax/v;
Vz(n) = exp(-(((dt*n)-(3*T))./(T/3)).^2);

xexc = round(xmax/2);
yexc = round(ymax/2);
%-----%

```

```

%---- coef. dos campos ----%
e1 = (1-s.*dt./(2.*e))./(1+s.*dt./(2.*e));
e2 = (dt./e)./(1+s.*dt./(2.*e));
h1 = (1-pm.*dt./(2.*u))./(1+pm.*dt./(2.*u));
h2 = (dt./u)./(1+pm.*dt./(2.*u));
%-----%

%---- vetores para varredura ----%
i = 2:xmax-1;
j = 2:ymin-1;
%-----%

%----- inicio do calculo -----%
for n=2:nmax

    disp(n);

%---- exciacao ----%
    Ez(xexc,yexc,2) = Ez(xexc,yexc,2) + Vz(n);
%-----%

%---- campo magnetico ----%
    Hx(i,j,3)=h1(i,j).*Hx(i,j,2)...
        -h2(i,j).*(Ez(i,j,2)-Ez(i,j-1,2))./d(i,j);

    Hy(i,j,3)=h1(i,j).*Hy(i,j,2)...
        -h2(i,j).*(-(Ez(i,j,2)-Ez(i-1,j,2))./d(i,j));
%-----%

%---- ABC teleport field ----%
    % xmin+3
    Hy(bx1+1,j,3)=Hy(bx1+1,j,3)-R*(h2(bx1+1,j)./d(bx1+1,j)).*Ez(bx1+1,j,1);
    % xmax-3
    Hy(bx2,j,3)=Hy(bx2,j,3)+R*(h2(bx2-1,j)./d(bx2-1,j)).*Ez(bx2-1,j,1);
    % ymin+3
    Hx(i,by1+1,3)=Hx(i,by1+1,3)+R*(h2(i,by1+1)./d(i,by1+1)).*Ez(i,by1+1,1);
    % ymax-3
    Hx(i,by2,3)=Hx(i,by2,3)-R*(h2(i,by2-1)./d(i,by2-1)).*Ez(i,by2-1,1);
%-----%

%---- campo eletrico ----%
    Ez(i,j,3)=e1(i,j).*Ez(i,j,2)...
        +e2(i,j).*((Hy(i+1,j,3)-Hy(i,j,3))./(d(i,j)/2+d(i+1,j)/2)...
            -(Hx(i,j+1,3)-Hx(i,j,3))./(d(i,j)/2+d(i,j+1)/2));
%-----%

%---- ABC teleport field ----%
    % xmin+3
    Ez(bx1,j,3)=Ez(bx1,j,3)-R*(e2(bx1+2,j)./(d(bx1,j)/2+d(bx1+1,j)/2)).*Hy(bx1+2,j,2);
    % xmax-3
    Ez(bx2,j,3)=Ez(bx2,j,3)+R*(e2(bx2-1,j)./(d(bx2,j)/2+d(bx2+1,j)/2)).*Hy(bx2-1,j,2);
    % ymin+3
    Ez(i,by1,3)=Ez(i,by1,3)+R*(e2(i,by1+2)./(d(i,by1)/2+d(i,by1+1)/2)).*Hx(i,by1+2,2);
    % ymax-3
    Ez(i,by2,3)=Ez(i,by2,3)-R*(e2(i,by2-1)./(d(i,by2)/2+d(i,by2+1)/2)).*Hx(i,by2-1,2);
%-----%

%---- PEC's ----%
    Ez(2,:,3)=0; Ez(xmax-1,:,3)=0;
    Ez(:,2,3)=0; Ez(:,ymax-1,3)=0;
%-----%

    M(:,n)=Ez(:,n); % salva resultados do campo Ez

%---- restauracao das matrizes ----%
    Ez(:,n,1)=Ez(:,n,2); Hx(:,n,1)=Hx(:,n,2); Hy(:,n,1)=Hy(:,n,2);
    Ez(:,n,2)=Ez(:,n,3); Hx(:,n,2)=Hx(:,n,3); Hy(:,n,2)=Hy(:,n,3);
%-----%

end
%-----%

```

```

%---- apresenta resultados na tela ----%
figure(1);
hh=uicontrol('Style','slider','Min',0,'Max',nmax-1,'SliderStep',[1/nmax 10/nmax],...
    'Callback',['vi=get(gca,'View');'...
        'n=round(get(hh,'Value'))+1;'...
        'surf(M(:,:,n)); axis ij;'...
        'shading interp;'...
        'set(gca,'View',vi);'...
        'title(n);'...
    ]);
set(gcf,'toolbar','figure');
%-----%

```

## (7) Campo\_Ez\_Teleport\_equação\_onda.m

```

%---- propriedades do vacuo ----%
eo = 8.854e-12;
uo = 4*pi*1e-7;
%-----%

%---- tamanho e discretização do dominio ----%
xmax = 100;
ymax = 100;

nmax = 200;

dx = 1e-3;          % discretizacao espacial no eixo x
dy = 1e-3;          % discretizacao espacial no eixo y
%-----%

%---- matrizes dos campos ----%
Ez(1:xmax,1:ymax,1:3) = 0;
Hx(1:xmax,1:ymax,1:3) = 0;
Hy(1:xmax,1:ymax,1:3) = 0;

M(1:xmax,1:ymax,1:nmax) = 0;          % matriz para salvar os resultados
%-----%

%---- propriedades do meio ----%
e(1:xmax,1:ymax) = eo;
u(1:xmax,1:ymax) = uo;
s(1:xmax,1:ymax) = 0;
pm(1:xmax,1:ymax) = 0;
%-----%

%--- descretizacao temporal ---%
v = 1./sqrt(uo*eo);
dt = (1/sqrt((1/dx)^2+(1/dy)^2))/v;
%-----%

%---- exciacao -----%
n = 1:nmax;
t = dt*n;
Lmax = max(10.*[dx dy]);
T = Lmax/v;
Vz(n) = exp(-(((dt*n)-(3*T))./(T/3)).^2);

xexc = round(xmax/2);
yexc = round(ymax/2);
%-----%

%---- coef. dos campos ----%
e1 = (1-s.*dt./(2.*e))./(1+s.*dt./(2.*e));
e2 = (dt./e)./(1+s.*dt./(2.*e));
h1 = (1-pm.*dt./(2.*u))./(1+pm.*dt./(2.*u));
h2 = (dt./u)./(1+pm.*dt./(2.*u));
%-----%

%---- posicionamento das fronteiras ----%
bx1 = 1+3;          % xmin+3
by1 = 1+3;          % ymin+3
bx2 = xmax-3;      % xmax-3
by2 = ymax-3;      % ymax-3

R = 0.99; % coef. de amortecimento;
%-----%

%---- buffer de predicao ----%
Eze(1:xmax,3,1:3)=0;
Hxe(1:xmax,3,1:3)=0;
%-----%
Ezd(1:xmax,3,1:3)=0;
Hxd(1:xmax,3,1:3)=0;
%-----%
Ezt(3,1:ymax,1:3)=0;

```

```

Hyt(3,1:ymax,1:3)=0;
%----%
Ezf(3,1:ymax,1:3)=0;
Hyf(3,1:ymax,1:3)=0;
%----%
%-----%

%---- alteração por Higdon ----%
v2=(1./sqrt(u.*e))./cos((pi/180)*45);
%----%
murt(1,:)=(v2(bx1,:)*dt-dx)./(v2(bx1,:)*dt+dx);
murf(1,:)=(v2(bx2,:)*dt-dx)./(v2(bx2,:)*dt+dx);
mure(:,1)=(v2(:,by1)*dt-dy)./(v2(:,by1)*dt+dy);
murd(:,1)=(v2(:,by2)*dt-dy)./(v2(:,by2)*dt+dy);
%----%
%-----%

%---- vetores para varredura ----%
i = 2:xmax-1;
j = 2:ymin-1;
%-----%

%----- inicio do calculo -----%
for n=2:nmax

    disp(n);

%---- exciacao ----%
Ez(xexc,yexc,2)=Ez(xexc,yexc,2)+Vz(n);
%-----%

%---- campo magnetico ----%
Hx(i,j,3)=h1(i,j).*Hx(i,j,2)...
-h2(i,j).*(Ez(i,j,2)-Ez(i,j-1,2))./dy;

Hy(i,j,3)=h1(i,j).*Hy(i,j,2)...
-h2(i,j).*(-(Ez(i,j,2)-Ez(i-1,j,2)))./dx);
%-----%

%---- ABC teleport field ----%
% xmin+3
Ezt(1,j,2)=(Ez(bx1+1,j,1)+murt(1,j).*(Ez(bx1+1,j,2)-Ezt(1,j,1)));
Hy(bx1+1,j,3)=Hy(bx1+1,j,3)-R*(h2(bx1+1,j))./dx).*Ezt(1,j,2);

% xmax-3
Ezf(1,j,2)=(Ez(bx2-1,j,1)+murt(1,j).*(Ez(bx2-1,j,2)-Ezf(1,j,1)));
Hy(bx2,j,3)=Hy(bx2,j,3)+R*(h2(bx2-1,j))./dx).*Ezf(1,j,2);

% ymin+3
Eze(i,1,2)=(Ez(i,by1+1,1)+mure(i,1).*(Ez(i,by1+1,2)-Eze(i,1,1)));
Hx(i,by1+1,3)=Hx(i,by1+1,3)+R*(h2(i,by1+1))./dy).*Eze(i,1,2);

% ymin+3
Ezd(i,1,2)=(Ez(i,by2-1,1)+murd(i,1).*(Ez(i,by2-1,2)-Ezd(i,1,1)));
Hx(i,by2,3)=Hx(i,by2,3)-R*(h2(i,by2-1))./dy).*Ezd(i,1,2);
%-----%

%---- campo eletrico ----%
Ez(i,j,3)=e1(i,j).*Ez(i,j,2)...
+e2(i,j).*((Hy(i+1,j,3)-Hy(i,j,3))./dx-(Hx(i,j+1,3)-Hx(i,j,3))./dy);
%-----%

%---- ABC teleport field ----%
% xmin+3
Hyt(1,j,3)=(Hy(bx1+2,j,2)+murt(1,j).*(Hy(bx1+2,j,3)-Hyt(1,j,2)));
Ez(bx1 ,j,3)=Ez(bx1 ,j,3)-R*(e2(bx1+2,j))./dx).*Hyt(1,j,3);

% xmax-3
Hyf(1,j,3)=(Hy(bx2-1,j,2)+murt(1,j).*(Hy(bx2-1,j,3)-Hyf(1,j,2)));
Ez(bx2,j,3)=Ez(bx2,j,3)+R*(e2(bx2-1,j))./dx).*Hyf(1,j,3);

% ymin+3
Hxe(i,1,3)=(Hx(i,by1+2,2)+mure(i,1).*(Hx(i,by1+2,3)-Hxe(i,1,2)));
Ez(i,by1 ,3)=Ez(i,by1 ,3)+R*(e2(i,by1+2))./dy).*Hxe(i,1,3);

```

```

% ymax-3
Hxd(i,1,3)=(Hx(i,by2-1,2)+murd(i,1).*(Hx(i,by2-1,3)-Hxd(i,1,2)));
Ez(i,by2,3)=Ez(i,by2,3)-R*(e2(i,by2-1)./dy).*Hxd(i,1,3);
%-----%

%---- PEC's ----%
Ez(2,:,3)=0; Ez(xmax-1,:,3)=0;
Ez(:,2,3)=0; Ez(:,ymax-1,3)=0;
%-----%

M(:, :, n)=Ez(:, :, 3); % salva resultados do campo Ez

%---- restauracao das matrizes ----%
Ez(:, :, 1)=Ez(:, :, 2); Hx(:, :, 1)=Hx(:, :, 2); Hy(:, :, 1)=Hy(:, :, 2);
Ez(:, :, 2)=Ez(:, :, 3); Hx(:, :, 2)=Hx(:, :, 3); Hy(:, :, 2)=Hy(:, :, 3);
%-----%

end
%-----%

%---- apresenta resultados na tela ----%
figure(1);
hh=uicontrol('Style','slider','Min',0,'Max',nmax-1,'SliderStep',[1/nmax 10/nmax],...
'Callback',['vi=get(gca,'View');'...
'n=round(get(hh,'Value'))+1;'...
'surf(M(:, :, n)); axis ij;'...
'shading interp;'...
'set(gca,'View',vi);'...
'title(n);'...
]);
set(gcf,'toolbar','figure');
%-----%

```

## (8) Campo3D\_1mur.m

```

%---- propriedades do vacuo ----%
eo = 8.854e-12;
uo = 4*pi*1e-7;
%-----%

%---- tamanho e discretizacao do dominio ----%
xmax = 50;
ymax = 50;
zmax = 50;

nmax = 100;

dx = 1e-3;          % discretizacao espacial no eixo x
dy = 1e-3;          % discretizacao espacial no eixo y
dz = 1e-3;

%-----%

%---- matrizes dos campos ----%
Ex(1:xmax,1:ymax,1:zmax,1:3)=0;
Ey(1:xmax,1:ymax,1:zmax,1:3)=0;
Ez(1:xmax,1:ymax,1:zmax,1:3)=0;
Hx(1:xmax,1:ymax,1:zmax,1:3)=0;
Hy(1:xmax,1:ymax,1:zmax,1:3)=0;
Hz(1:xmax,1:ymax,1:zmax,1:3)=0;

Mxy(1:xmax,1:ymax,1,1:nmax)=0; % matriz para salvar os resultados (plano xy)
Mxz(1:xmax,1,1:zmax,1:nmax)=0; % matriz para salvar os resultados (plano xz)
Myz(1,1:ymax,1:zmax,1:nmax)=0; % matriz para salvar os resultados (plano yz)
%-----%

%---- propriedades do meio ----%
e(1:xmax,1:ymax,1:zmax)=eo;
u(1:xmax,1:ymax,1:zmax)=uo;
s(1:xmax,1:ymax,1:zmax)=0;
pm(1:xmax,1:ymax,1:zmax)=0;
%-----%

%--- descretizacao temporal ---%
v = 1/sqrt(uo*eo);
dt = (1/sqrt((1/dx)^2+(1/dy)^2+(1/dz)^2))/v;
%-----%

%---- excitação ----%
Vz(2:62)=sin(2*pi*(0:60)/60).^3;
Vz(63:nmax)=0;

xexc = round(xmax/2);
yexc = round(ymax/2);
zexc = round(zmax/2);
%-----%

%---- coef. dos campos ----%
e1 = (1-s.*dt./(2.*e))./(1+s.*dt./(2.*e));
e2 = (dt./e)./(1+s.*dt./(2.*e));
h1 = (1-pm.*dt./(2.*u))./(1+pm.*dt./(2.*u));
h2 = (dt./u)./(1+pm.*dt./(2.*u));
%-----%

%---- posicionamento das fronteiras ----%
bx1 = 0+1;          % xmin+1
by1 = 0+1;          % ymin+1
bz1 = 0+1;          % zmax+1

bx2 = xmax-1;      % xmax-1
by2 = ymax-1;      % ymax-1
bz2 = zmax-1;      % zmax-1
%-----%

```

```

%---- coef. lo Mur ----%
v2=1./sqrt(u.*e);
murlx = (v2.*dt-dx)/(v2.*dt+dx);
murlly = (v2.*dt-dy)/(v2.*dt+dy);
murlz = (v2.*dt-dz)/(v2.*dt+dz);
%-----%

%---- vetores para varredura ----%
i = 2:xmax-1;
j = 2:ymin-1;
k = 2:zmax-1;
%-----%

%----- inicio do calculo -----%
for n=2:nmax

disp(n);

%---- excitacao ----%
Ez(xexc,yexc,zexc,1) = Ez(xexc,yexc,zexc,1) + Vz(n);
%-----%

%---- campo magnetico ----%
Hx(i,j,k,2)=h1(i,j,k).*Hx(i,j,k,1)...
-h2(i,j,k).*((Ez(i,j,k,1)-Ez(i,j-1,k,1))./dy-(Ey(i,j,k,1)-Ey(i,j,k-1,1))./dz);
Hy(i,j,k,2)=h1(i,j,k).*Hy(i,j,k,1)...
-h2(i,j,k).*((Ex(i,j,k,1)-Ex(i,j,k-1,1))./dz-(Ez(i,j,k,1)-Ez(i,j,k,1))./dx);
Hz(i,j,k,2)=h1(i,j,k).*Hz(i,j,k,1)...
-h2(i,j,k).*((Ey(i,j,k,1)-Ey(i-1,j,k,1))./dx-(Ex(i,j,k,1)-Ex(i,j-1,k,1))./dy);
%-----%

%---- campo eletrico ----%
Ex(i,j,k,2)=e1(i,j,k).*Ex(i,j,k,1)...
+e2(i,j,k).*((Hz(i,j+1,k,2)-Hz(i,j,k,2))./dy-(Hy(i,j,k+1,2)-Hy(i,j,k,2))./dz);
Ey(i,j,k,2)=e1(i,j,k).*Ey(i,j,k,1)...
+e2(i,j,k).*((Hx(i,j,k+1,2)-Hx(i,j,k,2))./dz-(Hz(i+1,j,k,2)-Hz(i,j,k,2))./dx);
Ez(i,j,k,2)=e1(i,j,k).*Ez(i,j,k,1)...
+e2(i,j,k).*((Hy(i+1,j,k,2)-Hy(i,j,k,2))./dx-(Hx(i,j+1,k,2)-Hx(i,j,k,2))./dy);
%-----%

%---- condicao de absorcao ----%
% xmin+1
Ey(bx1,j,k,2)=Ey(bx1+1,j,k,1)+murlx(bx1,j,k).*(Ey(bx1+1,j,k,2)-Ey(bx1,j,k,1));
Ez(bx1,j,k,2)=Ez(bx1+1,j,k,1)+murlx(bx1,j,k).*(Ez(bx1+1,j,k,2)-Ez(bx1,j,k,1));

% ymin+1
Ex(i,by1,k,2)=Ex(i,by1+1,k,1)+murlly(i,by1,k).*(Ex(i,by1+1,k,2)-Ex(i,by1,k,1));
Ez(i,by1,k,2)=Ez(i,by1+1,k,1)+murlly(i,by1,k).*(Ez(i,by1+1,k,2)-Ez(i,by1,k,1));

% zmin+1
Ex(i,j,bz1,2)=Ex(i,j,bz1+1,1)+murlz(i,j,bz1).*(Ex(i,j,bz1+1,2)-Ex(i,j,bz1,1));
Ey(i,j,bz1,2)=Ey(i,j,bz1+1,1)+murlz(i,j,bz1).*(Ey(i,j,bz1+1,2)-Ey(i,j,bz1,1));

% xmax-1
Ey(bx2,j,k,2)=Ey(bx2-1,j,k,1)+murlx(bx2,j,k).*(Ey(bx2-1,j,k,2)-Ey(bx2,j,k,1));
Ez(bx2,j,k,2)=Ez(bx2-1,j,k,1)+murlx(bx2,j,k).*(Ez(bx2-1,j,k,2)-Ez(bx2,j,k,1));

% ymax-1
Ex(i,by2,k,2)=Ex(i,by2-1,k,1)+murlly(i,by2,k).*(Ex(i,by2-1,k,2)-Ex(i,by2,k,1));
Ez(i,by2,k,2)=Ez(i,by2-1,k,1)+murlly(i,by2,k).*(Ez(i,by2-1,k,2)-Ez(i,by2,k,1));

% zmax-1
Ex(i,j,bz2,2)=Ex(i,j,bz2-1,1)+murlz(i,j,bz2).*(Ex(i,j,bz2-1,2)-Ex(i,j,bz2,1));
Ey(i,j,bz2,2)=Ey(i,j,bz2-1,1)+murlz(i,j,bz2).*(Ey(i,j,bz2-1,2)-Ey(i,j,bz2,1));
%-----%

```



```

%---- salva resultados do campo eletrico ----%
Mxy(:, :, 1, n)=sqrt(Ex(:, :, zexc, 2).^2+Ey(:, :, zexc, 2).^2+Ez(:, :, zexc, 2).^2);
Mxz(:, 1, :, n)=sqrt(Ex(:, yexc, :, 2).^2+Ey(:, yexc, :, 2).^2+Ez(:, yexc, :, 2).^2);
Myz(1, :, :, n)=sqrt(Ex(xexc, :, :, 2).^2+Ey(xexc, :, :, 2).^2+Ez(xexc, :, :, 2).^2);
%-----%

%---- restauracao das matrizes ----%
Ex(:, :, :, 1)=Ex(:, :, :, 2);    Ey(:, :, :, 1)=Ey(:, :, :, 2);    Ez(:, :, :, 1)=Ez(:, :, :, 2);
Hx(:, :, :, 1)=Hx(:, :, :, 2);    Hy(:, :, :, 1)=Hy(:, :, :, 2);    Hz(:, :, :, 1)=Hz(:, :, :, 2);
%-----%

end
%-----%

%---- apresenta resultados na tela ----%
Vol(1:xmax,1:ymax,1:zmax,1)=0; % aloca um volume para exibição

figure(1);
h=uicontrol('Style','slider','Min',0,'Max',nmax-1,'SliderStep',[1/nmax 10/nmax],...
    'Callback',[ 'vi=get(gca,'View');'...
                 'n=round(get(h,'Value'))+1;'...
                 'Vol(:, :, zexc)=Mxy(:, :, 1, n);'...
                 'Vol(:, yexc, :)=Mxz(:, 1, :, n);'...
                 'Vol(xexc, :, :)=Myz(1, :, :, n);'...
                 'slice(Vol,yexc,xexc,zexc);'...
                 'xlabel(''Y''); ylabel(''X''); zlabel(''Z'');'...
                 'shading interp; axis ij; axis equal;'...
                 'set(gca,'View',vi);'...
                 'title(n);'...
    ]);

mf = uimenu('Label','ferramentas');
uimenu(mf,'Label','zoom','Callback','zoom on; rotate3d off');
uimenu(mf,'Label','rotacao','Callback','zoom off; rotate3d on;');

set(gcf,'toolbar','figure');
%-----%

```

## (9) Campo3D\_2mur.m

```

%---- propriedades do vacuo ----%
eo = 8.854e-12;
uo = 4*pi*1e-7;
%-----%

%---- tamanho e discretizacao do dominio ----%
xmax = 50;
ymax = 50;
zmax = 50;

nmax = 100;

dx = 1e-3;          % discretizacao espacial no eixo x
dy = 1e-3;          % discretizacao espacial no eixo y
dz = 1e-3;

%-----%

%---- matrizes dos campos ----%
Ex(1:xmax,1:ymax,1:zmax,1:3)=0;
Ey(1:xmax,1:ymax,1:zmax,1:3)=0;
Ez(1:xmax,1:ymax,1:zmax,1:3)=0;
Hx(1:xmax,1:ymax,1:zmax,1:3)=0;
Hy(1:xmax,1:ymax,1:zmax,1:3)=0;
Hz(1:xmax,1:ymax,1:zmax,1:3)=0;

Mxy(1:xmax,1:ymax,1,1:nmax)=0; % matriz para salvar os resultados (plano xy)
Mxz(1:xmax,1,1:zmax,1:nmax)=0; % matriz para salvar os resultados (plano xz)
Myz(1,1:ymax,1:zmax,1:nmax)=0; % matriz para salvar os resultados (plano yz)
%-----%

%---- propriedades do meio ----%
e(1:xmax,1:ymax,1:zmax)=eo;
u(1:xmax,1:ymax,1:zmax)=uo;
s(1:xmax,1:ymax,1:zmax)=0;
pm(1:xmax,1:ymax,1:zmax)=0;
%-----%

%--- descretizacao temporal ---%
v = 1/sqrt(uo*eo);
dt = (1/sqrt((1/dx)^2+(1/dy)^2+(1/dz)^2))/v;
%-----%

%---- excitação ----%
Vz(2:62)=sin(2*pi*(0:60)/60).^3;
Vz(63:nmax)=0;

xexc = round(xmax/2);
yexc = round(ymax/2);
zexc = round(zmax/2);
%-----%

%---- coef. dos campos ----%
e1 = (1-s.*dt./(2.*e))./(1+s.*dt./(2.*e));
e2 = (dt./e)./(1+s.*dt./(2.*e));
h1 = (1-pm.*dt./(2.*u))./(1+pm.*dt./(2.*u));
h2 = (dt./u)./(1+pm.*dt./(2.*u));
%-----%

%---- posicionamento das fronteiras ----%
bx1 = 0+1;          % xmin+1
by1 = 0+1;          % ymin+1
bz1 = 0+1;          % zmax+1

bx2 = xmax-1;      % xmax-1
by2 = ymax-1;      % ymax-1
bz2 = zmax-1;      % zmax-1
%-----%

```

```

%---- coef.s de Mur ----%
v2=1./sqrt(u.*e);
%----%
mur1x=((v2.*dt-dx)./(v2.*dt+dx));
mur1y=((v2.*dt-dy)./(v2.*dt+dy));
mur1z=((v2.*dt-dz)./(v2.*dt+dz));
%----%
mur2x=(2*dx)./(v2.*dt+dx);
mur2y=(2*dy)./(v2.*dt+dy);
mur2z=(2*dz)./(v2.*dt+dz);
%----%
mur3x=(dx.*(v2.*dt).^2)./(2*dy^2.*(v2.*dt+dx));
mur3y=(dy.*(v2.*dt).^2)./(2*dx^2.*(v2.*dt+dy));
mur3z=(dz.*(v2.*dt).^2)./(2*dx^2.*(v2.*dt+dz));
%----%
mur4x=(dx.*(v2.*dt).^2)./(2*dz^2.*(v2.*dt+dx));
mur4y=(dy.*(v2.*dt).^2)./(2*dz^2.*(v2.*dt+dy));
mur4z=(dz.*(v2.*dt).^2)./(2*dy^2.*(v2.*dt+dz));
%-----%

%---- vetores para varredura ----%
i = 2:xmax-1;
j = 2:ymax-1;
k = 2:zmax-1;
%-----%

%----- inicio do calculo -----%
for n=2:nmax

disp(n);

%---- excitacao ----%
Ez(xexc,yexc,zexc,2) = Ez(xexc,yexc,zexc,2) + Vz(n);
%-----%

%---- campo magnetico ----%
Hx(i,j,k,3)=h1(i,j,k).*Hx(i,j,k,2)...
-h2(i,j,k).*(Ez(i,j,k,2)-Ez(i,j-1,k,2))./dy-(Ey(i,j,k,2)-Ey(i,j,k-
1,2))./dz);
Hy(i,j,k,3)=h1(i,j,k).*Hy(i,j,k,2)...
-h2(i,j,k).*(Ex(i,j,k,2)-Ex(i,j,k-1,2))./dz-(Ez(i,j,k,2)-Ez(i-
1,j,k,2))./dx);
Hz(i,j,k,3)=h1(i,j,k).*Hz(i,j,k,2)...
-h2(i,j,k).*(Ey(i,j,k,2)-Ey(i-1,j,k,2))./dx-(Ex(i,j,k,2)-Ex(i,j-
1,k,2))./dy);
%-----%

%---- campo eletrico ----%
Ex(i,j,k,3)=e1(i,j,k).*Ex(i,j,k,2)...
+e2(i,j,k).*(Hz(i,j+1,k,3)-Hz(i,j,k,3))./dy-(Hy(i,j,k+1,3)-
Hy(i,j,k,3))./dz);
Ey(i,j,k,3)=e1(i,j,k).*Ey(i,j,k,2)...
+e2(i,j,k).*(Hx(i,j,k+1,3)-Hx(i,j,k,3))./dz-(Hz(i+1,j,k,3)-
Hz(i,j,k,3))./dx);
Ez(i,j,k,3)=e1(i,j,k).*Ez(i,j,k,2)...
+e2(i,j,k).*(Hy(i+1,j,k,3)-Hy(i,j,k,3))./dx-(Hx(i,j+1,k,3)-
Hx(i,j,k,3))./dy);
%-----%

%---- condicao de absorcao ----%
% xmin+1
Ey(bx1,j,k,3)=-Ey(bx1+1,j,k,1)...
+mur1x(bx1,j,k).*(Ey(bx1+1,j,k,3)+Ey(bx1,j,k,1))...
+mur2x(bx1,j,k).*(Ey(bx1,j,k,2)+Ey(bx1+1,j,k,2))...
+mur3x(bx1,j,k).*(Ey(bx1,j+1,k,2)-2.*Ey(bx1,j,k,2)+Ey(bx1,j-1,k,2))+...
Ey(bx1+1,j+1,k,2)-2.*Ey(bx1+1,j,k,2)+Ey(bx1+1,j-1,k,2))...
+mur4x(bx1,j,k).*(Ey(bx1,j,k+1,2)-2.*Ey(bx1,j,k,2)+Ey(bx1,j,k-1,2))+...
Ey(bx1+1,j,k+1,2)-2.*Ey(bx1+1,j,k,2)+Ey(bx1+1,j,k-1,2));

Ez(bx1,j,k,3)=-Ez(bx1+1,j,k,1)...
+mur1x(bx1,j,k).*(Ez(bx1+1,j,k,3)+Ez(bx1,j,k,1))...
+mur2x(bx1,j,k).*(Ez(bx1,j,k,2)+Ez(bx1+1,j,k,2))...
+mur3x(bx1,j,k).*(Ez(bx1,j+1,k,2)-2.*Ez(bx1,j,k,2)+Ez(bx1,j-1,k,2))+...
Ez(bx1+1,j+1,k,2)-2.*Ez(bx1+1,j,k,2)+Ez(bx1+1,j-1,k,2))...

```

```

+mur4x(bx1, j, k). * (Ez(bx1, j, k+1, 2) - 2. * Ez(bx1, j, k, 2) + Ez(bx1, j, k-1, 2) + ...
Ez(bx1+1, j, k+1, 2) - 2. * Ez(bx1+1, j, k, 2) + Ez(bx1+1, j, k-1, 2));

% xmax-1
Ey(bx2, j, k, 3) = -Ey(bx2-1, j, k, 1) ...
+mur1x(bx2, j, k). * (Ey(bx2-1, j, k, 3) + Ey(bx2, j, k, 1)) ...
+mur2x(bx2, j, k). * (Ey(bx2, j, k, 2) + Ey(bx2-1, j, k, 2)) ...
+mur3x(bx2, j, k). * (Ey(bx2, j+1, k, 2) - 2. * Ey(bx2, j, k, 2) + Ey(bx2, j-1, k, 2) + ...
Ey(bx2-1, j+1, k, 2) - 2. * Ey(bx2-1, j, k, 2) + Ey(bx2-1, j-1, k, 2)) ...
+mur4x(bx2, j, k). * (Ey(bx2, j, k+1, 2) - 2. * Ey(bx2, j, k, 2) + Ey(bx2, j, k-1, 2) + ...
Ey(bx2-1, j, k+1, 2) - 2. * Ey(bx2-1, j, k, 2) + Ey(bx2-1, j, k-1, 2));

Ez(bx2, j, k, 3) = -Ez(bx2-1, j, k, 1) ...
+mur1x(bx2, j, k). * (Ez(bx2-1, j, k, 3) + Ez(bx2, j, k, 1)) ...
+mur2x(bx2, j, k). * (Ez(bx2, j, k, 2) + Ez(bx2-1, j, k, 2)) ...
+mur3x(bx2, j, k). * (Ez(bx2, j+1, k, 2) - 2. * Ez(bx2, j, k, 2) + Ez(bx2, j-1, k, 2) + ...
Ez(bx2-1, j+1, k, 2) - 2. * Ez(bx2-1, j, k, 2) + Ez(bx2-1, j-1, k, 2)) ...
+mur4x(bx2, j, k). * (Ez(bx2, j, k+1, 2) - 2. * Ez(bx2, j, k, 2) + Ez(bx2, j, k-1, 2) + ...
Ez(bx2-1, j, k+1, 2) - 2. * Ez(bx2-1, j, k, 2) + Ez(bx2-1, j, k-1, 2));

% ymin+1
Ex(i, by1, k, 3) = -Ex(i, by1+1, k, 1) ...
+mur1y(i, by1, k). * (Ex(i, by1+1, k, 3) + Ex(i, by1, k, 1)) ...
+mur2y(i, by1, k). * (Ex(i, by1, k, 2) + Ex(i, by1+1, k, 2)) ...
+mur3y(i, by1, k). * (Ex(i+1, by1, k, 2) - 2. * Ex(i, by1, k, 2) + Ex(i-1, by1, k, 2) + ...
Ex(i+1, by1+1, k, 2) - 2. * Ex(i, by1+1, k, 2) + Ex(i-1, by1+1, k, 2)) ...
+mur4y(i, by1, k). * (Ex(i, by1, k+1, 2) - 2. * Ex(i, by1, k, 2) + Ex(i, by1, k-1, 2) + ...
Ex(i, by1+1, k+1, 2) - 2. * Ex(i, by1+1, k, 2) + Ex(i, by1+1, k-1, 2));

Ez(i, by1, k, 3) = -Ez(i, by1+1, k, 1) ...
+mur1y(i, by1, k). * (Ez(i, by1+1, k, 3) + Ez(i, by1, k, 1)) ...
+mur2y(i, by1, k). * (Ez(i, by1, k, 2) + Ez(i, by1+1, k, 2)) ...
+mur3y(i, by1, k). * (Ez(i+1, by1, k, 2) - 2. * Ez(i, by1, k, 2) + Ez(i-1, by1, k, 2) + ...
Ez(i+1, by1+1, k, 2) - 2. * Ez(i, by1+1, k, 2) + Ez(i-1, by1+1, k, 2)) ...
+mur4y(i, by1, k). * (Ez(i, by1, k+1, 2) - 2. * Ez(i, by1, k, 2) + Ez(i, by1, k-1, 2) + ...
Ez(i, by1+1, k+1, 2) - 2. * Ez(i, by1+1, k, 2) + Ez(i, by1+1, k-1, 2));

% ymax-1
Ex(i, by2, k, 3) = -Ex(i, by2-1, k, 1) ...
+mur1y(i, by2, k). * (Ex(i, by2-1, k, 3) + Ex(i, by2, k, 1)) ...
+mur2y(i, by2, k). * (Ex(i, by2, k, 2) + Ex(i, by2-1, k, 2)) ...
+mur3y(i, by2, k). * (Ex(i+1, by2, k, 2) - 2. * Ex(i, by2, k, 2) + Ex(i-1, by2, k, 2) + ...
Ex(i+1, by2-1, k, 2) - 2. * Ex(i, by2-1, k, 2) + Ex(i-1, by2-1, k, 2)) ...
+mur4y(i, by2, k). * (Ex(i, by2, k+1, 2) - 2. * Ex(i, by2, k, 2) + Ex(i, by2, k-1, 2) + ...
Ex(i, by2-1, k+1, 2) - 2. * Ex(i, by2-1, k, 2) + Ex(i, by2-1, k-1, 2));

Ez(i, by2, k, 3) = -Ez(i, by2-1, k, 1) ...
+mur1y(i, by2, k). * (Ez(i, by2-1, k, 3) + Ez(i, by2, k, 1)) ...
+mur2y(i, by2, k). * (Ez(i, by2, k, 2) + Ez(i, by2-1, k, 2)) ...
+mur3y(i, by2, k). * (Ez(i+1, by2, k, 2) - 2. * Ez(i, by2, k, 2) + Ez(i-1, by2, k, 2) + ...
Ez(i+1, by2-1, k, 2) - 2. * Ez(i, by2-1, k, 2) + Ez(i-1, by2-1, k, 2)) ...
+mur4y(i, by2, k). * (Ez(i, by2, k+1, 2) - 2. * Ez(i, by2, k, 2) + Ez(i, by2, k-1, 2) + ...
Ez(i, by2-1, k+1, 2) - 2. * Ez(i, by2-1, k, 2) + Ez(i, by2-1, k-1, 2));

% zmin+1
Ex(i, j, bz1, 3) = -Ex(i, j, bz1+1, 1) ...
+mur1z(i, j, bz1). * (Ex(i, j, bz1+1, 3) + Ex(i, j, bz1, 1)) ...
+mur2z(i, j, bz1). * (Ex(i, j, bz1, 2) + Ex(i, j, bz1+1, 2)) ...
+mur3z(i, j, bz1). * (Ex(i+1, j, bz1, 2) - 2. * Ex(i, j, bz1, 2) + Ex(i-1, j, bz1, 2) + ...
Ex(i+1, j, bz1+1, 2) - 2. * Ex(i, j, bz1+1, 2) + Ex(i-1, j, bz1+1, 2)) ...
+mur4z(i, j, bz1). * (Ex(i, j+1, bz1, 2) - 2. * Ex(i, j, bz1, 2) + Ex(i, j-1, bz1, 2) + ...
Ex(i, j+1, bz1+1, 2) - 2. * Ex(i, j, bz1+1, 2) + Ex(i, j-1, bz1+1, 2));

Ey(i, j, bz1, 3) = -Ey(i, j, bz1+1, 1) ...
+mur1z(i, j, bz1). * (Ey(i, j, bz1+1, 3) + Ey(i, j, bz1, 1)) ...
+mur2z(i, j, bz1). * (Ey(i, j, bz1, 2) + Ey(i, j, bz1+1, 2)) ...
+mur3z(i, j, bz1). * (Ey(i+1, j, bz1, 2) - 2. * Ey(i, j, bz1, 2) + Ey(i-1, j, bz1, 2) + ...
Ey(i+1, j, bz1+1, 2) - 2. * Ey(i, j, bz1+1, 2) + Ey(i-1, j, bz1+1, 2)) ...
+mur4z(i, j, bz1). * (Ey(i, j+1, bz1, 2) - 2. * Ey(i, j, bz1, 2) + Ey(i, j-1, bz1, 2) + ...
Ey(i, j+1, bz1+1, 2) - 2. * Ey(i, j, bz1+1, 2) + Ey(i, j-1, bz1+1, 2));

```

```

% zmax-1
Ex(i,j,bz2,3)=-Ex(i,j,bz2-1,1)...
+mur1z(i,j,bz2).*(Ex(i,j,bz2-1,3)+Ex(i,j,bz2,1))...
+mur2z(i,j,bz2).*(Ex(i,j,bz2,2)+Ex(i,j,bz2-1,2))...
+mur3z(i,j,bz2).*(Ex(i+1,j,bz2,2)-2.*Ex(i,j,bz2,2)+Ex(i-1,j,bz2,2))+...
Ex(i+1,j,bz2-1,2)-2.*Ex(i,j,bz2-1,2)+Ex(i-1,j,bz2-1,2))...
+mur4z(i,j,bz2).*(Ex(i,j+1,bz2,2)-2.*Ex(i,j,bz2,2)+Ex(i,j-1,bz2,2))+...
Ex(i,j+1,bz2-1,2)-2.*Ex(i,j,bz2-1,2)+Ex(i,j-1,bz2-1,2));

Ey(i,j,bz2,3)=-Ey(i,j,bz2-1,1)...
+mur1z(i,j,bz2).*(Ey(i,j,bz2-1,3)+Ey(i,j,bz2,1))...
+mur2z(i,j,bz2).*(Ey(i,j,bz2,2)+Ey(i,j,bz2-1,2))...
+mur3z(i,j,bz2).*(Ey(i+1,j,bz2,2)-2.*Ey(i,j,bz2,2)+Ey(i-1,j,bz2,2))+...
Ey(i+1,j,bz2-1,2)-2.*Ey(i,j,bz2-1,2)+Ey(i-1,j,bz2-1,2))...
+mur4z(i,j,bz2).*(Ey(i,j+1,bz2,2)-2.*Ey(i,j,bz2,2)+Ey(i,j-1,bz2,2))+...
Ey(i,j+1,bz2-1,2)-2.*Ey(i,j,bz2-1,2)+Ey(i,j-1,bz2-1,2));

%-----%

%---- lMur ----%
% (zmin+1 || ymin+1) => x=min+1:max
Ex(i,by1,bz1,3)=Ex(i,by1,bz1+1,2)+mur1z(i,by1,bz1).*(Ex(i,by1,bz1+1,3)-
Ex(i,by1,bz1,2));
% (zmin+1 || ymax-1) => x=min+1:max
Ex(i,by2,bz1,3)=Ex(i,by2,bz1+1,2)+mur1z(i,by2,bz1).*(Ex(i,by2,bz1+1,3)-
Ex(i,by2,bz1,2));
% (zmin+1 || xmin+1) => y=min+1:max
Ey(bx1,j,bz1,3)=Ey(bx1,j,bz1+1,2)+mur1z(bx1,j,bz1).*(Ey(bx1,j,bz1+1,3)-
Ey(bx1,j,bz1,2));
% (zmin+1 || xmax-1) => y=min+1:max
Ey(bx2,j,bz1,3)=Ey(bx2,j,bz1+1,2)+mur1z(bx2,j,bz1).*(Ey(bx2,j,bz1+1,3)-
Ey(bx2,j,bz1,2));

% (zmax-1 || ymin+1) => x=min+1:max
Ex(i,by1,bz2,3)=Ex(i,by1,bz2-1,2)+mur1z(i,by1,bz2).*(Ex(i,by1,bz2-1,3)-
Ex(i,by1,bz2,2));
% (zmax-1 || ymax-1) => x=min+1:max
Ex(i,by2,bz2,3)=Ex(i,by2,bz2-1,2)+mur1z(i,by2,bz2).*(Ex(i,by2,bz2-1,3)-
Ex(i,by2,bz2,2));
% (zmax-1 || xmin+1) => y=min+1:max
Ey(bx1,j,bz2,3)=Ey(bx1,j,bz2-1,2)+mur1z(bx1,j,bz2).*(Ey(bx1,j,bz2-1,3)-
Ey(bx1,j,bz2,2));
% (zmax-1 || xmax-1) => y=min+1:max
Ey(bx2,j,bz2,3)=Ey(bx2,j,bz2-1,2)+mur1z(bx2,j,bz2).*(Ey(bx2,j,bz2-1,3)-
Ey(bx2,j,bz2,2));

% (xmin+1 || ymin+1) => z=min+1:max
Ez(bx1,by1,k,3)=Ez(bx1+1,by1,k,2)+mur1x(bx1,by1,k).*(Ez(bx1+1,by1,k,3)-
Ez(bx1,by1,k,2));
% (xmin+1 || ymax-1) => z=min+1:max
Ez(bx1,by2,k,3)=Ez(bx1+1,by2,k,2)+mur1x(bx1,by2,k).*(Ez(bx1+1,by2,k,3)-
Ez(bx1,by2,k,2));
% (xmax-1 || ymin+1) => z=min+1:max
Ez(bx2,by1,k,3)=Ez(bx2-1,by1,k,2)+mur1x(bx2,by1,k).*(Ez(bx2-1,by1,k,3)-
Ez(bx2,by1,k,2));
% (xmax-1 || ymax-1) => z=min+1:max
Ez(bx2,by2,k,3)=Ez(bx2-1,by2,k,2)+mur1x(bx2,by2,k).*(Ez(bx2-1,by2,k,3)-
Ez(bx2,by2,k,2));
%-----%
%-----%

%---- salva resultados do campo eletrico ----%
Mxy(:, :, 1, n)=sqrt(Ex(:, :, zexc, 3).^2+Ey(:, :, zexc, 3).^2+Ez(:, :, zexc, 3).^2);
Mxz(:, :, 1, n)=sqrt(Ex(:, yexc, :, 3).^2+Ey(:, yexc, :, 3).^2+Ez(:, yexc, :, 3).^2);
Myz(:, :, 1, n)=sqrt(Ex(xexc, :, :, 3).^2+Ey(xexc, :, :, 3).^2+Ez(xexc, :, :, 3).^2);
%-----%

%---- restauracao das matrizes ----%
Ex(:, :, 1)=Ex(:, :, 2); Ey(:, :, 1)=Ey(:, :, 2); Ez(:, :, 1)=Ez(:, :, 2);
Ex(:, :, 2)=Ex(:, :, 3); Ey(:, :, 2)=Ey(:, :, 3); Ez(:, :, 2)=Ez(:, :, 3);
Hx(:, :, 1)=Hx(:, :, 2); Hy(:, :, 1)=Hy(:, :, 2); Hz(:, :, 1)=Hz(:, :, 2);
Hx(:, :, 2)=Hx(:, :, 3); Hy(:, :, 2)=Hy(:, :, 3); Hz(:, :, 2)=Hz(:, :, 3);
%-----%

```

```

end
%-----%

%---- apresenta resultados na tela ----%
Vol(1:xmax,1:ymax,1:zmax,1)=0; % aloca um volume para exibição

figure(1);
h=uicontrol('Style','slider','Min',0,'Max',nmax-1,'SliderStep',[1/nmax 10/nmax],...
'Callback',['vi=get(gca,'View');'...
'n=round(get(h,'Value'))+1;'...
'Vol(:,zexc)=Mxy(:,1,n);'...
'Vol(:,yexc,:)=Mxz(:,1,,:n);'...
'Vol(xexc,,:)=Myz(1,,:n);'...
'slice(Vol,yexc,xexc,zexc);'...
xlabel('Y'); ylabel('X'); zlabel('Z');'...
'shading interp; axis ij; axis equal;'...
'set(gca,'View',vi);'...
'title(n);'...
]);

mf = uimenu('Label','ferramentas');
uimenu(mf,'Label','zoom','Callback','zoom on; rotate3d off');
uimenu(mf,'Label','rotacao','Callback','zoom off; rotate3d on;');

set(gcf,'toolbar','figure');
%-----%

```

## (10) Campo3D\_PML.m

```

%---- propriedades do vacuo ----%
eo = 8.854e-12;
uo = 4*pi*1e-7;
%-----%

%---- tamanho e discretização do dominio ----%
xmax=50;
ymax=50;
zmax=50;
nmax=100;

dx=1e-3;      % discretizacao espacial eixo x
dy=1e-3;      % discretizacao espacial eixo y
dz=1e-3;      % discretizacao espacial eixo z
%-----%

%--- descretizacao do tempo ---%
v=1./sqrt(uo*eo);
dt=(1/sqrt((1/dx)^2+(1/dy)^2+(1/dz)^2))/v;
%-----%

%---- matrizes dos campos ----%
Ex(1:xmax,1:ymax,1:zmax,1:2)=0;
Ey(1:xmax,1:ymax,1:zmax,1:2)=0;
Ez(1:xmax,1:ymax,1:zmax,1:2)=0;
Hx(1:xmax,1:ymax,1:zmax,1:2)=0;
Hy(1:xmax,1:ymax,1:zmax,1:2)=0;
Hz(1:xmax,1:ymax,1:zmax,1:2)=0;

Dx(1:xmax,1:ymax,1:zmax,1:2)=0;
Dy(1:xmax,1:ymax,1:zmax,1:2)=0;
Dz(1:xmax,1:ymax,1:zmax,1:2)=0;
Bx(1:xmax,1:ymax,1:zmax,1:2)=0;
By(1:xmax,1:ymax,1:zmax,1:2)=0;
Bz(1:xmax,1:ymax,1:zmax,1:2)=0;

Mxy(1:xmax,1:ymax,1,1:nmax)=0; % matriz para salvar os resultados (plano xy)
Mxz(1:xmax,1,1:zmax,1:nmax)=0; % matriz para salvar os resultados (plano xz)
Myz(1,1:ymax,1:zmax,1:nmax)=0; % matriz para salvar os resultados (plano yz)
%-----%

%-----PML-----%
er = max(max(max(e)))/eo;          % obtem o er do meio
L = 10;                            % espessura da camada PML
m = 4;                              % expoente polinomial
smax = (m+1)/(150*pi*dy*sqrt(er));  % obtem a maxima condutividade do PML

txa = L;  txb = L;                  % largura da PML no eixo x
tya = L;  tyb = L;                  % largura da PML no eixo y
tza = L;  tzb = L;                  % largura da PML no eixo z

bx1 = 2+txa;  bx2 = (xmax-1)-txb;   % posicao inicial e final da PML no eixo x
by1 = 2+tya;  by2 = (ymax-1)-tyb;   % posicao inicial e final da PML no eixo y
bz1 = 2+tza;  bz2 = (zmax-1)-tzb;   % posicao inicial e final da PML no eixo z

ip1 = 2:bx1; ip2 = bx2:xmax-1;      % varredura da PML no eixo x
jp1 = 2:by1; jp2 = by2:ymax-1;      % varredura da PML no eixo y
kp1 = 2:bz1; kp2 = bz2:zmax-1;      % varredura da PML no eixo z

sxpi = zeros(xmax,1,1); sxpm = zeros(xmax,1,1); % vetores de condutividade
syypi = zeros(1,ymax,1); syypm = zeros(1,ymax,1); % vetores de condutividade
szypi = zeros(1,1,zmax); szypm = zeros(1,1,zmax); % vetores de condutividade

sxpi(ip1) = smax*((bx1-ip1)./txa).^m; % condutividade no eixo x (posicoes "i")
sxpi(ip2) = smax*((ip2+0.5-bx2)./txb).^m; % condutividade no eixo x (posicoes "i")

sxpm(ip1) = smax*((bx1-ip1+0.5)./txb).^m; % condutividade no eixo x (posicoes "i-1/2")
sxpm(ip2) = smax*((ip2-bx2)./txb).^m; % condutividade no eixo x (posicoes "i-1/2")

syypi(jp1) = smax*((by1-jp1)./tya).^m; % condutividade no eixo y (posicoes "j")

```

```

syypi(jp2) = smax*((jp2+0.5-by2)./tyb).^m; % condutividade no eixo y (posicoes "j")

syym(jp1) = smax*((by1-jp1+0.5)./tyb).^m; % condutividade no eixo y (posicoes "j-1/2")
syym(jp2) = smax*((jp2-by2)./tyb).^m; % condutividade no eixo y (posicoes "j-1/2")

szypi(kp1) = smax*((bz1-kp1)./tza).^m; % condutividade no eixo z (posicoes "k")
szypi(kp2) = smax*((kp2+0.5-bz2)./tzb).^m; % condutividade no eixo z (posicoes "k")

szypm(kp1) = smax*((bz1-kp1+0.5)./tzb).^m; % condutividade no eixo z (posicoes "k-1/2")
szypm(kp2) = smax*((kp2-bz2)./tzb).^m; % condutividade no eixo z (posicoes "k-1/2")

sxi=s+repmat(sxpi,[1 ymax zmax]); % PML no eixo x + condutividade original do meio
sxm=s+repmat(sxpm,[1 ymax zmax]); % PML no eixo x + condutividade original do meio

syi=s+repmat(sypi,[xmax 1 zmax]); % PML no eixo y + condutividade original do meio
sym=s+repmat(syym,[xmax 1 zmax]); % PML no eixo y + condutividade original do meio

szi=s+repmat(szpi,[xmax ymax 1]); % PML no eixo z + condutividade original do meio
szm=s+repmat(szpm,[xmax ymax 1]); % PML no eixo z + condutividade original do meio

%-- coef. do PML --%
PML1xi = (1+sxi.*dt./2./eo);
PML2xi = (1-sxi.*dt./2./eo);
PML1xm = (1+sxm.*dt./2./eo);
PML2xm = (1-sxm.*dt./2./eo);

PML1yi = (1+syi.*dt./2./eo);
PML2yi = (1-syi.*dt./2./eo);
PML1ym = (1+sym.*dt./2./eo);
PML2ym = (1-sym.*dt./2./eo);

PML1zi = (1+szi.*dt./2./eo);
PML2zi = (1-szi.*dt./2./eo);
PML1zm = (1+szm.*dt./2./eo);
PML2zm = (1-szm.*dt./2./eo);
%-----%
%-----%

%---- coef. dos campos ----%
e1 = (1-s.*dt./(2.*e))./(1+s.*dt./(2.*e));
e2 = (dt./e)./(1+s.*dt./(2.*e));
h1 = (1-pm.*dt./(2.*u))./(1+pm.*dt./(2.*u));
h2 = (dt./u)./(1+pm.*dt./(2.*u));
%-----%

%---- excitação ----%
Vz(2:62)=sin(2*pi*(0:60)/60).^3;
Vz(63:nmax)=0;

xexc = round(xmax/2);
yexc = round(ymax/2);
zexc = round(zmax/2);
%-----%

%---- vetores para varredura ----%
i = 2:xmax-1;
j = 2:ymax-1;
k = 2:zmax-1;
%-----%

%----- inicio do calculo -----%
for n=2:nmax

    disp(n); % exibe o passo de simulacao

    p1 = mod(n+1,2)+1; % variavel auxiliar
    p2 = mod(n,2)+1; % variavel auxiliar

%---- excitacao ----%
Ez(xexc,yexc,zexc,p1) = Ez(xexc,yexc,zexc,p1)+Vz(n);
%-----%

```



```

% algoritmo FDTD + PML
%-----%
Bx(i,j,k,p2)=(PML2ym(i,j,k)/PML1ym(i,j,k)).*Bx(i,j,k,p1)...
              -(dt./PML1ym(i,j,k)).*( (Ez(i,j,k,p1)-Ez(i,j-1,k,p1))./dy...
              -(Ey(i,j,k,p1)-Ey(i,j,k-1,p1))./dz);

Hx(i,j,k,p2)=(PML2zm(i,j,k)/PML1zm(i,j,k)).*Hx(i,j,k,p1)...
              +(1./u(i,j,k)/PML1zm(i,j,k)).*(PML1xi(i,j,k).*Bx(i,j,k,p2)...
              -PML2xi(i,j,k).*Bx(i,j,k,p1));
%-----%
By(i,j,k,p2)=(PML2xm(i,j,k)/PML1xm(i,j,k)).*By(i,j,k,p1)...
              -(dt./PML1xm(i,j,k)).*( (Ex(i,j,k,p1)-Ex(i,j,k-1,p1))./dz...
              -(Ez(i,j,k,p1)-Ez(i-1,j,k,p1))./dx);

Hy(i,j,k,p2)=(PML2zm(i,j,k)/PML1zm(i,j,k)).*Hy(i,j,k,p1)...
              +(1./u(i,j,k)/PML1zm(i,j,k)).*(PML1yi(i,j,k).*By(i,j,k,p2)...
              -PML2yi(i,j,k).*By(i,j,k,p1));
%-----%
Bz(i,j,k,p2)=(PML2ym(i,j,k)/PML1ym(i,j,k)).*Bz(i,j,k,p1)...
              -(dt./PML1ym(i,j,k)).*( (Ey(i,j,k,p1)-Ey(i-1,j,k,p1))./dx...
              -(Ex(i,j,k,p1)-Ex(i,j-1,k,p1))./dy);

Hz(i,j,k,p2)=(PML2xm(i,j,k)/PML1xm(i,j,k)).*Hz(i,j,k,p1)...
              +(1./u(i,j,k)/PML1xm(i,j,k)).*(PML1zi(i,j,k).*Bz(i,j,k,p2)...
              -PML2zi(i,j,k).*Bz(i,j,k,p1));
%-----%
Dx(i,j,k,p2)=(PML2yi(i,j,k)/PML1yi(i,j,k)).*Dx(i,j,k,p1)...
              +(dt./PML1yi(i,j,k)).*( (Hz(i,j+1,k,p2)-Hz(i,j,k,p2))./dy...
              -(Hy(i,j,k+1,p2)-Hy(i,j,k,p2))./dz);

Ex(i,j,k,p2)=(PML2zi(i,j,k)/PML1zi(i,j,k)).*Ex(i,j,k,p1)...
              +(1./e(i,j,k)/PML1zi(i,j,k)).*(PML1xm(i,j,k).*Dx(i,j,k,p2)...
              -PML2xm(i,j,k).*Dx(i,j,k,p1));
%-----%
Dy(i,j,k,p2)=(PML2xi(i,j,k)/PML1xi(i,j,k)).*Dy(i,j,k,p1)...
              +(dt./PML1xi(i,j,k)).*( (Hx(i,j,k+1,p2)-Hx(i,j,k,p2))./dz...
              -(Hz(i+1,j,k,p2)-Hz(i,j,k,p2))./dx);

Ey(i,j,k,p2)=(PML2zi(i,j,k)/PML1zi(i,j,k)).*Ey(i,j,k,p1)...
              +(1./e(i,j,k)/PML1zi(i,j,k)).*(PML1ym(i,j,k).*Dy(i,j,k,p2)...
              -PML2ym(i,j,k).*Dy(i,j,k,p1));
%-----%
Dz(i,j,k,p2)=(PML2yi(i,j,k)/PML1yi(i,j,k)).*Dz(i,j,k,p1)...
              +(dt./PML1yi(i,j,k)).*( (Hy(i+1,j,k,p2)-Hy(i,j,k,p2))./dx...
              -(Hx(i,j+1,k,p2)-Hx(i,j,k,p2))./dy);

Ez(i,j,k,p2)=(PML2xi(i,j,k)/PML1xi(i,j,k)).*Ez(i,j,k,p1)...
              +(1./e(i,j,k)/PML1xi(i,j,k)).*(PML1zm(i,j,k).*Dz(i,j,k,p2)...
              -PML2zm(i,j,k).*Dz(i,j,k,p1));
%-----%

%---- PEC's ----%
%plano xy
Ex(:, :, 1, p2)=0;           Ey(:, :, 1, p2)=0;
Ex(:, :, zmax-1, p2)=0;     Ey(:, :, zmax-1, p2)=0;
%plano xz
Ex(:, 1, :, p2)=0;          Ez(:, 1, :, p2)=0;
Ex(:, ymax-1, :, p2)=0;     Ez(:, ymax-1, :, p2)=0;
%plano yz
Ey(1, :, :, p2)=0;          Ez(1, :, :, p2)=0;
Ey(xmax-1, :, :, p2)=0;     Ez(xmax-1, :, :, p2)=0;
%-----%

```

```

%---- salva resultados do campo eletrico ----%
Mxy(:, :, 1, n) = sqrt(Ex(:, :, zexc, p2).^2 + Ey(:, :, zexc, p2).^2 + Ez(:, :, zexc, p2).^2);
Mxz(:, 1, :, n) = sqrt(Ex(:, yexc, :, p2).^2 + Ey(:, yexc, :, p2).^2 + Ez(:, yexc, :, p2).^2);
Myz(1, :, :, n) = sqrt(Ex(xexc, :, :, p2).^2 + Ey(xexc, :, :, p2).^2 + Ez(xexc, :, :, p2).^2);
%-----%

end
%-----%

%---- apresenta resultados na tela ----%
Vol(1:xmax, 1:yymax, 1:zmax, 1) = 0; % aloca um volume para exibição

figure(1);
h = uicontrol('Style', 'slider', 'Min', 0, 'Max', nmax-1, 'SliderStep', [1/nmax 10/nmax], ...
    'Callback', ['vi=get(gca, 'View');'...
        'n=round(get(h, 'Value'))+1;'...
        'Vol(:, :, zexc)=Mxy(:, :, 1, n);'...
        'Vol(:, yexc, :) =Mxz(:, 1, :, n);'...
        'Vol(xexc, :, :) =Myz(1, :, :, n);'...
        'slice(Vol, yexc, xexc, zexc);'...
        'xlabel(''Y''); ylabel(''X''); zlabel(''Z'');'...
        'shading interp; axis ij; axis equal;'...
        'set(gca, 'View', vi);'...
        'title(n);'...
    ]);

mf = uimenu('Label', 'ferramentas');
uimenu(mf, 'Label', 'zoom', 'Callback', 'zoom on; rotate3d off');
uimenu(mf, 'Label', 'rotacao', 'Callback', 'zoom off; rotate3d on;');

set(gcf, 'toolbar', 'figure');
%-----%

```

# (11) Campo3D\_Teleport.m

```

%---- propriedades do vacuo ----%
eo = 8.854e-12;
uo = 4*pi*1e-7;
%-----%

%---- tamanho e discretizacao do dominio ----%
xmax = 50;
ymax = 50;
zmax = 50;

nmax = 100;

dx = 1e-3;          % discretizacao espacial no eixo x
dy = 1e-3;          % discretizacao espacial no eixo y
dz = 1e-3;

%-----%

%---- matrizes dos campos ----%
Ex(1:xmax,1:ymax,1:zmax,1:3)=0;
Ey(1:xmax,1:ymax,1:zmax,1:3)=0;
Ez(1:xmax,1:ymax,1:zmax,1:3)=0;
Hx(1:xmax,1:ymax,1:zmax,1:3)=0;
Hy(1:xmax,1:ymax,1:zmax,1:3)=0;
Hz(1:xmax,1:ymax,1:zmax,1:3)=0;

Mxy(1:xmax,1:ymax,1,1:nmax)=0; % matriz para salvar os resultados (plano xy)
Mxz(1:xmax,1,1:zmax,1:nmax)=0; % matriz para salvar os resultados (plano xz)
Myz(1,1:ymax,1:zmax,1:nmax)=0; % matriz para salvar os resultados (plano yz)
%-----%

%---- propriedades do meio ----%
e(1:xmax,1:ymax,1:zmax)=eo;
u(1:xmax,1:ymax,1:zmax)=uo;
s(1:xmax,1:ymax,1:zmax)=0;
pm(1:xmax,1:ymax,1:zmax)=0;
%-----%

%--- descretizacao temporal ---%
v = 1/sqrt(uo*eo);
dt = (1/sqrt((1/dx)^2+(1/dy)^2+(1/dz)^2))/v;
%-----%

%---- excitação ----%
Vz(2:62)=sin(2*pi*(0:60)/60).^3;
Vz(63:nmax)=0;

xexc = round(xmax/2);
yexc = round(ymax/2);
zexc = round(zmax/2);
%-----%

%---- coef. dos campos ----%
e1 = (1-s.*dt./(2.*e))./(1+s.*dt./(2.*e));
e2 = (dt./e)./(1+s.*dt./(2.*e));
h1 = (1-pm.*dt./(2.*u))./(1+pm.*dt./(2.*u));
h2 = (dt./u)./(1+pm.*dt./(2.*u));
%-----%

%---- posicionamento das fronteiras ----%
bx1 = 1+3;          % xmin+3
by1 = 1+3;          % ymin+3
bz1 = 1+3;          % zmax+3

bx2 = xmax-3;      % xmax-3
by2 = ymax-3;      % ymax-3
bz2 = zmax-3;      % zmax-3

R = 0.99; % coef. de amortecimento;
%-----%

```

```

%---- vetores para varredura ----%
i = 2:xmax-1;
j = 2:ymax-1;
k = 2:zmax-1;
%-----%

%----- inicio do calculo -----%
for n=2:nmax

disp(n);

p1=1+mod(n,3);
p2=1+mod(n+1,3);
p3=1+mod(n-1,3);

%---- excitacao ----%
Ez(xexc,yexc,zexc,p2) = Ez(xexc,yexc,zexc,p2)+Vz(n);
%-----%

%---- campo magnetico ----%
Hx(i,j,k,p3)=h1(i,j,k).*Hx(i,j,k,p2)...
-h2(i,j,k).*(Ez(i,j,k,p2)-Ez(i,j-1,k,p2))./dy-(Ey(i,j,k,p2)-Ey(i,j,k-
1,p2))./dz);
Hy(i,j,k,p3)=h1(i,j,k).*Hy(i,j,k,p2)...
-h2(i,j,k).*(Ex(i,j,k,p2)-Ex(i,j,k-1,p2))./dz-(Ez(i,j,k,p2)-Ez(i-
1,j,k,p2))./dx);
Hz(i,j,k,p3)=h1(i,j,k).*Hz(i,j,k,p2)...
-h2(i,j,k).*(Ey(i,j,k,p2)-Ey(i-1,j,k,p2))./dx-(Ex(i,j,k,p2)-Ex(i,j-
1,k,p2))./dy);
%-----%

%---- ABC teleport field ----%
% xmin+3
Hy(bx1+1,j,k,p3)=Hy(bx1+1,j,k,p3)-R*Ez(bx1+1,j,k,p1).*h2(bx1+1,j,k)./dx;
Hz(bx1+1,j,k,p3)=Hz(bx1+1,j,k,p3)+R*Ey(bx1+1,j,k,p1).*h2(bx1+1,j,k)./dx;

% xmax-3
Hy(bx2,j,k,p3)=Hy(bx2,j,k,p3)+R*Ez(bx2-1,j,k,p1).*h2(bx2-1,j,k)./dx;
Hz(bx2,j,k,p3)=Hz(bx2,j,k,p3)-R*Ey(bx2-1,j,k,p1).*h2(bx2-1,j,k)./dx;

% ymin+3
Hx(i,by1+1,k,p3)=Hx(i,by1+1,k,p3)+R*Ez(i,by1+1,k,p1).*h2(i,by1+1,k)./dy;
Hz(i,by1+1,k,p3)=Hz(i,by1+1,k,p3)-R*Ex(i,by1+1,k,p1).*h2(i,by1+1,k)./dy;

% ymax-3
Hx(i,by2,k,p3)=Hx(i,by2,k,p3)-R*Ez(i,by2-1,k,p1).*h2(i,by2-1,k)./dy;
Hz(i,by2,k,p3)=Hz(i,by2,k,p3)+R*Ex(i,by2-1,k,p1).*h2(i,by2-1,k)./dy;

% zmin+3
Hy(i,j,bz1+1,p3)=Hy(i,j,bz1+1,p3)+R*Ex(i,j,bz1+1,p1).*h2(i,j,bz1+1)./dz;
Hx(i,j,bz1+1,p3)=Hx(i,j,bz1+1,p3)-R*Ey(i,j,bz1+1,p1).*h2(i,j,bz1+1)./dz;

% zmax-3
Hy(i,j,bz2,p3)=Hy(i,j,bz2,p3)-R*Ex(i,j,bz2-1,p1).*h2(i,j,bz2-1)./dz;
Hx(i,j,bz2,p3)=Hx(i,j,bz2,p3)+R*Ey(i,j,bz2-1,p1).*h2(i,j,bz2-1)./dz;
%-----%

%---- campo eletrico ----%
Ex(i,j,k,p3)=e1(i,j,k).*Ex(i,j,k,p2)...
+e2(i,j,k).*(Hz(i,j+1,k,p3)-Hz(i,j,k,p3))./dy-(Hy(i,j,k+1,p3)-
Hy(i,j,k,p3))./dz);
Ey(i,j,k,p3)=e1(i,j,k).*Ey(i,j,k,p2)...
+e2(i,j,k).*(Hx(i,j,k+1,p3)-Hx(i,j,k,p3))./dz-(Hz(i+1,j,k,p3)-
Hz(i,j,k,p3))./dx);
Ez(i,j,k,p3)=e1(i,j,k).*Ez(i,j,k,p2)...
+e2(i,j,k).*(Hy(i+1,j,k,p3)-Hy(i,j,k,p3))./dx-(Hx(i,j+1,k,p3)-
Hx(i,j,k,p3))./dy);
%-----%

%---- ABC teleport field ----%
% xmin+3
Ez(bx1 ,j,k,p3)=Ez(bx1 ,j,k,p3)-R*Hy(bx1+2,j,k,p2).*e2(bx1+2,j,k)./dx;
Ey(bx1 ,j,k,p3)=Ey(bx1 ,j,k,p3)+R*Hz(bx1+2,j,k,p2).*e2(bx1+2,j,k)./dx;

```

```

% xmax-3
Ez(bx2,j,k,p3)=Ez(bx2,j,k,p3)+R*Hy(bx2-1,j,k,p2).*e2(bx2-1,j,k)./dx;
Ey(bx2,j,k,p3)=Ey(bx2,j,k,p3)-R*Hz(bx2-1,j,k,p2).*e2(bx2-1,j,k)./dx;

% ymin+3
Ez(i,by1,k,p3)=Ez(i,by1,k,p3)+R*Hx(i,by1+2,k,p2).*e2(i,by1+2,k)./dy;
Ex(i,by1,k,p3)=Ex(i,by1,k,p3)-R*Hz(i,by1+2,k,p2).*e2(i,by1+2,k)./dy;

% ymax-3
Ez(i,by2,k,p3)=Ez(i,by2,k,p3)-R*Hx(i,by2-1,k,p2).*e2(i,by2-1,k)./dy;
Ex(i,by2,k,p3)=Ex(i,by2,k,p3)+R*Hz(i,by2-1,k,p2).*e2(i,by2-1,k)./dy;

% zmin+3
Ex(i,j,bz1,p3)=Ex(i,j,bz1,p3)+R*Hy(i,j,bz1+2,p2).*e2(i,j,bz1+2)./dz;
Ey(i,j,bz1,p3)=Ey(i,j,bz1,p3)-R*Hx(i,j,bz1+2,p2).*e2(i,j,bz1+2)./dz;

% zmax-3
Ex(i,j,bz2,p3)=Ex(i,j,bz2,p3)-R*Hy(i,j,bz2-1,p2).*e2(i,j,bz2-1)./dz;
Ey(i,j,bz2,p3)=Ey(i,j,bz2,p3)+R*Hx(i,j,bz2-1,p2).*e2(i,j,bz2-1)./dz;
%-----%

%---- PEC's ----%
%plano xy
Ex(:, :, 1, p2)=0;      Ey(:, :, 1, p2)=0;
Ex(:, :, zmax-1, p2)=0; Ey(:, :, zmax-1, p2)=0;
%plano xz
Ex(:, 1, :, p2)=0;      Ez(:, 1, :, p2)=0;
Ex(:, ymax-1, :, p2)=0; Ez(:, ymax-1, :, p2)=0;
%plano yz
Ey(1, :, :, p2)=0;      Ez(1, :, :, p2)=0;
Ey(xmax-1, :, :, p2)=0; Ez(xmax-1, :, :, p2)=0;
%-----%

%---- salva resultados do campo eletrico ----%
Mxy(:, :, 1, n)=sqrt(Ex(:, :, zexc, p3).^2+Ey(:, :, zexc, p3).^2+Ez(:, :, zexc, p3).^2);
Mxz(:, 1, :, n)=sqrt(Ex(:, yexc, :, p3).^2+Ey(:, yexc, :, p3).^2+Ez(:, yexc, :, p3).^2);
Myz(1, :, :, n)=sqrt(Ex(xexc, :, :, p3).^2+Ey(xexc, :, :, p3).^2+Ez(xexc, :, :, p3).^2);
%-----%

end
%-----%

%---- apresenta resultados na tela ----%
Vol(1:xmax,1:ymax,1:zmax,1)=0; % aloca um volume para exibição

figure(1);
h=uicontrol('Style','slider','Min',0,'Max',nmax-1,'SliderStep',[1/nmax 10/nmax],...
    'Callback',[ 'vi=get(gca, 'View');'...
                'n=round(get(h, 'Value'))+1;'...
                'Vol(:, :, zexc)=Mxy(:, :, 1, n);'...
                'Vol(:, yexc, :)=Mxz(:, 1, :, n);'...
                'Vol(xexc, :, :)=Myz(1, :, :, n);'...
                'slice(Vol, yexc, xexc, zexc);'...
                'xlabel(''Y''); ylabel(''X''); zlabel(''Z'');'...
                'shading interp; axis ij; axis equal;'...
                'set(gca, 'View', vi);'...
                'title(n);'...
    ]);

mf = uimenu('Label','ferramentas');
uimenu(mf, 'Label', 'zoom', 'Callback', 'zoom on; rotate3d off');
uimenu(mf, 'Label', 'rotacao', 'Callback', 'zoom off; rotate3d on;');

set(gcf, 'toolbar', 'figure');
%-----%

```

## (12) Campo\_Ez\_Teleport\_atraso.m

```

%---- propriedades do vacuo ----%
eo = 8.854e-12;
uo = 4*pi*1e-7;
%-----%

%---- tamanho e discretizacao do dominio ----%
xmax = 50;
ymax = 50;
zmax = 50;

nmax = 100;

dx = 1e-3;          % discretizacao espacial no eixo x
dy = 1e-3;          % discretizacao espacial no eixo y
dz = 1e-3;
%-----%

%---- matrizes dos campos ----%
Ex(1:xmax,1:ymax,1:zmax,1:4)=0;
Ey(1:xmax,1:ymax,1:zmax,1:4)=0;
Ez(1:xmax,1:ymax,1:zmax,1:4)=0;
Hx(1:xmax,1:ymax,1:zmax,1:4)=0;
Hy(1:xmax,1:ymax,1:zmax,1:4)=0;
Hz(1:xmax,1:ymax,1:zmax,1:4)=0;

Mxy(1:xmax,1:ymax,1:1:nmax)=0; % matriz para salvar os resultados (plano xy)
Mxz(1:xmax,1:1:zmax,1:nmax)=0; % matriz para salvar os resultados (plano xz)
Myz(1:1:ymax,1:zmax,1:nmax)=0; % matriz para salvar os resultados (plano yz)
%-----%

%---- propriedades do meio ----%
e(1:xmax,1:ymax,1:zmax)=eo;
u(1:xmax,1:ymax,1:zmax)=uo;
s(1:xmax,1:ymax,1:zmax)=0;
pm(1:xmax,1:ymax,1:zmax)=0;
%-----%

%--- descretizacao temporal ---%
v = 1/sqrt(uo*eo);
dt = (1/sqrt((1/dx)^2+(1/dy)^2+(1/dz)^2))/v;
%-----%

%---- excitação ----%
Vz(2:62)=sin(2*pi*(0:60)/60).^3;
Vz(63:nmax)=0;

xexc = round(xmax/2);
yexc = round(ymax/2);
zexc = round(zmax/2);
%-----%

%---- coef. dos campos ----%
e1 = (1-s.*dt./(2.*e))./(1+s.*dt./(2.*e));
e2 = (dt./e)./(1+s.*dt./(2.*e));
h1 = (1-pm.*dt./(2.*u))./(1+pm.*dt./(2.*u));
h2 = (dt./u)./(1+pm.*dt./(2.*u));
%-----%

%---- posicionamento das fronteiras ----%
bx1 = 1+3;          % xmin+3
by1 = 1+3;          % ymin+3
bz1 = 1+3;          % zmax+3
bx2 = xmax-3;      % xmax-3
by2 = ymax-3;      % ymax-3
bz2 = zmax-3;      % zmax-3

R = 0.99;          % coef. de amortecimento;
a = sqrt(3);       % atraso temporal (muda esse valor para se adequar ao problema)
m = floor(a);
%-----%

```

```

%---- vetores para varredura ----%
i = 2:xmax-1;
j = 2:ymin-1;
k = 2:zmax-1;
%-----%

%----- inicio do calculo -----%
for n=2:nmax

disp(n);

p1 = 1+mod(n-1,4);
p2 = 1+mod(n,4);
p3 = 1+mod(n+1,4);
p4 = 1+mod(n+2,4);

%---- excitacao ----%
Ez(xexc,yexc,zexc,p3) = Ez(xexc,yexc,zexc,p3)+Vz(n);
%-----%

%---- campo magnetico ----%
Hx(i,j,k,p4)=h1(i,j,k).*Hx(i,j,k,p3)...
-h2(i,j,k).*(Ez(i,j,k,p3)-Ez(i,j-1,k,p3))./dy-(Ey(i,j,k,p3)-Ey(i,j,k-
1,p3))./dz);
Hy(i,j,k,p4)=h1(i,j,k).*Hy(i,j,k,p3)...
-h2(i,j,k).*(Ex(i,j,k,p3)-Ex(i,j,k-1,p3))./dz-(Ez(i,j,k,p3)-Ez(i-
1,j,k,p3))./dx);
Hz(i,j,k,p4)=h1(i,j,k).*Hz(i,j,k,p3)...
-h2(i,j,k).*(Ey(i,j,k,p3)-Ey(i-1,j,k,p3))./dx-(Ex(i,j,k,p3)-Ex(i,j-
1,k,p3))./dy);
%-----%

%---- ABC teleport field ----%
% xmin+3
Hy(bx1+1,j,k,p4)=Hy(bx1+1,j,k,p4)-R*(h2(bx1+1,j,k)./dx).*((m+1-
a)*Ez(bx1+1,j,k,p2)+(a-m)*Ez(bx1+1,j,k,p1));
Hz(bx1+1,j,k,p4)=Hz(bx1+1,j,k,p4)+R*(h2(bx1+1,j,k)./dx).*((m+1-
a)*Ey(bx1+1,j,k,p2)+(a-m)*Ey(bx1+1,j,k,p1));

% xmax-3
Hy(bx2,j,k,p4)=Hy(bx2,j,k,p4)+R*(h2(bx2-1,j,k)./dx).*((m+1-a)*Ez(bx2-1,j,k,p2)+(a-
m)*Ez(bx2-1,j,k,p1));
Hz(bx2,j,k,p4)=Hz(bx2,j,k,p4)-R*(h2(bx2-1,j,k)./dx).*((m+1-a)*Ey(bx2-1,j,k,p2)+(a-
m)*Ey(bx2-1,j,k,p1));

% ymin+3
Hx(i,by1+1,k,p4)=Hx(i,by1+1,k,p4)+R*(h2(i,by1+1,k)./dy).*((m+1-
a)*Ez(i,by1+1,k,p2)+(a-m)*Ez(i,by1+1,k,p1));
Hz(i,by1+1,k,p4)=Hz(i,by1+1,k,p4)-R*(h2(i,by1+1,k)./dy).*((m+1-
a)*Ex(i,by1+1,k,p2)+(a-m)*Ex(i,by1+1,k,p1));

% ymax-3
Hx(i,by2,k,p4)=Hx(i,by2,k,p4)-R*(h2(i,by2-1,k)./dy).*((m+1-a)*Ez(i,by2-1,k,p2)+(a-
m)*Ez(i,by2-1,k,p1));
Hz(i,by2,k,p4)=Hz(i,by2,k,p4)+R*(h2(i,by2-1,k)./dy).*((m+1-a)*Ex(i,by2-1,k,p2)+(a-
m)*Ex(i,by2-1,k,p1));

% zmin+3
Hy(i,j,bz1+1,p4)=Hy(i,j,bz1+1,p4)+R*(h2(i,j,bz1+1)./dz).*((m+1-
a)*Ex(i,j,bz1+1,p2)+(a-m)*Ex(i,j,bz1+1,p1));
Hx(i,j,bz1+1,p4)=Hx(i,j,bz1+1,p4)-R*(h2(i,j,bz1+1)./dz).*((m+1-
a)*Ey(i,j,bz1+1,p2)+(a-m)*Ey(i,j,bz1+1,p1));

% zmax-3
Hy(i,j,bz2,p4)=Hy(i,j,bz2,p4)-R*(h2(i,j,bz2-1)./dz).*((m+1-a)*Ex(i,j,bz2-1,p2)+(a-
m)*Ex(i,j,bz2-1,p1));
Hx(i,j,bz2,p4)=Hx(i,j,bz2,p4)+R*(h2(i,j,bz2-1)./dz).*((m+1-a)*Ey(i,j,bz2-1,p2)+(a-
m)*Ey(i,j,bz2-1,p1));
%-----%

```

```

%---- campo eletrico ----%
Ex(i,j,k,p4)=e1(i,j,k).*Ex(i,j,k,p3)...
+e2(i,j,k).*(Hz(i,j+1,k,p4)-Hz(i,j,k,p4))./dy-(Hy(i,j,k+1,p4)-
Hy(i,j,k,p4))./dz);
Ey(i,j,k,p4)=e1(i,j,k).*Ey(i,j,k,p3)...
+e2(i,j,k).*(Hx(i,j,k+1,p4)-Hx(i,j,k,p4))./dz-(Hz(i+1,j,k,p4)-
Hz(i,j,k,p4))./dx);
Ez(i,j,k,p4)=e1(i,j,k).*Ez(i,j,k,p3)...
+e2(i,j,k).*(Hy(i+1,j,k,p4)-Hy(i,j,k,p4))./dx-(Hx(i,j+1,k,p4)-
Hx(i,j,k,p4))./dy);
%-----%

%---- ABC teleport field ----%
% xmin+3
Ez(bx1 ,j,k,p4)=Ez(bx1 ,j,k,p4)-R*(e2(bx1+2,j,k)./dx).*((m+1-
a)*Hy(bx1+2,j,k,p3)+(a-m)*Hy(bx1+2,j,k,p2));
Ey(bx1 ,j,k,p4)=Ey(bx1 ,j,k,p4)+R*(e2(bx1+2,j,k)./dx).*((m+1-
a)*Hz(bx1+2,j,k,p3)+(a-m)*Hz(bx1+2,j,k,p2));

% xmax-3
Ez(bx2,j,k,p4)=Ez(bx2,j,k,p4)+R*(e2(bx2-1,j,k)./dx).*((m+1-a)*Hy(bx2-1,j,k,p3)+(a-
m)*Hy(bx2-1,j,k,p2));
Ey(bx2,j,k,p4)=Ey(bx2,j,k,p4)-R*(e2(bx2-1,j,k)./dx).*((m+1-a)*Hz(bx2-1,j,k,p3)+(a-
m)*Hz(bx2-1,j,k,p2));

% ymin+3
Ez(i,by1 ,k,p4)=Ez(i,by1 ,k,p4)+R*(e2(i,by1+2,k)./dy).*((m+1-
a)*Hx(i,by1+2,k,p3)+(a-m)*Hx(i,by1+2,k,p2));
Ex(i,by1 ,k,p4)=Ex(i,by1 ,k,p4)-R*(e2(i,by1+2,k)./dy).*((m+1-
a)*Hz(i,by1+2,k,p3)+(a-m)*Hz(i,by1+2,k,p2));

% ymax-3
Ez(i,by2,k,p4)=Ez(i,by2,k,p4)-R*(e2(i,by2-1,k)./dy).*((m+1-a)*Hx(i,by2-1,k,p3)+(a-
m)*Hx(i,by2-1,k,p2));
Ex(i,by2,k,p4)=Ex(i,by2,k,p4)+R*(e2(i,by2-1,k)./dy).*((m+1-a)*Hz(i,by2-1,k,p3)+(a-
m)*Hz(i,by2-1,k,p2));

% zmax+3
Ex(i,j,bz1 ,p4)=Ex(i,j,bz1 ,p4)+R*(e2(i,j,bz1+2)./dz).*((m+1-
a)*Hy(i,j,bz1+2,p3)+(a-m)*Hy(i,j,bz1+2,p2));
Ey(i,j,bz1 ,p4)=Ey(i,j,bz1 ,p4)-R*(e2(i,j,bz1+2)./dz).*((m+1-
a)*Hx(i,j,bz1+2,p3)+(a-m)*Hx(i,j,bz1+2,p2));

% zmin-3
Ex(i,j,bz2,p4)=Ex(i,j,bz2,p4)-R*(e2(i,j,bz2-1)./dz).*((m+1-a)*Hy(i,j,bz2-1,p3)+(a-
m)*Hy(i,j,bz2-1,p2));
Ey(i,j,bz2,p4)=Ey(i,j,bz2,p4)+R*(e2(i,j,bz2-1)./dz).*((m+1-a)*Hx(i,j,bz2-1,p3)+(a-
m)*Hx(i,j,bz2-1,p2));
%-----%

%--- PEC ---%
%plano xy
Ex(:, :, 1, p4)=0; Ex(:, :, zmax-1, p4)=0;
Ey(:, :, 1, p4)=0; Ey(:, :, zmax-1, p4)=0;
%plano xz
Ex(:, 1, :, p4)=0; Ex(:, ymax-1, :, p4)=0;
Ez(:, 1, :, p4)=0; Ez(:, ymax-1, :, p4)=0;
%plano yz
Ey(1, :, :, p4)=0; Ey(xmax-1, :, :, p4)=0;
Ez(1, :, :, p4)=0; Ez(xmax-1, :, :, p4)=0;
%-----%

%---- salva resultados do campo eletrico ----%
Mxy(:, :, 1, n)=sqrt(Ex(:, :, zexc, p4).^2+Ey(:, :, zexc, p4).^2+Ez(:, :, zexc, p4).^2);
Mxz(:, :, 1, n)=sqrt(Ex(:, yexc, :, p4).^2+Ey(:, yexc, :, p4).^2+Ez(:, yexc, :, p4).^2);
Myz(1, :, :, n)=sqrt(Ex(xexc, :, :, p4).^2+Ey(xexc, :, :, p4).^2+Ez(xexc, :, :, p4).^2);
%-----%

end
%-----%

```



```

%---- apresenta resultados na tela ----%
Vol(1:xmax,1:ymax,1:zmax,1)=0; % aloca um volume para exibição

figure(1);
h=uicontrol('Style','slider','Min',0,'Max',nmax-1,'SliderStep',[1/nmax 10/nmax],...
'Callback',['vi=get(gca,'View');'...
'n=round(get(h,'Value'))+1;'...
'Vol(:,zexc)=Mxy(:,1,n);'...
'Vol(:,yexc,:)=Mxz(:,1,,:n);'...
'Vol(xexc,,:)=Myz(1,,:n);'...
'slice(Vol,yexc,xexc,zexc);'...
xlabel('Y'); ylabel('X'); zlabel('Z');'...
'shading interp; axis ij; axis equal;'...
'set(gca,'View',vi);'...
'title(n);'...
]);

mf = uimenu('Label','ferramentas');
uimenu(mf,'Label','zoom','Callback','zoom on; rotate3d off');
uimenu(mf,'Label','rotacao','Callback','zoom off; rotate3d on;');

set(gcf,'toolbar','figure');
%-----%

```

## (13) Campo\_Ez\_Teleport\_equação\_onda.m

```
%---- propriedades do vacuo ----%
eo = 8.854e-12;
uo = 4*pi*1e-7;
%-----%

%---- tamanho e discretizacao do dominio ----%
xmax = 50;
ymax = 50;
zmax = 50;

nmax = 100;

dx = 1e-3;          % discretizacao espacial no eixo x
dy = 1e-3;          % discretizacao espacial no eixo y
dz = 1e-3;
%-----%

%---- matrizes dos campos ----%
Ex(1:xmax,1:ymax,1:zmax,1:3)=0;
Ey(1:xmax,1:ymax,1:zmax,1:3)=0;
Ez(1:xmax,1:ymax,1:zmax,1:3)=0;
Hx(1:xmax,1:ymax,1:zmax,1:3)=0;
Hy(1:xmax,1:ymax,1:zmax,1:3)=0;
Hz(1:xmax,1:ymax,1:zmax,1:3)=0;

%---- buffer para a equacao de onda ----%
Exe(1:xmax,1:4,1:zmax,1:3)=0;
Eze(1:xmax,1:4,1:zmax,1:3)=0;
Hxe(1:xmax,1:4,1:zmax,1:3)=0;
Hze(1:xmax,1:4,1:zmax,1:3)=0;
%----%
Exd(1:xmax,1:4,1:zmax,1:3)=0;
Ezd(1:xmax,1:4,1:zmax,1:3)=0;
Hxd(1:xmax,1:4,1:zmax,1:3)=0;
Hzd(1:xmax,1:4,1:zmax,1:3)=0;
%----%
Eyt(1:4,1:ymax,1:zmax,1:3)=0;
Ezt(1:4,1:ymax,1:zmax,1:3)=0;
Hyt(1:4,1:ymax,1:zmax,1:3)=0;
Hzt(1:4,1:ymax,1:zmax,1:3)=0;
%----%
Eyf(1:4,1:ymax,1:zmax,1:3)=0;
Ezf(1:4,1:ymax,1:zmax,1:3)=0;
Hyf(1:4,1:ymax,1:zmax,1:3)=0;
Hzf(1:4,1:ymax,1:zmax,1:3)=0;
%----%
Exb(1:xmax,1:ymax,1:4,1:3)=0;
Eyb(1:xmax,1:ymax,1:4,1:3)=0;
Hxb(1:xmax,1:ymax,1:4,1:3)=0;
Hyb(1:xmax,1:ymax,1:4,1:3)=0;
%----%
Exc(1:xmax,1:ymax,1:4,1:3)=0;
Eyc(1:xmax,1:ymax,1:4,1:3)=0;
Hxc(1:xmax,1:ymax,1:4,1:3)=0;
Hyc(1:xmax,1:ymax,1:4,1:3)=0;
%-----%

Mxy(1:xmax,1:ymax,1,1:nmax)=0; % matriz para salvar os resultados (plano xy)
Mxz(1:xmax,1,1:zmax,1:nmax)=0; % matriz para salvar os resultados (plano xz)
Myz(1,1:ymax,1:zmax,1:nmax)=0; % matriz para salvar os resultados (plano yz)
%-----%

%---- propriedades do meio ----%
e(1:xmax,1:ymax,1:zmax)=eo;
u(1:xmax,1:ymax,1:zmax)=uo;
s(1:xmax,1:ymax,1:zmax)=0;
pm(1:xmax,1:ymax,1:zmax)=0;
%-----%
```

```

%--- descretizacao temporal ---%
v = 1/sqrt(uo*eo);
dt = (1/sqrt((1/dx)^2+(1/dy)^2+(1/dz)^2))/v;
%-----%

%---- excitação ----%
Vz(2:62)=sin(2*pi*(0:60)/60).^3;
Vz(63:nmax)=0;

xexc = round(xmax/2);
yexc = round(ymax/2);
zexc = round(zmax/2);
%-----%

%---- coef. dos campos ----%
e1 = (1-s.*dt./(2.*e))./(1+s.*dt./(2.*e));
e2 = (dt./e)./(1+s.*dt./(2.*e));
h1 = (1-pm.*dt./(2.*u))./(1+pm.*dt./(2.*u));
h2 = (dt./u)./(1+pm.*dt./(2.*u));
%-----%

%---- posicionamento das fronteiras ----%
bx1 = 1+3;          % xmin+3
by1 = 1+3;          % ymin+3
bz1 = 1+3;          % zmax+3

bx2 = xmax-3;      % xmax-3
by2 = ymax-3;      % ymax-3
bz2 = zmax-3;      % zmax-3

R = 0.99; % coef. de amortecimento;
ang = 45; % angulo para maxima absorcao

%---- compensação por Higdon ----%
vp = (1./sqrt(u.*e))./cos((pi/180)*ang);
murt(1, :, :)=(vp(bx1, :, :)*dt-dx)./(vp(bx1, :, :)*dt+dx);
murf(1, :, :)=(vp(bx2, :, :)*dt-dx)./(vp(bx2, :, :)*dt+dx);
mure(:, 1, :)=(vp(:, by1, :)*dt-dy)./(vp(:, by1, :)*dt+dy);
murd(:, 1, :)=(vp(:, by2, :)*dt-dy)./(vp(:, by2, :)*dt+dy);
murb(:, :, 1)=(vp(:, :, bz1)*dt-dz)./(vp(:, :, bz1)*dt+dz);
murc(:, :, 1)=(vp(:, :, bz2)*dt-dz)./(vp(:, :, bz2)*dt+dz);
%-----%

%---- vetores para varredura ----%
i = 2:xmax-1;
j = 2:ymax-1;
k = 2:zmax-1;
%-----%

%----- inicio do calculo -----%
for n=2:nmax

disp(n);

p1=1+mod(n,3);
p2=1+mod(n+1,3);
p3=1+mod(n-1,3);

%---- excitacao ----%
Ez(xexc,yexc,zexc,p2) = Ez(xexc,yexc,zexc,p2)+Vz(n);
%-----%

%---- campo magnetico ----%
Hx(i,j,k,p3)=h1(i,j,k).*Hx(i,j,k,p2)...
-h2(i,j,k).*((Ez(i,j,k,p2)-Ez(i,j-1,k,p2))./dy-(Ey(i,j,k,p2)-Ey(i,j,k-
1,p2))./dz);
Hy(i,j,k,p3)=h1(i,j,k).*Hy(i,j,k,p2)...
-h2(i,j,k).*((Ex(i,j,k,p2)-Ex(i,j,k-1,p2))./dz-(Ez(i,j,k,p2)-Ez(i-
1,j,k,p2))./dx);
Hz(i,j,k,p3)=h1(i,j,k).*Hz(i,j,k,p2)...
-h2(i,j,k).*((Ey(i,j,k,p2)-Ey(i-1,j,k,p2))./dx-(Ex(i,j,k,p2)-Ex(i,j-
1,k,p2))./dy);
%-----%

```

```

%---- ABC teleport field ----%
% xmin+3
Eyt(1,j,k,p2)=Ey(bx1+1,j,k,p1)+murt(1,j,k).*(Ey(bx1+1,j,k,p2)-Eyt(1,j,k,p1)); %
equacao de onda
Ezt(1,j,k,p2)=Ez(bx1+1,j,k,p1)+murt(1,j,k).*(Ez(bx1+1,j,k,p2)-Ezt(1,j,k,p1)); %
equacao de onda
Hy(bx1+1,j,k,p3)=Hy(bx1+1,j,k,p3)-R*(h2(bx1+1,j,k)./dx).*Ezt(1,j,k,p2);
Hz(bx1+1,j,k,p3)=Hz(bx1+1,j,k,p3)+R*(h2(bx1+1,j,k)./dx).*Eyt(1,j,k,p2);

% xmax-3
Eyf(1,j,k,p2)=Ey(bx2-1,j,k,p1)+murt(1,j,k).*(Ey(bx2-1,j,k,p2)-Eyf(1,j,k,p1)); %
equacao de onda
Ezf(1,j,k,p2)=Ez(bx2-1,j,k,p1)+murt(1,j,k).*(Ez(bx2-1,j,k,p2)-Ezf(1,j,k,p1)); %
equacao de onda
Hy(bx2,j,k,p3)=Hy(bx2,j,k,p3)+R*(h2(bx2-1,j,k)./dx).*Ezf(1,j,k,p2);
Hz(bx2,j,k,p3)=Hz(bx2,j,k,p3)-R*(h2(bx2-1,j,k)./dx).*Eyf(1,j,k,p2);

% ymin+3
Exe(i,1,k,p2)=Ex(i,by1+1,k,p1)+mure(i,1,k).*(Ex(i,by1+1,k,p2)-Exe(i,1,k,p1)); %
equacao de onda
Eze(i,1,k,p2)=Ez(i,by1+1,k,p1)+mure(i,1,k).*(Ez(i,by1+1,k,p2)-Eze(i,1,k,p1)); %
equacao de onda
Hx(i,by1+1,k,p3)=Hx(i,by1+1,k,p3)+R*(h2(i,by1+1,k)./dy).*Eze(i,1,k,p2);
Hz(i,by1+1,k,p3)=Hz(i,by1+1,k,p3)-R*(h2(i,by1+1,k)./dy).*Exe(i,1,k,p2);

% ymax-3
Exd(i,1,k,p2)=Ex(i,by2-1,k,p1)+murd(i,1,k).*(Ex(i,by2-1,k,p2)-Exd(i,1,k,p1)); %
equacao de onda
Ezd(i,1,k,p2)=Ez(i,by2-1,k,p1)+murd(i,1,k).*(Ez(i,by2-1,k,p2)-Ezd(i,1,k,p1)); %
equacao de onda
Hx(i,by2,k,p3)=Hx(i,by2,k,p3)-R*(h2(i,by2-1,k)./dy).*Ezd(i,1,k,p2);
Hz(i,by2,k,p3)=Hz(i,by2,k,p3)+R*(h2(i,by2-1,k)./dy).*Exd(i,1,k,p2);

% zmin+3
Exb(i,j,1,p2)=Ex(i,j,bz1+1,p1)+murb(i,j,1).*(Ex(i,j,bz1+1,p2)-Exb(i,j,1,p1)); %
equacao de onda
Eyb(i,j,1,p2)=Ey(i,j,bz1+1,p1)+murb(i,j,1).*(Ey(i,j,bz1+1,p2)-Eyb(i,j,1,p1)); %
equacao de onda
Hx(i,j,bz1+1,p3)=Hx(i,j,bz1+1,p3)-R*(h2(i,j,bz1+1)./dz).*Eyb(i,j,1,p2);
Hy(i,j,bz1+1,p3)=Hy(i,j,bz1+1,p3)+R*(h2(i,j,bz1+1)./dz).*Exb(i,j,1,p2);

% zmax-3
Exc(i,j,1,p2)=Ex(i,j,bz2-1,p1)+murc(i,j,1).*(Ex(i,j,bz2-1,p2)-Exc(i,j,1,p1)); %
equacao de onda
Eyc(i,j,1,p2)=Ey(i,j,bz2-1,p1)+murc(i,j,1).*(Ey(i,j,bz2-1,p2)-Eyc(i,j,1,p1)); %
equacao de onda
Hx(i,j,bz2,p3)=Hx(i,j,bz2,p3)+R*(h2(i,j,bz2-1)./dz).*Eyc(i,j,1,p2);
Hy(i,j,bz2,p3)=Hy(i,j,bz2,p3)-R*(h2(i,j,bz2-1)./dz).*Exc(i,j,1,p2);
%-----%

%---- campo eletrico ----%
Ex(i,j,k,p3)=e1(i,j,k).*Ex(i,j,k,p2)...
+e2(i,j,k).*((Hz(i,j+1,k,p3)-Hz(i,j,k,p3))./dy-(Hy(i,j,k+1,p3)-
Hy(i,j,k,p3))./dz);
Ey(i,j,k,p3)=e1(i,j,k).*Ey(i,j,k,p2)...
+e2(i,j,k).*((Hx(i,j,k+1,p3)-Hx(i,j,k,p3))./dz-(Hz(i+1,j,k,p3)-
Hz(i,j,k,p3))./dx);
Ez(i,j,k,p3)=e1(i,j,k).*Ez(i,j,k,p2)...
+e2(i,j,k).*((Hy(i+1,j,k,p3)-Hy(i,j,k,p3))./dx-(Hx(i,j+1,k,p3)-
Hx(i,j,k,p3))./dy);
%-----%

%---- ABC teleport field ----%
% xmin+3
Hyt(1,j,k,p3)=Hy(bx1+2,j,k,p2)+murt(1,j,k).*(Hy(bx1+2,j,k,p3)-Hyt(1,j,k,p2)); %
equacao de onda
Hzt(1,j,k,p3)=Hz(bx1+2,j,k,p2)+murt(1,j,k).*(Hz(bx1+2,j,k,p3)-Hzt(1,j,k,p2)); %
equacao de onda
Ey(bx1,j,k,p3)=Ey(bx1,j,k,p3)+R*(e2(bx1+2,j,k)./dx).*Hzt(1,j,k,p3);
Ez(bx1,j,k,p3)=Ez(bx1,j,k,p3)-R*(e2(bx1+2,j,k)./dx).*Hyt(1,j,k,p3);

% xmax-3
Hyf(1,j,k,p3)=Hy(bx2-1,j,k,p2)+murt(1,j,k).*(Hy(bx2-1,j,k,p3)-Hyf(1,j,k,p2)); %
equacao de onda

```

```

    Hzf(1,j,k,p3)=Hz(bx2-1,j,k,p2)+murt(1,j,k).*(Hz(bx2-1,j,k,p3)-Hzf(1,j,k,p2)); %
equacao de onda
    Ey(bx2,j,k,p3)=Ey(bx2,j,k,p3)-R*(e2(bx2-1,j,k)./dx).*Hzf(1,j,k,p3);
    Ez(bx2,j,k,p3)=Ez(bx2,j,k,p3)+R*(e2(bx2-1,j,k)./dx).*Hyf(1,j,k,p3);

    % ymin+3
    Hxe(i,1,k,p3)=Hx(i,by1+2,k,p2)+mure(i,1,k).*(Hx(i,by1+2,k,p3)-Hxe(i,1,k,p2)); %
equacao de onda
    Hze(i,1,k,p3)=Hz(i,by1+2,k,p2)+mure(i,1,k).*(Hz(i,by1+2,k,p3)-Hze(i,1,k,p2)); %
equacao de onda
    Ex(i,by1 ,k,p3)=Ex(i,by1 ,k,p3)-R*(e2(i,by1+2,k)./dy).*Hze(i,1,k,p3);
    Ez(i,by1 ,k,p3)=Ez(i,by1 ,k,p3)+R*(e2(i,by1+2,k)./dy).*Hxe(i,1,k,p3);

    % ymax-3
    Hxd(i,1,k,p3)=Hx(i,by2-1,k,p2)+murd(i,1,k).*(Hx(i,by2-1,k,p3)-Hxd(i,1,k,p2)); %
equacao de onda
    Hzd(i,1,k,p3)=Hz(i,by2-1,k,p2)+murd(i,1,k).*(Hz(i,by2-1,k,p3)-Hzd(i,1,k,p2)); %
equacao de onda
    Ex(i,by2 ,k,p3)=Ex(i,by2 ,k,p3)+R*(e2(i,by2-1,k)./dy).*Hzd(i,1,k,p3);
    Ez(i,by2 ,k,p3)=Ez(i,by2 ,k,p3)-R*(e2(i,by2-1,k)./dy).*Hxd(i,1,k,p3);

    % zmin+3
    Hxb(i,j,1,p3)=Hx(i,j,bz1+2,p2)+murb(i,j,1).*(Hx(i,j,bz1+2,p3)-Hxb(i,j,1,p2)); %
equacao de onda
    Hyb(i,j,1,p3)=Hy(i,j,bz1+2,p2)+murb(i,j,1).*(Hy(i,j,bz1+2,p3)-Hyb(i,j,1,p2)); %
equacao de onda
    Ex(i,j,bz1 ,p3)=Ex(i,j,bz1 ,p3)+R*(e2(i,j,bz1+2)./dz).*Hyb(i,j,1,p3);
    Ey(i,j,bz1 ,p3)=Ey(i,j,bz1 ,p3)-R*(e2(i,j,bz1+2)./dz).*Hxb(i,j,1,p3);

    % zmax-3
    Hxc(i,j,1,p3)=Hx(i,j,bz2-1,p2)+murc(i,j,1).*(Hx(i,j,bz2-1,p3)-Hxc(i,j,1,p2)); %
equacao de onda
    Hyc(i,j,1,p3)=Hy(i,j,bz2-1,p2)+murc(i,j,1).*(Hy(i,j,bz2-1,p3)-Hyc(i,j,1,p2)); %
equacao de onda
    Ex(i,j,bz2 ,p3)=Ex(i,j,bz2 ,p3)-R*(e2(i,j,bz2-1)./dz).*Hyc(i,j,1,p3);
    Ey(i,j,bz2 ,p3)=Ey(i,j,bz2 ,p3)+R*(e2(i,j,bz2-1)./dz).*Hxc(i,j,1,p3);
%-----%

%--- PEC ---%
%plano xy
    Ex(:, :, 1, p3)=0;          Ey(:, :, 1, p3)=0;
    Ex(:, :, zmax-1, p3)=0;    Ey(:, :, zmax-1, p3)=0;
%plano xz
    Ex(:, 1, :, p3)=0;          Ez(:, 1, :, p3)=0;
    Ex(:, ymax-1, :, p3)=0;    Ez(:, ymax-1, :, p3)=0;
%plano yz
    Ey(1, :, :, p3)=0;          Ez(1, :, :, p3)=0;
    Ey(xmax-1, :, :, p3)=0;    Ez(xmax-1, :, :, p3)=0;
%-----%

%---- salva resultados do campo eletrico ----%
Mxy(:, :, 1, n)=sqrt(Ex(:, :, zexc, p3).^2+Ey(:, :, zexc, p3).^2+Ez(:, :, zexc, p3).^2);
Mxz(:, 1, :, n)=sqrt(Ex(:, yexc, :, p3).^2+Ey(:, yexc, :, p3).^2+Ez(:, yexc, :, p3).^2);
Myz(1, :, :, n)=sqrt(Ex(xexc, :, :, p3).^2+Ey(xexc, :, :, p3).^2+Ez(xexc, :, :, p3).^2);
%-----%

end
%-----%

%---- apresenta resultados na tela ----%
Vol(1:xmax,1:ymax,1:zmax,1)=0; % aloca um volume para exibição
figure(1);
h=uicontrol('Style','slider','Min',0,'Max',nmax-1,'SliderStep',[1/nmax 10/nmax],...
    'Callback',[ 'vi=get(gca, 'View');'...
    'n=round(get(h, 'Value'))+1;'...
    'Vol(:, :, zexc)=Mxy(:, :, 1, n);'...
    'Vol(:, yexc, :)=Mxz(:, 1, :, n);'...
    'Vol(xexc, :, :)=Myz(1, :, :, n);'...
    'slice(Vol, yexc, xexc, zexc);'...
    'xlabel(''Y''); ylabel(''X''); zlabel(''Z'');'...
    'shading interp; axis ij; axis equal;'...
    'set(gca, 'View', vi);'...
    'title(n);'...
]);

```

```
mf = uimenu('Label','ferramentas');
    uimenu(mf,'Label','zoom','Callback','zoom on; rotate3d off');
    uimenu(mf,'Label','rotacao','Callback','zoom off; rotate3d on;');

set(gcf,'toolbar','figure');
%-----%
```

# Referências

- [1] K. S. Yee, “Numerical solutions of initial boundary value problems involving Maxwell’s equations in isotropic media”, IEEE Trans. Antennas Propagate, vol. AP-14, pp. 302-307, May 1966.
- [2] B. Engquist and A. Majda, “Absorbing Boundary Conditions for the Numerical Simulation of Waves”, Mathematics of Computational Physics, vol. 31, pp. 629-651, 1977.
- [3] G. Mur, “Absorbing Boundary Conditions for the Finite-Difference Approximation of the Time-Domain Electromagnetic-Field Equations”, IEEE Trans. Electromagnetic Compatibility, vol. 23, pp. 377-382, Nov. 1981.
- [4] J. P. Berenger, “A Perfectly Matched Layer for the Absorption of Electromagnetic Waves”, J. Computational Physics, vol. 114, pp. 185-200, 1994.
- [5] S. D. Gedney, “An Anisotropic Perfectly Matched Layer Absorbing Medium for the Truncation of FDTD Lattices”, IEEE Trans. Antennas Propagate, vol. 44, pp. 1630-1639, Dec. 1996.
- [6] M. N. O. Sadiku, “Numerical Techniques in Electromagnetics”, 2<sup>nd</sup> ed. Boca Raton
- [7] R. Diaz, I. Scherbatko, “A New Radiation Boundary Condition for FDTD based on Self-Teleportation of fields”, IEEE MTT-S Digest, 2003, pp. 2073-2076.
- [8] R. E. Diaz, I. Scherbatko, “A New Multistack Radiation Boundary Condition for FDTD based on Self-Teleportation of fields”, Journal of Computational Physics, 203 (2005) 176-190.
- [9] R. E. Diaz and I. Scherbatko, “A Simple Stackable Re-Radiating Boundary Condition (rRBC) for FDTD”, IEEE Antennas Propagation Magazine, vol. 46, no. 1, pp. 124-130, Feb. 2004.
- [10] A. M. Milagre, S. E. Barbin, “Analysis of Influence of Non-Uniform Grids Method on the Self-Teleportation of Fields Method”, XXII SBrT, Sep. 2005, pp. 451-455.
- [11] A. M. Milagre, S. E. Barbin, C. G. Christodoulou, “Improved Self Teleportation of Fields Artificial Boundary Condition for FDTD Simulations”, In: 2006 IEEE Antennas and Propagation International Symposium, 2006, Albuquerque. pp. 2719-2722.

- [12] A. M. Milagre, S. E. Barbin, “A Comparison Between the Perfectly Matched Layer and the Self – Teleportation of Fields Conditions” In: International Telecommunication Symposium 2006, Fortaleza. pp. 386-390.
- [13] A. Taflove, S. C. Hagness, “Computational Electrodynamics: The Finite-Difference Time-Domain Method”, Artech House inc. 2005.
- [14] R. L. Higdon, “Numerical Absorbing Conditions for the Wave Equation”, *Math. of Computation*, vol. 49, no. 179, pp. 65-90, July 1987.
- [15] D. M. Sheen, S. M. Ali, M. D. Abouzahra and J. A. Kong, “Application of the Three – Dimensional Finite – Difference Time – Domain Method to the Analysis of Planar Microstrip Circuits”, *IEEE Trans. Microwave Theory Tech.* vol. 38, no. 7, pp 849 – 857, July 1990.
- [16] S. S. Zivanovic, K. S. Yee and K. K. Mei, “A Subgridding Method for the Time – Domain Finite – Difference Method to solve Maxwell’s Equations”, *IEEE Trans. Microwave Theory Tech.* vol. 39, no. 3, pp. 471-479, March 1991.
- [17] A. Vaccari, R. Pontalti, C. Malacarne and L. Cristoforetti, “A robust and efficient subgridding algorithm for finite – difference time – domain simulations of Maxwell’s equations”, *Jour. Computational Physics* 194 (2004) 117-139.
- [18] T. A. Manteuffel and A. B. White, Jr., “The Numerical Solution of Second-Order Boundary Value Problems on Nonuniform Meshes”, *Math. Computation*, vol. 47, no. 176, pp. 511-535, October 1986.
- [19] P. Monk and E. Süli, “Error Estimates for Yee’s Method on Non – uniform Grids”, *IEEE Trans. Magnetics*, vol. 30, no. 5, pp. 3200-3203, September 1994.
- [20] A. C. Balanis “Advanced Engineering Electromagnetics”, John Wiley & Sons inc. 1989.
- [21] L. Zhao and A. C. Cangellaris, “GT-PML: Generalized Theory of Perfectly Matched Layers and Its Application to the Reflectionless Truncation of Finite-Difference Time Domain Grids”, *IEEE Trans. Microwave Theory Tech.*, vol. 44, no. 12, pp. 2555-2563, Dec 1996.
- [22] J. P. Berenger, “Making Use of the PML Absorbing Boundary Condition in Coupling and Scattering FDTD Computer Codes”, *IEEE Trans. Electromagnetic Compat.*, vol. 45, no. 2, pp. 189-197, May 2003.



- [23] M. Fujii, M. M. Tentzeris and P. Russer, "Performance of Nonlinear Dispersive APMML in High-Order FDTD Schemes", IEEE MTT-S Digest 2003, pp. 1129-1132.
- [24] D. S. Katz, E. T. Thiele and A. Taflove, "Validation and Extension to Three Dimensions of the Berenger PML Absorbing Boundary Condition for FD-TD Meshes", IEEE Microwave Letters, vol. 4, no. 8, pp. 268-270, Aug. 1994.
- [25] K. K. Mei and J. Fang, "Superabsorption – A Method to Improve Absorbing Boundary Conditions", IEEE Trans. Antenna Propagat., vol. 40, no. 9, pp. 1001-1010, Sept. 1992.
- [26] D. T. Prescott and N. V. Shuley, "Reflection Analysis of FDTD Boundary Conditions – Part I: Time-Space Absorbing Boundaries", IEEE Trans. Microwave Theory Tech., vol. 45, no. 8, pp. 1162-1170, Aug. 1997.
- [27] D. T. Prescott and N. V. Shuley, "Reflection Analysis of FDTD Boundary Conditions – Part II: Berenger's PML Absorbing Layers", IEEE Trans. Microwave Theory Tech., vol. 45, no. 8, pp. 1171-1178, Aug. 1997.
- [28] B. Z. Wang, Y. Wang, W. Yu and R. Mittra, "A Hybrid 2-D ADI – FDTD Subgridding Scheme for Modeling On – Chip Interconnects", IEEE Trans. Advanced Packaging, vol. 24, no. 4, pp. 528-533, Nov. 2001.
- [29] J. N. Hwang, "A Compact 2-D FDFD Method for Modeling Microstrip Structures With Nonuniform Grids and Perfectly Matched Layer", IEEE Trans. Microwave Theory Tech., vol. 53, no. 2, pp. 653-659, Feb. 2005.
- [30] X. Zhang and K. K. Mei, "Time – Domain Finite Difference Approach to the Calculation of the Frequency – Dependent Characteristics of Microstrip Discontinuities", IEEE Trans. Microwave Theory Tech. vol. 36, no. 12, pp 1775-1787, Dec. 1988.
- [31] Y. Qian and T. Itoh, "FDTD Analysis and Design of Microwave Circuits and Antennas Software and Applications", Realize Inc., 1999.