

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

ESCOLA DE ENGENHARIA

Curso de Pós-Graduação em Engenharia Elétrica - CPGEE

**EFEITOS BIOLÓGICOS DAS RADIAÇÕES DE RÁDIO  
FREQUÊNCIAS DE TELEFONIA CELULAR  
-  
SIMULAÇÃO FDTD**

CLAUDIO ENRIQUE FERNÁNDEZ RODRÍGUEZ

Dissertação para obtenção do título de Mestre em Engenharia

Porto Alegre

2001

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

ESCOLA DE ENGENHARIA

Curso de Pós-Graduação em Engenharia Elétrica - CPGEE

**EFEITOS BIOLÓGICOS DAS RADIAÇÕES DE RÁDIO  
FREQUÊNCIA DE TELEFONIA CELULAR  
SIMULAÇÃO FDTD**

CLAUDIO ENRIQUE FERNÁNDEZ RODRÍGUEZ

Engenheiro Eletricista

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia Elétrica - CPGEE, como parte dos requisitos para a obtenção do título de Mestre em Engenharia.  
Área de concentração: Eletrônica e Comunicações.  
Desenvolvida no Laboratório de Comunicações Eletro-Óticas do Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul.

Porto Alegre

2001

ii

# **EFEITOS BIOLÓGICOS DAS RADIAÇÕES DE RÁDIO FREQUÊNCIA DE TELEFONIA CELULAR SIMULAÇÃO FDTD**

CLAUDIO ENRIQUE FERNÁNDEZ RODRÍGUEZ

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: \_\_\_\_\_

Prof. Álvaro Augusto de Almeida Salles, UFRGS

PhD. pela University College of London, UK

Banca Examinadora:

Prof. Francisco de Assis Tejo, UFPB

Dr. pela Universidade Federal da Paraíba – UFPB – 1996

Prof. Marcelo S. Alencar, UFPB.

PhD. pela University of Waterloo – 1993

Prof. Jorge R. S. Sabadal, UFRGS

Dr. pela Universidade Federal do Rio Grande do Sul – UFRGS – 1994

Prof. Jorge A. Lisbôa, UFRGS.

Dr. pela Universidade Federal do Rio Grande do Sul – UFRGS – 1988

Coordenador do CPGEE: \_\_\_\_\_

Prof. Dr. Altamiro A. Susin

Porto Alegre, agosto de 2001.

*para Juan Felipe y Diego*

## **AGRADECIMENTOS**

Aos Professores Dr. Álvaro Salles e Dr. Jorge Lisbôa pela orientação e colaboração em todas as etapas do trabalho.

Ao Professor Dr. Marco Vilhena por traduzir sua vasta experiência com métodos numéricos em certos conselhos.

Ao Engenheiro, MSc. Luis Schäffer pela colaboração na concepção e nas primeiras etapas do desenvolvimento do algoritmo apresentado.

Ao Engenheiro Rodrigo Stoll pela implementação dos algoritmos originais de forma adequada ao supercomputador.

Ao Engenheiro, MSc. (cdt) Sérgio Severo pelo intercâmbio de idéias sobre o algoritmo implementado e pela colaboração na implementação de rotinas de saída gráfica.

Ao Engenheiro (cdt) Mateus Bonadimann pela valiosa ajuda com a construção do domínio do problema, nas simulações e implementações de antenas.

Ao Engenheiro Paulo Serafini pela ajuda com o projeto, implementação e medições de antenas.

Aos funcionários do CESUP

A todos pelo incentivo além das colaborações, meus agradecimentos.

# SUMÁRIO

<b>LISTA DE FIGURAS.....</b>	<b>viii</b>
<b>LISTA DE TABELAS.....</b>	<b>x</b>
<b>LISTA DE ABREVIATURAS E SÍMBOLOS.....</b>	<b>xi</b>
<b>RESUMO.....</b>	<b>xiv</b>
<b>ABSTRACT .....</b>	<b>xv</b>
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1 MOTIVAÇÃO .....	1
1.2 PLANO DO TRABALHO .....	1
<b>2 CONTEXTO DO PROBLEMA .....</b>	<b>3</b>
2.1 DIFUSÃO DAS COMUNICAÇÕES PESSOAIS VIA RÁDIO .....	3
2.2 ONDAS ELETROMAGNÉTICAS .....	4
2.3 EFEITOS BIOLÓGICOS E RISCOS À SAÚDE .....	5
2.3.1 Efeitos Térmicos.....	6
2.3.2 Efeitos Não-Térmicos.....	7
2.3.3 Efeitos e Riscos .....	9
2.4 EXPOSIÇÃO HUMANA A CAMPOS ELETROMAGNÉTICOS.....	9
2.4.1 Parâmetros Dosimétricos.....	9
2.4.2 Principais Normas e Recomendações.....	11
2.4.3 Postura Precautória.....	12
2.5 CAMPO DISTANTE .....	12
2.5.1 Cálculos para uma Estação de Radiocomunicação.....	13
2.5.2 Medidas .....	16
<b>3 SIMULAÇÃO NUMÉRICA.....</b>	<b>17</b>
3.1 MÉTODO FDTD.....	18
3.1.1 Calculo dos Campos Elétricos e Magnéticos .....	19
3.2 MODELO DO DOMÍNIO DO PROBLEMA .....	24
3.3 MODELO DA FONTE .....	28
3.4 SIMULAÇÃO DO ESPAÇO LIVRE .....	30
3.5 CONVERGÊNCIA DO MÉTODO .....	32
3.6 VALIDAÇÃO DO ALGORITMO IMPLEMENTADO.....	33
3.7 O ALGORITMO IMPLEMENTADO .....	34
<b>4 RESULTADOS OBTIDOS.....</b>	<b>35</b>
4.1 COMPARAÇÕES COM OUTRAS SIMULAÇÕES .....	40
4.2 COMPARAÇÕES COM MEDIDAS .....	41

<b>5</b>	<b>PROJETO DE ANTENAS DIRETIVAS .....</b>	<b>43</b>
5.1	PROJETO DE ANTENAS USANDO FDTD .....	44
5.2	MEDIDAS DE PROTÓTIPOS.....	47
<b>6</b>	<b>CONCLUSÕES E RECOMENDAÇÕES .....</b>	<b>52</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>54</b>
	<b>ANEXOS .....</b>	<b>59</b>
ANEXO1	LISTAGEM DO PROGRAMA PRINCIPAL .....	59
ANEXO2	LISTAGEM DA ROTINA DE INDEXAÇÃO DOS TECIDOS .....	114

## LISTA DE FIGURAS

Fig. 1. Evolução do uso de telefones celulares no mundo.	3
Fig. 2. Evolução do uso de telefones celulares no Brasil.	4
Fig. 3. Espectro Eletromagnético.	5
Fig. 4. Limites de Exposição em Função da Frequência.	11
Fig. 5. Distâncias para o atendimento às normas ICNIRP e ANSI para uma ERB típica.	14
Fig. 6. Distâncias para o atendimento às legislações Suíça e Italiana ERB típica.	15
Fig. 7. Distâncias para o atendimento às legislações Suíça e Italiana: plano vertical	15
Fig. 8. Espectro electromagnético observado nas proximidades de duas ERB (Bandas A e B).	16
Fig. 9. Diagrama de Blocos dos algoritmos implementados.	16
Fig. 10. Célula de Yee.	20
Fig. 11. Célula de Yee modificada.	23
Fig. 12. Cortes sagitais (real e de ressonância magnética).	25
Fig. 13. Imagem de corte sagital idealizado, usado para a geração do domínio de cálculo.	25
Fig. 14. Dimensões do modelo utilizado.	26
Fig. 15. Plano da matriz gerada. Cada índice aponta a um material.	27
Fig. 16. Diagramas de irradiação de um dipolo e de uma fonte isotrópica.	28
Fig. 17. Simulação de um monopolo	29
Fig. 18. Campo Elétrico observado (vista superior e lateral para uma simulação teste de um monopolo).	29
Fig. 19. Pulso propagando para bordas descasada e casada.	30
Fig. 20. Monitorização ao longo das iterações dos valores de diferentes componentes de campo elétrico.	33
Fig. 21. Monitorização ao longo das iterações do valor de SAR calculado.	35
Fig. 22. SAR num corte sagital da cabeça do usuário em mW/g por V/m RMS.	36
Fig. 23. SAR na cabeça do usuário para um corte sagital, depois de 1ns, 2ns e 3ns de exposição.	37
Fig. 24. SAR na cabeça do usuário para diferentes cortes e distâncias.	38
Fig. 25. Elevados valores da SAR internos.	39



Fig. 26. SAR normalizada para uma potência entregue à antena $P_{del} = 600$ mW.	40
Fig. 27. Comparações de resultados simulados.	41
Fig. 28. Comparações de resultados medidos.	42
Fig. 29. Representação de uma matriz contendo a antena microstrip.	45
Fig. 30. Propagação de campos eletromagnéticos para a antena microstrip.	46
Fig. 31. Cortes transversais da antena microstrip.	46
Fig. 32. Campo elétrico propagado a partir da antena microstrip, para uma vista longitudinal.	47
Fig. 33. Propagação de campos eletromagnéticos a partir da antena microstrip, para uma vista de topo.	47
Fig. 34. Diagrama de irradiação medido de uma antena microstrip no plano horizontal	49
Fig. 35. Diagrama de irradiação medido de uma antena microstrip no plano vertical	50
Fig. 36. Taxa de onda estacionária medida na faixa de 837,5 a 865 MHz.	50

## **LISTA DE TABELAS**

Tabela 1. Parâmetros dos tecidos usados na simulação. ERRO! INDICADOR NÃO DEFINIDO.

Tabela 2. Parâmetros dos materiais usados na simulação. ERRO! INDICADOR NÃO DEFINIDO.

## LISTA DE ABREVIATURAS E SÍMBOLOS

$\Delta t$	Passo temporal (intervalo de tempo, diferença finita).
$\Delta x, \Delta y, \Delta z$	Intervalos das variáveis espaciais $x y z$ (diferenças finitas).
$\partial E/\partial t$	Diferencial de campo elétrico em relação ao tempo.
$\partial H/\partial t$	Diferencial de campo magnético em relação ao tempo.
$\partial m$	Diferencial de massa (massa do elemento).
$\partial V$	Diferencial de volume (volume do elemento).
$\partial W$	Diferencial de energia ( $J$ ) em Joules.
$\delta$	Profundidade de penetração pelicular, em ( $mm$ ) milímetros.
$\epsilon_0$	Constante dielétrica do vácuo, permissividade do vácuo.
$\epsilon_r$	Constante dielétrica relativa, permissividade relativa do meio.
$\epsilon_r'$	Parte real da constante dielétrica.
$\epsilon_r''$	Parte imaginária da constante dielétrica.
$\phi, \phi_m$	Coordenada esférica ( $r, \theta, \phi$ ).
$\lambda$	Comprimento de onda, em ( $m$ ) metros.
$\nu$	Frequência em ( $Hz, MHz, GHz$ ) Hertz, Megahertz ou Gigahertz.
$\mu$	Permeabilidade magnética.
$\pi$	Pi, 3,14159267.
$\rho$	Massa específica, em ( $kg/l$ ) kilograma por litro.
$\theta, \theta_m$	Coordenada esférica ( $r, \theta, \phi$ ).
$\sigma$	Condutividade, em ( $S/m$ ) Siemens por metro.
$\sigma^*$	Termo dual da condutividade, para o campo magnético.
$\nabla$	Operador nabla.
ABC	<i>Absorbing Boundary Condition</i> – Condição de Borda Absorvedora.
ADI	<i>Alternative Direction, Implicit</i> – Método numérico implícito de direção alternada.
ANSI	<i>American National Standards Institute</i> – Instituto de normatização dos Estados Unidos.
ANATEL	Agência Nacional de Telecomunicações.
Bandas A e B	Faixas de frequência outorgadas às operadoras do serviço móvel celular.
BIFA	<i>Bent Inverted F Antennas</i> – Antena de microlinha invertida tipo F sobre uma superfície curva.
$b_f$	Densidade de fluxo sangüíneo, em ( $m^3/s$ ) metros cúbicos por segundo.
C++	Linguagem de programação.
C95.1	Norma de exposição a campos eletromagnéticos do IEEE de 1992.
$C_b$	Calor específico do sangue arterial, em ( $J/kg \text{ } ^\circ C$ ) Joules por kilograma e por grau Celsius.
CDMA	<i>Code Division Media Access</i> – Multiplexação por código.
CENELEC	<i>Comité Européen de Normalisation Electrotechnique</i> – Comitê Europeu de Normatização Eletrotécnica.

COST	<i>CO</i> peration <i>eu</i> ropéenne dans le domaine de la recherche Scientifique et Technique – Modelo de rádio-propagação.
$C_p$	Calor específico, em (J/kg °C) Joules por quilograma e por grau Celsius.
CPU	Unidade Central de Processamento
CPW	<i>Coplanar Waveguide</i> – Estrutura planar com guias de onda coplanares.
CRAY T94	Supercomputador modelo T94 da CRAY.
$D_m$	Função de diretividade da antena.
DNA	Ácido Desoxirribonucléico
$dT/dt$	Diferencial de temperatura no intervalo diferencial de tempo $dt$ .
$E_x, E_y, E_z$	Componentes de Campo Elétrico (V/m) em Volts por metro.
ENV50166-2	Norma de exposição a campos eletromagnéticos do ICNIRP de 1998
ERB	Estação radiobase.
$f$	Frequência.
FCC	<i>Federal Communications Commission</i> – Agência governamental para a regulamentação das telecomunicações dos Estados Unidos.
FDTD	<i>Finite Differences Time-Domain</i> – Método numérico explícito das Diferenças Finitas no Domínio do Tempo.
FR-4 V0	Substrato dielétrico tipo <i>Perstorp</i> .
GB	Gigabyte, unidade de armazenamento de dados.
GFLOP	Unidade de velocidade de processamento.
$h$	Constante de Planck, $6,626 \times 10^{-34}$ (J×s) Joules multiplicados por segundos.
$h_{CONV}$	Perdas térmicas por convecção.
$h_E$	Perdas térmicas por evaporação.
$h_m$	Calor gerado pelo metabolismo.
$h_{RAD}$	Perdas térmicas por radiação.
$H_x, H_y, H_z$	Componentes do Campo Magnético (A/m)
HSP	<i>Heat-Shock Proteins</i> – Proteína de choque térmico.
ICNIRP	<i>International Commission on Non-Ionizing Radiation Protection</i> – Comissão Internacional de Proteção de Radiação Não Ionizante.
IEEE	<i>The Institute of Electrical and Electronics Engineers</i> – Instituto dos Engenheiros Eletro Eletrônicos dos Estados Unidos.
MoM	Método numérico dos Momentos.
MS 2711	Modelo de analisador de espectro marca Anritsu.
$n$	Índice de iteração.
$N_m$	Número de canais, para cada setor.
OMS	Organização Mundial de Saúde.
PCs	Computadores pessoais.
PIFA	<i>Planar Inverted F Antenna</i> – Antena de microlinha invertida tipo F.
$P_m$	Potência por canal efetivamente fornecida à antena.
PML	<i>Perfect Matching Layer</i> – Camada perfeitamente casada.
$R(\theta, \phi)$	Coordenada esférica de distância.
raios-X	Radiação com frequências de aproximadamente $10^{20}$ Hz.
RF	Radiofrequências.
RGB	<i>Red, Green &amp; Blue</i> , componentes de cor.
RMS	Valor médio quadrático.
RNA	Ácido ribonucléico
SAR	<i>Specific Absorption Rate</i> – Taxa de Absorção Específica em (mW/g)
SCSI	<i>Small Computer System Interface</i> – Inteface de computadores.
Slot-line	Tipo de estrutura planar (de fenda)

SMA	Tipo de conector.
$T$	Temperatura em ( $^{\circ}C$ ) graus Celsius.
$T_b$	Temperatura do sangue arterial.
$U_{max}$	Densidade de potência, em ( $mW/cm^2$ ) miliwatts por centímetro quadrado.
$V$	Volume em ( $m^3$ ) metros cúbicos.

## RESUMO

Neste trabalho é apresentado o desenvolvimento de um programa para cálculo de campos eletromagnéticos baseado no método das Diferenças Finitas no Domínio do Tempo (FDTD). Este programa é aplicado no cálculo da Taxa de Absorção Específica (SAR) na cabeça de um usuário de transceptores portáteis como os telefones celulares. Como introdução ao problema é realizada uma revisão dos efeitos biológicos e das recomendações para a exposição humana a radiofrequências e microondas. Os principais resultados obtidos com simulações realizadas com o programa desenvolvido. São mostrados e comparados com os resultados simulados e medidos por outros autores, assim como com as recomendações nacionais e internacionais. A título de recomendação final e como alternativa visando minimizar os riscos que a absorção do campo pode representar para a saúde dos usuários, é proposta a utilização de antenas que emitem mais no sentido oposto à cabeça. Este conceito é testado com o desenvolvimento de uma antena simples em *microstrip*.

## **ABSTRACT**

The development of a software to compute the electromagnetic fields, employing the Finite Difference Time Domain (FDTD) method is shown in this work. This software is used to estimate the Specific Absorption Rate (SAR) in the head of users of portable transmitters (such as cellular phones). A review of the biological effects of microwaves and radio frequencies, as well as the guidelines for human exposure are included. The relevant results obtained with this model, are compared with the guidelines as well as with results estimated or measured by other authors. The use of directive antennas, radiating energy away from the user's head, is proposed as an alternative aiming to reduce health risks. In order to test this concept, a very simple microstrip antenna is developed.

# 1 INTRODUÇÃO

O interesse no estudo dos efeitos biológicos de radiofrequência e microondas têm aumentado com a popularização das comunicações pessoais. O estudo específico da distribuição de campo para o caso do transmissor portátil requer o uso de uma ferramenta de cálculo numérico, como o Método das Diferenças Finitas no Domínio do Tempo FDTD, quando se deseja conhecer a distribuição de campo em campo próximo. A aplicação desta ferramenta não se restringe a esse problema. Os cálculos de espalhamento com ela realizados podem servir a diferentes aplicações, entre as quais o projeto de antenas diretivas que possam minimizar os efeitos biológicos adversos sobre os usuários dos transceptores portáteis.

## 1.1 MOTIVAÇÃO

Os níveis de exposição humana a campos eletromagnéticos são limitados em diferentes recomendações e legislações. Porém, as dificuldades encontradas para as medições e estimativas teóricas em condições de campo próximo levam à necessidade de comprovação do atendimento desses limites no caso dos transmissores portáteis.

Considerando estas dificuldades para a avaliação teórica e para a dosimetria dos campos eletromagnéticos na proximidade dos elementos radiantes, a pequena distância entre as antenas dos telefones celulares e a cabeça do usuário, o que configura esta situação de “campo próximo”, e o grande número de usuários que atualmente usam este tipo de telefonia, o desenvolvimento de um método de avaliação teórico para o estudo deste problema se torna relevante.

## 1.2 PLANO DO TRABALHO

Inicialmente, no Capítulo 2, uma discussão introdutória dos efeitos biológicos das radiações eletromagnéticas tentará colocar em contexto o problema, mostrando a abrangência da questão. A seguir, no Capítulo 3, será apresentado o desenvolvimento de uma ferramenta de análise numérica baseada no método de diferenças finitas no domínio do tempo (FDTD) para determinar a distribuição dos campos eletromagnéticos emitidos pela antena de um



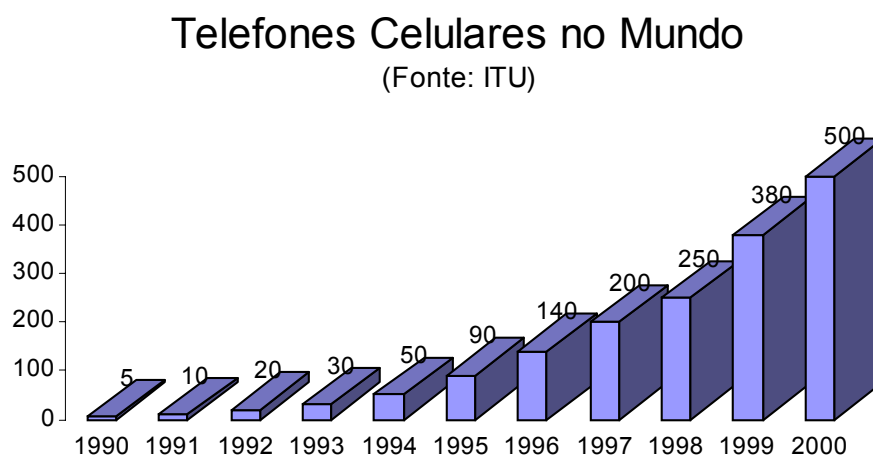
telefone celular e absorvidos pela a cabeça do usuário. No Capítulo 4, os resultados obtidos com o algoritmo desenvolvido serão comparados com outros resultados publicados, simulados e medidos e com alguns limites de exposição recomendados. No Capítulo 5, o projeto e teste preliminar de antenas diretivas, que minimizam os campos absorvidos na cabeça do usuário, é mostrado como solução conceitual ao problema. Considerando que o algoritmo implementado foi usado para simulações da antena projetada, este desenvolvimento serve também como exemplo da aplicação da ferramenta computacional desenvolvida. Finalmente, outras conclusões, recomendações e uma breve resenha de possíveis desenvolvimentos futuros são incluídos no Capítulo 6.

## 2 CONTEXTO DO PROBLEMA

A difusão do uso de transceptores portáteis, como os telefones celulares, tem levantado questionamentos a respeito dos efeitos que o campo eletromagnético induzido na cabeça dos usuários pode provocar. Para avaliar estas questões é necessário o desenvolvimento de uma solução numérica, implementada num *software* que tenha aplicação geral para problemas de propagação e absorção de campo eletromagnético, em campo próximo ou distante, e que possa ser aplicado para determinar a distribuição dos campos eletromagnéticos provenientes das antenas dos transceptores portáteis, sobre o usuário.

### 2.1 DIFUSÃO DAS COMUNICAÇÕES PESSOAIS VIA RÁDIO

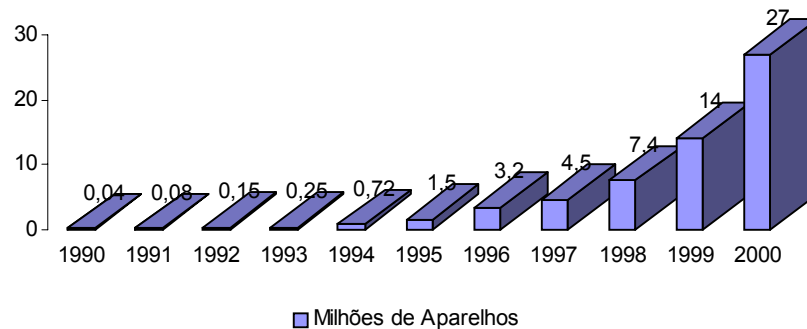
Embora o uso de radiofrequências remonte às experiências de Marconi em 1895 e Landell de Moura em 1879, e o primeiro uso corporativo do rádio em comunicações pessoais ocorreu em 1921 com as primeiras radiopatrulhas em Detroit - EEUU [1]. No entanto, é apenas a partir da década de 1980 com o surgimento da telefonia celular e, principalmente depois de sua difusão nos anos 1990, como ilustram as Figuras 1 e 2, que grandes segmentos da população começaram a usar transceptores pessoais de rádio.



**Figura 1. Evolução do uso de telefones celulares no mundo.**

## Telefones Celulares no Brasil

(Fonte: ANATEL)



**Figura 2. Evolução do uso de telefones celulares no Brasil.**

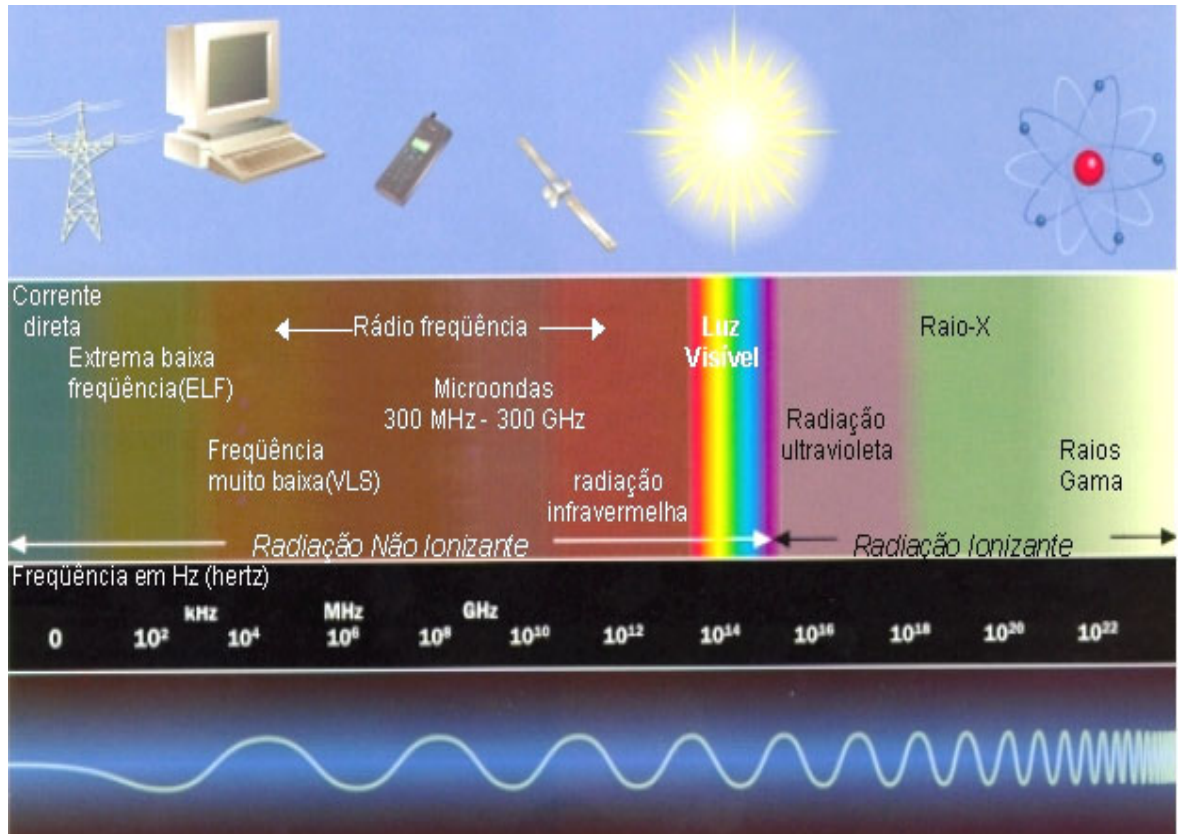
Hoje em dia são mais de 30 milhões de usuários apenas no Brasil e ainda há previsões de que a penetração acelerada desse serviço continue atingindo parcelas ainda maiores da população. Ao lado dessa popularização, tem crescido também a preocupação a respeito dos possíveis riscos à saúde causados pela exposição aos campos eletromagnéticos emitidos por esses equipamentos de comunicação. Note-se que nestes equipamentos, os campos emitidos não correspondem a um efeito desnecessário e indesejado, mas a parte fundamental do processo de comunicação.

## 2.2 ONDAS ELETROMAGNÉTICAS

Os canais de transmissão dos telefones celulares portáteis usam portadoras de radiofrequências, por exemplo, entre 800 MHz e 900 MHz para as bandas A e B, no Brasil. Desta forma, os sinais de voz que devem ser transmitidos são transformados em ondas eletromagnéticas com frequências próximas a 1 GHz (ou  $10^9$  Hz), as quais devem ser emitidas com potência suficiente para serem recebidas pelas estações de radiobase (ERB) do sistema celular.

Este tipo de ondas é classificado como sendo “não ionizante” (Figura 3) diferentemente das radiações de maior frequência (e.g., a partir de  $10^{15}$  Hz). De fato, radiação em comprimentos de onda menores que o da radiação ultravioleta começam a ter a capacidade de

ionizar a matéria (e são, portanto, consideradas “ionizantes”), pois possuem energia capaz de arrancar elétrons dos átomos irradiados.



**Figura 3. Espectro Eletromagnético.**

A energia  $E$  da onda depende linearmente da frequência  $\nu$  e da constante de Planck  $h$ , de acordo com a equação

$$E = h\nu \quad (1)$$

O objeto de estudo deste trabalho são as radiações não ionizantes na faixa de radiofrequências (ou próximas às de microondas) usadas pelos telefones celulares para a transmissão.

### 2.3 EFEITOS BIOLÓGICOS E RISCOS À SAÚDE

A interação de ondas eletromagnéticas com a matéria orgânica viva pode implicar em efeitos prejudiciais e também em aplicações médicas nas quais o risco deve ser controlado.

Para o caso das radiações ionizantes, diversas aplicações e severos efeitos danosos foram identificados, há já várias décadas. Por exemplo, a radioterapia ou as chapas de raios-X têm aplicações benéficas. Por outro lado, é conhecido que a exposição prolongada a esses mesmos raios-X pode provocar alguns tipos de tumores de pele [2] e outros efeitos prejudiciais, inequivocamente identificados desde as primeiras décadas de pesquisa e uso dessas radiações.

Para o caso das radiações não ionizantes além de problemas de compatibilidade eletromagnética que podem causar interferência, por exemplo, em equipamentos hospitalares [3, 4], alguns efeitos associados ao aquecimento provocado pela absorção da energia da onda, os efeitos térmicos, também têm sido identificados, existindo aplicações e recomendações de limitações.

O conhecimento e a identificação de efeitos biológicos provocados por radiofrequências como as dos telefones celulares traz, apesar do receio de que possam provocar danos à saúde, uma esperança otimista do desenvolvimento de novas aplicações médicas.

### 2.3.1 Efeitos Térmicos

Tecidos biológicos submetidos a campos de radiações não ionizantes podem sofrer aquecimento, dependendo da frequência e da potência do sinal. Em frequências mais baixas podem ocorrer correntes induzidas com as correspondentes perdas ôhmicas.

Em frequências para as quais o corpo humano ou parte dele podem se comportar como uma boa antena, ou seja em frequências com o comprimento de onda da mesma ordem de grandeza que o corpo humano ou suas partes, como na faixa de radiofrequências, os efeitos térmicos são causados pelo aquecimento direto dos tecidos biológicos como resultado da absorção da energia eletromagnética num meio dissipativo. Por exemplo, nos meios dielétricos dissipativos nas quais a permissividade (ou constante dielétrica) apresenta uma parte imaginária maior que  $\varepsilon_0$ , a parte imaginária ( $\varepsilon_r''$ ) da permissividade relativa pode ser relacionada a uma condutividade equivalente

$$\sigma = 2\pi f \varepsilon_0 \varepsilon_r'' \quad (S / m) \quad (2)$$

em que  $f$  é a frequência e  $\varepsilon_0$  é a permissividade do vácuo. Tanto os valores das partes real ( $\varepsilon_r'$ ) e imaginária ( $\varepsilon_r''$ ) da permissividade relativa ( $\varepsilon_r = \varepsilon_r' - j \varepsilon_r''$ ), bem como da condutividade equivalente  $\sigma$ , variam substancialmente com a frequência e com o tipo de tecido [5]. Ao contrário das radiações não ionizantes em comprimentos de onda menores (e.g.,

no infravermelho, ou no visível), as radiações em microondas e em rádio frequência (RF) não são absorvidas apenas pela pele, mas, dependendo da frequência, em camadas mais profundas de tecidos também. Calculando a profundidade de penetração pelicular  $\delta$ , para frequências de RF e microondas de acordo com

$$\delta = \sqrt{\pi f \mu \sigma_{eq}} \quad (3)$$

obtem-se, para valores de condutividade médios,  $\sigma_{eq}$  [6], de tecidos como pele ou osso, e para uma frequência  $f$  de 850 MHz, por exemplo,  $\delta_{pele} = 22 \text{ mm}$ ,  $\delta_{osso} = 16 \text{ mm}$ . Uma vez que os sensores de temperatura do corpo humano estão localizados somente na periferia do corpo (endoderme), efeitos prejudiciais aos tecidos podem ocorrer devido a aquecimentos excessivos em regiões mais profundas, sem serem percebidos pelas pessoas [7]. Note-se que as radiações predominantes no ambiente natural, provenientes do sol, estão nas faixas do visível e do infravermelho próximo, (e são absorvidas então nas camadas superficiais da pele) para as quais o organismo tem capacidade sensorial e imunidade evolutiva.

Então, genericamente, todos os efeitos que podem acarretar um aumento de temperatura nos tecidos são chamados efeitos térmicos. Esses efeitos térmicos têm sido estudados há muitas décadas, e os resultados da absorção dos campos eletromagnéticos por diferentes tipos de tecidos são relativamente bem conhecidos, existindo mesmo normas internacionalmente aceitas que estabelecem limites de exposição em função da frequência de operação, do tempo de exposição, do tipo de usuário, etc., [8, 9]. Os limites de exposição destas recomendações consideram que aquecimentos acima de 1 °C podem provocar efeitos teratogênicos ou degenerativos [10].

### 2.3.2 Efeitos Não-Térmicos

Além dos fenômenos de aquecimento, pode-se esperar que ocorra interação direta dos campos eletromagnéticos oscilatórios, principalmente com as moléculas polares presentes nos fluidos e tecidos biológicos. Os efeitos “não-térmicos” são, por exemplo, efeitos bioquímicos ou eletrofísicos causados, diretamente, pelos campos eletromagnéticos induzidos, e não por um aumento localizado ou distribuído de temperatura. Alguns efeitos “não-térmicos” relatados na literatura incluem efeitos nos sistemas nervoso, cardiovascular e imunológico, bem como no metabolismo e em fatores hereditários [11-13]. Entretanto, nessas áreas os resultados ainda são polêmicos, não existindo conclusões definitivas, o que poderá ainda demorar muitos anos. Alguns resultados são mesmo conflitantes, especialmente devido a técnicas experimentais não muito confiáveis. Por exemplo: (a) os métodos utilizados para

caracterizar os sintomas em estudos epidemiológicos; (b) dosimetria em RF e microondas, especialmente em campo próximo; e (c) a presença de influências estranhas não consideradas na interpretação dos resultados.

Alguns efeitos sobre funções específicas ou em cobaias foram claramente demonstrados, restando ainda as dúvidas sobre se estes efeitos se constituem em riscos à saúde. Entre estes efeitos se incluem a alteração no fluxo de íons através das membranas das células (afetando particularmente as propriedades eletro-fisiológicas das células nervosas), alteração na mobilidade dos íons de cálcio (particularmente nos tecidos do cérebro), alterações na síntese de DNA e na transcrição de RNA e efeitos na resposta de células normais a moléculas sinalizantes (incluindo hormônios, neurotransmissores e fatores de crescimento) [12]. Alterações no fluxo de cálcio em células, na barreira hemato-encefálica (*blood-brain barrier*, que protege o cérebro de certas toxinas) e no desenvolvimento de tumores cerebrais foram também relatados [14].

Entre outras, uma incerteza importante, que está para ser confirmada, é se esses efeitos não térmicos podem ou não ocorrer em taxas de absorção específicas bem abaixo daquelas observadas para os efeitos térmicos. Em cobaias, foram constatados efeitos no sistema imunológico, nas células nervosas e a ocorrência de câncer ou carcinomas para níveis de exposição abaixo dos limiares térmicos [12]. Também foram relatadas alterações no sistema endócrino, na química sangüínea e nos sistemas hematológicos e imunológicos para exposições prolongadas em níveis menores que os recomendados pelas normas [14].

Recentemente, foi relatada a indução de proteínas de choque térmico (HSP- *Heat-Shock Proteins*) em vermes do tipo nematóides (*Caenorhabditis Elegans*) com exposição prolongada a campos eletromagnéticos de baixa intensidade, (três ordens de grandeza abaixo dos valores para os quais ocorrem os efeitos térmicos) o que foi atribuído a mecanismos não térmicos que podem também ocorrer em tecidos humanos [15, 16]. Isto sugere, então, que os limites atuais de exposição devem ser reconsiderados.

Entre as muitas dúvidas que ainda estão por ser esclarecidas é possível ressaltar, por exemplo, como o campo eletromagnético atua em determinadas estruturas, como: (a) nos cromossomos ou nas moléculas de DNA que constituem os genes [5], e (b) na alteração da mobilidade dos íons (por exemplo, de cálcio), particularmente em tecidos do cérebro e nas propriedades eletro-fisiológicas das células nervosas [12]. Por outro lado, é sabido que um estágio inicial do desenvolvimento do câncer é caracterizado por uma alteração no código genético de células do corpo, isto é, a molécula de DNA.

### 2.3.3 Efeitos e Riscos

Ensaio reprodutíveis, como os observados em nematóides [15, 16], indicam que ondas eletromagnéticas não ionizantes podem interagir com seres vivos provocando respostas identificáveis mesmo em níveis de potência muito inferiores aos níveis para os quais se identificam os efeitos térmicos. A existência dos “efeitos não térmicos”, a rigor, pode ser provada com a repetição deste tipo de experiências. Porém, a identificação de um efeito como este, que não está associado diretamente a um risco à saúde, isto é, não é um efeito danoso ou não reversível, ainda não seria suficiente para rever os limites de exposição recomendados.

Reverendo o histórico das recomendações e legislações, observa-se que, à medida que as pesquisas avançam e novos resultados aparecem, as normas são atualizadas e novos limites mais restritivos são sugeridos [17]. A construção do conhecimento científico requer a verificação de estudos estatísticos demorados e a repetição de ensaios como o dos nematóides, com o teste de hipóteses alternativas para os fenômenos observados. No estágio atual das pesquisas uma maioria de publicações relata que não foram encontrados efeitos significativos das radiações de radiofrequências em potências abaixo das que provocam efeitos térmicos, em seres humanos. Isto não é ainda suficiente para poder descartar com rigor científico os resultados de uma quantidade já significativa de trabalhos que indicariam a existência desses efeitos não térmicos [18].

## 2.4 EXPOSIÇÃO HUMANA A CAMPOS ELETROMAGNÉTICOS

Diversas normas e leis definem os limites de exposição a partir dos efeitos térmicos, no caso das recomendações mais difundidas (ANSI, ICNIRP) [8, 9], ou de projeções que antecipam a demonstração da existência de efeitos não térmicos, prejudiciais, mesmo em níveis mais baixos (normas suíça e italiana) [19, 20].

### 2.4.1 Parâmetros Dosimétricos

Os limites indicados nas recomendações podem ser expressos em densidade de potência incidente (e.g., em  $mW/cm^2$ ), intensidade de campo elétrico ( $V/m$ ) ou magnético ( $A/m$ ), ou de um parâmetro dosimétrico largamente utilizado: a “Taxa de Absorção Específica” (ou SAR - *Specific Absorption Rate*). Esta última é definida como sendo a derivada no tempo do aumento de energia “ $\partial W$ ” absorvida ou dissipada num elemento de massa “ $\partial m$ ” contida num elemento de volume “ $\partial V$ ” cuja massa específica é “ $\rho$ ” [14, 21], e que analiticamente pode ser expressa por



$$SAR = \frac{\partial}{\partial t} \frac{\partial W}{\partial m} = \frac{\partial}{\partial t} \frac{\partial W}{\rho (\partial V)} (mW / g). \quad (4)$$

É possível dizer, em outras palavras, que a SAR quantifica a potência absorvida por unidade de massa. Utilizando o teorema do Vetor de Poynting para campos eletromagnéticos com excitação senoidal no domínio da frequência, a SAR pode ser também expressa por

$$SAR = \frac{\sigma}{2\rho} |E_i|^2 = \frac{\omega \epsilon_0 \epsilon_r}{2\rho} |E_i|^2 = \frac{|J_i|^2}{2\rho\sigma} (mW / g), \quad (5)$$

em que  $E_i$  e  $J_i$  são os valores de pico do campo elétrico e da densidade de corrente no local considerado. Observa-se que a SAR é diretamente proporcional ao aumento local de temperatura, responsável pelos efeitos térmicos

$$\frac{\partial T}{\partial t} = \frac{SAR}{C_p} (^\circ C / s), \quad (6)$$

em que  $T$  é a temperatura, e  $C_p$  é o calor específico do tecido ( $J/kg \times ^\circ C$ ). Podem ser implementadas aproximações mais precisas (como a da Equação 7), para o cálculo do aumento de temperatura  $dT/dt$  num volume  $V$ , levando em consideração as perdas térmicas por radiação  $h_{RAD}$ , convecção  $h_{CONV}$  e evaporação  $h_E$ , as trocas de calor pelo fluxo sanguíneo  $b_f$  (sendo  $T_b$  e  $C_b$  a temperatura e o calor específico do sangue arterial) e o calor gerado pelo metabolismo  $h_m$ .

$$mC \frac{\partial T}{\partial t} = [\nabla (k \nabla T) + h_m + h_{EM} + b_f C_b (T_b - T)] \times V - h_{RAD} - h_{CONV} - h_E. \quad (7)$$

A Equação 7 [22] não pode ser resolvida por métodos explícitos como FDTD, sendo indicado, para a solução da mesma, outros métodos numéricos, implícitos, como, por exemplo, ADI (*alternative direction, implicit*) [23].

Para exposição do corpo inteiro, por exemplo, pode-se considerar a SAR média, que será então a relação entre a potência total absorvida pelo corpo e sua massa. Entretanto, para aquecimentos localizados, como os produzidos pelos telefones celulares emitindo muito próximo (por exemplo, a 2 cm, 1 cm ou menos) da cabeça do usuário, deve ser usada a SAR local, que é definida como a potência absorvida (e.g., em  $mW$ ) por um grama de tecido [9].

## 2.4.2 Principais Normas e Recomendações

Em função dos bem estabelecidos efeitos térmicos, diferentes normas têm sido propostas [17] entre as quais se destacam a norma C95.1 de 1992 do IEEE/ANSI (*The Institute of Electrical and Electronics Engineers / American National Standards Institute*) [8] adotada nos Estados Unidos e a ENV50166-2, de 1996/1998 do ICNIRP / CENELEC (*International Commission on Non-Ionizing Radiation Protection / Comité Européen de Normalisation Electrotechnique*) [9], adotada por diferentes países europeus. Estas recomendações servem de base para diferentes legislações nacionais ou locais [19-21, 24, 25]. As expressões dos valores limites (em forma gráfica na Figura 4), para as faixas de radiofrequências e microondas, próximas às usadas em serviços de comunicação pessoal são dadas pelas Equações 8 e 9.

$$ANSI \rightarrow P\left(\frac{W}{m^2}\right) = \frac{f(MHz)}{150}, \quad (8)$$

$$ICNIRP \rightarrow P\left(\frac{W}{m^2}\right) = \frac{f(MHz)}{200}. \quad (9)$$

Para as frequências do celular das Bandas A e B do Brasil, a recomendação IEEE/ANSI implica em limites de densidade de potência de  $5,7 W/m^2$  ou para exposição local, uma SAR de  $1,6 mW/g$  enquanto que a recomendação ICNIRP adotada pela ANATEL define os seus limites de densidade de potência em  $4,3 W/m^2$  e  $SAR = 2 mW/g$ .

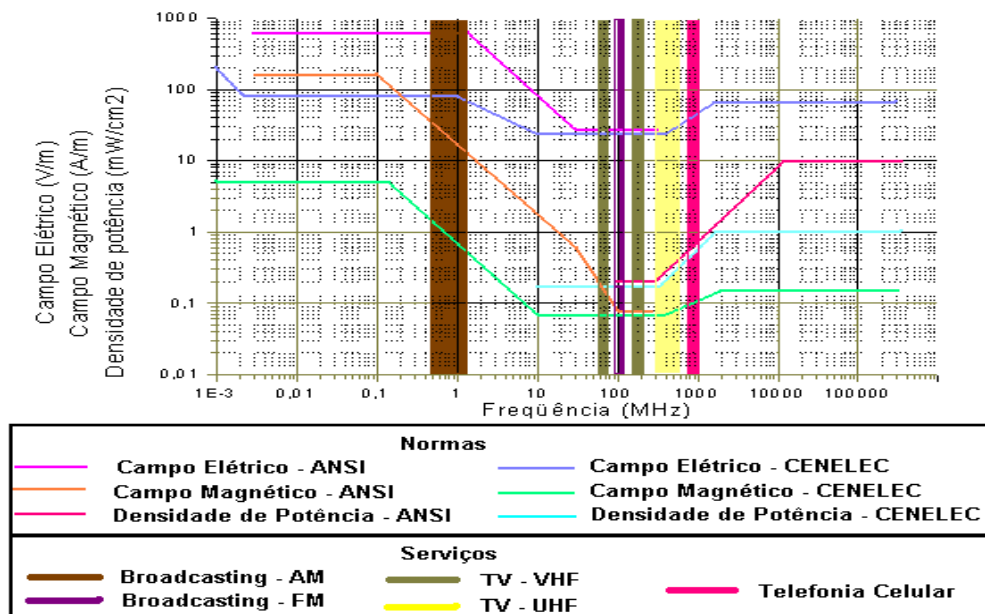


Figura 4. Limites de Exposição em Função da Frequência.

Embora os limites recomendados nessas normas sejam gerais, de validade em qualquer situação de irradiação, devido às dificuldades de avaliação em campo próximo até pouco tempo atrás, os organismos reguladores dos diferentes países (FCC no caso dos Estados Unidos) não avaliavam o atendimento as normas para transmissores portáteis, autorizando-os desde que não superassem a potência transmitida de  $740 \text{ mW}$ . Em função da relevância da questão comentada, tem havido um esforço para definir as metodologias de medida [26-28] de SAR e desde setembro de 2000, nos Estados Unidos é requerida uma declaração do fabricante indicando que os valores de SAR medidos para seus produtos atendem à norma C.95.1 IEEE/ANSI. Entretanto, um protocolo de medidas (especificando especialmente as distâncias entre a antena e a cabeça) ainda não foi oficialmente estabelecido.

### 2.4.3 Postura Precautória

O Princípio de Precaução é um critério de abordagem de riscos aplicado em circunstâncias com um alto grau de incerteza científica, refletindo a necessidade de tomar atitudes em face de riscos potencialmente sérios, sem esperar os resultados da pesquisa científica, [29]. Como ainda há incertezas a respeito dos efeitos não térmicos, alguns países, como Itália e Suíça invocando este princípio, adotaram recentemente limites mais restritivos [17, 19, 20]. O critério de precaução também foi recomendado para a exposição a radiofrequências e microondas pela Organização Mundial de Saúde – OMS [18, 29].

Baseados nesta atitude, o Decreto N° 381. italiano [20], e a norma suíça N° 814.710 [19] limitam os valores de densidade de campo em  $0,1 \text{ W/m}^2$  e  $0,043 \text{ W/m}^2$  respectivamente, valores dezenas de vezes menores que aqueles da recomendação ICNIRP.

## 2.5 CAMPO DISTANTE

O campo eletromagnético decresce com a distância à antena emissora como mostram as Expressões 10, 11 e 12, derivadas das equações de componentes de campo para um dipolo.

$$E_r \propto \frac{1}{r^2} + \frac{1}{jkr^3}, \quad (10)$$

$$H_\phi \propto \frac{1}{r} + \frac{1}{jkr^2}, \quad (11)$$

$$E_0 \propto \frac{1}{r} + \frac{1}{jkr^2} - \frac{1}{k^2 r^3}, \quad (12)$$

Desta forma, o aumento da distância à antena transmissora tem como resultado uma diminuição considerável dos campos. Então a contribuição aos valores dos campos, devida a antenas afastadas são pequenos, mesmo quando a antena emite elevadas potências. Inversamente, transmissores de baixa potência muito próximos do ponto onde se quer avaliar o campo podem ser responsáveis por valores elevados, mais se a distância for suficientemente pequena para ocorrer acoplamento eletromagnético ou se as componentes reativas (de valores imaginários) das equações passam a ser relevantes

### 2.5.1 Cálculos para uma Estação de Radiocomunicação

Para avaliar o cumprimento das normas de limitação de exposição a campos eletromagnéticos gerados por Estações de Radiocomunicação, dois tipos de cálculos são de interesse: a) o cálculo das “superfícies de contorno” que definem a região do espaço para a qual as normas são observadas e b) o cálculo do campo nos pontos em que serão realizadas as medidas. O cálculo das superfícies de contorno de atendimento à norma, para “campos distantes”, é baseado na expressão de Friis [30] para propagação no espaço livre, com linha de visada direta, resultando numa aproximação válida, desde que se confirme que as superfícies calculadas não incluem nenhum obstáculo. Em caso de realizar avaliações em pontos mais distantes, podem ser utilizados outros modelos que consideram as reflexões e difrações nos obstáculos presentes, como os de Lee-Picquenard, Korowajczuk, Okomura-Hata ou COST [31-36], que aproximam melhor a condição real do enlace entre a estação e o ponto de medida para as diferentes zonas: densamente urbanizadas, urbanizadas, semi-urbanizadas, rurais, florestais, por exemplo.

Do equacionamento derivado da expressão de Friis, aplicado a uma estação radiobase (ERB) de telefonia celular setorizada, a densidade de potência em um ponto  $(r, \theta, \phi)$ , situado na região distante da antena pode ser estimada por meio de

$$U(r, \theta, \phi) = \frac{\sum_{m=1}^3 P_m D_m(\theta - \theta_m, \phi - \phi_m) N_m}{4\pi r^2}, \quad (13)$$

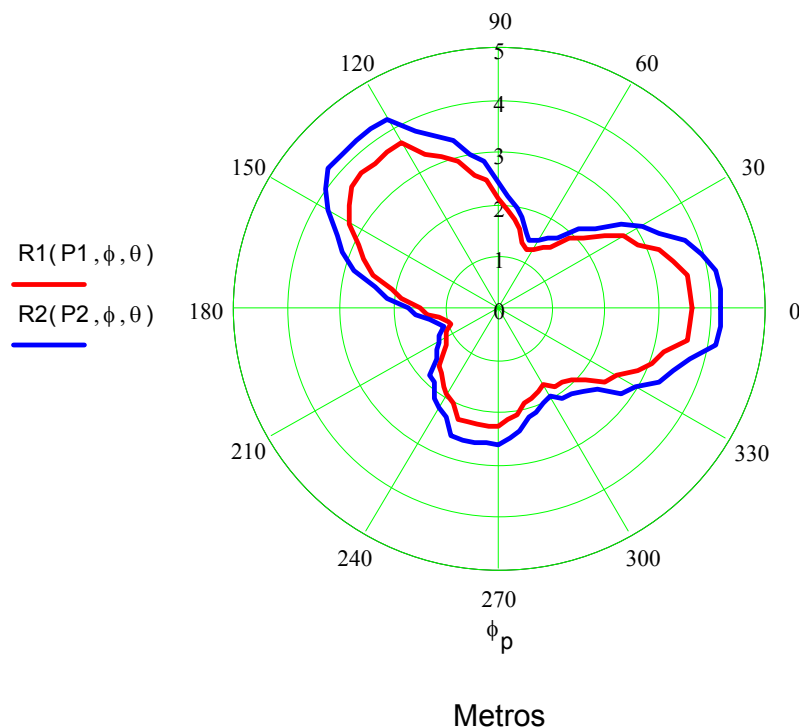
em que,  $P_m$  representa a potência por canal efetivamente fornecida à antena,  $D_m$  a função diretividade da antena utilizada e  $N_m$  o número de canais, para cada setor. Na fórmula,  $\theta_m$  e  $\phi_m$  indicam a orientação da antena de cada setor.

Desde que a densidade de potência é uma função monotonicamente decrescente da distância a antena, a superfície de contorno a partir da qual é observado o limite máximo de densidade de potência,  $U_{max}$ , pode ser definida a partir de uma distância  $R(\theta, \phi)$  dada por

$$R(\theta, \phi) = \sqrt{\frac{\sum_{m=1}^3 P_m D_m(\theta - \theta_m, \phi - \phi_m) N_m}{4\pi U_{max}}} \quad (14)$$

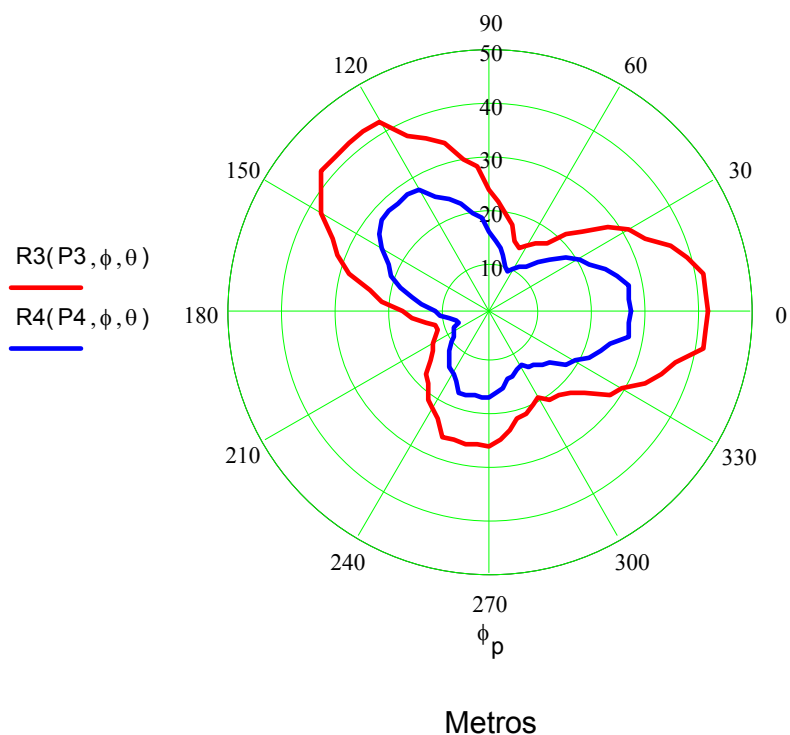
Com base neste equacionamento e tendo em vista que a faixa de transmissão utilizada no sentido da ERB para as estações móveis varia de 869 MHz a 894 MHz, tomando os valores de  $U_{max}$  para a frequência mais baixa, pode se calcular as superfícies de contorno onde ocorrem os valores limites de densidade de potência para a norma cujo cumprimento se quer verificar.

As Figuras 5 e 6 mostram os valores calculados para uma ERB típica, das distâncias mínimas para as quais os limites das normas ANSI e ICNIRP (Figura 5) e suíça e italiana (Figura 6) são atendidos.



**Figura 5** Distâncias calculadas para o atendimento às normas ICNIRP e ANSI para uma estação radiobase típica (aprox. 4 m) para um plano horizontal na altura das antenas.

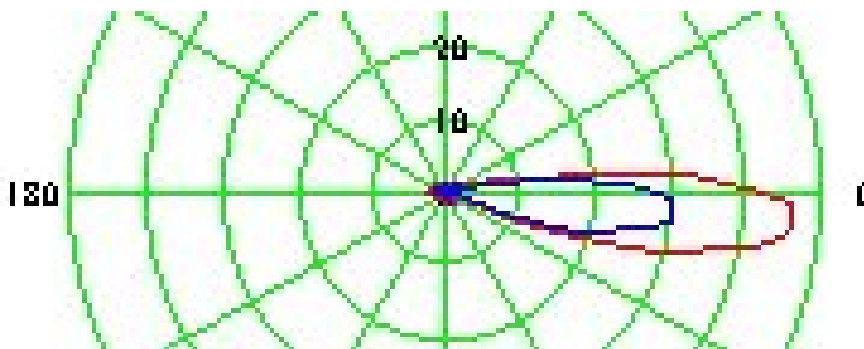
Para este exemplo de ERB típica, os limites das normas ANSI e ICNIRP são observados a partir de aproximadamente 5 m,



**Figura 6.** Distâncias calculadas para o atendimento às legislações Suíça e Italiana para uma estação radiobase típica (aprox. 30 e 40 m) para um plano horizontal na altura das antenas.

enquanto que os limites das normas suíça e italiana são observados a partir de aproximadamente 40 m.

As curvas das Figuras 5 e 6 correspondem a cortes de plano horizontal de superfícies que limitam a região do espaço para a qual os valores de densidade de campo irradiado superam as respectivas normas. No plano vertical, dada a diretividade típica das antenas usadas nas ERB que visa estender a cobertura, os limites são como na Figura 7.

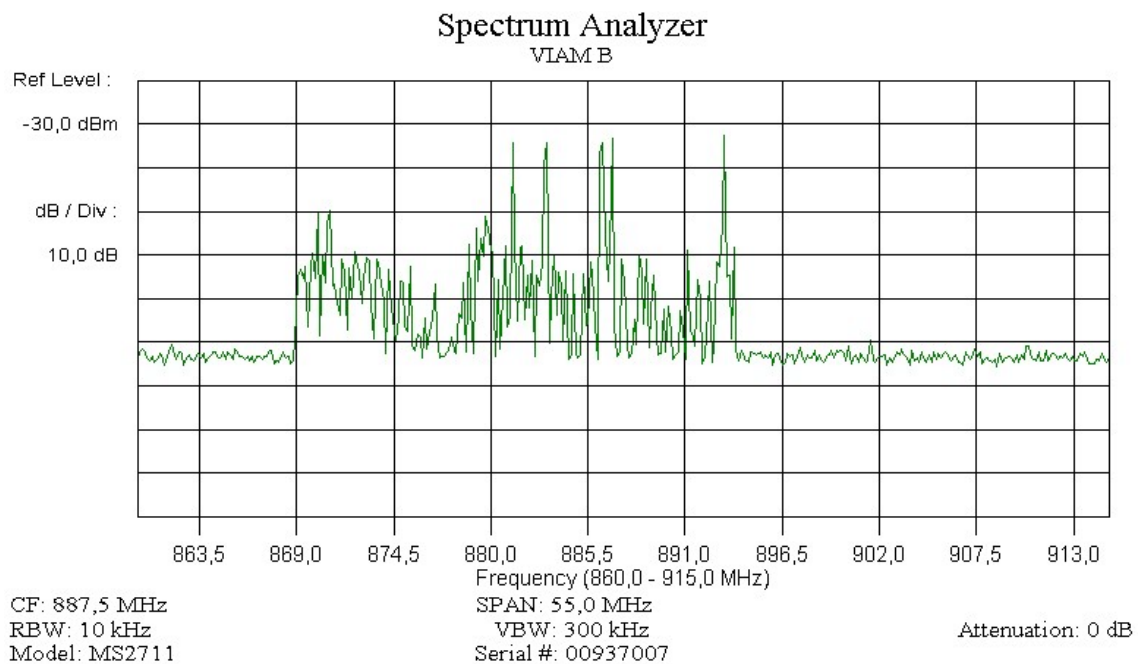


**Figura 7.** Distâncias calculadas para o atendimento às legislações Suíça e Italiana para uma estação radiobase típica para um plano vertical no eixo das antenas.

As superfícies de contorno das Figuras 5 a 7 se constituem num parâmetro de desempenho de leitura mais universal que os parâmetros dosimétricos citados nas recomendações.

## 2.5.2 Medidas

Por serem realizadas distantes da antena emissora, as medidas podem ser feitas com instrumentos como analisador de espectro ou medidor de campo de RF, com antenas adequadas. Instrumentos, técnicas de medida, e resultados comprovando os cálculos foram publicados em [37].



**Figura 8. Espectro eletromagnético observado nas proximidades de duas estações de radiobase (Bandas A e B).**

Os resultados das medidas de monitorização realizados comprovam as estimativas teóricas, na maioria dos casos. Porém, em alguns casos, falhas na execução do projeto ou elementos que não são considerados no projeto teórico para o atendimento às normas (como cabos de alimentação com perdas) podem ser responsáveis por níveis de radiação elevados, fato pelo qual não é prudente descartar a necessidade de medições.

### 3 SIMULAÇÃO NUMÉRICA

Prever o atendimento às recomendações existentes para o caso dos usuários de transceptores portáteis é um problema de campo próximo, no qual a cabeça do usuário está numa distância tal que interage com a antena modificando seu diagrama de irradiação. Diferente do problema em campo distante, o problema em campo próximo requer soluções numéricas. Os métodos mais usados para problemas semelhantes a este, de cálculo de propagação de ondas eletromagnéticas, são o Método dos Momentos (MoM), e o método das Diferenças Finitas no Domínio do Tempo (FDTD). Embora este último exija, geralmente, maiores recursos computacionais ele é mais adequado quando se trata de calcular a propagação da onda numa estrutura altamente não homogênea, como é um modelo apurado de corpo humano, e em particular, da cabeça.

#### Diagrama de Blocos do Solver FDTD

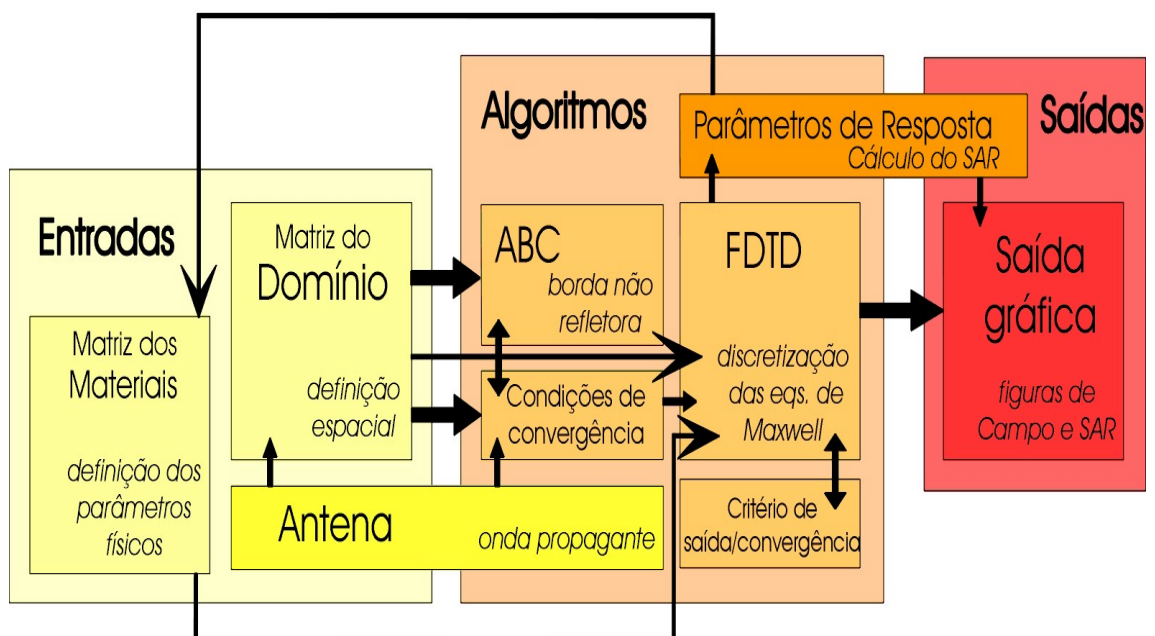


Figura 9. Diagrama de Blocos dos algoritmos implementados.



A série de algoritmos desenvolvida para encontrar a distribuição de campos vai além do algoritmo FDTD básico, incluindo rotinas de entrada de dados, para a definição do domínio do problema e da antena, rotinas de verificação e validação baseadas no monitoramento contínuo dos valores em pontos pré-escolhidos do domínio, e rotinas de saída, como rotinas de visualização gráfica de matrizes de grande porte.

### 3.1 MÉTODO FDTD

O método FDTD foi proposto inicialmente por Yee em 1967 [38] para resolver problemas de propagação e espalhamento, diretamente a partir das equações de Maxwell, de validade geral, também em campo próximo, diferentemente das equações de onda. Nas décadas seguintes à proposta original, diferentes autores têm contribuído com seus desenvolvimentos para aperfeiçoarem o método [39] o que, junto com o aumento da capacidade computacional disponível tem permitido a popularização do método nos últimos anos, existindo já pacotes de softwares comerciais baseados em FDTD [40].

Algoritmos FDTD foram utilizados em cálculos precisos de propagação da onda eletromagnética do telefone celular portátil na cabeça do usuário, visando estimar seus possíveis efeitos quanto à absorção de energia pela cabeça [41-45].

Embora os efeitos térmicos estejam na origem dos limites de exposição, esses são apresentados em valores de densidade de campo ou de SAR que podem ser relacionados com os valores de temperatura pela Equação 6. Para comparação com as limitações basta então calcular as componentes dos campos. Nas simulações apresentadas se calculam apenas os campos eletromagnéticos sem levar em conta a troca de calor do corpo ou a variação dos parâmetros dos tecidos em função da variação da temperatura.

Partindo de uma condição inicial de campo nulo na totalidade do domínio (ou de uma distribuição de campo previamente calculada ou estabelecida) é simulado um dipolo magnético, que será a condição de contorno do método numérico ou a fonte do campo eletromagnético e se faz com que este assuma as características semelhantes à antena de um aparelho celular, para a qual se tem frequência e potência pré-estabelecidas. Essas ondas se propagam a partir da fonte e, através de um número elevado de iterações, penetram na cabeça do usuário. Após a convergência numérica, o valor de campo calculado em todas as iterações é utilizado para o cálculo da SAR em cada célula do domínio. Esses dados calculados são

armazenados novamente em matrizes e transformados posteriormente em novas imagens, através das quais é possível visualizar as distribuições de intensidade de campo e da SAR. Nessas simulações, por simplificação, não foram incluídos no modelo os seguintes elementos: aparelho, mão e tronco do usuário. Esses elementos deverão ser considerados em simulações que queiram implementar um modelo apurado, pois a alteração nas características da antena, devido aos mesmos pode não ser desprezível, por se encontrarem no campo próximo, e, em particular, porque a barreira que o aparelho e a mão representam para as perdas de calor por convecção são certamente significativas para o caso de implementar equações de temperatura como a Equação 7.

### 3.1.1 Cálculo dos Campos Elétricos e Magnéticos

O método FDTD resolve as equações rotacionais de Maxwell (Equações 15 e 16), em forma de diferenças finitas, num domínio de cálculo previamente discretizado [46].

$$\frac{\partial \vec{H}}{\partial t} = -\frac{1}{\mu} \nabla \otimes \vec{E} - \frac{\rho}{\mu} \vec{H}, \quad (15)$$

$$\frac{\partial \vec{E}}{\partial t} = -\frac{1}{\varepsilon} \nabla \otimes \vec{H} - \frac{\sigma}{\varepsilon} \vec{E}. \quad (16)$$

A Equação 17 é a equação diferencial para a componente  $H_x$  do campo magnético, num sistema de coordenadas cartesianas, derivada da equação rotacional 15.

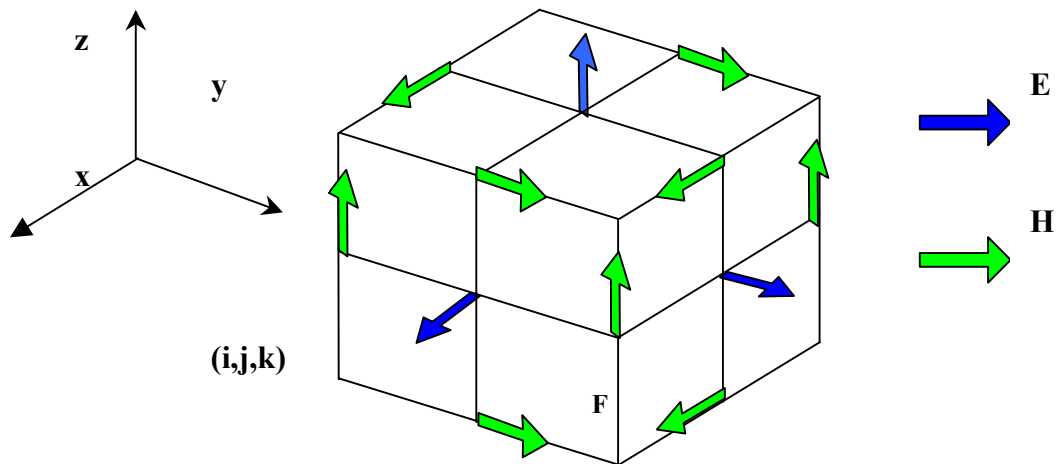
$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu} \left( \frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - \rho H_x \right). \quad (17)$$

Na proposta original de Yee, as equações de Maxwell em forma diferencial (como a Equação 17) são substituídas por equações de diferenças centrais de primeira ordem, tanto no tempo quanto no espaço. Desta forma as componentes usadas nas equações de diferenças estarão localizadas em pontos do domínio ou em intervalos de tempo vizinhos ao ponto que se está calculando. Por exemplo, a Equação 17, da componente  $H_x$  é substituída pela Equação 18.

$$\frac{H_x^{n+1/2}{}_{i,j,k} - H_x^{n-1/2}{}_{i,j,k}}{\Delta t} = \frac{1}{\mu_{i,j,k}} \left( \frac{E_y^n{}_{i,j,k+1/2} - E_y^n{}_{i,j,k-1/2}}{\Delta z} - \frac{E_z^n{}_{i,j+1/2,k} - E_z^n{}_{i,j-1/2,k}}{\Delta y} - \rho_{i,j,k} H_x^n{}_{i,j,k} \right) \quad (18)$$

Nesta equação explícita, o valor da componente  $H_x$ , num ponto  $(i, j, k)$  do domínio e num instante ou iteração  $n+1/2$ , posterior, é calculado a partir do valor de  $H_x$  no mesmo ponto  $(i, j, k)$  num instante anterior  $n-1/2$ , e das componentes  $E_y$  e  $E_z$  do rotacional avaliadas num instante equidistante ( $n$ ) como melhor aproximação e nos pontos da vizinhança  $(i, j, k+1/2)$ , por exemplo.

A solução completa do campo eletromagnético requer a solução de todas as componentes de ambos os campos elétrico e magnético das equações de Maxwell semelhantes à Equação 18 (seis para um problema espacial sem simetrias). Em alguns problemas pode ser conveniente o cálculo das equações de Maxwell em coordenadas cilíndricas para aproveitar simetrias.



**figura 10. Célula de Yee. Componentes de campo  $E$  no ponto central da célula são calculados usando componentes de campo  $H$  interpoladas.**

Desta forma, para usar diferenças centrais de intervalos unitários, como se visualiza na Figura 10, nos pontos onde é calculado o campo elétrico não se calcula o campo magnético, e vice-versa. O domínio se torna um par de malhas entrelaçadas onde os pontos de uma malha

nos quais se calcula um dos campos, elétrico ou magnético, são os centros dos segmentos das células da outra malha onde se calcula o outro campo. O algoritmo originalmente proposto alterna entre estas duas malhas e é descrito como *leap-frog* (“pulo do sapo”) já que o passo de discretização para um campo pula os pontos de discretização do outro, localizados em pontos com índices  $\frac{1}{2}$  se o passo for unitário (Equação 18). Os instantes onde um e outro campo são avaliados também são alternados e desencontrados.

Este tipo de algoritmo *leap-frog* originalmente proposto duplica o tamanho do domínio necessário. Para um problema no qual a malha representa a maior parte da memória usada para executar a simulação, isto implica em praticamente duplicar os recursos computacionais necessários. Algumas alternativas, como avaliar as diferenças em intervalos não unitários, podem ser testadas para evitar esta duplicação da malha, desde que se tenha cuidado em manter a convergência do método. O algoritmo implementado usa as Equações 19 a 24, que diferem das equações originalmente propostas, pelo fato de avaliar os rotacionais em pontos de discretização de índices inteiros e não em pontos de interpolação, mantendo o intervalo unitário. Aproximações equivalentes para problemas de duas dimensões se encontram em [47]. Por exemplo, os valores de campo elétrico (nas Equações 19 a 21) que deviam ser avaliados nesses pontos intermediários (como era feito na Equação 18) são aproximados aqui por valores pertencentes à malha do campo magnético, não mais com subíndices  $k+\frac{1}{2}$ ,  $k-\frac{1}{2}$  ou  $j+\frac{1}{2}$ ,  $j-\frac{1}{2}$ , mas  $k+1$ ,  $k$  ou  $j+1$ ,  $j$ , evitando a duplicação da malha. Esta solução tem se mostrado satisfatória do ponto de vista da convergência e validação dos resultados

$$\mathbf{Hx} \left| \begin{matrix} n+1 \\ i, j, k \end{matrix} \right. = \left( \begin{array}{c} 1 - \frac{\rho'_{i, j, k} \Delta t}{2 \mu_{i, j, h}} \\ \frac{\rho'_{i, j, k} \Delta t}{2 \mu_{i, j, h}} \end{array} \right) \mathbf{Hx} \left| \begin{matrix} n-1 \\ i, j, k \end{matrix} \right. + \left( \begin{array}{c} \frac{\Delta t}{\mu_{i, j, k}} \\ 1 + \frac{\rho'_{i, j, k} \Delta t}{2 \mu_{i, j, k}} \end{array} \right) \cdot \left( \begin{array}{c} \mathbf{Ey} \left| \begin{matrix} n \\ i, j, k+1 \end{matrix} \right. - \mathbf{Ey} \left| \begin{matrix} n \\ i, j, k \end{matrix} \right. \\ \mathbf{Ez} \left| \begin{matrix} n \\ i, j+1, k \end{matrix} \right. - \mathbf{Ez} \left| \begin{matrix} n \\ i, j, k \end{matrix} \right. \end{array} \right) \cdot \left( \begin{array}{c} \Delta z \\ \Delta y \end{array} \right) \quad (19)$$

$$\mathbf{Hy} \Big|_{i,j,k}^{n+1} = \left( \frac{1 - \frac{\rho'_{i,j,k} \Delta t}{2\mu_{i,j,h}}}{1 + \frac{\rho'_{i,j,k} \Delta t}{2\mu_{i,j,h}}} \right) \mathbf{Hy} \Big|_{i,j,k}^{n-1} + \left( \frac{\frac{\Delta t}{\mu_{i,j,k}}}{1 + \frac{\rho'_{i,j,k} \Delta t}{2\mu_{i,j,k}}} \right) \cdot \left( \frac{\mathbf{Ez} \Big|_{i+1,j,k}^n - \mathbf{Ez} \Big|_{i,j,k}^n}{\Delta x} - \frac{\mathbf{Ex} \Big|_{i,j,k+1}^n - \mathbf{Ex} \Big|_{i,j,k}^n}{\Delta z} \right) \quad (20)$$

$$\mathbf{Hz} \Big|_{i,j,k}^{n+1} = \left( \frac{1 - \frac{\rho'_{i,j,k} \Delta t}{2\mu_{i,j,h}}}{1 + \frac{\rho'_{i,j,k} \Delta t}{2\mu_{i,j,h}}} \right) \mathbf{Hz} \Big|_{i,j,k}^{n-1} + \left( \frac{\frac{\Delta t}{\mu_{i,j,k}}}{1 + \frac{\rho'_{i,j,k} \Delta t}{2\mu_{i,j,k}}} \right) \cdot \left( \frac{\mathbf{Ex} \Big|_{i,j+1,k}^n - \mathbf{Ex} \Big|_{i,j,k}^n}{\Delta x} - \frac{\mathbf{Ey} \Big|_{i+1,j,k}^n - \mathbf{Ey} \Big|_{i,j,k}^n}{\Delta z} \right) \quad (21)$$

$$\mathbf{Ex} \Big|_{i,j,k}^{n+1} = \left( \frac{1 - \frac{\sigma_{i,j,k} \Delta t}{2\varepsilon_{i,j,h}}}{1 + \frac{\sigma_{i,j,k} \Delta t}{2\varepsilon_{i,j,h}}} \right) \mathbf{Ex} \Big|_{i,j,k}^n + \left( \frac{\frac{\Delta t}{\varepsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k} \Delta t}{2\varepsilon_{i,j,k}}} \right) \cdot \left( \frac{\mathbf{Hz} \Big|_{i,j,k}^{n+1} - \mathbf{Hz} \Big|_{i,j-1,k}^{n+1}}{\Delta y} - \frac{\mathbf{Hy} \Big|_{i,j,k}^{n+1} - \mathbf{Hy} \Big|_{i,j,k-1}^{n+1}}{\Delta z} \right) \quad (22)$$

$$\mathbf{E}_y \Big|_{i,j,k}^{n+1} = \left( \frac{1 - \frac{\sigma_{i,j,k} \Delta t}{2 \varepsilon_{i,j,h}}}{1 + \frac{\sigma_{i,j,k} \Delta t}{2 \varepsilon_{i,j,h}}} \right) \mathbf{E}_y \Big|_{i,j,k}^n + \left( \frac{\frac{\Delta t}{\varepsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k} \Delta t}{2 \varepsilon_{i,j,k}}} \right) \cdot \left( \frac{\mathbf{H}_x \Big|_{i,j,k}^{n+1} - \mathbf{H}_x \Big|_{i,j,k-1}^{n+1}}{\Delta z} - \frac{\mathbf{H}_z \Big|_{i,j,k}^{n+1} - \mathbf{H}_z \Big|_{i-1,j,k}^{n+1}}{\Delta x} \right) \quad (23)$$

$$\mathbf{E}_z \Big|_{i,j,k}^{n+1} = \left( \frac{1 - \frac{\sigma_{i,j,k} \Delta t}{2 \varepsilon_{i,j,h}}}{1 + \frac{\sigma_{i,j,k} \Delta t}{2 \varepsilon_{i,j,h}}} \right) \mathbf{E}_z \Big|_{i,j,k}^n + \left( \frac{\frac{\Delta t}{\varepsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k} \Delta t}{2 \varepsilon_{i,j,k}}} \right) \cdot \left( \frac{\mathbf{H}_y \Big|_{i,j,k}^{n+1} - \mathbf{H}_y \Big|_{i-1,j,k}^{n+1}}{\Delta x} - \frac{\mathbf{H}_x \Big|_{i,j,k}^{n+1} - \mathbf{H}_x \Big|_{i,j-1,k}^{n+1}}{\Delta y} \right) \quad (24)$$

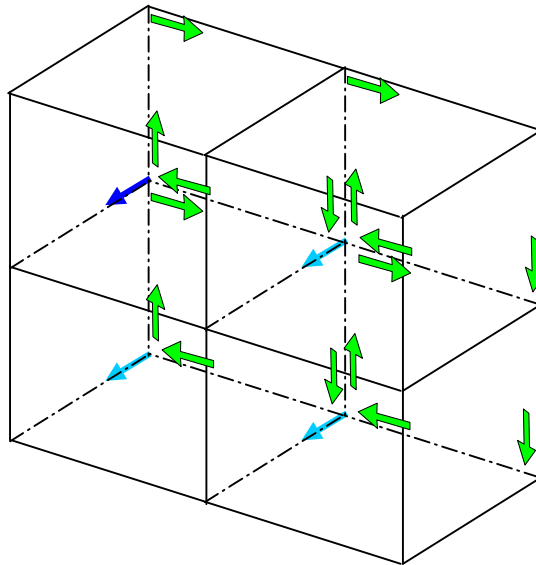


Figura 11. Componentes de campo usados na aproximação dos rotacionais das Equações 19 a 24 (modificação da célula de Yee).

O sucesso deste tipo de aproximação requer que se mantenha a consistência dos rotacionais das diferentes componentes, usando valores de uma dada componente num dado ponto sempre em duas equações diferentes, mas com sinais diferentes (em rotacionais calculados para duas células vizinhas), sob risco de o algoritmo divergir rapidamente. Outras aproximações, i.é., outras escolhas dos pontos para avaliar as componentes dos rotacionais, têm resultado em simulações que divergiram rapidamente.

Usando estas novas equações de diferenças (19 a 24), evita-se a duplicação da malha espacial. A duplicação no tempo é eliminada com uma perda da simetria das equações, em relação ao subíndice  $n$  das iterações. Isto, por sua vez, é contornado processando-se as equações na ordem que foram apresentadas. Este procedimento se justifica, pois, uma vez calculadas as componentes da iteração  $n+1$  do campo magnético, utilizando-se as Equações 19 a 21, poderão então ser calculadas as Equações 22 a 24 do campo elétrico, pois essas últimas equações são em função dos componentes calculados nas anteriores.

Com este conjunto de equações ordenadas pode-se resolver o valor de campo iterativamente, para qualquer instante, e depois de um tempo de transiente, obter valores de campo máximo ou médio, que serão usados no cálculo da SAR, de acordo com a Equação 5.

## 3.2 MODELO DO DOMÍNIO DO PROBLEMA

O domínio do problema é um modelo discreto de cabeça humana. Embora algumas normas e procedimentos recomendem o uso de modelos com morfologias padrão ou médias [48], que poderiam ser usados nessas simulações, foi desenvolvido um modelo próprio de cabeça humana, sobre uma malha ortogonal suficientemente fina para produzir um modelo detalhado. Uma base de dados médicos com imagens bidimensionais de diferentes cortes reais (sagittais e transversais), de Raios-X e Ressonância Magnética (Figura 12) com uma separação de 0,9 mm entre cortes sucessivos, servem de ponto de partida para determinar o domínio numérico onde o campo é calculado, utilizando o método FDTD.

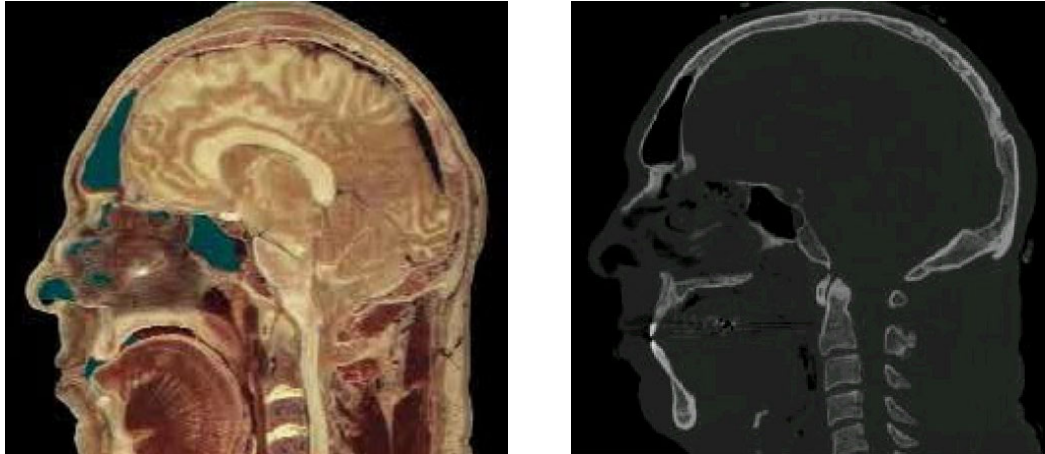


Figura 12. Cortes sagitais (real e de ressonância magnética).

A partir dessas imagens e com ajuda de ferramentas comerciais de processamento de imagens, foram geradas 208 imagens de cortes idealizados, as quais foram retocadas manualmente. Em cada imagem foram identificados e separados os diferentes tecidos que compõem o corte e que serão considerados homogêneos e invariantes nos cálculos: pele/gordura, músculos, ossos, cérebro, olhos e o próprio ar, usando para isto cores primárias para maior facilidade em posterior reconhecimento destas pelo algoritmo (Figura 13). Cada imagem foi programada para ser composta de  $225 \times 256$  pontos ou pixels, de forma que a distância entre pixels corresponda aos mesmos  $0,9 \text{ mm}$  da distância entre cortes para gerar uma malha homogênea de 57.600 células por corte, totalizando, aproximadamente, 12 milhões de pontos.

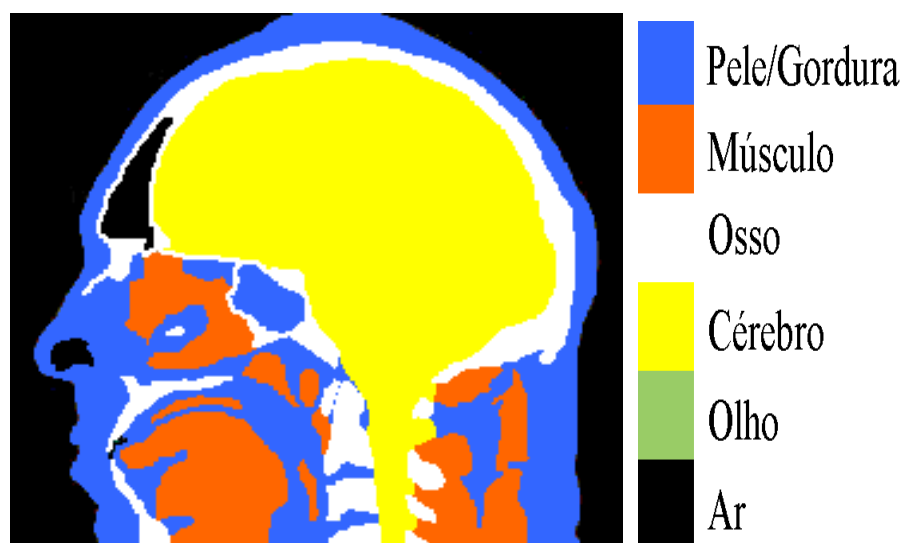
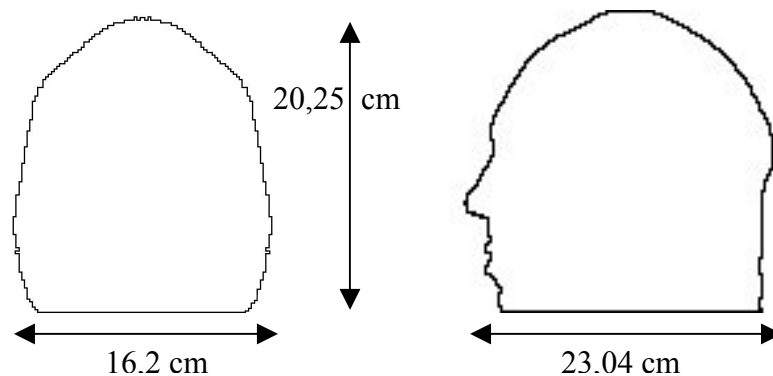


Figura 13. Imagem de cortes sagital idealizado, usada para a geração do domínio de cálculo.



As dimensões do modelo utilizado (mostradas na Figura 14) são compatíveis com as dos modelos recomendados [48] e com as apresentadas em trabalhos de outros autores [40-43].



**Figura 14. Dimensões do modelo utilizado.**

Após a determinação do espaço físico do problema, através de um conjunto de imagens bidimensionais, o programa deve criar uma matriz que será o domínio de cálculo. Cada célula desta matriz estará relacionada a uma localização no espaço e, portanto, a um tecido específico, referente à sua posição.

As imagens são convertidas em arquivos *bitmap*, com a dimensão apropriada, e é nesses arquivos que se usa uma rotina de reconhecimento de padrões para identificar, em cada ponto, o tecido correspondente. No caso dos arquivos utilizados, este reconhecimento é praticamente direto já que o arquivo de cada imagem, convertido em texto e uma vez eliminado seu cabeçalho, reproduz os índices das componentes de cor no padrão RGB. Indexando cada tecido com um valor (não necessariamente igual ao valor da componente RGB usada para a identificação), pode-se extrair uma matriz numérica bidimensional (Figura 15) de cada imagem de corte idealizado, de modo que durante o processamento da simulação, obtém-se através da soma das 208 matrizes correspondentes aos 208 cortes sagitais, uma matriz tridimensional (3D), representando o domínio do problema com a composição de 11.980.800 células.

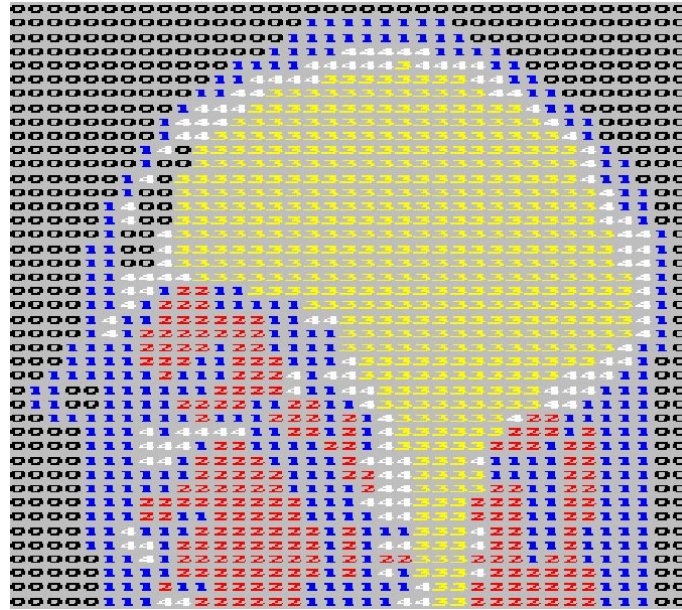


Figura 15. Plano da matriz gerada. Cada índice corresponde a um material.

Os diferentes meios que constituem o modelo da cabeça foram definidos de acordo com os parâmetros eletromagnéticos médios dos diferentes tecidos presentes [40, 43, 49, 50]. Assim tecidos que apresentam valores semelhantes foram considerados, por simplicidade, como um único meio. Desde que essas características são dependentes da frequência da radiação à qual são submetidos, os parâmetros correspondem à frequência simulada (850 MHz). Embora haja alteração desses parâmetros com a temperatura, para evitar cálculos de pouca relevância, os valores utilizados são mantidos constantes sem que haja uma correção ou realimentação a partir do cálculo dos valores de SAR. O objeto destas simulações é o cálculo da SAR para comparação com as recomendações e outros resultados publicados, não sendo incluído um modelo termodinâmico para, a partir da SAR calcular a temperatura. Não obstante, o aumento de temperatura previsto é da ordem de décimos de grau [51] o que não justifica alterações significativas nos valores dos parâmetros usados. A Tabela 1 expressa os principais parâmetros utilizados nas simulações (permissividade relativa, condutividade e massa específica).

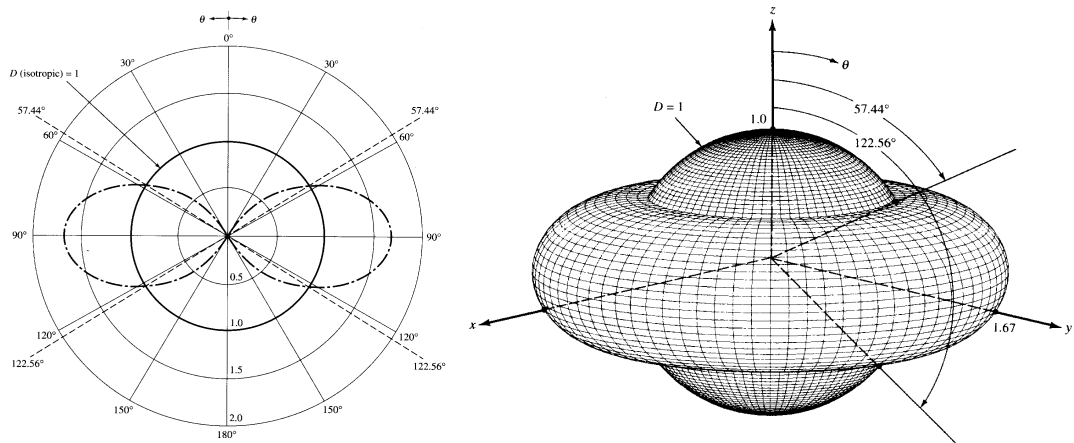
Tabela 1. Parâmetros dos tecidos usados na simulação.

Tecido	$\epsilon_r$	$\sigma$ [S/m]	$\rho$ [kg/l]
Pele/Gordura	34.5	0.600	1.10
Músculo	58.5	1.210	1.04
Osso	8.0	1.105	1.85
Cérebro	55.0	1.230	1.03
Olho	67.9	1.680	1.02

Analisando os valores da tabela acima se pode prever que os valores elevados para a constante dielétrica relativa provocarão uma concentração de densidade de fluxo elétrico nos tecidos correspondentes, o que, associado aos altos valores de condutividade, irá resultar em SAR elevada de acordo com a Equação 5.

### 3.3 MODELO DA FONTE

Muitos aparelhos celulares utilizados atualmente são dotados de antenas tipo monopolo, dipolo ou helicoidal, que apresentam diagramas de irradiação simétricos no plano horizontal, quando afastados de qualquer obstáculo. A Figura 16 mostra o diagrama de irradiação de um monopolo vertical, comparado com uma fonte isotrópica (bi e tri-dimensional). Quando da presença de um corpo no campo próximo da antena, esses diagramas ficam substancialmente alterados.

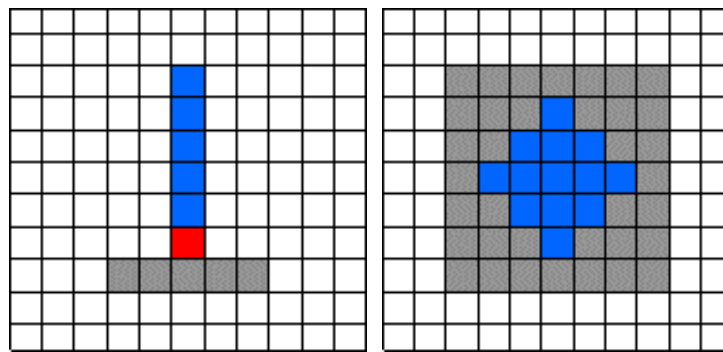


**Figura 16. Diagramas de irradiação de um dipolo e de uma fonte isotrópica.**

Existem diferentes maneiras de modelar a fonte irradiadora. Como primeira aproximação, uma vez definidos os pontos do espaço onde estará localizada a antena, é possível forçar neles o valor do campo elétrico, ou o campo magnético nas células da vizinhança de um plano normal à direção da antena, ajustado aos valores de corrente na antena, ou ainda ambos campos  $E$  e  $H$  simultaneamente. Neste caso, os pontos do domínio que representam a antena formam parte de uma coluna da matriz, e aparecerão no algoritmo como uma fonte de sinal ou “porta de entrada” eletromagnética.

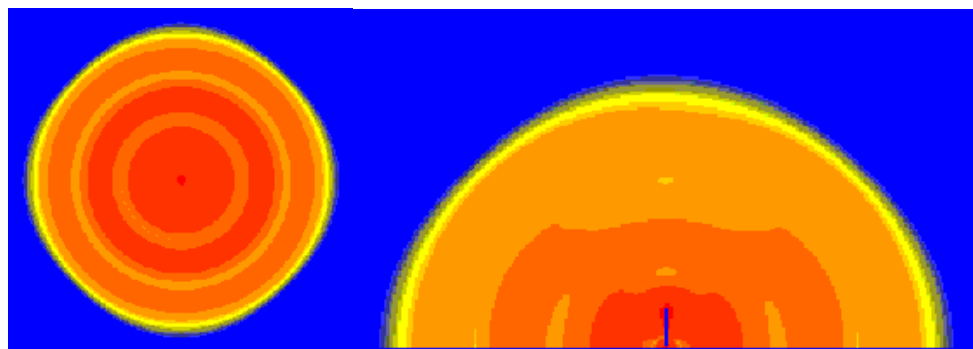
Uma aproximação mais precisa consiste em descrever o material da antena (metal) como condutor com um valor de condutividade elevado ou um condutor ideal (sem perdas) assumindo a condutividade do metal tendendo ao infinito ( $\sigma \rightarrow \infty$ ). Com essa aproximação, as

Equações 22 a 24, do cálculo de campo elétrico, vão sofrer simplificações, agilizando o processamento. Uma possível alternativa é manter as equações gerais, utilizando um valor de condutividade apropriado. A antena utilizada nas simulações é um modelo de monopolo como apresentado na Figura 17. O campo é forçado apenas na entrada de alimentação da antena ou em um ponto mais distante conectado à entrada da antena por células que estão totalmente circundadas por metal, simulando assim a alimentação por cabo coaxial. Se cada célula tem  $0,9\text{ mm}$  e o comprimento da antena for  $\lambda/4$  para aproximadamente  $850\text{ MHz}$ , ou seja,  $8,82\text{ cm}$ , ou 98 células, observa-se que o número de células destas figuras não guarda proporção com as utilizadas.



**Figura 17. Esquema das diferentes vistas (lateral e superior) da simulação de um monopolo com plano terra.**

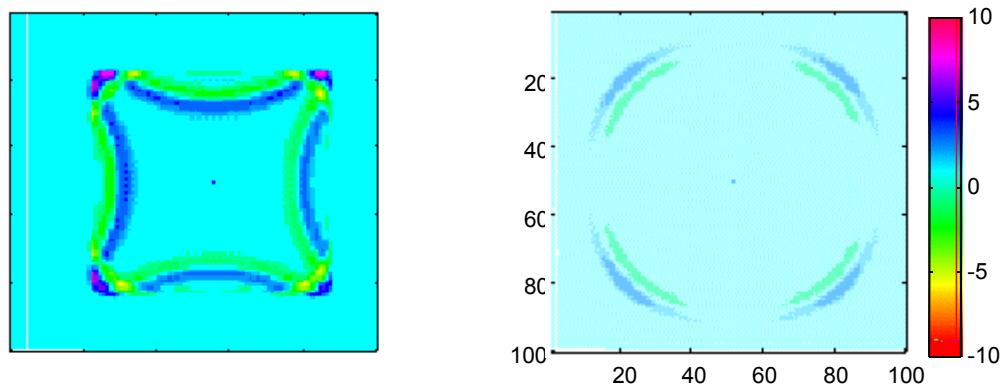
Para a validação do modelamento da fonte, foram realizadas diversas simulações sem o obstáculo representado pela cabeça (i.é. com a antena num domínio contendo somente ar). Após um número de iterações suficiente para a convergência dos valores de campo, observa-se, a partir dos valores na matriz de  $|E|$ , a isotropia na propagação, o comprimento de onda e a velocidade de propagação, como ilustrado na Figura 18.



**Figura 18. Campo Elétrico observado (vista superior e lateral para uma simulação teste de um monopolo).**

### 3.4 SIMULAÇÃO DO ESPAÇO LIVRE

Um problema que os algoritmos FDTD devem considerar é a simulação do espaço livre. O domínio computacional é necessariamente finito, então o domínio de simulação deve ser truncado. A primeira aproximação para uma condição de contorno na borda do domínio seria a de supor os campos suficientemente atenuados para considerarmos o seu valor nulo. Porém, imediatamente podem ser observadas as semelhanças entre esta solução e a modelagem de uma superfície metálica, perfeitamente condutora e, portanto, refletora. Este tipo de truncamento provoca então, reflexões da onda propagante nas bordas do domínio [46].



**Figura 19. Vista superior de um pulso propagando (campo elétrico) para uma condição de borda descasada (esquerda) e casada (direita). A borda absorvedora tem 18 células de largura.**

Entre as diferentes soluções propostas para uma eficiente simulação do espaço livre pode-se citar o cálculo do valor da condição de contorno usando equações de campo distante e o uso de equações e algoritmos específicos para as regiões limites do domínio, por exemplo, aproximando as equações diferenciais não mais por diferenças centrais e sim por diferenças laterais nos pontos do limite do domínio (para que não seja necessário estimar valores em pontos de “fora do domínio”) [39, 46]. Entre essas foram escolhidas, após alguns ensaios com outros métodos, duas soluções de condições de bordas absorvedoras (ABC). A primeira solução calculada a partir da proposta de Mur [46] considera aproximações da direção de propagação nos diferentes limites do domínio e, usando equações diferenciais de segunda ordem, encontra novas equações para a região de borda. A Equação 25 exemplifica a forma de cálculo já em diferenças finitas, de acordo com o equacionamento de Mur, de uma componente de campo  $W$  (válida para todas as componentes de campo  $E$  e campo  $H$ ) para o plano de  $x = 0$  de uma malha homogênea na qual  $\Delta x = \Delta y = \Delta z = \Delta$ .

$$\begin{aligned}
W_{0,j,k}^{n+1} = & -W_{1,j,k}^{n-1} + \frac{c\Delta t - \Delta}{c\Delta t + \Delta} \left( W_{1,j,k}^{n+1} + W_{0,j,k}^{n-1} \right) + \\
& \frac{2\Delta}{c\Delta t + \Delta} \left( W_{1,j,k}^n + W_{0,j,k}^n \right) + \\
& \frac{(c\Delta t)^2}{2\Delta(c\Delta t + \Delta)} \left( \begin{array}{l} W_{0,j+1,k}^n - 4W_{0,j,k}^n + W_{0,j-1,k}^n + \\ W_{1,j+1,k}^n - 4W_{1,j,k}^n + W_{1,j-1,k}^n + \\ W_{0,j,k+1}^n + W_{0,j,k-1}^n + W_{1,j,k+1}^n + W_{0,j,k-1}^n \end{array} \right).
\end{aligned} \tag{25}$$

Diferentes algoritmos de cancelamento de erros e variações da ABC foram propostos para melhorar a absorção da borda [46, 52]. A segunda solução utiliza um material de borda com características absorvedoras (“ABC”), e com a condição geradora da Perfect Matching Layer (“PML”) [46, 53, 54]. A PML é uma proposta recente [53, 54] que promete minimizar as reflexões das ondas incidentes na borda do domínio usando materiais de borda complexos e fenômenos como condutividade magnética, definidos a partir do modelo matemático e sem equivalentes reais. O termo  $\sigma^*$  é o termo dual da condutividade, para o campo  $H$ , nas equações de Maxwell, e é associado a cargas ou monopolos magnéticos não observados na realidade. A identidade da Equação 26 é o ponto de partida do desenvolvimento de uma PML [46].

$$\frac{\sigma}{\epsilon_0} = \frac{\sigma^*}{\mu_0} \tag{26}$$

Foi implementada, então, uma camada absorvedora de uma profundidade de 30 células do domínio com diferentes materiais com características anisotrópicas de condutividade. Com esta solução são obtidas atenuações superiores a 30 *dB* na reflexão e já foram relatadas atenuações de mais de 70 *dB* por outros autores [53, 54].

Ambos os tipos de borda são sensíveis ao ângulo de incidência da onda propagante. Desta forma as bordas são otimizadas para diferentes direções e têm melhor desempenho quando as ondas incidem nelas ortogonalmente. Em problemas de geometrias mais simples é fácil prever a direção de propagação para, a partir dela, ajustar a borda. Já no caso de um corpo ou cabeça

humana o espalhamento da onda dificulta este ajuste e prejudica o desempenho da borda, sem que isto comprometa a validade dos resultados calculados.

### 3.5 CONVERGÊNCIA DO MÉTODO

Métodos explícitos como o FDTD tendem a propagar rapidamente os erros [23] e divergir após um número suficiente de iterações, pelo que se torna particularmente importante diminuir as componentes de erro, representadas pelo fator  $O(\Delta x)^2$ , na equação abaixo [46].

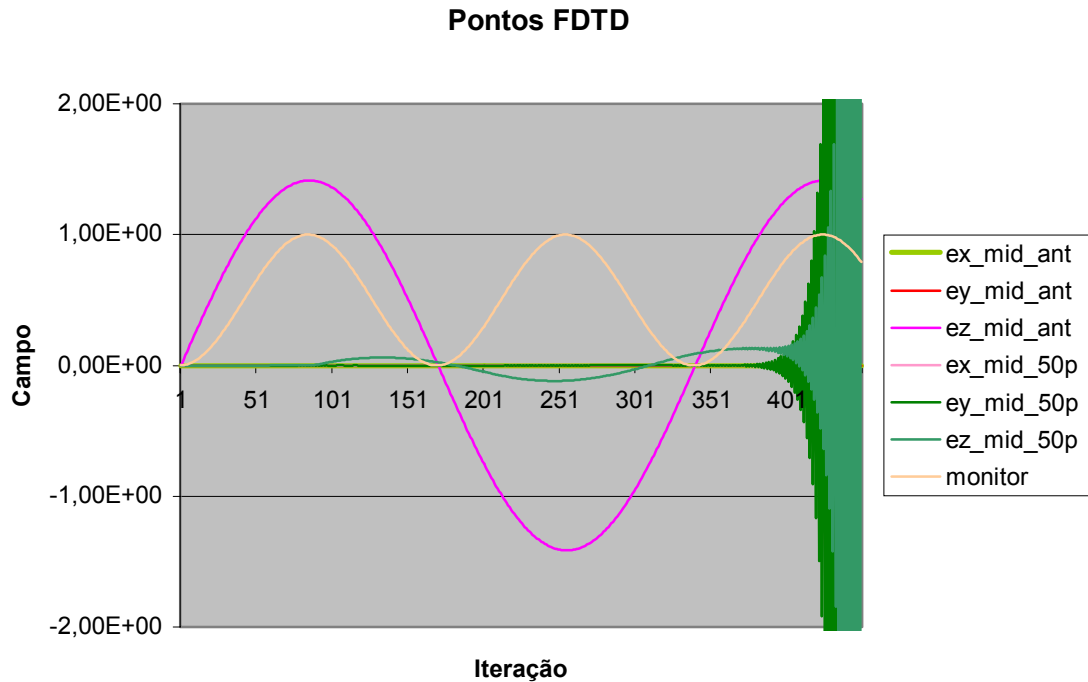
$$\frac{\partial u(i\Delta x, j\Delta y, k\Delta z, n\Delta t)}{\Delta x} = \frac{u^n_{i+\frac{1}{2},j,k} - u^n_{i-\frac{1}{2},j,k}}{\Delta t} + O[(\Delta x)^2]. \quad (27)$$

Os critérios de convergência do algoritmo desenvolvido são os mesmos de um algoritmo FDTD genérico [46]. Por exemplo, para o caso de uma malha de discretização homogênea com  $\Delta x = \Delta y = \Delta z = 0,9 \text{ mm}$  e para uma onda propagante de frequência igual a 850MHz, o intervalo de tempo entre iterações deve ser menor ou igual a  $\Delta t = 3,4665 \times 10^{-12}$  segundos de acordo com a Inequação 28.

$$\Delta t \leq \frac{1}{c \sqrt{\left(\frac{1}{\Delta x}\right)^2 + \left(\frac{1}{\Delta y}\right)^2 + \left(\frac{1}{\Delta z}\right)^2}}. \quad (28)$$

Diferente de outros métodos numéricos nos quais o erro diminui junto com o passo entre iterações  $\Delta t$ , para algoritmos FDTD em malha homogênea ( $\Delta x = \Delta y = \Delta z$ ) o máximo valor de  $\Delta t$  que permite a convergência, cancela o erro de discretização. Usando esses valores, se demonstra que o resultado da solução numérica (nos pontos nos quais é calculada) é exato, idêntico ao da solução algébrica, se esta existir, sendo cancelados os erros de quantização, tendo apenas o erro derivado do truncamento computacional. Por estas propriedades, este intervalo de tempo é chamado de *Magic Step* [46].

Na simulação implementada, desde que o método sofreu algumas adaptações e não se pode mais garantir o cancelamento dos erros de discretização, é calculado um valor de  $\Delta t$  de acordo com a Inequação 25, tomando cuidado para garantir que os erros de cálculo do truncamento não façam com que o valor do tempo entre iterações supere o *Magic Step* o que levaria a divergência da simulação. Observando os valores das diferentes componentes de campo calculadas, a cada iteração, para um conjunto de pontos distribuídos no domínio, pode-se observar o intervalo de convergência do método como na Figura 20.



**Figura 20. Monitorização ao longo das iterações dos valores de diferentes componentes de campo elétrico na qual se observa o intervalo de convergência (400 iterações neste caso).**

### 3.6 Validação do Algoritmo Implementado

Parte fundamental das simulações numéricas reside na confiabilidade dos resultados obtidos. Mesmo observadas as condições matemáticas de estabilidade e convergência e avaliados os erros de truncamento, é necessário o teste do algoritmo com problemas para os quais se conhece a solução. No caso presente, embora não se tenha testado o programa com uma bateria completa de problemas *benchmark*, a sua resposta foi comprovada para a propagação de pulsos e ondas senoidais em meios homogêneos, observando os fenômenos mais conhecidos de mudança de comprimento de onda na interface entre dois meios, reflexões e difração em fendas.

Para a simulação da propagação de uma onda no ar foi medida uma velocidade de propagação de 303.000.000 *m/s*. Este valor tem um erro de 1% que é inferior ao que pode ser justificado pela discretização do domínio. Também foram comprovados os diagramas de irradiação de antenas conhecidas (dipolos) em simulações realizadas em domínios suficientemente grandes para chegar a uma situação de campo distante.



### 3.7 O ALGORITMO IMPLEMENTADO

Na primeira etapa de desenvolvimento, os algoritmos utilizados para uma simulação em duas dimensões foram implementados em Matlab<sup>®</sup> para aproveitar as bibliotecas disponíveis como, por exemplo, as rotinas gráficas de saída. Logo após destas primeiras simulações, ficou evidente que uma plataforma PC não seria suficiente para poder processar uma simulação em três dimensões com o detalhamento desejado. Os algoritmos FDTD geralmente requerem massivos recursos computacionais, e, se considerarmos que o domínio de cálculo, incluindo as bordas possui mais de 24 milhões de pontos ( $268 \times 285 \times 316$ ), nos quais são calculadas as diferentes componentes dos diferentes campos, elétrico e magnético, não seria viável processar simulações com essa grandeza em computadores PCs usuais. Assim, como é necessária uma grande capacidade de memória (1,6 GB em algumas simulações) e de processamento, para realizar estas simulações, foi utilizado um computador de processamento paralelo CRAY T94 [55], com 5,2 GFLOP CPU, 2 GB RAM e unidade de disco de 9.5 GB SCSI, possibilitando assim esses trabalhos. O algoritmo foi implementado em C++, otimizado para o processamento paralelo, pois esta é uma das linguagens de programação com compilador disponível no supercomputador usado.

## 4 RESULTADOS OBTIDOS

Para efeitos de comparação com as normas, são calculados os valores de SAR correspondentes aos valores de campo obtidos. As recomendações requerem a medição de SAR ao longo de períodos de alguns minutos (e.g., 6 minutos para a norma ANSI C.95.1). Porém uma simulação com intervalos de tempo como este teria que realizar um elevado número de iterações (da ordem de  $10^{14}$  iterações), dado o valor correspondente de  $\Delta t$ , o que representa um valor inviável. Mesmo que se dispusesse de recursos computacionais para realizar tal quantidade de simulações a propagação de erros inevitáveis como os de truncamento levaria à divergência dos valores calculados.

Os resultados apresentados são estimativas da SAR obtidas a partir da Equação 5, para diferentes distância ou posições relativas entre a antena e o usuário e para diferentes tempos de exposição. Para estas estimativas se usa uma média de valores do campo  $E$  calculada com um número suficiente de iterações para ser representativo (com o cuidado de descartar os transientes). A Figura 21 abaixo mostra a evolução ao longo do tempo (ou seja, das iterações) do valor da SAR calculado desta forma para um ponto (ou célula) do interior da cabeça.

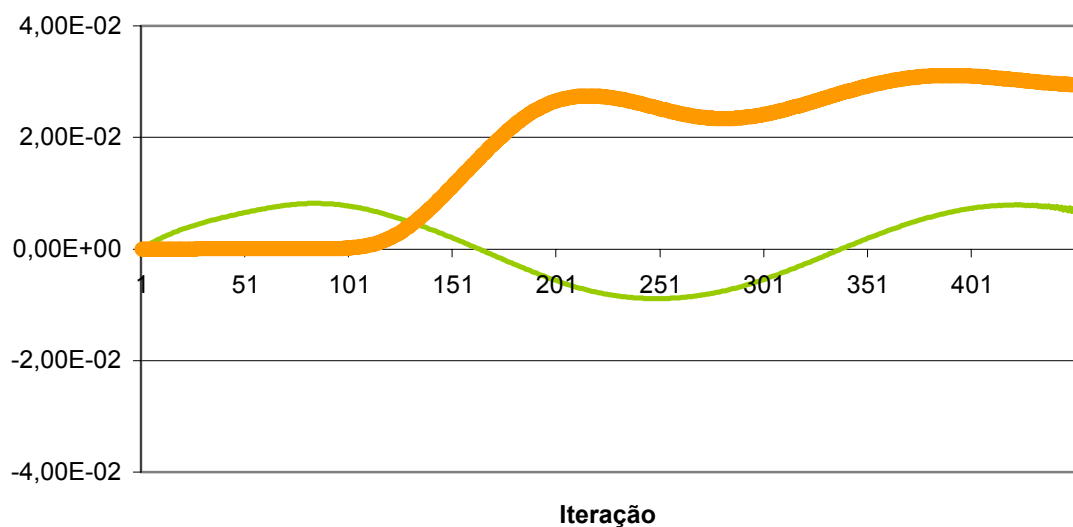
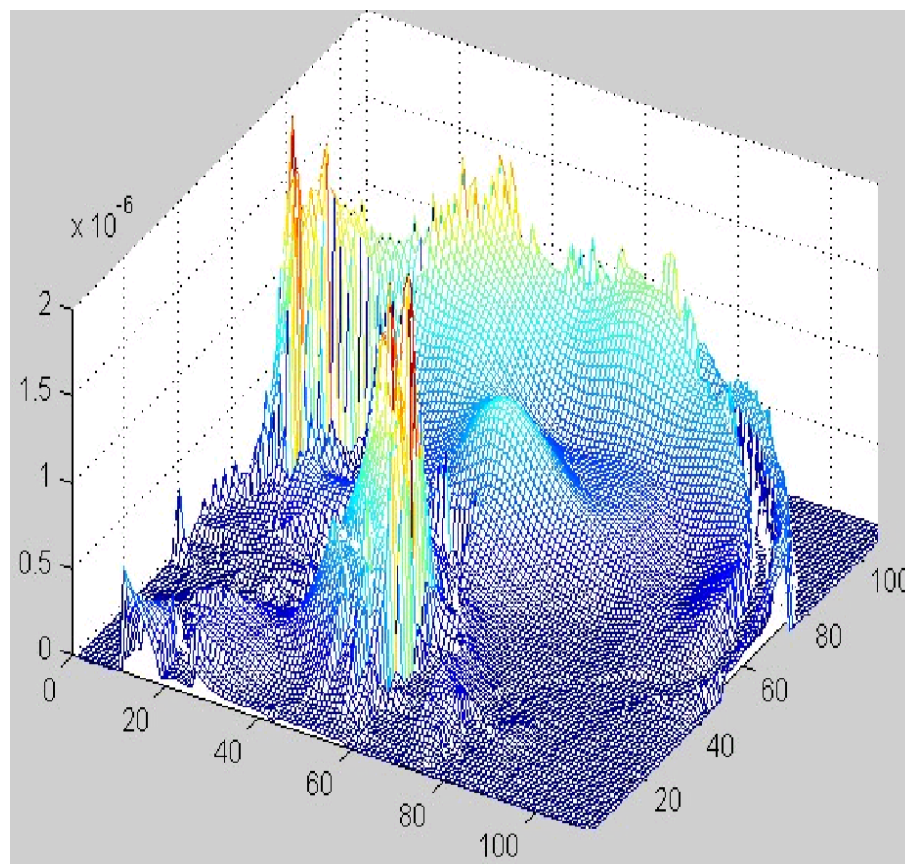


Figura 21. Monitorização ao longo das iterações do valor de campo elétrico na entrada da antena (curva fina) e do valor de SAR calculado (curva espessa).

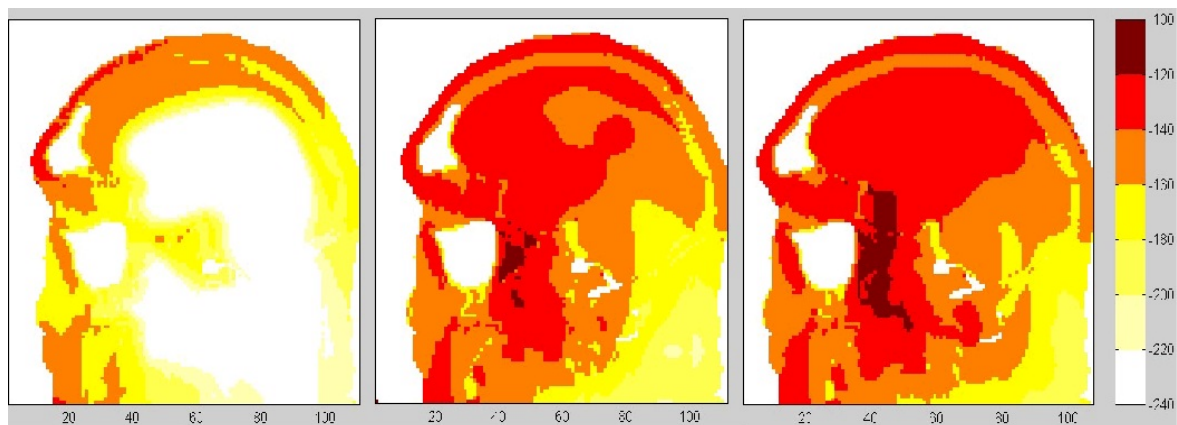
No exemplo da Figura 21, o valor de SAR converge de forma oscilatória. Esse comportamento oscilatório da SAR está de acordo com o modelo implementado. Não se deve a oscilações introduzidas pelo método numérico, mas é esperado teoricamente. A curva (SAR  $\times$  tempo) pode ser aproximada, então, por uma exponencial (com coeficientes complexos). Este tipo de aproximação pode ser usado para estimar valores rapidamente sem ter que processar todas as iterações necessárias para a convergência. A implementação deste tipo de aproximações pode ser muito útil em aplicações que exijam um elevado número de iterações em comparação com os recursos computacionais disponíveis. Durante as simulações são monitorados alguns pontos ao longo de todas as iterações, mas o conjunto dos valores de SAR calculado será mostrado em forma gráfica apenas para instantes determinados para economizar o de tempo de comunicação e o uso da capacidade de armazenamento. Na Figura 22, está representada a SAR (em  $mW/g$ ) simulada na cabeça do usuário, normalizada para um campo elétrico de  $1 V/m$  RMS na antena, na qual o plano horizontal representa um corte sagital da cabeça e no eixo vertical estão os valores da SAR.



**Figura 22. SAR num corte sagital da cabeça do usuário em  $mW/g$  para  $1 V/m$  RMS de excitação.**

Os valores obtidos nas simulações dependem de muitas variáveis do modelo, mas principalmente do posicionamento da antena em relação ao usuário, e ultrapassam em muitos casos os valores sugeridos nas normas (e.g., SAR = 1,6  $mW/g$  na norma ANSI [8] ou SAR = 2,0  $mW/g$  na norma ICNIRP [9]). A definição da posição relativa da antena em relação à cabeça do usuário para efeitos de comprovação do atendimento à norma ICNIRP, tanto em simulações como em medidas, tem sido discutida recentemente [48].

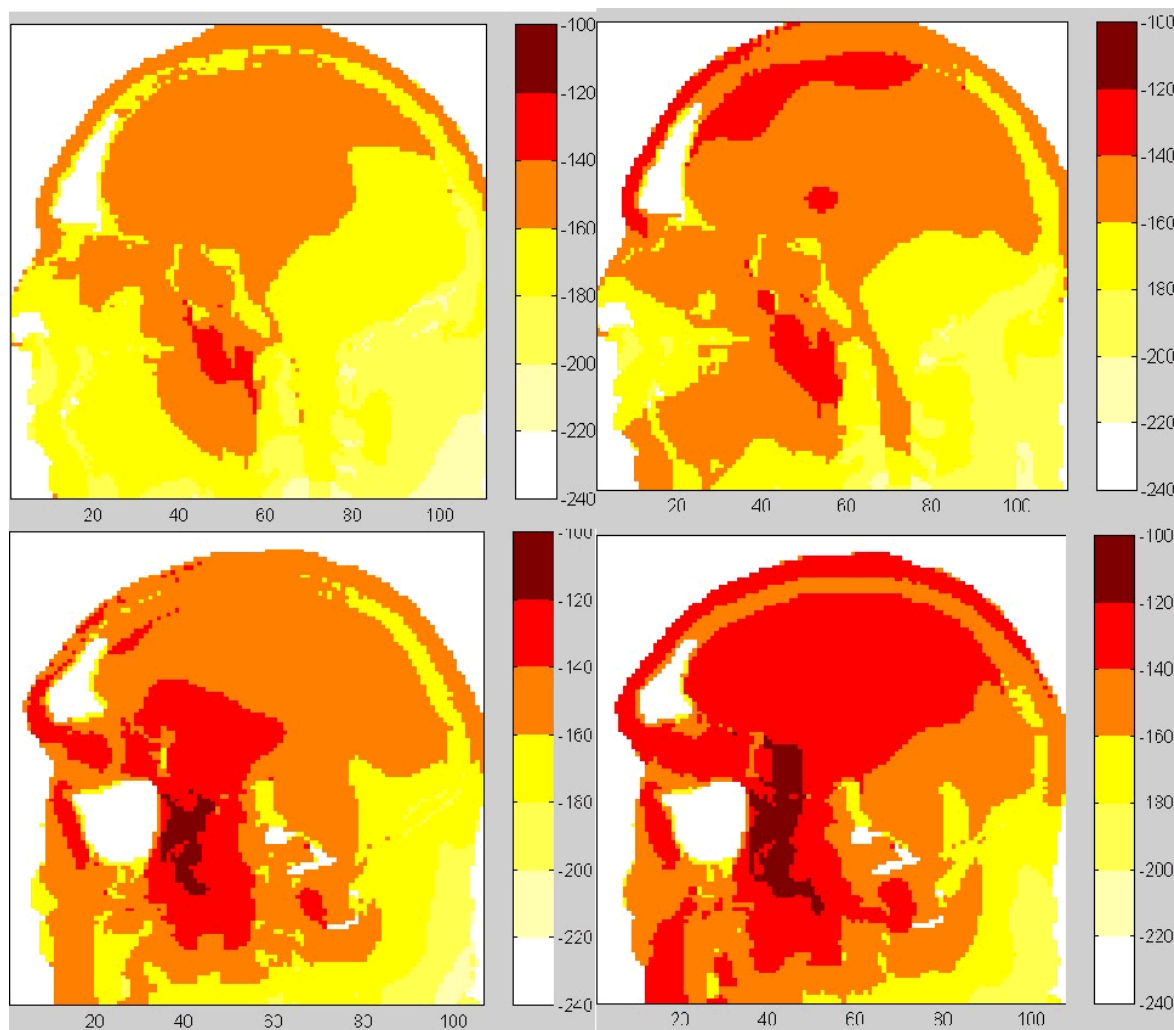
O algoritmo FDTD fornece em cada iteração, soluções aproximadas do valor de campo e SAR, no instante correspondente a essa iteração. Assim é possível visualizar através dele a evolução espacial das ondas se propagando e o aumento da SAR desde a condição inicial de campo nulo até a convergência como na Figura 23.



**Figura 23. Imagem obtida da SAR da cabeça do usuário para um corte sagital, depois de 1 ns, 2 ns e 3 ns de exposição.**

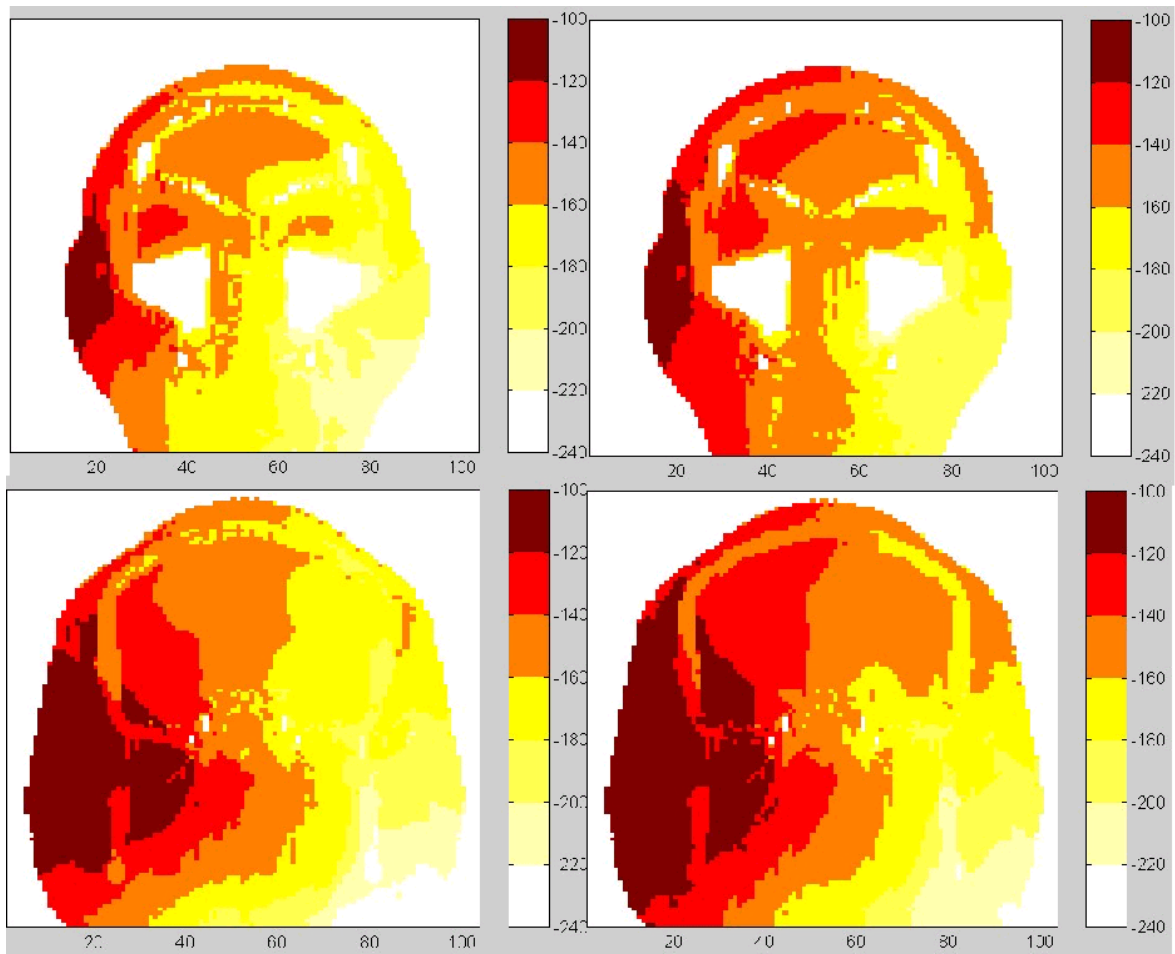
A escala de cor das figuras acima representa  $mW/g$  de SAR por  $V/m$  RMS de campo da fonte (i.é., num ensaio onde a fonte foi modelada como de 1  $V/m$  RMS) em escala logarítmica ( $10 \times \log(\text{SAR})$ ). As três figuras correspondem a 300, 600 e 900 iterações ou aproximadamente tempos de 1 ns, 2 ns e 3 ns.

Os valores obtidos na convergência numérica do método podem ser comparados dependendo da distância e posição onde a antena é colocada, e observados para diferentes cortes (Figura 24), comprovando a profundidade de penetração dos valores mais elevados da SAR (Figura 25).



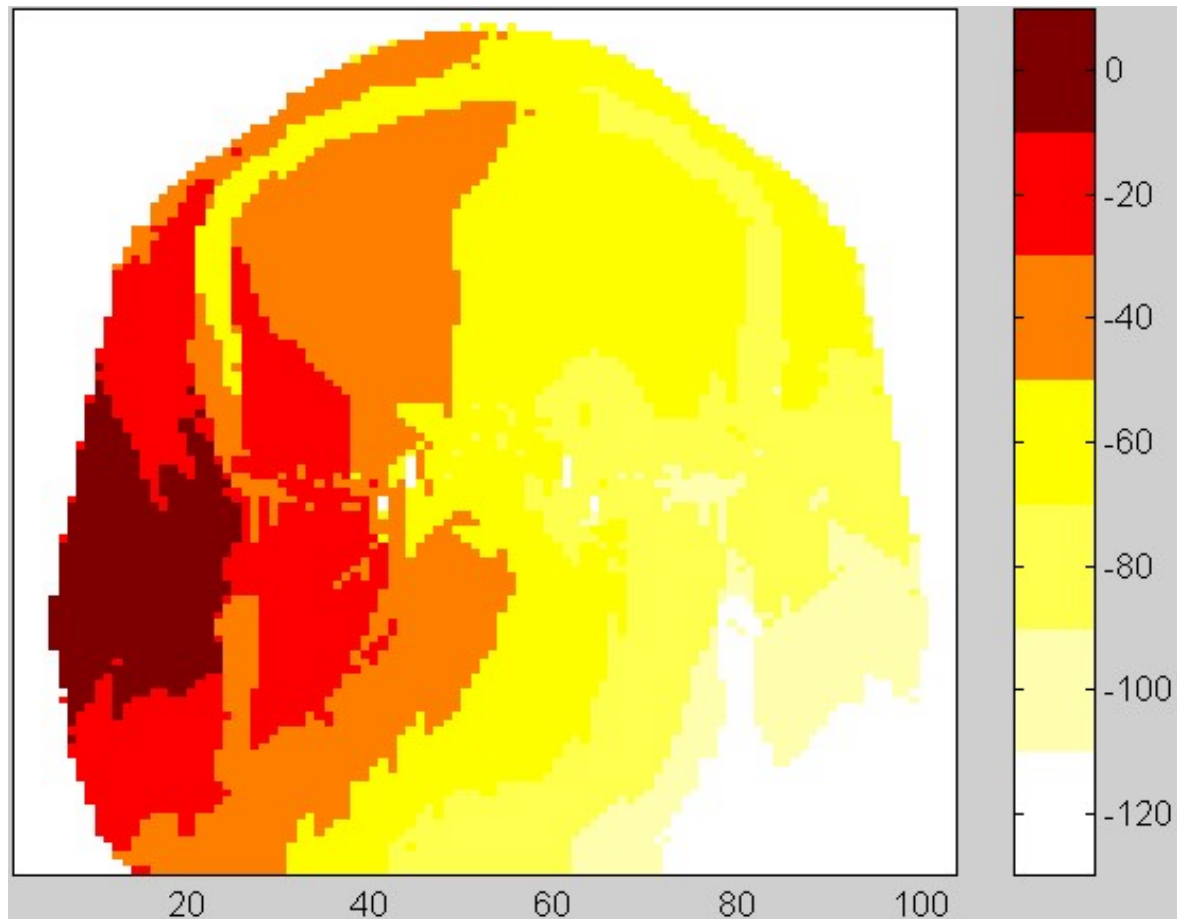
**Figura 24. Imagens obtidas da SAR da cabeça do usuário para dois cortes, mais distante da antena (acima) e mais próximo (abaixo). Quando a antena é colocada a 2 cm (esquerda) e 1 cm (direita).**

Todas as Figuras 23 a 25, correspondem a resultados obtidas após 900 iterações, equivalentes a um tempo de exposição de 3,12 ns, salvo o especificado para a Figura 23. Pode-se observar que a SAR é muito intensa nas áreas do cérebro próximas ao plano da antena e podem-se notar máximos locais em tecidos sensíveis, por exemplo, nos olhos (Figura 25).



**Figura 25. Os cortes transversais, neste caso na altura dos olhos (acima) e no plano da antena (abaixo) permitem observar elevados valores no interior da cabeça, para a antena a 2 cm e 1 cm, (esquerda e direita respectivamente).**

Na Figura 26 a potência entregue à antena é  $P_{del} = 600 \text{ mW}$ , para reproduzir o caso de um telefone celular operando no modo analógico em sua máxima potência. A escala de cor corresponde à SAR normalizada  $[10 \times \log(\text{SAR})]$  em que 0 dB corresponde a uma  $\text{SAR} = 1 \text{ mW/g}$ , então, o valor superior da cor mais escura (10 dB acima) corresponde a  $10 \text{ mW/g}$ .



**Figura 26.** Imagem transversal que apresenta os resultados normalizados [ $10 \times \log(\text{SAR})$ ], para uma antena a 2 cm de distância e sendo a potência entregue à antena  $P_{\text{del}} = 600 \text{ mW}$ .

#### 4.1 COMPARAÇÕES COM OUTRAS SIMULAÇÕES

Considerando os estudos realizados por outros autores que utilizam diferentes métodos numéricos computacionais ou até mesmo medidas em *phantoms*, é possível verificar uma similaridade muito grande nas conclusões dos diferentes estudos [40-43, 49-51]. A presença de áreas de concentração de campo e o cálculo de valores elevados, acima das recomendações internacionais são referidos na maioria dos trabalhos que abordam estas simulações. Uma comparação mais direta com os resultados dos diferentes trabalhos requer, porém reproduzir exatamente os diferentes parâmetros da simulação, como a distância, posição, características da antena entre outros. Dadas as variadas possibilidades de modelagem, e de apresentação dos resultados, os diversos autores encontram pequenas diferenças nos seus resultados, mas numa análise geral, pode-se concluir que as áreas mais afetadas pelo efeito do campo eletromagnético, obtidas neste trabalho, são equivalentes e com valores muito aproximados aos obtidos por outros autores. A Figura 27 mostra comparações dos valores obtidos em



diferentes simulações publicadas com os valores aqui obtidos. Os limites recomendados de  $1,6 \text{ mW/g}$  e  $2 \text{ mW/g}$ , respectivamente pelas normas IEEE/ANSI e ICNIRP, são ultrapassados em todas as projeções para distâncias menores que aproximadamente  $2 \text{ cm}$ .

### Comparações com Resultados de Simulações

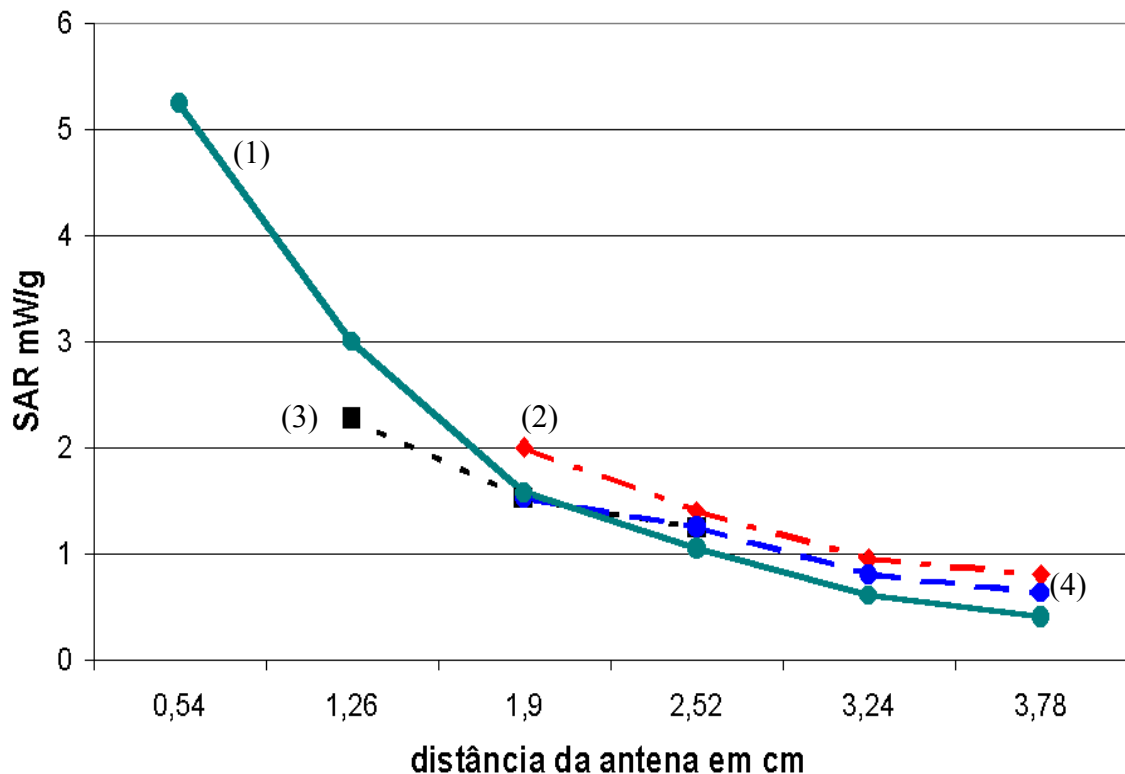


Figura 27. Gráfico da SAR simulada na cabeça do usuário. A curva (1) inteira (verde) obtida neste trabalho (para  $31,6 \text{ V}_{pp}/\text{m}$ ) é comparada com as curvas de vários autores: (2) ponto-e-traço (vermelha) [40], (3) pontilhada (preta) [49] e (4) tracejada (azul) [42] todas com  $P_{del} = 1 \text{ W}$ .

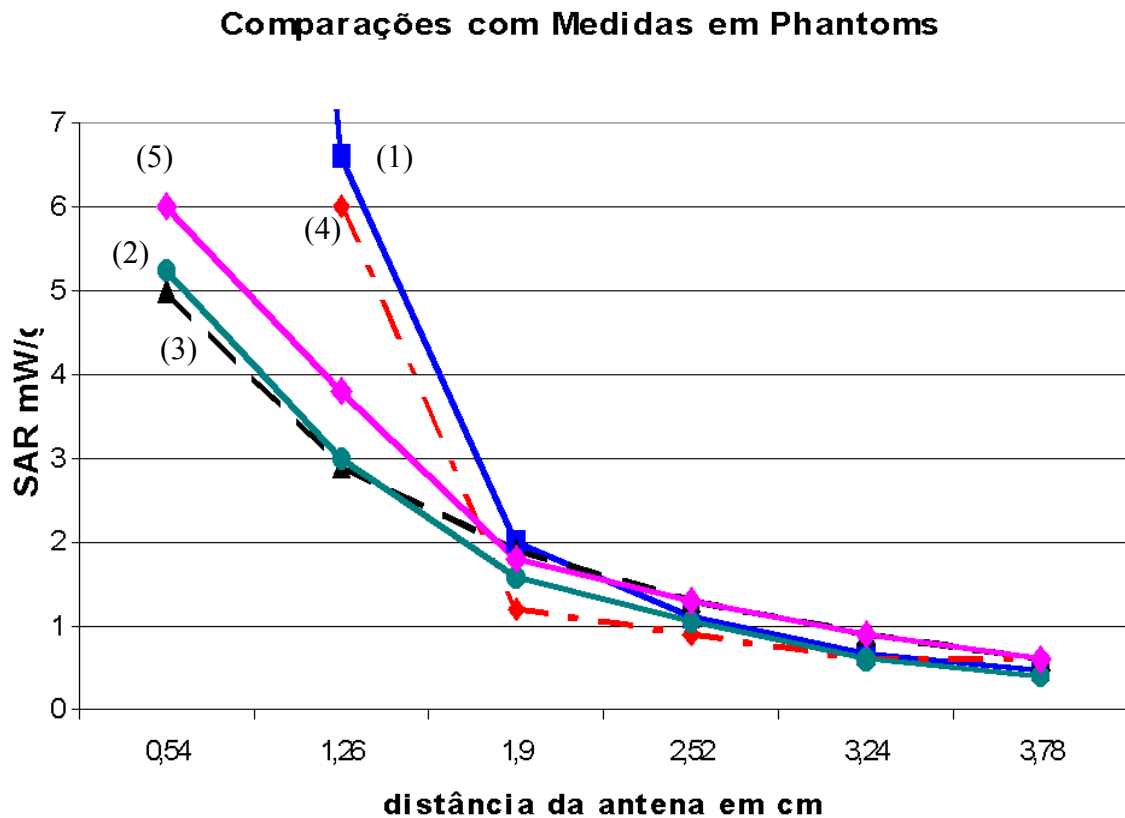
## 4.2 COMPARAÇÕES COM MEDIDAS

As medidas de SAR requerem a montagem de um sofisticado laboratório com *phantoms* ou modelos específicos que simulam o corpo humano, compostos de materiais (gel ou plásticos) com parâmetros eletromagnéticos equivalentes à média dos tecidos. Além dos *phantoms* são necessários transdutores, em geral em microlinhas, e braços posicionadores (automatizados) que interfiram o menos possível com os campos que estão sendo medidos. Um programa específico deve controlar os movimentos do conjunto para realizar medidas em



diferentes pontos e, dependendo dos transdutores usados, considerar a soma de componentes ortogonais para simular um transdutor isotrópico. Além desses equipamentos específicos, sofisticados equipamentos gerais como gerador e amplificador de RF, analisador de redes e paredes anecoicas são necessários para a montagem de um laboratório de medidas de SAR.

Assim como no caso das simulações, as medidas podem ter valores diferentes em função da indefinição da distância e posição da antena entre outros parâmetros mais relevantes.



**Figura 28.** Valores obtidos nas atuais simulações: no olho, com antena frontal (linha (1) inteira, azul) e *hotspot* interior ao crânio com antena lateral (linha (2) inteira, verde, pontos circulares), comparados com resultados medidos para antena lateral em modelo homogêneo (linha (3) tracejada, preta) e *phantom* não homogêneo (ponto-e-linha (4), vermelha) [50].  $P_{del} = 0.45 W$ . A curva (5) cor de rosa, inteira mostra resultados de medidas em *phantoms* e simulações apresentada em [51]

E possível verificar a concordância entre as simulações realizadas e medidas publicadas. O importante aumento da SAR, com a diminuição da distância entre a antena e a cabeça, é observado em todos os casos.

## 5 PROJETO DE ANTENAS DIRETIVAS

Os resultados das simulações anteriores e a revisão da literatura indicam que devam ser estudadas alternativas visando minimizar a energia absorvida na cabeça. Por outro lado, como o diagrama de irradiação no plano horizontal, para as antenas mais usadas em telefones celulares (tipo monopolo) é aproximadamente simétrico quando estão afastadas de qualquer obstáculo, quando operam próximas à cabeça têm seu diagrama de irradiação distorcido, sendo mais densidade de potência emitida na direção do usuário. Assim o uso de antenas diretivas representaria uma diminuição dos níveis de radiação sobre o usuário, sem que com isto se prejudiquem os níveis emitidos na direção oposta à cabeça, que é a direção preferencial para a comunicação.

Antenas planares (e.g., em “*Microstrip*”, CPW - “*Coplanar Waveguide*”, “*Slot-line*”, etc.) têm sido descritas na literatura científica [47, 56-64], podendo representar substanciais melhorias no desempenho dos telefones móveis, tanto em termos de diagrama de irradiação (por exemplo, uma curva do tipo cardióide [68]), relação frente-costas (por exemplo, da ordem de 15 dB), impedância de entrada, largura de banda, etc., resultando em melhoria da qualidade de comunicação, redução de consumo da bateria e dos riscos aos usuários.

Alguns autores desenvolveram antenas impressas para aplicações em telefones móveis, demonstrando algumas melhorias de desempenho (por exemplo, em termos de eficiência de irradiação, etc.) em relação à antena monopolo convencional [56-67]. Estas antenas podem ser compactas, e com características de produção rentáveis provenientes de matérias primas baratas e de fácil fabricação, além de serem robustas, pois sua estrutura não possui partes móveis ou salientes o que as torna ideais para transceptores portáteis. Além disto, devido a sua reduzida espessura, elas podem ser acomodadas em superfícies curvas ou planas como na capa de cobertura móvel do aparelho. Suas características de irradiação podem ser adaptadas para a aplicação desejada, como, por exemplo, seu diagrama de irradiação, relação frente costas, largura de banda, impedância de entrada, etc. Por exemplo, a PIFA (“*Planar Inverted F Antenna*”) e a BIFA (“*Bent Inverted F Antennas*”) mostraram estas melhorias [40, 61, 62]. Também a antena espiral descrita em [66] mostrou resultados interessantes.

Com o objetivo de testar uma aplicação diferente do algoritmo FDTD desenvolvido e ao mesmo tempo verificar conceitualmente a possibilidade de aplicação de antenas diretivas em transmissores portáteis, foi iniciado o estudo de antenas em microlinhas. A seguir se descreverá sumariamente o projeto, realização e os resultados preliminares medidos para o protótipo experimental de uma antena tipo monopolo de um quarto de onda em *microstrip*.

## 5.1 PROJETO DE ANTENAS USANDO FDTD

Antenas planares mais sofisticadas e de desempenho aprimorado já foram apresentadas na literatura [62, 66]. Não se pretende aqui mostrar o desenvolvimento de uma antena de alto desempenho, mas realizar um teste de conceito (a utilidade de antenas diretivas em microlinhas como alternativa para minimizar a potência absorvida na cabeça do usuário de transceptores portáteis) junto com uma aplicação do algoritmo FDTD para projeto.

A antena apresentada é constituída de duas seções. A primeira delas é uma linha de quarto de onda em aberto em sua extremidade e, portanto, apresenta semelhanças com uma antena monopolo, e a segunda opera como um transformador de quarto de onda, necessário para o casamento de impedância com o transmissor.

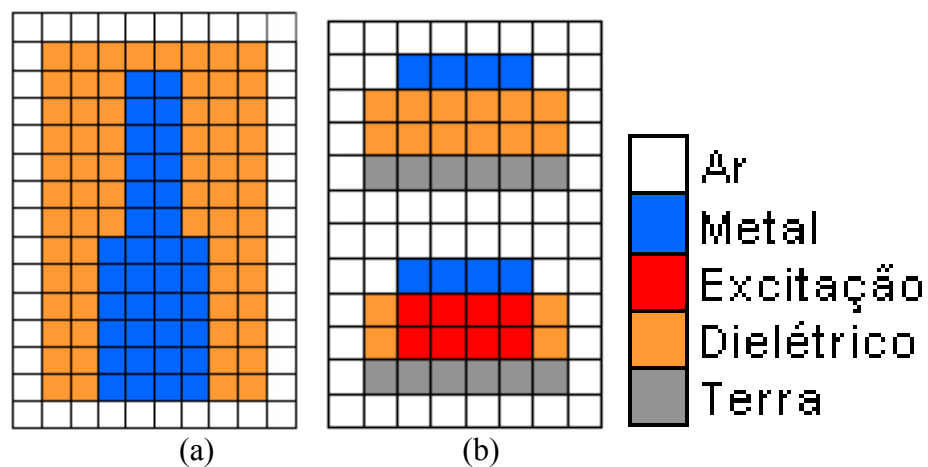
Foram utilizadas equações bem estabelecidas [47, 56-60, 67, 68] para o dimensionamento dessas duas seções. O transformador de quarto de onda é dimensionado para adaptar a baixa impedância na entrada do monopolo de quarto de onda à impedância vista dos terminais do transceptor portátil. Esta antena foi também modelada utilizando-se o método FDTD [67], que permite estimativa precisa dos campos na região próxima e distante, constituindo-se em importante ferramenta para o projeto e otimização dessas antenas.

A frequência central escolhida foi 840 MHz. O substrato disponível, “Perstorp” tipo FR-4 V0 com constante dielétrica relativa 4,6, espessura 1,58 mm e 0,035 mm de metalização, não é adequado para o uso em antenas, esperando-se de antemão que as antenas tenham baixa eficiência. A impedância de entrada do monopolo de quarto de onda foi estimada em 22,5  $\Omega$ , a impedância característica calculada do transformador de quarto de onda é 35,7  $\Omega$ , apresentando então em sua entrada uma impedância de 50  $\Omega$ . Foi utilizado um conector SMA para substituir o conector original, então mantendo menores possíveis as dimensões.

Nesta simulação se analisa o comportamento da antena *microstrip* descrita acima, para a região próxima, através de um modelo simplificado e um algoritmo FDTD. A antena foi

implementada dentro de um domínio contendo ar, para que o sinal se propagasse no espaço livre, ou seja, nessas simulações não foi incluída a cabeça como nas simulações anteriormente apresentadas.

Para tanto, foram inicialmente determinadas dimensões apropriadas para a malha (matriz), escolhidas de forma que o modelo fosse o mais fiel possível. Assim teremos um domínio cúbico com células com dimensões espaciais:  $\Delta x = \Delta y = \Delta z = 0,7 \text{ mm}$  e as dimensões da antena são compostas de múltiplos inteiros desses elementos. Nesta malha, determina-se através de índices, o material para cada posição (célula). Com isso, a antena é “desenhada” nos diversos “cortes” da matriz (Figura 29).



**Figura 29. Representação de uma matriz contendo a antena microstrip, onde: (a) vista superior (b) vista em corte transversal.**

Na Figura 29b está representado um corte transversal à antena, no qual está indicada a excitação, aplicada apenas nas células correspondentes à região restrita do dielétrico, no plano onde inicia a *microstrip*. Na sua menor dimensão, considera-se o metal como condutor perfeito, ou seja, com condutividade infinita e, nas células em que há metal, as equações de diferenças finitas que o método propaga são corrigidas de forma a eliminar as parcelas de campo dentro do metal, podendo desta forma evitar a mudança de malha para representar espessuras de metal muito menores que as que poderiam ser descritas dadas as dimensões das células. Os principais parâmetros do dielétrico utilizado estão apresentados na Tabela 2.

**Tabela 2. Parâmetros dos materiais usados na simulação.**

<b>Material</b>	$\epsilon_r$	$\sigma \text{ [S/m]}$
Ar	1	0
Metal /Plano Terra	-	$\infty$
Dielétrico (FR-4)	4,6	$2,1742 \times 10^{-3}$

Com a determinação do domínio do problema, todos os campos eletromagnéticos inicialmente são nulos. São inseridos, então, nos pontos referentes à excitação, campos que se propagarão por todo o domínio (antena/ar) ao longo das iterações. Após a convergência numérica do método, são disponibilizadas matrizes dos campos eletromagnéticos que serão transformadas em imagens para apreciação. A propagação dos campos pode-se perceber através de cortes transversais (Figuras 30 e 31), longitudinais (Figura 32) ou vista de topo (Figura 33).

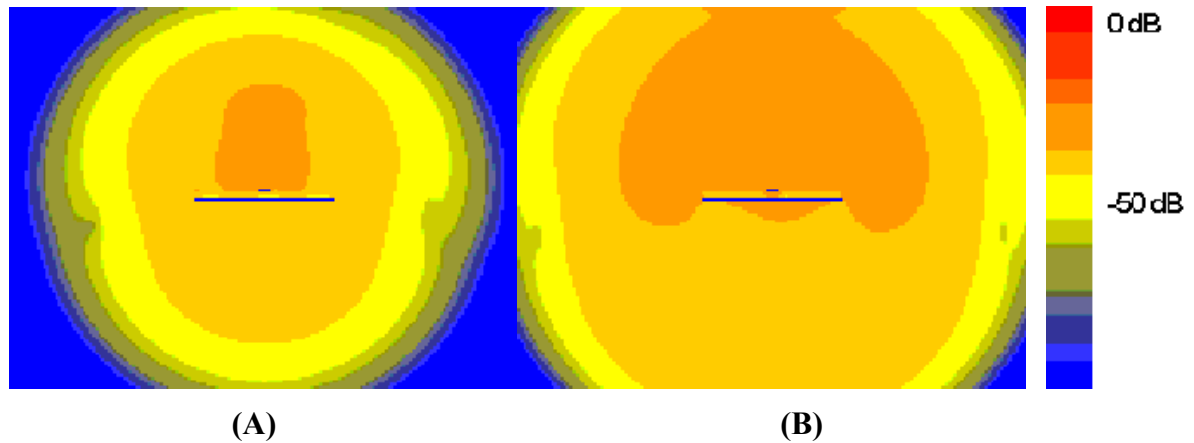


Figura 30. Imagens da propagação de campos eletromagnéticos para a antena microstrip, nas quais: (a) ocorre após 0,34 ns de propagação e (b) ocorre após 0,375 ns de propagação.

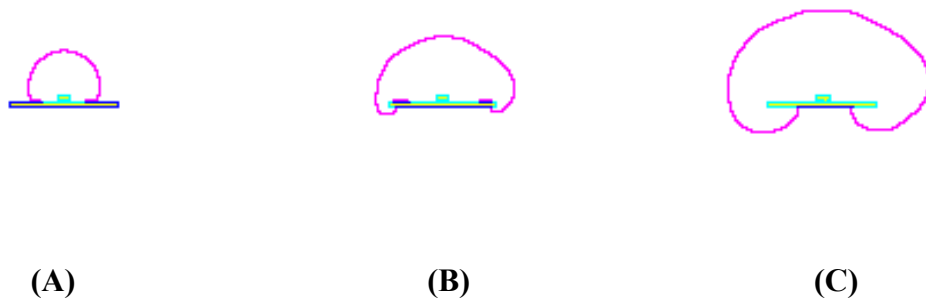


Figura 31. Imagens de cortes transversais da antena microstrip, nas quais: (a) ocorre após 0,69 ns, (b) ocorre após 0,75 ns e (c) ocorre após 0,84 ns.

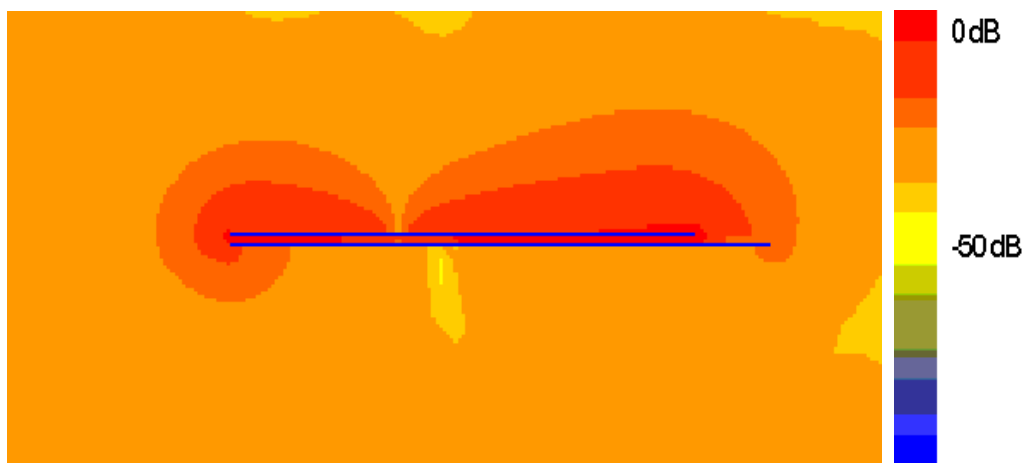


Figura 32. Campo elétrico propagado a partir da antena microstrip, para uma vista longitudinal após 0,86 ns.

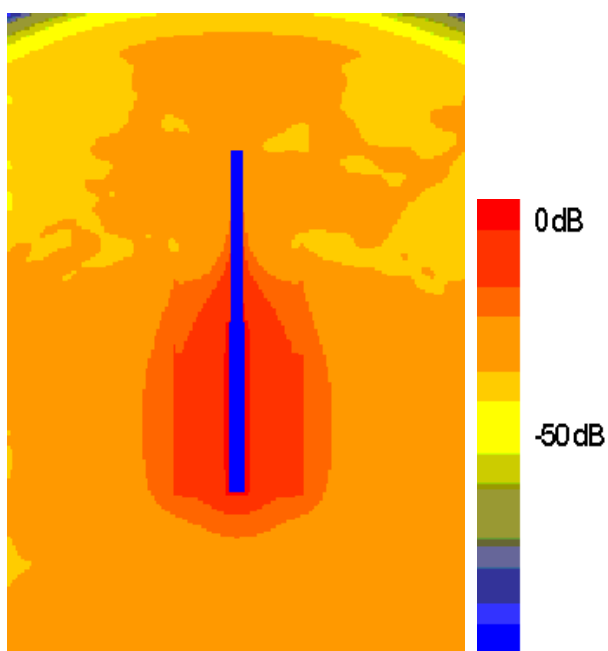


Figura 33. Propagação de campos eletromagnéticos a partir da antena microstrip, para uma vista de topo após 0,46 ns.

## 5.2 MEDIDAS DE PROTÓTIPOS

A avaliação das características da antena, tanto em campo próximo como em campo distante, é tarefa difícil de ser realizada já que as condições ideais para a medida normalmente não existem e controlar o ambiente de modo a simular o espaço livre pode não ser possível, desde que não se dispõe de uma câmara anecóica.

Por conveniência utilizou-se uma antena monopolo no modo recepção acoplada a um analisador de espectro portátil Anritsu MS 2711 (princípio da reciprocidade) e a fonte transmissora (transceptor de *trunking* com frequências e potências programáveis) fixa em local distante o suficiente para ser considerada no campo distante [47].

Um critério adequado para definir essa distância é considerar que a antena sob teste seja iluminada por uma frente de onda praticamente plana e com diferenças de fase entre seu centro e sua borda maior não superior a  $\lambda/16$  [70]. Tipicamente a distância  $R$  a partir do qual o campo pode ser considerado distante é dada pela inequação:

$$R \geq \frac{2d^2}{\lambda} \quad (29)$$

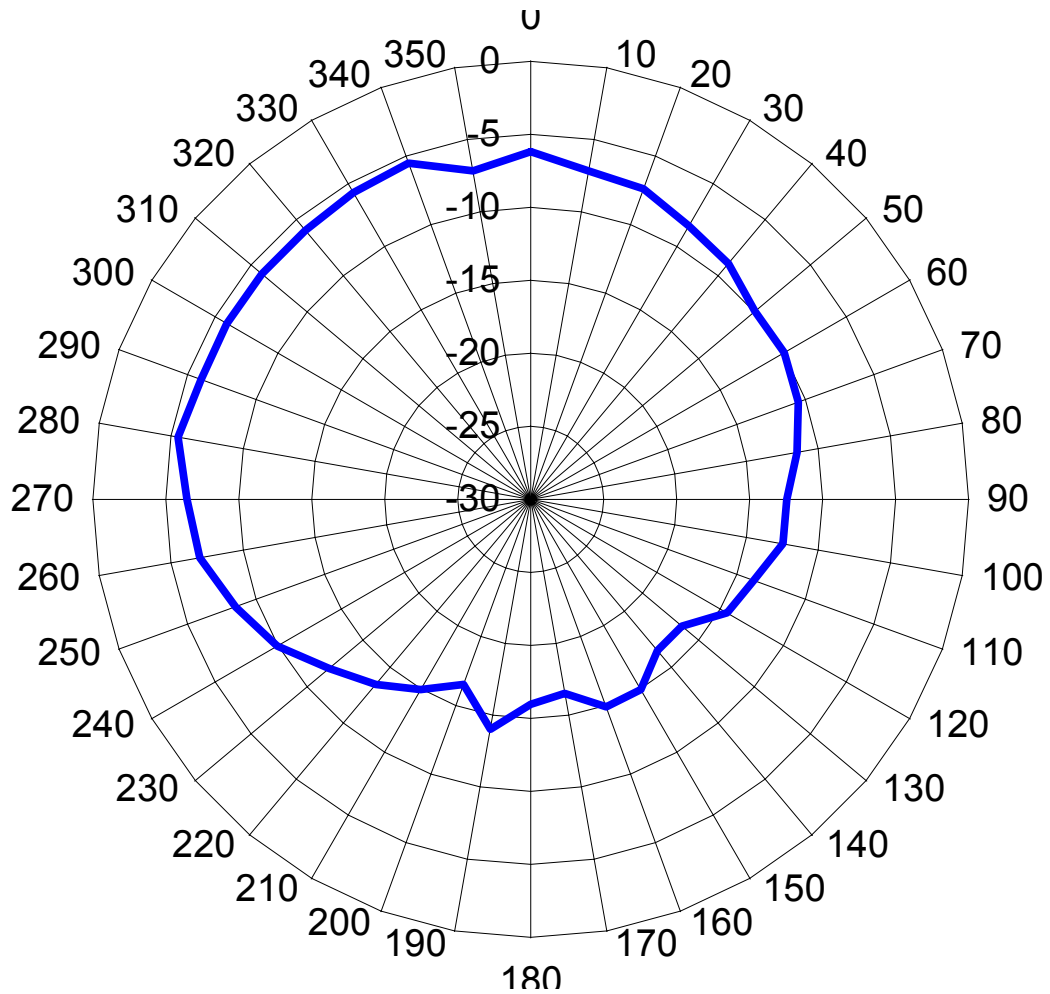
em que  $d$  é a maior dimensão da antena. Para a frequência de medida (cerca de 850 MHz) e para a antena em teste, esta distância é de 0,7 m.

Para aproximar o teste da condição de espaço livre foi utilizada a disposição de medição inclinada (*slant range*) [47, 67] de modo que a antena teste é colocada a uma altura maior que a antena transmissora.

Tal disposição tende a suprimir a onda refletida de terra e evita que outras reflexões degradem os valores de relação frente-costa existentes já que a onda que se dirige à antena também se dirige ao infinito.

Ajustes foram realizados em bancada (especialmente com elementos reativos e alargando a fita da seção do transformador de quarto de onda, que passou para uma impedância característica de cerca de 30  $\Omega$ ), visando a otimização dos principais parâmetros da antena. Entre outros, a taxa de onda estacionária da antena foi substancialmente reduzida ao longo da faixa desejada.

Os principais resultados preliminares obtidos serão mostrados a seguir. O diagrama de irradiação no plano horizontal está representado na Figura 34.



**Figura 34. Diagrama de irradiação da antena microstrip no plano horizontal. A parte frontal da antena corresponde ao ângulo  $0^\circ$  e os valores de potência, em  $dB$ , são em relação à potência equivalente irradiada por um monopolo de  $\lambda/2$ .**

Observa-se neste diagrama uma relação frente-costa da ordem de  $13\text{ dB}$ .

Na Figura 35 está mostrado o diagrama de irradiação no plano vertical medido para a parte frontal da antena (o plano da antena corresponde a  $0^\circ$ ).



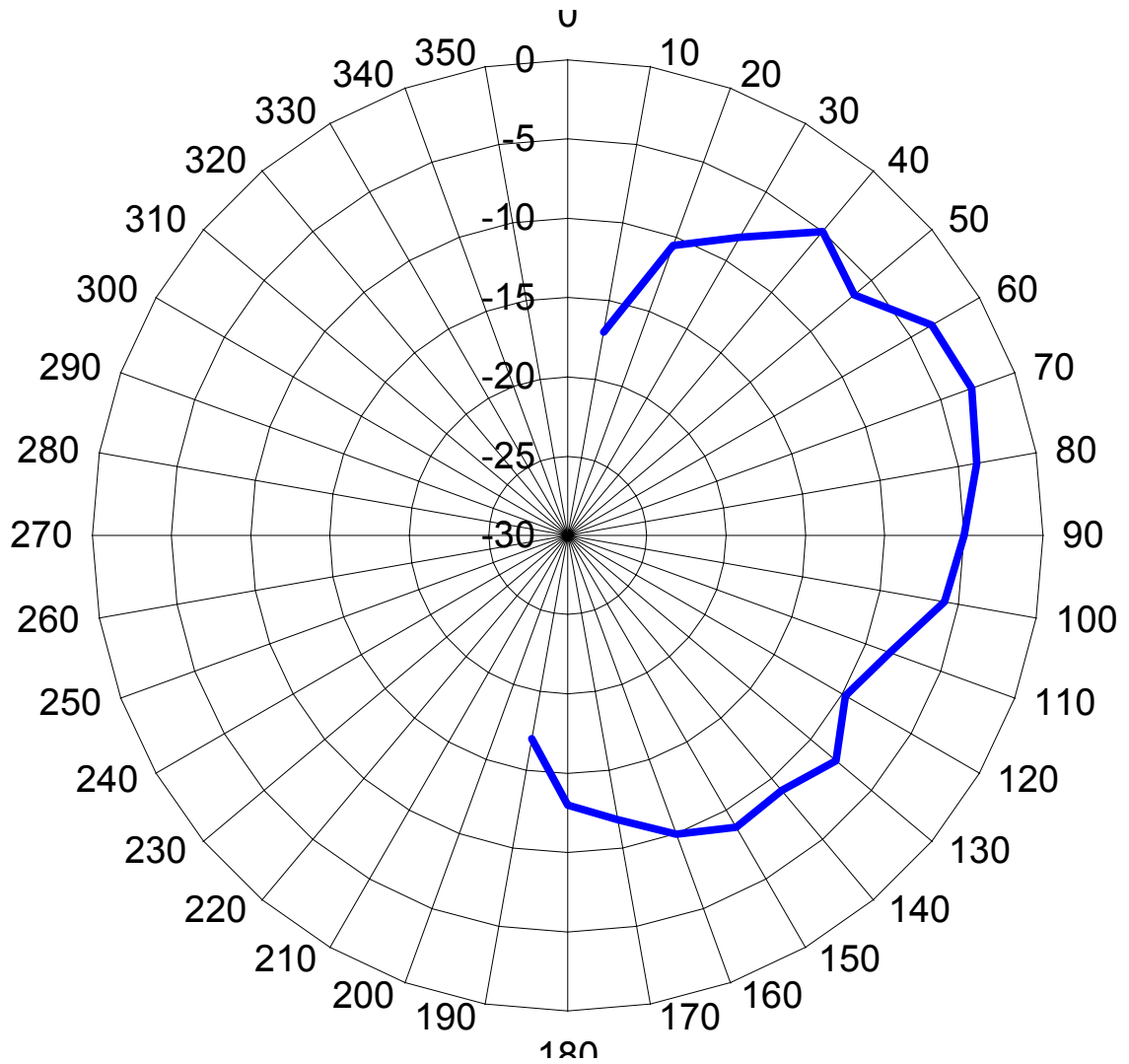


Figura 35. Diagrama de irradiação da antena microstrip no plano vertical. Onde 0 dB corresponde a o ganho de um monopolo de  $\lambda/2$  na direção de máximo ganho.

Na Figura 36 está mostrada a taxa de onda estacionária medida na entrada desta antena.

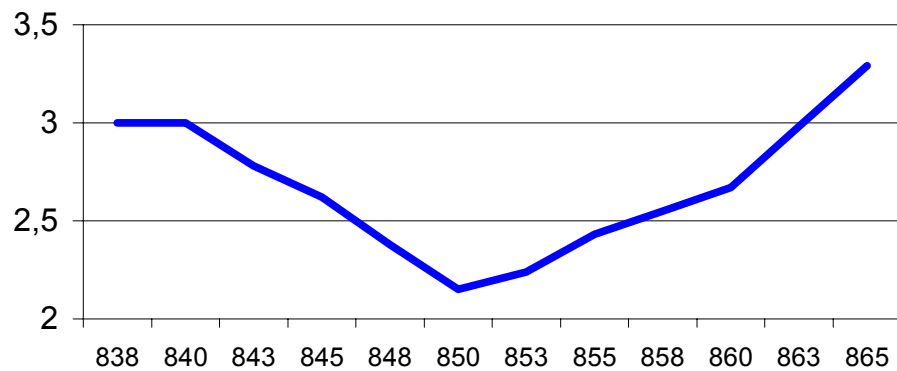


Figura 36. Relação de onda estacionária medida na faixa de 837,5 a 865 MHz.

Observa-se dessa figura que a razão de onda estacionária medida para a antena microstrip na faixa de 840 a 860 *MHz* mantém-se menor que três, o que é um resultado adequado para aplicações práticas.

De uma maneira geral, observa-se dos resultados preliminares medidos, que o diagrama de irradiação apresenta ângulos de abertura entre os pontos de meia potência da ordem de 180° no plano horizontal e de cerca de 90° no plano vertical, relação frente-costas da ordem de 13 *dB*, razão de onda estacionária na entrada menor que 3, para a faixa de frequências entre 840 e 860 *MHz*.

Esses resultados preliminares certamente poderão ser melhorados. Também, com a utilização de substratos com constantes dielétricas mais elevadas, obtém-se uma redução substancial nas dimensões destas antenas permitindo uma melhor integração nos aparelhos portáteis.

## 6 CONCLUSÕES E RECOMENDAÇÕES

Foram descritos sumariamente os efeitos da exposição à radiofrequência e microondas, em particular diferenciando os mesmos dos efeitos provocados por radiações ionizantes, que são conhecidos há mais tempo, e também da radiação solar, que é absorvida mais superficialmente. Entre estes efeitos, os mais conhecidos, que são os efeitos térmicos, dão origem a recomendações e normas de exposição. Algumas dessas normas tem um amplo reconhecimento internacional e indicam os limites de exposição em parâmetros de campos elétricos e magnéticos, e em SAR. Para verificar o atendimento a estas normas, basta então medir ou calcular os valores de campo ou da SAR.

Para o caso das estações de radiobase, os cálculos e medidas não apresentam maior dificuldade e mostram que os limites recomendados (mesmo os mais restritivos) são atendidos para as distâncias onde pode haver público. Já para o caso dos transceptores portáteis, tanto o cálculo como as medidas requerem ferramentas especiais. Para desenvolver estes cálculos e assim poder estimar os níveis de campo e SAR sobre o usuário de transceptores portáteis, foram desenvolvidos um modelo de cabeça humana (malha do método numérico) e um *software* baseado no método das Diferenças Finitas no Domínio do Tempo, FDTD.

A convergência do algoritmo desenvolvido foi testada, comparando diferentes soluções para as condições iniciais e condições de contorno do problema, e os resultados calculados foram validados. Estes resultados são convergentes com aqueles publicados por vários autores e indicam que a SAR cresce substancialmente quando a distância entre a antena e a cabeça decresce, ultrapassando  $2 \text{ W/Kg}$  para distâncias menores que  $2 \text{ cm}$ , por exemplo. Uma vez que uma percentagem substancial de usuários, talvez a maioria, opera os celulares com a antena monopolo a uma distância menor que  $2 \text{ cm}$  da cabeça e muitas vezes por longos períodos, as normas ICNIRP (e IEEE/ANSI) são então substancialmente excedidas e as recomendações de limitar o tempo de uso e cuidar para manter uma distância maior que  $2 \text{ cm}$  (ou tanto quanto for possível) entre o transceptor e a cabeça são pertinentes.

Como os atuais limites de exposição são essencialmente baseados nos efeitos térmicos (p.ex., ICNIRP e IEEE/ANSI), eles podem ser objeto de revisão, que os tornaria mais

restritivos, dependendo dos resultados das pesquisas em andamento (por exemplo, coordenadas pela Organização Mundial da Saúde - OMS), especialmente em efeitos não-térmicos. Por outro lado, é de se esperar pressões por parte da indústria para não diminuir os níveis de potência emitidos, o que poderia prejudicar a comunicação e o desenvolvimento de sistemas que permitam o intercâmbio de maiores quantidades de informação (imagem, vídeo, dados). Isto porque, para aumentar a quantidade de informação enviada as alternativas são o aumento de banda de frequências, a sofisticação das modulações e codificações usadas (e.g. CDMA, *spread spectrum*) e o aumento de potência.

Uma alternativa conceitual para permitir o desenvolvimento de telefones celulares que irradiem o usuário com níveis baixos, sem diminuir a potência transmitida nem prejudicar a comunicação, baseada em antenas planares, foi proposta e sua viabilidade mostrada com cálculos e medidas em protótipos.

Entre as perspectivas de desenvolvimentos futuros deste trabalho se incluem o aperfeiçoamento e complementação do *software* desenvolvido, com (a) a construção de interfaces amigáveis para a entrada de dados e geração de malha a antena, (b) a adaptação dos algoritmos e da malha para outras geometrias, implementando equações para outros sistemas de coordenadas, transformações (rotações escalas) e malhas heterogêneas (com células de diferentes tamanhos), (c) o desenvolvimento e teste de bordas absorvedoras mais eficientes, (d) implementação de cálculos para campo distante e sua validação.

Também a aplicação do *software* para diferentes problemas de espalhamento eletromagnético, em particular na continuação do estudo e desenvolvimento de antenas planares pode ser abordado.

Para obter um entendimento mais profundo dos efeitos térmicos que definem os limites das normas mais difundidas, é possível implementar equações de calor biológico (Equação 7). Partindo dos valores de SAR calculados com este *software* FDTD, o modelo pode ser criticado, e suas outras componentes implementadas e resolvidas por uma combinação de métodos numéricos.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] D'ÁVILA, C.K **Programa de Treinamento para Profissionais da ANATEL**, CEDET Centro de Desenvolvimento Profissional e Tecnológico, UnB, 1998
- [2] DOWD, S.E.; TILSON E.R. **Practical Radiation Protection and Applied Radiobiology** Univ. of Alabama, Birmingham. Brandon/Hill Allied Health List selection (#305) Apr 1999
- [3] IRNICH, W.; TOBISCH, R. Mobile Phones in Hospitals, **Biomedical Instrumentation and Technology**, pp 128 –34, Jan/Feb 1999.
- [4] IRNICH, W.; TOBISCH R. Mobilfunk im Krankenhaus: Einfluß von Mobiltelefonen auf Lebensrettende und Lebenserhaltende Medizintechnik **Schiele und Schön**, Berlin 1999
- [5] FISCHETTI, M. The Cellular Phone Scare. **IEEE Spectrum**, p. 43-47, Jun. 1993.
- [6] JENSEN, M. A.;RAHMAT-SAMII, Y. EM Interaction of Handset Antennas and a Human in Personal Communications. **Proc. of the IEEE**, v. 83, n. 1, p. 7-17, Jan. 1995.
- [7] SALLES, A. A. A. DE. Efeitos Biológicos dos Telefones Celulares Portáteis. **Revista da Sociedade Brasileira de Telecomunicações**, v. 11, n. 1, p. 71-80, Dez. 1996.
- [8] ANSI – American National Standards Institute. **IEEE C95.1-1991: IEEE Standard for Safety Levels with Respect to Human Exposure to Radio Frequency Electromagnetic Fields, 3 kHz to 300 GHz**. The Institute of Electrical and Electronics Engineers, Inc., 345 East 47 Street, New York, NY 10017-2394, USA.
- [9] ICNIRP GUIDELINES – Guidelines for Limiting Exposure to Time-Varying Electric, Magnetic and Electromagnetic Fields (Up to 300 GHz) – International Commission on Non-Ionizing Radiation Protection, **Health Physics**, v. 74, n. 4, p. 494-522, Apr. 1998.
- [10] SALLES, A. A. A. DE; FERNÁNDEZ, C. Riesgos a la Salud Provocados por los Teléfonos Celulares Móviles: una Discusión Reabierta. Los Efectos a Largo Plazo. **Revista Infosuis**, n. 4, p. 2-5, Out. 1999.
- [11] MOSELEY, H. **Non-Ionizing Radiation**. Medical Physics Handbooks 18, p. 38-61.
- [12] BERNHART, J. H. Non-Ionizing Radiation Safety: Radiofrequency Radiation, Electric and Magnetic Fields. **Physics on Medicine and Biology**, v. 37, n. 4, p. 807-844, 1992.

- [13] BRONZINO, J. D. Biologic Effects of Non-ionizing Electromagnetic Fields. **The Biomedical Engineering Handbook**, CRC Press & IEEE Press., p. 1388-1389 and p. 1380-1440, 1995.
- [14] STUCHLY, M. A. Mobile Communication Systems and Biological Effects on their Users. **The Radio Science Bulletin**, n. 275, p.7-13, Dec. 1995.
- [15] DE POMERAI, D. Non-Thermal Heat-Shock Response to Microwaves. **Nature**, v. 405, p. 417- 418, May 2000.
- [16] DE POMERAI, D. Microwave Radiation Induces a Heat-Shock Response and Enhances Growth in the Nematode *Caenorhabditis Elegans*, **IEEE Trans. On MTT**, v. 48, n. 11, p. 2076-2081, Nov. 2000.
- [17] SALLES, A. A. A DE. Biological Effects of Microwaves and RF. **SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference**. p. 611-617, 1999.
- [18] WORLD HEALTH ORGANIZATION **International EMF Project**, <http://www.who.int/emf/>.
- [19] SWITZERLAND FEDERAL AGENCY FOR ENVIRONMENT, FORESTS AND LANDSCAPE **Ordinance on Non-Ionizing Radiation Protection n. 814.710**, December 23, 1999, Art. 64.a., p. 14,
- [20] MINISTERO DELL' AMBIENTE ITALIANO, **DECRETO 10 settembre 1998, n. 381 - Regolamento recante norme per la determinazione dei tetti di radiofrequenza compatibili con la salute umana** pubblicato sulla G.U. n. 257, 03 novembre 1998.
- [21] European Union to Adopt ICNIRP-Based Guidelines; Italy Protests that Decision Ignores Long Term Effects, **Microwave News**, 19\_4, pp.1-3 Jul 1999.
- [22] GANDHI O, P, ET AL. Temperature Rise for the Human Head for Cellular Telephones and for Peak SARs Prescribed in Safety Guidelines **IEEE Trans. on MTT**, vol. 49, n. 9 pp. 1607-1613 Sept. 2001.
- [23] CARNAHAN, B. **Applied Numerical Methods**. Krieger Publishing Co., 1990, ISBN 0-89464-486-6.
- [24] ANATEL, **Diretrizes para Limitação da Exposição a Campos Elétricos, Magnéticos e Eletromagnéticos Variáveis no Tempo**, Agência Nacional de Telecomunicações, Brasília, Brasil, Dez. 1999
- [25] **Lei n. 8463** de 19/01/2000, Prefeitura Municipal de Pôrto.Alegre, RS.
- [26] FCC 96-326: FCC Updates its Guidelines for Evaluating the Environmental Effects of Radiofrequency Emissions: Action Implements Telecom Act provisions (ET Docket No. 93-62), **Report DC96-76** <http://www.fcc.gov/oet/headline/fcc96326.html>
- [27] RADIO FREQUENCY INTERAGENCY WORK GROUP (RFIA-WG) **RF Guidelines Statement**, Jun, 19th, 1999

- [28] QISHAN YU, ET AL. An Automated SAR Measurement System for Compliance Testing of Personal Wireless Devices **IEEE Trans. on Electromagnetic Compatibility**, 41, pp. 234-245 Aug 1999
- [29] WORLD HEALTH ORGANIZATION Mobile Phones: Conclusions and Recommendations, **Fact Sheet 193**, Jun 2000, pg.5 [www.who.int/inf-fs/en/fact193.html](http://www.who.int/inf-fs/en/fact193.html)
- [30] H. T. FRIIS, A Note on a Simple Transmission Formula **Proc. IRE.**, Vol. 34, May 1946.
- [31] Y.OKUMURA ET AL., Field Strength and Its Variability in VHF and UHF LandMobile Radio Service **Review of the Electrical Communication Laboratories** Vol. 16, Nos. 9 and 10, 1968, pp. 825-873.
- [32] M. HATA, Empirical Formula for Propagation Loss in Land Mobile Radio Services, **IEEE Transactions on Vehicular Technology**, Vol. VT 29, No 3, 1980, pp. 317-325.
- [33] LEE, WILLIAM C. Y., **Mobile Cellular Telecommunications Systems**, McGraw Hill, 1989, ISBN 0070380899 pp 97-143.
- [34] AHLIN, L.; ZANDER, J. **Principles of Wireless Communications**, Studentlitteratur 1998, ISBN 91-44-00762-0
- [35] BLAUNSTEIN, N. **Radio Propagation in Cellular Networks**, Artech House, 1999, ISBN: 1580530672
- [36] CÁTEDRA, M.; PÉREZ, J. **Cell Planning for Wireless Communications** Artech House 1999, ISBN 0-89006-601-9
- [37] SALLES, A.A. DE; LISBÔA, J.A.; FERNÁNDEZ C. E NEVES, E.S. Estimativas Teóricas e Medidas dos Campos Irrradiados por Estações Rádio Base **Anais do IX Simpósio Brasileiro de Microondas e Optoeletrônica**, João Pessoa – PB, Agosto, 2000.
- [38] YEE, K.S. Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media **IEEE Transactions on Antennas and Propagation**, vol. AP-14, n°5, May 1966, pp.302-307
- [39] TAFLOVE, A. **Advances in Computational Electrodynamics – The Finite Difference Time Domain Method**. Artech House, 1998, ISBN 0-89006-834-8.
- [40] <http://www.zeland.com/>
- [41] OKONIEWSKI, M.; STUCHLY, M. A. A Study of the Handset Antenna and Human Body Interaction. **IEEE T-MTT**, v. 44, n. 10, p. 1855-1864, Oct. 1996.
- [42] WATANABE, S. Characteristics of the SAR Distributions in a Head Exposed to Electromagnetic Fields Radiated by a Hand-Held Portable Radio. **IEEE T-MTT**, v. 44, n. 10, p. 1874-1883, Oct. 1996.

- [43] DIMBYLOW, P. J.; GANDHI, O. P. Finite-Difference Time-Domain Calculations of SAR in a Realistic Heterogeneous Model of the Head for Plane-Wave Exposure from 600 MHz to 3 GHz. **Phys. Med. Biol.**, v. 36, p. 1075-1089, Aug. 1991.
- [44] SALLES, A. A.; FERNÁNDEZ, C. R.; BONADIMAN, M. Simulação da Taxa de Absorção na Cabeça do Usuário do Telefone Celular. **Anais do IX Simpósio Brasileiro de Microondas e Optoeletrônica**, João Pessoa – PB, Agosto, 2000.
- [45] SALLES, A. A.; FERNÁNDEZ, C. R.; BONADIMAN, M. Distância da Antena e Potência Absorvida na Cabeça do Usuário de Telefone Celular. **Anais do XVIII Simpósio Brasileiro de Telecomunicações**, Gramado – RS, Setembro, 2000.
- [46] TAFLOVE, A. **Computational Electrodynamics – the Finite Difference Time Domain Method**. Artech House, 1995, 600p. ISBN 0-89006-792-9.
- [47] BALANIS, C. **Antenna Theory Analysis and Design**. 2.ed., John Wiley & Sons, 1997, 941p. ISBN 0-471-59268-4.
- [48] CENELEC – Basic Standard for the Measurement of Specific Absorption Rate Related to Human Exposure to Electromagnetic Fields from Mobile Phones (300 MHz - 3 GHz). Rev. Final Draft, TC-211, April, 2000.
- [49] CHEN, H.; WANG, H. Current and SAR Induced in a Human Head Model by the Electromagnetic Fields Irradiated from a Cellular Phone. **IEEE Trans. on Microwave T&T**, v. 42, n. 12, Dec. 1994.
- [50] HOMBACH, V. The Dependence of EM Energy Absorption Upon Human Head Modeling at 900 Mhz. **IEEE Trans. on Microwave T&T**, v. 44, n. 10, Oct. 1996.
- [51] SCHIAVONI, A.; BERTOTTO, P.; RICHIARDI, G.; BIELLI, P. SAR Generated by Commercial Cellular Phones – Phone Modeling, and Measurements. **IEEE T-MTT**, v. 48, n. 11, Nov. 2000.
- [52] GEDNEY, S.D. et al. Perfect Matched Layer Media With CFS for an Unconditionally Stable ADI-FDTD Method. **IEEE Trans. on Antennas & Propagation**, v. 49, n. 11, Nov. 2001.
- [53] BÉRENGER, J. P. Perfectly Matched Layer for the FDTD Solution of Wave-Structure Interaction Problems. **IEEE Trans. on Antennas & Propagation**, v. 44, n. 1, Jan. 1996.
- [54] BÉRENGER, J. P. Improved PML for the FDTD Solution of Wave-Structure Interaction Problems. **IEEE T. Antennas & Prop.**, v. 45, n. 3, Mar. 1997.
- [55] CENTRO NACIONAL DE SUPERCOMPUTAÇÃO, CESUP/RS,  
<http://www.cesup.ufrgs.br>
- [56] JAMES, J. R.; HALL, P. S.; WOOD, C. **Microstrip Antenna: Theory and Design**. Peter Peregrinus, 1981, ISBN: 0-906048-57-5.



- [57] POZAR, D. M.; SCHAUBERT, D. H. **Microstrip Antennas - The Analysis and Design of Microstrip Antennas and Arrays**. IEEE Press, 1985, ISBN 0-7803-1078-0.
- [58] BARTHIA, P.; RAO, K. V. S.; TOMAR, R. S. **Millimeter-Wave Microstrip and Printed Circuit Antennas**. Artech House, 1990.
- [59] VAN DER PAUW, L. J. The Radiation of Electromagnetic Power by Microstrip Configurations. **IEEE Trans. On MTT**, v. 25, n. 9, p. 719-725, Sep. 1977.
- [60] BELOHOUBEK, E.; DENLINGER, E. Loss Considerations for Microstrip Resonators. **IEEE Trans. on MTT**, v. 23, p.522-526, June 1975.
- [61] JENSEN, M. A.; RAHMAT-SAMII, Y. The Electromagnetic Interaction Between Biological Tissue and Antennas on a Transceiver Handset. **IEEE MTT-S International Microwave Symposium**, p. 367-370, 1994.
- [62] RAHMAT-SAMII, Y. Novel Antennas for Personal Communications Including Biological Effects. **SBMO/IEEE MTT-S IMOC'95 Proceedings**, p. 295-308.
- [63] SANAD, M.; AMANT, N. H. An Internal Integrated Microstrip Antenna for PCS/Cellular Telephones and Other Hand-held Portable Communication Equipment. **Microwave Journal**, p. 64-77, July 1998.
- [64] DEAL, W. R.; QIAN, Y.; ITOH, T. Planar Integrated Antenna Technology. **Microwave Journal**, p. 128-144, July 1999.
- [65] SANAD, M.; HASSAN, N. Compact Internal Multiband Microstrip Antennas for Portable GPS, PCS, Cellular and Satellite Phones. **Microwave Journal**, p. 90-98, Aug. 1999.
- [66] BEZERRA, J. M.; JUNQUEIRA, C.; RICHARDS, F. P. Antenas de Faixa-Larga de Polarização Circular Impressas em Microfitas. **XVIII Simpósio Brasileiro de Telecomunicações**, Gramado – RS, Setembro, 2000.
- [67] MALONEY, J. G.; SMITH, G. S.; SCOTT, W. R. Accurate Computation of the Radiation from Simple Antennas Using the Finite-Difference Time-Domain Method. **IEEE Trans. On AP**, v. 38, n. 7, p. 1059-1068, July 1990.
- [68] SALLES, A. A.; VIEIRA, J.E.; FRANCO, P.R.; FERNÁNDEZ, C.; SCHÄFFER, L.A.; ALVES, D.M. Simulações e Medidas dos Campos Irradiados pelos Telefones Celulares Portáteis. **Anais do VIII Simpósio Brasileiro de Microondas e Optoeletrônica**, Joinville – SC, Agosto, 1998, pg 384-388.
- [69] SAINATI, R. A. **CAD of Microstrip Antennas for Wireless Applications**. Artech, 1996, 255p. ISBN 0-890-06-562-4.
- [70] JASIK, H. **Antenna Engineering Handbook**. Mc.Graw-Hill, 1961.

## ANEXOS

### ANEXO1 LISTAGEM DO PROGRAMA PRINCIPAL

```

/*****
*
* UFRGS - UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
* DELET - DEPARTAMENTO DE ENGENHARIA ELETRICA E ELETRONICA
* LACOM - LABORATORIO DE COMUNICACOES ELETRO-OPTICAS
*
* PROGRAMA PARA CALCULO DE CAMPO ELETROMAGNETICO E SAR
*
* VERSAO: 3.0 / 2001
*
* A. A. DE SALLES
* CLAUDIO FERNANDEZ
* MATEUS BONADIMAN
*
*****/

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <float.h>
#include <time.h>
#include <signal.h>
#include <string.h>

#define VERSAO "3.0"

// SETTINGS
//
*****
**
#define NUM_MAX_ITERACAO 900
#define INTERVALO 300 // INTERVALO PARA GRAVACAO DE DADOS
#define AVISO_GRAVACAO 100 // PARA CADA INTERVALO ELE MANDA UMA MENSAGEM
PARA A TELA (ANDAMENTO)

#define GRAVA_FRAME_SAR 1 // 1 PARA GRAVAR A SAR
#define GRAVA_FRAME_E 1 // 1 PARA GRAVAR CAMPOS ELETROMAGNETICOS NA
REGIAO DE ESTUDO
#define GRAVA_ANTENA 1 // 1 PARA GRAVAR CAMPOS ELETROMAGNETICOS NA
REGIAO DA ANTENA
#define GRAVA_BORDA 0 // 1 PARA GRAVAR A SAR E |E| COM A BORDA
ABSORVEDORA
#define GRAVA_TELA 0 // 1 PARA GRAVAR A MATRIZ CAB (PARA DEBUG)
#define GRAVA_PTO 0 // 1 PARA GRAVAR UM DETERMINADO PONTO AO LONGO
DO TEMPO
#define MATHLAB 0 // 1 PARA GRAVAR NO FORMATO MATHLAB
#define QUAL_MATERIAL 1 // 1 PARA CABECA E 0 PARA O AR

```

```

#define GRAVA_SAG          1 // 1 PARA GRAVAR CORTES SAGITAIS
#define GRAVA_FRO         1 // 1 PARA GRAVAR CORTES FRONTAIS
#define GRAVA_COR         1 // 1 PARA GRAVAR CORTES CORONAI
#define TIPODANTENA       "Monopolo de 1/4 de onda"

// CONSTANTES DE TAMANHO E DISTÂNCIAS
#define TAM_BORDAX      30
#define TAM_BORDAY      30
#define TAM_BORDAZ      30
#define TAM_AR1         128 // Ar antes da antena
#define TAM_ANTENA      1 // Largura da antena
#define TAM_AR2         50 // Ar entre a antena e a cabeca
#define TAM_DA_CAB      104 // Largura da cabeca
#define TAM_AR3         28 // Ar entre a cabeca e a borda

// CONSTANTES DO ARQUIVO COM A MATRIZ DE ELEMENTOS (.FD)
#define ny 225 // LINHAS
#define nx 256 // COLUNAS
#define nz (2*TAM_BORDAZ)+TAM_AR1+TAM_ANTENA+TAM_DA_CAB+TAM_AR2+TAM_AR3

//
*****

// CONSTANTES USADAS EM TODO O PROGRAMA
#define FREQUENCIA      (850e6) // FREQUENCIA DA FONTE: 850 MHz
#define LIGHT_SPEED     (299792458.0)
#define MU_0             (1.2566370614359172953850573533118e-6)
#define EPS_0            (8.8541878176203898505365630317164e-12)
#define UM               (1.0)
#define DOIS             (2.0)

// CONSTANTES PARA CALCULO TERMICO
#define HA              10.5
#define HB              50
#define TAMB            20
#define TSANGUE        37

// CONSTANTES PARA DEFINICAO ESPACIAL

#define AR              0
#define PELE            1
#define MUSCULO         2
#define CEREBRO        3
#define OSSO            4
#define OLHO            5
#define BORDAX          6
#define BORDAY          7
#define BORDAZ          8
#define CANTO           9

#define METAL           10
#define TERRA           11
#define EXCITACAO      12
#define CAMPOX          13
#define CAMPOY          14
#define CAMPOZ          15
#define CAMPOSXY       16
#define CAMPOSXZ       17
#define CAMPOSYZ       18
#define TODCAMPOS      19

```

```
// CONSTANTES PARA A MATRIZ DE TECIDOS

#define MU      0
#define RO      1
#define SIGX    2
#define SIGY    3
#define SIGZ    4
#define EPS     5
#define MAGSIGX 6
#define MAGSIGY 7
#define MAGSIGZ 8
#define CALORESP 9
#define CONDUTERM 10
#define SANGUE  11

// CONSTANTES PARA A BORDA ABSORVEDORA

#define IMPED_O  377
#define MUDIVEPS (141925.729082)
#define TEMP_SIG (0.702)
#define TEMP_SIGM (TEMP_SIG*MUDIVEPS)

// CONSTANTES PROPOSTAS ESTATISTICAMENTE POR BERENGER

#define ALPHA    1.5851
#define BETA     -1.0156
#define GAMA     0.18495
#define M_REFL   0.01          // Maxima Reflexao aceita na PLM

// CONSTANTES PARA MEDIDAS

#define MEDE_VELOC 1          // MEDE A VELOCIDADE DA ONDA
#define DELTA_DIST 50        // PIXELS DE DISTÂNCIA ENTRE OS PONTOS 1 E 2
#define RAO_POT    100       // 0 NAO CALCULA -> RAO PARA CALCULO DE POT.
                                IRRADIADA

#ifdef _CRAYC
    #pragma _CRI bounds
#endif

// MATRIZ DE TECIDOS E BORDA

double material[20][12]=
/*
      mu      ro      sigx      sigy      sigz
eps     magsix  magsiy  magsiz  caloresp  conduterm  sangue*/

/* AR */
EPS_0,      0.0,      0.0,      0.0,      0.0,      0.0,      0.0},
      {MU_0, 0.0013,      0.0,      0.0,      0.0,
/* PELE */
34.5000*EPS_0,      0.0,      0.0,      0.0,      0.0,      0.0,      0.50,
8652},
      {MU_0, 1.1000,      0.6000,      0.6000,      0.6000,
/* MUSCULO */
58.5000*EPS_0,      0.0,      0.0,      0.0,      0.0,      3500,      0.60,
3488},
```

```

/* CEREBRO */      {MU_0, 1.0300, 1.2300, 1.2300, 1.2300,
55.0000*EPS_0,    0.0, 0.0, 0.0, 3500, 0.60,
37822},
/* OSSO */        {MU_0, 1.8500, 0.1050, 0.1050, 0.1050,
8.0000*EPS_0,    0.0, 0.0, 0.0, 1300, 0.30,
1401},

/* OLHO */        {MU_0, 1.0200, 1.6800, 1.6800, 1.6800,
67.9000*EPS_0,  0.0, 0.0, 0.0, 3900, 0.50,
0.0},

/* BORDAX */      {MU_0, 1.0, TEMP_SIG, 0.0,
0.0, EPS_0, TEMP_SIGM, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0},

/* BORDAY */      {MU_0, 1.0, 0.0, TEMP_SIG, 0.0,
EPS_0, 0.0, TEMP_SIGM, 0.0, 0.0, 0.0,
0.0},

/* BORDAZ */      {MU_0, 1.0, 0.0, 0.0,
TEMP_SIG, EPS_0, 0.0, 0.0, TEMP_SIGM,
0.0, 0.0, 0.0},

/* CANTO */       {MU_0, 1.0, TEMP_SIG, TEMP_SIG, TEMP_SIG,
EPS_0, TEMP_SIGM, TEMP_SIGM, TEMP_SIGM, 0.0, 0.0, 0.0},
/* METAL */       {MU_0, 1.0, 5.8e7, 5.8e7, 5.8e7,
EPS_0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0},

/* TERRA */       {MU_0, 1.0, 5.8e7, 5.8e7, 5.8e7,
EPS_0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0},

/* EXCITACAO */   {MU_0, 0.0013, 0.0, 0.0, 0.0,
EPS_0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0},

/* CAMPOX */      {MU_0, 1.0, 5.8e7, 5.8e7, 5.8e7,
EPS_0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0},

/* CAMPOY */      {MU_0, 1.0, 5.8e7, 5.8e7, 5.8e7,
EPS_0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0},

/* CAMPOZ */      {MU_0, 1.0, 5.8e7, 5.8e7, 5.8e7,
EPS_0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0},

/* CAMPOSXY */    {MU_0, 1.0, 5.8e7, 5.8e7, 5.8e7,
EPS_0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0},

/* CAMPOSXZ */    {MU_0, 1.0, 5.8e7, 5.8e7, 5.8e7,
EPS_0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0},

/* CAMPOSYZ */    {MU_0, 1.0, 5.8e7, 5.8e7, 5.8e7,
EPS_0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0},

```

```

/* TODCAMPOS */ {MU_0, 1.0, 5.8e7, 5.8e7, 5.8e7,
EPS_0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0}};

// DECLARACAO DE FUNCOES

void Gera_bordaz(int k);
void Gera_matriz_ar(int k);
void Gera_antenal(int k, int);
void faz_calc_hx(double, double, double, double, double, double, double, double
*atingiu_borda, double *degringolou);
void faz_calc_hy(double, double, double, double, double, double, double, double
*atingiu_borda, double *degringolou);
void faz_calc_hz(double, double, double, double, double, double, double, double
*atingiu_borda, double *degringolou);
void faz_calc_ex(double, double, double, double, double, double, double, double
*atingiu_borda, double *degringolou);
void faz_calc_ey(double, double, double, double, double, double, double, double
*atingiu_borda, double *degringolou);
void faz_calc_ez(double dx, double dy, double dz, double dt, double
*cpopele, double *cpomusc, double *cposso, double *cpoolho, double
*cpocereb, double *cpoar, double berenger_g, double sigma_zero, double,
double *atingiu_borda, double *degringolou);

void grava_tela_cab(int k);
void grava_rel_sar(double, double, double, double, double, double, double,
double, double, double, double, int, double, double, double, double, int, double,
double, int, int, double, double);
void signal_handler_fpu(int *p);
void signal_handler_sigsev(int *p);
void erro_alocacao(void);

int calcula_memoria_req(void);
unsigned int aloca_memoria(void);

void grava_sar_sag(int, int, int, int);
void grava_sar_fro(int, int, int, int, int, int);
void grava_sar_cor(int, int, int, int, int, int);
void grava_cpo_sag(int, int, int, int);
void grava_cpo_fro(int, int, int, int, int, int);
void grava_cpo_cor(int, int, int, int, int, int);

// DECLARACAO DE PONTEIROS

double ***ex, ***ey, ***ez;
double ***hx, ***hy, ***hz;
double ***vsar;

time_t horacom,horaini;

// MATRIZ DA ANTENA (DEVE TER MENOS QUE 100 PONTOS, POIS AQUI DECLARO COM
100)

int antena[100][3];

// DECLARACAO DE MATRIZES

int cab[nx+2*TAM_BORDERDAY][ny+2*TAM_BORDERDAX][nz];
double guarda_cpo[91][3];

// *****

```

```

// * INICIO DO ALGORITMO PRINCIPAL *
// *****

void main(int argc, char *argv[])
{
    int
i, j, k, inter, bytesalocados, contant, nro_fontes, auxi, auxj, auxk, ctagrav,
gravabx, gravaby, nmax, soma_iter_av, m_int_pto1, m_int_pto2, m_aux_zer;

    unsigned int interacao, interacoes;

    double
TempoAtualSimulado, TempoTotalSimulado, omega, dx, dy, dz, dt, diff_time, fonte,
cpopele, cpomusc, cpoosso, cpoolho, cpocereb, cpoar, log_g, berenger_g, sigma_zero,
nro_cel_borda, atingiu_borda, degringolou, nro_iterac, m_val_pto1, m_val_pto2,
    m_ant_pto1, m_ant_pto2, m_aux_pot;

    char nomarq[20], nompto[20];

    time_t time1, time2;

    FILE *arqppto, *arqppt;

#ifdef __WIN32__
    if(signal(SIGFPE, signal_handler_fpu) == SIG_ERR)
        printf("Erro Signal sigfpe\n");
    if(signal(SIGSEGV, signal_handler_sigsev) == SIG_ERR)
        printf("Erro Signal sigsegv\n");
#endif

    interacoes = NUM_MAX_ITERACAO;

    omega = DOIS*M_PI*FREQUENCIA;

    // VARIAVEIS PARA RELATORIO

    cpopele=0;
    cpomusc=0;
    cpoosso=0;
    cpoolho=0;
    cpocereb=0;
    cpoar=0;
    m_aux_zer=0;
    nro_iterac=0.0;
    degringolou=0.0;
    m_val_pto1=0.0;
    m_val_pto2=0.0;
    m_ant_pto1=0.0;
    m_ant_pto2=0.0;
    m_int_pto1=0;
    m_int_pto2=0;
    soma_iter_av=0;
    atingiu_borda=0.0;
    m_aux_pot=0.0;

    // VARIAVEIS PARA MAGIC-STEP

    dx = 9e-4; debug

```

```

dy = 9e-4;
dz = 9e-4;

// dt PARA MAGIC-STEP
dt = dx/(LIGHT_SPEED*sqrt(3));

fonte=0.0;

TempoTotalSimulado = interacoes*dt;
bytesalocados      = calcula_memoria_req();
bytesalocados      = aloca_memoria();

// CALCULOS PARA A BORDA DE BERENGER
if ( nx>ny && nx>(nz-(2*TAM_BORDAZ)) )
    nmax = nx;
else if ( ny>nx && ny>(nz-(2*TAM_BORDAZ)) )
    nmax = ny;
else if ( (nz-(2*TAM_BORDAZ))>nx && (nz-(2*TAM_BORDAZ))>ny )
    nmax = (nz-(2*TAM_BORDAZ));
else
    nmax = nx;

log_g = ALPHA+BETA*log10(nmax)+GAMA*pow(log10(nmax),2.0);
berenger_g = pow(10.0,log_g);

// Tamanho da borda devera ser SEMPRE maior que nro_cel_borda
nro_cel_borda = (1/log(berenger_g))*log(1-(sqrt(berenger_g)-
1)*log10(M_REFL)*ny*100000/(2*M_PI));

sigma_zero = -
((EPS_0*LIGHT_SPEED)/(2*dx))*log(berenger_g)/(pow(berenger_g,nmax)-
1))*log(M_REFL);

// IMPRIME CABECALHO NA TELA

fprintf(stdout,"*****
\n");
fprintf(stdout,"UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL (UFRGS)
\n");
fprintf(stdout,"  DEPARTAMENTO DE ENGENHARIA ELETRICA (DELET)
\n");
fprintf(stdout,"LABORATORIO DE COMUNICACOES ELETRO-OPTICAS (LACOM)
\n");
fprintf(stdout,"  SIMULADOR FDTD VERSAO %s - UFRGS - 2001
\n",VERSAO);
fprintf(stdout,"*****
\n\n");

if (QUAL_MATERIAL)
    fprintf(stdout, "Material em Estudo...: CABEÇA\n");
else
    fprintf(stdout, "Material em Estudo...: AR\n");

fprintf(stdout, "Número de Iterações..: %d\n",NUM_MAX_ITERACAO);
fprintf(stdout, "Intervalo de Gravação: %d\n",INTERVALO);
fprintf(stdout, "Memoria Necessaria...: %f MBytes\n",
((bytesalocados/1000000.0)+100));
fprintf(stdout, "Antena Utilizada.....: %s\n",TIPODANTENA);
fprintf(stdout, "Frequencia.....: %lg Hz\n",FREQUENCIA);

```



```

    fprintf(stdout, "Comprimento de Onda...: %lg
m\n",LIGHT_SPEED/FREQUENCIA);
    fprintf(stdout, "Distância Total.....: %lg m ou %lg
comprimento(s) de onda\n"
, LIGHT_SPEED*dt*NUM_MAX_ITERACAO
, ((LIGHT_SPEED*dt*NUM_MAX_ITERACAO)/(LIGHT_SPEED/FREQUENCIA)));

    fprintf(stdout, "%dx%dx%d celulas\ndx=%lg metros\ndy=%lg
metros\ndz=%lg\
metros \ndt=%lg segundos\n",nx+2*TAM_BORDAY, ny+2*TAM_BORDAX, nz, dx, dy,
dz,dt);

    if (GRAVA_FRAME_SAR)
        fprintf(stdout, "Grava a SAR a cada %d iteracoes, %lg
segundos\n",INTERVALO,INTERVALO*dt);
    if (GRAVA_FRAME_E)
        fprintf(stdout, "Grava o |E| a cada %d iteracoes, %lg
segundos\n",INTERVALO,INTERVALO*dt);
    if (GRAVA_ANTENA)
        fprintf(stdout, "Grava o |E| da ANTENA a cada %d iteracoes, %lg
segundos\n",INTERVALO,INTERVALO*dt);

    if (QUAL_MATERIAL)
        fprintf(stdout, "Distância da Antena a Cabeça = %lg
cm\n",TAM_AR2*dx*100);

    fprintf(stdout,"*****
\n\n");
    fflush(stdout);

    auxi = TAM_BORDAY+128;
    auxj = TAM_BORDAX+ny-3;
    auxk = TAM_BORDAZ+TAM_AR1;

    for (k = 0 ; k <= 90 ; k++)
    {
        i = ( auxi + (int)RAIO_POT*cos(k) );
        j = ( auxj - (int)RAIO_POT*sin(k) );

        fprintf(stdout, "i=%d j=%d k=%d\n",i,j,auxk);
    }
    fflush(stdout);

    contant= 1;
    inter = 0;

    // CARREGA OS ARQUIVOS PARA AS MATRIZES

    printf("Carregando Matrizes... \n");
    fflush(stdout);

    // CARREGA BORDA ABSORVEDORA NA DIMENCAO Z
    for( k = 0 ; k < TAM_BORDAZ ; k ++ )

```

```

        Gera_bordaz(k);

// CARREGA MATRIZ DE AR
for( k = TAM_BORDAZ ; k < TAM_BORDAZ+TAM_AR1 ; k ++ )
    Gera_matriz_ar(k);

printf("Carregando Antena... \n");
fflush(stdout);

// CARREGA ANTENA
for( k = TAM_BORDAZ+TAM_AR1 ; k < TAM_BORDAZ+TAM_AR1+TAM_ANTENA ; k
++ )
//    Gera_antena(k, contant++);
    Gera_matriz_ar(k);

// CARREGA MATRIZ DE AR
for( k = TAM_BORDAZ+TAM_AR1+TAM_ANTENA ; k <
TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2 ; k ++ )
    Gera_matriz_ar(k);

if (QUAL_MATERIAL)
{
    // CARREGA MATRIZ DA CABECA
    contant= 1;
    for( k = TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2 ; k <
TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB/2 ; k ++ )
        Gera_cabeca(k, contant++);
    contant= TAM_DA_CAB/2;
    for( k = TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB/2 ; k
< TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB ; k ++ )
        Gera_cabeca(k, contant--);
}
else
{
    // CARREGA AR
    for( k = TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2 ; k <
TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB ; k ++ )
        Gera_matriz_ar(k);
}

// CARREGA MATRIZ DE AR
for( k = TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB ; k <
TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB+TAM_AR3 ; k ++ )
    Gera_matriz_ar(k);

// CARREGA BORDA ABSORVEDORA NA DIMENCAO Z
for( k = TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB+TAM_AR3 ; k
< TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB+TAM_AR3+TAM_BORDAZ ; k
++ )

    Gera_bordaz(k);

// CRIA UM PLANO TERRA EM TODA A EXTENSAO DO DOMINIO
for( k = TAM_BORDAZ ; k < nz-TAM_BORDAZ ; k ++ )
{
    for( i = TAM_BORDAZ ; i<nx+TAM_BORDAZ ; i++)
    {
        cab[i][TAM_BORDAZ+ny-2][k]=TERRA;
    }
}

// "DESENHA" A ANTENA NO ESPACO

```

```

cab[TAM_BORDAY+128][TAM_BORDAX+ny-3][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-4][TAM_BORDAZ+TAM_AR1]=TODCAMPOS;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-5][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-6][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-7][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-8][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-9][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-10][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-11][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-12][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-13][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-14][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-15][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-16][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-17][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-18][TAM_BORDAZ+TAM_AR1]=CAMPOSXZ;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-19][TAM_BORDAZ+TAM_AR1]=TODCAMPOS;

cab[TAM_BORDAY+128][TAM_BORDAX+ny-3][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-4][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-5][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-6][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-7][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-8][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-9][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-10][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-11][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-12][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-13][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-14][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-15][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-16][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-17][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-18][TAM_BORDAZ+TAM_AR1]=EXCITACAO;
cab[TAM_BORDAY+128][TAM_BORDAX+ny-19][TAM_BORDAZ+TAM_AR1]=EXCITACAO;

printf("Carga de Matrizes Finalizada... \n");
fflush(stdout);

if (GRAVA_TELA)
{
    grava_tela_cab(TAM_BORDAZ+TAM_AR1);
    grava_tela_cab(2);
    for( k = 0 ; k < 4 ; k ++ )
        grava_tela_cab(k);
}

// INICIA MATRIZ COM A POSICAO PARA A FONTE (ANTENA)

printf("Iniciando posicionamento da fonte... \n");
fflush(stdout);

nro_fontes=0;

for( k = TAM_BORDAZ+TAM_AR1 ; k < TAM_BORDAZ+TAM_AR1+TAM_ANTENA ; k
++ )
{
    for( j = 0 ; j<2*TAM_BORDAX+ny ; j++)
    {
        for( i =0 ; i<2*TAM_BORDAY+nx ; i++)
        {

```

```

        if ( cab[i][j][k] == EXCITACAO )
        {
            antena[nro_fontes][0]=i;
            antena[nro_fontes][1]=j;
            antena[nro_fontes][2]=k;
            nro_fontes++;
        }
    }
}

// CALCULO FDTD

printf("Iniciando Calculo FDTD... \n");
fflush(stdout);

if (GRAVA_PTO)
{
    sprintf(nomarq,"campo_pto.txt");
    arqpto=fopen(nomarq,"w+");
}

if (RAIO_POT > 0 )
{
    sprintf(nompto,"potência_cpo.txt");
    arqpot=fopen(nompto,"w+");
}

time1 = time(NULL);
horacom = time(NULL);
time(&horaini);

for(interacao = 0; interacao < interacoes; interacao++)
{
    TempoAtualSimulado = dt*(double)interacao;
    nro_iterac++;

    if((((interacao % AVISO_GRAVACAO) == 0 ) && interacao))
    {
        soma_iter_av += AVISO_GRAVACAO;
        printf("Já se Passaram %d Iterações! \n",soma_iter_av);
        fflush(stdout);
    }

    // GRAVA ARQUIVOS NAS ITERACOES DESEJADAS
    if((((interacao % INTERVALO) == 0 ) && interacao))
    {
        inter++;
        if (GRAVA_FRAME_SAR)
            fprintf(stdout, "Gravando a SAR com um total de %d
iteracoes.\n",interacao);
        if (GRAVA_FRAME_E)
            fprintf(stdout, "Gravando o |E| com um total de %d
iteracoes.\n",interacao);
        if (GRAVA_ANTENA)
            fprintf(stdout, "Gravando a ANTENA com um total de
%d iteracoes.\n",interacao);
        fflush(stdout);

        if (GRAVA_ANTENA)
        {

```

```

ctagrav = 0;

if (GRAVA_SAG)
{
    // GRAVA TODAS AS IMAGENS SAGITAIS
    /*
    TAM_BORDAZ+TAM_AR1+TAM_ANTENA ; k ++ )
    {
        ctagrav++;
        grava_cpo_sag(inter,ctagrav,k,0);
    }*/

    // GRAVA AS IMAGENS SAGITAIS SELECIONADAS
    grava_cpo_sag(inter,1,TAM_BORDAZ+TAM_AR1-
100,0);
    grava_cpo_sag(inter,2,TAM_BORDAZ+TAM_AR1-
50,0);
    grava_cpo_sag(inter,3,TAM_BORDAZ+TAM_AR1-
25,0);

    grava_cpo_sag(inter,4,TAM_BORDAZ+TAM_AR1,0);
    grava_cpo_sag(inter,5,TAM_BORDAZ+TAM_AR1+2,0);

    grava_cpo_sag(inter,6,TAM_BORDAZ+TAM_AR1+10,0);

    grava_cpo_sag(inter,7,TAM_BORDAZ+TAM_AR1+35,0);
}

ctagrav = 0;
gravaby = 0;

if (GRAVA_BORDA)
    gravaby = TAM_BORDAY;

if (GRAVA_FRO)
{
    // GRAVA AS IMAGENS FRONTAIS
    for( i = TAM_BORDAY-gravaby ; i <
nx+TAM_BORDAY+gravaby ; i++)
    {
        ctagrav++;

        grava_cpo_fro(inter,ctagrav,i,TAM_BORDAZ+TAM_AR1,TAM_BORDAZ+TAM_AR1+T
AM_ANTENA,0);
    }
}

ctagrav = 0;
gravabx = 0;

if (GRAVA_BORDA)
    gravabx = TAM_BORDAX;

if (GRAVA_COR)
{
    // GRAVA TODAS AS IMAGENS CORONAIS
    /*
    ny+TAM_BORDAX+gravabx ; j++)
    {
        ctagrav++;

```

```

        grava_cpo_cor(inter,ctagrav,j,TAM_BORDAZ+TAM_AR1,TAM_BORDAZ+TAM_AR1+T
AM_ANTENA,0);
        }*/

        // GRAVA AS IMAGENS CORONAIS SELECIONADAS

        grava_cpo_cor(inter,1,TAM_BORDAX+28,TAM_BORDAZ,nz-TAM_BORDAZ,0);
        grava_cpo_cor(inter,2,TAM_BORDAX+98,TAM_BORDAZ,nz-TAM_BORDAZ,0);
        grava_cpo_cor(inter,3,TAM_BORDAX+128,TAM_BORDAZ,nz-TAM_BORDAZ,0);
        grava_cpo_cor(inter,4,TAM_BORDAX+128+8,TAM_BORDAZ,nz-TAM_BORDAZ,0);
        grava_cpo_cor(inter,5,TAM_BORDAX+128+16,TAM_BORDAZ,nz-TAM_BORDAZ,0);
        }
    }

    ctagrav = 0;

    // GRAVA AS IMAGENS SAGITAIS
    for( k = TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2 ; k <
TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB ; k ++ )
    {
        ctagrav++;
        if (QUAL_MATERIAL)
        {
            // GRAVA A SAR DA CABECA
            if (GRAVA_FRAME_SAR && GRAVA_SAG)

            grava_sar_sag(inter,ctagrav,k,interacao+1);
        }
        // GRAVA O MODULO DO CAMPO ELETROMAGNETICO
        if (GRAVA_FRAME_E && GRAVA_SAG)
            grava_cpo_sag(inter,ctagrav,k,1);
    }

    ctagrav = 0;
    gravaby = 0;

    if (GRAVA_BORDA)
        gravaby = TAM_BORDAY;

    // GRAVA AS IMAGENS FRONTAIS
    for( i = TAM_BORDAY-gravaby ; i < nx+TAM_BORDAY+gravaby ;
i++)
    {
        ctagrav++;
        if (QUAL_MATERIAL)
        {
            // GRAVA A SAR DA CABECA
            if (GRAVA_FRAME_SAR && GRAVA_FRO)

            grava_sar_fro(inter,ctagrav,interacao+1,i,TAM_BORDAZ+TAM_AR1+TAM_ANTE
NA+TAM_AR2,TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB);
        }
        // GRAVA O MODULO DO CAMPO ELETROMAGNETICO
        if (GRAVA_FRAME_E && GRAVA_FRO)
    }

```

```

        grava_cpo_fro(inter,ctagrav,i,TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2,T
AM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB,1);
    }

    ctagrav = 0;
    gravabx = 0;

    if (GRAVA_BORDA)
        gravabx = TAM_BORDAX;

    // GRAVA AS IMAGENS CORONAIS
    for( j = TAM_BORDAX-gravabx ; j < ny+TAM_BORDAX+gravabx ;
j++)
    {
        ctagrav++;
        if (QUAL_MATERIAL)
        {
            // GRAVA A SAR DA CABECA
            if (GRAVA_FRAME_SAR || GRAVA_COR)

                grava_sar_cor(inter,ctagrav,interacao+1,j,TAM_BORDAZ+TAM_AR1+TAM_ANTE
NA+TAM_AR2,TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB);
        }
        // GRAVA O MODULO DO CAMPO ELETROMAGNETICO
        if (GRAVA_FRAME_E && GRAVA_COR)

            grava_cpo_cor(inter,ctagrav,j,TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2,T
AM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB,1);
    }

}

// ATUALIZACAO DA ANTENA

fonte = sqrt(2)*sin(omega*TempoAtualSimulado);

for ( i=0 ; i<nro_fontes ; i++)
{
    auxi=antena[i][0];
    auxj=antena[i][1];
    auxk=antena[i][2];
    ey[auxi][auxj][auxk] = fonte;
}

// CALCULA OS CAMPOS MAGNETICOS

faz_calc_hx(dy,dz,dt,berenger_g,sigma_zero,nro_iterac,&atingiu_borda,
&degringolou);

faz_calc_hy(dx,dz,dt,berenger_g,sigma_zero,nro_iterac,&atingiu_borda,
&degringolou);

faz_calc_hz(dy,dx,dt,berenger_g,sigma_zero,nro_iterac,&atingiu_borda,
&degringolou);

// CALCULA OS CAMPOS ELETRICOS

faz_calc_ex(dy,dz,dt,berenger_g,sigma_zero,nro_iterac,&atingiu_borda,
&degringolou);

```

```

    faz_calc_ey(dx,dz,dt,berenger_g,sigma_zero,nro_iterac,&atingiu_borda,
&degringolou);

    faz_calc_ez(dx,dy,dz,dt,&cpopele,&cpomusc,&cposso,&cpoolho,&cpocereb
,&cpoar,

berenger_g,sigma_zero,nro_iterac,&atingiu_borda,&degringolou);

    if (MEDE_VELOC)
    {
        if (m_int_pto1 == 0 || interacao >= 50)
        {
            auxi = TAM_BORDAY+128;
            auxj = TAM_BORDAX+ny-11;
            auxk = TAM_BORDAZ+TAM_AR1+50;

            m_val_pto1 = sqrt(
(ex[auxi][auxj][auxk]*ex[auxi][auxj][auxk])+
(ey[auxi][auxj][auxk]*ey[auxi][auxj][auxk])+
(ez[auxi][auxj][auxk]*ez[auxi][auxj][auxk]));

            if ( m_val_pto1 < m_ant_pto1 )
            {
                m_int_pto1 = interacao-1;
                m_val_pto1 = m_ant_pto1;
            }
            else
                m_ant_pto1 = m_val_pto1;
        }

        if (m_int_pto2 == 0 || interacao >= 100)
        {
            auxi = TAM_BORDAY+128;
            auxj = TAM_BORDAX+ny-11;
            auxk = TAM_BORDAZ+TAM_AR1+100;

            m_val_pto2 = sqrt(
(ex[auxi][auxj][auxk]*ex[auxi][auxj][auxk])+
(ey[auxi][auxj][auxk]*ey[auxi][auxj][auxk])+
(ez[auxi][auxj][auxk]*ez[auxi][auxj][auxk]));

            if ( m_val_pto2 < m_ant_pto2 )
            {
                m_int_pto2 = interacao-1;
                m_val_pto2 = m_ant_pto2;
            }
            else
                m_ant_pto2 = m_val_pto2;
        }
    }

    if (RAIO_POT != 0)
    {
        if ( interacao > 100 )
        {
            // PONTOS DO CENTRO DA ESFERA

```



```

auxi = TAM_BORDAY+128;
auxj = TAM_BORDAX+ny-3;
auxk = TAM_BORDAZ+TAM_AR1;

for (k = 0 ; k <= 90 ; k++)
{
    i = ( auxi + (int)RAIO_POT*cos(k) );
    j = ( auxj - (int)RAIO_POT*sin(k) );

    m_aux_pot = sqrt(
(ex[i][j][auxk]*ex[i][j][auxk])+(ey[i][j][auxk]*ey[i][j][auxk])+
(ez[i][j][auxk]*ez[i][j][auxk]) );

    guarda_cpo[k][0] += m_aux_pot;

    if ( guarda_cpo[k][1] < m_aux_pot )
        guarda_cpo[k][1] = m_aux_pot;

    guarda_cpo[k][2] += sqrt(
(ex[i][j][auxk]*ex[i][j][auxk])+
(ey[i][j][auxk]*ey[i][j][auxk]) );
    }
}

if (GRAVA_PTO)
{
    fprintf(arqpto,"%d; ",interacao);

    auxi = TAM_BORDAY+128;
    auxj = TAM_BORDAX+ny-11;
    auxk = TAM_BORDAZ+TAM_AR1+50;
    fprintf(arqpto,"%%.4e;
",sqrt((ex[auxi][auxj][auxk]*ex[auxi][auxj][auxk])+
(ey[auxi][auxj][auxk]*ey[auxi][auxj][auxk])+(ez[auxi][auxj][auxk]*
ez[auxi][auxj][auxk])));

    auxi = TAM_BORDAY+128;
    auxj = TAM_BORDAX+ny-11;
    auxk = TAM_BORDAZ+TAM_AR1+100;
    fprintf(arqpto,"%%.4e;
",sqrt((ex[auxi][auxj][auxk]*ex[auxi][auxj][auxk])+
(ey[auxi][auxj][auxk]*ey[auxi][auxj][auxk])+(ez[auxi][auxj][auxk]*
ez[auxi][auxj][auxk])));

    auxi = TAM_BORDAY+178;
    auxj = TAM_BORDAX+ny-11;
    auxk = TAM_BORDAZ+TAM_AR1;
    fprintf(arqpto,"%%.4e;
",sqrt((ex[auxi][auxj][auxk]*ex[auxi][auxj][auxk])+
(ey[auxi][auxj][auxk]*ey[auxi][auxj][auxk])+(ez[auxi][auxj][auxk]*
ez[auxi][auxj][auxk])));

    auxi = TAM_BORDAY+228;

```

```

        auxj = TAM_BORDAX+ny-11;
        auxk = TAM_BORDAZ+TAM_AR1;
        fprintf(arqpto, "%.4e;
", sqrt((ex[auxi][auxj][auxk]*ex[auxi][auxj][auxk])+
(ey[auxi][auxj][auxk]*ey[auxi][auxj][auxk])+(ez[auxi][auxj][auxk]*
ez[auxi][auxj][auxk])));

        fprintf(arqpto, "\n");
    }

}

// *****
// FIM DOS CALCULOS
// INICIO DAS GRAVAÇÕES FINAIS
// *****

if (GRAVA_PTO)
{
    fprintf(stdout, "Salvando Arquivo com a Tabela de Campos nos
Pontos Escolhidos.\n");
    fclose(arqpto);
}

if (RAIO_POT > 0 )
{
    fprintf(stdout, "Salvando Arquivo com a Tabela de Campos com
Angulos de 0 a 90.\n");

    for (i = 0 ; i <= 90 ; i++)
    {
        fprintf(arqpot, "%d; ", i);

        fprintf(arqpot, "%.4e;
", guarda_cpo[i][0]/(NUM_MAX_ITERACAO-100));

        fprintf(arqpot, "%.4e; ", guarda_cpo[i][1]);

        fprintf(arqpot, "%.4e;
", guarda_cpo[i][2]/(NUM_MAX_ITERACAO-100));

        fprintf(arqpot, "\n");
    }
    fclose(arqpot);
}

// GRAVANDO ARQUIVOS FINAIS

if (GRAVA_FRAME_SAR)
    fprintf(stdout, "Gravando SAR FINAL com %lg segundos de
simulacao.\n", TempoAtualSimulado);
if (GRAVA_FRAME_E)
    fprintf(stdout, "Gravando |E| FINAL com %lg segundos de
simulacao.\n", TempoAtualSimulado);
if (GRAVA_ANTENA)
    fprintf(stdout, "Gravando ANTENA FINAL com %lg segundos de
simulacao.\n", TempoAtualSimulado);

fflush(stdout);

```

```

if (GRAVA_ANTENA)
{
    ctagrav = 0;

    if (GRAVA_SAG)
    {
        // GRAVA TODAS AS IMAGENS SAGITAIS
/*
        for( k = TAM_BORDAZ+TAM_AR1 ; k <
TAM_BORDAZ+TAM_AR1+TAM_ANTENA ; k ++ )
        {
            ctagrav++;
            grava_cpo_sag(99,ctagrav,k,0);
        }*/

        // GRAVA AS IMAGENS SAGITAIS SELECIONADAS
        grava_cpo_sag(99,1,TAM_BORDAZ+TAM_AR1-100,0);
        grava_cpo_sag(99,2,TAM_BORDAZ+TAM_AR1-50,0);
        grava_cpo_sag(99,3,TAM_BORDAZ+TAM_AR1-25,0);
        grava_cpo_sag(99,4,TAM_BORDAZ+TAM_AR1,0);
        grava_cpo_sag(99,5,TAM_BORDAZ+TAM_AR1+2,0);
        grava_cpo_sag(99,6,TAM_BORDAZ+TAM_AR1+10,0);
        grava_cpo_sag(99,7,TAM_BORDAZ+TAM_AR1+35,0);
    }

    ctagrav = 0;
    gravaby = 0;

    if (GRAVA_BORDA)
        gravaby = TAM_BORDAY;

    if (GRAVA_FRO)
    {
        // GRAVA AS IMAGENS FRONTAIS
        for( i = TAM_BORDAY-gravaby ; i < nx+TAM_BORDAY+gravaby ;
i++)
        {
            ctagrav++;

            grava_cpo_fro(99,ctagrav,i,TAM_BORDAZ+TAM_AR1,TAM_BORDAZ+TAM_AR1+TAM_
ANTENA,0);
        }
    }

    ctagrav = 0;
    gravabx = 0;

    if (GRAVA_BORDA)
        gravabx = TAM_BORDAX;

    if (GRAVA_COR)
    {
        // GRAVA TODAS AS IMAGENS CORONAIS
/*
        for( j = TAM_BORDAX-gravabx ; j < ny+TAM_BORDAX+gravabx ;
j++)
        {
            ctagrav++;

            grava_cpo_cor(99,ctagrav,j,TAM_BORDAZ+TAM_AR1,TAM_BORDAZ+TAM_AR1+TAM_
ANTENA,0);
        }*/
    }
}

```

```

// GRAVA AS IMAGENS CORONAIS SELECIONADAS
grava_cpo_cor(99,1,TAM_BORDAX+28,TAM_BORDAZ,nz-
TAM_BORDAZ,0);
grava_cpo_cor(99,2,TAM_BORDAX+98,TAM_BORDAZ,nz-
TAM_BORDAZ,0);
grava_cpo_cor(99,3,TAM_BORDAX+128,TAM_BORDAZ,nz-
TAM_BORDAZ,0);
grava_cpo_cor(99,4,TAM_BORDAX+128+8,TAM_BORDAZ,nz-
TAM_BORDAZ,0);
grava_cpo_cor(99,5,TAM_BORDAX+128+16,TAM_BORDAZ,nz-
TAM_BORDAZ,0);
}
}

ctagrav = 0;

// GRAVA AS IMAGENS SAGITAIS
for( k = TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2 ; k <
TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB ; k ++ )
{
    ctagrav++;
    if (QUAL_MATERIAL)
    {
        // GRAVA A SAR DA CABECA
        if (GRAVA_FRAME_SAR && GRAVA_SAG)
            grava_sar_sag(99,ctagrav,k,NUM_MAX_ITERACAO);
    }
    // GRAVA O MODULO DO CAMPO ELETROMAGNETICO
    if (GRAVA_FRAME_E && GRAVA_SAG)
        grava_cpo_sag(99,ctagrav,k,1);
}

ctagrav = 0;
gravaby = 0;

if (GRAVA_BORDA)
    gravaby = TAM_BORDAY;

// GRAVA AS IMAGENS FRONTAIS
// for( i = TAM_BORDAY-gravaby ; i < nx+TAM_BORDAY+gravaby ; i++)
for( i = 18 ; i < 19 ; i++) // debug
{
    ctagrav++;
    if (QUAL_MATERIAL)
    {
        // GRAVA A SAR DA CABECA
        if (GRAVA_FRAME_SAR && GRAVA_FRO)

            grava_sar_fro(99,ctagrav,NUM_MAX_ITERACAO,i,TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2,TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB);
    }
    // GRAVA O MODULO DO CAMPO ELETROMAGNETICO
    if (GRAVA_FRAME_E && GRAVA_FRO)

        grava_cpo_fro(99,ctagrav,i,TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2,TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB,1);
}

ctagrav = 0;
gravabx = 0;

```

```

        if (GRAVA_BORDA)
            gravabx = TAM_BORDAX;

        // GRAVA AS IMAGENS CORONAIS
        // for( j = TAM_BORDAX-gravabx ; j < ny+TAM_BORDAX+gravabx ; j++)
        for( j = 13 ; j < 14 ; j++) // debug
        {
            ctagrav++;
            if (QUAL_MATERIAL)
            {
                // GRAVA A SAR DA CABECA
                if (GRAVA_FRAME_SAR || GRAVA_COR)

                    grava_sar_cor(99,ctagrav,NUM_MAX_ITERACAO,j,TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2,TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB);
            }
            // GRAVA O MODULO DO CAMPO ELETROMAGNETICO
            // if (GRAVA_FRAME_E && GRAVA_COR)
            //debug
            grava_cpo_cor(99,ctagrav,j,TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2,TAM_BORDAZ+TAM_AR1+TAM_ANTENA+TAM_AR2+TAM_DA_CAB,1);
        }

        // RELATORIOS

        grava_rel_sar(dx,dy,dz,dt,cpopele,cpomusc,cposso,cpoolho,cpcereb,cpoar,bytesalocados,

nro_cel_borda,log_g,berenger_g,sigma_zero,nmax,atingiu_borda,degringolou,

m_int_ptol,m_int_pto2,m_val_ptol,m_val_pto2);

        // AVISOS FINAIS

        time2=time(NULL);

        fprintf(stdout, "\n");
        fprintf(stdout, "SIMULACAO CONCLUIDA!!\n");
        fprintf(stdout, "\n");
        fprintf(stdout, "Tempo Total Simulado = %lg segundos em %d
interacoes\n",TempoTotalSimulado,interacao);
        fprintf(stdout, "\n");
        diff_time = difftime(time2,time1);
        fprintf(stdout, "Tempo Total = %lg segundos\n",diff_time);
        fprintf(stdout, "Feito com %lg interacoes por
segundo\n", (double)diff_time/(double)interacao);
        fprintf(stdout, "\n");
        fflush(stdout);
    }

/*****
*****
**
**
** ROTINAS PARA P PROGRAMA PRINCIPAL
**
**
**
**

```

```

*****
*****/

/*****
*
*   FUNCAO:   Gera_bordaz()
*
*   RECEBE:   Numero do Corte
*   RETORNA:  NULL
*   FUNCAO:   Criar uma tela de AR com borda absorvedora
*   OBS:      Nesta funcao, i=linha e j=coluna
*
*****/
void Gera_bordaz(int k)
{
    int i,j,contai;

    contai=0;

    // IMPLEMENTA A BORDA SUPERIOR
    for(i = 0 ; i < TAM_BORDAX ; i++)
    {
        for(j = 0 ; j < nx+2*TAM_BORDAY ; j++)
        {
            if (j<TAM_BORDAY || j>=nx+TAM_BORDAY)
            {
                cab[j][contai][k] = CANTO;
            }
            else
            {
                cab[j][contai][k] = BORDAX;
            }
        }
        contai++;
    }

    // IMPLEMENTA A BORDA Z
    for(i = 0 ; i < ny ; i++)
    {
        for(j = 0 ; j < nx+2*TAM_BORDAY ; j++)
        {
            if (j<TAM_BORDAY || j>=nx+TAM_BORDAY)
            {
                cab[j][contai][k] = BORDAY;
            }
            else
            {
                cab[j][contai][k] = BORDAZ;
            }
        }
        contai++;
    }

    // IMPLEMENTA A BORDA INFERIOR
    for(i = 0 ; i < TAM_BORDAX ; i++)
    {
        for(j = 0 ; j < nx+2*TAM_BORDAY ; j++)
        {
            if (j<TAM_BORDAY || j>=nx+TAM_BORDAY)
            {

```

```

        cab[j][contai][k] = CANTO;
    }
    else
    {
        cab[j][contai][k] = BORDAX;
    }
}
contai++;
}
}

/*****
*
* FUNCAO: Gera_matriz_ar()
*
* RECEBE: Numero do Corte
* RETORNA: NULL
* FUNCAO: Criar uma tela de AR com borda absorvedora
* OBS: Nesta funcao, i=linha e j=coluna
*
*****/
void Gera_matriz_ar(int k)
{
    int i,j,contai=0;

    // IMPLEMENTA A BORDA SUPERIOR
    for(i = 0 ; i < TAM_BORDAX ; i++)
    {
        for(j = 0 ; j < nx+2*TAM_BORDAY ; j++)
        {
            if (j<TAM_BORDAY || j>=nx+TAM_BORDAY)
            {
                cab[j][contai][k] = CANTO;
            }
            else
            {
                cab[j][contai][k] = BORDAX;
            }
        }
        contai++;
    }

    // IMPLEMENTA O AR
    for(i = 0 ; i < ny ; i++)
    {
        for(j = 0 ; j < nx+2*TAM_BORDAY ; j++)
        {
            if (j<TAM_BORDAY || j>=nx+TAM_BORDAY)
            {
                cab[j][contai][k] = BORDAY;
            }
            else
            {
                cab[j][contai][k] = 0;
            }
        }
        contai++;
    }

    // IMPLEMENTA A BORDA INFERIOR
    for(i = 0 ; i < TAM_BORDAX ; i++)

```

```

    {
        for(j = 0 ; j < nx+2*TAM_BORDERDAY ; j++)
        {
            if (j<TAM_BORDERDAY || j>=nx+TAM_BORDERDAY)
            {
                cab[j][contai][k] = CANTO;
            }
            else
            {
                cab[j][contai][k] = BORDAX;
            }
        }
        contai++;
    }
}

/*****
*
* FUNCAO: Gera_antenal()
*
* RECEBE: Numero do Corte
* RETORNA: NULL
* FUNCAO: Passar a antena do arquivo .fd para uma matriz
* OBS: Nesta funcao, i=linha e j=coluna
*
*****/
void Gera_antenal(int k, int contant)
{
    char tmp[10], filename[15];
    int i,j,contai,num,soma_ai;

    FILE *fp;

    sprintf(filename,"/u/fdtdeb/teste/monop%d.fd",contant);
    fp=fopen(filename,"r");

    if(!fp)
    {
        printf("\nNao consigo abrir o arquivo %s",tmp);
        fflush(stdout);
        exit(0);
    }

    fgets(tmp,10,fp);
    num = atoi(tmp);
    fgets(tmp,10,fp);
    num = atoi(tmp);

    contai=0;

    // IMPLEMENTA A BORDA SUPERIOR
    for(i = 0 ; i < TAM_BORDERAX ; i++)
    {
        for(j = 0 ; j < nx+2*TAM_BORDERDAY ; j++)
        {
            if (j<TAM_BORDERDAY || j>=nx+TAM_BORDERDAY)
                cab[j][contai][k] = CANTO;
            else
                cab[j][contai][k] = BORDAX;
        }
    }
}

```



```

        }
        contai++;
    }

    contai=contai+ny-1;

    j = 0;

    do
    {

        // IMPLEMENTA A BORDA A ESQUERDA
        if (j == 0)
        {

            if (contai == TAM_BORDAX-1)
                break;

            for ( j = 0 ; j<TAM_BORDAY ; j++ )
                cab[j][contai][k] = BORDAY;
        }

        fgets(tmp,10,fp);
        num = atoi(tmp);

        if( num == -1 )
        {

            if (contai == TAM_BORDAX-1)
                break;

            // IMPLEMENTA A BORDA A DIREITA
            for ( soma_ai = 0 ; soma_ai<TAM_BORDAY ; soma_ai++ )
                cab[j++][contai][k] = BORDAY;

            contai--;
            j = 0;

        }
        else
        // PEGA VALORES DO ARQUIVO
        {
            cab[j][contai][k] = num;
            j++;
        }

    } while( !feof(fp) );

    contai=contai+ny+1; // Soma +2 pois no fim do .FD tem um (-1)

    // IMPLEMENTA A BORDA INFERIOR
    for(i = 0 ; i < TAM_BORDAX ; i++)
    {
        for(j = 0 ; j < nx+2*TAM_BORDAY ; j++)
        {
            if (j<TAM_BORDAY || j>=nx+TAM_BORDAY)
                cab[j][contai][k] = CANTO;
            else
                cab[j][contai][k] = BORDAX;
        }
        contai++;
    }

```

```

    }
    fclose(fp);
}

/*****
*
* FUNCAO: faz_calc_hx()
*
* RECEBE: Corte Inicial, Corte Final e Parametros Espaciais
* RETORNA: NULL
* FUNCAO: Calcula e Atualiza os Campos Hx para as Telas da Antena
* OBS:
*
*****/
void faz_calc_hx(double dy, double dz, double dt, double berenger_g, double
sigma_zero,
                double nro_iterac, double *atingiu_borda,
double *degringolou)
{
    int i,j,k,aux,dist_by,dist_bz;

    double mu,sigy,sigz,sigmy,sigmz,Day,Daz,Dby,Dbz;

    // ATUALIZANDO OS CAMPOS H
    #ifdef _CRAYC
        #pragma _CRI ivdep
        #pragma _CRI concurrent
    #endif
    for(i=0; i<(nx+2*TAM_BORDERDAY); i++)
    {
        for(j=0; j<(ny+2*TAM_BORDAX-1); j++)
        {
            for(k=0; k<nz-1; k++)
            {
                aux = cab[i][j][k];

                mu = material[aux][MU];

                if (j>=TAM_BORDAX && j<ny+TAM_BORDAX &&
i>=TAM_BORDERDAY && i<nx+TAM_BORDERDAY &&
                k>=TAM_BORDAZ && k<nz+TAM_BORDAZ)
                {
                    sigmy = material[aux][MAGSIGY];
                    sigmz = material[aux][MAGSIGZ];
                }
                else
                {
                    // IMPLEMENTA A BORDA BERENGER
                    if (i<TAM_BORDERDAY)
                        dist_by = TAM_BORDERDAY-i;
                    else if (i>=nx+TAM_BORDERDAY)
                        dist_by = i-TAM_BORDERDAY-nx;

                    if (k<TAM_BORDAZ)
                        dist_bz = TAM_BORDAZ-k;
                    else if (k>=nz+TAM_BORDAZ)
                        dist_bz = k-TAM_BORDAZ-nz;

                    sigy =
sigma_zero*pow(pow(berenger_g,1/TAM_BORDERDAY),dist_by);

```

```

                                sigz =
sigma_zero*pow(pow(berenger_g,1/TAM_BORDAZ),dist_bz);

                                sigmy = sigy*MU_0/EPS_0;
                                sigmz = sigz*MU_0/EPS_0;
                                }

                                Day = (UM - (sigmy*dt)/(DOIS*mu))/(UM +
(sigmy*dt)/(DOIS*mu));
                                Dby = (dt/mu) / (UM + (sigmy*dt)/(DOIS*mu));

                                Daz = (UM - (sigmz*dt)/(DOIS*mu))/(UM +
(sigmz*dt)/(DOIS*mu));
                                Dbz = (dt/mu) / (UM + (sigmz*dt)/(DOIS*mu));

                                hx[i][j][k] = ((Daz+Day)/DOIS)*hx[i][j][k] +
(Dbz*(ey[i][j][k+1] - ey[i][j][k])/dz
                                                - Dby*(ez[i][j+1][k] -
ez[i][j][k])/dy );

                                if (*atingiu_borda == 0)
                                {
                                        if (cab[i][j][k] == BORDAX || cab[i][j][k] ==
BORDAY || cab[i][j][k] == BORDAZ ||
                                                cab[i][j][k] == CANTO)
                                        {
                                                if (hx[i][j][k] != 0)
                                                        *atingiu_borda = nro_iterac;
                                        }
                                }

                                if (*degringolou == 0)
                                        if (hx[i][j][k] > 0 || cab[i][j][k] !=
EXCITACAO)
                                                *degringolou = nro_iterac;

/*                                if (cab[i][j][k] == TERRA)
                                {
                                        hx[i][j][k] = 0.0;
                                        hy[i][j][k] = 0.0;
                                        hz[i][j][k] = 0.0;
                                        ex[i][j][k] = 0.0;
                                        ey[i][j][k] = 0.0;
                                        ez[i][j][k] = 0.0;
                                }
                                else if (cab[i][j][k] == CAMPOX)
                                {
                                        hy[i][j][k] = 0.0;
                                        hz[i][j][k] = 0.0;
                                        ey[i][j][k] = 0.0;
                                        ez[i][j][k] = 0.0;
                                }
                                else if (cab[i][j][k] == CAMPOY)
                                {
                                        hx[i][j][k] = 0.0;
                                        hz[i][j][k] = 0.0;
                                        ex[i][j][k] = 0.0;
                                        ez[i][j][k] = 0.0;
                                }
                                else if (cab[i][j][k] == CAMPOZ)
                                {

```

```

        hx[i][j][k] = 0.0;
        hy[i][j][k] = 0.0;
        ex[i][j][k] = 0.0;
        ey[i][j][k] = 0.0;
    }
    else if (cab[i][j][k] == CAMPOSXZ)
    {
        hy[i][j][k] = 0.0;
        ey[i][j][k] = 0.0;
    }
    else if (cab[i][j][k] == CAMPOSXY)
    {
        hz[i][j][k] = 0.0;
        ez[i][j][k] = 0.0;
    }
    else if (cab[i][j][k] == CAMPOSYZ)
    {
        hx[i][j][k] = 0.0;
        ex[i][j][k] = 0.0;
    }
}*/
}
}
}

/*****
*
* FUNCAO: faz_calc_hy()
*
* RECEBE: Corte Inicial, Corte Final e Parametros Espaciais
* RETORNA: NULL
* FUNCAO: Calcula e Atualiza os Campos Hy para as Telas da Antena
* OBS:
*
*****/
void faz_calc_hy(double dx, double dz, double dt, double berenger_g, double
sigma_zero,
                double nro_iterac, double *atingiu_borda,
double *degringolou)
{
    int i,j,k,aux,dist_bx,dist_bz;

    double mu,sigx,sigz,sigmx,sigmz,Dax,Daz,Dbx,Dbz;

#ifdef _CRAYC
    #pragma _CRI ivdep
    #pragma _CRI concurrent
#endif
    for(i=0; i<(nx+2*TAM_BORDAY-1); i++)
    {
        for(j=0; j<(ny+2*TAM_BORDAX); j++)
        {
            for(k=0; k<nz-1; k++)
            {
                aux = cab[i][j][k];

                mu = material[aux][MU];

                if (j>=TAM_BORDAX && j<ny+TAM_BORDAX &&
i>=TAM_BORDAY && i<nx+TAM_BORDAY &&
                    k>=TAM_BORDAZ && k<nz+TAM_BORDAZ)

```

```

        {
            sigmx = material[aux][MAGSIGX];
            sigmz = material[aux][MAGSIGZ];
        }
    else
    {
        // IMPLEMENTA A BORDA BERENGER
        if (j<TAM_BORDAX)
            dist_bx = TAM_BORDAX-j;
        else if (j>=ny+TAM_BORDAX)
            dist_bx = j-TAM_BORDAX-ny;

        if (k<TAM_BORDAZ)
            dist_bz = TAM_BORDAZ-k;
        else if (k>=nz+TAM_BORDAZ)
            dist_bz = k-TAM_BORDAZ-nz;

        sigx =
sigma_zero*pow(pow(berenger_g,1/TAM_BORDAX),dist_bx);
        sigz =
sigma_zero*pow(pow(berenger_g,1/TAM_BORDAZ),dist_bz);

        sigmx = sigx*MU_0/EPS_0;
        sigmz = sigz*MU_0/EPS_0;
    }

    Dax = (UM - (sigmx*dt)/(DOIS*mu))/(UM +
(sigmx*dt)/(DOIS*mu));
    Dbx = (dt/mu) / (UM + (sigmx*dt)/(DOIS*mu));

    Daz = (UM - (sigmz*dt)/(DOIS*mu))/(UM +
(sigmz*dt)/(DOIS*mu));
    Dbz = (dt/mu) / (UM + (sigmz*dt)/(DOIS*mu));

    hy[i][j][k] = ((Dax+Daz)/DOIS)*hy[i][j][k] + (
Dbx*(ez[i+1][j][k]- ez[i][j][k])/dx
- Dbz*(ex[i][j][k+1] -
ex[i][j][k])/dz );

    if (*atingiu_borda == 0)
    {
        if (cab[i][j][k] == BORDAX || cab[i][j][k] ==
BORDAY || cab[i][j][k] == BORDAZ ||
            cab[i][j][k] == CANTO)
        {
            if (hy[i][j][k] != 0)
                *atingiu_borda = nro_iterac;
        }
    }

    if (*degringolou == 0)
        if (hy[i][j][k] > 0 || cab[i][j][k] !=
EXCITACAO)
            *degringolou = nro_iterac;

    /*
    if (cab[i][j][k] == TERRA)
    {
        hx[i][j][k] = 0.0;
        hy[i][j][k] = 0.0;
        hz[i][j][k] = 0.0;
        ex[i][j][k] = 0.0;
    }

```

```

        ey[i][j][k] = 0.0;
        ez[i][j][k] = 0.0;
    }
else if (cab[i][j][k] == CAMPOX)
{
    hy[i][j][k] = 0.0;
    hz[i][j][k] = 0.0;
    ey[i][j][k] = 0.0;
    ez[i][j][k] = 0.0;
}
else if (cab[i][j][k] == CAMPOY)
{
    hx[i][j][k] = 0.0;
    hz[i][j][k] = 0.0;
    ex[i][j][k] = 0.0;
    ez[i][j][k] = 0.0;
}
else if (cab[i][j][k] == CAMPOZ)
{
    hx[i][j][k] = 0.0;
    hy[i][j][k] = 0.0;
    ex[i][j][k] = 0.0;
    ey[i][j][k] = 0.0;
}
else if (cab[i][j][k] == CAMPOSXZ)
{
    hy[i][j][k] = 0.0;
    ey[i][j][k] = 0.0;
}
else if (cab[i][j][k] == CAMPOSXY)
{
    hz[i][j][k] = 0.0;
    ez[i][j][k] = 0.0;
}
else if (cab[i][j][k] == CAMPOSYZ)
{
    hx[i][j][k] = 0.0;
    ex[i][j][k] = 0.0;
}*/
    }
}
}

/*****
*
* FUNCAO: faz_calc_hz()
*
* RECEBE: Corte Inicial, Corte Final e Parametros Espaciais
* RETORNA: NULL
* FUNCAO: Calcula e Atualiza os Campos Hz para as Telas da Antena
* OBS:
*
*****/
void faz_calc_hz(double dy, double dx, double dt, double berenger_g, double
sigma_zero,
                double nro_iterac, double *atingiu_borda,
double *degringolou)
{
    int i,j,k,aux,dist_bx,dist_by;

```

```

double mu, sigx, sigy, sigmx, sigmy, Dax, Day, Dbx, Dby;

#ifdef _CRAYC
    #pragma _CRI ivdep
    #pragma _CRI concurrent
#endif
for(i=0; i<(nx+2*TAM_BORDERDAY-1); i++)
{
    for(j=0; j<(ny+2*TAM_BORDERAX-1); j++)
    {
        for(k=0; k<nz; k++)
        {
            aux = cab[i][j][k];

            mu = material[aux][MU];

            if (j>=TAM_BORDERAX && j<ny+TAM_BORDERAX &&
i>=TAM_BORDERDAY && i<nx+TAM_BORDERDAY &&
                k>=TAM_BORDERAZ && k<nz+TAM_BORDERAZ)
            {
                sigmx = material[aux][MAGSIGX];
                sigmy = material[aux][MAGSIGY];
            }
            else
            {
                // IMPLEMENTA A BORDA BERENGER
                if (j<TAM_BORDERAX)
                    dist_bx = TAM_BORDERAX-j;
                else if (j>=ny+TAM_BORDERAX)
                    dist_bx = j-TAM_BORDERAX-ny;

                if (i<TAM_BORDERDAY)
                    dist_by = TAM_BORDERDAY-i;
                else if (i>=nx+TAM_BORDERDAY)
                    dist_by = i-TAM_BORDERDAY-nx;

                sigx =
sigma_zero*pow(pow(berenger_g,1/TAM_BORDERAX),dist_bx);
                sigy =
sigma_zero*pow(pow(berenger_g,1/TAM_BORDERDAY),dist_by);

                sigmx = sigx*MU_0/EPS_0;
                sigmy = sigy*MU_0/EPS_0;
            }

            Dax = (UM - (sigmx*dt)/(DOIS*mu))/(UM +
(sigmx*dt)/(DOIS*mu));
            Dbx = (dt/mu) / (UM + (sigmx*dt)/(DOIS*mu));

            Day = (UM - (sigmy*dt)/(DOIS*mu))/(UM +
(sigmy*dt)/(DOIS*mu));
            Dby = (dt/mu) / (UM + (sigmy*dt)/(DOIS*mu));

            hz[i][j][k] = ((Dax+Day)/DOIS)*hz[i][j][k] + (
Dby*(ex[i][j+1][k]
                - ex[i][j][k])/dy -
Dbx*(ey[i+1][j][k] - ey[i][j][k])/dx );

            if (*atingiu_borda == 0)
            {

```

```

        if (cab[i][j][k] == BORDAX || cab[i][j][k] ==
BORDAY || cab[i][j][k] == BORDAZ ||
        cab[i][j][k] == CANTO)
        {
            if (hz[i][j][k] != 0)
                *atingiu_borda = nro_iterac;
        }
    }

    if (*degringolou == 0)
        if (hz[i][j][k] > 0 || cab[i][j][k] !=
EXCITACAO)
            *degringolou = nro_iterac;

/*
    if (cab[i][j][k] == TERRA)
    {
        hx[i][j][k] = 0.0;
        hy[i][j][k] = 0.0;
        hz[i][j][k] = 0.0;
        ex[i][j][k] = 0.0;
        ey[i][j][k] = 0.0;
        ez[i][j][k] = 0.0;
    }
    else if (cab[i][j][k] == CAMPOX)
    {
        hy[i][j][k] = 0.0;
        hz[i][j][k] = 0.0;
        ey[i][j][k] = 0.0;
        ez[i][j][k] = 0.0;
    }
    else if (cab[i][j][k] == CAMPOY)
    {
        hx[i][j][k] = 0.0;
        hz[i][j][k] = 0.0;
        ex[i][j][k] = 0.0;
        ez[i][j][k] = 0.0;
    }
    else if (cab[i][j][k] == CAMPOZ)
    {
        hx[i][j][k] = 0.0;
        hy[i][j][k] = 0.0;
        ex[i][j][k] = 0.0;
        ey[i][j][k] = 0.0;
    }
    else if (cab[i][j][k] == CAMPOSXZ)
    {
        hy[i][j][k] = 0.0;
        ey[i][j][k] = 0.0;
    }
    else if (cab[i][j][k] == CAMPOSXY)
    {
        hz[i][j][k] = 0.0;
        ez[i][j][k] = 0.0;
    }
    else if (cab[i][j][k] == CAMPOSYZ)
    {
        hx[i][j][k] = 0.0;
        ex[i][j][k] = 0.0;
    }
*/
}
}
}

```



```

    }
}

/*****
*
*   FUNCAO:   faz_calc_ex()
*
*   RECEBE:   Corte Inicial, Corte Final e Parametros Espaciais
*   RETORNA:  NULL
*   FUNCAO:   Calcula e Atualiza os Campos Ex para as Telas da Antena
*   OBS:
*
*****/
void faz_calc_ex(double dy, double dz, double dt, double berenger_g, double
sigma_zero,
                double nro_iterac, double *atingiu_borda,
double *degringolou)
{
    int i,j,k,aux,dist_by,dist_bz;

    double sigy,sigz,Cay,Caz,Cby,Cbz,eps;

#ifdef _CRAYC
    #pragma _CRI ivdep
    #pragma _CRI concurrent
#endif
    for(i=0; i<(nx+2*TAM_BORDERDAY); i++)
    {
        for(j=1; j<(ny+2*TAM_BORDERDAX); j++)
        {
            for(k=1; k<nz; k++)
            {
                aux = cab[i][j][k];

                eps = material[aux][EPS];

                if (j>=TAM_BORDERDAX && j<ny+TAM_BORDERDAX &&
i>=TAM_BORDERDAY && i<nx+TAM_BORDERDAY &&
                    k>=TAM_BORDERDAZ && k<nz+TAM_BORDERDAZ)
                {
                    sigy = material[aux][SIGY];
                    sigz = material[aux][SIGZ];
                }
                else
                {
                    // IMPLEMENTA A BORDA BERENGER
                    if (i<TAM_BORDERDAY)
                        dist_by = TAM_BORDERDAY-i;
                    else if (i>=nx+TAM_BORDERDAY)
                        dist_by = i-TAM_BORDERDAY-nx;

                    if (k<TAM_BORDERDAZ)
                        dist_bz = TAM_BORDERDAZ-k;
                    else if (k>=nz+TAM_BORDERDAZ)
                        dist_bz = k-TAM_BORDERDAZ-nz;

                    sigy =
sigma_zero*pow(pow(berenger_g,1/TAM_BORDERDAY),dist_by);
                    sigz =
sigma_zero*pow(pow(berenger_g,1/TAM_BORDERDAZ),dist_bz);
                }
            }
        }
    }
}

```

```

Cay = (UM - (sigy*dt)/(DOIS*eps)) / (UM +
(sigy*dt)/(DOIS*eps));
Cby = (dt/eps) / (UM + (sigy*dt)/(DOIS*eps));

Caz = (UM - (sigz*dt)/(DOIS*eps)) / (UM +
(sigz*dt)/(DOIS*eps));
Cbz = (dt/eps) / (UM + (sigz*dt)/(DOIS*eps));

if (cab[i][j][k] == TERRA || cab[i][j][k] ==
CAMPOSXZ || cab[i][j][k] == TODCAMPOS)
{
    Cay = -1;
    Caz = -1;
    Cby = 0;
    Cbz = 0;
}

ex[i][j][k] = ((Caz + Cay)/DOIS)*ex[i][j][k] + (
Cby*(hz[i][j][k]
- hz[i][j-1][k])/dy -
Cbz*(hy[i][j][k] - hy[i][j][k-1])/dz);

if (*atingiu_borda == 0)
{
    if (cab[i][j][k] == BORDAX || cab[i][j][k] ==
BORDAY || cab[i][j][k] == BORDAZ ||
cab[i][j][k] == CANTO)
    {
        if (ex[i][j][k] != 0)
            *atingiu_borda = nro_iterac;
    }
}

if (*degringolou == 0)
    if (ex[i][j][k] > 0 || cab[i][j][k] !=
EXCITACAO)
        *degringolou = nro_iterac;

/* if (cab[i][j][k] == TERRA)
{
    hx[i][j][k] = 0.0;
    hy[i][j][k] = 0.0;
    hz[i][j][k] = 0.0;
    ex[i][j][k] = 0.0;
    ey[i][j][k] = 0.0;
    ez[i][j][k] = 0.0;
}
else if (cab[i][j][k] == CAMPOX)
{
    hy[i][j][k] = 0.0;
    hz[i][j][k] = 0.0;
    ey[i][j][k] = 0.0;
    ez[i][j][k] = 0.0;
}
else if (cab[i][j][k] == CAMPOY)
{
    hx[i][j][k] = 0.0;
    hz[i][j][k] = 0.0;
    ex[i][j][k] = 0.0;
    ez[i][j][k] = 0.0;
}

```

```

    }
    else if (cab[i][j][k] == CAMPOZ)
    {
        hx[i][j][k] = 0.0;
        hy[i][j][k] = 0.0;
        ex[i][j][k] = 0.0;
        ey[i][j][k] = 0.0;
    }
    else if (cab[i][j][k] == CAMPOSXZ)
    {
        hy[i][j][k] = 0.0;
        ey[i][j][k] = 0.0;
    }
    else if (cab[i][j][k] == CAMPOSXY)
    {
        hz[i][j][k] = 0.0;
        ez[i][j][k] = 0.0;
    }
    else if (cab[i][j][k] == CAMPOSYZ)
    {
        hx[i][j][k] = 0.0;
        ex[i][j][k] = 0.0;
    }
}*/
}
}
}

/*****
*
* FUNCAO: faz_calc_ey()
*
* RECEBE: Corte Inicial, Corte Final e Parametros Espaciais
* RETORNA: NULL
* FUNCAO: Calcula e Atualiza os Campos Ey para as Telas da Antena
* OBS:
*
*****/
void faz_calc_ey(double dx, double dz, double dt, double berenger_g, double
sigma_zero,
                double nro_iterac, double *atingiu_borda,
double *degringolou)
{
    int i,j,k,aux,dist_bx,dist_bz;

    double sigx,sigz,Cax,Caz,Cbx,Cbz,eps;

#ifdef _CRAYC
    #pragma _CRI ivdep
    #pragma _CRI concurrent
#endif
    for(i=1; i<(nx+2*TAM_BORDAY); i++)
    {
        for(j=0; j<(ny+2*TAM_BORDAX); j++)
        {
            for(k=1; k<nz; k++)
            {
                aux = cab[i][j][k];

                eps = material[aux][EPS];

```

```

        if (j>=TAM_BORDAX && j<ny+TAM_BORDAX &&
i>=TAM_BORDAY && i<nx+TAM_BORDAY &&
            k>=TAM_BORDAZ && k<nz+TAM_BORDAZ)
        {
            sigx = material[aux][SIGX];
            sigz = material[aux][SIGZ];
        }
    else
    {
        // IMPLEMENTA A BORDA BERENGER
        if (j<TAM_BORDAX)
            dist_bx = TAM_BORDAX-j;
        else if (j>=ny+TAM_BORDAX)
            dist_bx = j-TAM_BORDAX-ny;

        if (k<TAM_BORDAZ)
            dist_bz = TAM_BORDAZ-k;
        else if (k>=nz+TAM_BORDAZ)
            dist_bz = k-TAM_BORDAZ-nz;

        sigx =
sigma_zero*pow(pow(berenger_g,1/TAM_BORDAX),dist_bx);
        sigz =
sigma_zero*pow(pow(berenger_g,1/TAM_BORDAZ),dist_bz);
    }

    Cax = (UM - (sigx*dt)/(DOIS*eps)) / (UM +
(sigx*dt)/(DOIS*eps));
    Cbx = (dt/eps) / (UM + (sigx*dt)/(DOIS*eps));

    Caz = (UM - (sigz*dt)/(DOIS*eps)) / (UM +
(sigz*dt)/(DOIS*eps));
    Cbz = (dt/eps) / (UM + (sigz*dt)/(DOIS*eps));

    if (cab[i][j][k] == TERRA || cab[i][j][k] ==
CAMPOSXZ || cab[i][j][k] == TODCAMPOS)
    {
        Cax = -1;
        Caz = -1;
        Cbx = 0;
        Cbz = 0;
    }

    ey[i][j][k] = ((Cax + Caz)/DOIS)*ey[i][j][k] + (
Cbz*(hx[i][j][k]
- hx[i][j][k-1])/dz -
Cbz*(hz[i][j][k] - hz[i-1][j][k])/dx);

    if (*atingiu_borda == 0)
    {
        if (cab[i][j][k] == BORDAX || cab[i][j][k] ==
BORDAY || cab[i][j][k] == BORDAZ ||
            cab[i][j][k] == CANTO)
        {
            if (ey[i][j][k] != 0)
                *atingiu_borda = nro_iterac;
        }
    }

    if (*degringolou == 0)

```

```

EXCITACAO)                if (ey[i][j][k] > 0 || cab[i][j][k] !=
                            *degringolou = nro_iterac;

/*                          if (cab[i][j][k] == TERRA)
                            {
                                hx[i][j][k] = 0.0;
                                hy[i][j][k] = 0.0;
                                hz[i][j][k] = 0.0;
                                ex[i][j][k] = 0.0;
                                ey[i][j][k] = 0.0;
                                ez[i][j][k] = 0.0;
                            }
                            else if (cab[i][j][k] == CAMPOX)
                            {
                                hy[i][j][k] = 0.0;
                                hz[i][j][k] = 0.0;
                                ey[i][j][k] = 0.0;
                                ez[i][j][k] = 0.0;
                            }
                            else if (cab[i][j][k] == CAMPOY)
                            {
                                hx[i][j][k] = 0.0;
                                hz[i][j][k] = 0.0;
                                ex[i][j][k] = 0.0;
                                ez[i][j][k] = 0.0;
                            }
                            else if (cab[i][j][k] == CAMPOZ)
                            {
                                hx[i][j][k] = 0.0;
                                hy[i][j][k] = 0.0;
                                ex[i][j][k] = 0.0;
                                ey[i][j][k] = 0.0;
                            }
                            else if (cab[i][j][k] == CAMPOSXZ)
                            {
                                hy[i][j][k] = 0.0;
                                ey[i][j][k] = 0.0;
                            }
                            else if (cab[i][j][k] == CAMPOSXY)
                            {
                                hz[i][j][k] = 0.0;
                                ez[i][j][k] = 0.0;
                            }
                            else if (cab[i][j][k] == CAMPOSYZ)
                            {
                                hx[i][j][k] = 0.0;
                                ex[i][j][k] = 0.0;
                            }
                            }*/
                        }
                    }
                }

/*****
*
*   FUNCAO:   faz_calc_ez()
*
*   RECEBE:   Corte Inicial, Corte Final e Parametros Espaciais
*   RETORNA:  NULL
*   FUNCAO:   Calcula e Atualiza os Campos Ez para as Telas da Antena
*/

```

```

*   OBS:
*
*****/
void faz_calc_ez(double dx, double dy, double dz, double dt, double
*cpopole, double *cpomusc,
                double *cposso, double *cpoolho, double
*cpcocereb, double *cpoar,
                double berenger_g, double sigma_zero, double
nro_iterac, double *atingiu_borda,
                double *degringolou)
{
    int i,j,k,aux,dist_bx,dist_by;

    double sigx,sigy,Cax,Cay,Cbx,Cby,modulo_e,cl_sar,eps;

#ifdef _CRAYC
    #pragma _CRI ivdep
    #pragma _CRI concurrent
#endif
    for(i=1; i<(nx+2*TAM_BORDAY); i++)
    {
        for(j=1; j<(ny+2*TAM_BORDAX); j++)
        {
            for(k=0; k<nz; k++)
            {
                aux = cab[i][j][k];

                eps = material[aux][EPS];

                if (j>=TAM_BORDAX && j<ny+TAM_BORDAX &&
i>=TAM_BORDAY && i<nx+TAM_BORDAY &&
                    k>=TAM_BORDAZ && k<nz+TAM_BORDAZ)
                {
                    sigx = material[aux][SIGX];
                    sigy = material[aux][SIGY];
                }
                else
                {
                    // IMPLEMENTA A BORDA BERENGER
                    if (j<TAM_BORDAX)
                        dist_bx = TAM_BORDAX-j;
                    else if (j>=ny+TAM_BORDAX)
                        dist_bx = j-TAM_BORDAX-ny;

                    if (i<TAM_BORDAY)
                        dist_by = TAM_BORDAY-i;
                    else if (i>=nx+TAM_BORDAY)
                        dist_by = i-TAM_BORDAY-nx;

                    sigx =
sigma_zero*pow(pow(berenger_g,1/TAM_BORDAX),dist_bx);
                    sigy =
sigma_zero*pow(pow(berenger_g,1/TAM_BORDAY),dist_by);
                }

                Cax = (UM - (sigx*dt)/(DOIS*eps)) / (UM +
(sigx*dt)/(DOIS*eps));
                Cbx = (dt/eps) / (UM + (sigx*dt)/(DOIS*eps));

                Cay = (UM - (sigy*dt)/(DOIS*eps)) / (UM +
(sigy*dt)/(DOIS*eps));
            }
        }
    }
}

```

```

Cby = (dt/eps) / (UM + (sigy*dt)/(DOIS*eps));

if (cab[i][j][k] == TERRA || cab[i][j][k] ==
CAMPOSXZ || cab[i][j][k] == TODCAMPOS)
{
    Cax = -1;
    Cay = -1;
    Cbx = 0;
    Cby = 0;
}

ez[i][j][k] = ((Cax + Cay)/DOIS)*ez[i][j][k] + (
Cbx*(hy[i][j][k]
- hy[i-1][j][k])/dx -
Cby*(hx[i][j][k] - hx[i][j-1][k])/dy);

if (*atingiu_borda == 0)
{
    if (cab[i][j][k] == BORDAX || cab[i][j][k] ==
BORDAY || cab[i][j][k] == BORDAZ ||
cab[i][j][k] == CANTO)
    {
        if (ez[i][j][k] != 0)
            *atingiu_borda = nro_iterac;
    }
}

if (*degringolou == 0)
    if (ez[i][j][k] > 0 || cab[i][j][k] !=
EXCITACAO)
        *degringolou = nro_iterac;

/*
if (cab[i][j][k] == TERRA)
{
    hx[i][j][k] = 0.0;
    hy[i][j][k] = 0.0;
    hz[i][j][k] = 0.0;
    ex[i][j][k] = 0.0;
    ey[i][j][k] = 0.0;
    ez[i][j][k] = 0.0;
}
else if (cab[i][j][k] == CAMPOX)
{
    hy[i][j][k] = 0.0;
    hz[i][j][k] = 0.0;
    ey[i][j][k] = 0.0;
    ez[i][j][k] = 0.0;
}
else if (cab[i][j][k] == CAMPOY)
{
    hx[i][j][k] = 0.0;
    hz[i][j][k] = 0.0;
    ex[i][j][k] = 0.0;
    ez[i][j][k] = 0.0;
}
else if (cab[i][j][k] == CAMPOZ)
{
    hx[i][j][k] = 0.0;
    hy[i][j][k] = 0.0;
    ex[i][j][k] = 0.0;
    ey[i][j][k] = 0.0;
}

```

```

    }
else if (cab[i][j][k] == CAMPOSXZ)
{
    hy[i][j][k] = 0.0;
    ey[i][j][k] = 0.0;
}
else if (cab[i][j][k] == CAMPOSXY)
{
    hz[i][j][k] = 0.0;
    ez[i][j][k] = 0.0;
}
else if (cab[i][j][k] == CAMPOSYZ)
{
    hx[i][j][k] = 0.0;
    ex[i][j][k] = 0.0;
}*/

// Calcula o SAR para cada ponto e soma com o
anterior
// material[aux][RO]*dx*dy*dz serve para corrigir um
fator de volume
// *1000 para termos mW/g ou W/Kg por V/m RMS
modulo_e = sqrt(ex[i][j][k]*ex[i][j][k] +
ey[i][j][k]*ey[i][j][k] + ez[i][j][k]*ez[i][j][k]);
cl_sar =
(material[aux][SIGX]/(2*material[aux][RO]*dx*dy*dz))*(modulo_e*modulo_e)*10
00;

vsar[i][j][k] = vsar[i][j][k] + cl_sar;

if (cab[i][j][k] == PELE)
    *cpopele += modulo_e;
else
if (cab[i][j][k] == MUSCULO)

    *cpomusc += modulo_e;
else if (cab[i][j][k] == OSSO)
    *cposso += modulo_e;
else if (cab[i][j][k] == OLHO)
    *cpoolho += modulo_e;
else if (cab[i][j][k] == CEREBRO)
    *cpocereb += modulo_e;
else if (cab[i][j][k] == AR)
    *cpoar += modulo_e;
    }
}
}

/*****
*
* FUNCAO: grava_tela_cab()
*
* RECEBE: Qual tela deve gravar
* RETORNA: NULL
* FUNCAO: Grava a tela de um Cab para DEBUG
* OBS:
*
*****/
void grava_tela_cab(int k)
{
    FILE *fp;

```



```

char mat[100];
int i,j;

sprintf(mat,"tela%d.txt",k);

printf("Gravando tela da matriz CAB: %s\n",mat);
fp=fopen(mat,"w+");

for(i = 0 ; i < 2*TAM_BORDAX+ny ; i++)
{
    for(j = 0 ; j < nx+2*TAM_BORDAY ; j++)
    {
        fprintf(fp,"%d ",cab[j][i][k]);
    }
    fprintf(fp,"\n");
}
fclose(fp);
}

/*****
*
* FUNCAO: calcula_memoria_req()
*
* RECEBE: NULL
* RETORNA: Valor da memoria que devera ser armazenada
* FUNCAO: Calcula a memoria requerida para a simulacao
* OBS:
*
*****/
int calcula_memoria_req(void)
{
    int bytesalocados = 0;
    bytesalocados += ( (nx+2*TAM_BORDAY) * (ny+2*TAM_BORDAX) * (nz) *
sizeof(double));
    bytesalocados += ( (nx+2*TAM_BORDAY+1) * (ny+2*TAM_BORDAX) * (nz) *
sizeof(double));
    bytesalocados += ( (nx+2*TAM_BORDAY+1) * (ny+2*TAM_BORDAX) * (nz) *
sizeof(double));
    bytesalocados += ( (nx+2*TAM_BORDAY-1) * (ny+2*TAM_BORDAX) * (nz) *
sizeof(double));
    bytesalocados += ( (nx+2*TAM_BORDAY) * (ny+2*TAM_BORDAX) * (nz) *
sizeof(double));
    bytesalocados += ( (nx+2*TAM_BORDAY) * (ny+2*TAM_BORDAX) * (nz) *
sizeof(double));
    bytesalocados += ( (nx+2*TAM_BORDAY) * (ny+2*TAM_BORDAX) * (nz) *
sizeof(double));
    return bytesalocados;
}

/*****
*
* FUNCAO: aloca_memoria()
*
* RECEBE: NULL
* RETORNA: Valor da memoria alocada
* FUNCAO: Alocar a memoria necessaria para a simulacao
* OBS:
*
*****/
unsigned int aloca_memoria(void)
{

```

```

int i,j,k;
unsigned int bytesalocados = 0;

ex = (double ***)malloc((nx+2*TAM_BORDERDAY)*sizeof(double **));
if( ex == NULL )
    erro_alocacao();
for(i=0; i<(nx+2*TAM_BORDERDAY); i++)
{
    ex[i] = (double **)malloc((ny+2*TAM_BORDERAX)*sizeof(double
*)));
    if( ex[i] == NULL )
        erro_alocacao();

    for(j=0; j<(ny+2*TAM_BORDERAX); j++)
    {
        ex[i][j] = (double
*)malloc((nz)*sizeof(double));
        if( ex[i][j] == NULL )
            erro_alocacao();

        for(k=0; k<(nz); k++)
        {
            ex[i][j][k] = 0.0;
        }
    }
    bytesalocados += ( (nx+2*TAM_BORDERDAY) * (ny+2*TAM_BORDERAX) * (nz) *
sizeof(double));

    ey = (double ***)malloc((nx+2*TAM_BORDERDAY)*sizeof(double **));
    if( ey == NULL )
        erro_alocacao();

    for(i=0; i<(nx+2*TAM_BORDERDAY); i++)
    {
        ey[i] = (double **)malloc((ny+2*TAM_BORDERAX)*sizeof(double
*)));
        if( ey[i] == NULL )
            erro_alocacao();

        for(j=0; j<(ny+2*TAM_BORDERAX); j++)
        {
            ey[i][j] = (double
*)malloc((nz)*sizeof(double));
            if( ey[i][j] == NULL )
                erro_alocacao();

            for(k=0; k<(nz); k++)
            {
                ey[i][j][k] = 0.0;
            }
        }
        bytesalocados += ( (nx+2*TAM_BORDERDAY) * (ny+2*TAM_BORDERAX) * (nz) *
sizeof(double));

        ez = (double ***)malloc((nx+2*TAM_BORDERDAY)*sizeof(double **));
        if( ez == NULL )
            erro_alocacao();
        for(i=0; i<(nx+2*TAM_BORDERDAY); i++)
        {

```

```

        ez[i] = (double **)malloc((ny+2*TAM_BORDAX)*sizeof(double
*)));
        if( ez[i] == NULL )
            erro_alocacao();
        for(j=0; j<(ny+2*TAM_BORDAX); j++)
            {
                ez[i][j] = (double
*)malloc((nz)*sizeof(double));
                if( ez[i][j] == NULL )
                    erro_alocacao();

                for(k=0; k<(nz); k++)
                    {
                        ez[i][j][k] = 0.0;
                    }
            }
        bytesalocados += ( (nx+2*TAM_BORDAY) * (ny+2*TAM_BORDAX) * (nz) *
sizeof(double));

        hx = (double ***)malloc((nx+2*TAM_BORDAY)*sizeof(double **));
        if( hx == NULL )
            erro_alocacao();

        for(i=0; i<(nx+2*TAM_BORDAY); i++)
            {
                hx[i] = (double **)malloc((ny+2*TAM_BORDAX)*sizeof(double
*)));
                if( hx[i] == NULL )
                    erro_alocacao();

                for(j=0; j<(ny+2*TAM_BORDAX); j++)
                    {
                        hx[i][j] = (double
*)malloc((nz)*sizeof(double));
                        if( hx[i][j] == NULL )
                            erro_alocacao();

                        for(k=0; k<(nz); k++)
                            {
                                hx[i][j][k] = 0.0;
                            }
                    }
                bytesalocados += ( (nx+2*TAM_BORDAY) * (ny+2*TAM_BORDAX) * (nz) *
sizeof(double));

                hy = (double ***)malloc((nx+2*TAM_BORDAY)*sizeof(double **));
                if( hy == NULL )
                    erro_alocacao();

                for(i=0; i<(nx+2*TAM_BORDAY); i++)
                    {
                        hy[i] = (double **)malloc((ny+2*TAM_BORDAX)*sizeof(double
*)));
                        if( hy[i] == NULL )
                            erro_alocacao();

                        for(j=0; j<(ny+2*TAM_BORDAX); j++)
                            {

```

```

        hy[i][j] = (double
*)malloc((nz)*sizeof(double));
        if( hy[i][j] == NULL )
            erro_alocacao();

        for(k=0; k<(nz); k++)
            {
                hy[i][j][k] = 0.0;
            }
    }
    bytesalocados += ( (nx+2*TAM_BORDERDAY) * (ny+2*TAM_BORDERDAX) * (nz) *
sizeof(double));

    hz = (double ***)malloc((nx+2*TAM_BORDERDAY)*sizeof(double **));
    if( hz == NULL )
        erro_alocacao();

    for(i=0; i<(nx+2*TAM_BORDERDAY); i++)
        {
            hz[i] = (double **)malloc((ny+2*TAM_BORDERDAX)*sizeof(double
*)));
            if( hz[i] == NULL )
                erro_alocacao();

            for(j=0; j<(ny+2*TAM_BORDERDAX); j++)
                {
                    hz[i][j] = (double
*)malloc((nz)*sizeof(double));
                    if( hz[i][j] == NULL )
                        erro_alocacao();

                    for(k=0; k<(nz); k++)
                        {
                            hz[i][j][k] = 0.0;
                        }
                }
            bytesalocados += ( (nx+2*TAM_BORDERDAY) * (ny+2*TAM_BORDERDAX) * (nz) *
sizeof(double));

            vsar = (double ***)malloc((nx+2*TAM_BORDERDAY)*sizeof(double **));
            if( vsar == NULL )
                erro_alocacao();

            for(i=0; i<(nx+2*TAM_BORDERDAY); i++)
                {
                    vsar[i] = (double
*)malloc((ny+2*TAM_BORDERDAX)*sizeof(double *));
                    if( vsar[i] == NULL )
                        erro_alocacao();

                    for(j=0; j<(ny+2*TAM_BORDERDAX); j++)
                        {
                            vsar[i][j] = (double
*)malloc((nz)*sizeof(double));
                            if( vsar[i][j] == NULL )
                                erro_alocacao();

                            for(k=0; k<(nz); k++)
                                {

```

```

                                vsar[i][j][k] = 0.0;
                                }
                                }
                                }
    bytesalocados += ( (nx+2*TAM_BORDAY) * (ny+2*TAM_BORDAX) * (nz) *
sizeof(double));

    return bytesalocados;
}

/*****
*
* FUNCAO:   grava_sar_sag()
*
* RECEBE:  NUMERO DO ARQUIVO, ITERACAO E NUMERO DO CORTE
* RETORNA: NULL
* FUNCAO:  Gravar em .TXT a Imagem SAGITAL da Cabeça com a SAR
* OBS:
*
*****/
void grava_sar_sag(int n, int indgrav, int k, int nrointer)
{
    FILE *fp;
    char mat[100];
    int i,j,gravbrx,gravbry;
    double sar;

    gravbrx = 0;
    gravbry = 0;

    if (indgrav<10)
    {
        sprintf(mat,"sar_sag%d_00%d.txt",n,indgrav);
    }
    else if (indgrav<100)
    {
        sprintf(mat,"sar_sag%d_0%d.txt",n,indgrav);
    }
    else
    {
        sprintf(mat,"sar_sag%d_%d.txt",n,indgrav);
    }

    printf("Gravando SAR no arquivo %s para o corte Sagital k = %d
\n",mat,k);

    if (GRAVA_BORDA)
    {
        gravbrx = TAM_BORDAX;
        gravbry = TAM_BORDAY;
    }

    fp=fopen(mat,"w+");

    if (MATHLAB)
        fprintf(fp,"[");

    for(j = TAM_BORDAX-gravbrx ; j < ny+TAM_BORDAX+gravbrx ; j++)
    {
        for(i = TAM_BORDAY-gravbry ; i < nx+TAM_BORDAY+gravbry ; i++)
        {

```

```

        sar = vsar[i][j][k]/nrointer;
        fprintf(fp,"%%.2e ",sar);
    }
    if (MATHLAB)
        fprintf(fp,";\n");
    else
        fprintf(fp,"\n");
}
if (MATHLAB)
    fprintf(fp,"]");
fclose(fp);
}

/*****
*
* FUNCAO: grava_sar_fro()
*
* RECEBE: NRO ARQUIVO, ITERACAO, COLUNA E NRO DO CORTE INICIAL E FINAL
* RETORNA: NULL
* FUNCAO: Gravar em .TXT a Imagem FRONTAL da Cabeça com a SAR
* OBS:
*
*****/
void grava_sar_fro(int n, int ctagrav, int nrointer, int i ,int kinic, int
kfinal)
{
    FILE *fp;
    char mat[100];
    int j,k,gravbrx;
    double sar;

    gravbrx = 0;

    if (ctagrav<10)
    {
        sprintf(mat,"sar_fro%d_00%d.txt",n,ctagrav);
    }
    else if (ctagrav<100)
    {
        sprintf(mat,"sar_fro%d_0%d.txt",n,ctagrav);
    }
    else
    {
        sprintf(mat,"sar_fro%d_%d.txt",n,ctagrav);
    }

    printf("Gravando SAR no arquivo %s para o corte Frontal i = %d
\n",mat,i);

    if (GRAVA_BORDA)
    {
        gravbrx = TAM_BORDAX;
    }

    fp=fopen(mat,"w+");

    if (MATHLAB)
        fprintf(fp,"[");

    for(j = TAM_BORDAX-gravbrx ; j < ny+TAM_BORDAX+gravbrx ; j++)

```

```

    {
        for(k = kinic ; k < kfinal ; k++)
        {
            sar = vsar[i][j][k]/nrointer;
            fprintf(fp, "%.2e ", sar);
        }
        if (MATHLAB)
            fprintf(fp, ";\n");

        else
            fprintf(fp, "\n");
    }
    if (MATHLAB)
        fprintf(fp, "];");
    fclose(fp);
}

/*****
*
* FUNCAO: grava_sar_cor()
*
* RECEBE: NRO ARQUIVO, ITERACAO, LINHA E NRO DO CORTE INICIAL E FINAL
* RETORNA: NULL
* FUNCAO: Gravar em .TXT a Imagem CORONAL da Cabeça com a SAR
* OBS:
*
*****/
void grava_sar_cor(int n, int ctagrav, int nrointer, int j ,int kinic, int
kfinal)
{
    FILE *fp;
    char mat[100];
    int i,k,gravbry;
    double sar;

    gravbry = 0;

    if (ctagrav<10)
    {
        sprintf(mat, "sar_cor%d_00%d.txt", n, ctagrav);
    }
    else if (ctagrav<100)
    {
        sprintf(mat, "sar_cor%d_0%d.txt", n, ctagrav);
    }
    else
    {
        sprintf(mat, "sar_cor%d_%d.txt", n, ctagrav);
    }

    printf("Gravando SAR no arquivo %s para o corte Coronal j = %d
\n", mat, j);

    if (GRAVA_BORDA)
    {
        gravbry = TAM_BORDAY;
    }

    fp=fopen (mat, "w+");

    if (MATHLAB)

```

```

        fprintf(fp,"[");

for(i = TAM_BORDERDAY-gravbry ; i < nx+TAM_BORDERDAY+gravbry ; i++)
{
    for(k = kinic ; k < kfinal ; k++)
    {
        sar = vsar[i][j][k]/nrointer;
        fprintf(fp,"%0.2e ",sar);
    }
    if (MATHLAB)
        fprintf(fp,";\n");
    else
        fprintf(fp,"\n");
}
if (MATHLAB)
    fprintf(fp,"]");
fclose(fp);
}

/*****
*
*   FUNCAO:   grava_cpo_sag()
*
*   RECEBE:  NRO DO ARQUIVO, ITERACAO, NRO DO CORTE E SE E ANTENA OU CAMPO
*   RETORNA: NULL
*   FUNCAO:  Gravar em .TXT a Imagem SAGITAL da Cabeça com a SAR
*   OBS:     qualq = 1 -> campo
*
*****/
void grava_cpo_sag(int n, int indgrav,int k, int qualg)
{
    FILE *fp;
    char mat[100];
    int i,j,gravbrx,gravbry;

    gravbrx = 0;
    gravbry = 0;

    if (indgrav<10)
    {
        if (qualg)
            sprintf(mat,"s0%d00%d.txt",n,indgrav);
        else
        {
            if (n<10)
                sprintf(mat,"s0%d00%d.txt.txt",n,indgrav);
            else
                sprintf(mat,"s%d00%d.txt.txt",n,indgrav);
        }
    }
    else if (indgrav<100)
    {
        if (qualg)
            sprintf(mat,"cam_sag%d_0%d.txt",n,indgrav);
        else
        {
            if (n<10)
                sprintf(mat,"s%d0%d.txt.txt",n,indgrav);
            else

```



```

    }
}
else
{
    if (qualg)
        sprintf(mat,"cam_sag%d_%d.txt",n,indgrav);
    else
        sprintf(mat,"ant_sag%d_%d.txt",n,indgrav);
}

if (qualg)
    printf("Gravando campo no arquivo %s para o corte Sagital k =
%d \n",mat,k);
else
    printf("Gravando antena no arquivo %s para o corte Sagital k =
%d \n",mat,k);

if (GRAVA_BORDA)
{
    gravbrx = TAM_BORDAX;
    gravbry = TAM_BORDAY;
}

fp=fopen(mat,"w+");

if (MATHLAB)
    fprintf(fp,"[");

//    for(j = TAM_BORDAX-gravbrx ; j < ny+TAM_BORDAX+gravbrx ; j++) //
original so q de cab p/ baixo
    for(j = ny+TAM_BORDAX+gravbrx-1 ; j >= TAM_BORDAX-gravbrx ; j--)
    {
        for(i = TAM_BORDAY-gravbry ; i < nx+TAM_BORDAY+gravbry ; i++)
        {
            fprintf(fp,"%%.4e
",sqrt((ex[i][j][k]*ex[i][j][k])+(ey[i][j][k]*ey[i][j][k])+(ez[i][j][k]*ez[
i][j][k])));
        }
        if (MATHLAB)
            fprintf(fp,";\n");
        else
            fprintf(fp,"\n");
    }
    if (MATHLAB)
        fprintf(fp,"]");
    fclose(fp);
}

/*****
*
*   FUNCAO:   grava_cpo_fro()
*
*   RECEBE:   NRO ARQUIVO, ITERACAO, COLUNA E NRO DO CORTE INICIAL,
*             FINAL E SE E ANTENA OU CAMPO
*
*   RETORNA:  NULL
*   FUNCAO:   Gravar em .TXT a Imagem FRONTAL do Campo Eletromagnetico
*   OBS:
*
*****/

```

```

void grava_cpo_fro(int n, int ctagrav, int i ,int kinic, int kfinal, int
qualg)
{
    FILE *fp;
    char mat[100];
    int j,k,gravbrx;

    gravbrx = 0;

    if (ctagrav<10)
    {
        if (qualg)
            sprintf(mat,"cam_fro%d_00%d.txt",n,ctagrav);

        else
            sprintf(mat,"ant_fro%d_00%d.txt",n,ctagrav);
    }
    else if (ctagrav<100)
    {
        if (qualg)
            sprintf(mat,"cam_fro%d_0%d.txt",n,ctagrav);
        else
            sprintf(mat,"ant_fro%d_0%d.txt",n,ctagrav);
    }
    else
    {
        if (qualg)
            sprintf(mat,"cam_fro%d_%d.txt",n,ctagrav);
        else
            sprintf(mat,"ant_fro%d_%d.txt",n,ctagrav);
    }

    if (qualg)
        printf("Gravando campo no arquivo %s para o corte Frontal i =
%d \n",mat,i);
    else
        printf("Gravando antena no arquivo %s para o corte Frontal i =
%d \n",mat,i);

    if (GRAVA_BORDA)
    {
        gravbrx = TAM_BORDAX;
    }

    fp=fopen(mat,"w+");

    if (MATHLAB)
        fprintf(fp,"[");

    for(j = TAM_BORDAX-gravbrx ; j < ny+TAM_BORDAX+gravbrx ; j++)
    {
        for(k = kinic ; k < kfinal ; k++)
        {
            fprintf(fp,"%%.4e
",sqrt((ex[i][j][k]*ex[i][j][k])+(ey[i][j][k]*ey[i][j][k])+(ez[i][j][k]*ez[
i][j][k])));
        }
        if (MATHLAB)
            fprintf(fp,";\n");
        else
            fprintf(fp,"\n");
    }
}

```

```

    }
    if (MATHLAB)
        fprintf(fp, "]\n");
    fclose(fp);
}

/*****
*
* FUNCAO: grava_cpo_cor()
*
* RECEBE: NRO ARQUIVO, ITERACAO, LINHA, NRO DO CORTE INICIAL E FINAL,
*         E SE E ANTENA OU CAMPO
*
* RETORNA: NULL
*
* FUNCAO: Gravar em .TXT a Imagem CORONAL do Campo Eletromagnetico
*
* OBS:
*
*****/
void grava_cpo_cor(int n, int ctagrav, int j ,int kinic, int kfinal, int
qualg)
{
    FILE *fp;
    char mat[100];
    int i,k,gravbry;

    gravbry = 0;

    if (ctagrav<10)
    {
        if (qualg)
            sprintf(mat,"cam_cor%d_00%d.txt",n,ctagrav);
        else
            sprintf(mat,"ant_cor%d_00%d.txt",n,ctagrav);
    }
    else if (ctagrav<100)
    {
        if (qualg)
            sprintf(mat,"cam_cor%d_0%d.txt",n,ctagrav);
        else
            sprintf(mat,"ant_cor%d_0%d.txt",n,ctagrav);
    }
    else
    {
        if (qualg)
            sprintf(mat,"cam_cor%d_%d.txt",n,ctagrav);
        else
            sprintf(mat,"ant_cor%d_%d.txt",n,ctagrav);
    }

    if (qualg)
        printf("Gravando campo no arquivo %s para o corte Coronal j =
%d \n",mat,j);
    else
        printf("Gravando antena no arquivo %s para o corte Coronal j =
%d \n",mat,j);

    if (GRAVA_BORDA)
    {
        gravbry = TAM_BORDAY;
    }
}

```

```

fp=fopen(mat,"w+");

if (MATHLAB)
    fprintf(fp,"[");

for(i = TAM_BORDAY-gravbry ; i < nx+TAM_BORDAY+gravbry ; i++)
{
    for(k = kinic ; k < kfinal ; k++)
    {
        fprintf(fp,"%%.4e
",sqrt((ex[i][j][k]*ex[i][j][k])+(ey[i][j][k]*ey[i][j][k])+(ez[i][j][k]*ez[
i][j][k])));
    }
    if (MATHLAB)
        fprintf(fp,";\n");
    else
        fprintf(fp,"\n");
}
if (MATHLAB)
    fprintf(fp,"]");
fclose(fp);
}

/*****
*
* FUNCAO: aloca_memoria()
*
* FUNCAO: Emitir sinal de erro
*
*****/
void signal_handler_fpu(int *p)
{
    printf("Erro fpe!\n");
    exit(1);
}

/*****
*
* FUNCAO: signal_handler_sigsev()
*
* FUNCAO: Emitir sinal de erro
*
*****/
void signal_handler_sigsev(int *p)
{
    printf("Erro sigsev!!\n");
    exit(1);
}

/*****
*
* FUNCAO: erro_alocacao()
*
* FUNCAO: Emitir sinal de erro
*
*****/
void erro_alocacao(void)
{
    printf("Erro Alocando Matrizes!! - Provavelmente Falta de
Memoria.\n");
    exit(1);
}

```

```

}

/*****
*
* FUNCAO: grava_rel_sar()
*
* RECEBE: Hora Inicial, Variaveis espaciais & Temporais
* RETORNA: NULL
* FUNCAO: Gravar em .txt um relatorio da simulacao
* OBS:
*
*****/
void grava_rel_sar(double dx, double dy, double dz, double dt, double
cpepele, double cpomusc,
double cpoosso, double cpoolho, double
cpocereb, double cpoar,
int bytesalocados, double
nro_cel_borda, double log_g,
double berenger_g, double sigma_zero,
int nmax, double atingiu_borda,
double degringolou, int m_int_pto1, int
m_int_pto2, double m_val_pto1,
double m_val_pto2)
{
FILE *fp;
char mat[100];
double diff_time,vel_onda_c;

time_t horat,hora2;

vel_onda_c=0.0;

sprintf(mat,"relatsar.txt");

printf("Gravando Relatorio da Simulacao:\n");
fp=fopen(mat,"w+");

hora2 = time(NULL);
time(&horat);

diff_time = difftime(hora2,horacom);

fprintf(fp,"***** \n");
fprintf(fp,"UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL (UFRGS) \n");
fprintf(fp,"DEPARTAMENTO DE ENGENHARIA ELETRICA (DELET) \n");
fprintf(fp,"LABORATORIO DE COMUNICACOES ELETRO-OPTICAS (LACOM) \n");
fprintf(fp,"SIMULADOR FDTD VERSAO %s - UFRGS - 2001
\n",VERSAO);
fprintf(fp,"*****
\n\n");

fprintf(fp,"RELATORIO DA SIMULACAO: \n\n");
fprintf(fp,"INICIO DA SIMULACAO.....: %s",ctime(&horaini));
fprintf(fp,"DATA E HORA FINAL.....: %s",ctime(&horat));
fprintf(fp,"TEMPO TOTAL DA SIMULACAO...: %lg seg ou %lg min ou %lg
h\n"
, diff_time, diff_time/60, diff_time/3600);
fprintf(fp,"MEDIA DE.....: %lg seg/int ou %lg min/int
ou %lg h/int\n"

```

```

,diff_time/NUM_MAX_ITERACAO, (diff_time/60)/NUM_MAX_ITERACAO,
      (diff_time/3600)/NUM_MAX_ITERACAO);
    fprintf(fp, "MEMORIA REQUERIDA.....: %lg
MBytes\n\n", (bytesalocados/1000000.0)+100);

    if (QUAL_MATERIAL)
        fprintf(fp, "\nMATERIAL DE ESTUDO.....: CABECA\n\n");
    else
        fprintf(fp, "\nMATERIAL DE ESTUDO.....: AR\n\n");

    fprintf(fp, "NUMERO TOTAL DE ITERACOES..: %d iteracoes ou %lg
segundo(s)\n", NUM_MAX_ITERACAO, NUM_MAX_ITERACAO*dt);
    if (QUAL_MATERIAL)
        fprintf(fp, "CORTES DO MATERIAL ESTUDADO: %d pixel(s) ou %lg
m\n", TAM_DA_CAB, TAM_DA_CAB*dx);
        fprintf(fp, "ARESTA DO CUBO (dx=dy=dz)...: %lg m\n", dx);
        fprintf(fp, "LINHAS.....: %d pixel(s) ou %lg
m\n", ny, ny*dx);
        fprintf(fp, "COLUNAS.....: %d pixel(s) ou %lg
m\n", nx, nx*dx);
        fprintf(fp, "TAMANHO DA BORDA.....: %d pixel(s) ou %lg
m\n", TAM_BORDAZ, TAM_BORDAZ*dx);
        fprintf(fp, "VOLUME DO CUBO.....: %lg m^3\n", dx*dy*dz);
        fprintf(fp, "DIFERENCIAL DO TEMPO (dt)..: %lg s\n\n", dt);

    fprintf(fp, "\nDADOS DE CONVERGENCIA:\n\n");

    if (degringolou == 0)
        fprintf(fp, "HOUE CAMPO MAIOR QUE ZERO.: NAO\n");
    else
    {
        fprintf(fp, "CAMPO SAI DA CONVERGENCIA..: %lg
ITERACOES\n", degringolou);
        fprintf(fp, "TEMPO PARA SAIR DA CONVERG.: %lg
s\n", degringolou*dt);
        fprintf(fp, "DISTÂNCIA PERCORRIDA.....: %lg m ou %lg
comprimento(s) de onda\n"
                , LIGHT_SPEED*dt*degringolou
, ((LIGHT_SPEED*dt*degringolou)/(LIGHT_SPEED/FREQUENCIA)));
    }

    if (atingiu_borda == 0)
        fprintf(fp, "CAMPO ATINGE A BORDA.....: NAO\n\n");
    else
    {
        fprintf(fp, "CAMPO ATINGE A BORDA EM....: %lg
ITERACOES\n", atingiu_borda);
        fprintf(fp, "TEMPO PARA ATINGIR A BORDA..: %lg
s\n", atingiu_borda*dt);
        fprintf(fp, "DISTÂNCIA PERCORRIDA.....: %lg m ou %lg
comprimento(s) de onda\n\n"
                , LIGHT_SPEED*dt*atingiu_borda
, ((LIGHT_SPEED*dt*atingiu_borda)/(LIGHT_SPEED/FREQUENCIA)));
    }

    fprintf(fp, "\nDADOS DA ANTENA:\n\n");

    fprintf(fp, "ANTENA UTILIZADA.....: %s\n", TIPODANTENA);

```

```

    fprintf(fp, "FREQUENCIA.....: %lg Hz\n", FREQUENCIA);
    fprintf(fp, "COMPRIMENTO DE ONDA.....: %lg
m\n", LIGHT_SPEED/FREQUENCIA);
    fprintf(fp, "DISTÂNCIA PERCORRIDA.....: %lg m ou %lg comprimento(s)
de onda\n"
            , LIGHT_SPEED*dt*NUM_MAX_ITERACAO
, ((LIGHT_SPEED*dt*NUM_MAX_ITERACAO)/(LIGHT_SPEED/FREQUENCIA)));
    fprintf(fp, "CORTES DE ANTENA .....: %d corte(s) ou %lg m
\n", TAM_ANTENA, (double)TAM_ANTENA*dx);
    fprintf(fp, "DISTÂNCIA (ANTENA/MATERIAL): %d corte(s) ou %lg m
\n\n", TAM_AR2, (double)TAM_AR2*dx);

    fprintf(fp, "\nDADOS DA BORDA:\n\n");

    fprintf(fp, "TAMANHO MINIMO PARA A BORDA: %lg\n", nro_cel_borda);
    fprintf(fp, "LARGURA MAXIMA DO DOMINIO..: %d\n", nmax);
    fprintf(fp, "G.....: %lg\n", berenger_g);
    fprintf(fp, "LOGARITMO DE G.....: %lg\n", log_g);
    fprintf(fp, "SIGMA ZERO.....: %lg\n\n", sigma_zero);

    if (MEDE_VELOC || m_int_ptol !=0 )
    {
        fprintf(fp, "\nVELOCIDADE DA ONDA:\n\n");
        fprintf(fp, "VALOR MAX DO PONTO 1.....: %lg\n", m_val_ptol);
        fprintf(fp, "OCORRIDO NA ITERACAO.....: %d\n", m_int_ptol);
        fprintf(fp, "VALOR MAX DO PONTO 2.....: %lg\n", m_val_pto2);
        fprintf(fp, "OCORRIDO NA ITERACAO.....: %d\n", m_int_pto2);
        fprintf(fp, "DISTÂNCIA ENTRE OS PONTOS: %d\n", DELTA_DIST);
        fprintf(fp, "VARIACAO DE ITERACOES.....: %d\n", m_int_pto2-
m_int_ptol);
        vel_onda_c=((DELTA_DIST*dx)/(m_int_pto2-m_int_ptol)*dt);
        fprintf(fp, "VELOCIDADE DA ONDA.....: %lg\n", vel_onda_c);
        fprintf(fp, "VELOCIDADE DA LUZ.....: %lg\n", LIGHT_SPEED);
        fprintf(fp, "PERCENTUAL DE ERRO.....:
%lg%\n", ((vel_onda_c*100)/LIGHT_SPEED));
        vel_onda_c=((DELTA_DIST*dx)/(m_int_pto2-m_int_ptol-1)*dt);
        fprintf(fp, "PERCENTUAL DE ERRO (-1)..:
%lg%\n", ((vel_onda_c*100)/LIGHT_SPEED));
        vel_onda_c=((DELTA_DIST*dx)/(m_int_pto2-m_int_ptol+1)*dt);
        fprintf(fp, "PERCENTUAL DE ERRO (+1)..:
%lg%\n\n", ((vel_onda_c*100)/LIGHT_SPEED));
    }

    fprintf(fp, "\nCAMPO TOTAL NOS TECIDOS:\n\n");

    fprintf(fp, "PELE.....: %.4e\n", cpopele);
    fprintf(fp, "MUSCULO.....: %.4e\n", cpomusc);
    fprintf(fp, "OSSO.....: %.4e\n", cpoosso);
    fprintf(fp, "OLHO.....: %.4e\n", cpoolho);
    fprintf(fp, "CEREBRO.....: %.4e\n", cpocereb);
    fprintf(fp, "AR.....: %.4e\n", cpoar);
    fprintf(fp, "TOTAL.....:
%.4e\n\n", cpopele+cpomusc+cpoosso+cpoolho+cpocereb+cpoar);

    if (QUAL_MATERIAL)
    {
        fprintf(fp, "SAR TOTAL NOS TECIDOS: \n\n");

        fprintf(fp, "PELE.....:\n");

```

```

        fprintf(fp, "MUSCULO.....:\n");
        fprintf(fp, "OSSO.....:\n");
        fprintf(fp, "OLHO.....:\n");
        fprintf(fp, "CEREBRO.....:\n");
        fprintf(fp, "TOTAL.....:\n");
    }

    fprintf(fp, "\nHISTORICO DE GRAVACAO E CALCULOS:\n\n");

    if (MEDE_VELOC)
        fprintf(fp, "CALCULOU A VELOCIDADE DA ONDA\n");

    if ( RAO_POT > 0 )
        fprintf(fp, "CALCULOU POTÊNCIA DA ONDA\n");

    if (GRAVA_FRAME_SAR)
        fprintf(fp, "SAR\n");

    if (GRAVA_FRAME_E)
        fprintf(fp, "MODULO DO CAMPO ELETROMAGNETICO\n");

    if (GRAVA_ANTENA)
        fprintf(fp, "MODULO DO CAMPO ELETROMAGNETICO NA ANTENA\n");

    if (GRAVA_SAG)
        fprintf(fp, "GRAVOU OS CORTES SAGITAIS\n");

    if (GRAVA_FRO)
        fprintf(fp, "GRAVOU OS CORTES FRONTAIS\n");

    if (GRAVA_PTO)
        fprintf(fp, "GRAVOU O CAMPO DE UM DETERMINADO PONTO\n");

    if (GRAVA_COR)
        fprintf(fp, "GRAVOU OS CORTES CORONAIS\n");

    if (GRAVA_FRAME_SAR || GRAVA_FRAME_E || GRAVA_ANTENA)
    {
        fprintf(fp, "GRAVADO A CADA %d ITERACOES\n", INTERVALO);

        if (MATHLAB)
            fprintf(fp, "**TODOS NO FORMATO MATHLAB\n");
        else
            fprintf(fp, "**TODOS NO FORMATO LEBIT\n");

        if (GRAVA_BORDA)
            fprintf(fp, "**TODOS COM A BORDA NA FIGURA\n");
        else
            fprintf(fp, "**TODOS SEM A BORDA NA FIGURA\n");
    }

    fclose(fp);
}

```



## ANEXO2 LISTAGEM DA ROTINA DE INDEXAÇÃO DOS TECIDOS

```

/*****
*           UFRGS - LACOM
*
*           Programa que le um BMP e gera um arquivo .fd para determinacao
*           do material (tecido) para programa ftdt.
*
*****/
/*****
*
*           O arquivo .fd e definido como sendo texto, um numero por linha
*           x       - largura da imagem em pixels
*           y       - altura da imagem em pixels
*           data1
*           ...
*           datan
*-1      - indica nova linha ( nao era necessario, incluido por seguranca)
*
*****/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "cabbmp.h"

#define AR           0
#define PELE        1
#define MUSCULO     2
#define CEREBRO     3
#define OSSO        4
#define OLHO        5
#define BORDAX      5
#define BORDAY      6
#define CANTO       7
#define ANTENA      8

//#define NAO_REDEFINIR_CORES 1
#define DEBUG                          1

void ident_cores(BYTE *buffer,int x,int y,unsigned char *cores,int *n);
void subs_cores(BYTE *buffer,int x,int y);

void subs_cores(BYTE *buffer,int x,int y)
{
    int xx;
    int i;
    int achado ;
    int nao_achados = 0;

    for(i = 0 ; i < x*y ; i++)
    {
        achado = 0;
        switch(buffer[i])
        {
            case 0: // Preto

```

```
        buffer[i] = 0;
        achado = 1;
        break;
case 53: // azul
    buffer[i] = PELE;
    achado = 1;
    break;
case 134: // verde
    buffer[i] = OLHO;
    achado = 1;
    break;
case 192: // vermelho
    buffer[i] = MUSCULO;
    achado = 1;
    break;
case 143:
    buffer[i] = CANTO;
    achado = 1;
    break;
case 183:
    buffer[i] = BORDAY;
    achado = 1;
    break;
case 199:
    buffer[i] = BORDAX;
    achado = 1;
    break;
case 210:
    buffer[i] = CEREBRO;
    achado = 1;
    break;
case 215: // branco
    buffer[i] = OSSO;
    achado = 1;
    break;
case 24: // verde
    buffer[i] = ANTENA;
    achado = 1;
    break;
// CORES DIFERENTES PARA O AR (TONS DIFERENTES
DE PRETO)
case 36:
    buffer[i] = 0;
    achado = 1;
    break;
case 42:
    buffer[i] = 0;
    achado = 1;
    break;
case 43:
    buffer[i] = 0;
    achado = 1;
    break;
case 79:
    buffer[i] = 0;
    achado = 1;
    break;
case 216:
    buffer[i] = 0;
    achado = 1;
    break;
```

```

        case 217:
            buffer[i] = 0;
            achado = 1;
            break;
        case 218:
            buffer[i] = 0;
            achado = 1;
            break;
        case 219:
            buffer[i] = 0;
            achado = 1;
            break;
        case 220:
            buffer[i] = 0;
            achado = 1;
            break;
    }
    if( !achado )
    {
        buffer[i] = buffer[i-1];
        nao_achados++;
    }
}
printf("\n----> Nao achados = %d\n",nao_achados);
}

void ident_cores(BYTE *buffer,int x,int y,unsigned char *cores,int *n)
{
    int i;
    int j;
    int total=0;
    int contem;

    cores[0] = buffer[0];
    total++;
    for(i = 0 ; i < x*y ; i++)
    {
        contem = 0;
        for(j = 0 ; j < total ; j++)
            if( cores[j] == buffer[i] )
            {
                contem = 1;
                break;
            }
        if ( !contem )
        {
            cores[total] = buffer[i];
            total++;
        }
    }
    *n = total;
}

void main(int argc,char *argv[])
{
    BYTE buffer[400000];
    char *ref;

```

```

char tmp[100];
unsigned char cores[256];
int n;
int x,y,i,j;

FILE *fp;

if(argc < 2)
{
    printf("Erro - Usar %s nome_do_arquivo\n",argv[0]);
    exit(1);
}

printf("Buffer de 200 Kbytes\nLendo BMP...");
if( le_bmp(argv[1],&x,&y,buffer) == -1 )
{
    perror("Erro abrindo arquivo:");
    exit(0);
}
printf(" OK\n");

#ifdef DEBUG
    printf("x=%d y=%d \n",x,y);
    le_teste(argv[1]);
#endif

#ifdef NAO_REDEFINIR_CORES
{
    printf("Indexando cores.... ");
    subs_cores(buffer,x,y);
    printf(" OK\n");
}
#endif

ident_cores(buffer,x,y,cores,&n);
for(i = 0 ; i < n ; i++)
    printf("cores[%d]=%d\n",i,cores[i]);

if( (ref = strchr(argv[1],'.')) == NULL )
{
    printf("Usar nome_do_arquivo.bmp\n");
    exit(1);
}
memcpy(tmp,argv[1],ref - argv[1]);
tmp[ref - argv[1]] = 0;
strcat(tmp, ".fd");

printf("Escrevendo %s...",tmp );
if( (fp = fopen(tmp,"w+")) == NULL )
{
    perror("Erro abrindo arquivo:");
    exit(1);
}
printf(" OK\n");
fprintf(fp,"%d\n",x); // Numero de Colunas
fprintf(fp,"%d\n",y); // Numero de Linhas
i = 0;
j = 0;
do
{
    fprintf(fp,"%d\n",buffer[i]);

```

```
        i++;
        j++;
        if ( j == x )
            {
                fprintf(fp, "-1\n");
                j = 0;
            }
    } while( i != x*y );
fclose(fp);
}
```