

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES

**RE-ENGENHARIA DO SOFTWARE SCMS PARA UMA LINGUAGEM
ORIENTADA A OBJETOS (JAVA) PARA USO EM CONSTRUÇÕES DE
PHANTOMS SEGMENTADOS**

RAFAEL GUEDES POSSANI

Dissertação apresentada como parte
dos requisitos para obtenção do
Grau de Mestre em Tecnologia
Nuclear

Orientador: Prof^o Dr. Hélio Yoriyaz

SÃO PAULO
2012

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES

**RE-ENGENHARIA DO SOFTWARE SCMS PARA UMA LINGUAGEM
ORIENTADA A OBJETOS (JAVA) PARA USO EM CONSTRUÇÕES DE
PHANTOMS SEGMENTADOS**

RAFAEL GUEDES POSSANI

Dissertação apresentada como parte
dos requisitos para obtenção do
Grau de Mestre em Tecnologia
Nuclear

Orientador: Prof^o Dr. Hélio Yoriyaz

SÃO PAULO
2012

Aos meus pais, Maura Guedes Possani e Lourival Possani, e a minha irmã, Renata Guedes Possani, pelo incentivo e apoio para a conclusão deste trabalho.

AGRADECIMENTOS

Aos meus pais pelo apoio, carinho e preocupação.

Ao meu orientador Prof. Dr. Helio Yoriyaz, pela oportunidade, ensinamentos, dedicação e cuidado. Obrigado, Helinho.

À minha irmã, pelo incentivo, conselhos e algumas broncas merecidas.

A todos meus familiares, em especial para Antonia, Bruno, Natalia e Nicea.

Aos amigos: Ariane, André, César, Denis, Douglas, Felipe, Gabriel, Gregório, Jeferson, Lee, Leo, Lubianka, Milena, Omar, Paulinha, Pedro Julio, Rodrigo, Talita, Tássio, Thiago, Vanessa e Vinicius, pelas conversas, cafés, cinemas, risadas e brigadeiros.

Aos professores e funcionários do IPEN.

À CNEN pela bolsa de estudos.

RE-ENGENHARIA DO SOFTWARE SCMS PARA UMA LINGUAGEM ORIENTADA A OBJETOS (JAVA) PARA USO EM CONSTRUÇÕES DE PHANTOMS SEGMENTADOS

RAFAEL GUEDES POSSANI

RESUMO

Metodologias recentes de planejamento dependem fortemente de imagens de tomografia computadorizada e a tendência é que os procedimentos de dosimetria interna na terapia usando medicina nuclear também sejam baseados em imagens, tais como, imagens de ressonância magnética (RM) e tomografia computadorizada (TC), que extraem informações anatômicas e histológicas, bem como, imagens funcionais ou mapas de atividades, provenientes de PET e SPECT. Estas informações, associadas a um software de transporte de radiação, são utilizadas para estimar a dose interna em pacientes submetidos a tratamento em medicina nuclear. Este trabalho visa a re-engenharia do SCMS, que é um software de interface entre o código MCNP e as imagens médicas, que carregam as informações do paciente em tratamento. Em outras palavras, as informações necessárias contidas nas imagens são interpretadas e apresentadas em um formato específico para o código MCNP, que executa a simulação do transporte de radiação. Portanto, o usuário não precisa compreender o complexo processo de introdução de parâmetros do MCNP, pois o SCMS é responsável por construir automaticamente dados anatômicos do paciente, bem como, os dados da fonte radioativa. O SCMS foi originalmente desenvolvido em Fortran-77 e neste trabalho, ele foi reescrito em uma linguagem orientada a objetos (JAVA). Novas funcionalidades e opções de dados também foram incorporadas ao software. Assim, o novo software tem uma série de melhorias, tais como interface gráfica intuitiva e um menu para a seleção do espectro de energia correspondente a um radioisótopo específico, armazenado em um banco de dados XML. A nova versão também trabalha com uma maior quantidade de materiais e o usuário pode especificar uma região de interesse na tomografia computadorizada para o cálculo da dose absorvida.

RE-ENGENHARIA DO SOFTWARE SCMS PARA UMA LINGUAGEM ORIENTADA A OBJETOS (JAVA) PARA USO EM CONSTRUÇÕES DE PHANTOMS SEGMENTADOS

RAFAEL GUEDES POSSANI

ABSTRACT

Recent treatment planning systems depend strongly on CT images and the tendency is that the internal dosimetry procedures in nuclear medicine therapy be also based on images, such as magnetic resonance imaging (MRI) and computed tomography (CT), to extract anatomical and histological information, as well as, functional imaging or activities map as PET and SPECT.

This information associated with a radiation transport simulation software is used to estimate internal dose in patients undergoing treatment in nuclear medicine. This work aims to re-engineer the software SCMS, which is an interface software between the Monte Carlo code MCNP, and the medical images, that carry information from the patient in treatment. In other words, the necessary information contained in the images are interpreted and presented in a specific format to the Monte Carlo MCNP code to perform the simulation of radiation transport. Therefore, the user does not need to understand the complex process of inputting data on MCNP, as the SCMS is responsible for automatically constructing anatomical data from the patient, as well as the radioactive source data. The SCMS was originally developed in Fortran-77. In this work it was rewritten in an object-oriented language (JAVA). New features and data options have also been incorporated into the software. Thus, the new software has a number of improvements, such as intuitive GUI and a menu for the selection of the energy spectra correspondent to a specific radioisotope stored in a XML data bank. The new version also supports new materials and the user can specify an image region of interest for the calculation of absorbed dose.

SUMÁRIO

	Página
1. INTRODUÇÃO.....	12
1.1 Medicina Nuclear	12
1.2 Imagens em Medicina Nuclear	13
1.3 Dosimetria Interna	15
1.4 Objetivo	17
2. MATERIAIS E MÉTODOS.....	18
2.1 O Método de Monte Carlo.....	18
2.2 O código de Monte Carlo MCNP5	21
2.2.1 Estrutura dos dados de entrada do MCNP5.....	22
2.2.2 Modelagem da geometria	25
2.2.3 Modelagem da fonte de radiação.....	33
2.2.4 O cálculo de dose: <i>Tallies</i>	42
2.3 O Software de Interface SCMS	46
2.4 Estrutura do Software SCMS	47
2.5 A linguagem de programação JAVA.....	50
2.6 Objetos Simuladores.....	52
2.7 O programa ImageJ	53
2.8 Arquivos <i>DICOM</i>	54
2.9 O programa ICCT	55
2.10 O programa ACPS	55
3. RESULTADOS	55
3.1 O software AMIGO 1.0	59
3.2 O software AMIGO 2.0.....	63
3.3 Validação do software AMIGO 1.0.....	70
3.4 Diferenças nos arquivos de saída dos softwares.....	80
3.5 Validação do software AMIGO 2.0.....	81
4. CONCLUSÃO.....	84
5. APÊNDICE A	86
6. REFERÊNCIAS BIBLIOGRÁFICAS	92

LISTA DE TABELAS

Tabela 1 - Lista com alguns mnemônicos para a construção de superfícies no MCNP.	22
Tabela 2 - Lista dos mnemônicos para uso nos <i>tallies</i>	43
Tabela 3 - Casos mais simples de testes dos softwares.	71
Tabela 4 - Casos de <i>benchmark</i> do software	75
Tabela 5 - Casos de <i>benchmark</i>	79
Tabela 6 - Caso 8: <i>benchmarks</i> para validação do AMIGO 2.0.....	82

LISTA DE FIGURAS

Figura 1 - Metodologia de cálculo de distribuição de dose com uso de imagens médicas específicas de cada paciente em tratamento acoplado ao código de Monte Carlo MCNP.	16
Figura 2 - Número de trabalhos publicados na revista <i>Physics in Medicine and Biology</i> ou na <i>Medical Physics</i> com o termo "Monte Carlo" no resumo ou no título. (ROGERS, 2006).....	18
Figura 3 - Células unitárias simples. O conjunto de células de mesmo tipo pode representar um órgão ou tecido.	23
Figura 4 - Geometria composta de diferentes células que compõe diferentes regiões representando diferentes tecidos do corpo humano.....	24
Figura 5 - Exemplo de um arquivo de entrada do MCNP.....	25
Figura 6 - Exemplo de uso dos mnemônicos <i>like</i> , <i>but</i> e <i>trcl</i>	26
Figura 7 - Reprodução tridimensional de uma geometria composta de 27 cubos.....	27
Figura 8 - Dados de entrada do MCNP correspondente à geometria do cubo que conterá 27 cubos menores formando um arranjo de 3x3x3.....	28
Figura 9 - Dados de entrada com estruturas repetidas da geometria do cubo menor (célula 2).	28
Figura 10 - Vinte e sete cubos construídos usando dois tipos de estruturas repetidas: ..	29
Figura 11 - Visualização de um exemplo de geometria com estruturas repetidas composto de 27 cubos. Os cubos do primeiro plano possuem materiais diferentes dos demais.....	30
Figura 12 - Exemplo do uso dos mnemônicos <i>mat</i> e <i>rho</i> na repetição de estruturas geométricas.....	30
Figura 13 - Geometria de 27 cubos utilizando estruturas repetidas e empregando o conceito de universo	32
Figura 14 - Utilizando estruturas repetidas.....	33
Figura 15 - Definindo estruturas repetidas usando atalhos.....	33
Figura 16 - Bloco 1 e Bloco 2 da simulação base para os exemplos sobre fontes.	36
Figura 17 - Fonte isotrópica emissora de fótons.....	37
Figura 18 - Fonte pontual com valores variáveis de energia.....	38
Figura 19 - Fonte com energia em histogramas	39
Figura 20 - Fonte com energia constante localizada em duas células distintas.....	39
Figura 21 - Fonte de radiação composta por duas células cúbicas dentro de um arranjo cúbico usando estruturas repetidas	40
Figura 22 - Sintaxe de referência a células.....	41
Figura 23 - Exemplo de referência às células dentro de um contexto de estruturas repetidas.....	42
Figura 24 - <i>Tally</i> para cálculo do fluxo médio de nêutrons numa superfície	44
Figura 25 - <i>Tally</i> que fornece três resultados para correntes de fótons	44
Figura 26 - Corrente de nêutrons.....	44
Figura 27 - Exemplo de <i>Tally</i> para estruturas repetidas	44
Figura 28 - <i>Tally</i> retangular usando estruturas repetidas.....	45
Figura 29 - <i>Tally</i> retangular com células variadas.....	45
Figura 30 - Diagrama esquemático do software SCMS.	47
Figura 31 - Objeto simulador humano matemático (RADAR, 2011).	52
Figura 32 - Implementação da funcionalidade “ <i>labeled break</i> ” em Java.....	56
Figura 33 - Exemplo de código utilizando laço em Fortran 77	56
Figura 34 - Objeto simulador de <i>voxel</i> cujo arquivo está em formato ASCII.....	58

Figura 35 - Objeto simulador de um paciente específico	58
Figura 36 - Arquivo de entrada de dados, SINP, do software SCMS.	59
Figura 37 - Tela inicial do software AMIGO 1.0.....	60
Figura 38 - Detalhe da caixa de seleção do software AMIGO 1.0.....	61
Figura 39 - Escolhendo um espectro de energia de emissão da fonte radioativa.....	62
Figura 40 - Tela de gerenciamento de materiais.....	63
Figura 41 - Metodologia para o cálculo de dose implementada para uso com o software AMIGO 2.0.....	64
Figura 42 - Tela inicial do segundo software do sistema AMIGO.....	65
Figura 43 - Tela para escolha da região de interesse para o cálculo da distribuição de dose.....	66
Figura 44 - Tela de seleção dos arquivos para o cálculo da dose.....	68
Figura 45 - Tela da aplicação <i>Doseview</i>	68
Figura 46 - <i>Doseview</i> com outros tipos de exibição.....	69
Figura 47 - Construção tridimensional do objeto simulador do Caso A1, Caso B1 e Caso C1	73
Figura 48 - Construção tridimensional dos objetos simuladores correspondentes aos casos A2, B2 e C2.	74
Figura 49 - Construção tridimensional do objeto simulador do Caso A3	76
Figura 50 - Construção tridimensional do objeto simulador do quarto grupo de testes.	77
Figura 51 - Construção do objeto simulador MAX.....	78
Figura 52 - Construção tridimensional do objeto simulador FAX	78
Figura 53 - Representação tridimensional do objeto simulador do caso teste 7.....	80
Figura 54 - Diferenças de espaçamento entre os valores nos arquivos de saída.	80
Figura 55 - Diferenças na exibição dos números	81
Figura 56 - Pequena diferença encontrada na descrição da fonte.....	81

LISTA DE SIGLAS

ACPS	Software para a conversão de DICOM para ASCII
AMIGO	<i>Algorithm for Medical Image Generating Object</i>
ASCII	<i>American Standard Code for Information Interchange</i>
BIN	Cada um dos componentes de um histograma
Célula	Conjunto de uma ou mais superfícies no MCNP
CT	<i>Computed Tomography</i>
FDP	Função de densidade de probabilidade
FILL	Comando do MCNP referente ao Universo de preenchimento
ICCT	<i>Image Converter Computed Tomography</i>
keV	Kilo elétron volt
LAT	Comando do MCNP referente ao tipo de preenchimento
mat	Comando do MCNP referente ao material
MCNP	<i>Monte Carlo N-Particle</i>
MeV	Mega elétron volt
PET	<i>Positron emission tomography</i>
PHANTOM	Objeto simulador
RHO	Comando do MCNP referente à densidade
RM	Ressonância magnética
SCMS	Software para Construção de Manequins Segmentados
SPECT	<i>Single photon emission computed tomography</i>
Tally	Acumulador estatístico
TC	Tomografia computadorizada
TRCL	Comando do MCNP para transformações na célula
Universo	Conjunto de uma ou mais células no MCNP
VOXEL	Pixel volumétrico
XML	Extensible Markup Language

1. INTRODUÇÃO

1.1 Medicina Nuclear

A medicina nuclear fornece imagens fisiológicas e funcionais do corpo humano sem a necessidade de procedimentos invasivos. Essas imagens mostram a funcionalidade das regiões internas do corpo humano realçadas por um contraste que pode ser iônico ou não iônico. Contrastes iônicos também são chamados de radiofármacos que são medicamentos contendo um ou mais radionuclídeos que, quando decaem, liberam radiação. Esta radiação então é captada por equipamentos eletrônicos e convertida em imagem. Os radiofármacos usados na medicina nuclear, tanto para diagnóstico por imagem quanto para terapia, são administrados no paciente via ação intravenosa, inalatória ou oral. Ao decair, o radionuclídeo presente no radiofármaco emite radiação gama com energia suficiente para atravessar o corpo do paciente sem sofrer muitas interações (espalhamento ou atenuação, por exemplo) até alcançar o sistema de detecção.

Por causa da radiação, a estrutura mínima de um centro de medicina nuclear envolve um local blindado, onde os radiofármacos são preparados; uma sala de realização de exames e; em casos de instalações que também ofereçam terapia, quartos de internação devidamente blindados para a proteção do público em geral, como acompanhantes ou médicos, por exemplo.

Os princípios básicos em medicina nuclear começaram a ser estabelecidos no final do século XIX com a descoberta da radioatividade natural por Henri Becquerel, em 1896, e de elementos radioativos por Marie e Pierre Curie, em 1898. Por essas descobertas os três cientistas ganharam o Prêmio Nobel de Física de 1903.

O princípio do traçador proposto em 1913 por George Charles de Hevesy fez com que a medicina nuclear tornasse possível. Ele confirmou o princípio através de experiências com nitrato de chumbo marcado com o nuclídeo radioativo ^{210}Pb , mostrando sua absorção e seu movimento em plantas. Por esse feito, Hevesy recebeu o Prêmio Nobel de Química de 1943. (De Hevesy, 1913)

Em 1930, Ernest Lawrence inventou o ciclotron (Lawrence e Livingstone, 1932), tornando possível o desenvolvimento de novos radionuclídeos, através do bombardeamento de núcleos-alvos por partículas positivas aceleradas, aumentando

assim o uso diagnóstico da medicina nuclear. O desenvolvimento de radiofármacos em larga escala só teve início com a construção de reatores nucleares, como o de Oak Ridge, nos Estados Unidos em 1946, e o de Harwell (Reino Unido), em 1947. (Robilotta, 2006)

Benedict Cassen (Cassen et al, 1951) construiu o primeiro mapeador retilíneo. Este dispositivo permitia a análise funcional de diversos órgãos e sistemas do corpo humano em duas dimensões. O mapeador retilíneo juntamente com a Câmara de cintilação de Hal Anger (Anger, 1958) foram os maiores marcos que revolucionaram os sistemas de imagens diagnósticas em medicina nuclear.

Com a introdução do radionuclídeo ^{99m}Tc como marcador, na década de 60, a medicina nuclear diagnóstica ganhou força; seu uso foi iniciado por Paul Harper e sua equipe. O ^{99m}Tc possui meia vida de 6 horas com emissão de radiação gama de baixa energia, fóton-gama com 140.511 keV, e facilidade em marcar fármacos, pois não é processado pelo organismo e permanece no sangue, possibilitando a análise do fluxo sanguíneo, o que o torna aplicável em estudos de quase todos os órgãos e sistemas do corpo humano. Por causa destas qualidades ele é o radionuclídeo mais utilizado em medicina nuclear (Cherry et al, 2003).

A partir da década de 70, com o crescimento da informática, novos dispositivos de diagnose por imagem foram desenvolvidos, tais como, o sistema de tomografia computadorizada por emissão de fótons únicos (*single photon emission computed tomography*), SPECT, e o tomógrafo por emissão de pósitrons, chamado de PET (*positron emission tomography*).

Atualmente a pesquisa para melhorar a instrumentação, com o uso de detectores mais eficientes e de eletrônica mais rápida, tem impulsionado tanto SPECT quanto PET em suas aplicações. Em especial, destaca-se o PET/CT, que é a combinação de PET com a tomografia computadorizada (CT) desenvolvido por David W. Townsend e sua equipe da Universidade de Pittsburgh (Beyer et al, 2000).

1.2 Imagens em Medicina Nuclear

O sistema de detecção mais comum usado em medicina nuclear é a Câmara de cintilação também chamada câmara de Anger. Sua configuração básica envolve um cristal cintilador e tubos fotomultiplicadores. O cristal cintilador detecta o fóton,

proveniente do decaimento do radiofármaco, este sinal, fótons, é registrado pelos fotomultiplicadores e um sistema eletrônico associado que computa as diferentes intensidades medidas dos fótons. Um colimador é usado entre o paciente e o detector para eliminar raios gama que não tenham direção perpendicular a ele (o que torna a imagem mais nítida). As imagens são produzidas com a ajuda de um computador integrado ao equipamento. (POWSNER, 2006)

As câmaras de cintilação possuem colimadores que são constituídos geralmente de chumbo. Tem formato de placas com uma determinada espessura e alguns orifícios. Eles funcionam como uma espécie de filtro de fótons espalhados para que a construção das imagens seja realizada a partir de sinais gerados por fótons que colidam com os detectores num ângulo mais perpendicular possível para garantir a boa qualidade da imagem final.

Imagens funcionais de SPECT são tridimensionais e tipicamente apresentadas em cortes transversais do paciente. A imagem é formada pela emissão de fótons provenientes do decaimento de radionuclídeos ligados a um fármaco específico e que são detectados por um sistema de detecção externo ao paciente.

O processo de aquisição das imagens é semelhante ao da câmara de cintilação, porém várias imagens planares do paciente são feitas e unidas matematicamente para criar a imagem tridimensional através de um algoritmo computacional de reconstrução.

Existem vários tipos de tomógrafos SPECT, alguns modelos possuem apenas um detector de raios gama, enquanto outros modelos, mais novos, possuem vários cabeçotes de detecção em formato de anel ao redor do paciente.

Assim como a câmara de cintilação, tomógrafos por emissão de fótons únicos também possuem colimadores na parte frontal de seus detectores para impedir a detecção de fótons espalhados.

Uma imagem PET corresponde a um mapa da distribuição de um radiofármaco emissor de pósitrons no corpo. Pósitrons emitidos pelos radiofármacos distribuídos pelo corpo sofrem aniquilação com um elétron. Esta aniquilação produz um par de fótons que são emitidos em direções opostas que por fim são captados por pequenos cristais de cintilação. Esses detectores formam uma espécie de anel ao redor do paciente para que cada par de fótons seja captado por cristais em posições opostas e em coincidência. Através de um algoritmo computacional de reconstrução similar ao SPECT a posição de aniquilação do pósitron é estimada e registrada na imagem PET.

A tomografia PET tem uma série de vantagens quando comparada à SPECT, sendo elas a alta resolução e sensibilidade, podendo chegar a um fator de 100 devido à ausência de colimadores. As principais desvantagens são seu alto custo e a baixa meia vida dos radiofármacos emissores que acarreta em maiores custos com equipamentos para sua fabricação e transporte. A baixa meia vida também tem sua vantagem ao paciente, pois o radiofármaco é rapidamente eliminado do seu organismo. (POWSNER, 2006)

1.3 Dosimetria Interna

Progressos na última década em relação à aquisição de imagens médicas e inovações técnicas na reconstrução destas imagens tem permitido novos avanços em diagnósticos e terapias em medicina nuclear (Badger e Fisher, 1994). Modalidades de imagens como tomografia computadorizada (CT) e ressonância magnética (RM) tem sido amplamente utilizadas para a visualização de áreas anatômicas, estudo de regiões de interesse e obtenção da geometria para sistemas de planejamento. A geometria consiste normalmente de uma matriz tridimensional cujos elementos são pequenos volumes denominados de “*voxels*”. Cada um destes elementos de volume é composto por diferentes materiais caracterizados pelas suas seções de choque (Li e Williamson, 1992).

Alguns autores tem realizado pesquisas em dosimetria utilizando-se dos recursos de CT e RM além de imagens funcionais fornecidas por SPECT e PET que fornecem a distribuição de radionuclídeos internamente em um paciente. (Sgouros, G. et AL, 1990; Jonsson, L. ET AL, 2000; Yoriyaz e Stabin, 1998). Nestas metodologias, a dosimetria é feita de forma específica para cada paciente utilizando-se, na maioria dos casos, técnicas de Monte de Carlo para a simulação do transporte de radiação e consequentemente a deposição de energia no meio. Dentre os códigos mais utilizados destacam-se o EGS4 (Nelson et al, 1985) e o código MCNP (Briesmeister, 2001). Este último tem sido amplamente utilizado em cálculos de reatores nucleares na divisão de física de reatores do IPEN, Instituto de Pesquisa Energéticas e Nucleares, para simulação de transporte de nêutrons e, mais recentemente, tem sido aplicado em cálculos de dose absorvida para aplicações médicas.

Mais recentemente, o MCNP foi incorporado como parte de uma metodologia para cálculos dosimétricos em planos de tratamento utilizando-se dados específicos de pacientes, através do uso de imagens médicas como CT e SPECT. Estas duas modalidades de imagens passam por um processo de fusão e posteriormente são armazenados em forma de arquivos de dados. Estes arquivos, por sua vez, são processados por um software de interface chamado SCMS (Software para Construção de Manequins Segmentados) (Yoriyaz, 2001; Stabin, 2002; Yoriyaz, 2000), que os converte no formato de dado de entrada no código MCNP para a realização do transporte de radiação e posteriormente o cálculo de distribuição de dose. Esta metodologia está apresentada de forma esquemática na Figura 1.

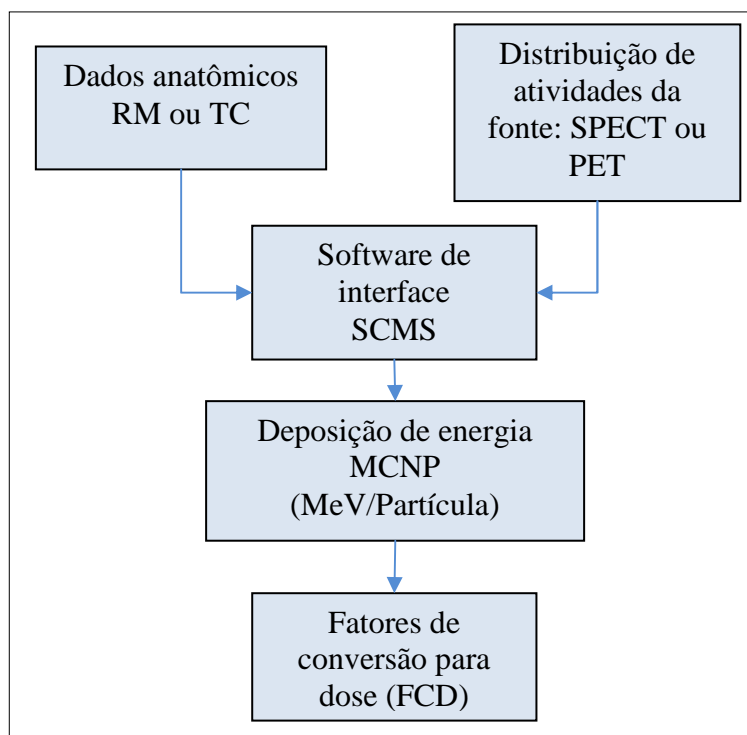


Figura 1 - Metodologia de cálculo de distribuição de dose com uso de imagens médicas específicas de cada paciente em tratamento acoplado ao código de Monte Carlo MCNP.

Metodologias de cálculo de distribuição de dose em medicina nuclear estão em constante desenvolvimento, visto que os modelos atuais são baseados em imagens médicas e algumas delas utilizam o Método de Monte Carlo para a simulação do transporte de radiação. No IPEN existe atualmente em desenvolvimento um sistema de cálculo de dose com o uso de imagens médicas e programas computacionais que usam o código de Monte Carlo MCNP.

Diversos outros programas estão atualmente em desenvolvimento com o intuito de formar um sistema integrado de cálculo de dose. Dentre estes programas, o SCMS tem a função de gerar um arquivo de dados de entrada contendo todas as informações anatômicas e da fonte de radiação do objeto simulador. Este arquivo de dados é gerado no formato específico para o programa de Monte Carlo MCNP5 para a simulação do transporte de radiação.

1.4 Objetivo

Originalmente, o programa SCMS foi escrito em FORTRAN 77 e com recursos computacionais limitados. Neste sentido, o objetivo geral do presente trabalho consiste em reescrever o software SCMS na linguagem Java, introduzindo novas funcionalidades computacionais e recursos gráficos, que não existem na versão atual.

Os objetivos específicos estão descritos abaixo:

- Desenvolver uma Interface Gráfica que forneça os dados necessários de forma mais fácil e eficaz;
- Adicionar opções para apresentação dos resultados através de recursos gráficos, por exemplo, a apresentação da distribuição de dose em forma de imagens;
- Desenvolver funções que possibilitem a especificidade do Software SCMS, ou seja, deixá-lo apto a trabalhar com imagens do próprio paciente.
- Criar um banco de dados com radiofármacos utilizados em medicina nuclear e possibilitar ao usuário adicionar outros;
- Tornar o SCMS um programa integrante de um sistema de cálculo de distribuição de dose para uso geral em radioterapia e medicina nuclear.

Com a criação de uma versão inteiramente nova do programa SCMS será possível sua utilização em diversas plataformas computacionais, além do fato de integrar o sistema de cálculo de distribuição de dose atualmente em desenvolvimento.

2. MATERIAIS E MÉTODOS

2.1 O Método de Monte Carlo

Ao longo dos últimos anos, a solução de problemas na área de ciências radiológicas utilizando-se técnicas de Monte Carlo tem crescido significativamente. Este crescimento é diretamente relacionado ao desenvolvimento de computadores cada vez mais rápidos e baratos, sendo acessíveis a todas as áreas do conhecimento.

Esta tendência é facilmente observada quando se consulta o número de artigos publicados com a palavra Monte Carlo no título ou no resumo de artigos de 2 periódicos na área de física médica, como é mostrado na Figura 2.

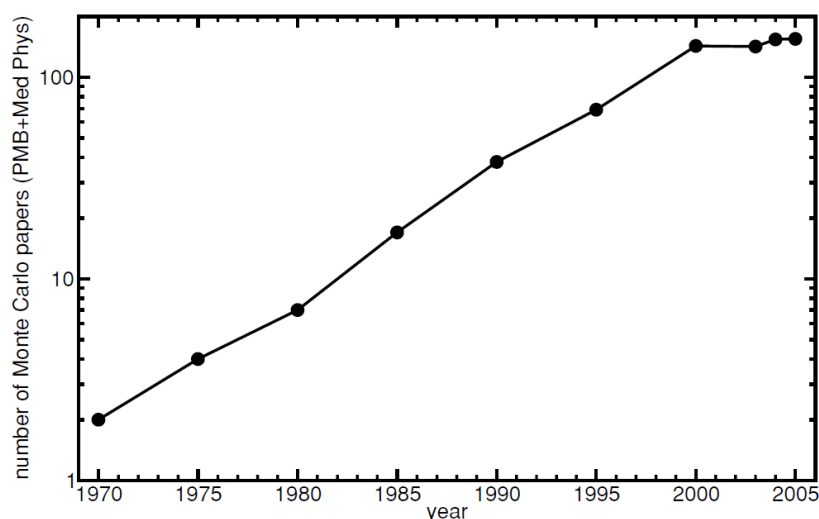


Figura 2 - Número de trabalhos publicados na revista *Physics in Medicine and Biology* ou na *Medical Physics* com o termo "Monte Carlo" no resumo ou no título. (ROGERS, 2006)

O método de Monte Carlo é um método estocástico que usa regras probabilísticas, sendo bastante utilizado em problemas complexos que não podem ser resolvidos com a mesma precisão se utilizado métodos determinísticos. O uso do método de Monte Carlo atinge várias áreas do conhecimento, tais como, o controle de tráfego de aviões e trens, jogos de estratégia, simuladores de conflitos militares, interação da radiação com a matéria, etc.

Para o uso do método de Monte Carlo existe apenas uma necessidade, esta necessidade consiste no fato de que o comportamento físico do processo a ser simulado

deve ser descrito por meio de funções de densidade de probabilidade, FDP, caso não exista essa possibilidade o método não poderá ser aplicado.

A partir das FDPs, os eventos probabilísticos individuais que compreendem um processo são simulados sequencialmente e seu processo de amostragem estatística é baseado na seleção de números aleatórios. (LOUREIRO, 2002)

De acordo com Hammersley (Hammersley, 1964) o nome "Monte Carlo" foi criado por John von Neumann, Stanislaw Ulam e Nicholas Metropolis enquanto trabalhavam no projeto Manhattan durante a Segunda Guerra Mundial nos laboratórios de Los Alamos, Estados Unidos da América, em referência a capital de Mônaco, conhecida pelos jogos de azar.

O método de Monte Carlo compreende uma área muito ampla da ciência, onde muitos processos, sistemas físicos e fenômenos são estudados. Neste trabalho o método de Monte Carlo é usado para simular o transporte da radiação ionizante através da matéria, onde o processo de simulação é realizado através da criação de partículas numa região, que é a fonte de radiação, geralmente com distribuição espacial aleatória (origem aleatória dentro da fonte). Em seguida, as partículas criadas são acompanhadas através de todo o sistema geométrico que representa a anatomia do paciente. Sempre que ocorre uma interação da partícula com a matéria, novos números aleatórios são gerados e através das funções de densidade de probabilidade avaliam-se quais serão as novas direções e as energias da partícula ou das novas partículas que venham a surgir.

A partir das interações determinam-se a penetração, a direção das partículas e a energia depositada para o cálculo da dose absorvida, que é a razão entre a energia depositada e a massa da região de interesse.

A precisão nos cálculos de dose está relacionada com o número de interação naquele volume de interesse, as principais discussões a respeito da simulação do transporte de radiação consistem em quantas histórias, ou horas de simulação, são necessárias para se obter uma incerteza aceitável, que geralmente está em torno de 5% e não maior que 10%. Estas discussões também estão relacionadas à exatidão dos dados, ou seja, quão bem o caso real pode ser simulado pelo modelo de geometria considerado.

Em geral, um problema que pode ser tratado com o método de Monte Carlo é determinado por um conjunto de parâmetros que se agregados determinam completamente o comportamento de um elemento do sistema em todas as situações modeladas dentro daquele sistema.

Em aplicações da técnica Monte Carlo na terapia com medicina nuclear, o usuário tem o interesse em modelar a criação e a interação de todas as partículas, até sua “morte”. Partículas são criadas, geralmente, em posições aleatórias dentro da região da fonte de radiação (podendo ser um órgão inteiro ou uma porção do corpo), com orientação inicial definida. As partículas viajam distâncias aleatórias antes de interagirem com átomos dentro dos tecidos de um corpo, depositando energia nestas interações, e, eventualmente, sendo absorvidas completamente ou escapando do corpo.

A completa caracterização de uma fonte de radiação irá incluir sua distribuição espacial, a distribuição angular da emissão do espectro e sua distribuição energética. Uma vez que uma partícula é criada e começa sua jornada para fora da fonte, a localização da primeira interação deve ser escolhida. Isso é feito através da amostragem da distribuição de probabilidade desta interação com os átomos do corpo. As probabilidades de interação de um fóton são bastante conhecidas: o efeito fotoelétrico, espalhamento Compton, Rayleigh, interação foto-nuclear e a produção de pares. Estes eventos ocorrem com probabilidades muito bem definidas de acordo com cada material com diferentes elementos químicos. Este processo geralmente é computacionalmente intenso envolvendo muitos cálculos, contudo, com o desenvolvimento de processadores cada vez mais rápidos, o tempo de simulação necessário para o término da simulação torna-se um problema cada vez menos importante.

Aplicações do processo de Monte Carlo para modelagem de qualquer tipo de sistema, mesmo os de complexidade moderada, não são possíveis de serem realizados sem o uso de um computador. Tabelas de números aleatórios foram impressas durante anos antes da ampla disponibilidade dos computadores para pesquisadores e para a comunidade científica em geral.

O uso de tais tabelas de números aleatórios para amostrar emissões de uma fonte de radiação e a interação das partículas seria, obviamente, uma tarefa trabalhosa e entediante ao extremo, e sem a possibilidade de gerar nenhum resultado satisfatório dentro de um tempo razoável. Isto mostra que o método de Monte Carlo pode ser utilizado mesmo sem computadores, embora seja muito mais trabalhoso.

O método de Monte Carlo alcançou ainda mais notoriedade na ciência quando os computadores se tornaram disponíveis para a comunidade científica em geral.

Atualmente o desempenho computacional é bastante alto, computadores que ocupam uma pequena parte de uma mesa possuem desempenho muito maior que

computadores que ocupavam uma sala inteira há algumas décadas atrás. Estações de trabalho de alto desempenho e computação distribuída continuam a aumentar o poder computacional em ordens de grandeza a cada década. Por este motivo as habilidades de investigação para modelar processos microscópicos, como o transporte de radiação, ou macroscópicos, como padrões climáticos, estão limitadas apenas a imaginação dos pesquisadores e à disponibilidade das funções densidade de probabilidade.

2.2 O código de Monte Carlo MCNP5

O código MCNP (*Monte Carlo N-Particle*), desenvolvido nos laboratórios de Los Alamos nos Estados Unidos da América, é um programa computacional baseado no Método de Monte Carlo para simulação do transporte de radiação no meio. (BRIESMEISTER, 2001)

Na sua atual versão, permite a simulação da interação de 3 tipos de radiação com a matéria, a saber: nêutron, fóton e elétron, e possui certas vantagens que o tornam atrativo em aplicações médicas. A faixa possível de energia de fótons e elétrons estende-se desde 1 keV até 100 MeV. Importantes modelos físicos de interações que ocorrem a baixas energias, como a produção de raios X e elétrons Auger, são considerados precisos para simulação de aquisição de imagens médicas, muito utilizadas em projetos de tomógrafos.

Em relação à geometria, este código possui uma vasta flexibilidade de modelagens geométricas que permite o uso simultâneo de recursos de geometria combinatória, com superfícies de até quarto grau em conjunto com recursos de estruturas repetidas, que permitem a reprodução geométrica exata de objetos digitalizados como no caso de imagens médicas.

O código MCNP tem sido utilizado com sucesso na solução de muitos problemas na área de física médica, visto que um número crescente de pesquisadores tem adotado este programa em cálculos de dosimetria numérica aplicada aos planos de tratamento em radioterapia, braquiterapia e radioimunoterapia (Solberg et al, 2001). Como exemplo, pode-se citar a utilização do código MCNP na reprodução do espectro de Bremsstrahlung gerado em aceleradores lineares em radioterapia (DeMarco, 1995); determinação de taxas de dose de fontes em braquiterapia (Mason, 1992); obtenção de propriedades dosimétricas para novas fontes em braquiterapia intravascular (Campos,

2000) e também na área de radioimunoterapia (RIT) e terapia por captura de nêutrons (BNCT). (Stabin, 2000; Yoriyaz, 2000; Evans, 2001)

O software SCMS, que foi inteiramente reescrito no presente trabalho, e cuja principal função é criar um arquivo de dados de entrada para o código MCNP5, baseia-se integralmente na estrutura de dados deste código; portanto, nas seções a seguir estão apresentadas as descrições detalhadas da estrutura de dados de entrada, que por sua vez, contém todas as informações necessárias para a simulação.

2.2.1 Estrutura dos dados de entrada do MCNP5

O MCNP permite ao usuário realizar simulações bastante complexas, onde é definida a geometria de forma tridimensional, o material de cada região da geometria, características da fonte de radiação, tempo de simulação e tipos de partículas a serem transportadas. Neste sentido, a construção de um arquivo de entrada para o MCNP pode ser demorada e bastante trabalhosa dependendo da complexidade do problema, mesmo para usuários experientes.

O arquivo de entrada do MCNP é um arquivo em formato ASCII organizado em blocos de programação. Sua programação é específica e realizada através de mnemônicos que são chamados de cartões no manual do MCNP. Na construção da geometria, primeiro definem-se as superfícies e posteriormente, criam-se volumes denominados células, através de operações booleanas que possibilitam a combinação de várias superfícies. A Tabela 1 mostra alguns dos mnemônicos possíveis no código MCNP para a construção de superfícies.

Tabela 1 - Lista com alguns mnemônicos para a construção de superfícies no MCNP.

Mnemônico	Tipo de superfície	Descrição	Equação
P	Plano	Geral	$Ax + By + Cz - D = 0$
PX	Plano	Normal ao eixo X	$x - D = 0$
S	Esfera	Geral	$(y - \bar{y})^2 + (z - \bar{z})^2 - R^2 = 0$
SO	Esfera	Centrada na origem	$x^2 + y^2 + z^2 - R^2 = 0$

C/X	Cilindro	Paralelo ao eixo X	$(x - \bar{x})^2 + (y - \bar{y})^2 + (z - \bar{z})^2 - R^2 = 0$
KX	Cone	No eixo X	$\sqrt{(y^2 + z^2)} - t(x - \bar{x}) = 0$
KY	Cone	No eixo Y	$\sqrt{(x^2 + z^2)} - t(y - \bar{y}) = 0$

Cada superfície é associada a um número de identificação desta superfície, e a partir destes números, criam-se as células que também possuem um número de identificação. As células são as estruturas básicas da geometria do MCNP, elas podem ser rotacionadas, transladadas ou até mesmo repetidas várias vezes. Também podem ser associadas a uma fonte de radiação, a um material e aos contadores estatísticos chamados *tallies* que armazenam as informações que o usuário requisitou da simulação.

As células definem um volume geométrico chamado “objeto simulador” e cada unidade que o compõe pode ter diferentes composições e densidade.

Outro conceito utilizado nesta metodologia é o de *Universo*, que pode corresponder a uma simples célula ou a um conjunto de células, tais como, uma rede ou uma malha constituída de várias células que se repetem.

Uma vez que o *Universo* é definido, ele pode ser utilizado para preencher outras células criando-se, desta forma, outros níveis de definição geométrica. A Figura 3 mostra células unitárias quaisquer que podem representar diferentes estruturas dentro de um objeto simulador. Estas células podem ser repetidas seguindo um padrão, de acordo com informações obtidas por imagens, como tomografias computadorizadas ou ressonâncias magnéticas, formando-se, desta forma, a geometria das estruturas desejadas, conforme pode ser visto na Figura 4.

Além de permitir a criação de estruturas irregulares, o recurso de Estruturas Repetidas possibilita o cálculo de dose em um ou mais células que representam os elementos de volume que compõe a estrutura.

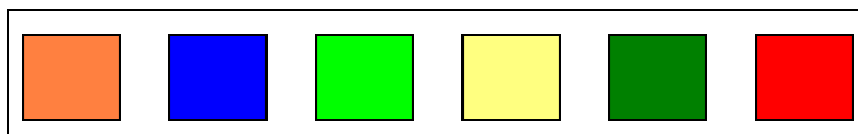


Figura 3 - Células unitárias simples. O conjunto de células de mesmo tipo pode representar um órgão ou tecido.

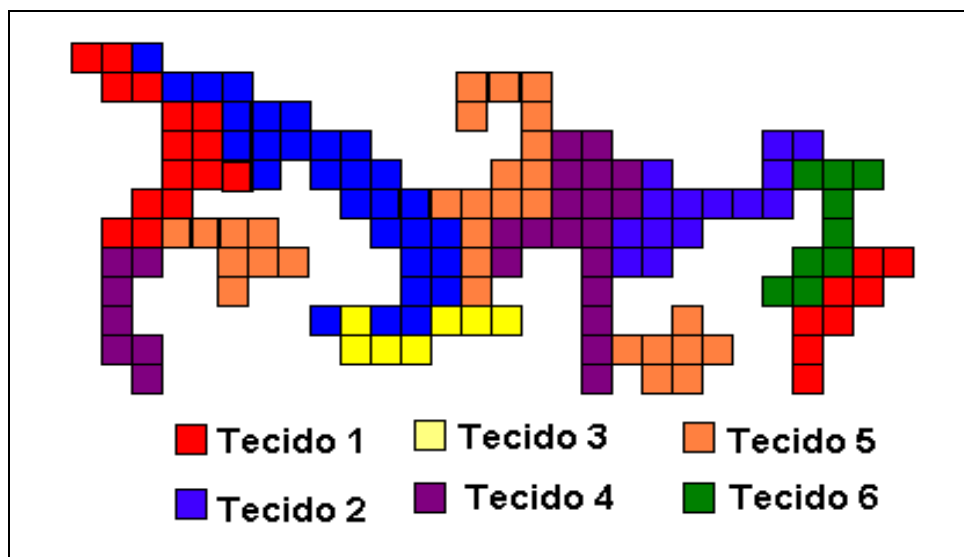


Figura 4 - Geometria composta de diferentes células que compõe diferentes regiões representando diferentes tecidos do corpo humano.

O arquivo de entrada do MCNP deve ter obrigatoriamente três blocos de informações, separados por uma linha em branco. No primeiro bloco, devem-se especificar as células da simulação; no segundo especificam-se as superfícies que compõe as células definidas no primeiro bloco e, por fim, no terceiro bloco definem-se as características da fonte de radiação, a composição dos materiais, o tempo de simulação ou o número de histórias da simulação e outras opções periféricas.

Um exemplo de arquivo de entrada do programa MCNP é apresentado na Figura 5. Neste exemplo o modelo geométrico corresponde a duas superfícies esféricas que estão declaradas no segundo bloco, visto que existem três células neste problema. O último bloco contém a descrição de uma fonte simples, de um *tally* e de quantas histórias serão simuladas, neste caso 10.000 histórias.


```

Título
c Células
10 0 -10   imp:n=1 (esfera de raio 1 cm correspondente a superfície 10)
20 0 -20 10 imp:n=1 (região entre as esferas de raios 1 e 2 cm correspondente as
                    superfícies 10 e 20)
30 0 20   imp:n=0

c Superfícies
10 so 1          (esfera de raio 1 cm) centrada na origem
20 so 2          (esfera de raio 2 cm) centrada na origem

F1:n 10 20
SDEF ERG 1
Definição dos materiais
nps 10000

```

Figura 5 - Exemplo de um arquivo de entrada do MCNP

2.2.2 Modelagem da geometria

Modelagens geométricas podem ser feitas definindo-se células individualmente, porém, em alguns casos, quando a geometria do problema é constituída por estruturas que se repetem, ou seja, ela é similar em várias partes, é possível reproduzir células, facilitando o processo da construção geométrica do problema.

Existem duas formas de repetir geometrias, sendo a primeira delas utilizando os mnemônicos *like* e *but* juntamente com os recursos de translação e rotação de objetos. Este processo de cópia é simples e necessita somente indicar o número da célula que se deseja copiar. Como a nova célula é idêntica à célula original, até mesmo sua posição é igual, por isso é necessário os comandos de translação e em alguns casos de rotação. O mnemônico associado à translação e rotação é denominado *trcl*. A sintaxe para o uso do mnemônico *trcl* é definido no final da linha na qual foi realizada a cópia de uma célula e, como superfícies e células, eles também podem ser associados a um valor numérico de identificação.

Na Figura 6 é mostrado o uso do *trcl*; existem duas formas de usá-lo: o usuário pode especificar qual será a rotação ou a translação diretamente na linha onde a célula copiada está sendo definida, ou associá-lo a um número e especificar as modificações no terceiro e último bloco do arquivo de entrada.

```

c Bloco 1
1 definir aqui a célula 1
2 like 1 but trcl = (2 0 0)      (repetição da célula 1)
3 like 1 but trcl=1            (repetição da célula 1)

...

c Bloco 3
tr1* 7 7 0 40 130 90 50 40 90 90 90 0

```

Figura 6 - Exemplo de uso dos mnemônicos *like*, *but* e *trcl*.

O exemplo acima mostra que as células 2 e 3 são cópias da célula 1. O *trcl* usado na célula 2 modifica apenas a posição espacial ao passo que na célula 3 a posição espacial é alterada e o objeto é rotacionado conforme os ângulos descritos no bloco 3. Observa-se que existem números especificando as alterações da célula 3, os três primeiros números estão relacionados à translação em x, y e z, no caso, x=7, y=7 e z=0. Os outros nove números especificam a rotação dada pelos ângulos entre os eixos iniciais antes da rotação (x,y,z) e os eixos após a rotação (x',y',z') na seguinte sequência: xx', yx', z,x', x,y', y,y',z,y', x,z', y,z' e z,z'. O asterisco informa ao MCNP que os valores de rotação dos ângulos são em graus, e a omissão do asterisco significa que são os cossenos dos ângulos.

O outro método de repetição geométrico utilizado pelo MCNP é um pouco mais complexo que o anterior e usa o conceito de *Universo*. Em resumo, assim como uma célula é formada por um conjunto de superfícies, um *Universo* pode ser tanto uma única célula quanto um conjunto de células. Para sua utilização é necessário os mnemônicos: *fill* e *u*. O número que é atribuído ao mnemônico *fill* indica o número do universo que preenche a célula que está caracterizada pelo mnemônico *fill*. Por exemplo, uma célula cuja definição exista o mnemônico "*fill=20*" será preenchido pelo universo de número 20 que pode, por sua vez, ser uma célula ou um conjunto de células. A especificação para *fill* pode ser tanto um valor único do *Universo*, como no exemplo acima, ou um conjunto de números, "*array*". O mnemônico *u* também possui um número de identificação que será usado para associar a célula que será preenchida com a célula que irá preenchê-la, por isso, o mnemônico *fill* deve ter, obrigatoriamente, um correspondente numérico *u*.

Para exemplificar o uso destes recursos geométricos, na Figura 7 é apresentado o exemplo de uma geometria com estruturas idênticas, no caso, um cubo composto de 27 cubos menores que se repetem. A construção desta geometria de forma convencional, ou seja, sem repetição, consistiria em definir todas as superfícies que definem os lados dos 27 cubos e em seguida construir a geometria de cada cubo individualmente. Entretanto, é claramente mais rápido construir um único cubo e repeti-lo 26 vezes para formar o arranjo final de 27 cubos. A construção deste arranjo cúbico usando este conceito pode ser realizada com qualquer método citado anteriormente. Usando o conceito de *Universo* o processo de construção será mais rápido, visto que o usuário não precisará se preocupar com a translação dos cubos copiados.

Usar estruturas repetidas para a construção de geometrias no MCNP torna o processo mais rápido, sobretudo quando se usa o conceito de *Universo*. Contudo, o uso do conceito de *Universo* tem uma desvantagem em relação ao tempo de simulação, pois ele a torna um pouco mais lenta quando comparado ao outro método de estruturas repetidas, *like e but*, ou até mesmo da construção convencional da geometria.

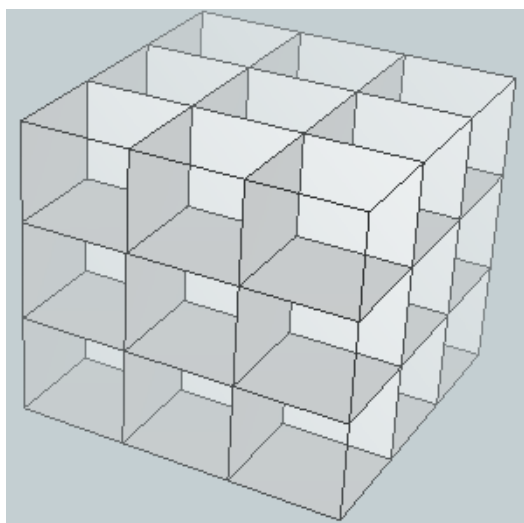


Figura 7 - Reprodução tridimensional de uma geometria composta de 27 cubos

Para construir um cubo similar ao da Figura 7, usando o conceito de *Universo*, o usuário precisa definir o cubo maior que conterá os cubos menores formando um arranjo de 3x3x3, assim, por exemplo, se os cubos menores tiverem 0.5 cm de lado, o cubo maior terá 1.5 cm de lado.

Na Figura 8 é exemplificada a descrição geométrica de um cubo no MCNP a partir de 6 planos. O primeiro número representa a identificação da célula, no caso

igual a 1, o segundo número representa a identificação do material da célula, o próximo número, “-1”, representa a densidade do material e os próximos seis números, 11, 12, 13, 14, 15 e 16, representam os seis planos que constituem as faces deste cubo. Por fim, utiliza-se o mnemônico *fill* para que esta célula seja preenchida por um universo, neste exemplo ele é preenchido pelo universo “1”. Em seguida, é necessário criar este universo, que neste caso é identificado pela célula de número 2, tal como mostrado na Figura 9. Em outras palavras, o universo de número 1 que preencherá a célula 1 é composta de outra célula que, neste exemplo, corresponde a célula 2.

```
c Bloco 1
c Cubo
1 1 -1 -11 12 -13 14 -15 16 fill=1
```

Figura 8 - Dados de entrada do MCNP correspondente à geometria do cubo que conterá 27 cubos menores formando um arranjo de 3x3x3.

```
c Bloco 1
c Cubo
1 1 -1 -11 12 -13 14 -15 16 fill=1
2 1 -1 -1 2 -3 4 -5 6 u=1 lat=1
```

Figura 9 - Dados de entrada com estruturas repetidas da geometria do cubo menor (célula 2).

A sequência numérica da célula 2 é similar a célula 1, ou seja, o primeiro número é o índice da célula, seguido pela identificação do material, “1”, e o valor de sua densidade, “-1”, os seis números sequenciais correspondem aos seis planos que formam o cubo menor que irá preencher a célula. Por fim, o mnemônico *u* representa o universo 1, e o mnemônico *lat=1* indica que a célula 2, na realidade, trata-se de uma rede quadrada cujos elementos individuais possuem as dimensões do cubo menor e preencherá a célula 1 dada pelo mnemônico *fill=1* (veja Figura 9). O mnemônico *lat* é a abreviação da palavra inglesa *lattice*, e pode assumir o valor 1 ou 2 no MCNP. Quando assume o valor igual a 1 significa que a grade, na qual a célula será repetida, terá o formato de um hexaedro e quando assume o valor 2 define um octaedro.

Nas duas figuras anteriores foram mostradas instruções para a construção de uma geometria repetida no MCNP usando o conceito de *Universo*. O código MCNP repetirá a célula 2 quantas vezes forem necessárias para que toda ou grande parte da

célula 1 fique preenchida. Contudo este exemplo ainda não está completo, pois ainda é necessário definir os 12 planos que formam as duas células (cubos) desta geometria, o exemplo está completo na Figura 10A, a Figura 10B, mostra essa mesma geometria, porém sem a definição das superfícies no bloco 2 e construída usando os mnemônicos *like* e *but*.

<pre> c Bloco 1 1 1 -1 -1 12 -13 14 -15 16 fill=1 2 1 -1 -1 2 -3 4 -5 6 u=1 lat =1 c Bloco 2 c Planos do cubo interno 1 px .5 2 px -.5 3 py .5 4 py -.5 5 pz .5 6 pz -.5 c Planos cubo principal 11 px 1.5 12 px -1.5 13 py 1.5 14 py -1.5 15 pz 1.5 16 pz -1.5 </pre> <p style="text-align: center;">A</p>	<pre> c Células c fatia central vertical 2 1 -1 -1 2 -3 4 -5 6 3 like 2 but trcl=(1 0 0) 4 like 2 but trcl=(-1 0 0) 5 like 2 but trcl=(1 1 0) 6 like 2 but trcl=(-1 1 0) 7 like 2 but trcl=(1 -1 0) 8 like 2 but trcl=(-1 -1 0) 9 like 2 but trcl=(0 1 0) 10 like 2 but trcl=(0 -1 0) c fatia lateral 11 like 2 but trcl=(0 0 1) 12 like 2 but trcl=(1 0 1) 13 like 2 but trcl=(-1 0 1) 14 like 2 but trcl=(1 1 1) 15 like 2 but trcl=(-1 1 1) 16 like 2 but trcl=(1 -1 1) 17 like 2 but trcl=(-1 -1 1) 18 like 2 but trcl=(0 1 1) 19 like 2 but trcl=(0 -1 1) c fatia lateral traseira 20 like 2 but trcl=(0 0 -1) 21 like 2 but trcl=(1 0 -1) 22 like 2 but trcl=(-1 0 -1) 23 like 2 but trcl=(1 1 -1) 24 like 2 but trcl=(-1 1 -1) 25 like 2 but trcl=(1 -1 -1) 26 like 2 but trcl=(-1 -1 -1) 27 like 2 but trcl=(0 1 -1) 28 like 2 but trcl=(0 -1 -1) </pre> <p style="text-align: center;">B</p>
--	---

Figura 10 - Vinte e sete cubos construídos usando dois tipos de estruturas repetidas:

(A) geometria construída usando-se os mnemônicos *u*, *fill* e *lat*; B) geometria construída usando se os mnemônicos *like* e *but*.

A Figura 10A mostra tanto as células do bloco 1 quanto as 12 superfícies definidas no bloco 2. O primeiro número da sintaxe para a construção de uma superfície é seu índice, este índice é utilizado na definição da célula no bloco 1, as duas letras que

seguem correspondem ao tipo de plano e, por fim, o valor “.5” e “.1.5” representam a distância da origem do plano cartesiano em centímetro, esta distância alinha os planos até o ponto na qual o plano toca o eixo.

Usando *like* e *but*, mostrados na Figura 10B, constrói-se a mesma geometria que representada na Figura 10A. Nota-se que a quantidade de instruções é maior, visto que é necessário criar um cubo e copia-lo 26 vezes. Além disso, na figura ainda não está apresentada as superfícies, portanto o arquivo final seria ligeiramente maior.

No exemplo anterior o material dos cubos é o mesmo, contudo em alguns casos esse material ou sua densidade pode variar. Na Figura 11 é mostrada uma geometria simples onde os 27 cubos estão separados em 3 linhas e 3 colunas e 3 planos, entretanto, os cubos do primeiro plano, representados com cores diferentes, são compostos por materiais diferentes dos demais. O recurso de estruturas repetidas permite especificar o material e a densidade destes cubos. A forma mais simples, num primeiro momento, seria usar o comando *like* em conjunto com o comando *but*. O usuário deve criar um cubo e repeti-lo em outras direções até formar 26 cubos, como cada cubo possui uma célula diferente é possível definir seu material e sua densidade usando os mnemônicos *mat* e *rho* como mostra a sintaxe na Figura 12.

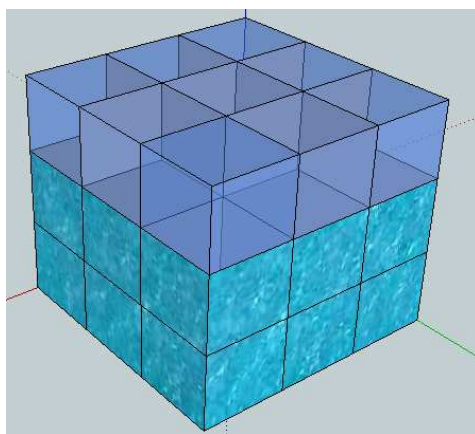


Figura 11 - Visualização de um exemplo de geometria com estruturas repetidas composto de 27 cubos. Os cubos do primeiro plano possuem materiais diferentes dos demais.

```
2 like 1 but mat=4 rho=-0.98
```

Figura 12 - Exemplo do uso dos mnemônicos *mat* e *rho* na repetição de estruturas geométricas

A sintaxe para o uso dos mnemônicos *mat* e *rho* é simples: primeiro copia-se a célula e depois se existir a necessidade é possível mudar o material e a densidade da nova célula. No exemplo da Figura 12, a célula 2 é geometricamente igual à célula 1, contudo seu material correspondente ao número 4 e sua densidade tem o valor de 0.98 g/cm³.

Modificar os valores do material e da densidade utilizando o conceito de Universo através dos mnemônicos *fill*, *u* e *lat*, também é possível. Esta solução é mais prática e rápida, sobretudo quando a geometria a ser simulada for muito grande como é mostrado na Figura 13. Neste caso, há uma série de diferenças nas instruções para a modelagem desta geometria em relação às instruções apresentadas na Figura 10A. A principal diferença localiza-se na linha referente à célula 2, onde se nota que a sintaxe do comando *fill* é claramente diferente do comando anterior. Agora, ao invés do *fill* ter o valor numérico referente a um universo, existe um intervalo ou campo (“array”), que representa a sequência de células (cubos), adotando como referência a dimensão dos três eixos do plano cartesiano, x, y e z, respectivamente. Além disso, em cada eixo x, y e z, as coordenadas percorrem os índices de -1 a 1, separadas por dois pontos. Estas coordenadas representam as localizações dos cubos que serão preenchidos no eixo x, y e z, respectivamente totalizando os 27 cubos.

Nota-se que os valores estão separados por fatias e que a primeira fatia possui valores diferentes, 2, das demais fatias, que possuem valor 1. Cada valor nestas fatias representa um universo, estes, por sua vez, estão definidos mais abaixo. As células 3 e 4 possuem valores de universo 1 e 2, respectivamente.

```

c Celulas
1 0 -11 12 -13 14 -15 16 fill=999
2 0 -301 302 -303 304 -305 306 u=999 lat=1
      fill=-1:1 -1:1 -1:1

c fatia 1
  2 2 2
  2 2 2
  2 2 2
c fatia 2
  1 1 1
  1 1 1
  1 1 1
c fatia 3
  1 1 1
  1 1 1
  1 1 1

c
3 1 -0.296 -1 2 -3 4 -5 6      u=1
4 like 3 but mat=4 rho=-0.98  u=2
50 0 #1

c Planos do cubo interno
1 px .5
2 px -.5
3 py .5
4 py -.5
5 pz .5
6 pz -.5
c lattices side
301 px .5
302 px -.5
303 py .5
304 py -.5
305 pz .5
306 pz -.5
c Planos cubo mundo
11 px 1.5
12 px -1.5
13 py 1.5
14 py -1.5

```

Figura 13 – Geometria de 27 cubos utilizando estruturas repetidas e empregando o conceito de universo

Desta forma é possível especificar qual o tipo de material e densidade de cada cubo desta geometria, pois cada cubo que será preenchido na geometria foi vinculado a uma célula que, por sua vez, possui um valor de material e de densidade.

Foram adicionados espaços e parágrafos na linha referente à célula 2 para facilitar a visualização, contudo ela poderia ser escrita na mesma linha, desde que não ultrapasse 80 caracteres por linha. Porém, se 80 caracteres forem insuficientes para a dimensão da geometria, o usuário pode utilizar a próxima linha desde que deixe 5 espaços em branco como é mostrado na Figura 14. Além disso, algumas geometrias

possuem universos iguais e sequenciais. O MCNP suporta comandos de atalho que concatenam essa informação como mostra a Figura 15, neste exemplo a letra “r” informe ao MCNP que repita o universo anterior, o número de repetições a serem realizadas é igual ao número que acompanha a letra.

```
2 0 -301 302 -303 304 -305 306 u=999 lat=1
fill=-1:1 -1:1 -1:1
2 2 2 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 2 2 2 1 1 1 1 1
```

Figura 14 - Utilizando estruturas repetidas

```
2 0 -301 302 -303 304 -305 306 u=999 lat=1
fill=-1:1 -1:1 -1:1
2 2r 1 5r 2 2r 1 5r 2 2r 1 5r
```

Figura 15 - Definindo estruturas repetidas usando atalhos

As instruções mostradas tanto na Figura 14 quanto na Figura 15 resultarão no mesmo resultado, apesar do primeiro caso ser um pouco mais legível, especialmente se ele for ordenado por fatias. Já o segundo caso é pouco legível, mas ocupa menos espaço no arquivo de dados de entrada, sendo útil para geometrias muito grandes como no caso de objetos simuladores que representam a anatomia humana.

2.2.3 Modelagem da fonte de radiação

Além da construção da geometria também é preciso definir as características da fonte de radiação para a simulação. No código MCNP a caracterização da fonte deve ser definida no terceiro bloco, junto com a definição dos *tallies*, ou seja, os cálculos de interesse da simulação, bem como, dados dos materiais e informações das opções do arquivo de saída da simulação.

O código permite quatro tipos diferentes de fontes: fontes genéricas, definidas pelo mnemônico SDEF, fontes de superfícies, SSR, fontes críticas, implementadas pelo mnemônico KCODE e, se nenhum destes três tipos de fonte forem adequados ao problema, o usuário pode construir sua própria fonte usando a subrotina chamada SOURCEX.

O MCNP não possui interface gráfica para verificar a posição ou até mesmo a emissão de partículas da fonte, por isso é bastante recomendável que durante a

construção da fonte o usuário verifique no arquivo de saída em que posição as partículas estão sendo criadas. Para isso utiliza-se a opção “110” no comando “*print*” do terceiro bloco. Este comando faz com que o arquivo de saída do código contenha as primeiras 50 partículas que foram originadas na fonte. Estas primeiras partículas são agrupadas em forma de tabela e contém sua posição inicial, sua energia, a célula de onde ela partiu, dentre outras características. Outra opção seria utilizar o recurso do cartão DBCN.

O comando DBCN é opcional e pode se utilizado no terceiro bloco, este comando acrescenta informações de *debug* no arquivo de saída da simulação. Ele possui uma série de parâmetros de entrada, sendo possível modificar configurações dos números aleatórios, adicionar mais informações nos *tallies*, modificar a geração de partículas secundárias, exibir informações detalhada das histórias das partículas, dentre outras opções.

Durante a modelagem da fonte de radiação o usuário precisa definir o tipo de partícula que será emitida, como: elétrons, fótons ou nêutrons, para isso existem os mnemônicos MODE e PAR.

As fontes de radiação podem ter distribuições energéticas variadas ou associadas a um espectro ou então monoenergéticas. Espectros de fontes são associados a uma distribuição de probabilidade para que a simulação use os diferentes valores de energia disponíveis.

A definição de distribuição de probabilidade no MCNP é feita pelos mnemônicos *SI* e *SP*. Eles correspondem, respectivamente, à informação de intensidade e à probabilidade, por isso podem ser relacionados tanto à energia da fonte quanto a sua distribuição espacial, por exemplo.

Somente fontes do tipo genéricas, SDEF, foram usadas neste trabalho, por isso os exemplos a seguir serão limitados a esse tipo de fonte. O mnemônico SDEF permite várias variáveis como parâmetros de entrada e é um dos comandos mais complexos do código MCNP. Somente um mnemônico SDEF é permitido numa simulação.

Uma forma simples de se definir uma fonte monoenergética é definir uma célula como fonte e com um valor fixo de energia, por exemplo, “SDEF ERG=1 cel=2”. Neste exemplo, ERG corresponde ao valor da energia, em MeV, e o mnemônico “cel” corresponde à célula onde está contida a fonte de radiação. Outro exemplo mais simples

de utilização seria apenas digitar “SDEF” no terceiro bloco, com isso será criada uma fonte pontual localizada no centro da geometria com energia de 14 MeV.

Fontes podem ser homogêneas, com chances de emissão iguais em toda sua extensão, ou heterogênea, com probabilidade de emissão variada. O processo de descrição de fontes heterogêneas também é suportado pelo código MCNP. Assim como sua energia, a localização da fonte também pode ser associada a uma distribuição de probabilidade.

O procedimento de descrição da localização da fonte é bastante extenso e trabalhoso em objetos simuladores muito grandes e que utilizam estruturas repetidas, pois é preciso definir cada ponto ou região onde se localiza a fonte e sua probabilidade.

Na Figura 16 é mostrado o primeiro e segundo bloco de uma geometria utilizando estruturas repetidas que será usada como base nos próximos exemplos. A geometria é semelhante às geometrias anteriores, ou seja, um cubo composto de cubos menores em seu interior num arranjo de $3 \times 3 \times 3$.

```

Bloco 1
Titulo: Teste de fontes
c Celulas
1 0 -11 12 -13 14 -15 16 fill=999
2 0 -301 302 -303 304 -305 306 u=999 lat=1
  fill=0:2 0:2 0:2
  2 2 2 1 1 1 1 1 1 2 2 2 1 1 1
  1 1 1 2 2 2 1 1 1 1 1 1
3 1 -0.296 -1 2 -3 4 -5 6 u=1
4 like 3 but mat=4 u=2
50 0 #1

C Bloco 2
c Planos do cubo interno
1 px 1
2 px 0
3 py 1
4 py 0
5 pz 1
6 pz 0
c lattices side
301 px 1
302 px 0
303 py 1
304 py 0
305 pz 1
306 pz 0
c Planos cubo mundo
11 px 3
12 px 0
13 py 3
14 py 0
15 pz 3
16 pz 0

```

Figura 16 - Bloco 1 e Bloco 2 da simulação base para os exemplos sobre fontes.

Antes do mnemônico SDEF, é necessário outro mnemônico chamado “mode”. Este comando especifica qual o tipo de partícula que o código MCNP irá acompanhar durante sua simulação. O mnemônico “mode” acompanha outros comandos que são: “p”, “n” e “e”, que fazem referência, respectivamente, a fótons, nêutrons e elétrons. Por exemplo, se o usuário definir “mode p”, significa que o MCNP só irá simular fótons, se for “mode e p”, o código irá simular tanto elétrons quanto fótons, e assim por diante.

No SDEF também é preciso definir qual partícula a fonte irá emitir, para isso usa-se o mnemônico PAR, este comando suporta apenas três opções: 1, 2 ou 3, sendo 1 para nêutrons, 2 para fótons e 3 para elétrons.

Se o comando PAR for omitido pelo usuário o código MCNP carregará valores padrões levando-se como referência o comando MODE. Sendo assim, a opção 1

significa que a fonte irá emitir nêutrons se o comando “*mode*” for “N” ou “P” ou “N P E”, se a opção for “2” e o “*mode*” for “P”, a fonte emitirá fótons, por fim, se o valor for “3” e o “*mode*” for “E” a fonte irá emitir elétrons.

Na Figura 17A é mostrado apenas uma parte do terceiro bloco de um arquivo de entrada do MCNP, onde está descrito uma fonte simples. Na Figura 17B é mostrada a respectiva ilustração gráfica da localização desta fonte.

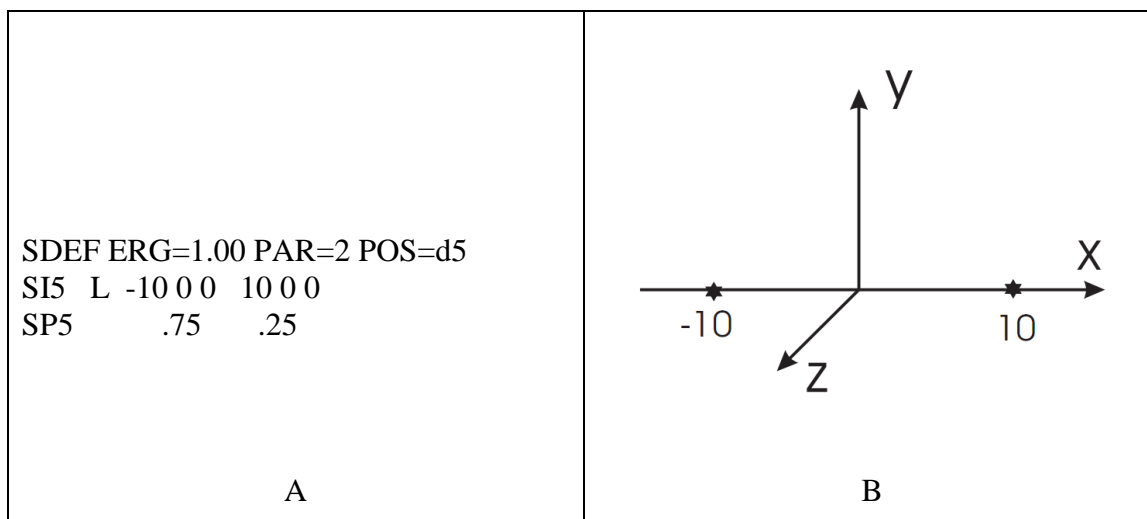


Figura 17 - Fonte isotrópica emissora de fótons

Nota-se que a energia dado pelo mnemônico, ERG, possui um símbolo de igual antes do valor de 1 MeV, o uso do símbolo é facultativo. Esta fonte emite fótons, pois o comando PAR é igual a 2. Neste exemplo, a fonte de radiação é caracterizada por duas fontes pontuais cujas localizações estão definidas como descrito a seguir. Sua posição espacial, POS, foi atribuída a uma distribuição “d5” como mostra as linhas 2 e 3. O comando SI faz referência a coordenadas de dois pontos onde as fontes estão localizadas. Os valores -10, 0 e 0 representam, respectivamente, as posições nas coordenadas x, y e z da primeira fonte pontual, enquanto os pontos 10, 0 e 0 representam a localização da segunda fonte.

A letra “L” indica que os valores da posição são discretos, a ausência da letra resultaria no valor padrão que são informações na forma de histograma.

O comando SP, neste caso, indica a probabilidade da emissão das fontes de radiação e os valores estão normalizados, isto quer dizer que na fonte localizada na primeira coordenada a probabilidade de emissão é de 75% enquanto na segunda fonte é de 25%.

A fonte de radiação descrita na Figura 18A diferentemente da fonte anterior não varia sua posição, neste caso a distribuição de probabilidade está associada à energia. Ela está localizada no centro do problema e a partícula de emissão é o fóton. A representação gráfica está mostrada na Figura 18B.

Sua energia está associada a uma distribuição “d1” como mostra a figura. A letra “L” depois do mnemônico SII significa, novamente, que os valores da energia da fonte são discretos. Esta fonte possui quatro valores diferentes para energia e a terceira linha mostra a probabilidade de cada energia, os valores estão normalizados, mas caso não estivessem o código MCNP faria a normalização.

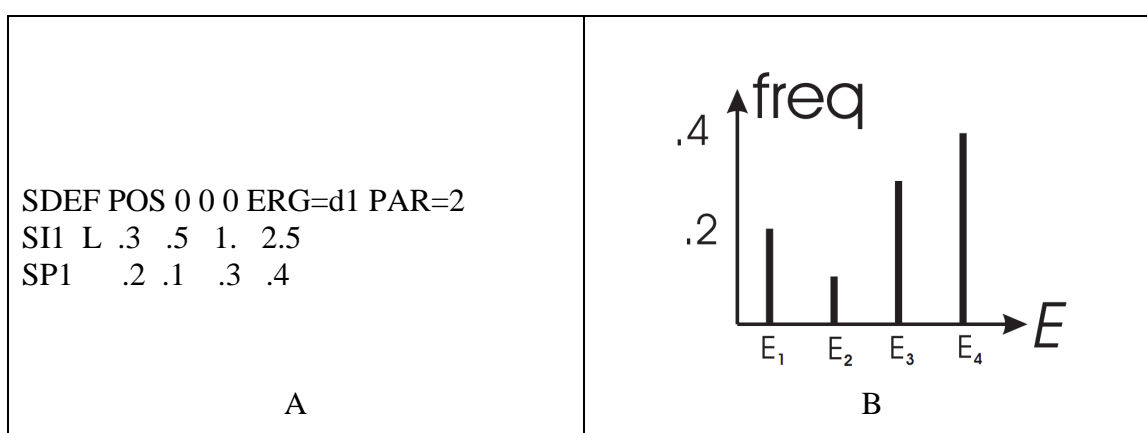


Figura 18 - Fonte pontual com valores variáveis de energia

No exemplo da Figura 18, a energia de 2.5 MeV, por exemplo, tem maior chance de ser emitida durante a simulação, com 40% de chance, já a energia de 0.5 MeV é a energia com menor chance de ser escolhida, pois tem apenas 10% de chance.

Na Figura 19A é mostrada uma fonte bastante similar a anterior, onde as partículas emitidas pela fonte são fótons que se originam no centro da geometria. Sua energia foi associada a uma distribuição “d1”, porém, diferentemente da fonte anterior, a distribuição faz associação à letra “H” depois do comando SII, esta opção define as energias em forma de histograma como mostra a Figura 19B, portanto os valores apresentados representam o limite superior de cada "bin" do histograma. Em resumo, esta distribuição está associada a um *range* de energia e não mais a valores pontuais como no último exemplo.

O comando “SP1” também difere da fonte anterior, a letra “D” é o comando padrão associado às probabilidades, portanto as fontes anteriores também continham este comando, porém não explicitadas. Outra diferença são os valores da probabilidade,

neste exemplo eles não foram normalizados pelo usuário, contudo o código MCNP se encarrega de normalizá-los.

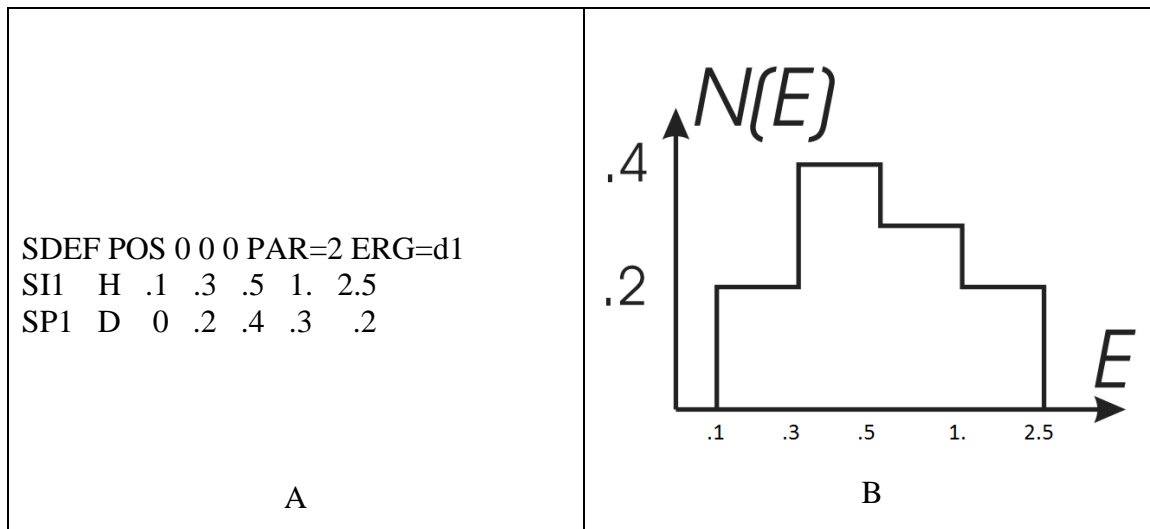


Figura 19 - Fonte com energia em histogramas

A fonte mostrada na Figura 20 apresenta energia constante de 1.25 MeV e está localizada no centro da geometria, pois a omissão do mnemônico “POS” faz com que o código MCNP carregue valores padrões para sua localização.

Neste exemplo a fonte de radiação está dividida entre duas células, seus números de identificação são 8 e 9, como mostra a segunda linha.

A última linha apresenta a probabilidade de emissão em cada localização da fonte, pois neste exemplo tanto a célula 8, quanto a célula 9 são as fontes do problema, portanto a fonte localizada na célula 8 tem 80% de chance de ser amostrada na simulação, contra apenas 20% da fonte localizada na célula 9.

<pre>SDEF ERG=1.25 CEL d1 SI1 L 8 9 SP1 0.8 0.2</pre>

Figura 20 - Fonte com energia constante localizada em duas células distintas

Geometrias com estruturas repetidas não possuem identificação direta à célula, contudo é possível associar uma fonte às células de uma geometria repetida. O código MCNP suporta este procedimento usando uma sintaxe própria para referenciar essas células como mostra a Figura 21A.

A Figura 21B mostra a representação gráfica da geometria descrita na Figura 16A, que consiste de um arranjo cúbico, com destaque para dois cubos na cor vermelha representando a fonte de radiação.

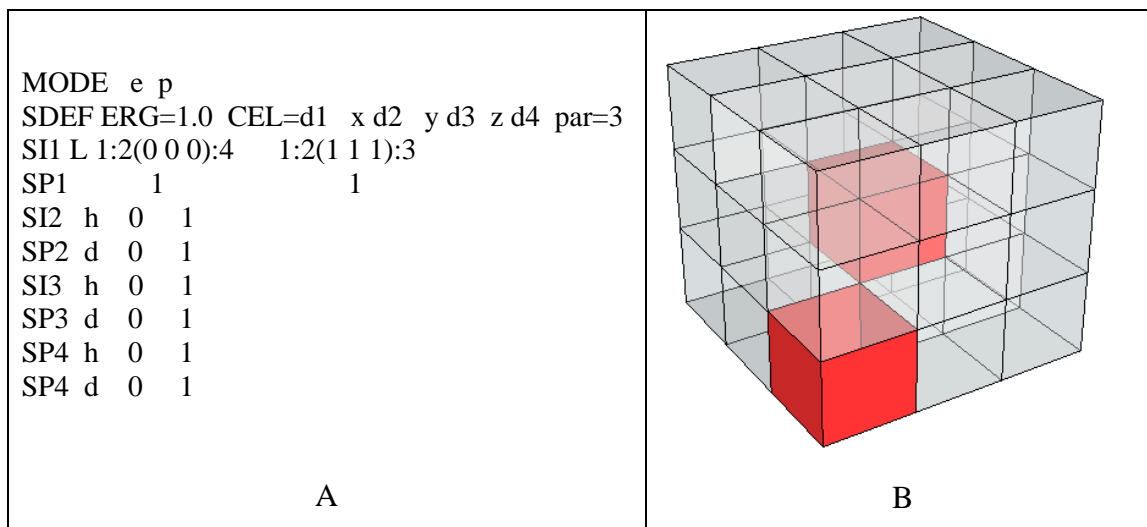


Figura 21 - Fonte de radiação composta por duas células cúbicas dentro de um arranjo cúbico usando estruturas repetidas

O código MCNP suporta múltiplas distribuições de probabilidade na caracterização de uma fonte, como no exemplo da Figura 21. Neste exemplo tanto a célula quanto a posição espacial estão vinculadas a uma distribuição. Outro ponto importante é a descrição da célula que será utilizada pela fonte. No exemplo acima são duas células com chances iguais de emissão especificadas pelo comando “SP1 1”.

O código MCNP permite definir fontes usando células dentro do contexto de estruturas repetidas desde que utilize uma sintaxe específica para definir a célula, pois neste caso não é possível usar um simples número para referenciar a célula. A sintaxe, que pode ser observada na terceira linha, segue o seguinte esquema: o primeiro número, 1 (um) neste caso, faz referência à célula que foi preenchida por um universo, o segundo número mostra qual célula contém o universo que preencheu a primeira, no caso a célula de número 2 (dois). Dentro dos parênteses existem três números relativos aos eixos x, y e z, respectivamente, que fazem referencia às coordenadas espaciais dentro do universo que contém a célula. Os valores utilizados na descrição da primeira célula são (0 0 0) e (1 1 1) para a segunda, desta forma, os cubos escolhidos foram, respectivamente, o primeiro cubo e o cubo central da geometria global que possui 27

cubos e cuja origem das coordenadas se encontra no canto inferior esquerdo onde se encontra o primeiro cubo.

Por fim, o número “4” na primeira célula e o número “3” na segunda célula fazem referências ao tipo de célula que existe na geometria repetida. A sintaxe para descrever uma célula na descrição da fonte pode ser resumida na Figura 22.

cp:cu(x y z):tc

Figura 22 - Sintaxe de referencia a células

Onde:

- cp é a célula principal que foi preenchida por um universo “n”
- cu é a célula que correspondente ao universo “n” e que preenche a célula principal
- x é a coordenada no eixo x da célula dentro do universo
- y é a coordenada no eixo y da célula dentro do universo
- z é a coordenada no eixo z da célula dentro do universo
- tc é o tipo de célula usada no universo

Na Figura 23 é mostrado um exemplo de referência às células dentro de um contexto de estruturas repetidas de forma mais detalhada, pois este exemplo possui o primeiro bloco com informações referentes à geometria.

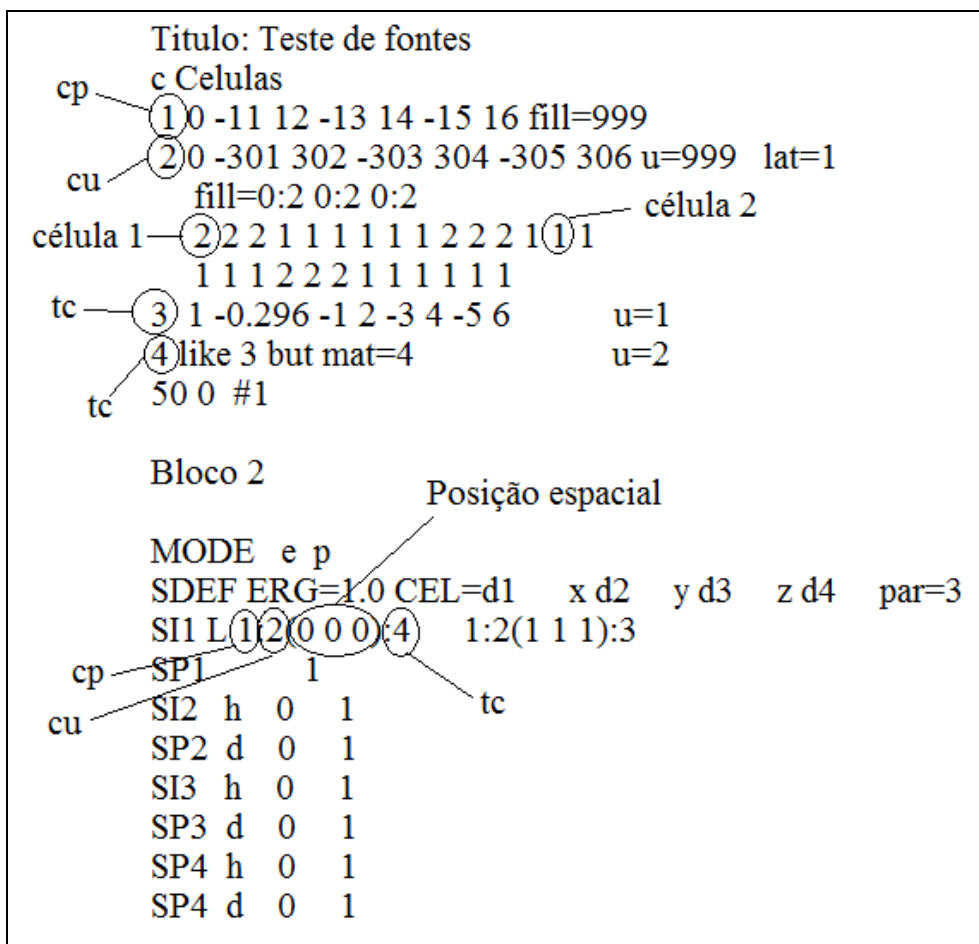


Figura 23 - Exemplo de referência às células dentro de um contexto de estruturas repetidas

2.2.4 O cálculo de dose: *Tallies*

O objetivo de qualquer simulação do transporte de radiação com a matéria é obter alguma informação do sistema que está sendo simulado, tais como: o fluxo de determinada partícula que atravessa uma superfície, pulsos de energia num detector, energia média depositada numa célula, energia de fissão depositada, dentre outras opções.

Para cada informação (*tally*) diferente que o usuário deseja é necessário um mnemônico diferente que deve ser colocado no terceiro bloco do arquivo de entrada. Na Tabela 2 é mostrada a lista de todos os mnemônicos disponíveis no MCNP para a coleta de informações durante a simulação. Como cada *tally* é responsável por uma informação diferente, a unidade dos dados de saída da simulação pode variar. Além disso, a maioria dos *tallies* também possui um modificador que altera a unidade de

cálculo de acordo com a necessidade do usuário, esse modificador é mostrado na última coluna da tabela.

Tabela 2 - Lista dos mnemônicos para uso nos *tallies*

Mnemônico	Descrição	Unidade	Unidade com modificador *
F1:N ou F1:p ou F1:E	Corrente integrada numa superfície	Partículas	MeV
F2:N ou F2:P ou F2:E	Fluxo médio numa superfície	Partícula/cm ²	MeV/cm ²
F4:N ou F4:P ou F4:E	Fluxo médio numa célula	Partícula/cm ²	MeV/cm ²
F5a:N ou F5a:P	Fluxo num detector pontual ou radial	Partícula/cm ²	MeV/cm ²
F6:N ou F6:N,P ou F6:P	Energia média depositada na célula	MeV/g	jerks/g
F7:N	Energia de fissão depositada	MeV/g	jerks/g
F8:P ou F8:P,E ou F8:E	Pulsos de energia num detector	Pulsos	MeV
+F8:E	Deposição de carga	Carga	Não existe

Se a geometria do problema foi criada de forma convencional, ou seja, sem estruturas repetidas, onde cada célula é definida individualmente é fácil adicionar um *tally* ao problema.

Na Figura 24 é mostrado o *tally* que calcula o fluxo médio de nêutrons em superfícies. São três superfícies, 1, 3 e 6, além disso, a instrução adicional inserida por meio da letra T no final da linha significa que foi requisitado, além dos fluxos médios em cada superfície mencionada anteriormente, o fluxo médio correspondente a soma dos fluxos nas superfícies 1, 3 e 6 divididos pela soma das áreas das respectivas superfícies.

F2:N 1 3 6 T

Figura 24 - **Tally** para cálculo do fluxo médio de nêutrons numa superfície

O exemplo de **tally** a seguir, mostrado na Figura 25, calcula a corrente de fótons em três superfícies, das quais uma delas é constituída pela soma das superfícies 1 e 2, a próxima é formada pela soma das superfícies 3, 4 e 5 e a última somente na superfície 6.

F1:P (1 2) (3 4 5) 6

Figura 25 - **Tally** que fornece três resultados para correntes de fótons

O próximo **tally**, mostrado na Figura 26, é similar ao anterior, pois calcula a corrente, mas neste caso, de nêutrons. São três valores de correntes, uma formada pela soma das superfícies 1, 2 e 3, a segunda pela soma das superfícies 1 e 4 e a última pela soma das superfícies anteriores, ou seja, 1, 2, 3 e 4.

F371:N (1 2 3) (1 4) T

Figura 26 - Corrente de nêutrons

Quando se utiliza estruturas repetidas também é possível definir o **tally** por célula ou por região. Neste caso, entretanto, a sintaxe, assim como no caso da definição da fonte radiação, é um pouco diferente, pois não existe referência direta à célula.

O **tally** na Figura 27, “*f8”, mostra um exemplo simples de como calcular a energia depositada por elétrons num universo contido numa célula. O primeiro número, “3”, está relacionado à célula principal onde se deseja obter o resultado. A célula de número “2” após o sinal de “menor” está relacionada à célula que corresponde ao **lattice** (grade) onde a célula “3” está inserida. Este exemplo vai retornar a somatória da energia depositada em todas as células 3 dentro da célula 2.

*f18:p,e (3 < 2)

Figura 27 - Exemplo de **Tally** para estruturas repetidas

Contudo, o usuário também pode ter a necessidade de calcular o *tally* separadamente, ou seja, em cada um das células dentro da geometria repetida (grade) e não apenas a somatória delas. Na Figura 28 é mostrado um exemplo de três *tallies* que calculam o valor da energia depositada por elétrons numa área selecionada pelo usuário e de forma separada.

```
*f18:p,e ( 3 < 2 [ 0:2  0:2  0:0 ]
*f28:p,e ( 3 < 2 [ 0:2  0:2  1:1 ]
*f38:p,e ( 3 < 2 [ 0:2  0:2  0:1 ]
```

Figura 28 - *Tally* retangular usando estruturas repetidas

Neste exemplo em particular, para o *tally* *f18, está se requisitando o valor da energia depositada nas células de número “3” presentes na célula “2”. Os números dentro dos colchetes representam as posições espaciais dentro da célula “2” (que é composta por uma grade), onde se deseja o valor de energia depositada e cuja célula seja de número “3”. Os valores significam, respectivamente, a primeira e a última posição da célula no eixo x separados por “:”. A mesma sintaxe funciona para o eixo y e para o eixo z.

No *tally* *f18 foi escolhido um conjunto de células que constitui uma matriz cujo arranjo é de 3x3x1 células localizadas no nível 0:0 em z (primeiro plano em z), ou seja, as informações serão estimadas dentro deste arranjo composto de 9 células, no primeiro plano em z, nível 0:0. Da mesma forma, no *tally* *f28 também serão estimados valores num arranjo de 3x3x1 de células, porém, no nível 1:1 em z (segundo plano em z). No terceiro *tally*, *f38, as informações serão coletadas num arranjo 3x3x2 (incluindo o primeiro e segundo planos em z) que engloba os dois primeiros *tallies*.

Em alguns casos, existem diferentes células com números diferentes dentro da mesma geometria repetida, portanto, neste caso, é preciso especificar quais células estão dentro desta região, como é mostrado na Figura 29. Este exemplo possui 23 células diferentes dentro do universo.

```
*f18:p,e (( 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 ) < 2 [ 0:2  0:2  0:1 ]
```

Figura 29 - *Tally* retangular com células variadas

No exemplo acima, o *tally* está definido para um arranjo matricial de 3x3x2 células, cujas coordenadas vão de 0 a 2 em x, 0 a 2 em y e 0 a 1 em z, com a diferença que, neste caso, as informações serão estimadas em todas as células identificadas entre 3 e 26 e não apenas na célula 3 como no exemplo anterior.

2.3 O Software de Interface SCMS

O software SCMS foi desenvolvido para servir como interface entre a obtenção de imagens médicas, que carregam todas as informações de um paciente específico, e o MCNP. Estas informações são transformadas num formato adequado que é aceito pelo código MCNP, para o cálculo de dose. Este processo é feito através da construção de uma representação geométrica da anatomia do paciente e também da especificação da fonte de radiação, a qual o paciente está sujeito durante o tratamento. Neste sentido, o SCMS explora todos os recursos do MCNP apresentados na seção anterior para criar um arquivo de dados de entrada de forma automática para ser utilizado pelo código MCNP.

Como mostrado na Figura 1, toda a informação necessária para a construção geométrica da anatomia do paciente é fornecida pelas imagens de tomografia computadorizada (TC) ou ressonância magnética (RM). Simultaneamente, o software SCMS processa as informações obtidas a partir das imagens de SPECT ou PET que fornecem dados sobre a distribuição espacial de atividades radioativas no paciente e transforma-os num formato adequado ao código MCNP para a simulação do transporte de radiação juntamente com as informações geométricas obtidas anteriormente.

A atual versão deste software se encontra na linguagem FORTRAN-77 e está restrito ao ambiente Windows com limitações de utilização quanto ao tempo de processamento e acessibilidade ao usuário final. Esta versão a ser modificada é constituída de 7 subrotinas, incluindo o programa principal. A parte central do software baseia-se no recurso de estruturas repetidas do código MCNP, cujas instruções estão baseadas nas informações fornecidas pelas imagens médicas, tanto na parte anatômica como na parte da caracterização da fonte de radiação.

2.4 Estrutura do Software SCMS

O programa computacional SCMS possui 7 subrotinas além do programa principal (MAIN) que são: PHANTOM, RESIZE, SETUP, UNIVMAP, FILLCARD, TALLIES, SOURCE1 e SOURCE2. O diagrama esquemático do software está mostrado na Figura 30.

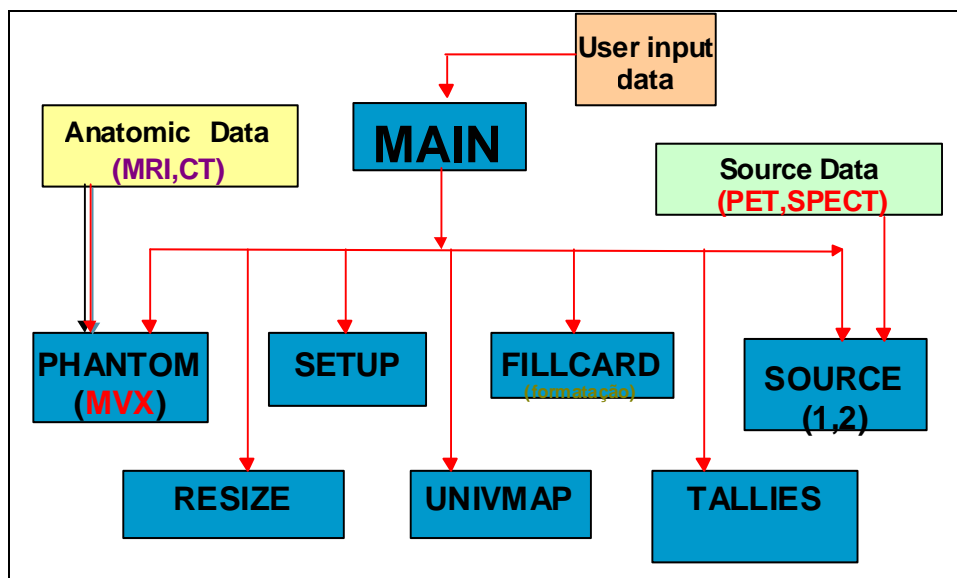


Figura 30 - Diagrama esquemático do software SCMS.

2.4.1 Programa Principal: MAIN

Este é o módulo principal onde todas as variáveis são definidas, bem como, as unidades de entrada e saída de dados e também onde são chamadas todas as subrotinas. Neste módulo também é lido o arquivo de entrada, que fornece as principais especificações do arquivo de imagens e outras informações para definição do problema, tais como: órgãos fontes, opção de cálculo de doses médias ou então de distribuição de doses, densidade de cada tecido ou órgão.

2.4.2 Subrotina PHANTOM

Basicamente, existem duas subrotinas de leitura de arquivo de imagens tomográficas: a subrotina PHANTOM1 e PHANTOM2. A escolha de uma delas depende do arquivo de imagem a ser lido. Toda a informação contida no arquivo de imagem de entrada é armazenada em uma matriz chamada MVX.

Os arquivos de imagens podem diferir em seus formatos de leitura e também suas dimensões matriciais, bem como no tamanho de cada elemento ou “voxel” dos quais são constituídos.

2.4.3 Subrotina RESIZE

Esta subrotina elimina as regiões exteriores ao organismo e que não interessam ao problema, mas ocupam espaço em memória. Isto é feito identificando os contornos ou os limites do organismo como o meio externo e, se for possível, redimensiona o arquivo original contido na matriz MVX criada pela subrotina PHANTOM1 e PHANTOM2.

2.4.4 Subrotina SETUP

Determinados órgãos do organismo podem ter doses absorvidas calculadas de forma mais detalhada e não apenas doses médias em seus volumes. A escolha dos órgãos de maiores interesses, onde se deseja obter valores de doses mais detalhados, podem ser selecionados através de especificações fornecidas no arquivo de dados de entrada como citado no programa MAIN.

Nesta subrotina, o volume de cada órgão de interesse é subdividido em pequenos sub-volumes ou sub-regiões onde as doses médias são calculadas. Desta forma, quanto menor o sub-volume maior será o detalhamento na determinação da distribuição de dose espacial no órgão de interesse em questão. No caso limite onde existiria o máximo de detalhamento, cada sub-volume corresponde ao próprio elemento de volume unitário do qual o órgão é constituído.

As relações espaciais entre as coordenadas i , j e k de cada elemento de volume ou “voxel” da matriz tridimensional MVX, e a localização espacial das sub-regiões são armazenadas na Matriz LOCALI. Ainda nesta subrotina são determinadas as massas de cada órgão e suas respectivas sub-regiões.

2.4.5 Subrotina UNIVMAP

Após a divisão do órgão ou dos órgãos em sub-regiões, os elementos de volume são renumerados de acordo com as suas localizações, atribuindo-se um número de identificação de acordo com a sub-região a qual pertence. Neste ponto determina-se a

quantidade de elementos de volume em cada sub-região que é armazenada na variável NVXORG. Também é determinada a quantidade de sub-regiões em cada órgão e o resultado armazenado na variável NREGION.

Cada órgão ou sub-região é relacionado a um valor de universo e que define as diferentes regiões na geometria do problema. Desta forma, cada órgão ou sub-volume do órgão fica especificado pelo número do universo e se torna a sua identificação. Esta forma de construção geométrica possibilita uma grande flexibilidade no detalhamento dos órgãos de interesse o aperfeiçoa a utilização de memória computacional.

O objeto simulador criado por este processo fica dividido em vários sub-volumes ou sub-regiões, cada um associado a um universo. A cada elemento de volume atribui-se o valor do universo correspondente ao sub-volume a qual ele pertence. Por exemplo, se um determinado elemento de volume também estará relacionado ao universo 20, desta forma, todos os elementos de volume de um mesmo sub-volume estará relacionado ao universo deste. A informação sobre os valores de universo correspondentes a cada elemento de volume pertence a um sub-volume que por sua vez pertence a um órgão é armazenada na matriz MVX.

2.4.6 Subrotina FILLCARD

Essencialmente esta subrotina escreve a matriz MVX criada pela subrotina UNIVMAP no formato adequado e compatível com o arquivo de entrada do programa MCNP-4B para que este possa reconhecer e processar os dados.

2.4.7 Subrotina TALLIES

Esta subrotina define os órgãos ou sub-volumes dos órgãos onde será computada a fração de energia depositada, para posterior cálculo da dose absorvida. O número de sub-volumes no órgão pode ser escolhido, de acordo com a informação fornecida pelo usuário, na opção Sinp do programa principal. Quanto maior este número, maior será o detalhamento da distribuição de dose na região ou órgão de interesse.

2.4.8 Subrotina SOURCE1, SOURCE2

A preparação do arquivo de fontes radioativas para o programa MCNP está dividida em duas subrotinas, a saber: SOURCE1 e SOURCE2. A subrotina SOURCE1 é utilizada para fontes distribuídas uniformemente no órgão ou região fonte. A subrotina SOURCE2 é utilizada em situações quando a fonte está distribuída de forma heterogênea.

2.5 A linguagem de programação JAVA

Linguagens como Fortran-77 utilizam programação estruturada, este tipo de programação preconiza que todos os programas possíveis podem ser reduzidos a apenas três estruturas: sequência, decisão e repetição. Programação estruturada não emprega diretamente conceitos como documentação de código e reutilização dos mesmos, tornando a manutenção trabalhosa, sobretudo em aplicações com muitas linhas de programação.

O conceito de programação orientada a objetos foi desenvolvido devido esta necessidade, este tipo de programação é paradigma de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamadas de objetos. Em resumo, a orientação a objeto surgiu devido à necessidade de códigos de máquinas mais claros, com isso, facilita a manutenção e reaproveitamento de código.

Algumas linguagens de computação foram desenvolvidas seguindo os conceitos de programação orientada a objetos, um grande exemplo é a linguagem Java.

A Linguagem de Programação Java, desenvolvida pela Sun Microsystems, teve sua origem nas pesquisas de um pequeno grupo de desenvolvimento, denominado Green. Este grupo tinha como objetivo criar novas gerações de dispositivos portáteis capazes de se comunicarem de forma diferente, ampliando sua potencialidade de uso, em que sua portabilidade estaria relacionada diretamente ao hardware utilizado, independentemente do sistema operacional.

As bibliotecas do Java, em relação ao controle de entrada e saída de dados, conhecidos como *input* e *output*, são orientadas a objetos usando os principais conceitos de interface, classes abstratas e polimorfismo. O polimorfismo no pacote “java.io”

utiliza fluxos de entrada e de saída para toda e qualquer operação, seja ela relativa a um arquivo ou também a entrada e saída padrão de um programa. As classes abstratas “InputStream” e “OutputStream” definem respectivamente o comportamento padrão dos fluxos em Java, em que, em um fluxo de entrada é possível ler bytes e no fluxo de saída escrever os bytes. (Wesley, 2000)

Os programas Java são compilados para uma forma intermediária de código denominado *bytecode*, que utiliza instruções e tipos primitivos de tamanho fixo, ordenação e uma biblioteca de classes padronizada. O *bytecode* é como uma linguagem de máquina destinada a uma única plataforma, a Máquina Virtual Java (*JVM – Java Virtual Machine*), um interpretador de *bytecode*. É possível utilizar a Máquina Virtual Java em qualquer plataforma, de forma que se obtém um programa computacional a ser executado em qualquer arquitetura que possua de uma *JVM* disponível.

A linguagem de programação Java foi projetada para ser compacta independente de plataforma e com a possibilidade da utilização em redes, utilizando o esquema de interpretação do código através de método *bytecode*. Em razão da linguagem ser interpretada, o desempenho pode ser melhorado devido à disponibilidade de compiladores do tipo JIN (*Just in Time*) que compilam o *bytecode* para código nativo da plataforma, otimizando o processo de execução.

Dentre alguns recursos para o desenvolvimento de aplicações é possível utilizar o recurso de múltiplas rotinas concorrentes, agregadas de elementos para a sincronização das rotinas. Estes fluxos de execução são denominados “*threads*”, recurso de programação importante para aplicações que exigem cálculos mais complexos. (James et al, 2000)(Deitel, 2003)(Berg, 1999)

A linguagem Java foi projetada para atender às necessidades do desenvolvimento de aplicações em um ambiente distribuído, em rede, e heterogêneo, por isso é tão utilizado dentro de empresas.

Segundo (TIOBE, 2011), Java é a linguagem de programação mais utilizada no mundo, sobretudo devido a sua característica multiplataforma, por ser gratuita e por possuir uma grande quantidade de bibliotecas com vários recursos já implementados, facilitando bastante o trabalho do desenvolvedor.

Outra grande vantagem da tecnologia são os ambientes de desenvolvimento integrados, *IDEs*, gratuitos como o Eclipse (ECLIPSE, 2011) e o Netbeans (NETBEANS, 2011), este último utilizado neste trabalho.

2.6 Objetos Simuladores

Um objeto simulador de corpo humano é uma representação digital ou matemática, total ou parcial do corpo humano. Sua principal finalidade é o cálculo de dose em um ou mais órgãos ou regiões, e neste sentido, geralmente são segmentados (divididos) em órgãos. Uma tomografia computadorizada ou ressonância magnética de um paciente é considerada um objeto simulador, contudo elas não são divididas por órgãos, e sim, por materiais.

Este trabalho utilizou dois tipos de objetos simuladores, alguns segmentados por órgãos, e outros, proveniente de pacientes específicos, segmentados por materiais.

Objetos simuladores geralmente são arquivos ASCII e são salvos sequencialmente em camadas ou fatias. Estas fatias sequenciais podem ser unidas por softwares específicos de processamento de imagens, dando origem a uma matriz tridimensional, onde cada unidade desta matriz é denominada *voxel*.

Segundo Hees (2006), o *voxel* numa visualização volumétrica é como o pixel para uma imagem bidimensional. Correspondem às menores partes unitárias das imagens, porém no caso do *voxel* trata-se de um elemento de volume.

Alguns objetos simuladores são representações geométricas (cilindros, cones, esferas, etc) dos órgãos do corpo humano com dimensões e valores bastante aproximados como mostrado na Figura 31. Entretanto, com o avanço tecnológico das técnicas de imagem foi possível criar objetos simuladores cuja forma geométrica é muito próxima da anatomia real.

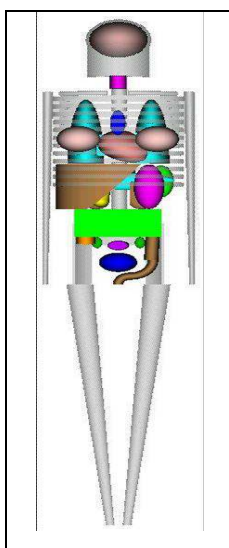


Figura 31 - Objeto simulador humano matemático (RADAR, 2011).

A dosimetria com objetos simuladores humanos teve início no final da década de 50, entretanto, devido a limitações computacionais da época, a representação humana era basicamente uma esfera maior contendo pequenas esferas em seu interior representando os órgãos. (RADAR, 2011)

Em 1975, a Comissão Internacional de Proteção Radiológica (ICRP - *International Commission on Radiological Protection*) definiu o primeiro objeto simulador padrão de um humano adulto usando dados dos cientistas Fisher e Snyder. (RADAR, 2011)

Posteriormente, em 1987, Cristy e Eckerman desenvolveram mais seis objetos simuladores representando outros modelos anatômicos, tais como, crianças e adolescentes, além de adultos com mais detalhes, e com maior número de órgãos que o objeto simulador de Fisher-Snyder (RADAR, 2011). Em 1995, Stabin e colaboradores desenvolveram objetos simuladores mais detalhados, além de três femininos com feto em diferentes estágios da gravidez. (RADAR, 2011) (Stabin et al, 1995)

Atualmente, os objetos simuladores não são mais desenvolvidos com sólidos geométricos, mas são produzidos a partir de imagens de tomografias computadorizadas, ressonâncias magnéticas ou até mesmo por fotos de corpos humanos milimetricamente fatiados. (VISIBLEHUMAN, 2011)

2.7 O programa ImageJ

O programa ImageJ é uma ferramenta livre e de código aberto (Abramoff et al, 2004), amplamente utilizado para processamento de imagens. Ele foi desenvolvido na linguagem de programação Java, e tem a vantagem de carregar diversos tipos de imagens, incluindo o formato *DICOM*.

O ImageJ é capaz de carregar imagens texto, formato ASCII, e também possui bibliotecas de colorização indo de cores quentes e frias a tons de cinza, dependendo da necessidade do usuário. Este programa possui também uma característica interessante, pois permite que novos *plugins* sejam adicionados sem que seu código precise ser recompilado, para isso esses *plugins* devem ser desenvolvidos utilizando sua biblioteca.

Como foi desenvolvido em Java ele herda sua característica multiplataforma, e por esta e todas as qualidades acima citadas de modificação este software é bastante utilizado no meio acadêmico.

2.8 Arquivos *DICOM*

O padrão *DICOM*, abreviação de *Digital Imaging and Communications in Medicine* (Comunicação de imagens digitais em medicina), foi criado com a finalidade de padronizar a formatação de imagens diagnósticas como tomografias, ressonâncias magnéticas, radiografias, ultrassonografias etc. Trata-se, portanto, de uma série de regras que permite que imagens médicas e informações associadas sejam trocadas entre equipamentos de diagnóstico geradores de imagens, computadores e hospitais. Atualmente está na terceira versão e foi desenvolvido pelo American College of Radiology (ACR) e pelo *National Electrical Manufacturers Association* (NEMA).

Além dos arquivos de imagem, o padrão armazena informações contendo o tipo de equipamento, nome e modelo, por exemplo, e informações do paciente como nome, data de nascimento, RG, altura e peso.

No começo dos anos 80 era impossível que alguém, além dos fabricantes de equipamentos de tomografia computadorizada ou de ressonância magnética, conseguisse decodificar as imagens que esses equipamentos geravam.

Físicos e radiologistas tinham interesse em usar essas imagens para efetuar um planejamento de dose para tratamentos utilizando radiação. O ACR e o NEMA se uniram e formaram um comitê para definir um padrão no ano de 1983.

A primeira edição do padrão, chamado inicialmente de ACR/NEMA 300, surgiu em 1985. Após seu lançamento ficou claro a necessidade de melhorias, pois o texto era vago e possuía contradições. A segunda edição só foi lançada em 1988, teve seus problemas corrigidos e foi mais aceita pelos fabricantes. Só a partir da terceira edição, no ano de 1993, o padrão passou a se chamar *DICOM*. Atualmente o padrão continua na terceira edição, contudo novas versões com melhorias continuam sendo lançadas, mas ao invés de edições numeradas e sequenciais, agora elas são referenciadas pelo ano da edição.

2.9 O programa ICCT

O ICCT, *Image Converter Computed Tomography*, é um software que converte arquivos DICOM em arquivos ASCII, que é o formato compatível com a atual versão do SCMS. Este software foi desenvolvido no IPEN como parte de um projeto de mestrado (MASSICANO, 2010) e foi usado para realizar as conversões dos arquivos no presente trabalho.

Além do arquivo do objeto simulador, criado a partir do arquivo *DICOM* do paciente, o ICCT também fornece o mapa de densidades dos materiais e a lista dos materiais, este último é gerado seguindo o formato de materiais utilizado pelo código MCNP.

2.10 O programa ACPS

Assim como o ICCT, no IPEN também está em desenvolvimento o software *Attenuation Correction PET SPECT* (ACPS) que efetua a correção de atenuação em imagens de PET e SPECT mediante o método de Chang de 1ª e 2ª ordem e gera a distribuição da atividade relativa no interior do paciente. A imagem corrigida resultante deste software será utilizada para a elaboração ou modelagem da distribuição da fonte radioativa dentro do objeto simulador, através da nova versão do SCMS em desenvolvimento neste trabalho.

3. RESULTADOS

O software SCMS original, desenvolvido em Fortran 77, possui cerca de duas mil linhas de programação, não está documentado, mas possui um manual de instruções em inglês.

A conversão do software para a linguagem de programação Java começou depois de um pequeno estudo sobre Fortran 77, este estudo se concentrou na sintaxe da linguagem, na forma de entrada e saída dos arquivos armazenados no computador e no tipo das variáveis utilizadas pelo software.

As linguagens de programação Fortran e Java têm suportes semelhantes quanto aos tipos de variáveis, mas a dimensão máxima delas é diferente, portanto pequenas alterações foram feitas em algumas variáveis. Variáveis com precisão dupla,

Double, utilizadas em Fortran puderam ser modificadas para o tipo “inteiro” quando o software foi reescrito na linguagem Java. Com essas alterações ganha-se em desempenho, pois menos memória é utilizada durante o tempo de execução do software.

Uma grande dificuldade durante a conversão do software foi a substituição do comando GOTO, presente somente na linguagem Fortran 77 e não em Java. Este comando transfere a sequência de execução de um software levando-o a qualquer outro lugar do código. Nas duas figuras abaixo, Figura 32 e Figura 33, são mostrados códigos semelhantes, mas escritos em linguagens diferentes. Neste código, a execução termina quando a condição é atingida, em Fortran é fácil interromper a execução de um laço triplo usando o comando GOTO, contudo em Java é preciso usar a funcionalidade “*labeled break*” para simular a mesma lógica da linguagem Fortran 77.

```
det_kmin:
{
  for (int k = 1; k <= kmax2; k++) {
    for (int j = 1; j <= jmax; j++) {
      for (int i = 1; i <= imax; i++) {
        if (!(mvx1[i][j][k] == 0)) {
          kmin = k;
          break det_kmin;
        }
      }
    }
  }
}
```

Figura 32 - Implementação da funcionalidade “*labeled break*” em Java

```
do 101 k=1,kmax2
  do 101 j=1,jmax
    do 101 i=1,imax
      if(mvx1(i,j,k).eq.0) then
        goto 101
      else
        kmin=k
      goto 102
    endif
  101 continue
```

Figura 33 - Exemplo de código utilizando laço em Fortran 77

Depois de completada a conversão do software para Java foram adicionados diversos recursos e opções à nova versão do software. Os testes comparativos das duas versões do software (velha e a nova) foram realizados a cada mudança significativa no software ou periodicamente para evitar que um eventual erro imperceptível persistisse em versões mais novas. O procedimento de testes comparativos consistiu em comparar a saída das duas versões dos softwares, em Fortran e em Java, verificando-se em seguida a existência de inconsistências entre as versões. O novo software recebeu o nome de AMIGO, cujo acrônimo significa *Algorithm for Medical Image Generating Object*.

Para manter a funcionalidade do SCMS em trabalhar com objetos simuladores em formato ASCII, usados principalmente em dosimetria para cálculos de radioproteção, e incluir a nova funcionalidade de carregar imagens tomográficas ao software, optou-se por dividir o desenvolvimento do novo SCMS em duas versões. A primeira versão do software, chamado de AMIGO 1.0, é basicamente uma versão nova do SCMS com modificações, que incluem uma interface gráfica, um gerenciador de materiais e a inserção dos dados que contém os espectros de emissão energética de radiação devido ao decaimento de radiofármacos usados em medicina nuclear. A segunda versão do software, chamado de AMIGO 2.0, embora compartilhe de bibliotecas e funções do AMIGO 1.0, foi desenvolvida para ser executada unicamente com objetos simuladores provenientes de imagens tomográficas de pacientes específicos.

A principal vantagem desta divisão de softwares é a manutenção do código, contudo, o usuário também é beneficiado, pois a utilização dos dois softwares fica simplificada, diminuindo erros na criação dos arquivos de dados de entrada.

Nas figuras abaixo são apresentados partes de arquivos, em formato ASCII, de objetos simuladores: o primeiro, Figura 34, representa tipicamente um objeto simulador usado em proteção radiológica onde o mesmo está segmentado por órgãos, de forma que, os *voxels* pertencentes a um mesmo órgão recebem o mesmo número de identificação. O segundo objeto simulador, Figura 35, é segmentado por tipo de material/tecido e é originado de imagens de tomografia de um paciente específico. Nota-se que, embora ambas as imagens possuam números inteiros como índice, no primeiro caso todos os *voxels* do mesmo órgão possuem o mesmo valor ou número de

identificação, enquanto, na segunda imagem o número de identificação de cada *voxel* corresponde ao tipo de tecido e não ao órgão a qual pertence.

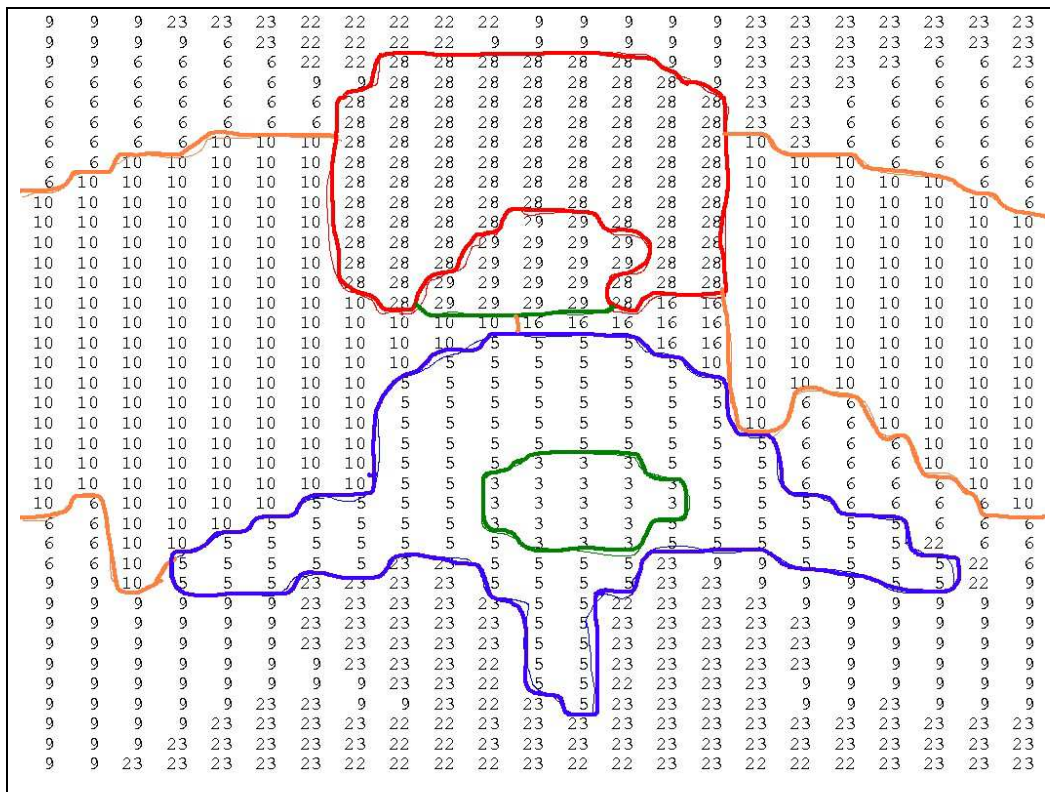


Figura 34 - Objeto simulador de *voxel* cujo arquivo está em formato ASCII

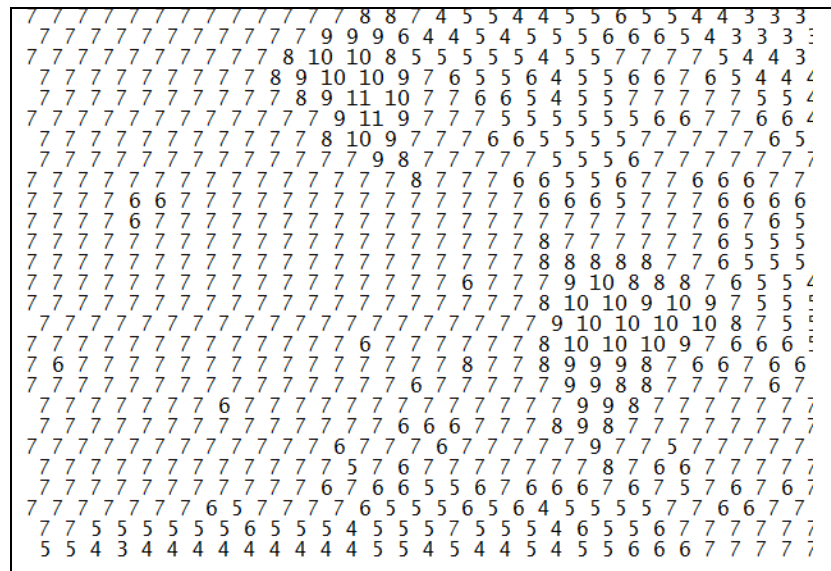
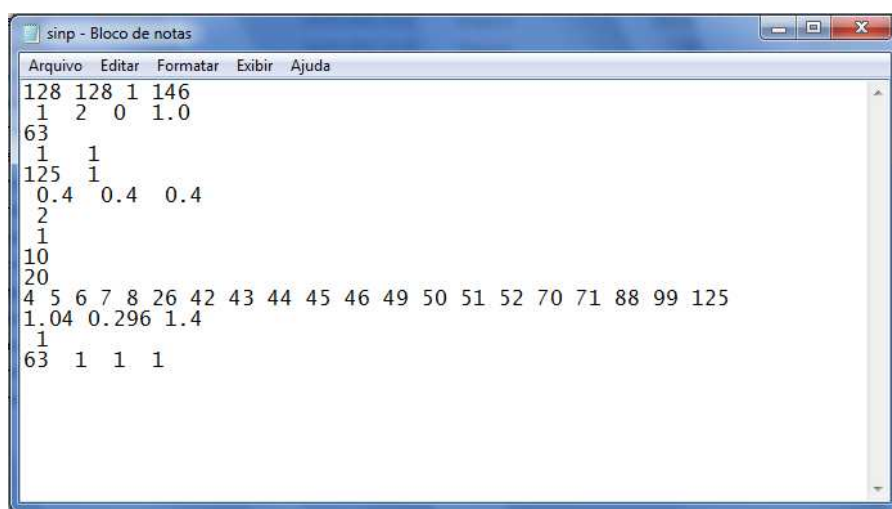


Figura 35 - Objeto simulador de um paciente específico

3.1 O software AMIGO 1.0

A principal mudança incorporada no software AMIGO 1.0 em relação ao SCMS, foi a implementação de uma interface gráfica ou *GUI* (*Graphical User Interface*), que tem a função de substituir o antigo arquivo de entrada chamado SINP. Este arquivo é composto de informações relacionadas aos dados anatômicos e da fonte radioativa, mesclado com opções de cálculo a serem transmitidas ao código de Monte Carlo, MCNP, veja Figura 36.



```

sinp - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
128 128 1 146
1 2 0 1.0
63
1 1
125 1
0.4 0.4 0.4
2
1
10
20
4 5 6 7 8 26 42 43 44 45 46 49 50 51 52 70 71 88 99 125
1.04 0.296 1.4
1
63 1 1 1

```

Figura 36 - Arquivo de entrada de dados, SINP, do software SCMS.

O software AMIGO 1.0 foi desenvolvido em Java, versão 1.6, e para o desenvolvimento de sua interface gráfica foi utilizado o *toolkit* do próprio Java chamado *Swing*. Os componentes do *Swing* foram explorados para que a interface gráfica ficasse simples e intuitiva como é mostrada na Figura 37. A interface gráfica possui botões, caixas de texto, caixas de seleção e rótulos.

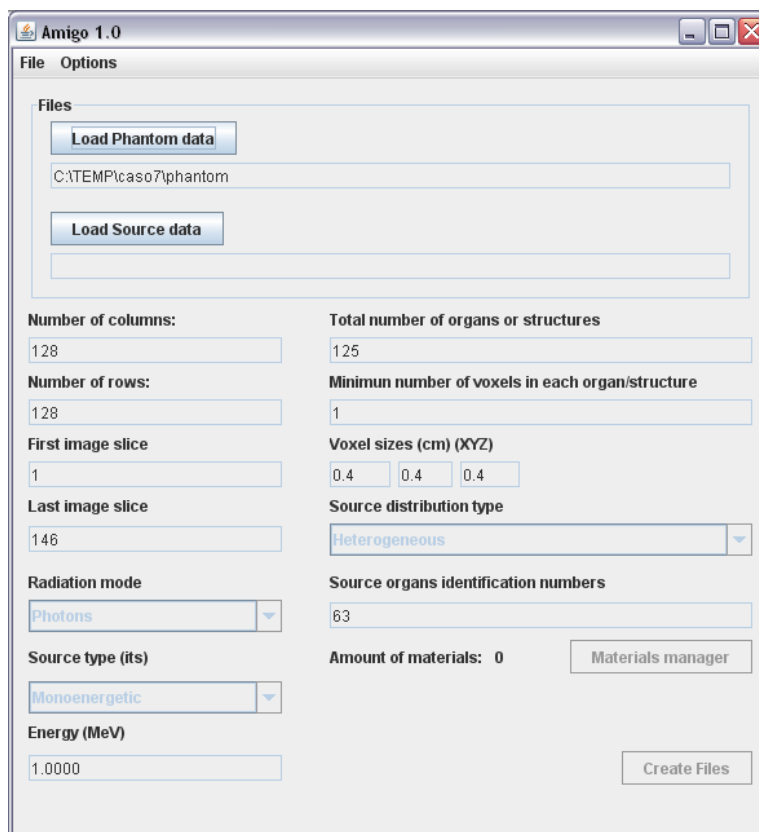


Figura 37 - Tela inicial do software AMIGO 1.0

Todos os menus, rótulos, botões e mensagens de erros estão em inglês, pois o software será disponibilizado para a comunidade internacional.

As mesmas opções de modificação do SCMS para construir um arquivo de entrada para o MCNP estão presentes no AMIGO 1.0, como a especificação do tipo de radiação, a energia da fonte, dimensões do *voxel*, dentre outras opções. Algumas opções foram melhoradas para deixar o preenchimento mais intuitivo como o caminho e o nome dos arquivos do objeto simulador e da fonte de radiação, pois agora estes arquivos podem ter qualquer nome e estar em qualquer lugar do computador do usuário.

Os campos *Number of columns*, *Number of rows*, *First image slice* e *Last image slice* informam ao software as dimensões da geometria dos arquivos de entrada, do objeto simulador e da fonte de radiação caso ela seja heterogênea, pois neste caso a fonte é descrita num arquivo ASCII.

Os próximos campos: *Radiation mode*, *Source type*, *Energy (MeV)*, *Source distribution type* e *Source organs identification numbers* são configurações da fonte de radiação. A Figura 38 mostra a caixa de seleção, *Radiation mode*, onde o usuário

consulta qual partícula foi selecionada e visualiza quais partículas podem ser selecionadas para o transporte no MCNP, são elas: elétrons, fótons e nêutrons.

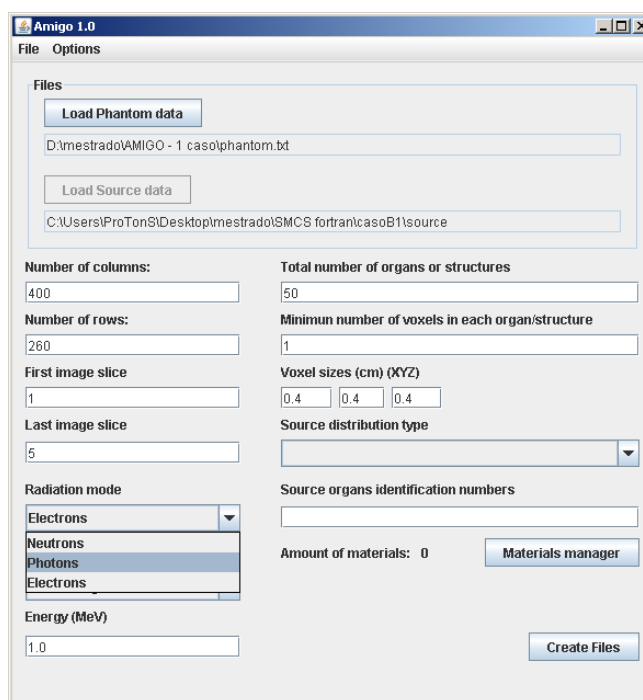


Figura 38 - Detalhe da caixa de seleção do software AMIGO 1.0.

O campo *Source type* é uma caixa de seleção em que seu valor pode ser *Spectrum* ou *Monoenergetic*. Este campo permite que o usuário escolha o tipo de fonte de radiação, podendo ser monoenergética, que possui uma única energia, ou por espectro energético. No exemplo da imagem acima está selecionado uma fonte monoenergética de fótons de 1 MeV.

Na Figura 39A é mostrado o que acontece na interface gráfica quando o usuário escolhe *Spectrum* no item *Source type*. O campo *Energy (MeV)* é substituído por um campo de seleção e por um botão onde o usuário escolhe qual espectro será incluído na simulação, que normalmente está relacionado a um radioisótopo específico. O botão *Plot* gera um gráfico deste espectro para que o usuário visualize e verifique se é o espectro selecionado, Figura 39B.

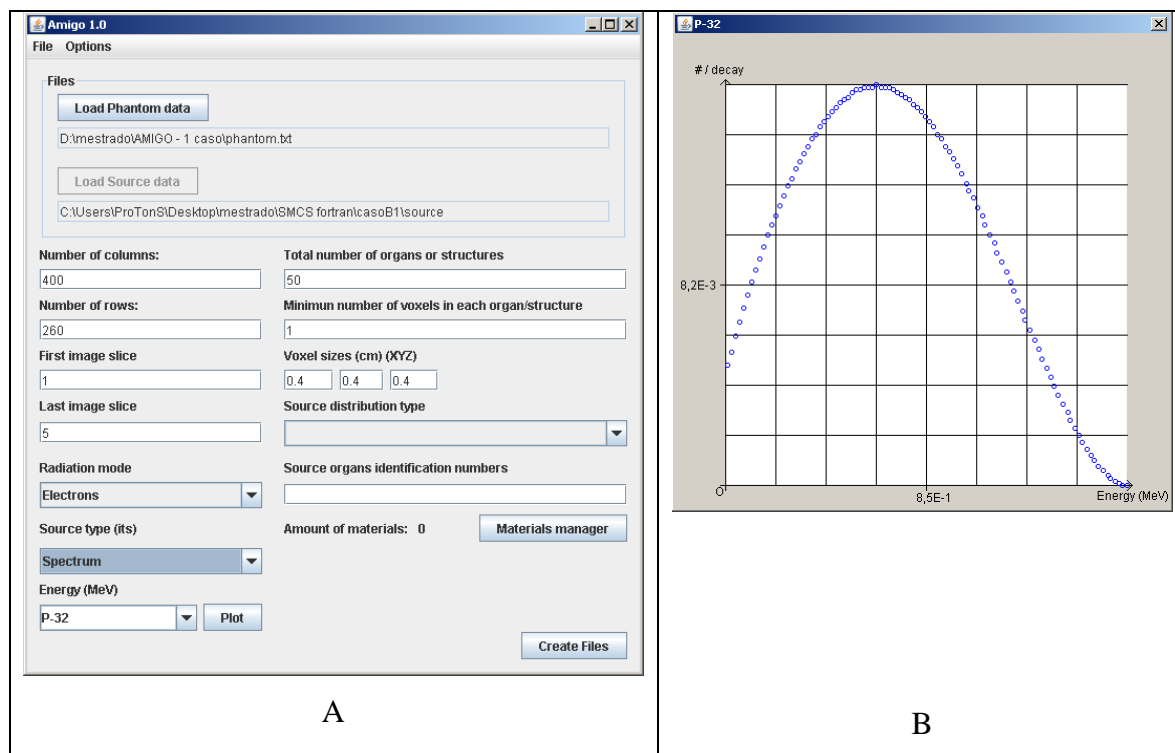


Figura 39 - Escolhendo um espectro de energia de emissão da fonte radioativa

O campo de seleção *Source distribution type*, Figura 38, possui duas opções: *Heterogeneous* ou *Homogeneous*, eles definem como será a fonte. Em ambos os tipos de fonte, homogênea ou heterogênea o usuário precisará selecionar o órgão ou órgãos onde a fonte da simulação estará localizada. Se a fonte de radiação for heterogênea o usuário precisará carregar, além do arquivo do objeto simulador, um arquivo adicional contendo a distribuição da fonte de radiação, se for homogênea o usuário terá apenas que selecionar pelo menos um órgão para ser a fonte. O campo *Source organs identification numbers* deve ser preenchido com o número do órgão ou os números dos órgãos que serão as fontes de radiação da simulação.

Os três últimos campos, *Total number of organs or structures*, *Minimum number of voxels in each organ/structure* e *Voxel sizes (cm)*, representam, respectivamente, a quantidade de órgãos diferentes que existem no arquivo de entrada do objeto simulador, o número mínimo de *voxels* em cada órgão e a dimensão desses *voxels*.

Toda a simulação utilizando o código MCNP precisa da definição dos materiais que compõem a geometria. Para facilitar o gerenciamento destes materiais foi implementado um gerenciador de materiais que pode ser acessado através do botão *Material Manager*, presente na interface gráfica.

O SCMS trabalha com apenas três materiais, mas isso foi modificado no AMIGO 1.0, visto que na presente versão o software suporta até 60 materiais. Este número foi escolhido por ser o dobro do necessário para se carregar um objeto simulador construído com o software ICCT. Foi adotada uma margem de segurança grande, pois o usuário talvez tenha necessidade de criar outros materiais não contemplados no software ICCT.

O gerenciador de materiais é mostrado na Figura 40. Alguns órgãos possuem materiais iguais, por isso o campo *Material of the organ(s)* pode ser preenchido com um ou mais órgãos, neste exemplo, o campo foi preenchido com o número “12”. Se, além do número “12”, também fosse adicionado o número “13”, ambos os órgãos, “12” e “13”, seriam preenchidos pelo material cujo número de identificação é 12 e o campo *Material of organ(s)* deveria ter os valores “12” e “13” separados por um espaço.

The screenshot shows a window titled "Material 12 : m12" with the following fields and controls:

- Maximum number of materials: 60
- Total of organs: 50
- Materials used in the project: A list of numbers from 7 to 50, with 12 highlighted.
- Buttons: First Material, Previous Material, Next Material, Last Material
- Name: m12
- Material Description: (Linha limitada a 60 caracteres)

m12	1000.	-7.5	6000.	-35.8	7000.	-3.1
	8000.	-38.1	12000.	-0.1	15000.	-4.8
	16000.	-0.2	17000.	-0.1	19000.	-0.1
	20000.	-10.3				
- Material density: 1.28
- Material of the organ(s): 12
- Button: organs that remain
- Buttons: Remove Material, Salvar, Sair

Figura 40 - Tela de gerenciamento de materiais

3.2 O software AMIGO 2.0

O software AMIGO 2.0 tem um fluxo de execução diferente do AMIGO 1.0 e do SCMS, pois ele foi desenvolvido para trabalhar apenas com objetos simuladores de pacientes específicos. Devido às diferenças nos arquivos de entrada, tanto o código de construção da geometria quanto às especificações dos *tallies* foram reescritos no

AMIGO 2.0. Por outro lado, o método de construção da fonte de radiação foi alterado, mas não precisou ser reescrito.

Tanto objetos simuladores de pacientes específicos quanto os mapas de atividades, provenientes de exames PET e SPECT, estão em formato *DICOM*, por causa disso o AMIGO 2.0 trabalha em conjunto com os softwares ICCT e ACPS como é mostrado na Figura 41.

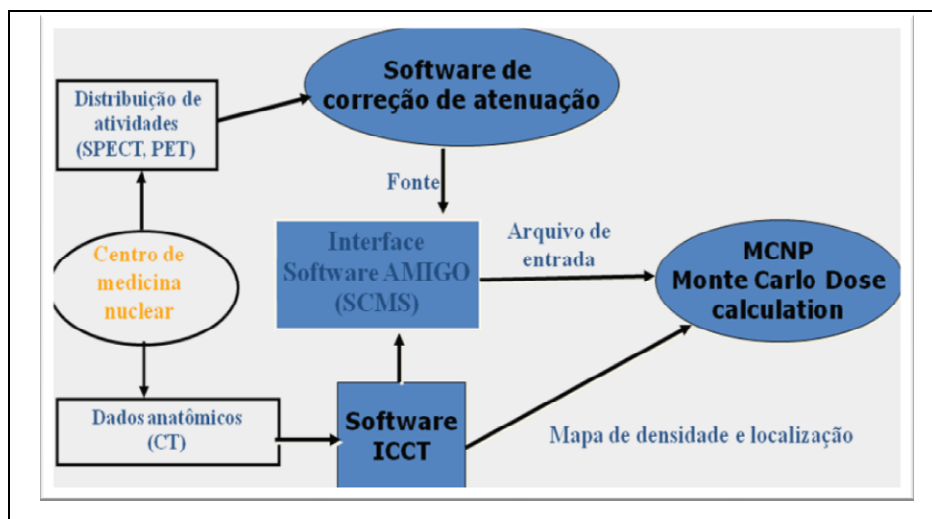


Figura 41 - Metodologia para o cálculo de dose implementada para uso com o software AMIGO 2.0

O fluxograma da nova metodologia mostra que tanto os arquivos contendo a fonte de radiação quanto o objeto simulador são encaminhados ao ACPS e ao ICCT, respectivamente, para serem processados e posteriormente utilizados no AMIGO 2.0.

A interface gráfica do AMIGO 2.0 é mostrada na Figura 42, onde se observa sua tela inicial. Comparando as duas interfaces gráficas, isto é, do AMIGO 1.0 e do AMIGO 2.0, nota-se que esta segunda interface é mais simples, pois existem menos opções de preenchimento. Por exemplo, não existem mais os itens *Source organs identification numbers* e *Source distribution type*, que são campos que fornecem informações relacionadas ao órgão ou órgãos onde a fonte de radiação está localizada, órgãos fonte. Como não existem mais a identificação dos órgãos estes campos foram suprimidos nesta versão do software.

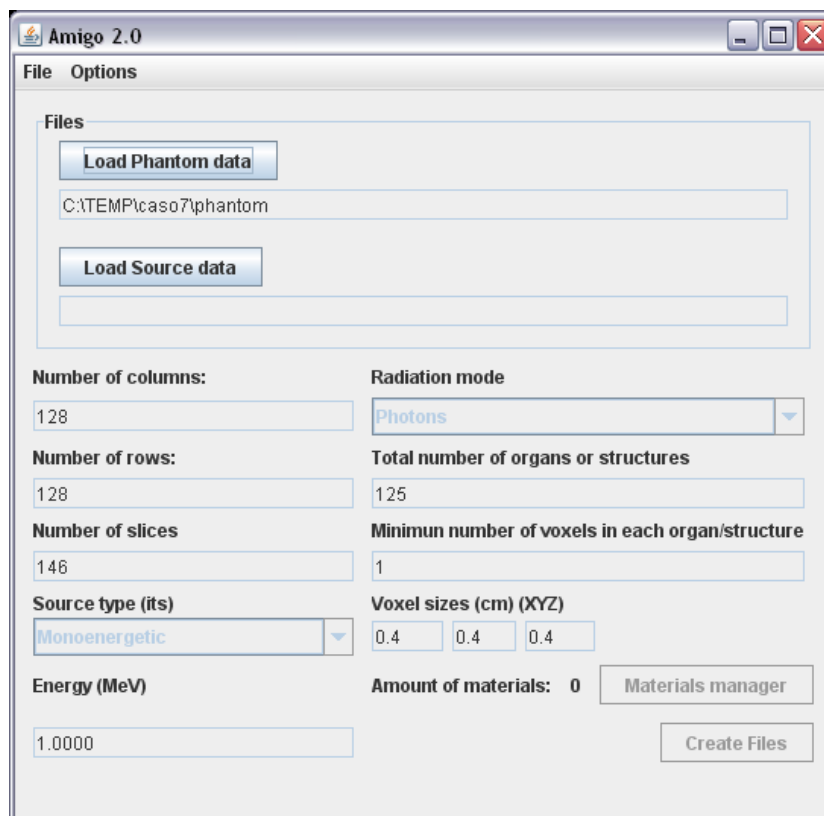


Figura 42 - Tela inicial do segundo software do sistema AMIGO

Diferentemente da versão anterior, onde é opcional, o uso de uma fonte heterogênea é obrigatório no AMIGO 2.0. Esta obrigatoriedade deve-se ao fato de não existir segmentação por órgãos em objetos simuladores de pacientes específicos, não sendo possível escolher um órgão para ser a fonte. A fonte de radiação no AMIGO 2.0 é definida utilizando o arquivo com os dados do mapa de atividades, obtidas através de imagens PET e SPECT, fornecido pelo software ACPS.

Como algumas fontes de radiação ocupam toda a dimensão do objeto simulador e sempre existe um limite de memória computacional, impôs-se uma limitação no tamanho de fonte, considerando-se somente valores superiores a centésima parte do seu valor máximo. Este critério foi adotado após diversas simulações de situações “reais” associado aos recursos computacionais no período de desenvolvimento do presente trabalho, podendo ser modificado, a medida que equipamentos com maior capacidade forem adquiridos, ou com novos avanços tecnológicos na área de computação permitindo maior capacidade de processamento.

Outras opções da interface gráfica foram mantidas, como a presença do gerenciador de materiais, a opção de adição de espectros de radiofármacos e o tipo da fonte de radiação que pode ser monoenergética ou com distribuição espectral.

Objetos simuladores de pacientes específicos não são segmentados por órgãos, uma vez que cada ponto da matriz bidimensional ou tridimensional representa uma região distinta, de forma que a estimativa de dose será sempre realizada no nível de *voxel*. Entretanto, foi desenvolvida uma opção em que o usuário tem a possibilidade de escolher a região de interesse onde se deseja estimar a distribuição de dose. Para especificar o local de cálculo de dose o usuário define um ROI (Região de Interesse) diretamente na imagem de tomografia computadorizada como é mostrado na Figura 43. Cada fatia da tomografia pode ter uma região de interesse onde a dose será calculada. Esta tela pode ser acessada através do menu *Options*, localizado acima da tela principal, e depois *Target Organs*.

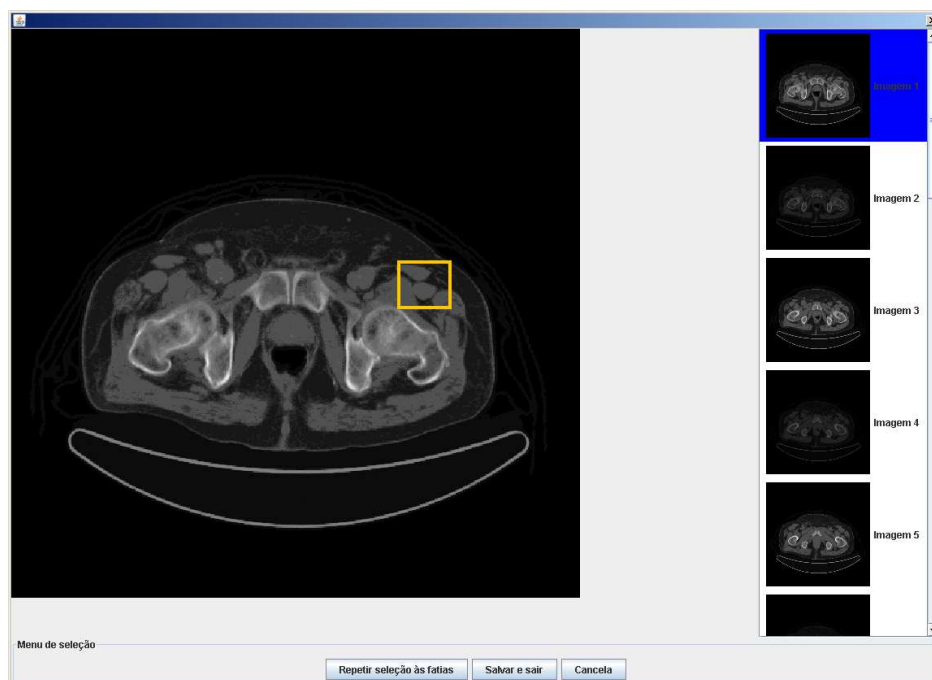


Figura 43 - Tela para escolha da região de interesse para o cálculo da distribuição de dose.

A tela onde o usuário define a região de interesse para o cálculo da dose possui uma barra de rolagem na lateral esquerda e é nesta barra que o usuário seleciona em quais fatias a dose será calculada e em qual região.

Alguns objetos simuladores podem possuir várias fatias, portanto, para automatizar o procedimento existe o botão, “*Repeat selection for all slices*”, onde o usuário repete a região de interesse atual em todas as outras fatias da tomografia computadorizada.

Depois que o usuário seleciona todas as opções necessárias ao software o arquivo de saída do AMIGO 2.0, que corresponde ao arquivo de entrada do código MCNP5, estará pronto para ser criado, a partir do acionamento do botão “*Create Files*” na tela principal.

O código MCNP5 permite uma série de configurações no arquivo de saída, das quais algumas já foram comentadas, como a exibição dos parâmetros das 50 primeiras partículas da fonte simulada. O código produz uma grande quantidade de informação estatística a respeito da simulação, como erro relativo de cada *tally* computado, além de testes para verificar a qualidade estatística da simulação.

O arquivo de saída do código MCNP5 também fornece o resultado dos *tallies* escolhidos pelo usuário. Esta informação fica separada numa parte do arquivo, contudo, para facilitar o carregamento e a manipulação dos dados por softwares de dosimetria, usou-se neste trabalho, uma opção do código MCNP5 que cria, além do arquivo de saída padrão, outro arquivo, chamado MCTAL, que armazena somente os valores dos *tallies* e suas incertezas.

O código MCNP5 não fornece a dose diretamente, mas a energia depositada em MeV/g ou MeV. A conversão para dose em unidade de Gray (J/kg) é efetuada utilizando-se o mapa de densidade dos materiais fornecida pelo software ICCT.

O mapa de densidade é um arquivo com as mesmas dimensões que o arquivo do objeto simulador e da fonte de radiação, mas, como o nome sugere, ele armazena a densidade do material em cada *voxel*. Portanto, além de criar o arquivo de entrada para o código MCNP5 o AMIGO 2.0 também efetua a conversão dos resultados em dose na região selecionada pelo usuário.

Para que a distribuição de dose pudesse ser visualizada em forma de imagem também foi desenvolvido o software de visualização chamado *Doseview*. Este aplicativo carrega os arquivos da fonte, o objeto simulador, o mapa de densidades fornecido pelo ICCT, o arquivo de saída do MCNP5, onde são salvos os dados da simulação, e, por fim, o arquivo MCTAL que contém os valores de energia depositada por *voxel* com os respectivos erros relativos.

Na Figura 44 é exibida a tela inicial do *Doseview* onde o usuário deve apontar onde estão os arquivos referentes à simulação. Já na Figura 45 é mostrada a tela do *Doseview* onde são exibidas as imagens do objeto simulador, lado esquerdo, e da distribuição de dose correspondente à região de interesse escolhido, lado direito.

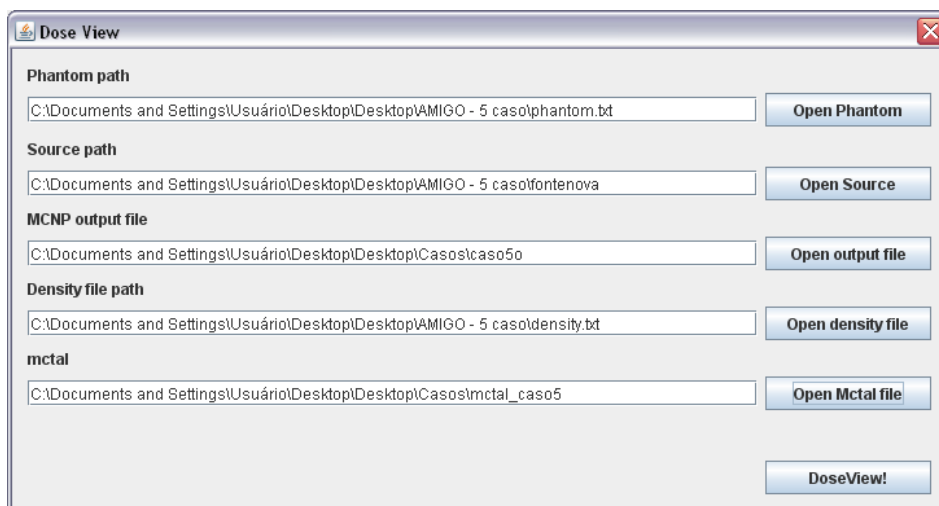


Figura 44 - Tela de seleção dos arquivos para o cálculo da dose

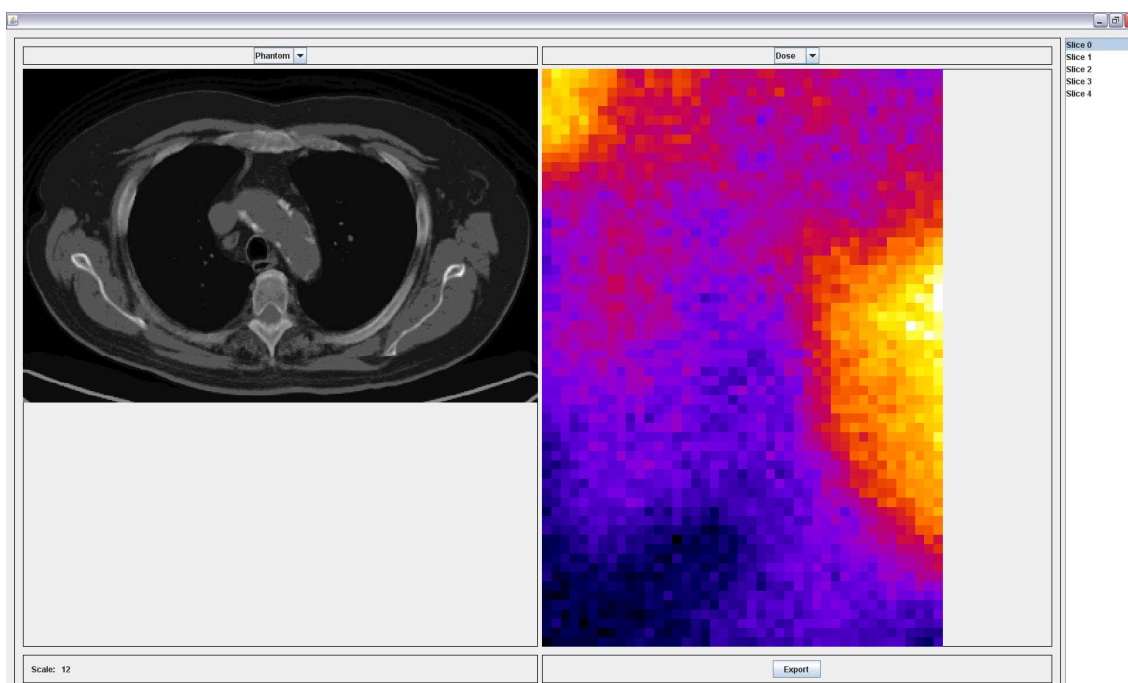


Figura 45 - Tela da aplicação *Doseview*.

Por fim, a Figura 46 exibe a distribuição da fonte de radiação ao lado da distribuição de dose na região de interesse obtida pela simulação.

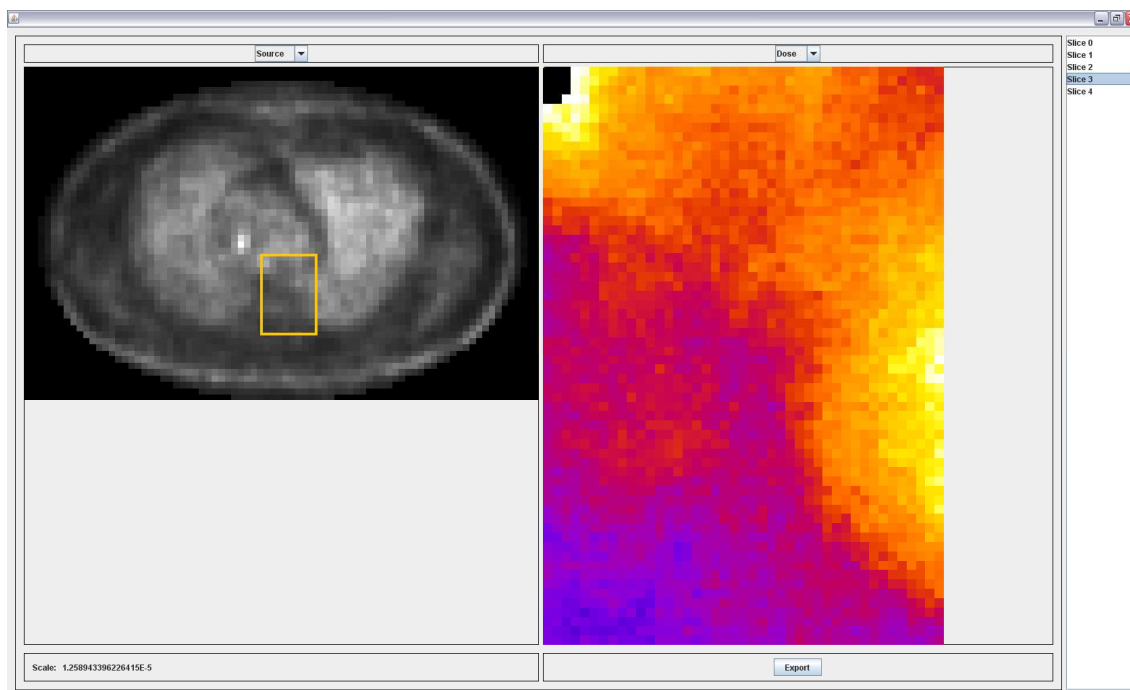


Figura 46 - *Doseview* com outros tipos de exibição

O software *Doseview* possui as opções de apresentar o objeto simulador, a distribuição da fonte de radiação, a distribuição de dose na região de interesse e a energia depositada para cada simulação realizada. A imagem da distribuição de dose exibida pelo software contempla apenas a região de interesse escolhida pelo usuário, portanto, na maioria dos casos, ela é muito menor que a imagem da fonte de radiação e do objeto simulador.

No painel da esquerda é exibido somente o objeto simulador ou a fonte de radiação, o primeiro é exibido por padrão, mas o usuário pode clicar na caixa de seleção e escolher visualizar a fonte de radiação, se preferir. Já no painel direito é exibida a distribuição de dose ou a energia depositada durante o cálculo da dose. Como no primeiro painel o usuário também pode optar por visualizar a energia depositada usando a caixa de seleção acima do painel.

Na parte inferior do software existe o campo de escala, *scale*, que mostra a escala aproximada da imagem em que o usuário estiver com o *mouse*. Na primeira imagem, Figura 45, onde é exibido o objeto simulador nota-se o número “12” referente ao material, já na segunda imagem, Figura 46, a escala é referente a dose em determinada região da fatia número “3”.

Para construir a escala de cores e associá-las aos valores de dose nas imagens foi utilizado o software ImageJ e um filtro de cores chamado fogo, onde cores

mais claras, como o branco, significam regiões com doses maiores e cores mais frias, como o azul, representam áreas com o valor de dose menor.

No campo à direita, a barra de rolagem permite que o usuário selecione as fatias onde serão obtidas as informações, pois as simulações geralmente são feitas em mais de uma fatia.

3.3 Validação do software AMIGO 1.0

Um total de 15 casos testes, *benchmarks*, foram propostos e executados para verificar os resultados do software AMIGO 1.0 com propósito de validação e averiguação de eventuais erros de programação. Os testes constituem em comparar a saída do SCMS com o AMIGO 1.0, verificando-se eventuais diferenças nos arquivos de saída.

Foram propostos quinze casos testes que abrangem uma variedade de situações onde se utilizam objetos simuladores diferentes. Alguns deles possuem arquivos com definições de fonte heterogênea, em outros as dimensões geométricas são variadas e por fim, há casos onde as fontes de radiação foram colocadas em órgãos diferentes.

Todos os casos testes foram executados no SCMS e seus os arquivos de saída foram armazenados, em seguida, o mesmo procedimento foi executado com o AMIGO 1.0. Por fim, os arquivos de saída de ambos os softwares foram, então, comparados usando-se o recurso do WORD da Microsoft e um pequeno aplicativo criado em linguagem Java com a finalidade de verificar as diferenças entre os arquivos.

Na Tabela 3 estão mostradas as principais características dos objetos simuladores utilizados nos primeiros casos testes. São modelos mais simples onde as dimensões do objeto simulador são reduzidas e a quantidade de objetos em seu interior é pequena se comparado aos demais modelos.

Tabela 3 - Casos mais simples de testes dos softwares.

	CasoA1	CasoB1	CasoC1	CasoA2	CasoB2	CasoC2
Eixo X	45	45	45	58	58	58
Eixo Y	25	25	25	43	43	43
Primeira fatia	1	1	1	1	1	1
Última fatia	20	20	20	53	53	53
Número de órgãos fonte	1	1	1	1	3	3
Partícula da fonte	Fótons	Fótons	Fótons	Fótons	Fótons	Fótons
Energia da fonte (MeV)	1,0	1,0	1,0	1,0	1,0	1,0
Órgãos que contém a fonte	3	3	3	3	2 5 10	2 5 10
Quantidade de órgãos no objeto simulador	3	3	3	12	12	12
Número mínimo de voxel em cada órgão	1	1	1	1	1	1
Comprimento do voxel para o eixo X (cm)	0,4	0,4	0,4	0,4	0,4	0,4
Comprimento do voxel para o eixo Y (cm)	0,4	0,4	0,4	0,4	0,4	0,4
Comprimento do voxel para o eixo Z (cm)	0,4	0,4	0,4	0,4	0,4	0,4
Distribuição da fonte	Homogênea	Heterogênea	Heterogênea	Homogênea	Homogênea	Heterogênea
Quantidade de órgãos/tecidos com material 1	1	1	1	10	10	10
Quantidade de órgãos/tecidos com material 2	1	1	1	1	1	1
Quantidade de órgãos/tecidos com material 3	1	1	1	1	1	1
Densidade do material 1 (grama/cm ³)	1,04	1,04	1,04	1,04	1,04	1,04
Densidade do material 2 (grama/cm ³)	0,296	0,296	0,296	0,296	0,296	0,296
Densidade do material 3 (grama/cm ³)	1,4	1,4	1,4	1,4	1,4	1,4
Quantidade de órgãos para calcular a energia depositada	1	1	1	3	5	5
Órgãos onde a energia depositada será calculada	1	1	1	2 6 9	2 6 9 11 12	2 6 9 11 12

A coluna da esquerda da tabela mostra a descrição dos parâmetros que o usuário deve inserir para criar um arquivo de entrada para o código MCNP. O nome do caso teste está localizado na primeira linha da tabela. Alguns parâmetros são extensos e outros são repetitivos, por isso eles foram omitidos, tal como, o tipo de fonte, que pode ser monoenergética ou espectro. Nestes primeiros 6 casos testes considerou-se sempre uma fonte monoenergética.

O usuário necessita especificar um ou mais objetos onde a energia depositada será calculada, contudo, existe a opção de dividir um objeto ou região em diversas sub-regiões onde será calculada a energia depositada de forma detalhada. O nível de detalhamento em sub-regiões é uma das opções de cálculo.

Como dito acima, os primeiros itens são referentes às dimensões geométricas do objeto simulador e da fonte heterogênea, se existir. Os outros parâmetros estão relacionados às características da fonte, da composição dos materiais e a escolha dos objetos onde a energia depositada será obtida com a possibilidade da divisão destes órgãos em sub-regiões.

O número de materiais destes casos testes está limitado a apenas 3, mesmo que o objeto simulador possua mais regiões/tecidos. Isso pode ser observado no item “Quantidade de órgãos no objeto simulador”, no Caso A2, por exemplo, existem 12 tecidos no objeto simulador, visto que 10 deles são compostos do material 1, segundo o item “Quantidade de elementos no material 1” da tabela. Os outros 2 tecidos são compostos, respectivamente, dos materiais 2 e 3.

Os casos testes estão agrupados de acordo com o tipo de objeto simulador, identificado pelo número no final do nome. Portanto, os casos A1, B1 e C1 são semelhantes entre si quanto à geometria do objeto simulador.

3.3.1 Casos testes: A1, B1 e C1

Na Figura 47 é mostrada a representação tridimensional do objeto simulador, correspondente aos casos testes A1, B1 e C1. Esta representação foi feita utilizando um dos *plugins* 3D do software *ImageJ* chamado *Volume Viewer* na versão 1.31. A geometria é composta de duas esferas de tamanhos e materiais diferentes contidas num espaço em forma de paralelogramo com um material diferente das esferas.

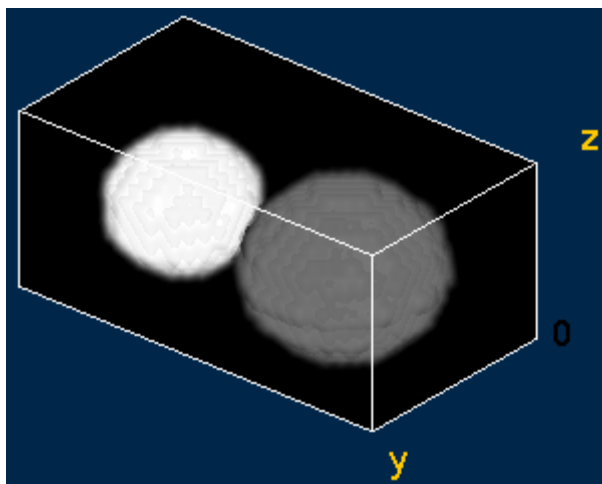


Figura 47 - Construção tridimensional do objeto simulador do Caso A1, Caso B1 e Caso C1

Nestes casos testes, a fonte se localiza numa das esferas (esfera fonte) e a energia depositada é obtida na outra (esfera alvo). No caso A1 a distribuição da fonte é homogênea já nos dois casos seguintes, B1 e C1, a fonte é heterogênea e requer um arquivo de entrada contendo dados desta fonte.

O arquivo da fonte tem as mesmas dimensões que o objeto simulador, contudo o tipo de dado do arquivo pode ser números decimais, diferentemente do objeto simulador que só pode conter números inteiros. Os valores da fonte são carregados e escritos no arquivo de entrada, visto que o MCNP5 se encarrega de normalizá-los.

Como mostra a Tabela 3, o Caso B1 e o Caso C1 possuem valores idênticos de entrada, contudo a quantidade de *voxels* na região de interesse para a obtenção da energia depositada é diferente. No Caso B1, o cálculo da energia depositada é requisitado em todos os *voxels* que compõe a esfera alvo. No Caso C1 o cálculo da energia depositada é requisitado em apenas dois *voxels* da esfera alvo. Este procedimento refina os dados de saída e o uso desta opção vai depender da necessidade do usuário por maiores níveis de detalhamento. Outro fator a ser considerado é o tempo de simulação, pois ele é diretamente proporcional a quantidade de *voxels*.

O primeiro conjunto de casos, por ser simples, poderia ser criado manualmente usando as ferramentas de construção de geometria repetida do código MCNP. Fazendo isso é possível verificar se o resultado da simulação é compatível ao criado pelo SCMS ou pelo AMIGO 1.0, ou seja, outra forma de validar o software.

3.3.2 Casos testes: A2, B2 e C2

O próximo conjunto de casos é um pouco mais complexo que o anterior, visto que a quantidade de materiais e objetos é maior. Estas informações podem ser observadas na Tabela 3.

A geometria destes casos é composta de elipsóides, veja Figura 48, e diferentemente do grupo anterior, neste conjunto de casos testes é muito trabalhoso criar a geometria manualmente usando as ferramentas de construção do MCNP5.

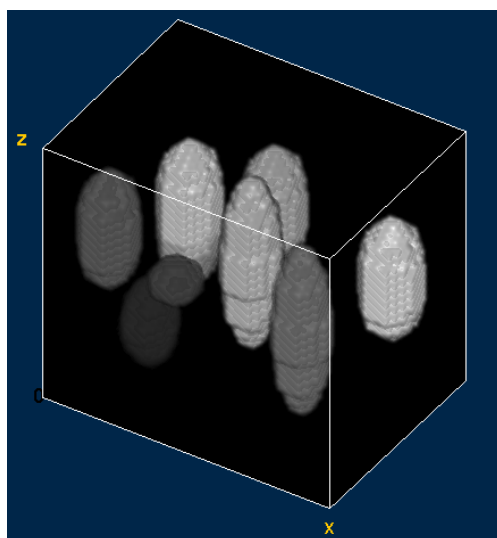


Figura 48 - Construção tridimensional dos objetos simuladores correspondentes aos casos A2, B2 e C2.

Da mesma forma que o caso anterior, a fonte de radiação se localizada em uma ou mais elipsóides e a energia depositada é calculada em outras elipsóides, visto que a quantidade de fontes de radiação varia. No Caso A2, por exemplo, apenas uma das elipsóides é a fonte, nos outros dois casos, Caso B2 e Caso C2, a fonte está distribuída em mais de um elipsóide.

O número de objetos onde a energia depositada será calculada varia de 3, para o Caso A2, a 5 para os outros casos. Por fim, dependendo do caso a distribuição da fonte se altera, podendo ser, heterogênea ou homogênea.

Os próximos casos testes são exibidos na Tabela 4 e são similares a tabela anterior, mas observa-se que alguns objetos simuladores possuem dimensões maiores.

Tabela 4 - Casos de *benchmark* do software

	CasoA3	CasoA4	CasoB4	CasoC4	CasoA5	CasoB5
Eixo X	80	128	128	128	158	158
Eixo Y	60	128	128	128	74	74
Primeira fatia	1	1	1	1	1	1
Última fatia	63	237	237	237	487	487
Número de órgãos fonte	2	1	1	1	1	1
Partícula da fonte	Fótons	Fótons	Fótons	Fótons	Fótons	Fótons
Energia da fonte (MeV)	1,0	1,0	1,0	1,0	1,0	1,0
Órgãos que contém a fonte	3 5	28	28	28	28	28
Quantidade de órgãos no objeto simulador	6	125	125	125	551	551
Número mínimo de voxel em cada órgão	1	1	1	1	1	1
Comprimento do voxel para o eixo X (cm)	0,4	0,4	0,4	0,4	0,36	0,36
Comprimento do voxel para o eixo Y (cm)	0,4	0,4	0,4	0,4	0,36	0,36
Comprimento do voxel para o eixo Z (cm)	0,4	0,4	0,4	0,4	0,36	0,36
Distribuição da fonte	Heterogênea	Homogênea	Heterogênea	Heterogênea	Homogênea	Heterogênea
Quantidade de órgãos/tecidos com material 1	4	104	104	104	414	414
Quantidade de órgãos/tecidos com material 2	1	1	1	1	1	1
Quantidade de órgãos/tecidos com material 3	1	20	20	20	136	136
Densidade do material 1 (grama/cm ³)	1,04	1,04	1,04	1,04	1,05	1,05
Densidade do material 2 (grama/cm ³)	0,296	0,296	0,296	0,296	0,26	0,26
Densidade do material 3 (grama/cm ³)	1,4	1,4	1,4	1,4	1,469	1,469
Quantidade de órgãos para calcular a energia depositada	1	1	1	3	1	3
Órgãos onde a energia depositada será calculada	2	11	11	10 12 28	28	10 12 28

3.3.3 Caso teste A3

O Caso A3 é similar aos dois últimos conjuntos de casos, onde o objeto simulador é composto por elipsóides. Este objeto simulador só possui um caso teste como é mostrada na Tabela 4. A representação tridimensional deste objeto simulador é mostrada na Figura 49.

Este caso é mais semelhante aos casos do grupo 1, pois sua geometria é simples e possui poucos tecidos, além disso, a energia depositada só é calculada em uma região ou órgão.

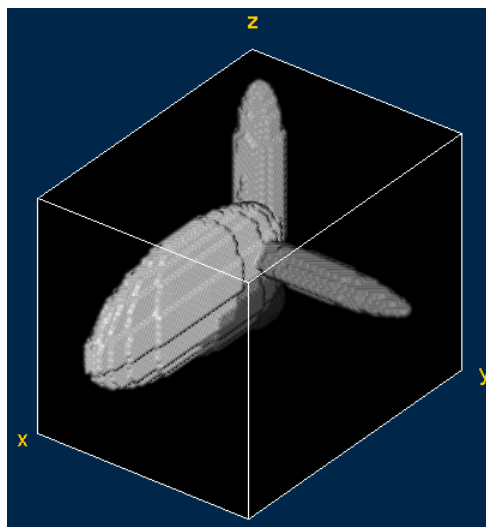


Figura 49 - Construção tridimensional do objeto simulador do Caso A3

3.3.4 Casos testes: A4, B4 e C4

Esse grupo de testes é o primeiro a usar um objeto simulador similar ao modelo anatômico de um corpo humano. Ele possui dimensões um pouco maiores que os casos anteriores, sobretudo no eixo Z, que representa as fatias do objeto simulador.

A maior diferença deste objeto simulador para os anteriores é a quantidade de segmentação, pois é dividido em 125 órgãos ou tecidos diferentes, o número é grande, pois representa o modelo de corpo humano próximo ao real. Na Figura 50 é mostrada a representação tridimensional, onde é possível distinguir órgãos internos e até mesmo os ossos.

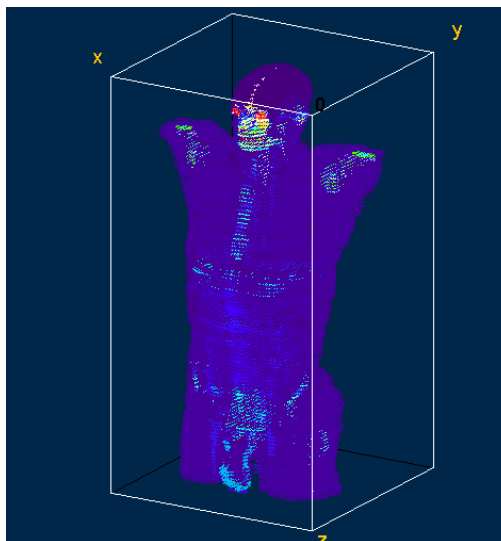


Figura 50 - Construção tridimensional do objeto simulador do quarto grupo de testes

O objeto simulador usado neste conjunto de casos é conhecido internacionalmente como “*phantom* de ZUBAL” (ZUBAL, 1994), porém algumas partes foram retiradas, tais como os membros superiores e inferiores, para aumentar o desempenho da simulação. Todos os órgãos neste modelo possuem um número de identificação. Em todos os casos do grupo 4 a fonte está localizada no órgão 28, que corresponde a tireóide. O procedimento de cálculo é similar em todos os três casos deste grupo, ou seja, um órgão é escolhido como fonte e a energia depositada é calculada em um ou mais órgãos próximos. Nos casos testes B4 e C4, por exemplo, foram modificados apenas os órgãos onde a energia depositada é calculada, já no caso A4 a distribuição da fonte é homogênea. No caso C4 a energia depositada também foi calculada no órgão onde a fonte se localiza.

3.3.5 Casos testes: A5 e B5

Os casos do grupo 5 são similares aos casos do grupo 4, pois ambos usam um objeto simulador humano, contudo, neste grupo, o objeto simulador, MAX5 (Kramer, 2003), é mais completo, possuindo membros inferiores e superiores e a quantidade de órgãos/tecidos é maior, num total de 551 tecidos.

Na Figura 51 é mostrada a representação tridimensional do objeto simulador usado nos dois testes do caso 5 onde nota-se a presença dos membros inferiores e superiores.

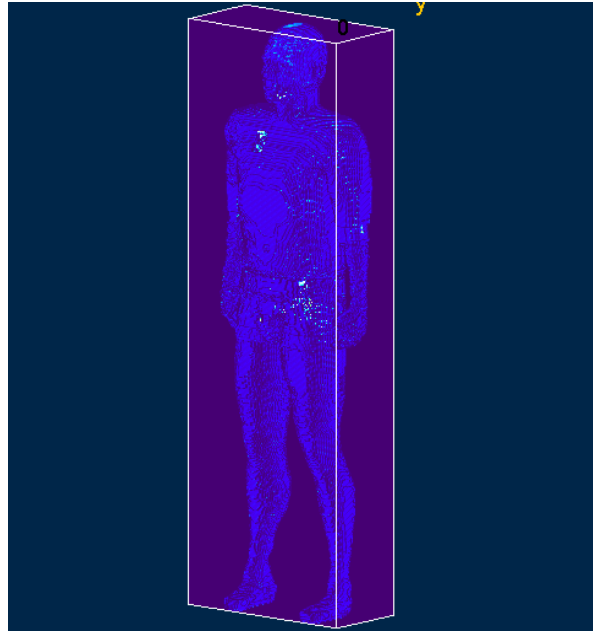


Figura 51 - Construção do objeto simulador MAX

3.3.6 Casos testes: A6 e B6

Os últimos 3 casos de benchmark são exibidos na Tabela 5. Observa-se que são semelhantes aos casos anteriores por utilizar um objeto simulador humano. Nos casos A6 e B6 foi usado o objeto simulador chamado FAX que simula um corpo feminino. (Kramer, 2004) A representação tridimensional deste objeto simulador pode ser vista na Figura 52.

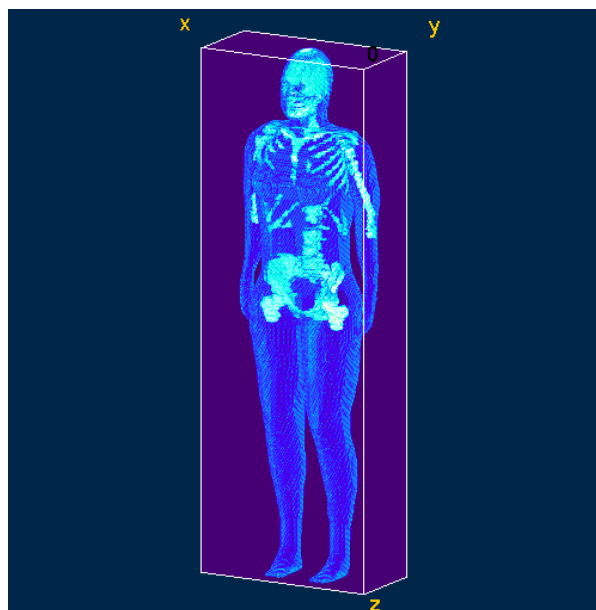


Figura 52 - Construção tridimensional do objeto simulador FAX

Tabela 5 - Casos de *benchmark*

	CasoA6	CasoB6	Caso7
Eixo X	158	158	128
Eixo Y	74	74	128
Primeira fatia	1	1	1
Última fatia	453	453	146
Número de órgãos fonte	1	1	1
Partícula da fonte	Fótons	Fótons	Fótons
Energia da fonte (MeV)	1.0	1.0	1.0
Órgãos que contém a fonte	27	27	63
Quantidade de órgãos no objeto simulador	551	551	125
Número mínimo de voxel em cada órgão	1	1	1
Comprimento do voxel para o eixo X (cm)	0,36	0,36	0,4
Comprimento do voxel para o eixo Y (cm)	0,36	0,36	0,4
Comprimento do voxel para o eixo Z (cm)	0,36	0,36	0,4
Distribuição da fonte	Homogênea	Heterogênea	Heterogênea
Quantidade de órgãos/tecidos com material 1	414	424	104
Quantidade de órgãos/tecidos com material 2	1	1	1
Quantidade de órgãos/tecidos com material 3	136	126	20
Densidade do material 1 (grama/cm ³)	1,05	1,05	1,04
Densidade do material 2 (grama/cm ³)	0,26	0,26	0,296
Densidade do material 3 (grama/cm ³)	1,469	1,469	1,4
Quantidade de órgãos para calcular a energia depositada	1	1	1
Órgãos onde a energia depositada será calculada	28	28	63

3.3.7 Caso teste 7

No caso teste 7 utiliza-se novamente o objeto simulador ZUBAL, que representa o modelo anatômico de um homem adulto. Como nos casos anteriores, um dos órgãos é definido como a fonte de radiação. Em particular, neste presente caso, o órgão fonte possui número de identificação 63, que corresponde a um tumor “artificial” localizado no fígado. O cálculo de dose, neste caso, é efetuado no próprio tumor. O objeto simulador é mostrado na Figura 53, onde nota-se que somente parte do objeto simulador é utilizada no cálculo da dose.

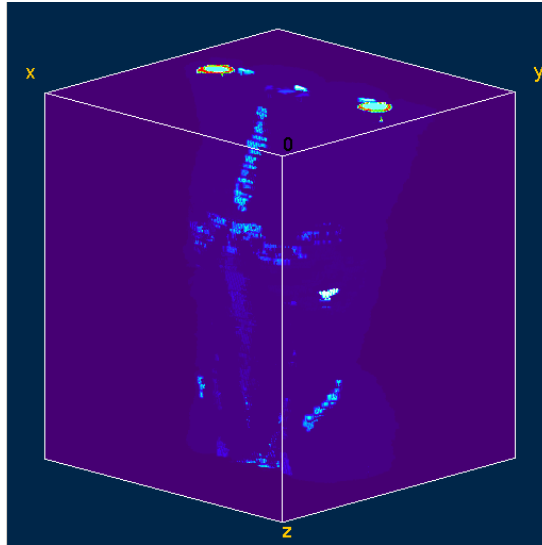


Figura 53 - Representação tridimensional do objeto simulador do caso teste 7

3.4 Diferenças nos arquivos de saída dos softwares

Neste item são discutidas as eventuais diferenças encontradas entre os arquivos de saída gerados pelos softwares SCMS, usado como referência, e os arquivos de saída gerados pelo software AMIGO 1.0. Esta nova versão do software foi inteiramente reescrita, sendo natural que exista eventuais diferenças entre os arquivos de saída, entretanto, essas diferenças não devem ser alterações que modifiquem as instruções de simulação e de cálculo de dose, ou seja, que provoquem alterações no resultado da simulação.

Comparando os arquivos de saída de todos os casos analisados e apresentados no item anterior, a principal diferença encontrada foi na formatação dos dados, isto é, o espaçamento entre os valores e no começo das linhas. Na Figura 54 são mostradas duas linhas de saída diferentes, a primeira linha corresponde à nova versão do software, AMIGO 1.0, enquanto a segunda a versão mais velha, SCMS. Estas linhas são do primeiro bloco de informações do MCNP, onde a geometria é definida. Notam-se diferenças de espaços no começo da linha e espaços entre o sinal de igual e o número dois. Estas diferenças, entretanto, não causam alterações na simulação do problema.

Arquivo 1, linha:	3	1	-1.04	7	-8	9	-10	11	-12	u=2	imp:e=1	imp:p=1
Arquivo 2, linha:	3	1	-1.04	7	-8	9	-10	11	-12	u= 2	imp:e=1	imp:p=1

Figura 54 - Diferenças de espaçamento entre os valores nos arquivos de saída.

Na Figura 55 é mostrado diferenças similares às encontradas na figura acima, isto é, espaços entre o sinal de igual e o número dez, além dessa diferença, a densidade do material, representada pelo valor 0.296, é mostrado de formas diferentes. Novamente, essas alterações não alteram nem a execução nem a simulação do problema.

Arquivo 1, linha:	11 like 3 but mat=2 rho=-0.296	u=10
Arquivo 2, linha:	11 like 3 but mat=2 rho= -.296	u= 10

Figura 55 - Diferenças na exibição dos números

Todas as diferenças encontradas entre os arquivos foram mínimas e não provocam alterações no modo de simulação do código MCNP. Na Figura 56 é mostrada outra diferença sutil que é apenas um espaço em branco logo no início das linhas.

Arquivo 1, linha:	sp1 160 170 166 210 228 205 164 176 226 254 246 208 151 204 239 252 229 185
Arquivo 2, linha:	sp1 160 170 166 210 228 205 164 176 226 254 246 208 151 204 239 252 229 185

Figura 56 - Pequena diferença encontrada na descrição da fonte.

Para todos os casos, foram realizadas simulações com o código MCNP5 e os resultados de energia depositada comparadas, não havendo discrepâncias entre elas. Como exemplo de um arquivo típico resultante do processamento do software AMIGO está apresentado no Apêndice A, o arquivo de dados de entrada do MCNP gerado, correspondente ao caso A1.

3.5 Validação do software AMIGO 2.0

Como o AMIGO 2.0 é um software novo e implementado para carregar objetos simuladores de pacientes específicos, não necessariamente segmentados por órgãos, os casos anteriores usados como *benchmarks*, tiveram que ser adaptados para poderem ser utilizados na validação.

O caso 1 foi escolhido por ser mais simples e, portanto, mais fácil de alterar. As dimensões do objeto simulador foram mantidas como mostra a Tabela 6 abaixo, O caso 1 foi subdividido em 4 novos casos e seu nome alterado para caso 8.

Tabela 6 - Caso 8: *benchmarks* para validação do AMIGO 2.0

	CasoA8	CasoB8	CasoC8	CasoD8
Eixo X	45	45	45	45
Eixo Y	25	25	25	25
Primeira fatia	1	1	1	1
Última fatia	20	20	20	20
Número de órgãos fonte	1	1	1	1
Partícula da fonte	Fótons	Fótons	Fótons	Fótons
Energia da fonte (MeV)	1.0	1.0	1.0	1.0
Órgãos que contém a fonte	3	3	3	3
Quantidade de órgãos no objeto simulador	3	3	3	3
Número mínimo de voxel em cada órgão	1	1	1	1
Comprimento do voxel para o eixo X (cm)	0.4	0.4	0.4	0.4
Comprimento do voxel para o eixo Y (cm)	0.4	0.4	0.4	0.4
Comprimento do voxel para o eixo Z (cm)	0.4	0.4	0.4	0.4
Distribuição da fonte	Homogênea	Homogênea	Heterogênea	Homogênea
Quantidade de órgãos/tecidos com material 1	1	1	1	1
Quantidade de órgãos/tecidos com material 2	1	1	1	1
Quantidade de órgãos/tecidos com material 3	1	1	1	1
Densidade do material 1 (grama/cm ³)	1.04	1.04	1.04	1.04
Densidade do material 2 (grama/cm ³)	0.296	0.296	0.296	0.296
Densidade do material 3 (grama/cm ³)	1.4	1.4	1.4	1.4
Quantidade de órgãos para calcular a energia depositada	1	2	3	1
Órgãos onde a energia depositada será calculada	1	1	1	1

Os *benchmarks* do caso 8 compartilham do mesmo objeto simulador, mas o arquivo da fonte de radiação foi alterado em todos os quatro casos. No caso A8 a fonte de radiação se localiza na esfera maior, no caso B8 a fonte de radiação está nas duas esferas, contudo, na esfera menor ela tem o dobro da intensidade que na esfera maior. No caso C8 a fonte está na esfera maior, mas sua intensidade varia em toda a distribuição da esfera, simulando uma fonte de radiação heterogênea. Por fim, no caso D8 a fonte de radiação está localizada na esfera menor.

O AMIGO 2.0 é um software dependente do ICCT, pois este fornece todas as informações anatômicas (arranjo geométrico, mapa de densidades e mapa com identificação de materiais) a serem utilizadas na simulação. Estas informações são fornecidas em arquivos gerados após o ICCT processar as imagens. Desta forma, no presente processo de validação, estes arquivos foram criados manualmente.

No MCNP não existe universo com número de identificação “zero”, por isso no SCMS os materiais são incrementados em uma unidade e convertidos em universos. No AMIGO 2.0 o material correspondente ao material zero, ar, é convertido para o universo 24, diferentemente do SCMS e o AMIGO 1.0. Por causa disso, durante a comparação, os universos correspondentes ao material 24 foram substituídos pelo universo 1 para que as comparações entre as saídas do SCMS e o AMIGO 2.0 pudessem ser realizadas normalmente.

Nos casos de *benchmark* do AMIGO 1.0, não existiram diferenças nos arquivos gerados quando comparados às saídas do SCMS, ocorrendo apenas alterações na formatação das linhas, como números diferentes de espaços em branco. No AMIGO 2.0, devido às diferenças nos softwares também existem diferenças nos arquivos, mas que não afetam na execução do problema.

A primeira diferença está localizada na construção da geometria, pois o material correspondente ao ar sempre é convertido para o universo 24, os demais materiais são mantidos com o mesmo número quando transformados em universos. Portanto, se os valores de universos forem substituídos pelos mesmos números de universo do SCMS a geometria permanece a mesma.

A descrição da fonte de radiação no arquivo de todos os casos do grupo 8, com exceção do caso B8, permaneceu inalterada na comparação entre o SCMS e o AMIGO 2.0, visto que houveram apenas modificações na quantidade de espaços em branco. No caso B8, onde existem duas fontes de radiação, a ordem dos *voxels* referentes às duas fontes está embaralhada (devido à forma de varredura). Isso se deve ao fato de que o SCMS, e conseqüentemente o AMIGO 1.0, construírem a fonte de radiação de forma diferente do AMIGO 2.0. A descrição da fonte de radiação é apresentada de forma distinta, mas quando ordenada *voxel* à *voxel* é comprovado que os dois softwares produzem a mesma descrição.

Em resumo, apesar de divergências de apresentação entre os dois arquivos, ambos produzem a mesma saída, quando simulados no código MCNP5.

4. CONCLUSÃO

A conversão do software SCMS para linguagem JAVA foi concluída e testada com êxito, não sendo encontradas diferenças relevantes que pudessem alterar o resultado final. Foram utilizados um total de 15 casos testes de *benchmark* propostos, cuja variação em extensão e complexidades geométricas cobrem uma vasta diversidade de situações, que podem ser encontradas em problemas de dosimetria em medicina nuclear.

Embora o desenvolvimento do software esteja encerrado, futuras modificações estão previstas para adicionar outras funcionalidades que aumentarão e facilitarão o uso do programa, deixando o mais simples e intuitivo. Além disso, está previsto sua integração com outros softwares atualmente em desenvolvimento e que faz parte de um projeto de desenvolvimento de um sistema de planejamento em medicina nuclear.

O desenvolvimento do AMIGO baseou-se no SCMS, por isso o método principal de execução do software ainda está implementado de forma estruturada, a divisão desta classe Java em um conjunto de novas classes utilizando orientação a objetos seria benéfica, pois facilitaria a manutenção da aplicação. Atualmente, o AMIGO possui uma nova interface gráfica, controle de erros quando os dados do software forem inseridos incorretamente, adição de espectro de energia, possibilidade de carregar objetos simuladores de um paciente específico, interface para visualização da dose simulada e suporte a até 60 materiais.

Todas as atualizações propostas no objetivo deste trabalho foram desenvolvidas. Além disso, novas funcionalidades que vieram a ser necessárias durante o desenvolvimento do software também foram adicionados ao trabalho, como, por exemplo, o software de visualização da distribuição de dose, o *DoseView*. Este software também pode ser melhorado com a adição de novos tipos de coloração nas imagens e a inserção de imagens novas provenientes do cálculo da dose, como a imagem do erro relativo da simulação no MCNP5, por exemplo. As imagens exibidas pelo *DoseView* são bidimensionais, portanto seria interessante desenvolver um método de representação tridimensional para a exibição da dose, do objeto simulador e da fonte de radiação usadas na simulação.

O procedimento de escolha da região de interesse (ROI) para o cálculo da energia depositada, que posteriormente é convertida em dose, pode ser atualizado. A escolha da ROI é feita com base numa seleção retangular, porém outras formas poderiam ser desenvolvidas, como regiões esféricas, à mão livre e até mesmo algum algoritmo de

segmentação poderia ser implementado, a fim de dividir o objeto simulador de um paciente específico em órgãos. Atualmente, apenas uma região de interesse pode ser inserida em cada fatia do objeto simulador, mas em alguns casos o usuário tem interesse em dois pontos distintos para o cálculo da dose, então seria interessante a possibilidade de escolha de um ou mais regiões de interesse em cada fatia.

O AMIGO trabalha apenas com o código MCNP5, mas pode ser ampliado ou alterado com o objetivo de criar arquivos de entrada para outros softwares de transporte de radiação, como o Geant4 ou o Penelope. O software está restrito a tratamentos de radiação utilizando medicina nuclear, entretanto, novas funcionalidades que contemplem a braquiterapia e a radioterapia podem ser incorporados.

O AMIGO 2.0 trabalha em conjunto com os softwares ICCT e ACPS que também foram desenvolvidos em Java e usam as bibliotecas do ImageJ para exibição das imagens na tela. Neste sentido, futuramente estes 3 softwares serão integrados em apenas um software.

5. APÊNDICE A

Arquivo de saída do SCMS correspondente ao caso de *benchmark* A1:

Title

```

1 0 -1 2 -3 4 -5 6 fill=9999 imp:e=1 imp:p=1
2 0 -301 302 -303 304 -305 306 lat=1 u=9999 imp:e=1 imp:p=1
   fill=0: 44 0: 24 0: 19
2 461r 3 2r 2 40r 3 4r 2 38r 3 6r 2 37r 3 6r 2 37r 3 6r 2 38r 3 4r 2 40r 3
2r 2 760r 3 4r 2 38r 3 6r 2 36r 3 8r 2 34r 3 10r 2 33r 3 10r 2 33r 3 10r 2
33r 3 10r 2 33r 3 10r 2 34r 3 8r 2 36r 3 6r 2 38r 3 4r 2 624r 3 4r 2 37r 3
8r 2 34r 3 10r 2 32r 3 12r 2 31r 3 12r 2 11r 4 2r 2 16r 3 12r 2 10r 4 4r 2
15r 3 12r 2 10r 4 4r 2 15r 3 12r 2 10r 4 4r 2 15r 3 12r 2 11r 4 2r 2 16r 3
12r 2 32r 3 10r 2 34r 3 8r 2 37r 3 4r 2 534r 3 4r 2 37r 3 8r 2 34r 3 10r 2
32r 3 12r 2 10r 4 4r 2 14r 3 14r 2 8r 4 6r 2 13r 3 14r 2 7r 4 8r 2 12r 3
14r 2 7r 4 8r 2 12r 3 14r 2 7r 4 8r 2 12r 3 14r 2 7r 4 8r 2 12r 3 14r 2 7r
4 8r 2 12r 3 14r 2 8r 4 6r 2 14r 3 12r 2 10r 4 4r 2 16r 3 10r 2 34r 3 8r 2
37r 3 4r 2 443r 3 6r 2 36r 3 8r 2 33r 3 12r 2 31r 3 12r 2 10r 4 4r 2 14r 3
14r 2 7r 4 8r 2 11r 3 16r 2 6r 4 8r 2 11r 3 16r 2 5r 4 10r 2 10r 3 16r 2
5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 10r 3 16r 2 5r 4
10r 2 10r 3 16r 2 6r 4 8r 2 12r 3 14r 2 7r 4 8r 2 13r 3 12r 2 10r 4 4r 2
15r 3 12r 2 33r 3 8r 2 36r 3 6r 2 352r 3 6r 2 35r 3 10r 2 32r 3 12r 2 30r
3 14r 2 9r 4 4r 2 14r 3 14r 2 7r 4 8r 2 11r 3 16r 2 5r 4 10r 2 10r 3 16r 2
5r 4 10r 2 9r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r
2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 9r 3 16r 2 5r 4 10r 2 10r 3
16r 2 5r 4 10r 2 11r 3 14r 2 7r 4 8r 2 12r 3 14r 2 9r 4 4r 2 15r 3 12r 2
32r 3 10r 2 35r 3 6r 2 307r 3 6r 2 35r 3 10r 2 32r 3 12r 2 10r 4 4r 2 14r
3 14r 2 7r 4 8r 2 11r 3 16r 2 5r 4 10r 2 10r 3 16r 2 4r 4 12r 2 8r 3 18r 2
3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r
2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 9r 3
16r 2 4r 4 12r 2 9r 3 16r 2 5r 4 10r 2 11r 3 14r 2 7r 4 8r 2 13r 3 12r 2
10r 4 4r 2 16r 3 10r 2 35r 3 6r 2 307r 3 6r 2 35r 3 10r 2 32r 3 12r 2 10r
4 4r 2 14r 3 14r 2 7r 4 8r 2 11r 3 16r 2 5r 4 10r 2 10r 3 16r 2 4r 4 12r 2
8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r
2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 3r 4
12r 2 9r 3 16r 2 4r 4 12r 2 9r 3 16r 2 5r 4 10r 2 11r 3 14r 2 7r 4 8r 2
13r 3 12r 2 10r 4 4r 2 16r 3 10r 2 35r 3 6r 2 262r 3 6r 2 35r 3 10r 2 32r
3 12r 2 30r 3 14r 2 9r 4 4r 2 13r 3 16r 2 6r 4 8r 2 10r 3 18r 2 4r 4 10r 2
9r 3 18r 2 3r 4 12r 2 8r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 6r 3 20r
2 1r 4 14r 2 6r 3 20r 2 1r 4 14r 2 6r 3 20r 2 1r 4 14r 2 7r 3 18r 2 2r 4
14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 3r 4 12r 2 8r 3 18r 2 4r 4 10r 2
10r 3 16r 2 6r 4 8r 2 12r 3 14r 2 9r 4 4r 2 15r 3 12r 2 32r 3 10r 2 35r 3
6r 2 217r 3 6r 2 35r 3 10r 2 32r 3 12r 2 30r 3 14r 2 9r 4 4r 2 13r 3 16r 2
6r 4 8r 2 10r 3 18r 2 4r 4 10r 2 9r 3 18r 2 3r 4 12r 2 8r 3 18r 2 2r 4 14r
2 7r 3 18r 2 2r 4 14r 2 6r 3 20r 2 1r 4 14r 2 6r 3 20r 2 1r 4 14r 2 6r 3
20r 2 1r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 3r
4 12r 2 8r 3 18r 2 4r 4 10r 2 10r 3 16r 2 6r 4 8r 2 12r 3 14r 2 9r 4 4r 2
15r 3 12r 2 32r 3 10r 2 35r 3 6r 2 262r 3 6r 2 35r 3 10r 2 32r 3 12r 2 10r
4 4r 2 14r 3 14r 2 7r 4 8r 2 11r 3 16r 2 5r 4 10r 2 10r 3 16r 2 4r 4 12r 2

```

```

8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r
2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 3r 4
12r 2 9r 3 16r 2 4r 4 12r 2 9r 3 16r 2 5r 4 10r 2 11r 3 14r 2 7r 4 8r 2
13r 3 12r 2 10r 4 4r 2 16r 3 10r 2 35r 3 6r 2 307r 3 6r 2 35r 3 10r 2 32r
3 12r 2 10r 4 4r 2 14r 3 14r 2 7r 4 8r 2 11r 3 16r 2 5r 4 10r 2 10r 3 16r
2 4r 4 12r 2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 2r 4
14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 3r 4 12r 2 8r
3 18r 2 3r 4 12r 2 9r 3 16r 2 4r 4 12r 2 9r 3 16r 2 5r 4 10r 2 11r 3 14r 2
7r 4 8r 2 13r 3 12r 2 10r 4 4r 2 16r 3 10r 2 35r 3 6r 2 307r 3 6r 2 35r 3
10r 2 32r 3 12r 2 30r 3 14r 2 9r 4 4r 2 14r 3 14r 2 7r 4 8r 2 11r 3 16r 2
5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 9r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4
12r 2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 9r
3 16r 2 5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 11r 3 14r 2 7r 4 8r 2 12r 3 14r
2 9r 4 4r 2 15r 3 12r 2 32r 3 10r 2 35r 3 6r 2 352r 3 6r 2 36r 3 8r 2 33r
3 12r 2 31r 3 12r 2 10r 4 4r 2 14r 3 14r 2 7r 4 8r 2 11r 3 16r 2 6r 4 8r 2
11r 3 16r 2 5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 10r 3
16r 2 5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 10r 3 16r 2 6r 4 8r 2 12r 3 14r 2
7r 4 8r 2 13r 3 12r 2 10r 4 4r 2 15r 3 12r 2 33r 3 8r 2 36r 3 6r 2 443r 3
4r 2 37r 3 8r 2 34r 3 10r 2 32r 3 12r 2 10r 4 4r 2 14r 3 14r 2 8r 4 6r 2
13r 3 14r 2 7r 4 8r 2 12r 3 14r 2 7r 4 8r 2 12r 3 14r 2 7r 4 8r 2 12r 3
14r 2 7r 4 8r 2 12r 3 14r 2 7r 4 8r 2 12r 3 14r 2 8r 4 6r 2 14r 3 12r 2
10r 4 4r 2 16r 3 10r 2 34r 3 8r 2 37r 3 4r 2 534r 3 4r 2 37r 3 8r 2 34r 3
10r 2 32r 3 12r 2 31r 3 12r 2 11r 4 2r 2 16r 3 12r 2 10r 4 4r 2 15r 3 12r
2 10r 4 4r 2 15r 3 12r 2 10r 4 4r 2 15r 3 12r 2 11r 4 2r 2 16r 3 12r 2 32r
3 10r 2 34r 3 8r 2 37r 3 4r 2 624r 3 4r 2 38r 3 6r 2 36r 3 8r 2 34r 3 10r
2 33r 3 10r 2 33r 3 10r 2 33r 3 10r 2 33r 3 10r 2 34r 3 8r 2 36r 3 6r 2
38r 3 4r 2 760r 3 2r 2 40r 3 4r 2 38r 3 6r 2 37r 3 6r 2 37r 3 6r 2 38r 3
4r 2 40r 3 2r 2 2639r
3 1 -1.04 7 -8 9 -10 11 -12 u= 2 imp:e=1 imp:p=1
4 like 3 but mat=3 rho= -1.400 u= 3
5 like 3 but mat=2 rho= -.296 u= 4
6 like 3 but u= 1
7 0 #1 imp:e=0 imp:p=0

```

C Cell Surfaces

```

7 px .0
8 px .400000
9 py .0
10 py .400000
11 pz .0
12 pz .400000
c lattices side
301 px .400000
302 px .0
303 py .400000
304 py .0
305 pz .400000
306 pz .0
c size of the array
1 px 18.000000

```

```

2 px 0.0
3 py 10.000000
4 py 0.0
5 pz 8.000000
6 pz 0.0

mode p e
sdef erg= 1.00 cel=d1 x d2 y d3 z d4 par=2
si1 l 1:2(32 11 2):5 1:2(33 11 2):5 1:2(34 11 2):5 1:2(31 12 2):5
    1:2(32 12 2):5 1:2(33 12 2):5 1:2(34 12 2):5 1:2(35 12 2):5
    1:2(31 13 2):5 1:2(32 13 2):5 1:2(33 13 2):5 1:2(34 13 2):5
    1:2(35 13 2):5 1:2(31 14 2):5 1:2(32 14 2):5 1:2(33 14 2):5
    1:2(34 14 2):5 1:2(35 14 2):5 1:2(32 15 2):5 1:2(33 15 2):5
    1:2(34 15 2):5 1:2(31 9 3):5 1:2(32 9 3):5 1:2(33 9 3):5
    1:2(34 9 3):5 1:2(35 9 3):5 1:2(30 10 3):5 1:2(31 10 3):5

...

    1:2(35 12 15):5 1:2(31 13 15):5 1:2(32 13 15):5
    1:2(33 13 15):5 1:2(34 13 15):5 1:2(35 13 15):5
    1:2(31 14 15):5 1:2(32 14 15):5 1:2(33 14 15):5
    1:2(34 14 15):5 1:2(35 14 15):5 1:2(32 15 15):5
    1:2(33 15 15):5 1:2(34 15 15):5

sp1 l 1717r
si2 h .0 .4000000
sp2 d 0. 1.
si3 h .0 .4000000
sp3 d 0. 1.
si4 h .0 .4000000
sp4 d 0. 1.
*f18:p,e (3<2)
*f28:p,e (4<2)
*f38:p,e (5<2)
c material composition of soft tissue
m1 1000. -0.10454 6000. -0.22663 7000. -0.02490
    8000. -0.63525 11000. -0.00112 15000. -0.00134
    16000. -0.00204 17000. -0.00133 19000. -0.00208
c material composition of lung tissue
m2 1000. -0.10134 6000. -0.10238 7000. -0.02866
    8000. -0.75752 11000. -0.00184 16000. -0.00225
    17000. -0.00266 19000. -0.00194
c material composition of bone tissue
m3 1000. -0.07337 6000. -0.25475 7000. -0.03057
    8000. -0.47893 11000. -0.00326 12000. -0.00112
    15000. -0.05095 16000. -0.00173 17000. -0.00143
    19000. -0.00153 20000. -0.10190

nps 600000
prtmp j 200000 j j
print 30 110

```


Caso A1 gerado pelo AMIGO 1.0:

IPEN SCMS2

```

1 0 -1 2 -3 4 -5 6          fill=9999  imp:e=1 imp:p=1
2 0 -301 302 -303 304 -305 306  lat=1 u=9999  imp:e=1 imp:p=1
   fill=0:44 0:24 0:19
2 461r 3 2r 2 40r 3 4r 2 38r 3 6r 2 37r 3 6r 2 37r 3 6r 2 38r 3 4r 2 40r 3
2r 2 760r 3 4r 2 38r 3 6r 2 36r 3 8r 2 34r 3 10r 2 33r 3 10r 2 33r 3 10r 2
33r 3 10r 2 33r 3 10r 2 34r 3 8r 2 36r 3 6r 2 38r 3 4r 2 624r 3 4r 2 37r 3
8r 2 34r 3 10r 2 32r 3 12r 2 31r 3 12r 2 11r 4 2r 2 16r 3 12r 2 10r 4 4r 2
15r 3 12r 2 10r 4 4r 2 15r 3 12r 2 10r 4 4r 2 15r 3 12r 2 11r 4 2r 2 16r 3
12r 2 32r 3 10r 2 34r 3 8r 2 37r 3 4r 2 534r 3 4r 2 37r 3 8r 2 34r 3 10r 2
32r 3 12r 2 10r 4 4r 2 14r 3 14r 2 8r 4 6r 2 13r 3 14r 2 7r 4 8r 2 12r 3
14r 2 7r 4 8r 2 12r 3 14r 2 7r 4 8r 2 12r 3 14r 2 7r 4 8r 2 12r 3 14r 2 7r
4 8r 2 12r 3 14r 2 8r 4 6r 2 14r 3 12r 2 10r 4 4r 2 16r 3 10r 2 34r 3 8r 2
37r 3 4r 2 443r 3 6r 2 36r 3 8r 2 33r 3 12r 2 31r 3 12r 2 10r 4 4r 2 14r 3
14r 2 7r 4 8r 2 11r 3 16r 2 6r 4 8r 2 11r 3 16r 2 5r 4 10r 2 10r 3 16r 2
5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 10r 3 16r 2 5r 4
10r 2 10r 3 16r 2 6r 4 8r 2 12r 3 14r 2 7r 4 8r 2 13r 3 12r 2 10r 4 4r 2
15r 3 12r 2 33r 3 8r 2 36r 3 6r 2 352r 3 6r 2 35r 3 10r 2 32r 3 12r 2 30r
3 14r 2 9r 4 4r 2 14r 3 14r 2 7r 4 8r 2 11r 3 16r 2 5r 4 10r 2 10r 3 16r 2
5r 4 10r 2 9r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r
2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 9r 3 16r 2 5r 4 10r 2 10r 3
16r 2 5r 4 10r 2 11r 3 14r 2 7r 4 8r 2 12r 3 14r 2 9r 4 4r 2 15r 3 12r 2
32r 3 10r 2 35r 3 6r 2 307r 3 6r 2 35r 3 10r 2 32r 3 12r 2 10r 4 4r 2 14r
3 14r 2 7r 4 8r 2 11r 3 16r 2 5r 4 10r 2 10r 3 16r 2 4r 4 12r 2 8r 3 18r 2
3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r
2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 9r 3
16r 2 4r 4 12r 2 9r 3 16r 2 5r 4 10r 2 11r 3 14r 2 7r 4 8r 2 13r 3 12r 2
10r 4 4r 2 16r 3 10r 2 35r 3 6r 2 307r 3 6r 2 35r 3 10r 2 32r 3 12r 2 10r
4 4r 2 14r 3 14r 2 7r 4 8r 2 11r 3 16r 2 5r 4 10r 2 10r 3 16r 2 4r 4 12r 2
8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r
2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 3r 4
12r 2 9r 3 16r 2 4r 4 12r 2 9r 3 16r 2 5r 4 10r 2 11r 3 14r 2 7r 4 8r 2
13r 3 12r 2 10r 4 4r 2 16r 3 10r 2 35r 3 6r 2 262r 3 6r 2 35r 3 10r 2 32r
3 12r 2 30r 3 14r 2 9r 4 4r 2 13r 3 16r 2 6r 4 8r 2 10r 3 18r 2 4r 4 10r 2
9r 3 18r 2 3r 4 12r 2 8r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 6r 3 20r
2 1r 4 14r 2 6r 3 20r 2 1r 4 14r 2 6r 3 20r 2 1r 4 14r 2 7r 3 18r 2 2r 4
14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 3r 4 12r 2 8r 3 18r 2 4r 4 10r 2
10r 3 16r 2 6r 4 8r 2 12r 3 14r 2 9r 4 4r 2 15r 3 12r 2 32r 3 10r 2 35r 3
6r 2 217r 3 6r 2 35r 3 10r 2 32r 3 12r 2 30r 3 14r 2 9r 4 4r 2 13r 3 16r 2
6r 4 8r 2 10r 3 18r 2 4r 4 10r 2 9r 3 18r 2 3r 4 12r 2 8r 3 18r 2 2r 4 14r
2 7r 3 18r 2 2r 4 14r 2 6r 3 20r 2 1r 4 14r 2 6r 3 20r 2 1r 4 14r 2 6r 3
20r 2 1r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 3r
4 12r 2 8r 3 18r 2 4r 4 10r 2 10r 3 16r 2 6r 4 8r 2 12r 3 14r 2 9r 4 4r 2
15r 3 12r 2 32r 3 10r 2 35r 3 6r 2 262r 3 6r 2 35r 3 10r 2 32r 3 12r 2 10r
4 4r 2 14r 3 14r 2 7r 4 8r 2 11r 3 16r 2 5r 4 10r 2 10r 3 16r 2 4r 4 12r 2
8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r
2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 3r 4

```

12r 2 9r 3 16r 2 4r 4 12r 2 9r 3 16r 2 5r 4 10r 2 11r 3 14r 2 7r 4 8r 2
 13r 3 12r 2 10r 4 4r 2 16r 3 10r 2 35r 3 6r 2 307r 3 6r 2 35r 3 10r 2 32r
 3 12r 2 10r 4 4r 2 14r 3 14r 2 7r 4 8r 2 11r 3 16r 2 5r 4 10r 2 10r 3 16r
 2 4r 4 12r 2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 2r 4
 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 2r 4 14r 2 7r 3 18r 2 3r 4 12r 2 8r
 3 18r 2 3r 4 12r 2 9r 3 16r 2 4r 4 12r 2 9r 3 16r 2 5r 4 10r 2 11r 3 14r 2
 7r 4 8r 2 13r 3 12r 2 10r 4 4r 2 16r 3 10r 2 35r 3 6r 2 307r 3 6r 2 35r 3
 10r 2 32r 3 12r 2 30r 3 14r 2 9r 4 4r 2 14r 3 14r 2 7r 4 8r 2 11r 3 16r 2
 5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 9r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4
 12r 2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 8r 3 18r 2 3r 4 12r 2 9r
 3 16r 2 5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 11r 3 14r 2 7r 4 8r 2 12r 3 14r
 2 9r 4 4r 2 15r 3 12r 2 32r 3 10r 2 35r 3 6r 2 352r 3 6r 2 36r 3 8r 2 33r
 3 12r 2 31r 3 12r 2 10r 4 4r 2 14r 3 14r 2 7r 4 8r 2 11r 3 16r 2 6r 4 8r 2
 11r 3 16r 2 5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 10r 3
 16r 2 5r 4 10r 2 10r 3 16r 2 5r 4 10r 2 10r 3 16r 2 6r 4 8r 2 12r 3 14r 2
 7r 4 8r 2 13r 3 12r 2 10r 4 4r 2 15r 3 12r 2 33r 3 8r 2 36r 3 6r 2 443r 3
 4r 2 37r 3 8r 2 34r 3 10r 2 32r 3 12r 2 10r 4 4r 2 14r 3 14r 2 8r 4 6r 2
 13r 3 14r 2 7r 4 8r 2 12r 3 14r 2 7r 4 8r 2 12r 3 14r 2 7r 4 8r 2 12r 3
 14r 2 7r 4 8r 2 12r 3 14r 2 7r 4 8r 2 12r 3 14r 2 8r 4 6r 2 14r 3 12r 2
 10r 4 4r 2 16r 3 10r 2 34r 3 8r 2 37r 3 4r 2 534r 3 4r 2 37r 3 8r 2 34r 3
 10r 2 32r 3 12r 2 31r 3 12r 2 11r 4 2r 2 16r 3 12r 2 10r 4 4r 2 15r 3 12r
 2 10r 4 4r 2 15r 3 12r 2 10r 4 4r 2 15r 3 12r 2 11r 4 2r 2 16r 3 12r 2 32r
 3 10r 2 34r 3 8r 2 37r 3 4r 2 624r 3 4r 2 38r 3 6r 2 36r 3 8r 2 34r 3 10r
 2 33r 3 10r 2 33r 3 10r 2 33r 3 10r 2 33r 3 10r 2 34r 3 8r 2 36r 3 6r 2
 38r 3 4r 2 760r 3 2r 2 40r 3 4r 2 38r 3 6r 2 37r 3 6r 2 37r 3 6r 2 38r 3
 4r 2 40r 3 2r 2 2639r
 3 1 -1.04 7 -8 9 -10 11 -12 u=2 imp:e=1 imp:p=1
 4 like 3 but mat=3 rho=-1.4 u=3
 5 like 3 but mat=2 rho=-0.296 u=4
 6 like 3 but u=1
 7 0 #1 imp:e=0 imp:p=0

c Cell Surfaces

7 px .0

8 px 0.4

9 py .0

10 py 0.4

11 pz .0

12 pz 0.4

c lattices side

301 px 0.4

302 px .0

303 py 0.4

304 py .0

305 pz 0.4

306 pz .0

c size of the array

1 px 18.0

2 px 0.0

3 py 10.0

```

4 py 0.0
5 pz 8.0
6 pz 0.0

mode p e
sdef erg=1.0 cel=d1 x d2 y d3 z d4 par=2
si1 l 1:2(32 11 2): 5 1:2(33 11 2):5 1:2(34 11 2):5 1:2(31 12 2):5
    1:2(32 12 2): 5 1:2(33 12 2):5 1:2(34 12 2):5 1:2(35 12 2):5
    1:2(31 13 2): 5 1:2(32 13 2):5 1:2(33 13 2):5 1:2(34 13 2):5
    1:2(35 13 2): 5 1:2(31 14 2):5 1:2(32 14 2):5 1:2(33 14 2):5
    1:2(34 14 2): 5 1:2(35 14 2):5 1:2(32 15 2):5 1:2(33 15 2):5
    1:2(34 15 2): 5 1:2(31 9 3):5 1:2(32 9 3):5 1:2(33 9 3):5
    1:2(34 9 3): 5 1:2(35 9 3):5 1:2(30 10 3):5 1:2(31 10 3):5

...

    1:2(35 12 15): 5 1:2(31 13 15):5 1:2(32 13 15):5
    1:2(33 13 15): 5 1:2(34 13 15):5 1:2(35 13 15):5
    1:2(31 14 15): 5 1:2(32 14 15):5 1:2(33 14 15):5
    1:2(34 14 15): 5 1:2(35 14 15):5 1:2(32 15 15):5
    1:2(33 15 15): 5 1:2(34 15 15):5

sp1 l 1717r
si2 h .0 0.4
sp2 d 0. 1.
si3 h .0 0.4
sp3 d 0. 1.
si4 h .0 0.4
sp4 d 0. 1.
*f18:p,e (3<2)
*f28:p,e (4<2)
*f38:p,e (5<2)
c material composition of soft tissue
m1 1000. -0.10454 6000. -0.22663 7000. -0.02490
    8000. -0.63525 11000. -0.00112 15000. -0.00134
    16000. -0.00204 17000. -0.00133 19000. -0.00208
c material composition of lung tissue
m2 1000. -0.10134 6000. -0.10238 7000. -0.02866
    8000. -0.75752 11000. -0.00184 16000. -0.00225
    17000. -0.00266 19000. -0.00194
c material composition of bone tissue
m3 1000. -0.07337 6000. -0.25475 7000. -0.03057
    8000. -0.47893 11000. -0.00326 12000. -0.00112
    15000. -0.05095 16000. -0.00173 17000. -0.00143
    19000. -0.00153 20000. -0.10190

nps 3e+7
prdmp j 5e+6 1 3
print 30 110

```

6. REFERÊNCIAS BIBLIOGRÁFICAS

(Abramoff et al, 2004) Abramoff, M.D.; Magelhaes, P.J.; Ram, S.J. "Image Processing with ImageJ". *Biophotonics International*, volume 11, issue 7, pp. 36-42, 2004.

(Anger, 1958) Anger, H. O. **Scintillation camera**. *Rev Sci Instrum*. 1958;29(1):27–33.

(Badger e Fisher, 1994) Badger, C.C.; Fisher, D.R. ***The Importance of Accurate Radiation Dosimetry in Radioimmunotherapy of Cancer***. Editorial. *The Journal of Nuclear Medicine*, Vol 35, No.2, Feb. 1994.

(Berg, 1999) Berg, C. ***Advanced Java 2 Development for Enterprise Application***, p 532-714, 1999.

(Beyer et al, 2000) Beyer, T.; Townsend, D. W.; Brun, T. **A combined PET/CT scanner for clinical oncology**. *J Nucl Med*. 2000; 41(8):1369–79.

(Briesmeister, 2001) Briesmeister, J. F. ***MCNP: A general Monte Cano N-Particle transport code, version 4C,***” Los Alamos National Laboratory Report. LA-13709-M, 2001.

(BRIESMEISTER, 2001) Briesmeister, J. F. ***MCNP: A general Monte Cano N-Particle transport code, version 4C***. Los Alamos National Laboratory Report. LA-13709-M, 2001.

(Campos, 2000) Campos, L. P. B. Cálculos dosimétricos em braquiterapia intravascular. Tese de Mestrado UFPe - Março de 2000.

(Cassen et al, 1951) Cassen, B.; Curtis, L.; Reed, C.; Libby, R. **Instrumentation for 131I use in medical studies**. *Nucleonics*. 1951;9(2):46–50.

(Cherry et al, 2003) Cherry, S.R.; Sorenson, J. A.; Phelps, M. E.; *Physics in Nuclear Medicine*, 3^a ed. , 523p., 2003, Saunders.

(De Hevesy, 1913) De Hevesy, G. **Radioelements as tracers in physics and chemistry**. Chem News. 1913;108:166. Lawrence EO, Livingstone MS. The production of high-speed light ions without the use of high voltage. Phys Rev. 1932; 40(1):19–30.

(Deitel, 2003) Deitel, P.J. **Java Como Programar**, p. 32-253, 2003.

(DeMarco, 1995) DeMarco, J. J. e colaboradores. **A verification of the Monte Carlo code MCNP of thick target bremsstrahlung calculations**. Med. Phys. v. 22, p. 11-16, 1995.

(ECLIPSE, 2011) *Eclipse - The Eclipse Foundation open source community website*. Disponível em: <www.eclipse.org> Acessado em: 24 ago. 2011.

(Evans, 2001) Evans, J. E. e outros **Absorbed dose estimates to structures of the brain and head using a high-resolution voxel-based head phantom**. Med. Phys., v. 28, p.780-786, 2001.

(Hammersley, 1964) Hammersley, J. M. and D. C. Handscomb. Monte Carlo Methods Editora: Wiley, 1ed, 1964.

(HEES, 2006) HEES H. 3D Computer Graphics, Número 1, p 350-351, Agosto 2006.

(James et al, 2000) James, G.; Bill, J.; Steele, G. **Java Language Specification**, p.47-512, 2000.

(Jonsson, L. e colaboradores, 2000) Jonsson, L. e colaboradores **The conjugate view method: evaluation of actin'it)' and absorbed dose calculations from Monte Carlo simulated scintillation camera images using experimental data in an anthropomorphic phantom**. J. Nucl. Med., v. 39. (Suppl). 234P, 2000.

(Kramer, 2003) Kramer, R. e colaboradores. **All about MAX: a male adult voxel phantom for Monte Carlo calculations in radiation protection dosimetry** Phys. Med. Biol. v. 48, p.1239-1262, 2003.

(Kramer, 2004) Kramer, R. e colaboradores. *All about FAX: a Female Adult voxel phantom for Monte Carlo calculation in radiation protection dosimetry* Phys. Med. Biol. v. 49, p.5203-5216, 2004.

(Li e Williamson, 1992) Li, Z.; Williamson, J.F. *Volume-based geometric modeling for radiation transport calculations*, Med. Phys. vol. 19, p.667-677, 1992.

(Mason, 1992) Mason, D. e colaboradores *Ytterbium-169: Calculated physical properties of a new radiation source for brachytherapy*. Med. Phys. v. 19, p.695-703, 1992.

(MASSICANO, 2010) Massicano, F, “Quantificação de imagens tomográficas para cálculo de dose em diagnose e terapia em medicina nuclear ”, IPEN, 2010. Dissertação (Mestrado) – Instituto de Pesquisas energéticas e Nucleares, São Paulo.

(Nelson et al, 1985) Nelson, W. R.; Hirayama, H.; Rogers, D. W. O. *The EGS-4 code system. Stanford Linear -Accelerator Center*. Report 265, 1985.

(NETBEANS, 2011) *Welcome to NetBeans*. Disponível em: <www.netbeans.org> Acessado em: 24 ago. 2011.

(Powsner & Powsner, 2006) Powsner, R. A.; Powsner, E. R. **Essential Nuclear Medicine Physics**, 2nd edition, Blackwell Publishers, 2006.

(RADAR, 2011) *Available Phantoms*. Disponível em: <<http://www.doseinfo-radar.com/RADARphan.html>> Acessado em: 23 jun. 2011.

(Robilotta, 2006) Robilotta, C. C. A tomografia por emissão de pósitrons: uma nova modalidade na medicina nuclear brasileira. Rev Panam Salud Publica. 2006; 20(2/3):134–42.

(ROGERS, 2006) Rogers, D. W. O. **Fifty years of Monte Carlo simulations for medical physics**. Volume 51, Number 13. 2006

(SALVAT et al, 2001) Salvat, F.; Fernandez-Varea, J. M.; Acosta, E.; Sempau, J. ***PENELOPE-A Code System for Monte Carlo Simulation of Electron and Photon Transport*** (Paris: OECD Publications). 2001.

(Sgouros, G. e colaboradores, 1990) Sgouros, G. e colaboradores. ***Treatment Planning for Internal Radionuclide Therapy: Three-Dosimensional Dosimetry for Nouniformly Distributed Radionuclides***. J. Nucl. Med. v. 31. p 1884-1891, 1990.

(Solberg et al, 2001) Solberg, T. D. e colaboradores. ***A review of radiation dosimetry applications using the MCNP Monte Carlo Code***. Radiochim. Acta, 89, p. 337-355, 2001.

(STABIN ET AL, 1995) M. G. Stabin, E. E. Watson, M. Cristy, J. C. Ryman, K. F. Eckerman, J. L. Davis, D. Marshall, and M. K. Gehlen, "Mathematical Models and Specific Absorbed Fractions of Photon Energy in the Nonpregnant Adult Female and at the End of Each Trimester of Pregnancy," Oak Ridge National Laboratory Report ORNL/TM-12907 (1995).

(Stabin, 2000) Stabin, M. G. e colaboradores ***Monte Carlo modeling of radiation dose distributions in intravascular radiation therapy***. Med. Phys. v. 27, p. 1086-1092, 2000.

(Stabin, 2002) Stabin, M. G. e colaboradores. ***Implementation of MCNP-Based radiation transport in radiotherapy treatment planning with patient-specific anatomic and physiologic data***. In: Brill A B. e Stabin M G. eds. Seventh International Radiopharmaceutical Dosimetry Symposium. Nashville, TN, Vanderbilt University, April 17-19, 2002.

(TIOBE, 2011) ***TIOBE Software: Tiobe Index***. Disponível em: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>> Acessado em: 24 ago. 2011.

(VISIBLEHUMAN, 2011). ***The National Library of Medicine's Visible Human Project***. Disponível em: <http://www.nlm.nih.gov/research/visible/visible_human.html> Acessado em: 23 jun. 2011.

(Wesley, 2000) Wesley, A. *Effective Java Programming Language Guide*, p.25-121, 2000.

(Yoriyaz & Stabin, 1998) Yoriyaz, H.; Stabin, M. *Use of MCNP 4B to calculate dose distributions in a voxelized phantom*. J. Nucl. Med. 39(5):83P, 1998.

(Yoriyaz, 2000) Yoriyaz, H. Desenvolvimento de uma metodologia computacional para cálculos em dosimetria interna. Tese de Doutorado - IPEN - Junho de 2000.

(Yoriyaz, 2000) Yoriyaz, H. e colaboradores *Absorbed fractions in a voxel-based phantom calculated with the MCNP-4Bcode*. Med. Phys., v. 27, p.1555-1562, 2000.

(Yoriyaz, 2001) Yoriyaz, H. e colaboradores. *Monte Carlo MCNP-4B-based absorbed dose distribution estimates for patient-specific dosimetry*. J. Nucl. Med., v. 42, p. 662-669, 2001.

(ZUBAL, 1994) Zubal, I.G., Harrell, C.R, Smith, E.O, Rattner, Z., Gindi, G. and Hoffer, P.B., Computerized three-dimensional segmented human anatomy, Medical Physics, 21(2):299-302, 1994.