



AUTARQUIA ASSOCIADA À UNIVERSIDADE DE SÃO PAULO

**DESENVOLVIMENTO DE UM PROGRAMA COMPUTACIONAL
PARA O TRATAMENTO DE SINAIS OBTIDOS PELA
RESSONÂNCIA PARAMAGNÉTICA ELETRÔNICA NA
DOSIMETRIA DE DOSES ALTAS**

ORLANDO RODRIGUES JÚNIOR

Tese apresentada como parte dos
requisitos para obtenção do Grau de
Doutor em Ciências na Área de
Tecnologia Nuclear - Aplicações.

Orientadora:
Dra. Letícia L.C. Rodrigues

São Paulo
2003

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES

Autarquia associada à Universidade de São Paulo

**DESENVOLVIMENTO DE UM PROGRAMA COMPUTACIONAL PARA O
TRATAMENTO DE SINAIS OBTIDOS PELA RESSONÂNCIA
PARAMAGNÉTICA ELETRÔNICA NA DOSIMETRIA DE DOSES ALTAS**

Orlando Rodrigues Júnior



Tese apresentada como parte dos requisitos para obtenção do Grau de Doutor em Ciências na Área de Tecnologia Nuclear - Aplicações

Orientadora:
Dra. Leticia L. C. Rodrigues

São Paulo
2003

À minha esposa Sandra
e aos meus filhos, Bruno e Priscila.

Agradecimentos

À Dra. Letícia Lucente Campos Rodrigues, pela orientação dedicada, confiança e incentivo.

Ao Instituto de Pesquisas Energéticas e Nucleares - IPEN, pela oportunidade oferecida para a realização desta pesquisa.

Ao Dr. Alberto Saburo Todo pela colaboração e amizade.

À Vanessa Rogéria de Lima pela colaboração na revisão das referências.

Ao amigo M.Sc. Ocimar Luiz Galante pelas discussões e pela ajuda constante durante as mais de 60 horas de medidas nos equipamentos de ressonância paramagnética eletrônica.

À Dra. Maria Teresa Lamy Freund e aos técnicos e colegas do Laboratório de Biofísica do Instituto de Física da Universidade de São Paulo pela oportunidade de realização das medidas.

Aos integrantes do Centro de Metrologia das Radiações pela amizade e estímulo.

Ao Dr. Cléber Nogueira de Souza pelo apoio e incentivo.

A todos que diretamente ou indiretamente colaboraram para a realização desta pesquisa.

Desenvolvimento de um Programa Computacional para o Tratamento de Sinais Obtidos pela Ressonância Paramagnética Eletrônica na Dosimetria de Doses Altas

Orlando Rodrigues Júnior

Resumo

Este trabalho apresenta o desenvolvimento de um sistema computacional para o tratamento matemático de espectros de Ressonância Paramagnética Eletrônica (RPE). O sistema foi desenvolvido para incorporar as atividades básicas de um laboratório de doses altas, incluindo a sua integração com um sistema de qualidade baseado na rastreabilidade das informações. O programa foi desenvolvido utilizando os conceitos da Programação Orientada a Objetos, permitindo o registro, o armazenamento e as análises das informações presentes no ciclo de atividades que compõem a dosimetria de doses altas pela técnica de RPE. Este ciclo compreende atividades desde o planejamento de uma dosimetria até a avaliação das doses em condições rotineiras ou em estimativas retrospectivas de dose em caso de acidente. Um novo método para o tratamento matemático de espectros brutos baseados em Filtros de *Wavelets* é proposto neste trabalho. A utilização deste método permite a extensão da técnica da alanina/RPE para doses baixas, aplicada à radioterapia, com incertezas totais menores que 5%.

Development of a Computational Program for Mathematical Treatment of Electron Paramagnetic Resonance Spectra in High-Dose Dosimetry.

Orlando Rodrigues Júnior

Abstract

This work presents the development of a computational system for the mathematical treatment of Electron Paramagnetic Resonance (EPR) spectrum. The system was developed to support the basic activities of high-doses dosimetry laboratory, including its integration with a quality system based on the traceability of the information. The program was developed using the concepts of the Object Oriented Programming, allowing the recording, the storage and the analysis of the information present in the activity cycle that composes the dosimetry by EPR technique. This cycle comprises activities involving the planning of dosimetry until the dose evaluation in routine conditions or the estimative of retrospective dose in cases of an accident. A new method for the treatment of raw spectra based in Wavelet Filters is proposed in this work. The use of this method allows the extension of the alanina/EPR technique for low doses, in radiotherapy range, with total uncertainty less than 5%.

Sumário

	Página
Resumo	i
Abstract	ii
1 Introdução	1
1.1 Visão Geral do Problema	1
1.2 Breve Retrospectiva Histórica da Dosimetria de Doses Altas	3
1.3 Objetivos Gerais e Específicos do Trabalho	5
1.4 Aspectos Originais do Trabalho	7
2 Bases Conceituais	8
2.1 A Ressonância Paramagnética Eletrônica	8
2.1.1 Origem do Sinal de RPE	11
2.1.2 Fator g e a Estrutura Hiperfina	15
2.1.3 Hamiltoniano de Spin	18
2.1.4 Espectrômetro de RPE	18
2.1.5 Formas do Espectro de RPE	21
2.2 A Dosimetria por RPE	22
2.2.1 Dosimetria com Alanina	23
2.2.2 Dosimetria de Acidente	28
2.2.3 Amostra Padrão	28
2.3 Problemas na Dosimetria por RPE	30
2.3.1 Doses Baixas	30
2.3.2 Retrospectiva de Dose	32
2.4 Tratamento Matemático dos Sinais	32

2.4.1 Suavização de Curvas	35
2.4.1.1 Filtro de Média Móvel	36
2.4.1.2 Filtro de Savitzky-Golay	37
2.4.2 Técnicas de Extração de Sinais em Espectros Brutos	39
2.4.2.1 Transformada de Fourier	40
2.4.2.2 Filtros de <i>Wavelets</i>	42
2.5 Estratégia para a Condução do Processo	48
3 Métodos Experimentais	50
3.1 Protocolos de Medida	50
3.2 Metodologias Adotadas Internacionalmente	51
3.3 Irradiações	54
3.4 Otimizando a Parametrização do Espectrômetro	56
3.5 Protocolo de Medida Proposto	61
3.5.1 Procedimentos Experimentais	62
3.5.2 Parametrização do Espectrômetro	63
3.5.3 Tratamento dos Espectros	65
4 Projeto Computacional	68
4.1 Modelo de Desenvolvimento de Software	68
4.2 Requisitos Básicos	71
4.3 Definição dos Processos do Laboratório	72
4.4 Concepção do Projeto Computacional	73
4.4.1 Levantamento dos Casos de Uso do Sistema	75
4.4.2 Elaboração dos Diagramas de Classes	77
4.4.3 Organização das rotinas numéricas	77

4.4.4 Detalhamento dos Processos	78
4.4.5 Implementação e Validação	80
5 Resultados e Discussão	86
5.1 Desempenho do Sistema	87
5.2 Parametrização dos Filtros de <i>Wavelets</i>	88
5.2.1 Extração de Ruídos de Alta Frequência	88
5.2.2 Desempenho dos Filtros de Wavelets	90
5.3 Extração Direta usando Wavelets	91
5.4 Identificando as Fontes de Erro no Pós-Processamento	92
5.5 Avaliação das Incertezas no Pós-Processamento	93
7 Conclusões	98
Apêndice - Programa DataEPR	100
Referências Bibliográficas	135

Capítulo 1

Introdução

1.1 Visão Geral do Problema

Nas últimas duas décadas, a dosimetria das radiações ionizantes passou por inúmeras transformações que envolveram o uso de tecnologias mais modernas, objetivando a melhoria da qualidade nos serviços. Essas tecnologias transformaram a pesquisa e o desenvolvimento de novos materiais. O computador teve um papel importante nessa transformação, automatizando processos e permitindo a construção de modelos mais próximos do mundo real. Os questionamentos científicos mudaram de enfoque. Antes, os problemas se concentravam em "o que se medir" e "onde se medir", e hoje as preocupações se concentram em "como medir de forma econômica, precisa e exata". Particularmente, na dosimetria de doses altas, algumas técnicas de medida tornaram-se padrões internacionais de rastreabilidade e controle de qualidade de serviços.

O Instituto de Pesquisas Energéticas e Nucleares - IPEN, dentro das suas atribuições, contribui diretamente no estabelecimento de metodologias e técnicas de dosimetria em diversos campos.

No Laboratório de Dosimetria de Doses Altas, a missão primordial é o desenvolvimento de técnicas dosimétricas para doses altas e o estabelecimento de metodologias para avaliação de doses externas tanto em situações rotineiras como em caso de acidente. As técnicas utilizadas envolvem métodos químicos baseados na espectrofotometria por absorção ótica de soluções e métodos físicos baseados na espectrometria por ressonância paramagnética eletrônica de pós. Esta última é o centro de interesse deste trabalho.

A Ressonância Paramagnética Eletrônica - RPE ou Ressonância de Spin Eletrônica - RSE é uma técnica espectroscópica avançada ideal para o estudo das propriedades magnéticas de certos materiais conhecidos como paramagnéticos.

Desde 1945, quando Zavoisky demonstrou a existência de um espectro de ressonância nas substâncias paramagnéticas, milhares de trabalhos passaram a aplicar a técnica de RPE em busca de um aprofundamento das propriedades magnéticas dos materiais⁽¹⁾. Esta técnica espectroscópica de investigação não-destrutiva de amostras paramagnéticas tem grande aplicabilidade em diversas áreas do estudo científico e tecnológico, incluindo datação arqueológica, identificação e quantificação de alterações nas estruturas cristalinas de substâncias e dosimetria das radiações ionizantes^(2,3).

Apesar da técnica de RPE ser considerada sofisticada e de custo relativamente elevado, graças aos excelentes resultados obtidos, o seu uso vem crescendo em instituições de pesquisa no mundo inteiro, principalmente no campo da dosimetria.

A aplicação na dosimetria compreende o intervalo das doses altas, 10^{-10} a 10^5 Gy¹, como na irradiação industrial, na retrospectiva de doses, na

¹ - gray: unidade especial de Dose Absorvida: 1 Gy = 1J/kg

dosimetria em caso de acidente e, mais recentemente, na dosimetria aplicada à radioterapia ⁽²⁻⁹⁾.

1.2 Breve Retrospectiva Histórica da Dosimetria de Doses Altas

A busca por bons materiais detectores da radiação ionizante é um dos objetivos permanentes de um laboratório de pesquisa aplicada na área da física das radiações. Em geral, métodos tradicionais de dosimetria são baseados nos efeitos que a radiação provoca em substâncias inorgânicas. Bons exemplos são encontrados nos dosímetros termoluminescentes, nas câmaras de ionização e também nos dosímetros químicos de Fricke ⁽¹⁾. No entanto, boa parte dos requisitos que um bom material detector da radiação deve possuir estão associados aos efeitos que a radiação pode provocar em tecidos animais.

Tentando vencer esse paradoxo, alguns materiais orgânicos começaram a ser investigados. Um dos efeitos biológicos da radiação, relevante e passível de quantificação com certa precisão, é a produção de radicais livres. Radicais livres em cristais orgânicos são produtos intermediários relativamente estáveis de uma seqüência de eventos e, como nos tecidos animais, estão associados com a absorção da radiação pelo material.

Bradshaw e colaboradores ⁽¹⁰⁾ propuseram em 1962 um novo método dosimétrico baseado no amino-ácido alanina e na ressonância paramagnética eletrônica como técnica de quantificação dos radicais livres formados. Esse estudo, bastante completo para época, investiga a sensibilidade da alanina para campos de radiação gama, elétrons e prótons, para diversas doses e condições de medida. A alanina era acondicionada em um tubo de polietileno com 1 cm de diâmetro e 2 a 4 cm de comprimento. Entretanto, por causa das dimensões do dosímetro não serem compatíveis com as da cavidade do espectrômetro utilizado,

era necessária a transferência da amostra para um outro recipiente com dimensões menores, a fim de se realizar uma medida. Em virtude deste problema, Bradshaw não mais utilizou o tubo de polietileno preenchido com alanina.

Regulla e Deffner, em 1982⁽⁵⁾, consolidaram os estudos na área com a apresentação de um método baseado no mesmo princípio. No trabalho, preocupações mais voltadas ao estabelecimento rotineiro e padronização da técnica são discutidas. A alanina é misturada com um aglomerante (parafina) para a obtenção de um detector com boas propriedades mecânicas. A interpretação do espectro da alanina irradiada, a construção de uma curva de calibração e o cálculo da dose e avaliações das incertezas no processo são alguns pontos que estão presentes nesse trabalho apontando para a padronização da técnica dosimétrica.

Além disso, a dosimetria passou a utilizar a RPE para investigar cenários acidentais envolvendo exposição à radiação⁽¹⁾. Conhecida como Retrospectiva de Dose, essa técnica é baseada na existência de materiais ou substâncias paramagnéticas no local do acidente que possam ser utilizados como dosímetros. A Retrospectiva de Dose é importante na investigação do acidente e no fornecimento de informações que possam ajudar no estabelecimento de procedimentos de Proteção Radiológica^(11,12).

Os bons resultados obtidos com dosímetros de alanina levaram muitos pesquisadores a investigarem novos materiais detectores, com formatos diferentes e utilizando encapsulamentos e outros aglomerantes para melhorar o desempenho mecânico e a resistência dos dosímetros.

Kojima e colaboradores⁽¹³⁾ apresentaram um detector baseado na alanina utilizando um polímero como aglomerante. Outros materiais com boas propriedades dosimétricas, como agarose gel, silicone, poli-vinil-cloreto (PVC), acetato de polivinila (PVA), também são citados na literatura^(1,14,15,16,17).

Uma iniciativa importante surgiu em 1985, quando a Agência Internacional de Energia Atômica, AIEA, coordenou um processo de intercomparação entre

diversos laboratórios espalhados pelo mundo para eleger o sistema dosimétrico que iria ser utilizado como sistema de referência num programa de intercomparação global chamado de *International Dose Assurance Service - IDAS* ⁽¹⁸⁾. O sistema dosimétrico escolhido foi o baseado na alanina/RPE. A técnica na alanina/RPE passou a ser considerada um padrão secundário para a dosimetria de doses altas.

A partir da década de 90, com a consolidação da técnica, os estudos se concentraram em melhorar o desempenho dos sistemas dosimétricos baseados na alanina/RPE, diminuindo o limite de detecção e permitindo o seu uso em aplicações no campo da radioterapia ^(6,9,18,19).

Em 1993, Bartolotta e colaboradores⁽¹⁹⁾ investigaram as fontes de incerteza no uso da alanina/RPE no intervalo de doses da radioterapia. Ainda em 1993, Nette e colaboradores⁽²⁰⁾, entre eles Regulla, apresentaram uma estratégia mundial coordenada pela AIEA para a criação de um serviço de intercomparação para a dosimetria em radioterapia baseado na técnica da alanina/RPE.

Nos últimos anos, os esforços da pesquisa em dosimetria se concentraram em otimizar procedimentos experimentais e no tratamento dos espectros de RPE. O tratamento matemático dos espectros brutos se tornou etapa importante no processo de avaliação da dose.

1.3 Objetivos Gerais e Específicos do Trabalho

Atualmente existem poucos programas computacionais disponíveis para o pós-processamento dos dados experimentais obtidos por RPE para materiais irradiados. Em geral, tais programas são voltados à aquisição de dados junto aos equipamentos de RPE e ao tratamento dos espectros em física básica, química e

biofísica, não atendendo às necessidades específicas da espectroscopia para fins dosimétricos.

Com o avanço tecnológico dos espectrômetros de RPE e a popularização do uso da técnica em dosimetria, a limitação dos resultados ficou associada aos problemas:

- Padronização do processo de fabricação dos dosímetros;
- Calibração do sistema dosimétrico;
- Otimização dos procedimentos experimentais;
- Melhor parametrização dos equipamentos de RPE;
- Interpretação dos espectros e das interferências no sinal para doses baixas, na ordem de *grays*, onde o sinal da amostra proveniente da irradiação é da mesma ordem de grandeza da composição de ruídos do sistema. Isso possibilitaria o uso da técnica em dosimetria para radioterapia.

Pensado nessas dificuldades foi elaborado um plano de trabalho para atender aos seguintes objetivos gerais:

- Estudo e desenvolvimento de técnicas matemáticas para a melhor interpretação dos sinais de RPE sob o ponto de vista de dosimetria;
- Criação de procedimentos operacionais otimizados para a realização das medidas;
- Desenvolvimento de um programa computacional para utilização no Laboratório de Dosimetria de Doses Altas do IPEN, tanto nas avaliações rotineiras como no caso de acidente.

1.4 Aspectos Originais do Trabalho

A proposta deste trabalho é o desenvolvimento de um programa computacional que consolide um conjunto de técnicas matemáticas e procedimentos operacionais, que possam auxiliar o pesquisador ou o técnico em dosimetria no processo da tomada de decisão em face aos resultados de uma análise espectroscópica para fins dosimétricos, tanto em condições normais, utilizando um dosímetro padrão de alanina ou de outra substância paramagnética, como em casos de análise retrospectiva de acidente.

O programa computacional foi desenvolvido na linguagem de programação Java, utilizando os conceitos do paradigma da Orientação a Objetos. Sua construção modularizada permite maior facilidade na manutenção e flexibilidade nas atualizações e inclusões.

Além disso, o programa foi concebido sob o prisma da rastreabilidade e confidencialidade dos dados, permitindo a compatibilidade com a filosofia do Sistema da Qualidade do IPEN. Estes esforços culminarão na construção de uma interface amigável, comercialmente viável, específica para a utilização em laboratórios de dosimetria de doses altas baseados na técnica espectroscópica de RPE.

No campo de tratamento matemático dos espectros brutos, o programa incorpora uma técnica inédita para a extração de sinais de RPE baseada em Filtros de *Wavelets*.

Capítulo 2

Bases Conceituais

2.1 A Ressonância Paramagnética Eletrônica

Os princípios da RPE envolvem os fundamentos da física clássica e aspectos da física quântica. Como o foco deste trabalho está no campo da Dosimetria das Radiações Ionizantes, são apresentados apenas os conceitos fundamentais para o entendimento do processo.

O spin do elétron foi descoberto em 1922 pelos cientistas Stern e Gerlach em experimentos com feixes de átomos de prata que se separavam em duas componentes ao passarem por um campo magnético não homogêneo. Esse resultado experimental trouxe uma dificuldade para a antiga teoria quântica, indicando a existência de um momento angular semi-inteiro⁽²¹⁾.

Em 1925, Goudsmit e Uhlenbeck formularam uma resposta para a estrutura fina do espectro atômico, sugerindo que a existência de um momento angular semi-inteiro fosse decorrente de um spin intrínseco do elétron⁽²¹⁾. Se o

momento angular intrínseco do spin for descrito por um número quântico s , pode-se esperar $2s+1$ valores possíveis para a componente z do momento angular intrínseco do elétron. Se m_s pode assumir somente dois valores, s deve ser $\frac{1}{2}$. Essa propriedade das partículas elementares é especificamente quântica e, por conseguinte, não admite uma interpretação clássica tradicional. No entanto, pode-se utilizar uma visão clássica para ilustrar o fenômeno.

Um elétron em órbita em volta de um núcleo dá origem a um momento angular orbital. Além disso, os elétrons são partículas carregadas e giram em torno do seu próprio eixo (momento angular intrínseco do elétron). O giro ou spin do elétron causa a movimentação de sua carga, provocando o surgimento do momento magnético de spin. A Figura 2.1 apresenta uma representação de um elétron orbitando em volta de um núcleo. ^(22,23,24)



Figura 2.1: Representação ilustrativa de um elétron orbitando ao redor de um núcleo.

Fazendo uma analogia macroscópica, cada elétron girando em torno de seu próprio eixo pode ser considerado equivalente a um pequeno ímã ou magneto.⁽¹⁾ A Figura 2.2 apresenta uma representação do modelo clássico do spin do elétron.

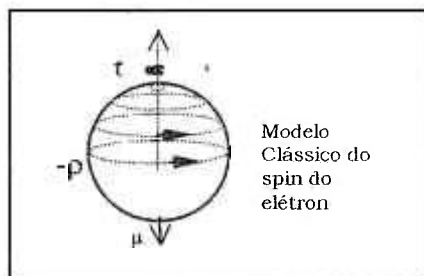


Figura 2.2: Ilustração do modelo clássico do Spin do elétron (momento angular intrínseco do elétron).

Em geral, esses pequenos magnetos estão dispostos aos pares (emparelhados) nos átomos e moléculas de uma substância, de tal forma que a contribuição líquida do momento magnético de spin é zero (Fig. 2.3a). A radiação natural ou artificial pode ionizar átomos ou moléculas, ou seja, separar elétrons emparelhados (Fig. 2.3b), gerando átomos ou moléculas com excesso ou falta de elétrons (desemparelhados). Nestas condições, o resultado líquido do momento magnético de spin é diferente de zero. Esses pequenos magnetos podem ser detectados e quantificados através da técnica espectroscópica de RPE, que consiste na medida da absorção de microondas pelo material sob o campo magnético externo. (Fig. 2.3c).

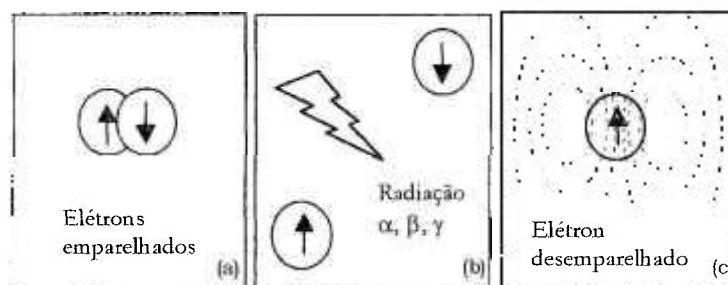


Figura 2.3: Formação de elétrons desemparelhados devido à interação com a radiação.

2.1.1 Origem do Sinal de RPE

Como já foi dito, materiais que apresentam um momento magnético de spin diferente de zero quando submetidos a um campo magnético são chamados de paramagnéticos. Esses pequenos magnetos, com direção e sentido, estão distribuídos aleatoriamente no interior da substância.

Quando elétrons desemparelhados são submetidos a um campo magnético externo adequado, a direção de cada magneto, inicialmente distribuída aleatoriamente no espaço (Fig. 2.4a), passa a se alinhar com o campo externo, gerando dois estados possíveis de energia, dependendo do sentido do alinhamento (Fig. 2.4b). Esse fenômeno é conhecido como Efeito Zeeman⁽²⁵⁾.

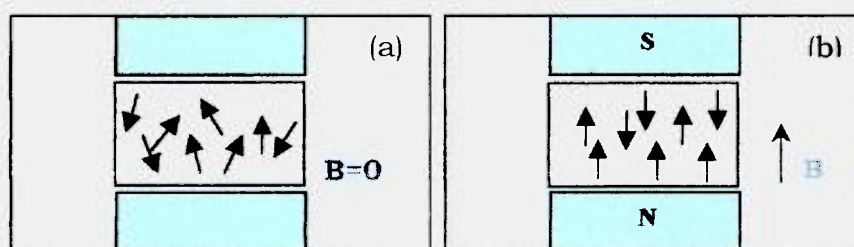


Figura 2.4: Orientação dos elétrons desemparelhados em substância paramagnética provocada por um campo magnético externo: (a) campo magnético externo $B=0$, (b) campo magnético externo $B \gg 0$.

O nível de energia mais baixo é mais populado que o nível de energia mais alto, e a técnica de RPE é uma forma de promover as transições eletrônicas do nível mais baixo E_1 , para o nível mais alto, E_2 , a partir da absorção de energia de microondas:

$$\Delta E = E_2 - E_1 = h\nu \quad (2.1)$$

onde:

h : é a constante de Planck ($6,63 \times 10^{-34}$ J.s); e

ν : é a frequência da radiação eletromagnética na região das microondas (s^{-1}).

A absorção de energia causa uma transição eletrônica do nível mais baixo de energia para o nível mais alto. A Figura 2.5 apresenta um diagrama simplificado da variação dos níveis de energia dos elétrons com o campo magnético externo.

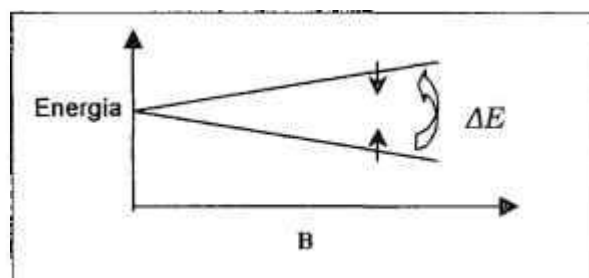


Figura 2.5: Transição de elétrons em níveis de energia em função do campo magnético externo.

O valor para a separação de energia $E_2 - E_1$ para um campo magnético B_0 é dado por^(23,25):

$$\frac{\Delta E}{h} = \frac{g\beta B_0}{h} \text{ (Hz)} \quad (2.2)$$

onde:

β : é o magneton de Bohr ($9,273 \times 10^{-24}$ J.K⁻¹);

g : é o fator de desdobramento espectroscópico ou fator de Landé e que é igual a 2,002319 para o elétron livre (átomo de hidrogênio);

B_0 : é o campo magnético (T); e,

h : é a constante de Planck ($6,63 \times 10^{-34}$ J.s).

Em uma espectroscopia convencional, o espectro de absorção de energia é obtido variando a frequência ν em função da absorção. Na RPE, devido a limitações da eletrônica associada às microondas, a frequência é mantida constante e varia-se o campo magnético. Um pico de absorção irá ocorrer quando o campo magnético atingir o valor necessário para promover a transição dos estados de energia. A posição das linhas na RPE é dada em termos dos valores de g . Na condição de ressonância, para um elétron livre, g assume o valor de 2,002319⁽²⁵⁾.

Uma questão relevante no estudo da condição de ressonância é como o processo de absorção de energia ocorre. Como foi visto, a atuação de um campo magnético externo provoca a formação de dois níveis de energia e que as transições de um nível para o outro podem ser induzidas com o fornecimento da energia necessária para essa transição.

Considerando um sistema de referências onde a direção do campo magnético externo estático coincide com a direção do eixo z, a aplicação de um campo de microondas de frequência fixa, com a componente do campo magnético perpendicular ao campo magnético estático, pode fornecer a energia necessária para mudar a direção dos *spins*. Essas transições podem ser tanto do estado superior para o inferior com a emissão de energia, como do estado inferior para o estado superior com absorção de energia.

No entanto, na condição de equilíbrio térmico, um número maior de elétrons estará no estado inferior de energia. Isso é uma consequência da Distribuição de Boltzmann dada por:

$$\frac{N_{\text{sup}}}{N_{\text{inf}}} = e^{-\frac{\Delta E}{kT}} \quad (2.3)$$

onde:

N_{sup} : é o número de spins no estado superior de energia;

N_{inf} : é o número de spins no estado inferior de energia;

k : é a constante de Boltzmann ($1,381 \times 10^{-23} \text{ J.K}^{-1}$); e,

T : é a temperatura (K).

Na condição de equilíbrio, a população nos dois estados é muito próxima, com pequena maioria no nível de menor energia. Ou seja, com o fornecimento de energia, seria esperada uma absorção líquida pequena, mas, diferente de zero até que as populações nos dois níveis fossem iguais e os processos de absorção e emissão fossem iguais. Nesta situação o sistema é dito saturado.

O equilíbrio só pode ser restabelecido se o sistema perder energia. Esse processo não é espontâneo e depende da interação dos *spins* com a rede cristalina na sua vizinhança, spin-rede, através de interações spin-spin. Os tempos que governam esses mecanismos de relaxação são chamados de: tempo de relaxação spin-rede T_1 e tempo de relaxação spin-spin T_2 ^(23,26).

2.1.2 Fator g e a Estrutura Hiperfina

Em geral, um espectro de ressonância paramagnética é bastante complexo, com linhas originadas de diferentes transições eletrônicas, que podem ser

subdivididas em subgrupos de linhas decorrentes das interações com o núcleo. Esse desdobramento é chamado de interação hiperfina, e a estrutura originária dessa interação é chamada de Estrutura Hiperfina⁽²³⁾. Além disso, existe a interação fina que ocorre em cristais com mais de um elétron desemparelhado causando uma degenerescência das linhas no espectro.

A interação hiperfina é a interação de um elétron desemparelhado com um núcleo com spin diferente de zero ($I > 0$) e pode acontecer de duas formas.

A primeira é chamada de interação anisotrópica e que pode ser interpretada como uma interação clássica de dois dipolos (spin-eletrônico e spin nuclear) separados por uma distância r e é dependente do ângulo formado pela linha que une os dois dipolos e o campo magnético externo.

A segunda interação hiperfina, chamada de interação isotrópica, decorre da probabilidade finita de se encontrar o elétron dentro do núcleo.

Os núcleos mais importantes em materiais orgânicos são H ($I = 1/2$), D ($I = 1$), ^{14}N ($I = 1$) e ^{13}C ($I = 1/2$). A informação sobre as linhas no espectro pode ser usada para o estudo da estrutura do radical livre. O número, a separação e as intensidades relativas destas linhas fornecem informações sobre o número de núcleos magnéticos, rotação dos radicais livres e sobre a força das interações hiperfinas.

A Figura 2.6 apresenta o diagrama da interação de 1 elétron desemparelhado com a estrutura hiperfina com um núcleo, $I=1/2$, de um sistema isotrópico num campo magnético externo B ⁽²⁴⁾.

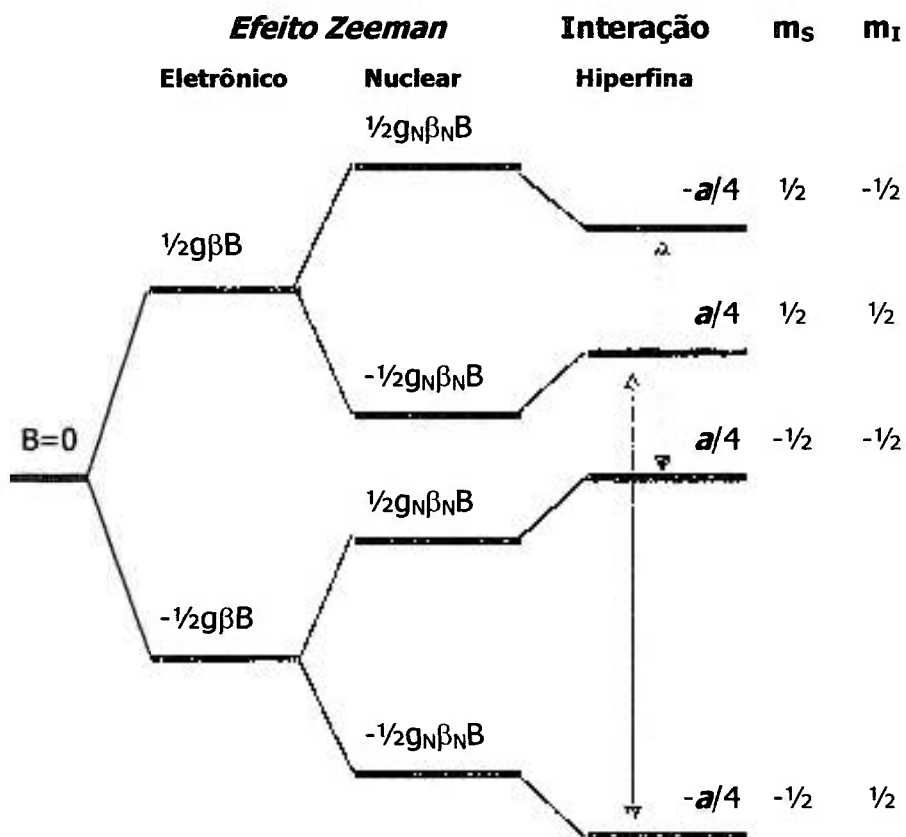


Figura 2.6: Diagrama dos níveis de energia para um elétron desemparelhado interagindo com um núcleo, $I=1/2$, com uma interação hiperfina isotrópica com constante a , assumindo $g_N\beta_N B > 1/2a$ ($g_N > 0$).

As linhas em azul na Figura 2.6 indicam as transições possíveis. Assumindo que S_z e I_z estão alinhados com o eixo z do campo magnético externo, a energia do sistema pode ser descrita como:

$$E = g\beta B m_s - g_N\beta_N B m_I + a m_s m_I \quad m_s = \pm 1/2 \text{ e } m_I = \pm 1/2 \quad (2.4)$$

Os dois primeiros termos são decorrentes do Efeito Zeeman, eletrônico e nuclear. O terceiro termo é proveniente da interação hiperfina. O termo para o Efeito Zeeman nuclear é muito pequeno comparado ao termo proveniente do Efeito Zeeman Eletrônico e em muitas aproximações pode ser desprezado.

As regras de seleção para as possíveis transições no caso isotrópico são: $\Delta m_S = \pm 1$ e $\Delta m_I = 0$. Observando o diagrama da Figura 2.6, apenas duas transições são possíveis e separadas pela distância a (em mT).

Um possível espectro para o esquema representado na Figura 2.6 teria dois picos de absorção separados pela distância a . No entanto os espectros de RPE são registrados na sua forma diferencial como mostra a Figura 2.7.

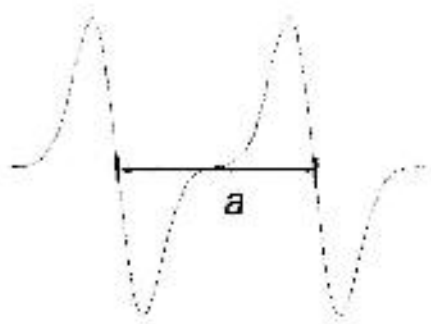


Figura 2.7: Sinal de RPE previsto para o esquema do diagrama da Figura 2.6.

Para n núcleos equivalentes, com a mesma constante de acoplamento a , têm-se $(2nI+1)$ linhas separadas pela distância a (mT). Neste exemplo, as interações consideradas eram isotrópicas, ou seja, os valores para g e para a eram independentes da direção do campo magnético externo. Em geral, para sólidos a anisotropia deve ser considerada.

2.1.3 Hamiltoniano de Spin

Em sistemas com muitos elétrons, com interações magnéticas entre os elétrons e entre elétrons e o campo magnético do núcleo, não é conveniente o uso direto da função de onda total para descrever o espectro observado. A frequência de oscilação, o tipo de estrutura cristalina e a orientação da substância no campo magnético externo são alguns parâmetros que influenciam a formação do espectro e dificultam a interpretação do mesmo.

Para facilitar o tratamento desses espectros foi criado o Hamiltoniano de Spin, que é uma forma simplificada de tratamento de estruturas mais complexas. Este operador atua nas partes da função de onda associadas ao *spin*, e contém constantes que podem ser obtidas pela integração sobre o espaço das coordenadas dos elétrons e seus *spins* individuais. O Hamiltoniano de Spin permite uma análise qualitativa e quantitativa dos fenômenos quânticos da interação dos radicais livres com o campo magnético.⁽²⁵⁾

Para o átomo de hidrogênio, pode-se descrever a degenerescência dos níveis de energia através do operador Hamiltoniano de Spin como:

$$H = g\beta(\hat{B} \cdot \hat{S}) - g_N\beta_N(\hat{B} \cdot \hat{I}) + A(\hat{I} \cdot \hat{S}) \quad (2.5)$$

2.1.4 Espectrômetro de RPE

Os espectrômetros de RPE são equipamentos complexos divididos basicamente em três sistemas: um sistema de geração do campo de radiação

magnética; uma cavidade para posicionamento de amostras e um sistema de geração de microondas.

Um conjunto de equipamentos eletrônicos associados permite o controle e a aquisição dos dados. As microondas geradas são conduzidas através de um guia de onda até a cavidade que é posicionada na região central entre os pólos do eletroímã.

Embora sejam mais comuns os espectrômetros operando na banda X, são encontrados comercialmente equipamentos operando em outras freqüências. A Tabela 2.1 apresenta os mais comuns.

Tabela 2.1: Designação dos espectrômetros de RPE por freqüência.

<i>Designação</i>	<i>Freqüência (GHz)</i>	<i>Campo Magnético (mT)</i>
S	3,0	107,0
X	9,5	339,0
K	23	820,0
Q	35	1.250,0

Em geral, para fins de dosimetria costuma-se utilizar espectrômetros operando na Banda-X. A Figura 2.8 apresenta um diagrama de blocos simplificado de um possível arranjo para um espectrômetro de RPE.

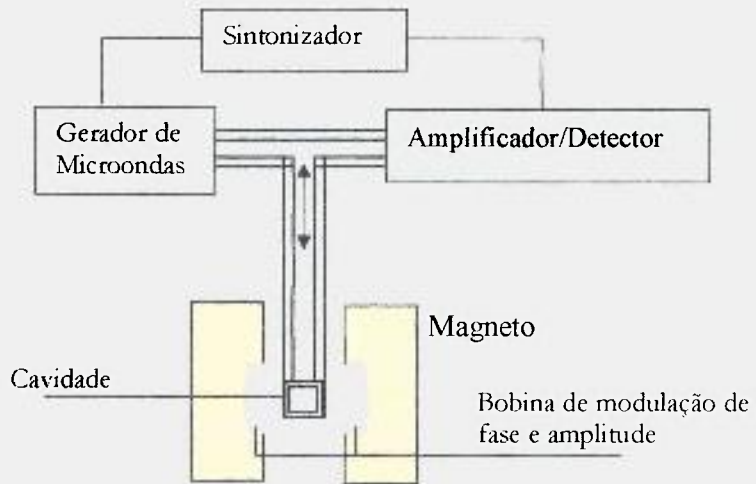


Figura 2.8: Diagrama de blocos de um espectrômetro de RPE^(23,24).

Em uma medida, posicionado o detector no centro da cavidade, ajustes de sintonia são efetuados até atingir-se a condição de ressonância. A absorção de energia é registrada através da mudança do fator de qualidade da cavidade.

Para minimizar o ruído do gerador de microondas, um campo magnético de baixa amplitude de modulação é aplicado junto ao eletroímã. A frequência de modulação desse campo pode variar de equipamento para equipamento.

O espectrômetro é um equipamento com alta sensibilidade, já que as quantidades medidas são muito pequenas. A potência de microondas e a amplitude de modulação são parâmetros que devem ser escolhidos com cautela para permitir medidas de boa qualidade.

2.1.5 Formas do Espectro de RPE

Diferentemente de outros espectros de absorção, onde se costuma avaliar picos e vales ou linhas de absorção, na espectrometria por RPE, o sinal obtido é a primeira derivada da curva de absorção. As vantagens de se utilizar o espectro derivado estão na melhoria da resolução e da relação sinal-ruído. Na Figura 2.9 são apresentados um típico espectro de absorção e a sua primeira derivada.

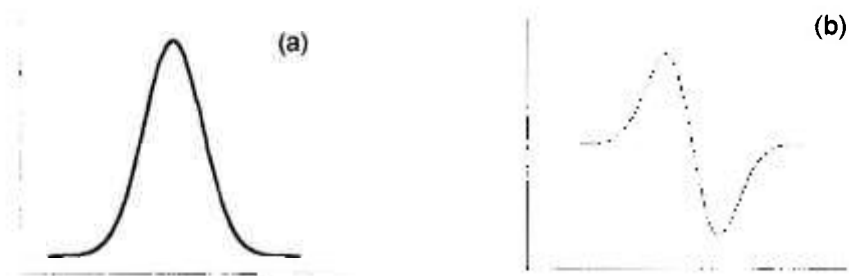


Figura 2.9: (a) Espectro de absorção característico com um pico; (b) Sinal de RPE característico – primeira derivada do espectro de absorção.

A função que descreve o formato do sinal de absorção depende das interações que ocorrem em cada substância. Em geral, distribuições do tipo Gaussiana ou Lorentziana são as utilizadas em programas de simulação de espectros⁽²⁴⁾.

Lorentziana:

$$I = \frac{I_0}{T_2^2 (H - H')^2 + 1} \quad (2.6)$$

Gaussiana:

$$I = I_0 e^{-b(H - H')^2 T_2^2} \quad (2.7)$$

onde:

I : é a intensidade do sinal;

I_0 : é a intensidade máxima do pico de absorção;

H : é o campo magnético variável;

H' : é o campo central ou na condição de ressonância;

T_2 : é o tempo de relaxação spin-spin;

b : é uma constante, usualmente igual a π .

2.2 Dosimetria por RPE

A RPE é uma técnica que vem sendo muito utilizada na dosimetria rotineira de doses altas de radiações gama, X, de nêutrons (indiretamente) e de elétrons, na dosimetria de acidentes e retrospectivas de dose, na datação arqueológica e geológica, na caracterização de materiais e análise de radicais⁽¹⁾.

Em geral, os sistemas dosimétricos europeus e americanos baseados na técnica da alanina/RPE utilizam dosímetro de alanina prensada com parafina com aproximadamente 230 mg (15% de parafina).

Esses dosímetros apresentam boas propriedades mecânicas e excelentes características dosimétricas, mas com uma componente de ruído, provocada pela

indução de radicais livres durante o processo de fabricação, que dificulta sua aplicação para doses abaixo de 10 Gy⁽⁵⁾.

No IPEN, foi desenvolvido um dosímetro de alanina de características distintas dos encontrados comercialmente^(16,27). A DL-alanina é encapsulada em uma fina camada cilíndrica de polietileno que garante uma resistência mecânica e proteção contra umidade e outros fatores externos adversos. O conjunto detector pode ser manufaturado sem prensagem em diversos tamanhos, com massas variando de 100 mg até 200 mg.

As vantagens estão associadas ao baixo custo de fabricação e às boas características dosimétricas.

Uma desvantagem é que para medidas abaixo de 10 Gy, o sinal nativo do tubo de polietileno pode interferir diretamente na medida, sendo necessário o uso de técnicas de pós-processamento dos espectros para a melhorar a avaliação da dose.

2.2.1 Dosimetria com Alanina

Na dosimetria rotineira, geralmente a alanina é utilizada como detector de radiação, apresentando boas características dosimétricas como estabilidade, reprodutibilidade, equivalência ao tecido humano e também baixo custo. Por causa da sua estrutura cristalina, a recombinação do radical é normalmente difícil, tornando a alanina um material estável para a maioria das condições de uso prático.⁽⁵⁾

A alanina é um aminoácido não essencial, alifático neutro com peso molecular de 89,1 g/mol, número atômico efetivo de 6,79 e fórmula molecular dada por $C_3 H_7 O_2$ ⁽¹⁶⁾.

As designações para as estruturas esteroisômeros estão apresentadas na Figura 2.10.

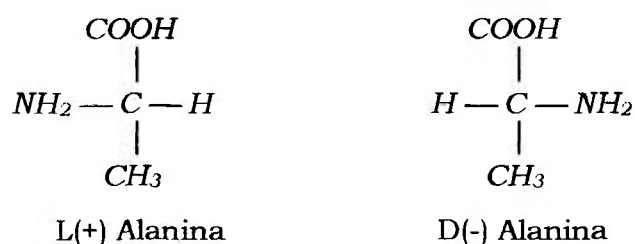
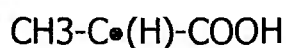


Figura 2.10: Representação dos esteroisômeros da alanina.

A DL-alanina, uma mistura dos dois esteroisômeros da alanina, apresenta bons resultados dosimétricos e é a utilizada pelo IPEN para o desenvolvimento dos detectores.

Pela ação da radiação ionizante, radicais livres são produzidos na alanina. A espécie paramagnética predominante à temperatura ambiente é o radical ⁽²⁸⁾:



A RPE é usada para determinar a concentração dos radicais livres, que é proporcional à dose absorvida pelo material para um grande intervalo de dose.

A dosimetria tradicional com alanina cobre um intervalo de $10\text{-}10^5$ Gy.

Para a realização de uma medida o detector é posicionado no centro da cavidade ressonante, utilizando um sistema de posicionamento que não interfira nas medidas.

Em geral, na dosimetria rotineira, um tubo de quartzo serve para o acondicionamento e posicionamento do detector dentro da cavidade. A Figura 2.11 apresenta um diagrama representativo da situação.

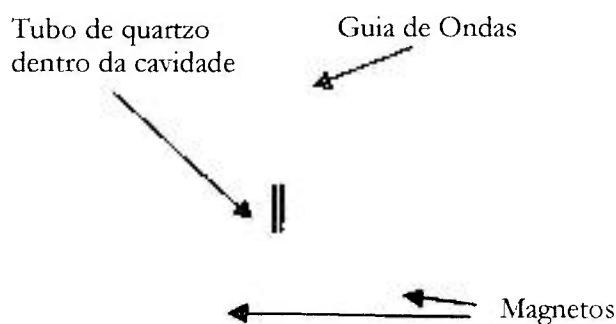


Figura 2.11: Diagrama de cavidade de microondas demonstrando o posicionamento da amostra.

A Figura 2.12 apresenta um gráfico do espectro diferencial de um detector de DL - alanina produzido no IPEN irradiado com 500 Gy de Co-60. O espectro é composto de 5 linhas com intensidades relativas de 1:4:6:4:1, decorrentes da interação do elétron desemparelhado formado na espécie predominante estável com 4 átomos de hidrogênio. Costuma-se expressar o sinal de RPE em unidades arbitrárias (u.a.), relativas às condições da parametrização do espectrômetro. O uso de amostras padrão permite a estimativa absoluta do número de spins formados na amostra. Costuma-se realizar uma normalização dos espectros antes da realização de qualquer comparação ou avaliação.

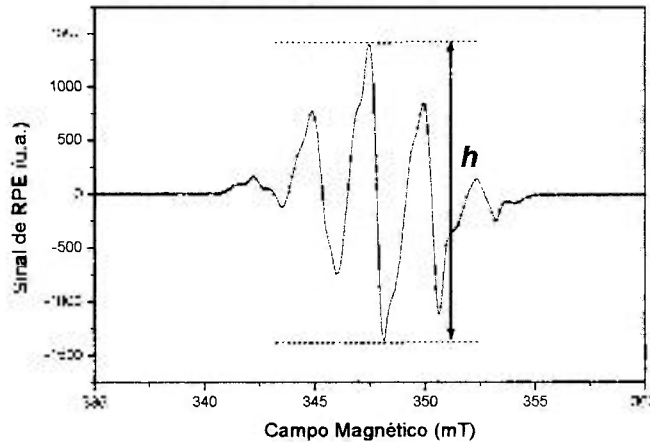


Figura 2.12: Espectro de RPE da DL-alanina irradiada com Co-60, 500 Gy.

A maior amplitude do sinal de pico do espectro diferencial da linha de absorção de RPE da alanina (h) é utilizada normalmente para se estabelecer uma relação entre a dose absorvida pelo material e a concentração de radicais formados induzidos pela radiação.

Essa relação depende das características do dosímetro, do equipamento de medida e de outros fatores que podem influenciar a construção da curva de calibração para o sistema.⁽⁵⁾

A equação para o cálculo da dose é dada por:

$$D(\text{Gy}) = f_c \cdot \left(\frac{PaP}{m} \right) \cdot \prod_i k_i \quad (2.8)$$

onde:

$D(\text{Gy})$: dose absorvida (Gy);

f_c : fator de calibração da curva de calibração;

PaP : é o valor normalizado pico a pico do maior elemento do espectro de RPE do detector (h);

m : massa do detector; e,

k_i : é o fator de correção para incluir alguns efeitos tais como: decaimento do sinal, variação da temperatura, variação da umidade e variação das condições de leitura do equipamento de RPE.

Uma das curvas de calibração para o dosímetro de alanina desenvolvido no IPEN está apresentada na Figura 2.13. Sendo o dosímetro de alanina um padrão secundário, essa curva foi obtida com o objetivo de permitir a dosimetria de campo ou de transferência. A curva apresenta boa resposta entre 10 e 10⁵ Gy⁽¹⁶⁾.

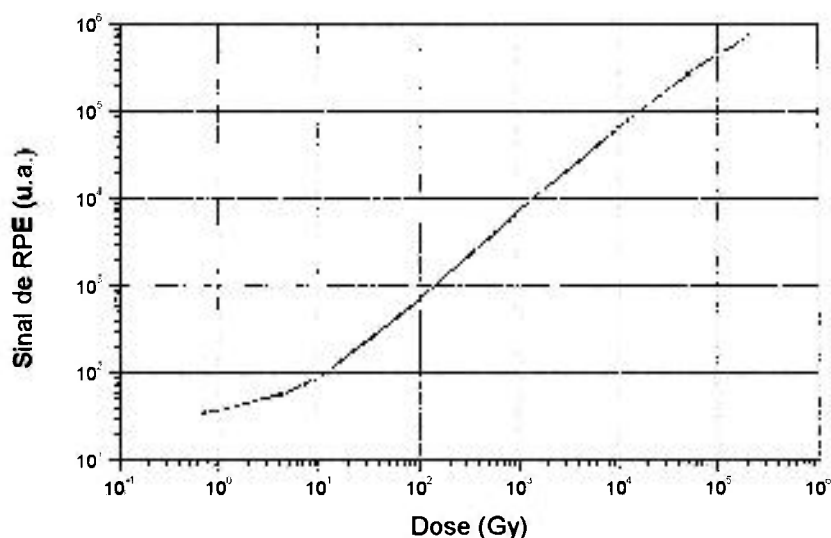


Figura 2.13: Curva de Calibração para o dosímetro de alanina desenvolvido no IPEN com 100 mg.

Como o dosímetro de alanina desenvolvido pelo IPEN^(16,27) tem características próprias, diferenciadas dos encontrados comercialmente, um conjunto de avaliações experimentais foi realizado, visando a otimização da parametrização do espectrômetro confrontando com os parâmetros sugeridos pela literatura e também para a validação e testes do programa computacional em desenvolvimento.

2.2.2 Dosimetria de Acidente

Acidentes como o nuclear de Chernobyl e o radiológico de Goiânia mostraram o potencial de exposição acidental a que o público em geral pode estar sujeito⁽¹²⁾.

Na dosimetria de acidente não se pode contar com uma amostra controlada para a medida. Assim sendo, deve-se realizar uma análise do local irradiado, coletando amostras de materiais paramagnéticos que possam ter sofrido a ação da radiação. Análises de amostras do mesmo material, obtidas de locais que não sofreram a irradiação podem ser utilizadas para o levantamento das curvas de calibração. Outra técnica, conhecida com técnica de aditivação de dose, é baseada na re-irradiação em condições controladas das amostras coletadas no cenário do acidente⁽¹²⁾.

Um detector muito utilizado é o açúcar, que apresenta sinal estável, disponibilidade numa forma relativamente pura, quase sempre é encontrado nos locais de acidente e ainda apresenta similaridade com o tecido humano.^(1,12)

Outros materiais que apresentam bons resultados são os ossos e o esmalte dentário.^(1,2)

2.2.3 Amostra padrão

As amostras padrão são úteis para testes de desempenho dos sistemas, calibração do espectrômetro e medidas quantitativas de concentração⁽²⁹⁾.

Idealmente, a amostra padrão deve conter espécies paramagnéticas estáveis de vida longa, ser facilmente preparada com métodos estáveis e controlados e deve ser caracterizada completamente, respeitando todos os parâmetros espectroscópicos como o tempo de relaxamento e a divisão da estrutura fina e hiperfina.

Além disso, a linha de ressonância deve ser estreita e preferencialmente homogênea. Como amostras padrão pode-se citar^(12,16):

- Manganês (Mn^{++}); e,
- DPPH (2,2-Difenil-1-Picrilhidrazil, $C_{18}H_{12}N_5O_6$).

O DPPH têm um fator g de 2.0036 ± 0.0003 . Um único cristal pequeno de DPPH é uma amostra suficiente para calibrar a fase e a amplitude de modulação do campo de um espectrômetro de RPE. O DPPH foi utilizado como padrão em todas as medidas realizadas.

2.3 Problemas da Dosimetria por RPE

2.3.1 Doses Baixas

Apesar do bom desempenho dos sistemas dosimétricos baseados na alanina na região entre 10 e 10^5 Gy, são relatados problemas quando se procura melhorar o limite inferior da técnica^(19,30,31,32,33,34).

O sucesso das medidas para doses muito baixas depende de fatores que envolvem o processo de fabricação do dosímetro, o sistema de irradiação usado para a construção da curva de calibração do lote e os métodos analíticos para a avaliação da dose^(31,35).

Para a região de doses baixas, menores que 10 Gy, a técnica passa a ter algumas dificuldades decorrentes da existência de interferências no sinal que não estão relacionadas aos radicais livres formados durante a irradiação do detector, prejudicando diretamente a exatidão e a precisão da avaliação de dose.

Alguns trabalhos recentes sugerem que a integral do sinal permitiria estimativas mais precisas da dose recebida pelo material irradiado, desde que o sinal de fundo do dosímetro pudesse ser avaliado com segurança.

A quantidade total de radicais livres formados, obtida pela integral do sinal de RPE, poderia ser utilizada como medida direta da dose absorvida pelo dosímetro. Mas essa técnica apresenta dificuldades quando a linha de base do espectro apresenta perturbações, tornando difícil sua aplicação para doses baixas⁽¹⁹⁾.

Ainda na rotina, o decaimento e a estabilidade dos sinais dos dosímetros de alanina podem afetar os resultados. Os processos poderiam ser acompanhados

e previstos com maior segurança, estabelecendo-se um programa de controle de qualidade de cada lote.

Para o melhor entendimento da importância do tratamento dos dados de RPE, a Figura 2.14 apresenta dois espectros de alanina irradiada com Co-60, com doses de 10 Gy e 10 kGy.

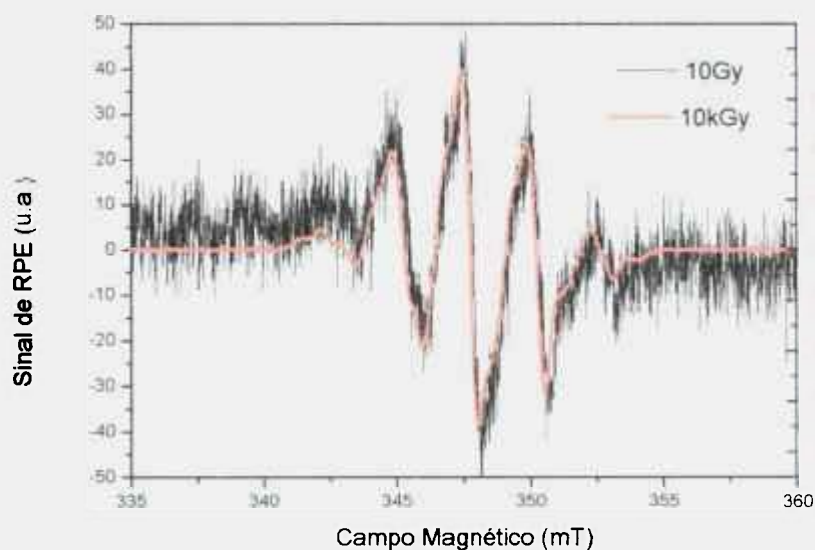


Figura 2.14: Espectros de alanina irradiada com Co-60.

Pode-se notar que o espectro para a dose mais baixa (10 Gy) apresenta uma perturbação que dificulta a avaliação do valor correto de h , que conseqüentemente irá afetar diretamente a avaliação da dose. Métodos matemáticos propostos aqui podem ser utilizados para auxiliar no processo de avaliação dos espectros.

2.3.2 Retrospectiva de Dose

O cenário do acidente é determinante para o sucesso da dosimetria. Cenários ricos em substâncias paramagnéticas estáveis podem permitir uma estimativa retrospectiva da dose mais realista. No entanto, a análise desses materiais é feita com grande dificuldade. Em todos os casos são muitas as fontes de interferência nos sinais.

Recentemente a AIEA, na sua publicação IAEA-TECDOC-1331⁽¹¹⁾, relatou em detalhes o processo de avaliação retrospectiva de dose baseado no esmalte dentário, listando as fontes de incertezas associadas ao processo:

- Decaimento do sinal;
- Preparação da amostra;
- Medida no espectrômetro de RPE;
- Tratamento matemático dos espectros; e,
- Sistema de Calibração.

2.4 Tratamento Matemático dos Sinais

Um conjunto de técnicas para o tratamento dos sinais foi desenvolvido para ser incorporado ao programa visando suportar o conjunto de atividades previstos na dosimetria com a RPE e mitigar ou eliminar as fontes de problema que foram discutidas anteriormente.

Técnicas para melhorar a qualidade dos sinais para doses baixas

Os espectros diferenciais obtidos pelo espectrômetro de RPE são originários de uma composição de sinais de diversas fontes^(30,31,34). Alguns desses sinais não são de origem do material ou substância que se quer medir, mas de interferências causadas basicamente pela interação do sistema de medida com o equipamento e da eletrônica associada ao equipamento. Esse tipo de sinal caracteriza uma perturbação no espectro e que em muitos casos pode interferir e impossibilitar a interpretação do sinal que se quer medir.

Pode-se dizer que o espectro bruto é composto do sinal intrínseco da amostra e outros sinais chamados de ruído e que interferem na medida:

$$\text{Espectro} = \text{Sinal} + \text{Ruído}$$

Os **ruídos** podem ser de diversas fontes e com diversos comportamentos.

Para melhor caracterizar um ruído é necessário investigar sua amplitude e variação no tempo. Uma grandeza que se costuma medir é a relação sinal/ruído - S/R. Quanto maior a relação melhor a qualidade do sinal. Técnicas para melhorar essa relação podem ser experimentais, como o aumento do número de leituras consecutivas da mesma amostra, ou analíticas, como o tratamento matemático do espectro bruto através de filtros⁽³⁵⁾.

Os **sinais** também precisam ser investigados. Alguns materiais detectores apresentam um sinal de fundo chamado de pré-dose que também precisa ser avaliado e até descontado do espectro bruto. Os sinais de fundo podem ter origem na matéria prima utilizada ou induzidos durante o processo de fabricação.

No caso dos dosímetros desenvolvidos no IPEN, onde a alanina pura é encapsulada em tubos de polietileno, essa contribuição é muito pequena.

Pode-se agrupar os seguintes tipos de interferências:

Ruídos de baixa frequência ou de linha de base:

Provenientes dos materiais utilizados para a construção do detector (agregante, encapsulamento) e para a realização da medida no equipamento (tubo de quartzo) e/ou por uma pré-dose induzida no material detector (processo de fabricação dos detectores). Esses ruídos, que influenciam a exatidão e a precisão do processo, podem ser avaliados por meio de medidas realizadas em separado e descontados do sinal bruto.

Ruídos de alta frequência:

Estão associados à eletrônica do equipamento e às condições de ressonância durante a medida. Influenciam muito a precisão da medida e são de origem randômica. Podem ser avaliados através de métodos matemáticos.

Técnicas para auxiliar a construção de curvas de calibração

Uma curva de calibração é o resultado de um conjunto de medidas feitas em condições controladas e com a dose estabelecida por metodologia independente.

A construção das curvas envolve, além de pequenas rotinas estatísticas para cálculo de médias e desvios, algoritmos de regressão linear, interpolação, integração e aproximação de funções.

Na literatura, são sugeridas diversas técnicas para o tratamento dos dados. A Tabela 2.2 apresenta um resumo das técnicas e suas aplicações.

Tabela 2.2: Técnicas matemáticas para o tratamento dos dados e suas aplicações.

Técnicas	Aplicação
Filtros Savitzky-Golay - Modificado ⁽³⁶⁾ Média Móvel ⁽⁴⁰⁾	Suavização (<i>Smoothing</i>) de curvas
Filtros de Fourier ^(37,41,64)	Eliminação de ruídos de alta e baixa frequência
Filtros de Wavelets ^(37,44,45,46,48)	Eliminação de ruídos e extração de sinal
Convolução e Deconvolução ^(38,39,47) Levenberg-Marquart ⁽⁴²⁾	Decomposição de sinais, subtração da linha de base, simulação de espectros
Mínimos Quadrados, Integração, Numérica e <i>Principal Component Analysis</i> - PCA ^(37,42,43)	Curva de Calibração

Como o objetivo deste trabalho é a melhoria da relação sinal/ruído dos espectros brutos, apenas as técnicas associadas ao processamento de sinais baseadas em filtros serão apresentadas em detalhes.

2.4.1 Suavização de Curvas

Muitas técnicas podem ser utilizadas na redução do ruído em espectros obtidos por espectrômetros. Em geral, os *softwares* de pós-processamento de dados utilizam algoritmos simplificados baseados em filtros conhecidos como filtros de suavização. Os métodos mais populares são: Filtro de Média Móvel e Filtro de Savitzky-Golay. Esses filtros são baseados na manipulação direta dos pontos no espectro e o processo de suavização pode resultar numa perda significativa de informação. A aplicação desses filtros deve ser feita com cautela, sendo mais indicada para análises qualitativas de espectros.

2.4.1.1 Filtro de Média Móvel

Os filtros de média móvel são uma família de filtros de suavização de ruído, que depende basicamente do processo de cálculo da média⁽⁴⁰⁾. O algoritmo é composto basicamente de três etapas:

1. Divide-se o espectro que se quer estudar em intervalos seqüenciais, conhecidos como janelas, com mesma largura cobrindo todo o espectro, com um número ímpar de pontos.
2. Para cada intervalo, calcula-se a média dos pontos por algum critério, armazena-se o valor no ponto médio do intervalo pela média calculada.
3. Desloca-se a janela para o próximo ponto e repete-se o procedimento.

No final, tem-se um novo conjunto de pontos calculados a partir da média móvel.

A Figura 2.15 apresenta um exemplo da aplicação de um filtro de média móvel (alanina irradiada com 1 Gy, Co-60).

As vantagens no uso do filtro de média móvel se concentram na fácil implementação computacional e na rapidez no processamento. Mas as desvantagens são diversas, principalmente para sinais com picos acentuados que são característicos nos espectros de RPE.

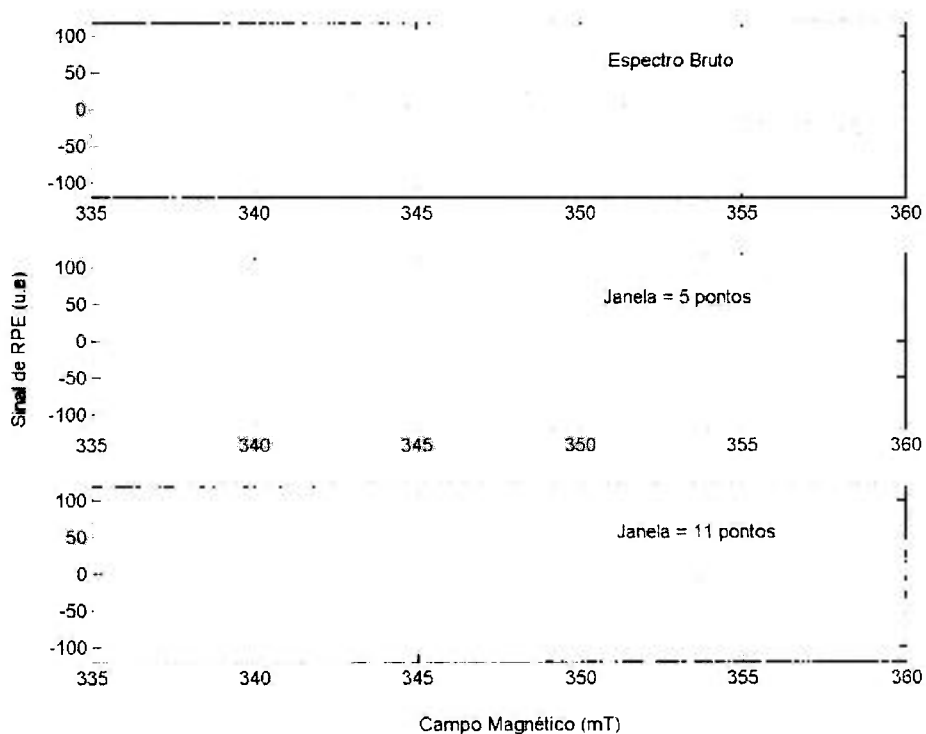


Figura 2.15: Exemplo de aplicação de um filtro de média móvel simples para alanina irradiada com 1 Gy de Co-60.

2.4.1.2 Filtro de Savitzky-Golay

Os problemas encontrados em filtros de média móvel podem ser atenuados utilizando o filtro de Savitzky-Golay⁽³⁷⁾. A técnica, desenvolvida na

década de 60, se tornou a mais popular dos filtros de suavização de espectros, sendo encontrado na maioria dos softwares comerciais da área para tratamento de espectros de ressonância magnética nuclear, cromatografia e espectrofotometria.

O processo é semelhante a do filtro de média móvel, mas neste caso o cálculo do novo ponto é feito a partir de um ajuste polinomial obtido pelo método dos mínimos quadrados. Isso possibilita a sua aplicação em espectros com picos acentuados. Novamente a implementação computacional é simples e eficiente.

O filtro de Savitzky-Golay apresenta bons resultados na suavização de espectros com componentes de baixa frequência e com relação sinal/ruído maior que 3. A técnica exige pontos obtidos no mesmo intervalo e continuidade nos pontos. Essas exigências são razoáveis quando os espectros são de RPE. O cuidado no uso deste filtro está em escolher o tamanho ideal da janela de pontos e o grau do polinômio que se deseja utilizar na aproximação.

Uma observação importante é que a utilização desse filtro em dosimetria pode-se concentrar apenas na suavização da região do espectro onde se quer extrair a informação desejada, ou seja, a avaliação do valor da amplitude h .

A Figura 2.16 apresenta um espectro bruto de RPE (alanina irradiada com 1 Gy, Co-60). Em (a) o espectro todo foi utilizado para a suavização e em (b) apenas o intervalo de (345-350) mT foi utilizado. Em ambos os casos o filtro atuou com um ajuste de grau 3 para o polinômio, para uma janela de 5 pontos.

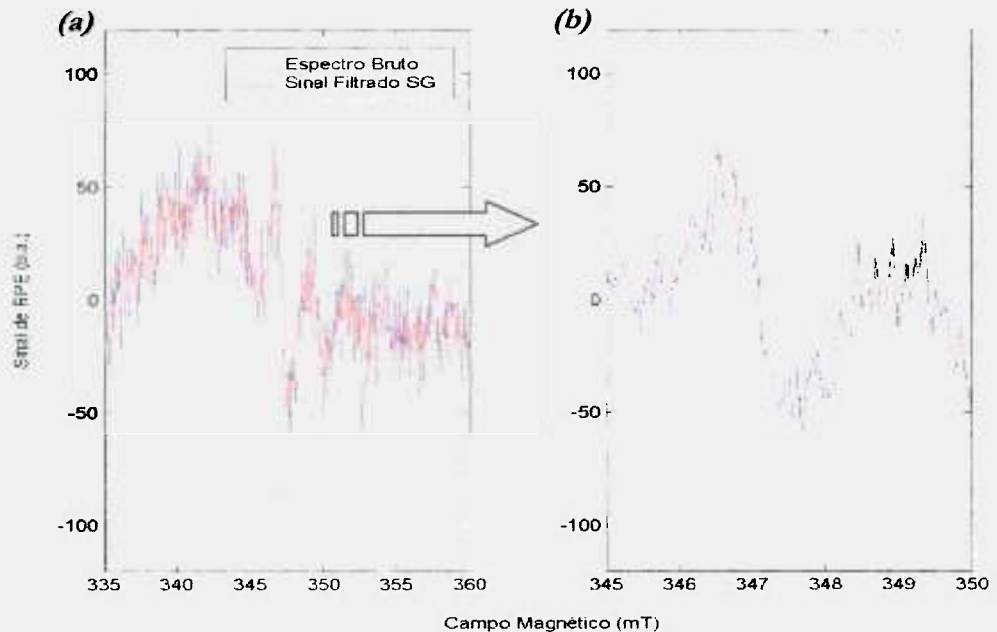


Figura 2.16: Exemplo de aplicação de um filtro de Savitzky-Golay: (a) espectro e sinal extraído; (b) aplicação no intervalo de (345-350) mT para a observação do efeito do filtro.

2.4.2 Técnicas de Extração de Sinais em Espectros Brutos

Até agora foram apresentadas técnicas de suavização baseadas em filtros que atuam diretamente sobre o espectro bruto substituindo os pontos originais por pontos gerados a partir de critérios locais como médias ou aproximações polinomiais. Esses filtros são pouco seletivos e incapazes de diferenciar num espectro bruto os sinais do ruído propriamente dito.

Sendo assim, o processo de suavização acaba removendo parte do sinal.

Mas existem outras famílias de filtros onde é possível a distinção das componentes de um espectro. Esses filtros atuam na extração do sinal de um espectro bruto.

Neste trabalho serão discutidas as famílias de filtros baseadas em transformadas de Fourier e em transformadas de Wavelets. Ambas as transformadas são rotações no espaço de uma representação da função em termos de uma nova função base.

2.4.2.1 Transformada de Fourier

A Transformada de Fourier - TF⁽³⁷⁾, denominada assim em homenagem ao físico francês Jean Baptiste Joseph Fourier (1768-1830), é uma ferramenta matemática largamente utilizada em processamento de sinais. A TF decompõe um sinal contínuo de extensão infinita em séries de senos e cossenos. Aplicando-se uma TF a um sinal no domínio do tempo ou espaço são obtidas informações sobre o domínio de frequências, com fases e amplitudes. No entanto, os espectros de RPE são discretos e limitados. Neste caso pode-se aplicar uma Transformada Discreta de Fourier – DTF⁽³⁵⁾.

Transformada de Fourier de um Sinal Discreto

A TF $X(w)$ de uma série de tamanho infinito de um sinal discreto x_n é definida como:

$$X(w) = \sum_{n=-\infty}^{\infty} x_n e^{-jwn} \quad (2.9)$$

e a transformada inversa como:

$$x_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(w) e^{jwn} dw \quad (2.10)$$

A maior diferença entre a TF de um sinal contínuo e a TF de um sinal discreto reside no fato da TF de um sinal discreto ser uma função periódica de frequência variável normalizada w .

Transformada Discreta de Fourier

Considerando um sinal discreto x_n , podemos escrever sua representação em termos de uma série de Fourier dada por:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{j(2\pi/N)kn} \quad 0 \leq n \leq N-1 \quad (2.11)$$

Onde os coeficientes X_k , conhecidos como coeficientes discretos da Série de Fourier, são dados por:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j(2\pi/N)kn} \quad 0 \leq k \leq N-1 \quad (2.12)$$

A TDF é utilizada na representação e na análise de espectros brutos discreto.

Filtragem utilizando a TDF

A idéia básica é a aplicação da transformada de Fourier no espectro bruto, definida pela Equação 2.12, para a obtenção da representação no domínio das frequências. Essa representação espectral pode ser analisada e frequências não associadas ao sinal que se quer extrair devem ser cortadas. Aplicando-se a transformada inversa, definida pela equação x, pode-se recuperar o sinal agora filtrado, sem as componentes do ruído. Os tipos de filtro possíveis são:

- Filtro *passa baixa*, quando são cortados os coeficientes a partir de um certo valor;
- Filtro *passa alta*, quando são cortados coeficientes até um certo valor; e,
- Filtro *passa faixa*, quando são escolhidos coeficientes entre dois valores de frequência.

2.4.2.2 Filtros de *Wavelets*

Outros métodos matemáticos estudados envolvem a decomposição do sinal com filtros mais avançados chamados de *Wavelets*^(46,47,48).

O princípio dos filtros baseados nas transformadas de *Wavelets* é parecido com a Transformada de Fourier mas com uma grande diferença: enquanto as

funções de base das TF são séries de senos e cossenos localizadas em intervalos de frequências, as funções de base das transformadas de Wavelets são localizadas também no espaço. Isso resulta numa transformada com grande poder de decomposição de sinais em suas partes fundamentais, ou seja, grande poder de extração do sinal e do ruído de espectros complexos.

Os estudos preliminares dessa técnica demonstraram a possibilidade de se discriminar a assinatura do espectro da alanina para doses muito baixas. O problema é a construção das funções de decomposição adequadas.

A idéia principal é construir uma função de base muito próxima ao sinal limpo característico da alanina e aplicar o filtro em sinais com relação S/R não muito boa, ou seja, para o intervalo de 2 a 10 Gy.

Essa técnica apresenta uma limitação associada ao processo de geração das funções de base. Caso o material analisado tenha uma linha de base dependente do decaimento, da qualidade da radiação, ou da dose, ou seja, os radicais livres formados não são muito estáveis em temperatura ambiente, o filtro não irá produzir o efeito desejado.

Em análises efetuadas com o amino-ácido arginina, onde o sinal de RPE sofre grande alteração na sua forma nos primeiros dias após a irradiação, os resultados não foram melhores que os métodos mais tradicionais de tratamento matemático de sinais, apresentados anteriormente.

Investigações sobre o uso de Wavelets

Embora a concepção do programa tenha sido feita baseada nos bons resultados obtidos utilizando técnicas mais tradicionais como as transformadas de

Fourier, o papel do estudo científico é também investigativo. Surgiu desde o início do processo de tratamento matemático de espectros, a pergunta:

Seria aplicável ao problema uma Transformada Discreta de Wavelet -TDW?

Se sim, qual seria a melhor abordagem? Se não, qual seria a justificativa?

Os estudos iniciais não indicavam nenhuma referência de aplicação de tal ferramenta no domínio dos sinais de RPE. Donoho⁽⁴⁶⁾ aplicou em caráter demonstrativo filtros de wavelets em espectros de absorção de RMN (ressonância magnética nuclear).

O processo, chamado de *Denoising Data* (redução de ruído), utilizava uma abordagem conhecida *Wavelet Shrinkage and thresholding* (encolhimento e ceifamento por Wavelets). No entanto, a aplicabilidade dessa técnica parece mais voltada à extração de sinais de espectros com ruído obtidos em condições especiais.

Zheng e colaboradores⁽⁴⁷⁾ desenvolveram uma ferramenta para a extração de sinais voltada ao tratamento de imagens onde existe uma grande tolerância a perdas. Um resumo dos estudos efetuados tentando aplicar a TDW para extração do sinal em espectros de RPE de baixa relação sinal-ruído e os resultados obtidos é apresentado a seguir.

Teoria das Wavelets aplicada na redução de ruído

O principal problema do tratamento matemático dos espectros é a perda no sinal provocada pela aplicação de filtros e outros algoritmos. O importante é

poder avaliar de forma razoável, o quanto tal técnica está influenciando no resultado final.

Nesse contexto, técnicas baseadas em transformadas de Fourier apresentaram bons resultados na maioria de espectros de RPE em que foram aplicadas. Apesar dos bons resultados, como o sinal a ser extraído tem sempre a mesma assinatura, ou seja, a forma básica do sinal de alanina irradiada, uma técnica de extração mais sofisticada poderia ser tentada.

A redução de ruídos utilizando a transformada de Wavelets funciona de forma diferente de um método de suavização (*smoothing*) mais convencional onde boa parte do sinal é atenuado durante o processo de suavização.

A transformada de Wavelet discreta procura extrair o sinal de um espectro perturbado com ruídos, em geral, de alta frequência.

O processo é composto de três etapas: aplicação de uma transformada de Wavelet, ceifamento de alguns coeficientes a partir de um critério adequado, e aplicação da inversa da transformada⁽⁴⁶⁾. O resultado seria um espectro mais limpo sem a presença de ruído de alta frequência.

Os problemas da aplicação da técnica estão concentrados em dois pontos: construção de Wavelets adequadas e escolha de um critério para o ceifamento dos coeficientes.

Considerando um sinal perturbado $X(n)$:

$$X(n) = S(n) + R(n) \quad (2.13)$$

onde:

$S(n)$ é o sinal que se quer extrair

$n = 1..N$, N é o número de pontos que compõem o espectro bruto

$R(n)$ é o ruído que pode ser expresso como:

$$R(n) = \sigma e(n) + r(n) \quad (2.14)$$

onde:

e é um ruído gaussiano branco

r é uma componente associada à linha base do espectro bruto

A transformada de Wavelet $W(a,b)$ pode ser dada por:

$$W(a,b) = \sum_{x=0}^N X(x) \psi_{a,b}(x) \quad (2.15)$$

onde:

a é o parâmetro de dilatação ou de escala da Wavelet

b é o parâmetro de translação ou posição

$\psi_{a,b}(x)$ é a função mãe da Wavelet que pode ser expressa como:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \Psi\left(\frac{x-b}{a}\right) \quad (2.16)$$

Os problemas consistem em encontrar um conjunto de funções que satisfaçam alguns critérios em termos de dilatações e translações. Também é desejável que sejam ortonormais, possibilitando a inversão da transformada.

Dentre as possíveis escolhas para a função $\psi_{a,b}(x)$ foram testadas as famílias de Wavelets mais comumente usadas em trabalhos envolvendo a extração de sinais em espectros com ruído: Haar, Daubechies e Symlet⁽⁴⁴⁾. A escolha envolve aspectos de simetria da função e de características do sinal que se quer extrair.

Para o critério de ceifamento dos coeficientes foi utilizada uma regra elaborada por Donoho conhecida como critério de soft-thresholding dado por ⁽⁴⁶⁾:

$$\delta = \sqrt{2 \log(N) \bar{\sigma}} \quad (2.17)$$

onde:

δ : é a meia-largura da janela de ceifamento; e,

$\bar{\sigma}$: é estimativa da variância do ruído, feita pela média dos coeficientes da transformada.

Depois do ceifamento dos coeficientes, uma transformada inversa é aplicada, restaurando o sinal. Esse processo pode ser repetido diversas vezes, resultando num nível de detalhamento no processo de extração do sinal.

O resultado é a extração do sinal do espectro bruto.

2.5 Estratégia para Condução do Processo

Considerando os problemas levantados, uma estratégia foi elaborada para suportar as atividades de um laboratório de dosimetria de doses altas baseado na técnica da alanina/RPE e que também pudesse atender as necessidades básicas para dosimetria de acidente.

Essa estratégia envolve a padronização dos processos através da definição de um protocolo de medida e a melhoria da qualidade dos resultados, possibilitando a redução do limite inferior da técnica para o intervalo da radioterapia com incertezas compatíveis com as necessidades de cada aplicação.

Doses decorrentes de avaliações acidentais apresentam outros requisitos. As incertezas são sempre maiores, pois muitos dos fatores não podem ser controlados. Aqui, o interesse é a padronização nos procedimentos para minimizar as incertezas associadas ao tratamento matemático dos espectros brutos. Sendo assim, neste caso, os procedimentos podem seguir os adotados para a dosimetria rotineira pela técnica da alanina/RPE.

Pela Equação (2.8) tem-se que a incerteza total na determinação da dose absorvida D depende: da incerteza no valor de f_c que está associada ao procedimento de calibração, da incerteza nos fatores de correção k_i , e na determinação das amplitudes h e h_0 , que dependem das características intrínsecas do tipo de dosímetro, e dos procedimentos de medida no RPE.

Considerando o dosímetro desenvolvido pelo IPEN, as fontes de incerteza podem ser agrupadas da seguinte forma:

- Incertezas associadas ao processo de fabricação do lote de detectores;
- Incertezas associadas ao procedimento de medida adotado, ou protocolo de medida; e

- Incertezas associadas à estimativa do valor das amplitudes h e h_0 .

Cada fonte contribui de forma diferente dependendo dos procedimentos experimentais adotados e da região de doses que se está investigando.

Como foi visto anteriormente, para doses acima da 10 Gy, a relação sinal/ruído é boa e as imprecisões na definição do valor de h são mais baixas. Para doses acima de 10 kGy, as incertezas estão mais fortemente associadas ao procedimento de medida e aos erros carregados do sistema de calibração.

Portanto, a estratégia para a melhoria na avaliação da amplitude h deve contemplar duas ações:

1. Estabelecimento de um protocolo de medida adequado com procedimentos experimentais que devem incluir a parametrização do espectrômetro e a avaliação das incertezas globais; e,
2. Tratamento matemático dos espectros brutos para a extração do sinal de RPE.

Capítulo 3

Métodos Experimentais

3.1 Protocolos de Medida

Tem-se estudado muito sobre a otimização dos procedimentos experimentais nas medidas de RPE nestes últimos anos. Isso decorre não apenas da necessidade de melhoria dos resultados na dosimetria com a alanina, mas principalmente de estudos e pesquisas envolvendo medidas retrospectivas de dose de esmalte dentário em situações de acidente^(11,12).

Como já foi dito, tanto em estudos retrospectivos como em avaliações acidentais, as estimativas de dose absorvida são cercadas de grandes fontes de incerteza. Sendo assim, os técnicos em dosimetria passaram a otimizar os procedimentos experimentais com o objetivo de diminuir as incertezas parciais introduzidas pela técnica de RPE.

3.2 Metodologias Adotadas Internacionalmente

Para possibilitar medidas abaixo de 1 Gy, muitos pesquisadores como Nagy e col.^(34,49), Hayes e col.^(31,50), Haskell e col.⁽³⁰⁾, Bartolotta e col.⁽¹⁹⁾ e Yourdanov e Gancheva⁽⁵¹⁾, têm apresentado metodologias que buscam melhorar a relação sinal/ruído e conseqüentemente diminuir as incertezas nos resultados. Essas metodologias envolvem procedimentos experimentais e um tratamento matemático dos espectros para melhorar a avaliação da dose.

Uma das principais dificuldades na avaliação das dose é a distorção da linha de base do espectro.

Em pesquisas publicadas em 1993, Bartolotta e colaboradores⁽¹⁹⁾ investigaram a distorção da linha de base que é notada em dosímetros não-irradiados. Esse sinal de baixa freqüência, chamado de ruído de fundo, intrínseco de cada dosímetro, varia de lote para lote e é originário do processo de fabricação e dos agregantes e encapsulantes presentes. As variações chegaram até 35% na flutuação no valor da linha de base média.

Além disso, tais sinais são fortemente dependentes da orientação do dosímetro na cavidade. Após a irradiação, a composição desses sinais com o sinal característico do acoplamento do tubo de quartzo utilizado durante as medidas e a cavidade ressonante pode gerar um ruído que impossibilite a avaliação precisa da dose realmente recebida pelo dosímetro.

Nos estudos de Bartolotta, para doses abaixo de 3 Gy, onde o ruído de fundo não pode ser negligenciado, um procedimento para a correção dos sinais era efetuado. No seu trabalho é utilizada a Transformada Rápida de Fourier - TRF, para filtrar os sinais de baixa freqüência que apresentam componentes no domínio de freqüências bem diferenciadas das do sinal proveniente da alanina irradiada.

Bartolotta alerta que tal procedimento deve ser utilizado com cautela, para que componentes não sejam filtrados indevidamente.

Mas a principal conclusão desse trabalho não envolve os métodos matemáticos de tratamento dos espectros, mas sim uma mudança significativa na condução dos procedimentos experimentais e na parametrização no espectrômetro de RPE.

Em geral, os espectros de RPE são obtidos utilizando amplitude de modulação baixa e potência de microondas baixa. Contudo, com essa parametrização a resolução é melhorada em detrimento da sensibilidade.

Nas aplicações em dosimetria somente as variações relativas da amplitude do pico são importantes. Portanto, pode-se selecionar uma parametrização para privilegiar a amplitude, mesmo que exista uma perda na resolução.

Bartolotta demonstrou que a utilização da sobremodulação em amplitude permite o melhor registro do sinal da alanina para doses no intervalo da radioterapia. Em geral, costuma-se utilizar uma amplitude de modulação menor do que a região de interesse que se quer observar. No caso da alanina irradiada, é recomendado o intervalo entre [0,05 - 0,6 mT]. Isso garante que a amplitude de modulação seja menor que a largura da linha.

O trabalho de Wieser e colaboradores⁽⁵²⁾, de 1993, discute a origem do sinal de fundo presente em dosímetros de alanina não- irradiados utilizando a parafina como agregante. Neste trabalho, a amplitude do pico central das 5 linhas do espectro da alanina é usada como indicador para a dose absorvida. Wieser discute a origem desse sinal de fundo: cavidade de microondas, tubo de quartzo e o próprio método de produção do detector de alanina.

Nos últimos anos os debates sobre o estabelecimento de protocolos de medidas para doses baixas foi se acirrando. Muitos pesquisadores passaram a

estudar e desenvolver procedimentos experimentais para melhorar o desempenho dos sistemas dosimétricos para doses baixas.

Métodos envolvendo a integral do espectro de RPE podem dar bons resultados, desde que a perturbação na linha de base provocada pelos sinais de fundo seja pequena ou possa ser descontada. Pass e Shames⁽⁵³⁾ publicaram em 2000 os resultados dos estudos envolvendo o processamento de sinais de RPE para esmalte dentário, comparando três metodologias diferentes.

Neste caso as técnicas de tratamento do espectro bruto envolviam a avaliação do sinal depois de subtraído o sinal de fundo, a integral do espectro diferencial e o uso da segunda derivada do espectro. A conclusão desse trabalho é que a melhor técnica era a utilização da segunda derivada, que não requeria a eliminação preliminar do sinal de fundo.

Ainda em 2000, Wieser e colaboradores⁽⁵⁴⁾ publicaram um outro trabalho envolvendo uma intercomparação de medidas de esmalte dentário efetuadas por 7 laboratórios espalhados pelo mundo.

Mais recentemente, em 2002, Nagy e colaboradores⁽³⁴⁾ elaboraram um estudo bastante elucidativo sobre os avanços na dosimetria de doses baixas, focando nos aspectos da avaliação das incertezas nas medidas.

Na Tabela 3.1, encontra-se um resumo dos principais parâmetros encontrados nos trabalhos envolvendo a dosimetria com alanina.

Tabela 3.1: Principais parâmetros do espectrômetro utilizados em medidas de doses baixas com alanina.

Parâmetro	Wieser 1993	Bartolotta 1993	Hayes 2000	Nagy 2002
<i>Potência de microondas (mW)</i>	4	1	2,5	0,25
<i>Amplitude de modulação (mT)</i>	1	1	1,46	1,4
<i>Constante de tempo</i>	-	-	82ms	1,3s
<i>Resolução (pontos)</i>	-	variável	1.024	1.024
<i>Janela de varredura (mT)</i>	2	-	10	2

3.3 Irradiações

Nas irradiações foram utilizadas três fontes de Co-60.

As fontes de cobalto pertencentes ao Centro de Tecnologia das Radiações CTR/IPEN foram calibradas com dosímetros químicos de sulfato ferroso (Fricke) e apresentam as seguintes características:

Panorâmica (Fabricante: YOSHIZAWA KIKO Co, Ltda. Japão).

Fonte: Cobalto metálico, encapsulado em aço inoxidável em forma de lápis.

Taxa de dose (dez/99)= 0,508 kGy/h - 10 cm atrás.

Atividade (janeiro /00)= 6×10^{13} Bq.

Gammacell 220 (Fabricante: Atomic Energy of Canada Ltd.)

Fonte: 26 lápis dispostos simetricamente; cada lápis com 7 pastilhas.

Taxa de dose (janeiro/2000): 6,55 kGy/h sem atenuador.

Atividade (janeiro/2000): $33,7 \times 10^{13}$ Bq.

A fonte de cobalto pertencente ao Centro de Metrologia das Radiações CMR/IPEN é calibrada por um sistema padrão secundário nível de radioterapia que possui certificado de calibração do National Physical Laboratory (NPL) da Inglaterra, com a rastreabilidade mantida por meio de intercomparações anuais com o Laboratório Nacional de Metrologia das Radiações Ionizantes (LNMRI) do IRD/CNEN. As características são:

Unidade de teleterapia de Co-60 Modelo IS (Fabricante Keleket Barnes Flexaray, USA)

Atividade nominal de 669 GBq (setembro/1999)

Para a realização das leituras no espectrômetro, os detectores irradiados foram retirados do porta-detector, que garante o equilíbrio eletrônico, e acondicionados em um tubo de quartzo e posicionados na cavidade ressonante do espectrômetro. As medidas de RPE foram feitas pelo menos duas horas após as irradiações utilizando um espectrômetro Bruker EMX operando na banda X, em condições ambientes com temperatura perto de 300 K.

3.4 Otimizando a Parametrização do Espectrômetro

Um dos cuidados na espectrometria por RPE é na escolha da parametrização correta para cada tipo de medida.

Na dosimetria utilizando a alanina, o principal aspecto observado é a amplitude do pico central. Neste caso, informações qualitativas de todo espectro são secundárias, mas podem ser utilizadas na indicação de possíveis interferências no sinal, geradas por outras substâncias paramagnéticas ou induzidas por problemas no equipamento de RPE. A maior preocupação é melhorar ao máximo a relação sinal/ruído para doses abaixo de 5 Gy. Para realizar esse estudo foram escolhidos os principais parâmetros que pela literatura podem influenciar a relação sinal/ruído.

Varreduras e amostragem de pontos

A Figura 3.1 apresenta dois espectros brutos da mesma amostra de alanina irradiada com 1 Gy com radiação gama do Co-60, mas com número de leituras acumulativas diferentes, chamadas de varreduras ou *scan*.

Teoricamente, para doses baixas, quanto maior o número de leituras acumulativas, melhor é a relação sinal/ruído.

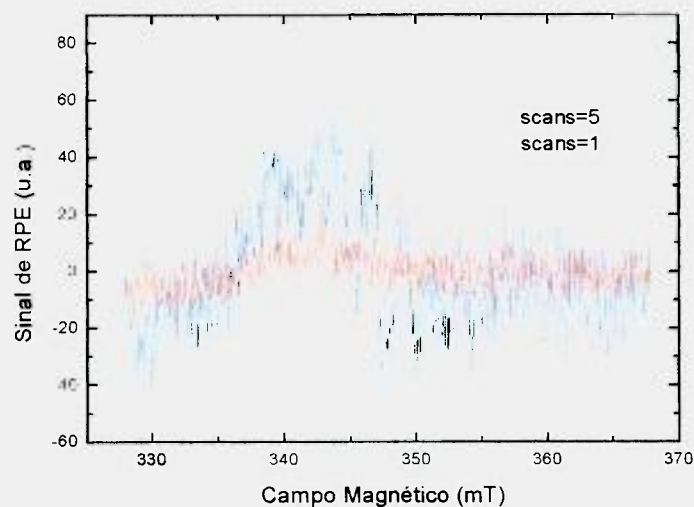


Figura 3.1: Espectros de alanina irradiada com 1 Gy e número de scans ou varreduras de 1 e de 5 vezes.

No entanto, o processo acumulativo consome bastante tempo e a partir de 50 *scans*, o tempo consumido pode comprometer a viabilidade da aplicação da técnica. Por exemplo, considerando a leitura de uma amostra utilizando uma janela de varredura de 10 mT com 1.024 pontos com 50 *scans*, sendo que cada varredura demora 41 s aproximadamente, tem-se um tempo total maior que 34 min para uma única amostra.

Foram adotados os valores de 5 até 10 *scans* para doses baixas e 1 *scan* para doses intermediárias e altas.

Potência de Microondas

Para avaliar a dependência do sinal RPE com a potência de microondas foram irradiados 4 dosímetros com a dose de 1 Gy. A potência foi variada de 8 a 32,5 mW. Como o detector usado tem apenas 100 mg de alanina, a potência de trabalho escolhida foi de 10 mW. A Figura 3.2 apresenta a curva obtida para a variação da resposta relativa com a potência de microondas.

Esse valor é nitidamente superior aos valores encontrados na literatura, mas garante uma sensibilidade adequada ao detector. Nas medidas tanto para doses intermediárias e altas como para doses baixas, foi adotado o valor de 10 mW.

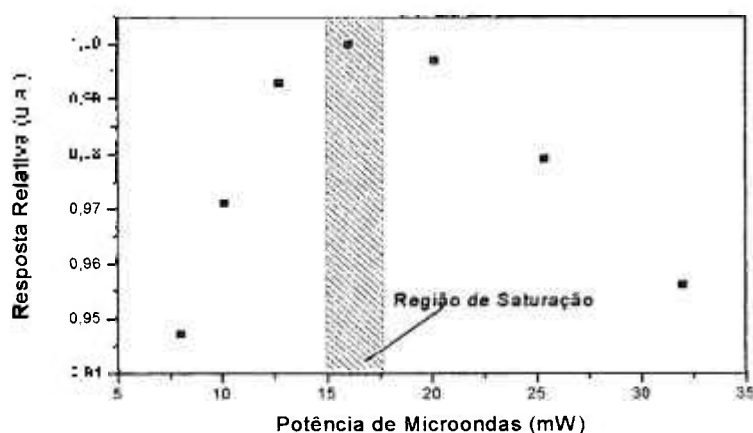


Figura 3.2: Resposta relativa da amplitude h em função da potência de microondas.

Amplitude de Modulação

Um estudo da variação da intensidade do espectro da alanina com a amplitude de modulação foi efetuado. A Figura 3.3 apresenta espectros brutos de alanina irradiada com 1 Gy para amplitudes de modulação de 0,1, 0,6 e 1,0 mT. A potência de microondas utilizada foi de 10 mW.

Como já foi dito, pesquisas demonstraram que a sobremodulação pode ser utilizada como técnica de melhoria na relação sinal/ruído. Isso é observado na Figura 3.3, onde maiores amplitudes de modulação resultaram em maiores intensidades de pico.

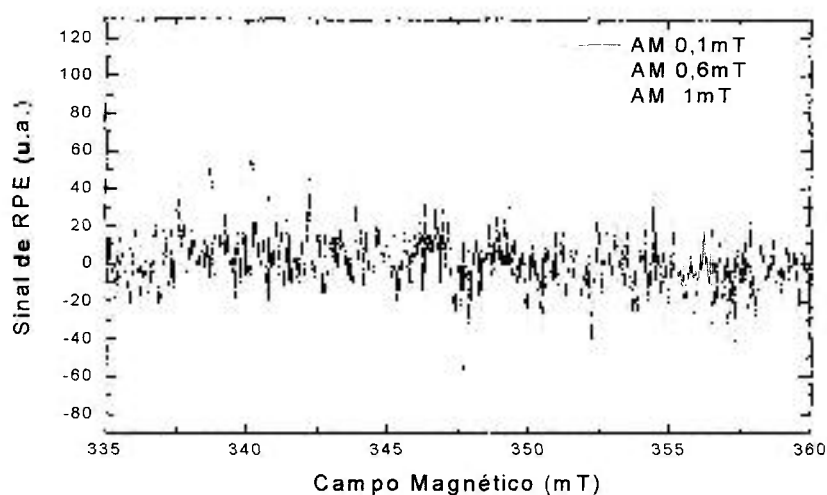


Figura 3.3: Diferentes espectros de alanina irradiada com 1 Gy para amplitudes de modulação (AM) de 0,1, 0,6 e 1,0 mT.

Amostragem

Um outro teste efetuado, verificou a variabilidade do sinal medido com a amostragem, ou seja, o tamanho da janela de varredura e o número de pontos. Foram feitas duas leituras do mesmo material, alanina irradiada com 1 Gy, com a mesma parametrização, alterando apenas a janela de varredura.

A Figura 3.4 apresenta três gráficos, a primeira varredura compreendeu o intervalo de 338 mT até 358 mT (centrada em 348 mT), ou seja, uma janela de varredura (*Sweep Width - SW*) de 20 mT e está representado no gráfico (a). O segundo gráfico, apresenta outra leitura do mesmo material utilizando uma janela de 10 mT (centrada em 348 mT) cobrindo o intervalo de 343 mT até 353 mT (b). No gráfico (c) estão superpostas as duas varreduras para o mesmo intervalo.

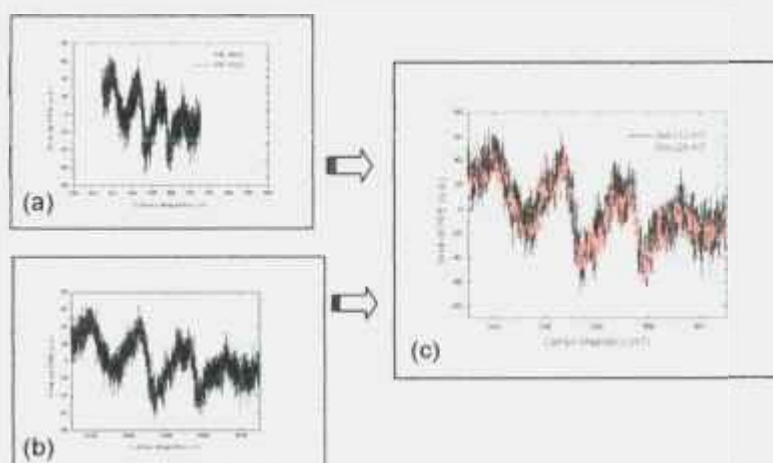


Figura 3.4: (a) Espectro da alanina irradiada com 1 Gy e janela de varredura de 20 mT; (b) mesma alanina mas com janela de varredura de 10 mT; (c) superposição dos gráficos (a) e (b) na região de 343 mT até 353 mT.

Pode-se observar que os resultados são muito próximos, variando apenas na densidade de pontos. A escolha da melhor janela de varredura de campo pode ser tomada em relação à região que se quer observar. Janelas de varredura muito grandes devem prever uma maior densidade de pontos. Janelas podem reduzir o tempo de cada leitura melhorando assim a usabilidade do equipamento.

3.5 Protocolo de Medida Proposto

Foi estabelecido um protocolo de medida baseado nas recomendações da American Society for Testing and Materials - ASTM^(65,66) para ser incorporado ao programa computacional que foi desenvolvido composto de:

1. Procedimentos operacionais para compreender todo o processo desde a fabricação do detector. Esses procedimentos envolvem a utilização da interface do programa durante todo o processo;
2. Procedimentos experimentais para a realização das medidas no espectrômetro. Aqui foi incluída a parametrização do espectrômetro; e,
3. Métodos de tratamento matemático no pós-processamento dos espectros para a avaliação da dose absorvida. Aqui estão consolidadas as técnicas matemáticas que apresentaram melhor performance na extração do sinal do espectro bruto.

O uso do mesmo protocolo de medida em situações envolvendo a dosimetria de acidente é viável, realizando pequenos ajustes na interface.

3.5.1 Procedimentos experimentais

Recomendações Gerais:

- *Seleção entre os tubos de quartzo que apresentavam menor interferência na linha de base. Cada tubo selecionado é identificado e medidas periódicas devem ser realizadas para a comparação do comportamento do espectro de RPE.*
- *Medidas dos detectores irradiados feitas pelo menos após 2h do término da irradiação.*
- *Uso de amostra padrão para contabilizar as flutuações do espectrômetro e detectar problemas com o ganho ou sujeira na cavidade, que podem invalidar uma leitura.*

Leitura para Doses Baixas (<10 Gy):

O procedimento adotado pelo IPEN para doses abaixo de 10 Gy é composto das seguintes atividades:

- 1) Medidas antes da irradiação: das amostras de referência, do tubo de quartzo (1 medida com 1 *scan*), e de cada detector (3 medidas acumulativas rotacionando o tubo em 120° com 5 *scans* cada uma).
- 2) Irradiação dos dosímetros: 2 dosímetros por ponto para medida rotineira e de 3 a 7 para construção de curvas de calibração.

- 3) Medidas depois da irradiação: das amostras de referência, do tubo de quartzo (1 medida com 1 *scan*), e de cada detector (3 medidas acumulativas rotacionando o tubo em 120° com intervalos de 5 a 20 *scans* cada uma).

Leitura para Doses Intermediárias (10-100 Gy) e Altas (> 100 Gy):

O procedimento adotado pelo IPEN para doses acima de 10 Gy é composto das seguintes atividades:

- 1) Medidas antes da irradiação: das amostras de referência, do tubo de quartzo (1 medida com 1 *scan*), e de 10% do lote (3 medidas acumulativas rotacionando o tubo em 120° com 1 *scans* cada uma).
- 2) Irradiação dos dosímetros: 2 dosímetros por ponto para medida rotineira e de 3 a 7 para construção de curvas de calibração.
- 3) Medidas de dosímetros irradiados: das amostras de referência, do tubo de quartzo (1 medida com 1 *scan*), e de cada detector (3 medidas acumulativas rotacionando o tubo em 120° com 1 *scans* cada uma).

3.5.2 Parametrização do Espectrômetro

Os principais parâmetros do espectrômetro que podem ser alterados para melhorar o desempenho das medidas estão listados na Tabela 3.2.

Tabela 3.2. Principais parâmetros utilizados na configuração do espectrômetro.

Parâmetro	Valores	
	Doses Baixas 0,5 - 10 Gy	Doses Intermediárias e Alta 10 - 10⁵ Gy
Potência de Microondas (mW)	10	5 - 10
Amplitude de Modulação (mT)	1 - 2	0,5 - 2
Número de varreduras por amostra	5 - 20 <i>scans</i>	1 - 5 <i>scans</i>
Janela de varredura (mT)	2 - 10	10 - 20
Amostragem de pontos	1024 - 4096	1024

O efeito da otimização da parametrização nos resultados é ilustrado na Figura 3.5. O espectro da alanina irradiada com 10 Gy de Co-60, inicialmente obtido com uma parametrização genérica, foi relido com a parametrização sugerida para Doses Intermediárias e Altas onde foram variadas a janela de varredura, a potência de microondas e a amplitude de modulação. Para melhorar ainda mais o espectro na janela escolhida, uma maior amostragem de pontos poderia ser implementada.

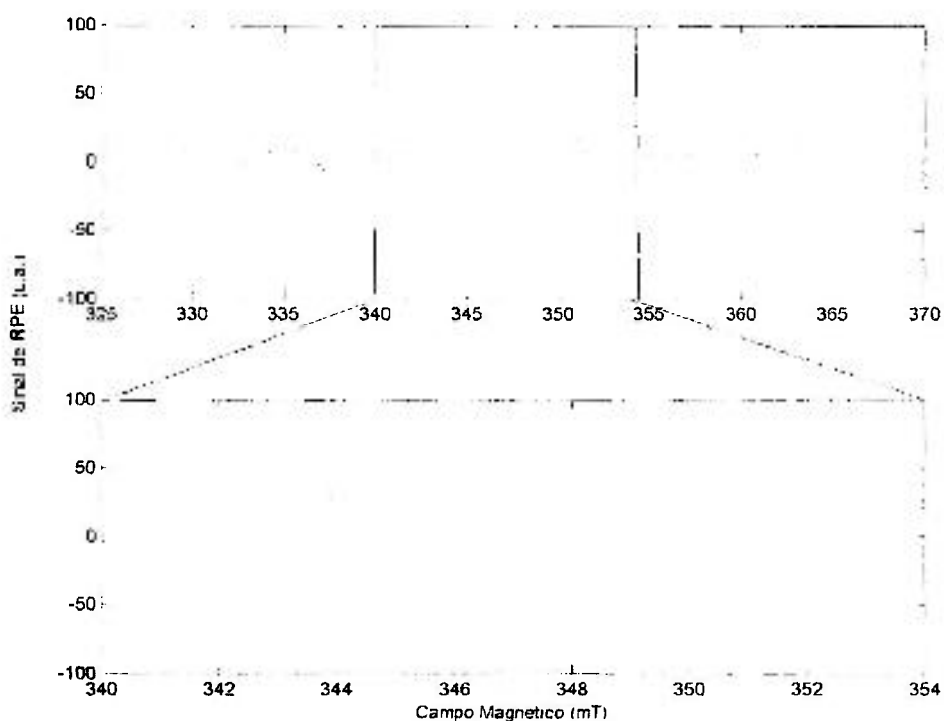


Figura 3.5: Efeitos de uma parametrização otimizada na aquisição de um espectro.

3.5.3 Tratamento dos Espectros

Como o objetivo do desenvolvimento do software é construir uma ferramenta para automatizar ao máximo os procedimentos para avaliação de doses, foi necessária a definição das técnicas mais adequadas ao tratamento matemático dos sinais de RPE.

Dentro do protocolo de medida, para as doses abaixo de 10 Gy, o processo de extração do sinal do espectro bruto com melhores resultados foi o baseado nos Filtros de *Wavelets*.

Na região de interesse do espectro, distorções na linha de base podem afetar diretamente o sinal. Pode-se optar pela subtração da linha de base de duas formas:

- Média da chamada **linha de base referencial** do lote (Figura 3.6), obtida pela suavização utilizando um ajuste polinomial simples dos espectros dos detectores de controle não-irradiados; ou,
- Média da linha de base tomada do próprio detector antes da irradiação, utilizando um ajuste polinomial simples (como estabelecido no procedimento experimental, onde todos os detectores para doses baixas são medidos antes da irradiação).

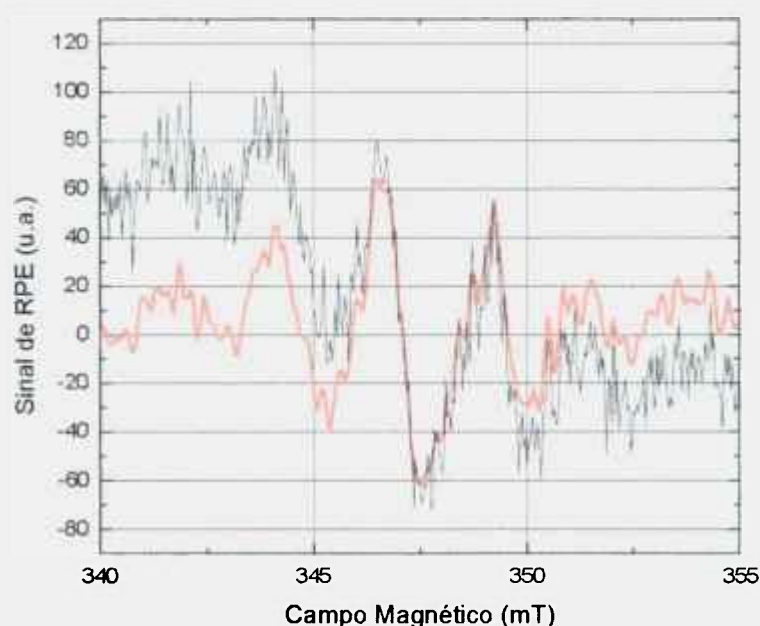


Figura 3.6: Aplicação do pós-processamento baseado no tratamento matemático utilizando Filtros de Wavelets. A subtração da linha de base foi feita utilizando a **linha de base referencial** do lote. O sinal filtrado é apresentado em vermelho.

Para doses no intervalo 10-100 Gy, uma suavização do espectro normalizado é suficiente para melhorar a avaliação do valor da amplitude h .

Para doses acima de 100 Gy, nenhum tratamento matemático é efetuado, e a amplitude h é avaliada diretamente do espectro.

Capítulo 4

Projeto Computacional

4.1 Modelo de Desenvolvimento de Software

Hoje em dia, a construção do conhecimento na área da Física Aplicada está fortemente apoiada na utilização de sistemas computacionais. Sistemas que podem simular modelos que tentam reproduzir o comportamento da natureza ou processar informações a partir de dados brutos e gerar conhecimento. Historicamente, os avanços nessa área sempre foram diretamente dependentes dos recursos computacionais disponíveis.

Mas além de capacidade computacional, nos últimos anos, alguns pesquisadores passaram a se preocupar em desenvolver software científico com características mais profissionais^(57,58). Isso se tornou possível graças ao

amadurecimento da Engenharia de Software que estabeleceu metodologias e técnicas para auxiliar o desenvolvimento de *software* com mais qualidade^(59,60).

Baseado nessa premissa, um estudo mais consistente utilizando o suporte metodológico da Engenharia de *Software*, foi considerado para o desenvolvimento do programa computacional chamado **DataEPR**. O início das atividades envolveram um planejamento da estratégia dessa abordagem que foi baseada em três etapas:

- 1 - Entendimento do problema e suas ramificações;
- 2 - Estabelecimento de uma estratégia para a busca da solução; e,
- 3 - Escolha de metodologias, técnicas e ferramentas para o desenvolvimento.

Dentre os modelos de desenvolvimento disponíveis, o Modelo de Desenvolvimento Evolucionário se mostrou o mais adequado. A Figura 4.1 apresenta um diagrama representativo do modelo.

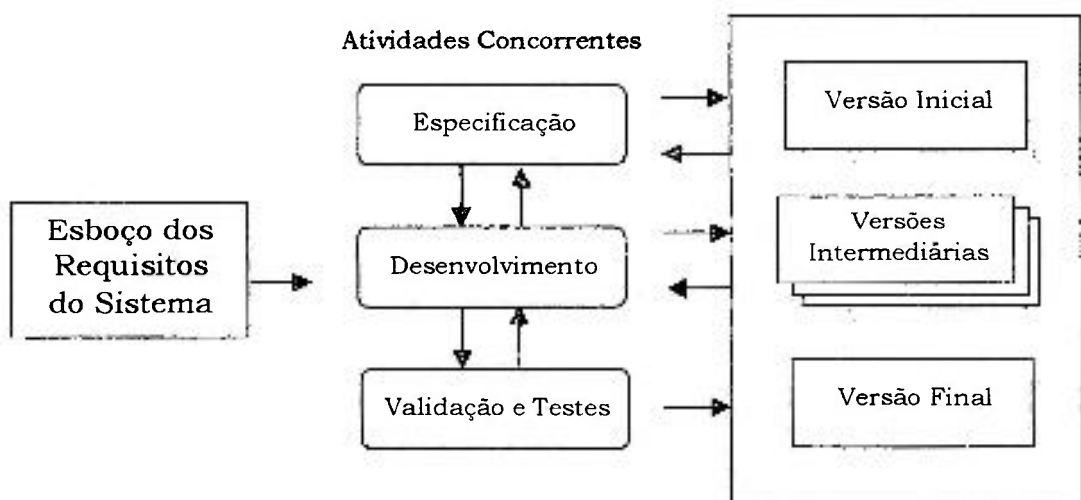


Figura 4.1 Modelo de desenvolvimento evolucionário de *Software*.

Na verdade, o modelo de desenvolvimento evolucionário compreende uma família de modelos conhecidos como modelos de prototipação de *software*, onde os processos são rápidos, os *softwares* não são muito complexos e a equipe de desenvolvimento é pequena.

O ciclo de desenvolvimento é iniciado com uma especificação básica mas completa dos requisitos que o *software* deve atender. Desse ponto em diante, o desenvolvimento é o foco do processo, objetivando sempre a consolidação de versões intermediárias que podem ser validadas, testadas e aprovadas pelo futuro usuário do sistema. Esse processo incremental de melhoria constante, tem muita afinidade com o método científico e a sua aderência ao problema foi muito grande.

Segundo Sommerville⁽⁶⁰⁾ as duas principais vantagens do modelo de desenvolvimento escolhido são o rápido desenvolvimento e a participação atuante do futuro usuário no desenvolvimento, o que melhora a proximidade do produto final às necessidades reais do usuário.

Escolhida a metodologia de desenvolvimento, ainda era necessária a adoção de uma técnica e as ferramentas necessárias.

Baseado na premissa que o Paradigma da Orientação a Objetos seria utilizado, a linguagem Java foi a escolhida como linguagem de programação e a Unified Modeling Language - UML^(61,63), como a linguagem de representação para o sistema. As principais razões para a escolha da linguagem Java^(62,65) para o desenvolvimento foram:

- Suporte à linguagem orientada a objetos;
- Disponibilidade de ferramentas gratuitas para suportar todo o processo de análise e de construção do programa;

- Familiaridade com a linguagem;
- Conjunto de rotinas matemáticas disponíveis;
- Portabilidade da linguagem, garantindo que boa parte do sistema desenvolvido não será dependente de plataforma;
- Facilidade de implementação da interface usuário computador; e
- Integração com a Internet.

4.2 Requisitos Básicos

Para definir o formato do sistema, uma técnica simples de análise de sistemas foi utilizada: a definição dos requisitos do sistema. De forma genérica, esses requisitos devem atender as principais necessidades de um laboratório de doses altas que realize medidas para a rotina com dosímetros de alanina e que também esteja capacitado para investigações em caso de acidente com radiação.

Os principais requisitos foram levantados na forma de características funcionais que o sistema deve atender:

- deve atender às necessidades de um serviço rotineiro de dosimetria de doses altas;
- deve ter flexibilidade para permitir a inclusão de novos procedimentos e práticas de dosimetria;
- não deve ser dependente do tipo de espectrômetro utilizado, apesar de poder utilizar as facilidades dos modelos mais comuns do mercado;

- deve permitir acesso aos dados de forma remota;
- deve permitir a inclusão de novos modelos matemáticos para o tratamento dos dados;
- deve poder incorporar o programa de qualidade do laboratório através da padronização dos procedimentos e inclusão de módulos para o cálculo da incerteza total do método;
- deve manter o histórico de medidas disponível para o usuário através de um banco de dados de espectros.

4.3 Definição dos Processos do Laboratório

As principais atividades do Laboratório de Doses Altas envolvendo a técnica de RPE foram agrupadas em duas macro-atividades:

Dosimetria rotineira: que envolve todas as atividades que compõem um processo de medida, desde o planejamento até a avaliação da dose. O programa deve ser utilizado durante a fase de planejamento para auxiliar na montagem das etapas e na documentação de rastreabilidade de um lote de dosímetros.

Dosimetria de Acidente (Retrospectiva de Dose): que envolve as atividades de investigação de novos materiais dosimétricos, simulações, construção de curvas de calibração e pela técnica de aditivação de dose e avaliações de dose.

A Figura 4.2 apresenta um diagrama simplificado do agrupamento das macro-atividades.

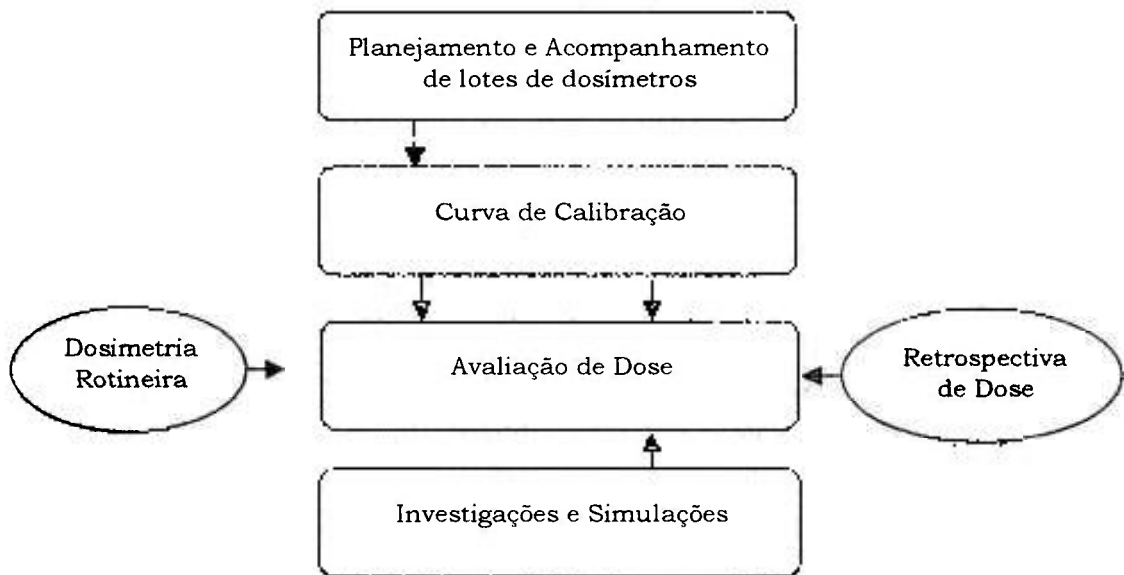


Figura 4.2: Macro-atividades envolvendo a técnica de RPE.

4.4 Concepção do Projeto Computacional

Para a concepção do projeto foi adotada uma metodologia de desenvolvimento chamada *Unified Modeling Language* - UML⁽⁶¹⁾, que é uma linguagem gráfica para a visualização, especificação, construção e documentação de projetos orientados a objetos.

As atividades foram divididas em cinco etapas: levantamento dos Casos de Uso, elaboração dos Diagramas de Classes, organização das rotinas numéricas,

detalhamento dos processos, implementação e validação. A UML norteou grande parte da modelagem do sistema, cobrindo desde a especificação de requisitos até a fase de documentação do *software*.

Apenas durante a construção da interface usuário-computador dentro da etapa de Implementação e Validação, outras metodologias foram necessárias para incorporar aspectos de design e usabilidade de *software*.

As ferramentas de desenvolvimentos escolhidas foram:

Plataforma de Desenvolvimento (Integrated Development Environment - IDE): NetBeans (versão 3.xx) integrado com o Poseidon UML (versão 1.x) para a construção dos diagramas da UML (Figura 4.3).

JDK: na sua versão 1.4.x

Banco de Dados: MySQL⁽⁶⁴⁾.

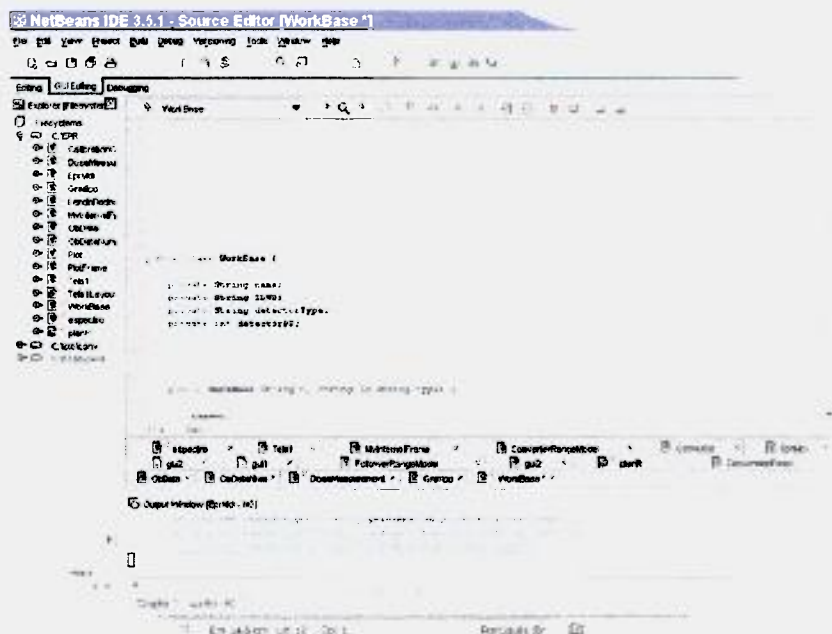


Figura 4.3: Ambiente de Desenvolvimento NetBeans 3.5.x

4.4.1 Levantamento dos Casos de Uso do Sistema

Os Casos de Uso são diagramas representativos do comportamento do sistema em termos de visualizações estáticas dos processos. Os Casos de Uso podem ser entendidos como cenários de uso do programa onde são identificados os atores e as suas atividades junto ao sistema.

Para a dosimetria rotineira, os cenários importantes são:

- Planejamento e acompanhamento de um lote de dosímetros;
- Levantamento de uma curva de calibração – conjunto de medidas;
- Avaliação de dose – registro de medidas, tratamento matemáticos para a extração do sinal e avaliação da dose;
- Estudos para a otimização da parametrização do equipamento – comparação de espectros e sinais análise de sensibilidade considerando a base de dados.

Para a dosimetria Retrospectiva de Dose incluindo os casos de acidente, pode-se listar os seguinte cenários de utilização do *software*:

- Investigações de novos materiais, simulações;
- Levantamento de uma curva de calibração - aditivação de dose;
- Avaliação de dose – registro de medidas, tratamento matemáticos para a extração do sinal e avaliação da dose; e,

- Estudos para a otimização da parametrização do equipamento – comparação de espectros e sinais análise de sensibilidade considerando a base de dados.

Na Figura 4.4 é apresentado um diagrama contendo o macro processo para a Dosimetria Rotineira. O processo engloba a fabricação de um lote, a construção de sua curva de calibração e do uso dos detectores na rotina.

O diagrama pode ser explodido em diagramas de Caso de Uso para cada sub-processo separadamente de acordo com a necessidade de detalhamento. Esse processo foi repetido para todos os possíveis casos de uso do programa.

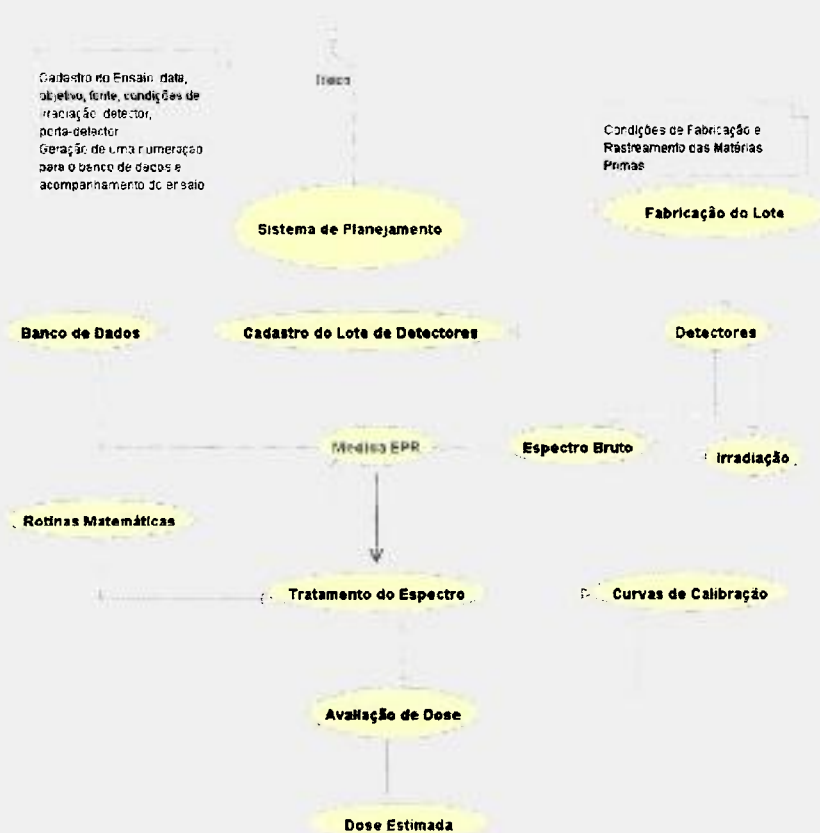


Figura 4.4: Macro-processo Dosimetria Rotineira com alanina.

4.4.2 Elaboração dos Diagramas de Classes

Os Diagramas de Classes apresentam as classes de objetos, interfaces, relacionamentos e colaborações e formam a base para o desenvolvimento de um projeto orientado a objetos. Na Figura 4.5 é apresentado um diagrama contendo algumas das classes de objetos que estão presentes no programa.

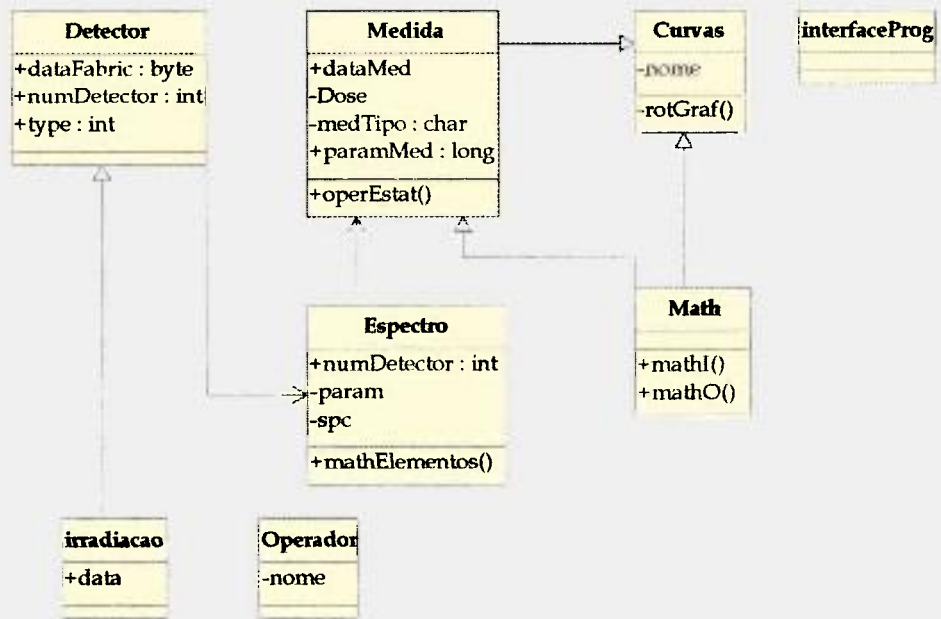


Figura 4.5: Exemplo de um Diagrama de Classes do sistema.

4.4.3 Organização das rotinas numéricas

Levantadas as classes dos objetos, foram codificadas as rotinas numéricas para o tratamento matemático dos espectros, divididos em 4 grupos.

As rotinas para *Manipulação de Entrada e Saída* englobam conversores de formato de arquivos de entrada de dados e para gravação, conversores do banco de dados e compactadores para gravação e recuperação de dados.

As rotinas para *Manipulação de Sinais* englobam as rotinas numéricas de Suavização de Curvas, Filtros de Fourier, Filtros de Wavelets e de outros métodos matemáticos para o tratamento visual do sinal e estimativa de dose^(37,66).

As rotinas para *Manipulação Gráfica* preocupam-se com o armazenamento e apresentação dos gráficos.

As rotinas para *Manipulação Estatística* englobam todo o conjunto de funções estatísticas, tais como médias e desvios, propagação de erros e regressões.

4.4.4 Detalhamento dos Processos

Para o detalhamento dos processos de interação, foram incorporados os procedimentos operacionais definidos no Protocolo de Medida (proposto em 3.4). Essa padronização permitiu a inclusão também de aspectos do Sistema de Qualidade do IPEN, como a geração de relatórios com formatação e numeração adequada e arquivamento de todas as informações do processo na base de dados.

Um exemplo da padronização dos processos é o diagrama apresentado na Figura 4.6. Esse diagrama demonstra o modelo básico do fluxo de atividades, e as informações geradas, para a aquisição de um espectro pelo equipamento de RPE presente em um procedimento operacional. O exemplo é para um equipamento da *Bruker*, mas pode ser adaptado para qualquer equipamento, conhecendo-se o tipo de arquivo gerado na aquisição dos espectros brutos.

O fluxo de atividades é composto de três etapas. A primeira etapa é o planejamento da medida que pode ser feito na própria interface do programa **DataEPR**.

Durante o planejamento, são fornecidas informações que irão compor um documento chamado de Folha de Controle de Medida. Ainda, dependendo do tipo de medida selecionada na interface, um arquivo de configurações baseado na parametrização presente no Protocolo de Medida estabelecido em 3.3 será criado. Esse arquivo ao ser carregado na interface do sistema de aquisição de dados do espectrômetro facilita muito o processo de parametrização do equipamento.

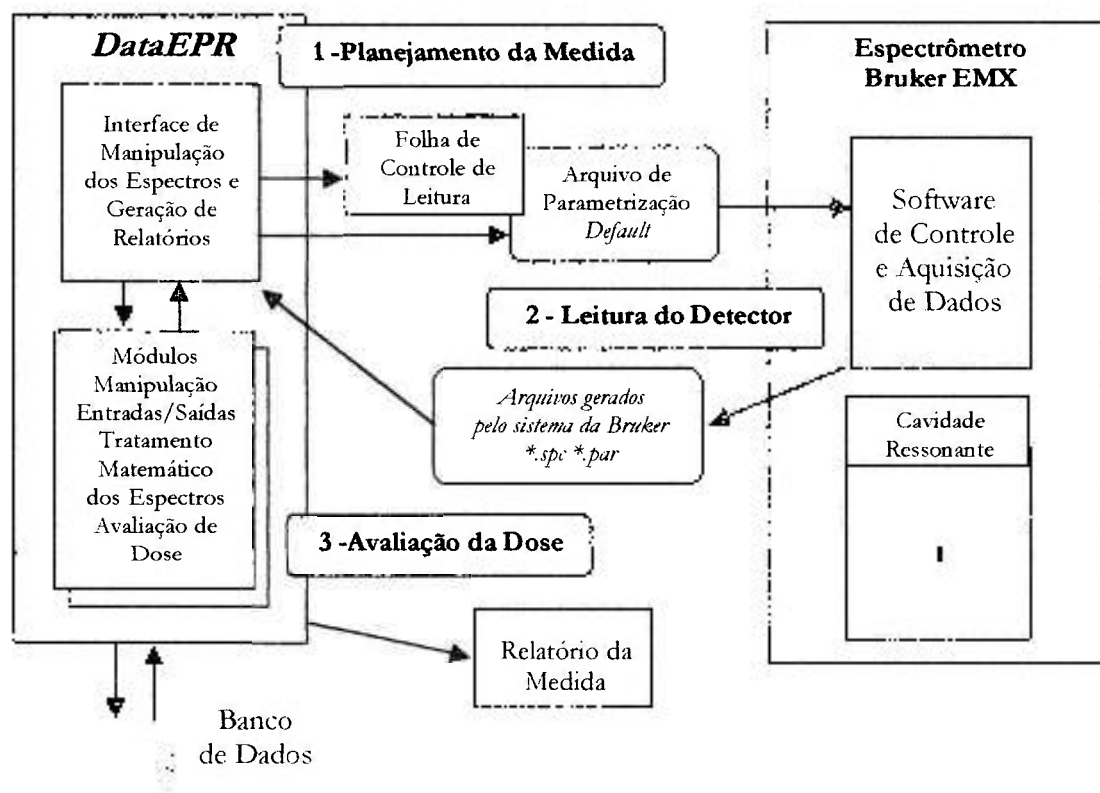


Figura 4.6: Diagrama simplificado do modelo básico do fluxo de atividades para uma medida no espectrômetro de RPE.

A segunda etapa envolve o processo de leitura no espectrômetro. O processo se inicia com a colocação do detector na cavidade do espectrômetro devidamente preparado (estabilizado) pronto para realizar leituras com uma parametrização que é definida no programa de aquisição. Nesta etapa pode ser necessária a utilização de amostras padrão para avaliação das flutuações do espectrômetro.

Depois de cada leitura, o programa de aquisição do espectrômetro grava o espectro e os parâmetros utilizados em arquivos distintos. Em um desses arquivos é gravada toda a parametrização usada para a realização da leitura em formato texto (*.par). No outro arquivo (*.spc) são armazenados os dados do espectro em pares ordenados (X=ponto,Y=leitura) utilizando um formato binário para economia de espaço.

A terceira etapa é a alimentação do programa *DataEPR* com os arquivos e informações gerados durante a etapa de leitura. Foram elaboradas rotinas específicas para a conversão dos arquivos gerados pelo espectrômetro em formato texto. A partir desse ponto, os arquivos são gravados em uma base de dados para sofrerem o tratamento e análise.

Os resultados da avaliação de dose podem compor um relatório com todas as informações importantes.

4.4.5 Implementação e Validação

Seguindo o modelo de desenvolvimento, as versões foram prototipadas em ciclos com a incorporação de especificações mínimas que foram verificadas e validadas em testes de uso do programa. Esses ciclos foram repetidos diversas vezes para incorporar as diversas funcionalidades previstas.

4.5 O programa DataEPR

4.5.1 Visão Geral

Para incorporar a filosofia de automação e de acompanhamento dos processos previstos na análise de Casos de Uso do sistema, a interface foi organizada de forma simples, respeitando os requisitos de usabilidade para processos de interação baseados em interfaces gráficas. A Figura 4.7 apresenta a tela inicial do programa. A organização compreendeu a criação de um espaço de trabalho, denominado *WorkBase*, que é necessário ser criado para que o usuário possa definir os processos, tanto para dosimetria rotineira como em retrospectiva de dose.



Figura 4.7: O programa DataEPR desenvolvido no IPEN.

Um *WorkBase*, representado no diagrama da Figura 4.8, pode conter informações de acompanhamento de um lote de dosímetros, curva de calibração e resultado de medidas, relatórios e todo o memorial das irradiações. Na criação de um *WorkBase* o usuário deve indicar se será utilizado em dosimetria rotineira, retrospectiva de doses, para construção apenas de uma curva de calibração ou em investigações genéricas.

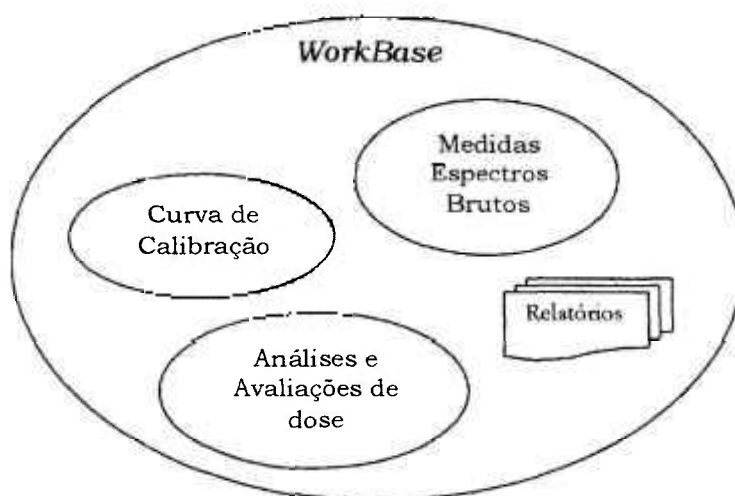


Figura 4.8: Composição de um *WorkBase*.

Os *WorkBases* podem ser selecionados e criados a partir do menu *WorkBase*. Para cada tipo o programa irá abrir uma tela solicitando informações específicas para a continuidade do processo. Na Figura 4.9 é mostrado o início do processo de criação de um *WorkBase* para uma curva de calibração. Um *WorkBase* pode apontar para outro *WorkBase*, ou seja, a curva de calibração criada em um *WorkBase* pode ser utilizada por outro.

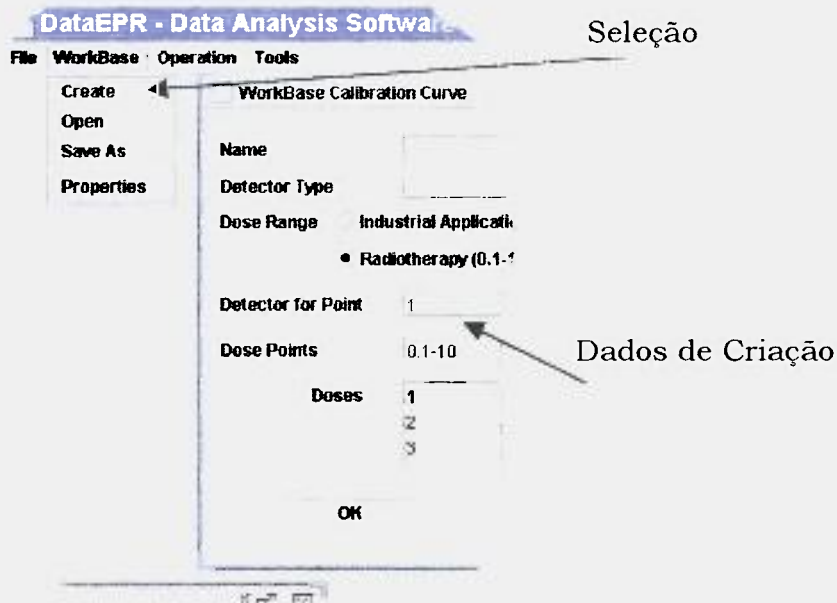


Figura 4.9: Criação de *WorkBase* para comportar uma curva de calibração.

A partir da criação de um *WorkBase* o usuário pode registrar informações sobre o lote de dosímetros, registrar medidas, requerer folhas de controle para ajudar nas medidas experimentais e para arquivamento e realizar análises sobre os espectros.

O programa também permite a visualização de arquivos de medidas. Na Figura 4.10 estão apresentados os dados de parametrização e um gráfico simplificado de um espectro bruto de alanina, para a dose de 100 Gy, Co-60.

Nesta primeira versão do programa o banco de dados conta com cerca de 280 espectros brutos e mais de 50 sinais que foram obtidos pela análise dos espectros. No entanto, o projeto inicial prevê uma versão para a Internet onde todas as medidas realizadas nos últimos anos seriam incorporadas para formar uma base de conhecimento. Uma listagens com as principais classes do programa está apresentada no Apêndice.

Capítulo 5

Resultados e Discussão

A teoria de Wavelets, apesar de não ser trivial, é facilmente implementada computacionalmente. Os resultados apresentados foram gerados a partir do tratamento de dados efetuado pelo programa *DataEPR*, mas, por questões de apresentação, os arquivos foram formatados utilizando o programa ORIGIN.

A Figura 5.1 demonstra o potencial da técnica das Wavelets. O valor de interesse é sempre a melhor determinação do valor da amplitude h .

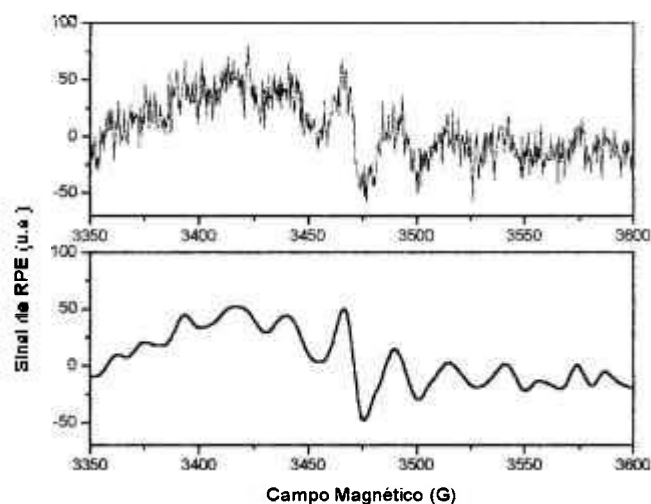


Figura 5.1: Espectro bruto contendo ruído de alta frequência em (a) e o sinal filtrado (ou extraído) sem a componente de alta frequência em (b).

5.1 Desempenho do Sistema

Os esforços computacionais para o processamento dos espectros foram avaliados. Foi considerado um computador *desktop* do tipo *Personal Computer* utilizado rotineiramente nos laboratórios de pesquisa (Intel Pentium III 550 MHz, HD de 20 GB, 256 MB de SDRAM com o MS Windows 98).

Atividade: construção de uma curva de calibração para doses baixas contendo 5 pontos, cada ponto obtido pelas medidas com 3 dosímetros, cada dosímetro processado através de um conjunto de 2 leituras não-acumulativas (antes e depois da irradiação), incluindo também o processamento de 5 detectores de controle, subtração da linha de base, e demais atividades de tratamento matemático, como regressão linear e construção da curva ajustada, todo o processo desde a abertura do *WorkBase* para a inclusão das leituras até a geração

da curva de calibração. Tempo total (incluindo irradiações e medidas no RPE) aproximadamente 2,5 dias. A avaliação do processo foi feita pela divisão do tempo total do processo de interação do usuário com a interface do programa **DataEPR** em quatro etapas:

- *Planejamento e elaboração do WorkBase;*
- *Introdução dos espectros brutos;*
- *Processamento; e*
- *Análises e Geração de Relatórios.*

As atividades envolvendo unicamente a interface do programa **DataEPR** contabilizaram cerca de 2h10min, sendo que a etapa de processamento, onde efetivamente as rotinas de cálculo atuaram, consumiu apenas 45 s.

5.2 Parametrização dos Filtros de *Wavelets*

Para a realização da parametrização do filtro de *Wavelets*, foram feitas diversas análises comparativas do desempenho dos filtros na extração do valor da amplitude h . Foram selecionados do banco de dados 12 espectros brutos de alanina irradiada com 1 Gy, com radiação gama do Co-60. Os espectros foram avaliados pelos algoritmos presentes no programa **DataEPR**.

5.2.1 Extração de Ruídos de Alta Frequência

Os testes para a escolha da melhor parametrização para filtrar o ruído de alta frequência, concentraram-se em dois pontos:

- escolha da melhor função Mãe da Wavelet ou Wavelet-mãe; e
- escolha do melhor critério de ceifamento dos coeficientes.

Foram usadas Wavelets das famílias: *haar*, *daubechies*, *coiflets* e *symlets*. No processo de filtragem dos espectros, as *daubechies* que não são simétricas apresentaram os resultados mais interessantes.

A Figura 5.2 apresenta um conjunto de gráficos do mesmo espectro filtrado pela família de Wavelets chamada de *daubechies* para diferentes níveis de resolução ⁽⁴⁴⁾.

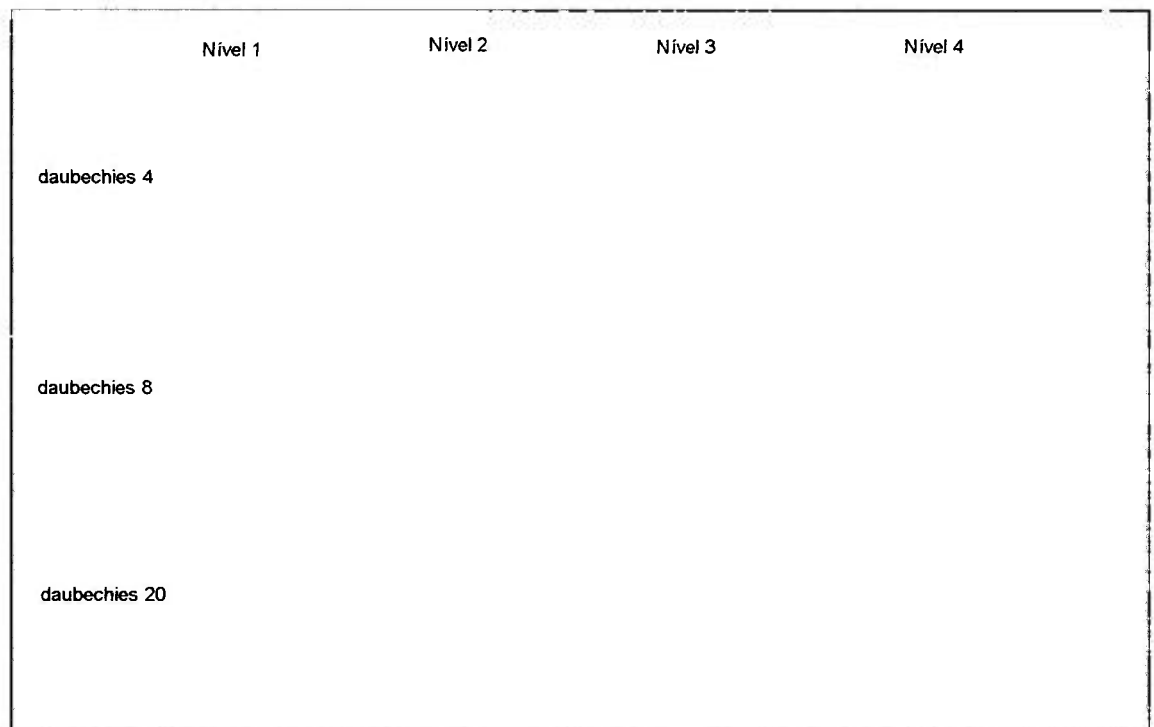


Figura 5.2: Resultados da aplicação do filtro baseado em algumas *Wavelets* da família *daubechies* e aplicada para diversos níveis de resolução. Espectro bruto em azul e sinal filtrado em vermelho.

5.2.2 Desempenho dos Filtros de Wavelets

Como medir o melhor desempenho dos filtros é uma pergunta de difícil resposta. Para os testes de desempenho dos filtros de *Wavelets* foram escolhidos três espectros brutos típicos provenientes de dosímetros de alanina irradiados independentemente com radiação gama do Co-60 para dose de 0,5 Gy, utilizando o protocolo de medida com 10 varreduras (*scans*).

Para as alterações na linha de base, provenientes do encapsulante, do tubo de quartzo e do próprio processo de preparação da alanina, o protocolo de medida realiza uma subtração dos espectros medidos antes e depois da irradiação.

Em geral, as avaliações de desempenho para a filtragem de sinais encontradas na literatura são baseadas em aspectos que não são aplicáveis diretamente ao domínio do tratamento de sinais para fins de dosimetria. Neste caso, foi estabelecido um critério baseando-se no grau de atenuação provocado pela filtragem dos espectros.

O grau de atenuação pode ser associado a uma perda provocada pelo processo de extração do sinal de um espectro bruto. Para os cálculos, foram considerados os valores das amplitudes pico a pico dos três principais elementos do sinal da alanina.

Nos testes executados, os resultados obtidos pelo filtro de *Wavelet* foram superiores aos encontrados pelos filtros baseados em Fourier em cerca de 7%. Ou seja, os filtros de *Wavelets* conseguiram extrair o sinal do espectro bruto com a menor atenuação.

5.3 Extração Direta usando Wavelets

Uma avaliação da capacidade de extração de filtros de *Wavelets* foi testada sem a subtração da linha de base. A idéia aqui é utilizar apenas um único tratamento matemático que teoricamente diminuiria a incerteza total. Foram utilizados 12 espectros de RPE medidos para detectores de alanina com doses de 1Gy, Co-60. Três resultados são apresentados na Figura 5.3.

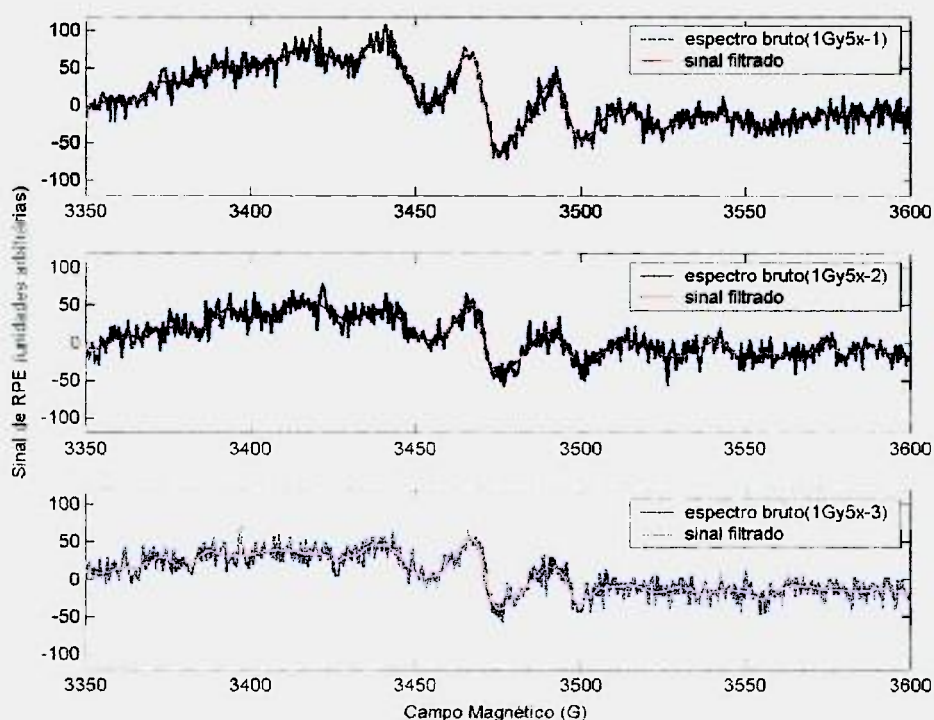


Figura 5.3: Espectros brutos em azul e o sinais filtrados em vermelho para três espectros de alanina irradiada com 1 Gy de Co-60, utilizando um filtro de *Wavelets* baseado na função db8, com quatro níveis de resolução.

A variação dos resultados foi feita comparando a extração direta (sem remoção da linha de base) com a extração partindo de espectros já subtraídos da linha de base. Como a subtração da linha de base do sinal pode promover uma atenuação nos máximos e mínimos presentes no sinal, o resultado da extração direta apresentou sistematicamente uma avaliação do valor da amplitude h maior em aproximadamente 3,5%.

Considerando que os sinais para as condições de parametrização para doses baixas apresentam resposta relativa entre 100 e 300 (u.a.), isso corresponde ao intervalo de 3,5 a 10,5. Os resultados sugerem que o processo pode ser utilizado desde que um novo fator de correção possa ser incluído na equação do cálculo da dose absorvida.

5.4 Identificando as Fontes de Erro no Pós-Processamento

Os erros introduzidos no pós-processamento dos espectros podem ser agrupados em duas categorias:

- *erros provenientes das hipóteses escolhidas para expressar o comportamento dos dados; e*
- *erros provenientes do tratamento numérico dos sinais.*

Os erros da primeira categoria envolvem considerações sobre a natureza estatística dos dados, por exemplo, supor que a curva de calibração possa ser ajustada pelo método dos mínimos quadrados diretamente sem considerações sobre os erros em X ou assumir a homocedasticidade dos dados.

Em geral existe uma incerteza associada aos valores de X , que na curva de calibração são as doses nos pontos escolhidos⁽³⁴⁾.

Os erros da segunda categoria envolvem os processos de suavização e de aproximação que são efetuados pelas técnicas numéricas nos espectros brutos. Também pode-se incluir aí erros que surgem no cálculo do valor médio para h quando as medidas são repetidas mais de uma vez em cada ponto. No presente estudo todos os pontos foram avaliados com três dosímetros.

Os efeitos de variação da parametrização do espectrômetro foram avaliados pela normalização das medidas pelas amostras de referência.

Os efeitos de anisotropia detectados pela rotação da amostra foram pouco significativos em face a outras fontes de incertezas. Em particular para o dosímetro desenvolvido no IPEN, onde o encapsulamento é diferente dos dosímetros desenvolvidos em outros países, as vantagens se concentram em uma melhor resistência mecânica promovida pelo tubo, mas que gera uma perturbação na linha de base que obriga a medida individualizada para cada detector antes e depois da irradiação.

5.5 Avaliação das Incertezas no Pós-Processamento

Considerando as dificuldades no estabelecimento das efetivas contribuições do pós-processamento na incerteza total, foi elaborado um conjunto de testes para avaliar o impacto das rotinas do tratamento numérico do sinal na incerteza total.

O processo compreendeu a fabricação de um lote de 36 detectores que foi subdividido em dois grupos de dosímetros. Um dos grupos foi utilizado para a construção da curva de calibração para as doses: 0,5, 1,0, 2,5, 5,0, 7,5 e 10 Gy, utilizando 3 detectores em cada ponto.

O grupo restante foi utilizado para irradiações e medidas comparativas em 5 pontos com 3 detectores por ponto. Além disso, 3 detectores utilizados para o controle do lote foram armazenados em condições controladas, sem sofrer irradiação.

O protocolo estabelecido no programa *DataEPR* realiza a subtração do espectro a partir da média dos sinais extraídos dos espectros brutos de todos os detectores utilizados para a construção da curva de calibração antes da irradiação (dose zero).

Os testes basearam-se no processo definido para atender aos dois procedimentos experimentais (rotina/calibração).

Depois da subtração de espectros devidamente normalizados pelas leituras das amostras de referência, o sinal foi extraído e o valor de h foi avaliado, tomando a média dos três detectores utilizados em cada ponto. Com esse valor, a dose absorvida pode ser estimada através da curva de calibração.

O primeiro conjunto de testes procurou apenas detectar o impacto do processo de determinação de h , ou seja, o impacto das técnicas numéricas de suavização ou extração do sinal do espectro bruto.

Para isso foram utilizados três tipos de abordagens: Filtros de Fourier, Filtros de Wavelets e Ajuste Polinomial baseado na técnica de Savitzky-Golay discutida anteriormente.

Os resultados do impacto na incerteza total estão apresentados na Figura 5.4. Neste teste, a construção da curva de calibração não considerou possíveis erros laboratoriais na determinação da dose ou seja, X livre de erros.

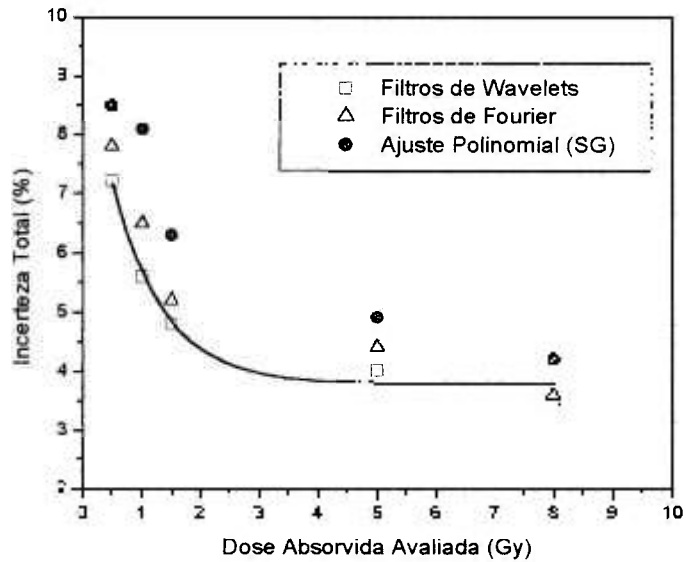


Figura 5.4. Incerteza total (1σ) em função do tipo de abordagem para a extração do sinal do espectro bruto.

Apesar da heterocedasticidade dos dados, os filtros de Wavelets demonstraram boa capacidade de extração dos sinais. Os filtros foram construídos a partir da família de *Wavelets daubechies*.

O segundo conjunto de testes procurou estimar o impacto do processo de determinação da curva de calibração. Neste teste, um ajuste polinomial foi usado na suavização do espectro.

Em geral, o método dos mínimos quadrados é adequado para o levantamento da curva entre 10^{-10} a 10^5 Gy, apesar de estudos demonstrarem a necessidade de segmentar essa região em mais de uma reta^(34,41).

Foram feitos dois testes, um considerando os valores de X, que representam a dose absorvida, livres de erros e um outro considerando um erro

laboratorial de 7% para todos os valores de dose. Os resultados estão apresentados na Figura 5.5.

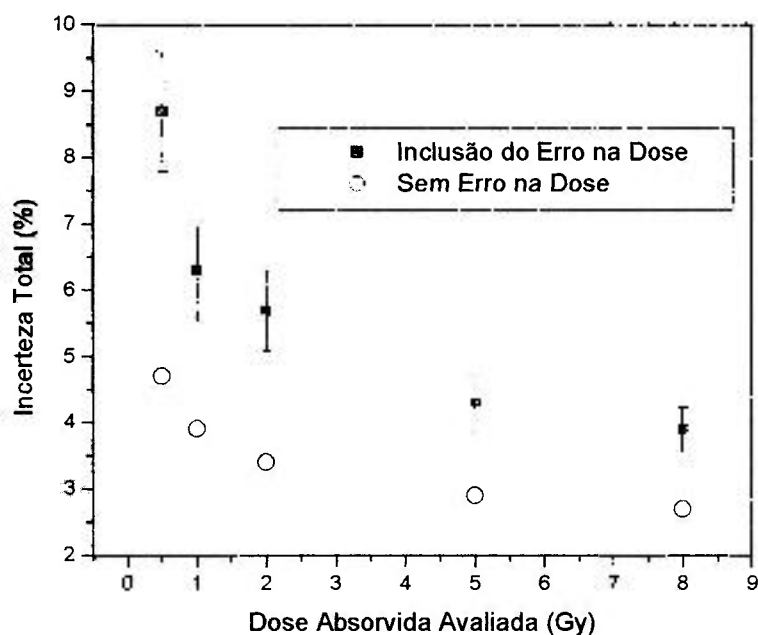


Figura 5.5. Incerteza total (1σ) em função da inclusão de erros na determinação das doses para a construção da curva de calibração.

Pode-se notar pelo gráfico da Figura 5.5, que com a inclusão de um erro laboratorial na determinação das doses para a construção da curva de calibração, as incertezas totais assumiram valores elevados, acima da meta que era abaixar a incerteza para 3%. Isso aconteceu devido aos problemas na determinação das doses nos sistemas de irradiação.

Uma observação importante, é que o número de repetições para cada ponto da curva de calibração é um fator importante para a redução dos erros⁽³⁴⁾.

No entanto, o processo com mais de 3 detectores por ponto de medida de dose consome um tempo substancial, e o levantamento de uma única curva de calibração com 5 pontos, com 7 repetições por ponto pode levar mais de 3 dias de trabalho, tornando a alanina/RPE pouco competitiva para dosimetria em radioterapia.

No presente estudo, apenas 3 detectores foram utilizados em cada ponto de medida de dose, permitindo a construção da curva em aproximadamente 2 dias.

Capítulo 7

Conclusões

A dosimetria pela técnica de RPE utilizando a alanina como detector da radiação apresenta excelentes resultados para a dosimetria de doses altas em um intervalo de doses de 10 Gy a 10^5 Gy, com uma incerteza total de 5%.

No entanto, para doses abaixo de 10 Gy, devido à baixa relação sinal/ruído, os erros podem superar os 5%. Para diminuir a incerteza total do método, cuidados devem ser tomados na definição dos procedimentos experimentais e na escolha das ferramentas matemáticas para o pós-processamento. Os bons resultados para a dosimetria de doses baixas dependem de um protocolo de medida composto de um procedimento experimental adequado, uma parametrização otimizada do espectrômetro e um conjunto de técnicas para o tratamento matemático com o melhor desempenho possível no processo de extração.

Nesse contexto, a partir dos resultados obtidos pode-se concluir:

- 1) O protocolo de medida desenvolvido e apresentado neste trabalho para o dosímetro de alanina desenvolvido no IPEN alcançou resultados excelentes e comparáveis aos resultados obtidos pelos sistemas dosimétricos internacionais.
- 2) A técnica matemática baseada em Filtros de *Wavelets* proposta apresentou os melhores resultados na extração de sinais de espectros brutos para doses entre 0,5 - 10 Gy, contabilizando incertezas totais médias menores que 5%.
- 3) A consolidação dos procedimentos do laboratório e a necessidade de padronização nas medidas só pode ser conseguida através da automação do processo. Nesse aspecto o programa computacional **DataEPR**, apesar de ainda estar na sua primeira versão, é uma ferramenta chave na convergência desse processo.

Apêndice - Programa *DataEPR*

DATAEPR - LISTAGENS PARCIAIS DA GUI, PLOTADOR E DE ROTINAS DE MANIPULACAO

Orlando Rodrigues Júnior

(comentários em Inglês em geral foram gerados pelos IDEs de desenvolvimento, principalmente na parte da interface.)

Rotina para a transformada de Wavelet - db4

```
import java.lang.Math.*;

/**
 * Daubechies D4
 * adaptado do Numerical Recipes versão Fortran
 * Orlando-22-04-2003 v0.3
 */
class db {
    protected final double sqrt_3 = Math.sqrt( 3 );
    protected final double denom = 4 * Math.sqrt( 2 );
    //
    //
    protected final double h0 = (1 + sqrt_3)/denom;
    protected final double h1 = (3 + sqrt_3)/denom;
    protected final double h2 = (3 - sqrt_3)/denom;
    protected final double h3 = (1 - sqrt_3)/denom;
    //
    // coeficientes da transformada
    //
    protected final double g0 = h3;
    protected final double g1 = -h2;
    protected final double g2 = h1;
    protected final double g3 = -h0;

    //
    // Coeficientes da transformada inversa
    //
    protected final double lh0 = h2;
    protected final double lh1 = g2; // h1
    protected final double lh2 = h0;
    protected final double lh3 = g0; // h3
    //
    //
    protected final double lg0 = h3;
    protected final double lg1 = g3; // -h0
    protected final double lg2 = h1;
    protected final double lg3 = g1; // -h2

}

/**
 *
 */
protected void transform( double a[], int n )
{
    if (n >= 4) {
        int i, j;
        int mm = n >> 1;

        double tmp[] = new double[n];

        i = 0;
        for (j = 0; j < n-3; j = j + 2) {
            tmp[i] = a[j]*h0 + a[j+1]*h1 + a[j+2]*h2 + a[j+3]*h3;
            tmp[i+mm] = a[j]*g0 + a[j+1]*g1 + a[j+2]*g2 + a[j+3]*g3;
            i++;
        }

        tmp[i] = a[n-2]*h0 + a[n-1]*h1 + a[0]*h2 + a[1]*h3;
        tmp[i+mm] = a[n-2]*g0 + a[n-1]*g1 + a[0]*g2 + a[1]*g3;

        for (i = 0; i < n; i++) {
```

```

        a[i] = tmp[i];
    }
} // transform

protected void invTransform( double a[], int n )
{
    if (n >= 4) {
        int i, j;
        int mm = n >> 1;
        int mmPls1 = mm + 1;

        double tmp[] = new double[n];

        tmp[0] = a[mm-1]*lh0 + a[n-1]*lh1 + a[0]*lh2 + a[mm]*lh3;
        tmp[1] = a[mm-1]*lg0 + a[n-1]*lg1 + a[0]*lg2 + a[mm]*lg3;
        j = 2;
        for (i = 0; i < mm-1; i++) {

            tmp[j++] = a[i]*lh0 + a[i+mm]*lh1 + a[i+1]*lh2 + a[i+mmPls1]*lh3;
            tmp[j++] = a[i]*lg0 + a[i+mm]*lg1 + a[i+1]*lg2 + a[i+mmPls1]*lg3;
        }
        for (i = 0; i < n; i++) {
            a[i] = tmp[i];
        }
    }
}

/**
 * transformada
 */
public void daubTrans( double s[] )
{
    final int N = s.length;
    int n;
    for (n = N; n >= 4; n >>= 1) {
        transform( s, n );
    }
}

/**
 * inversa
 */
public void invDaubTrans( double coef[])
{
    final int N = coef.length;
    int n;
    for (n = 4; n <= N; n <<= 1) {
        invTransform( coef, n );
    }
}
} // db

```

Classe DoseMeasurement

- classe para armazenar informações do WorkBase que não está incorporadas no Main

```

/*
 * DoseMeasurement.java
 *
 * Created on 11 de Maio de 2003, 11:28
 */

/**
 *
 * @author Orlando
 */
public class DoseMeasurement extends WorkBase{

```

```

int detectorDecay;//

/** Creates a new instance of DoseMeasurement */
public DoseMeasurement(String n,String id, String type) {
    super(n,id,type);
}

//
public int getdetectorDecay(){
    return this.detectorDecay;
}
//
public void setdetectorDecay(int ddy){
    this.detectorDecay=ddy;
}
}

```

Classe main com as rotinas de conversão dos arquivos *.SPC da Bruker

```

/**
 * @author Orlando
 * Parte do Sistema Para Avaliação de Espectros de EPR
 * ver doc sobre estrutura do sistema
 * Versão 0.7 - NetBeans 3.4
 * reorganizando as classes - abertura de múltiplos espectros
 *
 *
 */

import javax.swing.*;
import javax.swing.JInternalFrame;
import javax.swing.JDesktopPane;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import javax.swing.JMenuBar;
import javax.swing.JFrame;
import javax.swing.filechooser.FileFilter;
import java.awt.event.KeyEvent;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.KeyStroke;
import java.awt.event.*;
import java.awt.*;
import java.awt.geom.*;
import java.util.Vector;
import java.awt.BorderLayout;
import java.io.*;
import java.util.*;
import java.lang.Object;

public class EprMdi extends JFrame
{
    // Usando JDesktopPane para o MDI
    JDesktopPane desktop;

    /**
     * Constructor para EprMdi.
     */
    public EprMdi()
    {
        super("DataEPR - Data Analysis Software for EPR Dosimetry
                IPEN/CNEN- BRAZIL");

        // Criando GUI
        desktop = new JDesktopPane ();
        //Posso mudar a cor de fundo aqui:
        //desktop.setBackground(Color.green);
        //Usando JDesktopPane preciso colocar um primeiro frame na
        // área de trabalho:

```

```

createFrame();
setContentPane(desktop);
//Menu do MDI:
setJMenuBar(createMenuBar());
//acelerando o dragg
desktop.putClientProperty("JDesktopPane.dragMode","outline");
//maximizo a janela:
setVisible(true);
setExtendedState(this.MAXIMIZED_BOTH);

// aqui estou setando o LookAndFeel para a GUI
//podemos ter problemas aqui!
//final JFrame me=this;
//try{
//  UIManager.setLookAndFeel("com.sun.java.swing.plaf.motif.MotifLookAndFeel");
//  SwingUtilities.updateComponentTreeUI(me);
//} catch (Exception ignore){
//  System.err.println("Erro!!!!"+ignore);
//}

//mostro:
this.show();

}

/**
 * Method createMenuBar.
 * @return JMenuBar
 */
private JMenuBar createMenuBar()
{
//Criando um menu para a Janela MDI:
JMenuBar menuBar = new JMenuBar();
//File
JMenu menu = new JMenu("File");
JMenuItem menuItem=new JMenuItem("New");
menuItem.setMnemonic(KeyEvent.VK_N);
menuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N, KeyEvent.CTRL_MASK));
menu.setToolTipText("Nova Janela");
//Tratador de eventos Anônimo para a nova janela
menuItem.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        createFrame();
    }
});
menu.add(menuItem);
//Adiciono o menu ao menuBar:
menuBar.add(menu);
//WorkBase
menu =new JMenu("WorkBase");
menuItem=new JMenuItem("Create");
menuItem.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        createWB();
    }
});
menu.add(menuItem);
menuItem=new JMenuItem("Open");
menuItem.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String nomeArq=openFile();
        LendoDados lendo= new LendoDados();
        //
        Vector dados= new Vector();
        Vector dadosSPC=new Vector();
        //aqui tenho problemas ainda naum consegui fazer o frame do PLOT
        //abrir dentro do MDI
        dados=lendo.Leitura(nomeArq);
        dadosFrame(dados,nomeArq);
        dadosSPC=readSPC(nomeArq);
        PlotFrame plotDados = new PlotFrame();
        plotDados.setDados(dadosSPC, nomeArq);
    }
});
}

```

```

    //
    }
    });
    menu.add(menuItem);
    menuItem=new JMenuItem("Save As");
    menuItem.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            // createFrame();
        }
    });
    menu.add(menuItem);
    menu.addSeparator();
    menuItem=new JMenuItem("Properties");
    menuItem.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            //createFrame();
            JInternalFrame gframe = new JInternalFrame("Frame Interno",true,true,true,true);
            //gframe.setTitle("Orlando");
            gframe.setVisible(true);
            Container c=gframe.getContentPane();
            //chamo uma classe para desenho
            Grafico gg=new Grafico();
            c.add(gg, BorderLayout.CENTER);
            gframe.setSize(400,400);
            gframe.setOpaque(true);
            desktop.add(gframe);
        }
    });
    menu.add(menuItem);
    menuBar.add(menu);
//Operações:
    menu =new JMenu("Operation");
    menuBar.add(menu);
//Ferramentas:
    menu =new JMenu("Tools");
    menuBar.add(menu);
//Para o Help ficar posicionado na esquerda:
    menuBar.add(Box.createHorizontalGlue());
//Menu do Help
    menu =new JMenu("Help");
    menuItem=new JMenuItem("Search");
    menu.add(menuItem);
    menuItem=new JMenuItem("Exit");
    //Exit
    menuItem.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            System.exit(0);
        }
    });
    menu.add(menuItem);
    menuBar.add(menu);

//Adiciona a barra de menus ao frame:
    this.setJMenuBar(menuBar);
//Retorno o menuBar para o desktop:
    return menuBar;
}

// ===== Métodos:
/**
 * Method createFrame.
 */
private void createFrame()
{
    FrameInterno frame = new FrameInterno();
    frame.setVisible(true);
    desktop.add(frame);
    try {
        frame.setSelected(true);
    } catch (java.beans.PropertyVetoException e) {}
}
//

```

```

}
//Create WorkBase
private void createWB()
{
    Tela1 window = new Tela1();
    window.setTitle("WorkBase Calibration Curve");
    window.setVisible(true);
    desktop.add(window);
    try {
        window.setSelected(true);
    } catch (java.beans.PropertyVetoException e) {}
}

//Método de inserção de um frame de apresentação dos dados dos arquivos par
//e spc

private void dadosFrame(Vector dados, String nomeArq)
{
    final FrameInterno dadosframe = new FrameInterno();
    Vector d= new Vector();
    d=dados;
    JTextArea txt =new JTextArea(30,40);
    JScrollPane pCentro = new JScrollPane(txt);
    JPanel pSul=new JPanel();
    //renomeio o novo frame com o path+nome do arquivo aberto
    //rever o title da janela
    dadosframe.setTitle(nomeArq);
    JButton ok=new JButton("OK");
    ok.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e)
        {
            dadosframe.setVisible(false);
        }
    });
    pSul.add(ok);
    //redimensiono o frame:
    dadosframe.setSize(200, 500);
    dadosframe.setVisible(true);
    // getContentPane().add(pCentro,BorderLayout.CENTER);
    // getContentPane().add(pSul,BorderLayout.SOUTH);
    //Precisei referenciar diretamente!
    dadosframe.getContentPane().add(pCentro,BorderLayout.CENTER);
    dadosframe.getContentPane().add(pSul,BorderLayout.SOUTH);
    //varro o vector de pares param,valor
    for (int j = 0; j < d.size(); j++)
    {
        ObData p = (ObData) d.elementAt(j);
        // txt.setText("-----"+j);
        txt.append(p.getParam()+"="+p.getValor());
        txt.append("\n");
    }
    desktop.add(dadosframe);
}

|

-----

/*
 *openFile
 */

private String openFile()
{
    String nomeArq;
    File [] arquivos;
    JFileChooser fc = new JFileChooser();
    fc.setMultiSelectionEnabled(true);
    int opcao = fc.showOpenDialog(this);
}

```

```

if (opcao == JFileChooser.APPROVE_OPTION) {
    System.out.println("Arquivo selecionado: " + fc.getSelectedFile().getAbsolutePath());
    //se eu ler mais de um arquivo:
    arquivos=fc.getSelectedFiles();
    //contando o num de arquivos lidos:
    int num=arquivos.length;
    // System.out.println("num de arquivos==== "+num+"   file [1]+arquivos[0]+ " file [2]" + arquivos[1]);
    //nomeArq=fc.getSelectedFile().getName();
    nomeArq=fc.getSelectedFile().getAbsolutePath();
}
else{ nomeArq="";}
return nomeArq;
}

//=====================================================MATH FuNC
public static float bp(String f)
{
    float soma = 0.0F;
    for(int i = 1; i < f.length(); i++)
    {
        String dig = f.substring(i - 1, i);
        int ddig = Integer.parseInt(dig);
        if(ddig == 1)
        {
            double ii = Math.pow(2D, -i);
            soma += (float)ii;
        }
    }
    return soma;
}

public static float f_32(String bin_str)
{
    String ss = bin_str.substring(0, 1);
    int s = Integer.parseInt(ss);
    String e = bin_str.substring(1, 9);
    String f = bin_str.substring(9, 32);
    int ed = Integer.parseInt(e, 2);
    int fd = Integer.parseInt(f, 2);
    float ff = 0.0F;
    if(ed > 0 && ed < 255)
        ff = (float)(Math.pow(-1D, s) * Math.pow(2D, (ed - 127)) *( 1 + bp(f)));
    else
    if(ed == 0 && fd != 0)
        ff = (float)(Math.pow(-1D, s) * (Math.pow(2D, -126D)) * (0 + bp(f)));
    else
        ff = 0.0F;
    return ff;
}

public static String zerobin(int n_int)
{
    String n_bin = Integer.toBinaryString(n_int);
    int comp = n_bin.length();
    if(comp < 8){
        switch(comp)
        {
            {
                case 1: // "\001"
                    n_bin = "0000000"+n_bin;
                    break;

                case 2: // "\002"
                    n_bin = "000000"+n_bin;
                    break;

                case 3: // "\003"
                    n_bin = "00000"+n_bin;
                    break;

                case 4: // "\004"
                    n_bin = "0000"+n_bin;
                    break;
            }
        }
    }
}

```

```

        case 5: // '\005'
            n_bin = "000"+n_bin;
            break;

        case 6: // '\006'
            n_bin = "00"+n_bin;
            break;

        case 7: // '\007'
            n_bin = "0"+n_bin;
            break;
        default:
            //8
            break;
    }
}
return n_bin;
}

public Vector readSPC (String nfile)
{
    Vector dadosSPC = new Vector();
    //preciso recuperar a info sobre o tamanho do file
    int nn=4097;
    String spc;
    int tam = nfile.length();
    spc= nfile.replaceAll("par", "spc");
    //System.out.println(spc);
    //System.out.println(nfile);

    int count=0;
    try{
        FileInputStream file= new FileInputStream(spc);
        boolean eof=false;
        while (!eof){
            int d1=file.read();
            int d2=file.read();
            int d3=file.read();
            int d4=file.read();
            String b1=zerobin(d1);
            String b2=zerobin(d2);
            String b3=zerobin(d3);
            String b4=zerobin(d4);
            String b=b4+b3+b2+b1;
            float num=f_32(b);
            //System.out.print(count+" "+num+" ..... ");
            ObDataNum dd=new ObDataNum();
            if (count>4096){eof=true;}
            //CARREGANDO O contador e o valor em DD
            Double cc;
            cc= new Double(count);
            dd.setCount(cc.doubleValue());
            dd.setNum(num);
            //ADD NA LISTA dados
            dadosSPC.add(dd);

            if(d4 == -1)
                eof = true;
            else
                count++;
        } //while
        file.close();
        //System.out.println("\nBytes read: " + count);
    } //try
    catch(IOException e){
        System.out.println("Error -- " + e.toString());
    } //catch
    return dadosSPC;
}

```

```

//=====

/** Exit the Application */
private void exitForm(java.awt.event.WindowEvent evt) {
System.exit(0);
}

////////////////////////////////////
public static void main(String[] args)
{
    EprMdi frame=new EprMdi();
    //
    frame.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });

    frame.setVisible(true);
    //System.out.println("MDI");
}

/*
 * espectro.java
 *
 * Created on 10 de Março de 2003, 16:50
 */

/**
 *
 * @author Orlando
 */
import java.util.Vector;

public class espectro {
    Vector dadosSPC;
    Vector dadosPAR;
    String nomeArq;

    /** Creates a new instance of espectro */
    public espectro() {
    }

}

Frame Interno

/**
 * @author Orlando
 *
 */
import javax.swing.JInternalFrame;
import java.awt.event.*;
import java.awt.*;

public class FrameInterno extends JInternalFrame {
    static int openFrameCount = 0;
    static final int xOffset = 30, yOffset = 30;

    public FrameInterno() {
        super("Spectro " + (++openFrameCount),
            true, //resizable
            true, //closable
            true, //maximizable
            true); //iconifiable

        //...Se for criar a GUI. coloco aqui

        //...tamanho da Janela
        setSize(300,300);
}

```

```

        //Posição da Janela
        setLocation(xOffset*openFrameCount, yOffset*openFrameCount);
        //getContentPane().add(new Grafico());
        //System.out.println("dentro do MYFRAME mdi");
    }
}

```

Rotina para ler dados Tokenizados.

```

/*
 * LendoDados.java
 *
 * Created on 23 de Fevereiro de 2003, 21:46
 */
/**
 *
 *
 * @author Orlando
 */

import java.io.*;
import java.util.*;
import java.io.StreamTokenizer;
import java.util.Vector;
import java.io.FileReader;
import java.io.IOException;

public class LendoDados {
    Vector dados =new Vector();

    public Vector Leitura( String nomeArq) {
    try {
        String fromFile = nomeArq; //nome do arquivo de dados
        BufferedReader br = new BufferedReader(new FileReader(fromFile));
        System.out.println("arquivo repassado ==="+nomeArq);
        for(String line = br.readLine(); line != null; line = br.readLine())
        {
            StringTokenizer st = new StringTokenizer(line);
            ObData dd=new ObData();
            //CARREGANDO OS DOIS TOKENS NO OBJETO DD
            dd.setParam(st.nextToken());
            dd.setValor(st.nextToken()); //lendo outro token na mesma linha
            //ADD NA LISTA dados
            dados.add(dd);
        }
    } catch(FileNotFoundException e) {
        } catch (IOException e) {
        }
    }

    System.out.println("dados"+ dados.size());
    //uma busca da string JRE no compo PARAM
    //se fosse um arquivo muito longo usaria um HashMap em lugar
    //de Vector
    String busca = "ANZ";
    boolean encontrado = false;
    for (int j = 0; j < dados.size(); j++) {
        System.out.println(" j="+j);
        ObData p = (ObData) dados.elementAt(j);
        System.out.println("param="+p.getParam());
        if (p.getParam().equals(busca))
        {
            System.out.println("Parâmetro:"+p.getParam());
            System.out.println("Dados:"+p.getValor());
            encontrado = true;
            break;
        }
    }
    if (! encontrado) {
        System.out.println("Não Encontrado!");
    }
    return dados;
}

```

```
}

```

```
}

```

Classes para encapsular os dados:

```
/*
 * WorkBase.java
 *
 * Created on 14 de Março de 2003, 10:41
 */

/**
 *
 * @author Orlando
 */
public class WorkBase {

    private String name;          //nome do WorkBase
    private String IDWB;         // um ID para o WB, ainda preciso definir
    private String detectorType; //tipo do detector
    private int detectorPP;      //num de detectores por ponto

    /** Creates a new instance of WorkBase */
    //Construtor
    public WorkBase(String n, String Id,String type) {

        name=n;
        IDWB=Id;
        detectorType=type;
    }

    //métodos para manipulação dos objetos
    public int getdetectorPP(){
        return this.detectorPP;
    };
    public String getname(){
        return this.name;
    };
    public String getIDWB(){
        return this.IDWB;
    };
    public String getdetectorType(){
        return this.detectorType;
    }
    //
    public void setdetectorPP(int pp){
        this.detectorPP=pp;
    };
}

```

```
import java.lang.String;
/**
 * @author Orlando
 *
 */
public class ObData {
    // um objeto instanciado dessa classe
    // guarda o tipo de parâmetro e o seu valor
    private String param;
    private String valor;

    /**
     * Constructor for ObData.
     */
    public ObData(String p, String v) {
        //no construtor são setados os valores
        this.param=p;
    }
}

```

```
        this.valor=v;
    }
    public ObData(){
        this.param="";
        this.valor="";
    }

    void setParam(String pp) {
        /**
         * Method setParam
         * string
         */

        this.param=pp;
    }

    void setValor(String vv) {
        this.valor=vv;
    }

    String getParam() {
        return this.param;
    }
    String getValor() {
        return this.valor;
    }
}

/**
 * ObDataNum.java
 *
 * Created on 26 de Fevereiro de 2003, 10:49
 */

import java.lang.String;
/**
 * @author Orlando
 */
public class ObDataNum {
    // um objeto instanciado dessa classe
    // guarda o count e o num dos arquivos spc
    private double count;
    private double num;

    /**
     * Constructor for ObData.
     */
    public ObDataNum(double c, double n) {
        //no construtor são setados os valores
        this.count=c;
        this.num=n;
    }
    public ObDataNum(){
        this.count=0;
        this.num=0;
    }

    public void setCount(double cc) {
        this.count=cc;
    }

    public void setNum(double nn) {
        this.num=nn;
    }

    public double getCount() {
        return this.count;
    }
    public double getNum() {
        return this.num;
    }
}
```

```

}

Parte da Interface (VERSAO B1)
/*
 * planR.java
 *
 * Created on 11 de Agosto de 2003, 15:46
 */

/**
 *
 * @author rodrigues
 */
public class planR extends javax.swing.JInternalFrame {

    /** Creates new form planR */
    public planR() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    private void initComponents() { //GEN-BEGIN: initComponents
        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        jTextField2 = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        jTextField3 = new javax.swing.JTextField();
        jPanel4 = new javax.swing.JPanel();
        jButton3 = new javax.swing.JButton();
        jButton4 = new javax.swing.JButton();
        jPanel2 = new javax.swing.JPanel();
        jLabel4 = new javax.swing.JLabel();
        jComboBox1 = new javax.swing.JComboBox();
        jLabel5 = new javax.swing.JLabel();
        jComboBox2 = new javax.swing.JComboBox();
        jPanel3 = new javax.swing.JPanel();
        jLabel6 = new javax.swing.JLabel();
        jTextField4 = new javax.swing.JTextField();

        setTitle("Planning - Routine Dosimetry");
        setFrameIcon(new
javax.swing.ImageIcon("C:\\j2sdk1.4.0_03\\icons\\toolbarButtonGraphics\\development\\Application24.gif"));
        jPanel1.setBorder(new javax.swing.border.TitledBorder("General"));
        jPanel1.setName("General");
        jLabel1.setText("Name:");
        jPanel1.add(jLabel1);

        jTextField1.setColumns(3);
        jTextField1.setMinimumSize(new java.awt.Dimension(200, 20));
        jTextField1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField1ActionPerformed(evt);
            }
        });

        jPanel1.add(jTextField1);

        jLabel2.setText("Date:");
        jPanel1.add(jLabel2);

        jTextField2.setColumns(10);
        jTextField2.setText("dd/mm/aaaa");
        jPanel1.add(jTextField2);

        jLabel3.setText("Detector Type:");
        jPanel1.add(jLabel3);
    }
}

```

```

jTextField3.setColumns(10);
jPanel1.add(jTextField3);

jPanel4.setBorder(new javax.swing.border.TitledBorder("Dose Range:"));
jRadioButton3.setText("Industrial Application (10-100kGy)");
jPanel4.add(jRadioButton3);

jRadioButton4.setText("Radiotherapy (0.2-10)Gy");
jRadioButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton4ActionPerformed(evt);
    }
});

jPanel4.add(jRadioButton4);

jPanel1.add(jPanel4);

getContentPane().add(jPanel1, java.awt.BorderLayout.NORTH);

jPanel2.setBorder(new javax.swing.border.TitledBorder("Calibration Curve"));
jLabel4.setText("Curve Points:");
jPanel2.add(jLabel4);

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "2", "3", "4", "5", "6", "7", "8", "9", "10"
}));
jPanel2.add(jComboBox1);

jLabel5.setText("Dosimeters per Point:");
jPanel2.add(jLabel5);

jComboBox2.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "2", "3", "4", "5" }));
jPanel2.add(jComboBox2);

getContentPane().add(jPanel2, java.awt.BorderLayout.CENTER);

jPanel3.setLayout(new java.awt.FlowLayout(java.awt.FlowLayout.LEFT));

jPanel3.setBorder(new javax.swing.border.TitledBorder("Irradiation System"));
jLabel6.setText(" Informations:");
jPanel3.add(jLabel6);

jTextField4.setColumns(40);
jPanel3.add(jTextField4);

getContentPane().add(jPanel3, java.awt.BorderLayout.SOUTH);

pack();
} //GEN-END: initComponents

private void jRadioButton4ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jRadioButton4ActionPerformed
    // Add your handling code here:
} //GEN-LAST:event_jRadioButton4ActionPerformed

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jTextField1ActionPerformed
    // Add your handling code here:
} //GEN-LAST:event_jTextField1ActionPerformed

// Variables declaration - do not modify //GEN-BEGIN:variables
private javax.swing.JComboBox jComboBox1;
private javax.swing.JComboBox jComboBox2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;

```

```

private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JRadioButton jButton3;
private javax.swing.JRadioButton jButton4;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
// End of variables declaration//GEN-END:variables
}

/*
 * Tela1.java
 *
 * Created on 14 de Março de 2003, 16:24
 */

/**
 *
 * @author Orlando
 */

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.JComponent;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.io.*;
import java.util.*;
import java.lang.Object;

public class Tela1 extends JFrame {
    JLabel label_1;
    JTextField textfield_1;
    JLabel label_2;
    JTextField textfield_3;
    JLabel label_3;
    ButtonGroup cbg;
    JRadioButton radio_1;
    JRadioButton radio_2;
    static String b1 = "Bird";
    static String b2 = "Cat";
    JLabel label_4;
    JTextField textfield_2;
    JLabel label_5;
    JTextField textfield_4;
    JLabel label_6;
    JList list_1;
    JScrollPane sp_list_1;
    Grafico panel_2;
    JButton b_ok;
    JButton b_cancel;

    public Tela1() {
        super("Spectro ",
            true, //resizable
            true, //closable
            true, //maximizable
            true); //iconifiable
        Tela1Layout customLayout = new Tela1Layout();

        getContentPane().setFont(new Font("Helvetica", Font.PLAIN, 12));
        getContentPane().setLayout(customLayout);

        label_1 = new JLabel("Name");
        getContentPane().add(label_1);

        textfield_1 = new JTextField("");
        getContentPane().add(textfield_1);

        label_2 = new JLabel("Detector Type");

```

```

getContentPane().add(label_2);

textfield_3 = new JTextField("");
getContentPane().add(textfield_3);

label_3 = new JLabel("Dose Range");
getContentPane().add(label_3);

cbg = new ButtonGroup();
radio_1 = new JRadioButton("Industrial Application (10-100kGy", true);
radio_1.setActionCommand(b1);
cbg.add(radio_1);
getContentPane().add(radio_1);

radio_2 = new JRadioButton("Radiotherapy (0.1-10)Gy", false);
radio_2.setActionCommand(b2);
cbg.add(radio_2);
getContentPane().add(radio_2);

label_4 = new JLabel("Detector for Point");
getContentPane().add(label_4);

textfield_2 = new JTextField("1");
getContentPane().add(textfield_2);

label_5 = new JLabel("Dose Points");
getContentPane().add(label_5);

textfield_4 = new JTextField("0.1-10 ");
getContentPane().add(textfield_4);

label_6 = new JLabel("Doses");
getContentPane().add(label_6);

DefaultListModel listModel_list_1 = new DefaultListModel();
listModel_list_1.addElement("1");
listModel_list_1.addElement("2");
listModel_list_1.addElement("3");
list_1 = new JList(listModel_list_1);
sp_list_1 = new JScrollPane(list_1);
getContentPane().add(sp_list_1);

panel_2 = new Grafico();
getContentPane().add(panel_2);

b_ok = new JButton("OK");
getContentPane().add(b_ok);

b_cancel = new JButton("Cancel");
getContentPane().add(b_cancel);

setSize(getPreferredSize());
ActionListener aa =new ActionListener(){
public void actionPerformed(ActionEvent e)
{
//devo ler os valores preenchidos no form:
System.out.println("ok foi pressionado");
String nome=textfield_1.getText();
System.out.println("nome="+nome);
String id="xx-xx-xxxx";
String type=textfield_3.getText();
System.out.println("texto="+type);
// Criando um objeto para receber o WB
CalibrationCurve cc=new CalibrationCurve(nome,id,type);
}
};

```

```

        b_ok.addActionListener(aa);
        ActionListener ab =new ActionListener(){
            public void actionPerformed(ActionEvent e)
            {
                setVisible(false);
            }
        };
        b_cancel.addActionListener(ab);
    }
}

class Tela1Layout implements LayoutManager {

    public Tela1Layout() {
    }

    public void addLayoutComponent(String name, Component comp) {
    }

    public void removeLayoutComponent(Component comp) {
    }

    public Dimension preferredLayoutSize(Container parent) {
        Dimension dim = new Dimension(0, 0);

        Insets insets = parent.getInsets();
        dim.width = 334 + insets.left + insets.right;
        dim.height = 321 + insets.top + insets.bottom;

        return dim;
    }

    public Dimension minimumLayoutSize(Container parent) {
        Dimension dim = new Dimension(0, 0);
        return dim;
    }

    public void layoutContainer(Container parent) {
        Insets insets = parent.getInsets();

        Component c;
        c = parent.getComponent(0);
        if (c.isVisible()) {c.setBounds(insets.left+13,insets.top+19,122,19);}
        c = parent.getComponent(1);
        if (c.isVisible()) {c.setBounds(insets.left+141,insets.top+19,83,19);}
        c = parent.getComponent(2);
        if (c.isVisible()) {c.setBounds(insets.left+13,insets.top+45,122,19);}
        c = parent.getComponent(3);
        if (c.isVisible()) {c.setBounds(insets.left+141,insets.top+45,83,19);}
        c = parent.getComponent(4);
        if (c.isVisible()) {c.setBounds(insets.left+13,insets.top+70,70,19);}
        c = parent.getComponent(5);
        if (c.isVisible()) {c.setBounds(insets.left+90,insets.top+70,218,19);}
        c = parent.getComponent(6);
        if (c.isVisible()) {c.setBounds(insets.left+90,insets.top+96,218,19);}
        c = parent.getComponent(7);
        if (c.isVisible()) {c.setBounds(insets.left+13,insets.top+128,122,19);}
        c = parent.getComponent(8);
        if (c.isVisible()) {c.setBounds(insets.left+141,insets.top+128,70,19);}
        c = parent.getComponent(9);
        if (c.isVisible()) {c.setBounds(insets.left+13,insets.top+160,122,19);}
        c = parent.getComponent(10);
        if (c.isVisible()) {c.setBounds(insets.left+141,insets.top+160,70,19);}
        c = parent.getComponent(11);
        if (c.isVisible()) {c.setBounds(insets.left+77,insets.top+192,58,19);}
        c = parent.getComponent(12);
        if (c.isVisible()) {c.setBounds(insets.left+141,insets.top+192,70,64);}
        c = parent.getComponent(13);
        if (c.isVisible()) {c.setBounds(insets.left+224,insets.top+141,90,102);}
    }
}

```

```

        c = parent.getComponent(14);
        if (c.isVisible()) {c.setBounds(insets.left+58,insets.top+275,90,19);}
        c = parent.getComponent(15);
        if (c.isVisible()) {c.setBounds(insets.left+211,insets.top+275,90,19);}
    }
}

/*
 * PlotFrame.java
 *
 * Created on 22 de Fevereiro de 2003, 12:30
 */

/**
 *
 * @author Orlando
 */

import java.awt.*;
import javax.swing.*;
import java.util.Vector;
import java.io.*;
import java.util.*;

public class PlotFrame extends JFrame{

    Vector d;
    String file;

    /** Creates a new instance of PlotFrame */
    public PlotFrame()
    {
        this.setSize(600, 500);
        this.show();
        //this.setTitle("PlotFrame");
    }
    public void setDados (Vector dados, String nomeArq)
    {
        d=dados;
        file=nomeArq;
        int tam=d.size();
        double [] x=new double[tam];
        double [] y=new double[tam];
        System.out.println("Dentro de PLOTFRAME-setDados - dsize="+tam);
        for (int j = 0; j < tam; j++)
        {
            ObDataNum p = (ObDataNum) d.elementAt(j);
            x[j]= p.getCount();
            y[j]= p.getNum();
        }
        Plot canvas1;
        canvas1=new Plot();
        canvas1.setBounds(10,10, 500,400);
        canvas1.Plot(x,y,Color.black,"line");
        this.getContentPane().add(canvas1);
    }
}
}

```

Plotador (versão baseado no Livro ref. 62)

```
/*
 * Plot.java
 *
 * Created on 02 de Fevereiro de 2003, 09:19
 */

/**
 *
 * @author Orlando
 */
// Plotador
// 26-02-2003
//rev. 3.1 Orlando

import java.awt.*;
import java.io.PrintStream;
import java.util.Vector;

public class Plot extends Canvas
{

    protected int n_xticks;
    protected int n_yticks;
    protected double north_percent;
    protected double west_percent;
    protected double x_percent;
    protected double y_percent;
    protected double xmin;
    protected double xmax;
    protected double ymin;
    protected double ymax;
    protected Vector dataseries;
    protected Vector series_names;
    protected String title;
    protected String xlabel;
    protected String ylabel;
    protected boolean key;
    protected boolean zero_lines;
    protected boolean y_gridlines;
    protected boolean x_gridlines;
    protected boolean border;
    private Color color;
    private String type;
    private int west;
    private int north;
    private int xlength;
    private int ylength;
    private int n_series;
    private double tmp_xmax;
    private double tmp_ymax;
    private double tmp_xmin;
    private double tmp_ymin;
    private double xrange;
    private double yrange;
    private double xscale;
    private double yscale;
    private double x_series[];
    private double y_series[];
    private boolean cxmax;
    private boolean cymax;
    private boolean cxmin;
    private boolean cymin;

    Plot()
    {
        n_xticks = 5;
        n_yticks = 5;
        north_percent = 0.10000000000000001D;
        west_percent = 0.14999999999999999D;
    }
}
```

```

    x_percent = 0.8000000000000004D;
    y_percent = 0.6999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot();
}

protected void Plot()
{
    dataseries = new Vector();
}

protected void Clear()
{
    Plot();
}

protected void Plot(Vector v)
{
    dataseries = v;
    n_series = v.size() / 4;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(Vector v)
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.1000000000000001D;
    west_percent = 0.1499999999999999D;
    x_percent = 0.8000000000000004D;
    y_percent = 0.6999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(v);
}

Plot(double x0[], double y0[], Color c0, String t0)
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.1000000000000001D;
    west_percent = 0.1499999999999999D;
    x_percent = 0.8000000000000004D;
    y_percent = 0.6999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0, c0, t0);
}

protected void Plot(double x0[], double y0[], Color c0, String t0)
{
    dataseries = new Vector();
    dataseries.addElement(x0);
    dataseries.addElement(y0);
    dataseries.addElement(c0);
    dataseries.addElement(t0);
    n_series = 1;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(double x0[], double y0[], Color c0)
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.1000000000000001D;
    west_percent = 0.1499999999999999D;
    x_percent = 0.8000000000000004D;
    y_percent = 0.6999999999999996D;
    title = "";

```

```

        xlabel = "";
        ylabel = "";
        Plot(x0, y0, c0);
    }

protected void Plot(double x0[], double y0[], Color c0)
{
    dataserie = new Vector();
    dataserie.addElement(x0);
    dataserie.addElement(y0);
    dataserie.addElement(c0);
    dataserie.addElement("squares");
    n_series = 1;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(double x0[], double y0[])
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.10000000000000001D;
    west_percent = 0.14999999999999999D;
    x_percent = 0.80000000000000004D;
    y_percent = 0.69999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0);
}

protected void Plot(double x0[], double y0[])
{
    dataserie = new Vector();
    dataserie.addElement(x0);
    dataserie.addElement(y0);
    dataserie.addElement(Color.black);
    dataserie.addElement("squares");
    n_series = 1;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(int x0[], int y0[], Color c0, String t0)
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.10000000000000001D;
    west_percent = 0.14999999999999999D;
    x_percent = 0.80000000000000004D;
    y_percent = 0.69999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0, c0, t0);
}

protected void Plot(int x0[], int y0[], Color c0, String t0)
{
    double xd[] = castToDouble(x0);
    double yd[] = castToDouble(y0);
    dataserie.addElement(xd);
    dataserie.addElement(yd);
    dataserie.addElement(c0);
    dataserie.addElement(t0);
    n_series = 1;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(int x0[], int y0[], Color c0)
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.10000000000000001D;
    west_percent = 0.14999999999999999D;

```

```

    x_percent = 0.80000000000000004D;
    y_percent = 0.69999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0, c0);
}

protected void Plot(int x0[], int y0[], Color c0)
{
    double xd[] = castToDouble(x0);
    double yd[] = castToDouble(y0);
    dataseries.addElement(xd);
    dataseries.addElement(yd);
    dataseries.addElement(c0);
    dataseries.addElement("squares");
    n_series = 1;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(int x0[], int y0[])
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.10000000000000001D;
    west_percent = 0.14999999999999999D;
    x_percent = 0.80000000000000004D;
    y_percent = 0.69999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0);
}

protected void Plot(int x0[], int y0[])
{
    double xd[] = castToDouble(x0);
    double yd[] = castToDouble(y0);
    dataseries.addElement(xd);
    dataseries.addElement(yd);
    dataseries.addElement(Color.black);
    dataseries.addElement("squares");
    n_series = 1;
    setFont(new Font("Helvetica", 0, 9));
}

public double[] castToDouble(int a[])
{
    int n = a.length;
    double ad[] = new double[n];
    for(int i = 0; i < n; i++)
        ad[i] = Math.round(a[i]);

    return ad;
}

Plot(double x0[], double y0[], Color c0, String t0, double x1[], double y1[], Color c1,
      String t1)
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.10000000000000001D;
    west_percent = 0.14999999999999999D;
    x_percent = 0.80000000000000004D;
    y_percent = 0.69999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0, c0, t0, x1, y1, c1, t1);
}

protected void Plot(double x0[], double y0[], Color c0, String t0, double x1[], double y1[], Color c1,
                    String t1)

```

```

{
    dataseries.addElement(x0);
    dataseries.addElement(y0);
    dataseries.addElement(c0);
    dataseries.addElement(t0);
    dataseries.addElement(x1);
    dataseries.addElement(y1);
    dataseries.addElement(c1);
    dataseries.addElement(t1);
    n_series = 2;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(double x0[], double y0[], Color c0, double x1[], double y1[], Color c1)
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.10000000000000001D;
    west_percent = 0.14999999999999999D;
    x_percent = 0.80000000000000004D;
    y_percent = 0.6999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0, c0, x1, y1, c1);
}

protected void Plot(double x0[], double y0[], Color c0, double x1[], double y1[], Color c1)
{
    dataseries.addElement(x0);
    dataseries.addElement(y0);
    dataseries.addElement(c0);
    dataseries.addElement("squares");
    dataseries.addElement(x1);
    dataseries.addElement(y1);
    dataseries.addElement(c1);
    dataseries.addElement("squares");
    n_series = 2;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(double x0[], double y0[], double x1[], double y1[])
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.10000000000000001D;
    west_percent = 0.14999999999999999D;
    x_percent = 0.80000000000000004D;
    y_percent = 0.6999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0, x1, y1);
}

protected void Plot(double x0[], double y0[], double x1[], double y1[])
{
    dataseries.addElement(x0);
    dataseries.addElement(y0);
    dataseries.addElement(Color.black);
    dataseries.addElement("squares");
    dataseries.addElement(x1);
    dataseries.addElement(y1);
    dataseries.addElement(Color.black);
    dataseries.addElement("squares");
    n_series = 2;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(int x0[], int y0[], Color c0, String t0, int x1[], int y1[], Color c1,
     String t1)
{
    n_xticks = 5;

```

```

n_yticks = 5;
north_percent = 0.10000000000000001D;
west_percent = 0.14999999999999999D;
x_percent = 0.80000000000000004D;
y_percent = 0.69999999999999996D;
title = "";
xlabel = "";
ylabel = "";
Plot(x0, y0, c0, t0, x1, y1, c1, t1);
}

protected void Plot(int x0[], int y0[], Color c0, String t0, int x1[], int y1[], Color c1,
String t1)
{
double xd0[] = castToDouble(x0);
double yd0[] = castToDouble(y0);
double xd1[] = castToDouble(x1);
double yd1[] = castToDouble(y1);
dataseries.addElement(xd0);
dataseries.addElement(yd0);
dataseries.addElement(c0);
dataseries.addElement(t0);
dataseries.addElement(xd1);
dataseries.addElement(yd1);
dataseries.addElement(c1);
dataseries.addElement(t1);
n_series = 2;
setFont(new Font("Helvetica", 0, 9));
}

Plot(int x0[], int y0[], Color c0, int x1[], int y1[], Color c1)
{
n_xticks = 5;
n_yticks = 5;
north_percent = 0.10000000000000001D;
west_percent = 0.14999999999999999D;
x_percent = 0.80000000000000004D;
y_percent = 0.69999999999999996D;
title = "";
xlabel = "";
ylabel = "";
Plot(x0, y0, c0, x1, y1, c1);
}

protected void Plot(int x0[], int y0[], Color c0, int x1[], int y1[], Color c1)
{
castToDouble(x0);
castToDouble(y0);
castToDouble(x1);
castToDouble(y1);
dataseries.addElement(x0);
dataseries.addElement(y0);
dataseries.addElement(c0);
dataseries.addElement("squares");
dataseries.addElement(x1);
dataseries.addElement(y1);
dataseries.addElement(c1);
dataseries.addElement("squares");
n_series = 2;
setFont(new Font("Helvetica", 0, 9));
}

Plot(int x0[], int y0[], int x1[], int y1[])
{
n_xticks = 5;
n_yticks = 5;
north_percent = 0.10000000000000001D;
west_percent = 0.14999999999999999D;
x_percent = 0.80000000000000004D;
y_percent = 0.69999999999999996D;
title = "";
xlabel = "";
ylabel = "";

```

```

    Plot(x0, y0, x1, y1);
}

protected void Plot(int x0[], int y0[], int x1[], int y1[])
{
    castToDouble(x0);
    castToDouble(y0);
    castToDouble(x1);
    castToDouble(y1);
    dataseries.addElement(x0);
    dataseries.addElement(y0);
    dataseries.addElement(Color.black);
    dataseries.addElement("squares");
    dataseries.addElement(x1);
    dataseries.addElement(y1);
    dataseries.addElement(Color.black);
    dataseries.addElement("squares");
    n_series = 2;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(double x0[], double y0[], Color c0, String t0, double x1[], double y1[], Color c1,
      String t1, double x2[], double y2[], Color c2, String t2)
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.10000000000000001D;
    west_percent = 0.14999999999999999D;
    x_percent = 0.80000000000000004D;
    y_percent = 0.69999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0, c0, t0, x1, y1, c1, t1, x2, y2, c2, t2);
}

protected void Plot(double x0[], double y0[], Color c0, String t0, double x1[], double y1[], Color c1,
      String t1, double x2[], double y2[], Color c2, String t2)
{
    dataseries.addElement(x0);
    dataseries.addElement(y0);
    dataseries.addElement(c0);
    dataseries.addElement(t0);
    dataseries.addElement(x1);
    dataseries.addElement(y1);
    dataseries.addElement(c1);
    dataseries.addElement(t1);
    dataseries.addElement(x2);
    dataseries.addElement(y2);
    dataseries.addElement(c2);
    dataseries.addElement(t2);
    n_series = 3;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(double x0[], double y0[], Color c0, double x1[], double y1[], Color c1, double x2[],
      double y2[], Color c2)
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.10000000000000001D;
    west_percent = 0.14999999999999999D;
    x_percent = 0.80000000000000004D;
    y_percent = 0.69999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0, c0, x1, y1, c1, x2, y2, c2);
}

protected void Plot(double x0[], double y0[], Color c0, double x1[], double y1[], Color c1, double x2[],
      double y2[], Color c2)
{

```

```

    dataseries.addElement(x0);
    dataseries.addElement(y0);
    dataseries.addElement(c0);
    dataseries.addElement("squares");
    dataseries.addElement(x1);
    dataseries.addElement(y1);
    dataseries.addElement(c1);
    dataseries.addElement("squares");
    dataseries.addElement(x2);
    dataseries.addElement(y2);
    dataseries.addElement(c2);
    dataseries.addElement("squares");
    n_series = 3;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(double x0[], double y0[], double x1[], double y1[], double x2[], double y2[])
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.10000000000000001D;
    west_percent = 0.14999999999999999D;
    x_percent = 0.80000000000000004D;
    y_percent = 0.69999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0, x1, y1, x2, y2);
}

protected void Plot(double x0[], double y0[], double x1[], double y1[], double x2[], double y2[])
{
    dataseries.addElement(x0);
    dataseries.addElement(y0);
    dataseries.addElement(Color.black);
    dataseries.addElement("squares");
    dataseries.addElement(x1);
    dataseries.addElement(y1);
    dataseries.addElement(Color.black);
    dataseries.addElement("squares");
    dataseries.addElement(x2);
    dataseries.addElement(y2);
    dataseries.addElement(Color.black);
    dataseries.addElement("squares");
    n_series = 3;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(int x0[], int y0[], Color c0, String t0, int x1[], int y1[], Color c1,
      String t1, int x2[], int y2[], Color c2, String t2)
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.10000000000000001D;
    west_percent = 0.14999999999999999D;
    x_percent = 0.80000000000000004D;
    y_percent = 0.69999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0, c0, t0, x1, y1, c1, t1, x2, y2, c2, t2);
}

protected void Plot(int x0[], int y0[], Color c0, String t0, int x1[], int y1[], Color c1,
                    String t1, int x2[], int y2[], Color c2, String t2)
{
    double xd0[] = castToDouble(x0);
    double yd0[] = castToDouble(y0);
    double xd1[] = castToDouble(x1);
    double yd1[] = castToDouble(y1);
    double xd2[] = castToDouble(x2);
    double yd2[] = castToDouble(y2);
    dataseries.addElement(xd0);

```

```

    dataseries.addElement(yd0);
    dataseries.addElement(c0);
    dataseries.addElement(t0);
    dataseries.addElement(xd1);
    dataseries.addElement(yd1);
    dataseries.addElement(c1);
    dataseries.addElement(t1);
    dataseries.addElement(xd2);
    dataseries.addElement(yd2);
    dataseries.addElement(c2);
    dataseries.addElement(t2);
    n_series = 3;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(int x0[], int y0[], Color c0, int x1[], int y1[], Color c1, int x2[],
     int y2[], Color c2)
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.10000000000000001D;
    west_percent = 0.14999999999999999D;
    x_percent = 0.80000000000000004D;
    y_percent = 0.69999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0, c0, x1, y1, c1, x2, y2, c2);
}

protected void Plot(int x0[], int y0[], Color c0, int x1[], int y1[], Color c1, int x2[],
                    int y2[], Color c2)
{
    castToDouble(x0);
    castToDouble(y0);
    castToDouble(x1);
    castToDouble(y1);
    castToDouble(x2);
    castToDouble(y2);
    dataseries.addElement(x0);
    dataseries.addElement(y0);
    dataseries.addElement(c0);
    dataseries.addElement("squares");
    dataseries.addElement(x1);
    dataseries.addElement(y1);
    dataseries.addElement(c1);
    dataseries.addElement("squares");
    dataseries.addElement(x2);
    dataseries.addElement(y2);
    dataseries.addElement(c2);
    dataseries.addElement("squares");
    n_series = 3;
    setFont(new Font("Helvetica", 0, 9));
}

Plot(int x0[], int y0[], int x1[], int y1[], int x2[], int y2[])
{
    n_xticks = 5;
    n_yticks = 5;
    north_percent = 0.10000000000000001D;
    west_percent = 0.14999999999999999D;
    x_percent = 0.80000000000000004D;
    y_percent = 0.69999999999999996D;
    title = "";
    xlabel = "";
    ylabel = "";
    Plot(x0, y0, x1, y1, x2, y2);
}

protected void Plot(int x0[], int y0[], int x1[], int y1[], int x2[], int y2[])
{
    castToDouble(x0);
    castToDouble(y0);

```

```
cast ToDouble(x1);
cast ToDouble(y1);
cast ToDouble(x2);
cast ToDouble(y2);
dataseries.addElement(x0);
dataseries.addElement(y0);
dataseries.addElement(Color.black);
dataseries.addElement("squares");
dataseries.addElement(x1);
dataseries.addElement(y1);
dataseries.addElement(Color.black);
dataseries.addElement("squares");
dataseries.addElement(x2);
dataseries.addElement(y2);
dataseries.addElement(Color.black);
dataseries.addElement("squares");
n_series = 3;
setFont(new Font("Helvetica", 0, 9));
}

protected void test()
{
}

protected double max(double x[])
{
    double mx = x[0];
    for(int i = 1; i < x.length; i++)
        if(x[i] > mx)
            mx = x[i];

    return mx;
}

protected double min(double x[])
{
    double mn = x[0];
    for(int i = 1; i < x.length; i++)
        if(x[i] < mn)
            mn = x[i];

    return mn;
}

protected void setName(int n, String name)
{
    try
    {
        if(series_names == null)
        {
            series_names = new Vector();
            for(int i = 0; i < n_series; i++)
                series_names.addElement("");
        }
        series_names.setElementAt(name, n);
        return;
    }
    catch(Exception _ex)
    {
        return;
    }
}

protected void addName(String name)
{
    try
    {
        if(series_names == null)
        {
            series_names = new Vector();
            for(int i = 0; i < n_series; i++)
                series_names.addElement("");
        }
    }
}
```

```
    }
    series_names.addElement(name);
    return;
}
catch(Exception _ex)
{
    return;
}
}

protected void setSeriesNames(String names[])
{
    try
    {
        series_names = new Vector();
        for(int i = 0; i < names.length; i++)
            series_names.addElement(names[i]);

        return;
    }
    catch(Exception e)
    {
        System.err.println(e);
    }
}

protected void setSeriesNames(Vector names)
{
    series_names = names;
}

protected void setColor(int n, Color c)
{
    try
    {
        dataseries.setElementAt(c, n * 4 + 2);
        return;
    }
    catch(Exception _ex)
    {
        return;
    }
}

protected void setColor(Color c[])
{
    try
    {
        for(int i = 0; i < c.length; i++)
            dataseries.setElementAt(c[i], i * 4 + 2);

        return;
    }
    catch(Exception _ex)
    {
        return;
    }
}

protected void setColor(Vector c)
{
    try
    {
        for(int i = 0; i < c.size(); i++)
            dataseries.setElementAt(c.elementAt(i), i * 4 + 2);

        return;
    }
    catch(Exception _ex)
    {
        return;
    }
}
```

```
}

protected void setType(int n, String t)
{
    try
    {
        dataseries.setElementAt(t, n * 4 + 3);
        return;
    }
    catch(Exception _ex)
    {
        return;
    }
}

protected void setType(String t[])
{
    try
    {
        for(int i = 0; i < t.length; i++)
            dataseries.setElementAt(t[i], i * 4 + 3);

        return;
    }
    catch(Exception _ex)
    {
        return;
    }
}

protected void setType(Vector t)
{
    try
    {
        for(int i = 0; i < t.size(); i++)
            dataseries.setElementAt(t.elementAt(i), i * 4 + 3);

        return;
    }
    catch(Exception _ex)
    {
        return;
    }
}

protected void calculateDimensions()
{
    if(x_percent < 0.10000000000000001D)
        x_percent = 0.10000000000000001D;
    if(y_percent < 0.10000000000000001D)
        y_percent = 0.10000000000000001D;
    if(north_percent > 1.0D)
        north_percent = 0.90000000000000002D;
    if(west_percent > 1.0D)
        west_percent = 0.90000000000000002D;
    xlength = (int)Math.round(x_percent * (double)size().width);
    ylength = (int)Math.round(y_percent * (double)size().height);
    west = size().width - xlength - (int)Math.round((1.0D - x_percent - west_percent) * (double)size().width);
    north = size().height - ylength - (int)Math.round((1.0D - y_percent - north_percent) * (double)size().height);
}

protected void setXMax(double xmax)
{
    this.xmax = xmax;
    cxmax = true;
}

protected void calculateXMax()
{
    cxmax = false;
}

protected void setXMin(double xmin)
```

```

    {
        this.xmin = xmin;
        cxmin = true;
    }

    protected void calculateXMin()
    {
        cxmin = false;
    }

    protected void setYMax(double ymax)
    {
        this.ymax = ymax;
        cymax = true;
    }

    protected void calculateYMax()
    {
        cymax = false;
    }

    protected void setYMin(double ymin)
    {
        this.ymin = ymin;
        cymin = true;
    }

    protected void calculateYMin()
    {
        cymin = false;
    }

    protected void useDefaultnames()
    {
        series_names = new Vector();
        for(int i = 0; i < n_series; i++)
            series_names.addElement("Series ".concat(String.valueOf(String.valueOf(String.valueOf(i + 1)))));
    }

    protected void fillInDefaults()
    {
        for(int i = series_names.size(); i < n_series; i++)
            series_names.addElement("Series ".concat(String.valueOf(String.valueOf(String.valueOf(i + 1)))));
    }

    public void update(Graphics g)
    {
        paint(g);
    }

    public void paint(Graphics g)
    {
        g.setColor(getBackground());
        g.fillRect(0, 0, size().width, size().height);
        if(key)
        {
            x_percent = 0.55000000000000004D;
            north_percent = 0.10000000000000001D;
            west_percent = 0.14999999999999999D;
            y_percent = 0.69999999999999996D;
        }
        calculateDimensions();
        g.setColor(getForeground());
        g.setFont(getFont());
        FontMetrics fm = getFontMetrics(getFont());
        if(border)
            g.drawRect(0, 0, size().width - 1, size().height - 1);
        g.drawRect(west, north, xlength, ylength);
        g.drawRect(west + 1, north + 1, xlength - 2, ylength - 2);
        g.drawRect(west - 1, north - 1, xlength + 2, ylength + 2);
        if(dataseries.size() == 0)
    }

```

```

    return;
if(series_names == null)
    useDefaultNames();
if(series_names.size() < n_series)
    fillInDefaults();
g.setColor(getForeground());
tmp_xmax = xmax;
tmp_ymax = ymax;
tmp_xmin = xmin;
tmp_ymin = ymin;
xmax = xmin;
ymax = ymin;
for(int seriesindex = 0; seriesindex < n_series * 4; seriesindex += 4)
{
    try
    {
        x_series = (double[])dataseries.elementAt(seriesindex);
        y_series = (double[])dataseries.elementAt(seriesindex + 1);
    }
    catch(ClassCastException _ex)
    {
        x_series = castToDouble((int[])dataseries.elementAt(seriesindex));
        y_series = castToDouble((int[])dataseries.elementAt(seriesindex + 1));
    }
    if(max(x_series) > xmax)
        xmax = max(x_series);
    if(max(y_series) > ymax)
        ymax = max(y_series);
}

ymin = ymax;
xmin = xmax;
for(int seriesindex = 0; seriesindex < n_series * 4; seriesindex += 4)
{
    try
    {
        x_series = (double[])dataseries.elementAt(seriesindex);
        y_series = (double[])dataseries.elementAt(seriesindex + 1);
    }
    catch(ClassCastException _ex)
    {
        x_series = castToDouble((int[])dataseries.elementAt(seriesindex));
        y_series = castToDouble((int[])dataseries.elementAt(seriesindex + 1));
    }
    if(min(x_series) < xmin)
        xmin = min(x_series);
    if(min(y_series) < ymin)
        ymin = min(y_series);
}

xmax = xmax + xmax * 0.050000000000000003D;
xmin = xmin - xmax * 0.050000000000000003D;
ymax = ymax + ymax * 0.050000000000000003D;
ymin = ymin - ymax * 0.050000000000000003D;
if(cxmax)
    xmax = tmp_xmax;
if(cxmin)
    xmin = tmp_xmin;
if(cymax)
    ymax = tmp_ymax;
if(cymin)
    ymin = tmp_ymin;
yrange = ymax - ymin;
xrange = xmax - xmin;
yscale = (double)ylength / yrange;
xscale = (double)xlength / xrange;
label0:
for(int seriesindex = 0; seriesindex < n_series * 4; seriesindex += 4)
{
    try
    {
        x_series = (double[])dataseries.elementAt(seriesindex);
        y_series = (double[])dataseries.elementAt(seriesindex + 1);
    }
}

```

```

}
catch(ClassCastException _ex)
{
    x_series = castToDouble((int[])dataseries.elementAt(seriesindex));
    y_series = castToDouble((int[])dataseries.elementAt(seriesindex + 1));
}
color = (Color)dataseries.elementAt(seriesindex + 2);
type = (String)dataseries.elementAt(seriesindex + 3);
g.setColor(color);
int step = 1;
if(x_series == null)
    return;
if(x_series.length > xlength)
    step = x_series.length / xlength;
if(type.equalsIgnoreCase("circles"))
{
    int i = 0;
    do
    {
        if(i >= x_series.length)
            continue label0;
        if(xmin <= x_series[i] && x_series[i] <= xmax && ymin <= y_series[i] && y_series[i] <= ymax)
            g.fillOval(((int)(x_series[i] * xscale) - (int)(xmin * xscale)) + west - 2, ((ylength - (int)(y_series[i] * yscale)) +
(int)(ymin * yscale) + north) - 2, 5, 5);
        i += step;
    } while(true);
}
if(type.equalsIgnoreCase("line"))
{
    int i = 0;
    do
    {
        if(i >= x_series.length)
            continue label0;
        if(xmin <= x_series[i] && x_series[i] <= xmax && ymin <= y_series[i] && y_series[i] <= ymax && i <
x_series.length - 1 && xmin <= x_series[i + 1] && x_series[i + 1] <= xmax && ymin <= y_series[i + 1] && y_series[i + 1] <=
ymax)
            g.drawLine(((int)(x_series[i] * xscale) - (int)(xmin * xscale)) + west, (ylength - (int)(y_series[i] * yscale)) +
(int)(ymin * yscale) + north, ((int)(x_series[i + 1] * xscale) - (int)(xmin * xscale)) + west, (ylength - (int)(y_series[i + 1] *
yscale) + (int)(ymin * yscale) + north);
        i += step;
    } while(true);
}
if(type.equalsIgnoreCase("line-squares"))
{
    int i = 0;
    do
    {
        if(i >= x_series.length)
            continue label0;
        if(xmin <= x_series[i] && x_series[i] <= xmax && ymin <= y_series[i] && y_series[i] <= ymax && i <
x_series.length - 1 && xmin <= x_series[i + 1] && x_series[i + 1] <= xmax && ymin <= y_series[i + 1] && y_series[i + 1] <=
ymax)
            {
                g.drawLine(((int)(x_series[i] * xscale) - (int)(xmin * xscale)) + west, (ylength - (int)(y_series[i] * yscale)) +
(int)(ymin * yscale) + north, ((int)(x_series[i + 1] * xscale) - (int)(xmin * xscale)) + west, (ylength - (int)(y_series[i + 1] *
yscale) + (int)(ymin * yscale) + north);
                g.fillRect(((int)(x_series[i] * xscale) - (int)(xmin * xscale)) + west - 2, ((ylength - (int)(y_series[i] * yscale)) +
(int)(ymin * yscale) + north) - 2, 5, 5);
            }
        i += step;
    } while(true);
}
if(type.equalsIgnoreCase("line-circles"))
{
    int i = 0;
    do
    {
        if(i >= x_series.length)
            continue label0;
        if(xmin <= x_series[i] && x_series[i] <= xmax && ymin <= y_series[i] && y_series[i] <= ymax && i <
x_series.length - 1 && xmin <= x_series[i + 1] && x_series[i + 1] <= xmax && ymin <= y_series[i + 1] && y_series[i + 1] <=
ymax)

```

```

        {
            g.drawLine(((int)(x_series[i] * xscale) - (int)(xmin * xscale)) + west, (ylength - (int)(y_series[i] * yscale)) +
(int)(ymin * yscale) + north, ((int)(x_series[i + 1] * xscale) - (int)(xmin * xscale)) + west, (ylength - (int)(y_series[i + 1] *
yscale)) + (int)(ymin * yscale) + north);
            g.fillOval(((int)(x_series[i] * xscale) - (int)(xmin * xscale)) + west - 2, ((ylength - (int)(y_series[i] * yscale)) +
(int)(ymin * yscale) + north) - 2, 5, 5);
        }
        i += step;
    } while(true);
}
dataseries.setElementAt("squares", seriesindex + 3);
for(int i = 0; i < x_series.length; i += step)
    if(xmin <= x_series[i] && x_series[i] <= xmax && ymin <= y_series[i] && y_series[i] <= ymax)
        g.fillRect(((int)(x_series[i] * xscale) - (int)(xmin * xscale)) + west - 2, ((ylength - (int)(y_series[i] * yscale)) +
(int)(ymin * yscale) + north) - 2, 5, 5);
}

if(key)
{
    g.setColor(getForeground());
    int k_width = size().width - west - xlength - 25;
    int k_start = west + xlength + 15;
    g.drawRect(k_start, north, k_width, ylength);
    for(int seriesindex = 0; seriesindex < n_series * 4; seriesindex += 4)
    {
        g.setColor((Color)dataseries.elementAt(seriesindex + 2));
        type = (String)dataseries.elementAt(seriesindex + 3);
        if(type.equalsIgnoreCase("line") || type.equalsIgnoreCase("line-squares") || type.equalsIgnoreCase("line-circles"))
            g.drawLine(k_start + 20, north + 20 + (ylength / n_series / 4) * seriesindex, (k_start + k_width) - 20, north + 20 +
(ylength / n_series / 4) * seriesindex);
        if(type.equalsIgnoreCase("circles") || type.equalsIgnoreCase("line-circles"))
            g.fillOval(west + xlength + (size().width - west - xlength) / 2, north + 18 + (ylength / n_series / 4) * seriesindex,
5, 5);
        if(type.equalsIgnoreCase("squares") || type.equalsIgnoreCase("line-squares"))
            g.fillRect(west + xlength + (size().width - west - xlength) / 2, north + 18 + (ylength / n_series / 4) * seriesindex,
5, 5);
        g.setColor(getForeground());
        if(((String)series_names.elementAt(seriesindex / 4)).equalsIgnoreCase("")) ||
(String)series_names.elementAt(seriesindex / 4) == null)
            series_names.setElementAt("Series ".concat(String.valueOf(String.valueOf(String.valueOf(seriesindex / 4 +
1))))), seriesindex / 4);
        g.drawString((String)series_names.elementAt(seriesindex / 4), (k_start + k_width) / 2 -
fm.stringWidth((String)series_names.elementAt(seriesindex / 4)) / 2, north + 40 + (ylength / n_series / 4) * seriesindex);
    }
}
g.setColor(getForeground());
if(n_yticks > 1)
{
    for(int j = 0; j < n_yticks; j++)
    {
        if(String.valueOf(ymax - (yrange * (double)j) / (double)(n_yticks - 1)).length() >= 8)
            g.drawString(String.valueOf(ymax - (yrange * (double)j) / (double)(n_yticks - 1)).substring(0, 7), west -
fm.stringWidth(String.valueOf(ymax - (yrange * (double)j) / (double)(n_yticks - 1)).substring(0, 7)) - 7, north + (ylength * j) /
(n_yticks - 1) + 3);
        else
            g.drawString(String.valueOf(ymax - (yrange * (double)j) / (double)(n_yticks - 1)), west -
fm.stringWidth(String.valueOf(ymax - (yrange * (double)j) / (double)(n_yticks - 1))) - 7, north + (ylength * j) / (n_yticks - 1) +
3);
        g.drawLine(west, north + Math.round((ylength * j) / (n_yticks - 1)), west - 5, Math.round(north + (ylength * j) /
(n_yticks - 1)));
        if(y_gridlines)
            g.drawLine(west, north + Math.round((ylength * j) / (n_yticks - 1)), west + xlength, Math.round(north + (ylength *
j) / (n_yticks - 1)));
    }
}
if(n_xticks > 1)
{
    for(int j = 0; j < n_xticks; j++)
    {
        if(String.valueOf(xmin + (xrange * (double)j) / (double)(n_xticks - 1)).length() >= 8)

```

```

        g.drawString(String.valueOf(xmin + (xrange * (double)j) / (double)(n_xticks - 1)).substring(0, 7), (west + (xlength
* j) / (n_xticks - 1)) - fm.stringWidth(String.valueOf(xmin + (xrange * (double)j) / (double)(n_xticks - 1)).substring(0, 7)) / 2,
north + ylength + 20);
        else
            g.drawString(String.valueOf(xmin + (xrange * (double)j) / (double)(n_xticks - 1)), (west + (xlength * j) / (n_xticks
- 1)) - fm.stringWidth(String.valueOf(xmin + (xrange * (double)j) / (double)(n_xticks - 1))) / 2, north + ylength + 20);
        g.drawLine(west + (xlength * j) / (n_xticks - 1), north + ylength + 5, west + (xlength * j) / (n_xticks - 1), north +
ylength);
        if(x_grdlines)
            g.drawLine(west + (xlength * j) / (n_xticks - 1), north + ylength, west + (xlength * j) / (n_xticks - 1), north);
    }
}
if(zero_lines)
{
    if(xmin < 0.0D && xmax > 0.0D)
    {
        g.drawLine(-(int)Math.round(xmin * xscale) + west, north + ylength, -(int)Math.round(xmin * xscale) + west, north);
        g.drawLine(-(int)Math.round(xmin * xscale) + west + 1, north + ylength, -(int)Math.round(xmin * xscale) + west + 1,
north);
        g.drawLine(-(int)Math.round(xmin * xscale) + west - 1, north + ylength, -(int)Math.round(xmin * xscale) + west -
1, north);
    }
    if(ymin < 0.0D && ymax > 0.0D)
    {
        g.drawLine(west, (north + ylength + (int)Math.round(ymin * yscale)) - 1, xlength + west, (north + ylength +
(int)Math.round(ymin * yscale)) - 1);
        g.drawLine(west, north + ylength + (int)Math.round(ymin * yscale), xlength + west, north + ylength +
(int)Math.round(ymin * yscale));
        g.drawLine(west, north + ylength + (int)Math.round(ymin * yscale) + 1, xlength + west, north + ylength +
(int)Math.round(ymin * yscale) + 1);
    }
}
g.drawString(xlabel, (west + xlength / 2) - fm.stringWidth(xlabel) / 2, north + ylength + 40);
g.drawString(ylabel, west / 2 - fm.stringWidth(ylabel) / 2, north - 10);
Font f = getFont();
g.setFont(new Font(f.getName(), 1, f.getSize() + 2));
fm = getFontMetrics(getFont());
g.drawString(title, (west + xlength / 2) - fm.stringWidth(title) / 2, north - 10);
}
}
}

```

Referências Bibliográficas

1. IKEYA, M. ***New application of electron spin resonance*** – Dating, Dosimetry and Microscopy. London: Word Science Publishing, 1993.
2. SCHAMANN, A. Past and present of ESR in radiation applications. ***Appl. Radiat. Isot.***, v. 47, n. 11-12, p. 1151-1153, 1996.
3. REGULLA, D. F. EPR Dosimetry – present and future. In: IAEA TECHNIQUES FOR HIGH DOSE DOSIMETRY IN INDUSTRY, AGRICULTURE AND MEDICINE, Nov. 2-5, 1998. ***Proceedings ...*** Vienna: IAEA, 1999. (IAEA-TECDOC-1070).
4. MEHTA, K. High-dose standardization service of the IAEA. ***Appl. Radiat. Isot.***, v. 47, p. 1155-1159, 1996.
5. REGULLA, D. F.; DEFFNER U. Dosimetry by ESR spectroscopy of alanine. ***Int. J. Appl. Radiat. Isot.***, v. 33, p. 1101-1114, 1982.
6. DEFFNER, U.; REGULLA, D. F. Therapy-level dosimetry based on ESR spectroscopy of alanine free radicals. ***Phys. Med. Biol.***, v. 39, p.799-799, 1994.
7. URENA-NUNEZ, F.; GALINDO, S.; AZORIN, J. An Alanine-boron compound for thermal neutron fluence measurements. Part 1 Synthesis and Development. ***Appl. Radiat. Isot.***, v. 49, n. 12, p. 1657-1664, 1998.
8. GOHS, U. Application of an alanine dosimetry system for industrial irradiation and radiation protection. ***Appl. Radiat. Isot.***, v. 47, p. 1169-1170, 1996.

9. PASS, B. ET ALL. High radiation doses from radiotherapy measured by electron spin resonance dental enamel. *Radiat. Prot. Dosim.*, v. 76, n. 4, p. 239-247, 1998.
10. BRADSHAW, W. W.; CADENA, D. G.; CRAWFORD, G. W.; SPETZLER, H. A. The use of alanine as a solid dosimeter. *Radiat. Res.*, v. 17, p. 11-21, 1962.
11. INTERNATIONAL ATOMIC ENERGY AGENCY. *Use of electron paramagnetic resonance dosimetry with tooth enamel for retrospective dose assessment*. IAEA, Vienna: 2002. (IAEA-TECDOC-1331).
12. COSTA, Z. M. *Desenvolvimento de um sistema dosimétrico para situações de emergência envolvendo pessoas do público em geral*. 1999. Tese (Doutorado) - Instituto de Pesquisas Energéticas e Nucleares, São Paulo.
13. KOJIMA, T; TANAKA, Y.; MORITA, Y.; SEGUCHI Alanine dosimeters using polymers as binders. *Appl. Radiat. Isot.*, v. 37, n. 6, p. 517-520, 1986.
14. WIELOPOLSKI, L.; MARYANSKI, M.; CIESIELSKI, B.; FORMAN, A.; REINSTEIN, L E.; MEEK, A. G. Continuous three-dimensional radiation dosimetry in tissue-equivalent phantoms using electron paramagnetic resonance in L- α -alanine. *Med. Phys.*, v. 14, n. 4, p. 646-652, 1987.
15. OLSSON, S.; BERGSTRAND, E.S.; CARLSSON, A.K.; HOLE, E.O.; LUND, E. Radiation dose measurements with alanine/agarose gel and thin alanine films around a Ir-192 brachytherapy source, using ESR spectroscopy. *Phys. Med. Biol.*, v. 47, n. 8, p.1333-1356, 2002.
16. GALANTE, O. L.; *Padronização do Método de Dosimetria de Doses Altas pela Técnica de Ressonância Paramagnética Eletrônica*. 2000. Dissertação (Mestrado) - Instituto de Pesquisas Energéticas e Nucleares, São Paulo.

17. GALINDO, S.; URENANUNEZ, F. Comparison of alanine dosimeters using silicone as their binder to a commercial, polystyrene-bound, alanine dosimeter. *Rev. Mex. Fis.*, v. 43, n. 3, p. 468-476, 1997.
18. MEHTA, K.; GIRZIKOWSKY, R. Reference Dosimeter System of the IAEA. *Radiat. Phys. Chem.*, v. 46, n. 4-6, p. 1247-1250, 1995.
19. BARTOLOTTA, A.; FATTIBENE, P.; ONORI, S.; PANTALONI, M.; PETETTI, E. Sources of uncertainty in therapy level alanine dosimetry. *Appl. Radiat. Isot.*, v.44, n. 1-2, p. 13-17, 1993.
20. NETTE, H.P., ONORI, S., FATTIBENE, P., REGULLA, D.F., WIESER, A. Coordinated research efforts for establishing an International Radiotherapy Dose Intercomparasion Service Based on the alanine/ESR System. *Appl. Radiat. Isot.*, v.44, n. 1-2, p. 7-12, 1993.
21. EISBERG, R.; RESNICK, R. *Física Quântica - Átomos, Moléculas, Sólidos, Núcleo e Partículas*. 3º. Ed. Rio de Janeiro: Campus, 1985.
22. CUGUNOV L.; MEDNIS, A.; KLIAVA, J. Computer simulations of the EPR spectra for ions with $S > 1/2$ by the Eigenfield and Related Methods. *J. Magn. Reson. A*, v. 106, p. 153-158, 1994.
23. PAKE, G. E. **Paramagnetic Resonance – An Introductory Monograph**. New York, W.A. Benjamin, 1962.
24. BERSOHN, M.; BAIRD, J. .C. *An introduction to electron paramagnetic resonance*. New York, W.A.: Benjamin, 1966.
25. ABRAGAM, A.; BLEANEY, B. *Electron paramagnetic resonance of transition ions*. Dove Publication, 1970.
26. BANWELL, C. M.; MCCASH, E. M. **Fundamentals of Molecular Spectroscopy**. Fourth Edition, London: McGraw- Hill, 1994.

27. RODRIGUES JÚNIOR, O.; GALANTE, O. L.; CAMPOS, L. L. ***Um sistema dosimétrico de referência para o intervalo de dose da radioterapia baseado na alanina/RPE*** - V Congresso Regional do IRPA - International Radiation Protection Association - Recife, 29 abril 04 de maio de 2001.
28. MALINEN, E.; HEYDARI, M. Z.; SAGSTUEN, E.; HOLE, E. O. Alanine radicals, part 3: Properties of the components contributing to the EPR spectrum of X-irradiated alanine dosimeters. ***Radiat. Res.***, v.159, p. 23-32, 2003.
29. NAGY, V. Y.; SOKOLOV, D. P.; KOMOZIN, P. N.; DESROSIERS, M. F. Choosing reference samples for EPR concentration measurements. Parts 1,2,3 and 4. ***Analytica Chimica Acta***, v. 339, n. 1-2, p. 1-62, 1997.
30. HASKELL, E. H.; HAYES, R.B.; G.H.; KENNER A. High Sensitivity EPR Technique For Alanine Dosimetry. ***Radiat. Prot. Dosim.*** v. 77, n. 1-2, p. 43-49, 1998.
31. HAYES, R.B.; HASKELL, E.H. ; BARRUS, J.K.; KENNER, G.H. ; ROMANYUKHA, A.A. Accurate EPR radiosensitivity calibration using small sample masses. ***Nucl. Instrum. Methods Phys. Res.***, v. 441, n. 3, p. 535-550, 2000.
32. WIESER, A.; SIEGELE, R.; REGULLA, D. F. Influence of the Irradiation temperature on Free-radical Response of Alanine. ***Appl. Radiat. Isot.***, v. 40, n. 10-12, p. 957-959, 1989.
33. BERGSTRAND, E.; HOLE, E. O.; SAGSTUEN, E. A Simple Method for Estimating Dose Uncertainty in ESR/Alanine Dosimetry. ***Appl. Radiat. Isot.***, v. 49, n. 7, p. 845-854, 1998.
34. NAGY, V. AND SHOLOM, S.V. AND CHUMAK, V.V. AND DESROSIERS, M.F. Uncertainties in alanine dosimetry in the therapeutic dose range, ***Appl. Radiat. Isot.***, v. 56, n. 6, p. 917--929, 2002.

35. RUCKERBAUER, F.; SPRUNCK, M.; REGULLA, D. F. Numerical Signal Treatment of Optimized Alanine/ESR Dosimetry in the Therapy-level Dose Range. *Appl. Radiat. Isot.*, v. 47, n. 11-12, p. 1263-1268, 1996.
36. SAVITZKY, A.; GOLAY, M. J. E. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Anal. Chem.*, v. 36, n. 8, p. 1627, 1964.
37. PRESS, W. H.; TEUKOLSKY, S.A.; VETTERING, W. T.; FLAMMERY, B.P. **Numerical Recipes in FORTRAN**. Sec. Edition - New York: Cambridge Press, 1992.
38. VERMA, R. Profile Deconvolution Method for Small Computers. *Nucl. Instrum. Methods*, v. 212, p. 323-326, 1983.
39. SJONTOFT, E. A straightforward deconvolution method for use in small computers. *Nucl. Instrum. Methods*, v. 163, p. 519-522, 1979.
40. CERQUEIRA, E. O.; POPPI, R. J.; KUBOTA, L. T.; MELLO, C. Utilização de filtros de transformada de Fourier para a minimização de ruídos em sinais analíticos. *Quim. Nova*, v.23, n. 5, p. 690-698, 2000.
41. SILSBEE, R. H. Fourier Transform Analysis of Hyperfine Structure in ESR. *J. Chem. Phys.*, v. 45, n. 5, p. 1710, 1966.
42. GRIVET, J-P. Accurate Numerical Approximation to the Gauss-Lorentz Lineshape. *J. Mag. Reson.* v. 125, p.102-106, 1997.
43. HEINER, Y. Evolution Strategies Applied to Least-Squares curve fitting of Spectroscopic Data. *J. Quant. Spectrosc. Radiat. Transfer*, v.56, n. 5, p.769-782, 1996.
44. DAUBECHIES, I. **Ten Lectures on Wavelets**, Philadelphia: SIAM, 1992.
45. GRAPS, A. **An Introduction to Wavelets** , 1995. Disponível em <http://www.amara.com/ftpstuff/IEEEwavelet.ps.gz> Acessado em 09/08/2001.

46. DONOHO, D. L. De-noising by soft-thresholding. *IEEE Trans. On Inform. Theory*, v. 41, n. 3, p. 613-627, 1995.
47. ZHENG, X.; MO, J.; XIE, T. Studies of spline wavelet least square in processing electrochemical signals. *Talanta*, v. 48, p. 425-436, 1999.
48. ABALLE, A.; BETHENCOURT, M.; BOTANA, F. J.; MARCOS, M. Using wavelets transforms in the analysis of electrochemical noise data. *Electrochim. Acta*, v. 44, p. 4805-4816, 1999.
49. NAGY, V.; SLEPTCHONOK, O.F.; DESROSIERS, M.F.; WEBER, R.T.; HEISS, A.H. Advancedments in accuracy of the alanine EPR dosimetry system Part III: Usefulness of an adjacent reference sample. *Radiat. Phys. Chem.*, v. 59, n. 4, p. 429-441, 2000.
50. HAYES, R.B.; HASKELL, E.H.; WIESER, A.; ROMANYUKHA, A.A.; HARDY, B.L. ; BARRUS, J.K. Assessment of an alanine EPR dosimetry technique with enhanced precision and accuracy. *Nucl. Instrum. Methods Phys. Res. Sect. A*, v. 440, n. 2, p. 453-461, 2000.
51. YORDANOV, N. D.; GANCHEVA, V. Some new approaches in the field of solid state/EPR dosimetry. *Advances in ESR Applications*, v. 18, p. 227-231, 2002.
52. WIESER, A.; LETTAU, C.; FILL, U.; REGULLA, D.F. The Influence of non-radiation induced ESR background signal from paraffin-alanine probes for dosimetry in the radiotherapy dose range. *Appl. Radiat. Isot.*, v. 44, n. 1-2, p. 59-65, 1993.
53. PASS, B.; SHAMES, A. I. Signal processing for radiation dosimetry using EPR in dental enamel: comparison of three methods. *Radiat. Measurem.*, v.32, n. 3, p. 163-167, 2000.

54. WIESER, A., ONORI, S., ARAGNO, D., FATTIBENE, P., ROMANYUKHA, A., IGNATIEV, E., KOSHTA, A., SKVORTZOV, V., IVANNIKOV, A., STEPANENKO, V., CHUMAK, V., SHOLOM, S., HASKELL, E., HAYES, R., KENNER, G. Comparison of sample preparation and signal evaluation methods for EPR analysis of tooth enamel. *Appl. Radiat. Isot.*, v. 52, p. 1059-1064, 2000.
55. PRESSMAN, R. S. **Software Engineering - A practitioner's approach**. Fourth Edition. New York: McGraw Hill, 1997.
56. SOMMERVILLE, I. **Software engineering**. Wokingham, UK: A.P. M&Tbooks, 1998.
57. BOOCH, G., RUMBAUGH, J., JACOBSON, I. **UML - Guia do Usuário**, Rio de Janeiro: Ed. Campus, 2000.
58. **Java Technologies**. Disponível em: <<http://www.java.sun.com>>. Acesso em: 28 ago. 2003.
59. FURLAN, J. D. **Modelagem de Objetos através da UML – Análise e Desenho Orientados a Objetos**. São Paulo: Makron Books, 1998.
60. **Mysql**. Disponível em : <<http://mysql.com>>. Acesso em: 12 maio 2003.
61. HORTON, I. **Beginning Java 2, A Comprehensive tutorial to Java programming**, USA: Wrox Press. 1999.
62. LYON, D.A.; RAO, H. V. **Java digital signal processing**. New York: Pearson Educational, 2001.
63. SMITH, S. A.; ET AL. Computer Simulations in Magnetic Resonance. An Object-Oriented Programming Approach. *J. Magn. Reson. A*, v. 106, p. 75-105, 1994.
64. KIRSTE, B. Methods for Automated Analysis and Simulation of EPR Spectra. *Anal. Chim. Acta*, v. 265, p. 192-200, 1992.

65. AMERICAN SOCIETY FOR TESTING AND MATERIALS. ***Standard Practice for Use of the Alanine-EPR Dosimetry System***. ASTM, 2002. (ISO/ASTM 51607:2002(E)).
66. AMERICAN SOCIETY FOR TESTING AND MATERIALS. ***Standard Practice for Characterization and Performance of a High-Dose Radiation Dosimetry Calibration Laboratory*** . ASTM, 2002. (ISO/ASTM 51400:2002(E)).