

Este exemplar corresponde à redação final
da tese defendida pelo Sr. Gerald Weber e
aprovada pela Comissão Julgadora.

Campinas, 17 FEV 87

Carl Alberto

"APLICAÇÃO DO MÉTODO MONTE CARLO AO ESTUDO DE CORRENTE INDUZIDA POR
FEIXE ELETRÔNICO EM DISPOSITIVOS SEMICONDUTORES

GERALD WEBER

Orientador: Carlos Alberto Ribeiro

Trabalho apresentado ao Insti-
tuto de Física "Gleb Wataghin" da
Universidade Estadual de Campinas,
para a obtenção do grau de Mestre
em Ciências

Fevereiro de 1987

UNICAMP
BIBLIOTECA CENTRAL

Aos meus Pais, com muito ca-
rinho

Agradecimentos

Ao Ribeiro pela orientação, e pela grande amizade,

Ao Celso pelas medidas experimentais e pela paciência de aturar um teórico no laboratório,

Ao Ricardo pelas discussões (às vezes infundáveis),

Ao Mário pela assistência computacional,

Ao Máximo pelas discussões e pelos incentivos,

A Claudete e à Maura por tantos "é pra ontem" e outros pepinos,

A Dina, fiel guardiã dos livros e periódicos,

Aos meus Pais e à minha irmã pelo apoio irrestrito à minha escolha profissional,

A Maria José pelo carinho e apoio em horas de desânimo e dificuldades,

Ao Xandão, Roberto e Marcão e ao "Clubinho" como um todo pela amizade,

Ao Peter por uma amizade que alguns abalos não tornaram menos profunda,

Aos colegas, funcionários e professores que não mencionei,

A FAPESP não somente pelo apoio financeiro, mas também pela ausência de burocratices e a forma como honra seus compromissos,

A Telebrás S/A pelo apoio financeiro e pelo uso das instalações o LPD,

Meus Agradecimentos

Resumo

Simulamos a corrente induzida por feixe eletrônico para um diodo e um laser de heteroestrutura dupla pela técnica de simulação Monte Carlo. Obtemos pelo método Monte Carlo uma distribuição de energia característica com a qual calculamos a produção e difusão dos portadores minoritários e a sua coleção pela junção. Desta forma é possível estimar o comprimento de difusão nos dispositivos. A ferramenta produzida possibilita uma nova abordagem dos fenômenos característicos nos dispositivos semicondutores analisados por um microscópio eletrônico de varredura.

Abstract

We simulate the electron beam induced current for a diode and a double heterostructure laser with a Monte Carlo simulation technique. We obtain by this simulation method a characteristic energy distribution by which we calculate minority carrier production, diffusion and collection at the junction. By this way it is possible to us to make an estimative for the carrier diffusion length in semiconductor devices analized by a scanning electron microscope.

Índice

1.....	Introdução
3.....	Capítulo 1 - Método Monte Carlo para a Microsonda Eletrônica
13.....	Capítulo 2 - Perda de Energia em Função da Profundidade
19.....	Capítulo 3 - Corrente Induzida por feixe eletrônico (EBIC)
29.....	Capítulo 4 - Resultado da Simulação EBIC
41.....	Conclusão
43.....	Apêndice A - Equação de Perda de Energia
47.....	Apêndice B - Livre Caminho Médio
50.....	Apêndice C - Ângulo de Espalhamento
52.....	Apêndice D - Coordenadas do Elétron Simulado em Três Dimensões
56.....	Apêndice E - Energias de Ionização Radiativas em Semicondutores de "gap" direto
61.....	Apêndice F - Organização dos Programas
65.....	Apêndice G - Programas
97.....	Bibliografia

ERRATA

pg 19 O ultimo parágrafo - onde se lê "banda de condução" leia-se "banda de valência" e vice versa.

pg 24 1º parágrafo - leia-se "condição de normalização" em vez de "condição de contorno".

pg 58 Acrescente-se à legenda da figura E1: "(Vide referencia [1])".

pg 59 A expressão correta da equação (E.8) é:

$$mv = (2m_e + m_p)v$$

pg 60 A expressão correta da equação (E.11)

$$q = \frac{E_G}{\langle E \rangle}$$

da equação (E.17)

$$q = \frac{E_G}{E_G + \frac{3}{5} (E_{et} + E_{pt})} = \frac{5}{14}$$

e da equação (E.18)

$$\langle E \rangle = \frac{14}{5} E_G$$

"A idéia é que a realidade, seja a da Santa Sé, a de René Char ou a de Oppenheimer, é sempre uma realidade convencional, incompleta e parcelada. A admiração de alguns caras diante de um microscópio eletrônico não me parece mais fecunda do que a admiração das parteiras pelos milagres de Lourdes."

Julio Cortázar em "O Jogo da Amarelinha"

Introdução

O uso dos métodos de Monte Carlo na resolução de problemas em Microscopia Eletrônica de varredura não constitui nenhuma novidade. Em alguns casos, principalmente na microanálise, tem mesmo se tornado um método padrão [1].

Mas a microscopia não parou na microanálise. Novos métodos de trabalho tem surgido, notadamente na área de semicondutores e dispositivos. Novas técnicas utilizando-se de efeitos típicos ou predominantes em semicondutores tornaram-se comuns e de larga aplicação.

Se o microscópio eletrônico já era por natureza um dos principais instrumentos na pesquisa de dispositivos optoeletrônicos e de circuitos integrados, as técnicas de corrente induzida por feixe eletrônico e a catodoluminescência simplesmente o tornaram indispensável.

As abordagens teóricas destes novos métodos até agora elaborados são todas de cunho analítico e apresentam muitas hipóteses e aproximações. Tais aproximações eram na sua maioria muito exageradas e tinham uma validade razoável apenas para amostras de silício homogêneo.

O nosso trabalho tem por objetivo lançar mão do método Monte Carlo e desenvolver alguns modelos analíticos, e em conjunto obter resultados práticos para dispositivos a serem estudados. O dispositivo mais complexo discutido é um laser de heteroestrutura dupla. A complexidade de sua estrutura e a diversidade dos materiais torna-o impossível de ser estudado por qualquer modelo analítico.

A meta final é desenvolver uma ferramenta potencialmente poderosa e inédita para a análise de muitas outras situações experimentais envolvendo os fenômenos acima citados.

1 - Método Monte Carlo para a Microsonda Eletrônica

O método Monte Carlo é essencialmente uma forma de simulação estatística para processos que não podem ser reunidos em um único modelo ou abordagem estatística.

Vamos supor que um determinado processo físico se decomponha em diversos subprocessos interdependentes. Cada um destes subprocessos é conhecido ou suposto e é caracterizado pela sua média estatística e a sua distribuição (frequência com que ocorre um dado valor ou característica). A simulação consiste em percorrer muitas vezes todos estes subprocessos com auxílio de números aleatórios. Tomamos o primeiro subprocesso, escolhemos de alguma forma um número aleatório e através da distribuição característica do subprocesso obtemos um determinado resultado particular. Passamos ao segundo subprocesso cujas condições podem depender do primeiro e procedemos da mesma forma para obter um determinado resultado particular, referente a um novo número aleatório. Continuamos assim até passar por todos os subprocessos. Ao final obtemos um determinado resultado particular que caracteriza o processo total. Este procedimento é repetido muitas vezes sempre para números aleatórios novos. Desta forma chegamos à média e à distribuição do processo.

A situação física que pretendemos estudar apresenta a característica de decompor o processo em vários subprocessos cuja física é bem conhecida. Entretanto não é possível juntar estes subprocessos em um único modelo coerente. A única saída viável neste caso é partir para o procedimento descrito acima.

A desvantagem inerente a este tipo de procedimento é o problema da escolha do modelo em geral *ad hoc*. Torna-se importante testes que indiquem a coerência dos resultados para situações bem conhecidas. Outra desvantagem é que a convergência estatística para uma simulação Monte Carlo, isto é, o número de vezes que o procedimento precisa ser repetido para que se convirja para determinado resultado, é uma das piores que se tem notícia. Quando quadruplicando, por exemplo, o número de procedimentos apenas dobra a precisão.

Vamos situar o nosso problema físico. Um feixe de elétrons incide sobre um material com energia fixa entre 5 a 45 KeV. O nosso interesse consiste em saber como estes elétrons depositam a sua energia no interior da amostra em termos quantitativos.

Observamos inicialmente que os elétrons têm energia suficientemente grande para não termos que nos preocupar com o arranjo espacial do sólido, tendo que levar em conta apenas a densidade específica deste.

Teremos assim espalhamentos elásticos e inelásticos dos elétrons se sucedendo de uma forma não previsível (isto é, aleatória) em um meio que consideraremos como sendo homogêneo.

A teoria da perda de energia ao longo do caminho do elétron tal como deduzido por Bethe [1] vem a calhar nesta situação. Nesta teoria a perda de energia se dá continuamente ao longo da trajetória (vide apêndice A), contabilizando os espalhamentos inelásticos.

Vamos seguir a trajetória de um elétron simulado pelo modelo, que apresentaremos com maior detalhe mais adiante, tal como mostrado na figura 1. Em nenhum instante afirmamos ser este o comportamento real

do elétron, buscamos apenas reunir aspectos próximos ao comportamento médio dos elétrons.

O nosso elétron incide perpendicularmente sobre a superfície, com uma dada energia fixa. Neste ponto dizemos que ele sofre o seu primeiro espalhamento por um ângulo θ . Deste ponto em diante seguirá em linha reta por um caminho com λ de comprimento, ao fim do qual calculamos a sua nova energia $E(1)$. O ângulo de espalhamento θ e o livre caminho percorrido λ não são quantidades que podemos determinar a priori mas terão que provir de números aleatórios em associação a medidas estatísticas conhecidas. A partir de λ calculamos a energia restante $E(1)$.

Há um aspecto pragmático a ser considerado na simulação, no sentido de que o espalhamento não pode seguir indefinidamente, e é preciso escolher uma energia limite a partir da qual pararemos a simulação. O critério de parada através desta energia limite em geral obedece a considerações do modelo (que a partir de uma certa energia cessa de ser válido) ou do sinal que se pretende obter pela simulação (abaixo de uma certa energia pode não haver produção deste sinal).

Uma segunda situação que pode ocorrer é que a simulação leve o elétron para fora da amostra. Dissemos neste caso que o elétron foi retroespalhado.

Definidos os critérios de parada, aplicamos os testes ao elétron ora com energia $E(1)$. Se prosseguirmos com a simulação, tomamos um novo ângulo de espalhamento (sempre referente a direção do movimento) e um novo livre caminho percorrido, ao fim do qual calculamos uma nova energia $E(2)$ e assim por diante.

Para os elétrons seguintes procedemos sempre da mesma forma, sendo todo o processo governado pelos números aleatórios.

Para a perda de energia usamos a conhecida equação de Bethe cuja dedução simplificada pode ser encontrada no apêndice A. Aqui nós a usaremos sob a forma

$$dE/ds = -7.85 \times 10^4 (\rho Z/AE) \ln(1.16E/J) \quad , \quad (\text{keV/cm}) \quad (1)$$

onde E é a energia do elétron, Z é o número atômico, A a massa atômica, ρ a densidade específica (g/cm^3) e J o potencial médio de ionização.

Para materiais contendo vários elementos (e suficientemente homogêneos) podemos usar, segundo Heinrich [1]

$$dE/ds = -7.85 \times 10^4 (\rho/E) \sum_i Z_i \ln(1.16E/J_i) / \sum_i A_i \quad (2)$$

Para o livre caminho percorrido recorremos a equação (B.14) chamando a atenção quanto a sua dependência com o número aleatório R_1 e a energia do elétron

$$\lambda = \lambda(E; R_1) \quad (3)$$

Com estes elementos podemos montar a relação de recorrência

$$E' = E_0 + \frac{dE}{ds} \Big|_{E_0} \lambda(E_0; R_1) \quad (4)$$

Aqui devemos tomar alguns cuidados com o livre caminho percorrido. O produto

$$\frac{dE}{ds} \Big|_E \lambda(E) \propto E$$

na ausência do fator aleatório e nos limites usados na prática, é grosseiramente linear com a energia. Isto nos garante que a "quebra" do movimento em segmentos menores fornecerá resultados semelhantes no tocante ao caminho percorrido. Devemos assim ter especial cuidado com o comportamento da componente aleatória do livre caminho percorrido impondo-lhe um limite superior razoável. Tal limite não resultará em maiores problemas durante a simulação pois o livre caminho percorrido

tem uma distribuição de Poisson e valores duas vezes maiores que a média tornam-se muito improváveis.

Por fim resta o espalhamento angular cujo responsável é o espalhamento elástico (supomos que no espalhamento inelástico o desvio angular é desprezível). Por razões de simplicidade usaremos o espalhamento Rutherford clássico que descrevemos no apêndice C. O ângulo de espalhamento com a direção do movimento se escreve como

$$\theta = 2 \arctg(R_2 \frac{bz}{E}) \quad (5)$$

onde R_2 é uma variável aleatória uniformemente distribuída entre 0 e 1. Resta um parâmetro empírico b a ser ajustado com as primeiras simulações de forma a produzir o coeficiente de espalhamento correto. As simulações realizadas mostram em nosso trabalho que a dependência de b com a energia inicial E_i pode ser descrita por

$$b = 0.2798E_i^{-0.5333} \quad (6)$$

As razões para adotarmos um modelo tão simples são:

- i) Um modelo de primeiros princípios requer um tempo computacional incompatível com os nossos recursos e propósitos
- ii) Foi mostrado por Liljequist [3] que o resultado final da simulação Monte Carlo é dominado pelo modelo do livre caminho médio adotado, tendo o espalhamento angular pouca influência. O autor em questão mostra inclusive simulações onde vai ao extremo de colocar um ângulo de espalhamento fixo por todo o processo, obtendo resultados bastante coerentes.

Se imaginarmos um sistema de coordenadas esféricas centrado sobre o elétron, podemos dizer que a nova posição do elétron é definida pelo livre caminho percorrido (distância), pelo ângulo de espalhamento

elástico (ângulo polar) e finalmente por um ângulo azimutal que não consideramos até agora. A situação está ilustrada na figura 2. O ângulo azimutal é o ângulo que o parâmetro de impacto (vide apêndice C) forma com um eixo centrado sobre o átomo espalhador, como não há ângulo preferencial neste caso supomos que o ângulo azimutal é dado por

$$\phi = 2\pi R_3 \quad (7)$$

onde R_3 é uma variável aleatória uniformemente distribuída entre 0 e 1.

Note que o movimento será determinado por três variáveis aleatórias R_1 , R_2 e R_3 e correspondem aos três graus de liberdade do sistema tridimensional, fechando desta forma o nosso modelo de simulação.

Há uma dificuldade prática na implementação deste modelo que consiste no fato de termos que trabalhar simultaneamente com o referencial de laboratório e o da partícula sendo que esta última requer um sistema cartesiano diferente para cada espalhamento. Simetrias do problema levam evidentemente a simplificações que evitam o uso de matrizes de rotação e transferência. Toda esta questão encontra-se detalhada no apêndice D.

Distribuição angular

Um aspecto sumamente importante na simulação Monte Carlo da microsonda eletrônica é o comportamento dos elétrons retroespalhados, isto é, elétrons do feixe primário que incidem sobre a amostra, sofrem processos de espalhamento e em seguida saem novamente da amostra. Um bom teste para o modelo é a distribuição angular dos elétrons retroespalhados. Para se entender melhor o que acontece é conveniente adotar o sistema de eixos cartesianos indicado na figura 1.

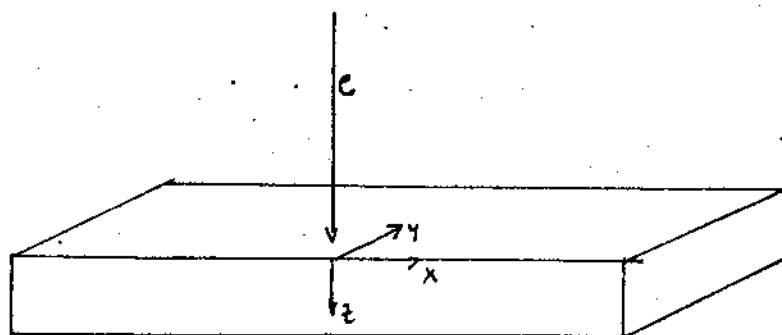


Figura 1

Onde o eixo z está direcionado para baixo, assim os elétrons incidentes vêm com a componente em z positiva e os elétrons retroespalhados com a componente negativa. Uma conclusão imediata a que se pode chegar sobre a possível uniformidade das distribuições angulares é em relação ao plano definido pelos eixos x e y quando temos uma incidência perpendicular a este plano: deve ser uma distribuição uniforme em uma

amostra homogênea e que supomos isotrópica (os elétrons primários não sofrem influência no espalhamento angular devido a planos do cristal por exemplo). Possuindo esta simetria a distribuição angular em relação a qualquer plano contendo o eixo z deve ser a mesma.

A figura 3 mostra os resultados obtidos pela nossa simulação para o ferro à uma tensão de aceleração de 30KeV. Nesta figura mostramos na forma de um histograma polar a média das energias dos elétrons retroespalhados em cada ângulo, onde na parte superior a distribuição é vista de lado e na parte inferior é vista como uma projeção no plano x' e y' tal como indicado na figura 2.

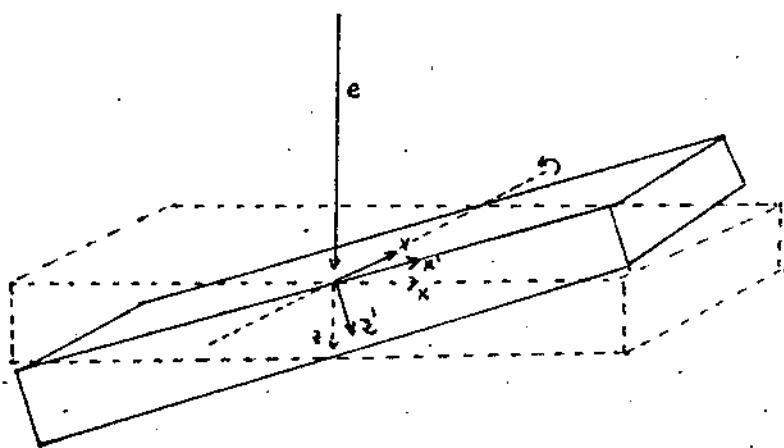


figura 2

Em seguida é mostrado como mudam as distribuições quando mudamos o ângulo de incidência da amostra. Podemos ver que à medida que variamos o ângulo a distribuição desloca o seu máximo também, sugerindo um comportamento de reflexão especular deste máximo.

Este teste é uma excelente averiguação não só do modelo mas também do próprio programa. Como exemplo citamos o fato que ocorreu conosco quando havia uma falha no nosso programa de simulação, na secção que calculava os parâmetros geométricos de movimento do elétron, realmente apareciam distorções visíveis nas distribuições angulares. Uma vez detetada a falha, nós a corrigimos e o programa voltou a fornecer resultados coerentes.

Consideramos portanto este teste como sendo excelente critério de avaliação do modelo e do programa de simulação.

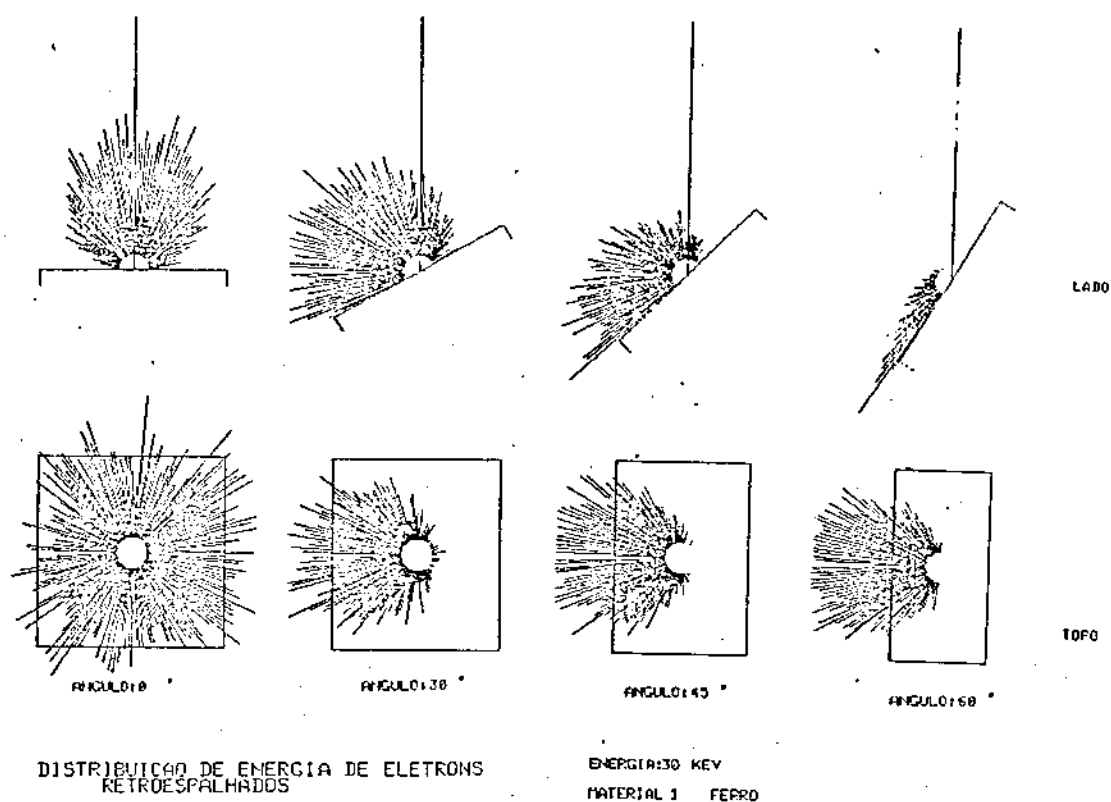


Figura 3

Variação da distribuição da energia com o ângulo. Usamos um tipo de histograma polar, o comprimento das linhas é proporcional à intensidade da energia retroespalhada.

2 - Perda de Energia em Função da Profundidade

Os resultados da dissipação de energia em função da profundidade merecem especial destaque, isto porque é com estes resultados que aplicamos os nossos modelos de EBIC (vide capítulo 3)

Para o tipo de modelo de EBIC que usamos será suficiente mostrarmos como a energia é dissipada em função da profundidade, isto é, a energia do feixe dissipada entre z e $z+dz$ sobre toda a extensão nas direções x e y .

Uma curva de dissipação de energia em função da profundidade (doravante chamada de curva dE/dz) com alta resolução (tipicamente 1000 intervalos) requer um tempo de simulação muito grande uma vez que a convergência estatística é lenta, isto é, a precisão aumenta com a raiz quadrada do número de trajetórias. Quadruplicar o número de trajetórias apenas dobra a precisão. Para ilustrar este efeito de convergência mostramos na figura 1 simulações com números crescentes de trajetórias para um laser de InGaAsP. Pode-se ver que a estrutura de camadas torna-se aparente a partir de 100.000 trajetórias. Apontamos algumas dificuldades na simulação de compostos quaternários do tipo do InGaAsP onde é necessário trabalhar com médias dos números atômicos, massa atômica e da densidade (parâmetros relevantes na simulação). A densidade (assim como o "gap") tem de ser calculada a partir de interpolações com os dados dos respectivos compostos ternários. Estes esquemas de interpolação são extensamente discutidos por Adachi [1]. O

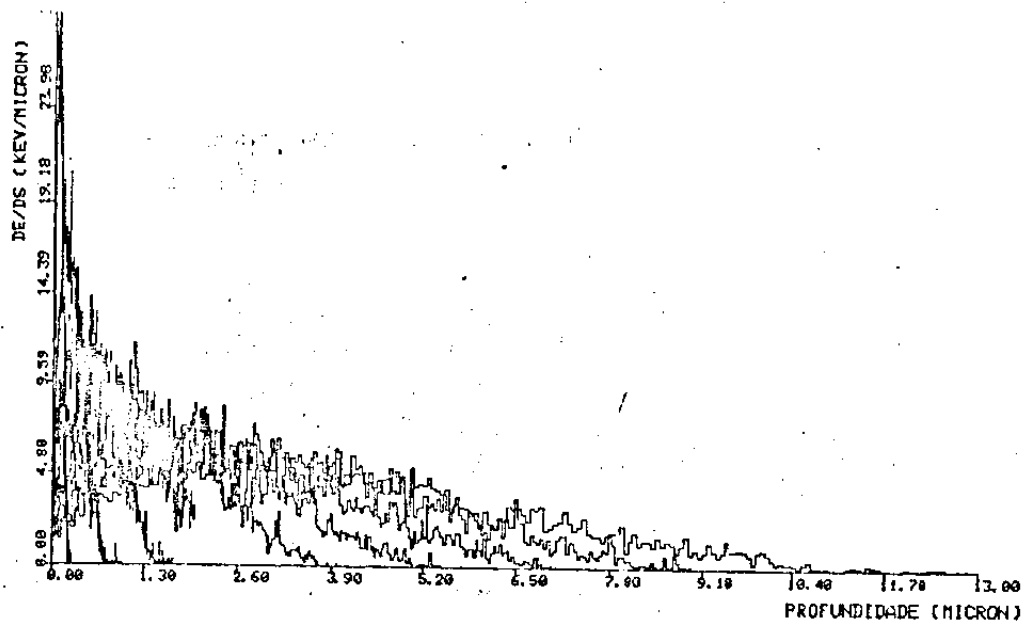
número atômico e a massa atômica podem ser encontrados por médias ponderadas com as frações de massa respectivas.

Para estruturas mais complexas como no caso do laser de InGaAsP é importante que a estrutura das camadas se torne visível. Na figura 2 mostramos detalhes da última curva dE/dz da figura 1 para ilustrar a sensibilidade em relação às interfaces.

Mostramos simulações para amostras homogêneas de GaAs (figura 3) e InP (figura 4), que serão utilizadas mais tarde na simulação de diodos de homojunção.

As curvas de perda de energia dE/dz têm ainda a vantagem de oferecer uma visão mais realista sobre o comportamento da microsonda. O que normalmente é adotado para os modelos de EBIC são distribuições de espalhamento dos elétrons obtidas por outras simulações Monte Carlo, originariamente efetuadas para microanálise. No caso da microanálise interessa saber onde o elétron primário realizou um espalhamento inelástico o suficientemente forte para produzir um fóton de raio-X característico. A distribuição dos espalhamentos fornece neste caso um formato peculiar de "pera", que acabou dando o nome de pera para a microsonda.

O uso das curvas dE/dz , portanto constituem uma maneira nova e mais realista nos cálculos de efeitos que dependem mais sensivelmente da distribuição de energia.



ENERGIA DISSIPADA

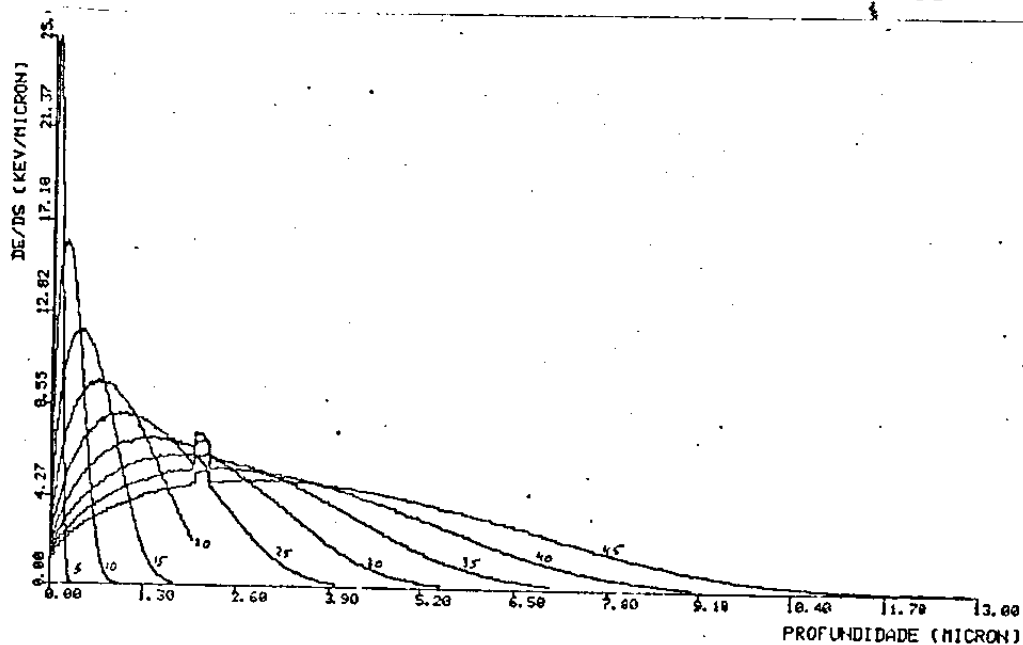
Laser de Três Camadas
In(0.7) Ga(0.3) As(0.6) P(0.4)

REGISTRO NR.: 1017

MICROSCOPIA ELETRONICA

SIMULACAO MONTE CARLO

LPD LABORATORIO DE PESQUISA
EM DISPOSITIVOS



ENERGIA DISSIPADA

Laser de Três Camadas
In(0.7) Ga(0.3) As(0.6) P(0.4)

REGISTRO NR.: 1015

MICROSCOPIA ELETRONICA

SIMULACAO MONTE CARLO

LPD LABORATORIO DE PESQUISA
EM DISPOSITIVOS

Figura 1

Simulações para um laser de InGaAsP para 1.000 e 500.000 trajetórias

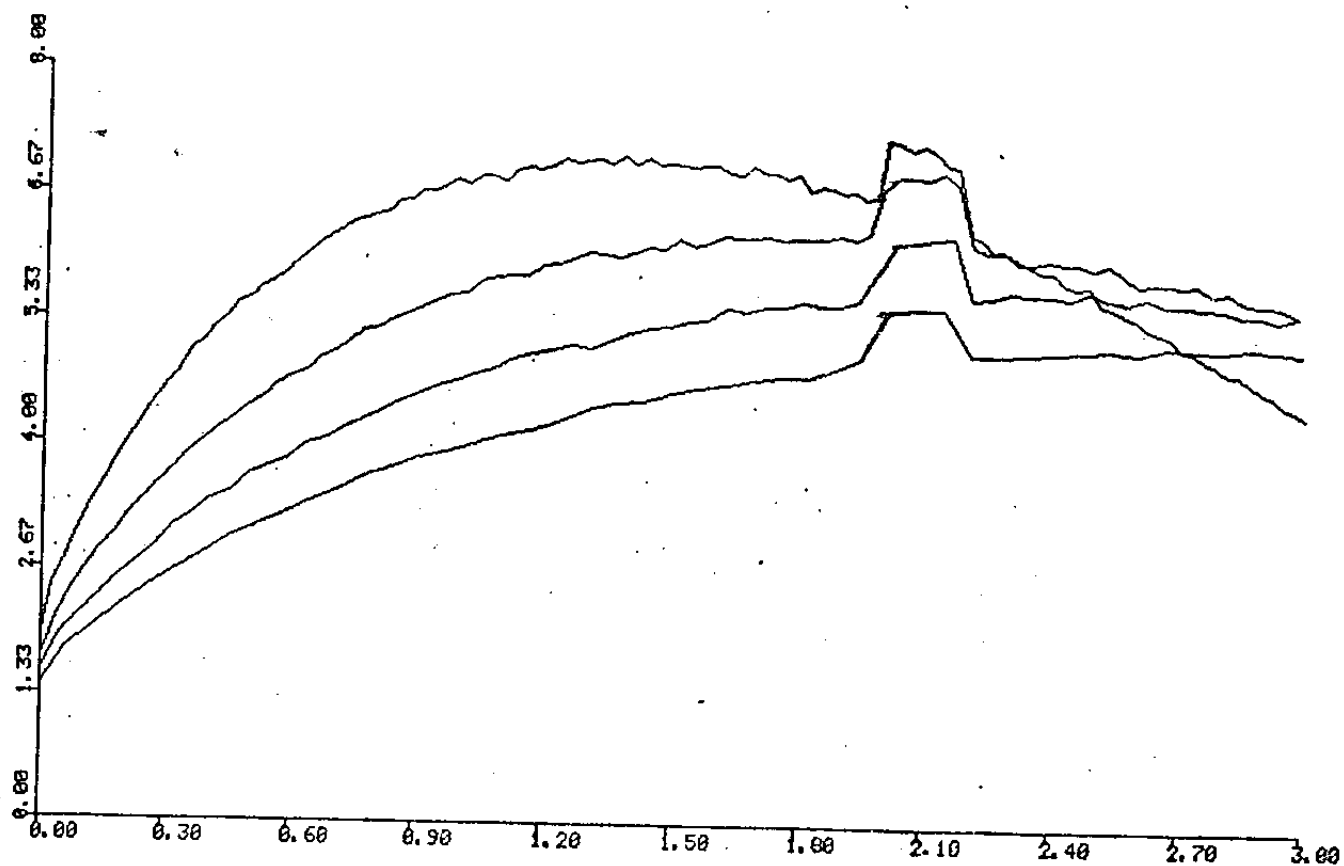


Figura 2

Detalhes da simulação de um laser de InGaAsP, 500.000 trajetórias

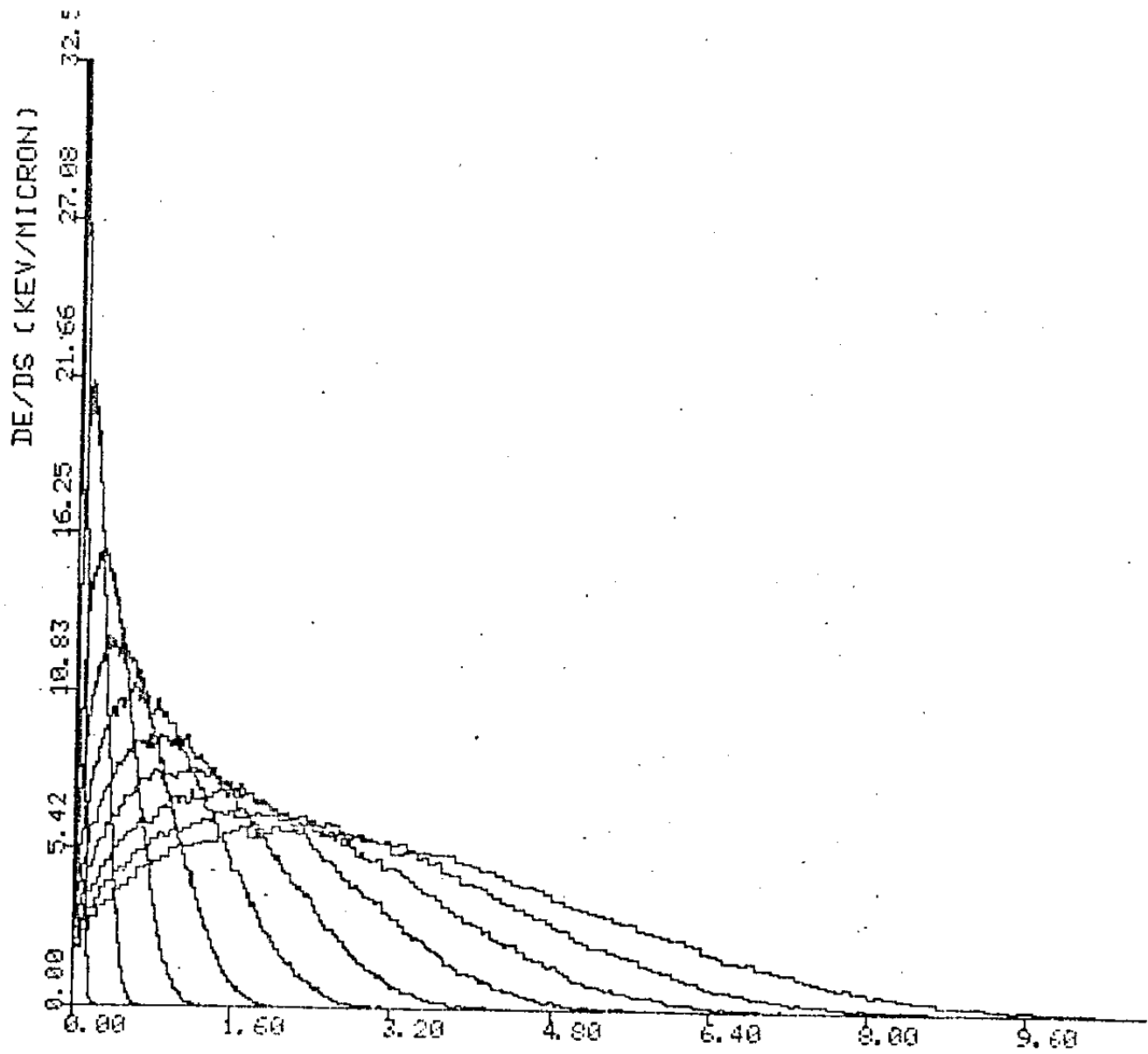
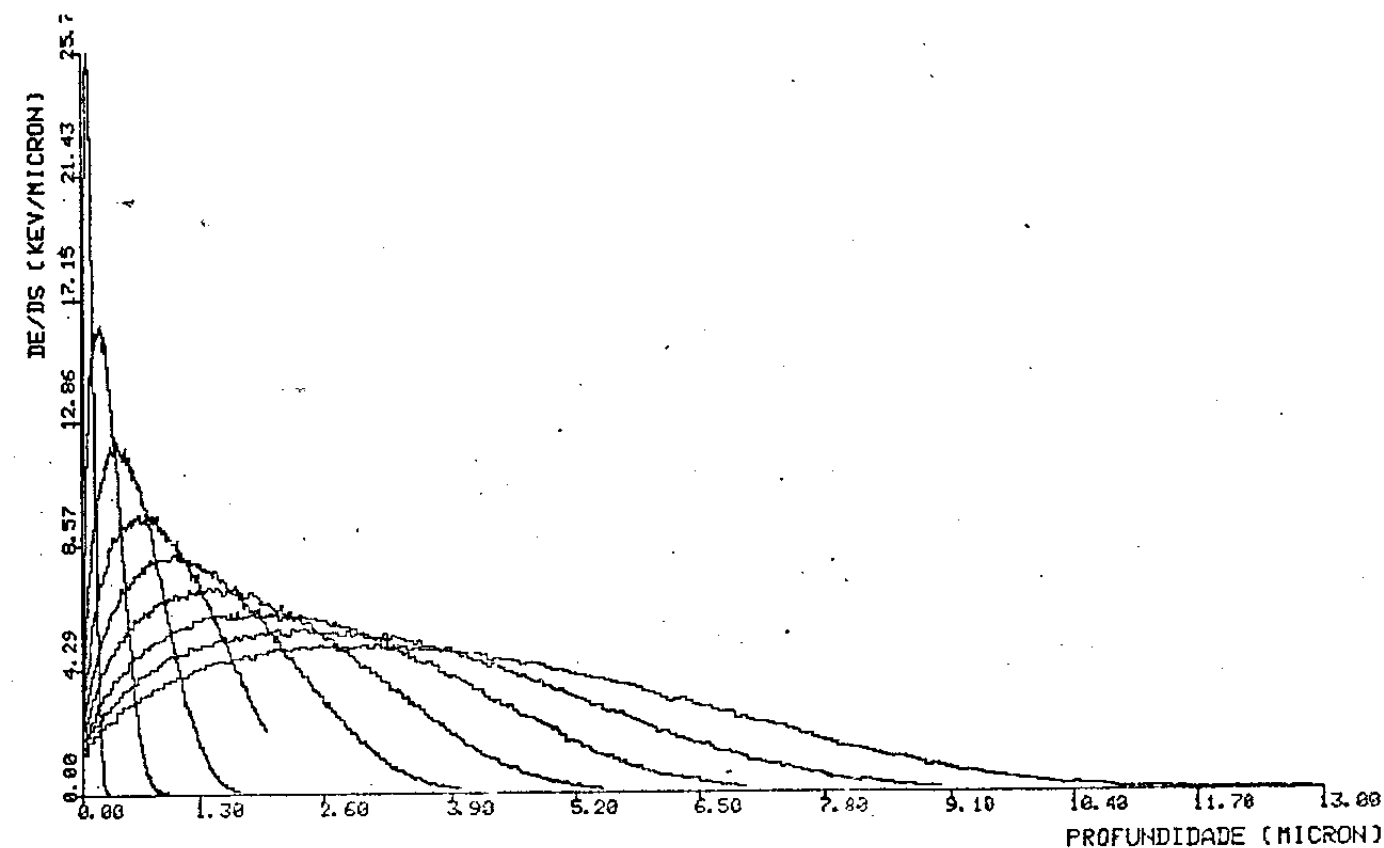


Figura 3

Simulação de diodo de homojunção de GaAs



ENERGIA DISSIPADA

INP
PURO

REGISTRO NR. : 1020

MICROSCOPIA ELETRONICA

SIMULACAO MONTE CARLO

LPD LABORATORIO DE PESQUISA
EM DISPOSITIVOS

Figura 4

Simulação de diodo de homojunção de InP

3 - Corrente induzida por feixe eletrônico (EBIC)

Antes de discutirmos o processo físico do fenômeno EBIC vamos fazer uma observação histórica. Ao contrário do que afirmam Hanoka e Bell[1], não foi Everhart[2], em 1958, o primeiro a fazer uma observação de EBIC em um microscópio eletrônico de varredura (SEM). Temos o artigo de Pfister[3] de 1957 com a primeira observação de EBIC. Por sinal, Pfister fez suas medidas para o Arseneto de Gálio publicando pela primeira vez as características de corrente versus tensão aplicada ao diodo no EBIC, com resultados muito bons para a época, sendo inclusive o primeiro a calcular a energia de ionização radiativa da qual falaremos mais adiante. Sendo a primeira fotografia com contraste EBIC feita por Lander et al.[4]

Mostramos na figura 1 o esquema experimental do modo EBIC. O feixe de elétrons incide em uma amostra semicondutora que possui uma junção (ou barreira Shottky) o suficientemente próxima da microsonda (região hachurada). Um sinal elétrico é produzido desta forma e é medido com o esquema eletrônico indicado. É possível a partir deste sinal obter imagens correspondentes à varredura do feixe sobre a amostra.

O feixe eletrônico gera, ao incidir, um grande número de pares elétron-buraco, ao promover elétrons da banda de condução à banda de valência. Os portadores minoritários determinam o comportamento da difusão e são portanto os portadores importantes a serem considerados. O

comprimento de difusão é suficientemente longo para que os portadores sejam coletados na junção, produzindo assim uma corrente entre os terminais do amperímetro.

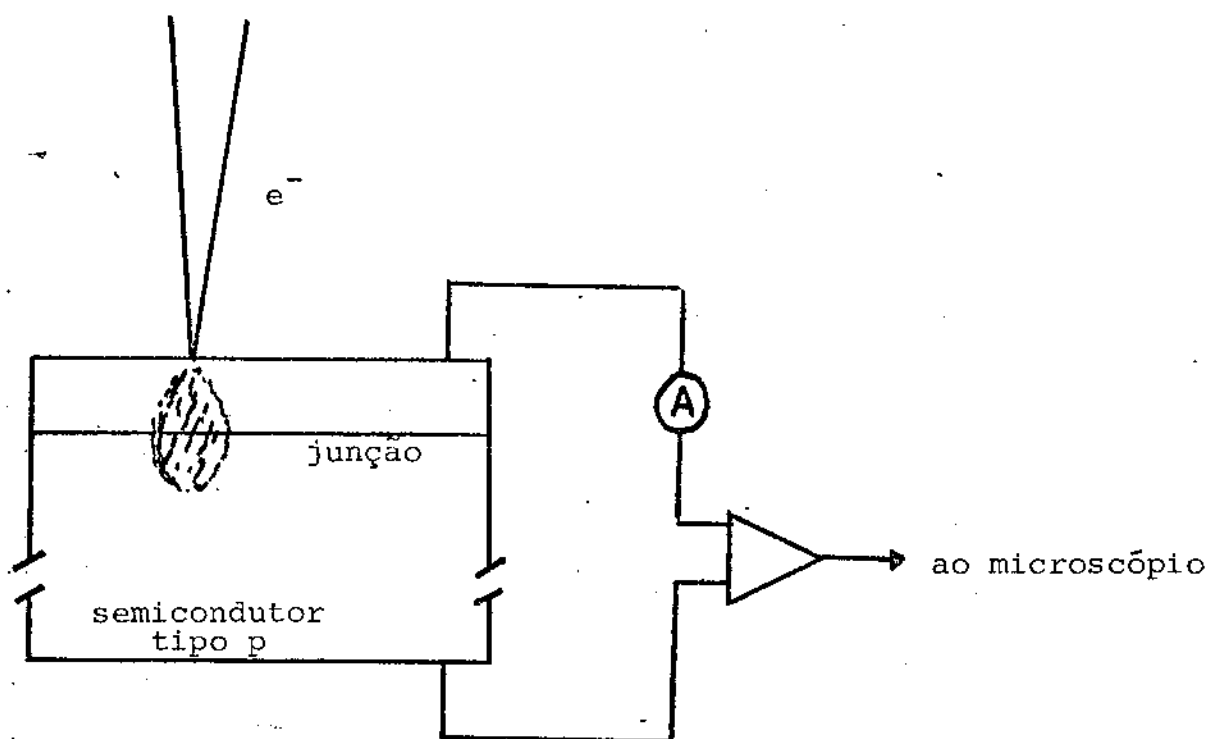


Figura 1

Esquema experimental do modo EBIC para um microscópio eletrônico de varredura

A taxa com que o feixe gera pares é discutida no apêndice E que trata da energia de ionização radiativa. A nossa preocupação neste tópico é o de determinar a difusão dos portadores, uma vez dada a geração de pares em função da posição. Vamos supor todo o processo, geração, difusão, recombinação com sendo estacionário.

A equação que governa o comportamento dos portadores minoritários é a equação de difusão, no caso para um material do tipo p

$$D\nabla^2 n(\vec{r}) - \frac{n(\vec{r})}{\tau(\vec{r})} + g(\vec{r}) = \frac{\partial n(\vec{r})}{\partial t} \quad (1)$$

onde D é a constante de difusão, $n(\vec{r})$ é concentração de elétrons em excesso, $\tau(\vec{r})$ é o tempo de vida local dos portadores e $g(\vec{r})$ é a função de geração determinada pela microsonda. Como já dissemos antes, vamos supor uma situação estacionária, assim o lado direito da equação (1) se anula. Ressaltamos que o tempo de vida dos portadores é uma das principais fontes do contraste de EBIC e que, além da posição, pode ser dependente da concentração de portadores, especialmente sob condições de alta injeção de portadores.

A função $n(\vec{r})$ possui uma função de Green que satisfaz a seguinte equação diferencial

$$D\nabla^2 G(\vec{r}, \vec{r}') - \frac{G(\vec{r}, \vec{r}')}{\tau(\vec{r})} = -g_0 \delta(\vec{r} - \vec{r}') \quad (2)$$

onde $\vec{r}'$ é a posição da fonte. No caso unidimensional a função de Green coincide com a própria função de concentração $n(\vec{r})$. Se acharmos a solução da equação de difusão para a fonte pontual obteremos, com auxílio da função de geração dada pela simulação, a solução total, isto é, a concentração de portadores minoritários gerados pela microsonda.

Equação (2) deve satisfazer às seguintes condições de contorno, vide [7],

$$G(\vec{r}, \vec{r}') = 0 \quad \text{para } \vec{r} \text{ pertencente a superfície da junção} \quad (3)$$

e para todas as outras superfícies

$$D \frac{\partial G(\vec{r}, \vec{r}')}{\partial n_i} = s_i G(\vec{r}, \vec{r}') \quad \vec{r} \in S_i \quad (i=1, 2, \dots, N) \quad (4)$$

onde s_i é a velocidade de recombinação superficial e n_i é a normal ao plano S_i .

Se supormos uma fonte plana e paralela à junção, a simetria reduz o nosso problema ao caso unidimensional que é de tratamento matemático simples e de fácil interpretação. Considerando a equação de Green unidimensional satisfazendo a equação de difusão

$$\frac{\partial^2 G(x, x')}{\partial x^2} - \frac{G(x, x')}{L^2} = -g_0 \frac{\delta(x-x')}{D} \quad (5)$$

onde definimos o comprimento de difusão

$$L = \sqrt{D\tau} \quad (6)$$

e escrevendo a transformada de Fourier da equação (5)

$$-k^2 G_F - \frac{G_F}{L^2} = -\frac{g_0 e^{ikx'}}{\sqrt{2\pi}} \quad (7)$$

portanto a transformada de Fourier da função de Green é

$$G_F = \frac{g_0}{D\sqrt{2\pi}} \frac{e^{ikx'}}{(k^2 + 1/L^2)} \quad (8)$$

onde k é o vetor recíproco. Calculando agora a transformada inversa temos

$$G(x, x') = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} G_F e^{-ikx} dk = \frac{g_0}{4\pi D} \int_{-\infty}^{\infty} \frac{e^{ikx'} e^{-ikx}}{(k^2 + 1/L^2)} dk \quad (9)$$

Esta integral imprópria possui pólos em $1/L$ e $-1/L$, podemos integrá-la então usando o método dos resíduos, obtendo desta forma [8]

$$G(x, x') = \frac{g_0 L}{2D} e^{-|x'-x|/L} \quad (10)$$

A equação (8) difere da solução dada por Hanoka e Bell [11] em um artigo de revisão (equação 10 daquele artigo), cuja solução não satisfaz a equação diferencial (2).

Aplicamos agora as condições de contorno a uma amostra semi-infinita com uma superfície plana em $x=0$, e a fonte pontual a uma distância $x=x'$ da superfície

$$D \frac{\partial G'(x, x')}{\partial x} = S G'(x, x') \quad \text{em } x=0 \quad (4')$$

o que implica em usar o método das imagens, isto é, supor uma fonte virtual em $x=-x'$. Após o ajuste das constantes que resultam ao impor a condição de contorno (4') temos

$$G'(x, x') = C_1 G(x, x') + C_2 G(x, x') =$$

$$= \frac{q_0 L}{2DN} \left\{ e^{-|x'-x|/L} + \frac{D/L-S}{D/L+S} e^{-(x'+x)/L} \right\} \quad (11)$$

onde N é uma constante de normalização que será determinada mais tarde.

Supomos agora uma junção a uma distância x'' da superfície e paralela à esta (vide figura 2), onde aplicamos a condição de contorno

$$G''(x, x') = 0 \quad \text{para } x = x' \quad (3')$$

Novamente temos que recorrer ao método das imagens, supondo uma fonte virtual a uma distância x'' da junção, isto é, a $2x'' + x'$ da superfície. A função de Green após o ajuste das constantes será

$$G''(x, x') = C_1 G'(x, x') + C_2 G'(x, 2x'' + x') =$$

$$= \frac{q_0 L}{2DN} \left\{ e^{-|x'-x|/L} + \frac{D/L-S}{D/L+S} e^{-(x'+x)/L} - e^{-(x-2x''-x')/L} - \frac{D/L-S}{D/L+S} e^{-(x-2x''+x')/L} \right\} \quad (12)$$

É importante notar que esta função de Green só é válida para o intervalo entre a superfície e a junção.

A corrente total coletada na junção será dada por [7]

$$I(x') = -qD \int \frac{\partial G}{\partial n} ds \quad (13)$$

e no nosso caso corresponde a

$$I(x') = qD \int_S \frac{\partial G(x, x')}{\partial x} \delta(x - x') dx \quad (13')$$

o que resulta em

$$I(x') = \frac{qq_0}{N} \left\{ e^{-(x''-x')/L} + \frac{D/L-S}{D/L+S} e^{-(x''+x')/L} \right\} \quad (14)$$

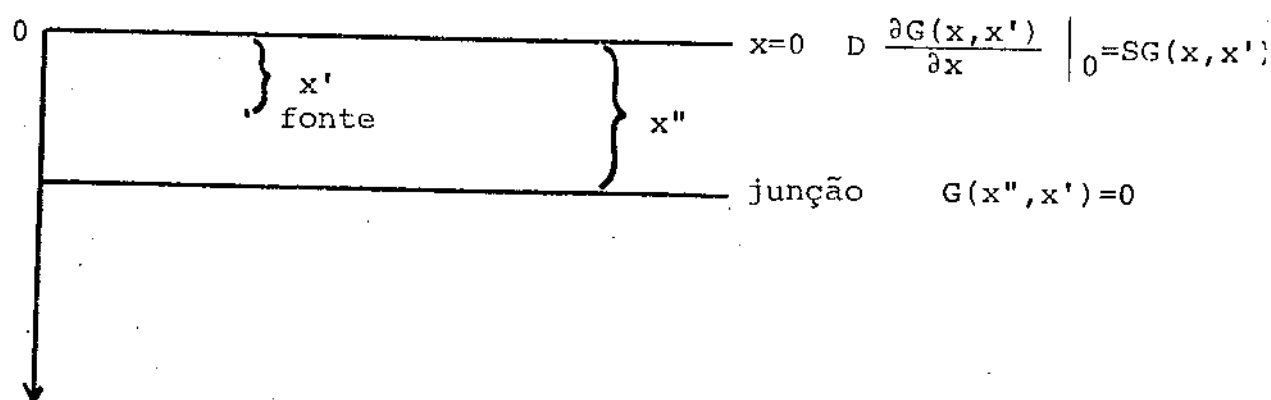


Figura 2

Geometria da amostra semi-infinita com uma junção a uma distância x''

A equação (14) deve satisfazer, para uma fonte localizada na junção, a condição de contorno

$$I(x'') = qg_0 \quad (15)$$

análoga a condição de contorno dada por Donolato [7] para a função corrente.

Satisfeita a condição de contorno (15) obteremos o fator de normalização

$$N = 1 + \frac{D/L-S}{D/L+S} e^{-2x''/L} \quad (16)$$

Finalmente a corrente coletada será

$$I(x') = qg_0 \left(1 + \frac{D/L-S}{D/L+S} e^{-2x''/L} - 1 \left(e^{-(x''-x')/L} + \frac{D/L-S}{D/L+S} e^{-(x''+x')/L} \right) \right) \quad (17)$$

É possível ainda introduzir a simplificação

$$I(x') = qg_0 \frac{\beta(x')}{\beta(x'')} \quad (18)$$

onde definimos a função

$$\beta(x) = e^{x/L} + \frac{D/L-S}{D/L+S} e^{-x/L} \quad (19)$$

que torna evidente que a equação (18) satisfaz a condição (13) em $x' = x''$.

No caso em que a origem da coordenada não coincide com a localização da superfície, por exemplo colocando a superfície a uma distância x_1 , temos a forma mais geral

$$I(x') = qg_0 \frac{\gamma(x', x_1)}{\gamma(x'', x_1)} \quad (20)$$

onde

$$\gamma(x, x_1) = e^{x/L} + \frac{D/L-S}{D/L+S} e^{+2x_1/L} e^{-x/L} \quad (21)$$

Para o caso mais simples onde temos o meio semi-infinito limitado apenas pela junção, sem possuir uma superfície que exige a condição dada pela equação (4), a solução imediata para uma junção a uma distância x'' é

$$I(x') = qg_0 \frac{\alpha(x')}{\alpha(x'')} \quad \text{onde } \alpha(x) = e^{-x/L} \quad (22)$$

Para obter a corrente total produzida por uma distribuição de N fontes pontuais entre a superfície e a junção basta efetuarmos a soma

$$I = \sum_1^N I(x_i) g_0(x_i) \quad (23)$$

onde $g(x')$ é dada pela simulação de Monte Carlo.

Se temos M regiões diferentes (que nos obrigam a aplicar soluções diferentes) a corrente total pode ser escrita como

$$I = \sum_1^M \sum_j^{N_j} I_1(x_j) g_0(x_j) \quad (24)$$

Para o caso de um diodo de junção pn tal como mostrado na figura 3 temos que a corrente total é

$$I_{PN} = q \left(\sum_{i < j} \frac{\beta_{L_1}(x_i)}{\beta_{L_1}(x_j)} g_0(x_i) + \sum_{i > j} \frac{\beta_{L_2}(x_i)}{\beta_{L_2}(x_j)} g_0(x_i) \right) \quad (25)$$

onde x_j é localização da junção.

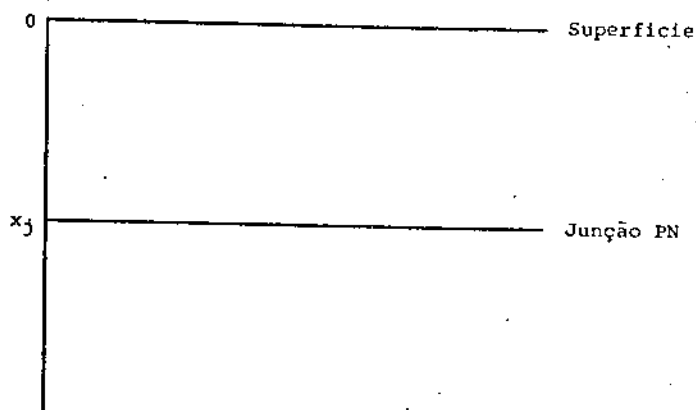


Figura 3

Para um Laser de heteroestrutura dupla mostrado na figura 4 a corrente total será dada por

$$I_{Laser} = q \left(\sum_{i < j} \frac{\beta(x_i)}{\beta(x_k)} g_0(x_i) + \sum_{j < i < k} \frac{\gamma(x_i, x_j)}{\gamma(x_k, x_j)} g_0(x_i) + \sum_{k < i < m} \frac{\alpha(x_i)}{\alpha(x_k)} g_0(x_i) \right) \quad (26)$$

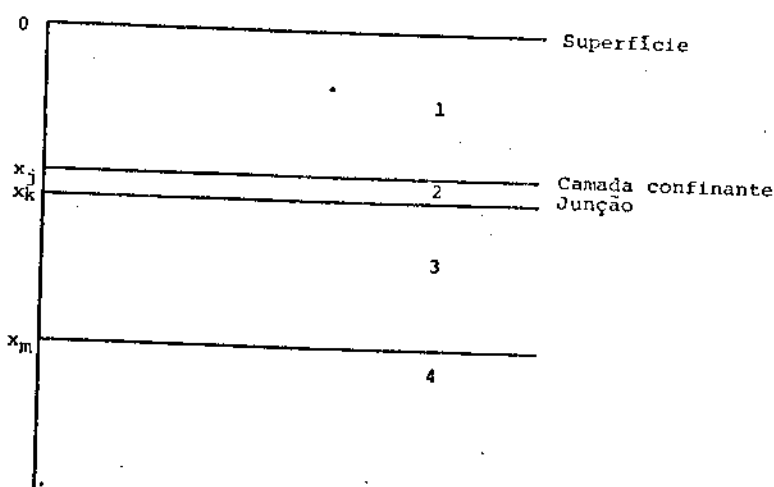


Figura 4

Laser de heteroestrutura dupla.

A razão para este tipo de modelo é que os portadores podem mover-se livremente da camada 1 para a 2 (confinante) mas não o contrário. Na interface 1-2 é conveniente colocar a velocidade de recombinação $s=0$. Isto significa que os portadores não são coletados da superfície o que seria uma propriedade desejável em uma camada que tem por função confinar os portadores. A interface 3-4 não oferece maiores preocupações pois a sonda em condições típicas de trabalho (5-45 keV) não a alcança.

Assumimos em todos estes modelos implicitamente o fato de que o comprimento de difusão L depende da camada em que se localiza a fonte.

Desenvolvemos neste capítulo um modelo simples e versátil para calcular a corrente total coletada a partir de resultados obtidos pelo método de Monte Carlo (resultados do capítulo 2). A aproximação mais importante foi considerar a sonda como um conjunto finito de fontes planas paralelas à junção, ponderadas segundo a energia dissipada em

função da profundidade. Evidentemente é possível considerar um conjunto de fontes pontuais mas neste caso seremos obrigados a usar curvas de perda de energia em três dimensões. Naturalmente esbarraremos em limitações como tempo de processamento computacional e espaço de memória em detrimento da precisão do cálculo.

Fica em aberto a escolha do comprimento de difusão dos portadores e das velocidades de recombinação superficial. Esta escolha deve seguir por considerações de natureza qualitativa das propriedades dos materiais envolvidos. O GaAs por exemplo terá a velocidade de recombinação tipicamente tendendo a infinito.

4 - Resultados da Simulação do EBIC

Para utilizarmos os resultados do capítulo precedente é necessário adicionar alguns parâmetros tais como a velocidade de recombinação superficial e o comprimento de difusão nas diversas regiões do dispositivo.

A velocidade de recombinação superficial pode ser tomada em dois extremos, $s=0$ e $s \rightarrow \infty$, neste caso a função (4.21) fica

$$\gamma(x, x_1) = e^{x/L} + e^{2x_1/L} e^{-x/L} \quad (1)$$

onde o sinal positivo está para a velocidade de recombinação nula e o negativo para a velocidade de recombinação tendendo ao infinito. Desta forma não será necessário nos preocuparmos com o coeficiente de difusão D que é eliminado desta forma.

Resta determinarmos o comprimento de difusão L . Aqui não temos como definirmos extremos e as medidas experimentais de L são as mais desconhecidas possíveis. Podemos por outro lado usar os modelos e ajustá-los para obtermos os mesmos valores das correntes coletadas medidas experimentalmente e estabelecer desta forma os valores de L .

Vamos apresentar primeiro alguns resultados para diodos de homojunção pn tal como discutido no capítulo 3, utilizando a solução (3.18)

$$I(x') = q g_0 \frac{\beta(x')}{\beta(x'')} \quad (2)$$

Como pode-se notar a equação (2) tem termos separados em função da localização da fonte (x') e da função (x''). Podemos nos valer desta pro-

priedade para calcularmos a corrente de EBIC de diversos diodos de homojunção com profundidade de junção diferentes. Isto é possível devido ao fato de que a microsonda não é sensível à dopagem do meio. Em outras palavras, a concentração das impurezas é muito pequena para influir no espalhamento dos elétrons do feixe. Desta forma podemos calcular a corrente EBIC de muitos diodos diferentes com os resultados de apenas uma única simulação Monte Carlo para um determinado semicondutor. Por exemplo diodos de GaAs com profundidades de junção a 1,2,3 etc. micron.

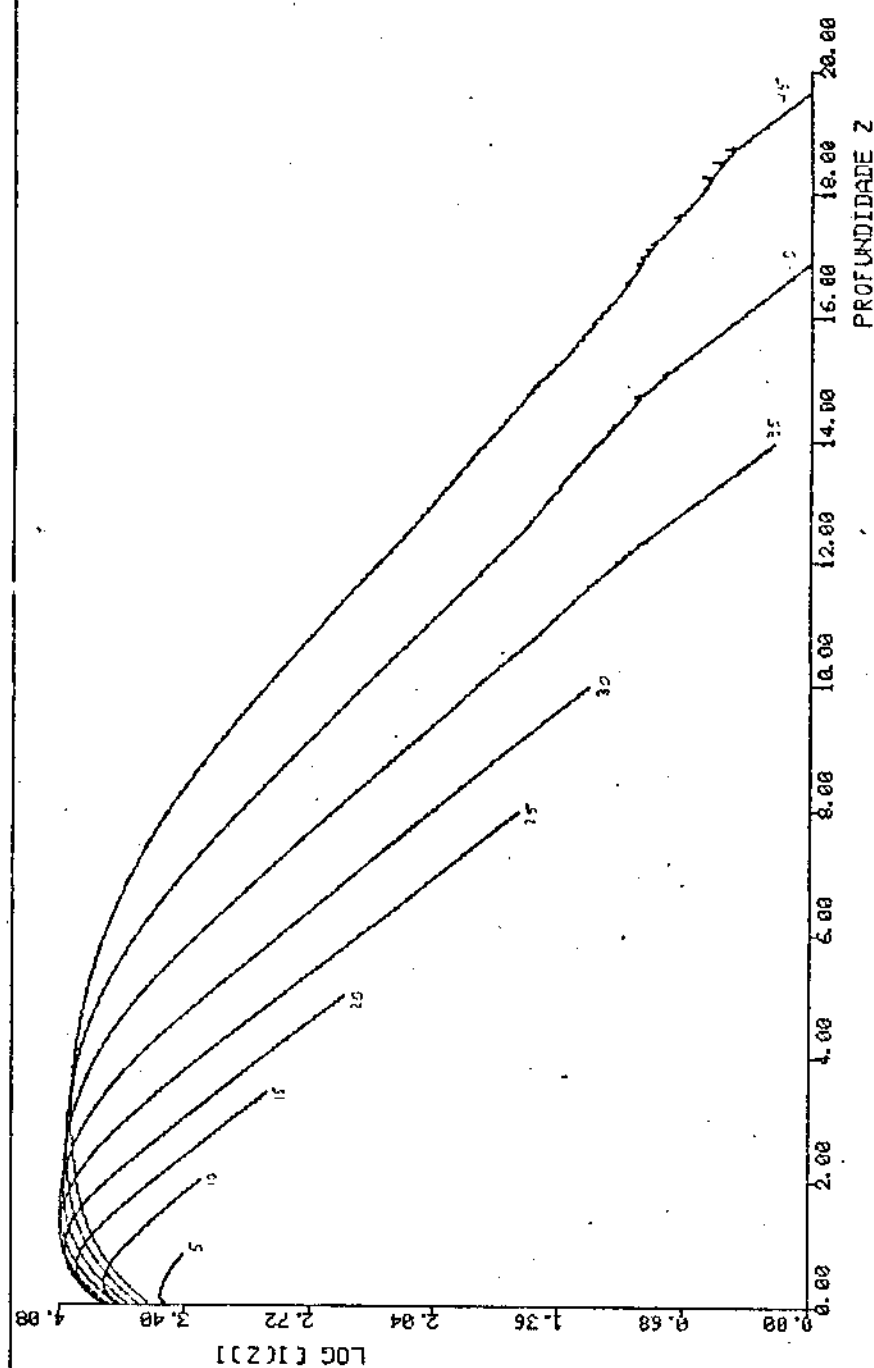
A forma da solução (2) nos permite montar um esquema de cálculo extremamente simplificado quando desejamos calcular a corrente a diversas profundidades diferentes da junção. De outra forma os cálculos levariam um tempo de computação muito maior.

Lembramos que a solução para o diodo se escreve como sendo (equação (3.25))

$$I_{PN} = q \left(\sum_{i < j} \frac{\beta_{L_1}(x_i)}{\beta_{L_1}(x_j)} g_0(x_i) + \sum_{i > j} \frac{\beta_{L_2}(x_i)}{\beta_{L_2}(x_j)} g_0(x_i) \right) \quad (3)$$

As funções α e β são calculadas em todo o intervalo que interessa e armazenados em um vetor. Feito isto basta fazer as somas da equação (3). O que evitamos, ao final de tudo, é recalcular as mesmas coisas muitas vezes. Na figura 1 e 2 mostramos os resultados para um diodo de GaAs com comprimentos de difusão idênticos antes e depois da junção. O que variamos no caso é a profundidade da junção. As correntes são calculadas até o alcance da microsonda. Na figura 1 temos a velocidade de recombinação superficial nula e na figura 2 esta velocidade é infinita. Podemos notar duas propriedades ao comparar as duas figuras. A influência da superfície diminui quando a microsonda penetra muito na

amostra (alta tensão do feixe) e quando a junção está longe da superfície como seria de se esperar. Nas figuras 3 e 4 repetimos os cálculos para um comprimento de difusão muito grande antes da junção em relação ao comprimento de difusão depois da junção. As figuras 5 e 6 invertem a situação, desta vez o comprimento de difusão é maior depois da junção.



CORRENTE COLETADA NA JUNCÃO

ARSENETO DE GALIO

PURO

 $L(1)=1.0$ $L(2)=1.0$ $S=0$

REGISTRO NR.: 166

MICROSCOPIA ELETRÔNICA

SIMULAÇÃO MONTE CARLO

LPD LABORATORIO DE PESQUISA
EM DISPOSITIVOS

Figura 1

Diodo de GaAs, calculado para $L(1)=1$ e $L(2)=1$. Velocidade de recombinação superficial nula.

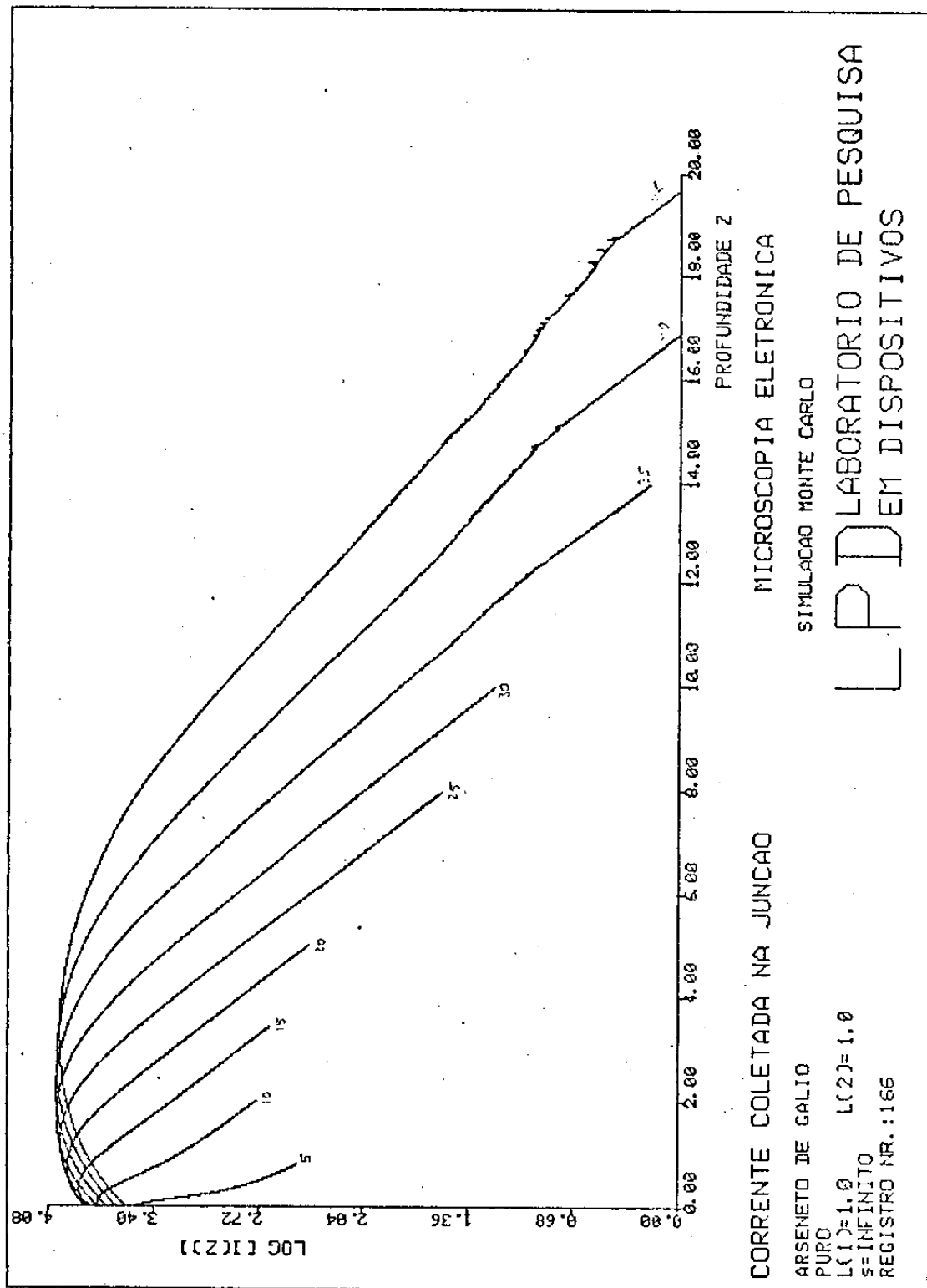


Figura 2

Idem Figura 1, para velocidade de recombinação superficial infinita.

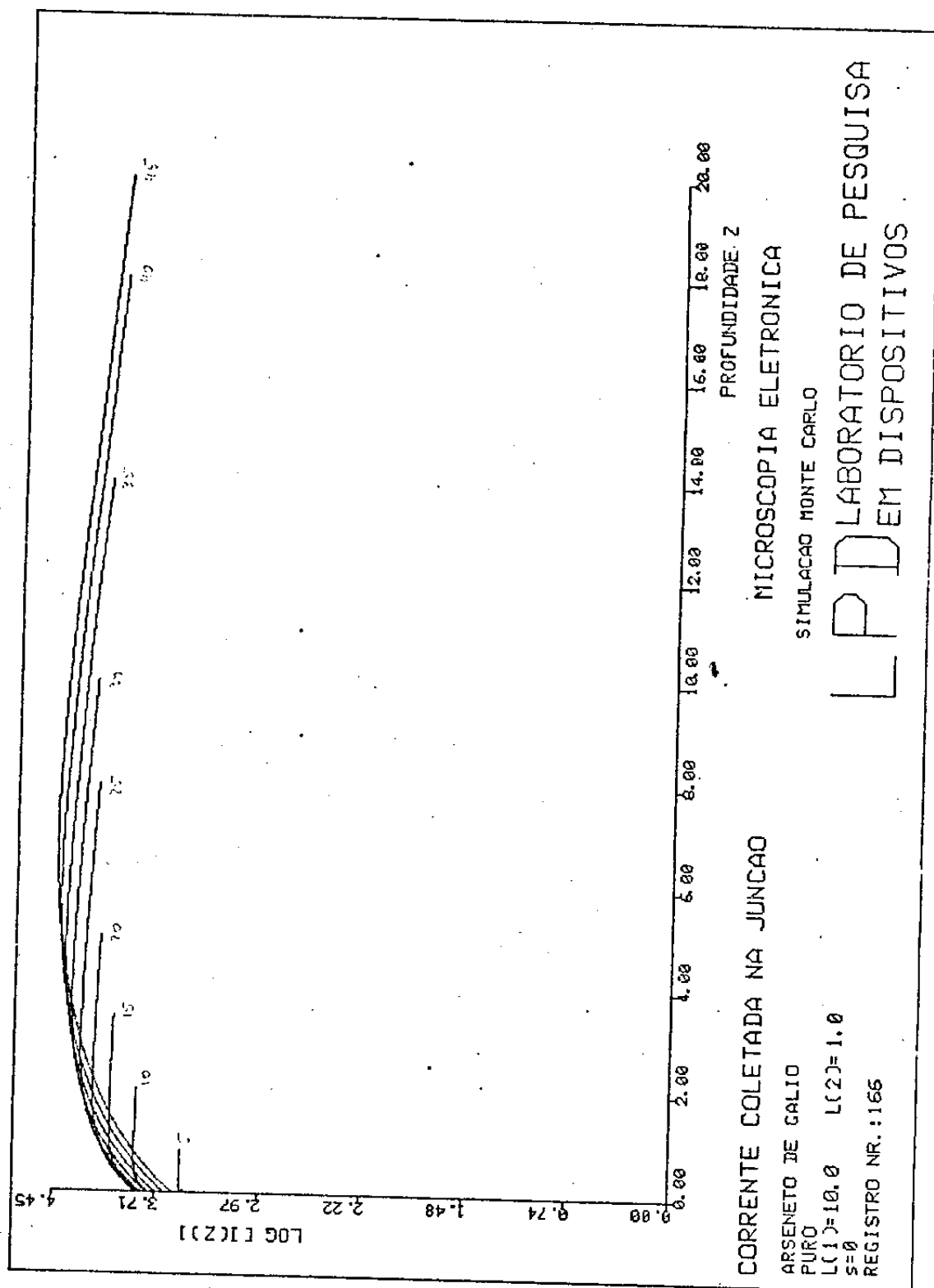


Figura 3

Diodo de GaAs, calculado para $L(1)=10$ e $L(2)=1$. Velocidade de recombinação superficial nula.

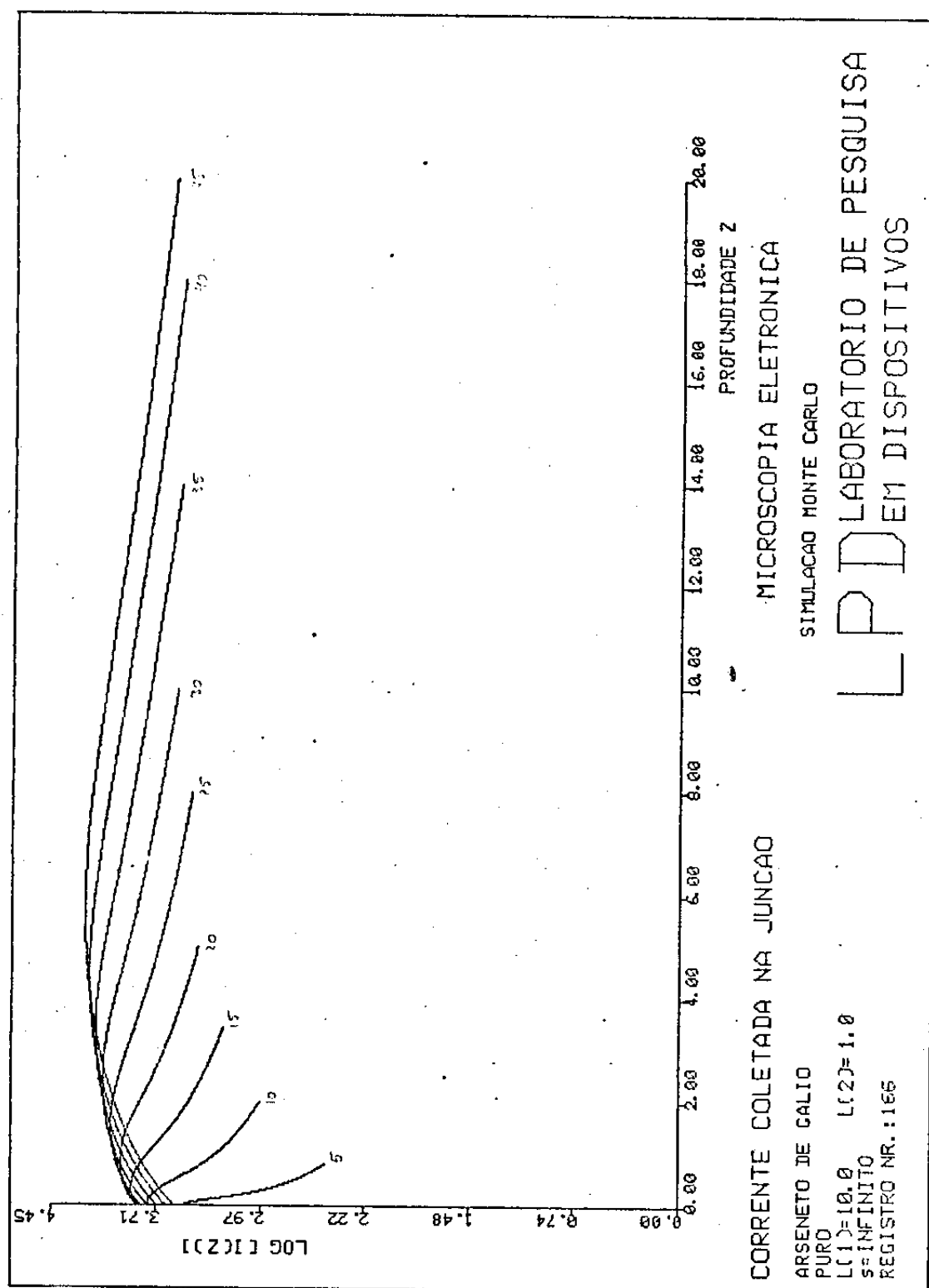


Figura 4

Idem Figura 3, para velocidade de recombinação superficial infinita.

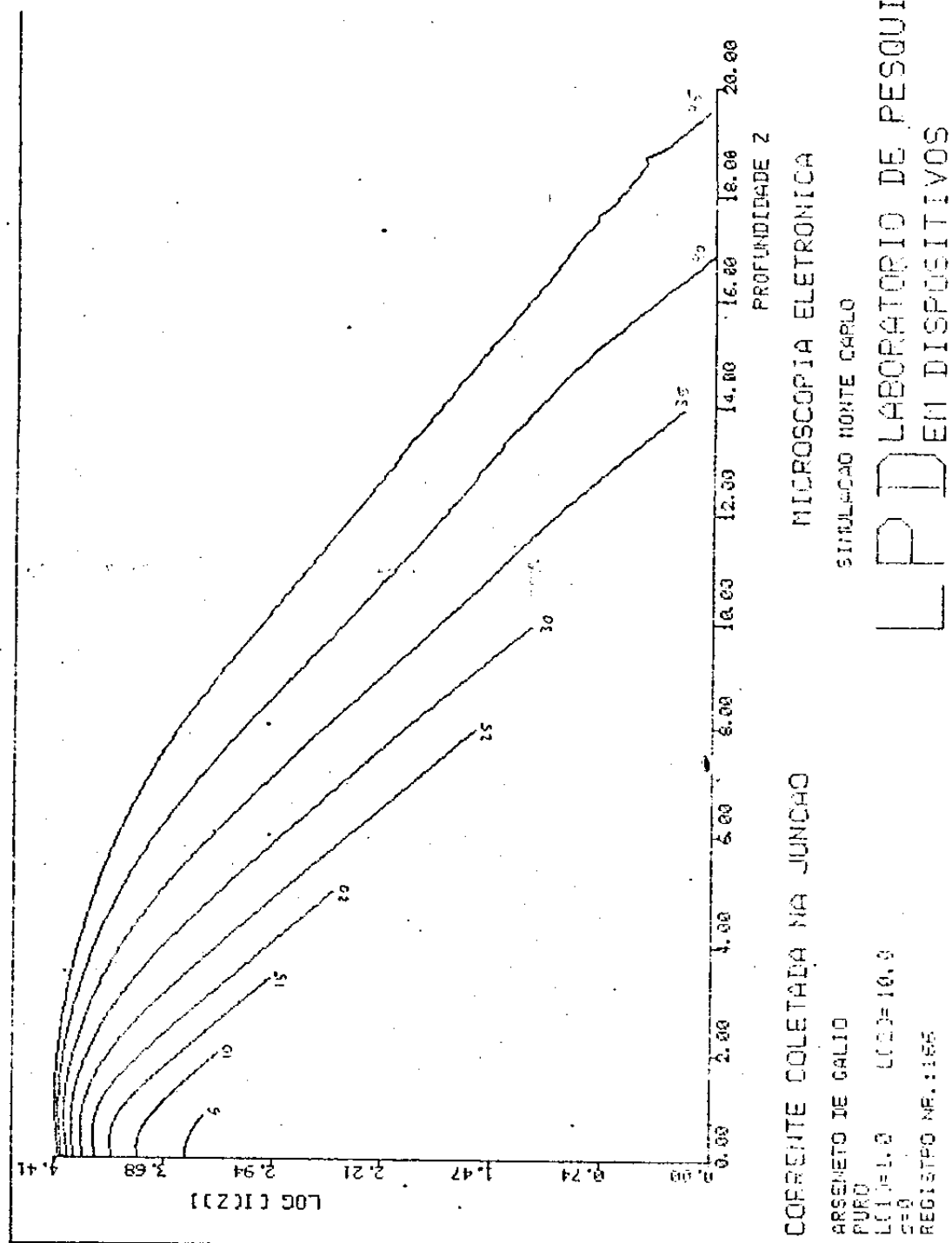


Figura 5

Diodo de GaAs, calculado para $L(1)=1$ e $L(2)=10$. Velocidade de recombinação superficial nula.

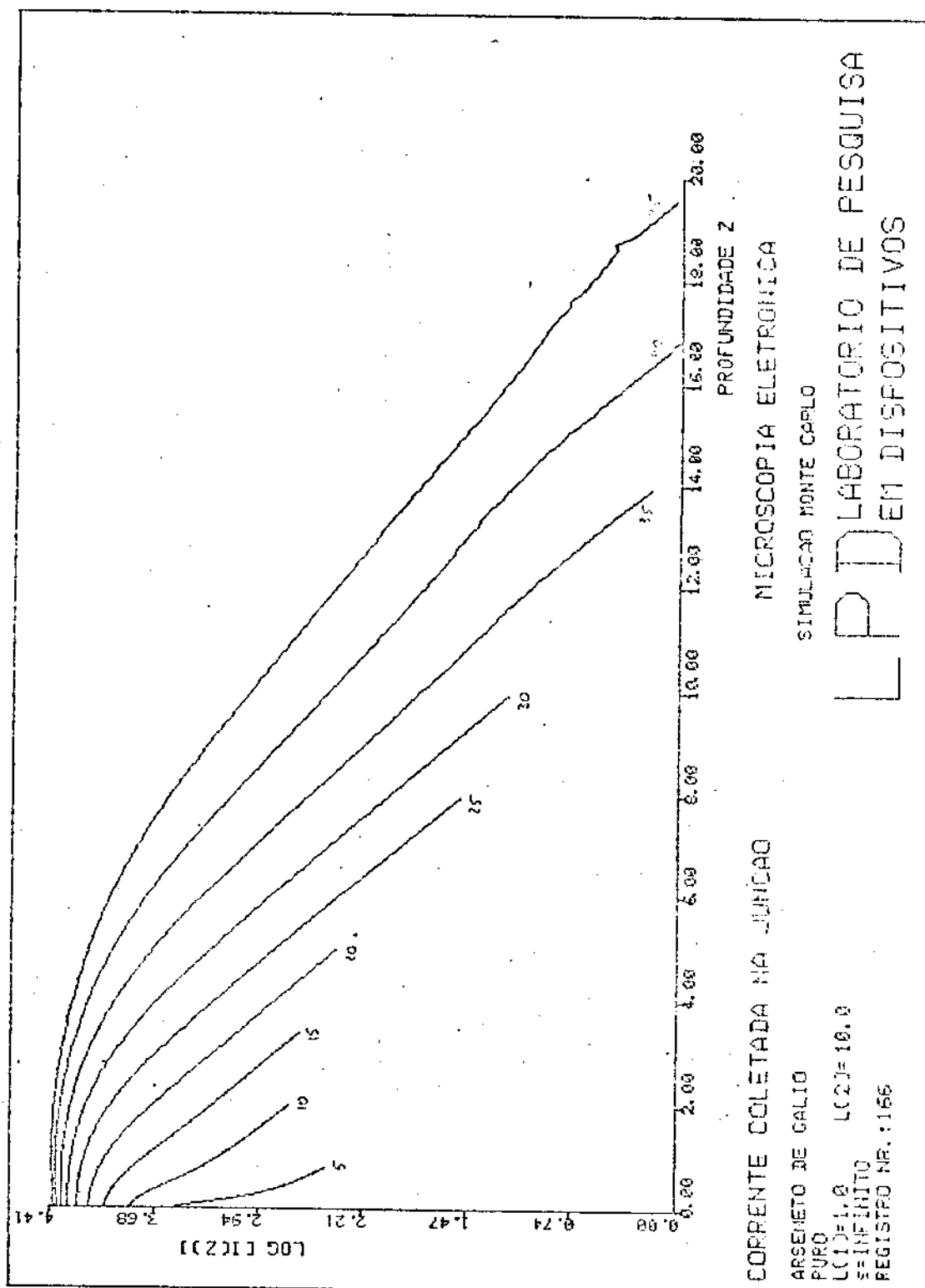


Figura 6

Idem Figura 5, para velocidade de recombinação superficial infinita.

Para um laser tal como o da figura 7 não é muito interessante o tipo de cálculo que efetuamos para o diodo. Isto porque as dimensões não são facilmente variáveis, ou seja, teriam que ser crescidos vários tipos de diodos diferentes.

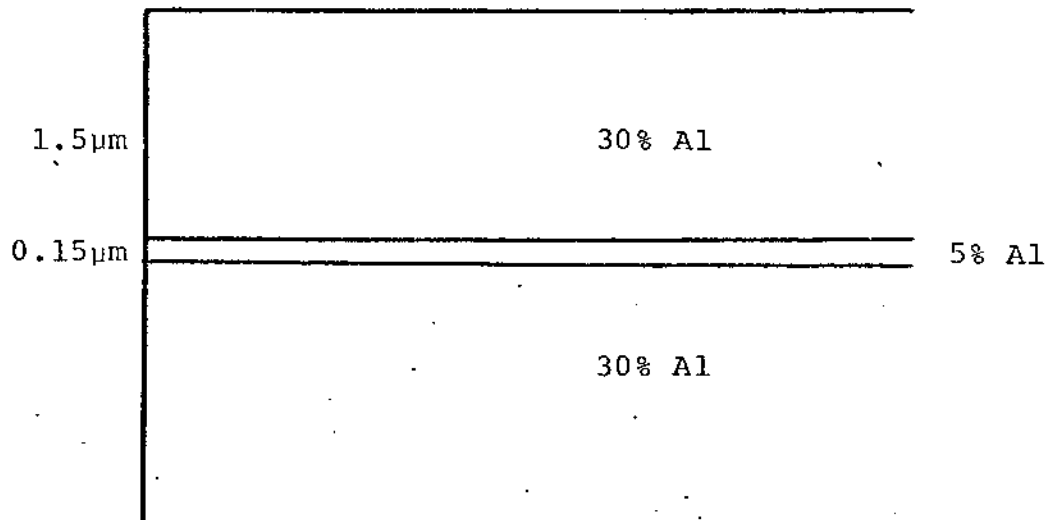


Figura 7

Laser de Heteroestrutura dupla

Existe grande interesse na obtenção de comprimentos de difusão a partir de medidas experimentais. Com o modelo que desenvolvemos é possível estimar esses comprimentos de difusão de cada camada do laser. Para isto usamos as medidas e os cálculos para diferentes tensões do feixe. Supomos que os comprimentos de difusão dos portadores não são afetados pela variação da tensão. Isto é razoável e encontra embasamento experimental. A questão da produção de portadores é discutida no apêndice E.

Usando as medidas para um laser de GaAsAl com as características indicadas na figura 7, podemos fazer um ajuste de curvas de forma a obter o melhor conjunto de comprimentos de difusão. Neste caso encontramos 0.3, 0.1 e 2.3 micron para as camadas indicadas como sendo a 1, 2 e a 3 na figura 7, respectivamente. Os dados das correntes (na realidade indicamos os ganhos de corrente sobre a corrente do feixe) estão indicados na tabela 1. Os cálculos foram realizados para uma velocidade de recombinação superficial infinita. Isto é razoável pois a recombinação superficial do GaAs é muito alta.

Energia	I ₁ (calculada)	I (experimental)
5	1.491	0
10	14.407	32
15	116.000	120
20	574.000	540
25	1364.000	1000
30	1575.000	1300
35	1501.000	1500
40	1327.000	1550
45	1161.000	1500

Tabela 1

Para efetuar este tipo de simulação é necessário calcular previamente a densidade específica e o "gap" do material de determinada camada. Isto em geral é feito por interpolações. Esquemas completos para

compostos ternários (GaAsAl) encontramos no artigo de revisão de Adachi [11], e para quaternários (InGaAsP) no artigo também de Adachi [2].

5 - Conclusão

Desenvolvemos uma ferramenta que nos permite calcular a corrente induzida por feixe eletrônico para um diodo simples e um laser de heterojunção.

A intenção principal deste trabalho foi o de substituir os modelos analíticos de microsonda por uma distribuição mais realista obtida a partir da simulação Monte Carlo. A vantagem adicional consiste no fato de podermos assim lidar com estruturas complexas que não poderiam ser abordadas por nenhum modelo analítico.

Uma das possíveis aplicações práticas e imediata seria a estimativa do comprimento de difusão dos portadores. Mas a motivação do nosso trabalho não foi esta. Criamos uma ferramenta e a testamos a partir de considerações e modelos simples e demonstramos a sua coerência. O fruto principal deste trabalho é o leque de aplicações e estudos que se abre daqui em diante. Desenvolver modelos que abordem outras situações experimentais de EBIC e de catodoluminescência tornou-se bem mais simples. Eis uma pequena lista do que pode ser feito:

- a) Análise de defeitos por EBIC. Diferentes tipos de defeitos podem ser propostos e seus efeitos sobre a corrente podem ser estudados e correlacionados com dados experimentais
- b) Estudos de contraste e resolução em EBIC e catodoluminescência.
- c) Estudo de catodoluminescência espectral. Isto pode ser um trabalho valiosíssimo uma vez que pode se lançar assim as bases

para a catodoluminescência quantitativa. O aparato teórico é muito parecido com o EBIC e não deve oferecer maiores dificuldades.

Naturalmente estes tópicos podem ser estendidos e subdivididos dependendo das aplicações.

A - Equação de Perda de Energia

O processo mais importante a ser considerado na simulação da trajetória de um elétron em um sólido é sem dúvida a perda de energia ao longo desta mesma trajetória. A perda de energia se dará nas colisões inelásticas com os elétrons dos átomos. Bethe [1] em 1930 deduziu a expressão quântica para esta perda de energia que é adotada mesmo nos trabalhos mais recentes em simulação Monte Carlo [2]. Não deduziremos aqui a equação de Bethe pelo procedimento quântico usual mas sim faremos o desenvolvimento de um cálculo semi-clássico, talvez um pouco menos rigoroso, que possui a vantagem de ser muito mais direto e menos tedioso. Para tanto usaremos um procedimento semelhante ao que foi usado por Jackson [3] para deduzir a perda de energia clássica de Bohr.

Imaginaremos inicialmente um elétron colidindo com um outro, em repouso, a uma velocidade v e com uma energia dada por

$$E = \gamma Mc^2 \quad (\text{A.1})$$

onde γ é dado por

$$\gamma = (1 + \beta^2)^{-1/2}$$

e b é o chamado parâmetro de impacto do elétron incidente.

Agora será necessário avaliarmos o campo causado pela partícula incidente.

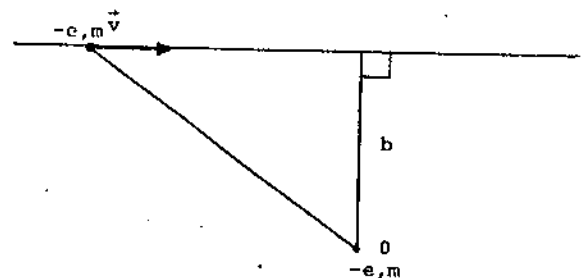


Figura A.1

Para calcularmos este campo consideremos a geometria indicada na figura A.2 onde indicamos dois referenciais cartesianos, um deles se movendo a uma velocidade v do outro. Fica claro que os eixos cartesianos correspondentes dos dois referenciais são sempre paralelos, e supomos que o desvio angular da partícula incidente é suficientemente pequeno para ser desprezado.

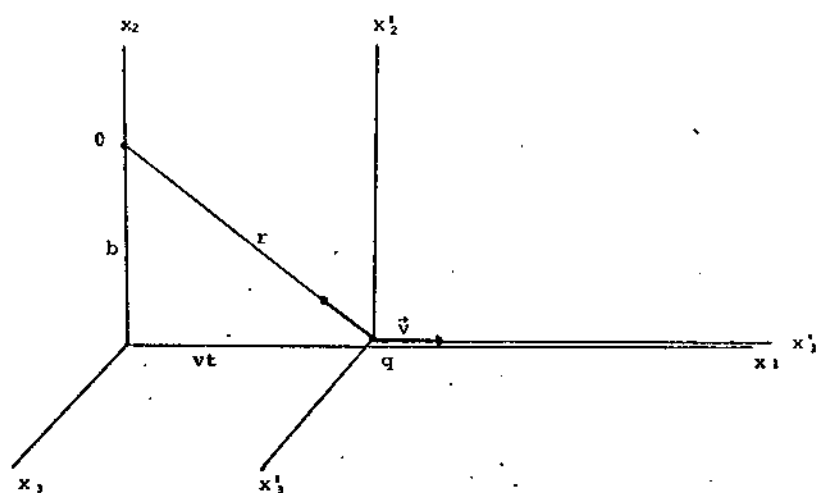


figura A.2

Pode se mostrar que, usando as transformações de Lorentz, os campos no referencial do elétron em repouso no ponto 0 serão

$$E_1 = + \frac{eyvt}{(b^2 + \gamma^2 v^2 t^2)^{3/2}} \quad (A.2)$$

$$E_2 = - \frac{\gamma eb}{(b^2 + \gamma^2 v^2 t^2)^{3/2}}$$

com todas as outras componentes nulas. Ainda pode se mostrar que o intervalo de tempo durante o qual os campos serão apreciáveis é

$$\Delta t = \frac{b}{\gamma v} \quad (A.3)$$

O intervalo de tempo dado por (A.3) se relaciona ao fato de que os campos dados por (A.2) têm a forma de lorentzianas de largura $b/\gamma v$.

Para calcularmos o impulso verificamos que apenas a componente transversal não fornece um integral nula no tempo. Assim o impulso será dado por

$$\Delta p = - \int_{-\infty}^{\infty} eE(t) dt = - \frac{2e^2}{bv} \quad (A.4)$$

e a energia transferido ao elétron é

$$\Delta E(b) = - \frac{(\Delta p)^2}{2m} = - \frac{2e^4}{mv^2} \left(\frac{1}{b^2} \right) \quad (A.5)$$

O elétron ao atravessar o sólido passa por diversos átomos e portanto a diversos valores de parâmetros de impacto. Vamos supor então que temos N átomos por unidade de volume, com Z elétrons por átomo. O número de elétrons localizados entre b e b+db em uma espessura dx, a qual supomos muito menor que o livre caminho médio de espalhamento elástico, será

$$dn = NZ 2\pi b db dx \quad (A.6)$$

Para calcularmos a perda de energia por unidade de distância percorrida pela partícula efetuamos a integral

$$\frac{dE}{dx} = - 4\pi NZ \frac{e^2}{mv^2} \int \Delta E(b) b db \quad (A.7)$$

vamos considerar agora os limites de integração escrevendo

$$\frac{dE}{dx} = - 4\pi NZ \frac{e^2}{mv^2} \int_{b_{\min}}^{b_{\max}} \frac{db}{b} = - 4\pi NZ \frac{e^2}{mv^2} \ln \frac{b_{\max}}{b_{\min}} \quad (A.8)$$

é necessário portanto avaliar entre quais limites a equação (A.8) é válida.

Para o limite máximo dizemos que ocorre quando o tempo em que os campos são consideráveis com o período do movimento do elétron, isto é

$$\Delta t(b_{\max}) \sim \frac{1}{\omega}$$

onde

$$b_{\max} = \frac{Yv}{\omega} \quad (A.9)$$

e ω é uma frequência atômica característica.

Para obter o limite mínimo consideramos o princípio de incerteza de Heisenberg

$$\Delta E \Delta t > \frac{\hbar}{2}$$

e novamente usando a equação (A.3) e aproximando

$$\Delta t \sim \frac{\hbar}{2E}$$

temos

$$b_{\min} = \frac{\hbar \gamma v}{2E} \quad (A.10)$$

o que leva finalmente a

$$\frac{dE}{dx} = -2\pi N Z \frac{e^2}{E} \ln \frac{2E}{\hbar \omega} \quad (A.11)$$

que é exatamente a equação de Bethe para a perda de energia num sólido calculada por procedimentos puramente quânticos.

B - Livre Caminho Médio

Vamos supor um potencial atômico blindado da forma

$$U = -\frac{Ze^2}{r} e^{-r/a} \quad (B.1)$$

A secção de espalhamento dentro da aproximação de Born será [1]

$$d\sigma = 4\pi^2 \left(\frac{Ze^2 ma}{\hbar^2} \right) \frac{d\Omega}{(q^2 a^2 + 1)^2} \quad (B.2)$$

onde $d\Omega$ é o elemento de ângulo sólido e q é dado por

$$q = 2k \sin \frac{\theta}{2} \quad (B.3)$$

onde k é o vetor de onda.

A secção de transporte da qual calcularemos o livre caminho médio é dada por [1]

$$\sigma_{tr} = \int_0^\pi (1 - \cos\theta) d\sigma = \int_0^\pi \frac{2k}{2k^2} d\sigma \quad (B.4)$$

o elemento de ângulo sólido se escreve em função de q

$$d\Omega = 2\pi \sin\theta d\theta = \frac{2\pi}{k^2} q dq \quad (B.5)$$

de onde obtemos

$$\sigma_{tr} = \frac{\pi}{k^4} \left(\frac{Ze^2 ma}{\hbar^2} \right)^2 \int_0^{2k} \frac{q^3 dq}{(q^2 a^2 + 1)^2} \quad (B.6)$$

fazendo a substituição $q^2 = x$ temos

$$\sigma_{tr} = \frac{4\pi}{k^4} \left(\frac{Ze^2 ma}{\hbar^2} \right)^2 \int_0^{4k^2} \frac{x dx}{2(xa^2 + 1)^2} \quad (B.7)$$

a integral em (B.7) pode ser resolvida por partes

$$\int_0^{4k^2} \frac{x dx}{(xa^2 + 1)^2} = -\frac{x}{a^2(xa^2 + 1)} \Big|_0^{4k^2} + \int_0^{4k^2} \frac{dx}{a^2(xa^2 + 1)} = -\frac{4k^2}{a^2(4k^2a^2 + 1)} + \frac{1}{a^4} \ln(4k^2a^2 + 1)$$

finalmente

$$\sigma_{tr} = \frac{2\pi Z^2 e^4 m^2}{k^4 \hbar^4} \left[\frac{(4k^2a^2 + 1) \ln(4k^2a^2 + 1) - 4k^2a^2}{(4k^2a^2 + 1)} \right] \quad (B.8)$$

para usarmos a aproximação de Born é feita a hipótese de que

$$ka \gg 1$$

então a equação (B.8) torna-se

$$\sigma_{tr} = \frac{2\pi Z^2 e^4 m^2}{k^4 \hbar^4} \ln(4k^2 a^2 - 1) \quad (B.9)$$

usando ainda a relação

$$k^2 = \frac{2mE}{\hbar^2}$$

temos

$$\sigma_{tr} = \frac{\pi Z^2 e^4}{2 E^2} \left(\ln\left(\frac{8mEa^2}{\hbar^2}\right) - 1 \right) \quad (B.10)$$

Devemos agora avaliar a, colocando-o em termos do raio do átomo no modelo de Thomas-Fermi

$$a = \frac{\hbar^2 Z^{1/3}}{me^2}$$

de onde vem

$$\sigma_{tr} = \frac{\pi Z^2 e^4}{2 E^2} \left(\ln\left(\frac{8E\hbar^2 Z^{2/3}}{e^2}\right) - 1 \right) \quad (B.11)$$

Para avaliarmos o livre caminho médio usamos a relação

$$\lambda_{tr} = \frac{A}{\rho N_0 \sigma_{tr}} \quad (B.12)$$

onde A é a massa atômica, ρ a densidade (em g/cm³) e N_0 a constante de Avogadro. Desta forma obtemos

$$\lambda_{tr} = \frac{2AE^2}{\pi Z^2 e^4 N_0 \rho} \left(\ln\left(\frac{8E\hbar^2 Z^{2/3}}{e^2}\right) - 1 \right)^{-1} \quad (B.13)$$

substituímos as constantes e convertendo convenientemente as unidades temos, em uma forma prática, o livre caminho médio

$$B.13' \quad \lambda_{tr} = 0.5112 \frac{AE^2}{\rho Z^2} \left(\ln(293EZ^{2/3}) - 1 \right)^{-1} \quad (B.13')$$

que é dado em micron embora a densidade continua mantida nas unidades originais e a energia em keV.

Com o livre caminho médio assim determinado calculamos, para o efeito da simulação Monte Carlo, o livre caminho percorrido como função de uma variável aleatória R_1 [2]

$$\lambda(R_1) = -\lambda_{tr} \ln(R_1) \quad (B.14)$$

onde R_1 tem uma distribuição uniforme e o livre caminho percorrido uma distribuição de Poisson.

C - Ângulo de Espalhamento

Vamos supor uma partícula movendo-se em um campo central U com uma certa energia cinética E . É conhecido da mecânica clássica que o ângulo ϕ_0 descrito pelo movimento não ligado é [1]

$$\phi_0 = \int_{r_{\min}}^{\infty} \frac{M/r^2 dr}{\sqrt{2m(E-U(r)) - M^2/r^2}} \quad (C.1)$$

onde M é o momento da partícula e usamos a geometria indicada na figura (C.1) onde ρ é o parâmetro de impacto.

O ângulo de desvio ou de espalhamento estará determinado por

$$\chi = |\pi - 2\phi_0| \quad (C.2)$$

Para efeito de cálculo colocamos o momento em função da energia.

$$M = mp\sqrt{2mE} \quad (C.3)$$

Substituindo na equação (C.1) e integrando para um potencial coulombiano

$$U = -\frac{(Ze^2)}{r} \quad (C.4)$$

temos

$$\phi_0 = \arccos \frac{Ze^2/2E\rho}{\sqrt{1+(Ze^2/E\rho)^2}} \quad (C.5)$$

usando a relação (C.2) temos

$$\rho^2 = \frac{Z^2 e^4}{4E^2} \operatorname{ctg}^2 \frac{\theta}{2} \quad (C.6)$$

ou, finalmente

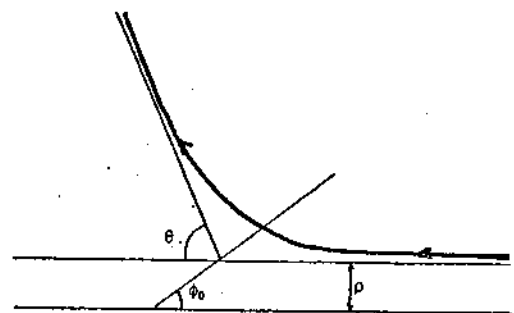


Figura C.1

$$\theta = 2 \arctg \left(\frac{Ze^2}{2\rho E} \right) = 2 \arctg (R_2 b Z / E) \quad (C.7)$$

onde R_2 é uma variável aleatória uniformemente distribuída entre 0 e 1 e b é um parâmetro ajustável a partir de valores experimentais. Em geral ajusta-se este valor para fornecer os coeficientes de retroespalhamento corretos.

D - Coordenadas do elétron em três dimensões

Para calcularmos a posição do elétron após cada espalhamento elástico é necessário ter um sistema de coordenadas sempre atualizado para podermos aplicar o espalhamento Rutherford.

Baseamo-nos aqui praticamente na técnica usada por Myklebust et alii. A idéia chave desta técnica consiste em uma mudança constante do sistema de coordenadas, mantendo, porém, sempre as relações com um sistema de coordenadas definido pela amostra, acompanhando o movimento do elétron. O sistema de coordenadas definido pela amostra é mostrado na figura D.1.

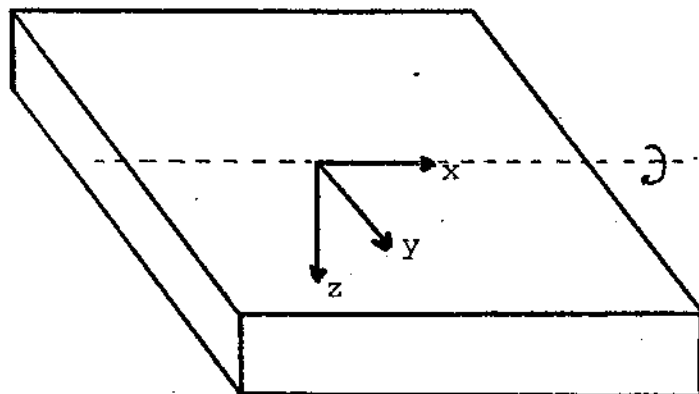


Figura D.1

O objetivo do sistema de coordenadas que acompanha o elétron é manter um eixo z' sempre na direção do movimento para aplicar de forma direta o modelo de espalhamento de Rutherford (vide apêndice C).

Vamos basear o nosso esquema de cálculo na figura D.2.

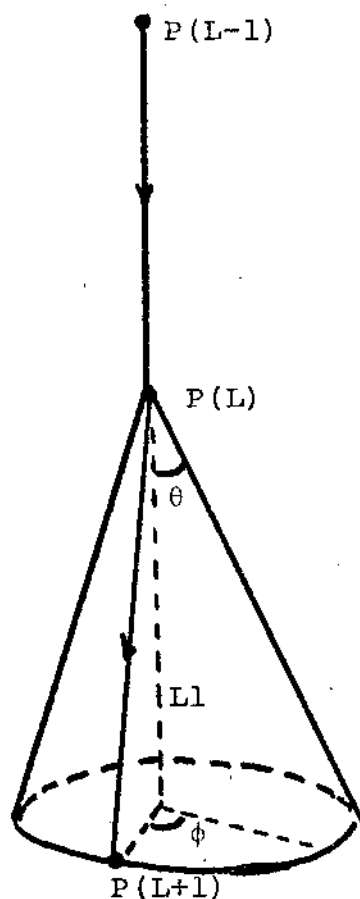


Figura D.2

O elétron vem do ponto $P(L-1)$ para o ponto $P(L)$ onde é espalhado por um ângulo θ em relação à direção do seu movimento e em um ângulo azimutal e uniformemente distribuído. Os cossenos diretores da reta L podem ser calculados a partir das coordenadas X, Y, Z dos pontos $P(L-1)$, $P(L)$. Depois de espalhado, o elétron vai para um ponto genérico $P(L+1)$ no perímetro da base do cone, cuja altura é o comprimento do segmento $L1$.

Para calcular $P(L+1)$, calculamos primeiro (X_0, Y_0, Z_0) , que representa o ponto de intersecção do plano com o segmento $L1$.

Seja $D(L)$ a distância entre $P(L)$ a $P(L+1)$ e a altura do cone H .
Então

$$H=D(L)\cos\theta \quad (D.1)$$

o raio da base

$$R=D(L)\sin\theta \quad (D.2)$$

as coordenadas (X_0, Y_0, Z_0)

$$X_0=X(L)+(Ha) \quad (D.3a)$$

$$Y_0=Y(L)+(Hb) \quad (D.3b)$$

$$Z_0=Z(L)+(Hc) \quad (D.3c)$$

Em seguida estabelecemos um novo conjunto de eixos de coordenadas (X', Y', Z') como na figura D.3, com o plano $X'Y'$ contendo a base do cone e o eixo Z' coincidindo com o segmento $L1$.

O eixo X é escolhido de modo a ser a reta de intersecção do plano da base com o plano $Y=Y_0$. Do conhecimento do ponto no plano da base (X_0, Y_0, Z_0) e dos cossenos diretores da normal ao plano, a equação da base do cone é:

$$aX+bY+cZ-(aX_0+bY_0+cZ_0)=0 \quad (D.4)$$

a reta de intersecção com $Y=Y_0$

$$aX+cZ-(aX_0+cZ_0)=0 \quad (D.5)$$

tem a forma

$$Z=-\frac{a}{c}X+K \quad K=\frac{aX_0}{c}+Z_0 \quad (D.6)$$

e os cossenos diretores de X' , com respeito aos eixos originais, são:

$$a_{X'}=\cos(\arctg(-a/c)) \quad (D.7a)$$

$$b_{X'}=\cos 90^\circ=0 \quad (D.7b)$$

$$c_{X'}=\cos(\arctg(-c/a)) \quad (D.7c)$$

Conhecendo os cossenos diretores X' e Z' , pode-se obter os cossenos diretores de Y' .

Para manter o sistema como de "mão direita", faz-se o produto vetorial de forma a obter

$$a_y' = b_z', \quad c_x' \quad (D.8a)$$

$$b_y' = c_z', \quad a_x' = -c_x', \quad a_z' \quad (D.8b)$$

$$c_y' = -a_x', \quad b_z' \quad (D.8c)$$

O ponto $P(L+1)$ é então

$$X'(L+1) = R \cos(R, 2\pi) \quad (D.9a)$$

$$Y'(L+1) = R \sin(R, 2\pi) \quad (D.9b)$$

$$Z'(L+1) = 0 \quad (D.9c)$$

As coordenadas acima podem ser transformadas para o sistema de coordenadas da amostra da forma

$$X = a_x' X'(L+1) + a_y' Y'(L+1) + X_0 \quad (D.10a)$$

$$Y = b_x' X'(L+1) + b_y' Y'(L+1) + Y_0 \quad (D.10b)$$

$$Z = c_x' X'(L+1) + c_y' Y'(L+1) + Z_0 \quad (D.10c)$$

E - Energias de ionização radiativa em semicondutores de "gap" direto

Analizamos neste tópico a produção de pares elétron-buraco como uma dependência do "gap" na banda de energia. Por produção entendemos o montante de pares elétron-buraco produzidos pelo feixe de elétrons primários. Em termos práticos nada se produziu de novo desde o artigo de Klein [1] em 1968. Os processos físicos ainda são mal explicados e têm-se apenas um idéia qualitativa destes. Vamos portanto nos basear completamente no artigo de Klein que é referência ainda usada como sendo a mais fundamental pelos autores que lidam com catodoluminescência.

No artigo de Klein é discutido processo de energia de ionização radiativa para radiações de altas energias em geral. É evidente que vamos nos ater apenas a elétrons de alta energia (que em nosso caso vai até 50 keV). Isto é justificado devido ao fato de que evidências experimentais apontadas por Klein mostram a independência da energia de criação de pares com a natureza da radiação primária.

Vamos definir a energia de ionização radiativa, ou energia de criação de pares, como sendo

$$Q = \frac{W}{e} \quad (E.1)$$

onde Q é a eficiência quântica de pares por elétron e W a energia do agente ionizador. Ficando assim estabelecida a proporcionalidade da eficiência quântica com a energia primária. A equação (1) dá portanto a medida da energia média gasta para produzir um par elétron-buraco através de quantidades experimentalmente acessíveis.

Em uma simples consideração teórica cada par criado é tratado como resultado de um processo elementar que não envolve realmente a física da interação da radiação com o meio. Evidentemente a energia para a criação de um par deve ser maior que o "gap" do meio, portanto podemos escrever

$$e = E_G + E' \quad (E.2)$$

onde dizemos que E' é a energia gasta para quaisquer outros propósitos que não a ionização. Especula-se que o processo ocorra em duas etapas: na primeira temos que a energia colocada dentro do sistema cria uma distribuição de não-equilíbrio dos portadores livres ao induzir transições da banda de valência à banda de condução. Esta é uma etapa que é concluída quase imediatamente, mesmo porque é provável que a ionização entre em competição com a geração de fónons. Na segunda etapa ocorre a termalização que é um processo bem mais lento. Temos portanto além das perdas pela ionização que já incluímos na equação (E.2) uma perda devida a geração de fónons (óticos) e devida a termalização que aparece devido ao fato de que a energia residual já não será mais suficiente para gerar outros pares. Fazendo esta pequena contabilidade pode-se chegar a expressão final obtida a partir de considerações empíricas

$$e = \frac{14}{5} E_G + r(h\nu_r) \quad (E.3)$$

onde r é o número médio de fónons óticos emitidos pela ionização e deve ser tratado neste caso como um parâmetro ajustável.

A figura E.1 mostra a dependência da energia de ionização radiativa com o "gap". Como pode ser visto esta dependência é praticamente linear.

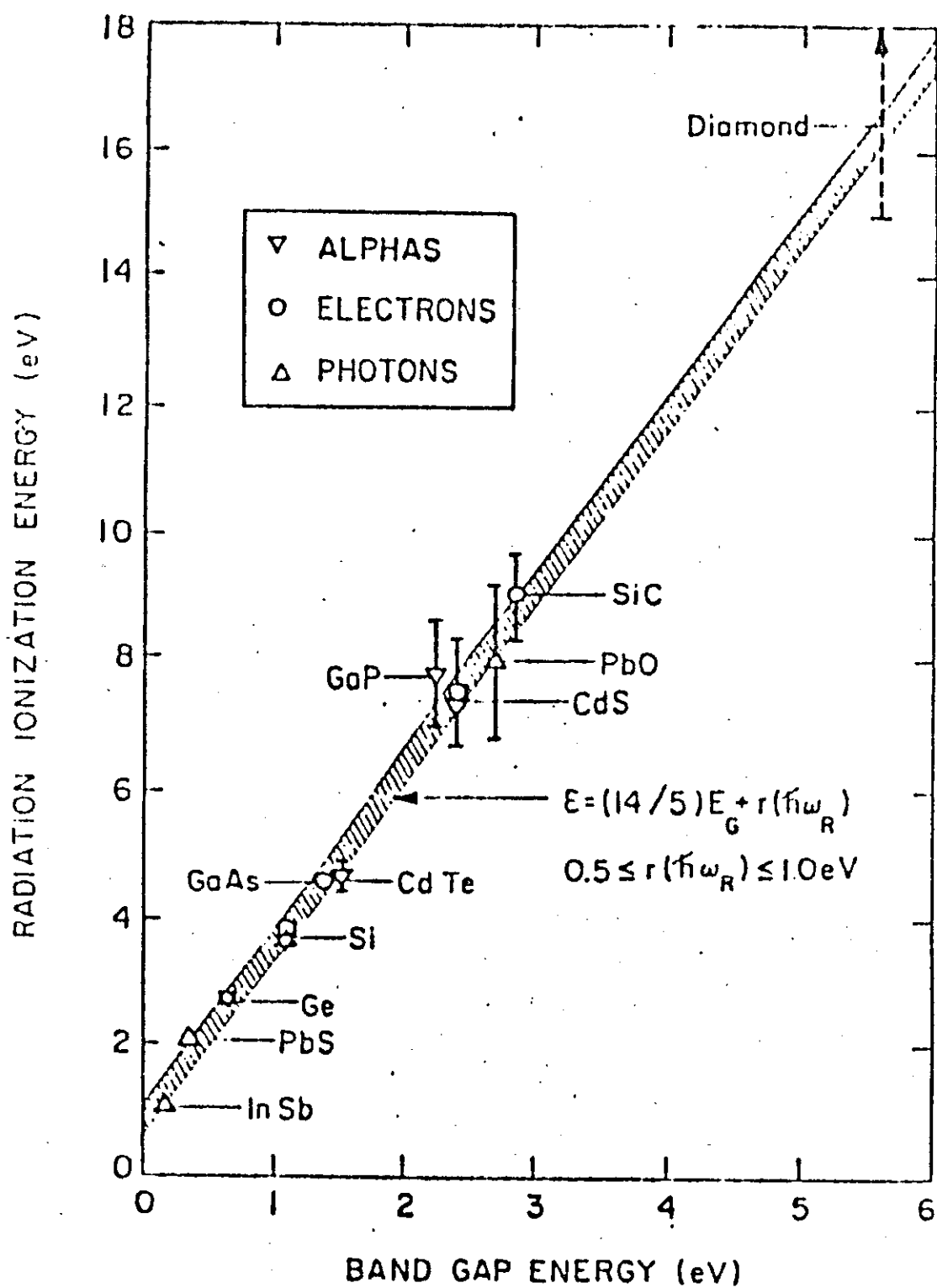


Figura E.1

Energia de ionização radiativa, ou energia média consumida por produção de par elétron-buraco com função do "gap".

É fácil mostrar que a relação (E.3) é razoável se seguirmos as simples considerações de Bogdankévitch et alii [2].

Usando as leis de conservação de energia e momento, vamos considerar o caso concreto da transição de um elétron da banda de valência à banda de condução resultante de uma colisão com outro elétron descrito pela reação



onde e_1 representa o elétron primário, e_v o elétron da banda de valência, e_1' o elétron difundido, e_c o elétron na banda de condução e p o buraco formado na banda de valência.

As leis de conservação de energia e momento podem ser escritas da forma seguinte

$$\begin{aligned} E_{e_1} &= E_{e_1'} + E_{e_c} + E_p + E_G \\ \vec{p}_{e_1} &= \vec{p}_{e_1'} + \vec{p}_{e_c} + \vec{p}_p \end{aligned} \quad (E.5)$$

onde E_G é o "gap".

Para determinar valor mínimo E_{e_1} para o qual a reação (E.4) ainda é possível, devemos calcular o extremo da função E_{e_1} levando em conta a equação (E.5). Pode-se demonstrar que E_{e_1} toma seu valor mínimo quando

$$\vec{v}_{e_1} = \vec{v}_{e_c} = \vec{v}_p = \vec{v} \quad (E.6)$$

colocando (E.6) em (E.5) obtemos

$$\frac{mv'^2}{2} = (2m_e + m_p) \frac{v^2}{2} + E_G \quad (E.7)$$

$$mv' = (2m_e + m_p) \frac{v}{2} + E_G \quad (E.8)$$

Resolvendo este sistema para v' e v obtém-se o limiar da energia de excitação

$$E_{et} = E_G \left(1 + \frac{m_e}{m_e + m_p} \right) \quad (E.9)$$

Se a partícula primária for um buraco o limiar da energia será

$$E_{pt} = E_G \left(1 + \frac{m_p}{m_e + m_p} \right) \quad (E.10)$$

Definimos o rendimento por

$$Q = \frac{E_G}{\langle E \rangle} \quad (E.11)$$

onde $\langle E \rangle$ é a energia média requerida para dar lugar a um par elétron-buraco. Seja esta energia média ainda

$$\langle E \rangle = E_G + \langle E_e \rangle + \langle E_p \rangle \quad (E.12)$$

onde

$$\langle E_e \rangle = \frac{\int_0^\infty E_e f_e(E) \rho_e(E) dE}{\int_0^\infty f_e(E) \rho_e(E) dE} \quad (E.13)$$

e

$$\langle E_p \rangle = \frac{\int_0^\infty E_p f_p(E) \rho_p(E) dE}{\int_0^\infty f_p(E) \rho_p(E) dE} \quad (E.14)$$

onde f_e e f_p são funções de repartição dos elétrons e dos buracos e ρ_e e ρ_p são as densidades de estado das bandas.

Como as partículas se concentram sobretudo na proximidade dos extremos das bandas podemos colocar ρ proporcional a E e dizemos que

$$\begin{aligned} f(E) &= \text{constante} & \text{para } 0 \leq E \leq E_t \\ f(E) &= 0 & \text{para } E > E_t \end{aligned} \quad (E.15)$$

Calculando desta forma as integrais (E.13) e (E.14) obtemos

$$\langle E_e \rangle = \frac{3}{2} E_{et} \quad \langle E_p \rangle = \frac{3}{5} E_{pt} \quad (E.16)$$

calculando assim a eficiência

$$Q = \frac{E_G}{E_G + \frac{3}{5}(E_{et} + E_{pt})} = \frac{5}{14} \quad (E.17)$$

ou seja

$$\langle E \rangle = \frac{5}{14} E_G \quad (E.18)$$

o que valida a nossa equação (E.3) obtida a partir de considerações empíricas.

A partir da curva de dissipação de energia (vide cap. 2), podemos obter o valor da energia disponível em um dado volume e calcularmos a produção de pares neste volume, diretamente pela expressão de eficiência quântica dada pela equação (E.11).

Assim sendo é possível atribuir a g_0 (vide capítulo 3, equação 2) o valor obtido para a expressão (E.11) como sendo um valor razoável do número de pares produzidos na região para qual se está aplicando a equação mencionada acima.
(eq. 2, cap. 3).

F - Organização dos Programas

A organização geral consiste das quatro etapas seguintes

- 1 - Preparação do arquivo de entrada
- 2 - Execução do programa de simulação
- 3 - Execução de programas suplementares
- 4 - Gráficos

1. Preparação do arquivo de entrada

Uma simulação Monte Carlo pode ter uma pesada carga de CPU e que em alguns casos pode passar de dezenas de horas. Obviamente um programa desta natureza não pode ser executado interativamente e os dados de entrada devem ser criteriosamente escolhidos pois um erro pode levar a um desperdício de horas de CPU e além disso, é claro, fornecer resultados inúteis.

Tendo estas considerações em mente foi desenvolvido o programa MONTA que constrói, de forma interativa, o arquivo de entrada para o programa de simulação.

A tarefa do programa MONTA é de criar o arquivo de dados DADOS.DAT no formato adequado à leitura do programa de simulação e de averiguar a coerência dos dados fornecidos pelo usuário.

O programa MONTA se estrutura da maneira seguinte:

- a) Lê o arquivo de DADOS.DAT da simulação anterior
- b) Interroga o usuário sobre modificações a serem feitas em

1. Denominação geral da simulação
2. Tarefa a serem executadas
3. Aproximações a serem usadas
4. Energias do feixe
5. Ângulos de inclinação da amostra
6. Parâmetros materiais da amostra (A,Z,) (vide capítulo 1)
7. Parâmetros estruturais da amostra (camadas, espessura)
8. Número de simulações, número de trajetórias, varredura (início, fim e total de pontos)

- c) Verifica se todos os dados são completos e coerentes. Se alguma falta ou incoerência for encontrada o programa aponta o erro e reinicia a sessão
- d) Cria o arquivo DADOS.DAT;+1 e fornece uma estimativa sobre o tempo de CPU necessário.

2. A execução do programa de simulação

A execução do programa MICROSSONDA_32 (Versão 3.2) em geral é feita não-interativamente e como critério para tal pode ser usado a estimativa de CPU em 1.d. A organização deste programa se dá da seguinte forma

- a) Lê o arquivo DADOS.DAT e interpreta as instruções nele contidas. Reserva espaço (abre arquivos) para os dados a serem gerados durante a simulação.
- b) É realizada a simulação propriamente dita pela subrotina ANDAR_32.

PAS

- c) Fecha os arquivos com os dados gerados e edita o relatório HISTORY.DAT;+1 contendo as características da simulação e alguns resultados.

Obs.: Ao executar o programa, é lido o arquivo HISTORY.DAT da simulação anterior. O primeiro número que aparece neste arquivo é o REGISTRO da simulação o qual é incrementado a cada simulação. Todos os arquivos e gráficos referentes a uma certa simulação contém este número.

3. Programas suplementares

Os programas suplementares se referem ao modelo de EBIC a ser aplicado usando dados gerados por MICROSSONDA_32 e contidos em PERFIL.DAT.

Para o caso de uma homojunção perpendicular ao feixe temos o programa EBIC_2 que, para comprimentos de difusão fixados, calcula a corrente total coletada na junção. O programa ainda varia a profundidade da junção de perto da superfície até uma distância previamente fixada. Este programa requer na ordem de alguns minutos de CPU.

Para o caso de um Laser semicondutor temos o programa EBIC_LASER que calcula a corrente EBIC para uma dada estrutura de Laser variando o comprimento de difusão fazendo um ajuste com dados experimentais correspondentes. Este programa requer na ordem de minutos de CPU.

4. Programas de desenho

DESENHAR_DEDS

Este programa desenha os perfis de dissipação para uma certa simulação, colocando no mesmo gráfico até dez curvas. Ajusta automaticamente a escala, funciona interativamente.

DESENHAR_GERAL

Tem a mesma função de DESENHAR_DEDS só que o usuário determina as escalas e as curvas a serem traçadas. É próprio para realçar detalhes nas curvas. Funciona interativamente.

DESENHAR_EBIC

Este programa traça as curvas de EBIC geradas pelo programa EBIC_2. Pode separar as contribuições das camadas em curvas diferentes ou somá-las. Funciona interativamente.

Para a execução destes programas usamos o VAX-11/780 do Centro de Computação Prof. J.D. Rogers. A linguagem usada é o VAX-11 PASCAL. Chamamos atenção ao fato de que este PASCAL pode recorrer a qualquer subrotina FORTRAN compatível com o VAX-11. Em alguns casos usamos TURBO-PASCAL, compatível com o microcomputador ITAUTEC 7000, principalmente para tarefas mais simples.

```

1  (*$S-X)
2  Program Monta(Input,Output,Dados);

5  Label 1,200,3;

7  Const Tra_Seg=0.117;

9  Type Material=( Retro,Trans,Absorvido,Cilindro,Mat_1,Mat_2,Mat_3,Mat_4,Mat_5,Mat_6,Mat_7,Mat_8,Mat_9,Mat_10,M);
10  Answer=( Atmico,Massa,Densidade,Peb,Nome,Apaga,At,Ma,De,No);
11  Slnao=(S,N);

13  Var Chave,Sse,Ssr,Sst,Ssd,Sss,Varredura,Alter,E_A,Random,
14  Desenh_Shape,Desenhar_En_Monitor,Aproximado,
15  Alcance_Sn,Alcance_Des,
16  Leitura,Reinicio,Diagnostico,Desenhar,Cilin,
17  Desenh_Ebic,Monolog,Reaproveltar,Escl,Term
18  Zt,Dens,Amass,Gap,E
19  Simular,A1,Tot_Sim,Pontos,R,A2,A3,R2,Trajectoria,
20  Caso,Hora,Minuto,Segundo,R1
21  Elnicial,Angulo,P1,Norma
22  Amostra
23  S_N
24  Resposta
25  Dados
26  Imat,Jmat,Hmat,Kmat
27  Inicio,Fim,Subs,Diametro,Tempo,Precisao_Shape,
28  Profundidade,Prof_Jun,Esp_Jun,Difusao_N,Difusao_P,
29  Eficiencia,Camada,Substrato
30  Definicao_1,Definicao_2
31  Superf

33  Procedure Vat(Sn:Boolean);
34  Begin
35  If Not Sn Then Write(' ');
36  End;
37  Procedure Tr;
38  Var I:Integer;
39  Begin
40  For I:=1 To 28 Do Write('xx');
41  Writeln;
42  End;
43  Begin
44  Write('terminal de video ? (s ou n):');Readln(S_N);
45  If S_N=S Then Term:=True Else Term:=False;
46  Write('precisa de esclarecimentos ? (s ou n):');Readln(S_N);
47  If S_N=S Then Escl:=True Else Escl:=False;
48  If Escl Then
49  Begin
50  If Term Then Tr;

```

```

51 WriteLn(' este e o programa monta criado por g.weber. ');
52 WriteLn(' o objetivo e montar o arquivo de entrada de dados para o programa de monte carlo. ');
53 WriteLn(' o presente programa fara mofificacoes em um arquivo anterior. ');
54 WriteLn(' e interrogara interativamente o usuario sobre as modifcacoes a ');
55 WriteLn(' serem realizadas. ');
56 WriteLn(' esclarecimentos sobre o significado das modifcacoes serao ');
57 WriteLn(' dados durante a execucao do programa. ');
58 WriteLn(' quando perguntar "pronto" s (sim) significa continuar nor- ');
59 WriteLn(' malmente e n (nao) violatara ao comeco do programa. ');
60 If Term Then Tr;
61 Write('pronto ? (s ou n): '); ReadLn(S_N);
62 If S_N=N Then Goto 3;
63 End;
64 (*Leitura do arquivo da simulacao anterior*)
65 3:Open(Dados, 'lweberldados2.dat', Old);
66 Reset(Dados);
67 ReadLn(Dados, Random, Sse, Ssr, Sst, Ssd, Sss);
68 ReadLn(Dados, Diagnostico, Monitor, Aproximado, Reaproveitar, Alcance_Sn);
69 ReadLn(Dados, Simular, Caso, Leitura);
70 A1:=0;
71 While Leitura Do
72 Begin
73 A1:=A1+1;
74 Read(Dados, Leitura, EInicial[A1], P1[A1]);
75 If Sss Then ReadLn(Dados, Normal[A1]) Else ReadLn(Dados);
76 End;
77 ReadLn(Dados, Leitura);
78 A2:=0;
79 While Leitura Do
80 Begin
81 A2:=A2+1;
82 ReadLn(Dados, Leitura, Angulo[A2])
83 End;
84 For Imat:=Mat_1 To Mat_10 Do
85 Begin
86 ReadLn(Dados, Leitura, Zt[Imat], Dens[Imat], Amass[Imat], Cap[Imat], Amostra[Imat]);
87 If Not Leitura Then Goto 1
88 End;
89 1: ReadLn(Dados, Tot_Sim); ReadLn(Dados, Varredura, Desenhar, Desenhar_En,
90 Desenhar_Shape, Desenhar_Ebic, Monolog, Alcance_Des);
91 If Varredura Then ReadLn(Dados, Inicio, Fim, Pontos);
92 ReadLn(Dados, Diametro, Cilin);
93 If Desenhar_Shape Then ReadLn(Dados, Precisao_Shape, Profundidade);
94 If Caso=2 Then
95 Begin
96 For Jmat:=Mat_1 To Imat Do ReadLn(Dados, Superf[Jmat]);
97 End;
98 ReadLn(Dados, Definicao_1);
99 ReadLn(Dados, Definicao_2);
100 Close(Dados);
101 200:Alter:=True;

```

```

102 (Xinquire uma definicao generica da amostra ou da simulacao)
103 While Alter Do
104   Begin
105     Writeln('definicao generica da amostra');
106     Writeln;
107     If Term Then Tr;
108     Writeln(Definicao_1);Writeln(Definicao_2);
109     If Term Then Tr;
110     Writeln;
111     Write('muda ? (s ou n):');Readln(S_N);
112     If S_N=S Then
113       Begin
114         Writeln('nova definicao');
115         Writeln('o campo');
116         Writeln('-----/');
117         Write('linha 1:');Readln(Definicao_1);
118         Write('linha 2:');Readln(Definicao_2);
119         End;
120       Else Alter:=False;
121     End;
122   Alter:=True;
123 (Xinquire tipos de atividades a serem exercidas na simulacao)
124 While Alter Do
125   Begin
126     If Term Then Tr;
127     Writeln('sim nao');
128     Val(Sse):Writeln('1:ebic');
129     Val(Ssr):Writeln('2:retroespalhados');
130     Val(Sst):Writeln('3:transmitidos');
131     Val(Ssd):Writeln('4:distribuido');
132     Val(Cilin):Writeln('5:cilindro');
133     Val(Sss):Writeln('6:perfil de energia');
134     Val(Alcance_Sn):Writeln('7:alcance');
135     If Term Then Tr;
136     Writeln('O:continuar');
137     Write('opcao:');Readln(R);
138     Case R Of
139       1:Sse:=Not Sse;
140       2:Ssr:=Not Ssr;
141       3:Sst:=Not Sst;
142       4:Ssd:=Not Ssd;
143       5:Cilin:=Not Cilin;
144       7:Alcance_Sn:=Not Alcance_Sn;
145       6:Sss:=Not Sss;
146       0:Alter:=False;
147     End;
148   Alter:=True;
149 (Xinquire alguns tipos de atividades e caracteristicas da simulacao)
150 While Alter Do
151   Begin

```

```

153 If Term Then Tr:
154   Writeln('          sim          nao');
155   Val(Random):Writeln('1:random');
156   Val(Varredura):Writeln('2:varredura');
157   Val(Monitor):Writeln('3:monitor');
158   Val(Aproximado):Writeln('4:aproximado');
159   Val(Reaproveltar):Writeln('5:reaproveltar');
160   If Term Then Tr:
161     Writeln('O:continuar');
162     Write('opcao:');Readln(R);
163     Case R Of
164       1:Random:=Not Random;
165       2:Varredura:=Not Varredura;
166       3:Monitor:=Not Monitor;
167       4:Aproximado:=Not Aproximado;
168       5:Reaproveltar:=Not Reaproveltar;
169       0:Alter:=False;
170     End;
171   End;
172   Alter:=True;
173   (Xinquire tipos de desenhos a serem realizados.NOTA:a secao de desenhos foi tirada do programa de simulacao)
174   While Alter Do
175     Begin
176       Writeln('          d e s e n h o s');
177       If Term Then Tr:
178         Writeln('          sim          nao');
179       If Sse Then
180         Begin
181           Val(Desenhar_Eblic):Writeln('1:eblic');
182           Val(Monolog):Writeln('2:eblic em monolog');
183         End
184       Else
185         Begin
186           Desenhar_Eblic:=False;
187           Monolog:=False;
188         End;
189       If Ssr Then
190         Begin
191           Val(Desenhar):Writeln('3:retro angular');
192           Val(Desenhar_En):Writeln('4:retro angular energia');
193         End
194       Else
195         Begin
196           Desenhar:=False;
197           Desenhar_En:=False;
198         End;
199       If Alcance_Sn Then
200         Begin
201           Val(Alcance_Des):Writeln('5:alcance');
202         End
203       Else Alcance_Des:=False;

```

```

204 If Sss Then
205   Begin
206     Val(Desenhar_Shape);WriteLn('6:perfil de energia');
207   End
208 Else Desenhar_Shape:=False;
209 If Term Then Tr;
210 WriteLn('O:continuar');
211 Write('opcao:');ReadLn(R);
212 Case R Of
213   1:Desenhar_Ebic:=Not Desenhar_Ebic;
214   2:Monolog:=Not Monolog;
215   3:Desenhar:=Not Desenhar;
216   4:Desenhar_En:=Not Desenhar_En;
217   5:Alicance_Des:=Not Alicance_Des;
218   6:Desenhar_Shape:=Not Desenhar_Shape;
219   0:Alter:=False;
220 End;
221 End;
222 Alter:=True;
223 (*Inquire energias de simulacao, no caso de calcular perfis de energia tambem inquire os alcances
224 sob o nome de NORMA para otimizar o espaco na hora da execucao. Obs.:O numero de simualcoes e' dado por
225 ENERGIAS X ANGULOS *)
226 While Alter Do
227   Begin
228     If Term Then Tr;
229     For A3:=1 To A1 Do
230       Begin
231         Write('eInicial['',A3:2,'',J'',EInicial[A3]:4:1,'', pi= ',Pi[A3]:6:2);
232         If Sss Then WriteLn('', norma= ',Norma[A3]:6:2) Else WriteLn;
233       End;
234     If Term Then Tr;
235     WriteLn('', nao mudar=0 , caso contrario introduza o indice desejado');
236     Write('opcao:');ReadLn(R);
237     If R > 0 Then
238       Begin
239         Write('.....energia:');ReadLn(EInicial[R]);
240         Write('parametro de impacto:');ReadLn(Pi[R]);
241         If Sss Then
242           Begin
243             Write('.....norma:');ReadLn(Norma[A3]);
244           End;
245         If R > A1 Then A1:=R;
246         If EInicial[R]=0.0 Then A1:=R-1
247       End
248     Else
249       If R < 0 Then
250         Begin
251           Write('e=',EInicial[-R]:4:1,'', pi=',Pi[-R]:6:2,'', norma = ');
252           ReadLn(Norma[-R]);
253         End
254       Else Alter:=False;

```

```

255 End;
256 Alter:=Not Alter;
257 (xidem do segmento anterior para angulosx)
258 While Alter Do
259 Begin
260   If Term Then Tr;
261   For A3:=1 To A2 Do Writeln('angulo[' ,A3:2, ']:',Angulo[A3]:5:1);
262   If Term Then Tr;
263   Writeln(' nao mudar=0 , caso contrario introduza o indice desejado');
264   Write(' .....opcao :');Readln(R);
265   If R > 0 Then
266   Begin
267     Write(' .....angulo :');Readln(Angulo[R]);
268     If R > A2 Then A2:=R;
269     If Angulo[R] < 0.0 Then A2:=R-1
270   End
271   Else Alter:=Not Alter;
272 End;
273 Alter:=Not Alter;
274 (x inquire Nome, Numero Atomico, Massa Atomica e Densidade de cada camadax)
275 Kmat:=Mat_1;
276 While Alter Do
277 Begin
278   If Term Then Tr;
279   For Jmat:=Mat_1 To Imat Do
280     Writeln(Jmat:7,Amostra[Jmat], ' z: ',Zt[Jmat]:5:1, ' a: ',Amass[Jmat]:6:2, ' d: ',Dens[Jmat]:6:3, ' gap: ',Gap[Jmat]:6:2);
281   If Term Then Tr;
282   Write(' .....muda ? ');Readln(S_N);
283   If S_N=S Then
284   Begin
285     Writeln('atomico,maxssa,deksndidade,peb,noxme ou apaga?');Readln(Resposta);
286     If Resposta <> Apaga Then
287     Begin
288       Writeln(' .....atual :',Kmat:7);
289       Writeln('para permanecer digite m');
290       Write(' .....qual ? ');Readln(Hmat);
291       If Hmat <> M Then Jmat:=Hmat Else Jmat:=Kmat;
292       Kmat:=Jmat;
293       If (Resposta <> Nome) And (Resposta <> No) Then
294       Begin
295         Writeln('novo valor:');Readln(Subs);
296       End
297     End;
298     Case Resposta Of
299       Atomico,At:Zt[Jmat]:=Subs;
300       Massa,Ma:Amass[Jmat]:=Subs;
301       Densidade,De:Dens[Jmat]:=Subs;
302       Peb:Gap[Jmat]:=Subs;
303       Nome,No:Begin
304         Write('novo nome:');Readln(Amostra[Jmat]);
305         Write('z=');Readln(Zt[Jmat]);

```

```

306 Write('a:');Readln(Amass[Jmat]);
307 Write('dens=');Readln(Dens[Jmat]);
308 End;
309 ABegin
310 Write('ultima linha:');Readln(Imat);
311 Jmat:=Imat;
312 End
313 End;
314 If Jmat > Imat Then Imat:=Jmat;
315 End
316 Else Alter:=Not Alter;
317 End;
318 Alter:=True;
319 (*Inquire sobre as geometrias de simulacao*)
320 While Alter Do
321 Begin
322 Writeln(' g e o m e t r i a s');
323 If Term Then Tr;
324 Writeln(Writeln(' 1: 5 camadas horizontais');
325 Writeln(' 2: homogeneo e inclinavel');
326 Writeln(' 3: duas camadas verticais');
327 If Term Then Tr;
328 Caso:=Caso-1;
329 Writeln(' 0: permanece ou confirma opcao: ',Caso:2);
330 Caso:=Caso+1;
331 Writeln('opcao:');Readln(R);
332 If R=0 Then Alter:=False Else Caso:=R+1;
333 End;
334 Alter:=Not Alter;
335 While Alter Do
336 Begin
337 Case Caso Of
338 2:Begin
339 If Term Then Tr;
340 For Jmat:=Mat_1 To Mat_10 Do
341 Writeln('Material ',(Ord(Jmat)-3):2,Superf(Jmat):8:3);
342 If Term Then Tr;
343 Writeln('0:.....mudar nada.....');
344 Write('opcao:');Readln(R);
345 End;
346 4:Begin
347 If Term Then Tr;
348 Writeln('1:comp. de dif. na primeira camada: ',Difusao_P:8:3);
349 Writeln('2:comp. de dif. na segunda camada: ',Difusao_N:8:3);
350 Writeln('3:eficiencia de producao de pares: ',Eficiencia:10:4);
351 If Term Then Tr;
352 Writeln('0:.....mudar nada.....');
353 Write('opcao:');Readln(R);
354 If R <> 0 Then R:=R+20;
355 End;
356 3:Alter:=False;

```



```

357 End;
358 If R=0 Then Alter:=False
359 Else
360   Begin
361     Jmat:=Cilindro;
362     For Ri:=1 To R Do Jmat:=Succ(Jmat);
363     Write('Material ',R:2,','); Readln(Superf[Jmat]);
364   End;
365 End;
366 Alter:=True;
367 (Xinquire sobre consideracoes gerais de simulacao)
368 While Alter Do
369   Begin
370     If Term Then Tr;
371     WriteLn(' 1:.....varredura:');
372     WriteLn(' 2:total de simulacoes:',Tot_Sim:5);
373     WriteLn(' 3:.....simular:',Simular);
374     WriteLn(' 4:.....dimetro:',Diametro);
375     WriteLn(' 5:.....precisao shape:',Precisao_Shape);
376     WriteLn(' 6:.....profundidade:',Profundidade);
377     If Term Then Tr;
378     WriteLn(' 0:.....muda nada.....');
379     WriteLn('opcao:'); Readln(R);
380     Case R Of
381       1:Begin
382         If Varredura Then
383           Begin
384             While Alter Do
385               Begin
386                 WriteLn('1:inicio',Inicio:6:2);
387                 WriteLn('2:fil',Fil:6:2);
388                 WriteLn('3:pontos',Pontos:6);
389                 WriteLn('4:nao mudar');
390                 Write('opcao'): Readln(R2);
391                 Case R2 Of
392                   1:Begin Write('inicio:'); Readln(Inicio);End;
393                   2:Begin Write('fil:'); Readln(Fil);End;
394                   3:Begin Write('pontos:'); Readln(Pontos);End;
395                   4:Alter:=False;
396                 End;
397               End;
398             Alter:=True;
399           End;
400         End;
401       2:Begin
402         Write('novo valor do total de simulacoes:'); Readln(Tot_Sim);
403       End;
404       3:Begin
405         Write('novo valor de simular:');
406         Readln(Simular)
407       End;

```

```

408 4:Begin
409   Write('novo valor do diametro:');Readln(Diametro);
410 End;
411 5:Begin
412   Write('nova precisao de shape:');Readln(Precisao_Shape);
413 End;
414 6:Begin
415   Write('novo profundidade:');Readln(Profundidade);
416 End;
417   O:Alter:=Not Alter;
418 End
419 End;
420 Reinicio:=False;
421 (x0 segmento que segue analisa os dados fornecidos e as modificacoes feitas e as rejelta no caso de
422 inconsistencia ou omissao. Dependendo do caso simplesmente reinicia a sessao outra vez desde o comecox)
423 If Sss Then
424   Begin
425     For A3:=1 To A1 Do
426       Begin
427         If Norma[A3]=0.0 Then
428           Begin
429             Writeln('xxx erro xxx');
430             Writeln('norma nula');
431             Writeln('xxx reinicio !!!');
432             Reinicio:=True;
433           End;
434         End;
435       End;
436     Trajetoria:=Tot_Sim*Simular;
437     If Varredura Then Trajetoria:=Trajetoria*Pontos;
438     If Trajetoria=0 Then
439       Begin
440         Writeln('xxx erro xxx');Writeln('zero trajetorias');
441         Writeln('xxx reinicio ');Reinicio:=Not Reinicio
442       End;
443     If A1=0 Then
444       Begin
445         Writeln('xxx erro xxx');Writeln('zero energias');
446         Writeln('xxx reinicio');
447         If Not Reinicio Then Reinicio:=Not Reinicio;
448       End;
449     Trajetoria:=Trajetoria*A1;
450     If A2=0 Then
451       Begin
452         Writeln('xxx alertaxxx');Writeln('nenhum angulo especificado');
453         Writeln('reinicio ?(s ou n):');
454         Readln(S_N);
455         If S_N=S Then
456           Begin
457             Writeln('xxx reinicio');
458             Reinicio:=True

```

```

459 End;
460 End;
461 Trajetoria:=TrajetoriaxA2;
462 If Varredura Then
463   Begin
464     If Fil=Inicio Then
465       Begin
466         Writeln('xxx erro xxx');Writeln('na varredura');
467         Writeln('inicio=fim');Writeln('xxx reinicio');
468         Reinicio:=True
469       End;
470     If Pontos=0 Then
471       Begin
472         Writeln('xxx erro xxx');Writeln('na varredura');
473       End;
474     Writeln('zero pontos');Writeln('xxx reinicio');
475     Reinicio:=True;
476   End;
477 End;
478 If Cilin And (Diametro=0.0) Then
479   Begin
480     Writeln('xxx alerta xxx');
481     Writeln('diametro e nulo');
482     Writeln('reiniciar?');
483     Readln(S_N);
484     If S_N=S Then
485       Begin
486         Writeln('xxx reinicio');
487         Reinicio:=True;
488       End;
489   End;
490   If Caso=2 Then
491     Begin
492       For Jmat:=lmat Downto Mat_2 Do
493         Begin
494           If Superf[Jmat] < Superf[Pred(Jmat)] Then Reinicio:=True;
495         End;
496       End;
497     If Reinicio Then Coto 200;
498     Writeln(Trajetoria,' trajetorias a serem simuladas');
499     Tempo:=1.0;
500     (*Estimativa do tempo de processamento previsto. NOTA: Esta secao nao se encontra atualizadax)
501     If Random Then Tempo:=1.0xTempo;
502     If Sse Then Tempo:=1.0xTempo;
503     If Ssr Then Tempo:=1.0xTempo;
504     If Sst Then Tempo:=1.0xTempo;
505     If Ssd Then Tempo:=1.0xTempo;
506     If Diagnostico Then Tempo:=1.0xTempo;
507     If Desenhx Then Tempo:=1.0xTempo;
508     Tempo:=Tempo*AlxA2*Tra_Seg*Simular;
509     If Varredura Then Tempo:=Pontos*Tempo;

```

```

510 (*Edicao do novo arquivo de dados*)
511 Open(Dados,'[weber\dados2.dat',New);
512 Rewrite(Dados);
513 Writeln(Dados,Random,Sse,Ser,Sst,Ssd,Sss);
514 Writeln(Dados,Diagnostico,Monitor,Aproximado,Reaproveltar,Alicance_Sn);
515 Leitura:=A1 > 0;
516 Writeln(Dados,Simular,Caso,Leitura);
517 For A3:=1 To A1 Do
518   Begin
519     Leitura:=A3 <> A1;
520     Write(Dados,Leitura,Elnicial[A3]:10:2,Pi[A3]:10:2);
521     If Sss Then Writeln(Dados,Normal[A3]) Else Writeln(Dados);
522   End;
523 Leitura:=A2 > 0;
524 Writeln(Dados,Leitura);
525 If Leitura Then
526   Begin
527     For A3:=1 To A2 Do
528       Begin
529         Leitura:=A3 <> A2;
530         Writeln(Dados,Leitura,Angulo[A3]);
531       End;
532   End;
533 For Jmat:=Mat_1 To Imat Do
534   Begin
535     Leitura:=Jmat <> Imat;
536     Writeln(Dados,Leitura,Zt[Jmat]:10:2,Dens[Jmat]:10:3,
537       Amas[Jmat]:10:3,Capt[Jmat]:10:3,' ',Amostr[Jmat]);
538   End;
539 Writeln(Dados,Tot_Sim);
540 Writeln(Dados,Varredura,Desenhar,Desenhar_En,Desenhar_Shape,Desenhar_Eblc,Monolog,Alicance_Des);
541 If Varredura Then Writeln(Dados,Inicio,Fim,Pontos);
542 Writeln(Dados,Diametro,Cilind);
543 If Desenhar_Shape Then Writeln(Dados,Precisao_Shape,Profundidade);
544 If Caso=2 Then
545   Begin
546     For Jmat:=Mat_1 To Imat Do Writeln(Dados,Superf[Jmat]);
547   End;
548 Writeln(Dados,Definicao_1);
549 Writeln(Dados,Definicao_2);
550 Close(Dados);
551 If Monitor Then
552   Begin
553     For A3:=1 To 60 Do Write('x');Writeln;
554     Writeln('monitor aconchado');
555     For A3:=1 To 30 Do Write('xx');Writeln;
556   End;
557 If Reaproveltar Then
558   Begin
559     For A3:=1 To 15 Do Write('==');Writeln;
560     Writeln('reaprovelta mento');

```

```
561   For A3:=1 To 15 Do Write('==');WriteLn;
562   End
563   Else
564     Begin
565       Hora:=Round(Tempo) Div 3600;
566       Tempo:=Tempo-Hora*3600;
567       Minuto:=Round(Tempo) Div 60;
568       Tempo:=Tempo-Minuto*60;
569       Segundo:=Round(Tempo-Minuto);
570       Write('Tempo de cpu ');
571       If Hora > 0 Then Write(Hora:3,' horas ');
572       If Minuto > 0 Then Write(Minuto:2,' minutos ');
573       If Segundo > 0 Then Write('e ',Segundo:2,' segundos');
574       WriteLn(' estimado');
575     End;
576   End.
```

```

1  (*$S-*)
2  Program Monte_Carlo(Output,Ebic,Retro,Dados,Historia,Dissipar);
3  Label 1;

5  (*Definicao dos tipos de variaveis *)
6  Type
7  Decisao
8  =Record
9  Ebic,Retro,Trans,Dist,Cilin,Shape :Boolean;
10 End;
11 Especial
12 =Real;
13 Material
14 =(Absorvido,Retros,Mat_1,Mat_2,Mat_3,Mat_4,Mat_5,Mat_6,Mat_7,Mat_8,Mat_9,Mat_10);
15 Umdez
16 =Array[1..10] Of Real;
17 Difmat
18 =Array[Mat_1..Mat_10] Of Real;
19 Difmat2
20 =Array[Retros..Mat_10] Of Real;
21 F103
22 =Record
23 E,P:Real;
24 End;
25 Zerocem
26 =Array[1..100] Of Real;
27 Estatistica
28 =Record
29 Simnao :Decisao;
30 Perfil :Record
31 Z :Array[0..1000] Of Real;
32 Alcance,Profundidade :Real;
33 End;
34 Retro_Angulo,Trans_Angulo,Retro_Energia,Trans_Energia :Record
35 A,B :Array[-100..100] Of Real
36 End;
37 Energia,Angulo,P1 :Umdez;
38 Retro_Coef,Trans_Coef,Sa,Sb,Retro_Coef_En :Array[1..10,1..10] Of Real;
39 A1,A2 :Integer;
40 End;
41 F101
42 =Record
43 Medio,Max:Real;
44 End;
45 F102
46 =Array[1..10] Of F101;
47 F104
48 =Packed Array[1..40] Of Char;
49 Umtrezentos
50 =Array[1..1000] Of Real;
51 Zerotrezentos=Array[0..1000] Of Real;
52 F100
53 =Array[1..10,1..50] Of Real;
54 Porta
55 =Text;
56 Nomes
57 =Array[Material] Of Packed Array[1..15] Of Char;

58 (*Declaracao das variaveis usadas no programa *)
59 Var
60 Saída
61 Zt,Dens,Pmi,Cof,Cof2,Cof3,Cof4,Cof5,Amass,Gap
62 Superf
63 Lcm,Epre
64 Random,Leitura,Varredura,Caminho,Desenhar,
65 Alcance_Sn,Alcance_Des,Monitor,Aproximado,
66 Estatistica;
67 Difmat;
68 Difmat2;
69 Zerotrezentos;

```

```

51 Desenharm_Eblic, Monolog, Reaproveltar,
52 Parada, Diagnostico, Desenharm_Shape, Desenharm_En
53 Imat, Jmat
54 I1, I2, Stop, Simular, A1, A2, Tot_Sim, Pontos,
55 Total_Stop, Sim_1, Sim_2, Sim_3, Sim_4, Kout,
56 A3, A4, A5, Registro, Caso, Contar, N_Alcance
57 Inicio, Fim, X1, Y1, Z1, Sds, Plim, Del_Var, Diametro,
58 Precisaao_Shape, Eficiencia, Fr, Dpasso, Passo
59 Dados, Eblic, Retro, Historia,
60 Dissipar
61 Amostra
62 Dia, Hora
63 Corrente_Eblic
64 Alcance
65 Definicao_1, Definicao_2
66 Norma

68 (*Funcao de gracao de numeros aleatorios. E usada uma funcao intrinseca do fortran que se localiza em RAN.FOR*)
69 Procedure Alea(Var I1, I2: Integer; Var R: Real); Fortran;

71 (*Funcoes aritmeticas nao suportadas intrinsecamente por VAX-11 PASCAL*)
72 Function MthsAlog10(Digit: Real): Real; Extern;
73 Function MthsAcos(Angle: Real): Real; Extern;

75 (*Inclusao da subrotina contida em ANDAR_32.PAS*)
76 *Include 'andar_32.pas/11st';

78 (*Funcao que calcula o potencial medio de ionizacao em funcao do numero atomico*)
79 Function Potencial(Z: Real): Real;
80 Begin
81 Potencial := Z*(14.0*(1.0-Exp(-0.1*XZ))+75.5/Z**((Z/7.5)-Z/(100.0+Z)))/1000.0
82 End;

84 Begin
85 (*Leitura do arquivo DADOS2.DAT que contem os parametros da simulacao*)
86 Open(Dados, 'weber\dados2.dat', Old);
87 Reset(Dados);
88 Read(Dados, Random, Salda, Simnao, Eblic, Salda, Simnao, Retro, Salda, Simnao, Trans, Salda, Simnao, Dist, Salda, Simnao, Shape);
89 Readln(Dados, Diagnostico, Monitor, Aproximado, Reaproveltar, Alcance_Sn);
90 Readln(Dados, Simular, Caso, Leitura);
91 A1 := 0;
92 While Leitura Do
93 Begin
94 A1 := A1 + 1;
95 Read(Dados, Leitura, Salda, Energia[A1], Salda, P[A1]);
96 If Salda.Simnao.Shape Then
97 Begin
98 Readln(Dados, Norma[A1]);
99 Norma[A1] := 1000/Norma[A1];
100 End;
101 End;

```

```

102 Readln(Dados,Leitura);
103 A2:=0;
104 While Leitura Do
105   Begin
106     A2:=A2+1;
107   Readln(Dados,Leitura,Salda,Angulo[A2])
108   End;
109 For Imat:=Mat_1 To Mat_10 Do
110   Begin
111     Readln(Dados,Leitura,Zt[Imat],Dens[Imat],Amass[Imat],Cap[Imat],Amostra[Imat]);
112     If Not Leitura Then Goto 1
113   End;
114   1:For Jmat:=Mat_1 To Imat Do
115     Begin
116       Pm[Jmat]:=Potencial(Zt[Jmat]);
117       Cof[Jmat]:=7.85*Zt[Jmat]/Amass[Jmat]*Dens[Jmat];
118       Cof2[Jmat]:=1.666/Pm[Jmat];
119       Cof3[Jmat]:=51.0*Amass[Jmat]/(Sqr(Zt[Jmat])*Dens[Jmat]*100.0);
120       Cof4[Jmat]:=Ln(294.0*Zt[Jmat]**(-0.6666))-1.0;
121       Cof5[Jmat]:=1.03*Pm[Jmat];
122     End;
123   Readln(Dados,Tot_Sim);
124   Readln(Dados,Varredura,Desenhar,Desenhar_Eq,Desenhar_Shape,Desenhar_Ebic,Monolog,Alicance_Des);
125   If Varredura Then
126     Begin
127       Readln(Dados,Inicio,Fim,Pontos);
128       Del_Var:=(Abs(Fim)+Abs(Inicio))/Pontos;
129     End;
130   Caminho:=(Imat > Mat_1) Or (Salda.Simnso.Shape);
131   Readln(Dados,Diametro,Salda.Simnso.Cilind);
132   If Desenhar_Shape Then Readln(Dados,Precisao_Shape,Salda.Perfil.Profundidade);
133   If Caso=2 Then
134     Begin
135       For Jmat:=Mat_1 To Imat Do
136         Begin
137           Readln(Dados,Superf[Jmat]);
138         End;
139     End;
140   Superf[Retros]:=0.0;
141   Readln(Dados,Definicao_1);
142   Readln(Dados,Definicao_2);
143   Close(Dados);
144   (*Fim da leitura de DADOS2.DAT *)
145   (*Leitura do numero de registro da simulacao anterior *)
146   Open(Historia,'[weber,history.dat',Old):Reset(Historia);
147   Readln(Historia,Registro):Close(Historia);
148   Registro:=Registro+1;
149   Contar:=0;
150
151   (*Inicio da simulacao *)

```



```

153 For Sim_1:=1 To Tot_Sim Do
154   Begin
155     If Saída.Simnáo.Shape Then
156       Begin
157         (xAbertura do arquivo PERFIL.DAT que contera' a energia dissipada em funcao da profundidade x)
158         Open(Dissipar, '[weber]perfil.dat', New);
159         Rewrite(Dissipar);
160         Writeln(Dissipar, Registro);
161         Writeln(Dissipar, Definicao_1);
162         Writeln(Dissipar, Definicao_2);
163         Writeln(Dissipar, Superf[Mat_1], Superf[Mat_2], Superf[Mat_3], Superf[Mat_4],
164           Superf[Mat_5]);
165         Writeln(Dissipar, Superf[Mat_6], Superf[Mat_7], Superf[Mat_8], Superf[Mat_9],
166           Superf[Mat_10]);
167       End;
168     If Saída.Simnáo.Retro Then
169       Begin
170         Open(Retro, '[weber]retro.dat', New);
171         Rewrite(Retro)
172       End;
173     For Sim_2:=1 To A1 Do
174       Begin
175         For Sim_3:=1 To A2 Do
176           Begin
177             If Pontos=0 Then Pontos:=1;
178             For Sim_4:=1 To Pontos Do
179               Begin
180                 If Varredura Then X1:=Del_Var*Sim_4+Inicio
181                 Else X1:=0.0;
182                 Y1:=0.0; Z1:=0.0;
183                 Saída.A1:=Sim_2;
184                 Saída.A2:=Sim_3;
185
186                 (xA simulacao propriamente dita e feita pela subrotina ANDAR contida em ANDAR_32.PAS*)
187                 Andar(Zt,Dens,Pml,Cof,Cof2,Cof3,Cof4,Cof5,Amass,Alcance[Sim_2],N_Alcance,11,12,Simular,Caso,X1,Y1,Z1,Diametro,
188                   Random,Caminho,Diagnostico,Monitor,Aproximado,Saída,Norma,Superf);
189
190                 (xAvallacao dos coeficientes de retroespalhamento x)
191                 Saída.Retro_Coeff_En[Sim_2,Sim_3]:=Saída.Retro_Coeff_En[Sim_2,Sim_3]/(Simular*Saída.Energia[Sim_2]);
192                 Alcance[Sim_2].Max:=Sqrt(Abs(Alcance[Sim_2].Max/(Alcance[Sim_2].Medio)/(Simular*(Simular-1)))));
193                 Alcance[Sim_2].Medio:=Alcance[Sim_2].Medio/Simular;
194                 Alcance[Sim_2].Max:=Alcance[Sim_2].Max+Alcance[Sim_2].Medio;
195                 Saída.Retro_Coeff[Sim_2,Sim_3]:=Saída.Retro_Coeff[Sim_2,Sim_3]/Simular;
196                 Saída.Trans_Coeff[Sim_2,Sim_3]:=Saída.Trans_Coeff[Sim_2,Sim_3]/Simular;
197                 If Saída.Simnáo.Shape Then
198                   Begin
199                     (xTranscricao do perfil de energia dissipada obtido x)
200                     Writeln(Dissipar, Saída.Energia[Sim_2], Saída.Angulo[Sim_3], Norma[Sim_2]);
201                     For A5:=0 To 1000 Do
202                       Begin
203                         Saída.Perfil.Z[A5]:=Saída.Perfil.Z[A5]*1000/Simular;

```

```

204 Write(Dissipar,Salda.Perfil,Z[A5]);
205 If (A5 Div 5)X5=A5 Then WriteIn(Dissipar);
206 Salda.Perfil.Z[A5]:=0.0;
207 End;
208 End;
209 End;
210 End;
211 End;
212 If Salda.Simnao.Shape Then Close(Dissipar);
213 If Salda.Simnao.Retro Then Close(Retro);
214 End;
215 (*Fim da simulacao *)

217 (*Elaboracao do relatorio da simulacao no arquivo HISTORY.DAT *)
218 Open(Historia,'Iweber\history.dat',New):Rewrite(Historia);
219 WriteIn(Historia,Registro);
220 Date(Dia):Time(Hora);
221 WriteIn(Historia,'x m l c r o s s o n d a - versao 3.2 - CCVAX2 *');
222 WriteIn(Historia,'simulacao realizada em ',Dia);
223 WriteIn(Historia,'as ',Hora);
224 WriteIn(Historia,Definicao_1);
225 WriteIn(Historia,Definicao_2);
226 For Jmat:=Mat_1 To lmat Do
227   Begin
228     Write(Historia,'material['',Jmat:6,''],AnostrafJmat]);
229     WriteIn(Historia,'z= ',Zt[Jmat:5:1],' dens= ',Dens[Jmat:5:1];
230     WriteIn(Historia,' pml= ',Pml[Jmat], ' cof= ',Cof[Jmat], ' cof2= ',Cof2[Jmat]);
231   End;
232 For A3:=1 To A1 Do
233   Begin
234     Write(Historia,'energia['',A3:2,''],Salda.Energia[A3]:6:2,' pl: ',Salda.PI[A3]:6:2);
235     If Salda.Simnao.Shape Then WriteIn(Historia,' norma= ',Norma[A3]:6:2) Else WriteIn(Historia);
236   End;
237 For A3:=1 To A2 Do WriteIn(Historia,'angulo['',A3:2,''],Salda.Angulo[A3]:6:2);
238 For A3:=1 To A1 Do
239   Begin
240     For A4:=1 To A2 Do
241       Begin
242         WriteIn(Historia,'e= ',Salda.Energia[A3]:6:2,' pl= ',Salda.PI[A3]:6:2,' a= ',Salda.Angulo[A4]:6:2,' r= ',Salda.Retro_Coef
243         A3,A4]:6:4,' re= ',Salda.Retro_Coef_En[A3,A4]:6:4,' t= ',Salda.Trans_Coef[A3,A4]:6:4);
244         If Salda.Simnao.Retro Then WriteIn(Historia,' sa= ',Salda.Sa[A3,A4], ' sb= ',Salda.Sb[A3,A4]);
245       End;
246     End;
247   WriteIn(Historia,'trajetorias simuladas ',Simular);
248   If Diametro > 0.0 Then WriteIn(Historia,' diametro do cilindro ',Diametro);
249   If Varredura Then
250     Begin
251       WriteIn(Historia,'varredura realizada de ',Pontos:6,' pontos');
252       WriteIn(Historia,'inicio: ',Inicio:6:2,' fim: ',Fim:6:2);
253     End;
254   Write(Historia,' geometria ');

```

```
255 Case Caso Of
256   2:Writeln(Historia,'5 camadas horizontais');
257   3:Writeln(Historia,'amostra homogenea inclinavel');
258   4:Writeln(Historia,'2 camadas verticais');
259 End;
260 Writeln(Historia,'Camadas');
261 For Jmat:=Mat_1 To lmat Do
262   Writeln(Historia,'material',Ord(Jmat):2,Superf(Jmat));
263   If Saida.Simnao.Retro Then Writeln(Historia,'retro realizado');
264   If Saida.Simnao.Shape Then Writeln(Historia,'perfil de energia em z da amostra realizado');
265   If Aproximado Then Writeln(Historia,'executado com aproximacao no livre caminho medio');
266   Writeln(Historia,' x m i c r o s s o n d a - versao 3.2- CCVAX2 *');
267   Close(Historia);
268 End.
```

```

1  Procedure Andar(      Zt,Dens,Pm1,Cof,Cof2,Cof3,
2      Cof4,Cof5,Amass
3      Var Alcance
4      Var N_Alcance
5      Var I1,I2,Simular,Caso
6      Xi,Y1,Z1,Diametro
7      Random,Caminho,Diagnostico,
8      Monitor,Aproximado
9      Var Salda
10     Norma
11     Superf
12
13  Label 2:
14
15  Var
16  X_O,X_A,X_D,Y_O,Y_A,Y_D,Z_O,Z_A,Z_D,X_I,Y_I,Z_I,
17  Long_1,E_1
18  I,J,Indice
19  H,Rn2,Rn3,Ltr,Max,Long,Al,Beta,DI,XO,YO,ZO,A,B,
20  C,Ax1,Cx1,Bx1,Ay1,By1,Cyl,T1,Sds,En,Ce,Ea,Ec,Pk,
21  R,X1,Y1,Z1,Abc,Amx,Senoa,Cossenoa,L,Rad
22  Local,Local_1,Local_2
23  Unlelt
24
25  (*Subrotina recursiva que determina a passagem de uma camada de material a outra e calcula as interseccoes entre
26  as interfaces, bem como a energia dissipada em cada camada. NAO ESTA ATIVADA !*)
27  Procedure Camadas( Var Local,Local_1
28      Var X_A,Y_A,Z_A,En,E_1,A,B,C
29      Var Cof,Cof2,Cof5
30      Var Superf
31
32      Var E_2,Long,X_I,Y_I,Z_I,T      :Real;
33      Local_2      :Material;
34
35  Begin
36  If Local_1 > Local Then Local_2:=Succ(Local) Else Local_2:=Pred(Local); E_2:=0.0;
37  If Local_2 > Retros Then
38      Begin
39      T:=(Z_A-Superf[Local_2])/C;
40      X_I:=X_A+AxT;
41      Y_I:=Y_A+ByT;
42      Z_I:=Superf[Local_2];
43      Long:=Sqrt((Sqr(X_I-X_A)+Sqr(Y_I-Y_A)+Sqr(Z_I-Z_A)));
44      E_1:=En-Cof[Local_2]*Long*ln(Cof2[Local_2]*En)/En;
45      If (Local_1 < Local_2) And (E_1 > Cof5[Local_2]) Then
46      Camadas(Local_2,Local_1,X_A,Y_A,Z_A,E_1,E_2,A,B,C,Cof,Cof2,Cof5,Superf);
47      X_A:=X_I; Y_A:=Y_I; Z_A:=Z_I;
48      En:=En-E_1-E_2; If En < Cof5[Local_1] Then Local_1:=Absorvido;
49      Local:=Local_1;
50  End;

```

```

51 End;
53 Begin
54 Senoa:=Sin(0.01745*Saída.Angulo[Saída.A2]);
55 Cosseno:=Cos(0.01745*Saída.Angulo[Saída.A2]);
56 X_O:=X1; Y_O:=Y1; Z_O:=Z1;
57 (*Esta seccao determina o local inicial em que o eletron se encontrax)
58 Case Caso Of
59   2:Begin
60     Local_1:=Retros;
61     While Z_O >= Superf[Local_1] Do Local_1:=Succ(Local_1);
62   End;
63   3:Begin
64     Rad:=Saída.Angulo[Saída.A2]*0.01745;
65     L:=X_O*Sin(Rad)/Cos(Rad);
66     If Z_O < -L Then Local_1:=Retros Else Local_1:=Mat_1;
67   End;
68   4:Begin
69     If Z_O < 0.0 Then Local_1:=Retros Else If X_O < 0.0 Then Local_1:=Mat_1 Else Local_1:=Mat_2;
70   End;
71 End;
72 (*Aqui inicia a simulacao ate o numero [Simular] de trajetorias *)
73 For I:=1 To Simular Do
74   Begin
75     A:=0.0; B:=0.0; C:=1.0;
76     En:=Saída.Energia[Saída.A1]; Ce:=Saída.P[Saída.A1];
77     Sds:=0.0;
78     Amax:=0.0;
79     X_A:=X_O; Y_A:=Y_O; Z_A:=Z_O;
80     Local:=Local_1;
81     Amax:=0.0;
82     (*O procedimento e' repetido ate o eletron ser retroespalhado ou absorvido *)
83     Repeat
84       (*O procedimento ALEA invoca uma subrotina FORTRAN para gerar numeros pseudoaleatorios*)
85       Alea(I1,I2,A1);
86       If A1=0.0 Then A1:=0.001; (*O numero aleatorio nao pode ser nulo*)
87       Beta:=2.0*Arctan(Ce*(1.0/Sqr(A1)))/En; (*Calculo do angulo de espalhamento*)
88       Ltr:=Sqr(En)*Cof3[Local]/Ln(En)+Cof4[Local]; (*Calculo do livre caminho medio*)
89       Alea(I1,I2,Rn3); (*Gera numero aleatorio para o livre caminho medio*)
90       If Rn3=0.0 Then Rn3:=0.001;
91       D1:=-Ltr*Ln(Rn3); (*Calcula o livre caminho percorrido*)
92       If D1=0.0 Then D1:=0.0001; (*O livre caminho percorrido nao pode ser nulo*)
93       (*Para esta seccao vide o calculo das trajetorias em 3 dimensoes*)
94       H:=D1*Cos(Beta); R:=D1*Sin(Beta);
95       XO:=X_A+H*A; YO:=Y_A+H*B; ZO:=Z_A+H*C;
96       If C=0.0 Then Axl:=0.0 Else Axl:=Cos(Arctan(-A/C));
97       Cxl:=Sqr(1.0-Sqr(Axl));
98       Bxl:=0.0;
99       Ayl:=B*A*Cxl; Byl:=C*A*Axl-Cxl*A; Cyl:=-Axl*B;
100      Abc:=Sqr(Sqr(Ayl)+Sqr(Byl)+Sqr(Cyl));
101      If Not ((Abc < 1.1) And (Abc > 0.9)) Then

```

```

1 Program Ebic(Dados,Result);
2   Var  Corr_AI,Corr_AO,Corr_B,Corr_C
3       Muda
4       Dados,Result
5       Registro,I,J
6       Pos,X,AI,AO,B,C,
7       Norma,Energia,Angulo
8       Define_1,Define_2,Arquivo_1,Arquivo_2
9       Superf
10      Funcao
11
12 Begin
13   (*Leitura do arquivo EBIC_INT.DAT que fornece as diretrizes*)
14   Open(Dados,'Ebic_int.dat',Old); Reset(Dados);
15   Readln(Dados,Arquivo_1); Readln(Dados,Arquivo_2); Close(Dados);
16   Open(Dados,Arquivo_1,Old); Reset(Dados); (*Abre arquivo para os resultados*)
17   Open(Result,Arquivo_2,New); Rewrite(Result); (*Abre arquivo para os resultados*)
18   Readln(Dados,Registro); Readln(Dados,Define_1); Readln(Dados,Define_2);
19   WriteLn(Result,Registro); WriteLn(Result,Define_1); WriteLn(Result,Define_2);
20   Readln(Dados,Superf[1],Superf[2],Superf[3],Superf[4],Superf[5]);
21   WriteLn(Dados,Superf[6],Superf[7],Superf[8],Superf[9],Superf[10]);
22   WriteLn(Result,Superf[1],Superf[2],Superf[3],Superf[4],Superf[5]);
23   WriteLn(Result,Superf[6],Superf[7],Superf[8],Superf[9],Superf[10]);
24   While Not Eof(Dados) Do
25   Begin
26     Readln(Dados,Energia,Angulo,Norma);
27     WriteLn(Result,Energia,Angulo,Norma);
28     Norma:=1.0/Norma;
29     For I:=0 To 1000 Do Read(Dados,Funcao[I]);
30     (*Inicia o calculo, como uma convolucao, das correntes de ebicx*)
31     For I:=0 To 1000 Do
32     Begin
33       For J:=1 To 10 Do
34       Begin
35         C:=Exp(-Superf[21]/J);
36         AI:=1/C-C;
37         AO:=1/C+C;
38         B:=1/C-Exp(-3*Superf[21]/J);
39         X:=Norma*I
40         If X < Superf[1] Then
41         Begin
42           (*Correntes na primeira camada, de cima para baixo. Obs.:Ultima camada crescida*)
43           Corr_AI[J]:=Corr_AI[J]+Funcao[I]*Exp(X/J)-Exp(-X/J))/AI; (*Corrente para recombinao superficial infinita*)
44           Corr_AO[J]:=Corr_AO[J]+Funcao[I]*Exp(X/J)+Exp(-X/J))/AO; (*Corrente para recombinao superficial nula*)
45         End
46       Else
47       Begin
48         If X < Superf[2] Then
49         Begin
50           (*Corrente da camada confinante*)
51           Corr_B[J]:=Corr_B[J]+Funcao[I]*Exp(X/J)-Exp(-(2*Superf[2]+X)/J))/B;

```

```

102 Begin
103   Cx1:=-Cx1;
104   By1:=Cx1-Cx1*A;
105 End;
106 Alea(I1,I2,Rn3): (xGera o numero aleatorio para o angulo radial de espalhamentox);
107 T1:=Rn3*6.28314;
108 X1:=RxCos(T1); Y1:=R*Sin(T1); Z1:=0.0;
109 X_D:=Ax1*X1+Ay1*Y1+X0; Y_D:=Bx1*X1+By1*Y1+Y0; Z_D:=Cx1*X1+Cy1*Y1+Z0;
110 Long:=Sqr(Sqr(X_D-X_A)+Sqr(Y_D-Y_A)+Sqr(Z_D-Z_A)); (*Caminho efetivamente percorrido*)
111 If Long=0.0 Then Long:=0.0001; (*O caminho efetivamente percorrido nao pode ser nulo*)
112 If Z_D > Amax Then Amax:=Z_D;
113 A:=(X_D-X_A)/Long; B:=(Y_D-Y_A)/Long; C:=(Z_D-Z_A)/Long; (*Calcula os novos cossenos diretores*)
114 Sds:=Sds+Long;
115 (*O segmento a seguir verifica a localizacao em termos das camadas ou o "status" de retroespalhadox*)
116 Case Caso Of
117 2: Begin
118   Local_2:=Retros;
119   While Z_D >= Superf[Local_2] Do Local_2:=Succ[Local_2]; E_1:=0.0;
120   (* If (Local_2 <> Local) And (Local_2 > Retros) Then
121   Camadas(Local,Local_2,X_A,Y_A,Z_A,En,E_1,A,B,C,Cof2,Cof5,Superf);*)
122   Local:=Local_2;
123 End;
124 3: Begin
125   Rad:=Saída.Angulo[Saída.A21]*0.01745;
126   L:=X_D*Sin(Rad)/Cos(Rad);
127   If Z_D < -L Then Local:=Retros Else Local:=Mat_1;
128 End;
129 4: Begin
130   If Z_D < 0.0 Then Local:=Retros Else If X_D < 0.0 Then Local:=Mat_1 Else Local:=Mat_2;
131 End;
132 End;
133 If (Local=Retros) Or (Local=Absorvido) Then Goto 2;
134 If Aproximado Then
135   Begin
136     If Ltr < 0.001 Then
137       Begin
138         Ec:=En;
139         En:=0.0;
140         Local:=Absorvido;
141         Goto 2;
142       End;
143       Ea:=En-Cof[Local]*Long*Ln(Cof2[Local]*En)/En; (*Calculo da nova energia do eletrón*)
144       Ec:=En-Ea+E_1; En:=Ea;
145       If En < Cof5[Local] Then (*Comparacao com 1.03 vezes o potencial medio de ionizacao*)
146         Begin
147           Local:=Absorvido;
148           Ec:=Ec+En+E_1;
149           En:=0.0;
150         End;
151       If Saída.Simnao.Shape Then (*Armazenamento do valor da energia dissipada*)
152         Begin

```

```

153  Indice:=Round((-X_DxSenoa+Z_DxCossenoa)*Normal[Salda.A1]);
154  If (Indice <= 1000) And (Indice > -1) Then Salda.Perfil.Z[Indice]:=Salda.Perfil.Z[Indice]+Ec;
155  End;
156  End;
157  X_A:=X_D; Y_A:=Y_D; Z_A:=Z_D; (*A nova posicao passa a ser a anterior*)
158  Until Local < Mat_1; (*Termina a simulacao da trajetoria individual*)
159  2: If Local=Retros Then
160  Begin
161      (*Calculo do coeficiente de retroespalhamento, por numero de eletrons e por energla*)
162      Salda.Retro_Coeff[Salda.A1,Salda.A2]:=Salda.Retro_Coeff[Salda.A1,Salda.A2]+1.0;
163      Salda.Retro_Coeff_Ent[Salda.A1,Salda.A2]:=Salda.Retro_Coeff_Ent[Salda.A1,Salda.A2]+En;
164  End;
165      (*Calculo do alcance medio e desvio padrao*)
166      Alcance.Medio:=Alcance.Medio+Amax; Alcance.Max:=Alcance.Max+Sqr(Amax);
167  End;
168  End;

```



```
51 End
52 Else
53   Begin
54     (xCorrente da camada pos-juncao. Obs.:Primeira camada crescidax)
55     Corr_C[J]:=Corr_C[J]+Funcao[I]*Exp(-X/J)/C;
56   End;
57 End;
58 For J:=1 To 10 Do
59   Begin
60     (xEscreve os resultadosx)
61     Writeln(Corr_A[J],Corr_A0[J],Corr_B[J],Corr_C[J]);
62   End;
63 End;
64 End;
65 End;
66 Close(Result): Close(Dados);
67 End.
```

```

1 Program Ebic(Dados,Result):
2   Var   Dados,Result,Experimental
3   Superf,Eg
4   Funcao
5   Corr_A1,Corr_A0,Corr_B,Corr_C,Energia_E,Angulo_E,
6   Corrente,A1,A0,B,C
7   Muda
8   lex,K,Registro,I,J,Aa,Bb,Cc,Melhor_A1,Melhor_B1,
9   Melhor_C1,Melhor_A0,Melhor_BO,Melhor_CO
10  Pos,X,Diferencial,Diferencao,
11  Minimo1,MinimoO,Soma1,SomaO,
12  Norma,Energia,Angulo
13  Define_1,Define_2,Arquivo_1,Arquivo_2,Arquivo_3
14  Begin
15  Open(Dados,'Ebic_int.dat',Old); Reset(Dados);
16  Readln(Dados,Arquivo_1); Readln(Dados,Arquivo_2); Readln(Dados,Arquivo_3);
17  Readln(Dados,Eg[1],Eg[2],Eg[3],Eg[4],Eg[5]);
18  Readln(Dados,Eg[6],Eg[7],Eg[8],Eg[9],Eg[10]);
19  Close(Dados);
20  Open(Dados,Arquivo_1,Old); Reset(Dados);
21  Open(Result,Arquivo_2,New); Rewrite(Result);
22  Readln(Dados,Registro); Readln(Dados,Define_1); Readln(Dados,Define_2);
23  Writeln(Result,Registro); Writeln(Result,Define_1); Writeln(Result,Define_2);
24  Readln(Dados,Superf[1],Superf[2],Superf[3],Superf[4],Superf[5]);
25  Readln(Dados,Superf[6],Superf[7],Superf[8],Superf[9],Superf[10]);
26  Writeln(Result,Superf[1],Superf[2],Superf[3],Superf[4],Superf[5]);
27  Writeln(Result,Superf[6],Superf[7],Superf[8],Superf[9],Superf[10]);
28  Open(Experimental,Arquivo_3,Old); Reset(Experimental);
29  lex:=0;
30  While Not Eof(Experimental) Do
31  Begin
32    lex:=lex+1;
33    Readln(Experimental,Energia_E[lex],Angulo_E[lex],Corrente[lex]);
34  End;
35  Close(Experimental);
36  For J:=1 To 40 Do
37  Begin
38    C[J]:=Exp(-Superf[2]/(J*0.1));
39    A[J]:=1/C[J]-C[J];
40    A0[J]:=1/C[J]+C[J];
41    B[J]:=1/C[J]+Exp(-3*Superf[2]/(J*0.1));
42  End;
43  While Not Eof(Dados) Do
44  Begin
45    Readln(Dados,Energia,Angulo,Norma);
46    Norma:=1.0/Norma;
47    For I:=0 To 1000 Do Read(Dados,Funcao[I]); Readln(Dados);
48    For I:=0 To 1000 Do
49      Begin
50        X:=Norma*I;

```

```

51  If X < Superf[1] Then
52      Begin
53          For J:=1 To 40 Do Corr_AI[J]:=Corr_AI[J]+Funcao[I]*(Exp(X/(0.1*XJ))-Exp(-X/(0.1*XJ)))/(A[I]*Eg[1]);
54          For J:=1 To 40 Do Corr_AO[J]:=Corr_AO[J]+Funcao[I]*(Exp(X/(0.1*XJ))+Exp(-X/(0.1*XJ)))/(AO[J]*Eg[1]);
55      End
56  Else
57      Begin
58          If X < Superf[2] Then
59              Begin
60                  For J:=1 To 40 Do Corr_B[J]:=Corr_B[J]+Funcao[I]*(Exp(X/(0.1*XJ))+Exp(-(2*Superf[2]+X)/(0.1*XJ)))
61                      /(B[J]*Eg[2]);
62              End
63          Else
64              Begin
65                  If X < Superf[3] Then For J:=1 To 40 Do Corr_C[J]:=Corr_C[J]+Funcao[I]*Exp(-X/(0.1*XJ))/(C[I]*Eg[3]);
66              End;
67          End;
68      End;
69      K:=0;
70      Repeat
71          K:=K+1;
72          If (Energia=Energia_E[K]) And (Angulo=Angulo_E[K]) Then
73              Begin
74                  MinimoI:=9999;
75                  MinimoO:=9999;
76                  For Aa:=1 To 40 Do
77                      Begin
78                          For Bb:=1 To 40 Do
79                              Begin
80                                  For Cc:=1 To 40 Do
81                                      Begin
82                                          SomaI:=Corr_AI[Aa]+Corr_B[Bb]+Corr_C[Cc];
83                                          SomaO:=Corr_AO[Aa]+Corr_B[Bb]+Corr_C[Cc];
84                                          Diferencial:=Abs(SomaI-Corrente[K]);
85                                          DiferencaO:=Abs(SomaO-Corrente[K]);
86                                          If Diferencial < MinimoI Then
87                                              Begin
88                                                  Melhor_AI:=Aa;
89                                                  Melhor_BI:=Bb;
90                                                  Melhor_CI:=Cc;
91                                                  MinimoI:=Diferencial;
92                                              End;
93                                          If DiferencaO < MinimoO Then
94                                              Begin
95                                                  Melhor_AO:=Aa;
96                                                  Melhor_BO:=Bb;
97                                                  Melhor_CO:=Cc;
98                                                  MinimoO:=DiferencaO;
99                                              End;
100                      End;
101                  End;

```

```

102 End;
103 WriteIn(Result); WriteIn(Result);
104 WriteIn(Result,'Energla=',Energla:5:1,' Angulo=',Angulo:6:1);
105 WriteIn(Result,Corr_AltMelhor_Alt),Corr_B[Melhor_B],Corr_C[Melhor_C];
106 WriteIn(Result,Corr_AO[Melhor_AO],Corr_B[Melhor_BO],Corr_C[Melhor_CO]);
107 WriteIn(Result,'L(ai)=',(Melhor_Alt/10):5:2,' L(bi)=',(Melhor_B/10):5:2,' L(ci)=',(Melhor_C/10):5:2);
108 WriteIn(Result,'L(ao)=',(Melhor_AO/10):5:2,' L(bo)=',(Melhor_BO/10):5:2,' L(co)=',(Melhor_CO/10):5:2);
109 WriteIn(Result,'soma(i)=',(Corr_Alt[Melhor_Alt]+Corr_B[Melhor_B]+Corr_C[Melhor_C]):10:3);
110 WriteIn(Result,'soma(o)=',(Corr_AO[Melhor_AO]+Corr_B[Melhor_BO]+Corr_C[Melhor_CO]):10:3);
111 WriteIn(Result,'Corrente experimental=',Corrente[K]:10:3);
112 WriteIn(Result,'diferenca(i)=','Minimo:10:3,' diferenca(o)=','Minimo0:10:3);
113 End;
114 Until K>=lex;
115 For J:=1 To 40 Do Corr_Alt[J]:=0.0;Corr_AO[J]:=0.0;Corr_B[J]:=0.0;Corr_C[J]:=0.0;
116 End;
117 Close(Result); Close(Dados);
118 End.

```

```

1 Program Desenho_Deds(Dados,Input,Output);
2 Var
3   P      Perfil
4   Dados
5   Registro,Kout,I,J,L,K
6   Esp_Jun,Prof_Jun,Camada,Substrato,D,
7   Max, Nor, Inc
8   Norma,Energia,Angulo
9   Definicao_1,Definicao_2,Arquivo
10 Procedure Axis(Xpage,Ypage:Real;Xstdescr Text:Packed Array[Integer] Of Char;Carc:Integer;Long,Angle,Minim,Incre:Real);Fortran
11 n;
12 Procedure Plots(Xlong,Ywide:Real; Outdev:Integer); Fortran;
13 Procedure Plot(Xpage,Ypage:Real; Pen:Integer); Fortran;
14 Procedure Symbol(Xpage,Ypage,Height:Real;
15   %Stdescr Text:Packed Array[Integer] Of Char;
16   Angle: Real; Nchar:Integer); Fortran;
17 Procedure Number(Xpage,Ypage,Height,Digit,Angle:Real;
18   Ndel:Integer); Fortran;
19 Procedure Factor(Fact:Real); Fortran;
20 Procedure Dashp(Xpage,Ypage,Dash:Real);Fortran;
21 Kout:=100;
22 Write('NOME DO ARQUIVO:'); Readln(Arquivo);
23 Open(Dados,'DBA1:WEBER.ESTOQUE'+Arquivo+'.DAT',Old);
24 Reset(Dados);
25 Readln(Dados,Registro); Readln(Dados,Definicao_1); Readln(Dados,Definicao_2);
26 If Registro > 1000 Then
27   Begin
28     Readln(Dados);Readln(Dados);
29   End
30 Else
31   Begin
32     Readln(Dados,Prof_Jun,Esp_Jun,Camada,Substrato);
33   End;
34 Max:=0.0; I:=0;
35 While Not Eof(Dados) Do
36   Begin
37     I:=I+1;
38     Readln(Dados,Energia[I],Angulo[I],Norma[I]);
39     Norma[I]:=1000/Norma[I];
40     For J:=0 To 1000 Do
41       Begin
42         Read(Dados,P[J]); If (J Div 5)*5=J Then Readln(Dados);
43         If P[J] < 0.0 Then P[J]:=0.0;
44       End;
45       For J:=0 To 199 Do
46         Begin
47           For K:=0 To 4 Do Perfill[I,J]:=Perfill[I,J]+P[J*5+K];
48           Perfill[I,J]:=Perfill[I,J]*0.2/Norma[I]; If Perfill[I,J] > Max Then Max:=Perfill[I,J];
49         End;
50       End;

```

```

51 Close(Dados);
52 Nor:=0.0; For J:=1 To I Do If Normal[I] > Nor Then Nor:=Normal[I];
53 (*FormularioX)
54 Plots(10.0,13.0,Kout);
55 Factor(0.393701);
56 Plot(1.0,1.0,-3); Plot(24.0,0.0,2); Plot(24.0,31.0,2);
57 Plot(0.0,31.0,2); Plot(0.0,0.0,2); Plot(2.0,0.0,3);
58 Plot(2.0,31.0,2); Plot(2.0,15.5,3); Plot(0.0,15.5,2);
59 (*GraficoX)
60 (*Mudanca De OrigemX) Plot(18.0,2.0,-3);
61 Symbol(-9.0,-1.0,0.3,'DE/DS (KEV/MICRON)',180.0,1);
62 Max:=15.0/Max; Nor:=25.0/Nor;
63 Symbol(1.0,20.0,0.3,'PROFUNDIDADE (MICRON)',90.0,1);
64 For L:=1 To I Do
65   Begin
66     Inc:=Normal[J]*Nor/200.0;
67     Plot(0.0,0.0,3);
68     For J:=0 To 199 Do
69       Begin
70         Plot(-Perfil[L,J]*Max,J*Inc,2);
71         Plot(-Perfil[L,J]*Max,(J+1)*Inc,2);
72       End;
73     End;
74     (*ElxosX)
75     Factor(2.5);
76     Axis(0.0,0.0,1,-2,10.0,90.0,0.0,2.5/Nor);
77     Axis(0.0,0.0,1,2,6.0,180.0,0.0,2.5/Max);
78     Factor(0.4);
79     (*Estrutura De CamadasX)
80     If Prof_Jun > 0.0 Then Begin Plot(0.0,Prof_Jun*Nor,3); Plot(-15.0,Prof_Jun*Nor,2); End;
81     If Esp_Jun > 0.0 Then Begin Plot(0.0,(Esp_Jun+Prof_Jun)*Nor,3); Plot(-15.0,(Esp_Jun+Prof_Jun)*Nor,2); End;
82     If Camada > 0.0 Then Begin Plot(0.0,Camada*Nor,3); Plot(-15.0,Camada*Nor,2); End;
83     If Substrato > 0.0 Then Begin Plot(0.0,Substrato*Nor,3); Plot(-15.0,Substrato*Nor,2); End;
84     (*Textox)
85     (*Mudanca De OrigemX) Plot(6.0,-2.0,-3);
86     Symbol(-4.0,0.5,0.4,'ENERGIA DISSIPADA',90.0,1);
87     Symbol(-3.0,0.5,0.3,Definicao_1,90.0,1);
88     Symbol(-2.5,0.5,0.3,Definicao_2,90.0,1);
89     Symbol(-1.0,0.5,0.3,'REGISTRO NR.: ',90.0,1);
90     Number(-1.0,4.0,0.3,Registro,90.0,-1);
91     Symbol(-4.0,17.0,0.4,'MICROSCOPIA ELETRONICA',90.0,1);
92     Symbol(-3.0,16.0,0.3,'SIMULACAO MONTE CARLO',90.0,1);
93     Symbol(-1.5,15.0,1.5,'LPD',90.0,1);
94     Symbol(-2.0,19.0,0.5,'LABORATORIO DE PESQUISA',90.0,1);
95     Symbol(-1.0,19.0,0.5,'EM DISPOSITIVOS',90.0,1);
96     Plot(0.0,0.0,399);
97   End.

```

```

1  (*$u+*)
2  Program Ebic_Juncao(Input,Output);
3  Label 1;
4  Var
5      P1,P2,Energia,Angulo      :Array[1..20] of Real;
6      Dados,Result              :Text;
7      Registro,j,k,h,i,m,n,iJuncao,ileit :Integer;
8      Juncao,Norma,F1,F2,F3,max   :Real;
9      Definicao_1,Definicao_2,Arquivo :String[40];
10     S_n                        :Char;
11  Begin
12      clrscr;
13      S_n:='n';
14      Write('Nome do Arquivo:'); Readln(Arquivo);
15      Assign(Dados,Arquivo+'.dat');
16      1:Reset(Dados);
17      If S_n='s' Then
18          Begin
19              clrscr;
20              WriteLn('Arquivo Pesquisado : ',Arquivo,'.dat');
21              Readln(Dados,Registro); Readln(Dados,Definicao_1); Readln(Dados,Definicao_2);
22              Readln(Dados);
23              Write('Profundidade da Juncao:'); Readln(Juncao); i:=0;
24              WriteLn(Definicao_1); WriteLn(Definicao_2); WriteLn('Registro Nr.: ',registro:5);
25              gotoxy(1,10); write('Energia:');
26              gotoxy(1,12); write('Juncao:');
27              gotoxy(30,8); write('Energia Angulo   i(s=0)   l(s=infinito)');
28              While not Eof(Dados) do
29                  Begin
30                      i:=i+1; Readln(Dados,Energia[i],Angulo[i],Norma);
31                      gotoxy(10,10); write(energia[i]:6:1);
32                      Norma:=1000.0/Norma;
33                      ljuncao:=Round(157*Juncao/norma); ileit:=ljuncao div 2;
34                      gotoxy(10,12); write(ljuncao:6); p1[i]:=0.0; p2[i]:=0.0;
35                      If ljuncao > 157 Then
36                          Begin
37                              for j:=1 to 84 do readln(dados);
38                              p1[i]:=0.0; p2[i]:=0.0;
39                              end
40                          Else
41                              Begin
42                                  For j:=1 to (ileit-1) do Readln(Dados);
43                                  If Odd(ljuncao) Then
44                                      Begin
45                                          Readln(Dados);
46                                          Read(Dados,F1,F2,F3); Readln(Dados);
47                                          End
48                                      Else
49                                          Begin
50                                              Read(Dados,F1,F2,F3); Readln(Dados,f1,f2,f3); Readln(Dados);

```

```

51 End;
52 For J:=1 to 83 do Readln(Dados);
53 P1[J]:=F1+F2; P2[J]:=F2+F3;
54 End;
55 gotoxy(30,8+1); write(energia[J]:5:1, ' ', angulo[J]:5:1);
56 if (p1[J]=0.0) And (p2[J]=0.0) Then
57 Begin
58 Write(' * Dados Nao Disponiveis *');
59 End
60 Else
61 Begin
62 Write(' ', p1[J]:10:2, ' ', p2[J]:10:2);
63 End;
64 if Eoln(Dados) Then Readln(Dados);
65 End;
66 for k:=1 to 4 do
67 begin
68 gotoxy(1,9+k); write(' ');
69 end; max:=0.0;
70 for J:=1 to 1 do
71 begin
72 if p1[J] > max Then max:=p1[J];
73 if p2[J] > max Then max:=p2[J];
74 end;
75 for J:=1 to 1 do
76 begin
77 h:=round(14*p1[J]/max);
78 l:=round(14*p2[J]/max);
79 for m:=1 to h do
80 begin
81 gotoxy(J,20-m); if m=h then write('O') else write(' ');
82 end;
83 for n:=1 to l do
84 begin
85 gotoxy(12+J,20-n); if n=l then write('I') else write(' ');
86 end;
87 end;
88 gotoxy(1,22); Write('Deseja continuar ? (s ou n):'); Readln(S_N);
89 if S_N='s' Then GoTo 1;
90 Close(dados);
91 End.

```



```

1 (SU+)
2 Program Ebic_Juncao(Input,Output);
3 Label 1;
4 Var
5   energia_2,angulo_2
6   Dados,Result,dados_2
7   Registro,f,J,K,H,L,M,N,Ijuncao,ileit,
8   dados_simul,dados_ex
9   Juncao,norma,F1,F2,F3,Max,Rie
10  Definicao_1,Definicao_2,Arquivo,
11  Arquivo_2
12  S_N
13 Begin
14   Clrscr;
15   S_N:='n';
16   Write('Nome do Arquivo:'); Readln(Arquivo);
17   Assign(Dados,Arquivo+'.dat');
18   Reset(Dados);
19   If S_N='s' Then
20     Begin
21       Clrscr;
22       Writeln('Arquivo Pesquisado : ',Arquivo,'.dat');
23       End;
24   Readln(Dados,Registro); Readln(Dados,Definicao_1); Readln(Dados,Definicao_2);
25   Readln(Dados);
26   Write('Profundidade da Juncao:'); Readln(Juncao); I:=0;
27   Writeln(Definicao_1); Writeln(Definicao_2); Writeln('Registro Nr.: ',Registro:5);
28   Write('R I E: '); Read(Rie);
29   Gotoxy(1,10); Write('Energia:');
30   Gotoxy(1,12); Write('Ijuncao:');
31   Gotoxy(30,8); Write('Energia Angulo I(s=0) I(s=Infinito)');
32   While Not Eof(Dados) Do
33     Begin
34       I:=1+I; Readln(Dados,Energia[I],Angulo[I],Norma);
35       Gotoxy(10,10); Write(Energia[I]:6:1);
36       Norma:=1000.0/Norma;
37       Ijuncao:=Round(167*Juncao/Norma); Ileit:=Ijuncao Div 2;
38       Gotoxy(10,12); Write(Ijuncao:6); P1[I]:=0.0; P2[I]:=0.0;
39       If Ijuncao > 167 Then
40         Begin
41           For J:=1 To 84 Do Readln(Dados);
42           P1[I]:=0.0; P2[I]:=0.0;
43         End
44       Else
45         Begin
46           For J:=1 To (Ileit-1) Do Readln(Dados);
47           If Odd(Ijuncao) Then
48             Begin
49               Readln(Dados);
50               Read(Dados,F1,F2,F3); Readln(Dados);

```

```

51 End
52 Else
53 Begin
54   Read(Dados,F1,F2,F3); Readln(Dados,F1,F2,F3); Readln(Dados);
55 End;
56 For J:=(11*10+1) To 83 Do Readln(Dados);
57   P1[J]:=(F1+F2)/R1e; P2[J]:=(F2+F3)/R1e;
58 End;
59 Gotoxy(30,8+1); Write(Energia[J]:5:1,' ',Angulo[J]:5:1);
60 If (P1[J]=0.0) And (P2[J]=0.0) Then
61   Begin
62     Write(' * Dados Nao Disponiveis *');
63   End
64 Else
65   Begin
66     Write(' ',P1[J]:10:2,' ',P2[J]:10:2);
67   End;
68   If Ecin(Dados) Then Readln(Dados);
69 End;
70 dados_simul:=1;
71 For K:=1 To 4 Do
72   Begin
73     Gotoxy(1,9+K); Write(' ');
74   End; Max:=0.0;
75   For J:=1 To 1 Do
76     Begin
77       If P1[J] > Max Then Max:=P1[J];
78       If P2[J] > Max Then Max:=P2[J];
79     End;
80   For J:=1 To 1 Do
81     Begin
82       H:=Round(14*P1[J]/Max);
83       L:=Round(14*P2[J]/Max);
84       For M:=1 To H Do
85         Begin
86           Gotoxy(J,20-M); If M=H Then Write('O') Else Write(' ');
87         End;
88       For N:=1 To L Do
89         Begin
90           Gotoxy(12+J,20-N); If N=L Then Write('I') Else Write(' ');
91         End;
92       End;
93       Gotoxy(1,22); write('Carrega Arquivo com dados experimentais? (s ou n): ');
94       read(S_N);
95       If (S_N='S') or (S_N='s') then
96         Begin
97           Gotoxy(1,22); direct; write('Nome do arquivo com os dados experimentais:');
98           read(arquivo_2); assign(dados_2,arquivo_2+'.dat'); reset(dados_2); l:=1;
99           while not eof(dados_2) do
100             begin
101               readln(dados_2,energia_2[J],angulo_2[J],experimental[J]);

```

```

102  l:=l+1;
103  end;
104  close(dados_2);
105  gotoxy(46,8); clreol; write(' I (experimental)');
106  dados_ex:=l;
107  for j:=1 to l do
108    begin
109      gotoxy(30,8+j); clreol; write(energia_2[j]:5:1,' ',angulo_2[j]:5:1,' ',
110      experimental[j]:10:2);
111    end;
112    for j:=(l+8) to 2l do
113      begin
114        gotoxy(30,j); clreol;
115      end;
116    For j:=1 To l Do
117      Begin
118        H:=Round(14*experimental[j]/Max);
119        For M:=1 To H Do
120          Begin
121            Gotoxy(j,20-M); Write('E');
122          End;
123        For N:=1 To h Do
124          Begin
125            Gotoxy(12+j,20-N); Write('E');
126          End;
127        End;
128      end;
129      Gotoxy(1,22); clreol; Write('Deseja continuar ? (s ou n):'); Readln(S_N);
130      If S_N='s' Then Goto 1;
131      Close(Dados);
132    End.

```

Bibliografia

Introdução

- [1] NBS Special Publication 460 "Use of Monte Carlo calculations in Electron Probe Microanalysis and Scanning Electron Microscopy" Edited by K.F.J.Heinrich et al.

1 - O Método Monte Carlo

- [1] K.F.J.Heinrich "Electron Beam X-Ray Microanalysis", Ed.Van Nostrand p.227 (1981)
- [2] D.Liljequist "A simple analysis of the transmission and backscattering of 10-30keV electrons in solid layers" J.Phys.D:Appl.Phys. 10,1363(1977)

2 - Resultados dE/dz

- [1] S.Adachi "Material parameters of $\text{In}(1-x)\text{Ga}(x)\text{As}(y)\text{P}(1-y)$ and related binaries" J.Appl.Phys 53(12),8775(1982)

3 - Corrente induzida por feixe eletrônico (EBIC)

- [1] J.I.Hanoka, R.O.Bell "Electron-beam-induced currents in semiconductors" Ann.Rev.Mater.Sci. 11,353(1981)

- [2] T.E.Everhart PhD Thesis Cambridge Univ. 1958
- [3] H.Pfister "Elektronenbestrahlung von p-n-Sperrschichten in GaAs" Z.Naturforschung 12a,217(1957)
- [4] J.J.Lander, H.Schreiber Jr., T.M.Buck, J.R.Matheus Appl.Phys.Lett. 3,206(1963)
- [5] W. van Roosbroeck "Injected Current Carrier Transport in a Semi-Infinite Semiconductor and the Determination of Lifetimes and Surface Recombination Velocities" J.Appl.Phys. 26(4),380(1955)
- [6] E.Butkov "Física Matemática" Ed.Guanabara Dois (1983)
- [7] C.Donolato "A reciprocity theorem for charge collection" Appl.Phys.Lett. 46(3),270(1985)
- [8] R.V.Churchill "Variáveis Complexas e suas Aplicações" Ed.McGraw-Hill (1976)
- [9] W.H.Hackett Jr. "Electron-beam Minority-Carriers Diffusion Profiles in Semiconductors" J.Appl.Phys. 43(4),1649(1972)

4 - Resultados da simulação de EBIC .

- [1] S.Adachi "GaAs, AlAs and Al(x)Ga(1-x)As: Material parameters for use in research and device applications" J.Appl.Phys. 58(3),R1(1985)
- [2] S.Adachi "Material parameters of In(1-x)Ga(x)As(y)P(1-y) and related binaries" J.Appl.Phys. 53(12),8775(1982)
- [2] C.A.Ribeiro, comunicação particular a partir de levantamentos bibliográficos

A - Equação de Perda de Energia

- [1] H.Bethe "Zur Theorie des Durchgangs schneller Korpuskularstrahlen durch Materie" Ann.d.Physik 5,325(1930)
- [2] A.J.Antolak, W.Williamson Jr. "Electron backscattering from bulk materials" J.Appl.Phys. 58(1),526(1985)
- [3] J.D.Jackson "Eletrodinâmica Clássica" segunda edição, Editora Guanabara Dois pág.478 (1983)

B - Livre Caminho Médio

- [1] L.Landau, E.Lifchitz "Mécanique Quantique" Editora Mir (1966)
- [2] L.Reimer "Monte-Carlo-Rechnungen zur Elektronendiffusion" Optik 27(2),86(1986)

C - Ângulo de Espalhamento

- [1] L.Landau, E.Lifchitz "Mecânica" Editora Mir (1978)

D - Coordenadas do Elétron em Três Dimensões

- [1] R.L.Myklebust, D.E.Newbury, H.Yakowitz "NBS Monte Carlo electron trajectory calculation program" NBS Special Publication 460 p.105 (1975)

E - Energias de Ionização Radiativa

- [1] C.A.Klein J.Appl.Phys. 39(4),2029(1968)

- [2] O.Bogdankévitch, S.Darzneki, P.Elisséev "Les Lasers à Semiconducteurs" Ed.Mir (1979)