

Parametrização de chuveiros atmosféricos inclinados do Observatório Pierre Auger

6 de Junho de 2011

Aluno: João Pinheiro Neto¹

Orientador: José Augusto Chinellato²



Resumo

O presente projeto visa parametrizar o sinal do detector de superfície do Observatório Pierre Auger. Consideramos aqui um novo setup do módulo de superfície, que introduz uma camada de solo ao redor do módulo. Utilizamos uma versão dedicada do toolkit de simulação Geant4, obtemos a curva de resposta que reproduz o depósito de energia das partículas mais comuns de um chuveiro atmosférico extenso. Em função da quantidade de análise de dados envolvida, escrevemos um software em shell script que automatizou esse processo. O software pode ser adaptado para outras tarefas que necessitem de fitting de dados em massa.

¹Email: joaopn at ifi.unicamp.br

²Email: chinella at ifi.unicamp.br

1 Introdução

A Terra é constantemente bombardeada por partículas subatômicas de origem cósmica, os chamados raios cósmicos. O estudo dos raios cósmicos de altíssima energia (UHECR) tem atraído muita atenção, e tem como objetivo principal a caracterização da origem de tais partículas. Ao colidir com a atmosfera, o raio cósmico (chamado de primário) inicia uma cascata de partículas e radiação, chamado de chuva atmosférica extensa (EAS), que pode ter na ordem de 10^{10} partículas[1] no seu máximo. Os UHECR tem energia da ordem de 10^{20} eV, e são as partículas mais energéticas do universo conhecido.

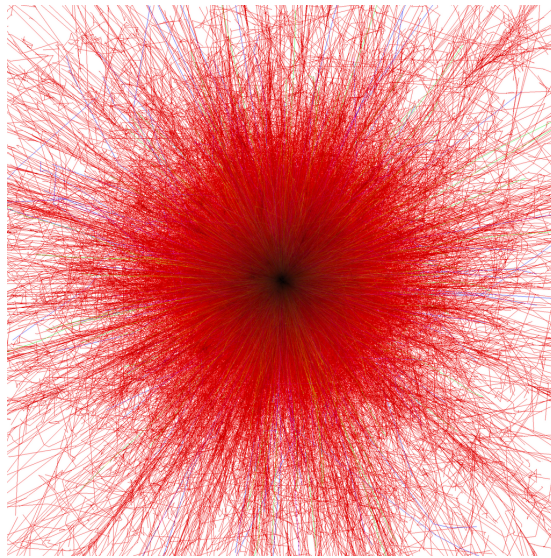


Figura 1: Simulação de chuva iniciada por um núcleo de Fe de 10^{13} eV, visto de baixo. Créditos: F. Schmidt, “CORSIKA Shower Images”

O estudo dos raios cósmicos de mais alta energia e chuveis resultantes é o propósito principal do Observatório Pierre Auger (OPA), que irá operar em dois sítios (na Argentina e em local a ser determinado no hemisfério norte), de forma a contemplar todo o céu. O Observatório utiliza uma técnica híbrida de detecção: o Detector de Fluorescência e o Detector de Superfície. O Detector de Fluorescência utiliza um esquema com telescópios para observar a luz de fluorescência decorrente do chuvi, e com isso observar o seu desenvolvimento longitudinal. O Detector de Superfície é responsável pela detecção das partículas que chegam ao solo, e é formado por tanques de água deionizada (chamados módulos de superfície), cada um contendo 3 fotomultiplicadoras, painéis solares, antenas de comunicação e eletrônica de suporte. As partículas do chuveis que chegam ao tanque interagem com a água no interior do tanque e produzem luz Cherenkov, que será detectada pelas fotomultiplicadoras.

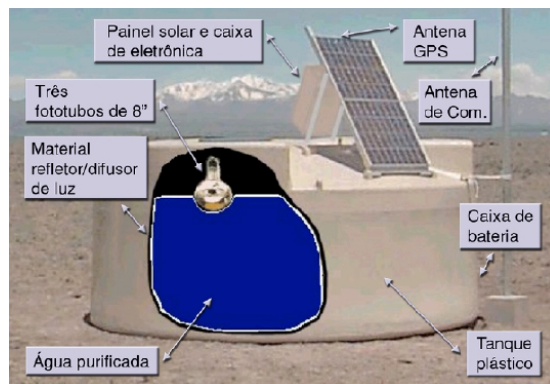


Figura 2: Esquema de um módulo de superfície do sítio Sul do OPA.

Atualmente o sítio Sul (Argentina), em pleno funcionamento, conta com 1600 detectores de superfície espalhados por uma área de 3000 km^2 . Dada a sua escala, o OPA já acumulou uma exposição total superior a todos os outros experimentos de chuviros atmosféricos reunidos.[2]

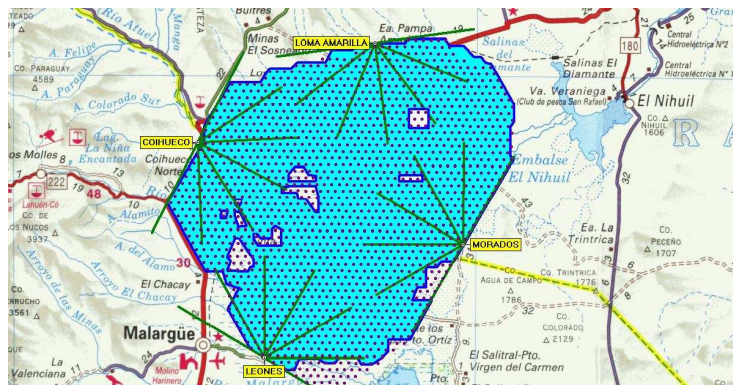


Figura 3: Esquema mostrando o sítio Sul. Nas extremidades do arranjo de tanques, temos os telescópios de fluorescência. Em azul são mostrados os tanques em funcionamento.

2 Simulação e parametrização do sinal do módulo de superfície

O objetivo deste trabalho é criar uma parametrização da resposta do detector de superfície a partículas individuais incidindo em ângulos inclinados. A escolha se deve ao fato de chuveiros quase-horizontais possuírem propriedades que os

diferenciam de chuveiros verticais.[3] A motivação é redução do tempo computacional da simulação do sinal de um EAS, que pode levar meses em função da sua grande cadeia de interações.

Utilizamos o toolkit Geant4³[6], em cima do qual foi desenvolvida por M.A. Muller a simulação tank0Mod[4], que simula o detector incluindo o solo ao seu redor. O tank0Mod utiliza 3 variáveis na simulação: tipo de partícula, ângulo de entrada e energia da partícula.

Fizemos simulações a valores discretos de energia no intervalo de eV até 10 TeV para os tipos de partículas mais relevantes de um EAS para obtermos uma parametrização geral do sinal que permita fazer interpolações para qualquer valor de energia. Podem ocorrer os seguintes casos para a função de distribuição de probabilidade do depósito de energia:

- Sem produção de sinal nas fotomultiplicadoras.
- Exponencial + gaussiana com 5 parâmetros.

Explicitamente, a curva escolhida para o ajuste da distribuição de fotoelétrons é

$$F(n) = \alpha e^{-\beta n} + \gamma \exp\left(-\frac{(n - \mu)^2}{2\sigma^2}\right) \quad (1)$$

onde n é o número de fotoelétrons, e os 5 parâmetros são α (constante linear da exponencial), β (constante do expoente da exponencial), γ (constante linear da gaussiana), μ (média da gaussiana) e σ (desvio padrão da gaussiana).

As partículas da frente do EAS foram injetadas em posições aleatórias em uma área de $6 \times 6\text{ m}^2$ (essa é a área de nossa configuração, que inclui o tanque Cherenkov mais uma área lateral de solo) na altura do topo do tanque, com um ângulo fixo de 60° (Esse ângulo tem relação com o ângulo zenital de entrada das partículas no tanque). Para cada partícula/energia simulamos em média 3000 vezes, de forma a obter uma boa estatística.

Quando uma partícula é injetada aleatoriamente na área acima, consideram-se as possibilidades:

- a) ela não provoca depósito de energia na água (é bom lembrar que pode haver um depósito por albedo). Nesse caso, ela não é considerada na estatística abaixo. A probabilidade de depósito de energia depende do tipo de partícula e de sua energia.
- b) ela provoca depósito de energia e nesse caso:
 - 1) não gera fótons detectáveis pelas fotomultiplicadoras
 - 2) gera fótons detectáveis pelas fotomultiplicadoras. Nesse caso pudemos determinar um formato universal para a distribuição do número de fotoelétrons (que pode ser facilmente convertido para VEM), que é a soma de uma exponencial mais uma gaussiana.

³Ele é considerado o “estado da arte” da simulação de interação de partículas, sendo muito usado tanto em física de altas energias como em física médica.

De posse dos parâmetros dos ajustes de tais curvas, que são 5, sendo 2 deles para a exponencial e 3 para a gaussiana, do tipo de partícula e de sua dependência com a energia, temos um procedimento de cálculo para o número de fotoelétrons.

Com os itens **a** e **b** acima citados, temos uma sequência de filtros a serem aplicados a uma partícula que chega a nossa configuração. O resultado do procedimento é o número de fotoelétrons gerados. A esse procedimento de cálculo daremos o nome de “função resposta”. Ressaltamos que com esse procedimento teremos um ganho de tempo de CPU da ordem de milhares de vezes. Como exemplo, podemos citar a simulação de uma partícula com energia da ordem de alguns *GeV*s que leva em torno de 200 *s*, e com as curvas de parametrização esse tempo será reduzido para a ordem de milissegundos.

Como nem toda partícula que chega à configuração gera fótons detectáveis, implementamos um fator de correção na forma de uma curva de probabilidade de interação em função da energia, para cada partícula. Ela é feita pelo fitting das energias simuladas, onde a probabilidade é dada pela razão entre o n° de simulações com depósito de energia e n° total de simulações.

3 Automatização da parametrização

Em função da quantidade enorme de dados a serem analisados, escolhemos escrever um conjunto de scripts capaz de automatizar a análise de dados. A linguagem escolhida foi Shell Script (mais especificamente, com o interpretador *bash*), em função da sua facilidade de manipulação de arquivos de texto. Por exemplo, a substituição de uma string de texto em um arquivo pode ser feita com a linha “*sed s/texto_antigo/texto_novo arquivo*”. Tal manipulação seria consideravelmente mais complexa se feita na linguagem C++, por exemplo.

Os scrips pegam os dados de contagem de fotoelétrons do detector⁴ e encontram o melhor fit das distribuições de probabilidade de fotoelétrons, a função de probabilidade de interação (muitas partículas atravessam o detector sem interagir) e evolução com a energia dos parâmetros das curvas ajustas. O algoritmo utilizado para os ajustes das distribuições de fotoelétrons é simples: manda-se o software ROOT[7] gerar e fazer o ajuste de diversos histogramas com números variados de bins e combinações variadas das curvas exponencial e gaussiana. Posteriormente escolhe-se o fit com menor χ^2 como o melhor. Decidimos tornar os scripts modulares, atacando uma parte do processo de análise por vez. Abaixo temos um esquema do funcionamento dos scrips:

⁴Este sera o único output da simulação, pois não estamos analisando a distribuição temporal do sinal.

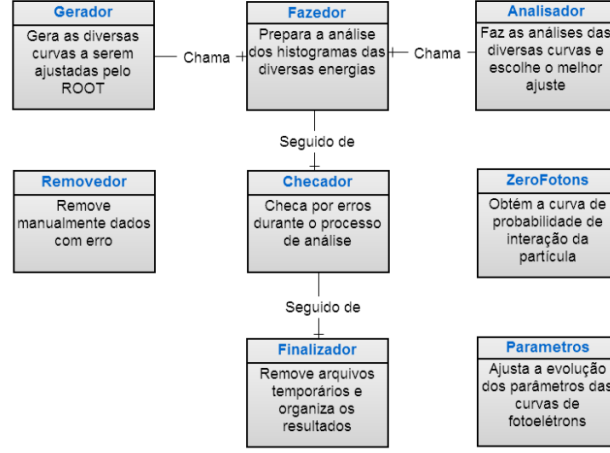


Figura 4: Esquema de funcionamentos dos scripts.

Uma dificuldade encontrada foi o funcionamento pouco previsível do ROOT diante de erros. Em muitos casos a seção de choque da partícula com o detector é tão pequena que não há uma real distribuição de probabilidade de fotoelétrons, e o ajuste da curva não converge. Nesses casos o ROOT pode travar de diversas formas, e o script teve que ser escrito robusto o suficiente para lidar com todas as formas com que o ROOT pode falhar.

Posteriormente foi escrito o script ZeroFotons, que obtém a probabilidade de interação da partícula. Este ofereceu o seu conjunto de desafios, como por exemplo a inabilidade do Shell Script de trabalhar com números de ponto flutuante. Visto que não há um modelo teórico para essa curva e o objetivo aqui é somente alcançar uma boa aproximação numérica, foi usada uma expressão polinomial com vários termos. Não há nenhuma tentativa de interpretar os termos da expressão. No momento atual o script utiliza o software gnuplot[8] e faz um ajuste do tipo

$$f(x) = ax^b + cx^d + ex^f + gx^h + ix^j + k \quad (2)$$

O último script desenvolvido foi o script Parametros, que faz uma análise similar ao script ZeroFotons para os 5 parâmetros que definem as curvas gaussianas e exponenciais ajustadas nos histogramas. Ele também utiliza o software gnuplot, e ajusta a mesma curva do ZeroFotons. Com isso obtemos a evolução em função da energia de cada parâmetro, e podemos reconstruir a curva de fotoelétrons para qualquer energia na faixa simulada.

4 Desenvolvimentos futuros

Pretendemos rodar mais simulações com o tank0Mod de forma a cobrir melhor o espaço de parâmetros. Devemos também desenvolver a função de resposta

da parametrização, utilizando os dados simulados e parametrizados. Esta deve obter de input o tipo de partícula, ângulo de entrada e energia, e devolver o n^o de fotoelétrons parametrizado. Ela deverá ser escrita na forma de uma biblioteca em C/C++, para que possa ser futuramente integrada no *framework* Offline⁵ da Colaboração Auger, acelerando a simulação de EASs.

Referências

- [1] Góra, D. *et al.*, *Astropart. Phys.* **16** 129 (2001)
- [2] Roulet, E. “Latest results from the Pierre Auger Observatory”. arXiv:1101.1825v1 [astro-ph.HE] (2011)
- [3] Ave, M. *et al.*, *Astropart. Phys.* **14** 91 (2000)
- [4] Müller, M. A., *Estudo sobre as Interações de Hádrons nos Módulos de Superfície e Adjacências, do Observatório Pierre Auger*, Tese de Doutorado, IFGW-Unicamp (2007)
- [5] Burtch, K. *Linux Shell Scripting with Bash*. Sams Publishing, Indiana. 2004. (referência utilizada em Shell Script)
- [6] Disponível em <http://wwwinfo.cern.ch/asd/geant4/geant4.html> (acessado em 06/06/11)
- [7] Disponível em <http://root.cern.ch/> (acessado em 06/06/11)
- [8] Disponível em www.gnuplot.info/ (acessado em 06/06/11)

⁵O Offline é o software utilizado pela Colaboração Auger para simulação, reconstrução e calibração de chuveis atmosféricos. Para essas tarefas, ele utiliza código próprio e softwares externos como Geant4, ROOT, Corsika e AIREs.

Anexo A

Código do script Fazedor

```
#!/bin/bash
#Joao PN - joaoxp-at-gmail.com

if [ ! -s nomes_histogramas ]; then
    mkdir Histogramas
    mv histo_* Histogramas
    mv saida_teste_histo_* Histogramas
    cd Histogramas
    echo histo_* > nomes_histogramas_temp
    sed -i s/histo_//g nomes_histogramas_temp
    tr ' ' '\n' < nomes_histogramas_temp >nomes_histogramas
    rm -f nomes_histogramas_temp #Remendo enquanto nao descubro como fazer o tr mexer no proprio arquivo
    cd ..
    mv Histogramas/nomes_histogramas nomes_histogramas
    mkdir Resultados

    #Organiza e copia os arquivos de analise das energias para as respectivas pastas
    exec 3< nomes_histogramas
    while read linha <&3; do
        var_energia_nome=$linha
        mkdir Analise_$var_energia_nome
        mkdir Resultados/$var_energia_nome
        cp Analisador Analise_$var_energia_nome/
        cp Gerador Analise_$var_energia_nome/
        cp histograma_base.C Analise_$var_energia_nome/
        cp histograma_base_uma_curva.C Analise_$var_energia_nome/
        cp Histogramas/histo_$var_energia_nome Analise_$var_energia_nome/
        cp Histogramas/histo_$var_energia_nome Resultados/$var_energia_nome
    done
fi
#Faz toda a analise de todos os histogramas de todas as energias
exec 3< nomes_histogramas
while read linha <&3; do
    var_energia_nome=$linha
    cd Analise_$var_energia_nome
    declare -x energia_nome=$var_energia_nome
    declare -x limite_de_barras="31"
    ./Gerador
    ./Analisador
    cd ..

    cp Analise_$var_energia_nome/FINALHISTOGRAMA_* Resultados/$var_energia_nome
    cp Analise_$var_energia_nome/pos-analise.C Resultados/$var_energia_nome

    cd Resultados/$var_energia_nome
    root -l -b -q pos-analise.C
    cd ..
    cd ..
done

exit 0
```


Anexo B

Código do script Gerador

```
#!/bin/bash
#Joao PN - joaoxp-at-gmail.com

#Espaço para código que fornece faixa de valores dos dados
#e: comprimento das barras
#Código para achar a energia máxima da plotagem

var_teste="0"
while [ $var_teste -eq 0 ]; do

    energia_final="0"
    teste_e="0"
    teste_e2="0"
    exec 3< histo_$(energia_nome)
    while read linha <&3; do
        if [ $(energia_final) -lt $linha ]; then
            energia_final=$linha
        fi
    done

    exec 3< histo_$(energia_nome)
    while read linha <&3; do
        if [ $(teste_e) -lt $linha ] && [ $linha -ne $(energia_final) ]; then
            teste_e=$linha
        fi
    done
    let "teste_e2 = teste_e + teste_e"
    if [ $(energia_final) -lt $teste_e2 ]; then
        var_teste="1"
    fi
    if [ $var_teste -eq 0 ]; then
        sed -i /$(energia_final)/d histo_$(energia_nome)
    fi

    if [ $(energia_final) -eq 0 ]; then
        var_teste="1"
    fi
done

#Faz tudo para 20, 25, ... ,limite_de_barras
numero_de_barras="20"
while [ $numero_de_barras -lt $limite_de_barras ]; do
    let "e = energia_final / numero_de_barras"
    #Gera os arquivos para análise
    numero_histo="0"
    #Copia arquivos para outra pasta, para organizacao
    while (($numero_histo < $numero_de_barras)); do
        mkdir analise$numero_de_barras
        cp histograma_base_uma_curva.C analise$numero_de_barras
        cp histograma_base.C analise$numero_de_barras
        cp histo_$(energia_nome) analise$numero_de_barras
        cd analise$numero_de_barras
        #Gera combinações de curvas

        #expo
```

```

tipo_curva="expo"
fitagem_final=$energia_final #Fitagem vai para todo o conjunto de dados
sed s/numero_histo/"$numero_histo"/g
<histograma_base_uma_curva.C>histograma"$numero_histo".C
sed -i s/tipo_curva/"$tipo_curva"/g histograma"$numero_histo".C
sed -i s/fitagem_final/"$fitagem_final"/g histograma"$numero_histo".C
sed -i s/energia_nome/"$energia_nome"/g histograma"$numero_histo".C
sed -i s/energia_final/"$energia_final"/g histograma"$numero_histo".C
sed -i s/numero_de_barras/"$numero_de_barras"/g histograma"$numero_histo".C

#expo-gaus
for (( i=1 ; i<=2*numero_de_barras ; i++ )); do
    let "l= i*e/2"
    expo_i="1"
    expo_f=$l
    gaus_i=$l
    gaus_f=$energia_final
    fitagem_final=$energia_final #Fitagem vai para todo o conjunto de dados

    #Substitui valores
    let "numero_histo = numero_histo + 1"
    sed s/numero_histo/"$numero_histo"/g
<histograma_base.C>histograma"$numero_histo".C
sed -i s/expo_i/"$expo_i"/g histograma"$numero_histo".C
sed -i s/expo_f/"$expo_f"/g histograma"$numero_histo".C
sed -i s/kaus_i/"$kaus_i"/g histograma"$numero_histo".C
sed -i s/kaus_f/"$kaus_f"/g histograma"$numero_histo".C
sed -i s/fitagem_final/"$fitagem_final"/g histograma"$numero_histo".C
sed -i s/energia_nome/"$energia_nome"/g histograma"$numero_histo".C
sed -i s/energia_final/"$energia_final"/g histograma"$numero_histo".C
sed -i s/numero_de_barras/"$numero_de_barras"/g histograma"$numero_histo".C
done

#gaus-expo
for ((i=1; i<2*numero_de_barras ; i++))
do
    let "l= i*e/2"
    expo_i=$l
    expo_f=$energia_final
    gaus_i="1"
    gaus_f=$l
    fitagem_final=$energia_final #Fitagem vai para todo o conjunto de dados

    #Substitui valores
    sed s/numero_histo/"$numero_histo"/g
<histograma_base.C>histograma"$numero_histo".C
sed -i s/expo_i/"$expo_i"/g histograma"$numero_histo".C
sed -i s/expo_f/"$expo_f"/g histograma"$numero_histo".C
sed -i s/kaus_i/"$kaus_i"/g histograma"$numero_histo".C
sed -i s/kaus_f/"$kaus_f"/g histograma"$numero_histo".C
sed -i s/fitagem_final/"$fitagem_final"/g histograma"$numero_histo".C
sed -i s/energia_nome/"$energia_nome"/g histograma"$numero_histo".C
sed -i s/energia_final/"$energia_final"/g histograma"$numero_histo".C
sed -i s/numero_de_barras/"$numero_de_barras"/g histograma"$numero_histo".C
let "numero_histo = numero_histo + 1"
done

#gaus

```

```

        tipo_curva="gaus"
        fitagem_final=$energia_final #Fitagem vai para todo o conjunto de dados
        sed s/numero_histo/"$numero_histo"/g
<histograma_base_uma_curva.C>histograma"$numero_histo".C
        sed -i s/tipo_curva/"$tipo_curva"/g histograma"$numero_histo".C
        sed -i s/fitagem_final/"$fitagem_final"/g histograma"$numero_histo".C
        sed -i s/energia_nome/"$energia_nome"/g histograma"$numero_histo".C
        sed -i s/energia_final/"$energia_final"/g histograma"$numero_histo".C
        sed -i s/numero_de_barras/"$numero_de_barras"/g histograma"$numero_histo".C
    done
    let "numero_de_barras = numero_de_barras + 5"
    cd ..
done
exit 0

```

Anexo C

Código do arquivo de configuração histograma_base.C

```

//[n umero_histo] - numero serial do histograma, inteiro
//[e xpo_i] - inicio da exponencial, inteiro
//[e xpo_f] - final da exponencial. inteiro
//[f itagem_final] - final da fitagem, inteiro

{
TCanvas *c1 = new TCanvas("c1", "c1",8,8,699,499);
c1->Range(0.75625,-56.875,1.19375,11.875);
c1->SetBorderSize(2);
c1->SetFrameFillColor(10);
c1->SetFillColor(10);

c1->Modified();
c1->cd();

co = new TH1F("co","Distribuicao de fotoeletrons - energia_nome",numero_de_barras,1,energia_final);

co->GetXaxis()->SetTitle("Fotoeletrons");
co->GetYaxis()->SetTitle("Ocorrencias");
co->GetYaxis()->SetTitleOffset(1.2);
co->GetXaxis()->SetTitleOffset(1.1);
//gStyle->SetOptStat("nemr");
gStyle->SetOptFit(111);

FILE *arq;
FILE *saida;
FILE *saida2;
saida=fopen("dadosnumero_histo","w");
saida2=fopen("chinumero_histo","w");
double x;
int nd;

arq=fopen("histo_energia_nome","r");
if(arq!=NULL)
{
do
{
nd=fscanf(arq,"%lf",&x);
if(nd==1)

```

```

        {
            co.Fill(x);
        }
    }
    while(nd==1);
    co.Draw();
}

Double_t norm = co->GetEntries();
if (norm) co->Scale(1/norm);

double par[5];
g1  = new TF1("g1","expo",expo_i,expo_f);
g2  = new TF1("g2","gaus",gaus_i,gaus_f);
total = new TF1("total","expo(0)+gaus(2)",1,fitagem_final);

g1->SetLineColor(1);
g2->SetLineColor(1);
total->SetLineColor(2);

co->Fit(g1,"R");
co->Fit(g2,"R+");

g1->GetParameters(&par[0]);
g2->GetParameters(&par[2]);

total->SetParameters(par);
co->Fit(total,"R+");

total->GetParameters(&par[5]);

fprintf(saida,"constant expo = %lf\n",par[0]);
fprintf(saida,"slope expo = %lf\n",par[1]);
fprintf(saida,"constant gaus = %lf\n",par[2]);
fprintf(saida,"mean gaus = %lf\n",par[3]);
fprintf(saida,"sigma gaus = %lf\n",par[4]);

c1->Print("histonumero_histo.eps");

double chi2;
chi2 = total->GetChisquare();
fprintf(saida2,"%lf",chi2);

fclose(arq);
fclose(saida);
fclose(saida2);

c1->Close();
delete co;
delete g1;
delete g2;
delete total;
delete c1;
}

```

Anexo D

Código do arquivo de configuração histograma_base_uma_curva.C

```
//[n umero_histo] - numero serial do histograma, inteiro
//[e xpo_i] - inicio da exponencial, inteiro
//[e xpo_f] - final da exponencial. inteiro
//[f inal] - final da fitagem, inteiro

{
TCanvas *c1 = new TCanvas("c1", "c1",8,8,699,499);
c1->Range(0.75625,-56.875,1.19375,11.875);
c1->SetBorderSize(2);
c1->SetFrameFillColor(10);
c1->SetFillColor(10);

c1->Modified();
c1->cd();

co = new TH1F("co","Distribuicao de fotoeletrons - energia_nome",numero_de_barras,1,energia_final);

co->GetXaxis()->SetTitle("Fotoeletrons");
co->GetYaxis()->SetTitle("Ocorrencias");
co->GetYaxis()->SetTitleOffset(1.2);
co->GetXaxis()->SetTitleOffset(1.1);
//gStyle->SetOptStat("nemr");
gStyle->SetOptFit(111);

FILE *arq;
FILE *saida;
FILE *saida2;
saida=fopen("dadosnumero_histo","w");
saida2=fopen("chinumero_histo","w");
double x;
int nd;

arq=fopen("histo_energia_nome","r");
if(arq!=NULL)
{
do
{
nd=fscanf(arq,"%lf",&x);
if(nd==1)
{
co.Fill(x);
}
}
while(nd==1);
co.Draw();
}

Double_t norm = co->GetEntries();
if (norm) co->Scale(1/norm);

double par[6];
g1 = new TF1("g1","tipo_curva",1,fitagem_final);
total = new TF1("total","tipo_curva(0)",1,fitagem_final);
```

```

g1->SetLineColor(1);
total->SetLineColor(2);

co->Fit(g1,"R");

g1->GetParameters(&par[0]);

total->SetParameters(par);
co->Fit(total,"R+");

total->GetParameters(&par[3]);

fprintf(saida,"constant expo = %lf\n",par[0]);
fprintf(saida,"slope expo = %lf\n",par[1]);
fprintf(saida,"constant gaus = %lf\n",par[2]);
fprintf(saida,"mean gaus = %lf\n",par[3]);
fprintf(saida,"sigma gaus = %lf\n",par[4]);

c1->Print("histonumero_histo.eps");

double chi2;
chi2 = total->GetChisquare();
fprintf(saida2,"%lf",chi2);

fclose(arq);
fclose(saida);
fclose(saida2);

c1->Close();
delete co;
delete g1;
delete total;
delete c1;
}

```

Anexo E

Código do script Analisador

```

#!/bin/bash
#!/bin/bash
#Joao PN - joaoxp-at-gmail.com

#Espaço para código que fornece faixa de valores dos dados
#e: comprimento das barras; n:número de barras do histograma

#Código para achar a energia máxima da plotagem
energia_final="0"
exec 3< histo_$(energia_nome)
while read linha <&3; do
    if [ $(energia_final -lt $linha) ]; then
        energia_final=$linha
    fi
done

echo "{ " > pos-analise.C

numero_de_barras="20"
while [ $numero_de_barras -lt $limite_de_barras ]; do

```

```

let "ndois = 4*numero_de_barras"
cd analise$numero_de_barras #Vai para o directorio dos dados

#Gera macro de análise e analisa os histogramas
echo "{" > analisador-root.C
for(( i=0 ; i<ndois ; i++ )); do
    echo 'gROOT->ProcessLine(".x histograma_numero.C");' >> analisador-root.C
    sed -i s/_numero/"$i"/g analisador-root.C
done
echo "}" >> analisador-root.C
root -l -b -q -n analisador-root.C

#Faz a análise do Chi^2
chi2_minimo="1"
numero_chi_minimo="0"
chi_erro="0.000000"
for(( i=0 ; i<"$ndois" ; i++ )); do
    chi2_teste=`cat chi$i`
    if [ "$chi2_minimo" \> "$chi2_teste" ] && [ "$chi2_teste" != "$chi_erro" ]; then
        chi2_minimo=$chi2_teste
        numero_chi_minimo=$i
    fi
done

cd ..

echo 'gROOT->ProcessLine(".x FINALHISTOGRAMA_numero_de_barras.C");' >> pos-analise.C
sed -i s/numero_de_barras/"$numero_de_barras"/g pos-analise.C

cp analise$numero_de_barras/histograma$numero_chi_minimo.C FINALHISTOGRAMA_$numero_de_barras.C
# cp analise$numero_de_barras/dados$numero_chi_minimo FINALDADOS_$numero_de_barras.C
# cp analise$numero_de_barras/chi$numero_chi_minimo FINALCHI_$numero_de_barras.C
# cp analise$numero_de_barras/histo$numero_chi_minimo.eps FINALGRAPH_$numero_de_barras.eps
sed -i s/dados$numero_chi_minimo/FINALDADOS_$numero_de_barras/g
FINALHISTOGRAMA_$numero_de_barras.C
sed -i s/histo$numero_chi_minimo.eps/FINALGRAPH_$numero_de_barras/g
FINALHISTOGRAMA_$numero_de_barras.C
sed -i s/chi$numero_chi_minimo/FINALCHI_$numero_de_barras/g
FINALHISTOGRAMA_$numero_de_barras.C
# root -l -b -q FINALHISTOGRAMA_$numero_de_barras.C
let "numero_de_barras = numero_de_barras + 5"
done

echo "}" >> pos-analise.C

exit 0

```

Anexo F

Código do script Checador

```

#!/bin/bash
#Joao PN - joaoxp-at-gmail.com

#Checa se a analise teve sucesso, e remove analise com erro
mkdir Erros
exec 3< nomes_histogramas
while read linha <&3; do
    cd Resultados/$linha

```

```

        echo Testando $linha
        if [ -s FINALCHI_20 ] && [ -s FINALCHI_25 ] && [ -s FINALCHI_30 ] && [ -s FINALDADOS_20 ] && [ -s
FINALDADOS_25 ] && [ -s FINALDADOS_30 ] && [ -s FINALGRAPH_20 ] && [ -s FINALGRAPH_25 ] && [ -s
FINALGRAPH_30 ]; then
            echo $linha OK
            cd ..
            cd ..
        else
            cd ..
            cd ..

            mv Histogramas/histo_$linha Erros/
            rm -fR Analise_$linha
            mv Resultados/$linha Erros/
            mv Histogramas/saida_teste_histo_$linha Erros/
            sed -i /$linha/d nomes_histogramas
            echo $linha com erro
            exit
        fi

done

```

Anexo G

Código do script Finalizador

```

#!/bin/bash

#Limpa arquivos de analise
rm -fR Analise_*

#Seleciona o plot com menor chi^2 entre 20, 25 e 30 barras
exec 3< nomes_histogramas
while read linha <&3; do
    cd Resultados/$linha
    numero_chi=20
    chi2=`cat FINALCHI_20`
    chi2_teste=`cat FINALCHI_25`

    if [ "$chi2" > "$chi2_teste" ]; then
        numero_chi=25
        chi2=$chi2_teste
    fi

    chi2_teste=`cat FINALCHI_30`

    if [ "$chi2" > "$chi2_teste" ]; then
        numero_chi=30
        chi2=$chi2_teste
    fi

    cd ..

    cp $linha/FINALDADOS_$numero_chi DADOS_$linha
    cp $linha/FINALGRAPH_$numero_chi GRAPH_$linha
    cp $linha/FINALCHI_$numero_chi CHI_$linha
    cp $linha/FINALHISTOGRAMA_$numero_chi.C HISTO_$linha

    rm -fR $linha

```



```
cd ..
done
```

Anexo H

Código do script Removedor

```
#!/bin/bash
printf "De o conjunto de dados a ser removido (formato: 10MeV): "
read linha

rm -f Resultados/CHI_$linha
rm -f Resultados/DADOS_$linha
rm -f Resultados/GRAPH_$linha
rm -f Resultados/CHI_$linha
rm -f Resultados/DADOS_$linha
mv Histogramas/histo_$linha Erros/
mv Histogramas/saida_teste_histo_$linha Erros/
sed -i /$linha/d nomes_histogramas
```

Anexo I

Código do script ZeroFotons

```
#!/bin/bash

##Faz a analise para zero fotoeletrons
mkdir Resultados/Zero_fotons
#Faz uma copia dos histogramas sem os dados "0"
cp Histogramas/histo_* Resultados/Zero_fotons/
cp nomes_histogramas Resultados/Zero_fotons
cd Resultados/Zero_fotons
exec 3< nomes_histogramas
while read linha <&3; do
    sed -i s/0// histo_$linha
    grep -v "^$" histo_$linha > temp_histo_$linha
    rm -f histo_$linha
    mv temp_histo_$linha histo_$linha
done
cd ..
cd ..

#Retira do saida_teste_histo a quantidade de simulacoes da particula

exec 3< nomes_histogramas
while read linha <&3; do
    count=0
    for (( i=0 ; i<=3100 ; i++ )); do
        if grep ">>> Event $i" Histogramas/saida_teste_histo_$linha; then
            count=$((count+1))
        fi
    done
    let 'count = count + 1'
    count2=`grep -c '$' Resultados/Zero_fotons/histo_$linha`
    echo $(echo "scale=4; $count2/$count" | bc) >> Resultados/Zero_fotons/FRACAO_ZERO
done

#Adiciona, com probabilidade 0, as energias que deram erro
DIR="Erros"
if [ "$(ls -A $DIR)" ]; then
    cp nomes_histogramas Erros/ENERGIAS_tmp
    cd Erros
    echo histo_* >> ENERGIAS_tmp
```

```

sed -i s/histo_//g ENERGIAS_tmp
tr ' ' '\n' < ENERGIAS_tmp >ENERGIAS

A=$(ls -l | wc -l)
let "B = A / 2"
Bvar=0
cd ..
while [ $Bvar -lt $B ]; do
    echo .0000 >> Resultados/Zero_fotons/FRACAO_ZERO
    let "Bvar = Bvar + 1"
done

mv Erros/ENERGIAS Resultados/Zero_fotons/ENERGIAS

else
    cp nomes_histogramas Resultados/Zero_fotons/ENERGIAS
fi

#Lista as energias das simulacoes
cp script-gnuplot Resultados/Zero_fotons/
cd Resultados/Zero_fotons
sed -i s/MeV//g ENERGIAS
sed -i s/GeV/000/g ENERGIAS
sed -i s/TeV/000000/g ENERGIAS

paste ENERGIAS FRACAO_ZERO > dados_zerofotons.dat
gnuplot script-gnuplot
cd ..
cp Zero_fotons/parametros_zero.dat parametros_zero.dat
cp Zero_fotons/zero_fotons.eps zero_fotons.eps

```

Anexo J

Código do script Parametros

```

#!/bin/bash

#Prepara os dados para analise
mkdir Resultados/Parametros
cp gnuplot-parametros Resultados/Parametros/
exec 3< nomes_histogramas
while read linha <&3; do
    cp Resultados/DADOS_$linha Resultados/Parametros/
    cd Resultados/Parametros
    sed -i s/'constant expo = '// DADOS_$linha
    sed -i s/'slope expo = '// DADOS_$linha
    sed -i s/'constant gaus = '// DADOS_$linha
    sed -i s/'mean gaus = '// DADOS_$linha
    sed -i s/'sigma gaus = '// DADOS_$linha
    cd ..
done
cp nomes_histogramas Resultados/Parametros/
cd Resultados/Parametros/

#Separa os dados de cada parametro em arquivos diferentes
exec 3< nomes_histogramas
while read linha <&3; do
    head -1 DADOS_$linha >> constant_expo

```

```

sed -i '1d' DADOS_$linha
head -1 DADOS_$linha >> slope_expo
sed -i '1d' DADOS_$linha
head -1 DADOS_$linha >> constant_gaus
sed -i '1d' DADOS_$linha
head -1 DADOS_$linha >> mean_gaus
sed -i '1d' DADOS_$linha
head -1 DADOS_$linha >> sigma_gaus
rm -f DADOS_$linha

done

#Cria o arquivo de dados de energia
cp nomes_histogramas ENERGIAS
sed -i s/MeV//g ENERGIAS
sed -i s/GeV/000/g ENERGIAS
sed -i s/TeV/000000/g ENERGIAS

#Cria os arquivos para serem analisados
paste ENERGIAS constant_expo > dados_constant_expo.dat
paste ENERGIAS slope_expo > dados_slope_expo.dat
paste ENERGIAS constant_gaus > dados_constant_gaus.dat
paste ENERGIAS mean_gaus > dados_mean_gaus.dat
paste ENERGIAS sigma_gaus > dados_sigma_gaus.dat

cp gnuplot-parametros gnuplot-constant_expo
cp gnuplot-parametros gnuplot-slope_expo
cp gnuplot-parametros gnuplot-constant_gaus
cp gnuplot-parametros gnuplot-mean_gaus
cp gnuplot-parametros gnuplot-sigma_gaus

sed -i s/var_parametro/constant_expo/g gnuplot-constant_expo
sed -i s/var_parametro/slope_expo/g gnuplot-slope_expo
sed -i s/var_parametro/constant_gaus/g gnuplot-constant_gaus
sed -i s/var_parametro/mean_gaus/g gnuplot-mean_gaus
sed -i s/var_parametro/sigma_gaus/g gnuplot-sigma_gaus

gnuplot gnuplot-constant_expo
gnuplot gnuplot-slope_expo
gnuplot gnuplot-constant_gaus
gnuplot gnuplot-mean_gaus
gnuplot gnuplot-sigma_gaus

```

Anexo K

Código do arquivo de configuração script-gnuplot

```

set term postscript eps enhanced color 20
set encoding iso_8859_1
set title '{/=36 Probabilidade de deposito de energia}'
set xlabel "{/=32 Energia (MeV)}"
set ylabel "{/=32 Probabilidade}";
set output "zero_fotons.eps"
f(x)=(a*(x**b)+c*(x**d)+e*(x**f)+g*(x**h)+i*x+j)
set logscale x
set fit logfile "log-dados"
fit f(x) 'dados_zerofotons.dat' via a,b,c,d,e,f,g,h,i,j
set size 1.5,1.5
plot [0.5:20000000] [0:1] 'dados_zerofotons.dat' lw 20 t '{/=32 Resultados da simula{\347}{\343}o}', f(x) lw 5 t '{/=32 10 Parametros}'

```

```

set print 'parametros_zero.dat'
print a
print b
print c
print d
print e
print f
print g
print h
print i
print j

```

Anexo L

Código do arquivo de configuração script-parametros

```

set term postscript eps enhanced color 20
set encoding iso_8859_1
set title '{/=36 Evolução do parâmetro var_parametro}'
set xlabel "{/=32 Energia (MeV)}"
set ylabel "{/=32 Probabilidade}";
set output "var_parametro.dat.eps"
f(x)=(a*(x**b)+c*(x**d)+e*(x**f)+g*(x**h)+i*x+j)
set logscale x
set fit logfile "log-dados"
fit f(x) 'dados_var_parametro.dat' via a,b,c,d,e,f,g,h,i,j
set size 1.5,1.5
plot [0.5:20000000] [-3:3] 'dados_var_parametro.dat' lw 20 t '{/=32 Resultados da simula{\347}{\343}o}', f(x) lw 5 t '{/=32
10 Parametros}'
set print 'parametros_var_parametro.dat'
print a
print b
print c
print d
print e
print f
print g
print h
print i
print j

```