Projeto de Física Computacional Simulação de um chuveiro electromagnético produzido por um fotão

Ana Santos 84364 e Inês Vieira 84406

Professor Fernando Barão

Instituto Superior Técnico 31 Dez, 2016

- Estrutura do código
 - Material.h
 - Particles.h
 - Propagator.h
 - Funcoes.h
 - propagate
- 2 Ideia das escalas
 - Produção de um par electrão-positrão
 - Radiação de energia pelo electrão ou positrão (bremsstrahlung)
 - Aniquilação do positrão
- 3 Amostragem das variáveis Aleatórias
 - Produção de um par electrão-positrão
 - Radiação de energia pelo electrão ou positrão (bremsstrahlung)
 - Aniquilação do positrão
- Perda de Energia
- Sesultados



Material.h

```
#ifndef __material_
#define __material_
#include<string>
using namespace std;
struct material{
double Z;
double ro;
double A;
double C0;
double U0;
double U1;
double a;
double m;
string name;
};
#endif
```

Figura: estrutura Material.h

Particle.h

```
class Particle{
 public:
  Particle();
  Particle(string.double.TVector3.TVector3):
  Particle(const Particle&):
  const Particle operator=(const Particle&);
  ~Particle(){;}
  void SetEnergy(double):
  void SetVivo(int):
  void SetMomentum(TVector3);
  void SetInitialP(TVector3);
  void SetFinalP(TVector3):
  void SetName(string):
  double GetEnergy();
  int GetVivo():
  TVector3 GetMomentum():
  TVector3 GetInitialP():
  TVector3 GetFinalP();
  string GetName():
  double& E():
  TVector3& p();
  TVector3& Xi();
  TVector3& Xf():
  string& Name():
  void Print();
  double m():
```

Figura: Metodos publicos da classe Particle

Particle.h

```
protected:
  double En;//energia
  TVector3 mo;//momento linear
  TVector3 xi;//posicao inicial
  TVector3 xf;//posicao final
  string N;
  int vivo;
```

Figura: Metodos privados da classe Particle

Propagator.h

```
class Propagator{
public:
    Propagator(){flag=1;};
    Propagator(){flag=1;};
    Propagator(){f;};
    Particles GetParticle();
    int Getflag();
    vector-Particle> PropagatePH();//fotao
    vector-Particle> PropagatePO();//fotao
    vector-Particle> PropagatePI();//eletrao
```

Figura: Metodos publicos da classe Propagator

Propagator.h

```
private:
material Al:
Particle P:
int flag;
///Criacao de pares
double ParR ():
void ParE(double Eq. double& Eele. double& Epos);
void ParAng(double Epart, double& anga, double& angb);
///Brem
double BremR(double E):
void BremE(double E, double& Efotao, double& Enovo):
void BremAng(double E, double& angf, double& azi);
//Aniquilacao de positrao
double AnPR():
double AnPu();
void AnPE(double u, double& E1, double& E2);
void AnPang(double u. double& teta1.double &teta2.double &fi1.double &fi2):
//perda de energia
void PE(double l)://muda a energia da particula do construtor se já estiver definida a xf
//Auxi
TVector3 MC(TVector3 S. TVector3 PS):
```

Figura: Metodos privados da classe Propagator

Funcoes.h

```
#ifndef __Funcoes__
#define __Funcoes__
#include"Particle.h"
#include"Material.h"
#include<vector>
class Funcoes{
public:
       Funcoes(vector<Particle>, vector<Particle>, vector<Particle>);
       ~Funcoes(){:}
       void Desenhar();
       void Hist(double):
       void Ficheiros();
private:
vector<Particle> vPH:
vector<Particle> vPO;
 vector<Particle> vEL:
#endif
```

Figura: Metodos publicos da classe Funcoes

propagate

```
vector<Particle> vPH:
vector<Particle> vPO;
vector<Particle> vEL:
void propagate(Particle P, material Al){
 if(P.Name()=="Photon"){
    Propagator prob(P,Al);
    vector<Particle> V1=prob.PropagatePH();
    Particle Paux=prob.GetParticle():
   lf(V1[0].E()>50e-6){
      propagate(V1[0],Al);
    if(V1[1].E()>50e-6){
      propagate(V1[1],Al);
    vPH.push_back(Paux);
  else if(P.Name()=="Electron"){
   Propagator prob(P,Al);
    vector<Particle> V2=prob.PropagateEL():
    Particle Paux=prob.GetParticle():
    for(int i=0;i<V2.size();++i){
     tf(V2[t],E()>5){
        propagate(V2[i],Al);
    vEL.push back(Paux);
  else if(P.Name()=="Positron"){
   Propagator prob(P.Al):
    vector<Particle> V3=prob.PropagatePO();
    Particle Paux=prob.GetParticle():
    for(int i=0;i<V3.size();++i){
     tf(V3[1].E()>5){
        propagate(V3[i],Al);
    vPO.push_back(Paux);
```

Figura: Função propagate

Produção de um par electrão-positrão

$$Probabilidade = \frac{7\rho}{9X_0} \tag{1}$$

$$X_0 = \frac{A}{4\alpha N_A Z^2 re^2 log(183Z^{\frac{1}{3}})}$$
 (2)

Probabilidade é uma constante igual a 0.0956476. λ_{int} é 10.4550

Radiação de energia pelo electrão ou positrão (bremsstrahlung)

A probabilidade de interação em função da Energia:

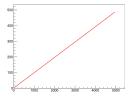


Figura: Probabildade de Br em função da energia

probabilidade para 10 MeV=2.29109 $\lambda_{int}=0.4365$ probabilidade para 100 MeV=11.1627 $\lambda_{int}=0.08958$ probabilidade para 5000 MeV=491.391 $\lambda_{int}=0.002035$

Aniquilação do positrão

A probabilidade de interação em função da Energia:

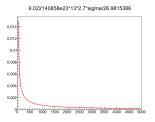


Figura: probabilidade em função da energia

```
probabilidade para 10 MeV=0.03 \lambda_{int}=33 probabilidade para 100 MeV=0.00502328 \lambda_{int}=199 probabilidade para 5000 MeV=0.00017744 \lambda_{int}=5635
```

Produção de um par electrão-positrão - Comprimento de interação

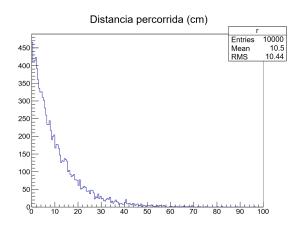


Figura: Comprimento de Interação

Produção de um par electrão-positrão - Energia

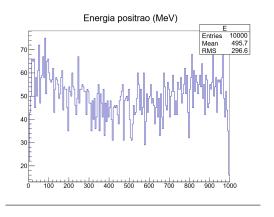


Figura: Energia Positrão para 1GeV

Produção de um par electrão-positrão - Energia

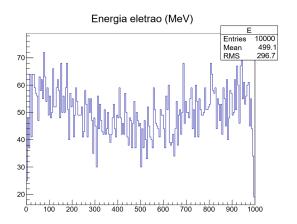


Figura: Energia Eletrão para 1GeV



Produção de um par electrão-positrão - Angulos

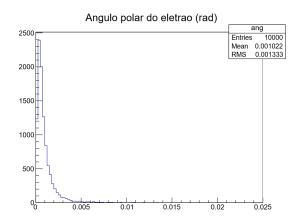


Figura: Angulo Positrão

Produção de um par electrão-positrão - Angulos

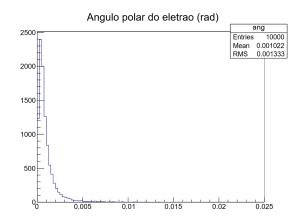


Figura: Angulo Electrão

Radiação de energia pelo electrão ou positrão- Comprimento de interação

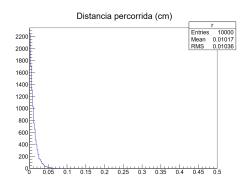


Figura: Comprimento de Interação

Radiação de energia pelo electrão ou positrão- Energia

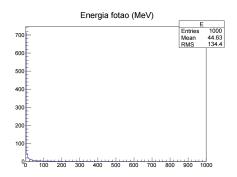


Figura: Energia do fotão

Radiação de energia pelo electrão ou positrão-Energia-função

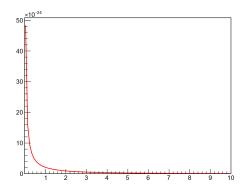


Figura: Função da distribuição

Radiação de energia pelo electrão ou positrão- Angulo

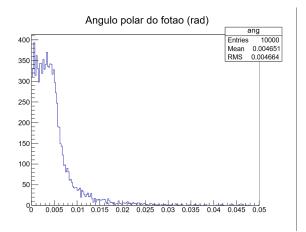


Figura: Angulo Polar



Aniquilação do positrão - Comprimento de interação

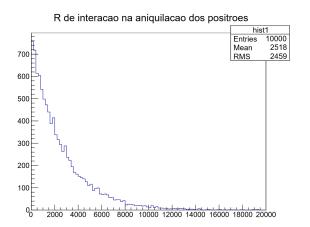


Figura: Comprimento de Interação

Aniquilação do positrão - u



Figura: u

Aniquilação do positrão - f(u)

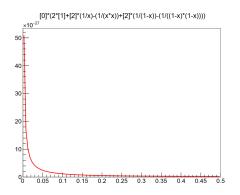


Figura: Função da distribuição

Perda de Energia

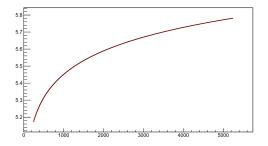


Figura: Gráfico da dE/dx em função da energia inicial

1GeV - Desenho xz

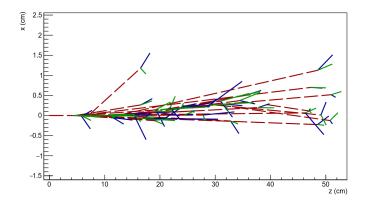


Figura: Desenho do Perfil longitudinal da cascata em x e z

1GeV - Desenho yz

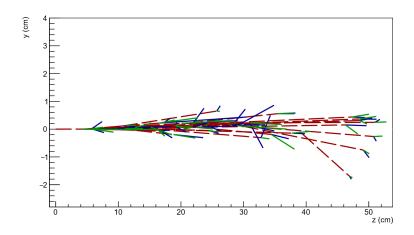


Figura: Desenho do Perfil longitudinal da cascata em y e z



1GeV - Desenho em 3D

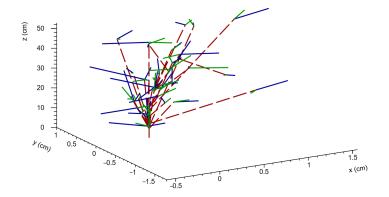


Figura: Desenho do Perfil longitudinal



1Gev - Histogramas

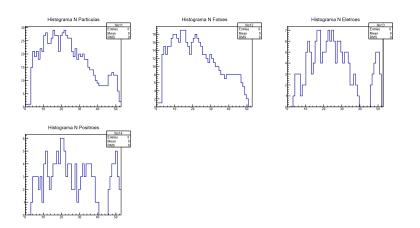


Figura: Histogramas

1Gev - Numero Maximo de Particulas

$$N_{max}$$
esperado = 23 $-N_{max}$ calculado = 27 $-Z$ = 14
Outros:
 N_{max} esperado = 23 $-N_{max}$ calculado = 28 $-Z$ = 27
 N_{max} esperado = 23 $-N_{max}$ calculado = 35 $-Z$ = 36
 N_{max} esperado = 23 $-N_{max}$ calculado = 30 $-Z$ = 20

2GeV - Desenho xz

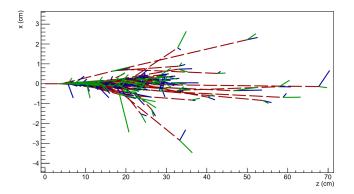


Figura: Desenho do Perfil longitudinal da cascata em x e z

Índice Estrutura do código Ideia das escalas Amostragem das variáveis Aleatórias Perda de Energia Resultados

2GeV - Desenho yz

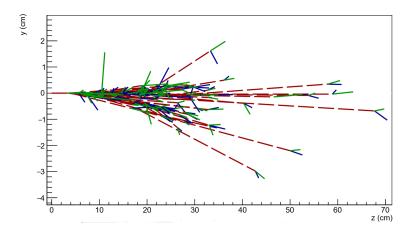


Figura: Desenho do Perfil longitudinal da cascata em y e z



2GeV - Desenho em 3D

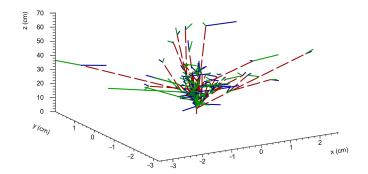


Figura: Desenho do Perfil longitudinal

2Gev - Histogramas

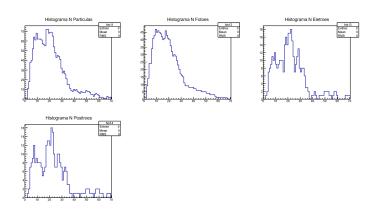


Figura: Histogramas

Índice Estrutura do código Ideia das escalas Amostragem das variáveis Aleatórias Perda de Energia Resultados

5GeV - Desenho xz

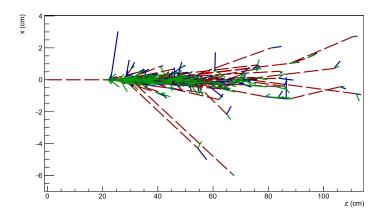


Figura: Desenho do Perfil longitudinal da cascata em x e z

Índice Estrutura do código Ideia das escalas Amostragem das variáveis Aleatórias Perda de Energia Resultados

5GeV - Desenho yz

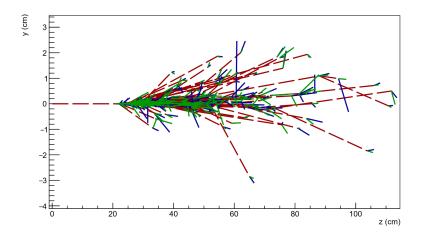


Figura: Desenho do Perfil longitudinal da cascata em y e z



5GeV - Desenho em 3D

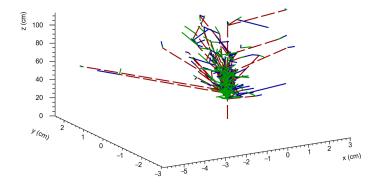


Figura: Desenho do Perfil longitudinal

5Gev - Histogramas

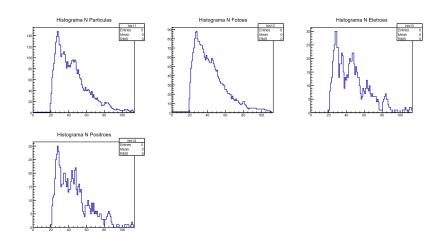


Figura: Histogramas

