Introdução à Análise de dados em FAE **(07/05/2024)**

Lista Monte Carlo

Professores: Eliza, Maurico, Sandro *Name:* Pedro Henrique de Castro Moura

EXERCICIO 1

```

{


int d=2;
int l=1;
TCanvas *c1 = new TCanvas("c1","c1" ,500 ,500);
c1->Divide(4, 3);

TNtuple *t1 = new TNtuple("t1", "t1", "x:theta");
TF1 *sineFunc = new TF1("sineFunc", Form("(%.2i*0.5)*TMath::Sin(x)", l), 0, M_PI);

TRandom rand;
// Gera n meros aleat rios de 0 a 1:

double random_number;
double random_angle;


float x;
float x_aim;
int N =1000;

std::vector<float> valores = {10, 50, 100, 1000, 10000, 2e6};
int k=1;

for (int i=0; i < valores.size(); i++){

    N = valores[i];
    // cout << N << " " <<i<< endl ;
    for (int i=0; i<N; i++){

    random_number = rand.Uniform();
    x = l*random_number;
    random_number = rand.Uniform();
    random_angle = 0 + (M_PI - 0)*random_number;

    // x_aim =(l/2) *( sin( random_number ));

    t1->Fill(x, random_angle);

    }

    c1->cd(k);
    t1->Draw("x: theta", " ", " ");
    sineFunc->Draw("same");


    //t1 ->Draw(" theta " ,"" ,"");
    // sineFunc -> Draw ("") ;
```

```
51
52      std::string message = Form("N=%d", N);
53
54      TNtuple *t2 = new TNtuple("message.c_str()","t2","x:theta");
55
56      for (int i = 0; i < t1->GetEntries(); i++){
57          t1->GetEntry(i);
58          double x = t1->GetArgs()[0]; //  valor de x do TNtuple
59          double theta = t1->GetArgs()[1]; //  valor de theta do TNtuple
60
61          if (sineFunc->Eval(theta)>=x) {
62              t2->Fill(x, theta); // Adiciona o ponto ao TNtuple t2
63              }
64
65          }
66
67   // Plotar os pontos pontos dentro de TF1
68      k++;
69      c1->cd(k);
70
71      t2->Draw ("x: theta ", "", "");
72      int m;
73
74      m=t2->GetEntries();
75
76      double pi=(2.0*N/m)*(float(l)/d);
77
78      float I;
79
80      I=d*M_PI*(float(m)/N);
81
82      cout<<Form("O valor de pi para N= %.2i: ", N)<<pi<<endl;
83      cout<<"Area efetiva I:"<<I<<endl;
84      k++;
85      }
86  }
```

Output:
O valor de π para N= 10: 3.33333
Area efetiva I : 1.88496
O valor de π para $N = 50$: 2.77778
Area efetiva I : 2.26195
O valor de π para $N = 100$: 1.96078
Area efetiva I : 3.20442
O valor de π para $N = 1000$: 2.57069
Area efetiva I : 2.44416
O valor de π para $N = 10000$: 2.78784
Area efetiva I : 2.25378
O valor de π para $N = 2000000$: 3.12259
Area efetiva I : 2.01217

EXERCICIO 3

```
1   {
2
3
4   double differentialCrossSection(double E, double theta, double m, double alpha){
5   double r0 = (alpha*alpha)/(2*m*m);
6   double E_prime = (E)/(1 + (E/m)*(1 - std::cos(theta)));
7   double dOmega = std::sin(theta);
8   return (r0*r0)*std::pow((E_prime/E), 2)*((E_prime/E)+(E/E_prime ) - dOmega*dOmega);
```

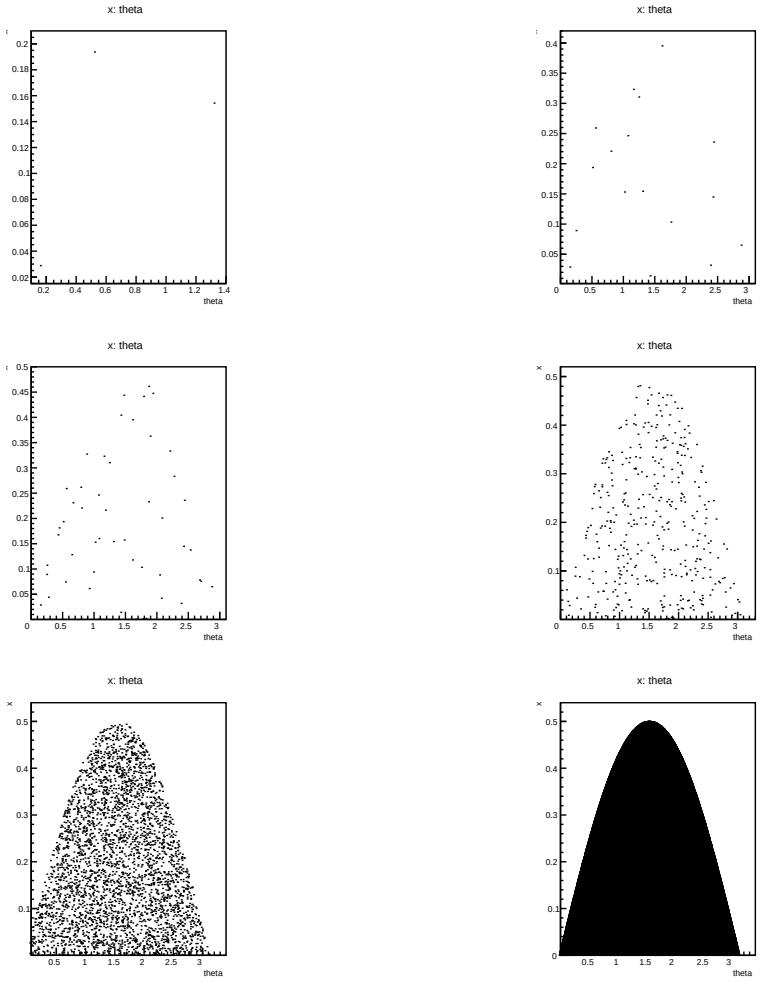

Figura 1: Exercício 1 e 2

```
9
10 }
11
12 int N = 100000;
13 double m = 0.5;
14 double alpha = 0.0072992701;
15
16 TCanvas *c1 = new TCanvas("c1","c1" , 400, 700);
17 c1->Divide(3, 1);
18
19 double E = 1.0; // Energia do f ton incidente (1 MeV)
20 // double E_prime = 9.0 e5;  // Energia do f ton espalhado (900 keV)
21
22 TNtuple *t1= new TNtuple("t1","t1","x");
23 TNtuple *t2= new TNtuple("t2","t2","x_aim:theta");
24 double random_number;
25 double x, x_aim, theta;
26 TRandom rand;
27 double max_value = 0;
28 float aux = 0;
29
30 for(float i=0; i<M_PI; i=i +0.01) {
31     if(i ==0) {
32     max_value = differentialCrossSection(E, i, m, alpha);
33     }
34     else if(max_value < differentialCrossSection(E, i, m, alpha)){
35     max_value = differentialCrossSection(E, i, m, alpha);
36     aux = i;
37 }
38 }
39
40     for(int j=0; j<N;j++) {
41         random_number = rand.Uniform();
42         x = max_value*random_number;
43         random_number = rand.Uniform();
44         theta = M_PI*random_number;
45         if(differentialCrossSection(E, theta, m, alpha) > x){
46             x_aim = x;
47             t2->Fill(x_aim, theta);
48         }
49
50 t1->Fill(x);
51 }
52
53 c1->cd(1);
54 t1->Draw("x>>h2","","");
55 h2->GetXaxis()->SetTitle("d# Sigma /d# Omega");
56 h2->SetTitle(" Area Total ");
57
58 c1->cd(2);
59 t2->Draw("theta>>h1","","");
60 h1->SetTitle("Area efetiva");
61 h1->GetXaxis()->SetTitle("# Theta");
62
63 c1->cd(3);
64 t2->Draw("x_aim >>h3","","");
65 h3->SetTitle("Area efetiva");
66 h3->GetXaxis()->SetTitle("d# Sigma /d# Omega");
67
68
69 }
```

EXERCICIO 4

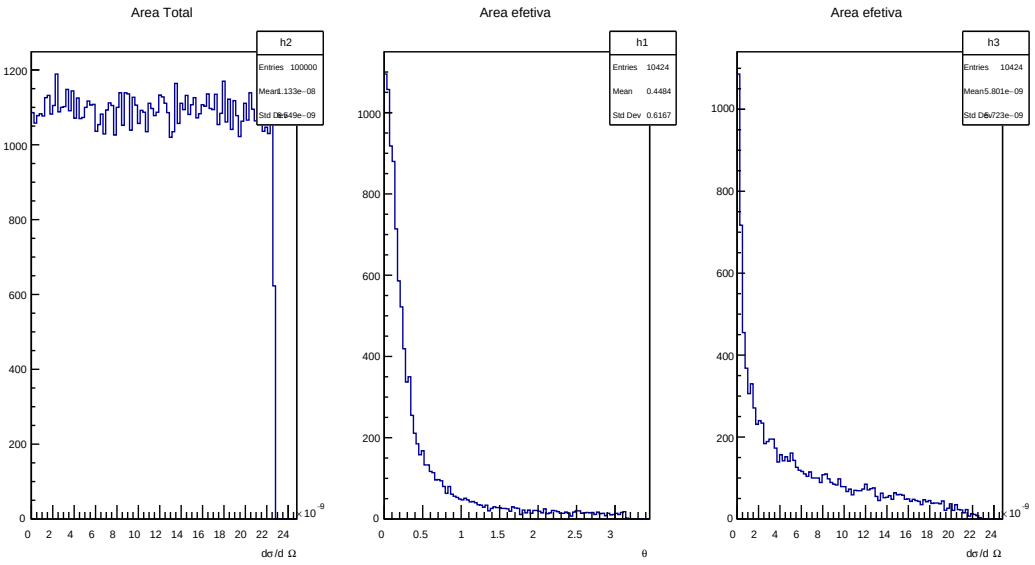

Figura 2: Exercício 3: $E = 1MeV$

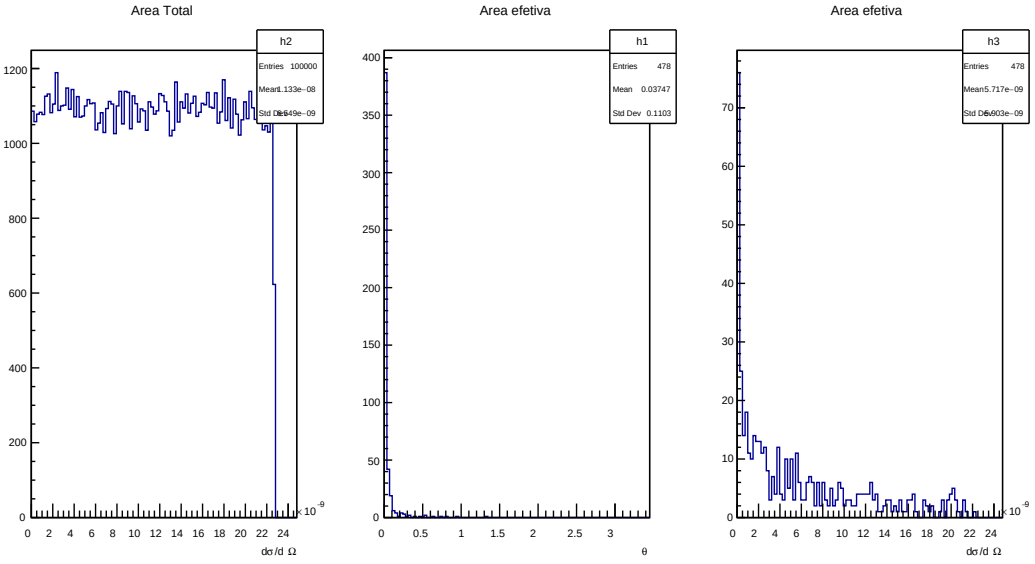

Figura 3: Exercício 3: $E = 5000MeV$

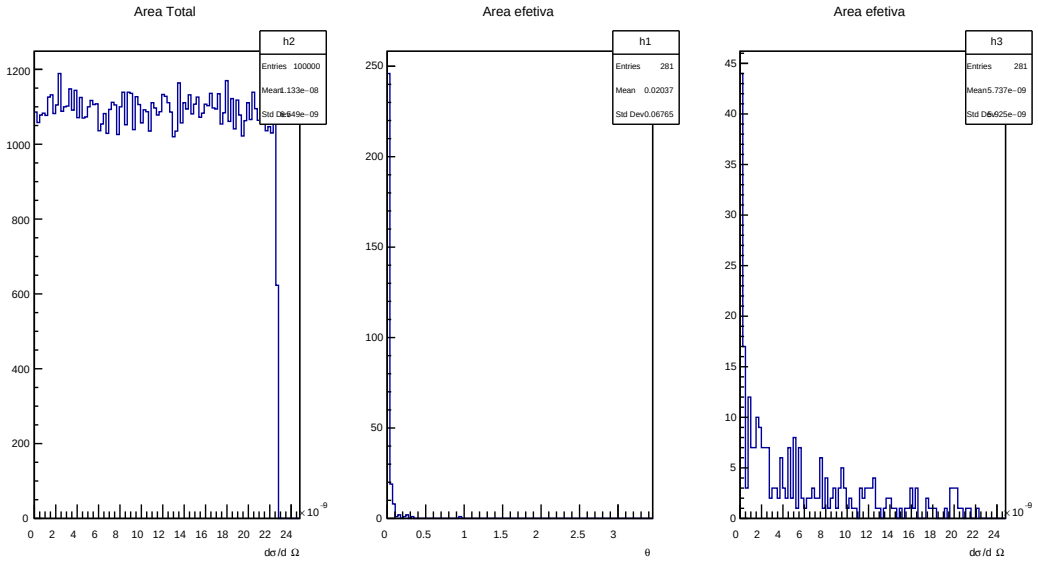

Figura 4: Exercício 3: $E = 15000MeV$

```python
import pythia8
from ROOT import TFile, TTree, vector

pythia = pythia8.Pythia()

pythia.readString("Beams:eCM = 13000.") # set center of mass energy
pythia.readString("HardQCD:all = on")    # Enable hard QCD processes

# Init ROOT ntuple

f = TFile("output.root", "RECREATE")
tree = TTree("tree", "Pythia Events")
n_particles = vector("float")()
n_eta = vector("float")()
n_phi = vector("float")()

# Branch definition
tree.Branch("Particle_pt", n_particles)
tree.Branch("Particle_eta", n_eta)
tree.Branch("Particle_phi", n_phi)

# Init Pythia generator
pythia.init()

# Event loop:
for iEvent in range(100):
    if not pythia.next():
        continue
    n_particles.clear()
    n_eta.clear()
    n_phi.clear()
    for particle in pythia.event:
        if particle.isFinal() and particle.isVisible():
            n_particles.push_back(particle.pT())
            n_eta.push.back(particle.eta())
            n_phi.push.back(particle.phi())

```

```
38      tree.Fill()
39
40  # Write and close ROOT file
41  f.Write()
42  f.Close
```