

Санкт-Петербургский политехнический университет Петра Великого
Физико-механический институт
Высшая школа фундаментальных физических исследований

Лабораторная работа

Генератор распада каона
По дисциплине "Специальный практикум"

Выполнил
студент гр. 5040302/10301

С. А. Буланова

Научный руководитель:
д.ф.-м.н.

Я. А. Бердников

«_____» _____ 2021 г.

Санкт-Петербург
2021

Задание

Дан пучок каонов с энергией 5 ГэВ. Масса каона 0.498 ГэВ. Смоделировать распад каонов на два π -мезона, рассчитать спектр инвариантных масс. Размыть импульсы по Гауссу, построить инвариантную массу в зависимости от поперечного импульса.

Приведем код:

```
1  //root macros
2  //include "TTree.h"
3  void generator(){
4      TFile fOut("kaon.root", "RECREATE");
5      TTree tree ("kaon", "kaon");
6      Double_t p1_x, p1_y, p1_z, p1_e;
7      tree.Branch("p1_x", &p1_x, "p1_x/D");
8      tree.Branch("p1_y", &p1_y, "p1_y/D");
9      tree.Branch("p1_z", &p1_z, "p1_z/D");
10     tree.Branch("p1_e", &p1_e, "p1_e/D");
11
12     Double_t p2_x, p2_y, p2_z, p2_e;
13     tree.Branch("p2_x", &p2_x, "p2_x/D");
14     tree.Branch("p2_y", &p2_y, "p2_y/D");
15     tree.Branch("p2_z", &p2_z, "p2_z/D");
16     tree.Branch("p2_e", &p2_e, "p2_e/D");
17
18     Double_t invmass;
19     tree.Branch("invmassa", &invmass, "invmass/D");
20
21     Double_t m_kaon = 0.498; //GeV
22     Double_t m_pion = 0.135;
23     Double_t p_kaon = 5.0; //GeV
24     Double_t beta = p_kaon / sqrt(p_kaon * p_kaon + m_kaon * m_kaon);
25
26     Int_t nEvents = 1e6;
27
28     TH2F * inv_mass = new TH2F("inv_mass", "; Transverce momentum, GeV; Invariant mas
29
30     for (Int_t i = 0; i < nEvents; i++){
31         Double_t e_pion = m_kaon / 2;
32         Double_t p_pion = sqrt(e_pion * e_pion - m_pion * m_pion);
33         Double_t cos_theta = 2 * gRandom -> Rndm() - 1;
34         Double_t sin_theta = sqrt(1 - cos_theta * cos_theta);
35         Double_t phi = gRandom -> Rndm() * 2 * 3.14159;
36
37         Double_t px1 = p_pion * sin_theta * cos(phi);
38         Double_t py1 = p_pion * sin_theta * sin(phi);
```

```

39     Double_t pz1 = p_pion * cos_theta;
40
41     Double_t px2 = -px1;
42     Double_t py2 = -py1;
43     Double_t pz2 = -pz1;
44
45     TLorentzVector pion1 (px1, py1, pz1, e_pion);
46     TLorentzVector pion2 (px2, py2, pz2, e_pion);
47
48     pion1.Boost(0., 0., beta);
49     pion2.Boost(0., 0., beta);
50
51     p1_x = pion1.Px();
52     p1_y = pion1.Py();
53     p1_z = pion1.Pz();
54     p1_e = pion1.E();
55
56     p2_x = pion2.Px();
57     p2_y = pion2.Py();
58     p2_z = pion2.Pz();
59     p2_e = pion2.E();
60
61     p1_x = gRandom -> Gaus(p1_x, 0.01);
62     p1_y = gRandom -> Gaus(p1_y, 0.01);
63     p1_z = gRandom -> Gaus(p1_z, 0.01);
64     p1_e = gRandom -> Gaus(p1_e, 0.01);
65
66     p2_x = gRandom -> Gaus(p2_x, 0.01);
67     p2_y = gRandom -> Gaus(p2_y, 0.01);
68     p2_z = gRandom -> Gaus(p2_z, 0.01);
69     p2_e = gRandom -> Gaus(p2_e, 0.01);
70
71     Double_t momentum = pow(p1_x + p2_x, 2) + pow(p1_y + p2_y, 2) + pow(p1_z + p2_z, 2);
72     invmass = sqrt(pow(p1_e + p2_e, 2) - momentum);
73     //std::cout << sqrt(momentum) << std::endl;
74     Double_t trans_mom = sqrt(pow(p1_y + p2_y, 2) + pow(p1_x + p2_x, 2));
75
76     tree.Fill();
77     inv_mass -> Fill(trans_mom, invmass);
78 }

```

```

79
80 TCanvas *c1 = new TCanvas ("c1", "invariant_mass", 1200, 800);
81 c1 -> cd();
82 inv_mass -> SetStats(0);
83 inv_mass -> Draw("colz");
84 //inv_mass -> Draw();
85 c1 -> Update();
86 c1 -> Modified();
87 c1 -> Print("invmass_01.png");
88
89 fOut.Write();
90 fOut.Close();
91 }

```

Для $\sigma = 0.01$ получим следующую зависимость инвариантной массы от поперечного импульса:

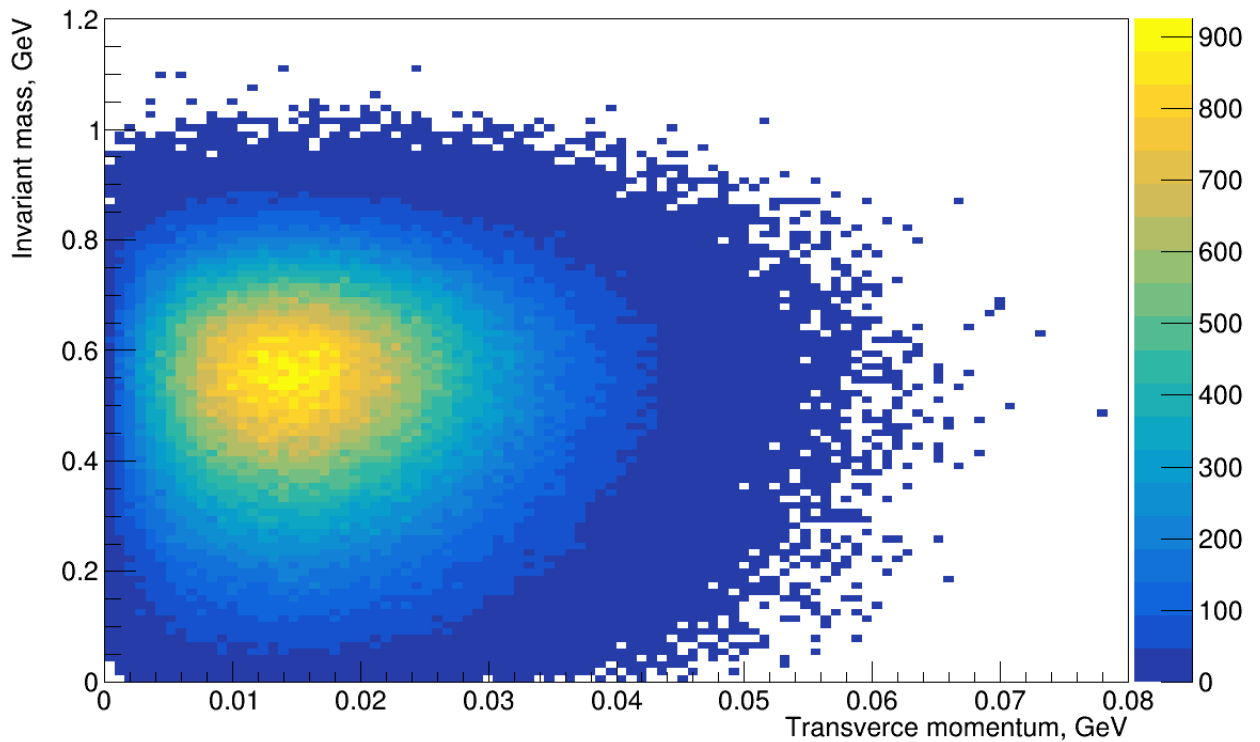


Рис. 1: Зависимость инвариантной массы от поперечного импульса, $\sigma = 0.01$

Приведем зависимости инв. массы от поперечного импульса для $\sigma = 0.5$ и $\sigma = 1.0$:

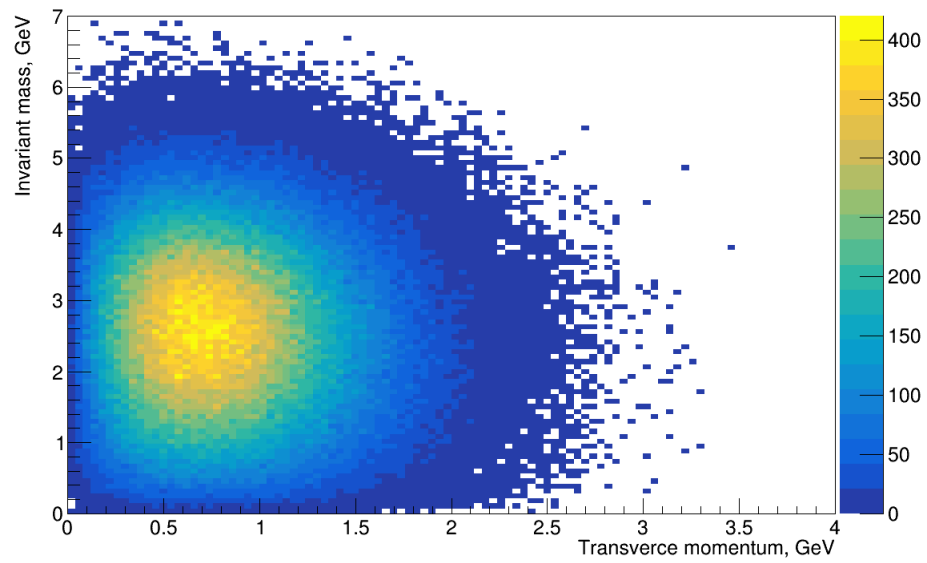


Рис. 2: Зависимость инвариантной массы от поперечного импульса, $\sigma = 0.5$

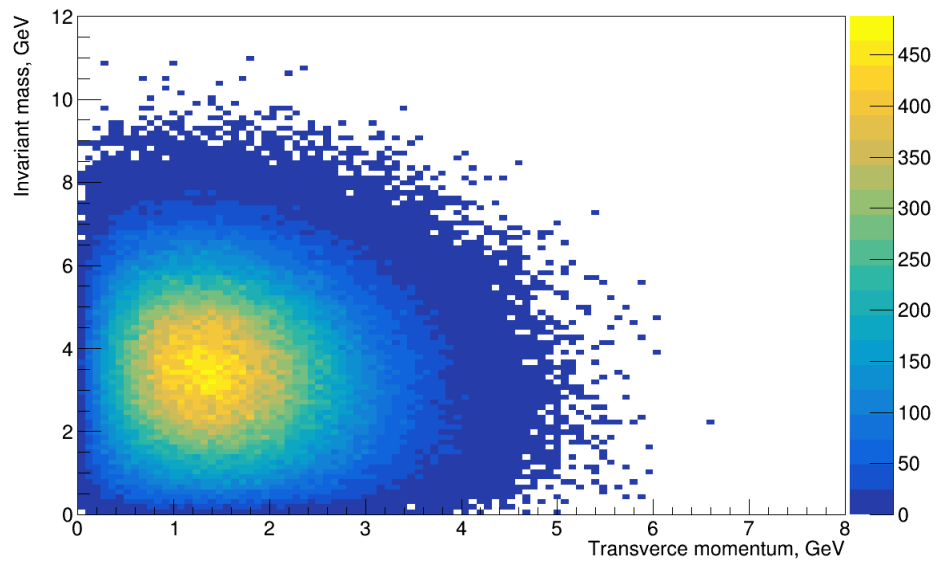


Рис. 3: Зависимость инвариантной массы от поперечного импульса, $\sigma = 1$