

Моделирование и обработка результатов измерений

Практические работы

Тема 1: «Метод наименьших квадратов»

- 1: линейный МНК с графическим изображением полученной функции
- 2: линейный МНК с ошибками по оси Y в каждой точке и с графиком функции
- 3: квадратичный МНК без учета ошибок, с графиком функции
- 4: фитирование экспериментальных распределений функцией Гаусса

Тема 2: «Моделирование и обработка результатов по распаду π^0 -мезона на 2 γ -кванта»

- 5: моделирование распада $\pi^0 \rightarrow \gamma + \gamma$ в системе покоя π^0 -мезона
- 6: преобразование кинематических характеристик вторичных частиц (γ -квантов) в лабораторную систему отсчета, формулы Лоренц-преобразования
- 7: моделирование детектора γ -квантов и запись модельных сигналов детектора во внешний файл
- 8: анализ экспериментальных данных по распаду π^0 -мезона. Чтение файла из работы № 7 и проверка кинематических параметров вторичных частиц
- 9: построение массового спектра системы двух γ -квантов и восстановление массы родительской частицы

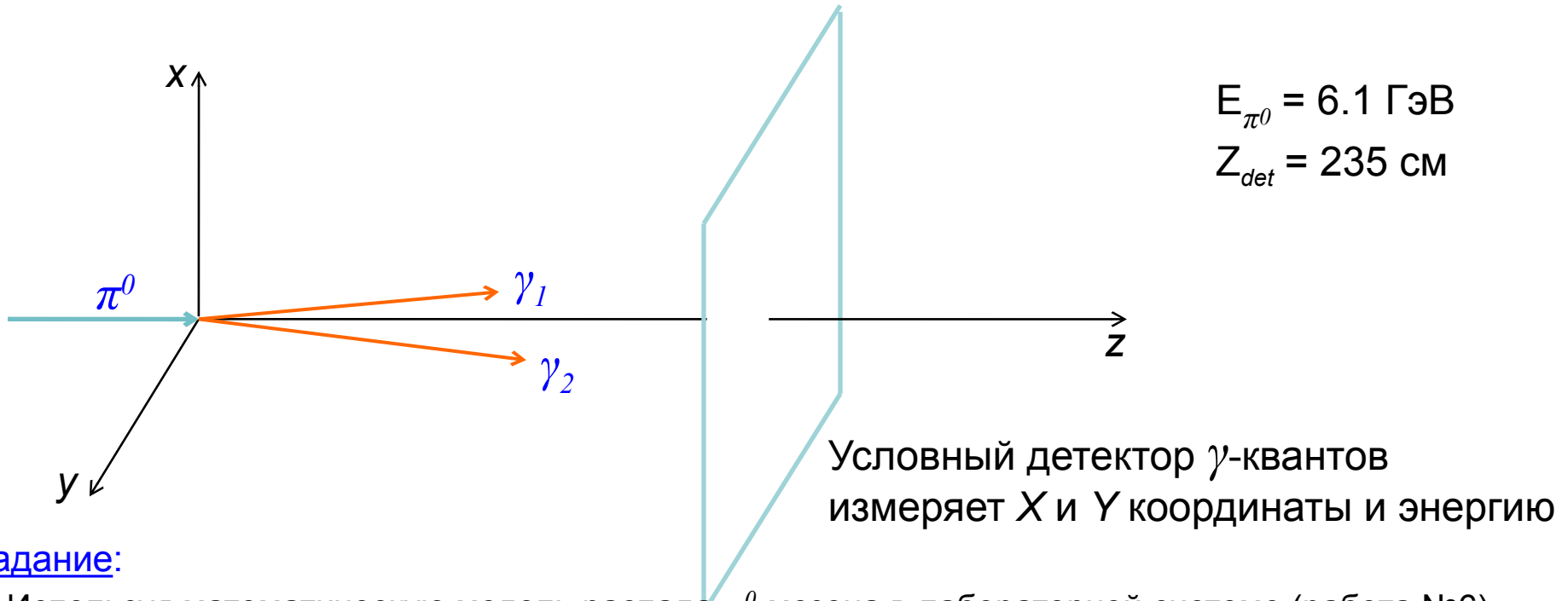
Поэтапное развитие программы с математической моделью распада частицы и детектирования продуктов распада

Создание второй программы, осуществляющей анализ модельных данных

Практическая работа №7

Распад $\pi^0 \rightarrow \gamma + \gamma$

Моделирование детектора γ -квантов и запись модельных сигналов детектора во внешний файл



Задание:

1. Используя математическую модель распада π^0 -мезона в лабораторной системе (работа №6), смоделировать прохождение γ -квантов через входное окно детектора и записать сигналы с детектора, т.е. энергии и координаты γ -квантов с помощью пакета Root в «дерево» (tree) с дальнейшим сохранением в Root-файле. Особое внимание обратить на редкие случаи непопадания γ -квантов в детектор, т. е. когда $p_z < 0$
2. Для контроля качества смоделированных данных построить двумерные гистограммы с X и Y координатами точки попадания каждого из γ -квантов в детектор

Создание и работа с «деревьями» в программном пакете ROOT

Подробнее: <https://root.cern.ch/root/html/doc/guides/users-guide/ROOTUsersGuide.html#trees>

Подключение необходимых библиотек

```
#include "TFile.h"  
#include "TTree.h"
```

Открытие нового файла, в котором будет сохранено «дерево»

```
TFile *my_file = new TFile("my_file_name.root", "NEW");
```

Проверка открытия файла

```
if (my_file == NULL)  
{ cout << "ERROR !" << endl; return -1; }
```

Создание «дерева»

```
TTree *my_tree = new TTree ("my_tree", "Name of my tree");
```

Определение всех переменных, составляющих «ветки» «дерева»

```
float px, py, pz;  
my_tree -> Branch("px", &px, "px/F");  
my_tree -> Branch("py", &py, "py/F");  
my_tree -> Branch("pz", &pz, "pz/F");
```

Заполнение всех веток «дерева» в цикле

```
for (i=.....)  
{  
    px=...; py=...; pz=...;  
    my_tree -> Fill();  
}
```

Сохранение «дерева» в файле и его закрытие

```
my_file -> Write();  
my_file -> Close();
```

C: символьная строка с "0" на конце
B: 8-битовое целое число со знаком
b: 8-битовое целое число без знака
S: 16-битовое целое число со знаком
s: 16-битовое целое число без знака
l: 32-битовое целое число со знаком
i: 32-битовое целое число без знака
L: 64-битовое целое число со знаком
l: 64-битовое целое число без знака
F: 32-битовое число с плавающей точкой
D: 64-битовое число с плавающей точкой