



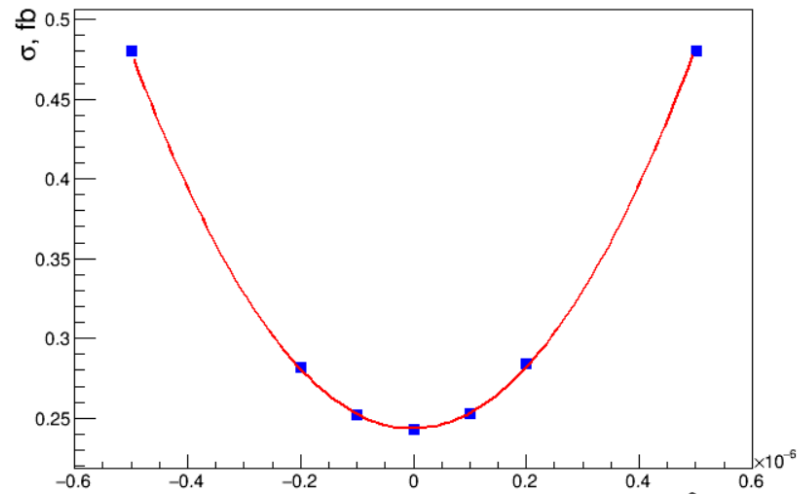
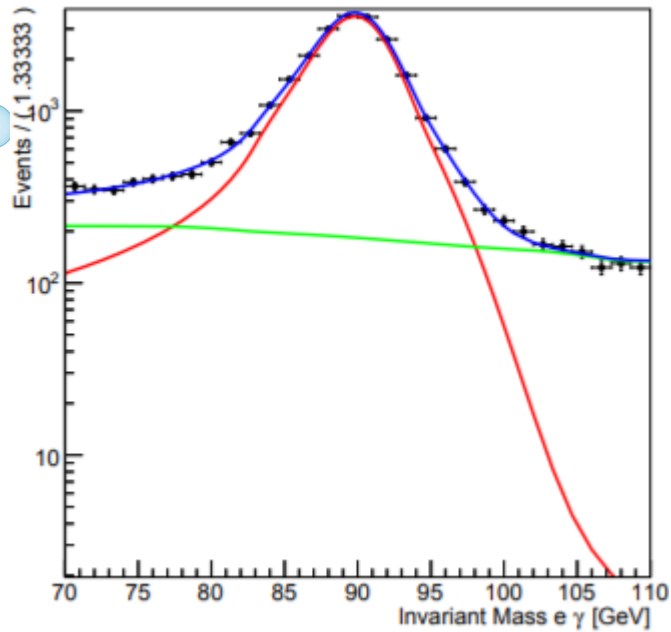
Лекции. Практические занятия

Солдатов Е.Ю.

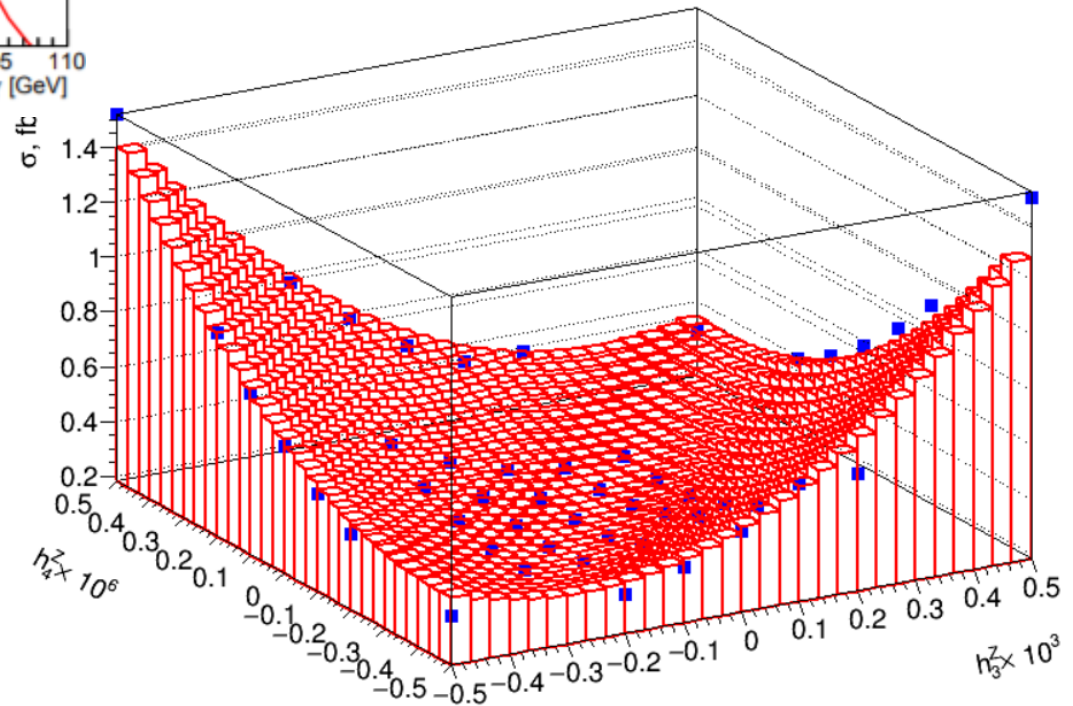
2024 г.

ФИТИРОВАНИЕ

Описание экспериментальных результатов теоретическими зависимостями.



Результат:
Параметры фитирующей
функции



ГИСТОГРАММЫ: ФИТИРОВАНИЕ

- В меню, открывающемся по клику правой кнопкой мыши при наведении на горизонтальную линию/точку гистограммы можно выбрать пункт FitPanel.

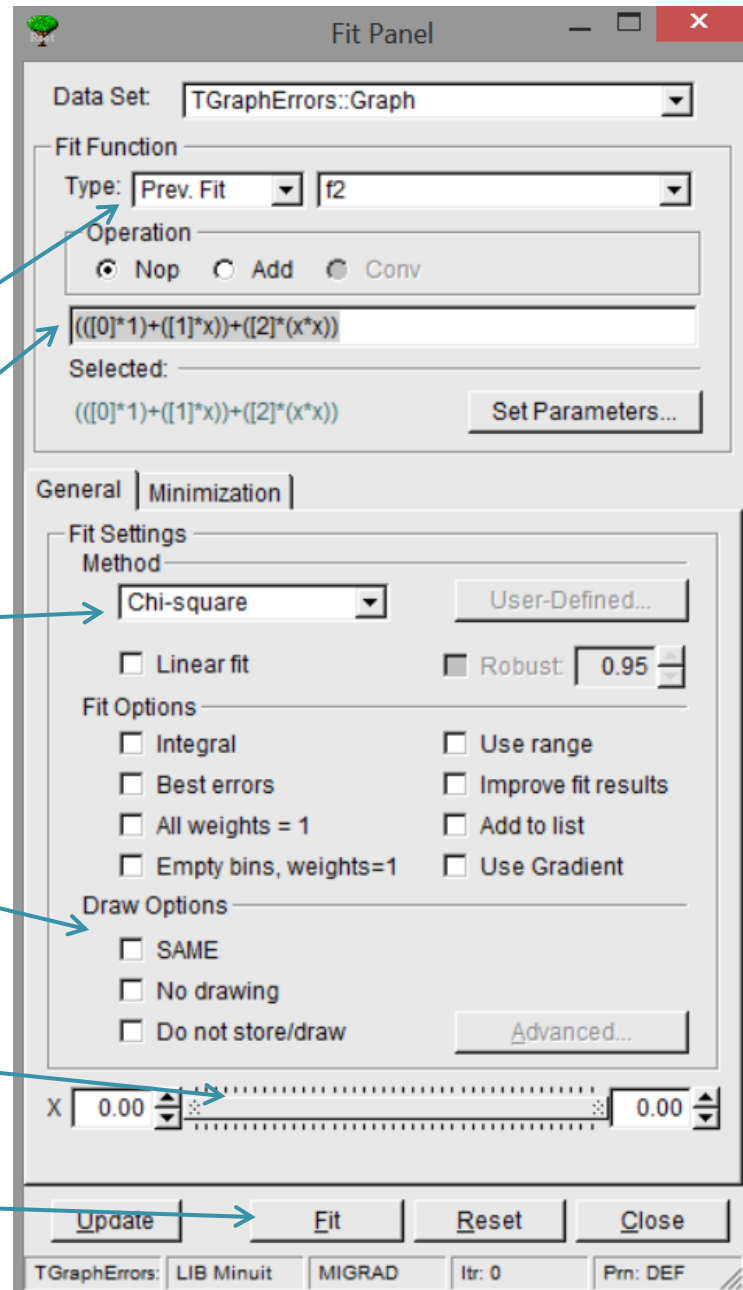
Выбор фитирующей функции из предустановленных или самостоятельно определённых

Опции фита, метод: какая функция минимизируется

Опции рисования

Варьирование диапазонов фита ползунками

Запустить фитирование



ГИСТОГРАММЫ: ФИТИРОВАНИЕ

- Вывод параметров фита в окне root

Пример типичного вывода для фитирования функцией Гаусса

NO.	NAME	VALUE	ERROR	SIZE	DERIVATIVE
1	Constant	3.14231e-001	1.42985e-002	2.79029e-005	1.07440e-002
2	Mean	-1.94760e-006	2.75504e-007	5.37743e-010	8.38081e+002
3	Sigma	3.43230e-006	1.92215e-008	4.50570e-003**	at limit **

- NO - номер параметра
- NAME - имя параметра
- VALUE - найденное значение параметра
- ERROR - погрешность значения

ГИСТОГРАММЫ: ФИТИРОВАНИЕ

- Для фитирования используется метод `THIF::Fit()`
- Фитирование встроенной функцией `hl->Fit("gaus")`

Встроенные функции:

“**gaus**” функция Гаусса с 3-мя параметрами:

$$f(x) = p0 * \exp(-0.5 * ((x - p1) / p2)^2)$$

“**expo**” экспонента с 2 параметрами:

$$f(x) = \exp(p0 + p1 * x)$$

“**polN**” полином степени N, где N - число между 0 и 9:

$$f(x) = p0 + p1 * x + p2 * x^2 + \dots$$

“**landau**” функция Ландау со средним и отклонением

“**gausn**” нормированная форма функции Гаусса с 3-мя параметрами:

$$f(x) = p0 * \exp(-0.5 * ((x - p1) / p2)^2) / (p2 * \sqrt{2 * \pi})$$

ГИСТОГРАММЫ: ФИТИРОВАНИЕ

Пример:

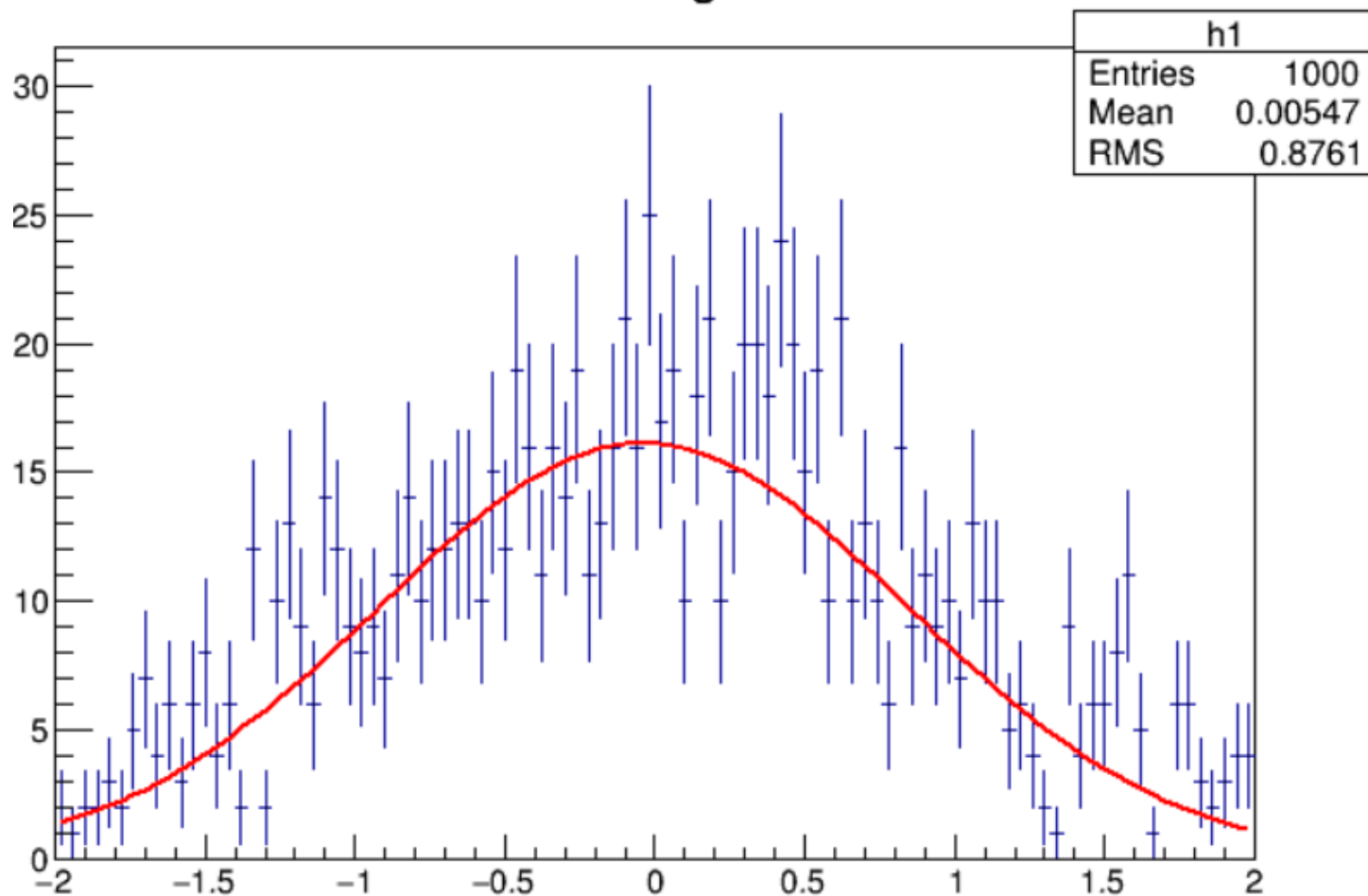
```
TH1F *h1 = new TH1F("h1","Random gauss",100,-2,2);
```

```
h1->FillRandom("gauss",1000);
```

```
h1->Draw("E");
```

```
h1->Fit("gauss");
```

Random gauss



ГИСТОГРАММЫ: ФИТИРОВАНИЕ

- Функции в ROOT реализованы классом TFI
- Создание функции $x \cdot \cos(x)$, определённой на интервале $(0, 5)$ будет выглядеть так:

```
TFI *func1 = new TFI("func1", "x*cos(x)", 0, 5)
```

- Создание функции с параметрами делается так:

```
TFI *func2 = new TFI("func2", "[0]*sin(x)*exp(-[1]*x)", 0, 5)
```

здесь [0] и [1] – свободные параметры функции.

- В случае полинома возможен равнозначный вариант:

```
TFI *func3 = new TFI("func3", "x++x*x*x)", 0, 5)
```

Здесь также будет 2 свободных параметра перед каждым из слагаемых.

- Чтобы фитировать гистограмму, нужно использовать метод `Fit(func1)`

- Для доступа к результатам фита:

```
Double_t param1 = func1->GetParameter(0)
```

```
Double_t param1_err = func1->GetParError(0)
```

- и его качественным характеристикам:

```
Double_t chi2 = func1->GetChisquare()
```

```
Double_t ndof = func1->GetNDF()
```

Ref. manual: <https://root.cern.ch/doc/master/classTFI.html>

ГИСТОГРАММЫ: ФИТИРОВАНИЕ

Пример:

```
TH1F *h1 = new TH1F("h1","Random gauss",100,-2,2);
```

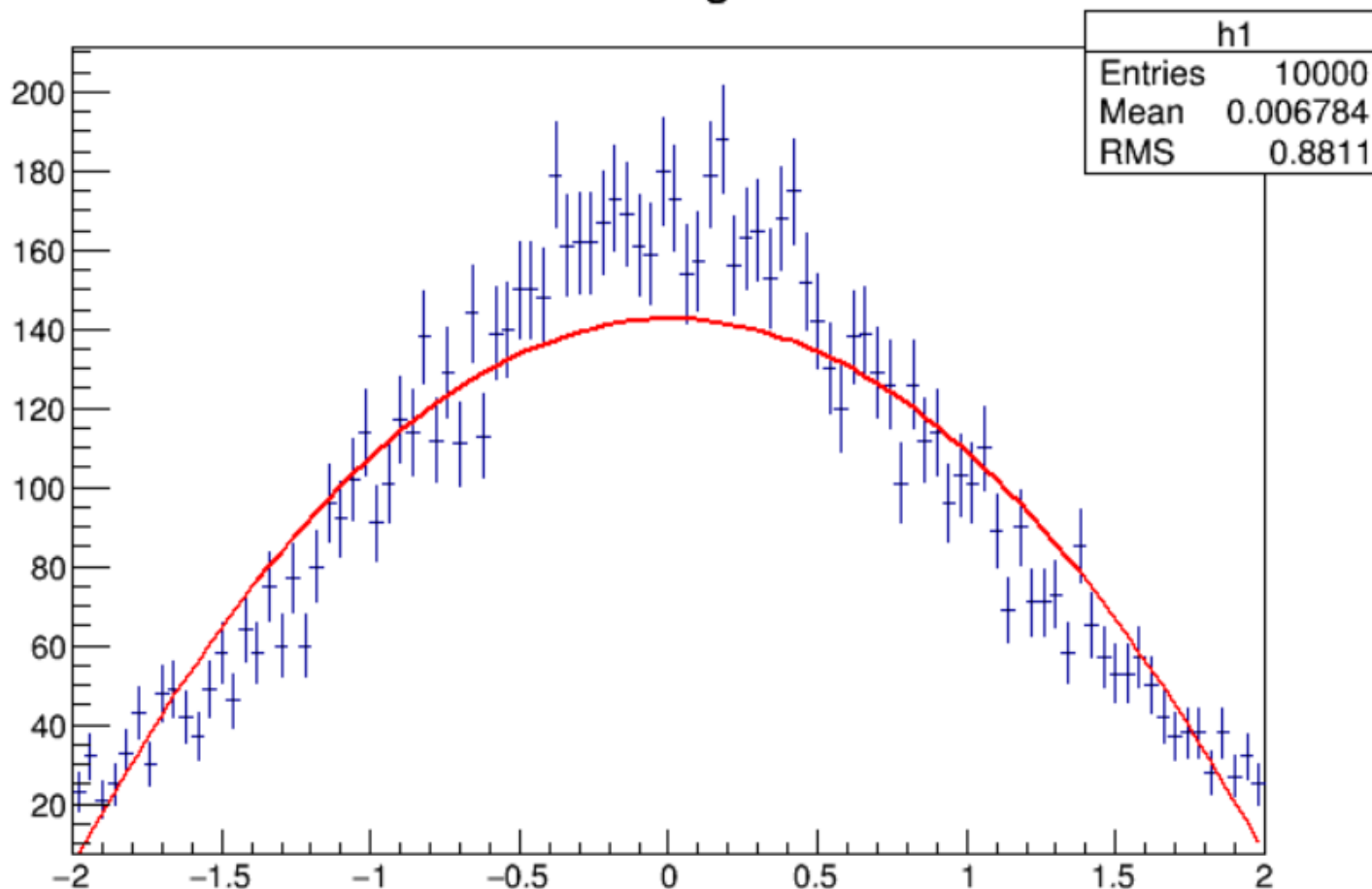
```
h1->FillRandom("gaus",10000);
```

```
TF1 *func1 = new TF1("func1","[0]+[1]*x+[2]*x*x",-2,2);
```

```
h1->Fit(func1,"R");
```

```
h1->Draw("E");
```

Random gauss



ГИСТОГРАММЫ: ФИТИРОВАНИЕ

- Более сложный пример – фитирование по кускам:

```
TFI* f1 = new TFI("m1","poll",20,81);  
TFI* f2 = new TFI("m2","gaus",81,101);  
TFI* f3 = new TFI("m3","poll",101,150);  
TFI* total = new TFI("mstotal","poll(0)+gaus(1)+poll(4)",20,150);
```

```
//Histogram h – from analysis
```

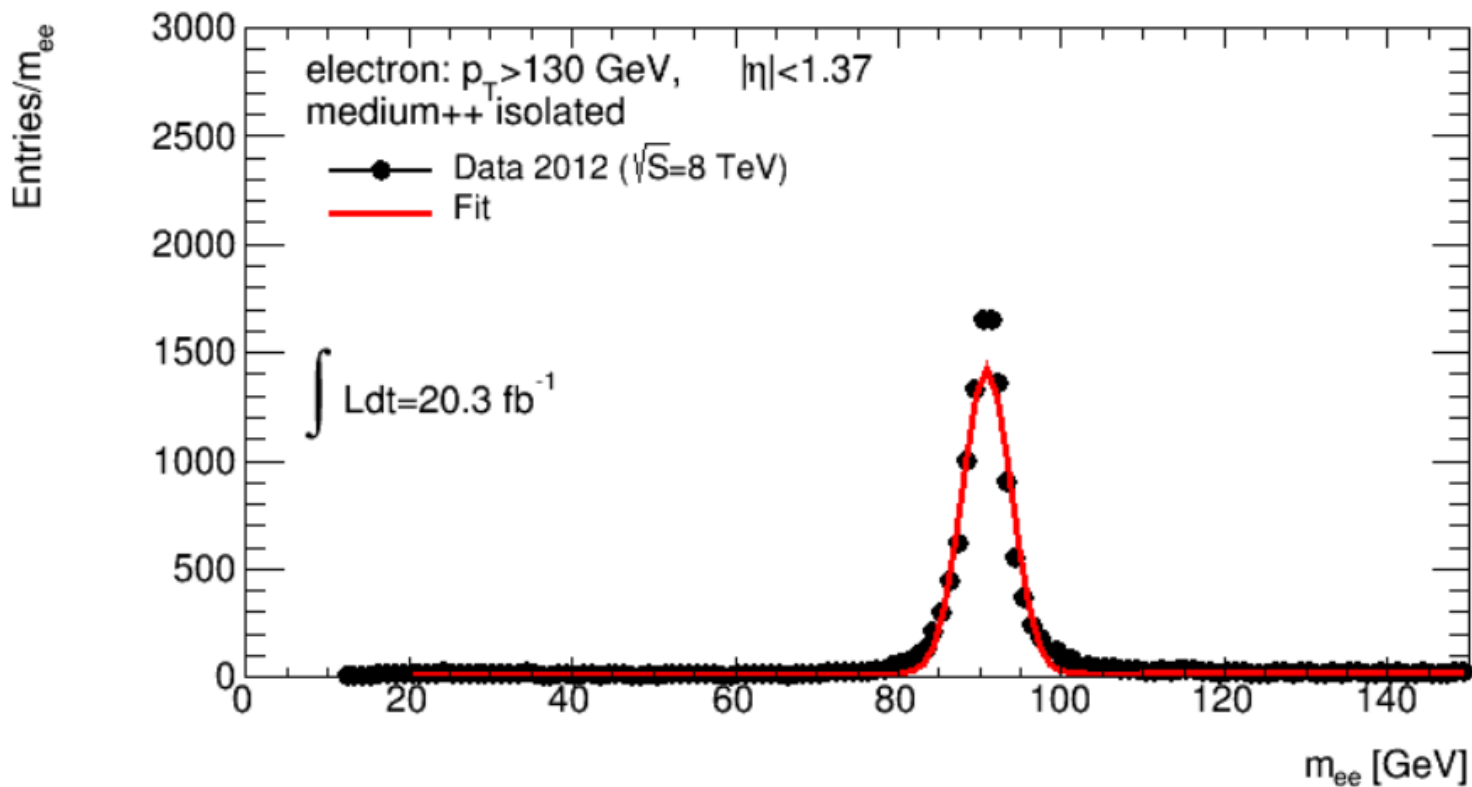
```
h->Fit(f1,"R");  
h->Fit(f2,"R+");  
h->Fit(f3,"R+");
```

```
// Get the parameters from the fit  
f1->GetParameters(&par[0]);  
f2->GetParameters(&par[3]);  
f3->GetParameters(&par[6]);
```

```
// Use the parameters on the sum  
total->SetParameters(par);  
h->Fit(total,"R+");
```

ГИСТОГРАММЫ: ФИТИРОВАНИЕ

Результат:



```
FCN=803.104 FROM MIGRAD STATUS=CONVERGED 332 CALLS 333 TOTAL
EDM=2.03919e-009 STRATEGY= 1 ERROR MATRIX UNCERTAINTY
```

1.5 per cent

EXT NO.	PARAMETER NAME	VALUE	ERROR	STEP SIZE	FIRST DERIVATIVE
1	p0	-3.19233e+001	4.34099e-001	3.74817e-005	-1.27996e-005
2	p1	1.41219e+003	2.05515e+001	2.61512e-002	-1.13951e-003
3	p2	9.08438e+001	3.08320e-002	6.10660e-006	-2.05498e-003
4	p3	3.00582e+000	3.39806e-002	-2.90244e-005	-2.07020e-003
5	p4	3.79416e+001	4.34099e-001	3.74226e-005	-1.27985e-005
6	p5	-1.41208e+003	2.05514e+001	-2.61617e-002	-1.13880e-003

ДВУМЕРНЫЕ ГИСТОГРАММЫ

- Работа с двумерной гистограммой аналогична одномерному случаю
- Основной класс TH2. Его производные TH2I, TH2F, TH2D, TH2C...
- При создании гистограммы следует указать число бинов как по оси X, так и по оси Y, а также соответствующие диапазоны изменения величин

Основные конструкторы:

```
TH2F *h1 = new TH2F("HistName", "Histogram  
title", NbinsX, xmin, xmax, NbinsY, ymin, ymax)
```

```
double xarray[NbinsX+1]={xmin,...,xmax};  
double yarray[NbinsY+1]={ymin,...,ymax};  
TH2F *h1 = new TH1F("HistName", "Histogram  
title", NbinsX, xarray, NbinsY, yarray)
```

- При заполнении следует передавать два значения (и, опционально, вес)
`h2->Fill(Xvalue, Yvalue)`
`h2->Fill(Xvalue, Yvalue, weight)`
- Рисование гистограммы осуществляется точно также
`h2->Draw()`
- По умолчанию 2D-гистограмма изображается как «облако» точек, плотность которого пропорциональна содержимому клетки.

ДВУМЕРНЫЕ ГИСТОГРАММЫ

Пример:

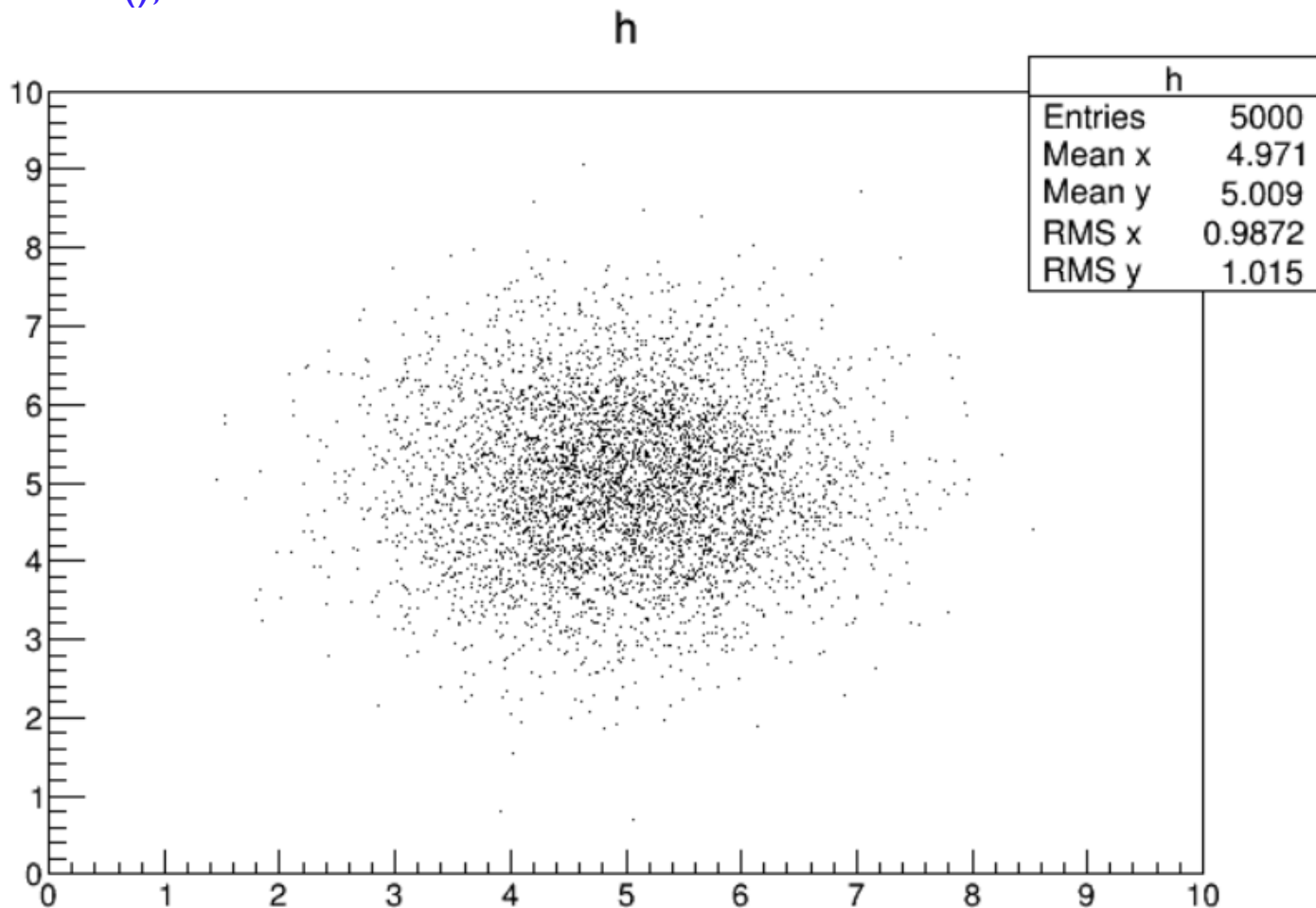
```
TH2F *h = new TH2F("h","h",100,0.,10.,100,0.,10.);
```

- TF2 *xyg = new TF2("xyg","xygaus",0,10,0,10);

```
xyg->SetParameters(1,5,1,5,1); //amplitude, meanx, sigmax, meany, sigmay
```

```
h->FillRandom("xyg");
```

```
h->Draw();
```

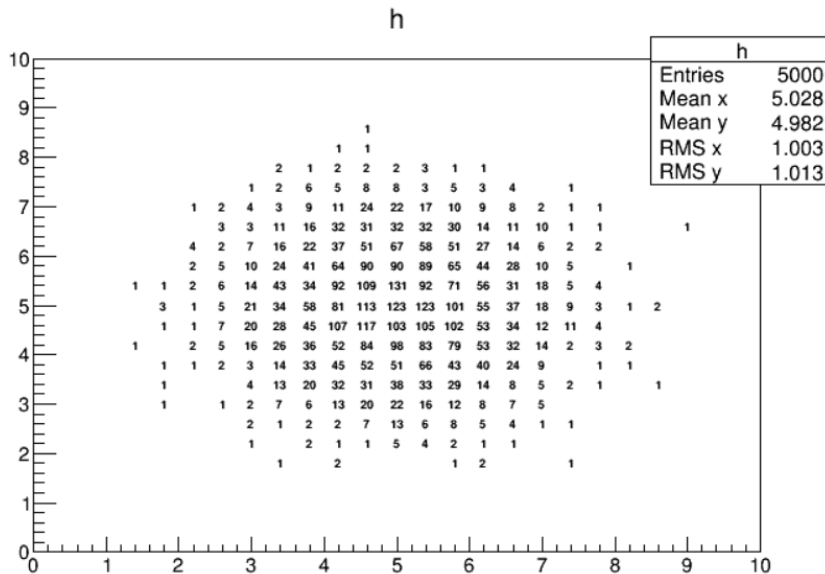


ДВУМЕРНЫЕ ГИСТОГРАММЫ

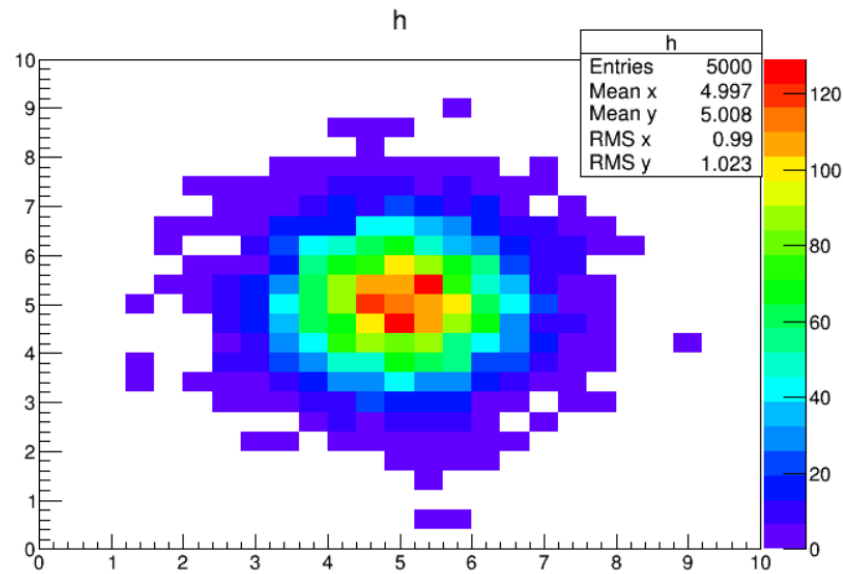
- Методу `Draw()` можно передавать различные опции рисования гистограмм. Для 2D есть специальные опции.

TEXT — напечатать значения содержимого клеток } 2D представление
COLZ – изобразить цветовую карту значений }

`h1->Draw("TEXT")`



`h1->Draw("COLZ")`

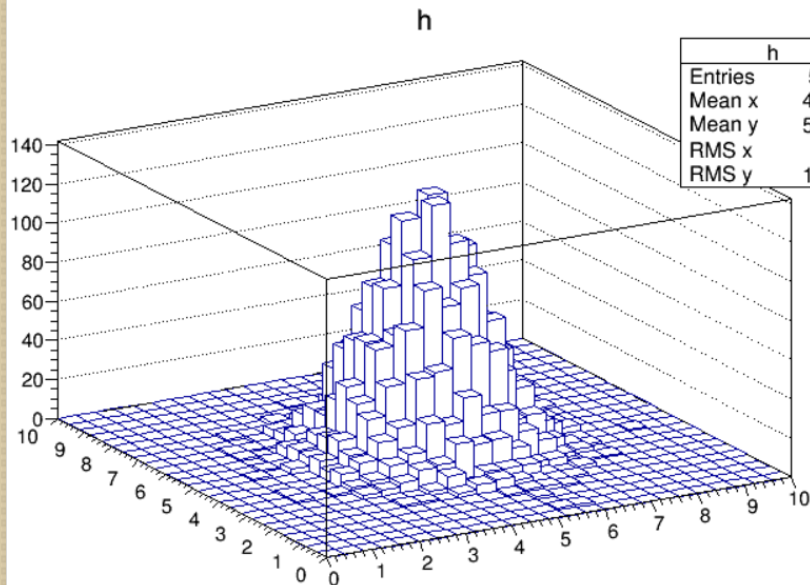


ДВУМЕРНЫЕ ГИСТОГРАММЫ

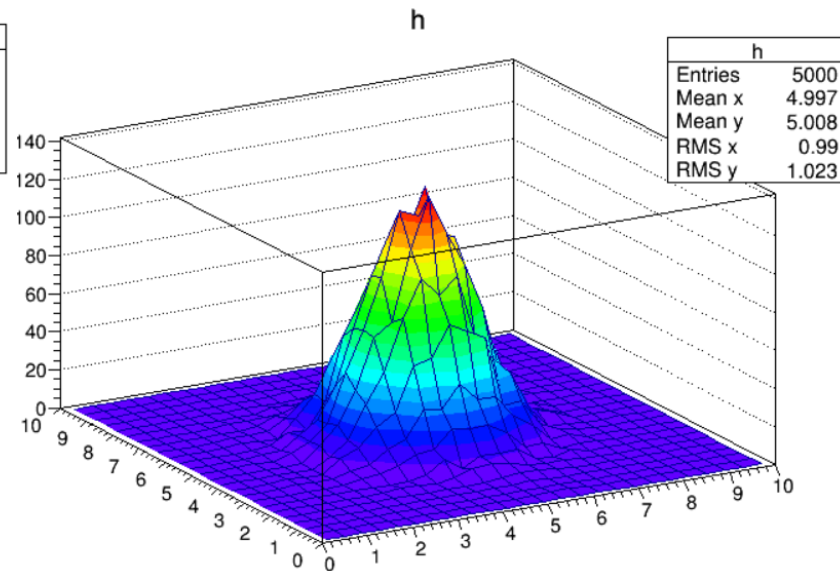
- Методу `Draw()` можно передавать различные опции рисования гистограмм. Для 2D есть специальные опции.

LEGO — нарисовать трехмерное изображение в виде столбцов } 3D
SURF — нарисовать ячеистую поверхность } представле
ние

```
h1->Draw("LEGO")
```



```
gStyle->SetPalette(1)  
h1->Draw("SURF1")
```



ПРОФИЛИ

Для изучения двухмерных распределений часто удобно использовать профили этих распределений на одну из осей.

Профили для двумерных гистограмм представляют собой одномерные гистограммы со средними значениями по одной из осей. Таким образом удобно изучать, например, корреляции между 2 переменными.

Базовый класс TProfile

```
root [5] g=h->ProfileX()  
(TProfile *) @0x11dd604  
root [6] g->Draw()
```

