

# 第二次作业

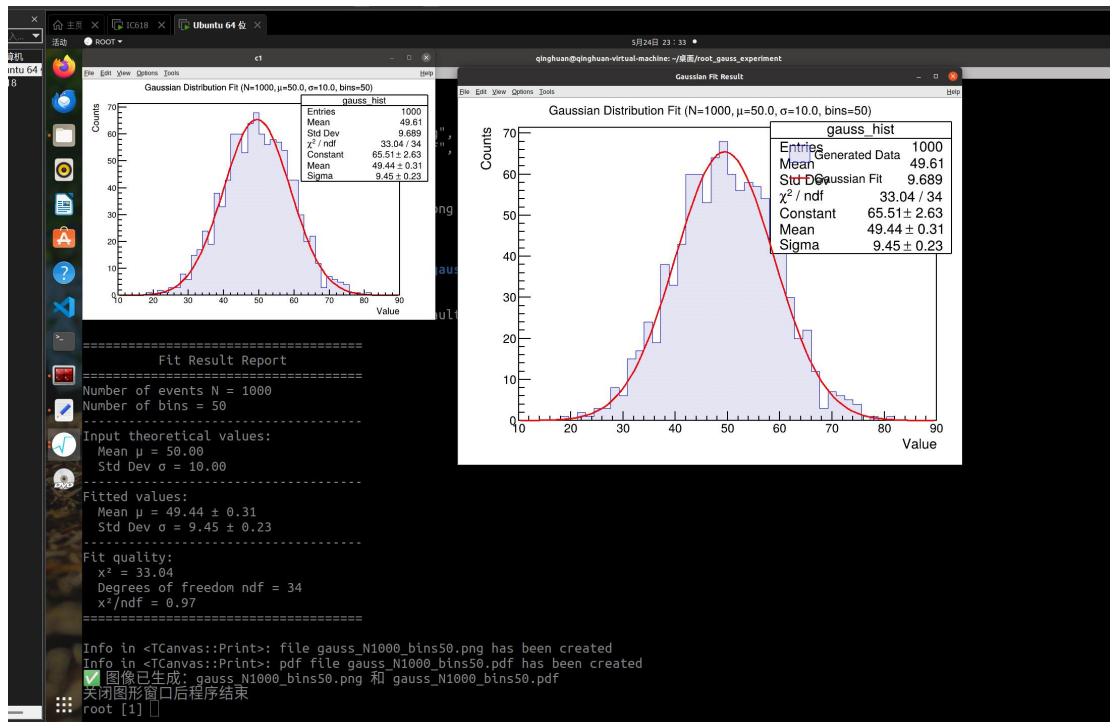
## 1. Root 任务

### (1) 代码

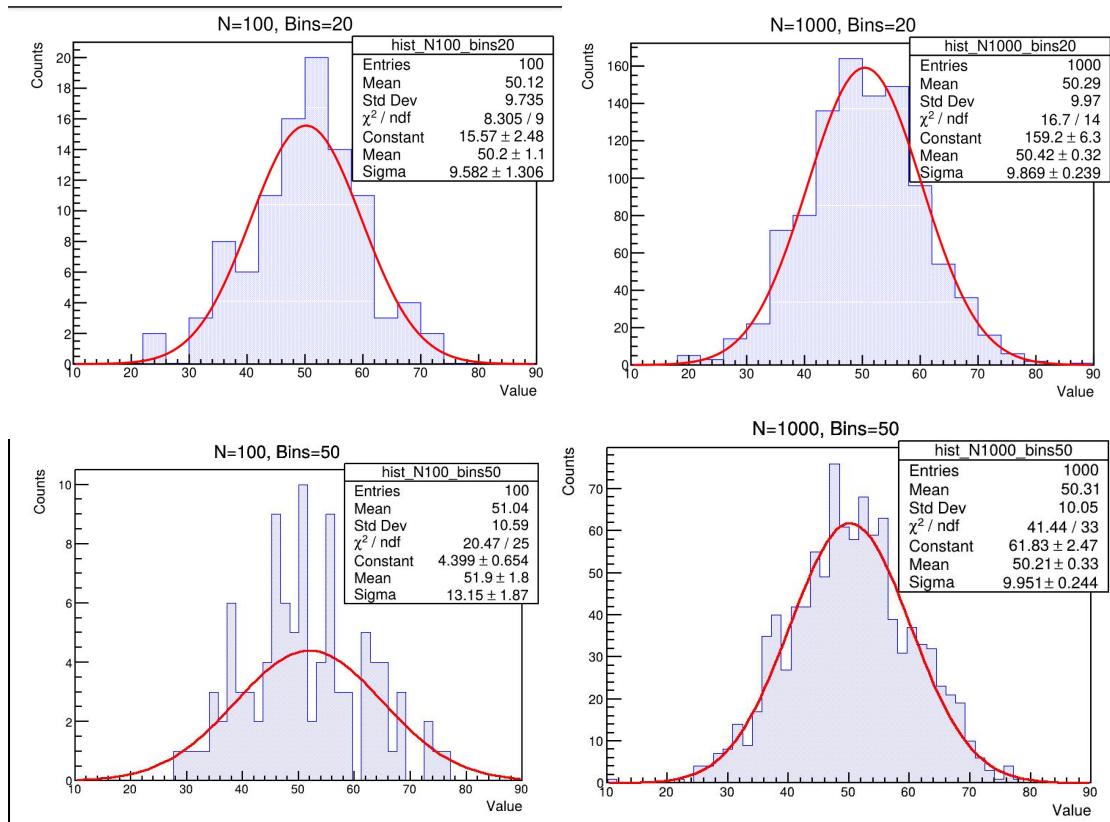
```
gauss_fit.cpp
~/桌面/root_gauss_experiment

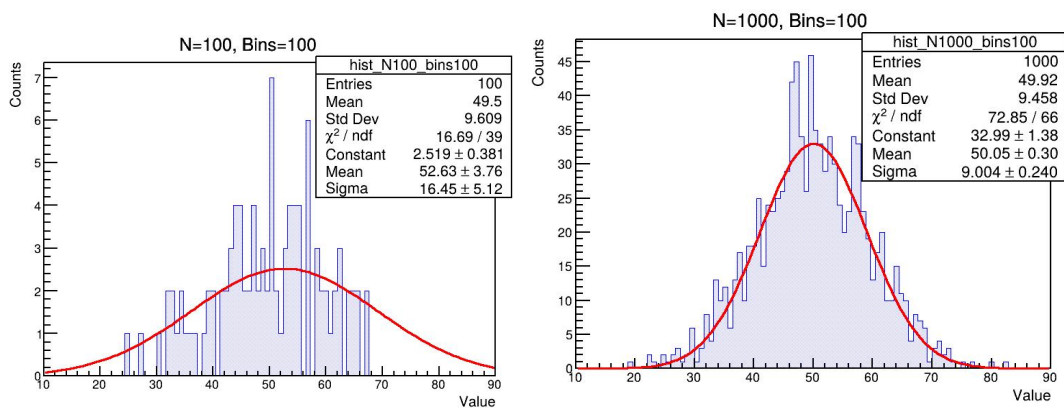
1 #include "TH1F.h"
2 #include "TF1.h"
3 #include "TRandom3.h"
4 #include "TCanvas.h"
5 #include "TApplication.h"
6 #include <cstdlib>
7
8 int main(int argc, char** argv) {
9     // 创建ROOT应用程序 (必须有, 否则无法显示图形)
10    TApplication app("GaussFitApp", &argc, argv);
11
12    // ===== 实验参数设置区 =====
13    const int N = 1000; // 随机数数量 (可修改: 100, 1000, 10000, 100000)
14    const double mean = 50; // 高斯分布均值 (固定为50)
15    const double sigma = 10; // 高斯分布标准差 (固定为10)
16    const int nbins = 50; // 直方图bin数量 (可修改: 10, 20, 50, 100, 200)
17    // =====
18
19    // 1. 创建随机数生成器 (使用系统时间作为种子, 保证每次运行结果不同)
20    TRandom3 rng(0);
21
22    // 2. 创建直方图
23    // 参数: 直方图名称, 直方图标题, bin数量, x轴最小值, x轴最大值
24    TH1F* hist = new TH1F("gauss_hist",
25        Form("高斯分布拟合 (N=%d, #mu=%1f, #sigma=%1f, bins=%d)",
26            N, mean, sigma, nbins),
27        nbins, mean - 4*sigma, mean + 4*sigma);
28
29    // 3. 生成N个高斯分布随机数并填充到直方图
30    for (int i = 0; i < N; i++) {
31        double x = rng.Gaus(mean, sigma); // 生成一个高斯分布随机数
32        hist->Fill(x); // 将随机数填充到直方图
33    }
34
35    // 4. 创建高斯拟合函数
36    // 参数: 函数名称, 函数形式 ("gaus"是ROOT内置的高斯函数), 拟合范围
37    TF1* gauss_func = new TF1("gauss_func", "gaus", mean - 4*sigma, mean + 4*sigma);
38
39    // 5. 执行拟合
40    // "q"参数: 安静模式, 不输出详细拟合信息
41    // "r"参数: 使用指定的范围进行拟合
42    hist->Fit(gauss_func, "QR");
43
44    // 6. 获取拟合结果
45    double fit_mean = gauss_func->GetParameter(1); // 拟合得到的均值
46    double fit_sigma = gauss_func->GetParameter(2); // 拟合得到的标准差
47    double mean_error = gauss_func->GetParError(1); // 均值的误差
48    double sigma_error = gauss_func->GetParError(2); // 标准差的误差
49    double chi2 = gauss_func->GetChisquare(); // 卡方值
50    double ndf = gauss_func->GetNDF(); // 自由度
51    double chi2_ndf = chi2 / ndf; // 归一化卡方 (最重要的拟合质量指标)
52
53    // 7. 在终端输出拟合结果
54    printf("\n");
55    printf("=====\n");
56    printf("          拟合结果报告\n");
57    printf("=====\n");
```

## (2) 实验过程



## (3) 实验图像





#### (4) 实验结论

- 随机数数量  $N$  越大：拟合结果越准确，参数误差越小，直方图越平滑，拟合质量越好。
- bin 数量存在最优值：太少会丢失细节产生系统偏差，太多会因统计涨落增大误差。
- 最佳实践：每个 bin 内平均 10-20 个统计量时拟合效果最好，统计量越大，最优 bin 数量也越大。

## 2. 插值与拟合的区别、原理及拟合质量评估

### (1) 核心区别

- 插值：要求构造的函数严格通过所有已知数据点，用于在已知点之间估算未知点的值，不考虑数据误差。
- 拟合：构造的函数不要求通过所有数据点，而是寻找与数据整体趋势最接近的曲线，允许数据点存在误差，用于揭示数据背后的规律。

## (2) 应用实例

- 插值应用：实验数据的中间值估算（如已知两个温度下的电阻值，估算中间温度的电阻）、图像缩放（通过插值算法补充像素）。
- 拟合应用：物理实验的规律验证（如本次高斯分布拟合，验证数据是否符合理论分布）、参数测量（通过拟合获取物理模型的未知参数）。

## (3) 常见原理

- 插值原理：通过多项式、样条等方法，构造连续函数，使该函数在已知点上的取值与数据完全一致。常见方法：线性插值、三次样条插值、拉格朗日插值。
- 拟合原理：通过最小化数据点与函数值的偏差（如最小二乘法），确定函数的最优参数，使整体误差最小。常见方法：最小二乘拟合、最大似然拟合。

## (4) 拟合质量评估

- $\chi^2$  检验：核心指标，理想值为 1（0.8-1.2 为良好），反映拟合函数与数据的整体偏差。
- 残差分析：残差（数据点与拟合值的差）随机分布在 0 附近，无明显趋势，说明拟合效果好。

- 参数误差：拟合得到的参数误差越小，说明参数测量精度越高。
- 决定系数  $R^2$ ：取值 0-1，越接近 1，说明拟合函数能解释的数据变异越多。

### 3. 基本粒子的分类、属性及实例

#### (1) 基本粒子的分类（标准模型框架）

按自旋分为费米子和玻色子两大类：

- 费米子（自旋为  $1/2$ ）：构成物质的基本单元，遵循泡利不相容原理，分为夸克和轻子。
- 玻色子（自旋为整数）：传递相互作用的媒介粒子，不遵循泡利不相容原理，分为规范玻色子和希格斯玻色子。

#### (2) 各类粒子的共同属性

- 费米子：具有半整数自旋，受泡利不相容原理约束，分为三代，质量随代数增大而增加。
- 夸克：参与强、电磁、弱相互作用，带分数电荷，存在颜色自由度，无法单独存在（夸克禁闭）。
- 轻子：不参与强相互作用，分为带电轻子和中微子，带电轻子参与电磁相互作用，中微子仅参与弱相互作用。
- 规范玻色子：传递四种基本相互作用，无质量（光子、胶

子)或质量较大 ( $W/Z$  玻色子)。

- 希格斯玻色子: 自旋为 0, 通过希格斯机制赋予其他粒子质量。

### (3) 实例

- 夸克实例: 上夸克 ( $u$ )、下夸克 ( $d$ )、粲夸克 ( $c$ )
- 轻子实例: 电子 ( $e^-$ )、 $\mu$  子 ( $\mu^-$ )、 $\tau$  中微子 ( $\nu\tau$ )
- 规范玻色子实例: 光子 ( $\gamma$ )、 $W^+$ 玻色子、胶子 ( $g$ )
- 希格斯玻色子: 希格斯玻色子 ( $H$ )

## 4. 径迹探测器通过电磁相互作用测量粒子信息的原因

(1) **相互作用强度适中:** 电磁相互作用强度适中, 既不会像强相互作用那样导致粒子发生大角度偏转或产生大量次级粒子, 也不会像弱相互作用那样发生概率极低, 能够稳定产生可测量的信号。

(2) **信号易于探测:** 带电粒子穿过介质时, 会与原子电子发生库仑相互作用, 使原子电离, 产生电子 - 离子对, 这些电荷信号可以通过电场收集、放大, 形成清晰的电信号, 便于电子学系统测量。

(3) **可同时获取多种信息：**通过测量粒子在探测器中的能量损失

( $dE/dx$ ) 和偏转轨迹，可以同时得到粒子的动量、电荷、粒子种类等信息：

- 偏转轨迹结合磁场可以计算粒子动量
- 能量损失 ( $dE/dx$ ) 与粒子的速度、电荷有关，可用于粒子鉴别

(4) **适用范围广：**几乎所有带电粒子（如电子、 $\mu$  子、质子等）都会参与电磁相互作用，而中性粒子（如光子、中微子）不参与电磁相互作用，无法被径迹探测器直接探测，因此径迹探测器主要针对带电粒子的电磁相互作用进行测量。