实验物理中的统计方法 作业4

1.

习题 3.3. 考虑锯齿分布

$$f(x) = \begin{cases} \frac{2x}{x_{max}^2} & 0 < x < x_{max}, \\ 0 & \text{其它。} \end{cases} \tag{3.1}$$

(a) 参考课本 3.2 节，利用变换方法寻找函数 $x(r)$，以产生服从 $f(x)$ 的随机数。用程序实现该方法，并生成一个直方图。(可以取 $x_{max} = 1$。)

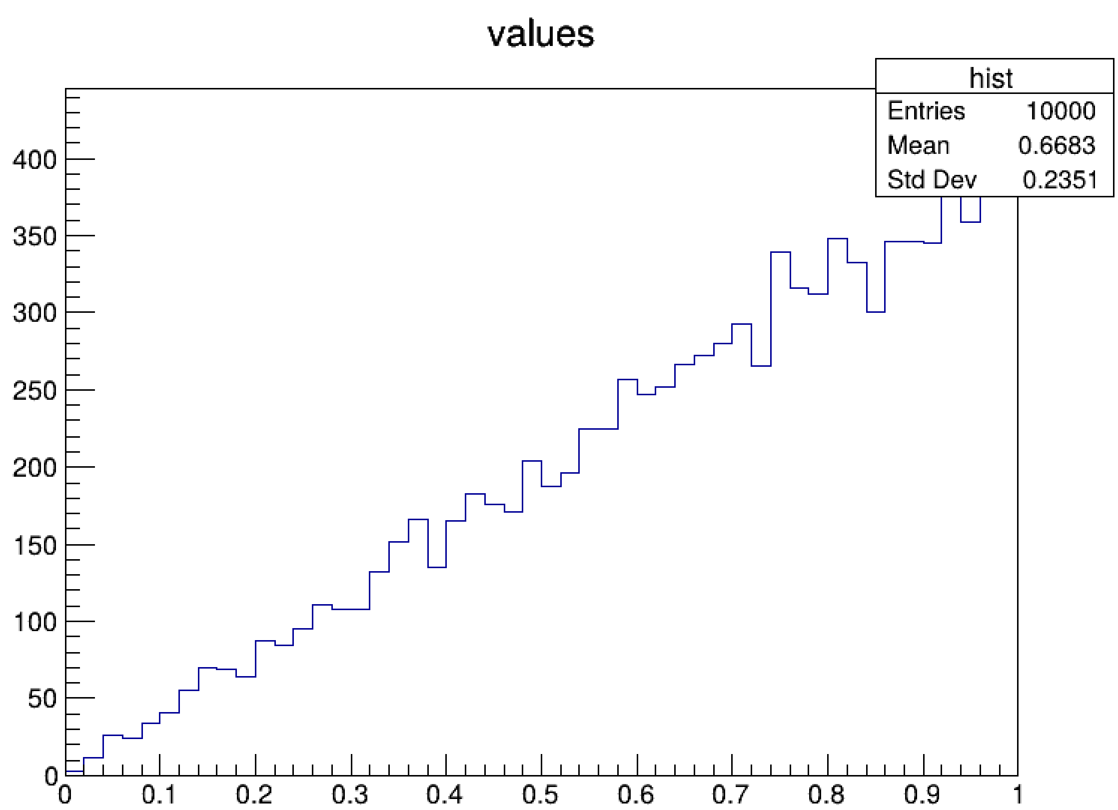
(b) 参考课本 3.3 节，编写一段程序，利用舍选法产生服从锯齿分布的随机数。画出相应的直方图。

(a)

Python代码实现:

```
# 变换法
import ROOT
import math
rnd = ROOT.TRandom3()
hist = ROOT.TH1F('hist', 'values', 50, 0, 1)
for i in range(10000):
    hist.Fill(math.sqrt(rnd.Rndm()))
canvas = ROOT.TCanvas('canvas', 'Canvas', 800, 600)
hist.Draw()
canvas.SaveAs('3.3.a.png')
```

直方图:

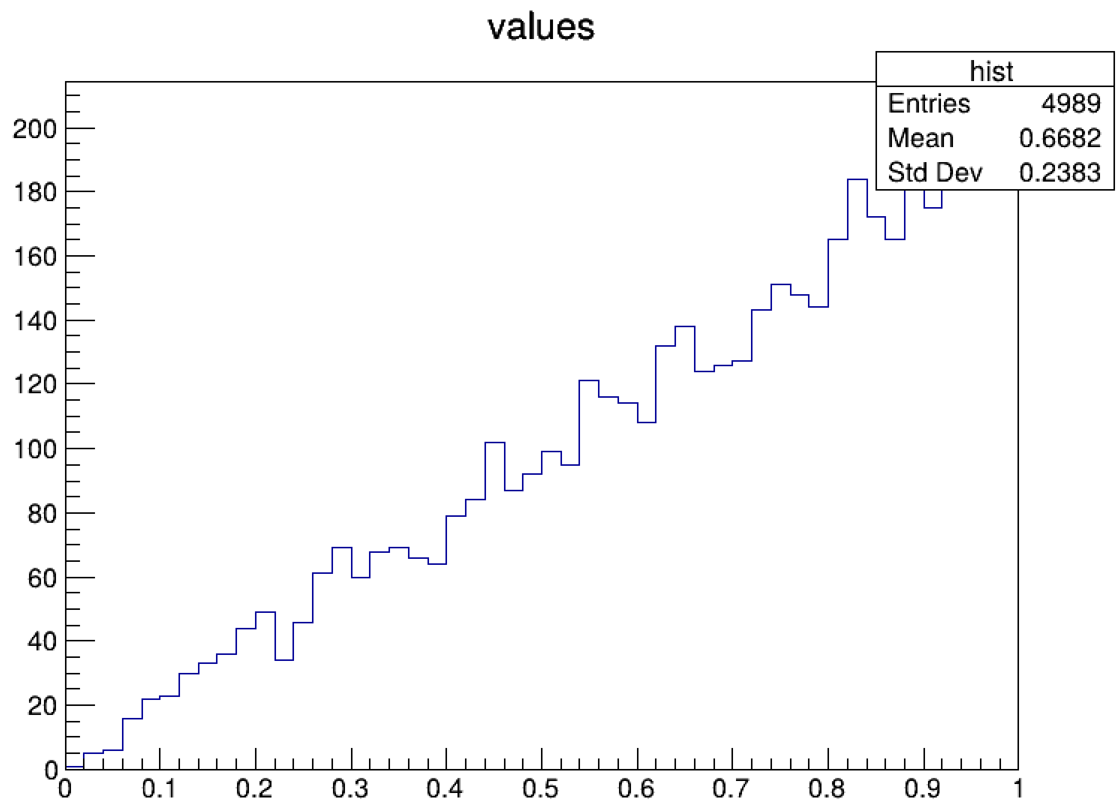


(b)

Python代码实现：

```
# 舍选法
import ROOT
rnd = ROOT.TRandom3()
def f(u, x):
    return u<2*x
hist = ROOT.TH1F('hist', 'values', 50, 0, 1)
for i in range(10000):
    x = rnd.Rndm()
    u = 2*rnd.Rndm()
    if f(u, x):
        hist.Fill(x)
canvas = ROOT.TCanvas('canvas', 'Canvas', 800, 600)
hist.Draw()

canvas.SaveAs('3.3.b.png')
```

直方图：



> 可以看出，变换法由于采样对于x方向的不均匀性导致x较大时采样少，偶然误差较大，MC法数据点没有得到全部应用导致相较而言随机误差较大。

2.

习题 3.5. 变量 t 服从均值 $\tau = 1$ 的指数分布，x 服从均值 $\mu = 0$，标准偏差 $\sigma = 0.5$ 的高斯分布。写一段 *MC* 程序，产生变量

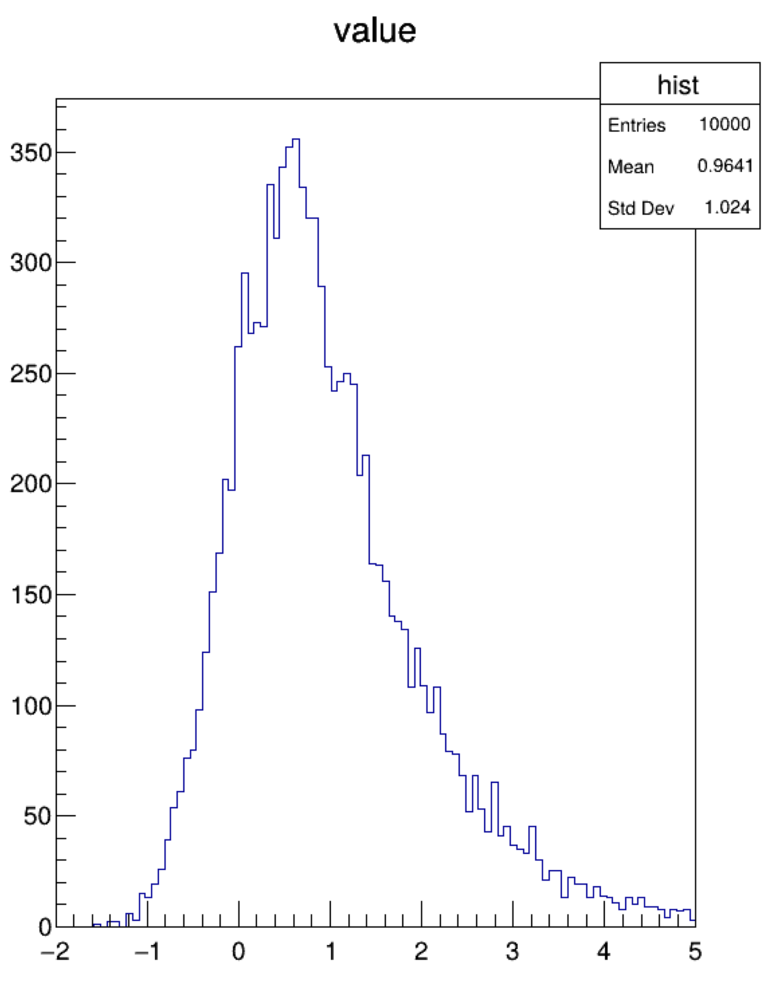
$$y = t + x. \tag{3.4}$$

这里 t 可以表示不稳定粒子的真实衰变时间，x 表示测量误差，所以 y 表示测量到的衰变时间。画出 y 的直方图。修改程序以研究 $\tau \ll \sigma$ 和 $\tau \gg \sigma$ 的情形。

正常参数Python代码实现：

```python
import ROOT
rnd = ROOT.TRandom3()
hist = ROOT.TH1F('hist', 'value', 100,-2,5)
for i in range(10000):
    t = rnd.Exp(1)
    x = rnd.Gaus(0, 0.5)
    y = t+x
    hist.Fill(y)

canvas = ROOT.TCanvas('canvas', 'Canvas', 600, 800)
hist.Draw()
canvas.SaveAs('3.5.png')
```

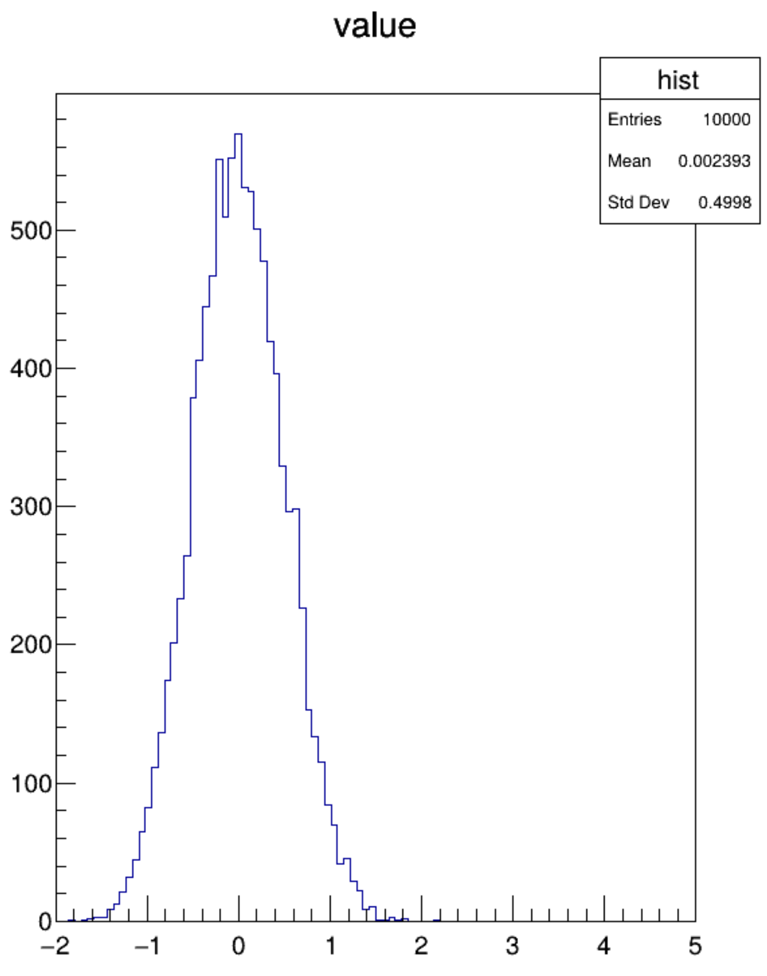
直方图:



取$\tau = 0.01, \sigma = 0.5$:

```python
import ROOT
rnd = ROOT.TRandom3()
hist = ROOT.TH1F('hist', 'value', 100,-2,5)
for i in range(10000):
    t = rnd.Exp(0.01)
    x = rnd.Gaus(0, 0.5)
    y = t+x
    hist.Fill(y)

canvas = ROOT.TCanvas('canvas', 'Canvas', 600, 800)
hist.Draw()
canvas.SaveAs('3.5.1.png')
```

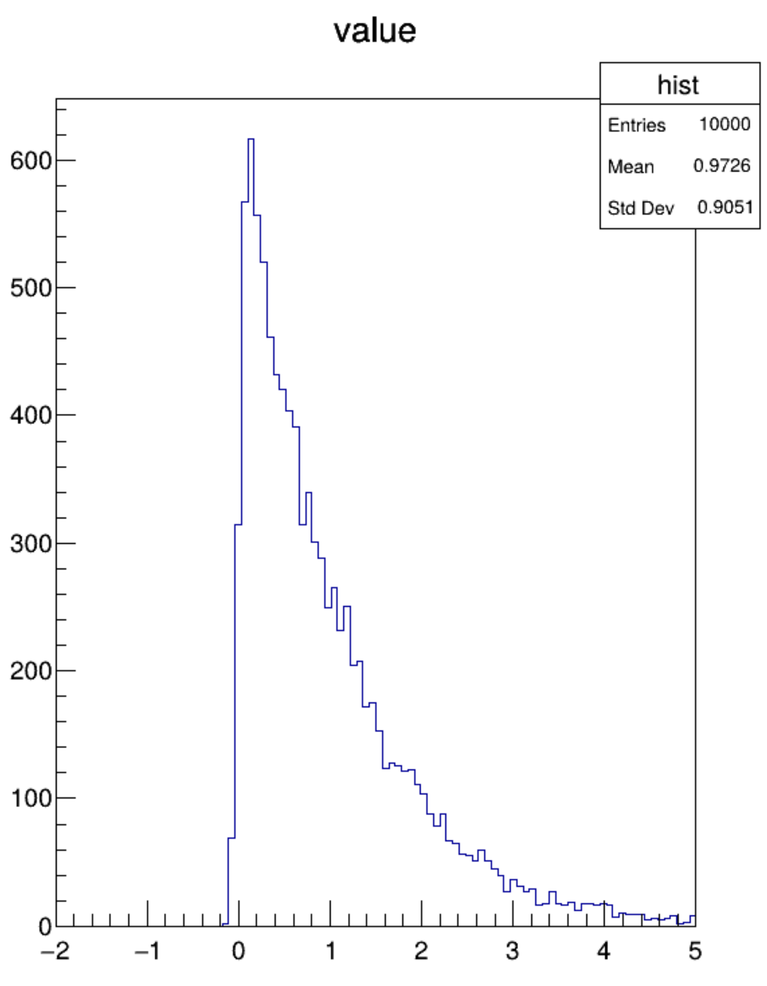
直方图:



取$\tau = 1, \sigma = 0.005$:

```python
import ROOT
rnd = ROOT.TRandom3()
hist = ROOT.TH1F('hist', 'value', 100,-2,5)
for i in range(10000):
    t = rnd.Exp(1)
    x = rnd.Gaus(0, 0.05)
    y = t+x
    hist.Fill(y)

canvas = ROOT.TCanvas('canvas', 'Canvas', 600, 800)
hist.Draw()
canvas.SaveAs('3.5.2.png')
```

直方图：



> 观察图像随参数的变化可以发现，当$\tau \ll \sigma$时，误差项占主导地位，图像基本呈现高斯分布，而当$\tau \gg \sigma$时，数据项占主要地位，呈现指数分布。

3.

习题 3.6. 考虑服从 *Cauchy(Breit-Wigner)* 分布的随机变量 x，

$$f(x)=\frac{1}{\pi}\frac{1}{1+x^2}. \tag{3.5}$$

(a) 证明如果 r 均匀分布于 $[0,1]$，则

$$x(r)=\tan[\pi(r-\frac{1}{2})] \tag{3.6}$$

服从 *Cauchy* 分布。

(b) 利用 *(a)* 中的结果，写一段小程序产生 *Cauchy* 分布的随机数。产生 10^4 个事例并画出直方图。

(c) 修改 *(b)* 中的程序，重复进行实验，每次实验 n 个独立的柯西分布数值（如取 $n=10$）。对每个样本，计算样本均值 $\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_i$。比较 $\bar{x}$ 的直方图与 x 的原始直方图。（参见习题 *10.8*。）

解:

(a)

由概率守恒:

$$f(x)\mathrm{d}x=g(r)\mathrm{d}r=\mathrm{d}r$$

$$f(x)=\left|\frac{\mathrm{d}r(x)}{\mathrm{d}x}\right|$$

$$r=\frac{\arctan x}{\pi}+\frac{1}{2}$$

$$\frac{\mathrm{d}r}{\mathrm{d}x}=\frac{1}{\pi}\frac{1}{1+x^2}$$

$$f(x)=\frac{1}{\pi}\frac{1}{1+x^2}$$

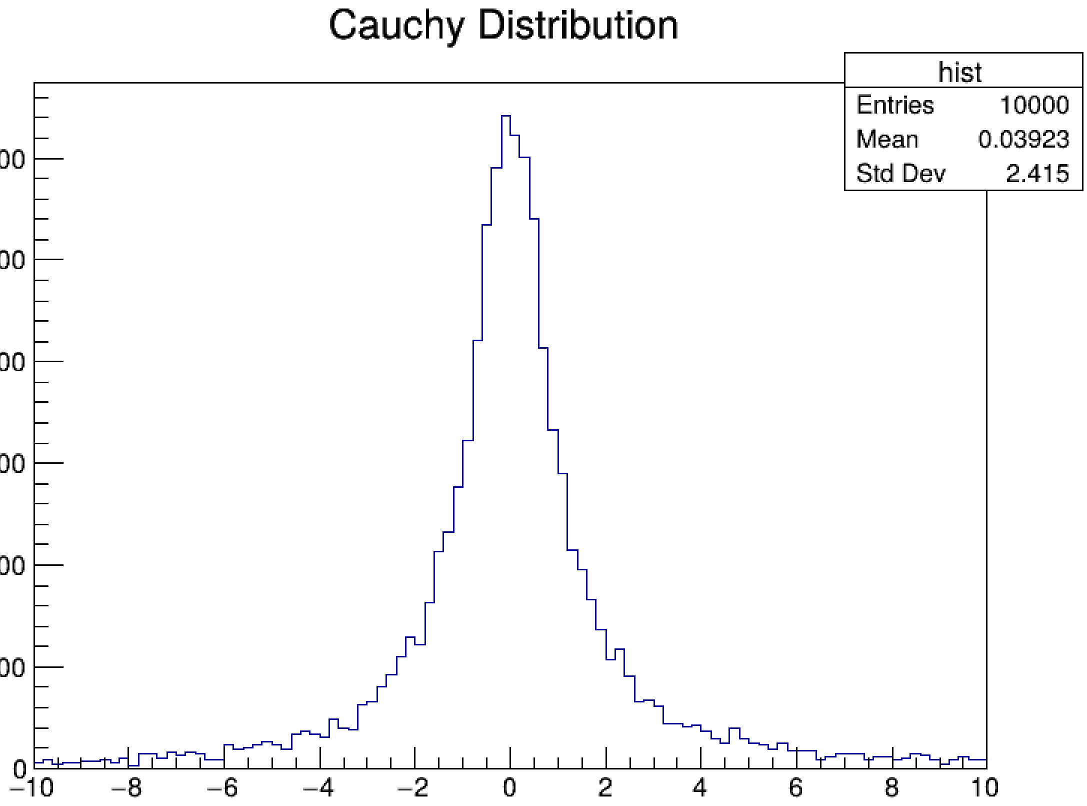
证毕。

(b)

Python代码实现:

```python
import ROOT
import math
rnd = ROOT.TRandom3()

hist = ROOT.TH1F('hist', 'Random Numbers', 100, -10, 10)

def f(x):
    return math.tan(math.pi*(x-0.5))

for i in range(10000):
    hist.Fill(f(rnd.Rndm()))

canvas = ROOT.TCanvas('canvas', 'Canvas', 800, 600)
hist.Draw()

canvas.SaveAs('3.6.png')
```

直方图:

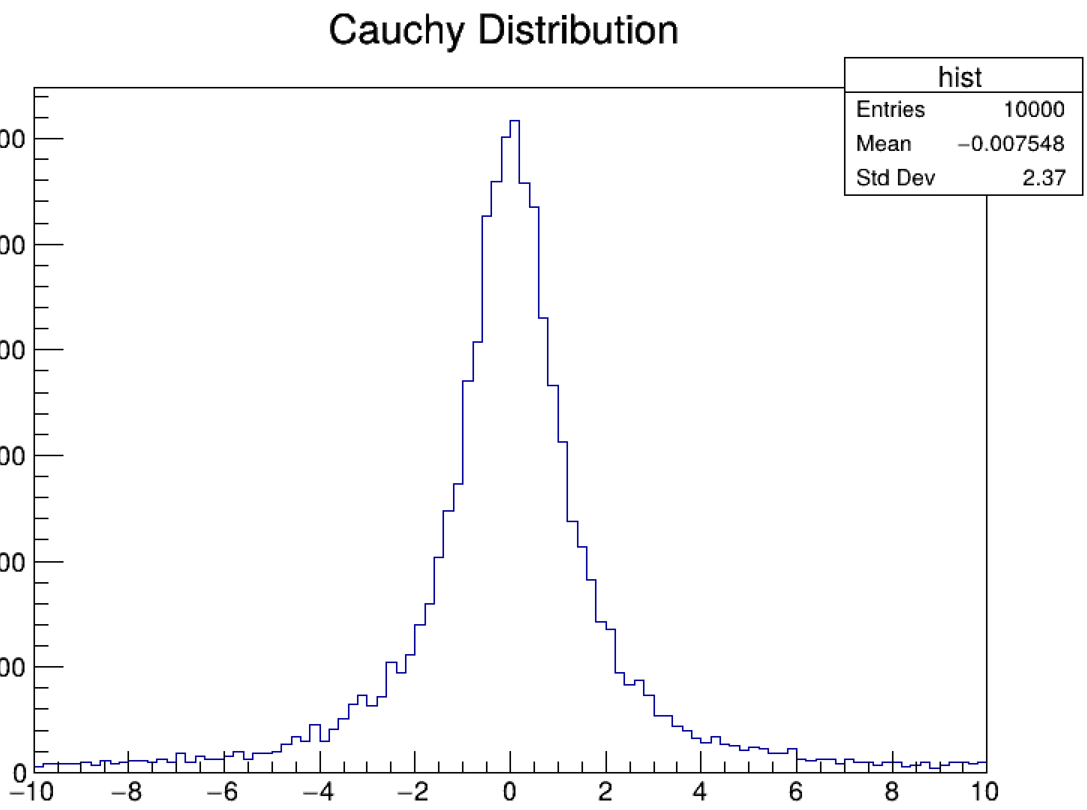


(c)

Python代码实现:

```
import ROOT
import math
rnd = ROOT.TRandom3()

hist = ROOT.TH1F('hist', 'Cauchy Distribution', 100, -10, 10)

def f(x):
    return math.tan(math.pi*(x-0.5))

for i in range(10000):
    sumx = 0
    for j in range(10):
        sumx += f(rnd.Rndm())
    hist.Fill(sumx/10)

canvas = ROOT.TCanvas('canvas', 'Canvas', 800, 600)
hist.Draw()

canvas.SaveAs('3.6.1.png')
```

直方图:



> 发现随着n增大，大数定律并不成立，分布仍然呈现 Cauchy 分布的形状。这是由于 Cauchy 分布的方差为无穷大导致的。

4.

习题 3.8. 假设二维随机变量 (r, θ) 表示二维平面上的某点的极坐标。写一段程序，产生 1000 对 (r, θ)，使其代表的点在以圆点为圆心的单位圆内均匀分布。

思路：

均匀分布意味着

$$f(r)\mathrm{d}rg(\theta)\mathrm{d}\theta = c \cdot r\mathrm{d}r\mathrm{d}\theta$$

故知：

$$f(r) = c_1 r$$
$$g(\theta) = c_2$$

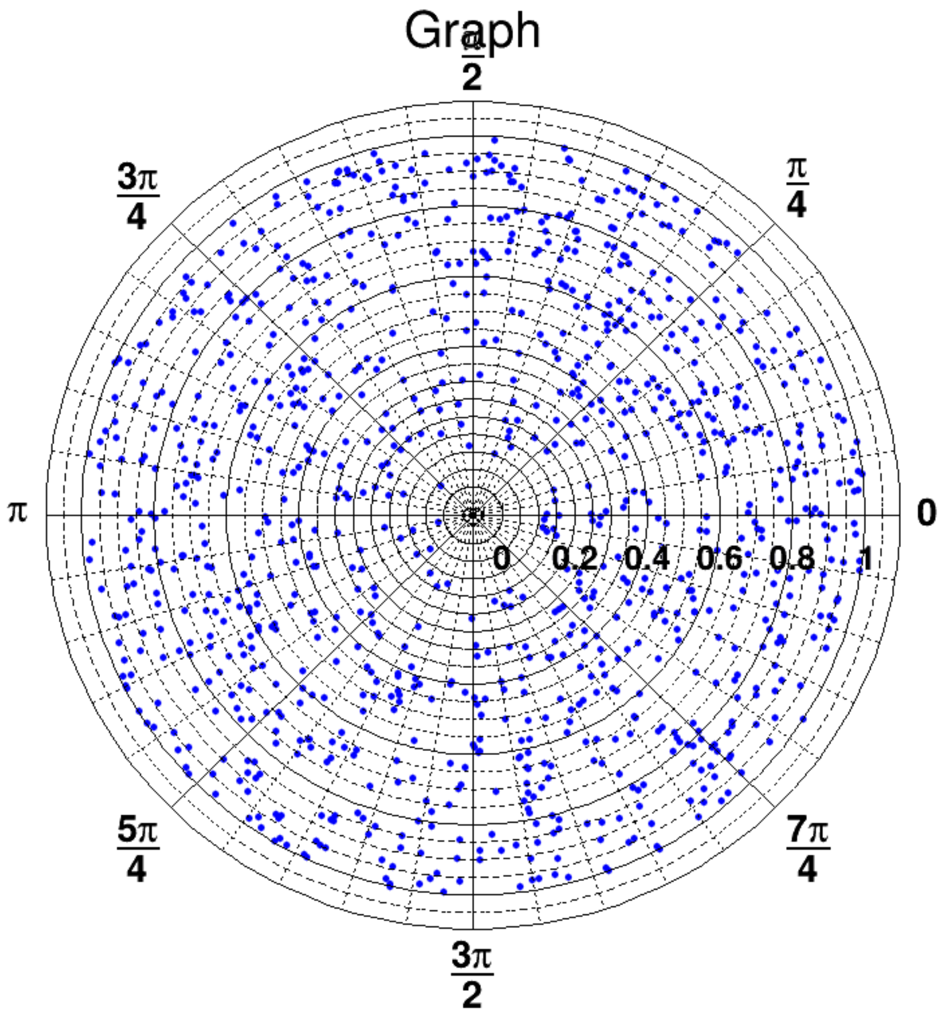
可以保证平面上的点均匀分布。

Python代码实现：

```
import ROOT
import math
radius = []
rnd = ROOT.TRandom3()
def f(x):
    return x
while True:
    u = rnd.Rndm()
    x = rnd.Rndm()
    if u <f(x):
```

```
        radius.append(x)
    if len(radius) == 1000:
        break
theta = []
for i in range(1000):
    theta.append(2*math.pi*rnd.Rndm())

graph = ROOT.TGraphPolar(1000)
for i in range(1000):
    graph.SetPoint(i, theta[i], radius[i])
canvas = ROOT.TCanvas("canvas", "Polar Scatter Plot", 800, 800)
graph.SetMarkerStyle(20)
graph.SetMarkerSize(0.6)
graph.SetMarkerColor(ROOT.kBlue)
graph.Draw("P")
canvas.Update()
canvas.Draw()
canvas.SaveAs("3.8.png")
```

直方图:



5.

习题 3.9. 写一段程序，用蒙特卡罗方法计算下面的定积分：

$$\int_0^1 \frac{e^{-x}}{\sqrt{x}} \mathrm{d}x$$

思路：

注意到：

$$0<\frac{\mathrm{e}^{-x}}{\sqrt{x}}<\frac{1}{\sqrt{x}}\quad,x\in(0,1)$$

而积分：

$$\int_0^1\frac{1}{\sqrt{x}}\mathrm{d}x=2$$

故知积分有限。

利用代换$t=\sqrt{x}$即可将积分区域上界化为有限值。

Python代码实现：

```python
import math
import ROOT

def f(x):
    return 2*math.e**(-1*x**2)
rnd = ROOT.TRandom3()
cnt = 0
for i in range(100000):
    u = 2*rnd.Rndm()
    x = rnd.Rndm()
    if u<f(x):
        cnt += 1
print(cnt/100000*2)
"""
result: 1.4936
"""
```

这一结果与数值积分的结果相同。

In[1]:= $\text{Integrate}\left[\frac{E^{-x}}{\sqrt{x}},\ \{x,\ 0,\ 1\}\right]$
积分

Out[1]= $\sqrt{\pi}\ \text{Erf}[1]$

In[2]:= $N\left[\sqrt{\pi}\ \text{Erf}[1]\right]$
数⋯ 误差函数

Out[2]= 1.49365