

云计算用于 G - M 计数器实验的数据处理

张润生 程敏熙 徐永康

(华南师范大学物理与电信工程学院 广东 广州 510006)

(收稿日期:2021-12-17)

摘要:基于云计算原理,用 Python 及 Flask 设计了一个辅助 G - M 计数器及放射性衰变统计规律实验数据处理的程序,可不受地域及设备的限制快速实现数据计算、G - M 计数管坪曲线和放射性衰变统计规律绘图等功能.

关键词:云计算 Python Flask 数据处理 G - M 计数器

近年来,由于低硬件成本、按需服务、动态调配、安全方便的特点,云计算平台成为辅助高校物理实验教学的重要资源^[1].运用云计算方法来辅助高校近代物理实验数据处理,不仅可以提高数据处理的速率和质量,还可以发展学生现代信息技术运用的能力^[2].

本文基于云计算原理,利用 Python 及 Flask 设计了一款辅助 G - M 计数器及放射性衰变统计规律实验数据处理的程序,实现了数据表格呈现、数值计算及绘图等功能.该程序可使用任意移动终端,通过人机交互输入实验数据,实时进行数据处理并输出数据处理结果.

1 实验原理及实现条件

1.1 Python 语言及 Flask 框架

Python 是一种面向对象、支持动态语义、内置高级数据结构、语法简洁优美、易于扩展的解释型脚本语言^[3],其包含的模块和扩展库提供了各种问题的解决方法,具有易学、易读、易维护等特点^[4].本文的编程语法采用 Python 语言.

Flask 是一种使用 Python 语言编写的 Web 开发框架,可以在短时间内实现功能丰富的中小型网站或 Web 服务,具有灵活、轻便、安全且容易上手等特点^[5,6].本文采用 Flask - WTF 作为表单设计的第三

方库,调用 Boken 库实现数据可视化.

1.2 G - M 计数器及 G - M 管的坪曲线

G - M 计数器是盖革(Geiger,1882 - 1945)和米勒(Muller,1905 - 1979)发明的一种专门探究电离辐射(α 粒子、 β 粒子、 γ 射线)强度的气体电离探测器,具有结构简单、易于加工、输出信号幅度大、使用方便、成本低等特点^[7].如图 1 所示,G - M 计数器由 G - M 计数管、高压电源及定标器组成.

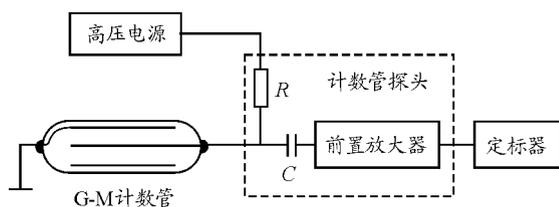


图 1 G - M 计数器实验装置图

在强度不变的放射源照射下,G - M 计数管的计数率 n 与计数时间 t 、计数 N 的关系为

$$n = \frac{N}{t} \quad (1)$$

计数率 n 随外加电压 V 变化的坪曲线如图 2 所示,其中坪长 L 为

$$L = V_2 - V_1 \quad (2)$$

坪坡度 T 为

$$T = \frac{n_2 - n_1}{\frac{1}{2}(n_1 + n_2)(V_2 - V_1)} \times 10^4 \quad (3)$$

作者简介:张润生(1997 -),男,在读硕士研究生,研究方向为学科教学(物理).

通讯作者:程敏熙(1962 -),男,博士,副教授,研究方向为光电技术与系统、物理实验设计.

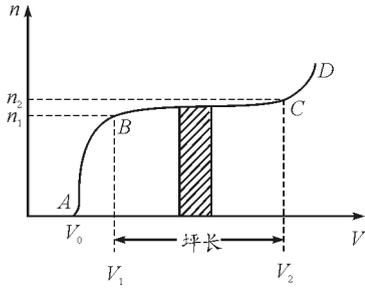


图2 G-M计数管的坪曲线

1.3 核衰变的统计规律

原子核放射性衰变过程属于随机事件,而大量的随机过程服从统计分布规律^[7,8].

如图3(a)所示,若某时间间隔内的平均计数 \bar{N} 比较小时,则某次测量相同时间间隔的计数为 N 的概率 $P(N)$ 服从不对称的泊松分布,即

$$P(N) = \frac{(\bar{N})^N}{N!} \times e^{-\bar{N}} \quad (4)$$

如图3(b)所示,当 \bar{N} 比较大时,泊松分布可以用高斯分布来替代,即

$$P(N) = \frac{1}{\sqrt{2\pi N}} \times e^{-\frac{(\bar{N}-N)^2}{2N}} \quad (5)$$

泊松分布和高斯分布的标准差 σ 满足

$$\sigma = \sqrt{\bar{N}} \approx \sqrt{N} \quad (6)$$

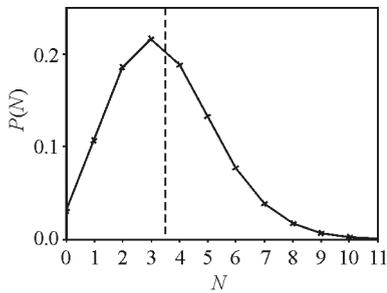
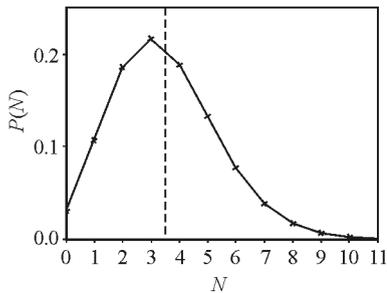
(a) $\bar{N}=3.5$ 时的泊松分布(b) $\bar{N}=20$ 时的高斯分布

图3 泊松分布与高斯分布曲线

2 程序编程与实验数据处理

2.1 测量G-M计数管的坪曲线

输入实验条件下显示的电压示数及对应的计数值,本程序在线处理绘制出G-M计数管的坪曲线,以此计算G-M计数管的坪长度、坪坡度,确定阈电压和适宜的工作电压范围.

(1) 以下是程序算法的部分代码:

```
V = [] # 输入电压数据
counts = [] # 输入计数值
count_rates = [] # 计算计数率并形成列表
for count in counts:
    count_rate = count * 3 / 5
    count_rates.append('% .3f' % count_rate)
z1 = np.polyfit(V, count_rates, 4) # 多项式拟合
p1 = np.poly1d(z1)
yvals = p1(V)
```

此部分为G-M计数管坪特性曲线的绘制提供数据支持,本实例的数据处理结果如表1所示.

表1 测量G-M计数管坪曲线的数据

序号	电压 V/伏	计数值 N	计数率 n/mm ⁻¹	序号	电压 V/伏	计数值 N	计数率 n/mm ⁻¹
1	320	402	241.2	8	460	1009	605.4
2	340	963	577.8	9	480	980	588.0
3	360	942	565.2	10	500	987	592.2
4	380	896	537.6	11	520	1050	630.2
5	400	921	552.6	12	540	1045	627.0
6	420	965	579.0	13	560	1165	699.0
7	440	994	596.4				

(2) 以下是实现数据可视化的部分代码:

```
plot1 = plt.plot(V, count_rates, '* ', label = 'Raw data')
plot2 = plt.plot(V, yvals, 'r', label = 'Curve fitting')
plt.xlabel('V/伏', fontproperties = 'SimHei')
plt.ylabel('n/min', fontproperties = 'SimHei')
plt.show()
```

此部分代码调用 Bokeh 方法,将计算结果转化为图像,实现平台上数据的可视化.由图 4 可知,本实例的阈电压为 320 V,适合工作频率范围的电压为 340 ~ 540 V,坪长度约为 200 V,坪坡度为

$$T = \frac{n_2 - n_1}{\frac{1}{2}(n_1 + n_2)(V_2 - V_1)} \times 10^4 = \frac{(627.0 - 577.8) \times 10^4}{\frac{1}{2}(627.0 + 577.8)(540 - 340)} = 4.08\% / (100 \text{ V}) \quad (7)$$

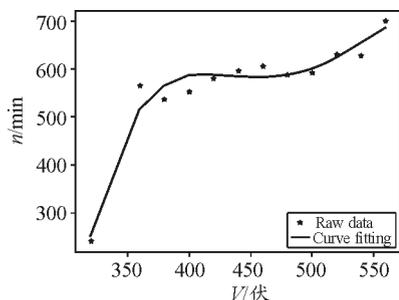


图 4 系统生成的 G-M 计数管坪特性曲线

2.2 验证泊松分布规律

不需要放射源,选定工作电压 440 V,计数时间 1 s,测量本底计数,每次计数值均小于 10,重复测量 300 次以上,输入实验测得的计数值及对应次数,本程序在线处理绘制出实验图像和理论图像,实验结果基本服从泊松分布.

(1) 以下是程序算法的部分代码:

```
cnt=[] # 计数值列表
listNs=[] # 对应次数列表
theoreticalPNs=[]
averN=0
N=np.array(listNs).sum() # 总测量次数
cnts=np.array(cnt)
truePNs=listNs/N # 求 P(N) 的实际值
for i in range(0,len(cnt)): # 求计数值的平均值
averN=averN+listNs[i]*cnts[i]/N
for i in cnts: # 求理论概率
theoreticalPN=((averN)**i)*(np.exp(-averN))/np.math.factorial(i)
```

theoreticalPNs.append(theoreticalPN)

用户可通过导入 Excel 表格或手动输入两种方式录入数据,此部分对输入的数据进行在线处理并获得加权平均值、标准差、概率实验值和理论值等数据.本实例的数据处理结果如图 5 所示,数据处理结果可以通过 Excel 形式导出保存.

← → ↺ 127.0.0.1:5000/poisson/table

泊松分布实验数据记录表

平均值 = 2.588 标准差 = 1.609

测量值	出现次数	实际概率	理论概率
0	32	0.104	0.075
1	67	0.218	0.195
2	56	0.182	0.252
3	63	0.205	0.217
4	46	0.149	0.140
5	27	0.088	0.073
6	12	0.039	0.031
7	3	0.010	0.012
8	1	0.003	0.004
9	1	0.003	0.001

导出excel

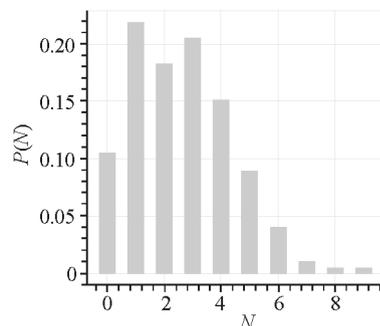
绘图

图 5 验证泊松分布实验的数据处理结果

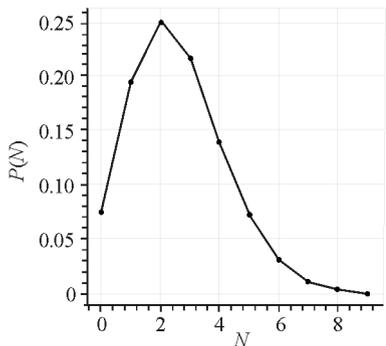
(2) 以下是实现数据可视化的部分代码:

```
P=figure(title="实测分布图",x_axis_label="N",y_axis_label="P(N)")
p1=figure(title="理论分布图",x_axis_label="N",y_axis_label="P(N)")
p2=figure(title="对比图",x_axis_label="N",y_axis_label="P(N)")
r=gridplot(children=[[p,p1],[p2]])
show(r)
```

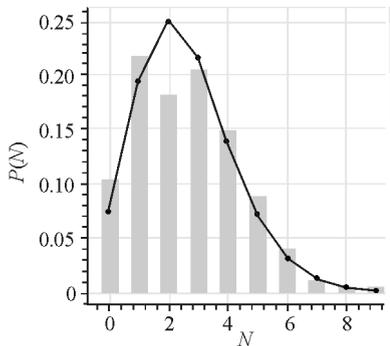
当用户点击“绘图”按钮后,程序会呈现该实验值的实测分布图、理论分布图以及以上两幅图的对比图,以使用户验证实验结果是否服从泊松分布.本实例处理得到图像如图 6 所示.



(a) 实测分布图



(b) 理论分布图



(c) 对比图

图 6 实测曲线与泊松分布理论曲线的对比

2.3 验证高斯分布规律

选择工作电压 440 V, 放置放射源, 计数时间 1 s, 每次计数大于 20, 本实验每次的计数值约 300/s, 重复测量 500 次以上, 输入实验条件下测得的所有计数值, 本程序在线处理绘制出实验图像和理论图像, 实验结果基本服从高斯分布。

(1) 以下是程序算法的部分代码:

```
def large (list, big) : # 分组函数
list1 = list  a=[]  b=[]
for i in range (0, len(list1)):
    if list1[i] >= big :
        for j in range (i, len(list1)):
            b. append(list1[j])
        break
else:
    a. append(list[i])
return a, b
def fung(avg, Q=[]): # 高斯分布计算
N = np. array(Q)
tpn = np. exp((- (N - avg) * * 2)/(2 * avg))/
```

```
((avg * (2 * np. pi)) * * 0.5)
```

```
return tpn
```

本程序以 $\frac{\sigma_N}{2}$ 为组距, 将所有数据进行分组, 计算各分组的频数、频率以及高斯分布的理论频率, 整体显示平均值 (avg)、总测量次数、标准差 (W)、组距 $\left(\frac{W}{2}\right)$ 的计算结果, 计算出 $avg \pm W$ 、 $avg \pm 2W$ 、 $avg \pm 3W$ 三个区间的概率。

如图 7 所示, 实验数据提交后, 程序在网页上以表格的形式呈现数据处理结果. 在本实例中, $avg \pm W$ 、 $avg \pm 2W$ 、 $avg \pm 3W$ 三个区间的概率分别为 0.6980, 0.9451, 0.9980, 与高斯分布的理论概率 68.5%, 95.4%, 99.7% 相近。



图 7 验证高斯分布实验的数据处理结果

(2) 以下是实现数据可视化的部分代码:

```
p = figure(title = ' 频率分布直方图 ',
x_axis_label
=' 计数率 N', y_axis_label = ' 频率 / 组距 ')
p1 = figure(title = " 理论正态分布图 ",
x_axis_label
=" 计数率 N", y_axis_label ="P(N)",)
p2 = figure(title = " 对比图", x_axis_label = "
计数率 N", y_axis_label ="P(N)")
r = gridplot(children = [[p, p1], [p2]])
show(r)
```

当用户点击“绘图”按钮后, 如图 8 所示, 程序会生成该实验的实验值分布图、理论分布图以及对比

图,直观显示实验结果是否服从高斯分布.

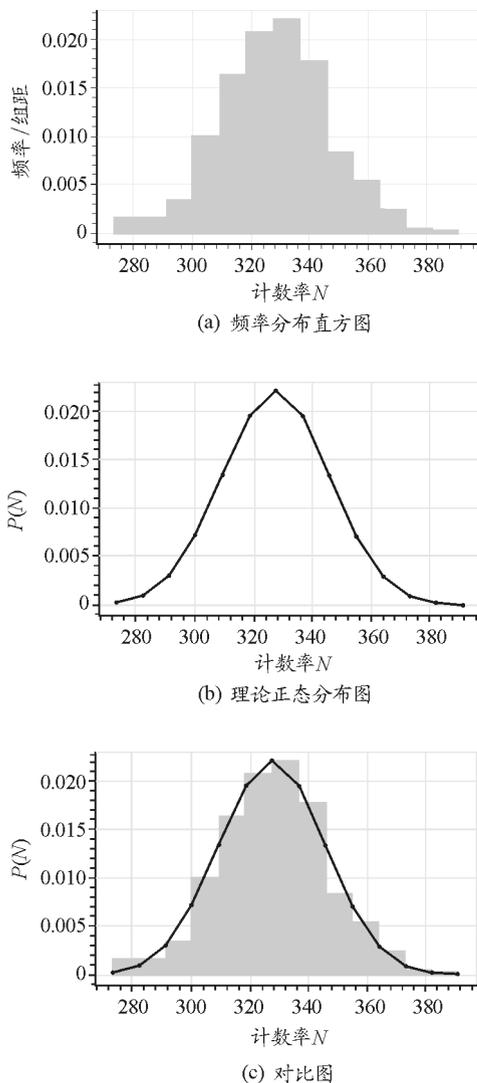


图8 实测曲线与高斯分布理论曲线的对比

3 结论

本文基于云计算平台,用 Python 及 Flask 设计了一个辅助盖革-米勒计数器及放射性衰变统计规律实验数据处理的程序,不同类型终端都可以使用该平台辅助实验教学,极大提高了师生教学实践和学习的效率和质量,实现了基于云平台的大学物理实验教学资源的功能拓展.

参考文献

- 1 孙柏祥. 云计算——高校教育信息化建设和发展的新模式[J]. 中国电化教育, 2010(5): 123 ~ 125
- 2 林瑜华. 云计算环境下高校实验教学模式的创新与实践[J]. 实验室研究与探索, 2011, 30(8): 271 ~ 274
- 3 李金. 自学 Python: 编程基础、科学计算及数据分析[M]. 北京: 机械工业出版社, 2018
- 4 黄六霞, 程敏熙, 林奇川. 基于云平台的 Lorenz 混沌数值模拟实验[J]. 物理实验, 2020, 40(2): 54 ~ 58
- 5 林奇川, 程敏熙. 基于云计算的大学物理实验数据在线处理[J]. 大学物理, 2019, 38(4): 28 ~ 32
- 6 朱贲. Python 语言的 Web 开发应用[J]. 电脑知识与技术, 2017, 13(32): 95 ~ 96
- 7 吴先球. 近代物理实验教程(第2版)[M]. 北京: 科学出版社, 2009. 79 ~ 85
- 8 吴思诚, 王祖铨. 近代物理实验(第2版)[M]. 北京: 北京大学出版社, 1995. 98 ~ 100
- 9 关毅铭, 程敏熙. 基于 Python 和梯度下降算法的物理实验数据一元线性拟合方法[J]. 物理通报, 2019(10): 92 ~ 95

Cloud Computing Data Processing of G - M Counter Experiment

Zhang Runsheng Cheng Minxi Xu Yongkang

(School of Physics and Telecommunication Engineering, South China Normal University, Guangzhou, Guangdong 510006)

Abstract: Based on the principle of cloud computing, a program for assisting G - M counter and radioactive decay statistical law experimental data processing was designed with python and Flask, which can quickly realize functions such as data calculation, G - M counting pipe plate curve and radioactive decay statistics drawing.

Key word: cloud computing; Python; Flask; data processing; G - M counter