

# 用电容 -电压法测量半导体中的杂质分布

黄 康 靖

2012 级 学号:masked student id<sup>\*</sup>

(日期: April 16, 2015)

在现代半导体物理学中, 掺杂半导体中的杂质分布是半导体材料的一个重要属性. 半导体工艺的相关设计和原理都依赖于这个属性. 本实验利用锁定放大器放大小电压信号, 使用电容 -电压法原理, 测得了  $p^+$ -n 结中 n 型半导体一侧的半导体材料杂质分布曲线.

**关键词:** 半导体, 杂质分布, 锁定放大器, 电容, 电压

---

<sup>\*</sup> [huangkangjing@gmail.com](mailto:huangkangjing@gmail.com)

## I. 引言

半导体物理学中, 半导体器件设计与制造的核心问题是如何控制半导体内的杂质分布, 以满足半导体器件实际应用的需要. 基于这个原因, 正确地检测杂质浓度及其分布是制备合格半导体器件的必要条件. 所以杂质的浓度及其分布的测量是半导体材料及器件研究中的基本测量之一. 通过测量不同直流偏压下  $p-n$  结势垒电容, 可以在不破坏器件本身的情况下比较迅速地测得单边突变  $p-n$  结中, 轻掺杂一侧的半导体的杂质浓度及其分布.

本实验利用上述的这一  $C-V$  法, 使用锁定放大器与函数信号记录仪等仪器, 测量了  $p^+-n$  结轻掺杂一侧的杂质浓度及其分布.

## II. 原理

### A. $p-n$ 结的势垒电容与杂质浓度的关系

我们知道, 当  $n$  型半导体与  $p$  型半导体在某一界面上相互接触时, 由于两种半导体中的载流子种类与符号不同,  $p$  型半导体中的空穴将会扩散到空穴较少的  $n$  型半导体去, 而  $n$  型半导体中的电子将会扩散到空穴较少的  $p$  型半导体中去. 然而, 在交接面上, 两种载流子会相互湮灭, 从而产生没有载流子, 有空间净电荷分布的耗尽区. 耗尽区内的电场使得载流子在其中作漂移运动. 这两种因素相互竞争的结果即是耗尽区的宽度达到了一个动态平衡, 即为  $p-n$  结的宽度.

当空间电荷区域宽度变化  $dw$  时, 相应的单位面积的空间电荷变化量

$$dQ = qN(w)dw \quad (1)$$

其中  $N(w)$  是空间电荷区域宽度  $w$  边界处的杂质浓度, 增加的电荷  $dQ$  将引起电场改变  $dE$ , 由泊松方程可以求得

$$dE = \frac{dQ}{\epsilon\epsilon_0} \quad (2)$$

相应地

$$d(V_R + V_D) = dV_R = wdE \quad (3)$$

以上两式消去  $dE$ , 得

$$\frac{dQ}{dV_R} = \frac{\epsilon\epsilon_0}{w} = \frac{C}{A} \quad (4)$$

整理得到

$$\begin{aligned} N(w) &= \frac{2}{q\epsilon\epsilon_0 A^2} \frac{1}{\frac{d(1/C^2)}{dV_R}} \\ &= - \frac{1}{q\epsilon\epsilon_0 A^2} \frac{C^3}{dC/dV_R} \end{aligned} \quad (5)$$

对于  $p^+ - n$  结来说, 我们有关系  $N_A \gg N_D$ , 那么  $x_n \gg x_p$ , 于是总空间电荷区宽度  $w = x_n + x_p \approx x_n$ , 那么, 我们得到的  $N(w)$  分布近似上就是  $n$  侧半导体内杂质浓度随深度的变化分布.

从此式我们可以看到, 只要求得 C-V 曲线, 按式 5 再结合式 4 即可算出杂质的分布情况.

## B. 势垒电容的测量

$p-n$  结势垒电容是一个随直流偏压变化的微分电容, 测量时采取的方法是在  $p-n$  结上偏置一定的反向直流偏压  $V_R$ , 同时再在  $V_R$  上叠加一个微小的交变电压信号  $v(t)$ , 这信号经过待测的  $p-n$  结电容  $C_x$  和一个与之相串接的已知电容  $C_0$ , 当  $C_0 \gg C_x$  时, 在  $C_0$  两端的电压为:

$$v_i = \frac{v(t)}{\frac{1}{j\omega C_x} + \frac{1}{j\omega C_0}} \cdot \frac{1}{j\omega C_0} \approx \frac{C_x}{C_0} v(t) \quad (6)$$

通过这一线性关系, 即可测得  $C_x$  的值.

### III. 实验

实验的测量方框图如图 1 所示. 图中的单刀双掷开关是为了切换测量实际容值和校准锁定放大器的测量信号输出而设计的. $D_1, D_2$  两个二极管和旁的电容组成的电路是锁定放大器的输入保护电路.

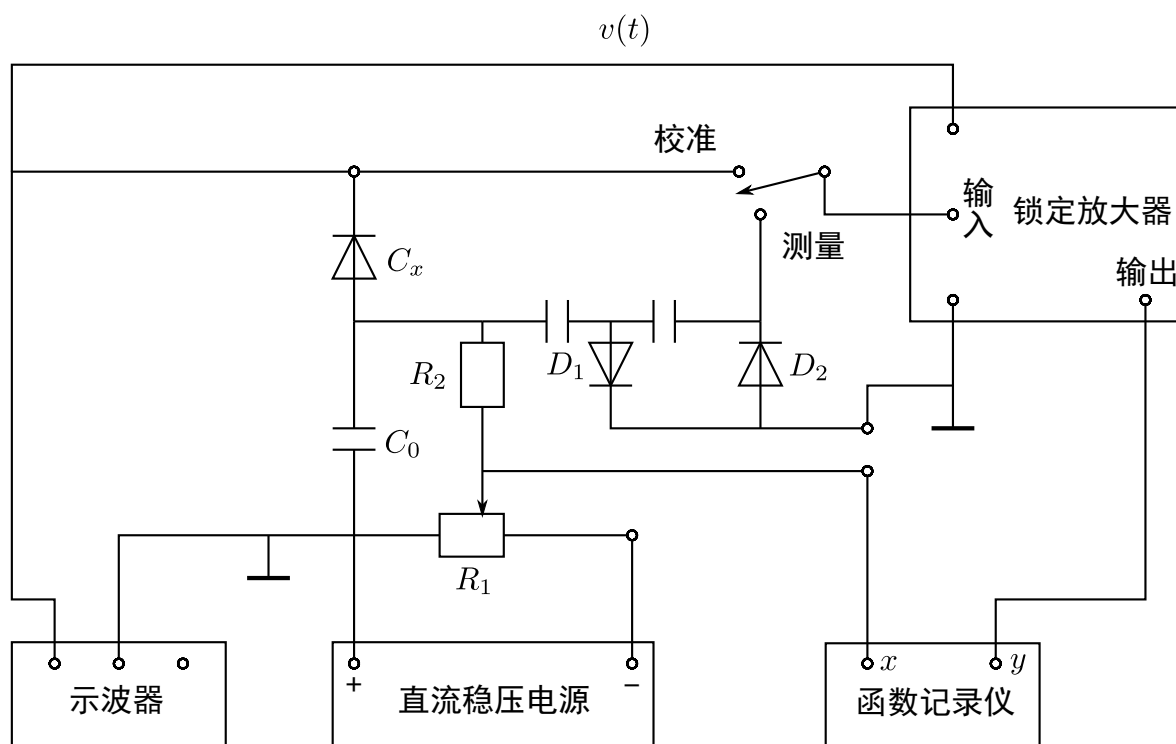


图 1: 实验装置的测量方框图, 图中锁定放大器型号为 Model 128 Lock-in Amplifier

在本实验中, 锁定放大器主要起到从噪声中提取固定频率的信号, 并且将其放大到可以测量的幅度的作用.

我们知道, 在半导体电子学实验中, 由于实验器材和实验素材的不可避免的缺陷和涨落因素, 在电路中会存在各种噪音, 影响实验的测量. 本次实验中, 由于实验的测量信号  $v_i$  十分微弱, 这一问题显得尤为显著. 而锁定放大器是一种交流电压表, 它通过相敏检波器和低通滤波器, 实现了测量深埋在噪声之中的周期重复信号的幅值与相位的能力, 本实验正是利用它的这一能力, 进行实验数据的测量.

本实验分以下三个部分进行:

- a. 通过若干个标准电容的测量, 证实了锁定放大器的输出与被测电容之间的线性

关系

- b. 实验的二极管样品有若干由于在反向加电压时会被击穿, 从而不适合测量. 击穿现象表现为在反向加电压时由于反向漏电流较大且随加的反向电压变化, 从而导致其相位会发生变化. 实验甄别了可用于测量的二极管.
- c. 实验测量了选定二极管样品的 C-V 特性曲线, 从而在此基础上计算出了杂质分布曲线.

## IV. 实验结果及分析讨论

### A. 锁定放大器输出电压与被测电容之间的线性关系

这一线性关系的证实性测量结果如图 2 所示.

从图表中可以看出, 证实性测量基本证实了这两个物理量之间的线性关系, 并且这一线性关系吻合的非常好, 这意味着我们的测量系统是满足条件  $C_0 \gg C_x$  的

### B. 可用二极管的确定

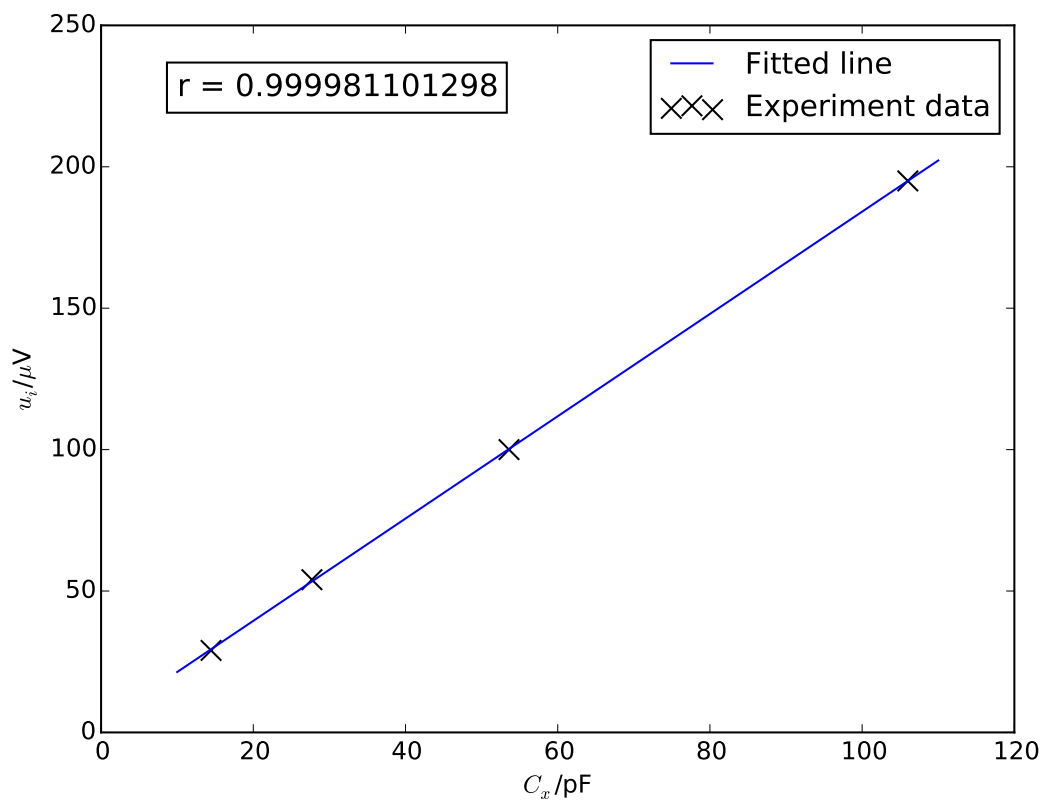
经过相位 - 反向电压关系的测量, 我们确定了一号和二号二极管的端电压相位是不基本不随电压值变化的, 可用于本实验测量.

### C. 杂质分布的测定

我们选定二号二极管进行杂质分布的测量, 首先, 我们测得二号二极管上的电容电压关系如图 3 所示.

基于测得的电容电压关系图线, 可以得到电容对电压的导数随电压的变化曲线如图 4 所示.

最后, 基于式 5 我们可以得到待测样品中杂质的分布图线如图 5 所示.



**图 2:** 本图是锁定放大器输出电压与被测电容之间的线性关系的验证性测量的测量图线. 图中蓝色线条为线性拟合线条, 黑色标记为测量数据点. 图上所写  $r$  值为拟合所得的线性相关系数

## V. 结论

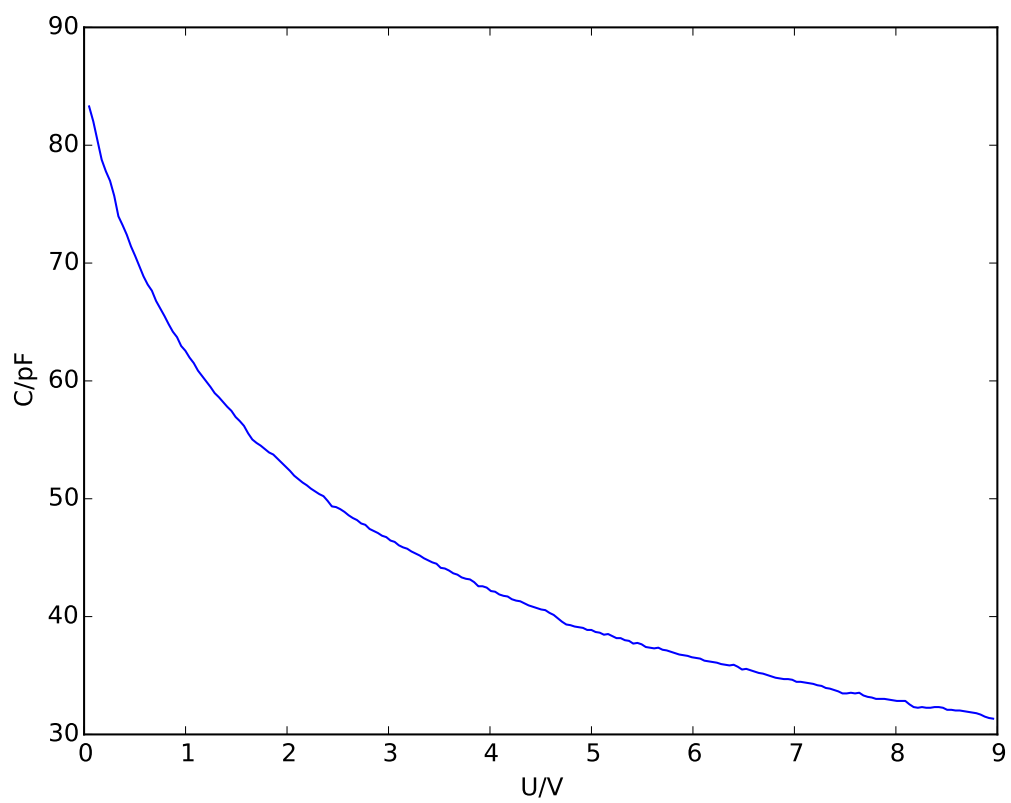
本实验通过电容电压法, 利用锁定放大器测量微小电压, 在此基础上成功地测得了  $p^+-n$  结中  $n$  型半导体侧的杂质浓度分布.

## VI. 致谢

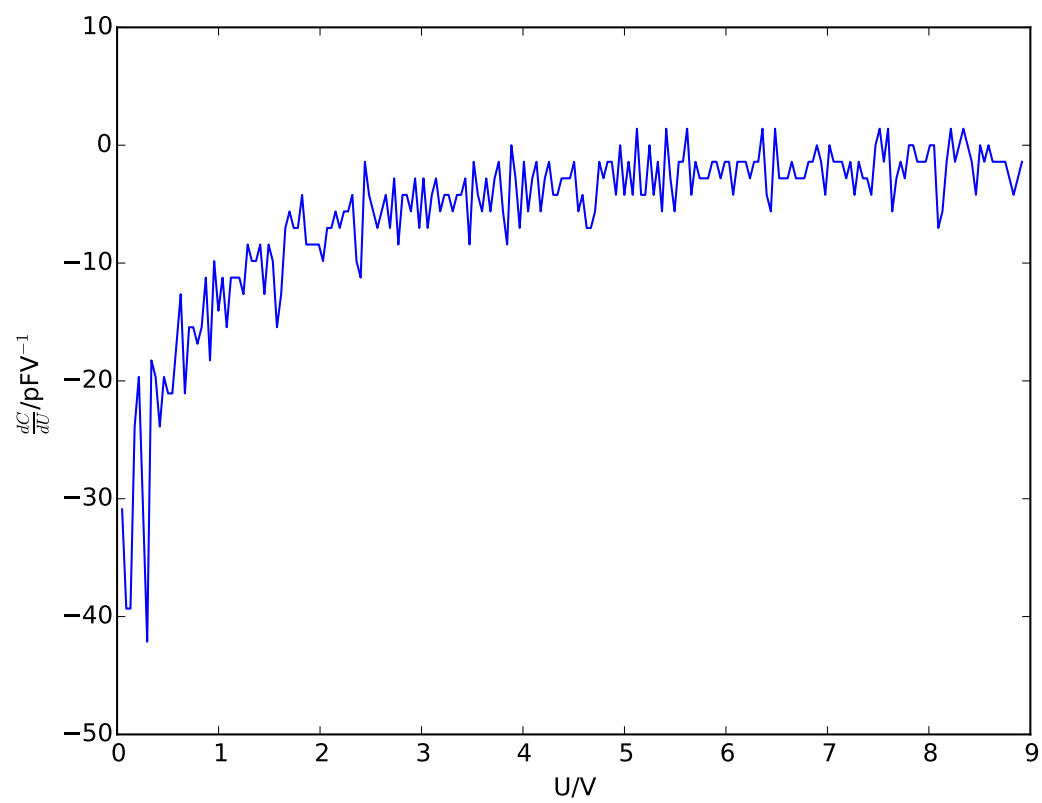
感谢在实验中许福军老师细致而专业的指导

---

[1] 吴思成, 王祖铨 2010 近代物理实验 (第三版) (北京: 高等教育出版社) 第 412 页.

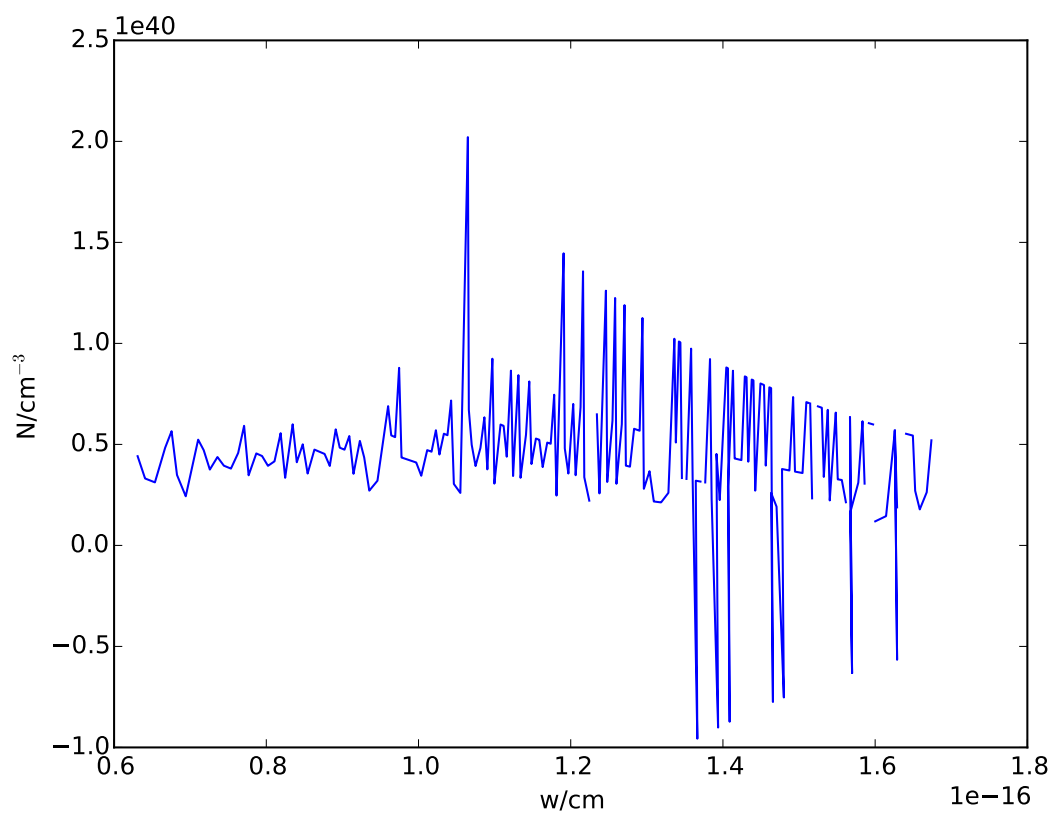


**图 3:** 二号二极管上的电容电压关系测量曲线, 由函数记录仪记录下来并且扫描进计算机后经过程序处理提取数据得到



**图 4:** 二号二极管上的电容对电压导数与电压的关系图线, 由电容电压关系图线计算得到





**图 5:** 最终测得的样品中杂质分布曲线, 注意其中有些点没有数值, 这是因为数值微分的误差, 导致这些点的电容对电压导数为 0, 从而按式 5 这里的分布值没有定义

## A. 代码

本实验的所有数据处理和绘图使用 python 语言完成. 具体的处理使用了 python 的 numpy, scipy 和 matplotlib 这数个开放源代码库套件.

相关的处理代码附在文后

data.py

```
1 from PIL import Image
2
3 xscale = 9.0 / ( 2795 - 613)
4 yscale = 106.0 / ( 2190 - 360)
5 x0 = 613
6 y0 = 2190
7
8 ystart = 2150
9
10 start = 625
11 end = 2795
12 step = 10
13
14 im = Image.open("data.png")
15 pixels = list(im.getdata())
16 width, height = im.size
17 pixels = [pixels[i * width:(i + 1) * width] for i in xrange(height)]
18
19 x = start
20 while x < end:
21     y = ystart
22     while pixels[y][x][3] == 0:
```

```

23         y = y - 1
24     print '{0},{1}'.format((x - x0) * xscale, (y0 - y) * yscale)
25     x = x + step

```

diff.py

```

1 from matplotlib import pyplot as plt
2 import numpy as np
3
4 u = []
5 c = []
6
7 with open('data.txt') as f:
8     line = f.readline()
9     while line:
10         u.append(float(line.split(',')[0]))
11         c.append(float(line.split(',')[1]))
12         line = f.readline()
13
14 du = 10 * 9.0 / ( 2795 - 613)
15 dc = np.divide(np.diff(c) , du)
16
17 plt.xlabel("U/V")
18 plt.ylabel(r"$\frac{dC}{dU} / pFV^{-1}$")
19 plt.plot(np.array(u[:-1]), dc)
20 plt.savefig("diff.pdf")

```

linear.py

```

1 import matplotlib.pyplot as plt
2 import numpy as np

```

```

3 from scipy.stats import linregress as lr
4
5 c = [14.45, 27.7, 53.6, 106]
6 u = [29, 54, 100, 195]
7
8 x = np.linspace(10, 110, 200)
9
10 k, b, r, _, _ = lr(c, u)
11
12 y = k * x + b
13
14 plt.plot(x, y, color='blue', label = 'Fitted_line')
15 plt.scatter(c, u, 100, color = 'black', marker = 'x', label = '
    Experiment_data')
16 plt.legend()
17 plt.xlabel("$C_x$/pF")
18 plt.ylabel("$u_i$/\mu V")
19 plt.text(10, 225, 'r = {0}'.format(r), bbox = {'pad':10, 'facecolor':
    white'}, fontsize = 15)
20 plt.savefig('plot1.pdf')

```

N.py

```

1 from matplotlib import pyplot as plt
2 import numpy as np
3
4 u = []
5 c = []
6

```

```

7 with open('data.txt') as f:
8     line = f.readline()
9     while line:
10         u.append(float(line.split(',')[0]))
11         c.append(float(line.split(',')[1]))
12         line = f.readline()
13
14 du = 10 * 9.0 / ( 2795 - 613)
15 dc = np.divide(np.diff(c) , du)
16 u = u[:-1]
17 c = c[:-1]
18
19 A = 5.03e-3
20 ee0 = 11.8 * 8.854e-14
21 q = 1.602e-19
22
23 N = (- 1.0 / q / ee0 / A / A) * np.array(c) ** 3 / np.array(dc)
24 w = ee0 * A / np.array(c)
25
26 plt.xlabel("w/cm")
27 plt.ylabel("N/cm$^{-3}$")
28 plt.plot(w, N)
29 plt.savefig("Nw.pdf")

```

uc.py

```

1 from matplotlib import pyplot as plt
2
3 u = []

```

```

4 c = []
5
6 with open('data.txt') as f:
7     line = f.readline()
8     while line:
9         u.append(line.split(',')[0])
10        c.append(line.split(',')[1])
11        line = f.readline()
12
13 plt.xlabel("U/V")
14 plt.ylabel("C/pF")
15 plt.plot(u, c)
16 plt.savefig('uc.pdf')

```