基于双波长电光效应测量高电压及基于声光效应测量未知波长方法实践探究

李卓洋 刘耿昊 林泽坤 李云帆 梁柏豪
中山大学物理与天文学院

摘 要:

我们基于电光效应的原理，构建了一套高电压测量系统，利用双波长激光作为信号载体，可以解决测压范围受限于半波电压的问题并进行了分析其误差以及适用条件。实际试验结果表明双波长法确实是可行的测量电压实验方案，

我们基于声光效应原理，推导了光波长与光屏到声光器件距离的关系。利用这个关系，我们设计了一套通过基准激光定标来测量未知激光波长的方案。实际实验的结果表明此方案不仅操作简单，并具有优秀的精确度。

关键词: pockels 效应；布拉格衍射；电光晶体；声光晶体；双波长法

1 引言

本文共分为两部分，第一部分是利用电光效应以及两种波长的激光测量高电压；第二部分利用声光效应测量未知激光波长。

2 双波长法测量高电压

电光效应是指晶体的折射率会在电场的作用下产生各向异性，当偏振光以特定角度通过晶体时，通过快轴的分量将产生相位延迟，使光线改变原本的偏振特性。电光效应与施加的电场强度、晶体材料、晶体宽度和长度等因素有关，当出射的光线通过检偏器后，光强将发生变化，由此便可以得知电场强度、晶体材料、晶体宽度和长度等信息，如果控制其他参数不变，则可以设计出一个测量电压的系统，利用双波长激光作为信号载体，可以解决测压范围受限于半波电压的问题。

2.1 电光效应

在外界电场的影响下，晶体内部的原子分子会发生极化而在宏观上引起晶体的光学性能改变。其中晶体折射率空间分布的改变被称为电光效应。某一方向的电场对沿某一传播的振动方向的折射率改变可以表示为:

$$\Delta\left(\frac{1}{n^2}\right)_{ij} = \gamma_{ijk}\vec{E_k} + b_{ijkl}\vec{E_k}\vec{E_l} \tag{1}$$

等式1右边一次项所引起的折射率线性变化即为一次电光效应，由于折射率 n_{ij} 具有对称性，所以对应的线性系数 γ_{ijk} 是一个具有 18 个独立分量的对称张量，可以用一个 3×6 的矩阵表示：$\gamma_{ijk} = \gamma_{mk}$, 其中 $i = 1,2,3; j = 1,2,3; m = ij$。一次电光效应只存在于不具有对称中心的晶体中，一次效应要比二次效应更显著，本文主要探究一次电光效应。可以将折射率与光传播方向、振动方向表示为折射率椭球，未加电场时，折射率椭球方程可以写为:

$$\frac{x^2}{n_{11}^2} + \frac{y^2}{n_{22}^2} + \frac{z^2}{n_{33}^2} = 1 \tag{2}$$

其中 x,y,z 是晶体主轴坐标，$\left(\frac{1}{n_1^2}\right),\left(\frac{1}{n_2^2}\right),\left(\frac{1}{n_3^2}\right)$ 是沿主轴方向折射率平方的倒数加上电场 $\vec{E_p}$ 后，各向异性产生，则折射率变为:

$$(\Delta n)_{ij} = \frac{1}{2}n^3\gamma_{ijk}E_k \tag{3}$$

椭球方程变为:

$$\frac{x^2}{n_{11}^2} + \frac{y^2}{n_{22}^2} + \frac{z^2}{n_{33}^2} + \frac{2yz}{n_{23}^2} + \frac{2xz}{n_{13}^2} + \frac{2xy}{n_{12}^2} = 1 \tag{4}$$

一次电光效应可根据电场方向与光线传播方向的不同分为两种:

1. 纵向电光效应：电场方向与传播方向平行时产生。如 KD*P 类型晶体
2. 横向电光效应：电场方向与传播方向垂直时产生。如 LiNbO3 晶体

当偏振光通过双轴晶体时将改变偏振特性，在检偏器的作用下，偏振特性将影响输出信号的幅值，由此可以传递所施电场的信息，例如通过改变电压强度调制在激光上的信号，当然也包括电压强度本身。

2.2 电光调制原理

激光作为高频载波，信号多调制于其强度上，也可以采用连续调幅、调频、调相以及脉冲调制等形式。强度调制是根据光载波电场振幅地平方比例调制信号。

原因：光接收器一般都是直接相应其所接受的光强度变化。

方法：机械调制、电光调制、声光调制、磁光调制、电源调制等

电光调制开算速度快、结构简单。电光调制根据所施加的电场方向的不同，可分为纵向电光调制和横向电光调制。

2.2.1 铌酸锂晶体横调制

表 1-1 电光晶体(electro-optic crystals)的特性参数

点群对称性	晶体材料	折射率 n_o	折射率 n_e	波长 (μm)	一次光电系数 ($10^{-12} m/V$)
$3m$	$LiNbO_3$	2.286	2.203	0.633	$\gamma_{13}=\gamma_{23}=10, \gamma_{33}=32$ $\gamma_{42}=\gamma_{51}=28, \gamma_{22}=6.8$ $\gamma_{12}=\gamma_{61}=-\gamma_{22}$
32	Quartz (SiO_2)	1.544	1.553	0.589	$\gamma_{41}=-\gamma_{52}=0.2$ $\gamma_{62}=\gamma_{21}=-\gamma_{11}=0.93$
$\bar{4}2m$	KH_2PO_4 (KDP)	1.5115	1.4698	0.546	$\gamma_{41}=\gamma_{52}=8.77, \gamma_{63}=10.3$
		1.5074	1.4669	0.633	$\gamma_{41}=\gamma_{52}=8, \gamma_{63}=11$
$\bar{4}2m$	$NH_4H_2PO_4$ (ADP)	1.5266	1.4808	0.546	$\gamma_{41}=\gamma_{52}=23.76, \gamma_{63}=8.56$
		1.5220	1.4773	0.633	$\gamma_{41}=\gamma_{52}=23.41, \gamma_{63}=7.828$
$\bar{4}3m$	KD_2PO_4 (KD*P)	1.5079	1.4683	0.546	$\gamma_{41}=\gamma_{52}=8.8, \gamma_{63}=26.8$
$\bar{4}3m$	GaAs	3.60		0.9	$\gamma_{41}=\gamma_{52}=\gamma_{63}=1.1$
		3.34		1.0	$\gamma_{41}=\gamma_{52}=\gamma_{63}=1.5$
		3.20		10.6	$\gamma_{41}=\gamma_{52}=\gamma_{63}=1.6$
$\bar{4}3m$	InP	3.42		1.06	$\gamma_{41}=\gamma_{52}=\gamma_{63}=1.45$
$\bar{4}3m$	ZnSe	2.60		0.633	$\gamma_{41}=\gamma_{52}=\gamma_{63}=2.0$
$\bar{4}3m$	β-ZnS	2.36		0.6	$\gamma_{41}=\gamma_{52}=\gamma_{63}=2.1$

上表展示了有关铌酸锂晶体电光系数的参数信息。本实验主要利用铌酸锂晶体横向电光效应进行电压测量。

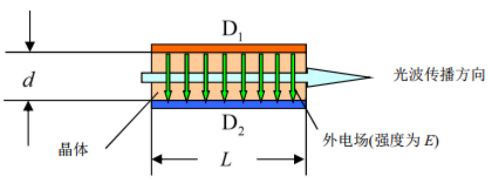

图 1: 横调制器

如图1所示的是横调制器结构图，其中 D_1, D_2 为晶体两端的电势，本实验不对激光进行调制，而是仅仅设置一个直流电压。电光效应引起的相位差 Γ 正比于电场强度 E 和作用距离 L, 而 $E = V/d$, 所以有

$$\Gamma \sim \frac{LV}{d} \tag{5}$$

铌酸锂晶体具有优良的加工性能及很高的电光系数, $\gamma_{22} = 6.8 \times 10^{-12}$ m/V, 常常用来做成横向调制器, 铌酸锂为单轴晶体, 有 $n_x = n_y = n_0 = 2.286, n_z = n_e = 2.203$

把晶体的通光方向设为 Z 方向，沿 X 方向施加电场 E。晶体由单轴变为双轴，新的主轴 X', Y', Z' 轴又称为感应轴，其中 X', Y' 绕 Z 轴转 45°，而 Z' 与 Z 轴重合。晶体的线性电光系数 γ 是一个三阶张量，受晶体对称性的影响 (表 1-1)，铌酸锂的线性电光系数矩阵为

$$\gamma = \begin{bmatrix} 0 & -\gamma_{22} & \gamma_{13} \\ 0 & \gamma_{22} & \gamma_{13} \\ 0 & 0 & \gamma_{33} \\ 0 & \gamma_{42} & 0 \\ \gamma_{42} & 0 & 0 \\ -\gamma_{22} & 0 & 0 \end{bmatrix} \tag{6}$$

施加电场后，得到电场强度矩阵（E，0，0），此时在 X 轴上加上电场后, 折射率的变化为:

$$\Delta n_{ij} = \gamma_{ijk} E_k \tag{7}$$

代入式6以及电场强度矩阵 (E,0,0) 得到:

$$\begin{bmatrix} \Delta\frac{1}{n_{11}^2} \\ \Delta\frac{1}{n_{22}^2} \\ \Delta\frac{1}{n_{23}^2} \\ \Delta\frac{1}{n_{12}^2} \\ \Delta\frac{1}{n_{13}^2} \\ \Delta\frac{1}{n_{23}^2} \end{bmatrix} \equiv \begin{bmatrix} 0 & -\gamma_{22} & \gamma_{13} \\ 0 & \gamma_{22} & \gamma_{13} \\ 0 & 0 & \gamma_{33} \\ 0 & \gamma_{42} & 0 \\ \gamma_{42} & 0 & 0 \\ -\gamma_{22} & 0 & 0 \end{bmatrix} \begin{bmatrix} E \\ 0 \\ 0 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \gamma_{42}E \\ -\gamma_{22}E \end{bmatrix} \tag{8}$$

当外加电场（E，0，0）时，电场作用下的光折射率椭球方程为

$$\frac{x^2}{n_0^2} + \frac{y^2}{n_0^2} + \frac{z^2}{n_e^2} + 2\gamma_{42}E_{xz}xz + 2\gamma_{22}E_{xy}xy = 1 \tag{9}$$

沿 Z 轴方向射入入射光，令上式的 Z=0，折射率椭球就变为与波矢垂直的折射率平面，如图2所示为加了电场后的折射率椭球截面图，经过坐标转换，得到截迹方程为:

$$\left(\frac{1}{n_0^2} + \gamma_{22}E\right)x'^2 + \left(\frac{1}{\mathrm{n}_0^2} - \gamma_{22}\mathrm{E}\right)\mathrm{y}'^2 = 1 \tag{10}$$

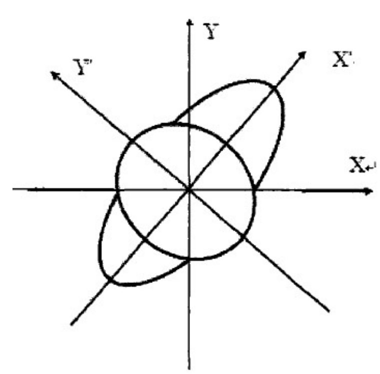

图 2: 沿 x 轴方向施加电场后的折射率椭球

故新主轴折射率为:

$$
\begin{aligned}
n_{x'} &= n_0 + \frac{1}{2}n_0^3\gamma_{22}E \\
n_{y'} &= n_0 - \frac{1}{2}n_0^3\gamma_{22}\mathrm{E}
\end{aligned}
\tag{11}
$$

当激光由晶体出射时两个分量会有一定的相位差。此相位差可以表示为:

$$\varphi = \frac{2\pi}{\lambda}\left(n_x - n_y\right)L = \frac{2\pi}{\lambda}n_0^3\gamma_{22}V\frac{L}{d} \tag{12}$$

式中: λ 为激光的波长，L 为晶体的通光长度，d 为晶体在 X 方向的厚度，V 是外加电压。$\varphi = \pi$ 时所对应的 V 为半波电压，于是可得:

$$V_\pi = \frac{\lambda \mathrm{d}}{2n_0^3\gamma_{22}L} \tag{13}$$

优点: 横调制器的电极不在光路中，工艺上容易解决。

缺点: 对波长很敏感, 当波长确定时又强烈依赖于距离 L。加工误差、装调误差引起的光波方向的稍许变化都会引起相位差的明显改变。

解决方法: 使用准直的激光; 使用一对晶体, 第一块晶体的 x 轴与第二块晶体的 z 轴相对, 使晶体的自然双折射部分相互补偿以消除或降低器件对温度、入射方向的敏感性; 巴比涅-索勒尔补偿器，将工作点偏置到特性曲线的线性部分。

2.3 锥光干涉

锥光干涉的实质就是偏振干涉，偏振光干涉的条件与自然光的干涉条件是一致的, 即: 频率相同、振动方向相同, 或存在互相平行的振动分量、位相差恒定。

当振动方向互相垂直的两束线偏振光经偏振片 P2 后，两束投射光的振幅为

$$\left.\begin{aligned}
A_{2o} &= A_0\sin\alpha = A_1\sin\theta\sin\alpha \\
A_{2e} &= A_e\cos\alpha = A_1\cos\theta\cos\alpha
\end{aligned}\right\} \tag{14}$$

其中, A_1 是射向波片 E_1 的线偏振光的振幅, θ 为起偏器 P_1 出射线偏振光方向与波片光轴的夹角, α 为检偏器 P_2 透光轴方向与波片光轴的夹角。

若两束光之间的相位差为 $\Delta\phi'$ ，那么合强度为

$$
\begin{aligned}
I = A^2 &= A_{2o}^2 + A_{2e}^2 + 2A_{2o}A_{2e}\cos\Delta\varphi' \\
&= A_1^2\left[\cos^2(\alpha+\theta) - \sin 2\theta\sin 2\alpha\sin^2\frac{\Delta\varphi'}{2}\right]
\end{aligned}
\tag{15}
$$

其中 $\Delta\phi'$ 是从偏振片 P_2 出射时两束光之间的相位差。入射在波片上的光是线偏光时，o 光和 e 光的相位相等，波片引入的相位差为 $\Delta\varphi = \frac{2\pi}{\lambda}\left(n_o - n_e\right)d$ ，其中 d 是波片的厚度。

产生锥光干涉是因为当在晶体前放置毛玻璃时，光会发射漫散射，沿各个方向传播。不同方向入射光经过晶体后会引入不同的相位差，不同入射角的入射光将落在接收屏上不同半径的圆周上，因为相同入射角的光通过晶体的长度是一样的，所以引入的相位差也是一样的，所以每一个圆环上光程差是一致的。从而就造成了圆环状的明暗干涉条纹。

因为正交偏振系统中, 设入射光振幅为 E, 入射面与起偏器的夹角为 a，经过前后两个偏振片后, 两束光的振幅为 $Ecos(a)sin(\theta)sin(\alpha)$,$Esin(a)cos(\theta)cos(\alpha)$。当光轴平行或垂直于起偏方向时，I 都趋向于 0。所以干涉图中有一个与偏振片透光方向相同的黑十字。如图3

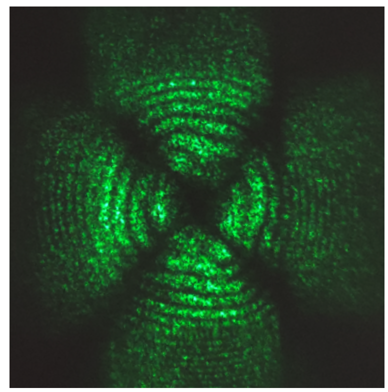

图 3: 锥光干涉条纹

3 双波长法测量高压

实验中利用锥光干涉调整两偏振片光轴相互正交，则由式15以及式13可知最终光功率计探测得到的光功率为:

$$
\begin{aligned}
P &= A_1^2 sin^2(2\theta)sin^2\left(\pi\frac{V}{V_\pi}\right) \\
&= P_{in}sin^2\left(\pi\frac{V}{V_\pi}\right)
\end{aligned}
\tag{16}
$$

3.1 交流电压

当加在晶体两端的电压为

$$V = V_0 + V_m sin\omega t$$

时,

$$P = P_{in}sin^2\left(\frac{\pi}{V_\pi}(V_0 + V_m sin\omega t)\right) \tag{17}$$

当交流电的直流分量 V_0 不断增加，交流等相位点将作周期性变化，即示波器的波形将发生周期性变化，而变化的周期与 V_π 有关, 也即与激光波长有关, 如图4, 当激光波长分别为 650nm,520nm 时，示波器波形不相同。双波长法可以利用这

种特性拓展高压的测量范围。用两种波长的激光进行测量，并同时拟合得到的两种示波器波形，最终得到电压值。

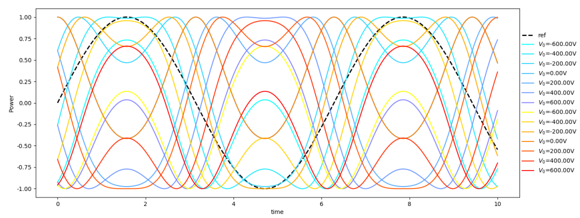

图 4: 增加 V_0 时波形变化。冷色调：波长为 650nm，暖色调：波长为 520nm

3.2 直流电压

当加在晶体两端的电压为 $V = V_0$ 时，功率是稳定值，无法通过读取示波器的波形进行测量，这时需使用光功率计直接测量该电压下激光的透射系数，同样的，测得的功率是电压的周期性函数，周期与半波电压也即激光波长有关，如图5所示。若使用两种周期的激光进行电压的测量，可以拓宽可测量的电压范围。

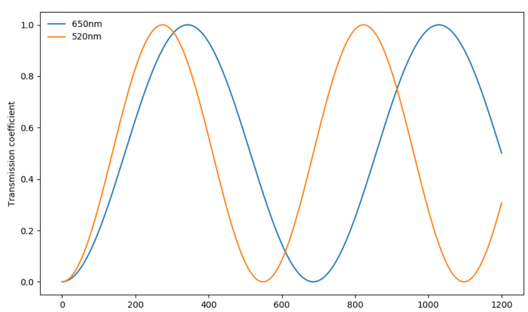

图 5: 透射系数随电压的变化

利用双波长测量电压，得到的是电压李萨如图上的一个点，如图6所示的是一段直流电压从 0-1200V，数据点在 520nmvs650nm 平面上画出的一条轨迹，它的位置可以表示为 $\vec{r} = sin^2\left(\frac{V_0\pi}{V_{\pi 520}}, sin^2\frac{V_0\pi}{V_{\pi 650}}\right)$.

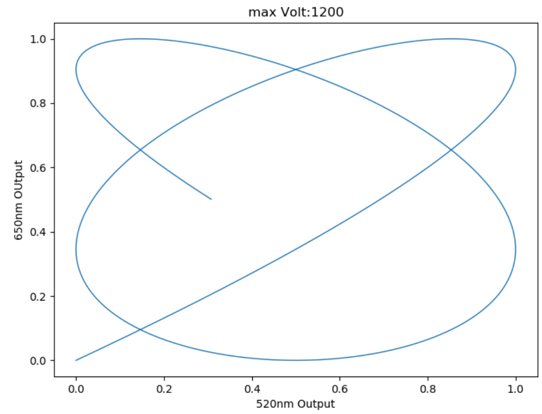

图 6: 520，650nm，0-1200 电压测量值

双波长法测量高电压，只需测得两种激光的理论波长，并对系统进行归一化定标得到理论的透射系数李萨如图。测量时只需得到两个波长的激光下光功率计的数值，归一化后得到透射系数，然后在理论李萨如图上找到对应的点即可完成对电压的测量，若测量点偏离理论轨迹，则只需取最近理论值作为模型预测值即可，此时的实验误差可以用测量点到最近理论点的距离表示。系统的最大测量范围取决于两种激光对应半波电压取整的最小公倍数，如图7为 520-650nm 波长测高压系统的测量范围，约为 1360V。

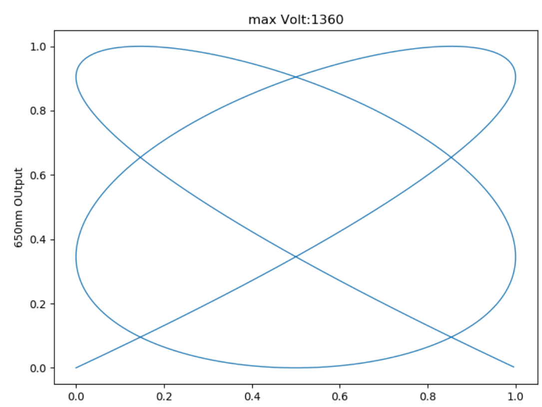

图 7: 520-650nm 波长测高压系统测量范围

双波长法的缺陷在于：

1. 无法测量李萨如图形交点处的电压值，必须假定待测电压有一定的连续性，才可通过轨迹的运行趋势判断测量值正处于哪个轨道中。

2. 对于轨迹上不同的点，要求的测量误差有所变化。在交点处，测量容许误差将为 0。

3. 平均测量容许误差与系统的测量范围成反比。如图8，当两激光波长为质数 557，619nm 时，测量范围增大至由 520-650nm 的 1360V 增加至 344783V，同时由于轨迹过于密集，其所要求的实验精度将变得很高。

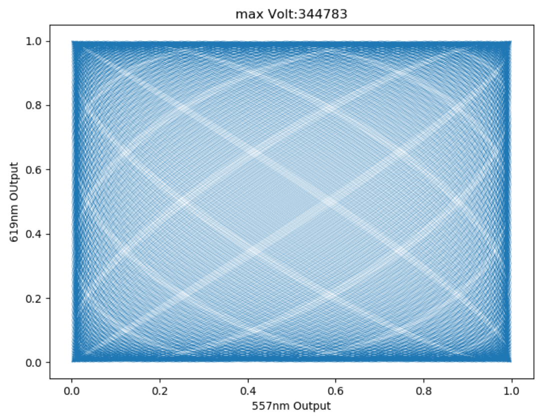

图 8: 557-619nm 波长测高压系统测量范围

3.3 实验装置、方案、测量结果

3.3.1 实验装置

电光实验主体装置如图9，由激光器产生的光功率为 P_{in} 的激光，经过偏振片后获得 $\pi/4$ 偏振光，经过电光晶体后转变为椭圆偏振光 P_{out} 由光功率计接收。

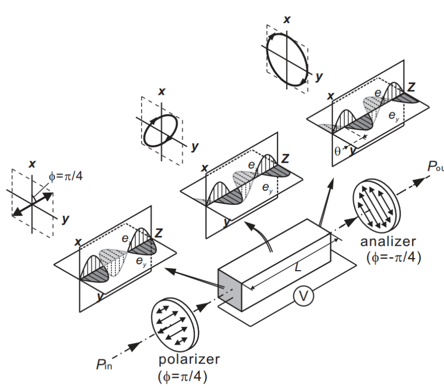

图 9: 实验装置图

3.3.2 实验方案

1. 按图所示搭建好光路，固定好主体部分的仪器，使它们等高共轴，安装 520nm 的准直激光器，使出射的激光通过主体装置，并被光功率计接收。

2. 调节晶体两端输入的直流电压为 600V，交流电压为 0V，转动旋钮使直流电压缓慢下降为 0，并使用摄像装备记录光功率计示数和实时电压。

3. 当晶体两端电压为 0V 时，记录光功率计的功率读数 P_{min}

4. 反转电源电极，调节直流电压从示数 0-600V，使用录像设备记录光功率计示数和实时电压。

5. 换用 650nm 的激光源，重复上述步骤。

6. 读取 4 段录像，获取光功率 P 以电压 V 的函数关系。找到每一段录像中光功率计功率读数最大值，记为 P_{max}

3.4 数据处理

数据在附录中。首先计算

$$T = \frac{P - P_{min}}{P_{max}}$$

得到透射系数。作出 T-V 图并使用

$$a \times sin(\frac{\pi V}{2V_{\pi}} + b)^2 + c$$

进行拟合，结果如图10.

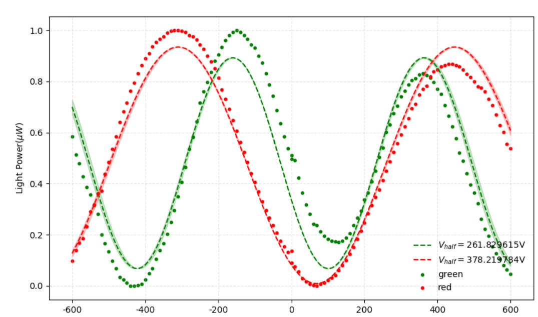

图 10: 拟合结果

拟合得到的半波电压分别为 $V_{520} = 261.83V, V_{650nm} = 378.22V$，对应的激光波长为 $496.29nm, 716.90nm$。对于该电压测量系统，可以使用这波长构建透射系数的李萨如图。如图11所示。

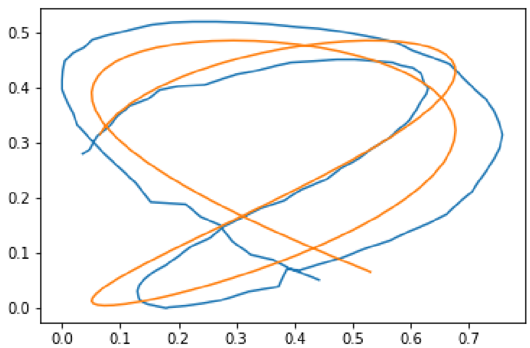

图 11: 拟合李萨如图与实际测量李萨如图

3.5 误差分析

在李萨如图 (11) 上，理论值与测量值的距离为

$$\sigma = \sqrt{(\Delta T_1)^2 + (\Delta T_2)^2}$$

,其中 $\Delta T_1, \Delta T_2$ 恰为两种波长激光测得的透射系数的误差。所以可以用这个 σ 衡量拟合结果与数据点的偏差。

若要保证理论预测值与数据点的偏差不过大，需要使模型的容许误差大于拟合结果与数据点的偏差，即李萨如图上模型的最大容许距离必须包含数据点，否则测量很可能会失败。如图12中灰色区域就是能保证数据匹配到正确的轨迹区间的最小半径构成的区域，如果数据点不在灰色区域中（如图中大部分蓝色点），则很可能发生“串线”的问题，使模型预测值偏离合理的范围。

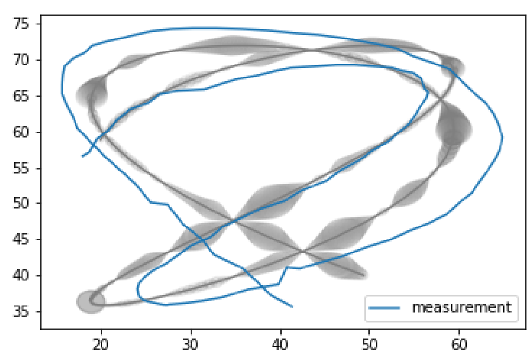

图 12: 模型最大容许误差

模型电压测量范围越大，则容许误差越小，如图8模型轨迹几乎遍布整个平面，使得几乎没有允许误差存在的空间。

对于本实验拟合结果，允许误差的统计直方图如图13所示：

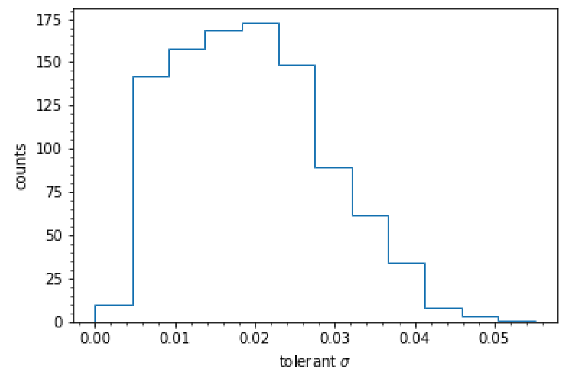

图 13: 容许误差统计直方图

该测量范围（-600V-600V）的平均容许误差约为 0.015，容许误差随电压的变化如图14所示，橙色曲线为数据值与你拟合结果的误差，蓝色曲线为拟合结果最大容许误差。

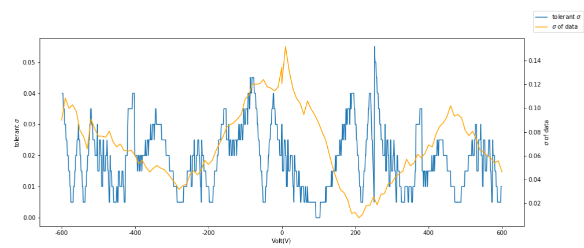

图 14: 容许误差与电压的关系

可见由于实验精度较低，若以距离最近的模型预测点作为预测值，则绝大部分数据点在该测量范围的模型下并不能获得合理的电压值。

若降低测量的电压范围，则模型能够获得合理地电压预测值, 如图15。

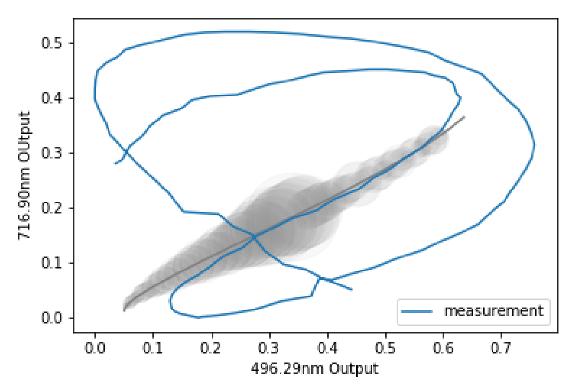

图 15: 减小电压测量范围能获得更大的容许误差

3.6 总结

我们以电光晶体的 Pockels 效应为基础，利用出射激光功率反算出横调制电压，并利用双波长法解决了半波电压对电压测量范围的限制，证明了只要激光波长选取得当，则理论上可以通过该系统测量到超过 3.4×10^5V 的电压。我们借助近代物理实验 IIE4 的光学设备，实现了对双波长测电压系统的定标、李萨如图形的建模、高电压的测量。最终证明双波长法虽然可以拓宽电压的测量范围，但其对精度的要求极高，实验过程中由于人员读数、仪器设备等误差影响，最终测量结果往往与真实值有一定的差距。

本实验方案操作过程简洁，随受限于已有的实验设备，无法对高于 600V 的电压进行精确测量，但仍然证明了双波长法测量直流高电压的可行性并总结出其优缺点，为高压测量提供了一个新的思路。

4 基于声光效应的简易波长测量方案

4.1 简介

近些年来，声光材料和声光器件的性能有了长足的进步，使得声光技术日渐趋于成熟，在光通讯，精密测量等领域开始发挥重要的作用。本文基于小组成员在实验课上学习到的声光效应原理，设计了一套操作简易，成本低廉，同时保持着较高精度的波长测量方案。并且通过进行实际测量验证了此方案具有优秀的精度。

本文的结构安排如下：在第一部分回顾了声光效应中 Bragg 衍射的原理。第二部分介绍了利用声光效应测量光波长的方法。第三部分介绍了实际测量时的实验装置，方案和结果。我们将在第四部分利用自动化程序分析实验结果，并在第五部分作总结。

4.2 原理回顾

当超声波在介质中传播时，由于能动量的传递，介质将产生时间和空间上的周期性的弹性应变，并导致其折射率也发生相应的改变。当光束通过这样的介质时就会像穿过光栅一样发生衍射，此即声光效应。

Bragg 衍射 超声波在声光介质中传播，其角频率为 ω_s，波长为 λ_s，波矢为 k_s。设入射光是平面波，传播方向与超声波方向垂直，角频率为 ω_0，波长和波矢为 λ_0,k_0。由于光速远大于声速，在光波通过的 Δt 内介质的折射率可看成是不变的。当弹性应变较小时，折射率的变化随时间以正弦的形式波动。

$$\Delta n(y,t)=\Delta n sin(\omega_s t-k_s y)$$

Δn 是声波折射率变化的幅值，由介质的光弹系数 P 和应变 S 决定（在各项同性介质中它们是标量）

$$\Delta n=-\frac{1}{2}n^3PS_0$$

当光波通过厚度为 L 的介质时，前后两点的相位差为

$$\Delta\Phi=k_0n(y,t)L=k_0n_0L+k_0\Delta nL\sin\left(w_st-k_sy\right)$$

右边的第二项就是超声波引起的附加相位差，随时间正弦震荡，这使得出射光的波振面随时间变化，从而使得光传播特性改变，产生了衍射。下面我们定量分析这种衍射的特征。

设超声波的传播方向为 y，光波传播方向为 x，坐标原点取在入射面与出射面中间，入射波可以写成 $E_i=Ae^{it}$。通过计算出射面上各点相位的改变，并将它们对出射面后的一点的贡献叠加。我们可以得到 xy 平面内，衍射角为 θ，距离出射面很远的一点的衍射光波为

$$E=Ce^{iwt}\int_{-\frac{b}{2}}^{\frac{b}{2}}e^{ik_0\Delta nL\sin(k_sy-w_st)}e^{-ik_0y\sin\theta}dy$$

可以利用贝塞尔函数将上式展开并积分

$$E=Cb\sum_{m=-\infty}^{\infty}J_m(k_0\Delta nL)e^{i(w-mw_s)t\frac{\sin[b(mk_s-k_0\sin\theta)/2]}{b(mk_s-k_0\sin\theta)/2}}$$

第 m 级衍射的光波为

$$E_m=E_0=CbJ_m(k_0\Delta nL)\frac{\sin\left[b\left(mk_s-k_0\sin\theta\right)/2\right]}{b\left(mk_s-k_0\sin\theta\right)/2}e^{i(w-mw_s)t}$$

利用函数 $\frac{sinx}{x}$ 在 $x=0$ 处取最大值，易得当 $mk_s-k_0sin\theta=0$，即 $\sin\theta=m\frac{k_s}{k_0}=m\frac{\lambda_0}{\lambda_s}$ 时有衍射振幅的极大值。

当声光作用的距离大于 $\frac{2\lambda_s^2}{\lambda}$，且光束相对于超声波波面有一定倾角入射时，会产生 Bragg 衍射，此时入射光同时因相位与振幅的震荡而被调制，$m>1$ 的衍射光刚好相互抵消了，只留下 0 级和 1 级（或-1 级）衍射光。此时衍射效率非常高，因此 Bragg 衍射在高精度领域有着广泛的应用。

4.3 测量方法

在上节我们已经知道，当 $\sin\theta=m\frac{\lambda_0}{\lambda_s}$ 时，衍射振幅有极大值。当发生 Bragg 衍射时，我们取 m=1，可得衍射角满足 $sin\theta_D=\frac{\lambda_0}{\lambda_s}$

这是介质中的衍射角，由于我们是在空气中测量偏转的角度，因此还要额外考虑介质折射率（空气的折射率认为是 1）。利用折射公式易得

$$sin\theta_{air} = n_d sin\theta_D = \frac{n_d\lambda_0}{\lambda_s}$$

空气中的偏转角 θ_{air} 通过测量 0 级到 1 级（或-1 级）的衍射光斑的中心距离 a 与光屏到声光介质的出射面距离 r 可以得到。

$$sin\theta_{air} = a/r$$

联立上面两式可得光波波长为

$$\lambda_0 = \frac{\lambda_s a}{n_d r}$$

如果我们已有一个已知波长的激光（例如，He-Ne 激光，波长为 632.8nm），则可以让其作为基准波长 λ_1，来测量未知激光的波长 λ_x。具体做法是固定两光斑的间距 a，分别测量基准激光和待测激光发生衍射时的光斑中心距离 r_1，r_x，此时有

$$\lambda_1 = \frac{\lambda_s a}{n_d r_1}$$

$$\lambda_x = \frac{\lambda_s a}{n_d r_x}$$

两式相除可得未知激光波长

$$\lambda_x = \frac{r_1\lambda_1}{r_x}$$

4.4 实验装置，方案, 测量结果

实验装置 实验装置如图所示，其中 1 为激光器，2 为偏振片，3 为声光器件，4 为光屏

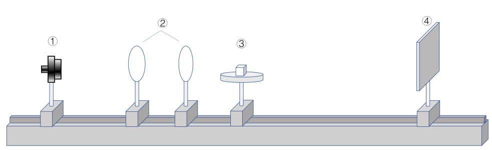

图 16

图 1 实验装置图

实验方案 1. 按图所示顺序搭建好光路，固定好所有仪器，并使它们等高共轴。

2. 定标

（1）打开基准激光器（如 He-Ne 激光），调节驱动电压到工作电压。粗调频率旋钮和声光器件的方向，此时光屏上应出现两个衍射光斑。

（2）调节光屏的位置使得衍射光斑正好处于刻度线上，便于后续测量。

（3）微调频率旋钮和声光器件的方向，使得光屏上的 1（-1）级衍射光斑最亮。记录下光屏此时的位置。

（4）利用光屏上的刻度线测量两个衍射光斑的间距 a_1。

3. 测量光屏的位置到声光器件的距离 r_1

4. 将基准激光器更换为待测激光器，此时两个光斑的相对距离发生改变。调整光屏位置，使得两个衍射光斑间距回到 a_1，测量光屏的位置到声光器件的距离 r_x。

5. 改变第二步定标时两个衍射光斑的间距 a_1，再重复上述步骤进行测量，得到多组数据。

测量结果 我们在实验室按上述方案进行了实际测量，选用 He-Ne 激光作为基准激光（波长为 632.8nm），选取已有的绿色激光（理论波长为 520nm）作为未知激光进行测量验证，得到了如下数据。

测量次数	r_x(cm)	r_1(cm)
1	33.6	27.8
2	34.9	28.1
3	35.1	29.3
4	35.4	28.9
5	36.0	29.2
6	37.5	30.3
7	38.3	32.0
8	39.5	32.5
9	42.0	33.4
10	53.8	42.9

4.4.1 实验数据分析

我们利用编写的数据分析程序对实验数据分析（程序的代码展示在附录 python 程序中），此程序具有如下功能：

1. 能够自动读取实验数据文件并利用最小二乘法计算并待测光波波长的拟合值，同时还能输出数据的标准差。

2. 在第一次计算完拟合值后，程序会判别哪些点与拟合值的不一致性超过 3σ 并把这些点删除，而后再进行一次最小二乘法得到最佳拟合值。

3. 可自动绘制每个数据点的值和误差，方便分析误差随距离变化的特点。

我们将上节测量结果的数据文件（txt，csv 均可）导入到程序所在文件夹，运行程序得到如下结果：

```
第一次绿光波长的拟合值是514.468245613222nm
标准差是9.297637661070402nm
检索并去除误差超过3sigma的数据点....
第二次绿光波长的最佳拟合值是514.468245613222nm
标准差是9.297637661070402nm
```

图 17

图 2 程序运行结果

可以看到，两次拟合的结果没有区别，说明数据点中没有出现 3σ 误差。因此我们认为最终的波长测量结果如下：

$$\lambda_x = 514.47 \pm 11.34(nm)$$

待测绿色激光的理论值是 $520nm$，因此我们可以看到此次测量的相对误差为 $\frac{520-514.47}{520}=1.06\%$。测量值与实际波长吻合, 误差被控制在 1% 的水平。

下面我们来分析数据中含有的误差。

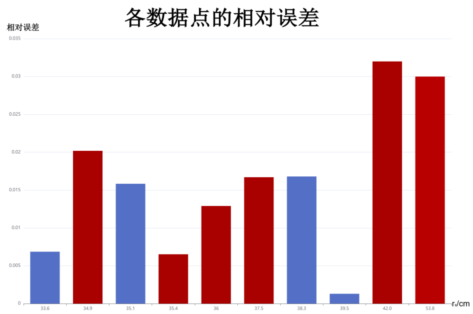

图 18

图 3 各数据点的相对误差，注意蓝色代表此点误差为正，红色代表此点误差为负

将每个点的相对误差制作成图表，我们发现既有偏大的数据, 也有偏小的数据, 并没有明显的证据说明测量中有显著的系统误差存在。

很显然, 系统中主要的误差来源是随机误差, 我们分析认为随机误差主要来源于以下因素:

1. 用导轨上的刻度尺，并不能很精确地测量出声光器件到光屏的距离，一方面是由于声光器件的出光面并不在刻度尺上，有一定垂直距离，读取位置时会产生误差。另一方面当时的环境很暗, 不利于读数, 这也会对测量带来影响。

2. 在切换激光源时要保持两光斑中心距离 $a1$ 不变，由于当时光斑很亮, 比较刺眼, 读取光斑中心位置有一定困难, 若读取的位置出现误差，那么 $a1$ 就没有严格地固定了，这样也会引入误差。

上述误差来源都可以通过改进实验测量仪器和方案得到解决，例如第一个误差来源可以通过提前将出光面和支架对应的刻度的距离测量出来，在后续测量中就只需要读取支架的刻度即可（支架的刻度是容易读出来的)。第二个误差来源可以通过将光屏换成 CCD 相机，利用 CCD 技术测量两光斑的距离来解决。我们相信，经过这些改进，测量的精度还有很大的提升空间。

4.5 总结

我们以声光效应中的 Bragg 衍射理论为基础，推导了光波波长与光屏到声光器件距离 r 以及光屏上两衍射光斑中心间距 a 的数学关系，并利用这个关系以及基准定标的思想得到了一个简便的测量未知激光波长的方法。而后借助常见的实验室设备，配合自己编写的实验数据分析程序，发展了一套操作简易的波长测量方案。最后我们按方案进行了一次实际测量实验，实验结果表明此方案用于测量未知激光波长有着很不错的精度，通过分析误差来源发现只需再添加一两个步骤和将光屏换成 CCD 相机，方案的测量精度还能得到进一步的提升。

本方案的实验操作过程十分简洁，且只需要常见的实验室仪器（He-Ne 激光，偏振片，声光器件，光屏）即可完成，并且能保持相当好的精度（1% 水平）。此方案可以快速, 低成本地实现对不同光波波长的测量, 在快速测量领域可以发挥重要的作用。

Listing 1: 声光效应绘图分析 python 代码

```python
# %%
import numpy as np
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt

#读取数据
data = np.loadtxt('data.txt', usecols=(0,1))
rx=data[:,0]
r1=data[:,1]

lambda_0=632.8 #在这里输入基准激光波长

#拟合
def func(x,k):
    return k*x
popt,pcov=curve_fit(func,rx,r1)
lambda_xbest=popt[0]*lambda_0

#计算标准差
s=0
lambda_xi=np.zeros( len(r1))
for i in  range( len(rx)):
    s=s+((r1[i]*lambda_0/rx[i])-(lambda_xbest))**2
    lambda_xi[i]=r1[i]*lambda_0/rx[i]
s=np.sqrt(s/( len(r1)-1))

print('第一次绿光波长的拟合值是'+ str(lambda_xbest)+'nm')
print('标准差是'+ str(s)+'nm')
print('检索并去除误差超过3sigma的数据点....')

##除去3sigma点
ind=np.empty(shape=(0,3))
for i in  range( len(r1)):
    err=np. abs(lambda_xi[i]-lambda_xbest)
    if err>=3*s:
        print('除去第'+ str(i)+'个点')
        ind=np.append(ind,i)
ind=ind.astype( int)
rx=np.delete(rx,ind)
r1=np.delete(r1,ind)

#第二次拟合
popt,pcov=curve_fit(func,rx,r1)
lambda_xbest=popt[0]*lambda_0

#第二次计算标准差
```

```python
s=0
lambda_xi=np.zeros( len(r1))
for i in  range( len(r1)):
    s=s+((r1[i]*lambda_0/rx[i])-(lambda_xbest))**2
    lambda_xi[i]=r1[i]*lambda_0/rx[i]
s=np.sqrt(s/( len(r1)-1))
print('第二次绿光波长的最佳拟合值是'+ str(lambda_xbest)+'nm')
print('标准差是'+ str(s)+'nm')
```

Listing 2: 电光效应绘图分析 python 代码

```python
# %%
import numpy as np
import matplotlib.pyplot as plt

# %%
def half_length_voltage(lamda):
    gamma = 6.8e-3;L=35;d=3;n0=2.286
    V_half=lamda*d/(2*n0**3*gamma*L)
    return V_half
def P(V,V_half,a=1,b=0,c=0):
    return a*np.sin(np.pi*V/(2*V_half)+b)**2+c
def Pt(t,V,V_half,omega=1,a=1,b=0,c=0):
    return a*np.cos(np.pi*V/(2*V_half)+2*np.sin(omega*t)+b)+c

# %%
#style=plt.style.available
#print(style)
#plt.style.use(style[0])
V=np.linspace(0,1200,1000)
P1=P(V,half_length_voltage(650))
P2=P(V,half_length_voltage(520))
plt.subplots(figsize=(10,6),dpi=100)
plt.plot(V,P1,label='650nm')
plt.plot(V,P2,label='520nm')
plt.legend(frameon=False,loc='upper left')
plt.xlabel('DC Voltage(V)')
plt.ylabel("Transmission coefficient")
plt.savefig('dctheo.png')

# %%
t=np.linspace(0,10,1000)
scale=600
V=np.linspace(-scale,scale,7)
fig,ax=plt.subplots(figsize=(15,6),dpi=100)
fig.subplots_adjust(right=0.9)
ax.plot(t,np.sin(t),c='black',lw=2,ls='--',label='ref')
for v in V:
    P1=Pt(t,v,half_length_voltage(650))
```

```
    color=v/scale/2+3/2
    ax.plot(t,P1,label=r'$V_{0}$'+'={:.2f}V'. format(v),c=[(color-1)/2,1-(color-1)/2,1])
    #plt.xticks(np.linspace(0,t[-1],10),labels=['{:.1f}'.format(i) for i in
     np.linspace(0,1,10)])
plt.xlabel('time')
plt.ylabel("Power")
for v in V:
    P1=Pt(t,v,half_length_voltage(520))
    color=v/scale/2+3/2
    ax.plot(t,P1,label=r'$V_{0}$'+'={:.2f}V'. format(v),c=[1,2-color,0])
    #plt.xticks(np.linspace(0,t[-1],10),labels=['{:.1f}'.format(i) for i in
     np.linspace(0,1,10)])
fig.legend(frameon=False,loc='right')
fig.savefig('actheo.png')

# %%
rang=1360
V=np.linspace(0,rang,9000)
x,y=[520,650]
P1=P(V,half_length_voltage(x))
P2=P(V,half_length_voltage(y))
plt.subplots(figsize=(8,6),dpi=100)
plt.plot(P1,P2,lw=1)
plt.xlabel( str(x)+'nm Output')
plt.ylabel( str(y)+'nm OUtput')
plt.title('max Volt:'+ str(rang))
plt.savefig('phase_of_'+ str(x)+'_'+ str(y)+'range'+ str(rang)+'.png')
plt.grid(alpha=0.6)

# %%
import matplotlib.animation as animation

fig, ax = plt.subplots()
xdata, ydata = [], []
ln, = ax.plot([], [], 'r-', animated=False)
ln2, = ax.plot([], [], 'b-', animated=False)
def init(): # 初始化的操作
    ax.set_xlim(0, 1200);  ax.set_ylim(0, 1)
    ax.set_xlabel('DC Voltage')
    ax.set_ylabel("Transmission coefficient")
    return ln

def update(frame): # 绘制某一个 frame 的操作
    V=np.linspace(0,frame,frame)
    P1=P(V,half_length_voltage(632))
    P2=P(V,half_length_voltage(520))
    ln.set_data(V, P1)
    ln2.set_data(V, P2)
    return ln
def update1(frame): # 绘制某一个 frame 的操作
```

```python
    V=np.linspace(0,frame,frame)
    P1=P(V,half_length_voltage(632))
    P2=P(V,half_length_voltage(520))
    ln.set_data(P1, P2)

    return ln
def init1(): # 初始化的操作
    ax.set_xlim(0, 1);  ax.set_ylim(0, 1)
    plt.xlabel('632nm Output')
    plt.ylabel("520nm OUtput")
    return ln

#制作动画
## update是更新函数，init初始函数；#frames总的帧数控制数组；#interval的单位是毫秒
    #repeat控制动画循环
ani = animation.FuncAnimation(fig, update, frames= range(1200),
  init_func=init, interval=1, repeat=False)
ani.save('anim3.gif', writer='pillow') #保存动画

fig1, ax = plt.subplots()
xdata, ydata = [], []
ln, = ax.plot([], [], 'r-', animated=False)
ln2, = ax.plot([], [], 'b-', animated=False)
ani1 = animation.FuncAnimation(fig1, update1, frames= range(1200),
  init_func=init1, interval=1, repeat=False)
ani1.save('anim.gif', writer='pillow') #保存动画
plt.show() #显示动画

# %%
green=red= np.linspace(-600,600,120)
green_P1 = np.array([29.1,29.4,30.7,32.2,33.1,34.5,35.9,36.7,37.6,38.7,40.5,
41.6,42.5,43.3,44.5,45.3,46.5,47.4,48.5,49.3,50.1,50.7,51.3,51.8,52.3,52.5,
53.0,53.2,53.6,53.8,54.2,54.4,54.6,54.8,55.0,54.9,54.9,54.6,54.3,54.2,54.1,
53.7,53.4,52.9,52.3,51.8,51.3,50.8,50.0,49.2,48.3,48.0,47.4,46.6,45.6,44.9,
44.3,43.7,41.6,  32.8,31.9,31.8,30.9,30.3,29.9,28.8,28.2,27.5,27.3,26.8,26.6,
26.1,25.5,25.2,24.9,24.6,24.2,23.9,23.7,23.6,23.5,23.0,22.7,22.3,22.1,
21.7,21.5,21.0,20.8,20.5,20.1,20.0,19.7,19.6,19.7,19.4,19.2,19.1,19.9,
18.6,18.3,18.1,17.7,17.9,17.9,17.8,17.6,17.2,17.1,16.8,16.4,16.1,15.7,15.5,
15.0,14.6,14.2,13.8,13.1])
red_P1 =
    np.array([36.1,36.3,36.2,36.1,35.9,35.6,35.3,35.0,34.7,34.5,34.1,33.3,32.9,32.4,32.2,31.6,31.2,30.7,30.1,29.1,28

# %%
P_front =
    np.array([36.1,36.3,36.2,36.1,35.9,35.6,35.3,35.0,34.7,34.5,34.1,33.3,32.9,32.4,32.2,31.6,31.2,30.7,30.1,29.1,28
P_back = np.array(
    list([37.3,37.3,37.3,37.3,37.2,37.0,36.8,36.6,36.3,35.8,35.6,34.9,34.5,34.1,33.4,32.8,32.2,31.5,31.0,30.5,29.8,2
red_P1=np.append(P_back,P_front)
red_P1
```

```
# %%
from scipy.optimize import curve_fit
green_P2 =
    np.append(np.array([40.8,42.2,44.4,47.0,49.5,52.3,54.4,56.6,58.7,60.0,61.5,62.2,63.2,64.0,64.6,
64.9,64.5,63.9,63.0,61.7,60.2,59.0,56.8,54.9,53.1,50.9,47.3,44.3,42.0,
38.6,35.7,32.9,30.3,28.0,25.7,23.9,22.6,20.8,19.1,18.1,16.9,16.0,15.8,
15.7,15.7,16.3,16.9,17.4,19.0,20.5,22.3,23.9,25.6,29.5,31.2,33.3,34.7,
36.8,39.3,41.0,44.4])[::-1],np.array([40.1,39.9,36.5,33.6,31.3,29.5,27.6,27.3,26.1,25.3,24.7,
24.3,24.2,24.1,24.3,24.9,25.8,27.3,28.8,30.1,32.3,33.6,35.8,37.8,40.5,42.3,
45.3,46.8,48.9,50.3,52.1,53.3,54.5,55.3,55.6,56.4,56.6,56.2,55.8,54.9,
53.6,52.6,50.5,48.4,46.4,44.1,41.3,39.7,37.4,35.2,33.6,31.6,28.6,26.6,25.3,
23.3,21.9,20.8,19.6,18.8,18.0]))

red_P2=np.append(np.array([40.9,41.7,42.6,43.8,44.9,46.1,47.2,48.5,50.0,51.5,
52.8,54.5,56.0,57.5,59.2,60.8,62.5,64.0,65.4,67.2,68.7,69.7,70.5,71.6,72.2,
73.0,73.5,73.7,74.1,74.3,74.4,74.4,74.3,74.0,73.5,73.1,72.6,72.0,70.9,70.2,
69.1,67.9,66.5,65.3,63.5,62.1,60.5,58.4,56.5,54.5,52.7,50.1,49.8,48.1,47.0,
44.8,43.0,42.3,41.2,39.6])[::-1],np.array([41.1,39.3,38.7,38.0,36.9,36.4,
36.1,35.9,35.8,36.1,36.3,36.6,37.0,37.4,38.1,38.9,39.7,40.6,41.6,42.9,44.1,
45.3,46.7,48.0,49.2,50.3,51.7,53.2,54.6,56.0,57.3,58.6,59.9,61.1,62.6,63.3,
64.7,65.5,66.,67.4,68.2,68.5,68.9,69.1,69.3,69.3,69.1,68.9,68.5,67.8,67.3,
66.7,65.9,65.7,65.2,64.0,63.1,61.7,60.1,59.0,57.2,56.6]))
'''poly = np.polyfit(x,green_P2,deg=2)
green_P2/=np.polyval(poly,x)
poly = np.polyfit(x,red_P2,deg=2)
red_P2/=np.polyval(poly,x)'''
x=np.append(np.linspace(-600,0,61),np.linspace(0,600,61))
plt.plot(green_P2,red_P2)
poptg,convg = curve_fit(P,x,green_P2,maxfev=1000000,p0=[270,1,0,59])
poptr,convr = curve_fit(P,x,red_P2,maxfev=100000,p0=[342,40,0,40])
datag=P(x,*poptg)
datar=P(x,*poptr)
plt.plot(datag,datar)
siga=np.sqrt((green_P2-datag)**2+(red_P2-datar)**2)
plt.subplots()
plt.plot(x,siga)

# %%
from scipy.optimize import curve_fit
V=np.linspace(-600,600,1000)
P1=P(V,563)
P2=P(V,505-121)
plt.subplots(figsize=(10,6),dpi=100)
#plt.plot(V,P1,label='650nm')
#plt.plot(V,P2,label='520nm')
x=np.append(np.linspace(-600,0,61),np.linspace(0,600,61))
green_P2=green_P2/green_P2. max()-green_P2. min()/green_P2. max()
red_P2=red_P2/red_P2. max()-red_P2. min()/red_P2. max()
poptg,convg = curve_fit(P,x,green_P2,maxfev=1000000,p0=[270,1,0,59])
poptr,convr = curve_fit(P,x,red_P2,maxfev=100000,p0=[342,40,0,40])
```

```
plt.scatter(x,green_P2,c='g',s=10,label='green')
poptgup=poptg.copy()+np.diag(convg)
poptgdown=poptg.copy()-np.diag(convg)
print(poptg)
plt.plot(x,P(x,*poptg),c='g',ls='--',label=r'$V_{half}=%f$V'%np. abs(poptg[0]))
plt.fill_between(x,P(x,*poptgup),P(x,*poptgdown),color='g',alpha=0.2)

plt.scatter(x,red_P2,c='r',  s=10,label='red')
plt.plot(x,P(x,*poptr),c='r',ls='--',label=r'$V_{half}=%f$V'%poptr[0])
poptrup=poptr.copy()+np.diag(convr)
poptrdown=poptr.copy()-np.diag(convr)
plt.fill_between(x,P(x,*poptrup),P(x,*poptrdown),color='r',alpha=0.2)

plt.xlabel('Voltage(V)')
plt.ylabel('Light Power($\mu W$)')
plt.grid(ls='--',alpha=0.3)
plt.legend(frameon=False)
plt.savefig('fit.png')

# %%
x=np.append(np.linspace(-600,0,61),np.linspace(0,600,61))
plt.plot(green_P2,red_P2)
poptg,convg = curve_fit(P,x,green_P2,maxfev=1000000,p0=[270,1,0,59])
poptr,convr = curve_fit(P,x,red_P2,maxfev=100000,p0=[342,40,0,40])
datag=P(x,*poptg)
datar=P(x,*poptr)
plt.plot(datag,datar)
siga=np.sqrt((green_P2-datag)**2+(red_P2-datar)**2)
plt.subplots()
plt.plot(x,siga)

# %%
for color in [green_P2,red_P2]:
    fft_green=np. abs(np.fft.fft(color))
    fft_greenx=np.fft.fftfreq( len(color))
    mask=np.where(fft_greenx>0)[0]
    y=fft_green[mask]
    x=fft_greenx[mask]
    plt.plot(x,y)
    plt.scatter(x[np.where(y==y. max())[0]],y. max())
    maxx=x[np.where(y==y. max())[0]][0]
    maxy=y. max()
    plt.text(maxx,maxy,'{:.8f}'. format(maxx))

# %%
from matplotlib.patches import Circle
'''cir1 = Circle(xy = xy, radius=radius,color='b',alpha=0.5)
ax.add_patch(cir1)'''
def nearest(point,x,y):
```

```
    line=np.vstack((x,y)).T
    dist=np. sum((line-point)**2,axis=1)
    res = line[np.where(dist==np. min(dist))[0]]
    return res
fig=plt.figure()
ax=fig.add_subplot(111)
ax.scatter(green_P2[0],red_P2[0])
ax.scatter(39.12706421, 41.60821506)
print(x[np.where(P(x,*poptg)==39.12706421)])
x=np.linspace(-600,3000,10000)
ax.plot(P(x,*poptg),P(x,*poptr))
plt.plot(green_P2,red_P2)

# %%

def dist(a,b):
    return np.sqrt((a[0]-b[0])**2+(a[1]-b[1])**2)
def is_nearby(i,xy_scan,datag,datar,j):
    last=(datag[i-1],datar[i-1])
     next=(datag[i+1],datar[i+1])
    gap= int(2*j/dist(last, next)+1)
    left=0 if i-gap<0 else i-gap
    right= len(datag) if i+gap> len(datag) else i+gap
    nearlist=[(x,y) for x in datag[left:right] for y in datar[left:right]]
    return xy_scan in nearlist

def find_radius(datag,datar,sigma):
    radiuslist=[]
    for i in  range(1,length-1):
        print('{:*<10s} {:d}%\r'. format('>'* int(i/(length-1)*10), int(i/(length-1)*100)),end='')
        xy_ori=(datag[i],datar[i])
        for j in np.arange(0,sigma,0.01):
            maxsigma=0
            for k in  range(length):
                xy_scan=(datag[k],datar[k])
                if dist(xy_ori,xy_scan)<j and j>0 and xy_ori!=xy_scan:
                    if not is_nearby(i,xy_scan,datag,datar,j):
                        maxsigma=j/2
                        break
            if maxsigma != 0:
                break
        radiuslist.append(maxsigma)
    return radiuslist

x=np.linspace(-600,600,1000)
datag=P(x,*poptg)
datar=P(x,*poptr)
length= len(datag)
radiuslist=find_radius(datag,datar,0.2)
```

```
np.save('radiulist_exp',radiuslist)


from matplotlib.patches import Circle
x=np.linspace(-600,600,10000)
#mask=[np.where(x<600)[0]]
#x=x[mask]
datag=P(x,*poptg)
datar=P(x,*poptr)
length= len(datag)
radiuslist=np.load('radiulist_exp.npy')


# %%
fig=plt.figure()
ax=plt.axes()
plt.plot(datag,datar,color='gray')
for i in  range( len(radiuslist)):
    cir1 = Circle(xy = (datag[i],datar[i]), radius=radiuslist[i],color='gray',alpha=0.03)
    ax.add_patch(cir1)
plt.plot(green_P2,red_P2,label='measurement')
plt.legend()
#plt.savefig('res.png')

# %%
plt.hist(radiuslist,bins=25,histtype='step')
plt.minorticks_on()
plt.xlabel(r'tolerant $\sigma$')
plt.savefig('tol_hist_res.png')

# %%
x=np.linspace(-600,3000,10000)[1:-1]
plt.subplots(figsize=(15,6))
plt.plot(x,radiuslist)
plt.xlabel('V')
plt.ylabel(r'tolerant $\sigma$')
plt.savefig('tol_Vmap.png')
```