

消光系数报告

宋熙喆 PB21000301

1 实验过程

本实验旨在利用既有的激光雷达观测数据反演消光系数 σ 随高度的变化。激光雷达是一种强有力的远程感测工具，通过测量返回信号的强度，可以推断大气中的各种光学特性。激光雷达系统的基本原理是发射一束激光脉冲，并测量其在大气中的返回信号强度。根据返回信号的强度随距离的衰减规律，可以推断出大气中的消光系数和后向散射系数。

首先，读取 Mie 散射数据与背景噪声数据。Mie 散射是描述大气中悬浮颗粒物对激光光束的散射现象，通过观测 Mie 散射信号，可以获得大气中气溶胶颗粒的分布信息。背景噪声信号是指未受激光照射的背景光强度，为了获得准确的散射信号，需要扣除背景噪声信号。通过激光雷达的采样率，可以得到距离分辨率 $h = 0.015$ km，并据此确定每个采样点对应的高度 r 。散射信号与背景噪声信号之差的绝对值作为初始数据，这一步称为“扣背景”，以减少噪声对数据的影响。

数据预处理完成后，使用小波滤波方法对数据进行平滑处理。小波滤波是一种有效的信号处理技术，可以在保留信号特征的同时去除噪声。选择适当的小波基函数和分解层数，将原始信号进行小波分解，再对分解系数进行阈值处理，最后重构得到平滑后的信号 $P(r)$ 。

接下来，采用 Klett 方法进行反演计算。Klett 方法是一种稳定的解析反演方法，通过假设消光系数与高度之间的关系，来计算不同高度的消光系数。具体来说，常量 k 取 0.7，第 50 与第 300 个数据点分别对应 $r_b = 0.75$ km 与 $r_m = 4.5$ km。已知

$$S(r) \equiv r^2 P(r) \quad (1)$$

根据此公式，可以计算信号 $S(r)$ ，其表达式为

$$S(r) = \ln [r^2 P(r)] \quad (2)$$

反演过程中，消光系数 $\sigma(r)$ 的计算公式为：

$$\sigma(r) = \frac{\exp [(S(r) - S_m)/k]}{\sigma_m^{-1} + \frac{2}{k} \int_r^{r_m} \exp [(S(r') - S_m)/k], dr'} \quad (3)$$

其中， $S_m = S(r_m)$ 是参考点 r_m 处的信号值， σ_m 为参考点 r_m 处的消光系数，其近似计算公式为：

$$\sigma_m \approx \frac{\exp [(S_b - S_m)/k] - 1}{\frac{2}{k} \int_{r_b}^{r_m} \exp [(S(r') - S_m)/k], dr'} \quad (4)$$

通过数值积分，可以得到随高度变化的消光系数分布 $\sigma(r)$ 。数值积分是将连续函数的积分离散化，适合于计算复杂积分的近似值。在本实验中，采用梯形积分方法对公式中的积分部分进行数值计算，以提高计算精度。

最后，将计算得到的消光系数 $\sigma(r)$ 绘制成图像，展示消光系数随高度的变化趋势。通过分析图像，可以直观地了解大气中不同高度的消光系数分布情况。这对于研究大气中的气溶胶分布和光学特性具有重要意义。

2 代码

```
import numpy as np
import pywt
import matplotlib.pyplot as plt

primitive = np.loadtxt('200812151717A.txt')
noise = np.loadtxt('200812151718A.txt')
h = 0.015
r = primitive[:, 0] * h
data = np.abs(primitive[:, 1] - noise[:, 1])

wavelet = 'db4'
level = 5
coeffs = pywt.wavedec(data, wavelet, level=level)
thr = 0.5
coeffs_thresh = [pywt.threshold(c, thr, mode='hard') for c in coeffs]
P = pywt.waverec(coeffs_thresh, wavelet)

k = 0.8
b = 50
m = 300
S = np.log(r ** 2 * P)

I = 0
for t in range(b, m + 1):
    if t == b or t == m:
        I += h * np.exp((S[t] - S[m]) / k) / 2
    else:
        I += h * np.exp((S[t] - S[m]) / k)
sigma_m = (k * np.exp((S[b] - S[m]) / k) - 1) / (2 * I)

J = 0
```

```
sigma = [0]
for r0 in range(2, 1025):
    J = 0
    for t in range(r0, m + 1):
        if t == b or t == m:
            J += h * np.exp((S[t] - S[m]) / k) / 2
        else:
            J += h * np.exp((S[t] - S[m]) / k)
    sig = np.exp((S[r0] - S[m]) / k) / (1 / sigma_m + 2 * J / k)
    sigma.append(sig)

plt.figure()
plt.plot(r, sigma)
plt.xlim([0.5, 5.0])
plt.xlabel('Distance / km')
plt.ylabel('Extinction Coefficient')
plt.show()
```

3 实验结果

得到如图 ?? 所示，可见 σ 随 r 的变化趋势与预期在较小高度较为符合，在较大高度的差距大概是小波滤波故障导致。消光系数数值明显大于预期。

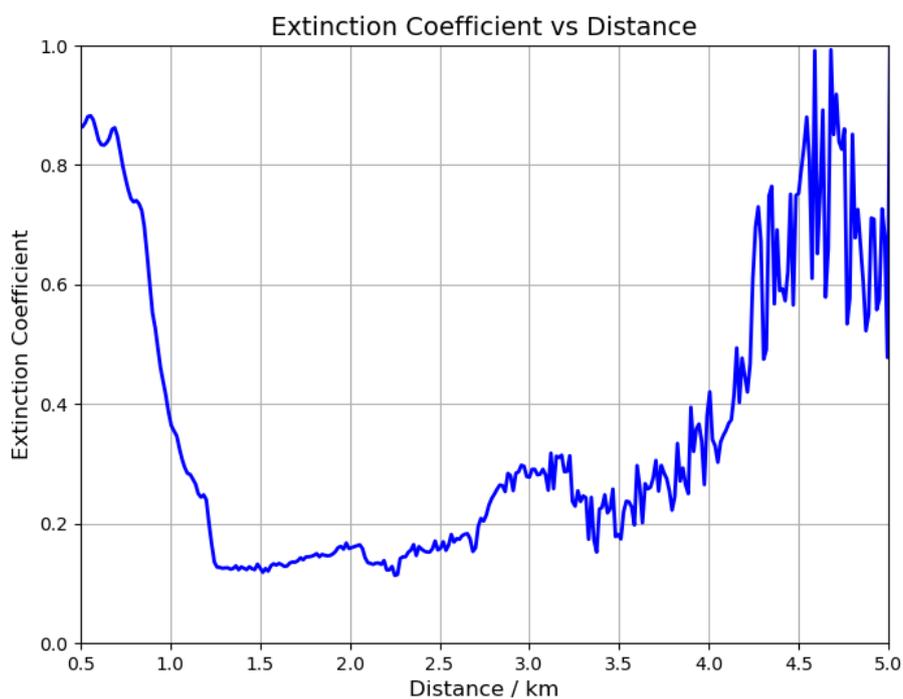


图 1: $\sigma-r$