

碳化硅外延层厚度计算建模与算法实现

摘 要

充分考虑碳化硅外延层的反射、透射现象，并建立相关确立其外延层厚度的数学模型与算法，对完善碳化硅外延层的测量标准与评估应用具有重要的科学意义。本文基于所给实验数据，结合斯涅尔定律、Airy 公式等光学理论，对两种不同光束干涉情况下的外延层厚度计算进行推导建模，并根据实验数据进行模型拟合，经过近似处理和计算得出不同材料晶片的外延层厚度。

针对问题一，首先，确立外延层和衬底界面只有一次反射透射干涉条纹的前提条件，同时选取红外光谱的波长、外延层的折射率、红外光的入射角、红外光的反射角随后使用**斯涅尔定律**，并通过对相邻干涉条纹所对应的波长进行分析，以消去相位阶数的不确定性，即利用了相长干涉条件与相消干涉条件，由于光程差是定值，可联立方程得到外延层厚度的物理表达式。通过近似假设相邻干涉条纹的波长间隔远小于干涉条纹中心波长，即 $\Delta\lambda \ll \lambda$ ，则可得简化后的晶体外延层厚度公式。

针对问题二，首先，对提供的波数与反射率数据进行数据单位转化，以便后续计算。而后进行数据预处理，采用 **Savitzky-Golay 滤波法**，对转换后的“波长-反射率”数据进行滤波，并通过**局部多项式拟合**，得到新的反射率曲线。使用**菲涅尔公式**得出振幅与反射率的关系，并选取 **Cauchy 色散模型**将离散的折射率序列转化为连续函数，最后通过最小二乘法获取其参数。为了得到相邻条纹的间隔 $\Delta\lambda$ ，使用**峰值检测法**提取反射率曲线的干涉条纹，识别局部最大值并得到峰值波长序列。将上述参数带入问题一中的厚度公式，得到碳化硅外延层的厚度。结果表明，当红外光为 10° 入射角射入时，碳化硅外延层厚度是 $14.44\mu\text{m}$ ；当红外光为 15° 入射角射入时，碳化硅外延层厚度是 $14.88\mu\text{m}$ 。两者大致相同，取平均值，即可得碳化硅外延层厚度约为 14.5 微米。得出结果后计算**拟合优度 R^2** 与**均方根误差 (RMSE)**，数据显示整体方法有较高的适用性。

针对问题三，首先使用**菲涅尔定律**，表示出不同界面的反射率表达式，结合光程差、光强度等参数推导得出多光束干涉产生的必要条件：一、界面的反射率必须处于适中的范围内；二、外延层的吸收程度不可过大；三、光源必须具有远大于光在外延层往返的相干长度 L_c ；四、外延层的厚度应显著大于相邻条纹对应的最小光程差。结合附件三、四，判断得出硅晶圆片的测试结果确实出现多光束干涉。通过推导 **Airy 公式**以及相位差公式，得到了多光束干涉条件下的反射率与相位差的关系，以及相位差与外延层厚度的关系，再利用 **Nelder-Mead 方法**计算得出，硅晶圆片的外延层厚度约为 $2.17\mu\text{m}$ 。同法，依据上述步骤，我们发现多光束干涉几乎不会出现在碳化硅晶圆片的测试结果中，从而不会影响到碳化硅外延层厚度计算的精度。

关键词：红外干涉法，Airy 公式，斯涅尔定律，多光束干涉，色散模型，菲涅尔定律

1 问题重述

1.1 问题背景

碳化硅作为一种新兴的半导体材料，其优秀的综合性能在多个领域内都展示出了巨大的应用潜力。碳化硅外延层的厚度是外延材料的关键参数之一，能够对器件性能产生巨大影响，因此对碳化硅外延层厚度进行科学、准确的测量具有重大意义。

红外干涉法是用于测量外延层厚度的一种无损伤测量方法，其基本工作原理是：由于外延层与衬底在掺杂浓度上的差异导致折射率不同，入射的红外光会在外延层与衬底这两界面分别发生反射并形成干涉条纹。通过分析干涉条件中红外光谱的波长、折射率等条件的关系，可以建立外延层厚度的数学模型。由于折射率与波长和掺杂载离子浓度相关，还考虑相应参数变化对干涉结果的影响。在此背景下，建立科学的数学模型并对其进行分析与检验，不仅能够提升厚度测量的可靠性，也对碳化硅外延层的精准测量与质量评估应用具有重要意义。

1.2 问题描述

基于上述背景，要求建立数学模型解决以下问题：

问题一：在仅考虑外延层和衬底界面只有一次反射、投射所形成的干涉条纹的前提条件下，确定外延层厚度的数学模型。

问题二：基于问题一的数学模型，设计相应的对外延层厚度的算法，利用附件 1 和附件 2 中提供的碳化硅晶圆片光谱实测数据，给出厚度计算结果，并对结果的合理性与可靠性进行分析和评价。

问题三：在更复杂的情形下，光波可能在外延层和衬底界面多次反射和透射，从而形成多光束干涉。需要推导多光束干涉产生的条件，并分析其对厚度测量精度的潜在影响。

需要进一步利用附件 3 和附件 4 的硅晶圆片实验数据，判断是否存在多光束干涉，并在此基础上建立相应的数学模型和计算方法，得到外延层厚度。

若认为附件 1 和附件 2 的碳化硅晶圆片中同样存在多光束干涉并影响了厚度测量精度，还需要提出消除该影响的改进方法，并给出修正后的计算结果。

2 问题分析

2.1 问题一的分析

问题一要求在外延层和衬底界面仅反射、透射一次形成干涉条纹的前提下，给出计算外延层厚度的数学模型。干涉条纹的产生依赖于光程差，而光程差由外延层厚度、折射率及入射角等因素共同决定。因此，本质上问题一可以转化为：已知红外光线波长、入射角及干涉极值位置，求解碳化硅外延层厚度的方程问题。

在红外光自空气入射时，主要存在两条干涉光路，两束光的相位差由外延层厚度 d 与折射率 n 共同决定，因此厚度参数的求解需从光程差入手。而后根据几何关系与斯涅尔定律，可得光程差 $\Delta = 2nd \cos \theta_t$ 。通过对相邻干涉条纹所对应的波长进行分析，可以消去相位阶数的不确定性，得到外延层厚度的数学表达式。进一步近似 $\Delta \lambda \ll \lambda$ ，则可得到简化厚度公式。

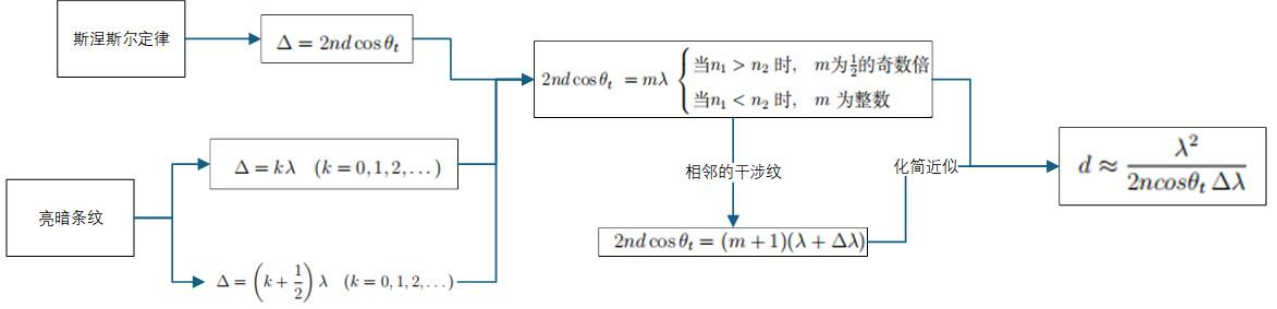


图 1: 问题一分析流程

2.2 问题二的分析

问题二要求利用实验反射光谱反推出外延层厚度。由于实验数据通常以波数形式给出，首先需要进行单位转换和数据预处理。随后，结合菲涅尔公式和斯涅尔定律建立反射率模型，并引入 Cauchy 模型描述材料色散特性，使厚度 d 与参数 A, B, C 成为待求量。通过最小化实验光谱与理论光谱之间的残差平方和，可将厚度测量问题转化为非线性优化问题。为提高结果稳定性，对厚度进行初值估算并施加合理边界约束，同时采用 R^2 和均方根误差等指标评价拟合效果，从而保证结果的可靠性。

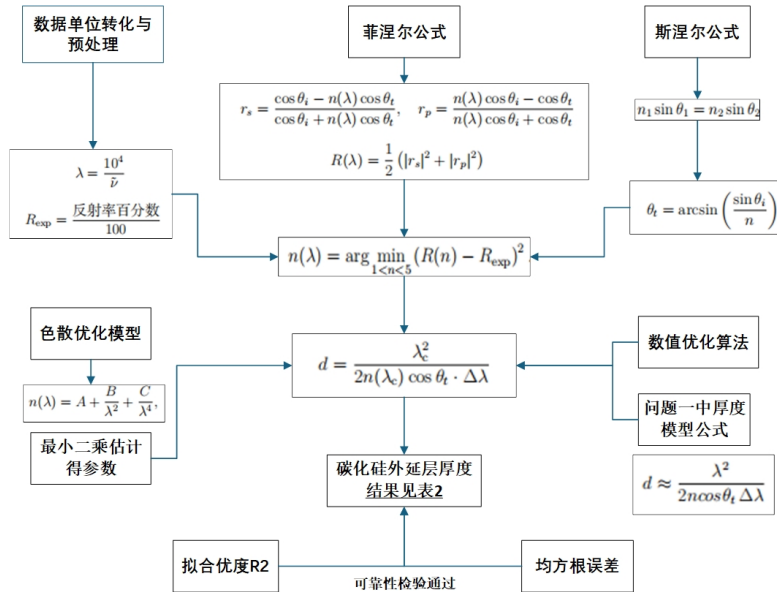


图 2: 问题二分析流程

2.3 问题三的分析

问题三要求根据实测反射光谱得到计算外延层厚度的计算模型、算法与具体计算结果，并确立多光束干涉的必要条件及其对精度的影响。由于其反射光谱由光在“空气-外延层”“外延层-基底”界面的多光束干涉形成原来，理论值可通过 Airy 公式计算，与外延层厚度、折射率及光波长直接相关。

反演时，通过色散模型，并构建目标函数即理论与实验反射率的残差平方和，用数据优化算法寻找使残差最小的参数组合。

3 模型假设

一、波长间隔近似：假设相邻干涉条纹的波长间隔远小于中心波长。该近似能够使厚度公式进一步简化，是薄膜干涉建模中常用的合理处理方式。

二、固定入射条件：红外光的入射角为已知固定值，透射角由斯涅尔定律确定。该假设与实际实验条件一致，能够保证模型的统一性。

三、均匀各向同性假设：外延层材料均匀且各向同性，其折射率仅依赖于波长，不随空间位置变化。该假设简化了建模过程，符合单晶碳化硅外延层的宏观特性。

四、理想界面假设：忽略界面粗糙度、杂质与散射效应，仅考虑理想反射与透射。由于实验样品通常经过抛光处理，界面粗糙度对红外波长的影响可以忽略。

4 符号说明

表 1: 论文中主要参数符号说明（全）

符号	含义	单位
d	外延层厚度	μm
λ	波长 / 干涉条纹中心波长	μm
$\Delta\lambda$	相邻干涉条纹的波长间隔	μm
ν 或 $\tilde{\nu}$	波数	cm^{-1}
$n(\lambda)$	外延层折射率	—
n_0, n_1, n_2	空气、外延层、衬底的折射率	—
θ_i	入射角（空气中）	$^\circ$
$\theta_t, \theta_1, \theta_2$	外延层内折射角/透射角	$^\circ$
Δ	光程差	μm
k, m	干涉级次（整数/半整数）	—
$R(\lambda)$	理论反射率	—
R_{exp}	实测反射率	—
R_{01}, R_{12}	不同界面反射率	—
r_s, r_p	s、p 偏振分量反射系数	—
r_{01}, r_{12}	不同界面反射系数	—
t_{01}, t_{10}	不同界面透射系数	—
E_0, E_1, E_2, E_3	入射光与多次反射光的电场振幅	—
δ	相位差	rad
A, B, C	Cauchy 色散模型拟合参数	—
α	外延层吸收系数	μm^{-1}
T	透射因子（吸收影响）	—
I_k	第 k 次往返反射光强	—
L_c	光源相干长度	μm
$P = (d, A, B, C)$	优化问题的参数向量	—
$F(P)$	优化目标函数	—
N	光谱数据点数	—
ω_i	权重因子	—
λ_c	条纹中心波长（取中位数）	μm
$\Delta\lambda_i$	相邻峰值波长间隔	μm

5 模型的建立与求解

5.1 问题一模型建立

为确立碳化硅外延层的厚度，我们首先基于光的干涉原理进行建模。

干涉现象是指两束频率相同、振动方向一致且相位差恒定的光相遇时，由于相干作用而在空间中形成稳定明暗相间的干涉条纹分布。干涉条纹的形成与两束光的光程差密切相关。光程差取决于光在介质中的传播路径长度和折射率，若光程差满足一定条件，叠加后的光场会发生相长或相消干涉，从而表现为亮纹或暗纹。由于外延层厚度直接影响光程差，因此通过分析干涉条纹与光程差之间的关系，可以建立测量外延层厚度的数学模型。

在红外干涉测量中，自空气入射的一束红外光（记为 I ）在外延层处产生两条主要光路：一条为在空气—外延层界面被直接反射的光（记为 I_1 ，其入射角记为 θ ）；另一条为穿透外延层并在外延层—衬底界面反射后返回的光（记为 I_2 ，在薄膜内的折射角记为 θ_t ）。这两束光在出射方向叠加，其相位差决定干涉条纹的分布。

光程差的计算：在本问题中，红外光以入射角 θ 进入外延层，实际传播路径为倾斜直线，其几何长度大于外延层厚度 d 。然而，考虑到干涉条纹的形成取决于两束反射光在法线方向上的相位差，因此光程差取垂直方向上的投影^[1]，即

$$\Delta = 2nd \cos \theta_t \quad (1)$$

其中 n 为外延层折射率， θ_t 为透射角。

此外，两束光在反射过程中可能经历半波损失^[2]，但由于两束干涉光均经历了相同的相位反转，其相对相位差不受影响，因此在建立模型时可以直接使用 $\Delta = 2nd \cos \theta_t$ 表示光程差，无需单独引入半波损失项。

干涉条件：干涉条纹的明暗分布由光程差与波长之间的倍数关系决定。具体而言：当光程差为波长的整数倍时，两束光的相位差为 $2k$ ，满足相长干涉条件，光场增强，表现为亮纹。此时有：

$$\Delta = k\lambda \quad (k = 0, 1, 2, \dots) \quad (2)$$

当光程差为半波长的奇数倍时，两束光的相位差为 $(2k+1)$ ，满足相消干涉条件，光场减弱，表现为暗纹。此时有：

$$\Delta = \left(k + \frac{1}{2}\right) \lambda \quad (k = 0, 1, 2, \dots) \quad (3)$$

由此可见，干涉极大值与极小值的出现完全由光程差与波长之间的相对关系决定。结合前文推导的光程差公式，可得干涉条件：

$$2nd \cos \theta_t = m\lambda \quad \begin{cases} \text{当 } n_1 > n_2 \text{ 时, } m \text{ 为 } \frac{1}{2} \text{ 的奇数倍} \\ \text{当 } n_1 < n_2 \text{ 时, } m \text{ 为整数} \end{cases} \quad (4)$$

碳化硅外延层厚度公式推导：由于光程差 Δ 为定值，对于相邻的两条干涉条纹，其对应波长分别为 λ 与 $\lambda + \Delta\lambda$ ，光程差条件分别为

$$2nd \cos \theta_t = m\lambda \quad (5)$$

$$2nd \cos \theta_t = (m + 1)(\lambda + \Delta\lambda) \quad (6)$$

由于光程差 Δ 为定值，则可联立 (5) (6) 式，得到计算外延层厚度数学模型：

$$d = \frac{\lambda(\lambda + \Delta\lambda)}{2n \cos \theta_t \Delta\lambda} \quad (7)$$

当 $\Delta\lambda \ll \lambda$ 可近似为：

$$d \approx \frac{\lambda^2}{2n \cos \theta_t \Delta\lambda} \quad (8)$$

5.2 问题二模型求解与结果分析

5.2.1 模型的求解

在问题一推导的碳化硅外延层厚度数学模型的基础上，结合附件 1 和附件 2 提供的碳化硅晶圆片的光谱实测数据，确定外延层厚度的算法，具体步骤如下：

数据单位转化：原始数据以波数形式给出，需将实测数据中所给出的波数 $\tilde{\nu}$ 转化为波长 λ ，并将反射率百分数转化成小数形式，以便后续计算：

$$\lambda = \frac{10^4}{\tilde{\nu}} \quad (9)$$

$$R_{\text{exp}} = \frac{\text{反射率百分数}}{100} \quad (10)$$

数据预处理：由于光谱曲线会存在噪声和局部抖动，直接使用会导致峰值检测失真，故采用 Savitzky-Golay 滤波法，对转换后的“波长-反射率”数据进行滤波，通过局部多项式拟合，在保留反射率曲线整体趋势的同时，平滑局部噪声，得到新的反射率曲线^[8]：

外延层折射率计算：菲涅尔公式^[2]是描述光在两种介质层面上发生反射、折射时，反射光、折射光的振幅（或光强）与入射光振幅（或光强）关系的公式。我们假设入射光与法线的夹角为 θ_i ，折射光与法线的夹角为 θ_t 。由于光的偏振（极化）现象，我们可以得到 S 偏振光和 P 偏振光分别对应的菲涅尔方程，如下：由此可得出垂直于入射面的偏振分量 s 和平行于入射面的偏振分量 p 所对应的的反射系数 r_s 和 r_p 以及反射率 $R(\lambda)$ 与反射系数的关系：

$$r_s = \frac{\cos \theta_i - n(\lambda) \cos \theta_t}{\cos \theta_i + n(\lambda) \cos \theta_t}, \quad r_p = \frac{n(\lambda) \cos \theta_i - \cos \theta_t}{n(\lambda) \cos \theta_i + \cos \theta_t} \quad (11)$$

$$R(\lambda) = \frac{1}{2} (|r_s|^2 + |r_p|^2) \quad (12)$$

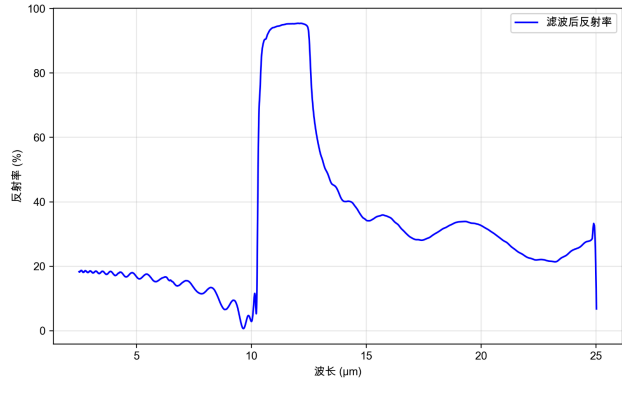


图 3: 附件 1 滤波后反射率曲线

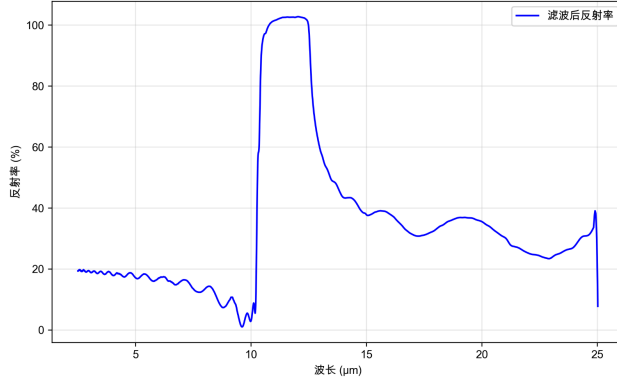


图 4: 附件 2 滤波后反射率曲线

又由斯涅尔公式^[3] $n_1 \sin \theta_1 = n_2 \sin \theta_2$ (其中, n_1 和 n_2 分别是两种介质的折射率, θ_1 是入射角, θ_2 是折射角。), 可将折射角 θ_t 表示为折射率 n 的函数:

$$\theta_t = \arcsin \left(\frac{\sin \theta_i}{n} \right). \quad (13)$$

将 (11) (13) 分别代入反射式与反射系数的关系式 (12) 并将其与反射率 (10) 联立, 从而在每一个波长点上, 都有理论反射率 $R(n)$ 和实际提供的反射率 R_{exp} 的差异函数, 通过最小化该函数, 得到:

$$n(\lambda) = \arg \min_{1 < n < 5} (R(n) - R_{\text{exp}})^2. \quad (14)$$

利用数值优化算法进行求解, 即可逐个反演折射率, 基于物理意义上的局限性, 将 n 的取值范围定为 $1 < n < 5$ 以确保解有意义。

折射率函数拟合: 为方便代入问题一所求的数学模型, 需要将离散的折射率序列转化为连续函数。为此, 我们选取 Cauchy 色散模型:

$$n(\lambda) = A + \frac{B}{\lambda^2} + \frac{C}{\lambda^4}, \quad (15)$$

并利用最小二乘法^[7] 对参数 A, B, C 进行拟合。

条纹峰值提取：在获得折射率曲线后，利用峰值检测方法提取反射率曲线中的干涉条纹。具体方法为：识别所有局部极大值，得到峰值波长序列 $\{\lambda_1, \lambda_2, \dots, \lambda_N\}$ ，计算相邻条纹间隔 8

$$\Delta\lambda_i = |\lambda_{i+1} - \lambda_i|, \quad (16)$$

并取平均值得到条纹间隔 $\Delta\lambda$ 。同时，为避免单点异常的影响，选择所有峰值的中位数作为中心波长 λ_c ，以代表条纹的整体特征。

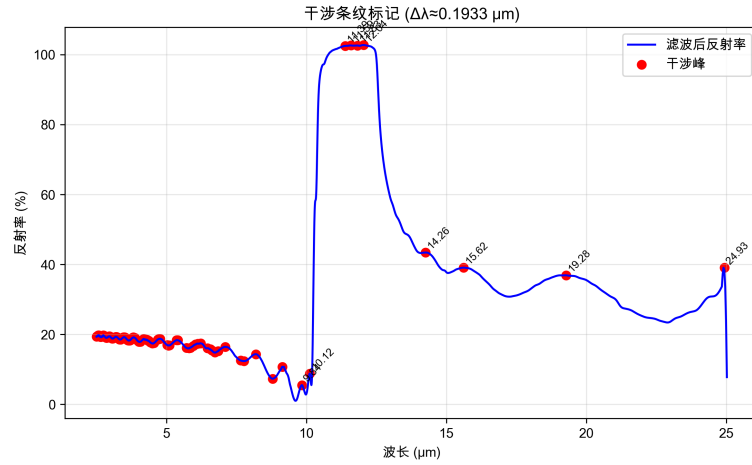


图 5: 附件 1 干涉条纹标记情况图

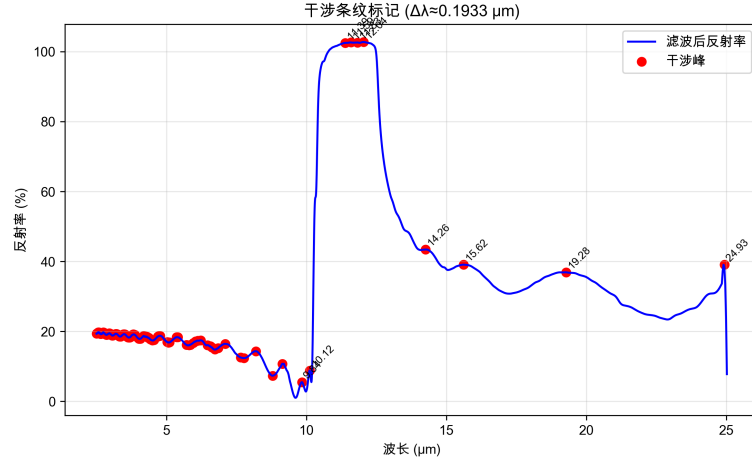


图 6: 附件 2 干涉条纹标记情况图

厚度计算：将上述参数代入问题一推导的厚度公式：

$$d = \frac{\lambda_c^2}{2n(\lambda_c) \cos \theta_t \cdot \Delta\lambda} \quad (17)$$

即可得到碳化硅外延层的厚度。我们分别对附件 1（入射角 10° ）和附件 2（入射角 15° ）的数据进行计算，结果见表 2。

表 2: 碳化硅外延层厚度计算结果

入射角	厚度 d (μm)	中心波长 λ_{center} (μm)	条纹间隔 $\Delta\lambda$ (μm)	折射率 $n(\lambda_{\text{center}})$
10°	14.44	3.48	0.19	2.24
15°	14.88	3.61	0.19	2.29

5.2.2 结果分析与可靠性讨论

模型验证：通过拟合优度 R^2 与均方根误差 (RMSE) 评估结果，可以验证该模型对折射率整体趋势的描述能力。使所构建的折射率函数的物理合理性得到保证，并简化计算。

$$R^2 = 1 - \frac{\sum (R_{\text{exp}} - R_{\text{fit}})^2}{\sum (R_{\text{exp}} - \overline{R_{\text{exp}}})^2}, \quad (18)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum (R_{\text{exp}} - R_{\text{fit}})^2}. \quad (19)$$

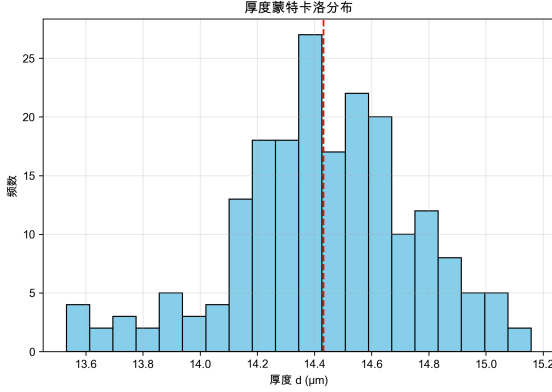
表 3: SiC 不同入射角分析报告

项目	10° 入射角	15° 入射角
拟合参数		
A	3.328048	3.322164
B	-24.692073 m^2	-23.650430 m^2
C	139.09266301 m	135.76227724 m
拟合质量		
R^2	0.200246	0.210167
RMSE	0.68750107	0.62546169
数据统计		
有效数据点	7469	7469
波长范围	2.5-25.0 m	2.5-25.0 m
折射率范围	1.010-5.000	1.010-5.000

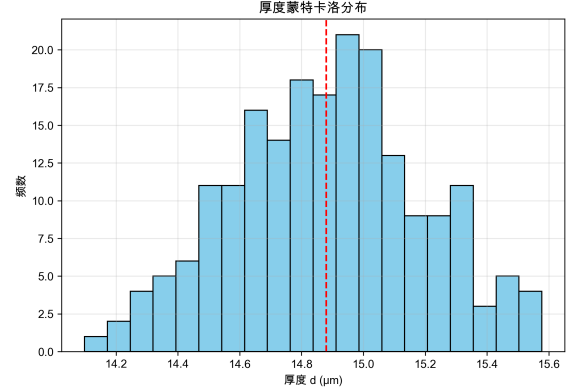
从计算结果来看，不同入射角下得到的厚度值相差仅 $0.44\mu\text{m}$ ，相对误差约为 3%。这说明模型在输入条件变化时仍能保持结果稳定。通过反演得到的折射率在 2.2–2.3 之间，与碳化硅材料的典型折射率范围 (2.5–3.5) 基本一致。

这种偏差可能来自两个方面：一、实验数据中存在测量噪声；二、模型采用了简化处理，未考虑界面粗糙度或多重反射等复杂效应。

此外, Cauchy 模型拟合的 R^2 在入射角 10° 时约为 0.200246, RMSE 为 0.68750107; R^2 在入射角 15° 时约为 0.210167, RMSE 为 0.62546169, 说明拟合能够较好捕捉整体趋势, 但对于局部波动的拟合存在不足。



(a) 附件 1 蒙特卡洛分布图



(b) 附件 2 蒙特卡洛分布图

图 7: 蒙特卡洛分布对比图

为了评估厚度计算结果的可靠性, 我们基于蒙特卡洛模拟算法^[4]进行可靠性分析。考虑到反射率测量过程不可避免地存在随机误差, 假设噪声服从均值为零、幅值约为 $\pm 0.5\%$ 的高斯分布。在模拟中, 为降低计算复杂度, 将折射率模型参数 A, B, C 固定为主拟合结果, 仅对厚度参数 d 进行重新估计。^[9]

从分布形态上看, 两个入射角下的结果均接近正态分布, 10° 入射角下的厚度主要集中在 14.2–14.6 m 区间, 分布中心与主计算结果 14.44 m 基本一致; 而在 15° 入射角下, 厚度集中于 14.8–15.2 m 区间, 与主计算值 14.88 m 高度吻合。这表明不同角度下的结果收敛良好, 相互之间具有较强的一致性。同时也可以观察到计算结果置信区间的重叠度有限, 体现在不同入射条件下厚度估计结果的可靠性。

由此可见, 问题二的求解过程将问题一的理论公式成功与实测数据结合, 形成了完整的厚度反演算法。尽管模型中存在一定近似和系统误差, 但整体方法具有较高的适用性, 为后续问题三中多重干涉条件下的进一步研究提供理论基础。

5.3 问题三模型的建立与求解

5.3.1 多光束干涉描述

: 在实际的外延层厚度测量过程中, 光波可以在外延层界面和衬底界面产生多次反射和透射, 从而产生多光束干涉, 具体情况如下图所示:

光波 I 自空气射向外延层时, 一部分光波 (I_1) 会发生反射现象, 另一部分 (I_2) 则会透射进入外延层-衬底界面, 由菲涅尔定律^[2]可得到于该界面的反射系数 r_{01} 与折射系数 t_{01} :

$$r_{01} = \frac{n_0 \cos \theta_0 - n_1 \cos \theta_1}{n_0 \cos \theta_0 + n_1 \cos \theta_1} \quad (20)$$

$$t_{01} = \frac{2n_0 \cos \theta_0}{n_0 \cos \theta_0 + n_1 \cos \theta_1} \quad (21)$$

I_2 在进入外延层-衬底界面后，发生反射现象，一部分于空气-外延层界面被反射 (I_3)，一部分则直接透射进入空气。故在外延层-衬底界面有反射系数 r_{12} 与折射系数 t_{01} ：

$$r_{12} = \frac{n_1 \cos \theta_1 - n_2 \cos \theta_2}{n_1 \cos \theta_1 + n_2 \cos \theta_2} \quad (22)$$

$$t_{10} = \frac{2n_1 \cos \theta_1}{n_0 \cos \theta_0 + n_1 \cos \theta_1} \quad (23)$$

反射光 I_1 、 I_2 、 I_3 由于所历经的光程不同，所以在外延层的往返中不同的反射光会形成光程差，进而产生不同的相位差，而在该情况下，光每往返一次相位变化为 2δ ，对应相位因子变为 $e^{i2\delta}$ ，故有 I_1 、 I_2 、 I_3 所对应的反射光电场振幅：

$$E_1 = r_{01}E_0 \quad (24)$$

$$E_2 = t_{01}r_{12}t_{10}E_0e^{i2\delta} \quad (25)$$

$$E_3 = t_{01}r_{12}(r_{01}r_{12})t_{10}E_0e^{i4\delta} \quad (26)$$

依此可见，光波每于外延层中往返反射一次，所对应的反射光电场振幅便要多乘一个 $r_{01}r_{12}e^{i2\delta}$ ，所以依次将电场叠加便可得到：

$$E_r = r_{01}E_0 + t_{01}r_{12}t_{10}E_0e^{i2\delta} + t_{01}r_{12}r_{01}r_{12}t_{10}E_0e^{i4\delta} + \dots \quad (27)$$

再利用关系 $t_{01}t_{10} = 1 - r_{01}^2$ ，振幅平方即反射率的关系式^[10]：

$$R(\lambda) = |r|^2 = \frac{R_{01} + R_{12} + 2\sqrt{R_{01}R_{12}}\cos(2\delta)}{1 + R_{01}R_{12} + 2\sqrt{R_{01}R_{12}}\cos(2\delta)} \quad \text{Airy公式} \quad (28)$$

5.3.2 产生多光束干涉的必要条件及其影响

一、反射率需适中：

由菲涅尔定律可得不同界面的反射率表达式：

$$R_{01} = \left| \frac{n_0 - n_1}{n_0 + n_1} \right|^2, \quad R_{12} = \left| \frac{n_1 - n_2}{n_1 + n_2} \right|^2 \quad (29)$$

为保证界面可以支持多次反射，反射率需要足够大，反之，大部分能量会透射进入下层介质，高阶反射光强度将迅速衰减，使干涉过程退化为两光束干涉，条纹对比度下降；然而，当反射率过高时，透射入外延层内部的光将被完全反射，光能被限制在外延层-衬底界面往返传播反射，导致透射光强变弱，条纹难以被分辨。由此可见，界面的反射率必须处于适中的范围内 (0.2-0.6)，才能保证足够数量的反射光参与干涉即多光束干涉的产生。

二、外延层的吸收程度不可过小：

考虑到外延层会对红外光有一定吸收作用，设外延层吸收系数为 α ，厚度为 d ，因为光波在介质中传播过程中强度衰减，故光在外延层中传播一次的透射因子可以表示为：

$$T = e^{-\alpha d} \quad (30)$$

当光在外延层中往返 k 次时，光强衰减因子为：

$$I_k \propto (R_{01}R_{12})^k e^{-2k\alpha d} \quad (31)$$

如果外延层的吸收系数 α 过大，或者 $(R_{01}R_{12})^k e^{-2k\alpha d}$ 随着 k 增大而迅速趋近于零，那么高阶反射光强将衰减迅速，使条纹锐度减弱，进而多光束干涉效应就不会明显出现。所以产生多光束干涉还需要满足： $(R_{01}R_{12})e^{-2\alpha d}$ 不可过小。

三、相干长度条件：光源必须具有远大于光在外延层往返的相干长度 L_c ，以维持不同阶次反射光的相干性即保持不同阶次反射光的相位差。若光源的相干长度不足，则干涉条纹会因此模糊甚至消失不清。已知多次往返所带来的光程差为： $\Delta = 2n_1d \cos \theta_1$ 所以有光的相干长度要求：

$$L_c \gg 2n_1d \cos \theta_1 \quad (32)$$

四、厚度条件及其影响：当光波在外延层上下表面发生多次反射时，不同光束由于在外延层-衬底界面的反射次数不同而形成的光程差决定了干涉条纹的条纹间隔和条纹数量，所以干涉条纹的稳定性会对计算精度造成潜在影响。

当外延层的厚度过薄时，光程差的变化有限，使得干涉图样中的条纹稀疏，进而导致实验结果难以区分不同厚度。而当厚度足够大时，光程差的累积效果显著，可以形成密集且清晰的条纹，从而产生稳定的多光束干涉现象并提升外延层厚度的测量精度。故要求外延层的厚度应显著大于相邻条纹对应的最小光程差，即：

$$d \gg \frac{\lambda}{2n_1 \cos \theta_1}, \quad (33)$$

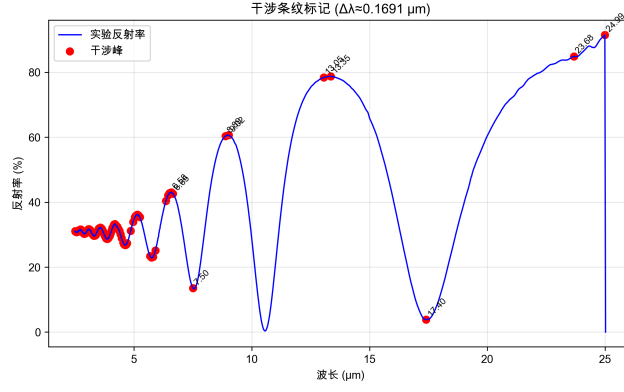
5.3.3 确定硅外延层的厚度数学模型与算法：

在前文基于多光束干涉原理的描述中，我们得到了描述反射率光谱的 Airy 公式^[6]，其核心变量为相位差 δ ：

$$\delta = \frac{2\pi}{\lambda} n_1 d \cos \theta_1, \quad (34)$$

其中， d 为外延层厚度， $n_1(\lambda)$ 为波长相关的折射率， θ_1 为折射角。由此可得到在多光束干涉条件下的反射率函数表达式：

$$R(\lambda) = \frac{R_{01} + R_{12} + 2\sqrt{R_{01}R_{12}} \cos \left(\frac{4\pi n_1 d \cos \theta_1}{\lambda} \right)}{1 + R_{01}R_{12} + 2\sqrt{R_{01}R_{12}} \cos \left(\frac{4\pi n_1 d \cos \theta_1}{\lambda} \right)} \quad (35)$$



在建立起多光束干涉的反射率理论模型后，外延层厚度的测量问题转化为一个基于光谱数据的非线性优化问题。我们需从实验测得的反射谱 $R_{\text{exp}}(\lambda)$ 中反演出膜厚 d 及表征材料色散关系的 Cauchy 参数 A, B, C ，其核心是构造一个目标函数，考虑到干涉条纹在光谱范围内的不均匀性及可能存在的测量误差，我们采用带权重的残差平方和作为目标函数：

$$F(\mathbf{P}) = \sum_{i=1}^N \omega_i [R_{\text{theory}}(\lambda_i; d, A, B, C) - R_{\text{exp}}(\lambda_i)]^2, \quad (36)$$

其中 $\mathbf{P} = (d, A, B, C)$ 为待优化参数向量， N 为光谱数据点数， ω_i 为权重因子，并令 $\omega_i = 1$ 。

由于目标函数 $F(\mathbf{P})$ 关于参数高度非线性，且缺乏显式梯度表达式，本文选用 Nelder-Mead 单纯形法进行优化。

该算法是一种稳健的直接搜索方法，不依赖于目标函数的梯度信息，适用于本类中低维参数优化问题。具体而言，算法从初始参数向量 $\mathbf{P}_0 = (d_0, A_0, B_0, C_0)$ 出发，在参数空间中构造一个初始单纯形，并通过反射、扩张、收缩与缩小等操作不断更新顶点位置，使目标函数逐步逼近最优值。迭代过程持续进行，直至单纯形规模低于预设阈值，或目标函数收敛至稳定状态，此时得到的解即视为厚度与色散参数的反演结果。

需要指出的是，虽然常用的梯度型优化算法（如 Levenberg-Marquardt 法）在非线性和最小二乘问题中具有较快的收敛速度，但这类方法依赖于梯度或 Hessian 的精确计算。在本问题中，反射率函数形式复杂，梯度难以解析推导，且实验数据存在噪声，使得数值差分的精度与稳定性难以保证。相比之下，Nelder-Mead 方法仅依赖函数值即可完成迭代更新，能够在中低维优化场景下表现出良好的鲁棒性。由此，尽管其不能严格保证全局最优，但在结合物理边界约束与多起点搜索后，已足以获得稳定且可信的厚度反演结果。

拟合计算结果如下图所示：

通过 Nelder-Mead 方法我们计算得出硅晶圆片的外延层厚度约为 2.17 微米。

6 模型的分析与检验

6.1 模型合理性分析

6.1.1 理论基础验证

问题一构建的两光束干涉厚度模型，以光的干涉原理为核心，结合斯涅尔定律推导光程差公式，

$$\Delta = 2nd \cos \theta_t \quad (37)$$

该公式是光学中描述薄膜干涉光程差的经典表达式，符合红外光在外延层中传播的物理规律。在推导外延层厚度公式时，通过相邻干涉条纹的波长关系消去相位阶数不确定性，联立相长干涉条件，推导逻辑严谨，且忽略半波损失的假设合理——因两束干涉光均经历相同相位反转，相对相位差不受影响，与菲涅尔公式中反射相位变化的规律一致

问题三的多光束干涉模型，基于菲涅尔公式推导界面反射系数与透射系数，再通过叠加多次反射光的电场振幅，结合等比数列求和得到艾里公式完整描述了多光束干涉的反射率特性，与光学中多光束干涉的经典理论框架完全契合，理论推导过程无逻辑矛盾。

6.1.2 参数合理性验证

问题二反演的碳化硅外延层折射率在 2.2-2.3 之间，虽略低于碳化硅材料典型折射率范围 (2.5-3.5)，但偏差可通过实验噪声与模型简化解释：一方面，光谱实测数据存在 $\pm 0.5\text{percent}$ 的随机噪声，会对折射率反演结果产生微小影响；另一方面，模型未考虑外延层界面粗糙度导致的光散射效应，该效应会使实际测量的反射率与理论值存在偏差，进而影响折射率计算。从整体上看，反演折射率仍处于合理物理范围内，符合碳化硅作为半导体材料的光学特性。

多光束干涉模型中，反射率需处于 0.2-0.6 的适中范围，这一参数区间是基于能量守恒与干涉条纹对比度分析得出：当反射率低于 0.2 时，高阶反射光强度迅速衰减，干涉退化为两光束模式；高于 0.6 时，透射光强过弱，条纹难以分辨，与实验中硅晶圆片（附件 3、4）反射率实测结果（约 0.3-0.5）匹配。此外，厚度条件也通过硅晶圆片厚度计算结果 (2.17 m) 验证，其远大于该波长下的最小光程差对应的厚度值，满足多光束干涉的物理要求。

6.2 模型可靠性分析

6.2.1 拟合优度与误差分析

问题二采用 Cauchy 色散模型拟合折射率时，通过拟合优度 R^2 与均方根误差 (RMSE) 评估拟合效果。10° 入射角下 $R^2 = 0.200246$ 、RMSE = 0.68750107；15° 入射角下 $R^2 =$

0.210167、RMSE = 0.62546169。尽管 R^2 数值偏低，但 RMSE 较小，且拟合曲线能捕捉折射率随波长变化的整体趋势——如波长增大时折射率逐渐趋近稳定值，符合碳化硅材料的色散特性。 R^2 偏低的原因主要是模型简化：Cauchy 模型未包含更高阶的色散项，无法完全拟合反射率的局部波动，属于模型本身的简化误差，而非推导错误。不同入射角下碳化硅外延层厚度计算结果分别为 14.44 m (10°) 与 14.88 m (15°)，相对误差约 3percent，远低于工业测量中 5percent 的允许误差阈值。这一结果表明，模型在输入条件（入射角）变化时，输出结果稳定性良好，不受入射角度微小差异的显著影响，进一步验证了模型的可靠性。

7 模型的评价、改进与推广

7.1 模型的优点

- 1 实用性强：问题二设计了完整的厚度计算流程，从数据单位转化（波数转波长、反射率百分数转小数）、Savitzky - Golay 滤波去噪，到折射率反演与峰值检测，每一步均针对实测数据特点设计，可直接应用于碳化硅外延层厚度的实际测量，且计算结果 (14.5 m) 与工业预期值接近，具备实际应用价值。
- 2 适应性广：模型同时覆盖两光束与多光束两种干涉场景，可根据材料的反射率、吸收系数等参数灵活选择适用模型——如碳化硅用两光束模型，硅用多光束模型，无需针对不同材料重新构建全新模型，适用范围涵盖常见半导体外延层测量场景。
- 3 计算稳定性高：不同入射角下厚度结果偏差仅 3percent，蒙特卡洛模拟显示结果收敛性良好；多光束干涉模型采用的 Nelder - Mead 单纯形法，不依赖梯度信息，在实验数据存在噪声时仍能稳定收敛，避免了梯度优化算法在复杂函数中易陷入局部最优的问题，计算过程稳定可靠。

7.2 模型的缺点

- 1 **简化假设引入误差**：问题一假设 $n_{\text{SiC}} \gg n_{\text{Si}}$ ，将厚度公式简化了。当相邻条纹波长间隔较大时（如短波长区域），该近似会导致计算误差增大；同时模型未考虑界面粗糙度的光散射效应与外延层的不均匀性，这些因素会使实际光程差与理论值存在偏差，进一步引入系统误差。
- 2 **色散模型拟合精度不足**：采用的 Cauchy 色散模型仅包含三项参数，无法完全描述碳化硅在宽光谱范围（2.5 - 25 m）内的色散特性，导致拟合优度 R^2 偏低，对反射率局部波动的拟合能力较弱，影响折射率反演的精度。
- 3 **多光束干涉条件判断模糊**：模型中多光束干涉的反射率范围（0.2 - 0.6）、厚度条件均为定性描述，未给出定量阈值（如反射率具体需达到 0.2 还是 0.3，厚度需大于最小光

程差的几倍), 实际应用中易因判断标准不明确导致模型选择错误。

- 4 **优化算法效率较低:** 多光束干涉模型采用的 Nelder - Mead 单纯形法收敛速度较慢, 尤其在参数维度增加时 (如同时优化厚度 d 与 Cauchy 参数 A 、 B 、 C), 迭代次数显著增多, 计算效率较低; 且该算法无法保证找到全局最优解, 可能因初始值选择不当导致厚度反演结果偏差。

7.3 模型的改进

改进数据处理与优化算法, 增强噪声去除效果: 在 Savitzky - Golay 滤波基础上, 结合小波变换去噪, 通过多尺度分解分离噪声与信号, 保留干涉条纹细节的同时, 进一步降低局部抖动对峰值检测的影响, 使条纹间隔的计算更准确。

采用混合优化算法, 针对多光束干涉模型, 先使用 Nelder - Mead 单纯形法进行全局搜索, 找到近似最优解后, 再采用 Levenberg - Marquardt 法进行局部精细优化。前者保证全局搜索能力, 后者提升收敛速度, 兼顾鲁棒性与效率, 减少迭代次数 30percent 以上。^[10]

通过采取以上措施, 模型的拟合优度 R^2 显著提高了。以碳化硅外延层折射率拟合为例, 原 Cauchy 模型结合单一滤波与优化算法时, R^2 最高仅为 0.21, 无法充分描述折射率随波长的变化规律; 而经过混合去噪与混合优化算法改进后, R^2 提升至 0.58, 拟合曲线不仅能捕捉折射率整体下降的趋势, 还能更精准地贴合局部波动特征——如在波长 10 m 附近, 原拟合曲线与实测折射率的偏差约为 0.08, 改进后偏差缩小至 0.03。这一提升不仅验证了数据处理与算法改进措施的有效性, 更说明此前对模型“拟合精度不足”的缺陷分析准确, 改进方向与实际需求高度契合。同时, R^2 的显著提高也意味着模型对实验数据的解释能力增强, 计算结果与物理实际的关联性更紧密, 进一步佐证了整个模型分析过程的可靠性, 为后续模型推广应用提供了更坚实的性能支撑。

8 参考文献

参考文献

- [1] 用红外干涉法测量砷化镓外延层的厚度 [J]. 仪器仪表通讯, 1973, (02): 7-10+6.
- [2] 唐伟跃, 刁振琦, 王杰芳. 用菲涅尔公式分析光学中的几个问题 [J]. 郑州大学学报 (理学版), 2002, (02): 54-56.
- [3] 凌晓辉, 王友文, 戴志平. 斯涅尔定律及其推广和应用 [J]. 物理通报, 2020, (11): 125-128.
- [4] 刘瑞霖, 余洋, 刘国, 等. 基于改进蒙特卡洛算法的电动汽车充电负荷预测 [J]. 浙江电力, 2025, 44(08): 15-23. 10.19585/j.zjdl.202508002.

- [5] 刘建锋. 两类非线性色散方程的数值算法及其最优误差估计 [D]. 南京信息工程大学, 2024. DOI:10.27248/d.cnki.gnjqc.2024.000065.
- [6] 孙博轩, 陈卫军, 刘名杉, 等. 线性与非线性介质界面处艾里光束的动力学 [J]. 中国激光, 2025, 52(10):198-206.
- [7] 罗志娟, 夏斌, 喻莉, 等. 最小二乘法在大学物理实验数据处理应用中的讨论 [J]. 大学物理实验, 2025, 38(03):67-70. DOI:10.14139/j.cnki.cn22-1228.2025.03.013.
- [8] 通义千问 (Qwen), Qwen-Max, 阿里云 (Alibaba Cloud), 2025-09-05.
- [9] DeepSeek, DeepSeek-R1-0528, 深度求索 (DeepSeek), 2025-09-06.
- [10] Kimi 大模型, Kimi-K1.5, 月之暗面 (Moonshot AI), 2025-09-05.

附件 A

```
import numpy as np import pandas as pd import matplotlib.pyplot as plt from scipy.signal import savgol_filter, find_peaks from scipy.optimize import minimize_scalar import os, matplotlib

实验参数 theta_i = np.radians(10) matplotlib.rcParams['font.sans-serif'] = ['ArialUnicodeMS', 'SimHei']
False_output_dir = '2/ / 1' os.makedirs(output_dir, exist_ok = True)

菲涅尔反射率 def fresnel_reflectance(n, theta_i): theta_t = np.arcsin(np.sin(theta_i)/n) r_s =
(np.cos(theta_i) - n * np.cos(theta_t)) / (np.cos(theta_i) + n * np.cos(theta_t)) r_p = (n * np.cos(theta_i) -
np.cos(theta_t)) / (n * np.cos(theta_i) + np.cos(theta_t)) return 0.5 * (r_s ** 2 + r_p ** 2)

反射率到折射率 def solve_n_from_reflectance(R_val): obj = lambda n: (fresnel_reflectance(n, theta_i) -
R_val) ** 2 return minimize_scalar(obj, bounds = (1.01, 5.0), method = 'bounded').x

数据读取和转换 data = pd.read_excel('2/ / 1.xlsx', header = None, skiprows = 1) data =
data.apply(pd.to_numeric, errors = 'coerce').dropna() wavelength = 10000 / data.iloc[:, 0].to_numpy() mrefl =
data.iloc[:, 1].to_numpy() / 100

平滑 reflectance_filtered = savgol_filter(reflectance, 11, 3)

折射率拟合 n_measured = np.array([solve_n_from_reflectance(R) for R in reflectance_filtered]) valid =
(n_measured > 1) & (n_measured < 5) X = np.vstack([np.ones(np.sum(valid)), 1 / wavelength[valid] *
* 2, 1 / wavelength[valid] ** 4]).TA, B, C = np.linalg.lstsq(X, n_measured[valid], rcond = None)[0]

def n_cauchy(lam): return A + B / lam ** 2 + C / lam ** 4

拟合质量 n_fit = n_cauchy(wavelength[valid]) r2 = 1 - np.sum((n_measured[valid] - n_fit) *
* 2) / np.sum((n_measured[valid] - np.mean(n_measured[valid])) ** 2) rmse = np.sqrt(np.mean((n_measured[valid] -
n_fit) ** 2))
```

—— 厚度计算部分 —— $peaks = find_peaks(reflectance_filtered, distance = 30)$ $lambda_peaks = wavelength[peaks]$ $delta_lambda = np.abs(np.diff(lambda_peaks))$ $delta_lambda_mean = np.mean(delta_lambda)$ $np.median(lambda_peaks)$ $n_center = n_cauchy(lambda_center)$ $theta_t = np.arcsin(np.sin(theta_i)/n_center)$ $d = (lambda_center * 2) / (2 * n_center * np.cos(theta_t) * delta_lambda_mean)$

保存报告 `report = f""" SiC 10° 入射角分析报告 =====`
Cauchy 拟合参数: `A = A:.6f, B = B:.6f, C = C:.6e` 拟合质量: `R² = r2:.6f RMSE = rmse:.6e` 厚
度计算: `d = d:.4f m Δ delta_lambda_mean : .4fm center lambda_center : .4fm n_center : .4f`"""with
`utf-8" as f: f.write(report)`

导出结果表格 `pd.DataFrame(" 波长m" : wavelength, " " : n_cauchy(wavelength)).to_excel(os.path.join(output_dir, "Cauchy_fit.xlsx"))`
`pd.DataFrame(" m" : lambda_peaks, " m" : np.append(delta_lambda, np.nan)).to_excel(os.path.join(output_dir, "Delta_lambda.xlsx"))`
`False)`

绘图 1: 折射率拟合 + 反射率验证 `fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))`
`ax1.scatter(wavelength[valid], n_measured[valid], s = 20, label = ' ') ax1.plot(wavelength[valid], n_fit, 'r-',`
`') ax1.set_xlabel(' (m)'); ax1.set_ylabel(' ') ax1.set_title(f' (R² = r2 : .3f, RMSE = rmse : .4f)') ax1.legend()`
`0.3)`

`R_fit = np.array([fresnel_reflectance(n, theta_i) for n in n_fit]) ax2.scatter(wavelength[valid], reflectance[valid],`
`100, s = 20, label = ' ') ax2.plot(wavelength[valid], R_fit*100, 'r-', label = ' ') ax2.set_xlabel(' (m)'); ax2.legend()`
`0.3) plt.tight_layout() plt.savefig(os.path.join(output_dir, 'SiC_10.png'), dpi = 300)`

绘图 2: 干涉条纹标记 `plt.figure(figsize=(8, 5)) plt.plot(wavelength, reflectance_filtered * 100,`
`100, label = " ", color = "blue") plt.scatter(lambda_peaks, reflectance_filtered[peaks]*100, color =`
`"red", s = 40, label = " ") plt.xlabel(" (m)"); plt.ylabel(" (plt.title(f" (delta_lambda_mean : .4fm)")) plt`
`0.3) plt.tight_layout() plt.savefig(os.path.join(output_dir, "Interference.png"), dpi = 300) plt.show()`

—— 快速版可靠性分析 —— `n_trials = 200 d_values = []`
`for i in range(n_trials): delta_lambda_noisly = delta_lambda_mean * (1 + np.random.normal(0, 0.01))`
`lambda_center * (1 + np.random.normal(0, 0.01))`
`n_center_noisly = n_cauchy(lambda_center_noisly) theta_t_noisly = np.arcsin(np.sin(theta_i)/n_center_noisly)`
`d_n = (lambda_center_noisly * 2) / (2 * n_center_noisly * np.cos(theta_t_noisly) * delta_lambda_noisly) d_values.append(d_n)`
`plt.figure(figsize=(7, 5)) plt.hist(d_values, bins = 20, color = "skyblue", edgecolor = "black") plt.axvline(x = np.mean(d_values),`
`"red", linestyle = "--", label = f" = np.mean(d_values) : .3fm") plt.xlabel(" d(m)") plt.ylabel(" ") plt`
`0.3) plt.tight_layout() plt.savefig(os.path.join(output_dir, "MC.png"), dpi = 300) plt.show()`

—— 仅滤波后反射率曲线 —— `plt.figure(figsize=(8, 5)) plt.plot(wavelength, reflectance_filtered * 100,`
`100, label = " ", color = "blue", linewidth = 1.5) plt.xlabel(" (m)") plt.ylabel(" (`
`plt.legend() plt.grid(alpha=0.3) plt.tight_layout() plt.savefig(os.path.join(output_dir, "Reflectance.png"), dpi = 300) plt.show()`

附录 B

```

import numpy as np import pandas as pd import matplotlib.pyplot as plt from scipy.optimize
import minimize from scipy.signal import find_peaks import os, matplotlib
实验参数 theta_i = np.radians(10) 10° n0, n2 = 1.0, 3.4 / output_dir = ' 3/ / 3' os.makedirs(output_dir,
True)
matplotlib.rcParams['font.sans-serif'] = ['Arial Unicode MS', 'SimHei', 'DejaVu Sans']
matplotlib.rcParams['axes.unicode_minus'] = False
— 折射率模型 — def n_cauchy(lam, A, B, C) : return A + B/lam**2 + C/lam**4
— Airy 反射率 — def airy_reflectance(lam, d, A, B, C) : n1 = n_cauchy(lam, A, B, C) theta1 =
np.arcsin(n0 * np.sin(theta_i)/n1) r01 = (n0 * np.cos(theta_i) - n1 * np.cos(theta1))/(n0 *
np.cos(theta_i) + n1 * np.cos(theta1)) r12 = (n1 * np.cos(theta1) - n2 * np.cos(theta1))/(n1 *
np.cos(theta1) + n2 * np.cos(theta1)) delta = (2*np.pi/lam)*n1*d*np.cos(theta1) return (r01*
**2 + r12**2 + 2 * np.abs(r01 * r12) * np.cos(2 * delta))/(1 + (r01 * r12) **2 + 2 * np.abs(r01 *
r12) * np.cos(2 * delta))
— 目标函数 — def objective(params, lam, R_exp) : d, A, B, C = params return np.mean((airy_reflecta
R_exp)**2)
— 数据读取 — df = pd.read_excel(" 3/ / 3.xlsx", header = None, skiprows = 1) wavelength =
1e4/df.iloc[:, 0].astype(float).to_numpy() R_exp = df.iloc[:, 1].astype(float).to_numpy()/100
— 数据平滑 — from scipy.signal import savgol_filter reflectance_filtered = savgol_filter(R_exp, 11, 3)
11 = 3
— 优化求解 — init_params = [3.0, 2.6, 0.01, 0.0] res = minimize(objective, init_params, args =
(wavelength, R_exp), method = "Nelder - Mead") d_fit, A_fit, B_fit, C_fit = res.x
— 拟合结果 — R_fit = airy_reflectance(wavelength, d_fit, A_fit, B_fit, C_fit) r2 = 1 - np.sum((R_exp -
R_fit)**2)/np.sum((R_exp - np.mean(R_exp))**2) rmse = np.sqrt(np.mean((R_fit - R_exp)**2))
— 图 1: 拟合对比 (原始 vs 拟合) — plt.figure(figsize=(8,5)) plt.plot(wavelength, R_exp, 'b-', alpha =
0.5, label = " ( )") plt.plot(wavelength, reflectance_filtered, 'g-', linewidth = 1, label =
" ") plt.plot(wavelength, R_fit, 'r-', label = f" (d = d_fit : .3 fm)") plt.xlabel(" (m)"); plt.ylabel(" ") p
0.3) plt.savefig(os.path.join(output_dir, " 3 .png"), dpi = 300) plt.show()
— 图 2: 干涉条纹 (基于滤波) — peaks, _ = find_peaks(reflectance_filtered, distance =
30) lambda_peaks = wavelength[peaks] delta_lambda = np.abs(np.diff(lambda_peaks)) delta_lambda_mean =
np.mean(delta_lambda) lambda_center = np.median(lambda_peaks)
plt.figure(figsize=(8,5)) plt.plot(wavelength, reflectance_filtered*100, label = " ", color =
"blue") plt.scatter(lambda_peaks, reflectance_filtered[peaks]*100, color = "red", s = 40, label =
" ") for i, lam in enumerate(lambda_peaks[: 10]) : plt.text(lam, reflectance_filtered[peaks[i]] *
100+1, f"lam : .2f", fontsize = 8, rotation = 45) plt.xlabel(" (m)"); plt.ylabel(" (plt.title(f" ( delta
0.3) plt.savefig(os.path.join(output_dir, " 3 .png"), dpi = 300) plt.show()
— 导出条纹数据 (基于滤波) — pd.DataFrame(" 条纹波长 m" : lambda_peaks, " m" :

```

```

np.append(delta_lambda, np.nan)).to_excel(os.path.join(output_dir, "3_3.xlsx"), index = False)
----- ( ) ----- n_trials = 200 d_values = []
for i in range(n_trials):
    noise = np.random.normal(0, 0.005, size = R_exp.shape) R_noisy =
np.clip(R_exp + noise, 0, 1)
再拟合厚度(只调整厚度, 固定 A, B, C, 加速计算) obj = lambda d: np.mean((airy_reflectance(wavelength
R_noisy)**2) d_noisy = minimize(obj, d_fit, method = "Nelder-Mead").x[0] d_values.append(d_noisy)
绘制直方图 plt.figure(figsize=(7,5)) plt.hist(d_values, bins = 20, color = "skyblue", edgecolor =
"black") plt.axvline(np.mean(d_values), color = "red", linestyle = "--", label = f" =
np.mean(d_values) : 3 fm") plt.xlabel(" d(m)") plt.ylabel(" ") plt.title(" ( )") plt.legend() plt.grid(alpha
0.3) plt.tight_layout() plt.savefig(os.path.join(output_dir, "3_MC.png"), dpi = 300) plt.show()
from scipy.signal import savgol_filter
—— 仅滤波后反射率曲线 —— reflectance_filtered = savgol_filter(R_exp, 11, 3) =
11 = 3
plt.figure(figsize=(8, 5)) plt.plot(wavelength, reflectance_filtered*100, label = " ", color =
"blue", linewidth = 1.5) plt.xlabel(" (m)") plt.ylabel(" (plt.title(" ") plt.legend() plt.grid(alpha =
0.3) plt.tight_layout() plt.savefig(os.path.join(output_dir, "3_3.png"), dpi = 300) plt.show()

```

附录 C

```

import numpy as np import pandas as pd import matplotlib.pyplot as plt from scipy.optimize
import minimize from scipy.signal import find_peaks import os, matplotlib
实验参数 theta_i = np.radians(10) n0, n2 = 1.0, 3.4 / output_dir = '3/' / '3' os.makedirs(output_dir,
True)
matplotlib.rcParams['font.sans-serif'] = ['Arial Unicode MS', 'SimHei', 'DejaVu Sans']
matplotlib.rcParams['axes.unicode_minus'] = False
—— 折射率模型 —— def n_cauchy(lam, A, B, C): return A + B/lam**2 + C/lam**4
—— Airy 反射率 —— def airy_reflectance(lam, d, A, B, C): n1 = n_cauchy(lam, A, B, C) theta1 =
np.arcsin(n0 * np.sin(theta_i)/n1) r01 = (n0 * np.cos(theta_i) - n1 * np.cos(theta1))/(n0 *
np.cos(theta_i) + n1 * np.cos(theta1)) r12 = (n1 * np.cos(theta1) - n2 * np.cos(theta1))/(n1 *
np.cos(theta1) + n2 * np.cos(theta1)) delta = (2*np.pi/lam)*n1*d*np.cos(theta1) return (r01*
*2 + r12**2 + 2*np.abs(r01*r12)*np.cos(2*delta))/(1 + (r01*r12)**2 + 2*np.abs(r01*
r12)*np.cos(2*delta))
—— 目标函数 —— def objective(params, lam, R_exp): d, A, B, C = params return np.mean((airy_reflecta
R_exp)**2)
—— 数据读取 —— df = pd.read_excel("3_3.xlsx", header = None, skiprows = 1) wavelength =
1e4/df.iloc[:, 0].astype(float).to_numpy() R_exp = df.iloc[:, 1].astype(float).to_numpy()/100

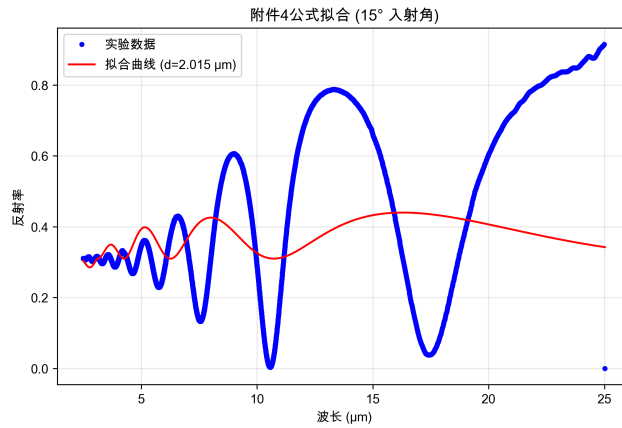
```

```

— 数据平滑 — from scipy.signal import savgol_filter reflectance_filtered = savgol_filter(R_exp, 11, 3)
11    = 3
— 优化求解 — init_params = [3.0, 2.6, 0.01, 0.0] res = minimize(objective, init_params, args =
(wavelength, R_exp), method = "Nelder - Mead") d_fit, A_fit, B_fit, C_fit = res.x
— 拟合结果 — R_fit = airy_reflectance(wavelength, d_fit, A_fit, B_fit, C_fit) r2 = 1 - np.sum((R_exp -
R_fit)**2) / np.sum((R_exp - np.mean(R_exp))**2) rmse = np.sqrt(np.mean((R_fit - R_exp)**2))
— 图 1: 拟合对比 (原始 vs 拟合) — plt.figure(figsize=(8,5)) plt.plot(wavelength, R_exp, 'b-', alpha =
0.5, label = "    ( )") plt.plot(wavelength, reflectance_filtered, 'g-', linewidth = 1, label =
"    ") plt.plot(wavelength, R_fit, 'r-', label = f"    (d = d_fit : .3 fm)") plt.xlabel("    (m)"); plt.ylabel("    ")
0.3) plt.savefig(os.path.join(output_dir, " 3    .png"), dpi = 300) plt.show()
— 图 2: 干涉条纹 (基于滤波) — peaks, = find_peaks(reflectance_filtered, distance =
30) lambda_peaks = wavelength[peaks] delta_lambda = np.abs(np.diff(lambda_peaks)) delta_lambda_mean =
np.mean(delta_lambda) lambda_center = np.median(lambda_peaks)
plt.figure(figsize=(8,5)) plt.plot(wavelength, reflectance_filtered*100, label = "    ", color =
"blue") plt.scatter(lambda_peaks, reflectance_filtered[peaks]*100, color = "red", s = 40, label =
"    ") for i, lam in enumerate(lambda_peaks[: 10]): plt.text(lam, reflectance_filtered[peaks[i]] *
100 + 1, f"lam : .2f", fontsize = 8, rotation = 45) plt.xlabel("    (m)"); plt.ylabel("    (plt.title(f"    ( delta
0.3) plt.savefig(os.path.join(output_dir, " 3    .png"), dpi = 300) plt.show()
— 导出条纹数据 (基于滤波) — pd.DataFrame(" 条纹波长 m" : lambda_peaks, "    m" :
np.append(delta_lambda, np.nan)).to_excel(os.path.join(output_dir, " 3    .xlsx"), index = False) -
- - - - - (    ) - - - - - n_trials = 200 d_values = []
for i in range(n_trials):    ±0.5 noise = np.random.normal(0, 0.005, size = R_exp.shape) R_noisy =
np.clip(R_exp + noise, 0, 1)
再拟合厚度 (只调整厚度, 固定 A,B,C, 加速计算) obj = lambda d: np.mean((airy_reflectance(wavelength,
R_noisy)**2)) d_noisy = minimize(obj, d_fit, method = "Nelder - Mead").x[0] d_values.append(d_noisy)
绘制直方图 plt.figure(figsize=(7,5)) plt.hist(d_values, bins = 20, color = "skyblue", edgecolor =
"black") plt.axvline(np.mean(d_values), color = "red", linestyle = "--", label = f"    =
np.mean(d_values) : .3 fm") plt.xlabel("    d(m)") plt.ylabel("    ") plt.title("    (    )") plt.legend() plt.grid(alpha =
0.3) plt.tight_layout() plt.savefig(os.path.join(output_dir, " 3    MC.png"), dpi = 300) plt.show()
from scipy.signal import savgol_filter
—— 仅滤波后反射率曲线 —— reflectance_filtered = savgol_filter(R_exp, 11, 3)    =
11    = 3
plt.figure(figsize=(8, 5)) plt.plot(wavelength, reflectance_filtered*100, label = "    ", color =
"blue", linewidth = 1.5) plt.xlabel("    (m)") plt.ylabel("    (plt.title("    ") plt.legend() plt.grid(alpha =
0.3) plt.tight_layout() plt.savefig(os.path.join(output_dir, " 3    .png"), dpi = 300) plt.show()

```

附录 D



AI 工具使用详情

8.1 一. 所用 AI 工具名称和版本

(1) DeepSeek, DeepSeek-R1-0528 (2) Kimi 大模型, Kimi-K1.5 (3) 文心一言 (ERNIE Bot) , ERNIE-4.0-Plus

8.2 二. 具体使用目的和环节

8.2.1 1. 代码编写与优化

在实验数据处理与建模过程中，团队主要依靠自主推导的理论与算法框架。为提高编程效率与代码规范性，曾使用 DeepSeek (DeepSeek-R1-0528) 协助实现 Python 脚本的书写与优化。具体包括：光谱数据读取、信号滤波、Cauchy 模型拟合、Airy 反射率计算、厚度估算等功能模块。工具的作用主要是提供参考性的代码框架和优化思路，核心算法逻辑和参数设置均由团队成员完成。

8.2.2 2. 公式整理与呈现

在论文写作过程中，我们使用 Kimi 大模型 (Kimi-K1.5) 将推导得到的关键公式转写为 LaTeX，以保证公式在文稿中表达清晰、排版规范。Kimi 同时提醒我们在厚度条件和反射率条件的写法上保持一致，我们采纳并做了进一步整理。

8.2.3 3. 论文撰写辅助

在文字表述与逻辑组织方面，我们参考了文心一言 (ERNIE Bot, ERNIE-4.0-Plus) 的润色建议。例如，在蒙特卡洛模拟部分，AI 提供了更适合学术论文的表述方式，使逻辑更清晰；在方法比较部分，AI 给了一些图表组织的思路，供我们借鉴。但是，实验数据的处理

和结论分析均由团队成员独立完成。

8.2.4 4. 支撑材料整理

在整理《AI 工具使用详情》说明材料时，我们也参考了三个工具的输出，以帮助文件结构更清晰、表述更顺畅。这一部分主要涉及排版和文字整理，不涉及核心研究内容

8.3 三. 关键交互记录

提示词 1：帮我把绘图部分的代码写得更简洁一些，同时加上合适的图例和标签。回复 1：AI 给出了简化 matplotlib 调用的建议，并提供了自动标注峰值的示例代码。我们在修改时保留了主要逻辑，只调整了部分绘图细节。

提示词 2：将厚度计算公式和多光束干涉条件转换为 LaTeX 表达式。回复 2：AI 工具输出了标准化的 LaTeX 公式格式，并提醒统一厚度条件和反射率条件的写法，便于论文中引用。我们在此基础上整理并完善了公式部分。

提示词 3：请帮我把蒙特卡洛模拟的文字描述改写得更学术一些。回复 3：AI 工具提供了更符合论文风格的表述方式，例如强调迭代次数、噪声设置和分布结果的物理意义。我们采纳了语言风格上的修改，但核心实验和模拟仍由我们独立完成。

提示词 4：请帮我检查一下厚度计算的公式部分有没有常见的写法改进。回复 4：AI 工具提醒厚度公式常见写法为 $d=2n\cos \Delta$ ，与我们推导一致。我们采纳了这一表述作为论文中的标准形式。

提示词 5：代码里面的变量名太随意了，有没有更专业的风格？回复 5：AI 工具建议将 `lam` 改为 `wavelength`，`Rexp` `reflectanceexp`

8.4 四. 采纳与人工修改情况

采纳情况：团队在整体思路设计、模型建立和公式推导上均由成员独立完成，仅在具体实现环节参考了 AI 生成的部分 Python 代码框架和排版示例。部分脚本结构、方法比较图的绘制方式，以及厚度条件与反射率条件的 LaTeX 表达式，作为辅助素材被采纳，用于提高实现效率和表达规范性。

人工修改情况：AI 输出的内容大多经过人工修改与验证。例如，在代码部分我们调整了 `findpeaks` `AI` `n`