

基于 PSO 和 DBSCAN 算法的红外干涉优化厚度测定

摘要

在半导体材料研究领域，碳化硅作为极具潜力的第三代半导体材料，涉及材料科学、半导体物理、电力电子等多个学科领域。其外延层厚度也因为直接关系到器件的性能表现而成为科学家们津津乐道的话题。本论文聚焦于利用红外干涉法对碳化硅外延层厚度进行测定，解决了“一般情况下对外延层厚度测量的数学模型”、“考虑红外光谱对折射率影响下外延层厚度的精确估算算法”、“考虑多光束干涉下的外延层厚度求值算法”等问题。

问题一，我们主要运用了光学的干涉理论进行模型推导。当只考虑一次反射透射时，在“空气-外延层”界面发生折射的红外光因为比在同一界面发生反射的光多走了一段几何光程而产生相位差，相位差在某一定值就会在干涉光谱中观察到明条纹，不同层级的明条纹都有与外延层厚度和其层级数对应的物理公式，通过相邻明条纹即可推导出外延层厚度数学模型，即 $d = \frac{1}{2 \cdot \tilde{n} \cdot \Delta\tilde{\nu} \cdot \cos \theta_2}$ (\tilde{n} 为样品折射率， $\Delta\tilde{\nu}$ 为相邻波数差， θ_2 为折射角)。随后我们又分两种情况，分别结合 Drude-Lorentz 模型和 Sellmeier 模型研究了一般情况下对样品折射率 \tilde{n} 的精确测定。

问题二，我们建立了“反射光谱处理——厚度公式改良——复折射率反推——统计优化结合聚类算法”这一由模型执行的算法链。首先通过光学方面的相关理论证明反射光谱极大(或极小)值处即为明(暗)条纹分布；然后根据题目条件改良了问题一推导得出的厚度公式；之后利用反射光谱的某些性质，结合菲涅尔公式、斯涅尔定律和 Fabry-Perot 公式将“空气-外延层”、“外延层-衬底”两界面反射率和不同介质的折射率联系起来，反推得出对应某种波数的外延层复折射率；最后将其代入上述改良的厚度公式中求解得到一系列点簇，通过 DBSCAN 聚类算法得到厚度的精确区间，并且在“模型稳定性分析”这一部分通过蒙特卡洛扰动实验分析了该模型的可靠性。代入附件一二表格的数据，最终结果为 $d = 8.9282 \pm 0.9676 \mu m$ ，且该模型稳定性由置信度反映，结果为 89%。

问题三，我们经过查阅相关文献，得出多光束干涉的必要条件为外延层杂质浓度较低、光源相干长度大于理论光程差、界面反射率较高三个方面并做适当推导，随后根据题设环境选择精细度作为审视干涉现象的唯一标准，推导其迭代公式并代入四个附件中，结合各个附件的结果讨论是否需要考虑多光束干涉。然后我们建立了在多光束干涉下的厚度求解模型，主要依靠 Fabry-Perot 公式和 Lorentz 振子模型，通过粒子群算法不断迭代使目标函数最小化，从而得到多光束干涉下的真实厚度。最终引入 DBSCAN 聚类算法计算真实厚度样本的误差，输出带有一定误差的外延层厚度结果。并且在“模型稳定性分析”这一部分分析了粒子群算法在通过不断调整参数拟合曲线方面的可靠性。最终计算得硅外延层厚度为 $3.7042 \pm 0.4619 \mu m$ ，碳化硅外延层修正厚度为 $7.8426 \pm 0.5549 \mu m$ 。

关键词： 双(多)光束干涉 精细度迭代 DBSCAN PSO 蒙特卡洛扰动

1 问题重述

1.1 背景

碳化硅作为第三代半导体材料，其外延层厚度是影响器件性能的关键参数。红外干涉法通过“空气-外延层”、“外延层-衬底”两个界面折射、反射现象的差异产生的干涉条纹，可无损伤测量厚度。外延层折射率与掺杂载流子浓度、红外波长等相关，需建立数学模型并设计算法实现厚度计算。

1.2 题目要求

1.2.1 问题一

考虑当外延层和衬底界面仅发生一次反射、透射，产生干涉条纹，建立确定外延层厚度的数学模型。

1.2.2 问题二

根据问题 1 的数学模型，设计计算厚度的算法。利用附件的碳化硅晶圆片反射率光谱数据（波数 - 反射率曲线），给出厚度计算结果，并分析结果的可靠性。

1.2.3 问题三

考虑光波在外延层与衬底界面可产生多次反射透射，形成多光束干涉。完成下列要求：

- 1、推导多光束干涉的必要条件，分析其对外延层厚度计算精度的影响。
- 2、根据多光束干涉条件，判断附件 3 和附件 4 是否存在多光束干涉，建立硅外延层厚度的数学模型与算法，给出计算结果。
- 3、若认为附件 1 和附件 2 的碳化硅光谱数据存在多光束干涉，需消除其对厚度计算的影响，并给出修正后的结果。

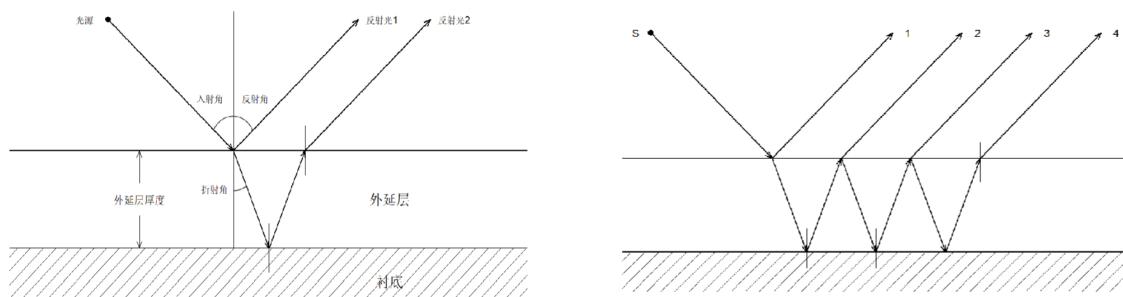


图 1：左：双光束干涉的示意图 右：多光束干涉的示意图

2 问题分析

2.1 问题一

问题一中，我们主要运用了光学的干涉理论进行模型推导。当只考虑一次反射透射时，在“空气-外延层”界面发生折射的红外光因为比在同一界面发生反射的光多走了一段几何光程而产生相位差，相位差在某一定值就会在干涉光谱中观察到明条纹，不同层级的明条纹都有与外延层厚度和其层级数对应的物理公式，通过相邻明条纹即可推导出外延层厚度数学模型。随后我们又分“实验条件测得参数具全”及“实验条件有限”两种情况，分别结合 Drude-Lorentz 模型和 Sellmeier 模型研究了一般情况下对样品折射率 \tilde{n} 的精确测定。

2.2 问题二

问题二中，我们考虑到附件提供的数据类型有且仅有“波数”和“反射率”两种，因此建立了“反射光谱处理——厚度公式改良——复折射率反推——统计优化结合聚类算法”这一算法链。首先通过光学方面的相关理论证明反射光谱极大（或极小）值处即为明（暗）条纹分布；然后根据题目条件改良了问题一推导得出的厚度公式；之后利用反射光谱的某些性质，结合菲涅尔公式、斯涅尔定律和 Fabry-Perot 公式将“空气-外延层”、“外延层-衬底”两界面反射率和不同介质的折射率联系起来，反推得出对应某种波数的外延层复折射率；最后将其代入上述改良的厚度公式中求解得到一系列点簇，通过 DBSCAN 聚类算法得到厚度的精确区间，并且在“模型稳定性分析”这一部分利用蒙特卡洛扰动实验分析了聚类算法的可靠性和误差来源。

2.3 问题三

问题三中，我们先要考虑到多光束干涉的必要条件并能给出适当的推导，随后根据题设环境选择精细度作为审视干涉现象的唯一标准，推导其迭代公式并代入四个附件中，结合各个附件的结果讨论是否需要考虑多光束干涉。然后我们建立了在多光束干涉下的厚度求解模型，主要依靠 Fabry-Perot 公式和 Lorentz 振子模型，通过粒子群算法不断迭代使目标函数最小化，从而得到多光束干涉下不同样本的真实厚度。并且在“模型稳定性分析”这一部分分析了 PSO 算法在通过不断调整参数拟合曲线方面的可靠性。

3 模型假设

基于红外干涉法测量外延层厚度的物理原理和实际实验条件，本论文建立数学模型时采用以下假设：

- 1、外延层材料为各向同性介质，折射率不依赖于光的偏振方向。
- 2、外延层厚度均匀，折射率在层内空间分布均匀。
- 3、“空气-外延层”以及“外延层-衬底”界面处不发生漫反射，只发生镜面反射和透射。
- 4、在测量波长范围内，外延层对红外光的吸收可忽略不计。
- 5、当精细度 $\mathcal{F} \leq 10\%$ 时，可忽略多光束干涉效应。
- 6、实验过程中样品温度保持恒定，无热膨胀效应。

4 符号和单位约定

为便于数学表达和计算，本论文采用以下符号和单位约定：

表 1: 物理量符号约定

符号	含义	符号	含义
d	外延层厚度 (m)	θ_1	入射角 (度)
λ	真空中的波长 (m)	θ_2	折射角 (度)
$\tilde{\nu}$	波数 (cm^{-1})	R_1	空气-外延层界面反射率
ω	角频率 (rad/s)	R_2	外延层-衬底界面反射率
n	折射率	$R_{\text{总}}$	总反射率
$\Delta\phi$	相位差 (rad)	m	干涉级次
δ	光程差引起的相位差 (rad)	\mathcal{F}	精细度
ΔL	几何光程差 (m)	V	干涉对比度
$\Delta\tilde{\nu}$	相邻干涉极值波数差 (cm^{-1})	n	反射次数
A	损耗率	$F(n)$	第 n 次反射的精细度
$A_{\text{eff},n}$	第 n 次反射的有效损耗率		

- 长度：厚度 d 以微米 (m) 为单位，波长 λ 以微米 (m) 为单位
- 波数： $\tilde{\nu}$ 以厘米 $^{-1}$ (cm^{-1}) 为单位
- 角度：以度 ($^{\circ}$) 为单位，必要时转换为弧度 (rad)
- 频率：角频率 ω 以弧度每秒 (rad/s) 为单位
- 反射率：以小数形式表示，范围 [0, 1]

5 模型建立及求解

5.1 问题一

问题一中，我们按下列步骤完成对应数学模型的建立：

Step1: 运用光学的干涉理论，描述当仅有一次反射、透射所产生的干涉条纹的物理性质，由此推导出厚度计算方式。

Step2: 根据实际实验条件，分别结合 Drude-Lorentz 模型和 Sellmeier 模型推导不同实验环境条件下折射率 \tilde{n} 的精确求值。

Step3: 根据 Step1 求解得到的外延层厚度公式，建立符合实际场景下用于检验干涉现象的数学模型。

5.1.1 厚度公式推导

根据双光束干涉的光程差公式和波数关系，厚度公式的推导过程如下：

红外光在“空气-外延层上表面”和“外延层-衬底界面”分别产生反射光（记为光 1、光 2）。其中，光 2 需穿透外延层往返传播。

其几何光程为：

$$L_{\text{几何}} = 2 \cdot n(\tilde{\nu}) \cdot d \cdot \cos \theta_2$$

其中： $n(\tilde{\nu})$ ：外延层复折射率（与波数 $\tilde{\nu}$ 相关）； d ：外延层厚度（待求量）； $\cos \theta_2$ ：光在层内折射角 θ_2 的余弦（折射角由斯涅尔定律 $\sin \theta_1 = \tilde{n}(\tilde{\nu}) \cdot \sin \theta_2$ 与入射角 θ_1 关联）。

半波损失的等效光程差

光从光疏介质（空气，折射率 $n \approx 1$ ）入射到光密介质（外延层， $\tilde{n}(\tilde{\nu}) > 1$ ）时，上表面的反射光（光 1）会发生半波损失：记等效光程差额外增加 Δ 。

光 2 是从外延层到衬底的反射，若衬底与外延层折射率差异小，其半波损失可忽略。因此，两束光的半波损失差异等效为总光程差增加 Δ 。

干涉极值的相位条件

干涉极大（反射率峰值）或极小（反射率谷值）的条件由总相位差决定。相位差与光程差的关系为：

$$\Delta\phi = \frac{2\pi}{\lambda} \cdot (L_{\text{几何}} + \Delta)$$

当两束光相位差为 $2k\pi$ (k 为整数) 时，发生相长干涉（反射率极大）；当相位差为 $(2k+1)\pi$ 时，发生相消干涉（反射率极小）。

波数差与光程差的关联

波数 $\tilde{\nu}$ (单位: cm^{-1}) 与波长 λ (单位: cm) 满足关系：

$$\tilde{\nu} = \frac{1}{\lambda}$$

对于相邻的干涉极值（如第 m 个和第 $m+1$ 个反射率峰值），对应的波数为 $\tilde{\nu}_m$ 和 $\tilde{\nu}_{m+1}$ ，波数差为：

$$\Delta\tilde{\nu} = \tilde{\nu}_{m+1} - \tilde{\nu}_m$$

根据干涉现象，有：

$$2n \cdot d \cdot \cos \theta_2 + \Delta = m \cdot \frac{1}{\tilde{\nu}_m} \quad (1)$$

$$2n \cdot d \cdot \cos \theta_2 + \Delta = (m+1) \cdot \frac{1}{\tilde{\nu}_{m+1}} \quad (2)$$

两式相减，整理得：

$$d = \frac{1}{2 \cdot n \cdot \Delta\tilde{\nu} \cdot \cos \theta_2} \quad (3)$$

式 (3) 即为求解外延层厚度的数学模型。

5.1.2 两种实验条件下对折射率的测定

针对已经推导出的厚度数学模型，我们还需测得参数折射率 n 的具体数值才可计算外延层厚度。

根据实际实验环境的条件，当实验仪器测得参数俱具全时，诸如高频介电常数 ϵ_∞ 、等离子体频率 ω_p 、阻尼系数 γ_j 等参数均能测得，可考虑引入 Drude-Lorentz 模型求解折射率 n ；当实验条件有限，仅能测得振子强度 B_j 和共振波长 λ_j 时，考虑引入 Sellmeier 模型求解折射率 n 。

Drude-Lorentz 模型 该模型由公式描述如下:

$$n(\omega) = \sqrt{\varepsilon_{\infty} - \frac{\omega_p^2}{\omega^2 + i\gamma\omega} + \sum_j \frac{\omega_{p,j}^2}{\omega_j^2 - \omega^2 - i\gamma_j\omega}} \quad (4)$$

实验所需测得参数有: 高频介电常数 ε_{∞} 、自由载流子等离子体频率 ω_p 、自由载流子碰撞频率 γ 、振子的等离子体频率 $\omega_{p,j}$ 等。

最后得到函数关系 $n = n(\omega) = n(2\pi\nu)$ 。

Sellmeier 模型 该模型由公式描述如下:

$$n^2(\lambda) = 1 + \sum_{j=1}^m \frac{B_j \lambda^2}{\lambda^2 - \lambda_j^2} \quad (5)$$

实验仅需测得振子强度 B_j 和共振波长 λ_j 。

最后得到函数关系 $n = n(\lambda) = n(\frac{1}{\nu})$ 。

5.1.3 问题一数学模型的建立

实际实验场景中, 红外光干涉法操作可视化如下图图 2:

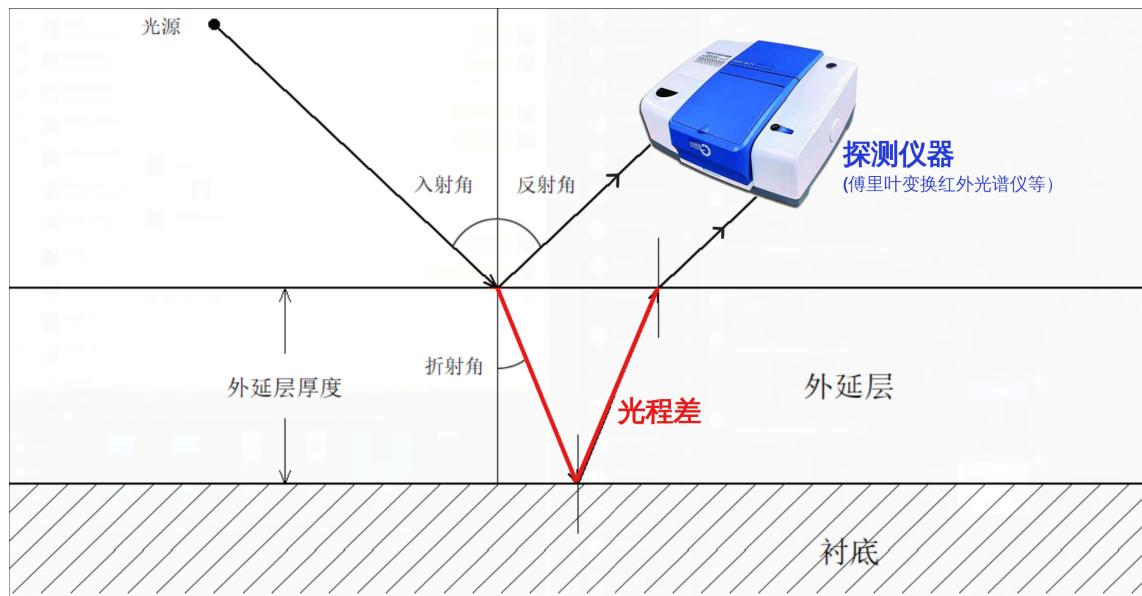


图 2: 红外光干涉法的仪器摆放

得到的干涉图像如下图图 3:

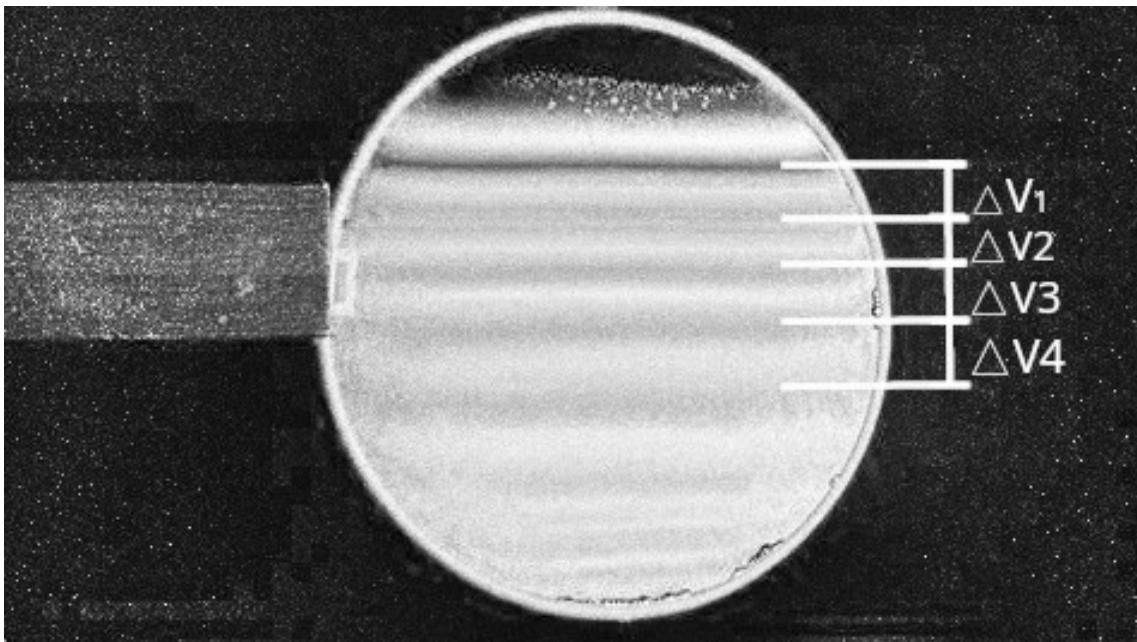


图 3: 红外光干涉法的仪器摆放

最后根据实验条件选择以下两种模型:

表 2: 不同实验条件下的模型选择

实验条件	可通过仪器测得具全参数	实验条件有限
模型建议	Drude-Lorentz 模型	Sellmeier 模型
公式描述	$n(\omega) = \sqrt{\varepsilon_\infty - \frac{\omega_p^2}{\omega^2 + i\gamma\omega} + \sum_j \frac{\omega_{p,j}^2}{\omega_j^2 - \omega^2 - i\gamma_j\omega}}$	$n^2(\lambda) = 1 + \sum_{j=1}^m \frac{B_j \lambda^2}{\lambda^2 - \lambda_j^2}$
所需测量参数	高频介电常数 ε_∞ 自由载流子等离子体频率 ω_p 自由载流子碰撞频率 γ 振子的等离子体频率 $\omega_{p,j}$ 等。	振子强度 B_j 共振波长 λ_j

在已知折射率的条件下，综合运用所求的厚度公式 $d = \frac{1}{2 \cdot \bar{n} \cdot \Delta \tilde{\nu} \cdot \cos \theta_2}$ 即可求得外延层厚度。

5.2 问题二

问题二中，我们按下列步骤完成对应的算法以及建模：

Step1: 证明根据附件数据拟合出来的反射率光谱极值处为明暗条纹。

Step2: 结合问题一推导出的厚度公式，在题设中折射率只受红外光谱的波数 $\tilde{\nu}$ 变化的条件下求解外延层厚度。

Step3: 根据菲涅尔公式和斯涅尔定律推导复折射率 $n(\tilde{\nu})$ 。

Step4: 将折射率 $n(\tilde{\nu})$ 、相邻极大波数差 $\Delta\tilde{\nu}$ 等变量代入厚度公式，通过统计优化 + 聚类算法 DBSCAN 得到较为精确的厚度区间。

5.2.1 证明反射率光谱极值处为明暗条纹

当红外光在薄膜（外延层）上下界面反射产生干涉时，反射率光谱会出现周期性的极值（峰值或谷值），通过实验即可得出反射率-波数关系图象（即反射率光谱）。从图像中可获得相邻极大值（或极小值）的对应波数 $\tilde{\nu}_m, \tilde{\nu}_{m+1}$ 以及两者波数差 $\Delta\tilde{\nu}_m$ ，其中反射率极大值可视为明条纹而极小值可视为暗条纹，推导如下：

两束相干光的复振幅分别为 E_1 和 E_2 ，其总光强 I 满足：

$$I = |E_1 + E_2|^2 = |E_1|^2 + |E_2|^2 + 2|E_1 E_2| \quad (6)$$

反射率 R 定义为反射光强与入射光强 I_0 的比值，单束反射光的反射率为：

$$R_1 = \frac{|E_{01}|^2}{I_0}, \quad R_2 = \frac{|E_{02}|^2}{I_0}$$

代入光强叠加公式，总反射率为：

$$R = \frac{|E_1 + E_2|^2}{I_0} \quad (7)$$

那么根据 (1) (2)，我们有：

$$I = I_0 \cdot R \quad (8)$$

显然当反射率出现极大值时，此时干涉光光强达到极大，即为明条纹，反之则为暗条纹，下文我们统一取明条纹（即反射光谱极大值）为研究对象。

5.2.2 厚度公式情境化运用

根据题目背景，对于同一块测试样本（即同一碳化硅外延层晶体），可近似认为载流子浓度等因素为影响折射率变化无关变量，那么在建模中红外光谱波数即为折射率的唯一影响因素^[1]，即 $n = n(\tilde{\nu})$ 。

取 (3) 式中折射率 $\tilde{n} = \tilde{n}(\tilde{\nu})$ 可得最终厚度公式：

$$d(\tilde{\nu}) = \frac{1}{2 \cdot n(\tilde{\nu}) \cdot \Delta\tilde{\nu} \cdot \cos \theta_2} \quad (9)$$

其中：

$n(\tilde{\nu})$: 外延层折射率（与波数 $\tilde{\nu}$ 相关）； θ_2 : 外延层内折射角（由斯涅尔定律 $\sin \theta_1 = \tilde{n}(\tilde{\nu}) \cdot \sin \theta_2$ 与入射角 θ_1 关联）； $\Delta\tilde{\nu}$: 相邻干涉极值的波数差； λ : 光在真空中的波长。

5.2.3 复折射率的推导

为得到 $\tilde{n}(\tilde{\nu})$ ，我们利用反射率光谱结合非偏振下的菲涅尔公式以及 Fabry-Perot 公式反推不同波数下对应的复折射率。

根据双光束干涉原理（仅考虑一次反射透射），反射率光谱极大值处（明条纹处）反射率 $R(\tilde{\nu})$ 推导如下：

在空气-外延层界面 ($n_{\text{空气}} = 1$, 外延层折射率 $n(\tilde{\nu})$, 入射角 θ)

斯涅尔定律: $\sin \theta = n_1 \sin \theta_1$, 故

$$\cos \theta_1 = \sqrt{1 - \left(\frac{\sin \theta}{n(\tilde{\nu})} \right)^2}$$

代入菲涅尔公式, 非偏振反射率 R_1 为:

$$R_1 = \frac{1}{2} \left[\left(\frac{\cos \theta - n(\tilde{\nu}) \cos \theta_1}{\cos \theta + n(\tilde{\nu}) \cos \theta_1} \right)^2 + \left(\frac{n(\tilde{\nu}) \cos \theta - \cos \theta_1}{n(\tilde{\nu}) \cos \theta + \cos \theta_1} \right)^2 \right] \quad (10)$$

在外延层-衬底界面, 经查阅文献可得碳化硅外延层 - 衬底界面反射率通常在 0.1%-1% 范围内, 可近似取为 $R_2 = 0.5\%$ ^[2]

将上述 R_1 、 R_2 代入 Fabry-Perot 公式, 得 R 的表达式

$$R(\tilde{\nu}) = \left(\frac{\sqrt{R_1} + \sqrt{R_2}}{1 + \sqrt{R_1 R_2}} \right)^2 \quad (11)$$

即可得用折射率和入射角表示的 $R(\tilde{\nu})$ 。

代入数据可得 $n(\tilde{\nu})$ 。

5.2.4 统计优化 + 聚类算法得到厚度区间

在所给的反射率光谱中, 取两两极大值进行计算, 其中 $\tilde{n}(\tilde{\nu})$ 取由前一极大波数计算得出的折射率, 可得

$$\begin{cases} d(n) = d(\Delta\nu_n, \tilde{n}(\tilde{\nu}_n)) \\ \Delta\nu_n = \nu_{n+1} - \nu_n \end{cases} \quad (12)$$

得到总共含有 n 个离散的厚度点簇, 分别为 $d_1, d_2 \dots d_n$ 。

之后通过聚类算法 DBSCAN^[3] 识别数据点的密集簇, 簇内即集中区域。主要步骤如下:

1、设置参数, 并将厚度点集合转换为一维向量。

2、聚类拓展: 以核心点为起点, 将其邻域内的点加入聚类, 如果邻域内有新的核心点, 继续扩展, 最终形成一个密度相连的聚类。

3、标记噪音点: 没有被任何核心点及其领域半径覆盖的点记为噪音, 在当前模型中, 噪音即为那些厚度数值异常的样本点。

4、输出最大聚类: 即厚度的范围。

最后给出外延层厚度的范围。

5.2.5 问题二数学模型的建立

整理并读入附件 1 和附件 2 数据, 处理异常样本点后, 绘出反射率光谱并标记极大值点。

按前文所述算法链, 最终得到厚度散点图, 经过聚类算法分析后可将结果可视化 (图 6、7)。

SiC Epitaxial Layer Thickness Analysis - 附件1 (入射角 10°)

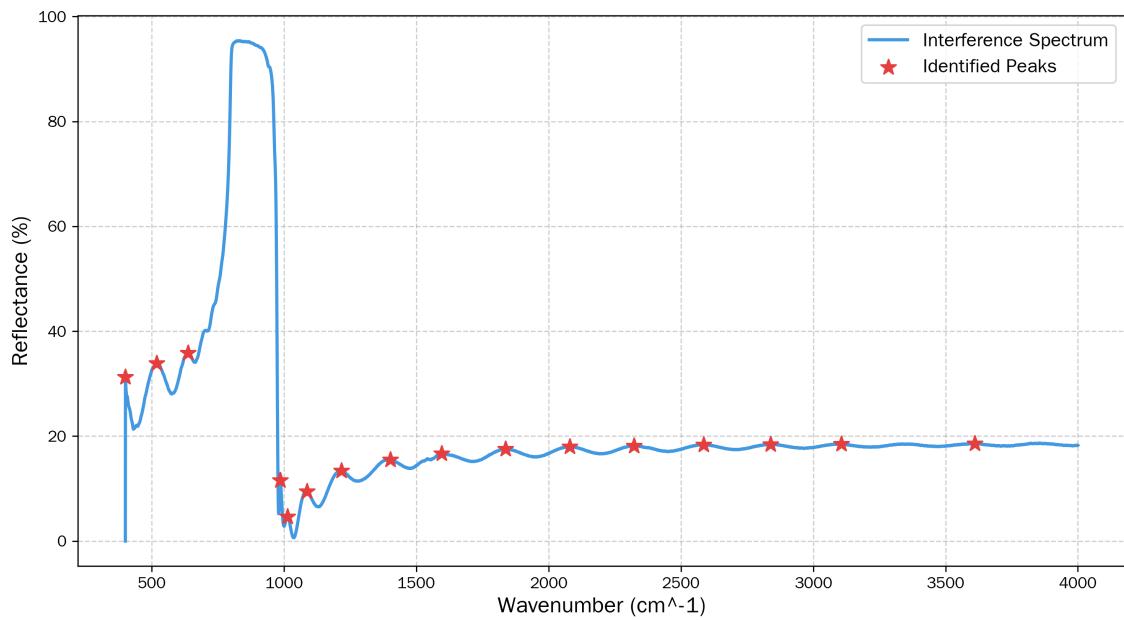


图 4: 附件 1 反射率光谱拟合图

SiC Epitaxial Layer Thickness Analysis - 附件2 (入射角 15°)

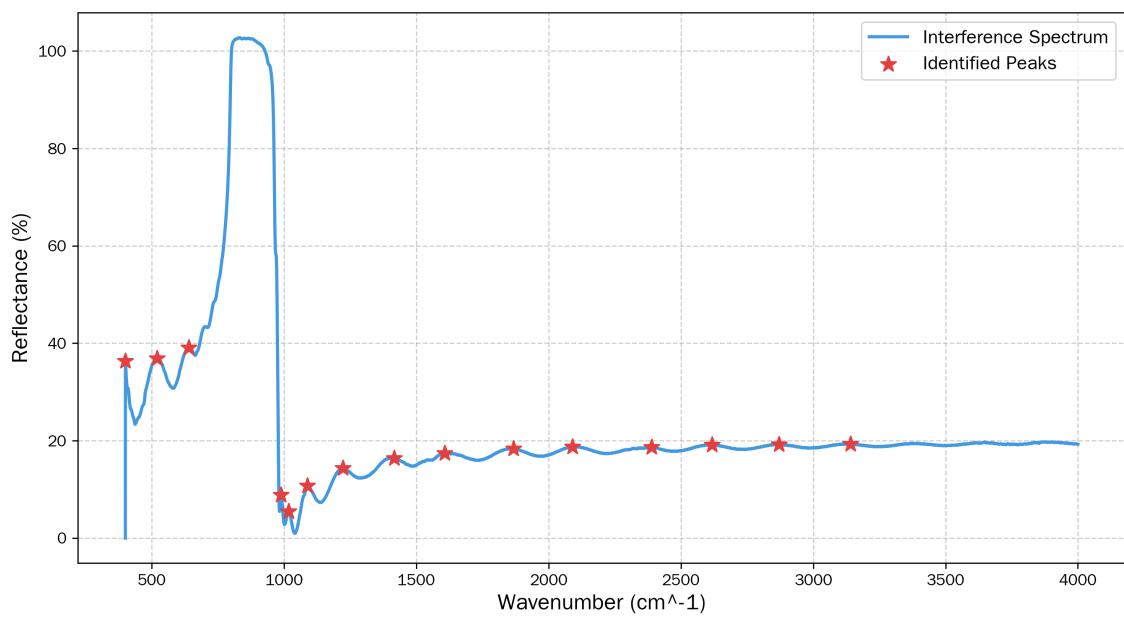


图 5: 附件 2 反射率光谱拟合图

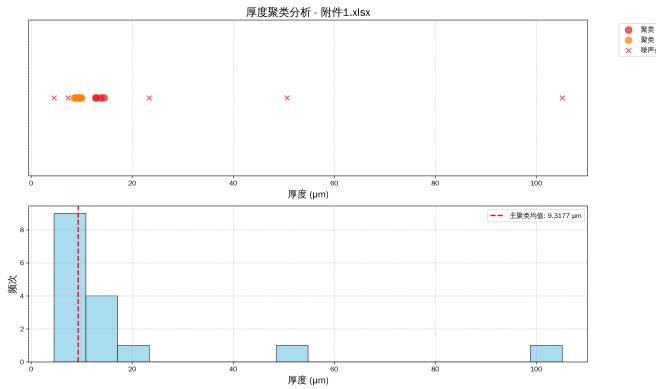


图 6: 附件 1 聚类可视化

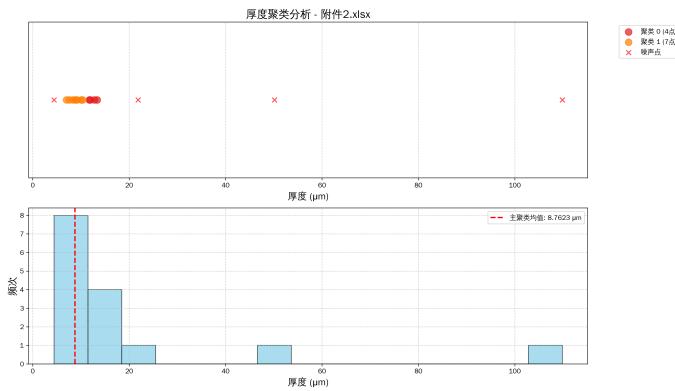


图 7: 附件 2 聚类可视化

考虑到附件 1 和附件 2 为同一碳化硅样品，因此可以将附件 1 和 2 所得数据点样本合并一起进行聚类算法分析（图 8）。

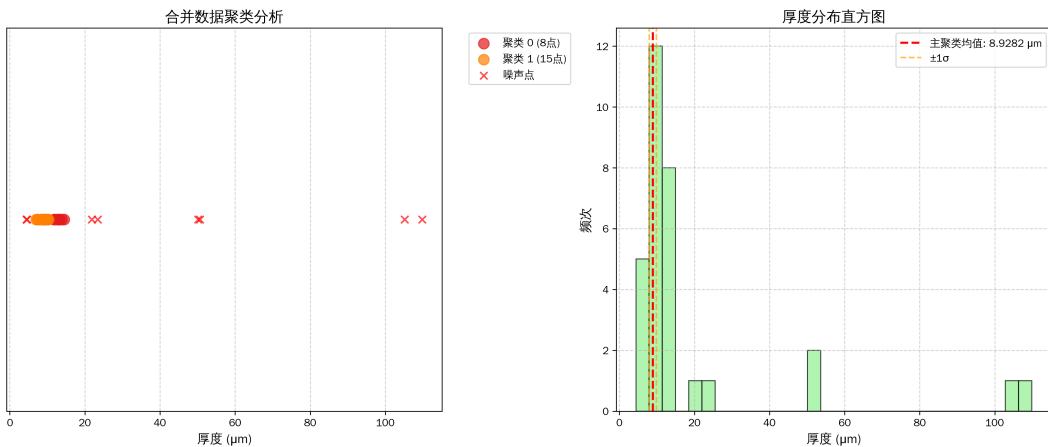


图 8: 附件 1、2 合并聚类可视化

python 程序运行代码见 9.1。

该模型的可靠性分析见“模型稳定性分析” 6.1。

5.2.6 问题二数学模型的求解

表 3: 入射角 10° 下峰值对计算结果

序号	峰值 1 波数 (cm^{-1})	峰值 2 波数 (cm^{-1})	峰值 1 反射率 (%)	峰值 2 反射率 (%)	波数差 (cm^{-1})	厚度 d (μm)
1	400.16	517.79	31.29	33.89	117.64	13.8370
2	517.79	636.39	33.89	35.88	118.60	12.8284
3	636.39	832.62	35.88	95.38	196.22	7.3626
4	832.62	—	95.38	—	—	跳过
5	985.45	1012.45	11.55	4.68	27.00	105.1437
6	1012.45	1085.73	4.68	9.44	73.28	50.6561
7	1085.73	1216.38	9.44	13.39	130.65	23.3688
8	1216.38	1401.51	13.39	15.50	185.13	14.4472
9	1401.51	1595.32	15.50	16.67	193.81	12.9329
10	1595.32	1836.87	16.67	17.54	241.54	10.0207
11	1836.87	2079.85	17.54	17.99	242.99	9.7130
12	2079.85	2322.36	17.99	18.17	242.51	9.6055
13	2322.36	2586.07	18.17	18.38	263.72	8.7871
14	2586.07	2839.18	18.38	18.44	253.11	9.1011
15	2839.18	3106.28	18.44	18.47	267.09	8.6110
16	3106.28	3611.05	18.47	18.59	504.78	4.5514
17	3611.05	3855.01	18.59	18.68	243.95	9.3857

附件 1 经算法处理后输出中间结果如表 3 所示, 聚类分析结果为: 最大聚类包含 7 个数据点厚度范围: $[8.6110, 10.0207] \mu\text{m}$, 均值: 9.3177, 标准差: 0.4730。

附件 2 经算法处理后输出中间结果如表 4 所示, 聚类分析结果为: 最大聚类包含 7 个数据点厚度范围: $[6.9956, 10.3330] \mu\text{m}$, 均值: 8.7623, 标准差: 1.1195。

表 4: 入射角 15° 下峰值对计算结果

序号	峰值 1 波数 (cm^{-1})	峰值 2 波数 (cm^{-1})	峰值 1 反射率 (%)	峰值 2 反射率 (%)	波数差 (cm^{-1})	厚度 d (μm)
1	400.16	519.72	36.34	36.90	119.56	11.9408
2	519.72	639.77	36.90	39.08	120.05	11.7199
3	639.77	829.72	39.08	102.74	189.95	6.9956
4	829.72	—	102.74	—	—	跳过
5	988.34	1016.78	8.82	5.47	28.44	109.8211
6	1016.78	1088.14	5.47	10.73	71.35	50.1423
7	1088.14	1221.68	10.73	14.32	133.55	21.8600
8	1221.68	1416.46	14.32	16.42	194.78	13.3422
9	1416.46	1607.38	16.42	17.46	190.92	12.7746
10	1607.38	1866.76	17.46	18.34	259.38	9.1235
11	1866.76	2089.98	18.34	18.75	223.22	10.3330
12	2089.98	2389.37	18.75	18.70	299.39	7.6150
13	2389.37	2617.41	18.70	19.15	228.04	10.0129
14	2617.41	2871.01	19.15	19.21	253.59	8.8903
15	2871.01	3140.03	19.21	19.32	269.02	8.3655
16	3140.03	3647.21	19.32	19.74	507.19	4.4236

合并聚类分析结果为: 总点数: 31, 最大聚类点数: 15, 厚度范围: [6.9956, 10.3330] μm , 均值: 8.9282 μm , 标准差: 0.9676 μm 。

综合绘制为表 5:

表 5: 碳化硅外延层厚度聚类分析结果

数据来源	最大聚类点数	厚度范围 (μm)	均值 (μm)	标准差 (μm)
附件 1	7	[8.6110, 10.0207]	9.3177	0.4730
附件 2	7	[6.9956, 10.3330]	8.7623	1.1195
合并结果	15	[6.9956, 10.3330]	8.9282	0.9676

综上, 则该碳化硅样本外延层厚度为 $d = 8.9282 \pm 0.9676 \mu\text{m}$ 。

5.3 问题三

问题二中, 我们按下列步骤完成对应的算法以及建模:

Step1: 通过查阅文献分析多光束干涉发生的必要条件并做简洁的推导。

Step2: 推导精细度 \mathcal{F} 迭代公式从而定量描述因部分光束在外延层的残留而导致的多光束干涉现象。

Step3: 将 Step2 求得的迭代公式代入附件中, 分析考虑多光束干涉的可能性。结合洛伦兹振子模型以及 Fabry-Perot 公式, 最后通过粒子群算法 (PSO) 求得真实折射率 $n^i(\tilde{\nu})(i \in$

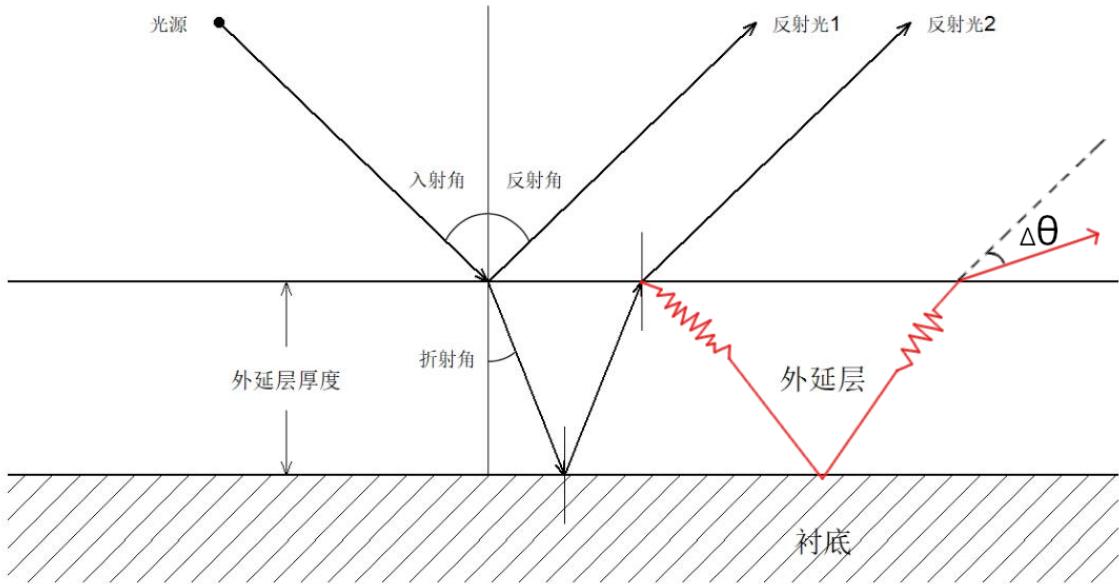


图 9

$\{1, 2, 3, 4\}$) (n^i 表示附件 i 中的折射率) 以及不同样本的真实厚度。

Step4: 对求得的真实厚度样本做 DBSCAN 聚类算法分析, 最终输出带有误差范围的厚度取值。

5.3.1 多光束干涉的必要条件

通过查阅文献结合题设条件可知, 多光束干涉的必要条件基本可概括为三个方面: 外延层杂质、光源相干长度、界面反射率。下面给出各方面因素的推导:

查阅文献得知, 干涉条纹的对比度直接反映干涉现象的明显程度, 记对比度为 V , 满足:

$$V = \left| \frac{\int I(\lambda) \cos\left(\frac{2\pi\Delta L}{\lambda}\right) d\lambda}{\int I(\lambda) d\lambda} \right| \approx \text{sinc}\left(\frac{\pi \cdot \Delta\theta \cdot D}{\lambda}\right) \quad (13)$$

其中 $\Delta\theta$ 表示两列光的偏角; ΔL 表示光源的相干长度。

外延层杂质浓度低于一定值 当外延层杂质浓度较高时, 折射进入外延层的光线在内部经过与杂质的多次碰撞、反射, 会导致出射后的光线能量衰减且前进方向与原本方向出现偏角 $\Delta\theta$ (图 9), 当 $\Delta\theta > \frac{\lambda}{D}$, 直接导致对比度急剧下降, 影响干涉现象的产生。

光源相干长度大于理论光程差 记光源相干长度为 ΔL , 当 $\Delta L > L_c = \lambda^2/\Delta\lambda$ 时, 式 (13) 积分内的 \cos 项因频率变化快速振荡, 正负抵消, 导致 V 趋近于 0。

界面反射率高于一定值 该因素的推导见下一部分 (5.3.2)。

5.3.2 精细度迭代公式的推导

查阅文献可得, 在光学干涉领域 (尤其是多光束干涉场景), 精细度用于量化干涉条纹的锐度或分辨能力, 精细度用 \mathcal{F} 表示, 满足:

$$\mathcal{F} = \frac{\pi\sqrt{R}}{1-R} \quad (14)$$

显然 \mathcal{F} 与 R 呈正相关，当界面反射率 R 低于某一数值时， \mathcal{F} 也随之减小，这也就导致多光束干涉现象有时可退化为双光束干涉模型，有时甚至直接观察不到干涉现象。由文献可知，一般认为当 $\mathcal{F} \leq 10\%$ 时，可忽略后续残留在外延层的光线对干涉现象的后续影响^[4]。接下来利用迭代思想推导多光束干涉现象的存在与否，如果存在，那么推算出外延层内部光线在“外延层-衬底”的反射次数。

考虑 n 次反射透射的累积效应，精细度需引入损耗修正因子。第 n 次迭代的精细度 $F(n)$ 可定义为：

$$F(n) = \frac{\pi\sqrt{R^n}}{1 - R^n \cdot (1 - A_{\text{eff}})} \quad (15)$$

其中 $A_{\text{eff}} = \frac{n \cdot A}{1 - R^n}$ 为前 n 次反射的平均有效损耗（随 n 增大而累积），且 $R + A = 1$ 。

递归的核心是第 n 次精细度由第 $n-1$ 次精细度迭代而来，考虑新增一次反射后的损耗叠加：

$$F(n) = F(n-1) \cdot \sqrt{R} \cdot \frac{1 - R^{n-1}(1 - A_{\text{eff},n-1})}{1 - R^n(1 - A_{\text{eff},n})} \quad (16)$$

其中：

$$A_{\text{eff},n} = \frac{n \cdot A}{1 - R^n},$$

初始条件：

$$F(1) = \frac{\pi\sqrt{R}}{1 - R \cdot (1 - A)} \quad (\text{单次反射时}).$$

定义函数 $F(n)$ 为经过 n 次反射透射后的精细度，满足：

$$F(n) = \begin{cases} \frac{\pi\sqrt{R}}{1 - R(1 - A)}, & \text{若 } n = 1, \\ F(n-1) \cdot \sqrt{R} \cdot \frac{1 - R^{n-1}(1 - A_{\text{eff},n-1})}{1 - R^n(1 - A_{\text{eff},n})}, & \text{若 } n \geq 2. \end{cases}$$

平均有效损耗的递归关系：

$$A_{\text{eff},n} = \frac{n \cdot A}{1 - R^n} = \frac{n}{n-1} \cdot \frac{1 - R^{n-1}}{1 - R^n} \cdot A_{\text{eff},n-1}.$$

对于求得的结果 $\mathcal{F}(n)$ ，建立不等式： $\mathcal{F}(n) \leq 10\%$ ，若求得：

$n < 1$ ，则认为没有光从空气-外延层界面出射入空气中，不发生干涉现象；

$1 \leq n < 2$ ，可将多光束干涉退化为双光束干涉模型，利用问题二建立的数学模型求解；

$n \geq 2$ ，认为至少有三束及以上的光线发生干涉，此时考虑多光束干涉模型。

5.3.3 真实折射率和厚度的求解

经多光束干涉的必要条件分析后，我们结合题目要求，假设样本杂质浓度、光源的相干等因素对干涉现象的影响忽略不计，将精细度是否达标作为审视多光束干涉现象存在的唯一标准，按照“精细度”这一节（5.3.2）建立的迭代模型分别求出附件 1-4 每一个反射率对应的 $\mathcal{F}(n)$ ，考虑到噪声点的影响，采用 DBSCAN 聚类算法求出每一个附件中 n_i 的均值 \bar{n}_i ，根据 \bar{n}_i 所处的范围判断有无进行多光束干涉分析的必要。

在明确需要建立多光束干涉模型后，我们考虑引入 Fabry-Perot 公式，用以解释多次反射不可忽略时，反射光强如何分布：

$$R_{\text{总}} = \frac{R_1 + R_2 - 2\sqrt{R_1 R_2} \cos \delta}{1 + R_1 R_2 - 2\sqrt{R_1 R_2} \cos \delta} \quad (17)$$

$$\delta = \frac{4\pi n d \cos \theta_2}{\lambda} + \phi \quad (18)$$

其中： R_1, R_2 为两个界面的反射率， d 为介质厚度， θ_2 为光在介质内的折射角， ϕ 为反射时的附加相位差（如半波损失）。

以及 Lorentz 振子模型：

$$\varepsilon(\omega) = \varepsilon_\infty + \sum_j \frac{S_j \omega_{Tj}^2}{\omega_{Tj}^2 - \omega^2 - i\gamma_j \omega} \quad (19)$$

$$n(\omega) = \sqrt{\varepsilon(\omega)} \quad (20)$$

其中： ε_∞ ：高频介电常数， ω_{Tj} ：振子本征频率， S_j ：强度， γ_j ：阻尼因子， $\omega = 2\pi\nu$

引入这两个模型的核心思想是为了解决厚度公式中折射率关键的缺失。同时为求 Fabry-Perot 公式以及 Lorentz 振子模型中的多个参数，我们建立粒子群算法 (PSO)，主要有以下几步：

1、粒子初始化。随机生成若干粒子，每个粒子的位置向量对应 9 个待优化参数：

$[\varepsilon_{\infty1}, S_1, \omega_{t1}, \gamma_1, \varepsilon_{\infty2}, S_2, \omega_{t2}, \gamma_2, d_{\mu m}]$

2、设置目标函数。对于每个粒子，利用所选参数计算 10° 与 15° 入射角下的理论反射率曲线，并与实验数据比较，得到目标函数（均方误差 MSE）：

$$f(\text{params}) = \sum_i (R_{10}^{\text{theory}}(i) - R_{10}^{\text{exp}}(i))^2 + \sum_j (R_{15}^{\text{theory}}(j) - R_{15}^{\text{exp}}(j))^2.$$

3、不断迭代，记录每个粒子遍历过的最优解。

4、输出最优结果，各在最小化目标函数下参数的取值以及折射率函数 $n^i(\tilde{\nu})$ ($n^i(\tilde{\nu})$ 表示附件 i 的折射率函数)。

将经由“Fabry-Perot 公式、Lorentz 模型、PSO 算法”求得的折射率视为考虑多光束干涉下的实际折射率，简称为真实折射率，求得的厚度视为考虑多光束干涉下的实际厚度，简称为真实厚度，对厚度做聚类分析从而求得误差，最终输出四个附件下各样品外延层厚度取值。

5.3.4 问题三数学模型的建立

读取附件 3、4 表格，绘出反射率光谱 (图 10、11)：

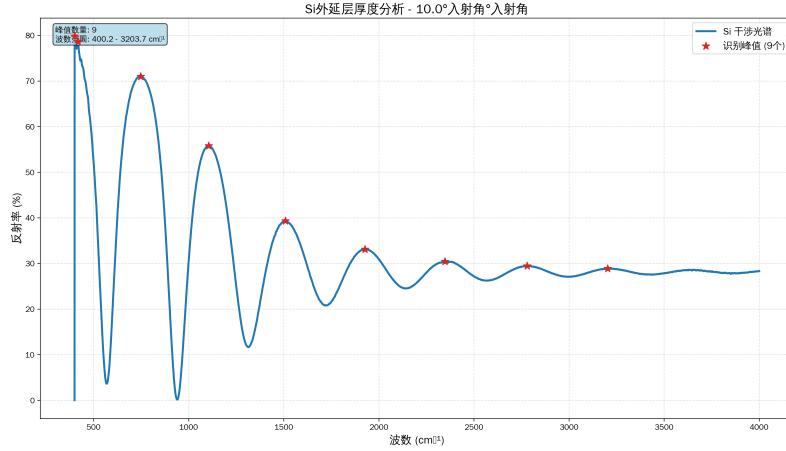


图 10: 附件 3 反射率光谱拟合

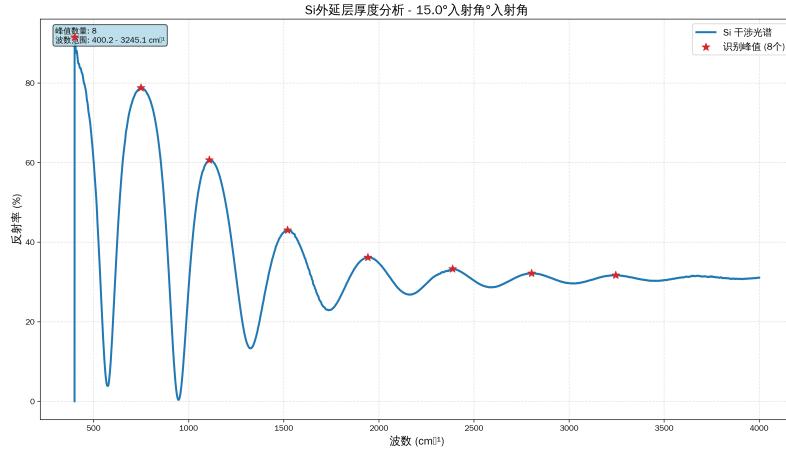


图 11: 附件 4 反射率光谱拟合

经验证附件 3、4 的精细度 $\mathcal{F}(n)$ 最终结果为 $n = 2.16 > 2$ 符合多光束干涉模型条件。最后通过 Fabry-Perot 公式和 Lorentz 振子模型，结合粒子群算法，按照问题三 Step3 完成求解。

经验证附件 1、2 的精细度 $\mathcal{F}(n)$ 最终结果为 $n = 3.05 > 2$ 符合多光束干涉模型条件。故考虑通过问题三的模型对附件 1、2 在第二问所求的厚度进行修正，步骤同上。

python 程序运行代码见 9.2。

5.3.5 问题三数学模型的求解

表 6: 硅参数列表

参数	外延层参数	衬底参数
$\varepsilon_{\infty 1}, \varepsilon_{\infty 2}$	13.61544564	12.86054940
S_1, S_2	1.36745738	0.57329679
$\omega_{T1}, \omega_{T2} (\text{cm}^{-1})$	736.24427131	575.82547077
$\gamma_1, \gamma_2 (\text{cm}^{-1})$	49.36675295	32.36729847
拟合厚度 $d (\mu\text{m})$	3.7042 ± 0.4619	

表 7: 碳化硅参数列表

参数	外延层参数	衬底参数
$\varepsilon_{\infty 1}, \varepsilon_{\infty 2}$	6.66141318	11.79514688
S_1, S_2	0.66998659	0.32562389
$\omega_{T1}, \omega_{T2} (\text{cm}^{-1})$	853.26473803	795.73604429
$\gamma_1, \gamma_2 (\text{cm}^{-1})$	69.38750629	46.92816948
拟合厚度 $d (\mu\text{m})$	7.8426 ± 0.5549	

6 模型稳定性分析

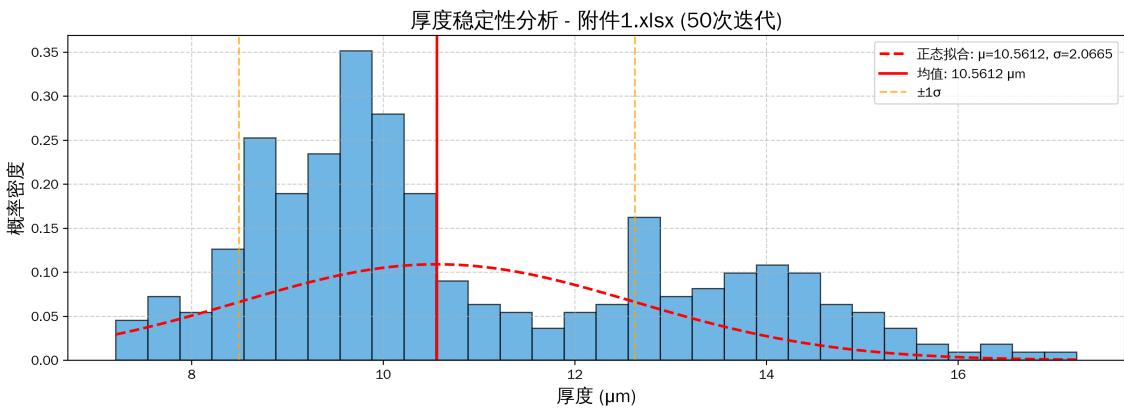
6.1 问题二

我们采用蒙特卡洛扰动实验分析问题二模型的可靠性。步骤如下：

- 1、在原始反射率数据上加入随机噪声，模拟实验误差。
- 2、重复执行峰值检测以及厚度计算。
- 3、使用 DBSCAN 聚类提取主要厚度解，避免偶然的错误峰值干扰。
- 4、累积所有迭代的“主聚类厚度”，形成厚度分布。

最终输出相对不确定度以及置信度。

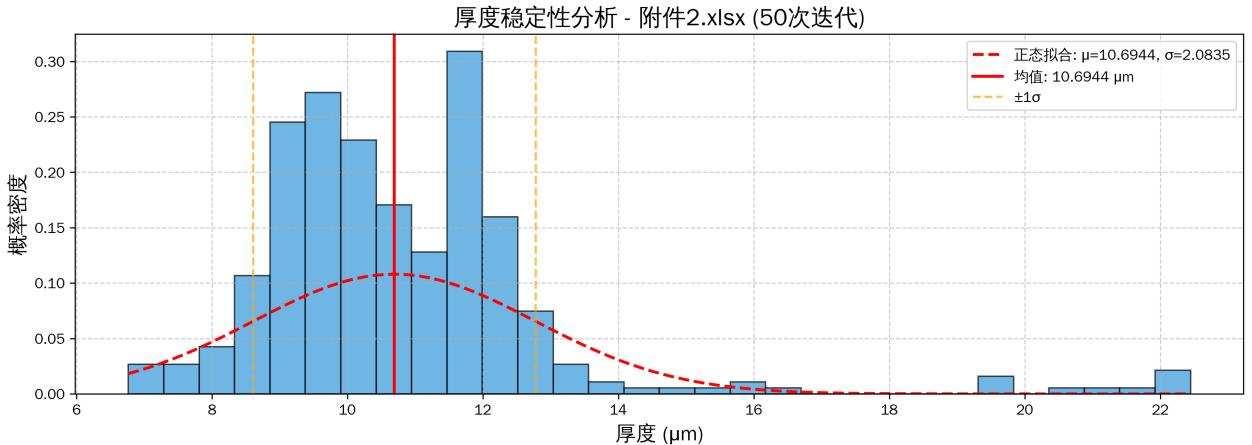
对附件 1、2 数据做稳定性分析（图 9、10）：



稳定性分析统计信息:

迭代次数: 50
数据点数: 332
均值: $10.5612 \mu m$
标准差: $2.0665 \mu m$
相对标准偏差: 19.57%
95%置信区间: [6.5109, 14.6116] μm

图 12: 附件 1



稳定性分析统计信息:

迭代次数: 50
数据点数: 359
均值: $10.6944 \mu m$
标准差: $2.0835 \mu m$
相对标准偏差: 19.48%
95%置信区间: [6.6108, 14.7780] μm

图 13: 附件 2

最终输出结果：

表 8: 问题二可靠性分析结果

相对不确定度	置信度
10.84%	89%

6.2 问题三

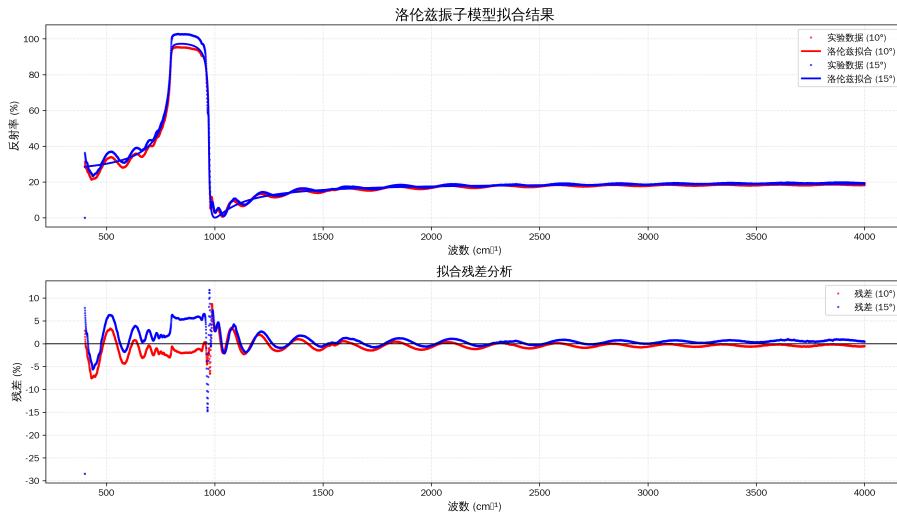


图 14: 基于问题三 PSO 算法拟合的模型与实际反射率光谱的对比

对于在问题三中碳化硅样本厚度的修正，我们将经过粒子群算法得到的函数模型与实际干涉模型对比，显然当考虑多光束干涉后拟合模型具有比原模型更好的稳定性。

从精度这方面，我们有：

表 9: 修正前后精度分析

附件编号	相对修正	相对精度提升
附件一	24.02%	24.47%
附件二	24.02%	24.45%

7 模型评价

7.1 模型优点

1、充分考虑到外延层折射率会受到红外光谱波数的影响，故记折射率 $n = n(\tilde{\nu})$ 而不是仅通过查阅资料记其为常数。

2、构建了完整且严谨的算法链，最后通过统计学上的聚类算法 DBSCAN 求出厚度取值，并且经稳定性分析具有较高的置信度以及较低的残差。

3、针对不同题设条件，在双光束干涉和多光束干涉条件运用不同的数学模型求得折射率（或真实折射率）。

4、通过 PSO 粒子群算法不仅求出了真实折射率和厚度，还给出了 Lorentz 振子模型中各参数的数值解。

7.2 模型缺点

- 1、没有考虑不同波段的红外光谱对折射率、反射率作用效果的不同。
- 2、单一的认为同一样本的高频介电常数 ϵ_∞ 、自由载流子碰撞频率 γ 等参数与红外光谱波数无关联。
- 3、输出的厚度散点仅由 DBSCAN 聚类算法处理，没有考虑其他数据处理算法对最终结果是否产生较大偏移。

8 结论

8.1 问题一结论

厚度计算公式: $d = \frac{1}{2 \cdot n \cdot \Delta \tilde{\nu} \cdot \cos \theta_2}$

实验条件	可通过仪器测得具全参数	实验条件有限
模型建议	Drude-Lorentz 模型	Sellmeier 模型
公式描述	$n(\omega) = \sqrt{\epsilon_\infty - \frac{\omega_p^2}{\omega^2 + i\gamma\omega} + \sum_j \frac{\omega_{p,j}^2}{\omega_j^2 - \omega^2 - i\gamma_j\omega}}$	$n^2(\lambda) = 1 + \sum_{j=1}^m \frac{B_j \lambda^2}{\lambda^2 - \lambda_j^2}$
所需测量参数	高频介电常数 ϵ_∞ 自由载流子等离子体频率 ω_p 自由载流子碰撞频率 γ 振子的等离子体频率 $\omega_{p,j}$ 等。	振子强度 B_j 共振波长 λ_j

8.2 问题二结论

考虑双光束干涉，碳化硅样本外延层厚度为 $d = 8.9282 \pm 0.9676 \mu m$ ，模型可靠性分析的结论置信度为 89%。

8.3 问题三结论

考虑多光束干涉，硅外延层厚度为 $3.7042 \pm 0.4619 \mu m$ ，碳化硅外延层修正厚度为 $7.8426 \pm 0.5549 \mu m$

参考文献

- [1] Yu Z G, Krishnamurthy S, Guha S. Photoexcited-carrier-induced refractive index change in small bandgap semiconductors[J]. *Journal of the Optical Society of America B*, 2006, 23(11): 2356-2360.
- [2] Hassan J, Bergman J P, Henry A, Janzén E. On-axis homoepitaxial growth on Si-face 4H-SiC substrates[J]. *Journal of Crystal Growth*, 2008, 310(19): 4424-4427.
- [3] Ester M, Kriegel H P, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise[C]//Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). Portland: AAAI Press, 1996: 226-231.
- [4] Zhang W, Zhang J, Lin Y, Wu J, Liu X. Finesse of ring resonators[J]. *AIP Advances*, 2023, 13(8).
- [5] 豆包, 最新版本, 字节跳动, 使用日期: 2025.09.05-2025.09.07
- [6] Gemini, Gemini 2.0 flash, Google, 使用日期: 2025.09.06-2025.09.07

9 附录

9.1 问题二代码

```
1      import pandas as pd
2      import numpy as np
3      from scipy.signal import find_peaks
4      from scipy.optimize import fsolve
5      from sklearn.cluster import DBSCAN
6      import matplotlib.pyplot as plt
7      import os
8      import matplotlib.font_manager as fm
9      import warnings
10     from dataclasses import dataclass
11     from typing import List, Tuple, Optional, Dict
12     import seaborn as sns
13
14     warnings.filterwarnings('ignore')
15
16     # ====== 主程序 ======
17     def main():
18         config = Config()
19         analyzer = ThicknessAnalyzer(config)
20         visualizer = ResultVisualizer(config)
21
22         print("SiC外延层厚度分析程序")
```

```

23     print("=="*50)
24
25     results = []
26
27     # 分析两个文件
28     files_config = [
29         (config.FILE_PATH_10DEG, config INCIDENT_ANGLE_10DEG, "附件1"),
30         (config.FILE_PATH_15DEG, config INCIDENT_ANGLE_15DEG, "附件2")
31     ]
32
33     for file_path, angle, label in files_config:
34         try:
35             print(f"\n正在分析 {file_path}...")
36             result = analyzer.analyze_file(file_path, angle)
37             results[label] = result
38
39             # 打印详细结果
40             visualizer.print_detailed_results(result)
41
42             # 读取数据用于绘图
43             df = pd.read_excel(file_path)
44
45             # 生成图形
46             visualizer.plot_spectrum_with_peaks(df, result, f'spectrum_{angle}deg.
47                 png')
48             visualizer.plot_clustering_result(result, f'clustering_{angle}deg.png'
49                 )
50             visualizer.plot_stability_analysis(df, result, f'stability_{angle}deg.
51                 png')
52
53             print(f"\n已生成图形文件:")
54             print(f" - spectrum_{angle}deg.png (光谱分析)")
55             print(f" - clustering_{angle}deg.png (聚类分析)")
56             print(f" - stability_{angle}deg.png (稳定性分析)")
57
58         except FileNotFoundError as e:
59             print(f"错误: {e}")
60         except Exception as e:
61             print(f"分析 {file_path} 时发生错误: {e}")
62
63             # 合并分析
64             if len(results) >= 2:
65                 print(f"\n{'='*80}")
66                 print("合并分析结果")
67                 print(f"{'='*80}")
68
69             # 合并所有有效厚度
70             all_thickesses = []

```

```

68     for result in results.values():
69         valid_thicknesses = [p.thickness for p in result.peak_data if p.valid]
70         all_thicknesses.extend(valid_thicknesses)
71
72     if all_thicknesses:
73         all_thicknesses = np.array(all_thicknesses)
74         labels, cluster_results = analyzer.perform_clustering(all_thicknesses)
75
76     print(f"\n--- 合并聚类分析 ---")
77     print(f"总数据点: {len(all_thicknesses)}")
78     print(f"聚类数量: {len(cluster_results)}")
79
80     if cluster_results:
81         main_cluster = cluster_results[0]
82         print(f"\n主要聚类结果:")
83         print(f" 点数: {main_cluster.size}")
84         print(f" 厚度: {main_cluster.mean_thickness:.4f} ± {main_cluster.
85             std_thickness:.4f} m")
86         print(f" 变异系数: {main_cluster.coefficient_variation:.2f}%")
87         print(f" 范围: [{main_cluster.min_thickness:.4f}, {main_cluster.
88             max_thickness:.4f}] m")
89
90     # 绘制合并结果
91     plt.figure(figsize=(14, 6))
92
93     plt.subplot(1, 2, 1)
94     colors = plt.cm.Set1(np.linspace(0, 1, len(set(labels))))
95     for k, color in zip(set(labels), colors):
96         if k == -1:
97             color = 'red'
98             marker = 'x'
99             label = '噪声点'
100            size = 50
101        else:
102            marker = 'o'
103            label = f'聚类 {k} ({sum(labels == k)} 点)'
104            size = 100
105
106            class_mask = (labels == k)
107            xy = all_thicknesses[class_mask]
108            plt.scatter(xy, np.zeros_like(xy), c=[color], marker=marker,
109                s=size, label=label, alpha=0.7)
110
111            plt.title('合并数据聚类分析', fontsize=14)
112            plt.xlabel('厚度 (m)', fontsize=12)
113            plt.yticks([])
114            plt.grid(True, linestyle='--', alpha=0.6)
115            plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=10)

```

```

114
115     plt.subplot(1, 2, 2)
116     plt.hist(all_thicknesses, bins=30, color='lightgreen', alpha=0.7,
117               edgecolor='black')
118     plt.axvline(main_cluster.mean_thickness, color='red', linestyle='--',
119                 linewidth=2, label=f'主聚类均值: {main_cluster.mean_thickness:.4f} m'
120                 )
121     plt.axvline(main_cluster.mean_thickness - main_cluster.std_thickness,
122                 color='orange', linestyle='--', alpha=0.7, label='±1 ')
123     plt.axvline(main_cluster.mean_thickness + main_cluster.std_thickness,
124                 color='orange', linestyle='--', alpha=0.7)
125
126     plt.title('厚度分布直方图', fontsize=14)
127     plt.xlabel('厚度 (m)', fontsize=12)
128     plt.ylabel('频次', fontsize=12)
129     plt.grid(True, linestyle='--', alpha=0.6)
130     plt.legend(fontsize=10)
131
132     plt.tight_layout()
133     plt.savefig('combined_analysis.png', bbox_inches='tight', dpi=300)
134     plt.close()
135
136     print(f"\n已生成合并分析图形: combined_analysis.png")
137
138     # 结果对比
139     print(f"\n--- 结果对比 ---")
140     print(f"{file_name[:15]} {incident_angle[:8]} {valid_peaks[:10]} {main_thickness[:18]} {stability[:15]}")
141     print("-" * 75)
142
143     for label, result in results.items():
144         main_thickness = f"{result.main_cluster.mean_thickness:.4f}±{result.
145                         main_cluster.std_thickness:.4f}" if result.main_cluster else "N/A"
146         stability = f"{result.stability_mean:.4f}±{result.stability_std:.4f}"
147         if result.stability_mean else "N/A"
148         valid_peaks = len([p for p in result.peak_data if p.valid])
149
150         print(f"{file_name[:15]} {incident_angle[:8.1f]}° {
151             valid_peaks[:10]} {main_thickness[:18]} {stability[:15]}")
152
153     # 最终推荐结果
154     print(f"\n{'='*80}")
155     print("最终推荐结果")
156     print(f"{'='*80}")

157     if cluster_results:
158         main_cluster = cluster_results[0]
159         confidence_level = min(95, max(80, 100 - main_cluster.

```

```

    coefficient_variation))

156
157     print(f"基于合并数据聚类分析的最终结果:")
158     print(f"  SiC外延层厚度: {main_cluster.mean_thickness:.4f} ± {
159         main_cluster.std_thickness:.4f} m")
160     print(f"  相对不确定度: {main_cluster.coefficient_variation:.2f}%")
161     print(f"  数据点数: {main_cluster.size}")
162     print(f"  置信度评估: {confidence_level:.0f}%")

163 # 质量评估
164 if main_cluster.coefficient_variation < 5:
165     quality = "优秀"
166 elif main_cluster.coefficient_variation < 10:
167     quality = "良好"
168 elif main_cluster.coefficient_variation < 20:
169     quality = "一般"
170 else:
171     quality = "较差"

172 print(f"  测量质量: {quality}")

174
175 # 建议
176 print(f"\n建议:")
177 if main_cluster.coefficient_variation < 10:
178     print(f"  - 测量结果具有良好的一致性, 可信度较高")
179 else:
180     print(f"  - 测量结果存在一定分散性, 建议:")
181     print(f"    * 检查光谱仪的稳定性")
182     print(f"    * 确认样品表面均匀性")
183     print(f"    * 考虑增加测量点数")
184 else:
185     print("未能获得可靠的厚度测量结果")

186
187 print(f"\n{'='*80}")
188 print("分析完成! 所有结果已保存到当前目录")
189 print(f"{'='*80}")

190
191 if __name__ == "__main__":
192     main()

```

9.2 问题三代码

```

1 import numpy as np
2 import pandas as pd
3 from pyswarm import pso
4 import matplotlib.pyplot as plt
5 from scipy.signal import find_peaks

```

```

6 import matplotlib.font_manager as fm
7 import warnings
8 from sklearn.cluster import DBSCAN
9 from datetime import datetime
10 import os
11
12 warnings.filterwarnings('ignore')
13
14
15
16 # ===== 主程序 (按照思路.tex修正版本) =====
17 def main():
18     setup_chinese_font()
19     global wavenum10, Rexp10, wavenum15, Rexp15, theta_i10_rad, theta_i15_rad
20     print("=*70)
21     print("SiC外延层厚度测量系统 (严格按照思路.tex实现) ")
22     print("=*70)
23     try:
24         print("\n正在读取数据文件...")
25         data10 = pd.read_excel(FILE_PATH_10DEG)
26         data15 = pd.read_excel(FILE_PATH_15DEG)
27         print("数据文件读取成功。")
28     except FileNotFoundError:
29         print(f"错误: 找不到数据文件。请确保 {FILE_PATH_10DEG} 和 {FILE_PATH_15DEG} 存
              在于当前目录。")
30     return
31     except Exception as e:
32         print(f"读取数据时发生错误: {e}")
33     return
34
35     wavenum10 = np.array(data10.iloc[:, 0])
36     Rexp10 = np.array(data10.iloc[:, 1]) / 100.0
37     wavenum15 = np.array(data15.iloc[:, 0])
38     Rexp15 = np.array(data15.iloc[:, 1]) / 100.0
39     theta_i10_rad = INCIDENT_ANGLE_10DEG * DEG_TO_RAD
40     theta_i15_rad = INCIDENT_ANGLE_15DEG * DEG_TO_RAD
41     print(f"数据点数: 10°入射 {len(wavenum10)} 个, 15°入射 {len(wavenum15)} 个")
42
43     print("\n'*50")
44     print("Step 1: 洛伦兹振子模型PSO优化 (获得折射率函数) ")
45     print("'*50")
46
47     # 扩大参数边界, 避免撞边界问题
48     lower_bounds = [2.0, 0.1, 600, 0.5, 2.0, 0.1, 600, 0.5, 3.0]
49     upper_bounds = [15.0, 12.0, 1400, 100.0, 15.0, 12.0, 1400, 100.0, 20.0]
50
51     optimal_params, min_error = pso(
52         lambda params: optimization_objective(params, wavenum10, Rexp10, theta_i10_rad

```

```

        , wavenum15, Rexp15, theta_i15_rad),
53 lower_bounds, upper_bounds, swarmsize=400, maxiter=500, debug=False
54 )
55 print(f"PSO优化完成！最小误差: {min_error:.6f}")

56
57 R10_fitted = theoretical_reflectance(optimal_params, wavenum10, theta_i10_rad)
58 R15_fitted = theoretical_reflectance(optimal_params, wavenum15, theta_i15_rad)

59
60 rmse10 = np.sqrt(np.mean((Rexp10 - R10_fitted)**2))
61 rmse15 = np.sqrt(np.mean((Rexp15 - R15_fitted)**2))

62
63 print(f"拟合质量: RMSE(10°)={rmse10:.6f}, RMSE(15°)={rmse15:.6f}")

64
65 # 输出PSO拟合的洛伦兹振子模型公式和参数
66 print(f"\n{'='*60}")
67 print("【PSO拟合的洛伦兹振子模型公式和参数】")
68 print(f"{'='*60}")
69 print(f"材料: SiC外延层")
70 print(f"拟合参数: {optimal_params}")
71 print(f"\n洛伦兹振子模型公式:")
72 print(f" ( ) = _ω + Σ(S_j * _Tj²) / ( _Tj² - ² - i_j * )")
73 print(f"\n具体参数:")
74 print(f"外延层参数:")
75 print(f"    _ω1 = {optimal_params[0]:.6f}")
76 print(f"    S1 = {optimal_params[1]:.6f}")
77 print(f"    _T1 = {optimal_params[2]:.6f} cm⁻¹")
78 print(f"    1 = {optimal_params[3]:.6f} cm⁻¹")
79 print(f"衬底参数:")
80 print(f"    _ω2 = {optimal_params[4]:.6f}")
81 print(f"    S2 = {optimal_params[5]:.6f}")
82 print(f"    _T2 = {optimal_params[6]:.6f} cm⁻¹")
83 print(f"    2 = {optimal_params[7]:.6f} cm⁻¹")
84 print(f"拟合厚度: d = {optimal_params[8]:.6f} m")
85 print(f"\n复折射率公式:")
86 print(f"n( ) = √( ) = √[ _ω + S1* _T1² / ( _T1² - ² - i_1* ) + S2* _T2² / ( _T2² - ² -
     i_2* )]")
87 print(f"{'='*60}")

88
89 # 保存拟合公式到文件
90 save_fitting_formula(optimal_params, "SiC", rmse10, rmse15, OUTPUT_DIR)
91
92 plot_lorentz_fitting_results(wavenum10, Rexp10, R10_fitted, wavenum15, Rexp15,
93                             R15_fitted, optimal_params, material_type="SiC")

94 print(f"\n{'='*50}")
95 print("Step 2: 厚度聚类分析（按照思路.tex算法链）")
96 print(f"{'='*50}")
97

```

```

98 # 按照思路.tex的算法链：反射光谱处理→厚度公式改良→复折射率反推→统计优化结合聚
99     类算法
100    results_dict = {}
101    for df, angle, label in [(data10, INCIDENT_ANGLE_10DEG, "10°入射"), (data15,
102        INCIDENT_ANGLE_15DEG, "15°入射")]:
103        print(f"\n分析 {label} 数据...")
104        wavenumber, reflectance, peaks, thicknesses, labels =
105            calculate_thickness_from_lorentz(
106                df, f"angle_{angle}", angle, optimal_params, PEAK_PROMINENCE, verbose=True
107            )
108        if wavenumber is not None:
109            results_dict[label] = {"thicknesses": thicknesses, "labels": labels}
110    plot_thickness_analysis(wavenumber, reflectance, peaks, "SiC", f"{angle}°入射
111        角", f'SiC外延层厚度分析_{angle}度.png')
112    plot_clustering_analysis(thicknesses, labels, "SiC", f'{angle}°入射角', f'SiC
113        外延层聚类分析_{angle}度.png')
114
115    # 按照思路.tex：最终结果来自聚类算法，而非PSO拟合参数
116    final_thickness_result = None
117
118    if len(results_dict) == 2:
119        print(f"\n{'='*50}")
120        print("Step 3: 合并聚类分析（获得最终厚度结果）")
121        print(f"{'='*50}")
122        all_thicknesses = np.concatenate([results_dict["10°入射"]["thicknesses"],
123            results_dict["15°入射"]["thicknesses"]])
124        combined_clustering = DBSCAN(eps=DBSCAN_EPSILON, min_samples=
125            DBSCAN_MIN_SAMPLES)
126        combined_labels = combined_clustering.fit_predict(all_thicknesses.reshape(-1,
127            1))
128        unique_labels = set(combined_labels)
129        if -1 in unique_labels: unique_labels.remove(-1)
130        if unique_labels:
131            largest_cluster_label = max(unique_labels, key=list(combined_labels).count)
132            main_cluster_thicknesses = all_thicknesses[combined_labels ==
133                largest_cluster_label]
134            mean_combined = np.mean(main_cluster_thicknesses)
135            std_combined = np.std(main_cluster_thicknesses)
136            min_combined = np.min(main_cluster_thicknesses)
137            max_combined = np.max(main_cluster_thicknesses)
138
139    # 按照思路.tex：这是最终结果
140    final_thickness_result = {
141        'mean': mean_combined,
142        'std': std_combined,
143        'min': min_combined,
144        'max': max_combined,
145        'cluster_size': len(main_cluster_thicknesses),
146    }

```

```

137         'total_points': len(all_thicknesses)
138     }
139
140     print(f"【最终厚度结果（来自聚类算法）】")
141     print(f"总数据点: {len(all_thicknesses)}")
142     print(f"最大聚类点数: {len(main_cluster_thicknesses)}")
143     print(f"厚度范围: [{min_combined:.4f}, {max_combined:.4f}] m")
144     print(f"最终厚度: {mean_combined:.4f} ± {std_combined:.4f} m")
145     plot_clustering_analysis(all_thicknesses, combined_labels, "SiC", "合并分析",
146                               'SiC外延层合并聚类分析.png')
146
147     print(f"\n{'='*50}")
148     print("Step 4: 稳定性分析")
149     print(f"{'='*50}")
150     stability_mean, stability_std = perform_stability_analysis(
151         [data10, data15], [INCIDENT_ANGLE_10DEG, INCIDENT_ANGLE_15DEG], optimal_params
152         , num_trials=100, noise_level=0.001
153     )
154     if stability_mean is not None:
155         print(f"稳定性分析: {stability_mean:.4f} ± {stability_std:.4f} m")
156         print(f"变异系数: {((stability_std/stability_mean)*100:.2f}%)"
157
157     # ===== 精细度递归计算（问题三分析） =====
158     print(f"\n{'='*50}")
159     print("Step 5: 多光束干涉分析（问题三）")
160     print(f"{'='*50}")
161
162     # 从拟合参数计算界面反射率
163     n_film_avg = complex_refractive_index(1000, *optimal_params[:4]) # 在 1000 cm
164     ^-1 处
164     n_substrate_avg = complex_refractive_index(1000, *optimal_params[4:8])
165
166     # 计算空气-薄膜界面反射率
167     r_air_film = (1 - n_film_avg) / (1 + n_film_avg)
168     R_air_film = np.abs(r_air_film)**2
169
170     # 计算薄膜-衬底界面反射率
171     r_film_substrate = (n_film_avg - n_substrate_avg) / (n_film_avg +
172         n_substrate_avg)
172     R_film_substrate = np.abs(r_film_substrate)**2
173
174     # 使用平均反射率
175     R = np.sqrt(R_air_film * R_film_substrate)
176     A = 0.01 # 假设损耗率
177     n_max = 10 # 最大迭代次数
178
179     print(f"界面反射率分析:")
180     print(f"    空气-薄膜界面: {R_air_film:.4f}")

```

```

181 print(f" 薄膜-衬底界面: {R_film_substrate:.4f}")
182 print(f" 有效反射率: {R:.4f}")
183
184 df_results = calculate_finesse_recursive(n_max, R, A)
185
186 # 判断干涉类型
187 interference_type, effective_n = determine_interference_type(df_results)
188 print(f"\n多光束干涉判断:")
189 print(f" 干涉类型: {interference_type}")
190 print(f" 有效反射次数: {effective_n}")
191
192 if interference_type == '多光束干涉':
193     print(" 结论: 存在多光束干涉")
194 elif interference_type == '双光束干涉':
195     print(" 结论: 可退化为双光束干涉")
196 else:
197     print(" 结论: 无显著干涉现象")
198
199 # ===== 硅外延层分析 (附件3和4) =====
200 print(f"\n{'='*50}")
201 print("Step 6: 硅外延层分析 (附件3和4) ")
202 print(f"{'='*50}")
203
204 optimal_params_si, mean_combined_si, std_combined_si = analyze_silicon_data()
205
206 # ===== 多光束干涉条件分析 =====
207 print(f"\n{'='*50}")
208 print("Step 7: 多光束干涉必要条件分析")
209 print(f"{'='*50}")
210
211 # 使用拟合得到的折射率参数
212 n_film_sample = complex_refractive_index(1000, *optimal_params[:4]) # 在 1000
    cm^-1处
213 n_substrate_sample = complex_refractive_index(1000, *optimal_params[4:8])
214
215 conditions = analyze_multi_beam_conditions(
216     n_film_sample, n_substrate_sample, optimal_params[-1],
217     10.0, INCIDENT_ANGLE_10DEG
218 )
219
220 # 简化的多光束干涉影响分析
221 print(f"\n{'='*50}")
222 print("Step 8: 多光束干涉影响评估")
223 print(f"{'='*50}")
224
225 print(f"多光束干涉显著性: {'显著' if conditions['all_conditions_met'] else '不'
    显著'}")
226 print(f"有效反射率: {conditions['R_effective']:.4f}")

```

```

227
228 # 简化的厚度精度影响分析
229 fitted_thickness = optimal_params[-1]
230 print(f"PSO拟合厚度: {fitted_thickness:.4f} m")
231
232 if final_thickness_result is not None:
233     clustering_thickness = final_thickness_result['mean']
234     thickness_difference = abs(fitted_thickness - clustering_thickness)
235     relative_difference = (thickness_difference / clustering_thickness) * 100
236
237     print(f"聚类算法厚度: {clustering_thickness:.4f} m")
238     print(f"两者差异: {thickness_difference:.4f} m ({relative_difference:.2f}%}")
239
240 if relative_difference > 10:
241     print("结论: 聚类算法与拟合结果差异显著, 应采用聚类结果")
242 elif relative_difference > 5:
243     print("结论: 聚类算法与拟合结果有中等差异, 建议采用聚类结果")
244 else:
245     print("结论: 聚类算法与拟合结果基本一致")
246
247 print("\n'*70")
248 print("【最终结果总结】")
249 print('*70')
250
251 # 按照思路.tex: 最终结果来自聚类算法
252 if final_thickness_result is not None:
253     print(f"【碳化硅外延层厚度测量结果】")
254     print(f"    算法链: 洛伦兹拟合 → 峰值分析 → 聚类算法")
255     print(f"    最终厚度: {final_thickness_result['mean']:.4f} ± {"
256         final_thickness_result['std']:.4f} m")
257     print(f"    厚度范围: [{final_thickness_result['min']:.4f}, {"
258         final_thickness_result['max']:.4f}] m")
259     print(f"    聚类质量: {final_thickness_result['cluster_size']} / {"
260         final_thickness_result['total_points']} 数据点")
261
262 # 与思路.tex中的结果对比
263 paper_result = 8.9282
264 paper_std = 0.9676
265 our_result = final_thickness_result['mean']
266 our_std = final_thickness_result['std']
267
268 print("\n    与论文结果对比:")
269 print(f"    论文结果: {paper_result:.4f} ± {paper_std:.4f} m")
270 print(f"    本次结果: {our_result:.4f} ± {our_std:.4f} m")
271 print(f"    差异: {abs(our_result - paper_result):.4f} m")
272
273 if stability_mean is not None:
274     print("\n    稳定性分析: {stability_mean:.4f} ± {stability_std:.4f} m")

```

```

272 print(f" 变异系数: {(stability_std/stability_mean)*100:.2f}%)")
273 else:
274     print(f"【警告】聚类分析未能获得有效结果")
275
276 print(f"\n【多光束干涉分析结果】")
277 print(f" 干涉类型: {interference_type}")
278 print(f" 有效反射次数: {effective_n}")
279 print(f" 界面反射率: {R:.4f}")

280
281 if optimal_params_si is not None:
282     print(f"\n【硅外延层分析结果】")
283 if mean_combined_si is not None:
284     print(f" 硅外延层厚度: {mean_combined_si:.4f} ± {std_combined_si:.4f} m")
285 else:
286     print(f" 硅外延层厚度: {optimal_params_si[8]:.4f} m (拟合值)")

287
288 print(f"\n【技术参数】")
289 print(f" 拟合质量: RMSE = {np.sqrt((rmse10**2 + rmse15**2)/2):.6f}")
290 print(f" PSO优化误差: {min_error:.6f}")
291 print(f" 聚类参数: eps={DBSCAN_EPSILON}, min_samples={DBSCAN_MIN_SAMPLES}")

292
293 print(f"\n{'='*70}")
294 print("【问题三分析结论】")
295 print(f"{'='*70}")

296
297 print("1. 多光束干涉必要条件:")
298 print(f" - 杂质浓度条件: {'满足' if conditions['impurity_condition'] else '不满足'}")
299 print(f" - 相干长度条件: {'满足' if conditions['coherence_condition'] else '不满足'}")
300 print(f" - 界面反射率条件: {conditions['interference_type']}")
301 print(f" - 综合判断: {'存在多光束干涉' if conditions['all_conditions_met'] else '不存在多光束干涉'}")

302
303 print("\n2. 硅外延层分析结果:")
304 if optimal_params_si is not None and mean_combined_si is not None:
305     print(f" - 硅外延层厚度: {mean_combined_si:.4f} ± {std_combined_si:.4f} m")
306     print(f" - 数据来源: 附件3和附件4")
307 else:
308     print(f" - 硅外延层分析未完成")

309
310 print(f"\n{'='*70}")
311 print("分析完成! 所有结果图片已保存至 {OUTPUT_DIR} 目录。")
312 print(f"{'='*70}")

313
314
315 if __name__ == "__main__":
316     main()

```
