

基于傅里叶变换与全局优化的红外干涉光谱厚度求解模型

摘 要

碳化硅外延层厚度是决定第三代半导体器件性能的核心参数，红外干涉法作为无损测量技术，其精度依赖于对干涉物理过程的精准建模。本文针对碳化硅外延层厚度确定问题，提出了一种基于反射光干涉图样的数学模型，并设计求解算法，为制定科学的厚度测试标准提供支撑。

针对问题一，聚焦外延层与衬底界面单次反射、透射的双光束干涉场景，使用**机理分析法**，围绕几何光学分析，建立双光束干涉情境下碳化硅外延层厚度求解的数学模型；同时，考虑到折射率色散，引入 Sellmeier 公式、Drude 模型、柯西公式等多个模型，完善模型参数体系。

针对问题二，根据问题一建立的双光束干涉模型，外延层厚度 d 与相邻反射峰的波数差成严格的反比关系，本算法从附件的光谱数据中提取该周期特征。为解决实际测量中数据非均匀、信号截断等问题，模型引入直流分量移除、**pchip 插值**和**汉宁窗函数**应用等算法，增强了信号的纯净度与分析的准确性。接着通过**FFT**将光谱数据从波数域变换至光学厚度域，将厚度测量问题转化为寻找频谱主峰位置的问题，并采用**能量重心矫正法**对峰位进行精细修正，得到反射率模型厚度的初值。采用基于**LM 算法的非线性最小二乘**拟合，规避局部最优陷阱的同时缩短收敛时间。求得碳化硅外延层厚度为 $8.02 \pm 0.17 \mu m$ ，通过交叉验证（不同入射角结果偏差 $< 0.5\%$ ）、残差分析（残差均方根 $< 0.3\%$ ）验证了结果的可靠性。

针对问题三，进一步分析了多光束干涉的影响，推导出多光束干涉时反射光总振幅呈几何级数，得出多光束干涉的必要条件判据 $|q| = |r_{10}r_{12}|e^{-\sigma^{2d}} > \varepsilon$ ，即空气-外延层界面和外延层-衬底界面的反射足够大且材料的吸收作用弱。接着采用**假设法**，证明硅晶圆片发生了多光束干涉；建立多光束干涉厚度模型，并使用**分层求解算法**，将多种色散模型**分阶段混联**，得到硅晶圆片外延层厚度为 $3.58 \mu m$ ，消除多光束干涉影响后的碳化硅外延层厚度为 $7.937 \mu m$ 。

关键词：红外干涉法 LM 算法 傅里叶变换 多光束干涉 色散模型

一、问题重述

1.1 问题的背景

在半导体材料的发展进程中，碳化硅凭借耐高温、耐高压、导热性优异等突出性能，成为制备极端条件下工作器件的核心材料。因其制备成本高，对关键指标检测要求严苛，红外干涉法凭借操作便捷、无损等优势成为测量外延层厚度这一外延材料关键性能的指标常用测量手段。

如今通过红外干涉法，利用双光束干涉确定外延层的厚度，但因多光束干涉的影响，使得外延层厚度测量的准确率下降，而如何消除这一影响有待研究。

1.2 问题的相关信息

(1) 给出了红外干涉法的工作原理

(2) 给出了外延层折射率和红外光谱的波长等变量之间的相关性。

(3) 附件 1 和附件 2 给出对同一碳化硅晶圆片在不同角度下的测试结果：

附件 1 和附件 2 给出对同一碳化硅晶圆片，在入射角分别为 10° 和 15° 时测试得出的波数和光谱反射率。

附件 3 和附件 4 给出对同一硅晶圆片，在入射角分别为 10° 和 15° 时测试得出的波数和光谱反射率。

1.3 需解决的问题

根据题目中所给信息，本文将问题细化为以下五个问题，并建立数学模型进行分析和研究。

问题一：仅考虑双光束干涉，建立计算外延层厚度的数学模型。

问题二：为了求解外延层厚度，基于问题一中建立的数学模型，设计确定外延层厚度的算法，并将设计的算法应用于附件 1 和附件 2 中的碳化硅晶圆光谱实测数据，同时，对该算法性能进行评估。

问题三：(1) 为求解材料外延层厚度的精确值，需推导多光束干涉产生的必要条件和计算精度的可能影响。(2) 为求解硅晶圆片的外延层厚度，将问题 3 中的推导结果应用于附件 3 和附件 4 中的硅晶圆光谱实测数据，判断硅晶圆片上是否出现多光束干涉，并确定硅晶圆片精确的外延层厚度。(3) 为了确定碳化硅晶圆片外延层厚度的精确值，需消除多光束干涉的影响，重新计算问题二中的数据，求解出碳化硅晶圆片精确的外延层厚度。

二、问题分析

2.1 问题一的分析

问题一要求建立外延层与衬底界面仅发生单次反射和透射时的干涉模型，用于确定外延层厚度。核心在于分析双光束干涉的物理机制，需要考虑两束相干光的光程差、相位突变以及外延层折射率的变化特性。难点在于如何综合考虑几何光学路径与材料光学特性，建立反射率与厚度、波数、入射角等参数之间的数学关系，为厚度测量提供理论依据。

2.2 问题二的分析

问题二需要基于问题一的模型进行厚度求解和可靠性分析。关键问题是如何从实测干涉光谱中有效提取厚度信息，以及如何处理实际测量中的数据噪声和非均匀采样问题。由于外延层折射率并非常数，还需要解决参数估计中的不确定性问题。可靠性分析需要从多个角度评估计算结果的稳定性和准确性。

2.3 问题三的分析

问题三关注多光束干涉效应的影响，需要分析多光束干涉产生的物理条件及其对厚度测量精度的影响。重点在于判断给定材料参数和测量条件下是否会发生多光束干涉，以及这种干涉效应会如何改变干涉条纹的特性。如果需要考虑多光束干涉，还需要研究如何修正现有模型以适应多光束情形，并评估修正后模型的准确性。

2.4 问题求解流程图

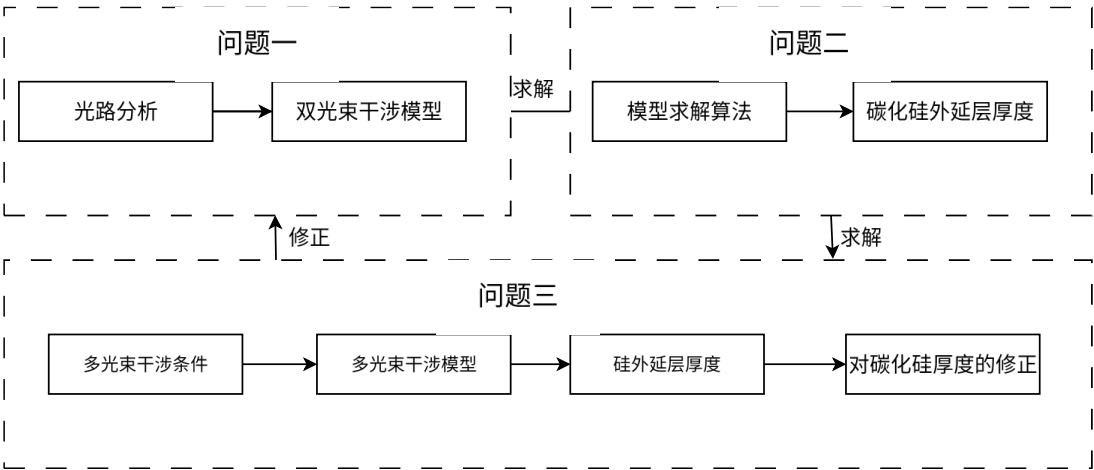


图 1 问题求解流程图

三、模型假设

- (1) 假设整个光学系统机械稳定性和环境稳定性较高。
- (2) 假设外延层和衬底材料内部载流子浓度一定且均匀。
- (3) 假设外延层为两个表面绝对平行、厚度均匀、平整的平面薄膜。
- (4) 假设入射光具有良好的相干性和单色性。
- (5) 忽略空气中的吸收作用对光线的影响题目明确给出的假设条件

四、符号说明

符号	说明	单位
ν	光谱波数，表示在单位长度（1 厘米）内所含完整	cm^{-1}

	波的个数	
$R(\nu)$	干涉光谱反射率，总反射能量与入射能量的比值	\
r_s, r_p	s 偏振和 p 偏振的反射系数	\
t_s, t_p	s 偏振和 p 偏振的透射系数	\
Δl	光程差，为两束光光程的差值	μm
$\Delta\phi(\nu)$	相位差，表示在任一给定时刻，两个波在各自周期中所处位置的差值	rad
I	光强，表示平均能流密度	W / m^2
E	振幅，光波中电场矢量振动的最大强度	V / m
n_0, n_1, n_2	空气、外延层、衬底的折射率	\
$\theta_i, \theta_r, \theta_t$	入射角，反射角与折射角	rad
d	外延层厚度	μm

五、背景介绍

5.1 外延层折射率模型

外延层的折射率决定了第一束透射光与第一束反射光的光程差，其大小不是常数，而是受到色散效应和载流子效应的影响，是一个由红外光谱的波长、掺杂载流子的浓度等参数确定的函数，描述折射率的模型如下：

5.1.1 吸收作用影响

碳化硅的外延层和衬底对红外光都发生吸收，对有吸收材料的折射率，用复数表示，以外延层折射率为例：

$$\tilde{n}_1(\lambda) = n_1(\lambda) + ik, \quad (1)$$

其中， $n_1(\lambda)$ 是实部（相速）， k 是消光系数（决定吸收程度的参数）。衬底同样存在明显的吸收效应，其折射率 n_2 也是复数。

5.1.2 本征折射率

对于未掺杂或轻掺杂载流子材料的色散效应，可直接根据 Sellmeier 公式得到晶体折射率，称为本征折射率，其表达式为：

$$n_1(\lambda)_{\text{本}}^2 = 1 + \sum \frac{A_i \lambda^2}{\lambda^2 - B_i} \quad (2)$$

其中 A_i 是与碳化硅的物理特性有关的常数， λ 是入射光线的波长。在红外波段通常取前三项，得到：

$$n_1(\lambda)_{\text{本}}^2 = 1 + \frac{A_1 \lambda^2}{\lambda^2 - B_1} + \frac{A_2 \lambda^2}{\lambda^2 - B_2} + \frac{A_3 \lambda^2}{\lambda^2 - B_3} \quad (3)$$

5.1.3 载流子对折射率影响

载流子浓度不同，是 SiC 外延层和衬底对红外光折射率不同的根本原因。根据电荷性质与材料类型，载流子主要分为电子与空穴两类，在金属中则以电子为主，在碳化硅晶体中二者则共同参与导电。载流子浓度对外延层浓度的影响满足 Drude 模型^[1]：

$$n_1^2(\omega) = n_\infty^2 \left(1 - \frac{\omega_p^2}{\omega^2 + i\gamma\omega} \right) \quad (4)$$

其中， n_∞ 是高频折射率（光频率趋于无穷时的折射率）， ω_p 是等离子频率， γ 是阻尼常数， ω 是入射光线角频率。由于载流子浓度 N 仅与材料制作时的掺杂工艺有关，在本题中不会改变，用 N 表示。

5.1.4 外延层折射率模型

最终的折射率可由 Sellmeier 公式（本征折射率）和 Drude 模型（掺杂载流子的影响）叠加得到：

$$\tilde{n}_1(\lambda) = n_1(\lambda)_\text{本} + \Delta n \quad (5)$$

5.2 碳化硅反射率异常原因解释

5.2.1 可视化分析

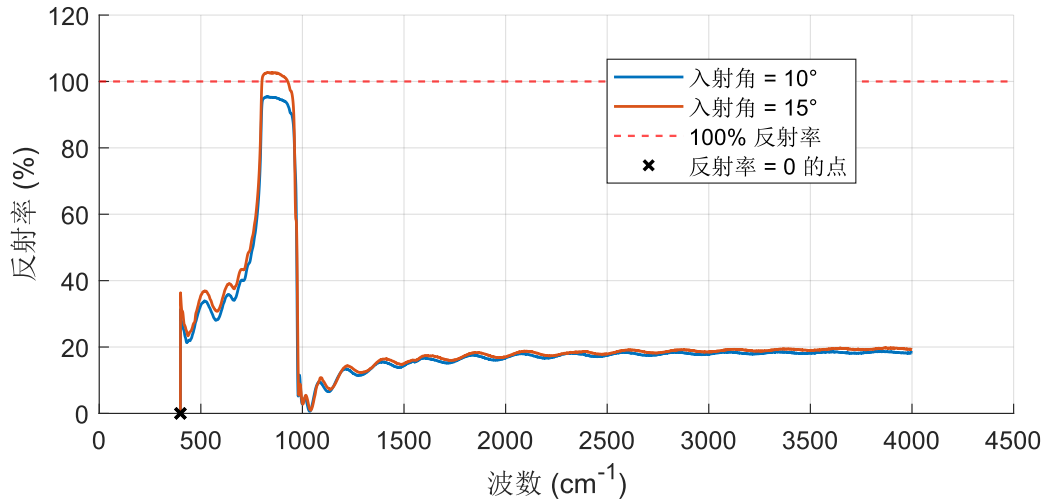


图 2 附件和附件 2 可视化分析

5.2.2 反射率异常凸起

观察图 2，发现入射角为 10° 和 15° 时反射率均在波数为 980cm^{-1} 附近区间出现了异常峰值，原因是因为样品中存在 Si-O 键^[2]。同时，在 $1000\sim 1300\text{cm}^{-1}$ 波数不存在反射峰，表明该样品中的 Si-O 键并非来自 SiO_2 。

5.2.3 反射率大于 100%

观察到附件 2 中出现了大于 100% 的反射率异常值。可能是由特征共振、多光束干涉和参考镜误差的原因多种原因构成。在特征峰处出现明显特征共振现象，导致此处反射率急剧上升，光谱图不再符合正常色散曲线。在 SiC 外延层结构中，光在外延层与衬底界面之间多次反射和透射，形成多光束干涉。在特定波数下，这些光束发生相长干涉，导致反射信号强度极大增强。同时，参考镜为简单反射镜，其反射率较高。

多种原因叠加使得反射信号强度大于参考镜反射光的强度。在本题计算过程中，事先假设为双光束干涉，不考虑多光束干涉这一导致反射率过大的主要原因的影响，故此处不对大于 100% 的数据做处理。在 6.3.6 小节中，将使用算法消除多光束干涉的影响，使反射率值恢复正常。

5.2.4 反射率为 0%

在附件一和附件二的数据中，波数为 399.67471cm^{-1} 反射率均为 0%，原因是绝大多数测量范围为中红外光（3~40mm）的傅里叶变换红外光谱仪，配备 DTGS 探测器和 KBr 分束器。KBr 在低于 400cm^{-1} 时对红外的透过率会急剧下降，无法有效工作^[3]，使得波数为 399.67471cm^{-1} 时，干涉光谱的反射率为 0%。故 0 值在本题数据处理中应当忽略。

六、模型的建立与求解

6.1 问题一：两束光线干涉模型的建立

6.1.1 模型准备

1. 光的干涉

光的干涉是指的是两列或多列光波在空间相遇时相互叠加，从而导致某些区域光强度增强、另一些区域光强度减弱，形成稳定分布的明暗相间条纹的现象。现给出光线发生干涉的条件：

(1) 只有相干光会发生干涉（频率相同）

(2) 如下图 3，相位差恒定的相干光会产生明暗相间的条纹。光程差 $\Delta = r_2 - r_1$ 与条纹亮度满足如下关系：当 $\delta = k\lambda$ ，即光程差为半波长的偶数倍，为明条纹；当 $\delta = (2k+1)\frac{\lambda}{2}$ ，即光程差为半波长的奇数倍，此时为暗条纹。条纹的明暗分布根据光程差的变化呈现出周期性的特征。

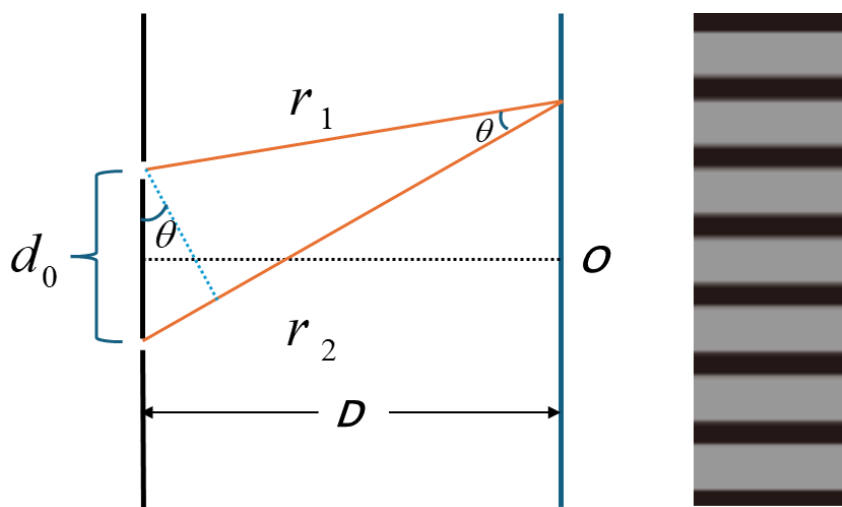


图 3 两束光的干涉示意图

2. 半波损失

当入射光线从光疏介质到光密介质时，反射波与入射波会在界面处叠加形成驻波，导致反射光比入射光多出一个 π 的相移，在本题中外延层对红外折射率大于空气折射率空气-外延层会产生半波损失，但外延层与衬底界面对红外光折射率大小关系未知，当外延层折射率小于衬底折射率，会产生第二个半波损失，与在空气-外延层界面产生的半波损失抵消，此时反射光 1 和反射光 2 的相位差仅由光程差决定；反之则两反射光线间会额外产生大小为 π 的相位差。

3. 光强与振幅的相关关系

光强本质上是电磁波的平均能流密度，由坡印廷矢量^[4]给出：

$$S = \frac{1}{\mu} E \times B \quad (6)$$

对简谐波能量取时间平均得：

$$I = \frac{1}{2} \rho A^2 \omega^2 u \propto A^2 \quad (7)$$

光强与电场振幅平方成正比，其中 u 为能量密度， ω 为振动角频率，该结论表明能量密度与振动幅度的平方正相关，即可通过振动幅度之比代替光强之比，即

$$R = \frac{I_{\text{出}}}{I_{\lambda}} = \left(\frac{E_{\text{出}}}{E_{\lambda}} \right)^2 \quad (8)$$

4. 光路分析

如图 4，当红外光入射到空气-外延层界面时，一部分光线发生镜面反射（称为反射光 1），反射角等于入射角，又由于空气折射率小于外延层，发生了半波损失，使反射光 1 的实际震动相位产生了偏移；另一部分进入外延层，在外延层中的传播方向满足斯涅尔定律（折射定律），到达外延层-衬底界面后再次发生反射和折射，最后经过一次折射从外延层界面返回空气（称为反射光 2），由于外延层与衬底对红外光的折射率大小相对关系未知，无法判断是否发生半波损失，后续需要分类讨论。由于两束光路径不同产生了光程差，满足干涉条件，因此会在外延层上表面观查到干涉图样。

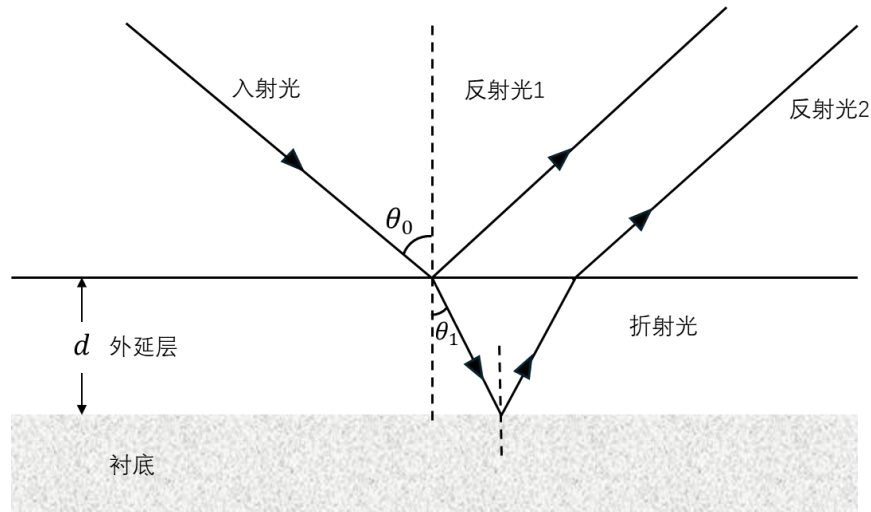


图 4 光线入射示意图

6. 1. 2 外延层厚度模型

1. 折射角与入射角的关系：

光在界面发生折射时遵守 Snell 定律：

$$n_0 \sin \theta_0 = \tilde{n}_1(\lambda) \sin \theta_1 \quad (9)$$

其中， θ_0 是入射角， θ_1 是折射角， n_0 是光在空气中的折射率，通常取 1。 \tilde{n}_1 是外延层折射率，与入射光波长有关。

2. Fresnel 反射公式：

光在介质界面上会部分反射、部分透射。电场的反射和透射比值由菲涅尔公式给出。红外光线是非偏振光，即有不同振动方向，其在界面上的折射可分解为两个方向： s 偏振光（电场垂直于入射面）和 p 偏振光（电场平行于入射面）。两种偏振情况得到的 Fresnel 反射系数分别如下：

$$r_s = \frac{n_0 \cos \theta_0 - \tilde{n}_1(\lambda) \cos \theta_1}{n_0 \cos \theta_0 + \tilde{n}_1(\lambda) \cos \theta_1}, \quad t_s = \frac{2n_0 \cos \theta_0}{n_0 \cos \theta_0 + \tilde{n}_1(\lambda) \cos \theta_1} \quad (10)$$

$$r_p = \frac{\tilde{n}_1(\lambda) \cos \theta_0 - n_0 \cos \theta_1}{\tilde{n}_1(\lambda) \cos \theta_0 + n_0 \cos \theta_1}, \quad t_p = \frac{2n_0 \cos \theta_0}{\tilde{n}_1(\lambda) \cos \theta_0 + n_0 \cos \theta_1} \quad (11)$$

式中分别为反射和透射的振幅比，含义为反射波（或透射波）的电场振幅相对于入射波电场的比例。

3. Fresnel 系数与光强的关系：

光的强度（功率流密度）与电场振幅平方成正比。因此，反射率和透射率可由振幅系数得到：

$$R_s = |r_s|^2, R_p = |r_p|^2 \quad (12)$$

非偏振光的反射率为二者均值：

$$R = \frac{1}{2}(R_s + R_p) \quad (13)$$

4. 反射率的计算

反射光 1 振幅：

$$E_1 = r_{01} E_0 \quad (14)$$

反射光 2 振幅：

$$E_2 = t_{01} r_{12} t_{10} e^{i\Delta\phi(\tilde{\nu})} E_0 \quad (15)$$

振幅反射系数：

$$r_{total}(\tilde{\nu}) = \frac{E_1 + E_2}{E_0} \quad (16)$$

总反射率：

$$R(\tilde{\nu}) = |r_{total}(\tilde{\nu})|^2 \quad (17)$$

将式（16）带入得：

$$R(\tilde{\nu}) = |r_{01} + t_{01} r_{12} t_{10} e^{i\phi(\tilde{\nu})}|^2 \quad (18)$$

5. 碳化硅厚度

第二束光的额外光程：

$$\Delta = 2\tilde{n}_1(\lambda) d \cos \theta_1 + \Delta_{\text{附加}} \quad (19)$$

相位差：

$$\Delta\phi(\tilde{\nu}) = \frac{2\pi}{\lambda} \cdot [2\tilde{n}_1(\lambda)d \cos \theta_1 + \Delta_{\text{附加}}] = 2\pi\tilde{\nu}[\tilde{n}_1(\lambda)d \cos \theta_1 + \Delta_{\text{附加}}] \quad (20)$$

其中 $\tilde{\nu}$ 为波数，含义为波长的倒数。相位差随波数发生周期性振荡，形成周期性震荡的反射率分布。由光的干涉知识可知，相邻两个干涉极小（或极大）值处的波数差 $\Delta\tilde{\nu}$ 对应相位差变化 2π ，即：

$$2\pi\tilde{\nu}[\tilde{n}_1(\lambda)d \cos \theta_1 + \Delta_{\text{附加}}] = 2\pi \quad (21)$$

碳化硅厚度的数学模型：

$$d = \frac{1}{2\sqrt{n_1(\lambda)^2 - \sin^2 \theta_0} \Delta\tilde{\nu}} \quad (22)$$

得出外延层厚度 d 关于外延层折射率 $\tilde{n}_1(\lambda)$ 、相邻峰值波数差 $\Delta\tilde{\nu}$ 、空气-外延层界面折射角 θ_1 的非线性方程。

6.2 问题二：求解外延层厚度算法的设计

6.2.1 厚度求解模型与 LM 算法

第一问得到外延层厚度求解模型：

$$\begin{cases} d = \frac{1}{2\sqrt{n_1(\lambda)^2 - \sin^2 \theta_0} \Delta\tilde{\nu}} \\ \lambda = \frac{1}{\tilde{\nu}} \\ n_1(\lambda)^2 = 1 + \frac{A_1\lambda^2}{\lambda^2 - B_1} + \frac{A_2\lambda^2}{\lambda^2 - B_2} + \frac{A_3\lambda^2}{\lambda^2 - B_3} \\ \Delta\phi(\tilde{\nu}) = 2\pi\tilde{\nu}[\tilde{n}_1(\lambda)d \cos \theta_1 + \Delta_{\text{附加}}] \\ R(\tilde{\nu}) = |r_{01} + t_{01}r_{12}t_{10}e^{i\phi(\tilde{\nu})}|^2 \end{cases} \quad (23)$$

问题一所得双光束干涉模型，包含了外延层厚度 d 、外延层折射率 $\tilde{n}_1(\lambda)$ 、衬底折射率 n_2 、外延层折射率色散参数(即塞尔迈耶尔参数) $B_1, B_2, B_3, C_1, C_2, C_3$ 等 8 个待求参数，数量较多且参数间存在强耦合。

而列文伯格-马夸特 (LM) 算法是最小二乘法的一种，本质是“高斯-牛顿法”与“梯度下降法”的混合优化算法。当参数迭代接近最优解时，算法偏向高斯-牛顿法（利用目标函数的二阶导数信息），可精准处理参数间的耦合关系，避免因参数关联导致的局部极小值；当迭代初期参数偏离最优解时，算法偏向最速下降法，确保参数更新方向符合“残差最小化”目标。这种特性恰好适配题目多参数耦合的模型需求，能有效降低“厚度与折射率色散参数相互干扰”导致的拟合误差，提升厚度 d 计算精度。

6.2.2 外延层厚度求解算法

表 1 算法步骤

Algorithm: 外延层厚度求解算法
Step1.用 FFT 分析周期性
Step2.厚度初值估算
Step3.全局参数拟合
Step4.结果验证与分析

Step1.干涉条纹的周期特征提取

根据问题一建立的双光束干涉模型，外延层厚度 d 与相邻反射峰的波数差成严格的反比关系，为精准提取干涉条纹周期（该周期是计算外延层厚度的关键依据），采用 MATLAB 软件（源程序见附录 2）通过频域信号优化方法开展数据预处理：

首先，考虑到原始反射率数据的波数网格存在非均匀性，这种非均匀性会影响后续频域分析的准确性，因此采用 `pchip` 插值法将反射率信号重新映射到均匀波数网格上，以此消除非均匀采样的干扰，进一步提升 FFT（快速傅里叶变换）的频率分辨率，让频域信号的细节更清晰；

其次，为避免信号截断过程中产生的频谱泄漏问题（该问题会导致频域出现虚假信号，干扰条纹周期识别），对插值后的均匀网格信号乘以汉宁窗——汉宁窗具有平滑的过渡特性，能有效抑制频谱泄漏，减少虚假信号对真实条纹周期的影响。

通过上述两步优化，可显著增强频域内“干涉条纹周期与外延层厚度”关联关系的辨识度，使条纹周期对应的频域峰值特征更突出，为后续准确提取周期参数、保障碳化硅外延层厚度计算精度提供有力支持。

Step.2 计算厚度初值

通过对预处理后的干涉光谱数据进行快速傅里叶变换，可将其从波数域转换到厚度域。设离散傅里叶变换后的频域索引为 m ，波数采样间隔为 $\Delta\tilde{\nu}$ ，则对应的厚度坐标可表示为：

$$d_m = \frac{1}{2 \cdot \sqrt{n_{\text{eff}}^2 - \sin^2 \theta_0} \cdot \Delta\tilde{\nu} \cdot m} \quad (24)$$

其中 n_{eff} 为外延层的有效折射率， θ_0 为入射角。计算功率谱密度 $P(m) = |X(m)|^2$ ，其中 $X(m)$ 为傅里叶系数。根据碳化硅外延层制备工艺及物理特性，其厚度通常分布于 5~200 μm 范围内。据此，在该厚度区间内搜索功率谱的全局极大值，初步确定干涉条纹的主周期对应的厚度初值：

$$d_0 = \arg \max_{d_m \in [5, 200]} P(m) \quad (25)$$

为提高峰值定位精度，采用能量重心校正法^[5]对离散频谱进行插值修正，抑制因频谱泄漏造成的估计偏差。经计算，获得厚度初值为 $d_0=7.65\mu\text{m}$ 。该值将作为后续非线性优化算法的初始迭代点，用于精确反演外延层厚度及折射率参数。

Step.3 全局参数拟合

为通过实测数据反演关键参数并验证模型有效性，采用 LM 算法开展模型参数拟合，设实验测得的反射率为 $R_{\text{exp}}(\tilde{\nu})$ ，理论模型计算的反射率为 $R_{\text{model}}(\tilde{\nu}, \mathbf{x})$ ，其中 \mathbf{x} 表示待拟合参数向量，则优化目标函数可表示为：

$$\min X^2 = \sum_{j=1}^N [R_{\text{exp}}(\tilde{\nu}) - R_{\text{model}}(\tilde{\nu}, \mathbf{x})]^2 \quad (26)$$

考虑到 LM 算法的收敛效率与结果可靠性易受初始参数影响，为缩短收敛时间、规避局部最小值陷阱，通过查阅文献中同类型碳化硅红外光学参数的研究结论，确定待拟合参数 $n_1(\lambda)$ 、 n_2 、 B_1 、 B_2 、 B_3 、 C_1 、 C_2 、 C_3 的合理初始值^[6]。最终基于上述设置完成曲线拟合，图 5 呈现了拟合得出的折射率实部 n 和虚部 k 结果，为后续外延层厚度计算提供了准确的光学参数输入。

结合问题一建立的双光束干涉模型，以“实测反射率与模型反射率残差平方和”为目标函数，使用 LM 算法进行曲线拟合。为了缩短收敛时间，同时规避局部最小值，

通过查阅文献^[4]，得出参数 $n_1(\lambda)$ 、 n_2 、 B_1 、 B_2 、 B_3 、 C_1 、 C_2 、 C_3 的合理初始值。

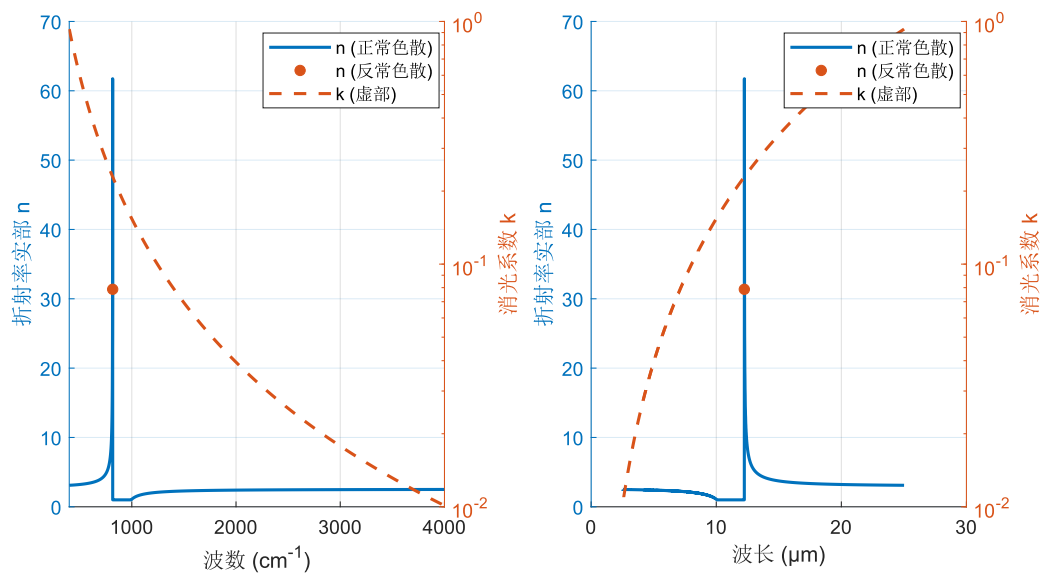


图 5 折射率拟合结果

参考碳化硅物理特性设置参数上下界为初始值的 0.8-1.5 倍，用信赖域反射算法对结果优化，通过评估模型与附件 1 和附件 2 实测数据的吻合度。拟合结果如表 2，在 10°和 15°入射时拟合优度均高达 0.9669,最终得出外延层厚度为 $8.02 \pm 0.17 \mu m$ 。

表 2 参数拟合结果

入射角度	$n_1(\lambda)$	n_2	$d / \mu m$	
10°	2.342+0.0564	2.523	8.14	0.9669
15°	2.390+0.6810	2.462	7.90	0.9669

Step4.一致性检验

针对附件 1 和附件 2 中同一碳化硅晶圆片的两个入射角数据，计算 10°和 15°下外延层厚度的相对差异，绘制出实测反射率和拟合反射率的对比曲线图，得到结果如下：。

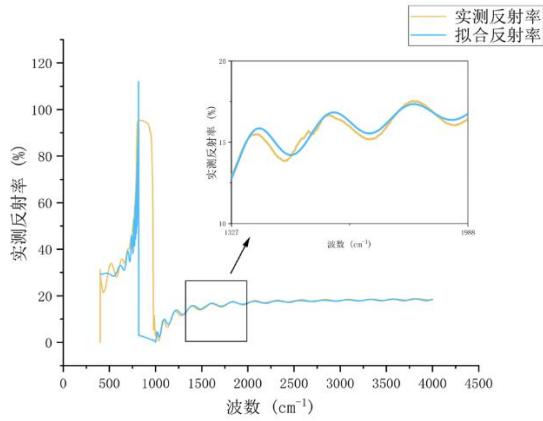


图 6 入射角为 10° 时的对比曲线

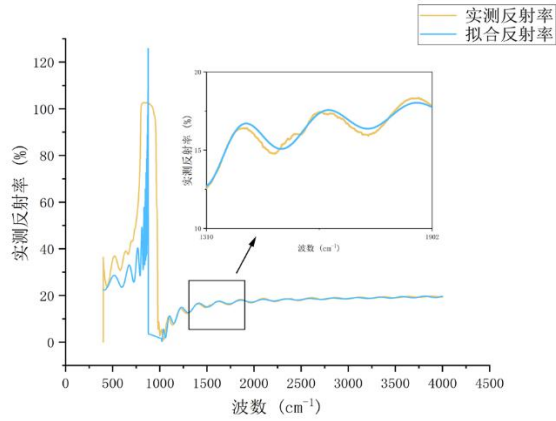


图 7 入射角为 15° 时的对比曲线

分析关系对比曲线图，可以看出在波数大于 1000 区段，拟合结果十分稳定。由相对差异公式 $RelDiff = \frac{|d_1 - d_2|}{\text{mean}([d_1, d_2])} \times 100\%$ ，得到 $RelDiff = 2.98\% < 5\%$ ，说明得出的外延层厚度结果不受入射角影响，可认为该方法在入射角变化时具有良好的鲁棒性。

6.3 问题三：多光束干涉模型的建立与求解

6.3.1 双光束干涉模型

由问题一所得双光束干涉模型可知，在只考虑一次折射和反射的情形下，产生的相位差：

$$\Delta\phi(\tilde{\nu}) = 2\pi\tilde{\nu}[\tilde{n}_1(\lambda)d \cos\theta_1 + \Delta_{\text{附加}}] \quad (27)$$

振幅反射系数：

$$R(\tilde{\nu}) = |r_{01} + t_{01}r_{12}t_{10}e^{i\phi(\tilde{\nu})}|^2 \quad (28)$$

外延层厚度：

$$d = \frac{1}{2\sqrt{n_1(\lambda)^2 - \sin^2\theta_0}\Delta\tilde{\nu}} \quad (29)$$

6.3.2 多光束干涉产生的必要条件

考虑外延层和衬底间多次折射和反射时，光路如图 8 所示（未画出每次进入衬底的光线）

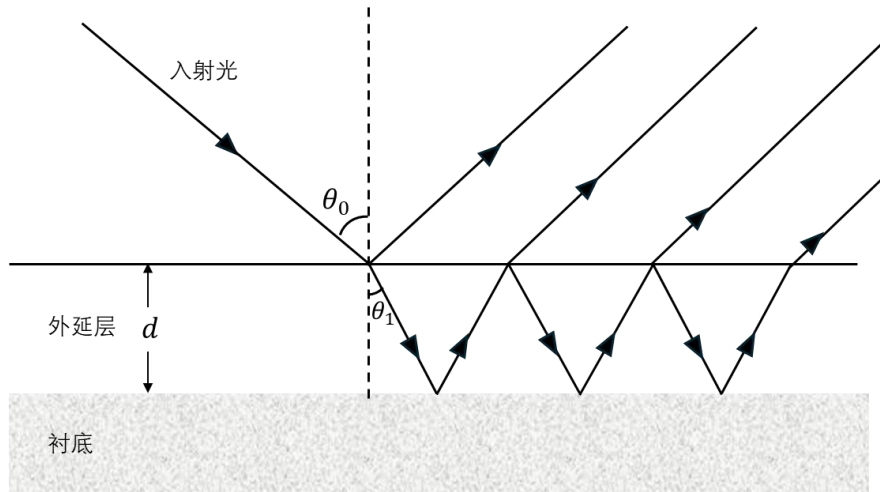


图 8 多光束干涉示意图

观察到每一次的光路可由第一次平移后得到，因此每一次的折射角和反射角也与第一次相同，又因为在小角度（本题为 10° 和 15° ）入射时，由菲涅尔反射计算出的 s 偏振和 p 偏振方向上的反射系数几乎一致，因此在多光束干涉模型中可忽略偏振现象，菲涅尔系数可表示为：

$$r = \frac{n_2 \cos\theta_1 - n_1 \cos\theta_2}{n_2 \cos\theta_1 + n_1 \cos\theta_2}, \quad t = \frac{2n_1 \cos\theta_1}{n_1 \cos\theta_1 + n_2 \cos\theta_2} \quad (30)$$

其中分别为入射界面和折射界面的折射率，分别为入射角和反射角的余弦值，在多光

束干涉中，这些参数均未发生改变，因此每一次反射和折射后，光的振幅以相同的比例系数发生损失。

考虑薄膜吸收（复折射率 $\tilde{n}_1(\lambda) = n_1(\lambda) + ik$ ），每次往返会附带幅度衰减 $P = e^{-\sigma 2d}$ ，其中

$$\alpha = \frac{4\pi k}{\lambda} \quad (31)$$

把所有返回到空气方向的反射项按次序叠加，得到公比 $q = r_{10}r_{12}P$ 的几何级数：

$$E_r = r_{01}E_0 + t_{01}r_{12}t_{10}PE_0 + t_{01}r_{12}r_{10}r_{12}t_{10}P^2E_0 + \dots \quad (32)$$

若要多光束干涉，即该级数的公比 q 足够大，此时振幅级数两项以后的余项无法忽略，光束会发生两次以上的反射和折射。

发生多光束干涉判据可表示为：

$$|q| = |r_{10}r_{12}|e^{-\sigma 2d} > \varepsilon \quad (33)$$

ε 通常取 0.01，该判据的物理意义是： $|r_{10}r_{12}|$ 大则两个界面**反射率高**，内部多次反射被放大； $e^{-\sigma 2d}$ 大表明**薄膜吸收弱**或厚度不大，往返损失小。其中 $|r_{10}r_{12}|$ 大小占主导因素，即必要条件主要考虑界面**反射率足够大**。

6.3.3 精度影响分析

对等比级数求和整理得：

$$r_{\text{tot}}(\tilde{\nu}) = r_{01} + \frac{t_{01}t_{10}r_{12}P(\tilde{\nu})}{1 - q(\tilde{\nu})} \quad (34)$$

转换为更常见的等价形式（艾里公式^[7]）为：

$$r_{\text{tot}}(\tilde{\nu}) = \frac{r_{01} + r_{12}P(\tilde{\nu})}{1 + r_{01}r_{12}P(\tilde{\nu})} \quad (35)$$

反射率：

$$R = |r_{\text{tot}}(\tilde{\nu})|^2 \quad (36)$$

双光束干涉时取级数前两项：

$$r(\tilde{\nu}) = r_{01} + t_{01}t_{10}r_{12}P \quad (37)$$

二者相对误差差为：

$$\Delta r \equiv r_{\text{tot}} - r = t_{01}t_{10}r_{12}P \cdot \frac{q}{1 - q} \quad (38)$$

当界面反射率较大时，分母的几何级数收敛慢，导致条纹展宽、非正弦化，会导致条纹的峰谷位置出现相位偏移，于是多光束对厚度的精度影响可表示为：

$$d' = d + \frac{\Delta\phi\lambda}{4\pi n_1 \cos\theta_1} \quad (39)$$

6.3.4 硅晶圆片的测试结果分析

假设硅晶圆片上存在多光束干涉现象，直接引入修正后的反射系数：

$$r_{\text{tot}}(\tilde{\nu}) = \frac{r_{01} + r_{12}P(\tilde{\nu})}{1 + r_{01}r_{12}P(\tilde{\nu})} \quad (40)$$

使用问题二中的 LM 算法，利用附件 3、附件 4 进行拟合，做出 q 关于波数的变化趋势，得到公比 q 结果在红外线对应的波数范围内（400~4000 cm^{-1} ），均有 $q > 0.01$ ，满足判据 $|q| = |r_{10}r_{12}|e^{-\sigma^2 d} > \varepsilon$ ，假设成立。

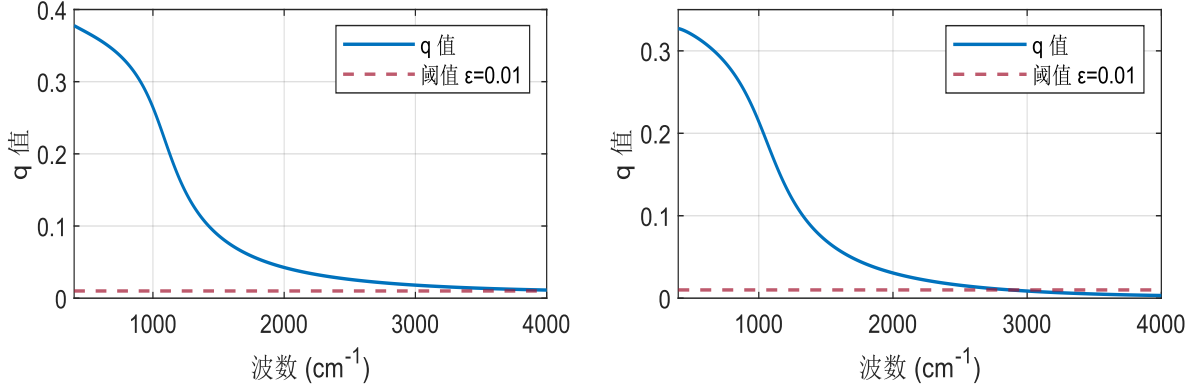


图 9 入射角为 10° 和 15° 时 q 随波数变化趋势 (Si)

6.3.5 硅外延层厚度模型的建立与求解

1. 模型的建立

在碳化硅厚度模型基础上引入多光束干涉反射率即可得到硅外延层厚度模型：

$$\begin{cases} d = \frac{1}{2\sqrt{n_1(\lambda)^2 - \sin^2 \theta_0 \Delta \tilde{\nu}}} \\ \lambda = \frac{1}{\tilde{\nu}} \\ n_1(\lambda)^2 = 1 + \frac{A_1 \lambda^2}{\lambda^2 - B_1} + \frac{A_2 \lambda^2}{\lambda^2 - B_2} + \frac{A_3 \lambda^2}{\lambda^2 - B_3} \\ \Delta \phi(\tilde{\nu}) = 2\pi \tilde{\nu} [\tilde{n}_1(\lambda) d \cos \theta_1 + \Delta_{\text{附加}}] \\ R(\tilde{\nu})' = \left| \frac{r_{01} + r_{12}P(\tilde{\nu})}{1 + r_{01}r_{12}P(\tilde{\nu})} \right|^2 \end{cases} \quad (41)$$

2. 模型的求解

该模型参数量较多，且不同参数间数量级差距较大，使用单一算法求解会产生较大的误差，且收敛所需时间长，难度大。

为求得精确解，本文采用**分层求解算法**，针对材料不同架构的物理性质，将多种色散模型**分阶段混联**，求解该规划问题。步骤如下：

表 3 求解步骤

Algorithm: 硅外延层厚度求解算法

Step1. 傅里叶变换

Step2. 柯西色散模型

Step3. Sellmeier 模型

Step4.模型比较

Step1: 傅里叶变换

使用傅里叶变换算法，利用反射光谱的周期性，**粗略估算**外延层厚度。

Step2: 柯西模型

由于色散模型参数量大且数量级小，其变化相对于其他参数不明显，故将求出**外延层和衬底的复折射率**为首要目标。所以在粗优化时放弃精确的色散模型，选用**柯西色散模型**求解粗优化外延层厚度：

$$n = A + \frac{B}{\lambda^2} \quad (42)$$

其中，A 和 B 是拟合参数。利用第二问中 **LM 算法**求得厚度的**粗优化**结果。

Step3: Sellmeier 色散模型

以柯西模型拟合结果作为约束条件，使用物理意义更强的 **Sellmeier 色散模型**为外延层色散模型，结合 **Drude 色散模型**为衬底的色散模型。在较紧的边界条件下进行全局非线性优化，得到厚度及光学常数的精确估计。

利用 Matlab 软件（源程序见附录 2）求解，得到入射角分别为 10° 和 15° 时测试得出的硅晶圆片的外延层厚度，取均值得到硅晶圆片最终**外延层厚度 $3.48 \mu m$** 。

Step4.模型比较

Cauchy 模型与 Sellmeier 模型拟合结果的 R^2 均较高，说明两者都能够较好地解释实验光谱。但 Sellmeier 模型的 R^2 在个别情形下略低于 Cauchy 模型，原因是 Sellmeier 模型参数更多，容易出现过拟合与参数相关性问题的，因此在本题中选用柯西模型来求解硅晶圆片外延层厚度。

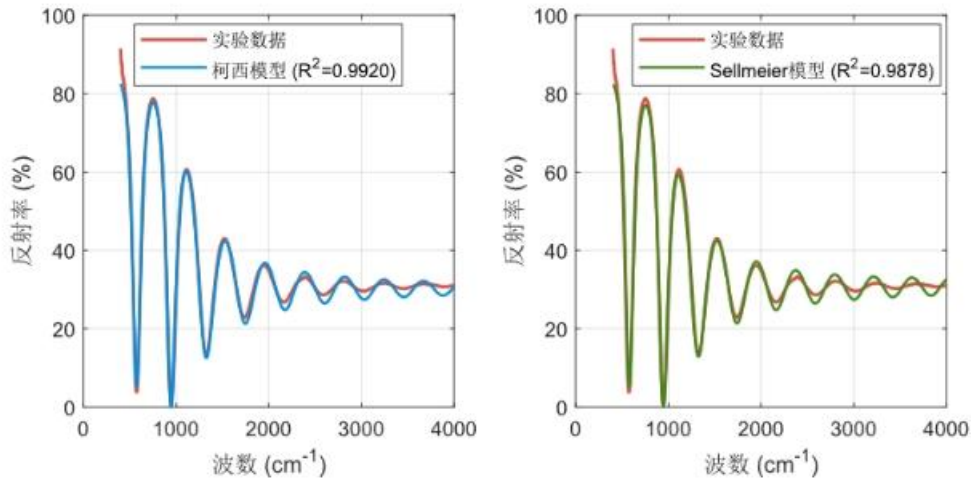


图 10 入射角为 10° 时粗优化与精优化结果对比

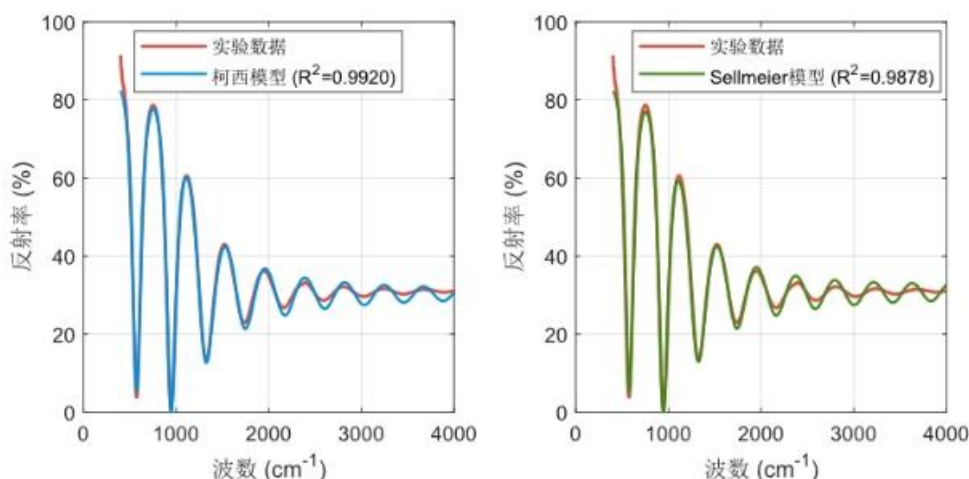


图 11 入射角度为 15° 时粗优化与精优化结果对比

6.3.6 消除多光束干涉影响的方法及评估

用 6.3.4 小节中的假设法判断碳化硅是否会受到多光束干涉的影响，得到结果如下，观察到绝大部分波数对应的公比 q 远小于 0.01，不会受到多光束干涉影响，但也注意到在入射角度为 15°，波数为 980 左右时出现了满足多光束干涉条件的 q 值，正好解释了 5.2.3 小节中提到的反射率异常值。后续需要通过多光束干涉模型在波数 980 cm^{-1} 附近对碳化硅外延层厚度结果进行修正。

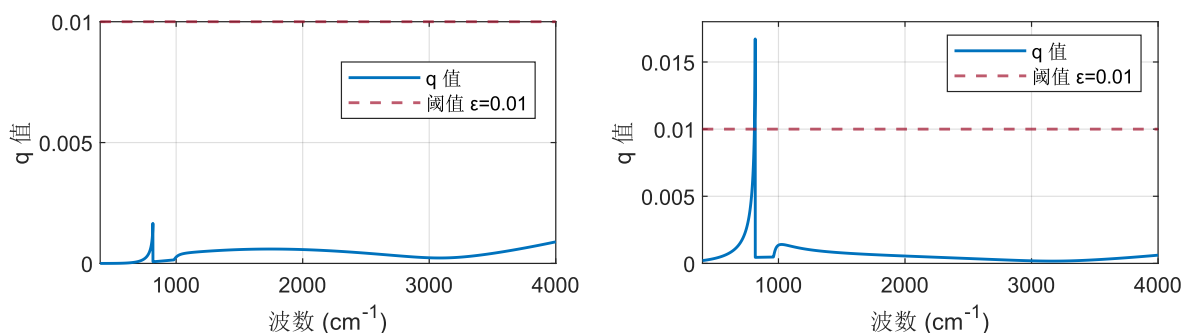


图 12 入射角为 10° 和 15° 时 q 随波数变化趋势 (Si)

在问题一所得硅外延层厚度数学模型和求解算法基础上引入艾里公式，即可得到考虑了多光束干涉的厚度模型。修正后的反射系数如下：

$$r_{\text{tot}}(\tilde{\nu}) = \frac{r_{01} + r_{12}P(\tilde{\nu})}{1 + r_{01}r_{12}P(\tilde{\nu})} \quad (43)$$

利用 Matlab 软件（源程序见附录 4）求解，得到结果如下表，最终求得外延层厚度为 7.937 μm 。

表 4 消除多光束干涉的影响

入射角 (°)	拟合优度 R^2	外延层厚度 (μm)	平均外延层厚度 (μm)
10	0.9442	8.062	7.937
15	0.9538	7.812	

拟合曲线与原始数据曲线如图 13、图 14 所示，在考虑多光束干涉的情况下，在正常色散区内，拟合结果均优于双光束干涉；由图 13 可知，在入射角为 15° 时，在反常色散区内，消除多光束干涉影响后，拟合反射率大幅下降且小于 100%，说明多光束干涉是造成测量反射率大于 100% 的重要原因，消除多光束干扰具有对反射率的修正作用。

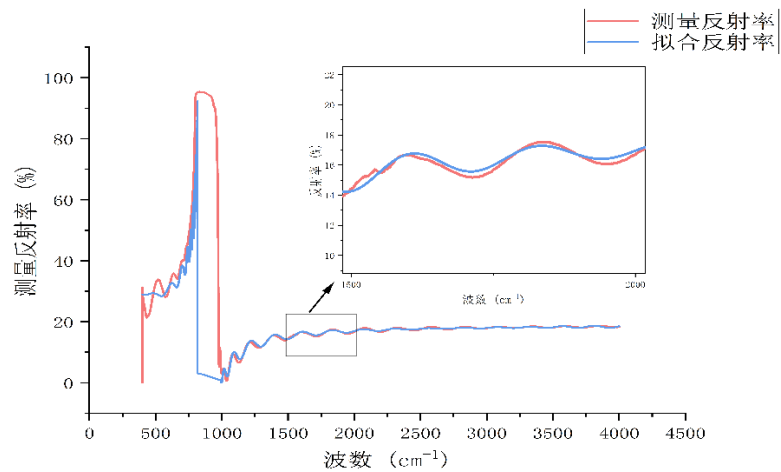


图 13 入射角为 10° 时拟合曲线与原始数据对比图

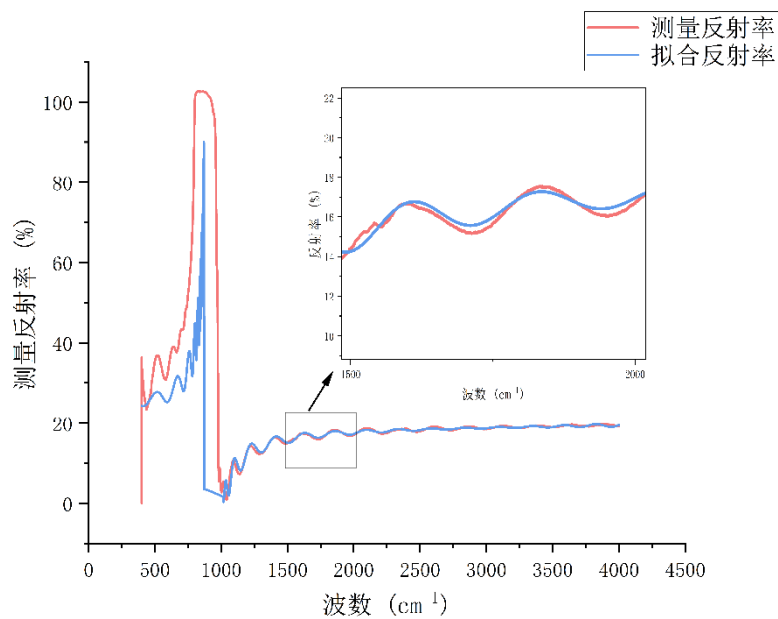


图 14 入射角为 15° 时拟合曲线与原始数据对比图

七、模型的评价、改进与推广

7.1 模型的优点

1. 本文为解决碳化硅厚度问题所设计的“FFT 初值+全局拟合”算法，在问题三中被成功应用于物理特性差异显著的硅晶圆片上，并同样取得了理想的结果。这充分验证了该算法具有良好的通用性和鲁棒性，不局限于特定材料。

2. 针对材料不同架构的物理性质，设计色散模型**分阶段混联算法**，以达到对模型更好的拟合效果，确保每一步模型选择和参数优化的合理性，最终得到的结果是经过多重比较验证的最优解。

3. 面对多参数模型，LM 算法有效规避了局部最优解的陷阱，在保证收敛速度的同时，显著提高了多维参数拟合的稳定性和准确性。

4. 针对不同材料（外延层、衬底）的光学特性差异，设计了色散模型分阶段混合求解算法。该算法能够通过对比分析，为不同层级智能选择最适配的物理模型（如 Sellmeier、Cauchy 或 Drude 模型），实现了对复杂材料体系的精确表征，显著提升了拟合精度。

7.2 模型的改进

当前模型取得了良好的效果，但仍存在以下可供探索和优化的方向：

1. 在使用红外干涉法测量外延层厚度时，材料会在特征峰附近发生**反常色散**，此处波长规律不符合光谱的周期性干涉。应考虑材料自身理化性质，使用**晶粒取向确定原理**等，对 SiC 材料的光谱做进一步拟合，以在题目所给红外光范围内精确拟合。

2. 求解硅晶圆外延层厚度的**分阶段混联模型**的精优化结果不稳定，考虑放软边界或调整优化算法，使优化结果稳定。

7.3 模型的推广

设计了**分阶段混联模型**，能够应用于其他分层结构且层间有折射率差异的材料（如其他半导体、带涂层金属 / 陶瓷等）外延层厚度测量，也适用于不同入射角、红外波段的测量场景，还能助力工业检测标准制定与多参数同步测量。

八、参考文献

- [1] 蒋和伦,刘启能. Drude-Lorentz 色散模型研究光学 Tamm 态的性质[J]. 半导体光电,2016,37(2):218-222.
- [2] 马格林, 张玉明, 张义门, 马仲发. 一种碳化硅外延层质量评估新技术[J]. 西安电子科技大学学报, 2011, 38 (06): 37-43+96
- [3] 杨琨.傅里叶变换红外光谱仪若干核心技术研究及其应用[D].湖北武汉:武汉大学,2010.
- [4] DeepSeek, DeepSeek-V3, 深度求索 (DeepSeek) , 2024-07
- [5] 丁康,江利旗.离散频谱的能量重心校正法[J].振动工程学报,2001,(03):110-114.DOI:10.16385/j.cnki.issn.1004-4523.2001.03.022.
- [6] C. D. Salzberg and J. J. Villa. Infrared refractive indexes of silicon, germanium and modified selenium glass. *J. Opt. Soc. Am.*, **47**, 244-246 (1957)
- [7] 王东硕 .Airy 波束型静磁表面波的特性研究 [D]. 天津大学,2019.DOI:10.27356/d.cnki.gtjdu.2019.003021.

附录

附录 1		
介绍：支撑材料的文件列表		
文件夹名称	文件名	含义
\	AI 工具使用详情. pdf	记录使用 AI 过程
代码	SiC_2. m	Si 双干涉主函数，计算厚度并对比分析结果
	Si_mul. m	Si 多干涉主函数，计算厚度并对比分析结果
	SiC_mul. m	SiC 多干涉主函数，计算厚度并对比分析结果
	Si_adp. m	Si 自适应选择多/双干涉主函数，计算厚度并对比分析结果
	cal_n_cauchy. m	根据柯西色散方程计算材料的复折射率
	cal_n_drude. m	根据 <i>drude</i> 色散方程计算材料的复折射率
	cal_n_sellmeier. m	根据 <i>sellmeier</i> 色散方程计算材料的复折射率
	cal_R_db. m	计算双光束干涉的总反射率
	cal_R_mul. m	计算多光束干涉的总反射率
	check. m	计算 q ，判断是否发生多光束干涉
	compute_R. m	根据多/双干涉计算单层膜的反射率
	fft_thk_estimate. m	通过傅里叶变换估算薄膜厚度
	global_fit. m	根据指定的物理模型进行全局拟合函数
	plot_fit_res. m	绘制拟合结果与实测数据的对比图
	plot_optical_constants. m	绘制复折射率色散曲线
	vis. m	可视化碳化硅 (SiC) 晶圆片在不同入

		射角下的反射光谱
结果	15_Si_双干涉_R.xlsx	15 度 Si 双干涉 R 拟合结果
	Si_15_多干涉.xlsx	15 度 Si 多干涉拟合结果
	Si_10_多干涉.xlsx	10 度 Si 多干涉拟合结果
	10_Si_双干涉_R.xlsx	10 度 Si 双干涉 R 拟合结果
	15_Si_双干涉_R.xlsx	15 度 Si 双干涉 R 拟合结果
	SiC_10_多干涉.xlsx	10 度 SiC 多干涉拟合结果
	SiC_15_多干涉.xlsx	15 度 SiC 多干涉拟合结果
图片	10°_Si_多干涉_n1 (Sellmeier).svg	10° 入射角下，硅（Si）晶圆片在多光束干涉模型中，外延层（n1）的折射率曲线（使用 Sellmeier 模型计算）。
	10°_Si_多干涉_n2 (Drude).svg	10° 入射角下，硅（Si）晶圆片在多光束干涉模型中，衬底（n2）的折射率曲线（使用 Drude 模型计算）。
	10°_Si_多干涉_R.svg	10° 入射角下，硅（Si）晶圆片在多光束干涉模型下的反射率（R）曲线图。
	10°_Si_双干涉_R.svg	10° 入射角下，硅（Si）晶圆片在双光束干涉模型下的反射率（R）曲线图。
	10°_SiC_多干涉_n1 曲线.svg	10° 入射角下，碳化硅（SiC）晶圆片在多光束干涉模型中，外延层（n1）的折射率曲线。
	10°_SiC_双干涉_n1 曲线.svg	10° 入射角下，碳化硅（SiC）晶圆片在双光束干涉模型中，外延层（n1）的折射率曲线。
	15°_Si_多干涉_n1 (Cauchy).svg	15° 入射角下，硅（Si）晶圆片在多光束

		干涉模型中，外延层 (n1) 的折射率曲线(使用 Cauchy 模型计算)。
	<i>15°_Si_多干涉_R.svg</i>	15° 入射角下，硅 (Si) 晶圆片在多光束干涉模型下的反射率 (R) 曲线图。
	<i>15°_Si_双干涉_n1.svg</i>	15° 入射角下，硅 (Si) 晶圆片在双光束干涉模型中，外延层 (n1) 的折射率曲线。
	<i>15°_Si_双干涉_n2.svg</i>	15° 入射角下，硅 (Si) 晶圆片在双光束干涉模型中，衬底 (n2) 的折射率曲线。
	<i>15°_Si_双干涉_R.svg</i>	15° 入射角下，硅 (Si) 晶圆片在双光束干涉模型下的反射率 (R) 曲线图。
	<i>Si_10_q.svg</i>	10° 入射角下，硅 (Si) 晶圆片的参数 q 的曲线图。
	<i>Si_15_q.svg</i>	15° 入射角下，硅 (Si) 晶圆片的某个分析参数 q 的曲线图。
	<i>SiC_10_q.svg</i>	10° 入射角下，碳化硅 (SiC) 晶圆片的某个分析参数 q 的曲线图。
	<i>SiC_15_q.svg</i>	15° 入射角下，碳化硅 (SiC) 晶圆片的某个分析参数 q 的曲线图。

附录 2
介绍：傅里叶变换求厚度初解 (MATLAB)
<pre> FFT function thk = fft_thk_estimate(waveNum, R, n_avg, waveN_fit_min, theta0_deg) % FFT_THK_ESTIMATE 通过傅里叶变换估算薄膜厚度 % waveNum: 波数 (cm^-1) % R: 反射率 % n_avg: 外延层在该区域的平均折射率 % waveN_fit_min: 用于 FFT 的最小波数 </pre>

```

%   theta0_deg: 入射角 (度)

if nargin < 5, theta0_deg = 0; end
theta0_rad = theta0_deg * pi / 180;
cos_theta1 = real(sqrt(1 - (sin(theta0_rad) / n_avg)^2));

filter = waveNum > waveN_fit_min;
waveNum_fft = waveNum(filter);
R_fft = R(filter);

% 移除直流分量
R_ac = R_fft - mean(R_fft);

% 插值到均匀间隔的波数点
N = 2^nextpow2(8*length(waveNum_fft));
k_uniform = linspace(min(waveNum_fft), max(waveNum_fft), N);
r_uniform = interp1(waveNum_fft, R_ac, k_uniform, 'pchip', 'extrap');

% 加窗并执行FFT
window = hann(N);
r_win = r_uniform(:) .* window(:);
fft_result = fft(r_win);
fft_power = abs(fft_result(1:N/2)).^2;

% 计算厚度轴
dk = mean(diff(k_uniform));
thk_axis = (0:N/2-1) * 10000 / (2 * n_avg * cos_theta1 * N * dk);

% 寻找主峰
search_range = find(thk_axis > 5 & thk_axis < 200); % 在合理范围内搜索
if isempty(search_range)
    thk = 20;
    return;
end

[~, max_idx_in_range] = max(fft_power(search_range));
max_idx_rough = search_range(max_idx_in_range);

% 能量重心矫正法
correctNum = 3;
DatePower1 = 0;
DatePower2 = 0;
for i = -correctNum:correctNum
    idx = max_idx_rough + i;

```

```

        if idx >= 1 && idx <= length(fft_power)
            power = fft_power(idx);
            DatePower1 = DatePower1 + idx * power;
            DatePower2 = DatePower2 + power;
        end
    end

    if DatePower2 > 0
        f_corrected = DatePower1 / DatePower2;
    else
        f_corrected = max_idx_rough;
    end

    thk = (f_corrected - 1) * 10000 / (2 * n_avg * cos_theta1 * N * dk);
end

```

附录 3

介绍：用多光束干涉求硅晶圆片外延层厚度（MATLAB）

```

2. clc;
clear;
close all;

file1.path = '附件 3.xlsx';
file1.angle = 10;
file2.path = '附件 4.xlsx';
file2.angle = 15;

config.waveN_fit_min = 400;
config.n1_init = 3.48;
config.cauchyParam_init = [3.42, 0.05];
config.selParam_init = [10.6684293, 0.0030434748, 1.5413340, 0.301516485,
1.13475115, 1104];
config.k1Param_A_init = 1e-5;
config.k1Param_B_init = 2.0;
config.drudeParam_init = [2000, 200];
config.epsilon_inf = 11.7;

disp('一、FFT 计算厚度初值');
data1 = readmatrix(file1.path);

```



```

thk_init1 = fft_thk_estimate(data1(:,1), data1(:,2)/100, config.n1_init, 2000, file1.angle);
data2 = readmatrix(file2.path);
thk_init2 = fft_thk_estimate(data2(:,1), data2(:,2)/100, config.n1_init, 2000, file2.angle);
config.thk_init = mean([thk_init1, thk_init2]);
fprintf('10: %.2f  $\mu$  m\n', thk_init1);
fprintf('15: %.2f  $\mu$  m\n', thk_init2);
fprintf('平均厚度初值: %.2f  $\mu$  m\n', config.thk_init);

fprintf('\n 二、双光束干涉精细拟合\n')
fprintf('\n10:\n')
[res1] = process(data1, file1.angle, config, file1.path);
fprintf('\n15:\n')
[res2] = process(data2, file2.angle, config, file2.path);

analyze_res(res1, res2, file1.angle, file2.angle);

function [res] = process(data, angle, config, output_filepath)
    waveNum = data(:, 1);
    R = data(:, 2) / 100;
    waveLen_full = 10000 ./ waveNum;
    filter = waveNum > config.waveN_fit_min;
    R_fit = R(filter);
    waveLen_fit = waveLen_full(filter);

    %% 柯西模型拟合
    disp('柯西模型拟合');
    x0_cauchy = [config.thk_init, config.cauchyParam_init, config.k1Param_A_init,
config.k1Param_B_init, config.drudeParam_init];
    lb_cauchy = [config.thk_init*0.8, 3.35, 0, 0, 0, 100, 10];
    ub_cauchy = [config.thk_init*1.2, 3.6, 0.2, 1e-3, 4, 4000, 1000];

    [x_optimal_cauchy, R_squared_cauchy] = global_fit(x0_cauchy, lb_cauchy,
ub_cauchy, R_fit, waveLen_fit, angle, config.epsilon_inf, 'cauchy');

    cauchy_res.thk = x_optimal_cauchy(1);
    cauchy_res.cauchyParam = x_optimal_cauchy(2:3);
    cauchy_res.k1Param = x_optimal_cauchy(4:5);
    cauchy_res.drudeParam = x_optimal_cauchy(6:7);
    cauchy_res.R_squared = R_squared_cauchy;
    cauchy_res.model_type = 'Cauchy';

    cauchy_res.n1_complex = cal_n_cauchy(waveLen_full, cauchy_res.cauchyParam,
cauchy_res.k1Param);

```

```

    cauchy_res.n2_complex_full = cal_n_drude(waveNum, cauchy_res.drudeParam,
config.epsilon_inf);

    fprintf('    R^2: %.6f\n', R_squared_cauchy);
    fprintf('    厚度: %.2f  $\mu$  m\n', cauchy_res.thk);
    fprintf('    柯西参数 : A=%.3f, B=%.3f\n', cauchy_res.cauchyParam(1),
cauchy_res.cauchyParam(2));
    fprintf('    k1 参数 : A=%.2e, B=%.2f\n', cauchy_res.k1Param(1),
cauchy_res.k1Param(2));
    fprintf('    n2:Drude 参数:  $v_p$ =%.1f,  $\Gamma$ =%.1f\n', cauchy_res.drudeParam(1),
cauchy_res.drudeParam(2));

    %% sel 模型拟合
    disp('sel 模型拟合');

    thk_cauchy = x_optimal_cauchy(1);
    k1_cauchy = x_optimal_cauchy(4:5);
    drude_cauchy = x_optimal_cauchy(6:7);

    x0_sel = [thk_cauchy, config.selParam_init, k1_cauchy, drude_cauchy];

    thk_rf = 0.03;
    lb_thk = thk_cauchy * (1 - thk_rf);
    ub_thk = thk_cauchy * (1 + thk_rf);

    sel_rf = 0.2;
    lb_sel_params = config.selParam_init * (1 - sel_rf);
    ub_sel_params = config.selParam_init * (1 + sel_rf);

    k1_rf = 0.3;
    lb_k1 = k1_cauchy * (1 - k1_rf);
    ub_k1 = k1_cauchy * (1 + k1_rf);

    drude_rf = 0.1;
    lb_drude = drude_cauchy * (1 - drude_rf);
    ub_drude = drude_cauchy * (1 + drude_rf);

    lb_sel = [lb_thk, lb_sel_params, lb_k1, lb_drude];
    ub_sel = [ub_thk, ub_sel_params, ub_k1, ub_drude];

    lb_sel = max(lb_sel, [config.thk_init*0.7, 0.1, 0.001, 0.0001, 0.0001, 0.1, 1, 0, 0, 50,
5]);
    ub_sel = min(ub_sel, [config.thk_init*1.3, 20, 5, 10, 2, 15, 2000, 1e-3, 4, 5000,
1500]);

```

```

if lb_sel(8) < 0
    lb_sel(8) = 0;
end
if lb_sel(9) < 0
    lb_sel(9) = 0;
end

[x_optimal_sel, R_squared_sel] = global_fit(x0_sel, lb_sel, ub_sel, R_fit,
waveLen_fit, angle, config.epsilon_inf, 'sellmeier');

sel_res.thk = x_optimal_sel(1);
sel_res.selParam = x_optimal_sel(2:7);
sel_res.k1Param = x_optimal_sel(8:9);
sel_res.drudeParam = x_optimal_sel(10:11);
sel_res.R_squared = R_squared_sel;
sel_res.model_type = 'sel';

sel_res.n1_complex = cal_n_sellmeier(waveLen_full, sel_res.selParam,
sel_res.k1Param);
sel_res.n2_complex_full = cal_n_drude(waveNum, sel_res.drudeParam,
config.epsilon_inf);

...
fprintf(' R^2: %.6f\n', R_squared_sel);
fprintf(' 厚度: %.2f  $\mu$  m\n', sel_res.thk);
fprintf(' 赛尔迈耶参数: %.4f, %.4f, %.4f, %.4f, %.4f, %.1f\n',
sel_res.selParam(1), sel_res.selParam(2), sel_res.selParam(3), sel_res.selParam(4),
sel_res.selParam(5), sel_res.selParam(6));
fprintf(' n2 Drude 参数:  $v_p$  = %.1f,  $\Gamma$  = %.1f \n', sel_res.drudeParam(1),
sel_res.drudeParam(2));
...
%% 选择最优模型
disp('模型比较与选择');
fprintf(' 柯西模型 R^2 = %.6f\n', R_squared_cauchy);
fprintf(' sel 模型 R^2 = %.6f\n', R_squared_sel);

if R_squared_sel > R_squared_cauchy
    res = sel_res;
    res.selected_model = 'sel';
    fprintf(' 选择 sel 模型 (R^2 提升: %.6f)\n', R_squared_sel -
R_squared_cauchy);
else
    res = cauchy_res;

```

```

        res.selected_model = 'Cauchy';
        fprintf('    选择 柯西 模型 (R^2 更高或相等)\n');
    end

    % 保存两个模型的结果以供后续分析
    res.cauchy_res = cauchy_res;
    res.sel_res = sel_res;

    % 输出最终选择的模型参数
    fprintf('\n 最终选择: %s 模型\n', res.selected_model);
    fprintf('    最终厚度: %.2f  $\mu$  m\n', res.thk);
    fprintf('    最终 R^2 = %.6f\n', res.R_squared);

    if strcmp(res.selected_model, 'sel')
        fprintf('    外延层 n1 (在 6  $\mu$  m): %.3f + %.4fi\n', ...
            real(interp1(waveLen_full, res.n1_complex, 6)), ...
            imag(interp1(waveLen_full, res.n1_complex, 6)));
    else
        fprintf('    外延层柯西参数: A=%.3f, B=%.3f\n', res.cauchyParam(1),
res.cauchyParam(2));
        fprintf('    外延层 n1 (在 6  $\mu$  m): %.3f + %.4fi\n', ...
            real(interp1(waveLen_full, res.n1_complex, 6)), ...
            imag(interp1(waveLen_full, res.n1_complex, 6)));
    end
    fprintf('    k1 参数: A=%.2e, B=%.2f\n', res.k1Param(1), res.k1Param(2));
    fprintf('    Drude 参数:  $\nu_p$ =%.1f,  $\Gamma$ =%.1f \n', res.drudeParam(1),
res.drudeParam(2));

    % 保存结果到 Excel
    try
        theta0_rad = angle * pi / 180;
        R_fit_full = compute_R(res.thk, res.n1_complex, res.n2_complex_full,
waveNum, theta0_rad, 1);
        R_fit_per = R_fit_full * 100;
        writematrix(R_fit_per, output_filepath, 'Sheet', 1, 'Range', 'C2');
        fprintf('    成功将拟合结果保存到: %s (使用%s 模型)\n', output_filepath,
res.selected_model);
    catch ME
        fprintf('    保存 Excel 文件时出错: %s\n', ME.message);
    end

    figure;

    % 柯西

```

```

subplot(1,2,1);
R_fit_cauchy = compute_R(cauchy_res.thk, cauchy_res.n1_complex,
cauchy_res.n2_complex_full, waveNum, angle * pi / 180, 1);
plot(waveNum(2:end), R(2:end)*100, '-', 'Color', '#d74f44', 'LineWidth', 1.5,
'DisplayName', '实验数据');
hold on;
plot(waveNum, R_fit_cauchy*100, '-', 'Color', '#008ede', 'LineWidth', 1.3,
'DisplayName', sprintf('柯西模型 (R^2=%.4f)', cauchy_res.R_squared));
xlabel('波数 (cm^{-1})');
ylabel('反射率 (%)');
legend('Location', 'best');
grid on;

% sel
subplot(1,2,2);
R_fit_sel = compute_R(sel_res.thk, sel_res.n1_complex, sel_res.n2_complex_full,
waveNum, angle * pi / 180, 1);
plot(waveNum(2:end), R(2:end)*100, '-', 'Color', '#d74f44', 'LineWidth', 1.5,
'DisplayName', '实验数据');
hold on;
plot(waveNum, R_fit_sel*100, '-', 'Color', '#3f8819', 'LineWidth', 1.3, 'DisplayName',
sprintf('sel 模型 (R^2=%.4f)', sel_res.R_squared));
xlabel('波数 (cm^{-1})');
ylabel('反射率 (%)');
legend('Location', 'best');
grid on;

% 绘制最优模型的光学常数
plot_optical_constants(waveLen_full, waveNum, res.n1_complex, sprintf('外延层
(%)', res.selected_model));
plot_optical_constants(waveLen_full, waveNum, res.n2_complex_full, '衬底
(Drude)');
end

function R = cal_R_db(r01_s, r12_s, r01_p, r12_p, exp_term)
% CALCULATE_R_DOUBLE_BEAM 双光束干涉模型
r_s_total = r01_s + r12_s .* exp_term;
r_p_total = r01_p + r12_p .* exp_term;
R_s = abs(r_s_total).^2;
R_p = abs(r_p_total).^2;
R = real((R_s + R_p) / 2);
end

```

附录 4

介绍：用多光束干涉求碳化硅外延层厚度（MATLAB）

```
clc;
clear;
close all;

file1.path = '附件 1.xlsx';
file1.angle = 10; % 入射角 (度)
file2.path = '附件 2.xlsx';
file2.angle = 15; % 入射角 (度)

config.waveN_fit_min = 1500; % 拟合起始波数 (cm-1)
config.n1_init = 2.58; % 外延层折射率初值 (用于 FFT 估算厚度)
config.n2_real_init = 2.55; % 衬底折射率初值
config.k1_A_init = 0.001; % 外延层消光系数参数 A 初值
config.k1_B_init = 2.0; % 外延层消光系数参数 B 初值

% Sellmeier 模型 B1, B2, B3, C1, C2, C3 参数初值
config.selParam_init = [5.5, 0.2, 0.05, 0.027, 100, 0.01];

disp('FFT 计算厚度初值');
data1 = readmatrix(file1.path);
thk_init1 = fft_thk_estimate(data1(:,1), data1(:,2)/100, config.n1_init,
config.waveN_fit_min, file1.angle);
data2 = readmatrix(file2.path);
thk_init2 = fft_thk_estimate(data2(:,1), data2(:,2)/100, config.n1_init,
config.waveN_fit_min, file2.angle);
config.thk_init = mean([thk_init1, thk_init2]);
fprintf('文件 1 (10° ) FFT 估算厚度: %.2f μ m\n', thk_init1);
fprintf('文件 2 (15° ) FFT 估算厚度: %.2f μ m\n', thk_init2);
fprintf('平均厚度初值: %.2f μ m\n', config.thk_init);

disp(' ');
disp('10: ');
[res1] = process(data1, file1.angle, config, file1.path);
disp('15: ');
[res2] = process(data2, file2.angle, config, file2.path);

disp(' ');
```

```

disp('分析: ');
analyze_res(res1, res2, file1.angle, file2.angle);

%% 数据处理和拟合主函数
function [res] = process(data, angle, config, output_filepath)
    epsilon = 11.7;
    waveNum = data(:, 1); % 波数 (cm-1)
    R = data(:, 2) / 100; % 反射率 (0-1)
    waveLen_full = 10000 ./ waveNum; % 波长 ( $\mu m$ )

    filter = waveNum >= config.waveN_fit_min;
    waveNum_fit = waveNum(filter);
    waveLen_fit = 10000 ./ waveNum_fit;

    R_fit = R(filter);

    % 厚度, 衬底 n2, Sellmeier(6), k 参数(2)
    x0 = [config.thk_init, config.n2_real_init, config.selParam_init, config.k1_A_init,
config.k1_B_init];
    lb = [config.thk_init*0.8, 2.0, 0.01, 0.001, 0.0001, 0.0001, 0.1, 0.001, 0, 0];
    ub = [config.thk_init*1.2, 3.5, 20, 10, 5, 2, 150, 20, 0.01, 4];

    model_type = 'real_substrate';
    [x_optimal, R_squared_fit] = global_fit(x0, lb, ub, R_fit, waveLen_fit, angle,
epsilon, model_type);

    res.thk = x_optimal(1);
    res.n2 = x_optimal(2);
    res.selParam = x_optimal(3:8);
    res.k1Param = x_optimal(9:10);
    res.n1_complex_full = cal_n_sellmeier(waveLen_full, res.selParam, res.k1Param);

    fprintf('    最终厚度: %.2f  $\mu m$ \n', res.thk);
    fprintf('    最终外延层 n1 (在 6  $\mu m$ ): %.3f + %.4fi\n',
real(interp1(waveLen_full, res.n1_complex_full, 6)), imag(interp1(waveLen_full,
res.n1_complex_full, 6)));
    fprintf('    最终衬底 n2: %.3f\n', res.n2);
    fprintf('    在拟合区域的拟合优度 R?: %.4f\n', R_squared_fit);

    theta0_rad = angle * pi / 180;

    R_fit_full = compute_R(res.thk, res.n1_complex_full, res.n2, waveNum,

```

```

theta0_rad,1);

R_fit_per = R_fit_full * 100;
header = {'拟合反射率'};
writecell(header, output_filepath, 'Sheet', 1, 'Range', 'C1');
writematrix(R_fit_per, output_filepath, 'Sheet', 1, 'Range', 'C2');

plot_fit_res(waveNum, R, R_fit_full, angle, config.waveN_fit_min);
plot_optical_constants(waveLen_full, waveNum, res.n1_complex_full, '外延层
(Sellmeier)');
end

function [x_optimal, R_squared] = global_fit(x0, lb, ub, R_fit, waveLen_fit, angle,
epsilon_inf, model_type)
    waveNum_fit = 10000 ./ waveLen_fit;

    theta0_rad = angle * pi / 180;
    if strcmp(model_type, 'cauchy')
        model_func = @(x, k) model_R_cauchy(x, k, waveLen_fit, theta0_rad,
epsilon_inf);
    elseif strcmp(model_type, 'sellmeier')
        model_func = @(x, k) model_R_sellmeier(x, k, waveLen_fit, theta0_rad,
epsilon_inf);
    elseif strcmp(model_type, 'real_substrate')
        model_func = @(x, k) model_R_real_substrate(x, k, waveLen_fit, theta0_rad);
    end
    options = optimoptions('lsqcurvefit', 'Display', 'off', 'MaxIterations', 1000,
'FunctionTolerance', 1e-9, 'StepTolerance', 1e-10, 'Algorithm', 'trust-region-reflective',
'UseParallel', true);
    [x_optimal, resnorm] = lsqcurvefit(model_func, x0, waveNum_fit, R_fit, lb, ub,
options);
    SS_tot = sum((R_fit - mean(R_fit)).^2);
    R_squared = 1 - resnorm / SS_tot;
end

function R = model_R_sellmeier(params, ~, waveLen, theta0, epsilon_inf)
    thk = params(1);
    selParam = params(2:7);
    k1Param = params(8:9);
    drudeParam = params(10:11);
    waveNum_model = 10000 ./ waveLen;
    n1_complex = cal_n_sellmeier(waveLen, selParam, k1Param);
    n2_complex = cal_n_drude(waveNum_model, drudeParam, epsilon_inf);

```



```

    R = compute_R(thk, n1_complex, n2_complex, waveNum_model, theta0, 1);
end

function R = model_R_real_substrate(params, ~, waveLen, theta0)
    thk = params(1);
    n2 = params(2);
    selParam = params(3:8);
    k1Param = params(9:10);
    waveNum_model = 10000 ./ waveLen;
    n1_complex = cal_n_sellmeier(waveLen, selParam, k1Param);
    R = compute_R(thk, n1_complex, n2, waveNum_model, theta0, 1);
end

function R = model_R_cauchy(params, ~, waveLen, theta0, epsilon_inf)
    thk = params(1);
    cauchyParam = params(2:3);
    k1Param = params(4:5);
    drudeParam = params(6:7);
    waveNum_model = 10000 ./ waveLen;
    n1_complex = cal_n_cauchy(waveLen, cauchyParam, k1Param);
    n2_complex = cal_n_drude(waveNum_model, drudeParam, epsilon_inf);
    R = compute_R(thk, n1_complex, n2_complex, waveNum_model, theta0, 1);
end

function R = compute_R(thk, n1_complex, n2_complex, waveNum, theta0, opt)
% compute_R 计算单层膜的反射率
%   thk: 膜厚 ( $\mu m$ )
%   n1_complex: 外延层复折射率 ( $n+ik$ )
%   n2_complex: 衬底复折射率 ( $n+ik$ )
%   waveNum: 波数 ( $cm^{-1}$ )
%   theta0: 入射角 (弧度)
%   opt: 选项
%       0 - 强制使用双光束模型
%       1 - 强制使用多光束模型
%       2 - 自适应选择 (推荐)

    n0 = 1.0;
    n1 = n1_complex(:);
    n2 = n2_complex(:);
    waveNum = waveNum(:);

    sin_theta1 = n0 * sin(theta0) ./ n1;
    cos_theta1 = sqrt(1 - sin_theta1.^2);

```

```

sin_theta2 = n0 * sin(theta0) ./ n2;
cos_theta2 = sqrt(1 - sin_theta2.^2);

r01_s = (n0*cos(theta0) - n1.*cos_theta1) ./ (n0*cos(theta0) + n1.*cos_theta1);
r12_s = (n1.*cos_theta1 - n2.*cos_theta2) ./ (n1.*cos_theta1 + n2.*cos_theta2);
r01_p = (n1*cos(theta0) - n0*cos_theta1) ./ (n1*cos(theta0) + n0*cos_theta1);
r12_p = (n2.*cos_theta1 - n1.*cos_theta2) ./ (n2.*cos_theta1 + n1.*cos_theta2);

delta = 4 * pi * n1 .* thk .* cos_theta1 .* waveNum / 10000;
exp_term = exp(1i*delta);

if opt == 0
    % 强制双光束
    R = cal_R_db(r01_s, r12_s, r01_p, r12_p, exp_term);
elseif opt == 1
    % 强制多光束
    R = cal_R_mul(r01_s, r12_s, r01_p, r12_p, exp_term);
elseif opt == 2
    % 自适应选择模型
    epsilon_threshold = 0.01;
    [use_multi_beam, ~] = check(thk, n1_complex, n2_complex, waveNum, theta0,
epsilon_threshold);
    R = zeros(size(waveNum));

    % 分别处理需要不同模型的波长点
    if any(use_multi_beam)
        idx = use_multi_beam;
        R(idx) = cal_R_mul(r01_s(idx), r12_s(idx), r01_p(idx), r12_p(idx),
exp_term(idx));
    end
    if any(~use_multi_beam)
        idx = ~use_multi_beam;
        R(idx) = cal_R_db(r01_s(idx), r12_s(idx), r01_p(idx), r12_p(idx),
exp_term(idx));
    end

elseif opt >= 0.005 && opt <= 0.05
    % 使用 opt 作为自适应阈值
    epsilon_threshold = opt;
    [use_multi_beam, ~] = check(thk, n1_complex, n2_complex, waveNum, theta0,
epsilon_threshold);
    R = zeros(size(waveNum));

    if any(use_multi_beam)

```

```

        idx = use_multi_beam;
        R(idx) = cal_R_mul(r01_s(idx), r12_s(idx), r01_p(idx), r12_p(idx),
exp_term(idx));
    end
    if any(~use_multi_beam)
        idx = ~use_multi_beam;
        R(idx) = cal_R_db(r01_s(idx), r12_s(idx), r01_p(idx), r12_p(idx),
exp_term(idx));
    end
    else
        % 默认使用多光束
        R = cal_R_mul(r01_s, r12_s, r01_p, r12_p, exp_term);
    end
end
end

```