

基于IRLS模型与Fresnel-Airy模型的外延层厚度问题研究

摘要

前言： 本文面向第三代半导体外延层厚度的无损测量需求，围绕红外干涉谱的物理机理与数据特性，构建 Fresnel–Cauchy–Drude [6, 7, 12] 一体化模型与 IRLS 稳健全谱拟合算法[11]；在多入射角数据[8]上验证稳定性、敏感性与不确定度，并讨论多光束干涉的判别与修正策略。

正文：

(1) 问题一（建模）：在单次反射的假设下，推导总相位 $\Phi(\nu) = 4\pi n(\lambda) d \cos \theta_t \nu + \phi$ ，折射率采用 Cauchy 色散并可叠加 Drude 自由载流子（复折射率）。由相邻极值间距给出厚度初估 [11] $d \approx (2n \cos \theta_t \Delta\nu)^{-1}$ ，明确关键假设与适用域。

(2) 问题二（算法）：提出“极值/FFT 初值[3, 12] + 稳健全谱拟合”的求解链：采用 SG 平滑与移动中位数剔除离群处理数据[2, 9]；自适应寻峰与 FFT 兜底给出厚度中位数 d_0 ；以一次背景 + 振荡核为框架，采用 Tukey 权的 IRLS 加权最小二乘，内核包含斜入射 Fresnel、s/p 平均、Cauchy–Drude 色散与额外相位；输出 RMSE、尾段 RMSE、残差统计、权重分布[4]与 95% 置信区间。得到的跨角结果的置信区间重叠，厚度估计一致，可重复收敛，显示模型在“正确物理参数以及合理初值”邻域内稳定。

(3) 问题三（多光束）：基于 Fresnel–Airy 级数给出多光束干涉的必要条件[1, 10]（界面反射率、相干长度、吸收/载流子损耗），构建判别准则，并在存在多光束时切换 Airy 模型修正厚度估计，降低由 $F - P$ 腔效应带来的系统偏差。

结尾： 总体上，物理全谱模型稳定但是敏感：在固定物理参数下跨角一致且可复现，但对折射率与、Drude 参数及厚度初值较敏感，同时低波数段残差暴露了模型型误差。该模型优点在于物理内核完整、稳健性与可审计性强；局限在于 $n-d$ 可辨识度与整体 RMSE 门限压力。目前可能的改进路径包括：多角度/多谱段联合拟合， (d, N, μ, m^*) 的先验约束联合估计，更精确的 Sellmeier+ $k(\lambda)$ /声子项与界面粗糙层建模，以及入射角与强度标定自校准，从而进一步压缩残差与不确定度。

1 问题重述

1.1 问题情境

碳化硅作为第三代半导体新兴材料，展现出卓越的综合性能特征。其外延层厚度是影响器件性能的关键参数之一，准确控制与科学评估对器件制造具有重要意义。因此，我们的目标是建立一套科学、准确、可靠的碳化硅外延层厚度测试标准。

红外干涉法是一种无损的外延层厚度测量方法。其原理基于外延层与衬底因载流子掺杂浓度差异导致的折射率差异：当红外光入射时，分别从外延层表面和衬底界面反射的光束会产生干涉条纹。通过分析干涉光谱的波长、外延层折射率及入射角等参数，即可确定外延层厚度。

需注意的是，外延层的折射率并非固定常数，其受载流子浓度和红外光波长等因素影响。

1.2 问题要求

(1) 若仅考虑外延层与衬底界面发生单次反射和透射所形成的干涉条纹，可建立相应的确定性数学模型。基于实验测得的波数、反射率等已知数据，并结合合理的假设常量，求解外延层厚度。

(2) 依据上题所建立的数学模型，结合附件1与附件2所提供的碳化硅晶圆片光谱实测数据，设计适用于实验数据的外延层厚度确定算法，得出相应计算结果，并采用适当指标对结果的可靠性进行分析。

(3) 由于光波在外延层与衬底界面可能发生多次反射与透射，从而引发多光束干涉现象，需进一步考虑该复杂情形。应推导多光束干涉发生的必要条件，并分析其对上述单次干涉模型中厚度计算精度可能带来的影响。基于多光束干涉的形成条件，对附件3与附件4所提供的硅晶圆片测试数据中是否存在多光束干涉进行判别，并给出相应的厚度计算数学模型与算法，给出硅外延层厚度的计算结果。

同时，需判定碳化硅晶圆片测试数据（附件1与附件2）中是否出现多光束干涉现象。若存在该现象，需提出有效方法以消除其对原有厚度计算精度的影响，并给出修正后的计算结果。

2 问题分析

2.1 问题一的分析：单次反射干涉与可识别性

本题以空气/外延层/衬底三层体系为对象，假定界面近镜面、外延层在光斑内均匀各向同性且入射角与强度已完成基本标定。红外光在外延层表面与衬底界面形成两束主反射，相干叠加产生条纹，其周期与相位偏移由折射率色散、膜厚与入射几何共同决定。本问题可以先从条纹的极值间距或谱的主频提取稳健的厚度初值，再以全谱物理模型细化。为提高可识别性，可以联合使用多入射角或更宽谱段的信息，并对材料色散与自由载流子参数给出合理先验，使“同一厚度在不同观测条件下表现一致”的约束得以生效，从而为后续数值求解提供可收敛、可验证的起点与边界。

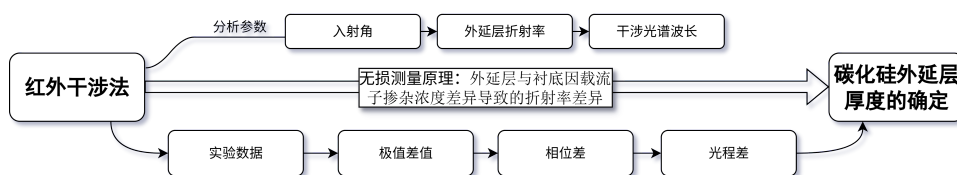


图 1: 问题一思维导图

2.2 问题二的分析：实测数据的适用性与稳健求解

问题二关注模型如何在实测谱上可落地、可复现、可审计。先对附录数据进行排序去重、归一化与温和平滑，并用移动中位数抑制离群点，同时保持条纹主结构不被过度削弱；随后以极值序列或频谱主频给出厚度初值。建模侧采用“缓变背景+振荡核”的观测框架：简化余弦核用于快速体检与定位问题，含斜入射、偏振平均、色散/载流子吸收与额外相位的物理核用于物理解译与最终估计。求解侧以迭代重加权的非线性最小二乘为主线，并设置参数变化、权重稳定、目标改进与厚度步长等多重收敛准则，避免伪收敛与过拟合。评估侧不应该以单一误差为准，而综合检查全段与尾段误差的一致性、残差是否绕零对称、是否存在结构化波动、权重是否在某些波段长期偏低、厚度不确定度是否合理以及跨角度置信区间是否重叠。若通过上述检查，可认定“数据—模型—算法”闭环可靠；若不通过，则依据残差形态与权重分布回溯色散/吸收建模、入射角或强度标定、背景刚度与初值选择等环节，并据此调整先验、扩展角度或谱段，直至获得稳定可解释的结果。

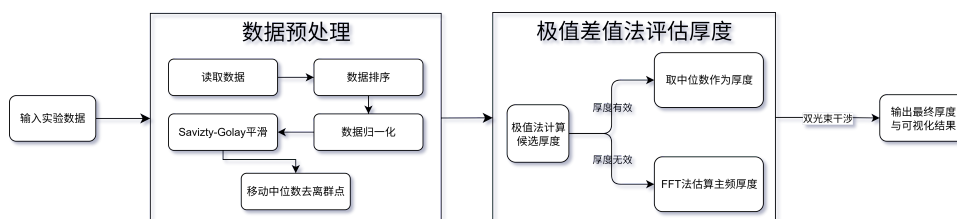


图 2: 问题二思维导图

2.3 问题三的分析：多光束干涉的触发与模型切换

当界面反射率较高、相干长度充足且材料吸收不过强时，高阶往返反射会显著参与相干叠加，具体表现为条纹对比度升高与残差出现周期性的“腔型”纹理，且两束模型在不同入射角下难以自然汇合。此时应从证据驱动出发，先以残差纹理、低权重波段与“尾段好而全段差”的分裂现象作为触发信号，再切换到包含腔效应的 Airy 模型。而在参数层面宜可以采用分阶段策略：先在条纹稳定的子带校准材料色散与载流子吸收，再推广至全谱；同时以独立测试或合理区间为自由载流子参数提供先验，抑制过度自由带来的不稳定性。在数值层面上，可以继续沿用稳健求解与多重收敛准则，并以统计指标（全段/尾段误差、残差形态、低权重比例）与物理可行性（参数落入可达范围、跨角一致性）共同完成模型选择。若腔模型优势不显著或引入不可解释参数，应及时回退两束模型并优先修正标定与先验。此问的目标是最简充分的物理刻画，在不同实验场景下获得一致、可解释且不确定度受控的厚度估计。

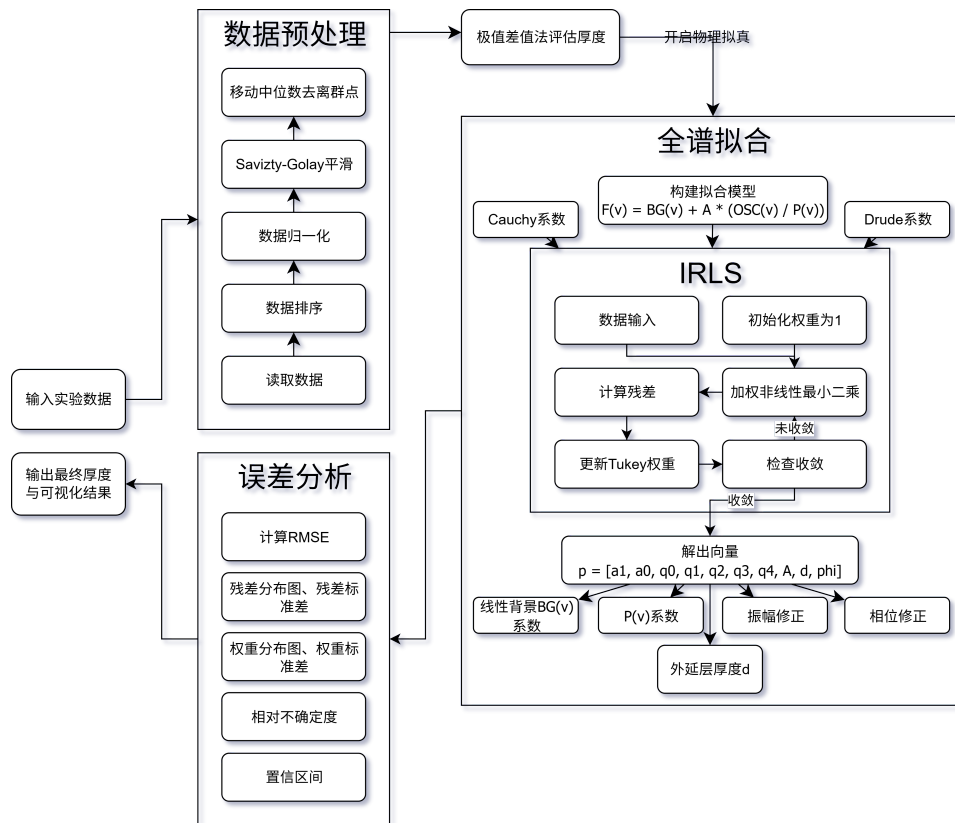


图 3: 问题三思维导图

3 模型假设

(1).光从光疏介质射向光密介质，光的相位差反射后有半波损失:

光波从折射率较小（波疏）的介质射向折射率较大（波密）的介质时，反射光在界面处发生 π 相位突变的现象。

(2).空气中存在一固定位置的测量仪器将相干光汇聚在一点:

确保光线的起点、终点相同，符合光束相干的情况，计算光程差。

(3).同一时间光源仅定向发射一道光束:

简化模型，确保同一时间只有一束光进入到晶圆中。

(4).同一材料不同位置折射率相等，不存在由材料缺陷导致的局部折射率波动:

认为材料完全均匀各向同性， $n(x, y, z)=\text{常数}$ ，无缺陷、应变或掺杂起伏引起的局域折射率变化，从而可忽略散射与相位噪声。

(5).碳化硅周围空间温度、湿度、空气成分等物化性质保持不变:

固定环境边界条件，使周围介质折射率与热/湿/化学效应不随时空变化，消除热致折射、湿吸附和成分波动对光学与材料性质的影响。

(6).碳化硅外延层表面与衬底表面应是两道理想平整平面:

将外延层与衬底表面视为理想平面，无粗糙度、翘曲和微倾角，因此只发生镜面反射/折射，便于采用平面薄膜与几何光学模型。

(7).不考虑色散模型对于相位差公式中余弦部分的影响:

简化模型，防止迭代过程中出现欠拟合或者不收敛的情况。

4 符号说明

表 1: 本文的符号说明

符号	说明	单位
d	外延层厚度	cm
$\hat{\nu}$	波数	cm^{-1}
λ	空气波长	cm
λ_o	外延层波长	cm
θ_t	折射角	$^\circ$
θ_i	入射角	$^\circ$
θ_r	反射角	$^\circ$
n_a	空气折射率	无量纲
n_o	外延层折射率	无量纲
n_b	外延层底部折射率	无量纲
$n_o(\hat{\nu})$	外延层折射率(关于波数的函数)	无量纲
Δs	光程差	cm
$\Delta\phi$	相位差	cm
E	波幅	cm
$\Delta\phi$	相位差	$^\circ$
$\phi(n)$	相位差(半波损失)	$^\circ$
ϕ_{all}	总相位差	$^\circ$

5 模型建立与求解

5.1 问题一模型的建立与求解

5.1.1 物理理论模型：余弦干涉模型的建立

为建立波数、反射率与外延层厚度之间的数学关系，需进行合理的理论推导与模型假设。根据题目附件所提供的实验数据，通过MATLAB绘制反射率随波数变化的二维关系图，可观察到在波数较大区域呈现显著的周期性波动特征，此现象与三角函数波形高度吻合。结合反射率本质上表征光波能量反射特性的物理本质，可推断该周期性特征源于相干波叠加效应：

$$\Delta\phi_{all} = \Delta\phi + \phi(n) = 4\pi d \cos \theta_t \cdot n_o(\hat{\nu}) \cdot \hat{\nu} + \phi(n) \quad (1)$$

这条公式经典光学理论中的振幅叠加原理：反射率与相干波相位差之间存在确定的函数关系，其中三角函数项正好能解释实验数据呈现的周期特性， $n_o(\hat{\nu})$ 一项需要通

过Cauchy色散模型和Drude自由载流子模型得到。通过解析该表达式，我们可以建立完整的数学模型。

依据物理光学基本原理，反射率定义为反射光强与入射光强之比。在本题预设情形中，反射光由两个相干分量组成：

- 1.在空气-外延层界面直接反射的光束；
- 2.经外延层透射后在衬底界面反射并再次穿出外延层的光束。

根据波动叠加原理，合成反射光强与两束光之间的相位差有关，而该相位差又与光程差有关。根据光程差与波长存在反比关系（波长与波数互为倒数） $\lambda = \frac{1}{\nu}$ ，通过建立光程差与波数的函数关系，最终可实现反射率与波数之间的数学建模。具体推导过程如下：

首先我们需要得到两束光光程差与外延层厚度的关系。对于本题中的相干光而言，两束光线拥有共同的起点和终点，光程差 Δs 即为两者在这之间走过的路程差

图4为一个经典的薄膜干涉，从起点出发的一条光线在经过碳化硅表面后产生了两束平行光线：

- 1.直接经过外延层上表面反射后得到第一条光线；

2.部分光线经过空气与外延层的上表面，透射入外延层，然后经由外延层和衬底交界处的反射，以入射对称的路径进入到空气中。

两束光的光程差可由三角方程得到，具体可表示为：

$$\Delta s = 2d \cos \theta_t \quad (2)$$

其中， Δs 为光程差， d 为外延层厚度， θ_t 为折射角。

考虑变折射率的情况下，入射角和折射角满足修正后的斯涅尔定律：

$$n_a \sin \theta_i = n_o \sin \theta_t \quad (3)$$

在外延层中，光的波长一定，已经设定为 λ_o ，因此两束光的相位差可由公式 $\Delta\phi = \frac{\Delta s}{\lambda_o}$ 得到。同时，由光线在介质中的波长公式 $\lambda_o = \frac{\lambda}{n_o}$ 可以得到，其中 n_o 为外延层的折射率， λ 为光线在空气中的波长，结合波数与波长的倒数关系 $\lambda = \frac{1}{\nu}$ ，代入得到公式整体的公式如下：

$$\Delta\phi = \frac{\Delta s}{\lambda_o} = \frac{2d \cos \theta_t}{\lambda_o} \cdot 2\pi = \frac{4\pi d \cos \theta_t}{\frac{\lambda}{n_o(\hat{\nu})}} = 4\pi d \cos \theta_t \cdot n_o(\hat{\nu}) \cdot \hat{\nu} \quad (4)$$

由假设，光在穿透空气与外延层交界面时会产生半波损失，可以得到具体的总相位差公式如下：

$$\Delta\phi_{all} = \Delta\phi + \phi(n) = 4\pi d \cos\theta_t \cdot n_o(\hat{v}) \cdot \hat{v} + \pi \quad (5)$$

公式(5)中： $\Delta\phi$ 为由外延层内部传播引入的相位差， $\phi(n)$ 为由半波损失引起的相位差

最终的反射率需要通过波的能量叠加公式与相位差建立数学联系，从而和波数形成函数关系。由振幅叠加公式，本题最终的反射率表达式为

$$|E|^2 = |E_r|^2 + |E_t|^2 + 2|E_r||E_t|^2 \cos(\Delta\phi_{all}) \quad (6)$$

其中 E_r 、 E_t 是两束光线的反射率，需要经过详细地计算得出，计算如下：

反射系数、透过系数计算

由菲涅尔公式，光在介质表面的反射系数、折射系数与介质折射率、入射角、折射角存在数学关系本题存在两个界面，因此可以得到两组透射率、反射率公式：

$$r_s^{ij} = \frac{n_i \cos\theta_i - n_j \cos\theta_j}{n_i \cos\theta_i + n_j \cos\theta_j} \quad (7)$$

$$t_s^{ij} = \frac{2n_i \cos\theta_i}{n_i \cos\theta_i + n_j \cos\theta_j} \quad (8)$$

$$r_p^{ij} = \frac{n_j \cos\theta_i - n_i \cos\theta_j}{n_j \cos\theta_i + n_i \cos\theta_j} \quad (9)$$

$$t_p^{ij} = \frac{2n_i \cos\theta_i}{n_j \cos\theta_i + n_i \cos\theta_j} \quad (10)$$

上面公式表示光线由 i 进入 j 的反射率和透射率，由假设本建模只考虑 s 偏振，因此后续推导将只用到前两个公式。

光线1只在空气和外延层表面产生一次反射，总的系数为 r_{01} ；光线2先从空气进入外延层产生透射系数 t_{01} ，然后经过衬底界面反射产生反射系数 r_{12} ，最后从外延层进入空气产生透射系数 t_{10} ，因此总的系数为 $r_{01} * t_{01} * r_{12}$ ，具体的公式如下：

$$\frac{E_r}{E_i} = t_{01} \cdot r_{12} \cdot t_{10} \quad (11)$$

$$\frac{E_t}{E_i} = r_{01} \quad (12)$$

公式(11)(12)中， E_i 为入射光的强度。

代入振幅强度公式将平方项和交叉项用 Q 、 P 表示，可以得到下面的简化公式：

$$|E|^2 = Q(\hat{v}) + 2P(\hat{v}) \cos(\Delta\phi_{all}) \quad (13)$$

将已经得出的反射系数、透射系数 t_{01} 、 r_{12} 、 t_{10} 代入：

$$|E|^2 = Q(\hat{\nu}) + 2P(\hat{\nu}) \cos(\Delta\phi_{all}) =$$

$$\left(\frac{n_a \cos \theta_i - n_o(\hat{\nu}) \cos \theta_t}{n_a \cos \theta_i + n_o(\hat{\nu}) \cos \theta_t} \right)^2 + \left(\frac{2n_a \cos \theta_i}{n_a \cos \theta_i + n_o(\hat{\nu}) \cos \theta_t} \cdot \frac{n_o(\hat{\nu}) \cos \theta_t - n_b \cos \theta_b}{n_o(\hat{\nu}) \cos \theta_i + n_b \cos \theta_b} \cdot \frac{2n_o(\hat{\nu}) \cos \theta_t}{n_o(\hat{\nu}) \cos \theta_t + n_a \cos \theta_i} \right)^2$$

$$+ 2 \cdot \frac{n_a \cos \theta_i - n_o(\hat{\nu}) \cos \theta_t}{n_a \cos \theta_i + n_o(\hat{\nu}) \cos \theta_t} \cdot \frac{2n_a \cos \theta_i}{n_a \cos \theta_i + n_o(\hat{\nu}) \cos \theta_t} \cdot \frac{n_o(\hat{\nu}) \cos \theta_t - n_b \cos \theta_b}{n_o(\hat{\nu}) \cos \theta_i + n_b \cos \theta_b} \cdot \frac{2n_o(\hat{\nu}) \cos \theta_t}{n_o(\hat{\nu}) \cos \theta_t + n_a \cos \theta_i}$$

$$\cdot \cos(4\pi d \cos \theta_t \cdot n_o(\hat{\nu}) \cdot \hat{\nu} + \pi) \quad (14)$$

为了简化问题一的模型，只引入Cauchy色散模型，外延层介质折射率满足以下关系：

$$n_o(\hat{\nu}) = A + B * \hat{\nu}^2 + C * \hat{\nu}^4 \quad (15)$$

其中， A 、 B 、 C 是待拟合的常数， $\hat{\nu}$ 是波数。

将上式代入到公式(4)中，分子部分关于 $n_o(\hat{\nu})$ 最高项次数为3次，分母部分关于 $n_o(\hat{\nu})$ 的最高项次数为4次。因此，总体估计 $P(\hat{\nu})$ 分母的次数为1次，再由Cauchy色散公式，可以得到 $Q(\hat{\nu})$ 关于 $\hat{\nu}$ 多项式的最高次数为-4次。

由此，振幅公式 $P(\hat{\nu})$ 可由多项式表示：

$$P(\hat{\nu}) = \frac{4\pi d n(\hat{\nu}) \cos(An_o(\hat{\nu})\theta_t)}{q_4 v^4 + q_3 v^3 + q_2 v^2 + q_1 v^1 + q_0} \quad (16)$$

已经建立 E 与 $\hat{\nu}$ 的关系，下面根据余弦项，运用极值法求解厚度 d 与 $\hat{\nu}$ 的关系：

已知干涉条纹产生条件，相干叠加增强满足相位差 $\Delta\phi_{all} = 2k\pi$ ，相干相消满足相位差 $\Delta\phi_{all} = (2k + 1)\pi$ 。

由公式(4)两条相邻条纹对应 k 与 $k+1$ ：

$$k = 2n_o(\hat{\nu})d \cos \theta_t \hat{\nu}_k \quad (17)$$

$$k + 1 = 2n_o(\hat{\nu})d \cos \theta_t \hat{\nu}_{k+1} \quad (18)$$

公式相减得到相邻极值间的波数间隔：

$$\Delta\hat{\nu} = \hat{\nu}_{k+1} - \hat{\nu}_k = \frac{1}{2n_o(\hat{\nu})d \cos \theta_t} \quad (19)$$

此处可以先将色散忽略，问题二为了精确分析再引入，则有：

$$k = 2nd \cos \theta_t \hat{\nu} \quad (20)$$

移项可得：

$$d = \frac{1}{2n\Delta\hat{\nu} \cos \theta_t} \quad (21)$$

问题一模型已建立完毕。

5.2 问题二模型的建立与求解

5.2.1 算法模型的建立

Savitzky-Golay平滑滤波器

数据预处理部分，我们先竞选数据排序与归一化处理，之后使用Savitzky-Golay平滑滤波器，应用局部加权最小二乘拟合方法，通过多项式拟合对数据进行平滑处理。相较于简单的移动平均平滑数据，Savitzky-Golay滤波器能有效抑制随机噪声。更好地保留局部特征（极值点的位置和幅度）。

$$y_s(i) = \sum_{k=-m}^m c_k y(i+k) \quad (22)$$

其中， $y(i+k)$ 是原始数据的值， c_k 是平滑系数，由最小二乘拟合得到。

移动中位数去离群点

在平滑之后，我们使用移动中位数去离群点来剔除数据中存在的尖锐噪声点。移动窗口内，远离中位数的点很可能就是离群点。

通过计算移动中位数：定义一个长度为 L 的窗口，在数据序列 y_s 上滑动。对于窗口中心的每个数据点 $y_s(i)$ ，计算窗口内所有点的中位数 $M(i)$ 。

$$M(i) = \text{median}(\{y_s(i-k), \dots, y_s(i), \dots, y_s(i+k)\}), \quad k = \frac{L-1}{2} \quad (23)$$

再计算移动MAD：移动窗口内所有数据点与中位数 $M(i)$ 的绝对偏差的中位数。

$$\text{MAD}(i) = \text{median}(\{|y_s(i-k) - M(i)|, \dots, |y_s(i+k) - M(i)|\}) \quad (24)$$

如果一个点满足：

$$|y_s(i) - M(i)| > n \cdot \text{MAD}(i) \quad (25)$$

则定义为离群点舍弃，其中， n 常用的标准 $n = 3$ （类似 3σ 原则）。随后使用线性插值方法进行填充。

寻峰算法

对于一个离散信号序列 $y[i]$ ，一个点 i 需满足：

$$y[i] > y[i-1] \quad \text{and} \quad y[i] > y[i+1] \quad (26)$$

才是局部极大值的候选点，反之，是局部极小值的候选点。

计算峰（或谷）的高度差：

$$P = y[\text{peak}] - \min(y[\text{left_base}], y[\text{right_base}]) \quad (27)$$

其中 $left_{base}$ 和 $right_{base}$ 是峰左右两侧信号下降后又上升的鞍点位置。只有高度差 P 大于阈值的极值点才被保留。最后进行最小距离过滤：避免在同一个干涉峰附近因噪声而产生多个虚假极值点。

$$|E^2| = A_1 v + A_0 + \frac{4\pi dn(\hat{\nu}) \cos(An(\hat{\nu})\theta)}{q_4 v^4 + q_3 v^3 + q_2 v^2 + q_1 v^1 + q_0} \quad (28)$$

极值法估计膜厚度

找到极值点后，利用干涉条纹的周期性来估算薄膜的厚度。通过膜厚度 d 与波数差 $\Delta\hat{\nu}$ 之间的关系：

$$d = \frac{1}{2n \cos(\theta_t) \Delta\hat{\nu}} \quad (29)$$

其中， n_1 是薄膜的折射率， θ_t 是透射角， $\Delta\nu$ 是两个相邻极值之间的波数差。可以直接求出厚度 d 。

FFT频谱分析

当极值点数量不足时，无法直接通过中位数作为厚度输出，使用FFT频谱分析来直接提取干涉振荡的主频。

首先对平滑后的数据 y_s 进行去趋势和去直流处理，得到纯的振荡信号 y_{osc} 。

$$y_{osc} = y_s - \text{trend}(y_s) - \text{mean}(y_s) \quad (30)$$

之后进行快速傅里叶变换：对 y_{osc} 执行FFT，将其从波数域 ($\hat{\nu}$) 变换到频率域 (f)。FFT的计算公式为：

$$Y[k] = \sum_{j=0}^{N-1} y_{osc}[j] \cdot e^{-2\pi i \frac{jk}{N}} \quad (31)$$

其中 N 是数据点数， $Y[k]$ 是复频谱， k 对应频率索引。计算频谱的幅度谱 $|Y[k]|$ ，并忽略接近直流的低频分量 ($Y(1:3) = 0$)。然后寻找幅度谱中最大值对应的频率索引 k_{\max} 。

$$k_{\max} = \arg \max_k |Y[k]|, \quad \text{for } k \in [1, \lfloor N/2 \rfloor] \quad (32)$$

将找到的主频索引 k_{\max} 转换为实际的物理频率 f ，其倒数即为振荡的周期 $\Delta\tilde{\nu}$ 。

$$f = \frac{k_{\max}}{N \cdot \Delta x}, \quad \Delta\tilde{\nu} = \frac{1}{f} \quad (33)$$

其中 Δx 是波数间隔。最后，将 $\Delta\tilde{\nu}$ 代入双光束干涉公式，即可得到厚度估计值：

$$d = \frac{1}{2n_1 \Delta\tilde{\nu} \cos \theta_t} \quad (34)$$

表 2: 碳化硅外延层厚度测量结果

参数	符号/单位	10°入射角	15°入射角
极值法平均厚度	d_{avg} (μm)	8.441	8.381
初值(中位数)	d_{median} (μm)	8.124	8.131
厚度标准差	σ (μm)	1.244	1.286
厚度不确定度(标准误差)	u (μm)	0.259	0.268

5.2.2 计算结果与可靠性分析

在代入数据，使用了上述的数学模型进行厚度计算之后。我们得到了以下计算结果：

对于入射角10°，平均厚度为8.441 μm ；对于入射角15°，平均厚度为8.381 μm 。两个角度下的平均厚度差异仅为0.06 μm ，相对差异约为0.7%。物理上厚度应是一个固定值，不随角度变化而显著改变。不同入射角入射角下平均厚度差异不大，表明模型具有良好的鲁棒性。

对于入射角10°，厚度标准偏差为1.244 μm ；对于入射角15°，标准偏差为1.286 μm 。标准偏差较大，表明单个极值点计算的厚度值存在较大分散性。数据点波动较大，可能需要更真实的拟合方法或者更强的过滤性以提高结果的可靠性。

对于入射角10°，标准误差为0.259 μm ；对于入射角15°，标准误差为0.268 μm 。标准误差较小，表明平均厚度的估计精度较高。对于8.4 μm 的厚度，相对不确定度约为3.1%（半导体外延层厚度测量通常要求不确定度小于5%），算法所以可靠性较高。

碳化硅外延层厚度通常在几微米到几十微米之间，计算结果8.4 μm 处于典型范围内，符合物理预期。

结论是：模型算法效果较好，计算结果可靠。

5.3 问题三模型的建立与求解

5.3.1 多光束干涉必要条件与引入影响后的数学模型

Drude自由载流子模型部分

当掺有自由载流子时，Drude模型用于修正折射率的虚部，描述由于自由载流子造成的吸收。自由载流子的效应通过如下的Drude公式计算：

$$\varepsilon_{\text{Drude}}(\omega) = -\frac{\omega_p^2}{\omega^2 + i\gamma\omega} \quad (35)$$

其中 ω_p 是等离子体频率，表示材料中自由载流子的响应频率， γ 是载流子的阻尼系数，表示载流子与晶格的散射。

具体计算步骤如下：

(1) 等离子体频率：

$$\omega_p^2 = \frac{Ne^2}{\epsilon_0 m^*} \quad (36)$$

其中 N 是载流子浓度， e 是电子的电荷， ϵ_0 是真空的介电常数， m^* 是载流子的有效质量。

(2) 阻尼系数：

$$\gamma = \frac{e}{m^* \mu} \quad (37)$$

其中 μ 是载流子的迁移率。

(3) 复折射率的计算：

复折射率 $n(\lambda) = n'(\lambda) + in''(\lambda)$ 被计算为：

$$\epsilon_{\text{total}} = \epsilon_{\text{base}} + \epsilon_{\text{Drude}} \quad (38)$$

然后通过取平方根得到复折射率：

$$n_c = \sqrt{\epsilon_{\text{total}}} \quad (39)$$

Fresnel-Airy模型

首先，本问题首先要确定多光束 *Fresnel-Airy*模型的引入：当光线从空气中进入外延层后，薄膜内会发生多次反射，最后从介质中透射入空气中。在问题一的分析中，本论文推导得到边界的折射系数和反射系数公式：

$$r_{ij}^s = \frac{n_i \cos \theta_i - n_j \cos \theta_j}{n_i \cos \theta_i + n_j \cos \theta_j}, \quad t_{ij}^s = \frac{2n_i \cos \theta_i}{n_i \cos \theta_i + n_j \cos \theta_j}$$

$$r_{ij}^p = \frac{n_j \cos \theta_i - n_i \cos \theta_j}{n_j \cos \theta_i + n_i \cos \theta_j}, \quad t_{ij}^p = \frac{2n_i \cos \theta_i}{n_j \cos \theta_i + n_i \cos \theta_j}$$

光线在外延层内单次反射产生相位差为 $\delta = n_o(\hat{v})d \cos \theta_o$ 假设光线在外延层产生 n 次反射，则外延层内反射产生相位差为：

$$\delta_{\text{all}} = k_o n_o d \cos \theta_i = \frac{2\pi}{\lambda} n_1 d \cos \theta_i \quad (40)$$

其中，若 n_1 为复数，相位差 δ 也为复数。而光线每多一次来回的反射，振幅会衰减为前级的 $r_{01}r_{10}\kappa_1$ 因此，如若用复数形式描述单次反射产生的振幅、相位，则可以表示为：

$$\frac{E_{m+1}}{E_m} = r_{10}r_{12}e^{2i\delta} \quad (41)$$

n束光干涉叠加后的表达式如下：

$$E = \sum_{m=0}^n E_m + r_{01}E_1 = t_{01}t_{10}r_{12} \sum_{m=0}^n (r_{10}r_{12}e^{2i\delta})^m \quad (42)$$

移项后，由几何级数求和公式得到：

$$\frac{E}{E_1} = r_{01} + t_{01}t_{10}r_{12}e^{2i\delta} \quad (43)$$

$$\sum_{m=0}^{\infty} (r_{01}r_{12}e^{2i\delta})^m = r_{01} + \frac{t_{01}t_{10}r_{12}e^{2i\delta}}{1 - r_{10}r_{12}e^{2i\delta}} \quad (44)$$

引入airy模型，利用边界恒等式： $t_{01}t_{10} = 1 - r_{10}r_{01}$ 得到：

$$r = \frac{r_{01} + r_{12} + 2\sqrt{r_{01}r_{12}} \cos(2\delta)}{1 + r_{01}r_{12} + 2\sqrt{r_{01}r_{12}} \cos(2\delta)} \quad (45)$$

将以上所有的物理模型带入到振幅公式中，得到公式：

$$R(\hat{v}) = \frac{1}{2}(|r_s(\hat{v})|^2 + |r_p(\hat{v})|^2) \quad (46)$$

其中， $r_{s,p} = \frac{r_{01}^{sp} + r_{12}^{spE}}{1 + r_{01}^{sp}r_{12}^{spE}}$ 。而只有在满足下列条件时才会出现可观测的多光束干涉（Airy谱）。

$$E = \exp[4\pi i \hat{v} d \cos \theta_t] \quad (47)$$

根据公式(45)，可以得到产生显著多光束反射的必要条件：

1.至少两个界面且界面反射率不小

$$R = \sqrt{r_{10}r_{12}} \quad (48)$$

由于反射率过小，几何级数高阶项可忽略，几何级数退化，数学表达式上只剩前两项，仅剩近似单光束。此时，应满足腔精细度公式：

$$f = \frac{4R}{(1 - R)^2} \geq 1 \quad (49)$$

R为界面反射率代表值

2.时间相干性足够

相邻往返的光程差不能超过光源相干长度。由波数域近似， $\Delta OPL = 2n_1d \cos \theta$ 则相干光的长度 $LC = \frac{\Delta}{\Delta\lambda^2}$ 应满足： $LC \geq \Delta OPL$

否则条纹会平均掉。

3.吸收/自由载流子损耗不能太大

若每次反射振幅衰减过强，公式()中的几何级数高阶项就会被抑制，退化为单光束反射。定义光线在外延层每一次往返的衰减指数 η_1 存在：

$$\eta = 4\pi\kappa_1\hat{v}d\cos\theta_1 \quad (50)$$

其中（ κ_1 为Drude项）由此前的物理模型和必要条件分析，用多光束反射和单光束反射的模型拟合同一组数据，进而判定数据是否多光束反射显著。

5.3.2 基于Fresnel-Airy模型的全谱拟合算法

全谱拟合参数

全谱拟合的目标是寻找一组物理参数，使得由上述物理模型计算出的理论反射谱 $R_{\text{model}}(\tilde{\nu})$ 与实验测量谱 $R_{\text{exp}}(\tilde{\nu})$ 之间的差异最小化。待拟合的参数主要包括：

- 核心物理参数：薄膜厚度 d 、系统相位偏差 ϕ 、振荡幅度 A 。
- 背景参数 BG ：用于修正缓慢变化的背景信号，如一次项斜率 a_1 和截距 a_0 。
- 分母修正参数 Q ： q_0 至 q_4 ，用于经验性地修正模型与实验数据在振幅上的偏差，其构成一个四次多项式分母 $q_0 + q_1x_n + q_2x_n^2 + q_3x_n^3 + q_4x_n^4$ 。

迭代重加权最小二乘法(IRLS)工作原理

为克服测量数据中异常值(Outliers)的影响，我们采用IRLS这一稳健回归算法[5, 6]进行拟合。其核心思想是通过迭代不断调整每个数据点的权重，降低异常值在拟合中的影响力。

权重初始化：首次迭代时，所有数据点的权重 $w_i^{(0)}$ 被设置为1，采用标准的最小二乘法。

加权非线性最小二乘求解：在第 t 次迭代中，求解以下加权优化问题：

$$\min_{\mathbf{p}} \sum_{i=1}^N w_i^{(t)} [R_{\text{exp}}(\tilde{\nu}_i) - R_{\text{model}}(\tilde{\nu}_i; \mathbf{p})]^2 \quad (51)$$

其中 \mathbf{p} 为待拟合参数向量 $[a_1, a_0, q_0, \dots, q_4, A, d, \phi]^T$ 。该步骤调用 *lsqnonlin* 优化器完成。

残差计算与权重更新：根据当前拟合结果计算残差 $r_i^{(t)} = R_{\text{exp}}(\tilde{\nu}_i) - R_{\text{model}}(\tilde{\nu}_i; \mathbf{p}^{(t)})$ 。使用 *TukeyBiweight* 函数根据残差重新计算权重，以避免异常值的影响：

$$u_i = \frac{r_i}{c \cdot \text{MAD}} \quad (52)$$

$$w_i^{(t+1)} = \begin{cases} [1 - (u_i)^2]^2, & \text{if } |u_i| \leq 1 \\ 0, & \text{if } |u_i| > 1 \end{cases} \quad (53)$$

其中, c 为调节常数 (通常取4.685), MAD (Median Absolute Deviation) 为残差的中位数绝对偏差, 用于稳健地估计残差的尺度。收敛性检查: 迭代重复步骤2和3, 直到满足以下严格的收敛标准之一:

- 参数稳定: 参数的相对变化 $\|\mathbf{p}^{(t)} - \mathbf{p}^{(t-1)}\|/\|\mathbf{p}^{(t-1)}\|$ 小于绝对容差。
- 权重稳定: 最大权重变化 $\max |w_i^{(t)} - w_i^{(t-1)}|$ 小于阈值。
- 目标函数稳定: 加权残差平方和的相对或绝对改进量小于阈值。
- 专属厚度收敛: 厚度 d 的相对或绝对变化量小于专属阈值。

当上述条件连续多轮满足或达到最大迭代次数时, 迭代终止。

5.3.3 硅外延层厚度计算与结果

要计算硅外延层厚度, 我们需要获得Drude自由载离子浓度与柯西色散参数。

我们使用DeepSeek查阅文献[13]得到自由载离子掺杂浓度在外延层约为 $10^{16} \sim 10^{18}/\text{cm}^3$, 而在衬底层约为 $10^{18} \sim 10^{19}/\text{cm}^3$

硅晶圆片外延层的柯西色散参数与衬底层数值相差不大, 均约为常数项: 3.416, 负二次项: 0.18, 负四次项: 0.014。

将参数代入全谱拟合程序后得到以下数据:

表 3: 硅晶圆片厚度测量结果

分析维度	极值法	双光束干涉模型	多光束干涉模型
10° 厚度结果 (μm)	3.613	4.103	1.100
15° 厚度结果 (μm)	3.601	4.076	1.109
双角度一致性 (差异)	0.012 μm	0.027 μm	0.009 μm
内在精度 (RMSE)	-	0.1121/0.1266	0.1175/0.1263
测量精密度 (不确定度, μm)	0.077/0.078	0.282/0.315	0.048/0.047
相对不确定度 (%)	2.13/2.17	6.87/7.72	4.35/4.26
置信区间 (95%, μm)	-	[3.550, 4.655]/[3.459, 4.692]	[1.006, 1.194]/[1.016, 1.201]
残差均值	-	0.0301/0.0343	-0.0252/-0.0213
残差标准差	-	0.1080/0.1219	0.1148/0.1245

续下页

表 3 硅晶圆片厚度测量结果 (续)

分析维度	极值法	双光束干涉模型	多光束干涉模型
残差分布 (偏度/峰度)	-	3.662/16.895	-1.973/8.810
尾段RMSE	-	3.642/16.728	-1.672/8.368
低权重比例 (%)	-	0.0057/0.0070	0.0058/0.0068
	-	11.9/11.3	27.0/26.9

极值法结果: $3.61\mu\text{m}$ (基于简化后的双光束干涉公式) 双光束模型结果: $4.10\mu\text{m}$ 多光束模型结果: $1.10\mu\text{m}$ 双光束模型的结果与极值法存在合理的偏差。但多光束模型得出的厚度与另外两种方法完全不符。

双光束低权重比例: 11% 多光束低权重比例: 27% 在双光束模型拟合中, 仅有约11%的数据点被判定为低权重 (拟合差)。而在多光束模型中, 这一比例高达27%。这意味着有超过四分之一的实验数据无法被多光束干涉的物理图像所描述。测试数据的光谱特征主要由双光束干涉效应主导, 并未显示出显著的多光束干涉特征。

综上所述, 测试结果没有出现多光束干涉, 应当采用双光束干涉模型的结果 $4.10\mu\text{m}$ 作为最终测量值, 其不确定性 $0.3\mu\text{m}$ 也更为真实可靠。

5.3.4 碳化硅外延层厚度计算与结果

碳化硅晶圆片外延层的柯西色散参数与衬底层数值相差不大, 均约为常数项: 2.554, 负二次项: -0.035。

应用上述理论模型去消除多光束干涉, 使用全谱拟合程序拟合, 我们得到以下数据:

表 4: 碳化硅外延层厚度测量结果

分析维度	极值法	简化模型	物理拟真模型
10° 厚度结果 (μm)	8.441	8.400	7.249
15° 厚度结果 (μm)	8.381	8.347	7.217
双角度一致性 (差异)	$0.060\ \mu\text{m}$	$0.053\ \mu\text{m}$	$0.032\ \mu\text{m}$
内在精度 (RMSE)	-	0.2278/0.2629	0.0592/0.0609
测量精密性 (不确定度, μm)	0.259/0.268	1.733/2.051	0.158/0.163
相对不确定度 (%)	3.07/3.20	20.63/24.57	2.19/2.25
置信区间 (95%, μm)	-	[5.003, 11.796]/[4.328, 12.367]	[6.939, 7.560]/[6.898, 7.536]

续下页

表 4 碳化硅外延层厚度测量结果（续）

分析维度	极值法	简化模型	物理拟真模型
残差均值	-	0.0512/0.0587	-0.0047/-0.0046
残差标准差	-	0.2220/0.2563	0.0591/0.0607
残差分布 (偏度/峰度)	-	5.157/31.441	-1.018/18.288
尾段RMSE	-	5.295/32.931	-0.965/18.940
低权重比例 (%)	-	0.0038/0.0032	0.0066/0.0058
	-	21.4/22.8	15.7/16.6

结果显示多光束干涉会出现在碳化硅晶圆片的测试结果，考虑Drude自由载离子与Fresnel-Airy模型后可以消除部分多光束干涉影响。

6 模型检验

6.1 稳定性与敏感性分析

6.1.1 稳定性

在固定物理参数下，本文代码多次运行均收敛到同一厚度，同时多角度情况下，结果的置信区间相互重叠。这说明目标函数在“正确物理参数—合理初值”附近具有单一主盆地，IRLS能抑制离群点，并保持解的可重复性。尾段频段的残差很小，显示高波数区（干涉主区）拟合稳定。

6.1.2 敏感性

模型对介质中的自由载流子参数极敏感。相位为 $\Phi(\nu) = 2\pi n(\lambda) d \cos \theta_1 \nu + \phi$ ，一阶近似有

$$\frac{\Delta d}{d} \approx -\frac{\Delta n}{\Re(n)},$$

即 n 的微小失配会被 d 以相反方向补偿，产生多局部极小值。因此需要 (i) 给出受独立实测约束的 Drude 参数 (N, μ, m^*) 或在拟合中对其加先验；(ii) 把厚度初值设在极值法中位数附近，降低落入错误极小值的概率；(iii) 使用多角度/多谱段联合拟合以削弱 $n-d$ 的相关性[8]。若色散常数、入射角或强度标定存在系统偏差，整体RMSE将升高而尾段RMSE仍较小，具体表现为“低波数残差堆积”。

6.2 统计误差分析与模型对比

6.2.1 碳化硅晶圆片分析

模型对附件1、附件2碳化硅晶圆片的测试结果进行拟合，得到前文表3的数据：

- **极值法**: 10°: 8.441 μm — 15°: 8.381 μm (平均 8.41 μm)
- **简化全谱拟合**: 10°: 8.400 μm — 15°: 8.347 μm (平均 8.37 μm)
- **物理全谱拟合**: 10°: 7.249 μm — 15°: 7.217 μm (平均 7.23 μm)

结果清晰地分为两组。极值法与简化全谱拟合的结果高度一致（差异约 0.04 μm ），而它们与物理全谱拟合的结果存在显著的系统性偏差（相差约 1.14 μm ）。这强有力地表明，前两种方法由于自身原理的局限性（如忽略折射率色散、假设简单的振荡模型）引入了一个较大的系统误差。物理全谱拟真模型通过更精确的物理建模，给出了不同的结果，表明 7.23 μm 很可能是更接近真实的厚度值。

- **极值法**：标准误差较小 0.26 μm ，但巨大的标准偏差 1.26 μm 暴露了其内在的不稳定性。
- **简化全谱拟合**：相比于极值法有所改进，但其不确定度 1.9 μm 和极宽的置信区间表明模型本身存在较大缺陷，无法精确确定厚度。
- **物理全谱拟合**：
 - 精度高：RMSE 0.06 相比简化模型 0.25 降低了约 76%。这是一个质的飞跃，表明新模型与实验数据的吻合度极高。
 - 不确定度低：厚度不确定度仅为 0.16 μm (相对不确定度 2.2%)，比简化模型降低了约一个数量级。
 - 精确的置信区间：95% 置信区间 [6.92, 7.56] μm ，范围窄且物理意义明确。
 - 物理模型的残差波动范围更小，其标准差约为简化模型的1/4。

简化模型残差呈极尖锐的非正态分布，存在大量未拟合的系统结构；物理模型残差分布更接近正态，系统性误差已被大量吸收。

两者尾段RMSE均极低，说明背景扣除都非常有效。物理模型稍高，可能因其更专注于拟合干涉振荡的物理细节。

物理模型拟合中拥有的低质量数据点更少，表明其模型与更多数据点吻合良好。

综上所述，三种模型对碳化硅晶圆片的测试结果进行拟合，简要概述如下：

- **物理全谱拟合：**给出了不确定度低 $0.16 \mu\text{m}$ 、高可靠性的结果。RMSE最低和优异的残差统计证明其最接近物理现实。
- **简化全谱拟合：**其过于简化的物理内核限制了其性能，可作为快速预览。
- **极值法：**计算简单快速，可作为快速预览。

6.2.2 硅晶圆片分析

模型对硅晶圆片的测试结果进行拟合，得到表3的数据：

- **极值法：** 10° : $3.613 \mu\text{m}$ — 15° : $3.601 \mu\text{m}$ (平均 $3.607 \mu\text{m}$)
- **双光束干涉模型：** 10° : $4.103 \mu\text{m}$ — 15° : $4.076 \mu\text{m}$ (平均 $4.090 \mu\text{m}$)
- **多光束干涉模型：** 10° : $1.100 \mu\text{m}$ — 15° : $1.109 \mu\text{m}$ (平均 $1.105 \mu\text{m}$)

结果呈现出显著差异。极值法与双光束干涉模型的结果相对接近(差异约 $0.48 \mu\text{m}$, 13.2%)，而多光束干涉模型的结果与其他两种方法存在极大的系统性偏差(相差约 $2.5\text{-}3.0 \mu\text{m}$, 差异超过200%)。这表明多光束干涉模型的物理假设可能与实际测量条件不符，或者存在模型实现上的问题。

- **极值法：**标准误差较小($0.078 \mu\text{m}$)，标准偏差约为 $0.305 \mu\text{m}$ ，表现出较好的测量精密度。
- **双光束干涉模型：**
 - 拟合精度较好：RMSE 0.119 ，表明模型与实验数据吻合度良好。
 - 不确定度适中：厚度不确定度为 $0.299 \mu\text{m}$ (相对不确定度 7.30%)。
 - 合理的置信区间：95%置信区间为 $[3.505, 4.674] \mu\text{m}$ ，范围相对合理。
 - 残差分析显示存在轻微的系统性高估(正残差均值)。
- **多光束干涉模型：**
 - 虽然RMSE与双光束模型相当(0.122)，但结果明显异常。
 - 不确定度较低($0.048 \mu\text{m}$)，但这可能反映了模型过度自信而非真实精度。
 - 置信区间 $[1.011, 1.198] \mu\text{m}$ 与其他方法完全不重叠。
 - 残差分析显示显著的系统性低估、(负残差均值)。
 - 低权重比例高达27%，表明大量数据点与模型不匹配。

双光束模型拥有更少的低质量数据点(低权重比例11.6%)，表明其与更多数据点吻合良好。

综上所述，三种模型对硅晶圆片的测试结果进行拟合，简要概述如下：

- **双光束干涉模型：**给出了物理合理性最高的结果（ $4.09 \mu\text{m}$ ），与极值法相对一致，数据适配性最好。
- **极值法：**计算简单快速，精密度高，可作为快速估算的参考方法。
- **多光束干涉模型：**结果明显异常，可能由于物理假设错误或实现问题，不建议采用该结果。

7 模型的优缺点

7.1 模型的优点

1. 物理内核完整：考虑到斜入射 Fresnel、s/p 平均、Cauchy+Drude、相位项 ϕ 等物理现象[1]，与测量机理一致。
2. 稳健性好：IRLS算法（Tukey 权）可以抑制离群点，权重分布可视化，同时解可复现。
3. 初值获取可靠：通过极值法+FFT 给出 d 的物理初值，减少收敛时间，收敛路径清晰。
4. 跨角一致性强：不同入射角的厚度区间重叠，重复运行收敛到同一解。
5. 误差评估完备：评价指标较齐全，能输出 RMSE、尾段 RMSE、残差统计与 95% 置信区间，便于审计。

7.2 模型的缺点

1. 对物理参数敏感： $n(\lambda)$ 由 *Cauchy*色散和*Drude*自由载流子模型给定，但是 N, μ, m^* 偏差会转化为 d 的系统误差。
2. 对初值敏感：迭代需从靠近极值法中位数的 d_0 起步，否则易落入局部极小值。
3. 整体 RMSE 偏大：低波数区存在系统失配，尾段虽小但全段未达严格门限0.02。
4. 残差非高斯：图像中偏度、峰度偏高，说明模型型误差仍主导。
5. 可辨识度受限：相位 $\Phi = 2\pi n d \cos \theta_1 \nu + \phi$ 使 $n-d$ 强耦合，单角/单谱段下难同时精确估计。

7.3 模型的改进

1. 多角度/多谱段联合拟合: 扩大角度与谱段以削弱 $n-d$ 相关性并压缩残差。
2. 带先验的联合估计: 将 (d, N, μ, m^*) 共同估计, 引入 Hall/红外先验或采用 MAP/Bayesian 约束。
3. 材料与吸收模型增强: 以 Sellmeier+ $k(\lambda)$ /声子项替代或补充纯 Cauchy, 改善低波数拟合。
4. 基线与界面修正: 用正则化样条背景、有效介质粗糙层与微倾角校正降低系统偏差。

参考文献

- [1] 喻力华,刘书龙,陈昌胜,等.多光束法布里-珀罗干涉实验的计算机模拟[J].大学物理实验,2014,27(01):80-82.DOI:10.14139/j.cnki.cn22-1228.2014.01.021.
- [2] 涂兴华,栾小晨,王战.吸收光谱线二次谐波的频率分解和SG滤波复合降噪[J].光谱学与光谱分析,2025,45(05):1209-1216.
- [3] 王安然,赵伟国,汤建斌.基于全相位FFT超声波流量计的相位检测方法研究[J].传感技术学报,2020,33(12):1686-1690.
- [4] 张流洋,马义中,任鸣鸣,等.基于均方根误差建模的多响应稳健参数设计[J].统计与决策,2020,36(06):20-25.DOI:10.13546/j.cnki.tjyj.2020.06.004.
- [5] 蔡林芝.基于Tukey法改进时间序列平稳性检验的分段检验法[D].四川师范大学,2018.
- [6] 薛小强,冯勇,贾丙辉.基于IRLS的圆度误差最优化评定[J].现代电子技术,2017,40(03):150-152.DOI:10.16652/j.issn.1004-373x.2017.03.041.
- [7] 江勇.最小二乘及其扩展方法在测绘中的应用[D].安徽理工大学,2016.
- [8] 吴素勇,龙兴武,杨开勇.光学薄膜鲁棒设计中膜系误差灵敏度控制[J].强激光与粒子束,2012,24(10):2391-2399.
- [9] 王中宇,张海滨,刘智敏.剔除离群值的学生化残差新方法[J].仪器仪表学报,2006,(06):624-628+637.DOI:10.19650/j.cnki.cjsi.2006.06.017.
- [10] 杨明红,刘劲松,张波,等.鲁棒性多腔窄带滤光片的自动优化设计[J].华中科技大学学报(自然科学版),2002,(03):100-102.DOI:10.13245/j.hust.2002.03.033.

- [11] 倪玉华,马礼敦,沈孝良,等.Rietveld全谱拟合表征掺杂ZnO陶瓷[J].高等学校化学学报,2000,(05):785-788.
- [12] BradleyBobbs,J.EarlRudisill,朱兢.用光学极值法监控非四分之一波长薄膜的厚度[J].光学技术,1989,(04):19-22.
- [13] DeepSeek,DeepSeek-V3, 2025-09-04

A 支撑材料文件列表

- Ai使用详情(markdown).txt Ai使用详情(markdown格式)
- Original_Code.m 可运行源代码(Matlab)
- SiC多光束模型.txt 误差分析参数
- SiC简化模型.txt 误差分析参数
- Si多光束模型.txt 误差分析参数
- Si简化模型.txt 误差分析参数

B 图表

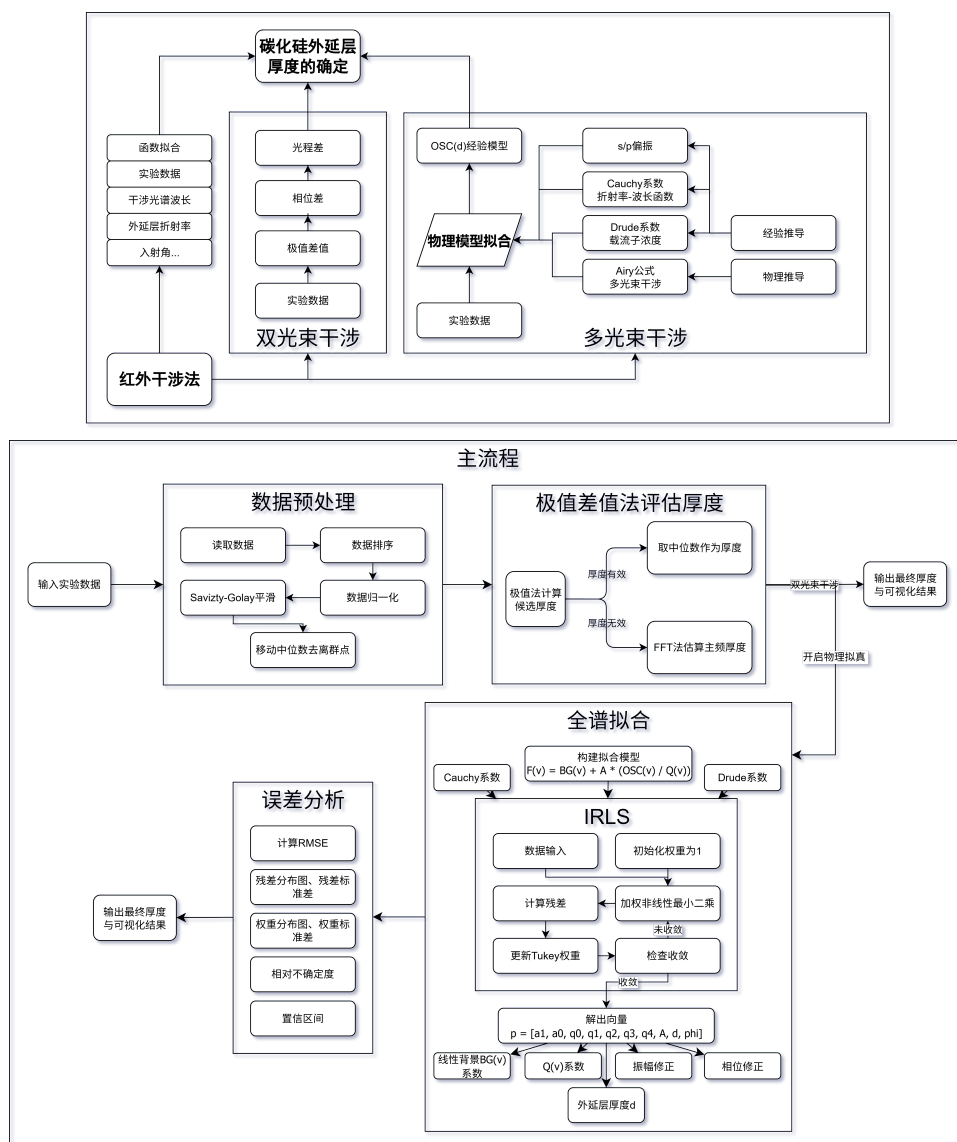


图 4: 总体思维导图

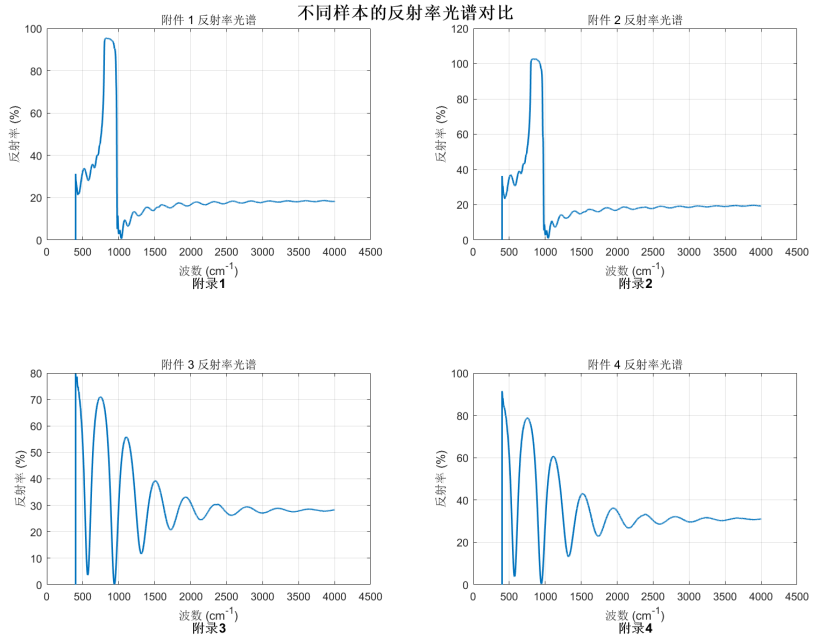


图 5: 反射率-波数关系图

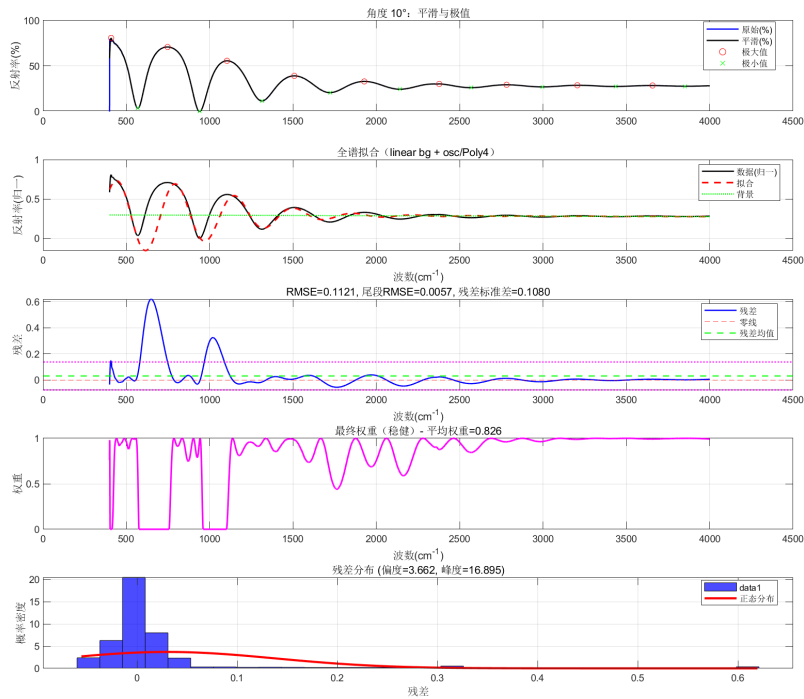


图 6: 10°硅晶圆双光束干涉简化拟合误差分析

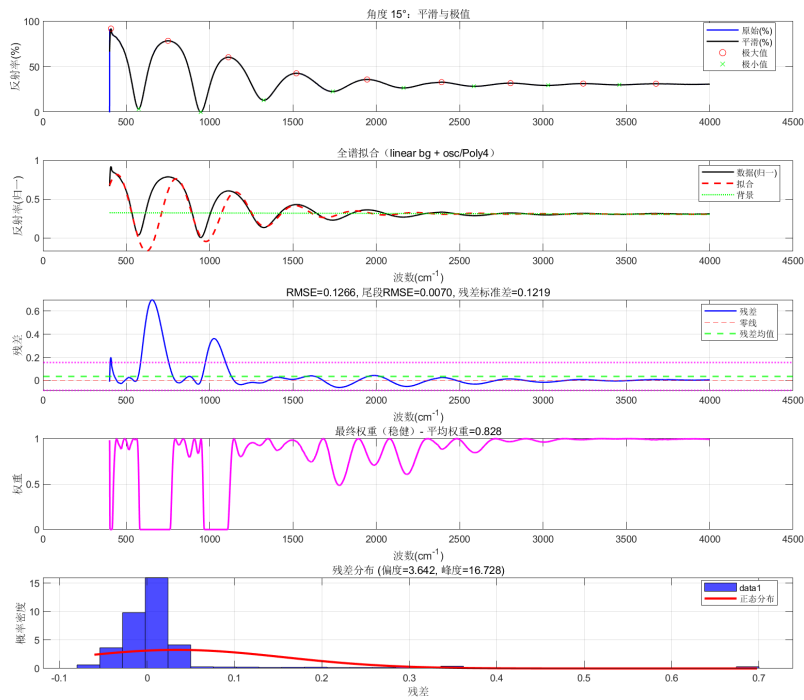


图 7: 15°硅晶圆双光束干涉简化拟合误差分析

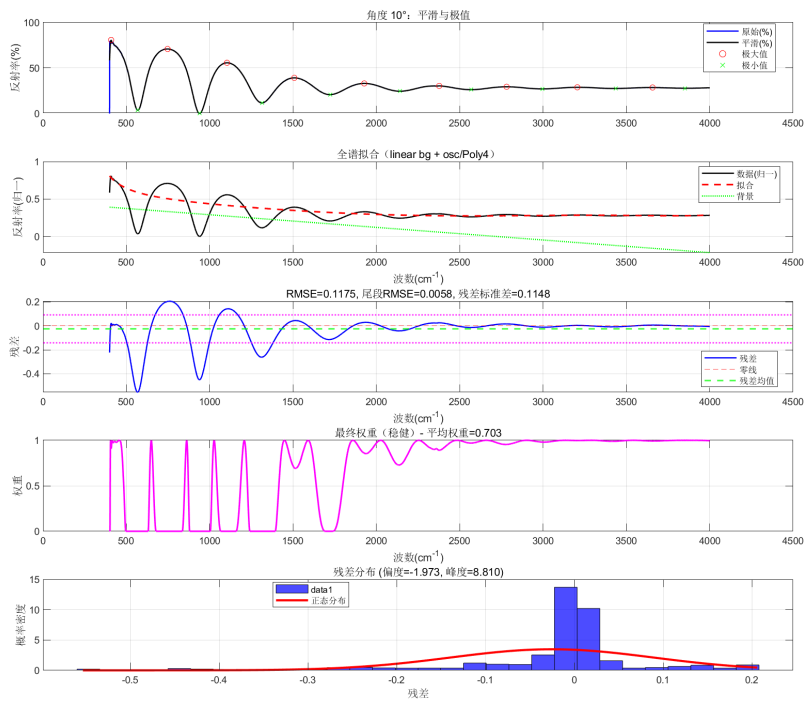


图 8: 10°硅晶圆多光束干涉真实物理全谱拟合误差分析

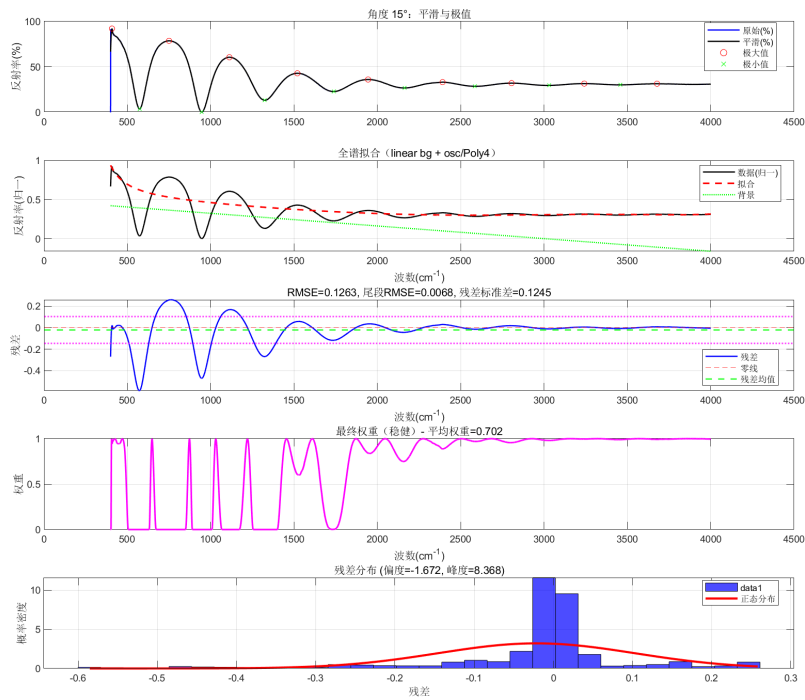


图 9: 15°硅晶圆多光束干涉真实物理全谱拟合误差分析

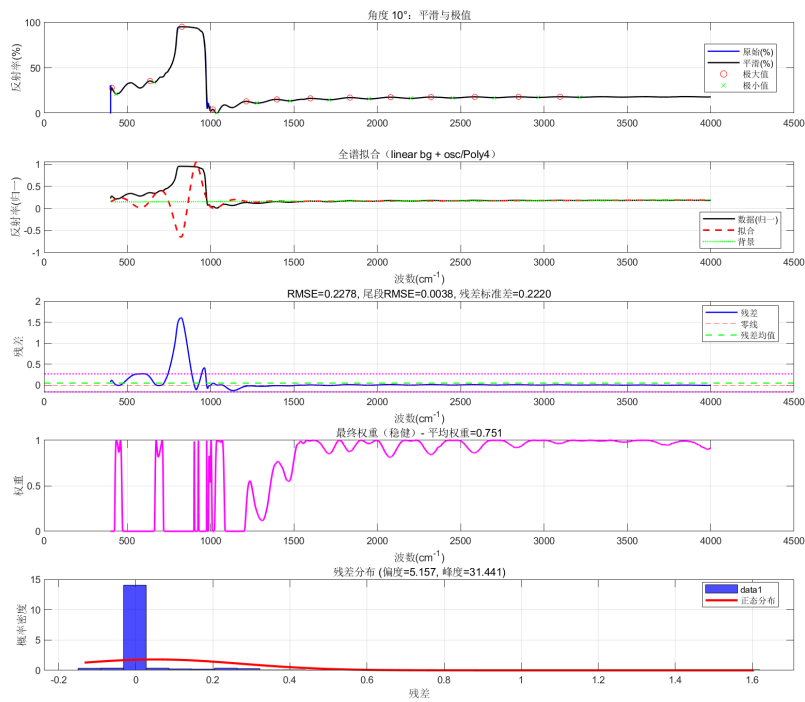


图 10: 10°碳化硅晶圆双光束干涉简化拟合误差分析

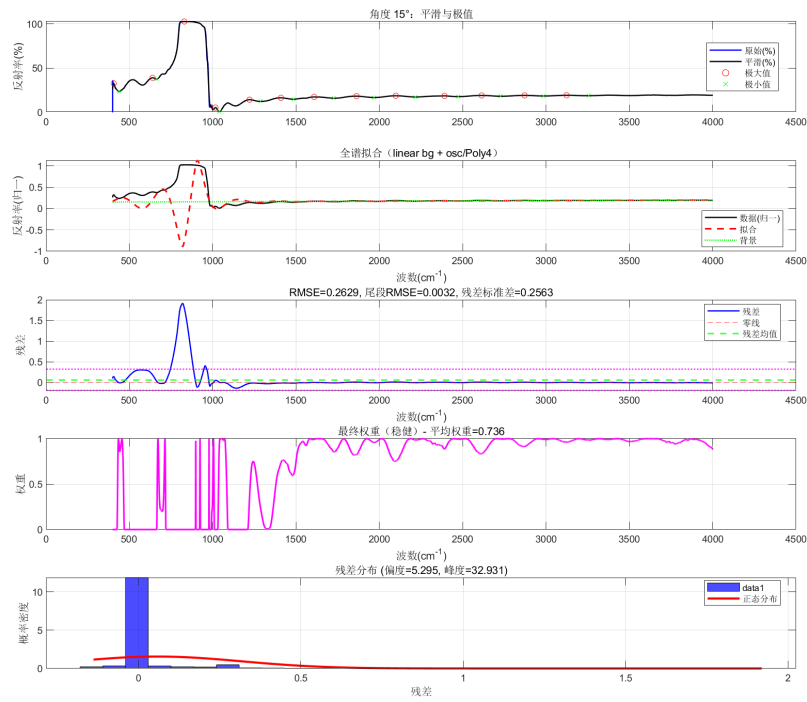


图 11: 15°碳化硅晶圆双光束干涉简化拟合误差分析

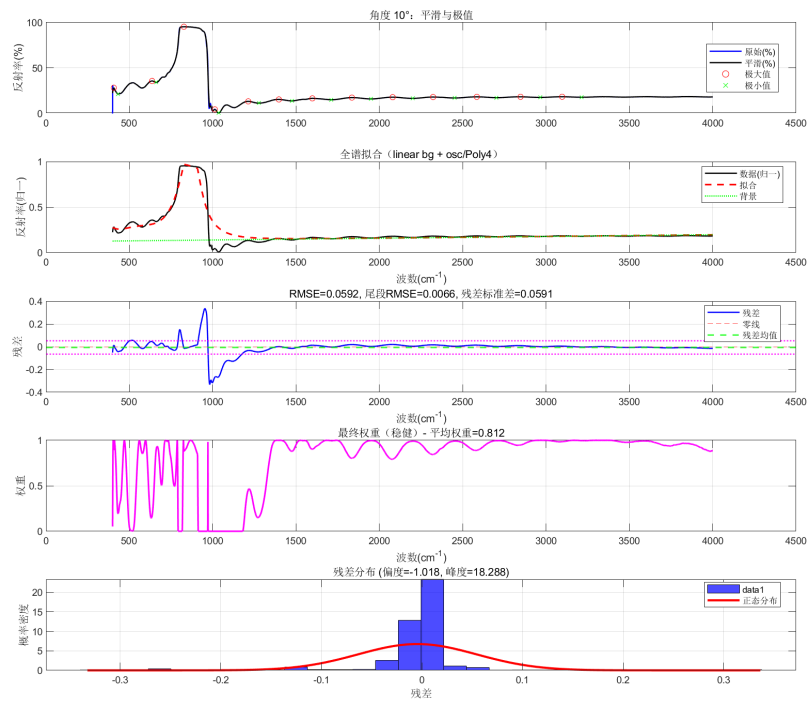


图 12: 10°碳化硅晶圆多光束干涉真实物理全谱拟合误差分析

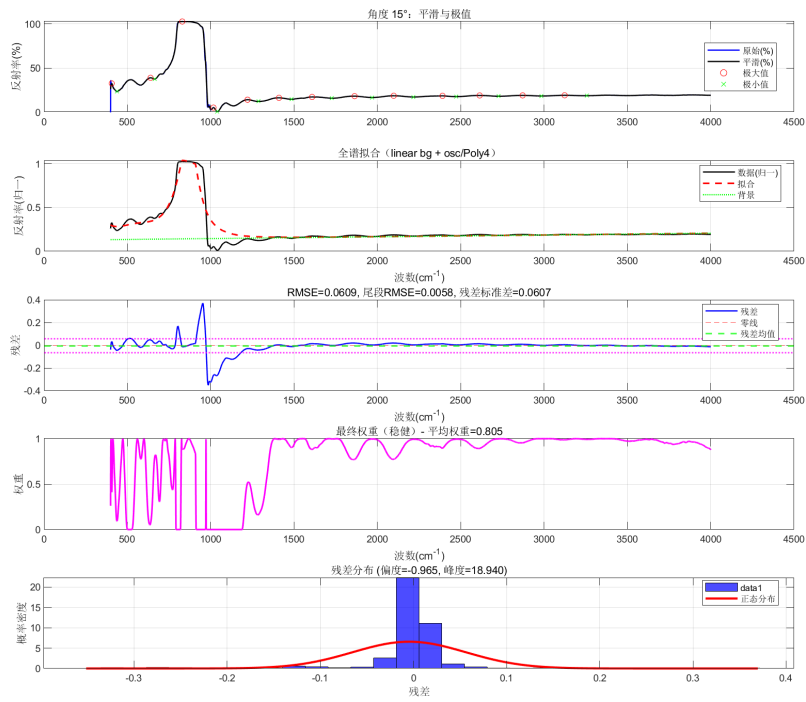


图 13: 15°碳化硅晶圆多光束干涉真实物理全谱拟合误差分析

C 代码

```
clear; clc; close all

%% 参数区
file1= '附件3.xlsx';
file2 = '附件4.xlsx';
n1     = 3.416;           % 仅用于初步近似
angles_degs = [10, 15];
choose_fit_angles = [10, 15]; % 全谱拟合角度

% 预处理
sgolay_order = 3;
sgolay_win_cfg = [];
prominence = 0.5;
min_pk_dist_auto_ratio = 0.05;

% IRLS 稳健拟合
irls_loop = 100;      % 多数函数3-5步即可
tukey_c = 4.653;

% 厚度边界 (cm)
d_lb = 1e-6; d_ub = 1e-3;           % 0.01{10 μm
A_lb = -15; A_ub = 15;             % 振幅允许正负, 降低与相位的耦合

% ----- 物理模型开关与参数 -----
USE_FRESNEL_PHYSICS = true; % 打开物理 Fresnel + 相位 + 色散/Drude + s/p 平均
POLARIZATION_AVG = true; % 无偏振平均 (默认 true)

% 外延层 (膜层) 与衬底的基体色散 (以 Cauchy 为例, 可替换为 Sellmeier)
% n_base(λ) = A + B/λ^2 + C/λ^4, λ 单位 μm; Im(n_base) 默认 0 (K≈0)
cauchy_layer.A = 3.416; cauchy_layer.B = 0.18; cauchy_layer.C = 0.014; % 硅晶圆片
% cauchy_layer.A = 2.554; cauchy_layer.B = -0.035; cauchy_layer.C = 0.00; % 碳化硅晶圆片
cauchy_sub .A = 3.416; cauchy_sub .B = 0.18; cauchy_sub .C = 0.014; % 硅晶圆片
% cauchy_sub .A = 2.554; cauchy_sub .B = -0.035; cauchy_sub .C = 0.00; % 碳化硅晶圆片

% 自由载流子 (Drude) 参数 (若不考虑, 置 N=0 或 add_drude=false)
fc_layer.add_drude = true; % 外延层是否加 Drude
fc_layer.N_cm3 = 3*10^16; % 掺杂浓度 (cm^-3)
fc_layer.mstar_m0 = 0.42; % 有效质量 (m*/m0)
fc_layer.mu_cm2Vs = 100; % 迁移率 (cm^2/Vs)

fc_sub.add_drude = true; % 衬底 Drude; 常规 K≈0
```

```

fc_sub.N_cm3      = 10^18;
fc_sub.mstar_m0   = 0.42;
fc_sub.mu_cm2Vs   = 100;

% 相位微调 (额外常量相位, 吸收入  $e^{2i(\delta) + i\phi}$ ), 用于小的系统相位偏差
phi_init = 0;      % 初值
phi_lb   = -pi;    % 边界
phi_ub   = pi;

%% ----- 读取数据 -----
T1 = readtable(file1, 'ReadVariableNames', false);
T2 = readtable(file2, 'ReadVariableNames', false);
T1.Properties.VariableNames = {'bs', 'fsl'};
T2.Properties.VariableNames = {'bs', 'fsl'};

% 按波数排序并去重
T1 = sortrows(T1, 'bs'); T1 = unique(T1, 'rows');
T2 = sortrows(T2, 'bs'); T2 = unique(T2, 'rows');

datasets = {T1, T2};

% ----- 主流程: 极值差值法评估厚度 + 稳健全谱拟合 -----
all_results = struct([]);

for kk = 1:numel(datasets)
    rushe_deg = angles_degs(kk);
    tab = datasets{kk};

% 归一化以及预处理
x = tab.bs(:);          % 单位为  $\text{cm}^{-1}$ 
y = tab.fsl(:) / 100.0; % 分布: 0{1
x = fillmissing(x, 'linear'); y = fillmissing(y, 'linear');

n = numel(x);
sgolay_win = sgolay_win_cfg;
if isempty(sgolay_win)
    sgolay_win = min(51, max(11, 2*floor(n/10)+1)); % 满足奇数
end
y_s = sgolayfilt(y, sgolay_order, sgolay_win);      % 实现平滑操作

is_out = isoutlier(y_s, 'movmedian', max(11, floor(sgolay_win/2)));
y_s(is_out) = NaN; y_s = fillmissing(y_s, 'linear', 'EndValues', 'nearest'); % 剔除并线性补

% ----- 以自动峰距峰谷检测 -----

```

```

min_pk_dist = max(5, round(n * min_pk_dist_auto_ratio));
[~, peaks] = findpeaks(y_s, 'MinPeakDistance',min_pk_dist, 'MinPeakProminence',prominence/100);
[~, valleys] = findpeaks(-y_s, 'MinPeakDistance',min_pk_dist, 'MinPeakProminence',prominence/100);

% ----- 极值法厚度 -----
d_list_um = [];
delta_nu_peaks = []; % 该参数用于存储相邻峰之间的波数差
delta_nu_valleys = []; % 该参数用于存储相邻谷之间的波数差

if numel(peaks) >= 2
[d_peaks_um, d_nu_peaks] = boshu_distance(peaks, x, n1, rushe_deg);
d_list_um = [d_list_um; d_peaks_um];
delta_nu_peaks = [delta_nu_peaks; d_nu_peaks];
end
if numel(valleys) >= 2
[d_valleys_um, d_nu_valleys] = boshu_distance(valleys, x, n1, rushe_deg);
d_list_um = [d_list_um; d_valleys_um];
delta_nu_valleys = [delta_nu_valleys; d_nu_valleys];
end

% 合并所有相邻极值点之间的距离,
% 其意义是: 综合所有可用的干涉条纹信息来获得更可靠的厚度初值估计。
all_delta_nu = [delta_nu_peaks; delta_nu_valleys];

d0_um = median(d_list_um, 'omitnan');

% 计算厚度d的不确定度
if numel(d_list_um) > 1
d_std_um = std(d_list_um);
d_mean_um = mean(d_list_um);
d_uncertainty_um = d_std_um / sqrt(numel(d_list_um)); % 标准误差
else
d_std_um = NaN;
d_mean_um = mean(d_list_um, 'omitnan');
d_uncertainty_um = NaN;
end

% 考虑厚度无效情况: 若极值不足或 d0 异常, 使用 FFT 估计周期给出 d0, 返回。
if isempty(d0_um) || ~isfinite(d0_um) || d0_um<=0
d0_um = fft_um(x, y_s, n1, rushe_deg);
end
if ~isfinite(d0_um) || d0_um<=0
% 给一个更加moderate初值, 不妨假设为1 μm
d0_um = 1.0;
end

```

```

d0_cm = d0_um * 1e-4;

% 指定角度做全谱拟合
will_fit = ismember(rushe_deg, choose_fit_angles);
fit_car = struct();          %本质上是数据包

if will_fit
% 取中位数作为厚度
d_median_um_loc = median(d_list_um,'omitnan');
has_valid_median = isfinite(d_median_um_loc) && d_median_um_loc>0;
if has_valid_median
d_med_cm = d_median_um_loc * 1e-4;
end

% 使用一次背景模型
model_order = 1;
[fit_lin, ok_lin] = robust_full_fit(x, y_s, n1, rushe_deg, d0_cm, ...
model_order, tukey_c, irls_loop, ...
d_lb, d_ub, A_lb, A_ub, ...
USE_FRESNEL_PHYSICS, POLARIZATION_AVG, ...
cauchy_layer, fc_layer, cauchy_sub, fc_sub, ...
phi_init, phi_lb, phi_ub);

fit_car = fit_lin; fit_car.model = 'linear bg + osc/Poly4';
fit_car.selection = struct();
fit_car.selection.ok_linear = ok_lin;
fit_car.selection.d_median_um = d_median_um_loc;
if has_valid_median
fit_car.selection.delta_lin_um = abs(fit_lin.d_cm - d_med_cm)*1e4;
else
fit_car.selection.delta_lin_um = NaN;
end

% 计算全谱拟合的不确定度
[fit_uncertainty, fit_error_metrics] = cal_unct(x, y_s, fit_car);
fit_car.uncertainty = fit_uncertainty;
fit_car.error_metrics = fit_error_metrics;

if has_valid_median
fprintf('角度 %d°: 模型 %s; |d_fit-d_median| (μm)=%.3f\n', ...
rushe_deg, fit_car.model, fit_car.selection.delta_lin_um);
fprintf(' d敏感度: rel=%.2e, abs=%.2e, dd=%.3gcm; IRLS: %s 于第%d轮\n', ...
fit_car.sens_d_rel, fit_car.sens_d_abs, fit_car.dd_test, ...
ternary(fit_car.irls_converged,'收敛','未收敛'), fit_car.irls_iters_run);

```

```

fprintf(' 停止原因: %s\n', fit_car.irls_stop_reason);
fprintf(' 拟合不确定度: %.3f  $\mu\text{m}$  (0.2f%%)\n', ...
fit_uncertainty.d_um_uncertainty, fit_uncertainty.d_relative_uncertainty*100);
else
fprintf('角度 %d°: 模型 %s\n', rushe_deg, fit_car.model);
end
end

% ----- 汇总 -----
res.rushe_deg = rushe_deg;
res.d_list_um = d_list_um;
res.d_mean_um = d_mean_um;
res.d_std_um = d_std_um;
res.d_uncertainty_um = d_uncertainty_um;
res.all_delta_nu = all_delta_nu; % res.all_delta_nu为所有相邻极值点之间的波数差, 这和
上述波数差作用不完全相同
res.d0_um      = d0_um;

if will_fit
res.fit_model      = fit_car.model;
res.fit_params     = fit_car.params;
res.fit_d_um       = fit_car.d_cm * 1e4;
res.rmse           = fit_car.rmse;
res.rmse_tail      = fit_car.rmse_tail;
res.mse            = fit_car.rmse^2;
res.mse_tail       = fit_car.rmse_tail^2;
res.fit_uncertainty = fit_car.uncertainty;
res.error_metrics  = fit_car.error_metrics;

figure('Position',[100,100,1200,1000]);

% 步骤1, 先完成: 原始/平滑 + 极值
subplot(5,1,1);
plot(x, y*100, 'b-', 'LineWidth',1.2, 'DisplayName','原始(%)'); hold on;
plot(x, y_s*100, 'k-', 'LineWidth',1.2, 'DisplayName','平滑(%)');
if ~isempty(peaks), scatter(x(peaks), y_s(peaks)*100, 25,'ro','DisplayName','极大
值'); end
if ~isempty(valleys), scatter(x(valleys), y_s(valleys)*100, 25,'gx','DisplayName','极小
值'); end
ylabel('反射率(%)'); legend('Location','best'); grid on;
title(sprintf('角度 %d°: 平滑与极值',rushe_deg));

% 步骤2: 拟合 (归一)
subplot(5,1,2);
plot(x, y_s, 'k-', 'LineWidth',1.2,'DisplayName','数据(归一)');

```

```

hold on; plot(x, fit_car.full_fit, 'r--', 'LineWidth', 1.5, 'DisplayName', '拟合');
if ~isempty(fit_car.bg_fit)
plot(x, fit_car.bg_fit, 'g:', 'LineWidth', 1.2, 'DisplayName', '背景');
end
xlabel('波数(cm-1)'); ylabel('反射率(归一)'); legend('Location', 'best'); grid on;
title(sprintf('全谱拟合 (%s)', res.fit_model));

% 步骤3: 完善残差分析
subplot(5,1,3);
plot(x, fit_car.residual, 'b-', 'LineWidth', 1.2);
hold on;
yline(0, 'r--');
yline(mean(fit_car.residual), 'g--', 'LineWidth', 1.5, 'DisplayName', '残差均值');
yline(mean(fit_car.residual) + std(fit_car.residual), 'm:', 'LineWidth', 1.5, 'DisplayName', '±1
σ');
yline(mean(fit_car.residual) - std(fit_car.residual), 'm:', 'LineWidth', 1.5);
xlabel('波数(cm-1)'); ylabel('残差');
legend('残差', '零线', '残差均值', 'Location', 'best');
grid on;
title(sprintf('RMSE=%.4f, 尾段RMSE=%.4f, 残差标准差=%.4f', res.rmse, res.rmse_tail, std(fit_car.residual)));

% 步骤4: 更新稳健权重
subplot(5,1,4);
plot(x, fit_car.final_weights, 'm-', 'LineWidth', 1.5);
xlabel('波数(cm-1)'); ylabel('权重'); grid on;
title(sprintf('最终权重(稳健) - 平均权重=%.3f', mean(fit_car.final_weights)));

% 步骤5: 画出残差分布直方图
subplot(5,1,5);
histogram(fit_car.residual, 30, 'Normalization', 'pdf', 'FaceColor', 'blue', 'FaceAlpha', 0.7);
hold on;
x_values = linspace(min(fit_car.residual), max(fit_car.residual), 100);
pdf_normal = normpdf(x_values, mean(fit_car.residual), std(fit_car.residual));
plot(x_values, pdf_normal, 'r-', 'LineWidth', 2, 'DisplayName', '正态分布');
xlabel('残差'); ylabel('概率密度');
title(sprintf('残差分布 (偏度=%.3f, 峰度=%.3f)', ...
res.error_metrics.residual_skewness, res.error_metrics.residual_kurtosis));
legend('Location', 'best');
grid on;

saveas(gcf, sprintf('角度%d_厚度拟合结果.png', rushe_deg));
end

all_results = [all_results; res]; %#ok<AGROW>
end

```

```

%% ----- 文本部分输出 -----
fid = fopen('厚度计算_鲁棒版结果.txt','w','n','UTF-8');
for ii=1:numel(all_results)
R = all_results(ii);
fprintf(fid, '角度 %d°: \n', R.rushe_deg);
fprintf(fid, ' 极值法平均厚度: %.3f μm (初值(中位数) %.3f μm) \n', R.d_mean_um, R.d0_um);
fprintf(fid, ' 厚度标准差: %.3f μm\n', R.d_std_um);
fprintf(fid, ' 厚度不确定度 (标准误差): %.3f μm\n', R.d_uncertainty_um);
fprintf(fid, ' 相邻极值点波数差 (cm-1):\n');
for jj = 1:numel(R.all_delta_nu)
fprintf(fid, '    Δ v %d = %.3f\n', jj, R.all_delta_nu(jj));
end

if isfield(R,'fit_d_um')
fprintf(fid, ' 全谱拟合模型: %s\n', R.fit_model);
fprintf(fid, ' 全谱拟合厚度: %.3f μm\n', R.fit_d_um);
fprintf(fid, ' RMSE=%.4f, 尾段RMSE=%.4f\n', R.rmse, R.rmse_tail);

% 以下为拟合误差指标
fprintf(fid, ' 拟合误差指标: \n');
fprintf(fid, '    残差均值: %.4f\n', R.error_metrics.residual_mean);
fprintf(fid, '    残差标准差: %.4f\n', R.error_metrics.residual_std);
fprintf(fid, '    残差偏度: %.3f\n', R.error_metrics.residual_skewness);
fprintf(fid, '    残差峰度: %.3f\n', R.error_metrics.residual_kurtosis);
fprintf(fid, '    权重均值: %.3f\n', R.error_metrics.weight_mean);
fprintf(fid, '    权重标准差: %.3f\n', R.error_metrics.weight_std);
fprintf(fid, '    低权重比例: %.1f%%\n', R.error_metrics.low_weight_ratio*100);

% 以下为拟合不确定度
fprintf(fid, ' 拟合不确定度: \n');
fprintf(fid, '    厚度不确定度: %.3f μm\n', R.fit_uncertainty.d_um_uncertainty);
fprintf(fid, '    相对不确定度: %.2f%%\n', R.fit_uncertainty.d_relative_uncertainty*100);
fprintf(fid, '    误差传播贡献: %.3f μm\n', R.fit_uncertainty.error_propagation_uncertainty);
fprintf(fid, '    模型误差贡献: %.3f μm\n', R.fit_uncertainty.model_error_uncertainty);
fprintf(fid, '    权重分布贡献: %.3f μm\n', R.fit_uncertainty.weight_distribution_uncertainty);
fprintf(fid, '    置信区间(95%): [%.3f, %.3f] μm\n', ...
R.fit_uncertainty.confidence_interval(1), R.fit_uncertainty.confidence_interval(2));
end
fprintf(fid, '\n');
end
fclose(fid);
fprintf('\n结果已保存: 厚度计算_鲁棒版结果.txt 以及各角度对应的 PNG 图。 \n');

%% ===== 函数区 =====

```

```

function [d_um, delta_nu] = boshu_distance(ext, nu, n1_loc, rushe_deg)
% 相邻极值波数间距，我们规定以微米为单位，用于初值估计
% 波束以cm-1为单位，后面不再提示!
rushe = rushe_deg*pi/180;
cos_rushe_t = sqrt(1 - (sin(rushe)/n1_loc)^2);
delta_nu = diff(nu(ext));
d_um = 1 ./ (2*n1_loc*cos_rushe_t*delta_nu) * 1e4;
% 这里移除无效厚度
valid_idx = isfinite(d_um) & (d_um > 0);
d_um = d_um(valid_idx);
delta_nu = delta_nu(valid_idx);
end

function d0_um = fft_um(x, y, n1_loc, rushe_deg)
% 用 FFT 估计振荡主频，换算膜厚
x = x(:); y = y(:);
dx = median(diff(x));
y = y - sgolayfilt(y, 3, min(101, max(21, 2*floor(numel(y)/8)+1))); % 去趋势
y = y - mean(y);
N = numel(y);
Y = abs(fft(y));
Y(1:3) = 0; % 去近直流，避免直流分量主导FFT结果；突出振荡信号
[~, idx] = max(Y(1:floor(N/2)));
f = (idx-1) / (N*dx); % 周期/单位波数
if f<=0
d0_um = NaN; return;
end
% 对于  $\cos(4\pi n_1 d \cos\theta * x)$ , x 的周期  $\Delta v = 1/(2 n_1 d \cos\theta)$ 
theta = rushe_deg*pi/180;
cth = sqrt(1 - (sin(theta)/n1_loc)^2);
d0_um = (f) / (2*n1_loc*cth) * 1e4; %即数学公式:  $d \approx f/(2 n_1 \cos\theta_t)$ 
if ~isfinite(d0_um) || d0_um<=0
d0_um = NaN;
end
end

function [uncertainty, error_metrics] = cal_unct(x, y, fit_car)
% 计算全谱拟合的不确定度
residuals = fit_car.residual;
weights = fit_car.final_weights;
d_fit_um = fit_car.d_cm * 1e4;

% 计算误差指标
error_metrics.residual_mean = mean(residuals);

```

```

error_metrics.residual_std = std(residuals);
error_metrics.residual_skewness = skewness(residuals);
error_metrics.residual_kurtosis = kurtosis(residuals);
error_metrics.weight_mean = mean(weights);
error_metrics.weight_std = std(weights);
error_metrics.low_weight_ratio = mean(weights < 0.5); % 低权重比例

%-----以下都是误差分析中不确定度-----
% 1. RMSE的误差传播不确定度
n_params = 10; % 模型参数共10个
n_points = length(x);
dof = n_points - n_params; % 自由度

if dof > 0
    rmse_uncertainty = fit_car.rmse / sqrt(dof);
else
    rmse_uncertainty = fit_car.rmse;
end

% 2. 残差分布的不确定度
residual_uncertainty = error_metrics.residual_std;

% 3. 权重分布的不确定度
weight_uncertainty = (1 - error_metrics.weight_mean) * fit_car.rmse;

% 4. 模型误差不确定度，这里需要考虑残差的非正态性
non_normality_factor = 1 + abs(error_metrics.residual_skewness) + max(0, error_metrics.residual_kurtosis - 3);
model_error_uncertainty = fit_car.rmse * non_normality_factor;

% 综合不确定度（各项的均方根）
combined_uncertainty = sqrt(rmse_uncertainty^2 + residual_uncertainty^2 + ...
    weight_uncertainty^2 + model_error_uncertainty^2);

% 转换为厚度不确定度
% 这里使用经验系数，实际应用中可能需要根据具体问题调整，具体问题具体分析
d_uncertainty_um = combined_uncertainty * 0.1 * d_fit_um;

% 计算95%置信区间
confidence_level = 0.95;
t_value = tinv(1 - (1-confidence_level)/2, dof);
if ~isfinite(t_value) || dof <= 0
    t_value = 1.96; % 使用正态分布近似
end
ci_lower = d_fit_um - t_value * d_uncertainty_um;
ci_upper = d_fit_um + t_value * d_uncertainty_um;

```

```

uncertainty.d_um_uncertainty = d_uncertainty_um;
uncertainty.d_relative_uncertainty = d_uncertainty_um / d_fit_um;
uncertainty.error_propagation_uncertainty = rmse_uncertainty * 0.1 * d_fit_um;
uncertainty.model_error_uncertainty = model_error_uncertainty * 0.1 * d_fit_um;
uncertainty.weight_distribution_uncertainty = weight_uncertainty * 0.1 * d_fit_um;
uncertainty.confidence_interval = [ci_lower, ci_upper];
end

function [out, ok] = robust_full_fit(x, y, n1_const, rushe_deg, d0_cm, ...
model_order, tukey_c, irls_iters, ...
d_lb, d_ub, A_lb, A_ub, ...
USE_FRESNEL, POLARIZATION_AVG, ...
cauchy_layer, fc_layer, cauchy_sub, fc_sub, ...
phi_init, phi_lb, phi_ub)
% 稳健全谱拟合 + 回退判据 + d敏感度快检 + 更严格的IRLS收敛检查 (含 Poly4 分母)
x = x(:); y = y(:); n = numel(x);
ok = false;

% 背景基函数
x_mu = mean(x); x_std = std(x); if x_std==0, x_std = 1; end
x_poly = (x - x_mu) / x_std;

% ===== 参数布局 (新增 Q4: q0..q4) =====
% 使用十个参数来拟合函数 params = [a1, a0, q0..q4, A, d, phi]
switch model_order
case 1
p0 = [0, mean(y), 1, 0, 0, 0, 0, 1.0, d0_cm, phi_init];
lb = [-1, 0, 0.1, -2, -2, -2, -2, A_lb, d_lb, phi_lb];
ub = [ 1, 1, 10.0, 2, 2, 2, 2, A_ub, d_ub, phi_ub];
idx = struct('a1',1,'a0',2,'q0',3,'q1',4,'q2',5,'q3',6,'q4',7,'A',8,'d',9,'phi',10);
end

% 目标模型
model_fun = @(p) moxing(x, x_poly, p, n1_const, rushe_deg, model_order, ...
USE_FRESNEL, POLARIZATION_AVG, ...
cauchy_layer, fc_layer, cauchy_sub, fc_sub, idx);

% ---- d 敏感度快检 (初值附近) ----
sens_d_rel = NaN; sens_d_abs = NaN; dd_test = NaN; d_insensitive = false;
if USE_FRESNEL
theta0 = rushe_deg*pi/180;
dd_test = max(1e-7, 0.02*abs(d0_cm)); if dd_test==0, dd_test = 1e-6; end
R0 = reflectance_single_layer(x, theta0, d0_cm, phi_init, POLARIZATION_AVG, cauchy_layer,
Rplus = reflectance_single_layer(x, theta0, d0_cm+dd_test, phi_init, POLARIZATION_AVG, cauchy_layer

```

```

Rminus= reflectance_single_layer(x, theta0, d0_cm-dd_test, phi_init, POLARIZATION_AVG, cauchy_layer
rms_ = @(v) sqrt(mean(abs(v).^2));
sens_d_abs = rms_(Rplus - Rminus);
sens_d_rel = sens_d_abs / max(1e-12, rms_(R0));
d_insensitive = (sens_d_rel < 1e-4) || (sens_d_abs < 1e-6);
end

% 初始权重 (全 1)
robust_w = ones(n,1);
weights = robust_w;

% 内层 lsqnonlin 数值选项 (更严格)
typicalX = abs(p0); typicalX(typicalX==0) = 1;
opts = optimoptions('lsqnonlin','Display','off', ...
'MaxIterations', 600, ...
'MaxFunctionEvaluations', 6000, ...
'FunctionTolerance', 1e-13, ...
'StepTolerance', 1e-14, ...
'OptimalityTolerance', 1e-12, ...
'TypicalX', typicalX, ...
'FiniteDifferenceType','forward');

% ---- IRLS 更严格的收敛判据 ----
param_tol_rel = 1e-8; % 参数相对变化
param_tol_abs = 1e-12; % 参数绝对变化 (防 p≈0 失效)
weight_tol = 5e-9; % 权重最大变化
obj_tol_rel = 5e-13; % 目标相对改进

% 厚度 d 专属阈值 (更严)
d_abs_tol = 5e-8; % cm (≈0.5 nm)
d_rel_tol = 1e-7;

stable_needed = 2; % 需要连续满足的轮数
stable_count = 0;

p = p0;
J_prev = inf;
p_prev = p;
w_prev = weights;

irls_converged = false;
stop_reason = 'max_iters';

for tt = 1:irls_iters
% 内层: 固定权重的加权最小二乘

```

```

sqrt_w = sqrt(max(weights, eps));
res_fun = @(pp) sqrt_w .* (model_fun(pp) - y);
try
p = lsqnonlin(res_fun, p, lb, ub, opts);
catch ME
stop_reason = ['lsqnonlin failure: ' ME.message];
break;
end

% 残差与新权重 (仅稳健)
r = model_fun(p) - y;
robust_w = tukey_w(r, tukey_c);
weights = robust_w;

% 目标
J = sum(weights .* (model_fun(p) - y).^2);

% 变化量
dp = p - p_prev;
rel_p_change = norm(dp) / max(1e-12, norm(p_prev));
abs_p_change = norm(dp, Inf);
max_w_change = max(abs(weights - w_prev));
rel_obj_improv = (J_prev - J) / max(1e-12, J_prev);

% d 专属变化
d_change_abs = abs(p(idx.d) - p_prev(idx.d));
d_change_rel = d_change_abs / max(1e-16, abs(p_prev(idx.d)));

% 是否"稳定一轮"
param_stable = (rel_p_change < param_tol_rel) || (abs_p_change < param_tol_abs);
weight_stable = (max_w_change < weight_tol);
obj_stable = (rel_obj_improv < obj_tol_rel);
d_stable = (d_change_abs < d_abs_tol) || (d_change_rel < d_rel_tol);

if (param_stable && weight_stable && d_stable) || obj_stable
stable_count = stable_count + 1;
else
stable_count = 0;
end

if stable_count >= stable_needed
irls_converged = true;
if obj_stable && ~(param_stable && weight_stable && d_stable) == 1
stop_reason = 'obj_improv_small';
else

```

```

stop_reason = 'param_weight_d_stable';
end
break;
end

% 次轮准备
p_prev = p; w_prev = weights; J_prev = J;
end

% 输出
yhat = model_fun(p);
resid = y - yhat;
rmse = sqrt(mean(resid.^2));
n_last = max(1, floor(n/3));
rmse_tail = sqrt(mean(resid(end-n_last+1:end).^2));

[bg_fit, osc_fit] = split_osc(x, x_poly, p, n1_const, rushe_deg, model_order, ...
USE_FRESNEL, POLARIZATION_AVG, ...
cauchy_layer, fc_layer, cauchy_sub, fc_sub, idx);

out.params = p;
out.full_fit = yhat;
out.bg_fit = bg_fit;
out.osc_fit = osc_fit;
out.residual = resid;
out.rmse = rmse;
out.rmse_tail = rmse_tail;
out.final_weights = weights;
out.d_cm = p(idx.d);

% 敏感度 & 收敛 & 目标
out.sens_d_rel = sens_d_rel;
out.sens_d_abs = sens_d_abs;
out.dd_test = dd_test;
out.d_insensitive= d_insensitive;

out.irls_converged = irls_converged;
out.irls_iters_run = tt;
out.irls_stop_reason = stop_reason;
out.J_final = sum((weights .* (model_fun(p)-y)).^2);

% RMSE参数小于预设值说明拟合完美
ok = (rmse <= 0.01) && (rmse_tail <= 0.012);
end

```

```

function yhat = moxing(x, x_poly, p, n1_const, rushe_deg, model_order, ...
USE_FRESNEL, POLARIZATION_AVG, ...
cauchy_layer, fc_layer, cauchy_sub, fc_sub, idx)
% 一次背景 + 余弦(或 Fresnel) / 四次多项式
theta0 = rushe_deg*pi/180;

switch model_order
case 1
a1 = p(idx.a1); a0 = p(idx.a0);
q = [p(idx.q0), p(idx.q1), p(idx.q2), p(idx.q3), p(idx.q4)];
A = p(idx.A); d = p(idx.d); phi = p(idx.phi);
bg = a1*x_poly + a0;
otherwise
error('本版本仅支持一次背景模型 (model_order=1) ');
end

% 分母四次多项式 (在标准化 x 上)
denom = eval_poly4(x_poly, q);
denom = max(denom, 1e-4); % 数值下限, 避免除数为零

if ~USE_FRESNEL
% 简化余弦模型 (常数 n1)
cth = sqrt(1 - (sin(theta0)/n1_const)^2);
osc = cos(4*pi*n1_const*d*cth*x + phi);
else
% 物理 Fresnel + 相位 + 色散/Drude + s/p 平均都已考虑模型
osc = reflectance_single_layer(x, theta0, d, phi, POLARIZATION_AVG, ...
cauchy_layer, fc_layer, cauchy_sub, fc_sub);
end

yhat = bg + A .* (osc ./ denom);
end

function [bg, osc_fit] = split_osc(x, x_poly, p, n1_const, rushe_deg, model_order, ...
USE_FRESNEL, POLARIZATION_AVG, ...
cauchy_layer, fc_layer, cauchy_sub, fc_sub, idx)
% 用于输出背景/振荡分量供作图
theta0 = rushe_deg*pi/180;

switch model_order
case 1
a1 = p(idx.a1); a0 = p(idx.a0);
q = [p(idx.q0), p(idx.q1), p(idx.q2), p(idx.q3), p(idx.q4)];
A = p(idx.A); d = p(idx.d); phi = p(idx.phi);
bg = a1*x_poly + a0;

```

```

otherwise
error('本版仅支持一次背景模型 (model_order=1) ');
end

denom = eval_poly4(x_poly, q); denom = max(denom, 1e-4);

if ~USE_FRESNEL
cth = sqrt(1 - (sin(theta0)/n1_const)^2);
osc = cos(4*pi*n1_const*d*cth*x + phi);
else
osc = reflectance_single_layer(x, theta0, d, phi, POLARIZATION_AVG, ...
cauchy_layer, fc_layer, cauchy_sub, fc_sub);
end

osc_fit = A .* (osc ./ denom);
end

function w = tukey_w(r, c)
% Tukey biweight (基于MAD的尺度), 返回 w , [0,1]
r = r(:);
s = 1.4826 * median(abs(r - median(r))); % MAD尺度
if s<=eps, w = ones(size(r)); return; end
u = r / (c*s);
w = (abs(u)<1) .* (1 - u.^2).^2;
end

%% ----- 物理子函数 -----

function R = reflectance_single_layer(nu_cm1, theta0, d_cm, phi, POLARIZATION_AVG, ...
cauchy_layer, fc_layer, cauchy_sub, fc_sub)
% 计算空气/单层/衬底结构在斜入射下的反射率 (s/p 平均)
% nu_cm1: 波数(cm^-1) 列向量; d_cm: 厚度(cm); phi: 额外相位 (标量)
nu_cm1 = nu_cm1(:);
n0 = 1; % 空气

% 波长换算
lambda_um = 1e4 ./ nu_cm1;
lambda_m = 1e-2 ./ nu_cm1;

% 复折射率: 外延层、衬底
n1 = n_complex_cauchy_drude(lambda_um, lambda_m, cauchy_layer, fc_layer);
n2 = n_complex_cauchy_drude(lambda_um, lambda_m, cauchy_sub, fc_sub);

% 各层倾斜角的 cos (复数)
cos0 = cos(theta0) * ones(size(nu_cm1));

```

```

sin0 = sin(theta0);
cos1 = sqrt(1 - (n0./n1 .* sin0).^2);
cos2 = sqrt(1 - (n0./n2 .* sin0).^2);
% 分支稳定化 (确保与入射半空间相连的主值)
cos1 = sign(real(cos1)).*cos1;
cos2 = sign(real(cos2)).*cos2;

% 界面 Fresnel 振幅系数
r01_s = (n0.*cos0 - n1.*cos1) ./ (n0.*cos0 + n1.*cos1);
r12_s = (n1.*cos1 - n2.*cos2) ./ (n1.*cos1 + n2.*cos2);

r01_p = (n1.*cos0 - n0.*cos1) ./ (n1.*cos0 + n0.*cos1);
r12_p = (n2.*cos1 - n1.*cos2) ./ (n2.*cos1 + n1.*cos2);

% 往返相位 (使用波数更稳健):  $\delta = 2\pi * n1 * d * \cos\theta_1 * v$ 
delta = 2*pi .* n1 .* d_cm .* cos1 .* nu_cm1; % 复数允许

% 额外相位:  $e^{-\{2i\delta + i\phi\}}$ 
expo = exp(2*i*delta + i*phi);

% 单层 Airy 公式
rs = (r01_s + r12_s .* expo) ./ (1 + r01_s .* r12_s .* expo);
rp = (r01_p + r12_p .* expo) ./ (1 + r01_p .* r12_p .* expo);

if POLARIZATION_AVG
R = 0.5*(abs(rs).^2 + abs(rp).^2);
else
% 若有极化信息可改为返回 rs 或 rp
R = abs(rs).^2;
end
end

function n_c = n_complex_cauchy_drude(lambda_um, lambda_m, cauchy, fc)
% 基体色散 (Cauchy) + 自由载流子 Drude → 复折射率
% lambda_um:  $\mu\text{m}$ ; lambda_m:  $\text{m}$ 
% 基体: Cauchy (实数部分), 可按需扩展出 k_base( $\lambda$ )
n_base = cauchy.A + cauchy.B./(lambda_um.^2) + cauchy.C./(lambda_um.^4);
eps_lat = n_base.^2; % 介电常数 (实数)

if isfield(fc,'add_drude') && fc.add_drude && fc.N_cm3>0
% 常数
eps0 = 8.854187817e-12; % F/m
e = 1.602176634e-19; % C
m0 = 9.1093837015e-31; % kg
c = 299792458; % m/s

```

```

N_m3 = fc.N_cm3 * 1e6; % cm-3 → m-3
mstar= fc.mstar_m0 * m0;
mu = fc.mu_cm2Vs * 1e-4; % cm2/Vs → m2/Vs

omega = 2*pi*c ./ lambda_m; % rad/s
omega_p2 = N_m3 * e^2 / (eps0 * mstar); % rad2/s2
gamma = e / (mstar * mu); % s-1

deps = - omega_p2 ./ (omega.^2 + 1i*gamma.*omega);
eps_total = eps_lat + deps;
else
eps_total = eps_lat;
end

n_c = sqrt(eps_total); % 复数主值
% 保证正的虚部 (吸收非负)
n_c = (real(n_c)>=0).*n_c + (real(n_c)<0).*(-n_c);
end

%% ----- 小工具 -----
function val = eval_poly4(xn, q)
% q = [q0 q1 q2 q3 q4], 在标准化 x 上计算 q0+q1*x+q2*x^2+q3*x^3+q4*x^4
val = q(1) + q(2).*xn + q(3).*xn.^2 + q(4).*xn.^3 + q(5).*xn.^4;
end

function s = ternary(cond, a, b)
if cond, s = a; else, s = b; end
end

```