universität wien

# MASTERARBEIT | MASTER'S THESIS

Titel | Title

## Infrared Reflection Spectroscopy and Optical Modelling for the Characterization of Polymer Films


verfasst von | submitted by

Cornelia Jakopin BSc


angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

| | |
|---|---|
| Studienkennzahl lt. Studienblatt \| Degree programme code as it appears on the student record sheet: | UA 066 606 |
| Studienrichtung lt. Studienblatt \| Degree programme as it appears on the student record sheet: | Masterstudium Drug Discovery and Development |
| Betreut von \| Supervisor: | Bernhard Lendl |

# Acknowledgements

First and foremost, I want to express my gratitude to Bernhard Lendl for providing me the opportunity to conduct this thesis in his research group. I am also extremely thankful to Georg Ramer, whose constant guidance and support was invaluable throughout these past months.

Additionally, I would like to thank the entire CAVS group (Lisa, Alicja, Kathi, Yide, Michi, Lena, Elisabeth, Shilpa, Sebi, Ufuk, Loisi, Felix, Verena, Margaux, Iskander, Dani, Paul, Harald, Nelson, Jésus, Dominik, Savda, Pily). The open discussions, and shared moments made this experience very insightful and enjoyable.

My sincere appreciation goes to the industry partner, KAI/Infineon for providing their support throughout the project. I am especially thankful to Eva, Mischa, Lars and Max whose encouragement and readiness to help were of importance.

Finally, I am profoundly grateful to my close friends Sebi, Irina, Miro and my family. Their encouragement, emotional support, and uplifting words carried me through the challenges and made this journey fulfilling and memorable.

# Abstract

Fourier transform infrared spectroscopy (FTIR) is a fast and non-destructive analytical method widely used for material characterization. However, when applied to the analysis of polymer-coated silicon wafers, it is complicated by challenges such as interference fringes and spectral distortions, which hinder accurate interpretation of the spectrum. To overcome these difficulties, this thesis focuses on the development of advanced computational methods for refining FTIR spectra and enabling reliable interpretation.

This study explores the impact of thickness variations of the polymer layer on the optical behavior with the ultimate goal of gaining information on the bulk properties of the polymers. Polymers such as polymethyl-methacrylate (PMMA) and polystyrene (PS) were analyzed using FTIR spectroscopy. Furthermore, it was observed that the substrate (silicon) has a pronounced fringe pattern and thus distorts the spectrum. This observation makes the spectral correction more challenging. For this correction, an algorithm was developed. To ensure the accuracy of the algorithm, the calculations implemented were verified and tested by applying them to the simulated data. Moreover, the algorithm was able to determine the thickness of the polymer layer and refractive index of the polymers studied.

This research successfully corrected the absorption spectra for thin polymer layers on silicon substrates. In addition to determining the polymer layer thickness and refractive index with high precision, modifications of the experimental setup were required. 3D printed apertures were implemented that improved the control of the angle of incidence. Despite these advancements, the absorption spectra of thicker polymer layers presented greater analytical challenges because of the increased interference effects in the spectra.

These findings have practical applications within the semiconductor industry, in particular in the production of wafers. It enables monitoring of the thickness and uniformity of the polymer layer during the fabrication process. Therefore, this work contributes to the advancement of FTIR spectroscopy as a tool for material characterization, enabling its application in research and industry.

# Kurzfassung

Fourier-Transformations-Infrarotspektroskopie (FTIR) ist eine zerstörungsfreie und einfache Analysemethode, die häufig zur Materialcharakterisierung verwendet wird. Bei der Anwendung auf polymerbeschichtete Siliziumproben wird die Interpretation der Spektra jedoch aufgrund von Interferenzmustern und Verzerrungen in den Spektra erschwert. Der Fokus dieser Arbeit liegt auf der Entwicklung eines mathematisch unterstützten Algorithmus, der FTIR-Spektren korrigiert und eine zuverlässige Interpretation ermöglicht.

Im Rahmen dieser Arbeit wurde auch der Einfluss von unterschiedlichen Schichtdicken der Polymere auf die optischen Eigenschaften untersucht, mit dem Ziel, Informationen über die Struktur und Eigenschaften innerhalb der Schichten zu gewinnen. Dabei werden Polymere wie Polymethylmethacrylat (PMMA) und Polystyrol (PS) mit Hilfe FTIR-Spektroskopie analysiert. Außerdem wurde festgestellt, dass das Substratmaterial, nämlich Silizium, ausgeprägtere Interferenzmuster aufweist als Gold. Diese Beobachtung erschwert die spektrale Korrektur. Zur Behebung dieses Problems, wurde ein Algorithmus entwickelt, dessen Genauigkeit durch Berechnungen überprüft und anhand von simulierten Daten getestet wurde. Der Algorithmus ist in der Lage, sowohl die Schichtdicke als auch den Brechungsindex der untersuchten Polymerschichten zu bestimmen.

In dieser Arbeit konnten Absorptionsspektren für dünne Polymerschichten auf Siliziumsubstraten erfolgreich korrigiert werden. Darüber hinaus konnte die Schichtdicke und der Brechungsindex der Polymere bestimmt werden. Diese Bestimmung wurde optimiert durch Anpassungen des experimentellen Aufbaus. Es wurden 3D-gedruckte Blenden eingesetzt, um den Einfallswinkel einzuschränken. Trotz dieser Fortschritte stellten die Absorptionsspektren dickerer Polymerschichten wegen stärkerer Interferenzeffekte größere analytische Herausforderungen dar.

Diese Erkenntnisse finden praktische Anwendung in der Halbleiterindustrie, insbesondere bei der Herstellung von Wafern. Darüber hinaus ermöglicht dies die Kontrolle von Dicke und Homogenität der Polymerschicht während des Herstellungsprozesses. Diese Arbeit leistet einen Beitrag zur Weiterentwicklung von FTIR-Spektroskopie und ermöglicht die Materialcharakterisierung von Polymerproben.

# Contents

# 1 Introduction

## 1.1 Aim of Thesis

The goal of this thesis is to better understand the mid-infrared spectra of polymer films on high-refractive-index materials (e.g. silicon) or reflective materials (e.g. gold). Specifically, the main focus of this project lies on the refinement of FTIR spectra of polymer-coated silicon wafers through advanced computational data analysis. Additionally, the variations in polymer layer thickness and composition are investigated and their influence on the optical properties is determined. This research has significant potential to establish a robust database for the automatic removal of interference effects visible in the spectra, facilitate the comparison of unknown compounds and reduce the reliance on manual analysis.

This project is conducted within the Institute of Chemical Technologies and Analytics in the research group process analytics with the support of an industrial partner. By combining experiments with simulation models, this research seeks to provide a comprehensive understanding of the interaction between IR light and these materials.[1]

The polymers which have been studied throughout this thesis included imides, polymethyl-methacrylate (PMMA) and polystyrene (PS) which where chosen due to their exceptional physical, mechanical and optical properties. These attributes make them valuable for applications that require optical transparency, mechanical strength and corrosion resistance. However, analyzing their infrared spectra when spin-coated on silicon wafers presents a challenge because of the influence of the film thickness on the spectra and the formation of fringes.[2–4]

For the analysis of these materials FTIR spectrometers from Bruker and Thermo Fischer were utilized. Despite having distinct software interfaces and appearances, the measurement settings could be configured in both, achieving functional equivalence.

This project addresses these complexities by developing methods for the refinement of infrared spectra and analyzing the interplay of film thickness and optical properties. Furthermore, it aids in the characterization of polymer-coated materials.[5,6]

## 1.2 Infrared Spectroscopy

Spectroscopy is the study of the interaction of matter with any part of the electromagnetic spectrum which is comprised of radiation at different wavelengths ranging

from gamma rays, UV-VIS, infrared to radio waves. These waves travel at the speed of light and can be described through the following relation:[1]

$$E = h\ f = \frac{h\ c}{\lambda} \tag{1}$$

where E corresponds to the energy of a photon, $f$ is the frequency of the radiation and $h$ is Planck's constant which is $6.634 \times 10^{-34}\,\mathrm{J\,s}$. The frequency is equivalent to the speed of light $c_0$ $3 \times 10^{8}\,\mathrm{m\,s^{-1}}$ divided by the wavelength $\lambda_0$ in vacuum.[1]

The infrared region comprises the wavenumbers of 10-12 500 $\mathrm{cm^{-1}}$. It is divided into the mid-infrared (400-4000 $\mathrm{cm^{-1}}$), near-infrared (4000-12 500 $\mathrm{cm^{-1}}$) and far-infrared (10-400 $\mathrm{cm^{-1}}$).[7]

The underlying phenomenon of infrared spectroscopy is based on the ability of a molecule to absorb infrared radiation. Mid-infrared spectral features can be related to the vibrational modes of molecules. These vibrational modes are determined by the structure of the molecule and are categorized as bond stretching or bending vibrations. These independent modes correspond to the vibrational degrees of freedom. Each molecule exhibits different vibrations, producing a unique spectrum for each molecule. Thus, mid-infrared spectroscopy is used to elucidate the molecular composition of materials.[8,9]

The absorption of infrared photons induces transitions between the rotational and vibrational energy levels. For the absorption to transpire, the frequency needs to correspond to one of the molecule's natural vibrational modes. The energy of mid-IR photons is too low to lead to excitation of electronic energy states.[1,8,10]

Molecular vibrations can be described using either the harmonic or anharmonic oscillator (figure 1). The harmonic oscillator assumes that the atoms vibrate with harmonic motion and therefore, the force that acts on the molecule is proportional to their displacement. The potential energy of the harmonic oscillator is described by eqn. (2).[8]

In contrast, the anharmonic oscillator also considers the potential energy curve of each bond (eqn. (3)). The vibrational frequency of a molecule is dependent on the bond strength $k$ and the reduced mass $\mu$ of the bonded atoms.[8]

$$V_{i,v} = h\ v_{osc}(v + \frac{1}{2}) \tag{2}$$

$$V_{i,v} = h\ v_{osc}(v + \frac{1}{2}) + h v_{osc} x (v + \frac{1}{2})^2 \tag{3}$$

where $h$ corresponds to Planck's constant, $v_{osc}$ is the fundamental frequency of the mode, $v$ denotes the vibrational quantum number of the same mode and $x$ is the anharmonicity constant.[8]

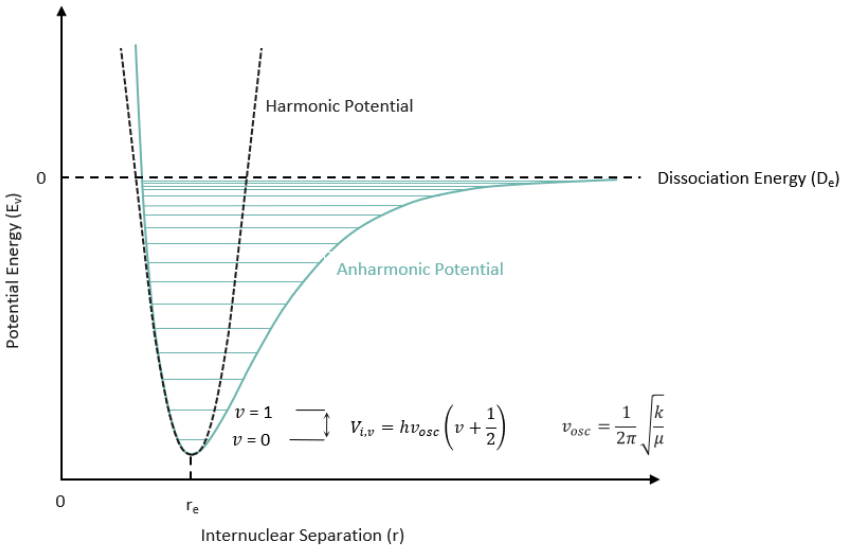



Figure 1: The potential energy of a molecule is plotted as a function of the atomic displacement during a vibration of the harmonic and anharmonic oscillator.[8]

A non-linear molecule with N atoms has 3N degrees of freedom that are categorized into three translational degrees of freedom in three perpendicular directions, three rotational degrees of freedom around the x,y and z axis and 3N-6 vibrational degrees of freedom. In contrast, a linear molecule only has two rotational degrees of freedom, resulting in 3N-5 normal vibrational modes.[8,10]

Functional groups are expected to exhibit characteristic absorption frequencies that would allow their presence in a compound to be elucidated. However, the interpretation of such a spectrum can be more complicated for a compound comprised of many atoms. This complexity arises from the various effects such as steric effects, electrical effects, size, phase change, electronegativity of neighboring atoms, overlapping and hydrogen bonding which all may cause boradening of bands and shifts in frequency.[10]

Infrared spectroscopy not only characterizes the molecular structure but also quantifies the concentration of the molecule, as the absorbed infrared energy is proportional to the concentration of the molecule.[10]

### 1.2.1 Mid-Infrared Spectroscopy

Mid-infrared (mid-IR) provides insights into the composition and optical characteristics of materials. It is a favored technique, due to its non-destructive behavior and its speed. Moreover, it enables the measurement of molecules in any phase (gas, liquid or solid) and identifies physical and chemical defects within the molecular structure. Thus, mid-IR spectroscopy can aid in understanding the surface and interface behavior of solid-state devices which is of particular importance in the semiconductor industry.[8,11]

The functional group region of the spectrum, which spans from 1600 to 4000 $\mathrm{cm}^{-1}$, is characterized by peaks that correspond to the stretching motions of the atoms. This region is easy to interpret, as the peaks are evenly spaced. In contrast to the fingerprint region, which ranges from 500 to 1600 $\mathrm{cm}^{-1}$, is more complex, with overlapping peaks. However, the fingerprint is very representative and unique to each molecule making mid-IR spectroscopy a highly selective technique.[9]

### 1.2.2 Fourier Transform Infrared Spectroscopy

FTIR spectroscopy is a vital and established analysis tool which is widely implemented as a spectroscopy method that relies on interferometer such as the Michelson interferometer as its fundamental component. It has a high energy throughput, resulting in a higher signal to noise ratio in a short amount of time. FTIR has a wide range of applications, including material science, environmental monitoring and medical research.[8,12,13] Additionally, FTIR can be used as a first-line analysis tool for the identification of functional groups on chemical compounds, enabling both bulk and surface analysis[9]

This sensitive spectroscopy method is based on a two-beam interferometer known as the Michelson interferometer, which was first designed by Michelson in 1891. This instrument first splits a polychromatic light beam into two paths due to a beamsplitter. The two light beams of which one is reflected to a fixed mirror and the other one partially transmitted to a mirror that moves along an axis are then returned to the beamsplitter. Due to one mirror being fixed and the other one moving along an axis a path difference is introduced. At the beamsplitter the two light beams interfere constructively or destructively depending on the path difference between the two arms and are again partially reflected or transmitted. The intensity of the light beams which reach the detector or return to the source is affected by the path difference. The variation in intensity can be measured overtime as a function of the path difference resulting in an interferogram containing the

spectral information. This data is subsequently converted into a frequency domain spectrum by applying Fourier transformation. The resulting spectrum displays the intensity of light absorbed at different wavenumbers proportional to the frequencies, providing a detailed characterization of the molecular structure of the material being analyzed.[7,8,13]

The signal that is recorded by the detector is proportional to the intensity of the source, the beamsplitter efficiency, sample absorption, detector response and amplifier characteristics. The measured interferogram is a composite of the individual interferograms that correspond to each wavelength when radiation of multiple wavelengths are emitted by the light source.[8]

FTIR instruments are equipped with a helium-neon laser which aids in reading out the mirror position for the interferometer, a resistively heated silicon carbide rod known as Globar and a mercury-cadmium-telluride (MCT) detector (in the case of the instruments used in this work). A Globar is a broadband infrared source which emits wavelengths that are subsequently modulated via an interferometer. The MCT detector belongs to the class of photon detectors, which operates by absorbing infrared radiation to excite electrons, thereby generating an electrical signal. To minimize thermal noise and achieve optimal performance, the detector requires cooling, usually with liquid nitrogen (77K).[8,14]

A distinct feature of the MCT detector is its ability to respond to weak light intensities while being robust to changes in modulation frequency. It is used for wavelengths between 3-30 µm. This sets it apart from pyroelectric detectors, belonging to the class of thermal detectors, which has a spectral range of 8-14 µm. The pyroelectric detector is less suitable for high-frequency applications as it is more susceptible to these changes. This type of detector is operated at room temperature and has a lower sensitivity as well as slower response, but it is cheaper.[14–16]

In addition, the MCT detector is able to capture the entire infrared band from the near IR to the far IR making it very versatile in its application. MCT detectors can achieve high absorbance sensitivity due to its high resolution measurements of light intensity.[13,14,16]

### 1.2.3 Optical Properties

As light travels through a medium, its electromagnetic waves interact with the material's atoms or molecules, inducing a dipole moment. This dipole moment, in turn, reradiates an electromagnetic field, which is a result of the oscillating charge, and absorbs some of the energy at specific oscillation frequencies corresponding to

the energy required to excite the atom or molecule. The collective effect of these dipole fields, combined with the incident field, results in the total macroscopic field within the material. The dielectric function provides a mathematical framework for understanding this interaction, relating the displacement field, incident electric field, and electric polarization to each other, and thereby, describing the material's response to electromagnetic radiation.[17]

Furthermore, the dielectric function ($\varepsilon$) relates to the complex refractive index ($n'$) via the relationship: $\varepsilon = n'^2$. These two functions can be summarized as the optical properties of a material and are comprised of the real and imaginary part, providing information on the alteration of light through interaction with a material. The complex refractive index is given by equation (4).[17–19]

$$n_i^{'}(\tilde{\nu}) = n_i(\tilde{\nu}) + i\kappa_i(\tilde{\nu}) \tag{4}$$

The refractive index is given by $n'(\tilde{v})$ and the imaginary refractive index, also known as the absorption index, is given by $\kappa(\tilde{v})$.[20]

### 1.2.4 Bouger-Beer-Lambert's Law

The Bouger-Beer-Lambert's law (eqn. (5)) is a fundamental law of quantitative spectroscopy, describing the attenuation of light traveling through an isotropic and homogeneous material. It states that the absorbance of a sample is proportional to the molar concentration and thickness of the sample by taking the molar extinction factor into account.[8,21]

$$A(\tilde{\nu}) = log_{10}\frac{I(\tilde{\nu})}{I_0(\tilde{\nu})} = \epsilon\, c\, d \tag{5}$$

where $A(\tilde{\nu})$ is the absorbance, $I(\tilde{\nu})$ is the intensity of the transmitted radiation, $I_0(\tilde{\nu})$ is that of the incident radiation, $\epsilon$ is the molar absorption coefficient, $c$ is the molar concentration and $d$ the sample thickness.[8]

Bouger-Beer-Lambert's law is only applicable when the absorption bands solely focus on specific wavelengths and are low in absorption. Hence, nonlinearities can arise due to several effects including instrumental or chemical effects, leading to inaccurate results. Chemical effects occur due to polymerization, association, and complexation while instrumental effects can arise from the spectrometer's resolution and sample placement. To reduce these effects, the spectrometer's settings and sample preparation require optimization. Furthermore, real light sources have a finite spectral bandwidth, leading to averaging errors. It is crucial that the measurement

is repeatable.[8,22]

Furthermore, if the absorption is high Bouger-Beer-Lambert's law cannot be applied because of local field effects. In addition, multiple reflections within the sample change the light wave amplitude due to interference. This leads to changes in the electric field intensity, which is also dependent on the location and sample thickness, resulting in a non-uniform absorbance distribution and shifts in the position of the absorption maximum.[22]

## 1.3 Infrared Reflectance of Thin Films

The analysis of infrared reflectance spectra of thin films is intricate, requiring careful considerations of the light's passage through three distinct layers and the angle of incidence. This complexity arises from the interplay between measurement conditions and material properties, which can induce intensity variations and line shape distortions in the spectra. As a result, the interpretation of these necessitates a higher degree of precision and care in data analysis to ensure accurate and reliable results.[23]

A crucial step in the analysis of thin films is the selection of a suitable substrate. A highly reflective material, such as a metal like gold which has a reflectivity of 98%, is preferable. The high reflectivity of these substrates can be attributed to the mobile electrons within the material, which are not able to propagate the electromagnetic wave. Such substrates are easily fabricated and relatively inexpensive in comparison to transparent materials that are commonly used for mid-IR spectroscopy, such as calcium fluoride ($CaF_2$).[20]

The three layer systems composed of air, polymer and substrate produce fringes, which originate from the interference between the reflected light from the air-film interface and the film-substrate interface depicted in figure 2 and depend on the wavenumber. Interference is caused by overlapping of waves of identical wavelength, leading to constructive or destructive interference.[3,6]

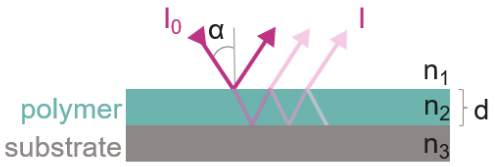



Figure 2: Three layer system comprised of air, thin film of a polymer and substrate.[3]

The formation of the fringes is illustrated in figure 3, and complicates data analysis, the traditional model, Bouger-Beer-Lambert's law, assumes linear absorbance that is proportional to the concentration and path length, but does not describe this complex system entirely.[24] Thus, the analysis of thin films needs to take wave optics and electric field intensities into account as band ratios and positions are dependent on the thickness of the film. As the absorption is proportional to the electric field intensity, an increased or decreased absorption may be caused by an electric field standing wave (EFSW).[3,6,24]

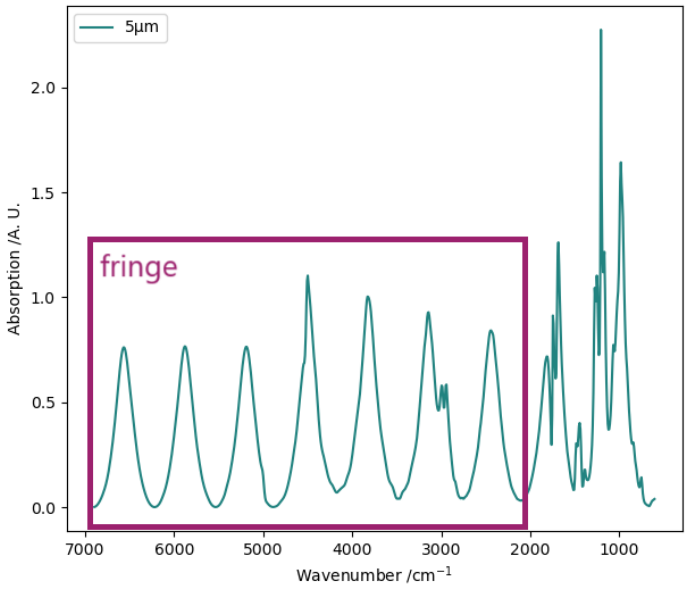



Figure 3: Depiction of the effect of multiple reflection within the polymer resulting in fringes.

Furthermore, reflection only takes place at an interface where the complex refractive index changes. The complex reflection amplitude is governed by Fresnel's equations (eqn. (6a)) and is a function of the refractive indices of air, film and substrate as well as the angle of incidence, polarization of light and film thickness. Polarization refers to the orientation of the electric field vector of the electromagnetic wave relative to the plane of incidence. The incoming light is composed of two orthogonal components: one with the electric field vector perpendicular to the plane of incidence (s-polarization) and another with the electric field vector lying within the plane of incidence (p-polarization). The combination of these two components through linear superposition describes the overall electric field vector of the incident wave and influences how the wave interacts with the interface.[8,20,25,26]

$$r_j = \frac{r_{12,j} + r_{23,j}\exp(2i\phi)}{1 + r_{12,j}r_{23,j}\exp(2i\phi)}, \quad j = s, p \tag{6a}$$

$$r_{12,s} = \frac{k'_1 - k'_2}{k'_1 + k'_2}, \quad r_{23,s} = \frac{k'_2 - k'_3}{k'_2 + k'_3} \tag{6b}$$

$$r_{12,p} = \frac{n'^2_1 k'_2 - n'^2_2 k'_1}{n'^2_1 k'_2 + n'^2_2 k'_1}, \quad r_{23,p} = \frac{n'^2_2 k'_3 - n'^2_3 k'_2}{n'^2_2 k'_3 + n'^2_3 k'_2} \tag{6c}$$

$$k'_1 = n'_1 \cos\alpha \tag{6d}$$

$$k'_i(\tilde{\nu}) = \sqrt{n'_i(\tilde{\nu})^2 - \sin^2\alpha}, \quad i = 2, 3 \tag{6e}$$

$$\phi(\tilde{\nu}) = 2\pi\tilde{\nu}k'_2(\tilde{\nu})d \tag{6f}$$

$\phi$ is the phase, which is dependent on the thickness of the polymer film $d$, the angle of incidence $\alpha$, the refractive index $n$, which changes across the absorption bands, and the wave vector $k$. The wave vector characterizes the direction and spatial frequency of wave propagation throughout the medium. The air/film interface is represented by $r_{12}$, while the film/metal interface corresponds to $r_{23}$. Moreover, the total reflectance $R$ is defined as $R = |r|^2$. Furthermore, the first medium is assumed to be gas or vacuum, having a refractive index of 1 and an absorption index of 0.[20,26,27]

### 1.3.1 Dispersion Analysis

The wavelength-dependent change of the refractive index is known as dispersion. Dispersion analysis aims to determine the refractive index spectrum of a material. To determine the polymer layer thickness and refractive index of the polymer, the measured data are fitted using the dielectric function which governs the optical behavior of the material such as reflectivity, absorption and penetration depth as outlined by Mayerhöfer *et al.* [20]. The dielectric function follows the Kramers-Kronig relations which relates the real and imaginary parts of the refractive index to each other. To obtain the refractive index and polymer layer thickness dispersion analysis is performed using an appropriate model to fit the measured data.[3,20]

One possible method for modeling the dielectric function is by applying the model of the damped harmonic oscillator. This method implements a non-linear least-square fit using the approximation of the Drude-Lorentz oscillator. This method is more efficient and removes noise compared to a direct fit in which the optical constants are independently adjusted at each wavelength and is more robust than applying the

Kramers-Kronig based technique. This is applicable because it is used to describe a shape with increasing absorption toward lower frequencies, which is the case for free-carrier absorptions such as those of metals, heavily doped semiconductors and transparent conductive oxides.[17,20,28]

A different approach for obtaining the imaginary refractive index is via a triangular basis function (eqn. (7)) described by Kuzmenko. This function is crucial, limiting the possible optical functions to only those that are plausible. Moreover, it reduces the number of free parameters needed to describe complex optical functions. Each oscillator should be narrow and provide spectral weight only in its region, which is achieved by setting its width to a fixed value.[20,21,28]

$$\kappa_i(\tilde{\nu}) = \begin{cases} \frac{\tilde{\nu}-\tilde{\nu}_{i-1}}{\tilde{\nu}_i-\tilde{\nu}_{i-1}}, & \tilde{\nu}_{i-1} < \tilde{\nu} \leq \tilde{\nu}_i \\ \frac{\tilde{\nu}_{i+1}-\tilde{\nu}}{\tilde{\nu}_{i+1}-\tilde{\nu}_i}, & \tilde{\nu}_i < \tilde{\nu} \leq \tilde{\nu}_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$$\begin{aligned} n_i(\tilde{\nu}_i) &= \frac{2}{\pi}\mathrm{P}\int_0^\infty \frac{\mathrm{k}''(\tilde{\nu}')}{\tilde{\nu}'^2-\tilde{\nu}^2}d\tilde{\nu}' \\ &= \frac{1}{\pi} * \left[\frac{g(\tilde{\nu}-\tilde{\nu}_{i-1})}{\tilde{\nu}_i-\tilde{\nu}_{i-1}} - \frac{(\tilde{\nu}_{i+1}-\tilde{\nu}_{i-1})g(\tilde{\nu}-\tilde{\nu}_i)}{(\tilde{\nu}_i-\tilde{\nu}_{i-1})(\tilde{\nu}_{i+1}-\tilde{\nu}_i)} + \frac{g(\tilde{\nu}-\tilde{\nu}_{i+1})}{\tilde{\nu}_{i+1}-\tilde{\nu}_i}\right] \\ g(x) &= (x+y)\ln|x+y| + (x-y) \end{aligned} \tag{8}$$

The variational dielectric function is built by implementing the triangular basis function. This approach represents the dielectric function as linear superposition of triangular functions, each centered at a specific frequency ($\tilde{\nu}_i$). The triangular line shape corresponds to a trapezoidal integration.[28] Thus, the index of absorption function is given by the values of the vertices $A_i$. The real refractive index can then be calculated as depicted in eqn. (9).[20]

$$n(\tilde{\nu}_i) = n_\infty + \sum_{i=2}^{N-1} A_i n_i(\tilde{\nu}_i) \tag{9}$$

where N is the number of points and $n_\infty$ is given by $\sqrt{\epsilon_{inf}}$ and is the initial estimate for the refractive index.[20]

When all parameters are allowed to vary freely, the fitting process often becomes numerically unstable. Additionally, this increases the complexity and ambiguity of the procedure, making routin operation difficult. This can be avoided by specifying some of the parameters.[28] However, the selection of the parameters influences the different structural elements, thereby limiting the range of the permitted parameter values.[11]

## 1.4 Preprocessing of the Spectrum

Spectra are often preprocessed to improve the accuracy, reliability and interpretability of the data, due to containing noise or other artifacts, that can distort the data. The following section outlines the key methods employed to process the data, with a focus on baseline correction, absorbance correction and the implementation of splines.

### 1.4.1 Baseline Correction

Baseline corrections are a standard procedure in FTIR spectroscopy, as baseline drift is a prevalent problem that can compromise both quantitative and qualitative analysis. The objective of this is to subtract the baseline from the spectrum, thereby isolating the signal bands and ensuring that they remain intact.[29,30]

Polynomial fitting is a widespread method, however, it is not always ideal, because its predefined structure offers less flexibility which is needed to handle datasets with noise or irregular shapes. Despite its simple application and effectiveness, the polynomial fit is prone to over- or under-fitting.[29,30]

A more robust method is the penalized least squares approach, which takes into account the fidelity and smoothness of the fitted baseline. Furthermore, such algorithms are fast and flexible in their implementation. Baek *et al.* [31] proposed the method of asymmetrical reweighted penalized least squares (arPLS) which is commonly applied to datasets containing high levels of signal noise. The objective of this technique is to fit a smooth baseline to the signal by minimizing the difference between the signal and estimated baseline. To ensure smoothness, deviations from smoothness in the baseline are penalized. This technique assigns weights to each data point during the iterative process to prevent the baseline from following the peaks. Signals in close proximity to the baseline receive similar weights due to the noise being more prominent in those regions while the weights are decreased for signals that deviate from the baseline. The arPLS method offers several advantages, including automatic adaptation of boundaries, handling of missing values, and control of smoothness. ArPLS has three parameters that need to be set by the user: the smoothing parameter ($\lambda$), the termination condition ratio and the maximum number of iterations. The choice of $\lambda$ is crucial, as it significantly affects the performance of the baseline correction. A larger value for $\lambda$ leads to a smoother curve, therefore, it is necessary to select an appropriate value to accurately and reliably depict the results.[29–32]

### 1.4.2 Model-Based Baseline Correction

As described by Mayerhöfer *et al.* [20] the following procedure was followed to correct the absorption. Utilizing equation (6) an initial guess could be obtained for the refractive index $n_\infty$ and polymer layer thickness $d$, for the spectral region above the range where fundamental infrared absorptions occur. Subsequently, an iterative fitting procedure is then employed to refine the estimate of the initial refractive index $n_\infty$ and polymer layer thickness $d$ by reducing the deviations between experimental and simulated values.[20]

The implementation of a baseline correction is necessary as the interference fringes compromise the accuracy of the first estimate of $\kappa$. This can be performed using equation (10).[20]

$$A_{corr} = \frac{-\lg(R_{meas}/R_{0,meas})}{-\lg(R_{calc}(n_\infty)/R_{0,calc})} \tag{10}$$

$\mathrm{R}_{meas}/\mathrm{R}_{0,meas}$ corresponds to the measured spectrum while $\mathrm{R}_{calc}(n_\infty)/\mathrm{R}_{0,calc}$ corresponds to the simulated spectrum. This equation also considers the reflectance, which is influenced by the incidence angle, polarization and the optical constants of both the substrate and film.[20]

As the baseline correction does not take dispersion distortions or intensity changes based on the EFSW effect into account, these need to be factored in. To obtain an initial estimate for the absorption index $\kappa$, equation (11) is used.[20]

$$\kappa_{app}(\tilde{\nu}) = \frac{A_{corr}}{\lg e \cdot 4\pi\tilde{\nu} \cdot 2\frac{d}{\sqrt{1-\frac{1}{n_2^2}\sin^2\alpha}}} \tag{11}$$

To generate improved values for the refractive index, the obtained absorption index was then placed into eqn. 7 and 8 which could iteratively determine better values when subsequently placed into eqn. (12).[20]

$$\kappa_{i+1}(\tilde{\nu}) = \kappa_i(\tilde{\nu})\frac{A_{\mathrm{meas}}}{A_{\mathrm{calc,i}}} \tag{12}$$

According to Mayerhöfer *et al.* [20] this equation, however, tends to converge too quickly. To address this, it is only used for the first iteration and all subsequent iterations implement eqn. (13).[20]

$$\kappa_{i+1}(\tilde{\nu}) = \kappa_i(\tilde{\nu})\frac{A_{\mathrm{calc,i}} - A_{\mathrm{meas}}}{\lg e \cdot 4\pi\tilde{\nu} \cdot 2\frac{d}{\sqrt{1-\frac{1}{n_2^2}\sin^2\alpha}}} \tag{13}$$

### 1.4.3 Spline

The implementation of polynomials for dispersion analysis can be inefficient due to the possibility of producing noisy results. A more suitable alternative is the implementation of splines, which divide the wavelength range into intervals and apply a polynomial function within each interval. This parameterization of the dielectric function is a vital step in accurately modeling the optical properties of the materials and reduces the number of fitting parameters. These piecewise functions are smoothly connected at joining points which are known as "knots". Splines are used to describe the imaginary part $k$ and the real part $n$ can be subsequently determined through the application of Kramers-Kronig relation.[17,33,34]

Each spline only affects the shape of the curve for the given interval, without influencing other parts of the spectrum. Furthermore, Johs *et al.* [34] demonstrated that if all the spline coefficients are positive, the convex-hull property ensures that $\kappa$ remains positive. The basis function only depends on the node position, thus, the amplitudes of each node shape the curve and can be treated as linearly independent variables.[17,33,34]

Determining the optimal number of data points is crucial to mitigate the risk of under- or overfitting. A reduced number of knots may fail to capture the underlying trends in the data, resulting in underfitting, while an excessive number of points can lead to overfitting. Consequently, it is crucial to evaluate the optimal number of knots. By decreasing the knot spacing, the fit can be improved and by optimizing the knot positions, a reduction in the number of knots is facilitated, leading to an improvement of the overall fit quality.[33,34]

## 1.5 Simulation of Data

To validate the developed algorithm with the help of the equations mentioned above, simulated data was generated. The simulations were performed using the transfer-matrix-method (TMM) package in python written by Byrnes [35] which is a tool employed to study wave propagation in multilayered systems. Specifically, the coh_tmm sub-package was used, as this package calculates the TMM for coherent cases. Thus, it is suitable for simulating the reflectance of polymer films on a substrate in a standard environment.[35–38]

The TMM algorithm is based on the Fresnel equations, as explained in section 1.3, to compute the transmission and reflection at the interface of each layer within the multilayered material. The overall transmission and reflection of an incident

plane wave through the sample can be computed by taking the different thicknesses and refractive indices of each layer into account. The complex refractive indices are constant throughout each layer and the layers are assumed to be optically flat, which allows for a one-dimensional calculation. The coefficients of transmission and reflection for a particular wavelength and angle of incidence strongly depend on these parameters. Moreover, even slight variations in wavelength can result in significant deviations for the same multi-layer film.[36–38]

# 2 Experimental Methods

## 2.1 Sample Preparation

The sample set consisted of two categories: experimentally prepared polymer samples with a fully known composition and industrial manufactured polymer samples with an unspecified chemical composition.

The developed manufactured samples consisted of a silicon substrate with a thickness of 300 µm. They were cut into 30 mm × 30 mm pieces, rinsed with acetone and dried with pressurized air.

For the polymer layers, polystyrene (PS) and polymethyl-methacrylate (PMMA) were applied onto the substrates. These polymer films were fabricated by dissolving each polymer correspondingly in toluene at 80°C under continuous stirring for 4-6 h. Subsequently, each polymer was spin-coated with a POLOS SPIN150i spin coater onto the substrates. The system settings of the spin coater were adjusted to achieve various thicknesses. For a thinner coat, the acceleration settings were set to 2000 rpm/s and for a thicker coat to 1000 rpm/s. This was repeated for each polymer solution and the coated substrates were dried for 30 min.

The industrial polymer samples are manufactured on silicon substrate and were coated with an imide photoresist. The samples were produced according to standard conditions.

In addition to silicon substrates, which are commonly used in semiconductor manufacture gold was also implemented for comparison purposes, as Mayerhöfer *et al.* have previously demonstrated successful removal of the fringes using this substrate.[20] The refractive indices for both of the materials were obtained from literature.[39,40]

## 2.2 Determination of Polymer Layer Thickness

The layer thicknesses of the spin-coated samples were measured using a contact profilometer (Dektak XT, Bruker). Obtained values are presented in table 1. A contact profilometer determines the layer thickness by tracing the surface of the polymer with a needle, which records the changes in height as it moves across the sample. Since the polymer did not fully cover the entire wafer, the Dektak XT was able to measure the thickness by tracing over the edge of the polymer layer. As the needle moves over the edge, it detects the step height between the polymer surface and the underlying surface, enabling the determination of the polymer layer thickness. The measurement was performed at a resolution of 0.1 µmpt, with the stylus force set to 2 mg and a scan length of 900 µm. The motorized stage ensures high accuracy positioning and the repeatability of the measurement. Furthermore, the step height repeatability of this instrument is >5 Å while the vertical resolution is 1 Å.[41]

The polymer layer thickness of the industrial manufactured sample was initially estimated to be approximately 10 µm, which was later verified using a focused ion beam scanning electron microscope (FIB-SEM). During this method, thin sections of the samples are created by a focused ion beam, exposing cross-sections. The electron beam intersects the ion beam at an angle of 52° above the surface of the sample, to enable real-time SEM imaging. Before performing FIB milling the wafer was coated with PtPd and C was used to cover the surface during the cut, to enhance the samples conductivity and to ensure a high quality cross-section. To calculate the film height a factor of $\tan(52°) \approx 1.27$ was applied to account for the angle of electron beam.[42] The polymer layer thickness could then be determined by measuring the polymer on the image. The obtained value is shown in table 1.

Table 1: Layer thickness of each sample determined by profilometer measurements and FIB cross section.

| **Samples** | Si-wafer coated with PMMA | | | Si-wafer coated with PS | | |
|---|---|---|---|---|---|---|
| | 12 wt% | 8 wt% | 4 wt% | high | medium | low |
| $\mathbf{d}_{measured}$ **(µm)** | 2.02 | 1.18 | 0.19 | 3.26 | 2.50 | 0.93 |
| **Sample Industry** | Imide coated Si-wafer | | | | | |
| $\mathbf{d}_{measured}$ **(µm)** | 10.15 | | | | | |

## 2.3 FTIR Measurements

FTIR spectra were collected using the ThermoScientific Nicolet iN10 MX FTIR and the Bruker Hyperion 3000-MCT FTIR. Both of these instruments implement a Globar and a MCT detector and have a 15x Cassegrain objective.[43,44]

### 2.3.1 Adaptation of Angle of Incidence

For configuration of the angle of incidence, apertures were developed for the ThermoScientific Nicolet iN10 MX FTIR. The aperture models were first drawn in Autodesk Inventor 2025 (figure 4) and subsequently 3D printed. The printing process was carried out using the HP Type MJF5200 printer which uses the polyamide Multi Jet Fusion (MFJ) powder printing technology. As printing material the polyamide PA12 was utilized.

The maximum angle of incidence for this FTIR instrument is 45°. To configure the angle of incidence, the apertures contained 1 mm slits, restricting the angle of incidence to a narrower range. To sample a variety of angles, the following have been selected: 17° (ranging from 14.59° to 19.35°) (fig. 4) due to its use in the algorithm, 20° (ranging from 17.67° to 22.27°) and 22° (ranging from 19.73° to 24.20°).

For the measurements, the apertures were placed into a holder which was placed in front of the lens as illustrated in figure 5. The samples without aperture were measured within a 300 µm × 300 µm window with 256 scans and a resolution of 8 $cm^{-1}$. The higher number of scans was necessary to achieve an optimal signal-to-noise ratio.

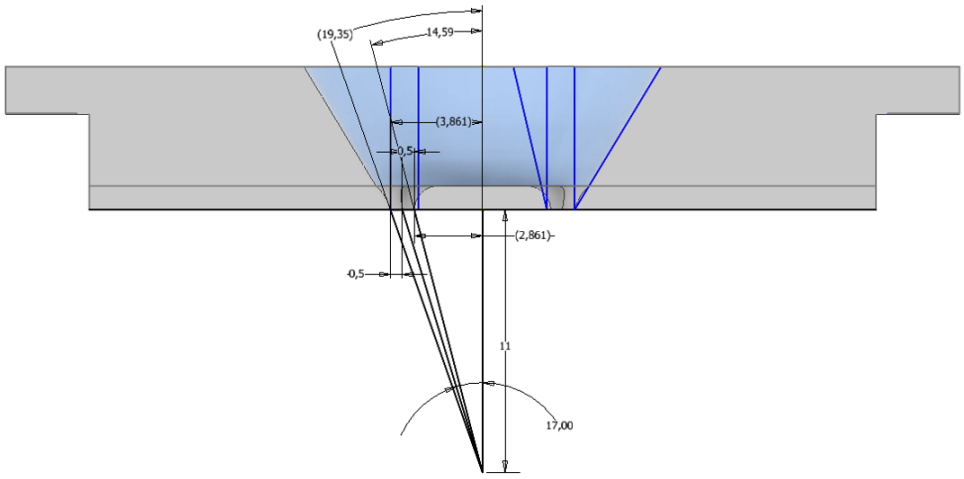



Figure 4: Cross-section of the aperture confining the angle to a range of 14.59°-19.35°, with the median being 17°. The angles as well as the distances are labeled. The measured distance between the sample and the aperture was 11 mm.

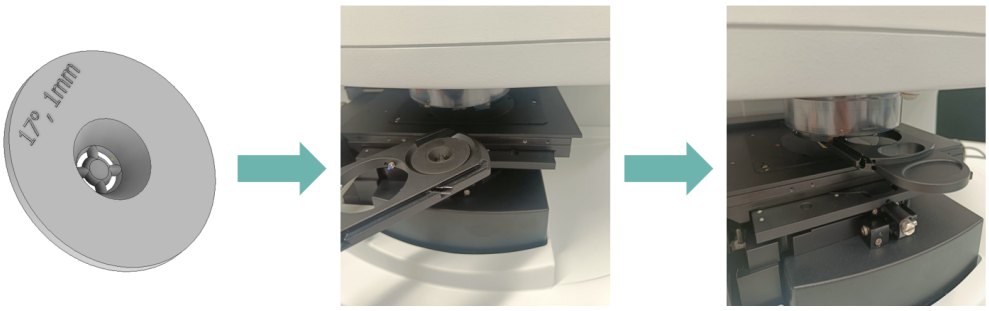



Figure 5: On the left-hand side, the 3D model of the aperture of 17° is illustrated. The image in the middle shows the aperture inserted into the holder. On the right-hand side, the holder is placed in front of the lens of the ThermoScientific FTIR.

# 3 Results

The focus of the analysis was to determine the polymer layer thickness and correct the absorption spectra of the samples, to enable accurate interpretation of the bulk polymer on silicon. To accomplish this, a stepwise approach was developed. An algorithm was developed based on the equations outlined in sections 1.3 and 1.4.

The algorithm begins with iteratively refining the polymer layer thickness value and refractive index from an initial starting value, so that it can be executed, the angle of incidence and wavenumber range also had to be predetermined. Once these parameters are established, the absorption spectra are corrected through further iterative refinement of each parameter.

As the first measurements were performed with the Hyperion 3000-MCT the median angle of 17° was implemented in equation (6). This proved to yield inaccurate results, therefore, the experimental setup was adjusted by adapting the angle of incidence with the 3D printed apertures. Hence, it was assumed that by defining the angle of incidence, the error was minimized and the correction of the absorption spectra could be improved.

To guarantee the applicability of the developed method, the algorithm was initially tested and optimized with simulated data. The next step was to test the algorithm on prepared samples and subsequently apply it to the industrial manufactured samples. By following these steps the goal was to demonstrate its applicability in the industry.

## 3.1 Simulation of PS coated wafers

The simulations enable the verification of the algorithm's performance as well as gaining insights into its data handling behavior.

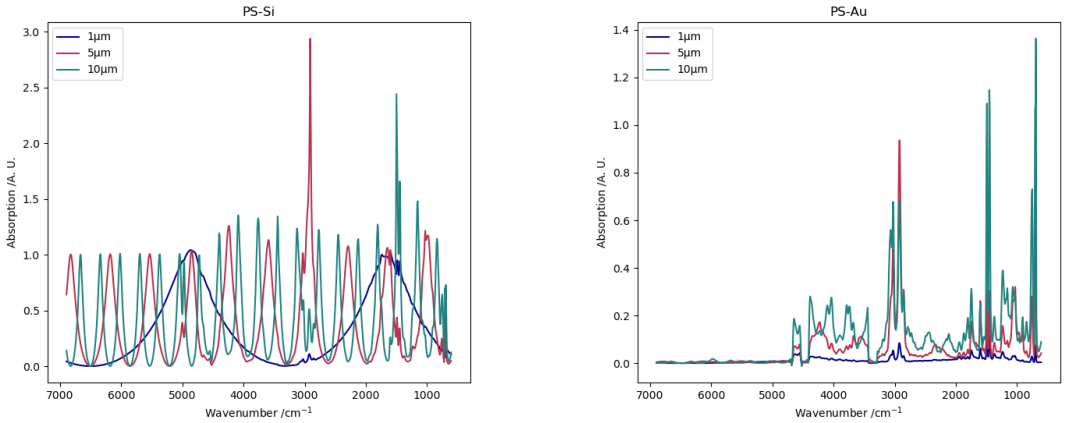
### 3.1.1 Verification of Implementation

As outlined in section 1.5, the multilayered system was modeled using the TMM package in Python. However, for the specific application the implementation in the package was found to be quite slow. Therefore, a faster code based on the equations described in section 1.3 was implemented and its output was validated against that of the TMM package. For this validation, the refractive index of air was set to 1, while that of the polymer was assumed to be 1.5 and that of silicon to be 4. The polymer layer thickness was set to 5 µm and the wavenumber to $1000\,\mathrm{cm}^{-1}$.

These parameters were then incorporated into the reflection coefficient for the s-polarized light. Notably, when tested across a range of wavenumbers, angles and thicknesses, the manual calculation and the TMM package produced a result that is effectively equivalent with a negligible deviation of $8.7 \times 10^{-13}$ , thereby confirming the accuracy of the implementation. The advantage of using the equations directly instead of relying on the python package for the removal of the fringes, is the significant reduction in computation time, which is crucial for a large data volume. The manual calculation was completed in 790 µs, whereas the TMM package required 4.69 ms. It should be noted that this difference is mainly due to the increased complexity of the TMM package, which can calculate arbitrary stacks of layers and provides reflection data, transmission data and layer specific absorption data.

### 3.1.2 Simulation of Reflection-Absorption Spectra

The TMM package incorporates several free parameters that need to be specified in advance for a simulation. These are refractive indices of the materials, the angle of incidence, the vacuum wavelength, the layer thicknesses and polarization of the incident light. The polymer layer was assumed to be PS, and the spectral values were obtained from existing literature.[2] To incorporate a broad range of values and stay consistent with the dimensions of the developed samples and those provided by the industrial partner, the layer thicknesses were set to 1 µm, 5 µm and 10 µm. Furthermore, the angle of incidence was set to 17°, corresponding to the median angle of the Hyperion 3000-MCT FTIR.




(a) Simulated absorption spectra of Si-wafers coated with PS of various thicknesses (1, 5 and 10 µm) at an incidence angle of 17°.

(b) Simulated absorption spectra of Au-wafers coated with PS of various thicknesses (1, 5 and 10 µm) at an incidence angle of 17°.

Figure 6: Modeled spectra of PS on silicon and gold wafers.

The modification of the substrate from silicon to gold has a significant impact on the FTIR spectra, as visualized in the figure 6. Silicon exhibits more pronounced fringes. Replacing silicon with gold leads to a change in refractive index, as gold has a higher reflectivity than silicon. However, the reflectivity at the air-polymer interface remains unchanged.

### 3.1.3 Determination of Refractive Index and Polymer Layer Thickness

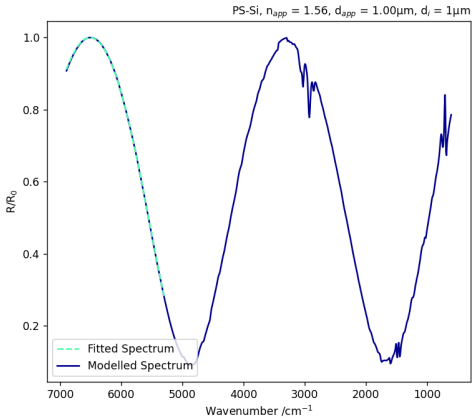
The simulated PS spectra were subsequently fitted to determine starting values for the base refractive index and polymer film thickness. Within the fit, these two parameters were constrained by specifying bounds. In this case, the refractive index was set to be in the range between 1 and 2, with an initial value of 1.5. Due to the fact that the refractive index varies with the wavenumber and the algorithm calculates an average over the wavenumber range, the index is expected to fall between 1.4 and 1.6.

For the polymer layer, the initial value was based on the predetermined thickness. For the 1 µm layer, the bounds were set to 0 and 4, for the 5 µm layer, they were defined as 2 and 9 µm and for the 10 µm PS layer, the bounds were set to 5 and 15 µm.

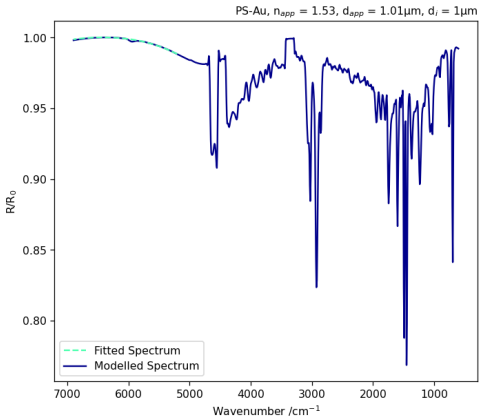
This procedure utilized the least squares and shgo (simplicial homology global optimization) algorithm introduced by Virtanen *et al.* [45], both of which are part of SciPy's optimization modules, to iteratively refine the values for the refractive

index and polymer layer thickness. The non-linear least squares algorithm handles nonlinear minimization problems with box constraints. It aims to find the best fit by minimizing the sum of the squares of the errors between the modeled and fitted data. However, it can converge to a local minimum, in particular when the starting value is unknown, as observed in fig. 8. To avoid this, a global optimization algorithm (shgo) is implemented. Shgo is ideal for locating a global minimum within bounds. It achieves this by sampling several starting points according to a specific method, from which it identifies the local minimum. To ensure a thorough analysis of the search space, it was initialized with 200 sampling points. The specified sampling points assist in identifying regions of interest for further optimization. Increasing the number of sampling points enhances coverage of the search space and improves the likelihood of locating the global optimum.[45–48]

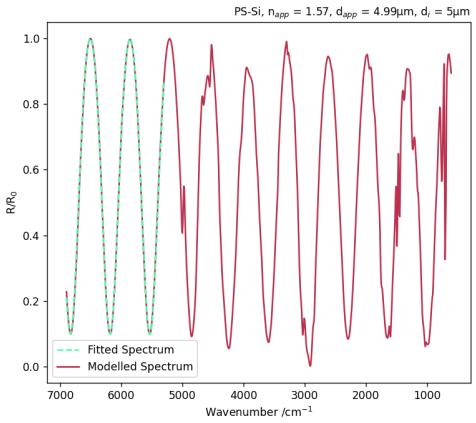
By applying the above to the simulated data of the PS-Si and PS-Au samples, figures 7a-7f were obtained.



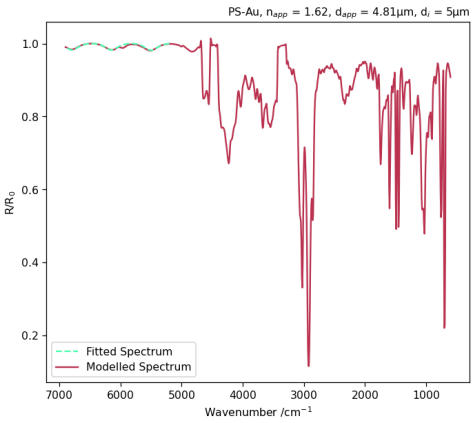

(a) Reflection spectra of the simulated PS-Si sample with a layer thickness of1 µm along with its corresponding fit, displaying an excellent alignment. The obtained refractive index and polymer layer thickness are 1.56 and 1.00 µm respectively, closely aligned to the expected values.



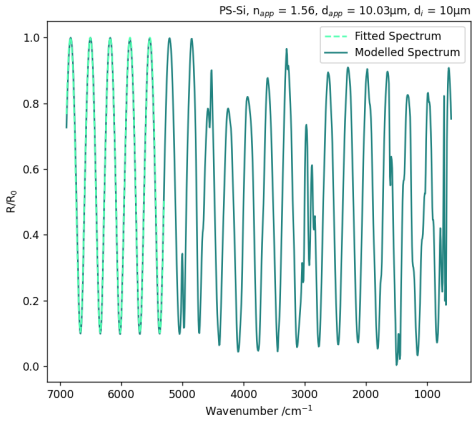

(b) The simulated PS-Au sample with a layer thickness of 1 µm is illustrated alongside its corresponding fit, displaying an excellent alignment. The obtained refractive index and polymer layer thickness are 1.53 and 1.01 µm respectively, matching closely to the expected values.



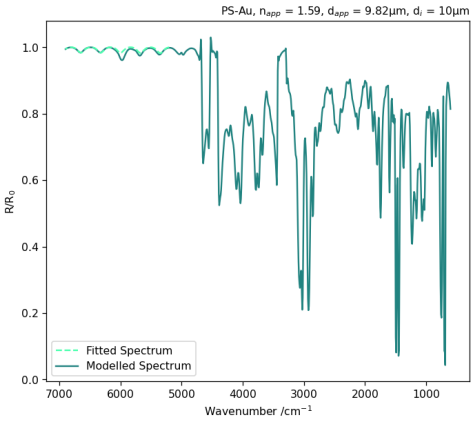

(c) Illustration of the simulated PS-Si sample with a layer thickness of 5 µm along with its corresponding fit, displaying an excellent alignment. The obtained refractive index and polymer layer thickness are 1.57 and 4.99 µm respectively, displaying an excellent agreement with the expected values.




(d) Reflection spectra of the simulated PS-Au sample with a layer thickness of5 µm alongside its corresponding fit, displaying an excellent alignment. The obtained refractive index and polymer layer thickness are 1.62 and 4.81 µm respectively, matching closely to the expected values.




(e) Illustration of the simulated PS-Au sample with a layer thickness of10 µm along with its corresponding fit, displaying a good alignment. The obtained refractive index and polymer layer thickness are 1.56 and 10.03 µm respectively, which are in accordance with the expected values.




(f) Reflection spectra of the simulated PS-Au sample with a layer thickness of10 µm along with its corresponding fit, displaying an excellent alignment. The obtained refractive index and polymer layer thickness are 1.59 and 9.82 µm respectively, differing slightly from the expected values of 1.5 and 10 µm.

Figure 7: These plots depict the simulated spectra of PS with their corresponding fit in a turquoise dashed line.

For the simulated PS-Si model, the refractive index has an average value of 1.56 and the polymer layer thickness is very close to the expected value. In contrast, for the

PS-Au model, the polymer layer thickness deviates slightly from the expected value. Furthermore, the fit does not fully match to the simulated PS-Au spectrum due to its sinusoidal curve behavior, which is not influenced by the peaks. The obtained fit results are presented in table 2.

Table 2: Fit results of the simulated PS coated silicon and gold substrates.

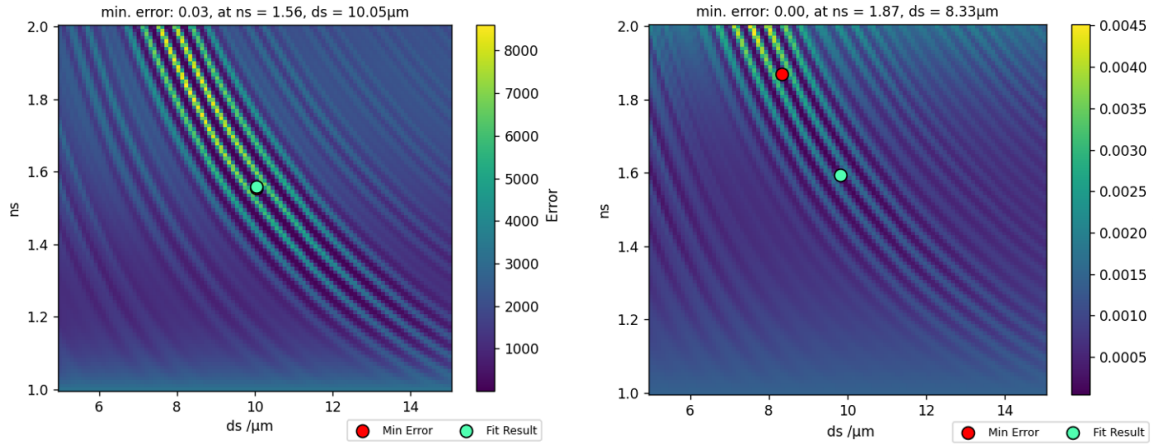
| **Layer Thickness** | 1µm | | 5µm | | 10µm | |
|---|---|---|---|---|---|---|
| **Parameter** | n | d ($\mu$m) | n | d ($\mu$m) | n | d ($\mu$ms) |
| **PS** | 1.56 | 1.00 | 1.57 | 4.99 | 1.56 | 10.03 |
| **PMMA** | 1.53 | 1.01 | 1.62 | 4.81 | 1.59 | 9.82 |

### 3.1.4 Ambiguity of Refractive Index and Thickness Fit Results

There should only be a single correct result for the fit of each sample. However, while the reflection at the interfaces change only slowly as the refractive index changes the interference due to phase shift changes more rapidly with refractive index, wavelength and sample thickness, leading to the oscillating baseline fringes. This change is described by the expression $\exp(2i\phi)$ in (6f). Furthermore, as $\phi = 2\pi\tilde{\nu}k_2'(\tilde{\nu})\,d$ a change in the refractive index can be compensated by an opposite change in the sample thickness.

To illustrate this relationship, a heatmap was generated for the PS-coated silicon and gold wafers with a layer thickness of 10 µm. The heatmap represents the residual error between the target spectrum and the modeled spectrum. To obtain a single error per combination of these two parameters, the squared sum of the residuals was used. The parameter space was defined with bounds of 1 and 2 for the refractive index and for the polymer layer they were adjusted to 5 and 15 µm.

The darker curved line represents a smaller minimum error than the lighter line. Moreover, the obtained value for the fit of the previous section is depicted as a turquoise dot, while the red dot represents the minimum residual error.




(a) Depiction of the error grid of PS-Si with a polymer layer thickness of 10 µm. The minimum error and the fit result overlap at a refractive index of 1.56 and a thickness of 10.05 µm.

(b) Depiction of the error grid of PS-Au with a polymer layer thickness of 10 µm. The minimum error is observed at a refractive index of 1.87 and a thickness of 8.33 µm and deviates from the fit result.

Figure 8: Illustration of the minimum error between the refractive index (ns) and the polymer layer thickness (ds) for each simulation accordingly. The color intensity corresponds to the magnitude of each error, darker regions correspond to lower errors, while lighter regions have a higher error.

For the silicon wafer sample, the minimum error is identical to the fit result and matches closely to the expected values of the refractive index and polymer layer thickness of 1.5 and 10 µm respectively. In contrast, the gold wafer sample deviates, with the minimum error falling below the expected values, while the fit displays a more accurate result. A trend emerges when comparing the results: higher polymer layer thicknesses lead to fit results with a lower refractive index values and vice versa.

### 3.1.5 Correction of the Absorbance

The absorption, refractive index and polymer layer thickness values were subsequently iteratively refined, as outlined in sections 1.4.1-1.4.3.

In initial experiments it was found that the results of the fit tended to oscillate, i.e. neighboring data points would start to move towards high and low values in an alternating fashion. Therefore, the optimization was extended by a penalty: keeping the spectrum smooth (i.e. a low absolute sum of the second derivative). The weighting factor ($\text{diff}_{weight}$) applied to the second derivative was set to $10^{-20}$, to enforce the smoothness by penalizing high curvatures in the absorption index. This value was experimentally determined and is a compromise to enforce smoothness

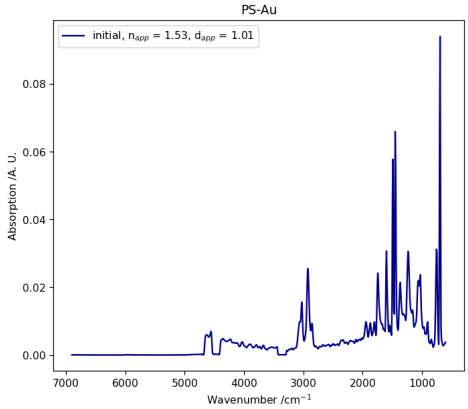
and avoid over- or under-regulating which would influence the absorption spectrum.

It was also observed that in some cases the imaginary part of the refractive index kept growing with each iteration of the fit. A second penalty was introduced for the magnitude of the imaginary part of the refractive index ($\text{coeff}_{weight}$). This was set to $10^{-5}$, to regulate the absorption index and avoid large values, as a higher absorption index increases the error. This parameter was chosen, due to the fact that it achieved the most optimal result and ensures numerical stability by preventing extreme values without restricting the flexibility of the algorithm.

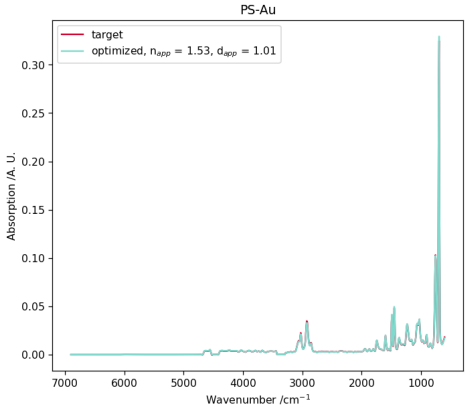
Finally, the maximum number of iterations ($\max_{nfev}$) was set to 30 for the absorption index and 15 for the subsequent step. This parameter determines the number of iterations prior to termination and account for the computational budget. The number of iterations for the absorption index failed to converge and setting it to a higher number did not yield a better result, thus it was set to this limit as it achieved the most plausible result. Furthermore, setting $\max_{nfev}$ higher also had an impact on the computation time, impacting the speed of the algorithm negatively. In contrast, the refractive index and polymer layer thickness reached the criteria and were aborted beforehand. These modifications collectively contributed to a more efficient optimization process.

To validate the accuracy of the algorithm and the resulting spectrum, the absorption spectrum of the polymer was plotted alongside the optimized spectrum. Although these two spectra are not identical in nature, it acts as a comparison to ensure that all prominent peaks are preserved. Thus, variations in the absorption indices may remain.

For demonstration purposes, the polymer-coated Si and Au wafers with a layer thickness of 1 and 10 µm were selected. In the following graphs on the left-hand side the "initial" is depicted which represents the data before iterative refinement. On the right-hand side the target is depicted in red which refers to the expected graph of PS, derived from values reported in the literature, while the curve labeled as "optimized" in turquoise is the output graph.[2,40]

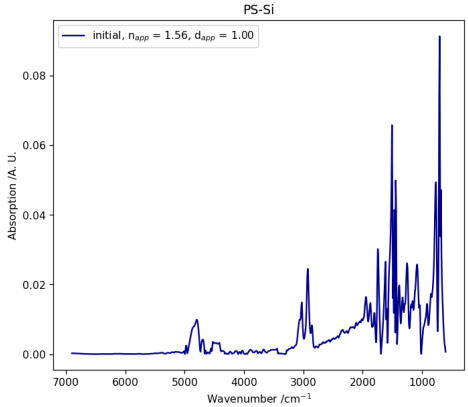



(a) Absorption spectrum of the initial simulated PS-Au model with a layer thickness of 1 µm.

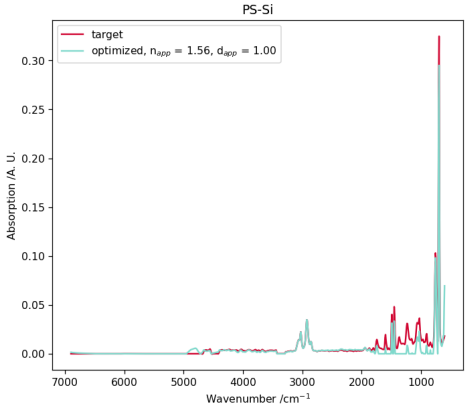



(b) The optimized absorption spectrum is plotted along side the target spectrum of PS.

Figure 9: For the PS-Au model with a polymer layer thickness of 1 µm the refractive index resulted in a value of 1.53 and the polymer layer thickness in 1.01 µm correlating well with the expected values.



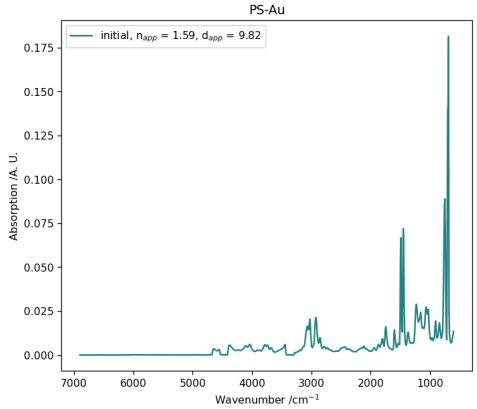

(a) Absorption spectrum of the initial simulated PS-Si model with a layer thickness of 1 µm before optimization



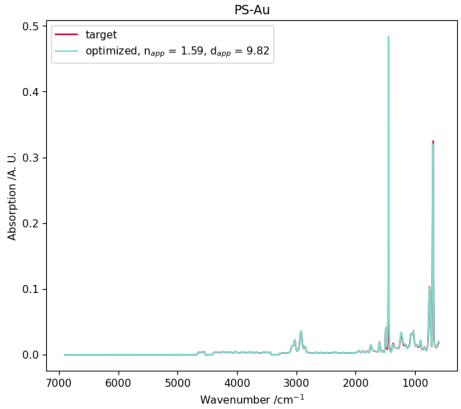

(b) Optimized absorption spectrum plotted along side the target spectrum of PS.

Figure 10: For the PS-Si model with a polymer layer thickness of 1 µm the refractive index resulted in a value of 1.53 and the polymer layer thickness in 1.01 µm correlating well with the expected values.

When comparing the figure 9 to 10, it is noticeable that the optimized spectrum of the gold substrate aligns more closely with the curve of the target spectrum, specifically in the fingerprint region, whereas the spectrum of the silicon substrate drops to the baseline after each band, however, the overall shape remained largely consistent.




(a) Absorption spectrum of the initial simulated PS-Au model with a layer thickness of 10 µm.




(b) Optimized absorption spectrum plotted alongside the target spectrum of PS.

Figure 11: For the PS-Au model with a polymer layer thickness of 10 µm the refractive index resulted in a value of 1.59 and the polymer layer thickness in 9.82 µm correlating well with the expected values.

The optimized graph in figure 11b contains all the prominent bands that are characteristic of the polymer, however, the peak at 1400 cm$^{-1}$ is a lot higher. This is illustrated more clearly in figure 12.

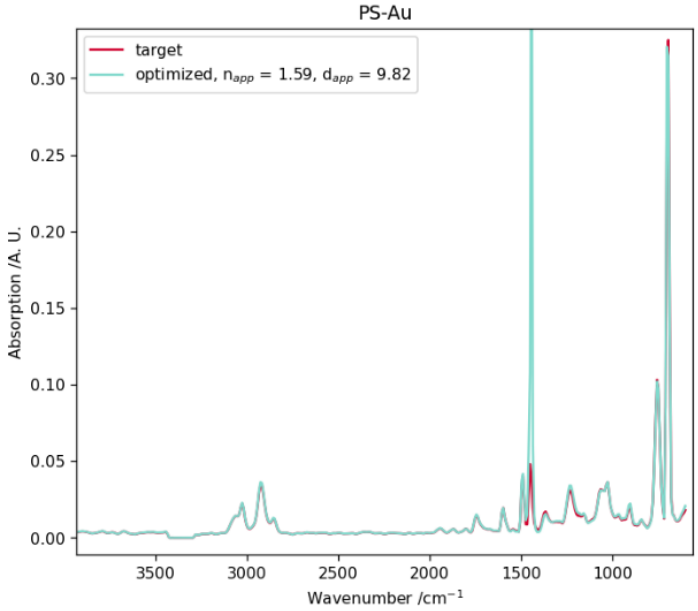
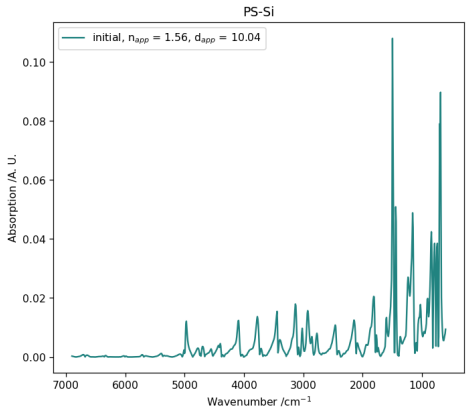
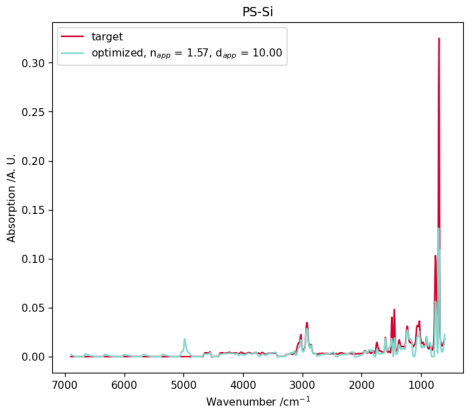



Figure 12: This is a close up view of the absorption spectrum of the PS coated gold wafer with a polymer layer thickness of 10 µm.




(a) Absorption spectrum of the initial simulated PS-Si model with a layer thickness of 10 µm.




(b) Optimized absorption spectrum plotted alongside the target spectrum of PS.

Figure 13: For the PS-Si model with a polymer layer thickness of 10 µm the refractive index resulted in a value of 1.57 and the polymer layer thickness in 10 µm correlating well with the expected values.

The optimization of the absorption spectrum of PS-Si with a polymer layer thickness of 10 µm was only partially effective. When comparing the figure 13a with 14, the fringes are less pronounced and the most prominent peaks are visible, however, the ultimate goal of removing the fringes was unsuccessful even though a refractive index of 1.57 and a polymer layer thickness of 10 µm was achieved. A zoomed in view of this spectrum is visible in figure 14.

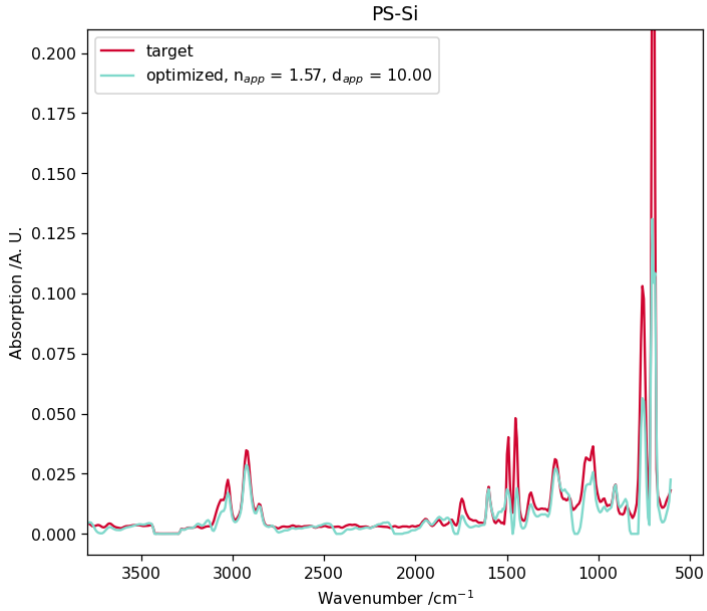
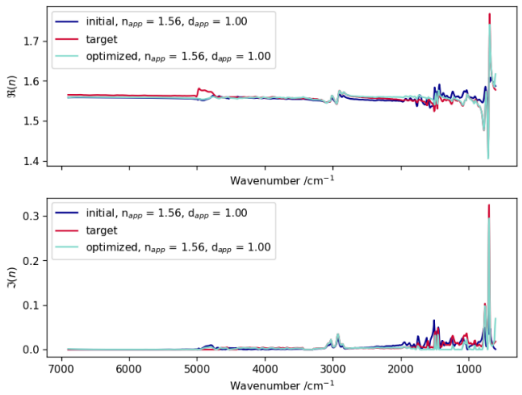


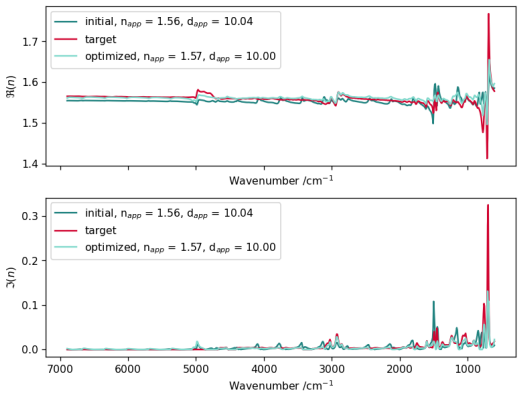

Figure 14: A closer view of the corrected absorption spectrum of the simulated PS-Si model.

The graphs in figures 9-13 exhibit a high degree of similarity. The refractive index and polymer layer thickness remained quite similar when rounded to the second decimal number, illustrating that the initial value was already approximated well. These values are summarized in table 3. Upon closer inspection, the simulated PS-Si model with a polymer layer thickness of 10 µm still displays faintly visible fringes after optimization.

In addition, the imaginary and real part of the optimization process can be plotted, reflecting the quality of the optimization process. These plots relate to each other via the Kramers-Kronig relation. The plots for the silicon model were chosen, as the optimized spectra have a slightly higher deviation from the target spectrum.




(a) PS-Si model with a layer thickness of 1 µm.




(b) PS-Si model with a layer thickness of 10 µm

Figure 15: The imaginary and real part of the the absorption spectrum of the PS-Si models.

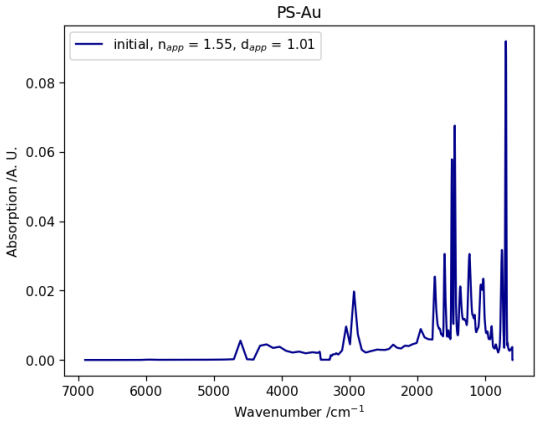
The curves in the figures 15 overlap, indicating a successful optimization process, despite the discrepancies in the absorption spectra. However, the curves are not identical, because the optimized spectrum reflects the bulk properties of the polymer, including contributions from its refractive index and absorption process which results in different optical behavior.

Furthermore, the algorithm exhibits instability because the iterative refinement of the absorption index fails to converge. The optimization was manually terminated after 30 iterations. Allowing a significantly higher number of iterations may lead to divergence and falsify the accuracy of the determined parameters.

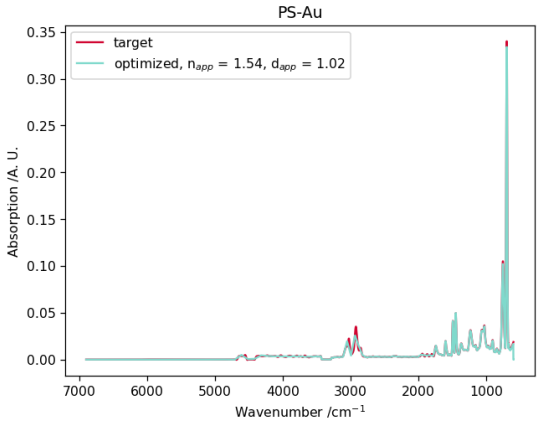
The optimization process demonstrates that the closer the initial values for the reflective index and polymer layer thickness are to their expected values, the fewer iterations the algorithm performs to converge to the solution.

To improve the algorithm further, the degrees of freedom to be optimized were reduced by applying spline interpolation to the data. Different spline knot spacings were chosen across the spectrum, depending on the size of the spectral features. The wavenumber range was divided into 4 intervals with different knot spacings, namely 590-1750 $\mathrm{cm}^{-1}$ with 4 $\mathrm{cm}^{-1}$ spacing, 1751-2800 $\mathrm{cm}^{-1}$ with 30 $\mathrm{cm}^{-1}$ spacing, 2801-3600 $\mathrm{cm}^{-1}$ with a mean of 4 $\mathrm{cm}^{-1}$ spacing and 3601-6900 $\mathrm{cm}^{-1}$ with 60 $\mathrm{cm}^{-1}$ spacing. These specifications were selected, so that the fingerprint region and the peak at 2900 $\mathrm{cm}^{-1}$ preserved their accuracy. In the other two intervals the inclusion of more wavenumbers for the calculation could be taken without losing any information and enabling the removal of the fringes with fewer degrees of freedom. The results are

depicted in figures 16a-19b.

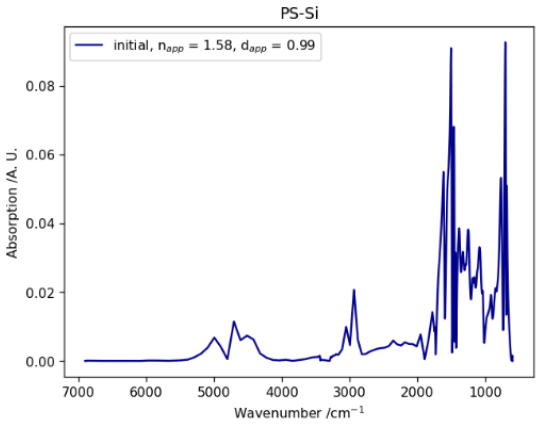



(a) Absorption spectrum of the initial simulated PS-Au model with a layer thickness of 1 µm before optimization.

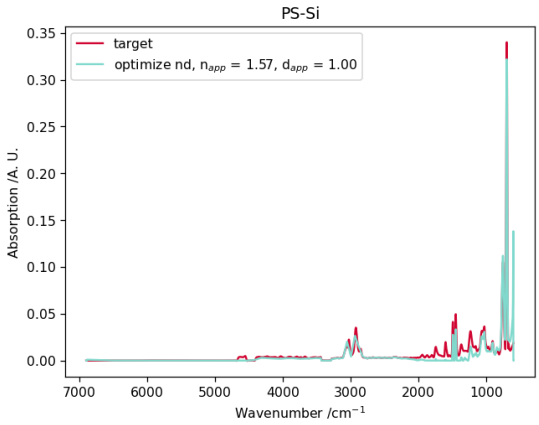



(b) Optimized absorption spectrum of PS-Au, plotted alongside its fit.

Figure 16: The obtained values for the refractive index and polymer layer thickness are 1.54 and 1.02 µm respectively, which are close to the expected values for the PS-Au model.



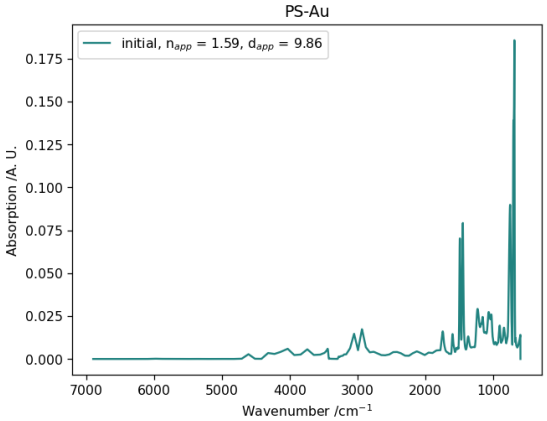

(a) Illustration of the initial PS-Si plot with a layer thickness of 1 µm before optimization.



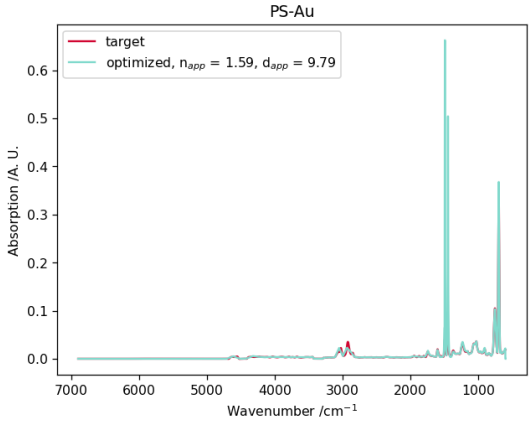

(b) Depiction of the obtained absorption spectra from the optimization process.

Figure 17: The obtained values for the refractive index and polymer layer thickness are 1.57 and 1.00 µm respectively, which are close to the expected values for the PS-Si model.

The PS-Si model with a layer thickness of 1 µm in figure 17 shows an improvement. The optimized spectrum follows the curve more closely without the trend towards the baseline after each band.

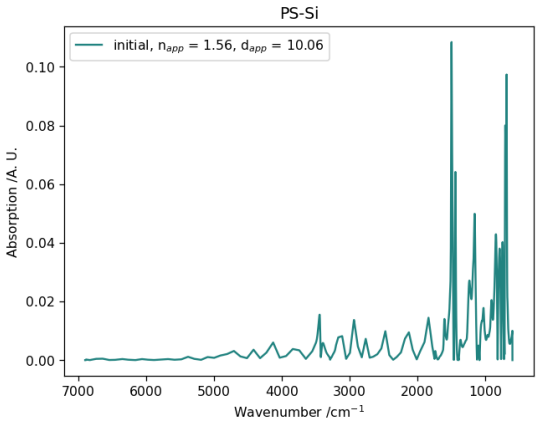



(a) Absorption spectrum of the initial PS-Au model with a layer thickness of 10 µm.

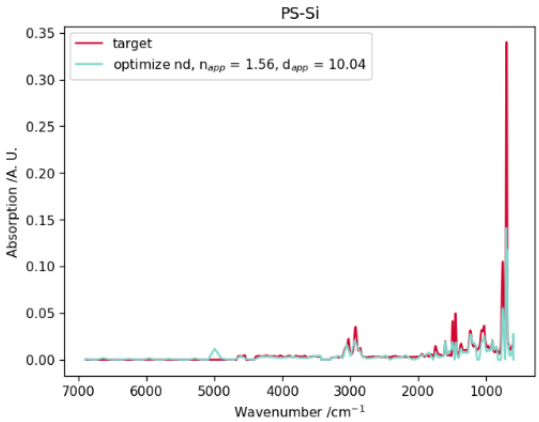



(b) Optimized absorption spectrum of PS-Au, plotted alongside its fit.

Figure 18: The obtained values for the refractive index and polymer layer thickness are 1.59 and 9.79 µm respectively, which are close to the expected values for the PS-Au model.




(a) Illustration of the initial PS-Si plot with a layer thickness of 10 µm before optimization.




(b) Depiction of the obtained absorption spectra from the optimization process.

Figure 19: The obtained values for the refractive index and polymer layer thickness are 1.56 and 10.04 µm respectively, which are close to the expected values for the PS-Si model.

The spectra of the simulated PS-coated gold wafer closely overlap with the optimized spectra. The obtained refractive index and polymer layer thickness values are identical to the expected values listed in table 3, indicating a good approximation of the target spectra. Moreover, the fringes were successfully removed.

Moreover, the PS-coated Si-wafer spectra in figure 17 and 19 show a good correlation with the target spectrum. The obtained refractive index and polymer layer thickness

are optimal, which would suggest a good approximation, however, in graph 19b, which displays the polymer with a layer thickness of 10 µm on the silicon substrate the fringes are still slightly visible. To provide a clearer visualization, a close-up view is shown in figure 20. The fringes are not as prominent as in the curve displaying the initial data, which is being corrected.

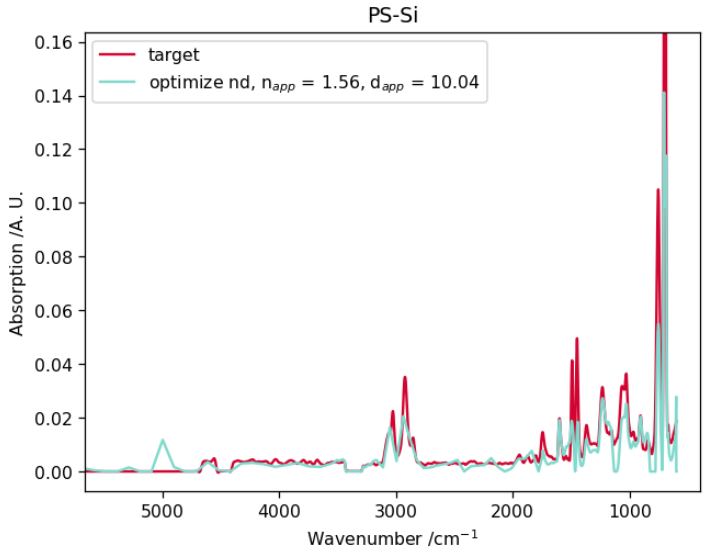



Figure 20: Absorption spectra of the PS-Si coated wafer sample after the optimization procedure.

The calculated refractive index and polymer layer thickness of the samples in figures 16-19 are summarized in table 3.

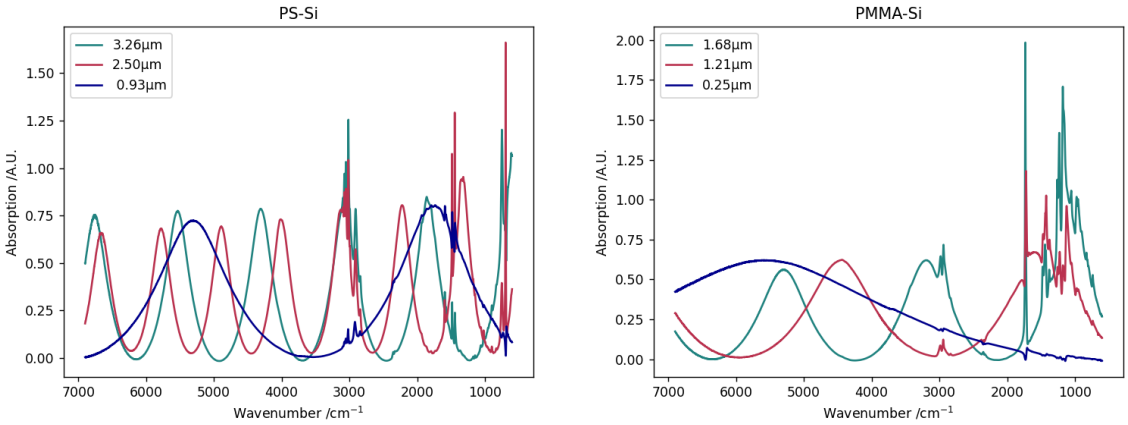
Table 3: Comparison of the optimization results with and without spline correction for different polymer layer thicknesses on silicon and gold substrates.

| | Without Spline | | With Spline | |
|---|---|---|---|---|
| **Parameter** | n | d ($\mu$m) | n | d ($\mu$m) |
| **PS-Si (1 µm)** | 1.56 | 1.00 | 1.57 | 1.00 |
| **PS-Si (10 µm)** | 1.57 | 10.00 | 1.56 | 10.04 |
| **PS-Au (1 µm)** | 1.53 | 1.01 | 1.54 | 1.02 |
| **PS-Au (10 µm)** | 1.59 | 9.82 | 1.59 | 9.79 |

The obtained results for the refractive index and polymer layer thickness of both optimization methods are accurate and are close to the expected value.

## 3.2 PS and PMMA samples

The fabricated samples were measured with the Hyperion 3000-MCT with a resolution of $2\,\mathrm{cm}^{-1}$ and 64 scans. The obtained absorption spectra are illustrated in the following figure 22.




(a) Illustration of the spin-coated PS-Si samples with varying polymer layer thicknesses.

(b) Illustration of the spin-coated PMMA-Si samples with varying polymer layer thicknesses.

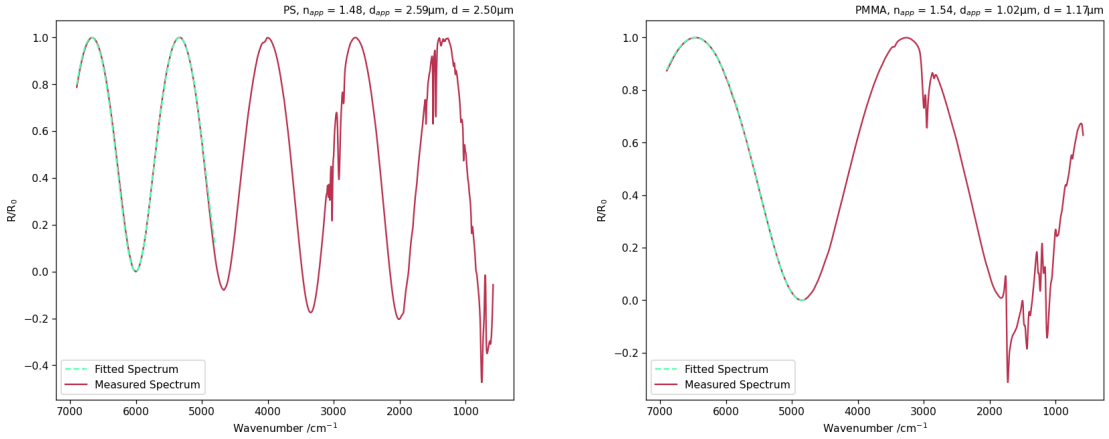
Figure 21: Absorption spectra of the PMMA and PS samples.

To illustrate the optimization algorithm, the PS-Si sample with a layer thickness of 2.50 µm and the PMMA-Si sample with a thickness of 1.17 µm were selected for the entire optimization process.

### 3.2.1 Determination of Refractive Index and Polymer Layer Thickness

The absorption spectra of PMMA and PS coated Si wafers were subsequently fitted to refine the values of the refractive index and polymer layer thickness. Within the fit, these two parameters were constrained by specifying appropriate bounds. These were set to 1.2 and 1.8 for the refractive index, with an initial value of 1.5. For the polymer layer, the bounds were set to 0 and 4 µm and the initial value was based on the predetermined thickness.

Additionally, this procedure utilized the least squares and shgo (simplicial homology global optimization) algorithm, which are part of SciPy's optimization modules, to iteratively refine the values for the refractive index and polymer layer thickness.




(a) The reflection spectrum of PS-Si wafer with a layer thickness of 2.50 µm is visualized alongside its fit.

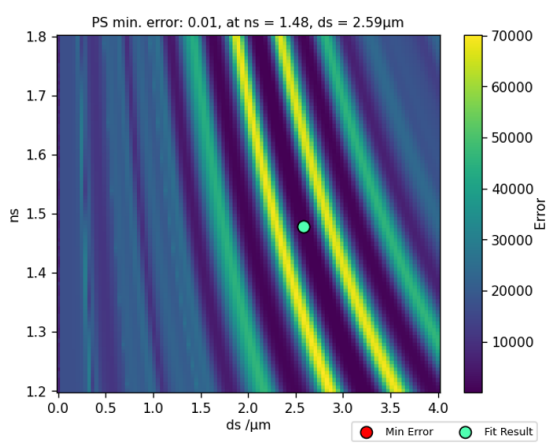
(b) The reflection spectrum of PMMA-Si wafer with a layer thickness of 1.17 µmis visualized along side its fit.

Figure 22: Illustration of the fit result.

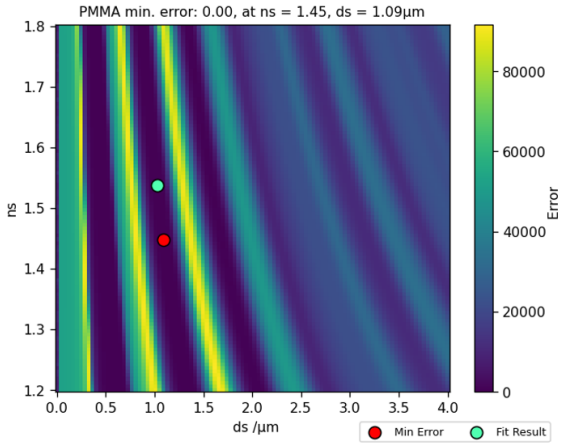
The obtained values, summarized in table 4, deviate from the expected values, which could be attributed to the use of an approximate angle of 17°. Therefore, apertures were developed to confine the angle of incidence and decrease the error within the algorithm.

### 3.2.2 Possible Fit Results

Due to the nature of the optimization process, there are several possible results. The refractive index and polymer layer thickness are interconnected in a complex manner, thus, it creates the appearance that an increase in one parameter leads to a decrease in the other. To illustrate this relationship, a heatmap was generated for the PS and PMMA coated silicon wafers. The heatmap represents the residual error between the target spectrum and the modeled spectrum. To obtain a single error per combination of these two parameters, the squared sum of the residuals was used. The parameter space was defined with bounds of 1-2 for the refractive index and 0-4 µm for the layer thickness, as these limits correspond to the constraints applied during the fit. The color intensity corresponds to the magnitude of each error, darker regions correspond to lower errors, while lighter regions have a higher error.




(a) Visualization of the error grid of PS-Si with a polymer layer thickness of 2.50 µm..




(b) Visualization of the error grid of PMMA-Si with a polymer layer thickness of 1.17 µm.

Figure 23: The heatmap displays the minimum error between the refractive index (ns) and the polymer layer thickness (ds) for each sample accordingly.

The obtained values for the refractive index and polymer layer thickness are depicted in table 4.

Table 4: Results of the minimum error from the heatmap and of the fit for a PMMA and PS coated silicon wafer with the bounds set to n: 1.2-1.8 and d: 0-4 µm.

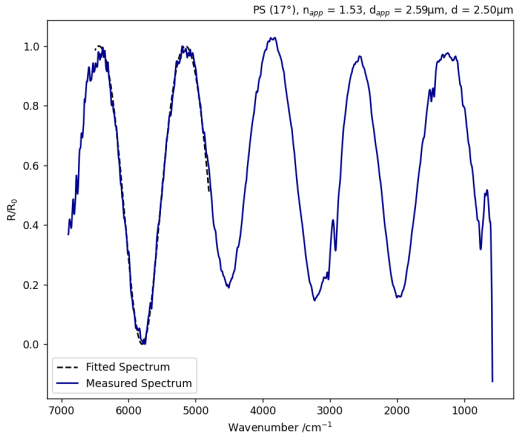
| | fit result | | | minimum error | | |
|---|---|---|---|---|---|---|
| **Parameter** | n | d ($\mu$m) | % deviation | n | d ($\mu$m) | % deviation |
| **PS (2.50 µm)** | 1.48 | 2.59 | 3.60 | 1.48 | 2.59 | 3.60 |
| **PMMA (1.17 µm)** | 1.52 | 1.02 | -12.82 | 1.45 | 1.09 | -6.84 |

The results of the fit and the minimum error for PS are identical, but deviate for PMMA. The result of the fit is lower than the obtained value from the minimum error. However, while the fit for PMMA resulted in a lower polymer layer thickness it has a higher refractive index than the result for the minimum error.

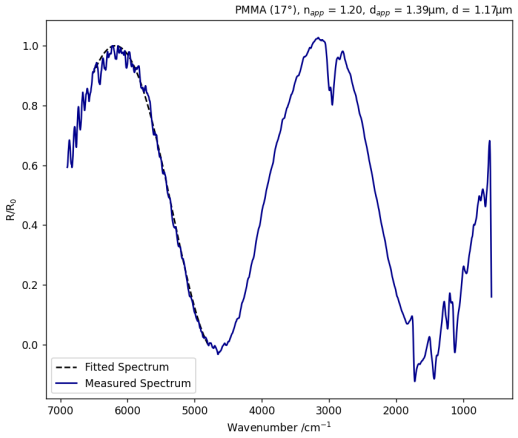
### 3.2.3 Application of Apertures

The 3D-printed apertures were implemented to narrow the range of the angle of incidence and to increase precision due to the confinement of the angle. Thus, depending on which aperture was being used, the angle of incidence was changed to either 17°, 20° or 22°. The spectra were measured at a resolution of 8 $cm^{-1}$ and 256 scans.

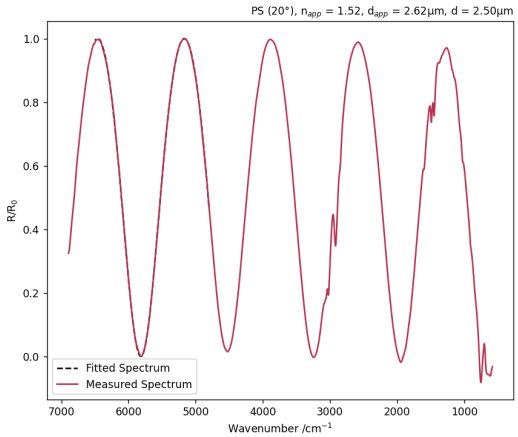
Furthermore, the bounds remained identical to those used for the fit without apertures, hence, they were defined as 1.2-1.8 for the refractive index and 0-4 µm for the polymer layer thickness. By applying these constraints, the following graphs were obtained.



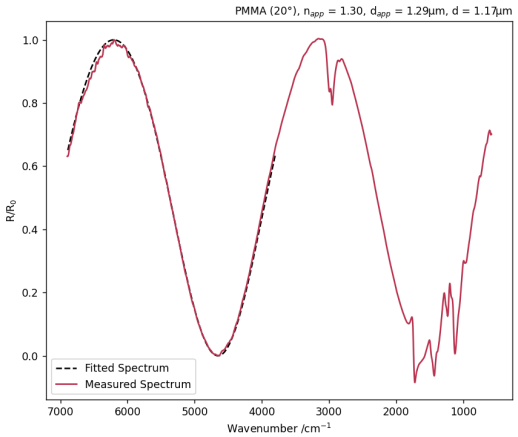

(a) The reflection spectrum alongside its fit of the PS-coated Si wafer at an angle of 17°. The obtained refractive index and polymer layer thickness are 1.53 and 2.59 µm respectively, aligning well with the expected values.



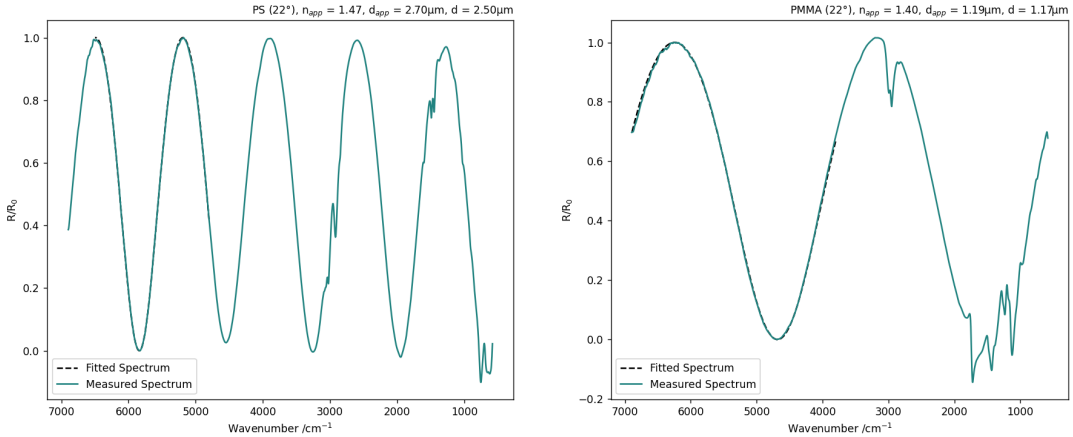

(b) The reflection spectrum alongside its fit of the PMMA-coated Si wafer at an angle of 17°. The obtained refractive index and polymer layer thickness are 1.20 and 1.39 µm respectively, while the refractive index is slightly below the expected range the polymer layer thickness matches closely to 1.17 µm.




(c) The reflection spectrum alongside its fit of the PS-coated Si wafer at angle of 20°. The obtained refractive index and polymer layer thickness are 1.52 and 2.62 µm respectively, matching the expected values.




(d) The reflection spectrum alongside its fit of the PMMA-coated Si wafer at an angle of 20°. The obtained refractive index and polymer layer thickness are 1.30 and 1.29 µm respectively, differing from the expected values.




(e) The reflection spectrum alongside its fit of the PS-coated Si wafer at an angle of 22°. The obtained refractive index and polymer layer thickness are 1.47 and 2.70 µm respectively, matching with the expected values

(f) The reflection spectrum alongside its fit of the PMMA-coated Si wafer at an angle of 22°. The obtained refractive index and polymer layer thickness are 1.40 and 1.19 µm respectively, aligning well with the expected values

Figure 24: Application of the apertures on one of the PS and PMMA samples and illustrating the resulting fit.

To illustrate the results of the fit in figure 24, the obtained refractive index and polymer layer thickness are summarized in the table 5.

Table 5: Fit results for a PMMA and PS coated silicon wafer with the bounds set to n: 1.2-1.8 and d: 0-4 µm.

| **Aperture** | none | | 17° | | 20° | | 22° | |
|---|---|---|---|---|---|---|---|---|
| **Parameter** | n | d ($\mu$m) | n | d ($\mu$m) | n | d ($\mu$m) | n | d ($\mu$m) |
| **PS (2.50 µm)** | 1.48 | 2.59 | 1.53 | 2.59 | 1.52 | 2.48 | 1.47 | 2.70 |
| **PMMA (1.17 µm)** | 1.54 | 0.92 | 1.20 | 1.19 | 1.30 | 1.21 | 1.40 | 1.19 |

Table 6: Deviation percentages of the polymer layer thickness of each aperture from the table 5 for the PMMA and PS sample.

| **Apertures** | 17° (% dev.) | 20° (% dev.) | 22° (% dev.) |
|---|---|---|---|
| **PS (2.50µm)** | 3.60 | -0.80 | 8.00 |
| **PMMA (1.17µm)** | 1.71 | 3.42 | 1.71 |

Even though the depicted fits align excellently with the spectrum, the refractive index and polymer layer thickness deviate slightly from the expected values. For

the PS coated silicon wafer, the obtained thickness values deviate more than for the PMMA sample. For both samples the average refractive index is 1.40 which is slightly lower than the obtained values of the simulations in section 3.1.3, which achieved an average value of 1.57.

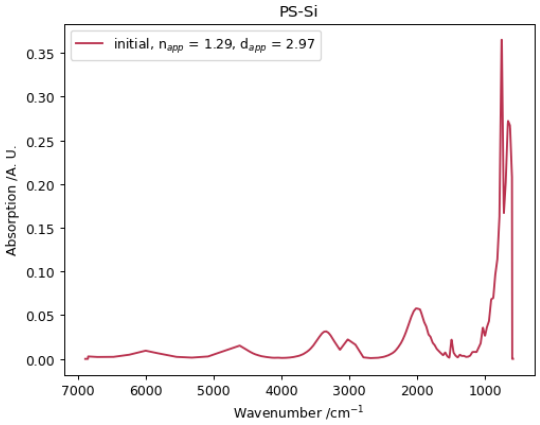
Furthermore, when the obtained results for the PMMA-Si sample is compared to those in table 4, it is evident that the determined values with the implementation of the apertures result in more accurate values. While the obtained values for the PS-Si sample fluctuate and has the most optimal result when the angle of incidence is confined to 20°.

### 3.2.4 Correction of the Absorbance

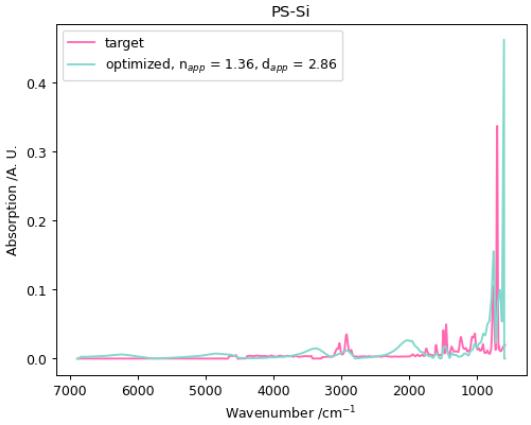
For further improvement, the spectra were corrected by iteratively refining the absorption index, refractive index and polymer layer thickness, as outlined in sections 1.4.1-1.4.3.

Furthermore, to improve the algorithm further, the degrees of freedom to be optimized were reduced by applying spline interpolation to the data. Different spline knot spacings were chosen across the spectrum, depending on the size of the spectral features. The wavenumber range was divided into four intervals with different knot spacings, namely 590-2000 $\mathrm{cm}^{-1}$ with 8 $\mathrm{cm}^{-1}$ spacing, 2001-3200 $\mathrm{cm}^{-1}$ with a mean of 30 $\mathrm{cm}^{-1}$ spacing, 3201-4500 $\mathrm{cm}^{-1}$ with a mean of 8 $\mathrm{cm}^{-1}$ spacing and 4501-6900 $\mathrm{cm}^{-1}$ with 30 $\mathrm{cm}^{-1}$ spacing. These specifications were selected, so that the fingerprint region and the peak at 2900 $\mathrm{cm}^{-1}$ preserved their accuracy. In the other two intervals the inclusion of more wavenumbers for the calculation could be taken without losing any information and enabling the removal of the fringes with fewer degrees of freedom. The results are depicted in figures 25-26.

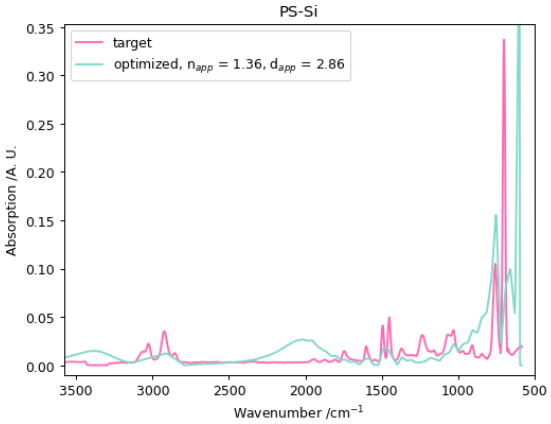
For the implementation of the optimization procedure, the measured spectrum of the PS and PMMA sample without aperture was used.




(a) Absorption spectrum of the PS-Si.



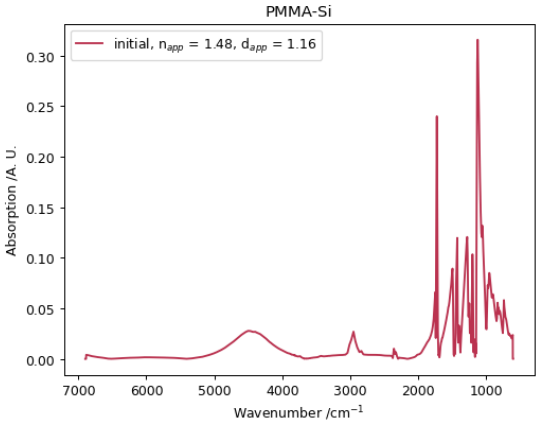

(b) Optimized absorption spectrum of the PS-Si sample.



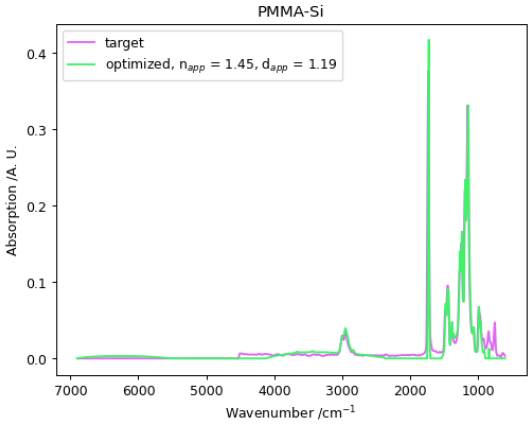

(c) Close up view of the absorption spectrum of the PS-Si sample.

Figure 25: The resulting graphs for the PS coated Si-wafer after having applied the optimization algorithm with a polymer layer thickness of 2.50 µm.

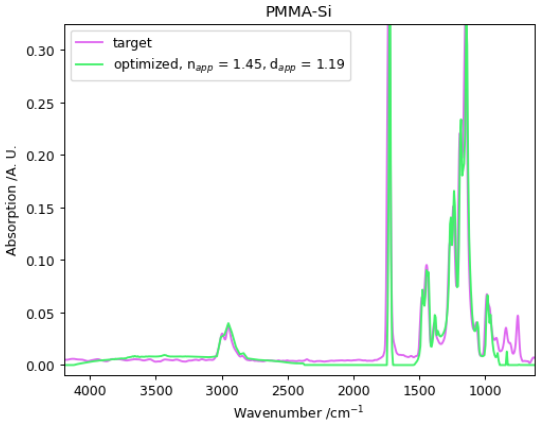
The fringes were successfully removed from the absorption spectrum of the PS coated Si-wafer, but not all of the bands were preserved (fig. 25). Furthermore, due to the iterative optimization process, the value of the refractive index was slightly improved from 1.29 to 1.36 and the layer thickness was improved from 2.97 µm to 2.86 µm, deviating by 14.40% from the measured layer thickness of 2.50 µm.




(a) Absorption spectrum of the PMMA-Si sample.




(b) Optimized absorption spectrum of the PMMA-Si sample.



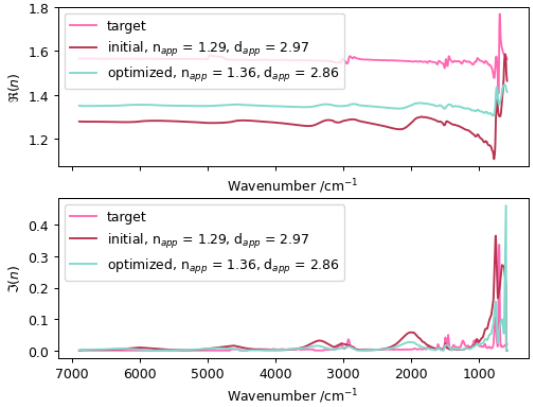

(c) Close up view of the absorption spectrum of the PMMA-Si sample.

Figure 26: The resulting graphs for the PMMA coated Si-wafer after having applied the optimization algorithm with a polymer layer thickness of 1.17 µm

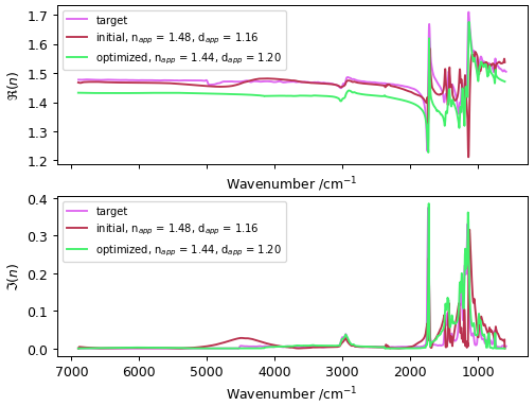
The fringes were successfully removed from the absorption spectrum of the PMMA-Si sample (fig. 26). The data for the target spectrum for PMMA was taken from literature.[2] When comparing the initial spectrum depicted in red to the approximated spectrum in green, the fringes have been successfully removed. Furthermore, due to the iterative optimization process, the value of the refractive index was improved from 1.48 to 1.45 and the layer thickness was improved from 1.15 µm to 1.19 µm, deviating by 1.71% from the measured layer thickness of 1.17 µm.

A comparison of figure 25 and 26 illustrates that the approximation algorithm works more effectively for PMMA as the polymer coat because the obtained spectrum aligns closer with its target. This is also shown in the plot of the real and imaginary part of the absorption spectrum in figure 27. This could be due to the fact that

PMMA has fewer fringes, which require removal.




(a) PS-Si sample with a polymer layer thickness of 2.50 µm.




(b) PMMA-Si sample with a polymer layer thickness of 1.17 µm.

Figure 27: The real and imaginary part of the the absorption spectrum of PS and PMMA sample on silicon substrate.

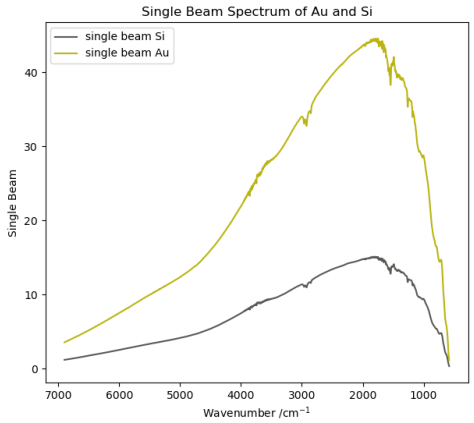
Figure 27 highlights the fact that the optimization procedure for PMMA was successful, because the curves overlap in the real part, while in the graph of the PS sample they lie further apart.

## 3.3 Sample A

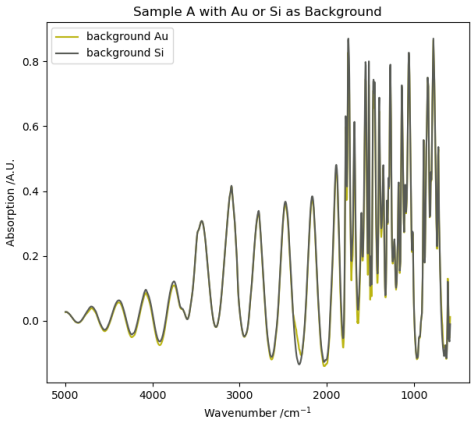
### 3.3.1 Absorption Spectrum

Ordinarily, the substrate of the sample was chosen as the background for each measurement. However, for simplicity, the industrial partner chose gold due to its high reflectivity. This difference in reflectivity between gold and silicon is depicted in figure 28a.

To validate the assumption that the difference between gold and silicon as background measurement is negligible, sample A was measured with gold as background. Subsequently, the same spectrum was reprocessed with the software OMNIC Picta from Thermo Scientific. For the reprocessing of this spectrum the single beam of the background measurement was modified to that of the substrate illustrated in figure 28b.




(a) Single beam spectrum of Au and Si which have undergone atmospheric correction.




(b) This plot illustrates the change in the spectra when the background is changed from silicon to gold.

Figure 28: Visualization of the impact of the background measurement on the absorption spectra of sample A.

As depicted in figure 28b, the absorption spectra of sample A are almost identical, suggesting that there is no difference in using gold or silicon as background signal which is automatically subtracted by the FTIR instrument's software (Omnic Picta). As a consequence, the gold background spectrum can be used interchangeably with silicon.

Furthermore, by implementing the apertures, the following graph (figure 29) was obtained.

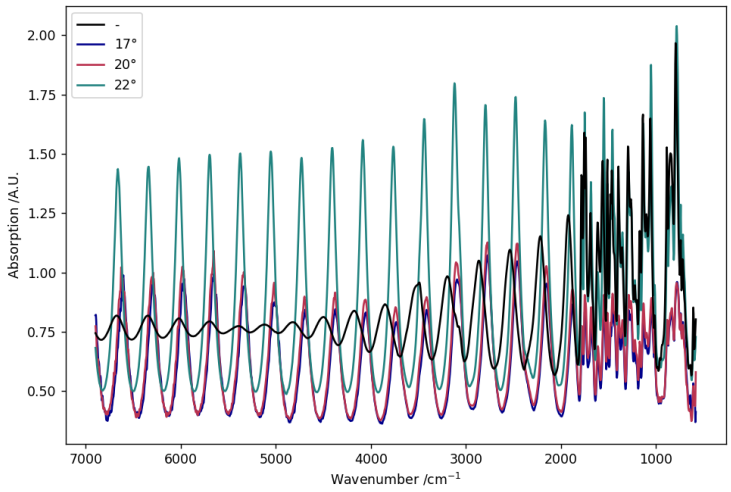



Figure 29: Absorption spectra of sample A.

The black line represents the spectrum of sample A measured without restricting the angle of incidence, whereas the other three lines correspond to measurements obtained using apertures. The apertures significantly affect the peak intensities, which increase as the angle of incidence becomes larger and are more uniform in the higher wavenumber region.

The baseline was corrected with the arPLS algorithm ($\lambda = 1 \times 10^5$) to facilitate the fitting of each spectrum, and figure 30 was obtained.

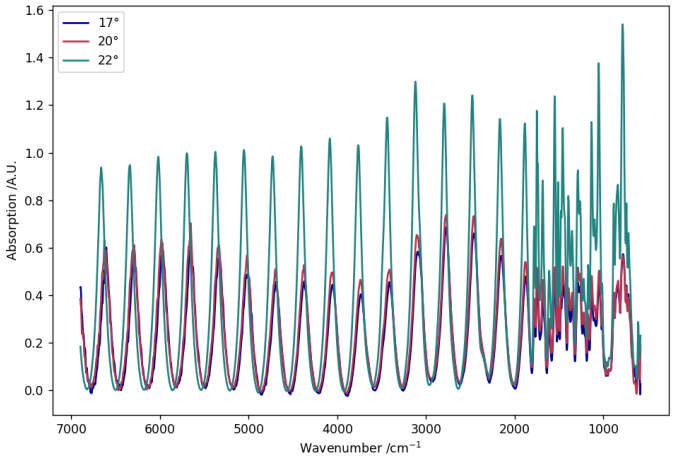
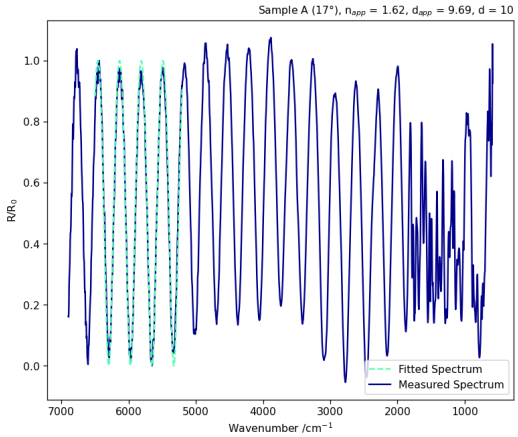



Figure 30: Baseline corrected absorption spectra of sample A.

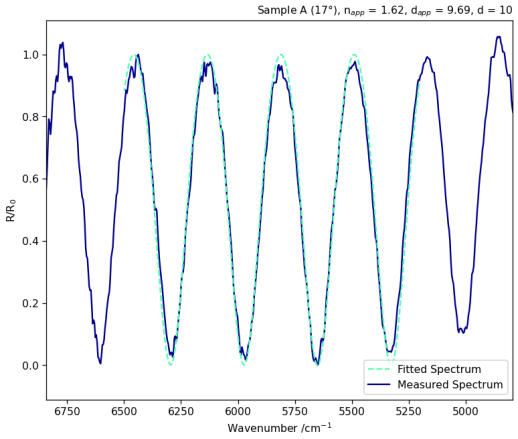
### 3.3.2 Determination of Refractive Index and Polymer Layer Thickness

Sample A was fitted to refine the values of the refractive index and polymer film thickness. The fit was constrained by specifying appropriate bounds for these two parameters. The refractive index was set to 1 and 2, with an initial value of 1.5. For the polymer layer, the bounds were set to 5 and 15 µm, with an initial value of 10 µm based on the provided layer thickness from the industrial partner. These bounds were chosen, as a larger search space enables a higher flexibility in determining the values correctly. Moreover, the bounds for the refractive index were determined based on the simulation model of PS.

By applying the above-mentioned specifications to the algorithm and processing the measured spectra, the graphs below were obtained (figure 31-32). Each measurement was performed at the same spot on the sample to facilitate a direct comparison of the different apertures and minimize the variability due to sample inhomogeneity.

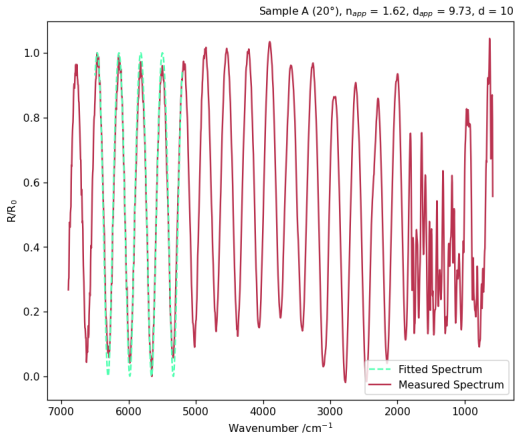



(a) Reflection spectrum of sample A, along with its fit.

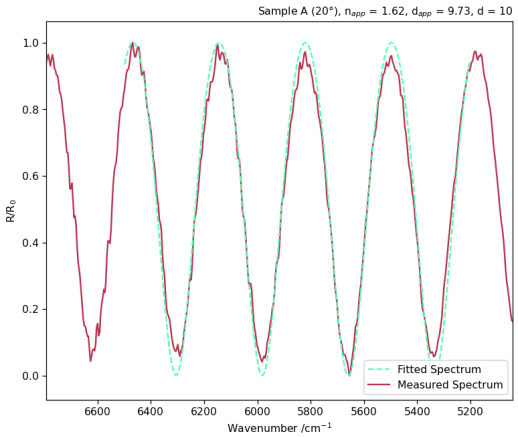



(b) A zoomed in view of the reflection spectrum for sample A and its fit.

Figure 31: Reflection spectra of sample A measured at a median angle of incidence of 17°. The obtained refractive index and polymer layer thickness are 1.62 and 9.69 µm, respectively.

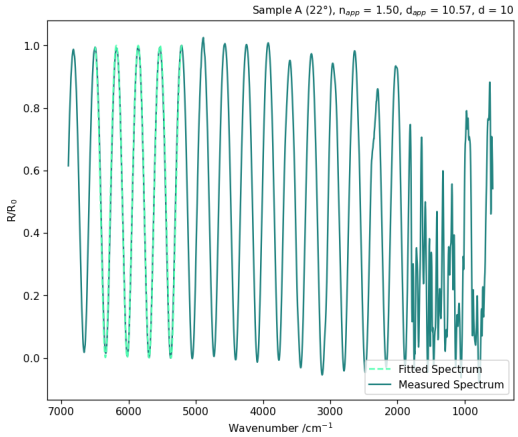



(a) Reflection spectrum of sample A, along with its fit.

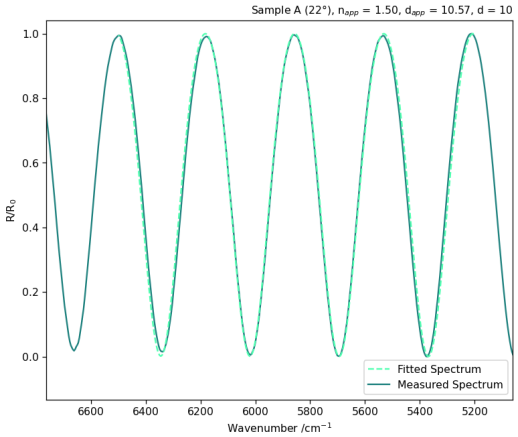



(b) A zoomed in view of the reflection spectrum for sample A and its fit.

Figure 32: Reflection spectra of sample A measured at a median angle of incidence of 20°. The obtained refractive index and polymer layer thickness are 1.62 and 9.37 µm, respectively.




(a) Reflection spectrum of sample A, along with its fit.




(b) A zoomed in view of the reflection spectrum for sample A and its fit.

Figure 33: Reflection spectra of sample A measured at a median angle of incidence of 22°. The obtained refractive index and polymer layer thickness are 1.50 and 10.57 µm, respectively.

The obtained fit values are summarized in table 7. The curve of the fit and the measured spectrum strongly align. In addition, a FIB cross-sectioning procedure was performed on sample A to determine the polymer layer thickness depicted in figure 34, resulting in a value of 10.15 µm. Since this analysis method damages the sample, the wafer was cut into two pieces. One piece was used for the FIB cross-sectioning procedure, while the other was used for the FTIR measurement. These two procedures were carried out in close proximity to the edge to minimize deviations caused by sample inhomogeneity and to obtain more accurate results.

Among the three apertures analyzed, the 22° aperture provides the best match, with a calculated polymer layer thickness of 10.17 µm that closely approximates the measured FIB cut value.

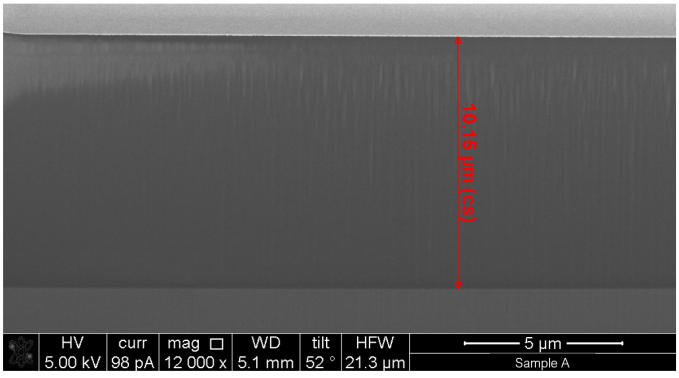
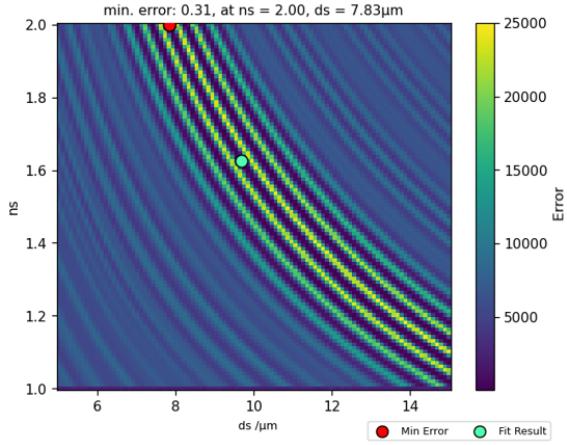


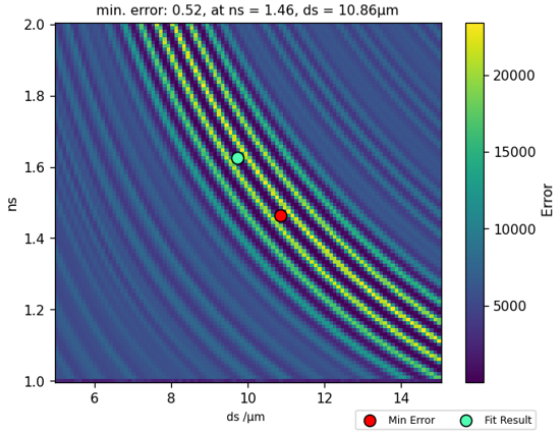

Figure 34: The cross-section of sample A with the polymer layer on top and silicon on the bottom, containing the measurement of the polymer layer thickness.

### 3.3.3 Possible Fit Results

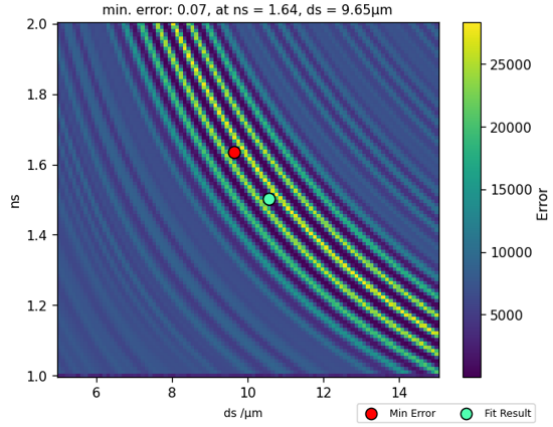
To illustrate the possible results, a residual error analysis between the target spectrum and the modeled spectrum was performed and depicted as a heatmap. The parameter space was selected as 1-2 for the refractive index and 5-15 µm for the polymer layer thickness, as these were the bounds for the fit.




(a) The error grid of sample A measured with the aperture of 17° is illustrated in this plot. The minimum error hereby lies at 2 and 7.83 µm.




(b) The error grid of sample A measured with the aperture of 20° is illustrated in this plot. The minimum error hereby lies at 1.46 and 10.86 µm.




(c) The error grid of sample A measured with the aperture of 22° is illustrated in this plot. The minimum error hereby lies at 1.64 and 9.65 µm.

Figure 35: The heatmap displays the minimum error between the refractive index (ns) and the polymer layer thickness (ds) for each simulation accordingly. The color intensity corresponds to the magnitude of each error, darker regions correspond to lower errors, while lighter regions have a higher error.

The plots in figure 35 are near identical to each other, highlighting the fact that there are several solutions in regard to the polymer layer thickness and refractive index which are linearly dependent. As the color intensity corresponds to the magnitude of each error, darker regions correspond to lower errors, while lighter regions have a higher error. Furthermore, with a higher refractive index, the layer thickness is lower. The calculated minimum error of the aperture of 22° is the closest to the obtained value from the fit results, while the minimum error of the other two heatmaps deviate.

Table 7: Results of the fit and of the minimum error of the heatmap for sample A with the bounds set to n: 1.2-1.8 and d: 5-15 µm. The measured polymer layer thickness was 10.15 µm.

| | fit result | | | minimum error | | |
|---|---|---|---|---|---|---|
| **Parameters** | n | d ($\mu$m) | % deviation | n | d ($\mu$m) | % deviation |
| **17°** | 1.62 | 9.69 | -4.53 | 2.00 | 7.83 | -22.86 |
| **20°** | 1.62 | 9.73 | -4.14 | 1.46 | 10.86 | 7.00 |
| **22°** | 1.50 | 10.57 | 4.14 | 1.64 | 9.65 | -4.93 |

When comparing the value obtained from the cross-section in figure 34 to the fit results presented in table 7, it reveals a good correlation with all aperture-based measurements. The deviation between the FIB cut and fitted polymer layer thickness is approximately 4%.

Meanwhile, the heatmap analysis differs from the fit results. In contrast, the values obtained for the aperture at 22° is show the closest match to the FIB cut measurement, with a deviation of 4.93%.

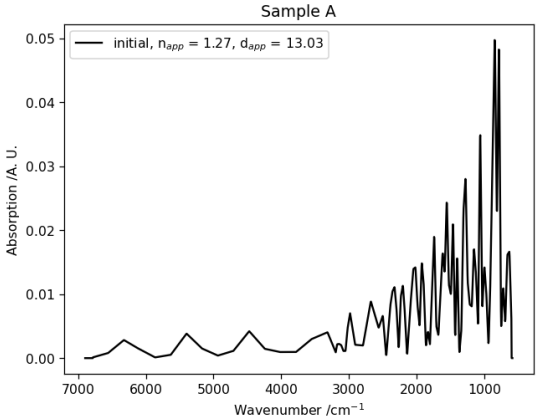
### 3.3.4 Correction of the Absorbance

For further improvement, the spectra were refined by employing the optimization algorithm as outlined in sections 1.4.1-1.4.3.

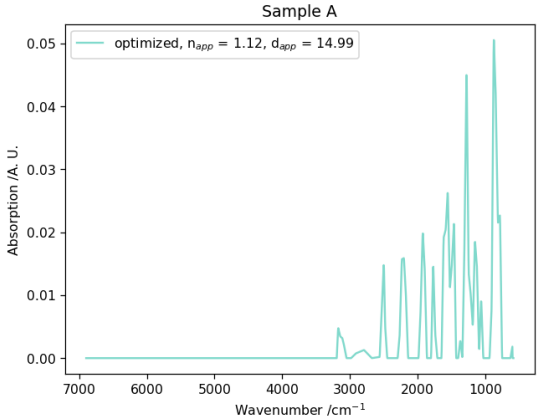
Different spline knot spacings were chosen across the spectrum, depending on the size of the spectral features. The wavenumber range was divided into 4 intervals with different knot spacings, namely 590-2500 $cm^{-1}$ with 8 $cm^{-1}$ spacing, 2501-3000 $cm^{-1}$ with 30 $cm^{-1}$ spacing, 3001-3200 $cm^{-1}$ with a mean of 8 $cm^{-1}$ spacing and 3201-6900 $cm^{-1}$ with 60 $cm^{-1}$ spacing. These specifications were selected, so that the fingerprint region and the peak at 3100 $cm^{-1}$ preserved their accuracy. In the other two intervals the inclusion of more wavenumbers for the calculation could be taken

without losing any information and enabling the removal of the fringes with fewer degrees of freedom.

The graphs in figure 36a-39b were obtained after undergoing the optimization procedure.

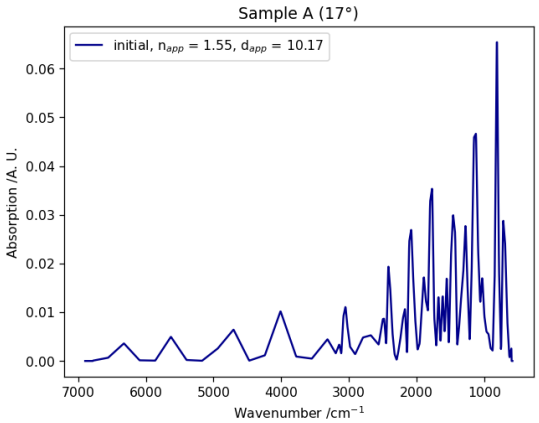



(a) Absorption spectrum of sample A.

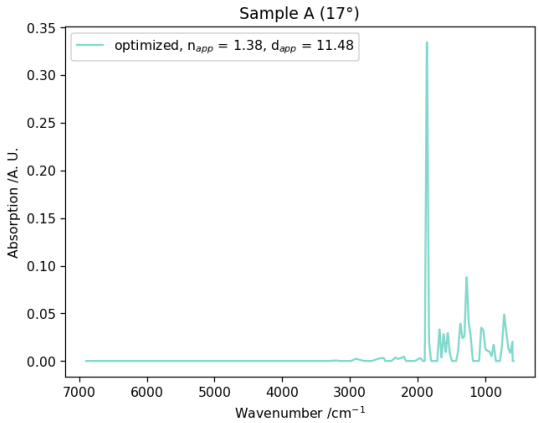



(b) Optimized absorption spectrum of sample A.

Figure 36: Comparison between the absorption spectra of sample A before and after the optimization procedure of the absorption spectrum achieving a refractive index of 1.12 and polymer layer thickness of 14.99 µm.

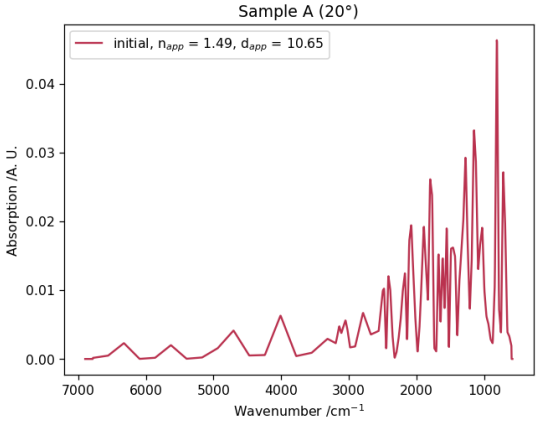



(a) Absorption spectrum of sample A measured at an angle of incidence of 17°.

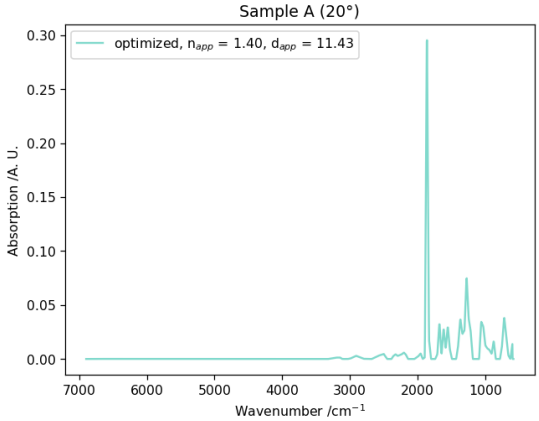



(b) Optimized absorption spectrum of sample A.

Figure 37: Comparison of the absorption spectra of sample A before and after optimization of the absorption spectrum reaching a refractive index value of 1.38 and a polymer layer thickness value of 11.48 µm.

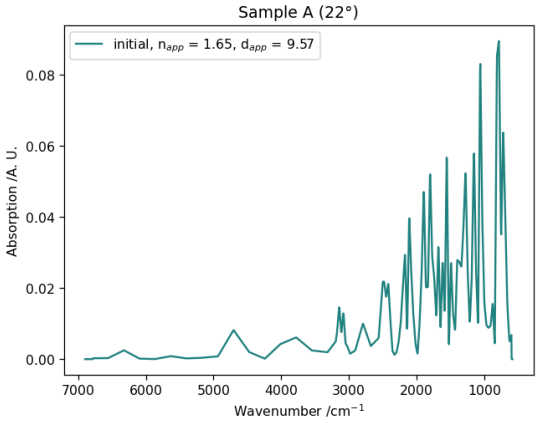



(a) Absorption spectrum of sample A measured at an angle of incidence of 20°

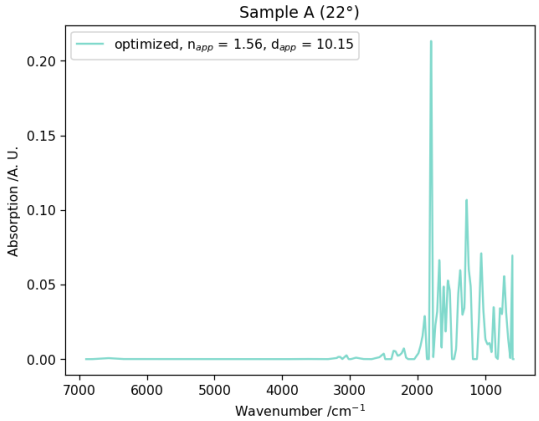



(b) Optimized absorption spectrum of sample A.

Figure 38: Comparison of the absorption spectra of sample A before and after optimization of the absorption spectrum reaching a refractive index value of 1.40 and a polymer layer thickness value of 11.43 µm.




(a) Absorption spectrum of sample A measured at an angle of incidence of 22°.

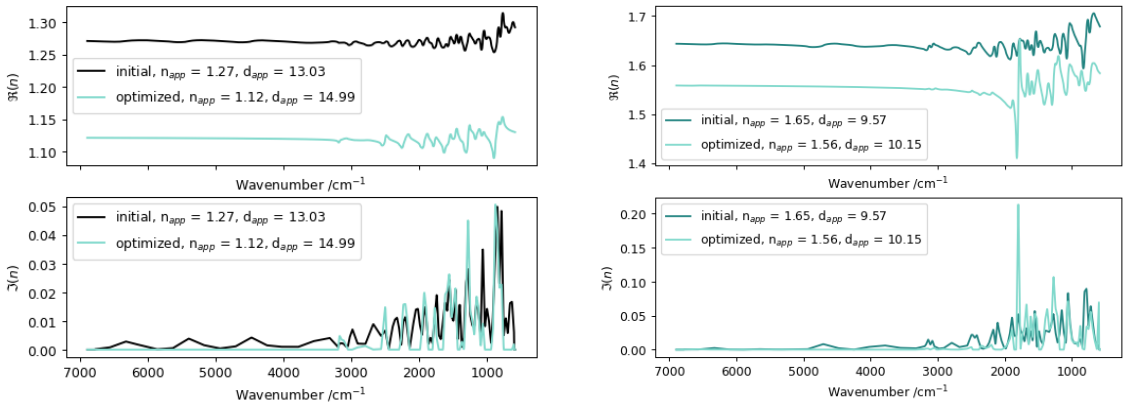



(b) Optimized absorption spectrum of sample A.

Figure 39: Comparison of the absorption spectra of sample A before and after optimization of the absorption spectrum reaching a refractive index value of 1.56 and a polymer layer thickness value of 10.15 µm.

When comparing the corrected spectra with each other, the graphs in the figure of the optimized spectrum all appear different even though it's the same polymer. Furthermore, the graph of sample A in figure 36b is similar to the one measured with a median incidence angle of 22° in figure 39b. However, in these two plots the peaks are in a different relation to each other, for example, the peak at 1900 cm$^{-1}$ is a lot higher in the spectrum measured with the aperture in comparison to sample

A in figure 36b. In addition, the imaginary and real parts were plotted to evaluate the performance of the optimization procedure.




(a) Sample A

(b) Sample A measured at a median angle of 22°.

Figure 40: The real and imaginary part of the absorption spectra of sample A.

The real part of sample A measured without aperture and at a median angle of 22° differ. For sample A in figure 40a the initial and optimized curve are further apart, suggesting that the optimization procedure was less successful.

Furthermore, the graph in figure 37b of the measured sample with a median angle of 17° is similar to the one measured at an angle of 20° in figure 38b. This illustrates that the algorithm is not successful in correcting each spectrum. When taking the results from the refractive index and polymer layer thickness into account, it is evident that the sample measured at a median angle of 22° reaches the expected values of 1.5 and 10.15 µm, while the other measurements differ. The obtained refractive index and polymer layer thickness are listed in the table 8.

Table 8: The refractive index and polymer layer thickness obtained for sample A after undergoing the optimization procedure.

| **Sample A** | n | d ($\mu$m) | % dev |
|---|---|---|---|
| **-** | 1.12 | 14.99 | 47.68 |
| **17°** | 1.38 | 11.48 | 13.10 |
| **20°** | 1.40 | 11.43 | 12.61 |
| **22°** | 1.56 | 10.15 | 0.00 |

The results for the refractive index and polymer layer thickness exhibit the largest deviation from the expected values of 1.5 and 10.15 µm for sample A when the

experimental setup was not adapted. This highlights that the use of the apertures results in a better approximation of these values.

Although the fitting approach in section 3.3.2 successfully determined the polymer layer thickness, the algorithm designed for correcting the absorption spectrum was not effectively executed, illustrating that this algorithm works better for thin polymer layers. Thus, the use of apertures proves beneficial in determining both the refractive index and the polymer layer thickness with high accuracy.

As discovered this algorithm does not work effectively for thicker samples, therefore, in the future a different approach could be implemented. For example, ellipsometry could be used instead, which can provide more data for the model due to having control over the angle of incidence and gaining more insight into the polarization of the incoming and transmitted light beam.

# 4 Conclusion and Outlook

The objective of this research project was to remove the interference fringes and correct the absorption spectrum via optical modeling. The algorithm was verified through simulations of PS and PMMA samples. A noticeable difference was observed when comparing gold and silicon as substrates, namely silicon caused more pronounced fringes.

Furthermore, the correction of the absorption spectrum was successfully executed for the PMMA-Si sample with a thin polymer layer, whereas for the PS-Si sample, it was not executed perfectly. While the fringes were removed, not all of the peaks were preserved. However, realizing this for sample A proved to be more challenging due to its thicker polymer layer.

The estimation of the polymer layer thickness and refractive index was accomplished via the fit algorithm. Initially, the median angle of incidence for the Hyperion 3000-MCT (17°) was applied, however, the analysis revealed that this approach was inadequate. Thus, to determine the angle of incidence and to decrease the error introduced by this estimation, apertures were 3D printed. By modifying the experimental setup it became apparent that the angle of incidence has an impact on the absorption spectrum. It affects the fringes and thus, implementing the correct angle within the fit algorithm provides a more accurate representation of the reality. Consequently, obtaining accurate values with minimal percentage deviations from the measured polymer layer thickness.

The hypothesis that defining the angle of incidence improves the correction of the spectra was shown to be inaccurate. While it did achieve an improved result for the polymer layer thickness, which is also reflected in the evaluation of the minimum error, it did not improve the correction of the absorption spectrum. This is because the correction of the absorption spectra depends not only on the angle of incidence but also on the noise in the data and the intrinsic properties of the material. These factors, therefore, impose limitations on the calculations performed by the algorithm.

Accurately approximating the polymer layer thickness is relevant for determining variations in the polymer layer thickness across an entire wafer. Such measurements are crucial within the production process of the wafers, as they enable quality control and verification of the consistency within the fabrication procedure. This method for determining these values is advantageous due to its ease of application and quick execution, resulting in lower resource usage and reduced time cost.

In addition, this work serves as the foundation for further research. As the reflection measurements of these samples provide information of the bulk which in turn leads to a more comprehensive understanding of the structure and crystallinity of the polymer within the layer.

# 5 References


(1) Khan, S. A.; Khan, S. B.; Khan, L. U.; Farooq, A.; Akhtar, K.; Asiri, A. M. In *Handbook of Materials Characterization*, Sharma, S. K., Ed.; Springer International Publishing: Cham, 2018, pp 317–344.

(2) Zhang, X.; Qiu, J.; Zhao, J.; Li, X.; Liu, L. *Journal of Quantitative Spectroscopy and Radiative Transfer* **2020**, *252*, 107063.

(3) Jitian, S.; Bratu, I. *AIP Conference Proceedings* **2012**, *1425*, 26–29.

(4) Lee, J. *Vibrational Spectroscopy* **2017**, *90*, 104–111.

(5) Tazreiter, M.; Christian, P.; Schennach, R.; Grießer, T.; Maria Coclite, A. *Analytical Methods* **2017**, *9*, Publisher: Royal Society of Chemistry, 5266–5273.

(6) Mayerhöfer, T. G.; Pahlow, S.; Hübner, U.; Popp, J. *The Analyst* **2020**, *145*, 3385–3394.

(7) Ozaki, Y. *Analytical Sciences: The International Journal of the Japan Society for Analytical Chemistry* **2021**, *37*, 1193–1212.

(8) Griffiths, P. R.; Haseth, J. A. D., *Fourier Transform Infrared Spectrometry*, 2nd; John Wiley & Sons: 2007.

(9) Poliskie, M.; Clevenger, J. O. *Metal Finishing* **2008**, *106*, 44–47.

(10) Ismail, A. A.; Van De Voort, F. R.; Sedman, J. In *Techniques and Instrumentation in Analytical Chemistry*; Elsevier: 1997; Vol. 18, pp 93–139.

(11) Grosse, P. *Microchimica Acta* **1991**, *104*, 309–323.

(12) Gerwert, K.; Kötting, C. In Wiley *Encyclopedia of Life Sciences*, 1st ed.; Wiley: 2010.

(13) Thompson, J. M. In *Infrared Spectroscopy*, 1st ed.; Jenny Stanford Publishing: 2018, pp 89–112.

(14) Rogalski, A. *Infrared Physics & Technology* **2002**, *43*, 187–210.

(15) Gravrand, O.; Rothman, J.; Castelein, P.; Cervera, C.; Baier, N.; Lobre, C.; Borniol, E. D.; Zanatta, J. P.; Boulade, O.; Moreau, V.; Fieque, B.; Chorier, P. In *Infrared Technology and Applications XLII*, Infrared Technology and Applications XLII, SPIE: 2016; Vol. 9819, pp 495–507.

(16) Hasegawa, T.; Taniguchi, K.; Sato, Y. *Vibrational Spectroscopy* **2009**, *51*, 76–79.

(17) Hilfiker, J. N.; Tiwald, T. In *Spectroscopic Ellipsometry for Photovoltaics: Volume 1: Fundamental Principles and Solar Cell Characterization*, Fujiwara, H., Collins, R. W., Eds.; Springer International Publishing: Cham, 2018, pp 115–153.

(18) Bassan, P.; Byrne, H. J.; Lee, J.; Bonnier, F.; Clarke, C.; Dumas, P.; Gazi, E.; Brown, M. D.; Clarke, N. W.; Gardner, P. *Analyst* **2009**, *134*, Publisher: The Royal Society of Chemistry, 1171–1175.

(19) Buffeteau, T.; Desbat, B. *Applied Spectroscopy* **1989**, *43*, Publisher: SAGE Publications Ltd STM, 1027–1032.

(20) Mayerhöfer, T. G.; Pahlow, S.; Hübner, U.; Popp, J. *The Analyst* **2018**, *143*, 3164–3175.

(21) Jacob, W.; Keudell, A. v.; Schwarz-Selinger, T. *Brazilian Journal of Physics* **2000**, *30*, Publisher: Sociedade Brasileira de Física, 508–516.

(22) Oshina, I.; Spigulis, J. *Journal of Biomedical Optics* **2021**, *26*, Publisher: SPIE, 100901.

(23) Yen, Y. S.; Wong, J. S. *The Journal of Physical Chemistry* **1989**, *93*, 7208–7216.

(24) Teolis, B.; Loeffler, M.; Raut, U.; Famá, M.; Baragiola, R. *Icarus* **2007**, *190*, 274–279.

(25) Heavens, O. S. *Reports on Progress in Physics* **1960**, *23*, 1.

(26) Ivanova, E. A.; Müller, W. H.; Rickert, W.; Vilchevskaya, E. N. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* **2025**, *105*.

(27) Zhang, J. X.; Hashino, K. In 2019, pp 231–309.

(28) Kuzmenko, A. B. *Review of Scientific Instruments* **2005**, *76*.

(29) Wang, Q.; Yan, X.-L.; Chen, X.-C.; Shuai, P.; Wang, M.; Zhang, Y.-H. *Nuclear Science and Techniques* **2022**, *33*, 148.

(30) Zhang, F.; Tang, X.; Tong, A.; Wang, B.; Wang, J.; Lv, Y.; Tang, C.; Wang, J. *Spectroscopy Letters* **2020**, *53*, 222–233.

(31) Baek, S.-J.; Park, A.; Ahn, Y.-J.; Choo, J. *Analyst* **2015**, *140*, Publisher: Royal Society of Chemistry, 250–257.

(32) Erb, D. pybaselines: A Python library of algorithms for the baseline correction of experimental data, 2024.

(33) Likhachev, D. V. In *Modeling Aspects in Optical Metrology VI*, Modeling Aspects in Optical Metrology VI, SPIE: 2017; Vol. 10330, pp 348–355.

(34) Johs, B.; Hale, J. S. *physica status solidi (a)* **2008**, *205*, 715–719.

(35) Byrnes, S. J. Multilayer optical calculations, 2020.

(36) Luce, A.; Mahdavi, A.; Marquardt, F.; Wankerl, H. *Journal of the Optical Society of America A* **2022**, *39*, 1007.

(37) Garhoom, W. A. A.; Shadidi, Z. A. *East European Journal of Physics* **2021**, Number: 4, 164–171.

(38) Puhan, J.; Lipovšek, B.; Bűrmen, Á.; Fajfar, I. *IEEE Photonics Journal* **2017**, *9*, Conference Name: IEEE Photonics Journal, 1–12.

(39) Olmon, R. L.; Slovick, B.; Johnson, T. W.; Shelton, D.; Oh, S.-H.; Boreman, G. D.; Raschke, M. B. *Physical Review B* **2012**, *86*, Publisher: American Physical Society, 235147.

(40) Shkondin, E.; Takayama, O.; Panah, M. E. A.; Liu, P.; Larsen, P. V.; Mar, M. D.; Jensen, F.; Lavrinenko, A. V. *Optical Materials Express* **2017**, *7*, Publisher: Optica Publishing Group, 1606–1627.

(41) Dektak XTL Brochure [PDF] `https://www.bruker.com/en/meta/forms/bns-form-pages/brochures/dektak/dektak-xtl.html` (accessed 05/27/2025).

(42) Wirth, R. *Chemical Geology* **2009**, *261*, 217–229.

(43) Nicolet™ iN10 MX Infrared Imaging Microscope `https://www.thermofisher.com/order/catalog/product/IQLAADGAAGFAHDMAPE` (accessed 06/10/2025).

(44) HYPERION II FT-IR and QCL Microscope `https://www.bruker.com/en/products-and-solutions/infrared-and-raman/ft-ir-microscopes/hyperion-ii-ft-ir-and-qcl-microscope.html` (accessed 05/27/2025).

(45) Virtanen, P. et al. *Nature Methods* **2020**, *17*, Publisher: Nature Publishing Group, 261–272.

(46) Endres, S. C.; Sandrock, C.; Focke, W. W. *Journal of Global Optimization* **2018**, *72*, 181–217.

(47) Branch, M. A.; Coleman, T. F.; Li, Y. *SIAM Journal on Scientific Computing* **1999**, *21*, 1–23.

(48) Maeder, M.; Zuberbuehler, A. D. *Analytical Chemistry* **1990**, *62*, 2220–2224.

Additionally, artificial intelligence tools such as GPT4o and Llama 3.1 70b were implemented throughout this thesis to look up synonyms.

# List of Figures

# List of Tables

# A Appendix

The following code was implemented throughout this thesis to obtain the figures and values of the refractive index and polymer layer thickness.

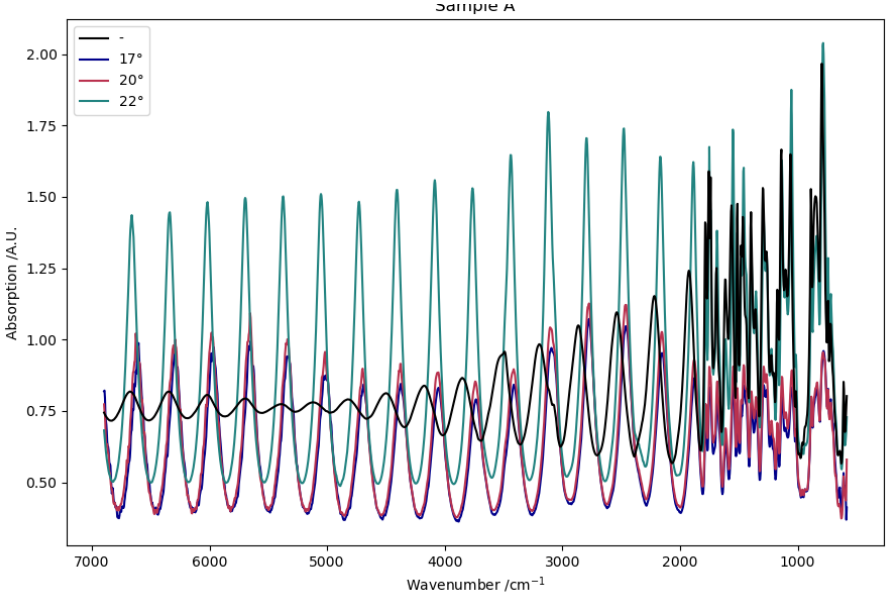
```
import numpy as np
import xarray as xr
import  matplotlib.pyplot as plt
import pathlib
get_ipython().run_line_magic('matplotlib', 'widget')
import glob
import yaml
from yaml import load, dump
from yaml import CLoader as Loader, CDumper as Dumper
from scipy.optimize import least_squares
from matplotlib import colormaps
import re
from scipy.optimize import leastsq
from scipy.special import xlogy
from pybaselines import Baseline, utils
import os
import pandas as pd

def k1_prime(n1, alpha):
    return n1 * np.cos(alpha)

def k2_prime(n2, alpha):
    return np.sqrt(n2**2 - np.sin(alpha)**2)

def k3_prime(n3, alpha):
    return np.sqrt(n3**2 - np.sin(alpha)**2)


def phi_nu(nu, k2_p, d, alpha):
    return 2 * np.pi * nu * k2_p * d

def R(n_polymer, n_substrate, d, alpha, nu, pol):
    """ calculates reflection spectrum relative to substrate
    n_polymer:   number, xr.DataArray,
                 refractive index of polymer layer
```

```
n_substrate: function f(nu) returns refractive index of
    substrate
                refractive index of substrate
d:              number, xr.DataArray,
                thickness of polymer layer in µm
alpha:          number, xr.DataArray,
                angle of incidence in radians from surface
                    normal
nu:             number, xr.DataArray,
                wavenumber in um-1
pol:            {"s", "p"}
                polarization


returns:
--------

reflection of polymer layer/substrate stack relative to
    substrate reflection
"""

n1 = 1
k1_p = k1_prime(n1, alpha)
k2_p = k2_prime(n_polymer, alpha)
k3_p = k3_prime(n_substrate, alpha)
phi = phi_nu(nu, k2_p, d, alpha)

if pol == 's':
    r12_s = (k1_p - k2_p) / (k1_p + k2_p)
    r23_s = (k2_p - k3_p) / (k2_p + k3_p)
    r_polymer = (r12_s + r23_s * np.exp(2 * 1j * phi)) /
        (1 + r12_s * r23_s * np.exp(2 * 1j * phi))
    r_substrate = (k1_p - k3_p) / (k1_p + k3_p)
    R_values = abs(r_polymer)**2 / abs(r_substrate)**2

elif pol == 'p':
    r12_p = (-k1_p * n_polymer**2 + k2_p * n1**2) / (k1_p
        * n_polymer**2 + k2_p * n1**2)
    r23_p = (-k2_p * n_substrate**2 + k3_p * n_polymer
        **2) / (k2_p * n_substrate**2 + k3_p * n_polymer
```

```
            **2)
        r_polymer = (r12_p + r23_p * np.exp(2 * 1j * phi)) /
            (1 + r12_p * r23_p * np.exp(2 * 1j * phi))
        r_substrate = (-k1_p * n_substrate**2 + k3_p * n1**2)
            / (k1_p * n_substrate**2 + k3_p * n1**2)
        R_values = abs(r_polymer)**2 / abs(r_substrate)**2

    elif pol ==  'both':
        r12_s = (k1_p - k2_p) / (k1_p + k2_p)
        r23_s = (k2_p - k3_p) / (k2_p + k3_p)
        r12_p = (-k1_p * n_polymer**2 + k2_p * n1**2) / (k1_p
            * n_polymer**2 + k2_p * n1**2)
        r23_p = (-k2_p * n_substrate**2 + k3_p * n_polymer
            **2) / (k2_p * n_substrate**2 + k3_p * n_polymer
            **2)

        r_polymer_s = (r12_s + r23_s * np.exp(2 * 1j * phi))
            / (1 + r12_s * r23_s * np.exp(2 * 1j * phi))
        r_substrate_s = (k1_p - k3_p) / (k1_p + k3_p)

        r_polymer_p = (r12_p + r23_p * np.exp(2 * 1j * phi))
            / (1 + r12_p * r23_p * np.exp(2 * 1j * phi))
        r_substrate_p = (-k1_p * n_substrate**2 + k3_p * n1
            **2) / (k1_p * n_substrate**2 + k3_p * n1**2)

        R_values = (abs(r_polymer_s)**2+abs(r_polymer_p)**2)
            / (abs(r_substrate_s)**2 + abs(r_substrate_p)**2)

    return R_values


def target_spectrum_ev(folder_path):
    """loading data into an xarray"""
    file_list = glob.glob(os.path.join(folder_path, "*.CSV"))

    data_arrays = []
    filenames = []

    for file in file_list:
```

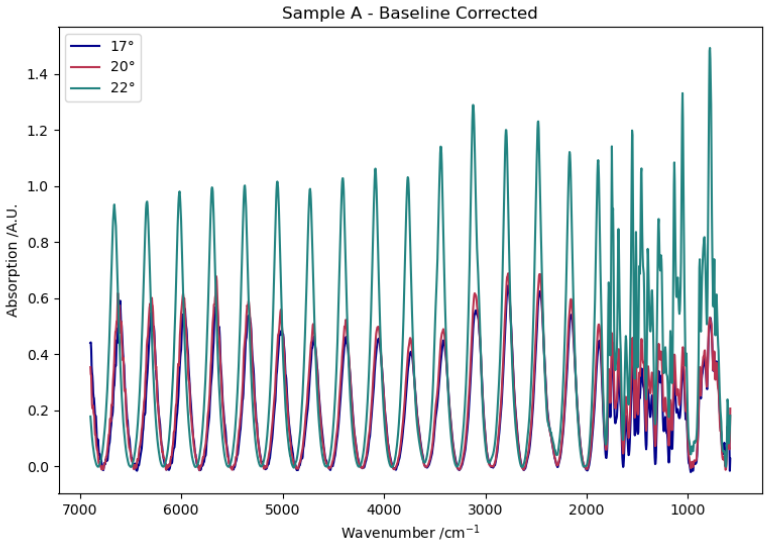
```
        df = pd.read_csv(file, header=None, names=["nu", "
            absorption"], delimiter=",")
        nu = df['nu'].values
        absorption = df['absorption'].values
        filename = os.path.basename(file)[:-4]
        da = xr.DataArray(absorption,
                          dims=["nu"],
                          coords={"nu": nu})

        data_arrays.append(da)
        filenames.append(filename)
    combined_da = xr.concat(data_arrays, dim="idx")
    combined_da.coords["filename"] = ("idx", filenames)
    return combined_da


ImidSA = target_spectrum_ev("file_path")

ImidSA = ImidSA.sortby("nu").dropna(dim = "nu").sel(nu=slice
    (580, 6900))
ImidSA = ImidSA.sortby("filename")
angle_SA = np.array([17, 20, 22, 17])
layer_SA = np.array([10, 10, 10, 10])
ImidSA = ImidSA.assign_coords(angle=('idx', angle_SA))
ImidSA = ImidSA.assign_coords(layer=('idx', layer_SA))



fig, ax = plt.subplots(figsize=(9,6))
colorlist = ["darkblue", "#BA324F", "#218380", "black"]
angle = [str(17)+"°", str(20)+"°" , str(22)+"°", "-"]
for i, spec in enumerate(ImidSA):
    ax.plot(spec.nu, spec, label=angle[i], color=colorlist[i
        ])
plt.xlabel('Wavenumber /cm$^{-1}$')
plt.ylabel('Absorption /A.U.')
handles, labels = ax.get_legend_handles_labels()
labels, handles = zip(*sorted(zip(labels, handles), key=
    lambda t: t[0]))
ax.legend(handles, labels, loc='upper left')
```

```
plt.tight_layout(rect=[0, 0, 1, 1])
plt.title("Sample A")
ax.invert_xaxis()
```




```
def correction_arPLS(target_spectra):
    corrected_list = []
    for i in range(target_spectra.sizes['idx']):
        spec = target_spectra.isel(idx=i)
        baseline_fitter = Baseline(x_data=spec.nu)
        bkg_1, _ = baseline_fitter.arpls(spec, lam=1E5)
        corrected_spectra = spec - bkg_1
        max_values = corrected_spectra.max(dim="nu")
        min_values = corrected_spectra.min(dim="nu")
        normalize_spectra = (corrected_spectra - min_values)
           / ( max_values - min_values)
        normalize_spectra = normalize_spectra.assign_coords(
           filename=spec.coords['filename'])
        corrected_list.append(normalize_spectra)

    combined_corrected_spectra = xr.concat(corrected_list,
       dim='idx')
    return combined_corrected_spectra

ImidSA_arPLS = correction_arPLS(ImidSA)
```

```
fig, ax = plt.subplots(figsize=(9,6))
angle = [str(17)+"°", str(20)+"°" , str(22)+"°", "-"]
colorlist = ["darkblue", "#BA324F", "#218380"]
for i, spec in enumerate(ImidSA_arPLS[:3]):
    ax.plot(spec.nu, spec, label=angle[i], color=colorlist[i
       ])
plt.xlabel('Wavenumber /cm$^{-1}$')
plt.ylabel('Absorption /A.U.')
handles, labels = ax.get_legend_handles_labels()
labels, handles = zip(*sorted(zip(labels, handles), key=
   lambda t: t[0]))
ax.legend(handles, labels, loc='upper left')
plt.title("Sample A - Baseline Corrected")
ax.invert_xaxis()
```




```
def load_yaml_file(path):
    """loads a yaml file from the specified path.
    path : str
        file path of the yaml file

    returns:
    --------
```

```
    dict
        returns it as a dictionary
    """
    with open(path, 'r') as file:
        return yaml.load(file, Loader=Loader)


yaml_data = {}
file_patterns = glob.glob("/home/jovyan/shared/CJ/10_08 new
   samples/literature values/*.yml.txt")
file_dict = {os.path.basename(file_path).split("_")[0]:
   file_path for file_path in file_patterns}

for name, f in file_dict.items():
    data = load_yaml_file(f)
    yaml_data[name]=(data)

def yaml_to_da(data):
    """converts refractive index data (n + ik) from the yaml
       file into an xarray DataArray
    data : dict
         dictionary containing refractive index data from the
             yaml file

    returns:
    -------
    xarray.DataArray
        A DataArray with the complex refractive index (n + ik
           ) as values,
        wavenumber as coordinates, and a single dimension `
           wn_values`.
    """
    data_string = data['DATA'][0]['data']
    wn_values = []
    n_values = []
    k_values = []

    for value in data_string.split("\n"):
        parts = value.split()
        if len(parts) == 3:
```

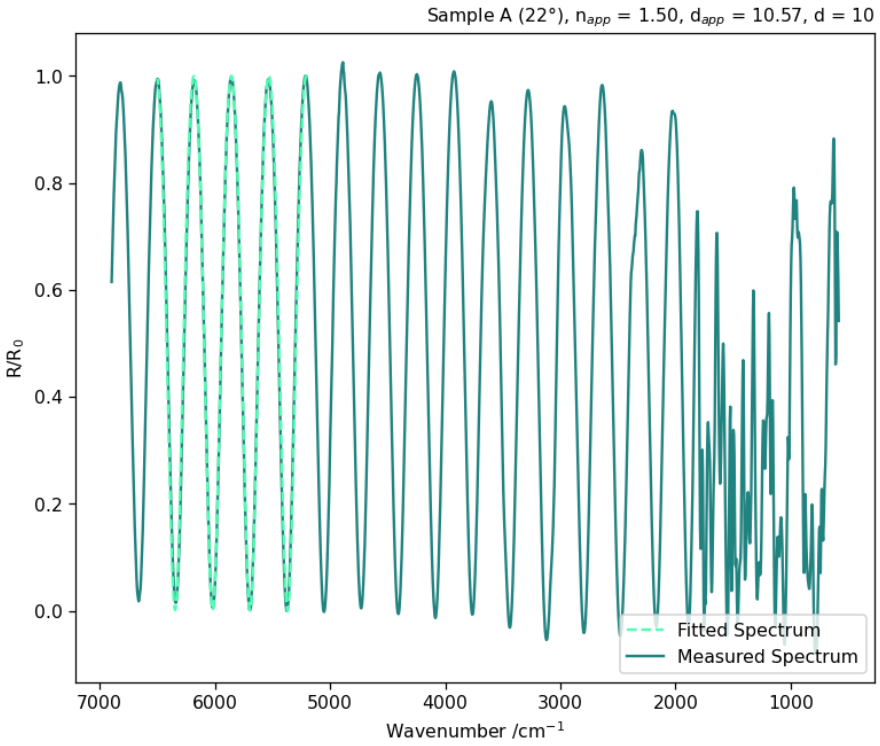
```
            wn_values.append(float(parts[0]))
            n_values.append(float(parts[1]))
            k_values.append(float(parts[2]))

    wn_values = np.array(wn_values)
    n_values = np.array(n_values)
    k_values = np.array(k_values)

    da = xr.DataArray(n_values+1j* k_values,
        dims=["wn_values"],
        coords={"wn_values": wn_values})
    return da

refind = {name:yaml_to_da(ymd) for name, ymd in yaml_data.
   items()}

from scipy.interpolate import interp1d
def get_interpfun(da):
    interpfun = interp1d(da.wn_values.to_numpy(), da.to_numpy
       (), kind="quadratic", bounds_error=False, fill_value="
       extrapolate")
    return lambda wn: xr.apply_ufunc(interpfun, wn)

refinds = {name:get_interpfun(da) for name, da in refind.
   items()}



thickness_SA = np.array([10, 10, 10, 10])
ImidSA_arPLS = ImidSA_arPLS.assign_coords(layer=('idx',
   thickness_SA))


class Initializer():
    """
    Initializes and performs fitting operations for comparing
        simulated
    and target spectra.

    This class manages the parameters, bounds, and the
```

```
    fitting process for determining
the optimal fit between a target spectrum and a
    calculated model.
"""
def __init__(self, substrate_refinds, target_spectrum,
    pol, alpha, d, n_polymer, fitted_params=["n_polymer",
    "d"], bounds=None):
     self.target_spectrum = target_spectrum
     self.n_substrate = substrate_refinds(1E4/
        target_spectrum.nu)
     self.nu = target_spectrum.nu
     self.pol = pol
     self.alpha = alpha
     self.d = d+0.2
     self.n_polymer = n_polymer
     self.fitted_params = fitted_params
     if bounds is None:
         bounds = {}
     self.bounds = bounds

     self.all_params = {"n_polymer":n_polymer, "d":d, "
        alpha":alpha,  "pol":pol, "n_substrate":self.
        n_substrate, "nu":self.nu/1E4}

     self._model_fun_args = ["n_polymer", "d", "alpha",  "
        pol", "n_substrate", "nu"]


def _init_fit(self):
     params = []
     param_names = []
     fixed = {}
     bounds_arrays = [[],[]]
     for p_name in self._model_fun_args:
         if p_name in self.fitted_params:
             params.append(self.all_params[p_name])
             param_names.append(p_name)
             bounds = self.bounds.get(p_name,[-np.inf, np.
                inf])
             bounds_arrays[0].append(bounds[0])
```

```
                bounds_arrays[1].append(bounds[1])
            else:
                fixed[p_name] = self.all_params[p_name]
        self.params_initial = params
        self.params_names = param_names
        self.fixed = fixed

        def update_args(params):
            d = self.fixed.copy()
            d.update({pn: arg for pn, arg in zip(self.
               params_names, params)})
            return d

        self._model_fun_partial = lambda params: self.
           model_fun(**update_args(params))
        self.bounds_arrays = bounds_arrays

    @staticmethod
    def model_fun( n_polymer, d, alpha,  pol, n_substrate, nu
       ):
        """ calculates the difference betwenn measured and
           calculated spectrum
        params:       number, xr.DataArray,
                      variables that are being optimized (
                         n_polymer and alpha)
        d:            number, xr.DataArray,
                      thickness of polymer layer in µm
        pol:          {"s", "p", "both"}
                      polarization
        n_substrate: function f(nu) returns refractive index
           of substrate
                      refractive index of substrate

        returns:
        --------

        difference between target_spectrum and calculated
           spectrum
        """
        R_res= R(n_polymer, n_substrate, d, alpha, nu, pol)
```

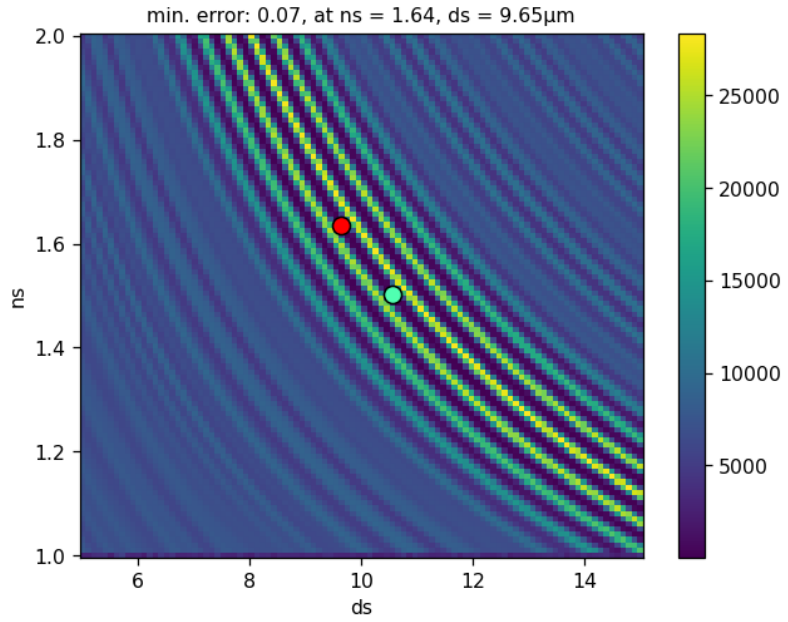
```
        R_res =( R_res - R_res.min())/(R_res.max() - R_res.
            min())
        return R_res


    def fit_fun(self, params):
        diff = self._model_fun_partial(params) - self.
            target_spectrum
        return diff.values.flatten()

    def run_fit(self, *args, **kwargs):
        opt = least_squares(self.fit_fun, self.params_initial
            ,  bounds=self.bounds_arrays, *args, **kwargs)
        self.opt = opt

    @property
    def fit_curve(self):
        return self._model_fun_partial(self.opt.x)

    @property
    def fit_results(self):
        return {pn: val for pn,val in zip(self.params_names,
            self.opt["x"])}



from scipy.optimize import shgo
class InitializerGlobal(Initializer):
    def fit_fun(self, params):
        diff = self._model_fun_partial(params) - self.
            target_spectrum
        return np.sum(diff.values.flatten()**2)
    def run_fit(self, *args, **kwargs):
        opt = shgo(self.fit_fun,  bounds=list(zip(*self.
            bounds_arrays)), *args, **kwargs)
        self.opt = opt
```

```
s = 10**-ImidSA_arPLS[1].sel(nu=slice(580, 6900))
s = s.pipe(lambda arr:(arr - arr.sel(nu=slice(5200,6500)).min
   ())/( arr.sel(nu=slice(5200,6500)).max() - arr.sel(nu=
   slice(5200,6500)).min()))
init = InitializerGlobal(refinds["Si"], s.sel(nu=slice
   (5200,6500)), pol="both", d=10, alpha=s.angle/180*np.pi,
   n_polymer = 1.5, bounds={"n_polymer":[1, 2], "d":[5, 15]},
    fitted_params=["n_polymer", "d"])

init._init_fit()
init.run_fit(n=200)
print(init.fit_results["d"])

fig, ax = plt.subplots(figsize=(7,6))
init.fit_curve.plot.line(x="nu", label='Fitted Spectrum',
   linestyle= "--", color= "#50FFB1", zorder = 3)
s.plot.line(x="nu", color = "darkblue", label='Measured
   Spectrum',)
init.fit_results
plt.title("")

filename = s.filename.values
text_str = f'Sample A (20°), n$_{{app}}$ = {init.fit_results
   ["n_polymer"]:.2f}, d$_{{app}}$ = {init.fit_results["d
   "]:.2f}, d = {s.layer.values}'
plt.title(text_str, loc="right", fontsize=10)

plt.xlabel('Wavenumber /cm$^{-1}$')
plt.ylabel('R/R$_0$')

ax.invert_xaxis()

handles, labels = ax.get_legend_handles_labels()
labels, handles = zip(*sorted(zip(labels, handles), key=
   lambda t: t[0]))
ax.legend(handles, labels, loc='lower right')
plt.tight_layout(rect=[0, 0, 1, 1])
plt.show()
```




```
ds = np.linspace(5, 15, 100)
ns = np.linspace(1, 2, 100)

ds = xr.DataArray(ds, coords={"ds":ds})
ns = xr.DataArray(ns, coords={"ns":ns})


errors = xr.apply_ufunc(lambda n,d: np.sum(init.fit_fun([n,d
    ])**2), ns, ds, vectorize=True)

error_min_index = errors.argmin(dim=["ns", "ds"])
min_ns = errors["ns"][error_min_index["ns"]].item()
min_ds = errors["ds"][error_min_index["ds"]].item()
min_error = errors.isel(ns=error_min_index["ns"], ds=
    error_min_index["ds"]).item()


result_d = init.fit_results["d"]
result_n = init.fit_results["n_polymer"]


plt.figure()
```

```
errors.plot.pcolormesh(x="ds", y="ns")
plt.scatter(min_ds, min_ns, color="red", s=80, label=f"Min
    Error: n={min_ns:.2f}, d={min_ds:.2f}$\mu$m", zorder=3,
    edgecolors="black")
plt.scatter(result_d, result_n, color="#50FFB1", s=80, label=
    f"Fit Result: n$_{{app}}$ = {result_n:.2f}, d$_{{app}}$ =
    {result_d:.2f}$\mu$m", zorder=3, edgecolors="black")
ax.legend()
plt.title(f"min. error: {min_error:.2f}, at ns = {min_ns:.2f
    }, ds = {min_ds:.2f}$\mu$m", fontsize = 10)
```




```
def init_n_d(s):
    s = 10**-s
    s = s.pipe(lambda arr:(arr - arr.sel(nu=slice(5800,6500))
       .min())/( arr.sel(nu=slice(5800,6500)).max() - arr.sel
       (nu=slice(5800,6500)).min()))
    if s.angle == 17:
        init = InitializerGlobal( refinds["Si"], s.sel(nu=
           slice(5800,6500)), pol="both", d=10, alpha=17/180*
           np.pi, n_polymer = 1.5, bounds={"n_polymer":[1,
           2], "d":[5, 15]}, fitted_params=["n_polymer", "d"
           ])
    if s.angle == 20:
```

```
            init = InitializerGlobal( refinds["Si"], s.sel(nu=
                slice(5800,6500)), pol="both", d=10, alpha=20/180*
                np.pi, n_polymer = 1.5, bounds={"n_polymer":[1,
                2], "d":[5, 15]}, fitted_params=["n_polymer", "d"
                ])
        if s.angle == 22:
            init = InitializerGlobal( refinds["Si"], s.sel(nu=
                slice(5800,6500)), pol="both", d=10, alpha=22/180*
                np.pi, n_polymer = 1.5, bounds={"n_polymer":[1,
                2], "d":[5, 15]}, fitted_params=["n_polymer", "d"
                ])

        init._init_fit()
        init.run_fit(n=200)
        return init.fit_results


def renamer(d):
        ren_dict = {"n_polymer":"app_n", "d":"app_d"}
        for k in list(d.keys()):
            if k in ren_dict:
                d[ren_dict[k]] = d[k]
                del d[k]
        return d

init_res_SA = [init_n_d(s) for s in ImidSA_arPLS.transpose("
    idx",...)]
ImidSA_arPLS= ImidSA_arPLS.assign_coords({n:("idx", [ renamer
    (l)[n]for l in init_res_SA]) for n in ["app_n", "app_d"]})



def fit_fun(params, alpha,  pol, n_substrate, target_spectrum
    , return_difference=True):
        """ calculates the difference betwenn measured and
            calculated spectrum using the reflection spectrum
        params:       number, xr.DataArray,
                      variables that are being optimized (
                          n_polymer and alpha)
        d:            number, xr.DataArray,
```

```
                thickness of polymer layer in µm
    pol:        {"s", "p", "both"}
                polarization
    n_substrate: function f(nu) returns refractive index of
       substrate
                refractive index of substrate
    target_spectrum:  xr.DataArray
                measured spectrum on Hyperion
    returns:
    --------

    difference between target_spectrum and calculated
       spectrum
    """
    n_polymer = params[0]
    d = params[1]
    nu = target_spectrum.nu/1E4
    R_res= R(n_polymer, n_substrate, d, alpha, nu, pol)

    if return_difference:
        diff = target_spectrum - (R_res)
        difference = diff.values
        return difference.flatten()
    return (R_res)


class Kappa():
    """
    A class that iteratively improves the absorption index,
       refractive index and polymer layer thickness.
    """
    def __init__(self, target_spectrum, alpha, n_subs,
       fit_fun, nu=None):
        self.target_spectrum = target_spectrum
        if nu is None:
            nu = target_spectrum.nu
        self.nu = nu
        self.k_to_n_arr = self.get_k_to_n_arr(self.nu)
        self.fit_fun = fit_fun
        self.alpha = alpha
```

```
        self.kappa_new = None
        self.n_new = None
        self.A_meas = 10**-target_spectrum #reflection
            spectrum

        self.app_n = target_spectrum.app_n.values
        self.app_d = target_spectrum.app_d.values
        self.parameters = {"alpha": self.alpha, "pol": 'both'
            , "n_substrate": n_subs(1E4/self.nu), "
            target_spectrum": self.A_meas}


    @staticmethod
    def get_k_to_n_arr(nu):
        nu_new = nu.rename(nu="nu_out")
        nu_m = nu.shift(nu=+1)
        nu_p = nu.shift(nu=-1)
        nu_i = nu
        def g(x,y):
            return xlogy(x-y, np.abs(x-y)) + xlogy(x+y, np.
                abs(x+y))
        term1 = g(nu_new, nu_m)/(nu_i - nu_m)
        term2 = (nu_p - nu_m)*g(nu_new, nu_i)/((nu_i-nu_m)*(
            nu_p-nu_i))
        term3 = g(nu_new, nu_p)/(nu_p - nu_i)
        return 1/np.pi*(term1-term2+term3).dropna("nu")

    def n_from_k(self, k, n_inf):
        n = self.k_to_n_arr.dot(k) + n_inf
        n = n.rename(nu_out="nu") + 1j*k
        return n


    @staticmethod
    def kappa_app(A_corr, nu, d, n_polymer, alpha):
        nu = nu/1E4
        a = A_corr / (np.log10(np.e) * 4 * np.pi * nu * 2 * d
            / (np.sqrt(1 - 1 / n_polymer**2 * np.sin(alpha)
            **2)))
```

```
        return a

    def compute_initials(self):
        A_calc_initial = self.fit_fun([self.app_n, self.
            app_d], **self.parameters, return_difference=
            False)
        A_corr = -np.log10(self.A_meas/ A_calc_initial)

        self.kappa_new = np.abs(self.kappa_app(A_corr, self.
            nu, self.app_d, self.app_n, self.alpha))
        self.n_new = self.n_from_k(self.kappa_new, self.
            app_n)
        return self.n_new

    def optimize_ratio(self, return_scaling=False):
        A_calc = self.fit_fun([self.n_new, self.app_d], **
            self.parameters, return_difference=False)
        self.kappa_prev = self.kappa_new.copy()
        scaling = np.log10((self.A_meas / A_calc))
        self.kappa_new = self.kappa_prev * scaling
        self.n_new = self.n_from_k(self.kappa_new, self.app_n
            )
        if return_scaling:
            return self.n_new, scaling
        return self.n_new


    def optimize_delta(self, return_delta=False):
        A_calc = self.fit_fun([self.n_new, self.app_d], **
            self.parameters, return_difference=False)
        self.kappa_prev = self.kappa_new.copy()

        delta = ( -np.log10(A_calc) +np.log10(self.A_meas))/\
                (np.log10(np.e)*4*np.pi*self.nu/1E4*self.
                    app_d/np.sqrt(1-np.sin(self.alpha)**2/self
                    .app_n**2))
        self.kappa_new = self.kappa_prev + delta
        self.n_new = self.n_from_k(self.kappa_new, self.app_n
            )
        if return_delta:
```

```
            return self.n_new, delta
        return self.n_new


    def fit_function(self, kappa, fit_fun, coeff_weight=None,
        diff2_weight=None):
        if kappa.shape[0] != len(self.nu):
            raise ValueError(f"Expected kappa of length {len(
                self.nu)}, but got {kappa.shape[0]}")

        kappa_new = xr.DataArray(kappa, dims=["nu"], coords={
            "nu": self.nu})
        n_new = self.n_from_k(kappa_new, self.app_n)
        A_calc = fit_fun([n_new, self.app_d], **self.
            parameters, return_difference=False)
        error = ((self.A_meas) - (A_calc)).values
        if coeff_weight is not None:
            error = np.append(error, np.sqrt(np.abs(kappa_new
                .values)) * np.sqrt(coeff_weight))
        if diff2_weight is not None:
            error = np.append(error, np.sqrt(np.abs(kappa_new
                .diff(dim="nu", n=2).values)) * np.sqrt(
                diff2_weight))
        return error

    def fit_function_nd(self, kappa_n, fit_fun,coeff_weight=
        None,  diff2_weight=None):
        """fit function using kappa as first part and n_infty
            as last element"""

        n_infty = kappa_n[-2]
        d = kappa_n[-1]
        kappa_new = xr.DataArray(kappa_n[:-2], dims=["nu"],
            coords={"nu": self.nu})
        n_new = self.n_from_k(kappa_new, n_infty)
        A_calc = fit_fun([n_new, d], **self.parameters,
            return_difference=False)
        error = ((self.A_meas) - (A_calc)).values
        if coeff_weight is not None:
            error = np.append(error, np.sqrt(np.abs(kappa_new
```

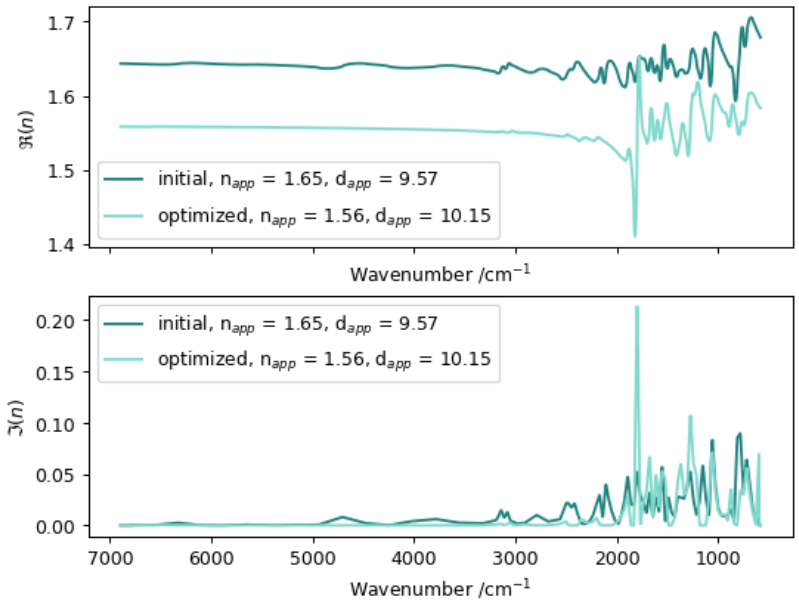
```
            .values)) * np.sqrt(coeff_weight))
        if diff2_weight is not None:
            error = np.append(error, np.sqrt(np.abs(kappa_new
                .diff(dim="nu", n=2).values)) * np.sqrt(
                diff2_weight))
        return error

    def optimize_least_squares(self,coeff_weight=None,
        diff2_weight=None,  *args, **kwargs):
        res = least_squares(self.fit_function, np.abs(self.
            kappa_new.values), verbose=2,bounds=(0, np.inf),
                            args=(self.fit_fun,coeff_weight,
                                diff2_weight), *args, **kwargs
                                )
        self.kappa_new = xr.DataArray(res["x"], dims=["nu"],
            coords={"nu": self.nu})
        self.n_new = self.n_from_k(self.kappa_new, self.app_n
            )
        return self.n_new

    def optimize_least_squares_nd(self,coeff_weight=None,
        diff2_weight=None, *args, **kwargs):
        res = least_squares(self.fit_function_nd, np.
            concatenate([np.abs(self.kappa_new.values), [self.
            app_n, self.app_d]]),
                            verbose=2,bounds=(0, np.inf),
                                args=(self.fit_fun,
                                coeff_weight, diff2_weight), *
                                args, **kwargs)
        self.app_n = res["x"][-2]
        self.app_d = res["x"][-1]
        self.kappa_new = xr.DataArray(res["x"][:-2], dims=["
            nu"], coords={"nu": self.nu})
        self.n_new = self.n_from_k(self.kappa_new, self.app_n
            )
        return self.n_new

    def error_fun_ndonly(self, nd, n_bare):
        A_calc = -np.log10(self.fit_fun([n_bare+nd[0], nd
            [1]], **self.parameters, return_difference=False))
```

```
        return (-np.log10(self.A_meas)  -A_calc)

    def optimize_least_squares_nd_only(self, *args, **kwargs)
        :
        n_bare = self.n_from_k(self.kappa_new, 0)

        res = least_squares(self.error_fun_ndonly, [self.
            app_n, self.app_d], verbose=2,bounds=((1,5), (2,
            15)),args=(n_bare,) , *args, **kwargs)
        self.app_n = res["x"][0]
        self.app_d = res["x"][1]
        self.n_new = self.n_from_k(self.kappa_new, self.app_n
            )
        return self.n_new



class Kappa_Spline(Kappa):
    """
    Fits Kappa with a spline, allows for adaption of
        resolution of the wavenumbers
    """
    def __init__(self, target_spectrum, alpha, n_subs,
        fit_fun):
        nu_orig = target_spectrum.nu
        nu = xr.concat([nu_orig.sel(nu=slice(None, 2500)).
            coarsen(nu=8, boundary="pad").mean(),
                        nu_orig.sel(nu=slice(2501, 3200)).
                            coarsen(nu=100, boundary="pad").
                            mean(),
                        nu_orig.sel(nu=slice(3201, 4500)).
                            coarsen(nu=8, boundary="pad").mean
                            (),
                        nu_orig.sel(nu=slice(4501, None)).
                            coarsen(nu=100, boundary="pad").
                            mean()], dim="nu")
        super().__init__(target_spectrum, alpha, n_subs,
            fit_fun, nu = nu)
        self.nu_orig = nu_orig
```

```
        self.k_to_n_arr = self.get_k_to_n_arr(nu, self.
            nu_orig)
        self.k_to_k_full_arr = self.get_k_to_k_full_arr(nu,
            self.nu_orig)
        self.parameters["n_substrate"] = n_subs(1E4/self.
            nu_orig)

    @staticmethod
    def get_k_to_n_arr(nu,  nu_new=None):
        if nu_new is None:
            nu_new = nu
        nu_new = nu_new.rename(nu="nu_out")
        nu_m = nu.shift(nu=+1)
        nu_p = nu.shift(nu=-1)
        nu_i = nu
        def g(x,y):
            return xlogy(x-y, np.abs(x-y)) + xlogy(x+y, np.
                abs(x+y))
        term1 = g(nu_new, nu_m)/(nu_i - nu_m)
        term2 = (nu_p - nu_m)*g(nu_new, nu_i)/((nu_i-nu_m)*(
            nu_p-nu_i))
        term3 = g(nu_new, nu_p)/(nu_p - nu_i)
        return -1/np.pi*(term1-term2+term3).dropna("nu")

    @staticmethod
    def get_k_to_k_full_arr(nu,  nu_new):
        nu_new = nu_new.rename(nu="nu_out")
        nu_m = nu.shift(nu=+1)
        nu_p = nu.shift(nu=-1)
        nu_i = nu
        lower =  (nu_new - nu_m)/(nu_i - nu_m)
        upper = (nu_p - nu_new)/(nu_p - nu_i)
        out = xr.where(np.logical_and((nu_m<nu_new),(nu_new<=
            nu_i)), lower, 0) + xr.where(np.logical_and((nu_i<
            nu_new),(nu_new<=nu_p)), upper, 0)
        return out.dropna("nu")

    def n_from_k(self, k, n_inf):
        n = self.k_to_n_arr.dot(k) + n_inf + 1j*self.
            k_to_k_full_arr.dot(k)
```

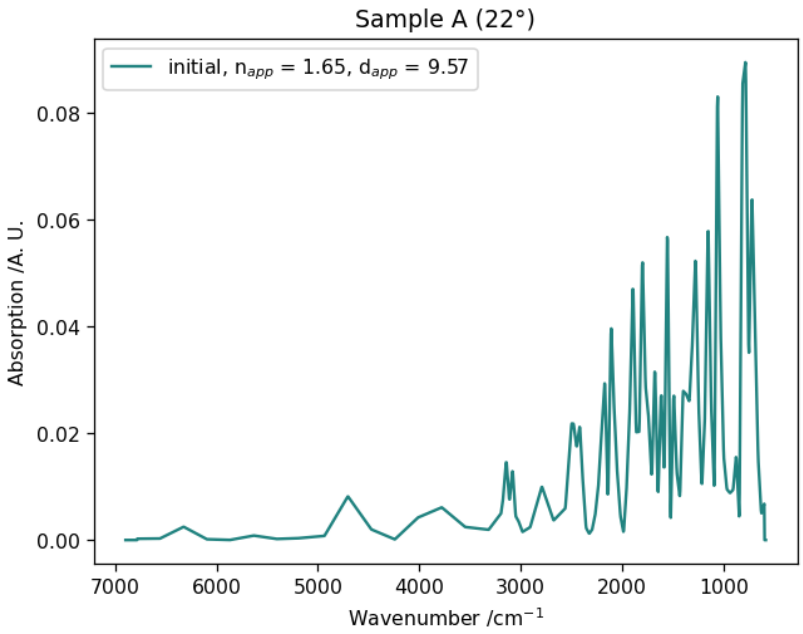
```
        n = n.rename(nu_out="nu")
        return n

    def compute_initials(self):
        A_calc_initial = self.fit_fun([self.app_n, self.app_d
            ], **self.parameters, return_difference=False)
        A_corr = -np.log10(self.A_meas / A_calc_initial)
        self.kappa_new = np.abs(self.kappa_app(A_corr, self.
            nu_orig, self.app_d, self.app_n, self.alpha)).
            interp(nu=self.nu)
        self.n_new = self.n_from_k(self.kappa_new, self.app_n
            )
        return self.n_new



spcidx = 3
colorlist = ["#218380", "#BA324F", "darkblue", "black"]
cur_spec = ImidSA_arPLS.isel(idx=spcidx).sel(nu=slice
    (580,6900))


def fit_spectrum_coeff(s, title=None):
    s =  s.assign_coords({n:("idx", [ renamer(l)[n]for l in [
        init_n_d(s)]]) for n in ["app_n", "app_d"]}).isel(idx
        =0)
    fitting_instance = Kappa_Spline(target_spectrum=s, alpha=
        s.angle/180*np.pi, n_subs=refinds["Si"], fit_fun=
        fit_fun)
    fig, ax = plt.subplots()

    newval = fitting_instance.compute_initials()
    newval.imag.plot.line(x="nu", label = f'initial, n$_{{app
        }}$ = {fitting_instance.app_n:.2f}, d$_{{app}}$ = {
        fitting_instance.app_d:.2f}', color="black")

    plt.xlabel('Wavenumber /cm$^{-1}$')
    plt.ylabel('Absorption /A. U.')
    ax.legend()
    ax.invert_xaxis()
```

```
    plt.title("Sample A")
    plt.show()
    return fitting_instance

fitters_I78 = {}
indices = [3]
for idx in indices:
    s = ImidSA_arPLS.isel(idx=[idx])
    fitters_I78[idx] = fit_spectrum_coeff(s)
```




```
def fit_spectrum(s, title=None):
    fig, axs = plt.subplots(nrows=2, sharex=True)
    axs[0].set_title(title)
    axs[0].set_ylabel(r"$\Re(n)$")
    axs[0].set_xlabel("Wavenumber /cm$^{-1}$")
    axs[1].set_ylabel(r"$\Im(n)$")
    axs[1].set_xlabel("Wavenumber /cm$^{-1}$")
    def plot_step(nk, label, color):
        axs[0].plot(nk.nu, nk.real, label=label, color=color)
        axs[1].plot(nk.nu, nk.imag, label=label, color=color)
        axs[1].legend(loc = "upper left")
        axs[0].legend(loc = "center left")
        fig.canvas.draw()
```

```
    s =  s.assign_coords({n:("idx", [ renamer(l)[n]for l in [
        init_n_d(s)]]) for n in ["app_n", "app_d"]}).isel(idx
        =0)
    fitting_instance = Kappa_Spline(target_spectrum=s, alpha=
        s.angle/180*np.pi, n_subs=refinds["Si"], fit_fun=
        fit_fun)
    stepval = fitting_instance.compute_initials()
    plot_step(stepval,f'initial, n$_{{app}}$ = {
        fitting_instance.app_n:.2f}, d$_{{app}}$ = {
        fitting_instance.app_d:.2f}', color="black")
    fitting_instance.optimize_least_squares(coeff_weight=1E
        -5, max_nfev=30, diff2_weight=1E-20,loss="soft_l1")
    plot_step(fitting_instance.optimize_least_squares_nd_only
        (max_nfev=15), f'optimized, n$_{{app}}$ = {
        fitting_instance.app_n:.2f}, d$_{{app}}$ = {
        fitting_instance.app_d:.2f}', color="#80D9CC")
    axs[0].invert_xaxis()
    return fitting_instance


fitters_sim = {}
indices = [3]
for idx in indices:
    s = ImidSA_arPLS.isel(idx=[idx])
    fitters_sim[idx] = fit_spectrum(s)
```




```
def fit_spectrum_coeff(s, title=None):
    s =  s.assign_coords({n:("idx", [ renamer(l)[n]for l in [
        init_n_d(s)]]) for n in ["app_n", "app_d"]}).isel(idx
        =0)
    fitting_instance = Kappa_Spline(target_spectrum=s, alpha=
        s.angle/180*np.pi, n_subs=refinds["Si"], fit_fun=
        fit_fun)

    fig, ax = plt.subplots()
    newval = fitting_instance.compute_initials()
    fitting_instance.optimize_least_squares(coeff_weight=1E
        -5, max_nfev=30, diff2_weight=1E-20, gtol=.5e-8,loss="
        soft_l1")
    fitting_instance.optimize_least_squares_nd_only(max_nfev
        =15).imag.plot.line(x="nu", label = f'optimized, n$_{{
        app}}$ = {fitting_instance.app_n:.2f}, d$_{{app}}$ = {
        fitting_instance.app_d:.2f}', color="#80D9CC")

    plt.xlabel('Wavenumber /cm$^{-1}$')
    plt.ylabel('Absorption /A. U.')
    ax.legend()
    ax.invert_xaxis()
    plt.title("Sample A")
    plt.show()
```

```
    return fitting_instance

fitters_I78 = {}
indices = [3]
for idx in indices:
    s = ImidSA_arPLS.isel(idx=[idx])
    fitters_I78[idx] = fit_spectrum_coeff(s)
```