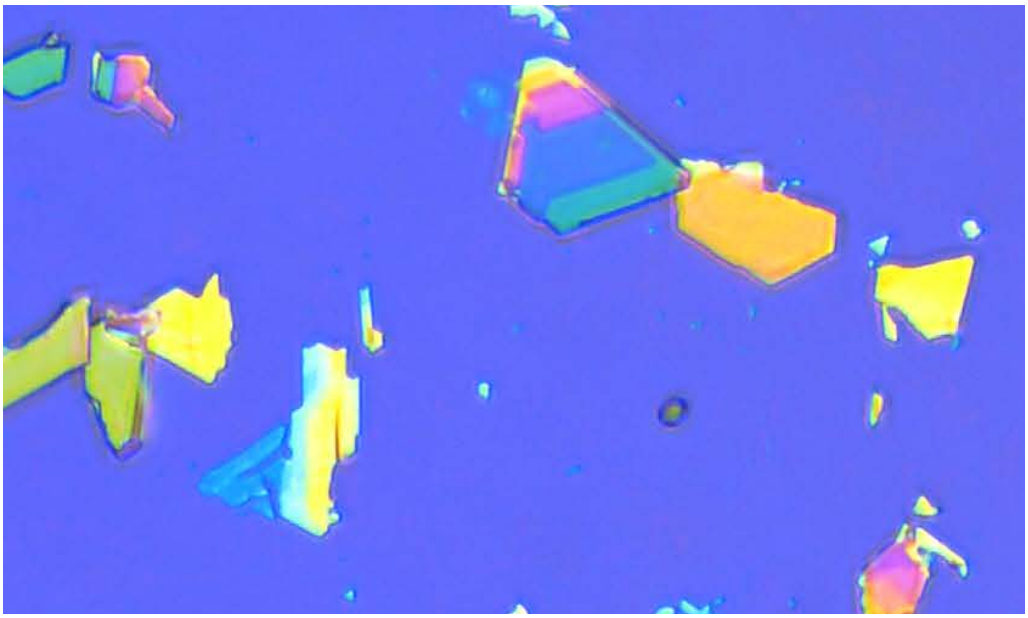
Thickness Identification of 2D Materials by Optical Imaging

Exploring the Relation between Colour and Thickness of $FePS_3$ and $NbSe_2$



Elias Baumann

Supervisors:
Yee Ling Willems-Ong
Niklas Kercher (ETH)
Realgymnasium Rämibühl
December 2023

Abstract

In recent decades, 2D materials have been researched extensively for their exceptional properties and diverse applications. Their thickness strongly influences their characteristics and therefore it must be determined prior to any experiment. This paper examines the approach of thickness identification by optical imaging on behalf of a theoretical model and experimental data, and tests its reliability for application in research, for both the insulating material $FePS_3$ and the conductive metallic $NbSe_2$.

The results confirm that the theoretical model accurately describes the contrast and colour of an insulator on a two-layered (SiO_2/Si) substrate. The correlation indicates, that the model's underlying assumptions were well chosen - even more so, since there is a significant divergence between the experimental results for the conductive material and the theoretical model, for which insulation is assumed. Further, they suggest that the thicknesses of an insulating flake up to 200 nm correspond to distinct RGB values.

This fact is used to develop an application, which is capable to yield, from a list of RGB values, the thicknesses, to which they are likely to correspond. The application is evaluated and its reliability is demonstrated. This report therefore confirms the high practicality of thickness identification by optical imaging, providing a fast, non-invasive, large-scale and cheap method to determine the thickness of 2D materials.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Mrs. Yee Ling Willems-Ong for guiding and supporting me with useful tips and giving very valuable feedback. Additionally, I thank the Department of Materials at ETH Zürich for granting me access to their instruments. Finally, I want to give a special thanks to Mr. Niklas Kercher for introducing me to 2D materials, providing me with the opportunity to conduct research in their lab and answering a lot of my questions. I am grateful for his patient explanations and his way of giving tips, which motivated me to push even further. I hope that my results and application can contribute to his workflow for the production of 2D materials for his doctorate in magnetism and interface physics.

Contents

List of Figures

Glossary of Symbols

$\vec{B}$ - Magnetic field

$\vec{B}_0$ - Magnetic field amplitude

c - Speed of light

C - Contrast

d - Thickness of a layer

e - Euler's number

E - Energy

$\vec{E}$ - Electric field

$\vec{E}_0$ - Electric field amplitude

ε - Electrical permittivity

ε_0 - Electrical permittivity in vacuum

$FePS_3$ - Iron phosphorus trisulphide

h-BN - Hexagonal boron nitride

h - Planck constant

i - Imaginary unit

I - Intensity

I_{levelled} - Levelled intensity

IPA - Isopropanol

$\vec{j}$ - Current density

$\vec{k}$ - Direction of wave propagation

λ - Wavelength

λ_0 - Wavelength in vacuum

μ - Magnetic field constant

μ_0 - Magnetic field constant in vacuum

n - Refractive index

n_0 - Refractive index in vacuum and air

$\vec{\nabla}$ - Nablaoperator

$NbSe_2$ - Niobium diselenide

ν - Frequency

ω - Angular velocity

φ - Phase

$\Delta\varphi$ - Phase shift

$\Delta\phi_1$ - Phase shift of first layer

$\vec{r}$ - Position

r_{01} - Relative reflectivity from medium 0 to medium 1

r_{01}^* - Complex conjugate of r_{01}

R - Relative reflected intensity

R_s - Relative reflected intensity of the substrate

ρ - Charge density

s_{av} - Adjacent substrate intensity value

S_{av} - Overall substrate intensity value

Si - Silicon

SiO_2 - Silicon dioxide

subscript$(_e)$ - Incident

subscript$(_r)$ - Reflected

subscript$(_t)$ - Transmitted

t - Time

t_{01} - Relative transmission from medium 0 to medium 1

θ_e - Incident angle

θ_r - Reflected angle

θ_t - Transmitted angle

v - Velocity of light in a medium

w.r.t. - With respect to

$\vec{x}, \vec{y}, \vec{z}$ - Unit vectors

1 Introduction

1.1 Two Dimensional Materials

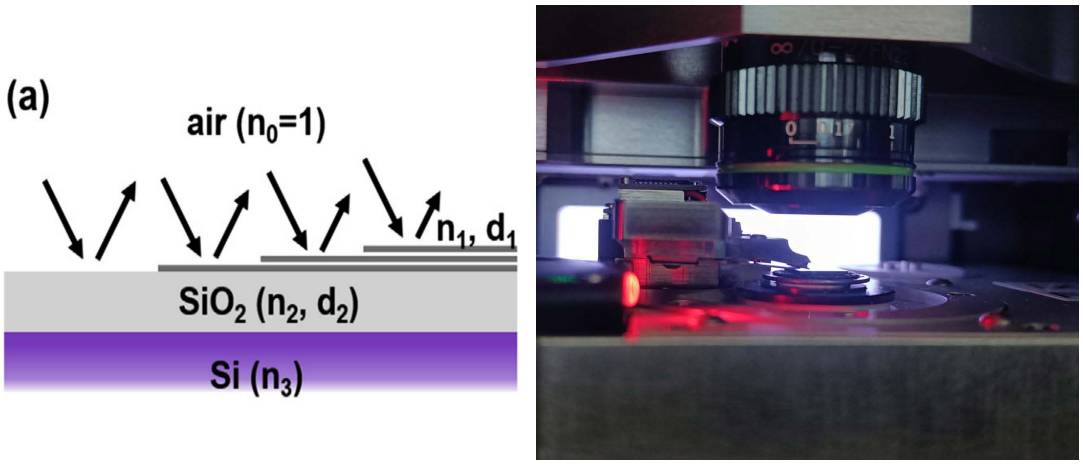
Imagine you were able to design a material with properties of your choice. Almost that much is promised by 2D materials. A two-dimensional material is a thin layer of a material that extends in only one dimension, typically in the nanometer scale. For example, graphene, the very first 2D material discovered, is a very thin layer of graphite. Very often, 2D materials differ from their 3D counterpart not only by size, but also in their properties. Due to their exceptional characteristics, such as high conductivity, flexibility and strength, while also being almost transparent, there is a lot of current research dedicated to them - especially, since they can be stacked onto each other, creating an unlimited amount of combinations of features [12]. Although most 2D materials have only been discovered in recent years, they are not only exciting for exploring new physical phenomena, but there is a vast range of potential applications, for example optical coatings, water purification systems, and drug delivery systems [9].
Typically, two-dimensional materials are produced from materials, which are built up such that the atoms within one layer are bound together by strong bonds, but the connection between the different layers is caused by much weaker Van der Waals forces. Therefore they can be split up into separate/small stacks of layers, called two dimensional materials [3]. The separation process is known as exfoliation. There are various types of exfoliation, for example mechanical exfoliation, which is the one used for this investigation, chemical vapour deposition, epitaxial growth, or liquid exfoliation [10].

1.2 Aim and Research Question

It has been shown, that the properties of 2D materials strongly depend on their thickness [3, 11]. In particular, optical properties change a lot with different numbers of layers. The goal of this investigation is to answer the research question: "How does the optical contrast and the colour of a three-layer system, consisting of thin flakes of the insulator $FePS_3$ and of the metal $NbSe_2$ on SiO_2/Si substrate, change as a function of flake thickness?" In order to establish an answer, a theoretical model is derived and the behaviour of the correlation between light absorption reflection of the mentioned materials and their thickness is determined experimentally.
In a second step the question: "Can the results be used to predict thicknesses for exfoliated insulators?" is answered. The goal is to develop a Python application, which can convert a list of RGB colours into the corresponding thicknesses. If applied for whole images, this application provides a way to identify the flakes of a desired thickness very fast, with low costs, and non-invasive. This enables precise and large-scale thickness characterisation to facilitate the production of 2D materials for further research.

1.3 Choice of Materials

As mentioned above, there are two key materials: Iron phosphorous trisulfide ($FePS_3$) and niobium diselenide ($NbSe_2$). Both are of layered structure, which means, that the layers are held together by weaker forces than there are within a layer, and they can therefore be exfoliated conveniently. The electric conductivity is the main difference between the two materials. $FePS_3$ is an insulator, while $NbSe_2$ has conductive properties, and can even serve as a superconductor at very low temperatures [2, 11]. They have been chosen in such a way to examine whether the theoretical model works for insulators, and whether it also holds for conductive materials, regardless of its fundamental assumptions.
The substrate chips, onto which the flakes are placed after the exfoliation, also play a significant role for the contrast and colour. Conventionally, they are made up of two materials: a 285 nm thick layer of silicon dioxide (SiO_2) on top of 525 µm silicon (Si). Because the flakes are up to 200 nm thick, and the SiO_2 layer is also in this range, the Si layer can be considered semi-infinite. This term describes the assumption that the material extends infinitely below a flat surface.



(a) Flakes on substrate. Taken from [11] (b) AFM

Figure 1: Material configuration and AFM

1.4 Key Instrument: Atomic Force Microscope (AFM)

For thickness identification, particularly of insulating materials, the current procedural standard includes the use of an Atomic Force Microscope. At its core, there is a cantilever with a small spike at its tip. For this study, the contact mode was used. In this mode, the cantilever is traced back and forth just slightly above the sample (a few tens of nanometres). The very precise movement to control the height is done by piezo-materials, which extend a tiny bit, if voltage is applied.
Repulsion and attraction forces, caused mainly by Van der Waals forces, deflect the cantilever [13]. The subsequent movement of a laser ray, which is reflected by the cantilever, is detected by a diode. Thus, the cantilever deflection is measured. A feedback loop then changes the voltages over the piezo-cylinders to adapt the position of the cantilever in such a way, that the deflection is normalised again, to prevent a crash of the cantilever and the sample. In such a

way, an area of at most 30 µm×30 µm can be scanned very precisely ($\pm 0.5 - 3$ nm). Images of dimensions 256×256 and 512×512 pixels were collected in this area - which took between 4 and 8.5 minutes. Since there are different ways of measuring the cantilever deflection, an AFM image contains multiple channels for various sensors[1]:

- Deflection: The signal, which is given from the diode to the piezo-cylinders, which control the height, is recorded. This signal can serve to detect flakes, as it has strong minima and maxima at the edges of flakes.
- Height: The channel, which contains the voltage, which is applied to the piezo-cylinders to control the height, is called 'Height'. This measurement is very precise for thin flakes, but due to non-linearity between extension and voltage, this measurement can be inaccurate for thicknesses above a few tens of nanometres. Additionally, the piezos can change their behaviour after high voltages, which increases the inaccuracy.
- Z-sensor: The Z-sensor measures the change of height of the cantilever. This signal is more linear than the Height measurement for thicker flakes. However, the noise level is higher than for the other two measured values.

[1]Source: adapted from the user manual of the AFM

2 Theoretical Background

2.1 The Model of Light and Light Spectra

To understand what occurs when light is hitting the three layer system, let us consider a light ray which is advancing in space. It behaves as a wave, more specifically, as an electromagnetic wave [15]. Such a wave consists of an electric and a magnetic field, $\vec{E}$ and $\vec{B}$. Both fields are propagating in the direction $\vec{k}$ and oscillating with the same frequency ν (Figure 2) [15]. $\vec{E}, \vec{B}$ and $\vec{k}$ are all orthogonal to each other:

$$\vec{E} \perp \vec{B} \perp \vec{k} \tag{1}$$

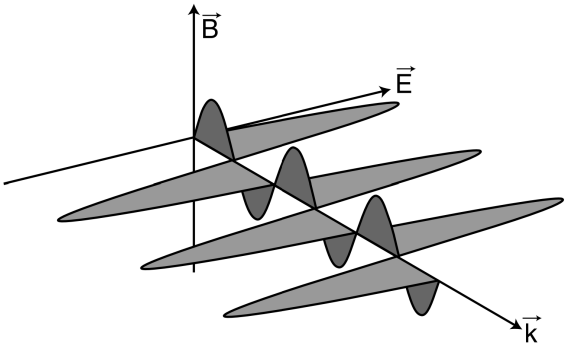


Figure 2: The electric and magnetic components of an electromagnetic wave. Taken from [15]

The energy E of a light wave is denoted by

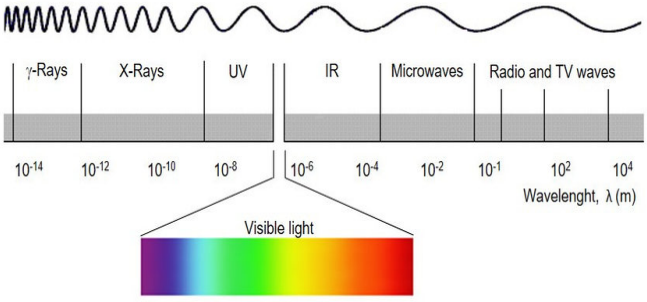
$$E = h\nu \tag{2}$$

where h is the Planck constant and ν the frequency. Since the amount of energy loss in our situation is negligibly small, the light ray has a constant frequency in all media.
In vacuum, a light wave is advancing at a constant speed $c = 1/\sqrt{\varepsilon_0 \mu_0} \approx 3 \times 10^8\,\mathrm{m\,s^{-1}}$ (ε_0 = electrical permittivity, μ_0 = magnetic field constant)[15]. The relation between the speed of a light wave and its frequency is called lambda λ. It denotes the distance between two consecutive peaks of the electric/magnetic field.

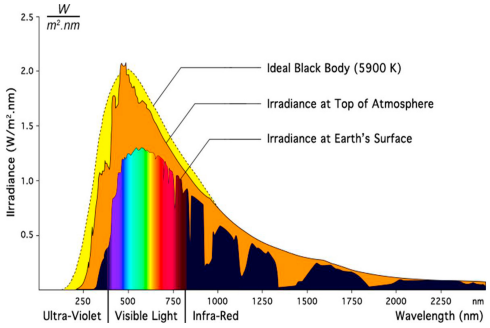
$$\lambda = \frac{c}{\nu} \tag{3}$$

Electromagnetic waves occur in a huge variety of wavelenghts. As humans, we can perceive only a tiny fraction (Figure 3a) - light is visible only in the range $380\,\mathrm{nm} \leq \lambda \leq 770\,\mathrm{nm}$ [15]. Within this range, we observe a variety of colours corresponding to different wavelengths. Generally, the smaller the wavelength, the higher the frequency and thus according to equation (2), the higher the energy. A body, such as the sun, emits light waves of various wavelengths. The amount of intensity per wavelength, which is emitted by a body is called the bodies spectrum. What we perceive as white light is a mixture of various light waves in the range of the visible spectrum.

The reason why sunlight normally appears as white is shown in Figure 3b: The light emitted by the sun consists of almost all wavelengths in the visible range. Since every light source and light detecting device has their own specific emission or sensitivity spectrum, often much less continuous than the one of the sun, the comparisons of intensities over different wavelengths are hard and require calibration.



(a) The electromagnetic spectrum. Taken from [4]



(b) The electromagnetic spectrum of the sun. Taken from [14]

Figure 3: Electromagnetic spectra

2.2 Refractive Index and Absorption

As stated above, the speed of light in a vacuum is c. However, as soon as light passes through matter, it interacts and as a result, it is slowed down. The velocity v of light in a medium is denoted by

$$v = \frac{1}{\sqrt{\varepsilon\varepsilon_0\mu_0}} = \frac{1}{\sqrt{\varepsilon}}c \tag{4}$$

where ε is the material specific electrical permittivity [15]. This difference in velocity must be taken into account to describe what happens when light hits an interface between two media, in which its velocity varies. The relation between the velocity in a medium to the speed of light in vacuum is denoted by the absolute refractive index n defined as

$$n = \sqrt{\varepsilon} = \frac{c}{v} \text{ [15].} \tag{5}$$

The absolute refractive index is material specific. Air has a refractive index very close to 1. E.g glass, typically has a refractive index of 1.5 [15]. At an interface, some of the light ray is reflected back at the same angle (incident angle θ_e = reflected angle θ_r). Figure 4 indicates, that the transmitted angle θ_t of the normal $\vec{u}_n$ and the transmitted ray is not equal to θ_e nor to θ_r. This appearance is called refraction. Using the refractive indices, θ_t can be computed by Snell's law, which is derived from Maxwell's equations and Fermat's principle [15]:

$$n_e \sin(\theta_e) = n_t \sin(\theta_r) \tag{6}$$

A problem arises from the example of a prism: When white light shines through a prism, it is split into its components. Waves of varying frequencies are refracted to a greater or lesser

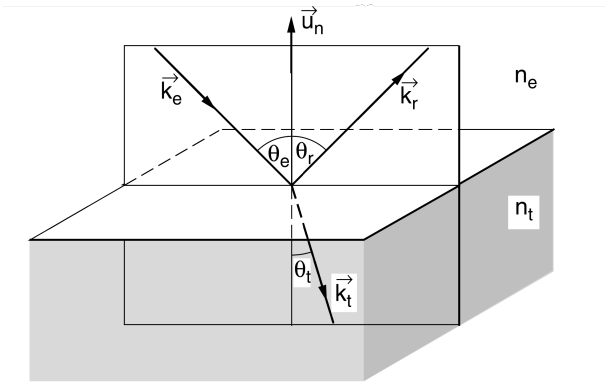


Figure 4: The incident, reflected and transmitted ray. Modified after [15]

extent, therefore they must have interacted unequally strong with the material. This example reveals that the refractive index depends on the light wave frequency. The phenomenon is called dispersion [7]. It reveals, that the definition of n in equation (5) is not complete. Dispersion is caused by absorption. As light passes through matter, some of the waves hit atoms[2]. Depending on the energy (which is proportional to the frequency ν of the wave), the atom behaves in different ways: The electrons surrounding an atom can be in a number of states. Here, we distinguish between an excited state and the normal state. Relative to the normal state, the excited state has a higher energy level. When a light wave arrives with an energy lower than the difference between the normal and the excited state, the energy of the wave is transformed to kinetic and thermal energy, which means that the atom shakes more strongly [7]. If the energy of the wave is greater than the difference between the electron states, they are excited to the higher energy level. As soon as they drop down to the lower state again (happens around 10^8 times per second [7]), and therefore loose the potential energy stored in the higher state, a wave of the same energy as the incoming wave is released in a random direction [7].
Depending on the energy of the incident wave, it is absorbed to a greater or a lesser extent. Since the energy of a wave is proportional to its frequency, the absorption depends on the frequency. The value of absorption is usually determined experimentally for a range of frequencies (e.g. by ellipsometry). Finally, the refractive index for a specific medium n_{medium} is defined as

$$n_{\text{medium}} = n + ik \qquad \text{where } i^2 = -1 \tag{7}$$

n in equation (7) denotes the ratio between the speed of light in vacuum and the velocity in the medium, and k denotes the dampening of the oscillation (since the average amplitude decreases) [7]. For computations in this study, the refractive indices were taken from `https://refractiveindex.info/`.
Introducing the refractive index creates a need to define the wavelength λ more generally than in equation (3), since the wavelength changes proportional to the velocity of light.

$$\lambda_{\text{medium}} = \frac{v}{\nu} = \frac{c}{n_{\text{medium}}\nu} \tag{8}$$

[2]When talking about absorption, light usually is described as a photon. Since these terminologies require explanations beyond the scope of this study, the terms for a wave are used.

2.3 Interference

2.3.1 Wave Description

It follows from Maxwell's equations (introduced in section 3) that both the electric and the magnetic field can be described using sinusoidal functions [15]. From the frequency, the angular velocity $\omega = 2\pi\nu$ can be determined. The functions for the wave depend on the time t, the position $\vec{r}$, the phase φ and the amplitudes $\vec{E}_0$ and $\vec{B}_0$

$$\vec{E} = \vec{E}_0 \cos\left(\vec{k}\vec{r} - \omega t + \varphi\right) \tag{9}$$

$$\vec{B} = \vec{B}_0 \cos\left(\vec{k}\vec{r} - \omega t + \varphi\right) \tag{10}$$

For the theory model, a different notation, obtained by Euler's formula, is used [15]:

$$\vec{E} = \vec{E}_0 e^{i(\vec{k}\vec{r} - \omega t + \varphi)} = \vec{E}_0 \left(\cos\left(\vec{k}\vec{r} - \omega t + \varphi\right) + i \sin\left(\vec{k}\vec{r} - \omega t + \varphi\right)\right) \tag{11}$$

$$\vec{B} = \vec{B}_0 e^{i(\vec{k}\vec{r} - \omega t + \varphi)} = \vec{B}_0 \left(\cos\left(\vec{k}\vec{r} - \omega t + \varphi\right) + i \sin\left(\vec{k}\vec{r} - \omega t + \varphi\right)\right) \tag{12}$$

All computations are done this notation, because it makes them more convenient; in the end, only the real part of the result will be taken and is physically measurable.

2.3.2 Addition of Two Waves of Equal Frequency

With the help of equations (9) and (10), it is possible to describe what is happening when two waves of the same frequency are travelling in the same direction. In this case, the waves can be added together. The formal calculation is done in reference [7]. As a result of the addition, there is a new wave. The most important factor is the phase shift between the two phases of the waves, $\Delta\varphi = \varphi_1 - \varphi_2$ [7]. The effect is shown in Figure 5. In the upper graph, the phase

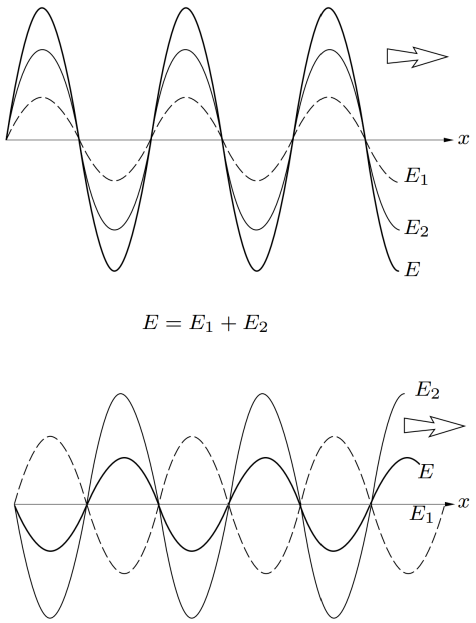


Figure 5: The principle of interference. E, E_1, E_2 denote the electric field. Taken from [7]

shift $\Delta\varphi = 0$; the waves add up to a new wave of higher amplitude. This is called constructive interference [7]. The lower graph shows the opposite; if the phase shift $\Delta\varphi = \pi$, the maxima of one wave coincide with to the minima of the other wave and result in a new wave of lower amplitude. This is called destructive interference [7]. If the amplitudes of the two waves were equal, the resulting wave would even have an amplitude of 0.

2.4 Principle of Thickness Identification by Optical Imaging

Both the refractive index and interference play a crucial role in the principle of determining thickness using optical imaging. A beam of white light is shone perpendicularly onto the surface of a flake. Some of the light is reflected, and some enters the material. All of the rays, which are returning to the surface from within the materials, for example the part of the transmitted ray, which is reflected at the bottom of the flake, have a phase shift compared to the reflected ray. This phase shift depends on the refractive index of the material and on the thickness of the flake. Due to differences in phase shift for various thicknesses, the wavelengths, of which the reflected rays undergo constructive interference, and the wavelengths, for which destructive interference occurs, are changing with thickness [2, 11]. Thus measuring the colour of the reflected light encodes the thickness of the flakes. This method has a variety of benefits: measuring the colour requires a lot less expensive technique than other types of thickness measurements. It is also much faster and non-invasive (the flakes are not damaged).

2.5 Colour Codes in Digital Cameras

Finally, a word on digital colour encoding. As soon as light waves hit a camera sensor, their intensity is registered. Shortly before, all wavelengths are filtered into three categories: red, green and blue. Together the three intensity values for red, green and blue, each between 0 and 255, form the RGB value, into which any colour, that displayed on a screen, is encrypted by additive colour blending. In this paper, intensity values are presented without indicated units. Unfortunately, there are factors, which increase the difficulty of mapping wavelengths onto RGB values: for example, a short wavelength, which we perceive as violet, has high red and blue values. On the other end of the spectrum, a long wavelength, corresponding to red, is represented as a high red value. However, in the middle of the spectrum, red values are very low. Thus the translation between wavelength and RGB value has to be approximated.

3 Theoretical Model

To develop a well-founded hypothesis, we derive a theoretical model[3] for the expected amount of light reflected by a three-layer system. If the experimental data supports the hypothesis, this would further validate the underlying assumptions behind the theoretical model. The model is based on Fresnel's Laws, which in turn can be derived directly from Maxwell's equations. Since it appears often in literature with very little explanations, we will derive it step by step in the following chapter.

3.1 Reflection and Transmission - Fresnel Equations for Perpendicular Incidence

3.1.1 Boundary Conditions

As mentioned in section 2, let us consider light as an electromagnetic wave. A light wave consists of an oscillating electric and magnetic field which are perpendicular to each other. We can define a coordinate system, where both fields travel in $+z$ direction, thus the electric field oscillates in the xz-plane and the magnetic field in the yz-plane (Figure 2 with $\vec{k} = \vec{z}$). To get started, we set our boundary conditions: The light behaves as a wave when switching from medium to medium - so the tangential component of the electric field strength $\vec{E}$ and its counterpart the magnetic field strength $\vec{H} = \frac{1}{\mu_0 \mu}\vec{B}$ must be continuous [15]. This implies that the x and y coordinates of $\vec{E}$ and $\vec{B}$ respectively must be equal just above and below the boundary. Additionally, since energy conservation holds, the frequency of the light does not depend on the medium and is constant. Using the notation $e^{i(\vec{k}\vec{z}-\omega t)}$ for our wave, representing $\cos(\vec{k}\vec{z}-\omega t) \cdot i\sin(\vec{k}\vec{z}-\omega t)$, we get the following condition:

$$\vec{E}_{0,e}e^{i(\vec{k}_e\vec{z}-\omega t)} + \vec{E}_{0,r}e^{i(\vec{k}_r\vec{z}-\omega t)} = \vec{E}_{0,t}e^{i(\vec{k}_t\vec{z}-\omega t)} \tag{13}$$

$$\vec{B}_{0,e}e^{i(\vec{k}_e\vec{z}-\omega t)} + \vec{B}_{0,r}e^{i(\vec{k}_r\vec{z}-\omega t)} = \vec{B}_{0,t}e^{i(\vec{k}_t\vec{z}-\omega t)} \tag{14}$$

Where $\vec{E}_{0,e},\ \vec{E}_{0,r},\ \vec{E}_{0,t}$ denote the amplitude of the incident, reflected and transmitted electric fields, respectively. For simplicity, we consider the case in which the angle of incidence is perpendicular to the surface of the sample. This assumption is reasonable, since the microscope has a built-in light source going through its lens and hits the sample from directly above. Since we are interested in the reflected intensity, we can focus only on the amplitudes [15]:

$$\vec{E}_{0,e} + \vec{E}_{0,r} = \vec{E}_{0,t} \tag{15}$$

$$\vec{B}_{0,e} + \vec{B}_{0,r} = \vec{B}_{0,t} \tag{16}$$

[3] Dr. Hugo Anders indicates ROUARD as the original source [1].

3.1.2 Maxwell's Equations

For our purposes, we are presupposing to be treating isotropic, non-magnetic ($\mu = 1$), homogeneous, even surfaced materials. Additionally, the materials are assumed be insulating, such that charge density $\rho = 0$ and current density $\vec{j} = 0$. This criterion is not fulfilled for $NbSe_2$. If the model would hold for $NbSe_2$, the necessity of the assumption should be reconsidered.
Under the assumptions, Maxwell's equations take the form:

$$\varepsilon\varepsilon_0 \vec{\nabla} \cdot \vec{E} = 0 \tag{17}$$

$$\vec{\nabla} \cdot \vec{B} = 0 \tag{18}$$

$$\vec{\nabla} \times \vec{E} = -\frac{\partial B}{\partial t} \tag{19}$$

$$\vec{\nabla} \times \vec{B} = \mu_0 \varepsilon \varepsilon_0 \frac{\partial E}{\partial t} \tag{20}$$

where $\vec{\nabla}$ is the Nablaoperator, ε and ε_0 are the electrical permittivity for the material and vacuum resp., and μ_0 is the magnetic field constant [15]. For $\vec{x}$, $\vec{y}$, and $\vec{z}$ being the unit vectors of a three dimensional Cartesian coordinate system, $\vec{\nabla} = \frac{\partial}{\partial \vec{x}} + \frac{\partial}{\partial \vec{y}} + \frac{\partial}{\partial \vec{z}}$. Plugging in $\vec{E} = \vec{E}_0 e^{i(\vec{k}\vec{z} - \omega t)}$ and $\vec{B} = \vec{B}_0 e^{i(\vec{k}\vec{z} - \omega t)}$ into equation (19), we derive:

$$\vec{\nabla} \times \vec{E}_0 e^{i(\vec{k}\vec{z} - \omega t)} = -\frac{\partial}{\partial t}(\vec{B}_0 e^{i(\vec{k}\vec{z} - \omega t)}) \tag{21}$$

Travelling along the z-axis, $\vec{E}$ only oscillates in the xz-plane of the coordinate system that we are working in. Its derivative w.r.t. $\vec{x}$ or $\vec{y}$ therefore is 0. Using the cross-multiplication rules, keeping $\vec{E}_0 = \text{const.}$, and expanding yields:

$$\left(\frac{\partial}{\partial \vec{x}} e^{i(\vec{k}\vec{z} - \omega t)} + \frac{\partial}{\partial \vec{y}} e^{i(\vec{k}\vec{z} - \omega t)} + \frac{\partial}{\partial \vec{z}} e^{i(\vec{k}\vec{z} - \omega t)}\right) \times \vec{E}_0 = -(-i\omega) \cdot \vec{B}_0 e^{i(\vec{k}\vec{z} - \omega t)} \tag{22}$$

$$\left(0 + 0 + i\vec{k} \cdot e^{i(\vec{k}\vec{z} - \omega t)}\right) \times \vec{E}_0 = i\omega \cdot \vec{B}_0 e^{i(\vec{k}\vec{z} - \omega t)} \tag{23}$$

$$i\vec{k} \times \vec{E}_0 e^{i(\vec{k}\vec{z} - \omega t)} = i\omega \cdot \vec{B}_0 e^{i(\vec{k}\vec{z} - \omega t)} \tag{24}$$

$$\vec{k} \times \vec{E}_0 = \omega \cdot \vec{B}_0 \tag{25}$$

Solving for $\vec{B}_0$ results in:

$$\vec{B}_0 = \frac{1}{\omega}(\vec{k} \times \vec{E}_0) \tag{26}$$

This equation describes the relation between the magnetic and the electric field of a light wave - one cannot exist without the other, they are perpendicular to each other (cross product), and both are oscillating with the same frequency.

3.1.3 Developing Fresnel's Equations

Combining the results from the previous two sections, we insert (26) into (16).

$$\frac{1}{\omega}(\vec{k}_e \times \vec{E}_{0,e}) + \frac{1}{\omega}(\vec{k}_r \times \vec{E}_{0,e}) = \frac{1}{\omega}(\vec{k}_t \times \vec{E}_{0,t}) \tag{27}$$

ω cancels, since it is independent of the medium. We can further simplify: the incident ($_e$) and reflected ($_r$) rays are anti-parallel, therefore $\vec{k}_r = -\vec{k}_e$. The transmitted ($_t$) ray has the same direction as $\vec{k}_e$, but is slower due to the surrounding matter: $\vec{k}_t = \frac{n_t}{n_e}\vec{k}_e$.

$$(\vec{k}_e \times \vec{E}_{0,e}) - (\vec{k}_e \times \vec{E}_{0,r}) = (\frac{n_t}{n_e}\vec{k}_e \times \vec{E}_{0,t}) \tag{28}$$

$$\vec{E}_{0,e} - \vec{E}_{0,r} = \frac{n_t}{n_e}\vec{E}_{0,t} \tag{29}$$

$$n_e\vec{E}_{0,e} - n_e\vec{E}_{0,r} = n_t\vec{E}_{0,t} \tag{30}$$

Using both conditions, which were set up in section 3.1.1, we combine (30), originating from (16), and (15), and solve for both $E_{0,r}$ and $E_{0,t}$ separately.

$$\text{from (15): } \vec{E}_{0,e} + \vec{E}_{0,r} = \vec{E}_{0,t} \tag{31}$$

$$\text{from (30): } \frac{n_e}{n_t}(\vec{E}_{0,e} - \vec{E}_{0,r}) = \vec{E}_{0,t} \tag{32}$$

$$n_t(\vec{E}_{0,e} + \vec{E}_{0,r}) = n_e(\vec{E}_{0,e} - \vec{E}_{0,r}) \tag{33}$$

$$n_t\vec{E}_{0,r} + n_e\vec{E}_{0,r} = n_e\vec{E}_{0,e} - n_t\vec{E}_{0,e} \tag{34}$$

$$\vec{E}_{0,r} = \frac{n_e - n_t}{n_e + n_t}\vec{E}_{0,e} \tag{35}$$

$$\text{Similarly, } \vec{E}_{0,t} - \vec{E}_{0,e} = \vec{E}_{0,e} - \frac{n_t}{n_e}\vec{E}_{0,t} = \vec{E}_{0,r} \tag{36}$$

$$\vec{E}_{0,t} + \frac{n_t}{n_e}\vec{E}_{0,t} = 2\vec{E}_{0,e} \tag{37}$$

$$\vec{E}_{0,t}(n_e + n_t) = 2n_e\vec{E}_{0,e} \tag{38}$$

$$\vec{E}_{0,t} = \frac{2n_e}{n_e + n_t}\vec{E}_{0,e} \tag{39}$$

We define the relative reflectivity r_{et} and relative transmission t_{et}:

$$r_{et} = \frac{n_e - n_t}{n_e + n_t} = \frac{\vec{E}_{0,r}}{\vec{E}_{0,e}} \qquad t_{et} = \frac{2n_e}{n_e + n_t} = \frac{\vec{E}_{0,t}}{\vec{E}_{0,e}} \tag{40}$$

To arrive at the relative reflected intensity R, in which we are interested, we square r or, in case r is complex, multiply it by its complex conjugate

$$R = \frac{I_r}{I_e} = r_{et}r_{et}^* \tag{41}$$

Note: there is a difference between $r_{et} = \frac{n_e - n_t}{n_e + n_t}$ and $r_{te} = \frac{n_t - n_e}{n_t + n_e}$; in fact, $r_{te} = -r_{et}$. Further, $t_{et} = 1 + r_{et}$, since $\frac{2n_e}{n_e+n_t} = \frac{n_e-n_t}{n_e+n_t} + \frac{n_e+n_t}{n_e+n_t} = \frac{n_e-n_t}{n_e+n_t} + 1$.

3.2 Reflection of Light on Interfaces

The main source for this section is [1].

3.2.1 Single Interface

To demonstrate, how Fresnel's equations work, suppose a light ray is travelling in air ($n_{air} = n_0 = 1$) and hits an even surface in the xy-plane ($n_{matter} = n_1 = 1.5$) perpendicularly. How much of the light is reflected?

$$R = r_{01} r_{01}^* = r_{01}^2 \text{ (here } n_0, n_1 \in \mathbb{R} \text{ have no complex part)} \tag{42}$$

$$R = \left(\frac{n_0 - n_1}{n_0 + n_1}\right)^2 = \left(\frac{1 - 1.5}{1 + 1.5}\right)^2 = 0.04 = \underline{\underline{4\%}} \text{ is reflected.} \tag{43}$$

3.2.2 Two Interfaces

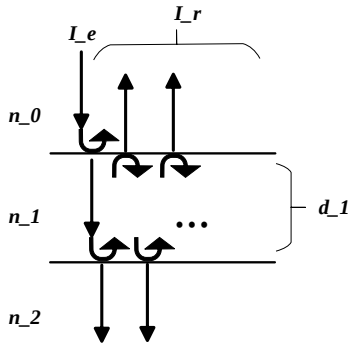


Figure 6: Model of light rays passing through two interfaces perpendicularly. The three dots indicate infinite repetition. Modified after [5].

If a ray of intensity I_e is entering a material with a small thickness d_1, and there is another material with refractive index $n_2 \neq n_1$ underneath, calculating the reflected intensity I_r becomes more challenging. Figure 6 illustrates the situation: $r_{01} r_{01}^*$ of the incoming ray is reflected at the first interface. The transmitted portion enters the upper material, some of it is absorbed, some of it passes on to the lower material, and some of it is reflected off the second interface. The latter travels back up until it reaches the upper interface. Here, some of the ray is transmitted - this part is observable and contributes to the reflected intensity I_r - and some is reflected again. This creates three aspects, that we have to worry about:

- Absorption: how much intensity is lost when travelling through matter?
- Interference: the rays passing through the material and returning have a phase shift compared to all the other rays, which have been either directly reflected or travelled a shorter or longer distance inside the material.
- Repetition: The ray can be reflected on both interfaces repeatedly - each time, a smaller amount of the original ray passes through the first interface and accounts for the reflected intensity. This happens infinitely many times.

Absorption

A part of the transmitted light ray is absorbed by the matter. The process of absorption is discussed in section 2.2. n is defined such that it accounts for absorption.

Interference

The phenomenon of interference, explained in section 2.2, is the reason, why the amount of light reflected off the first interface and the ray which has passed down and up once through the upper material can not simply be added together. The second ray has a phase-shift $\Delta\phi_1$ compared to the first one. We can deduce the phase-shift with the help of the equation of the wave and the Figure 6:

$$\vec{E}_0 e^{i(\vec{k}\vec{z}-\omega t)} \tag{44}$$

Here, $\vec{k}\vec{z}$ denotes the phase of the wave w.r.t. a certain position $\vec{z}$. The phase-shift $\Delta\phi_1$ is built up in the same way - there occurs a phase shift because of spatial difference. As the wave travels the distance d_1 within the upper material, it picks up a phase shift of $\Delta\phi_1 = |\vec{k}_1|d_1$. Since $|\vec{k}_1| = n_1|\vec{k}_0| = \frac{2\pi}{\lambda}n_1$, where $\vec{k}_0$ denotes the original $\vec{k}$ from above, expanding yields:

$$\Delta\phi_1 = \frac{2\pi n_1 d_1}{\lambda} \tag{45}$$

Repetition until Infinity: Developing the Equation

Let us examine the first few rays displayed in Figure 6 to illustrate the process. Initially, $r_{tot} \approx r_{01}$. A fraction passes the upper interface (t_{01}), picks up a phase-shift of $\Delta\phi_1$, and some of it is reflected at the lower interface (r_{12}). A fraction of the ray is transmitted and travels away in the lowest medium. As the reflected ray travels up again, it picks up a second phase shift, before most of it is transmitted through the upper interface from below. For the moment,

$$r_{tot} \approx r_{01} + t_{01}e^{-i\Delta\phi_1}r_{12}e^{-i\Delta\phi_1}t_{10}. \tag{46}$$

But we have to continue the path of the ray, which has travelled once down and up, isn't transmitted but reflected (r_{10}), picks up $\Delta\phi_1$, is reflected again (r_{12}), picks up $\Delta\phi_1$ and is transmitted through the upper interface (t_{10}).

$$r_{tot} \approx r_{01} + t_{01}r_{12}t_{10}e^{-i2\Delta\phi_1} + t_{01}r_{12}r_{10}r_{12}t_{10}e^{-i4\Delta\phi_1} \tag{47}$$

Continuing the paths, we arrive at:

$$r_{tot} = r_{01} + t_{01}r_{12}t_{10}e^{-i2\Delta\phi_1} + t_{01}r_{12}r_{10}r_{12}t_{10}e^{-i4\Delta\phi_1} + t_{01}r_{12}r_{10}r_{12}r_{10}r_{12}t_{10}e^{-i6\Delta\phi_1} + \dots \tag{48}$$

$$r_{tot} = r_{01} + t_{01}r_{12}t_{10}e^{-i2\Delta\phi_1}(1 + r_{10}r_{12}e^{-i2\Delta\phi_1} + (r_{10}r_{12}e^{-i2\Delta\phi_1})^2 + (r_{10}r_{12}e^{-i2\Delta\phi_1})^3 + \dots \tag{49}$$

$$r_{tot} = r_{01} + t_{01}r_{12}t_{10}e^{-i2\Delta\phi_1}\sum_{n=0}^{\infty}(r_{10}r_{12}e^{-i2\Delta\phi_1})^n \tag{50}$$

Applying the formula for an infinite geometric series, since $r_{10}r_{12}e^{-i2\Delta\phi_1} < 1$, and expanding yields the equation for the relative reflectivity of a two-layer system:

$$r_{tot} = r_{01} + \frac{t_{01}r_{12}t_{10}e^{-i2\Delta\phi_1}}{1 - r_{10}r_{12}e^{-i2\Delta\phi_1}} \tag{51}$$

$$r_{tot} = \frac{r_{01} - r_{01}r_{10}r_{12}e^{-i2\Delta\phi_1} + t_{01}r_{12}t_{10}e^{-i2\Delta\phi_1}}{1 - r_{10}r_{12}e^{-i2\Delta\phi_1}} \tag{52}$$

$$r_{tot} = \frac{r_{01} + r_{01}r_{01}r_{12}e^{-i2\Delta\phi_1} + (1 + r_{01})(1 - r_{01})r_{12}e^{-i2\Delta\phi_1}}{1 + r_{01}r_{12}e^{-i2\Delta\phi_1}} \tag{53}$$

$$r_{tot} = \frac{r_{01} + r_{01}^2 r_{12}e^{-i2\Delta\phi_1} + r_{12}e^{-i2\Delta\phi_1} - r_{01}^2 r_{12}e^{-i2\Delta\phi_1}}{1 + r_{01}r_{12}e^{-i2\Delta\phi_1}} \tag{54}$$

$$r_{tot} = \frac{r_{01} + r_{12}e^{-i2\Delta\phi_1}}{1 + r_{01}r_{12}e^{-i2\Delta\phi_1}} \tag{55}$$

3.2.3 Three Interfaces

To predict the amount of reflected light from a flake of thickness d_1 on a SiO_2/Si substrate, where the thickness of SiO_2 is denoted by d_2, we can simply plug in the same equation for the relative reflectivity $r_{12} \to r'_{12}$. Figure 7 visualises this mathematical idea. To arrive at a final formula, we define:

$$r_{tot} = \frac{r_{01} + r'_{12}e^{-i2\Delta\phi_1}}{1 + r_{01}r'_{12}e^{-i2\Delta\phi_1}} \qquad\qquad r'_{12} = \frac{r_{12} + r_{23}e^{-i2\Delta\phi_2}}{1 + r_{12}r_{23}e^{-i2\Delta\phi_2}} \tag{56}$$

Plugging in and simplifying yields:

$$r_{tot} = \frac{r_{01} + \frac{r_{12}+r_{23}e^{-i2\Delta\phi_2}}{1+r_{12}r_{23}e^{-i2\Delta\phi_2}}e^{-i2\Delta\phi_1}}{1 + r_{01}\frac{r_{12}+r_{23}e^{-i2\Delta\phi_2}}{1+r_{12}r_{23}e^{-i2\Delta\phi_2}}e^{-i2\Delta\phi_1}} \tag{57}$$

$$r_{tot} = \frac{\frac{r_{01}(1+r_{12}r_{23}e^{-i2\Delta\phi_2}) + r_{12}e^{-i2\Delta\phi_1} + r_{23}e^{-i2\Delta\phi_2}e^{-i2\Delta\phi_1}}{1+r_{12}r_{23}e^{-i2\Delta\phi_2}}}{\frac{1+r_{12}r_{23}e^{-i2\Delta\phi_2} + r_{01}r_{12}e^{-i2\Delta\phi_1} + r_{01}r_{23}e^{-i2\Delta\phi_2}e^{-i2\Delta\phi_1}}{1+r_{12}r_{23}e^{-i2\Delta\phi_2}}} \tag{58}$$

$$\boxed{r_{tot} = \frac{r_{01} + r_{12}e^{-i2\Delta\phi_1} + r_{23}e^{-i2(\Delta\phi_1+\Delta\phi_2)} + r_{01}r_{12}r_{23}e^{-i2\Delta\phi_2}}{1 + r_{01}r_{12}e^{-i2\Delta\phi_1} + r_{01}r_{23}e^{-i2(\Delta\phi_1+\Delta\phi_2)} + r_{12}r_{23}e^{-i2\Delta\phi_2}}} \tag{59}$$

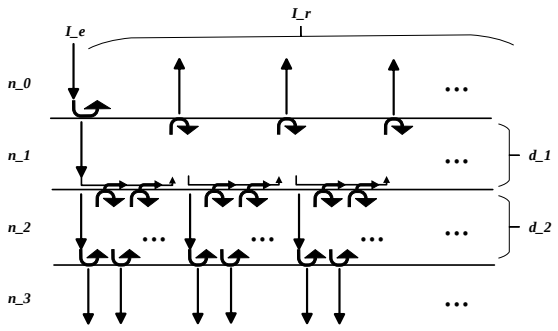


Figure 7: Model of light rays passing through three interfaces perpendicularly. Three dots indicate infinite repetition. Modified after [1].

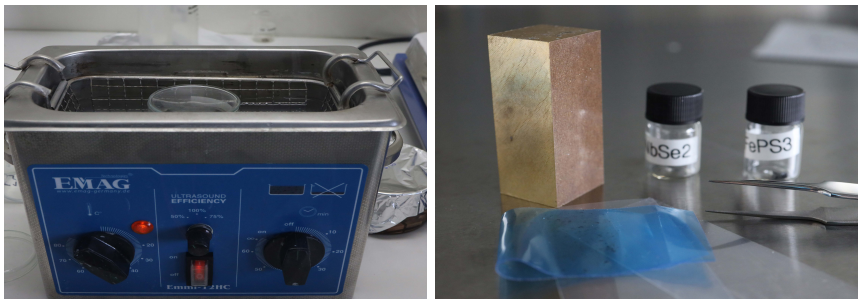
4 Methodology

4.1 Experiment

In order to answer the research question and produce data for further applications, the height and the red, green, blue and greyscale intensity values of various flakes of both iron phosphorous trisulfide $FePS_3$ and niobium diselenide $NbSe_2$ were measured. Materials of current procedural standard were used to achieve best precision. This is necessary as the results should be used for further research. The experiment can be replicated with less expensive material (except for the AFM), but lower precision has to be expected. A small preliminary experiment, including data analysis, proved to be very insightful and is highly recommended. A short summary of the Methodology can be found in the Appendix.

4.1.1 Materials

- SiO_2/Si substrate chips with markers
- Tweezers (metal and rubber)
- 10 ml acetone
- 10 ml isopropanol (IPA)
- Ultrasonic bath
- Water (if possible, de-ionised)
- Gaseous Nitrogen
- Two small beakers with lid
- Small boxes to store substrate chips (ideally with adhesive gel coating, e.g. Polydimethylsiloxan (PDMS))
- Adhesive tape
- Brick (weight=50 g)
- Optical microscope with camera
- Atomic Force Microscope
- SiO_2/Si substrates with markers (cleaned)



(a) Ultrasonic bath & closed beaker (b) Materials for exfoliation

Figure 8: Materials

4.1.2 Substrate Preparation

SiO_2/Si substrate chips, onto which markers had been written with gold and which were sealed with a coating to prevent contamination and physical deformation, were provided. To prepare them for further use, this coating had to be removed in a process called 'resist stripping'. Acetone removes most of the coating, IPA removes small remainders.

1. The water in the ultrasonic bath was heated up to 40°C. The acetone and the chips were put into a beaker with a lid using tweezers. The lid was closed and placed into the water for 20 minutes.

2. The ultrasonic cleaning was switched on for 15 minutes.

3. Isopropanol was poured into another beaker. Again using tweezers, the chips were transferred from the acetone beaker to the IPA and sonicated for another 15 minutes.

4. The chips were then taken out individually and dried immediately using gaseous nitrogen to prevent traces of drying. Finally, the solvents were disposed separately.

4.1.3 Exfoliation

The process of obtaining flakes from a bulk crystal is called exfoliation and is described in this section. The same process has been done for both materials and for each substrate chip.

1. A small piece of crystal was placed on the blue adhesive tape (Figure 8b) using tweezers. The piece was removed from the tape immediately, leaving behind a very small residue on the tape. If the material is harder, it is necessary to press the piece slightly onto the tape before removing it, to ensure that there is an adequate amount of material on the tape.

2. The tape was then folded together (adhesive sides touching), then gently torn apart, so that the material is distributed on two spots on the tape. When folding the tape repeatedly, while shifting it such that the spots do not land on the same spot every time, the cohesive parts of material become thinner, but unfortunately smaller as well. The folding was done 8 times - this number can be adapted depending on whether larger and thicker or thin and small pieces are required.

3. One of the substrate plates was placed onto the adhesive tape with tweezers (SiO_2/Si surface onto adhesive side). The brick was placed on top of the plate for 10-15 s to apply slight pressure and ensure that there is some material on the substrate. The substrate plate was then torn gently off the tape to make the sample finally ready for investigation.

4.1.4 Optical Imaging

The freshly prepared samples were placed under the optical microscope and illuminated by a lamp through the lens of the microscope. Note that the optical images were taken in the first 30 minutes after exfoliation to prevent distortion by oxidation. All images for data collection were taken with the 50X magnification lens (e.g. appendix and Figure 9a, image size: 243 µm × 182 µm). To find the same flakes later in the AFM, another image with the 20X mag. lens was taken (where the adjacent markers on the substrate are visible; image size: 768 µm × 576 µm). As the resolution of 2592×1944 pixels is relatively low for the small flakes, images were only taken, if the flakes were large enough to allow a colour value to be obtained later on and if the flakes were of distinct height (some big flakes have a wavy surface, where the height varies over short distances). Nevertheless, an even distribution of data for all colours and heights was pursued. To make the images comparable, the camera was set to 5.7 ms exposure, 1.9x gain,

100.00 saturation and 0.6 gamma. Furthermore, auto-white balance was done using a white test stripe and the aperture was closed as much as possible for sharpness over the whole height. Any shading, sharpening or contrast was turned off, and the files were saved in png-format.

4.1.5 AFM Imaging

The substrate chip was fixed on the AFM sample holder using adhesive carbon tape. The flakes of which pictures had been taken were scanned in the AFM (256 × 256 or 512 × 512 pixels (depending on the complexity of the flakes), each picture 30 µm× 30 µm, contact mode), to determine the exact (± 0.5-3 nm) height. A sample image is shown in Figure 9c.

4.1.6 Safety Regulations

During the process of substrate preparation and exfoliation, gloves and goggles were worn. Any substances used in contact with the substrate plate were disposed separately. The substrate plates were only moved around with tweezers to prevent contamination. Substrate stripping and exfoliation was done under fume hoods. Solvents used (acetone, isopropanol) must not get in contact with skin or mucous membranes and must be disposed separately.

4.2 Data Evaluation

The data evaluation was done using Gwyddion for AFM images, Spyder as Python compiler, Microsoft Excel for data storage, and Desmos Scientific Calculator for curve analysis. The corresponding versions can be found in the References.

4.2.1 Measuring Colour Values and Computation of Contrast

To obtain the contrast values for each measured thickness, the colour values at the precisely same position had to be determined first. To speed up the process of data extraction, a Python script was written (see appendix). Upon entering the values of the desired points x and y coordinates in the optical image, the program yields the RGB and greyscale values for every pixel in a 5×5 pixel square to the bottom right of the indicated point, as well as the average red, green, blue and greyscale values, and their standard deviation over the square and the x, y values. It then displays the image, marking the measured square black to check the position. Finally, the extracted values were copied to Microsoft Excel and the following calculations were done:

$$\text{Levelled RGB and greyscale values } I_{\text{levelled}} = I - s_{av} + S_{av} \tag{60}$$

$$\text{Contrast: } C = \frac{S_{av} - I_{\text{levelled}}}{S_{av}} \tag{61}$$

Where I is the red, green, blue or greyscale value of a flake, s_{av} the corresponding adjacent substrate value (average of the three measurements), and S_{av} the corresponding average of all substrate values. Such a 5x5 pixel square was attained for each profile and three times for the substrate close to the area being investigated for each group of flakes on one AFM image.

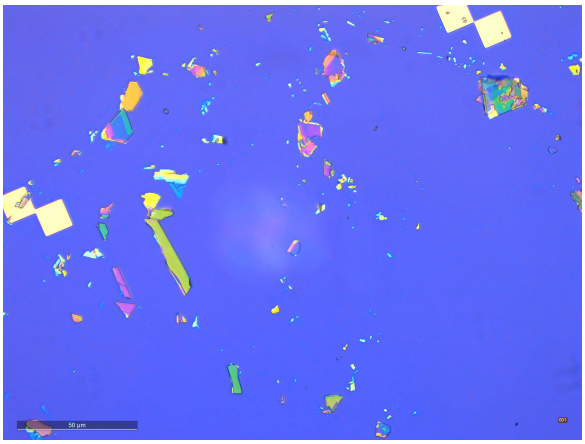
4.2.2 Measuring Height

The heights of individual flakes were extracted from AFM images using Gwyddion. The data of the Z-Sensor contains a map of heights that requires processing for more precise results. The maps were levelled, using the 'Level data by fitting a plane through three points', because when placing a sample into the AFM, there is always a slight inclination. This is due to the sample load platform, which is not perfectly flat - in the frame of nanometres, this tilt is significant. The level of the plane was then set to zero such that the displayed height is with respect to the substrate surface (For plane fit: Averaging radius = 3; the three points were distributed in different corners and onto the substrate).
To measure the height of a flake, profiles were extracted along straight lines (white lines in Figure 9c; thickness = 2 pixels). Each line goes from smooth substrate to a flat part of a flake, containing an edge as sharp as possible. The lines were drawn horizontally, following the direction in which the data was recorded to enable high precision. A few profiles of interesting flakes were drawn vertically instead of horizontally, because neighbouring flakes would prevent accurate height measurements. For each flake, 3 profiles were drawn per area with similar height to reduce the effect of outliers arising throughout the data evaluation.
For each AFM image, a graph (Figure 9d) was produced, showing all extracted profiles. Using the 'Fit critical dimension' function in Gwyddion, the height of the step in each profile, including the measurement error, was determined and recorded in Microsoft Excel. All processed AFM images and the corresponding graphs are shown in the appendix.

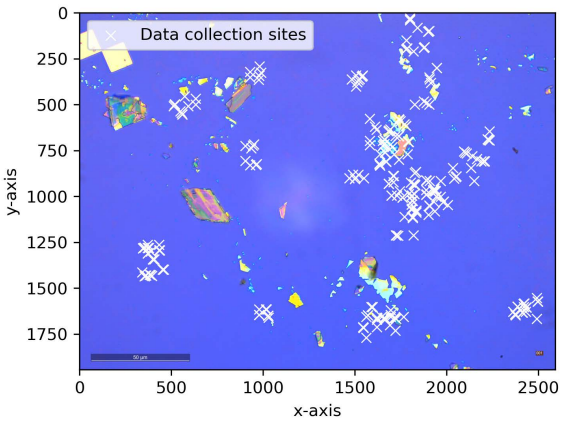
5 Results

5.1 Experimental Data $FePS_3$

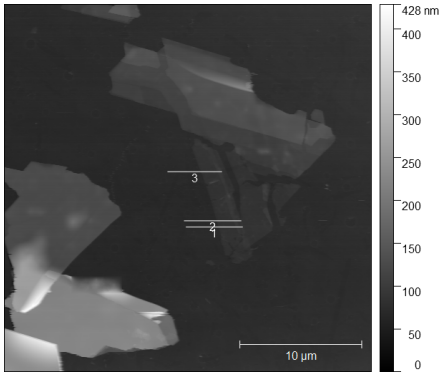
In this subchapter, the experimental results for the reflected intensity, contrast and colour per thickness of $FePS_3$ are presented. They have been extracted from pictures like the ones shown in Figure 9. Both in Figure 9a and 9b, a slight blur in the middle of the image and a few darker dots are visible. They are part of the microscope illumination and camera system, and to prevent errors in measuring, the data has been measured in places without such contamination, as indicated in Figure 9b by white crosses. All the optical and AFM images can be found in the appendix.



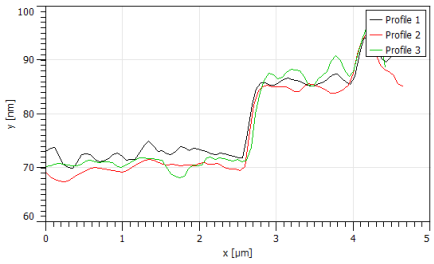
(a) Optical microscope



(b) Data collection sites



(c) Processed AFM image



(d) Profiles from AFM image

Figure 9: Example of $FePS_3$ data

5.1.1 Reflected Intensity $FePS_3$

Figure 10 shows the levelled reflected intensity for the red, green, blue and greyscale channels. The values have been levelled with the equation (60) as described in the end of section 4. A few values are above 255 due to the levelling. Measurements of this material have been taken up to 200 nm thickness on two different days. The black points have been measured on the first day, the coloured (& grey) points on the second day with fresh samples. Although attention was paid to the settings of the camera, factors beyond our control resulted in a certain systematic error. The pictures from the second day are overall slightly darker than the ones taken on the

first day. Any other small measurement errors are indicated by the bars at one point: horizontal lines indicate the error in the height measurement, vertical lines indicate the standard deviation of the colour measurement (More elaboration on errors in section 6.3).
For each colour, peaks are reached at different thicknesses: the shortest wavelengths, contributing mainly to the colour blue, are at a very high level on the thinnest flakes. Another peak is at around 95-115 nm. The medium wavelengths, contributing mainly to the colour green, reach a peak between 30 nm and 60 nm. The data from the second day suggests a peak at 30-40 nm. The longest wavelengths, corresponding to the colour red, peak at approximately 60 nm. Both the red and green channels decrease rapidly after this first peak, and increase again and peak at around 125-160 nm (green) and 170-200 nm (red).

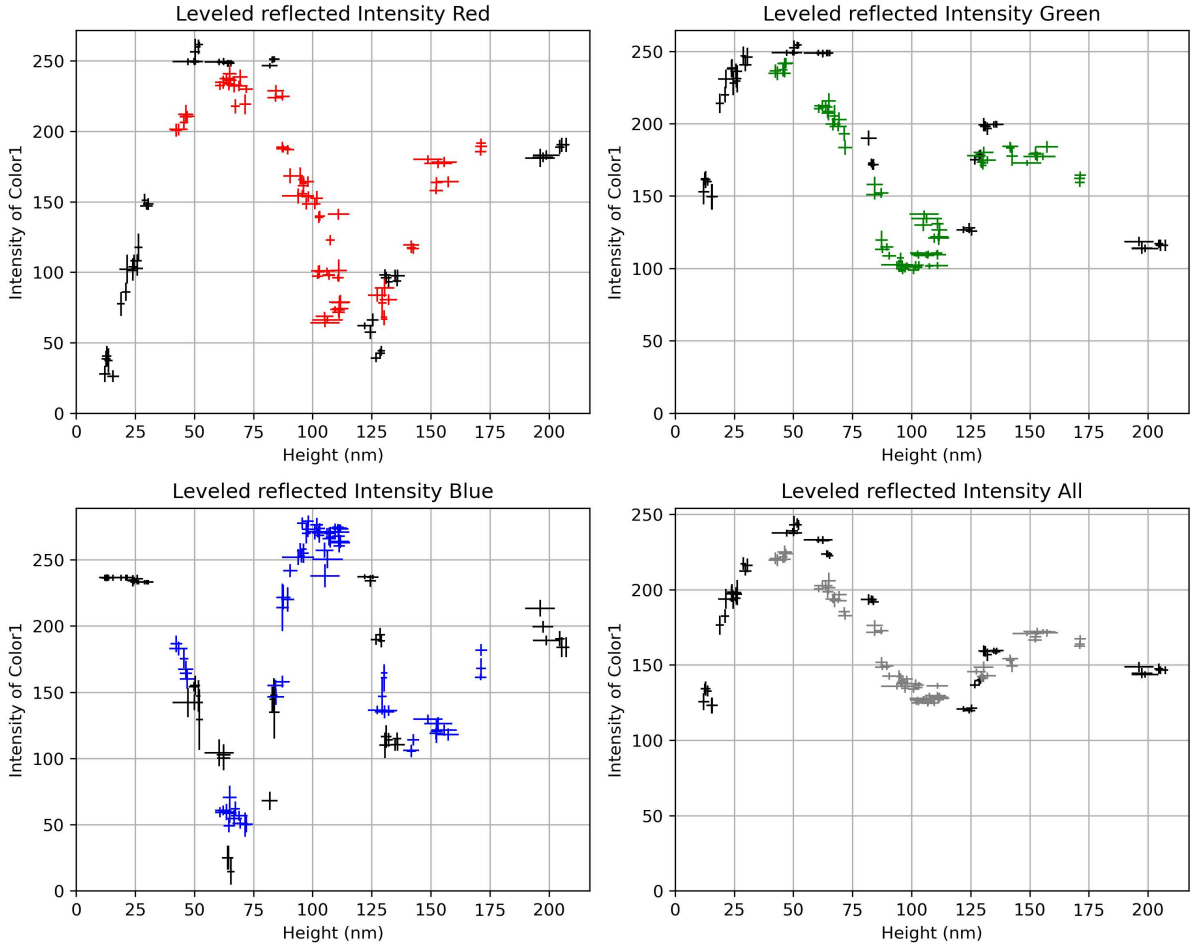


Figure 10: Measured RGB values $FePS_3$

5.1.2 Measured Contrast $FePS_3$

From the reflected intensity, the contrast with respect to the surrounding substrate has been calculated using the equation (61) and plotted in Figure 11. Contrast values have a benefit over the levelled reflected intensity: the systematic error due to non-recorded changes in the camera become less significant. Since the bias of the flake is similar to the one of the surrounding substrate, by calculating their differences, the bias becomes smaller. Therefore showing the

measured contrast facilitates replication of this study by decreasing the significance of the camera settings. The points were fitted using a damped sinusoidal function of the form

$$\text{Contrast } C = Ae^{-\lambda d}\cos(\omega d + \phi) + c, \text{ where } d = \text{thickness and there are 5 parameters,} \quad (62)$$

which yielded r^2 values above 0.83. They will be used in the next section.

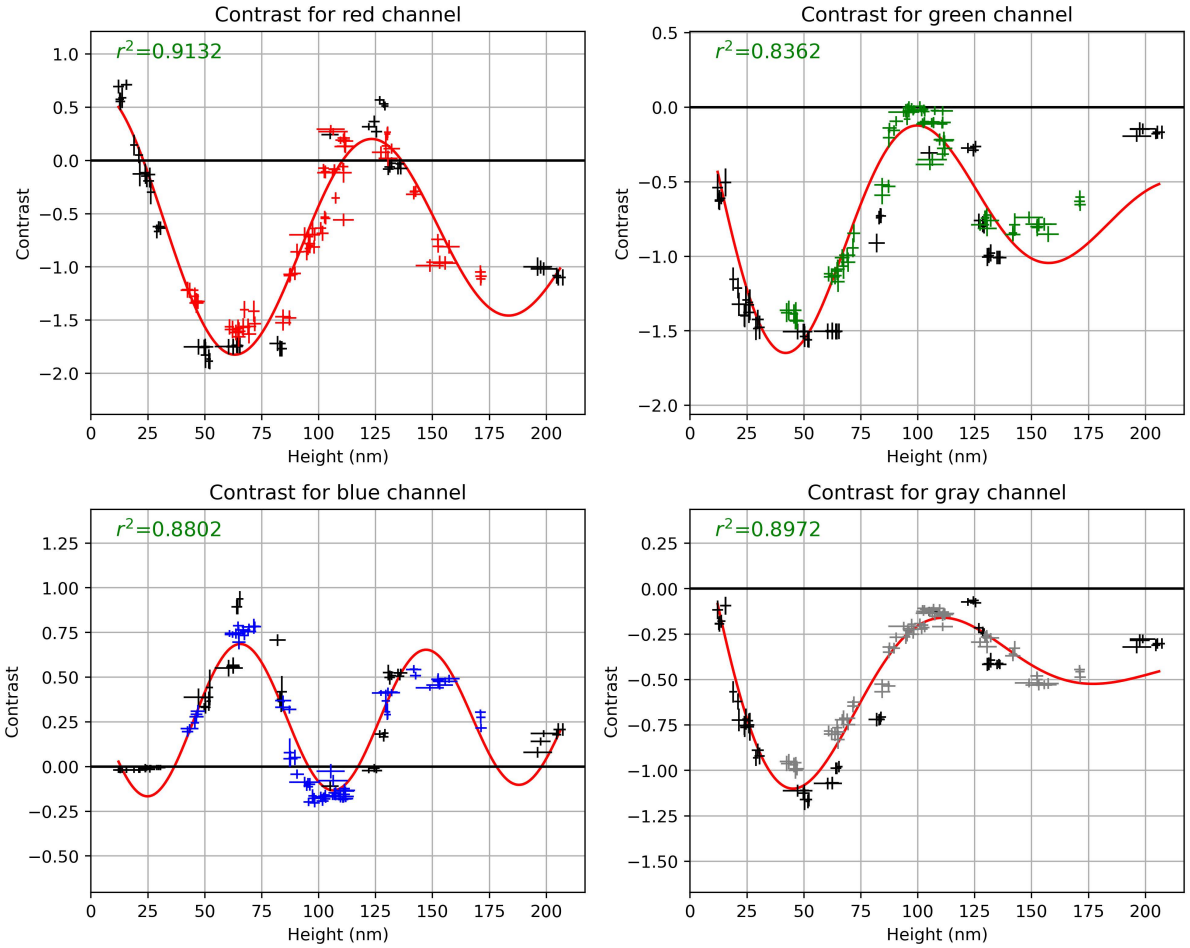


Figure 11: Measured contrasts $FePS_3$

$$\text{Red: } C = 1.378e^{-d/300.60}\cos(\frac{2\pi}{120.25}\cdot d - 0.222) - 0.711 \quad (63)$$

$$\text{Green: } C = 1.389e^{-d/114.70}\cos(\frac{2\pi}{115.51}\cdot d + 0.700) - 0.697 \quad (64)$$

$$\text{Blue: } C = -0.445e^{-d/1020.3}\cos(\frac{2\pi}{81.574}\cdot d + 4.359) + 0.268 \quad (65)$$

$$\text{All: } C = 1.358e^{-d/69.394}\cos(\frac{2\pi}{131.85}\cdot d + 0.694) - 0.422 \quad (66)$$

For each channel, the contrast values are interpreted as such: if $C = 0$, then the intensity value is the same as for the surrounding substrate ($I_{R_s} = I_R$). If $C = 1$, then $I_R = 0$; it follows that the intensity value is 0. If $C < 0$, the value of the flake is higher than the value of the

surrounding substrate. Overall, it follows from equation (61), that

$$I_R = (1 - C) \cdot I_{R_s} \tag{67}$$

In Figure 11, the local minima reflect the peaks from Figure 10. The fit-functions were differentiated w.r.t d to find minima and maxima; the values for d if $\frac{\mathrm{d}C}{\mathrm{d}d} = 0$ were computed using Desmos Scientific Calculator and they are shown in the Table 1.

Colour	**Contrast Maxima at d(nm)**	**Contrast Minima at d(nm)**
Red	3.03 , 123.3	63.2, 183.4
Green	99.7	42, 157.5
Blue	65.6, 147.2	24.8, 106.4, 188.0
Grey	111	45.2, 177.0

Table 1: Maxima and minima of measured contrast for $FePS_3$

5.1.3 Measured Colours $FePS_3$

From the fit-functions, which yield the contrast values, the measured colours were generated and plotted in Figure 12. Using the equation (67): $I_R = (1 - C) \cdot I_{R_s}$, where C is computed by the contrast fit-functions which depend on the thickness and the average measured substrate values I_{R_s} are $91, 99, 231$ for red, green and blue respectively.

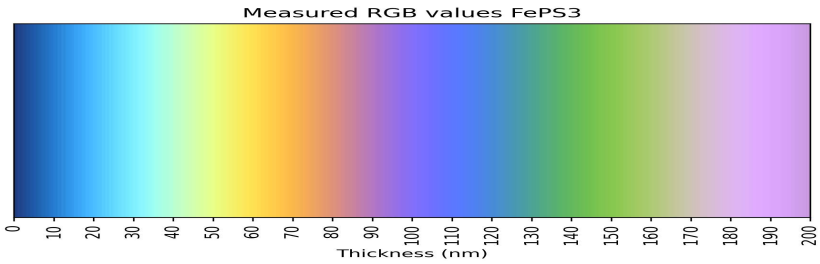


Figure 12: Measured colours $FePS_3$

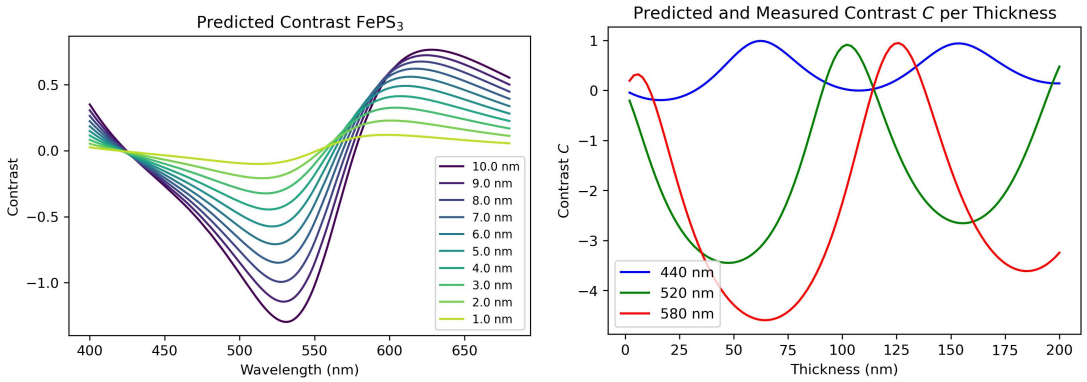
Note that due to this procedure, colours below 10 nm thickness might be misleading, as data for the fit-functions is lacking. For the other thicknesses, this graph indicates the measured colours of flakes; the very thin flakes are dark blue (note that this is a purer blue than around 110-120 nm). They then become lighter up until around 40 nm. Between 45 nm and 85 nm, the flakes are yellow and become gradually more orange. Between 80 nm and 100 nm, flakes may appear violet. Going on, the flakes become blue again until; this time a more greyish blue. From 130 nm onwards, green is the dominant colour, before turning pink again around 170 nm until 200 nm. Note that the only colours which are multiple times are blue (0-40 nm; 105-125 nm) and violet/pink (90-100 nm; 180-200 nm). The results are discussed in section 6.

5.2 Computing the Theoretical Model for $FePS_3$

From the equation (59) developed in the section 3, we generated various different kinds of plots to allow comparisons to the experimental data.

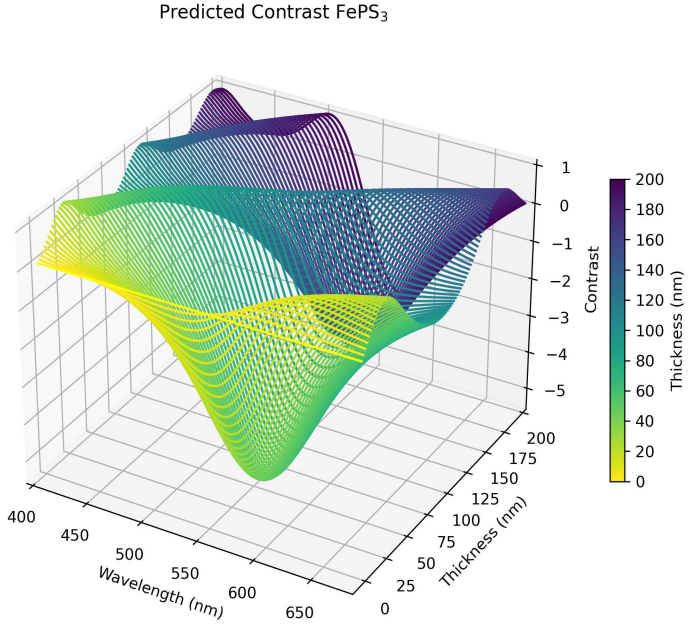
5.2.1 Predicted Contrast $FePS_3$

To allow comparison to the experimental data and related studies, we compute the contrast $C = \frac{R_s - R}{R_s}$ with respect to the surrounding substrate ($R_s = r_{tot} r^*_{tot}$, as defined in equation (55)). For thicknesses 1-10 nm, the results are shown in Figure 13a.



(a) Predicted contrasts $FePS_3$ (1-10 nm)

(b) Predicted contrast $FePS_3$ (0-200 nm)



(c) Predicted contrasts $FePS_3$ (0-200 nm; 101 lines)

Figure 13: Computing the predicted contrast

For the thinnest flakes it can be seen that the decrease in contrast at a wavelength of approximately 535 nm and the increase at 630 nm per nm are constant; the contrast changes linearly with the flake thickness. Around the blue wavelengths, the contrast is close to 0 and does not vary for thin flakes. Computations for thicknesses between 0 nm and 200 nm are shown in Figures 13b and 13c. Estimated minima and maxima are shown in Table 2. They were determined only

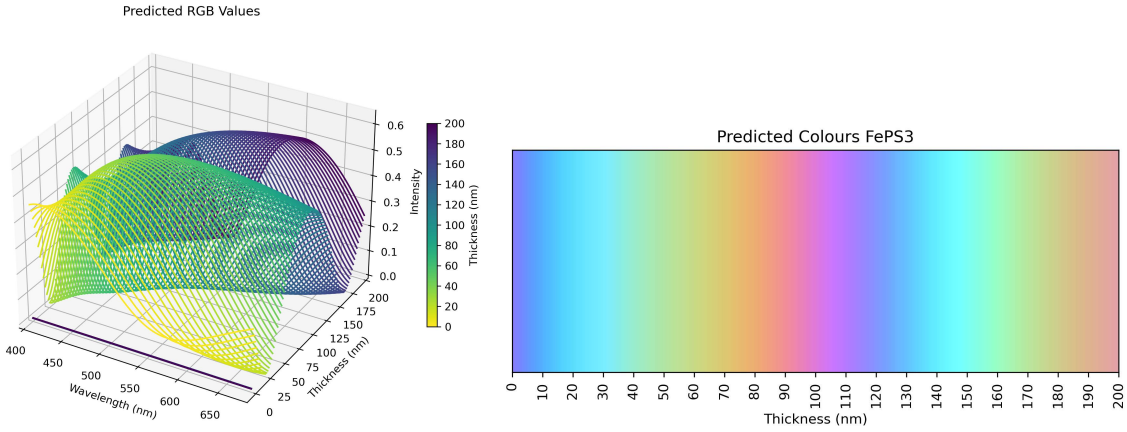
visually because of the complex construction of the computation.

Colour	**Contrast Maxima at d(nm)**	**Contrast Minima at d(nm)**
Red	10, 125	65, 185
Green	105	50, 155
Blue	60, 155	20, 105, 195

Table 2: Maxima and minima of predicted contrast for $FePS_3$

5.2.2 Predicted Colours $FePS_3$

As with the measured results, a plot of the predicted colours per thickness has been generated. To facilitate the computation, the colours have been generated directly from the values for R rather than C. The wavelengths were first converted to RGB values. Because they do not translate 1:1, an approximation was used (see code in Appendix). Subsequently, for every wavelength, this combination of red, green and blue values was multiplied by its corresponding value of R. These products were added together for all wavelengths and divided by the sum of all R (one per wavelength) for each channel separately. To adjust the spectra as emitted by the microscope light and as detected by the camera sensor, to the spectra of the theory model (which is based on equal intensities across all wavelengths), and to the spectral sensitivity of the human eye, the values were slightly calibrated: as a result, the values of the blue channel were doubled and additionally all values were multiplied by 1.3 to achieve a proper brightness.



(a) Predicted RGB intensity $FePS3_3$ (b) Predicted colours $FePS_3$ (0-200 nm)

Figure 14: Predicted colours and reflected intensity for $FePS_3$

Once again, we observe a wide range of colours; they start rather dark, something in between violet and blue. Up until 35 nm thickness, light blue is strongly saturated. The blue value decreases whilst red colours become more intense until 90 nm. Between 100 nm and 110 nm the colour is pink. something like greyish blue starts at 115 nm and turns into light blue until 160 nm, where some green shows, before from 180 nm until 200 nm turning orange and red again.

5.3 Experimental Data $NbSe_2$

In this subchapter, the experimental results for the reflected intensity, contrast and colour per thickness of $NbSe_2$ are presented. For this material, samples were collected only up to 100 nm, all on the same day. The optical and AFM images can be found in the appendix. The order in which the data is presented is the same as for $FePS_3$.

5.3.1 Reflected Intensity $NbSe_2$

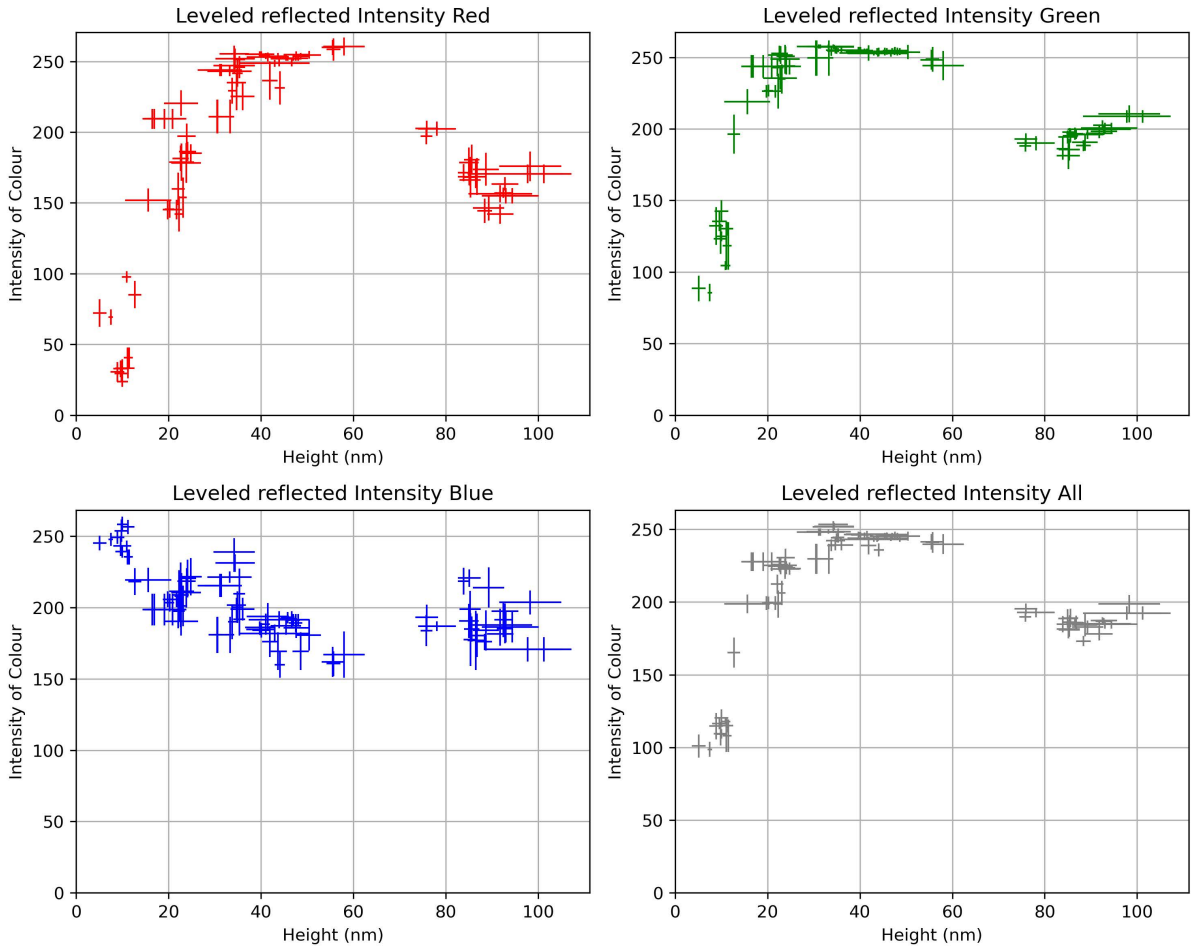


Figure 15: Reflected intensity $NbSe_2$

Figure 15 shows the reflected light intensity per thickness split up into the colour channels. Blue colours show the highest reflected intensity for thicknesses around 10 nm. Until 60 nm the values decrease by 100. There is a cloud of flakes with thickness between 70 nm and 105 nm, ranging from 170 up to 230. Green wavelengths are reflected the most between 20 nm and 50 nm. From 0 nm to 20 nm, there is a rapid increase from 80 to 255; after 50 nm, there is a gap, but there is some decrease: at 75 nm, the values are around 190, and increasing to 210 at 100 nm. The red channel peaks between 30 nm and 60 nm. There is no evidence whether it goes on to higher thicknesses or whether the values decrease already at 60 nm; for sure they are between 60-110 lower for 75-110 nm thick flakes. Here again, there is a rapid increase between 10 nm and 30 nm, extending over the whole range.

5.3.2 Measured Contrast $NbSe_2$

For $NbSe_2$, the contrast has been computed in the same way, according to equation (61). The results are displayed in Figure 16. Like the $FePS_3$ contrast values, have been fitted using a damped sinusoidal function in the same form as equation (62). The equations are

$$\text{Red: } C = 2.577e^{-d/44.02}\cos\left(\frac{2\pi}{128.98}\cdot d + 0.521\right) - 0.741 \tag{68}$$

$$\text{Green: } C = 3.466e^{-d/23.85}\cos\left(\frac{2\pi}{138.07}\cdot d + 0.844\right) - 0.911 \tag{69}$$

$$\text{Blue: } C = -3.161e^{-d/40.15}\cos\left(\frac{2\pi}{1578.43}\cdot d + 1.470\right) + 0.124 \tag{70}$$

$$\text{Greyscale: } C = 2.175e^{-d/30.082}\cos\left(\frac{2\pi}{132.43}\cdot d + 0.735\right) - 0.618 \tag{71}$$

The r^2 values have been computed for these functions and are shown in the graph.

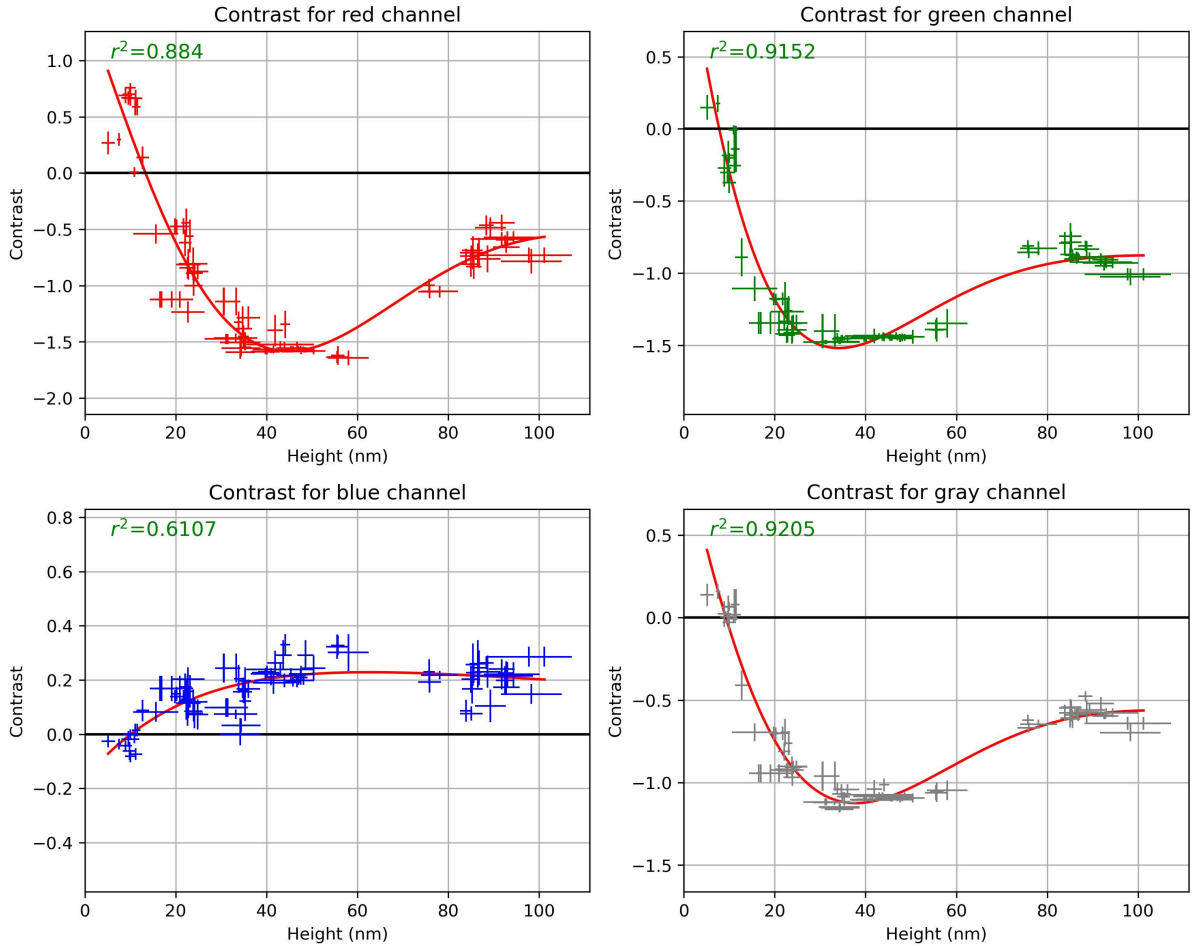


Figure 16: Measured contrasts $NbSe_2$

The data behaves analogously to the data described in section 5.3.1. The maxima and minima obtained from the fit-functions are for the red channel: minimum at 44.9 nm; green channel: minimum at 34.1 nm; blue channel: maximum at 65.1 nm. Note that the r^2 value for the blue channel is relatively low; for the others, the r^2 value is above 0.88.

5.3.3 Measured Colours $NbSe_2$

From the contrast fit-functions, the measured colours have been reconstructed and displayed in Figure 17. Again, the equation for the reflected intensity $I_R = (1 - C)I_{R_s}$ was used, where I_{R_s} denotes the measured substrate value for each channel. The thinnest flakes are blue; they then start to turn green at 10 nm, reaching a pale yellowish at 35 nm to 50 nm thickness. There is a smooth transition to a greyish green between 55 nm and 100 nm thickness.

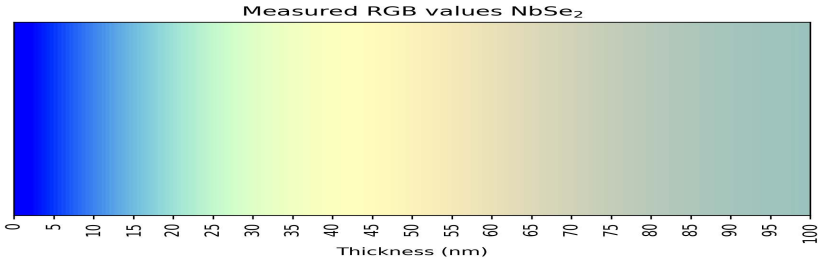
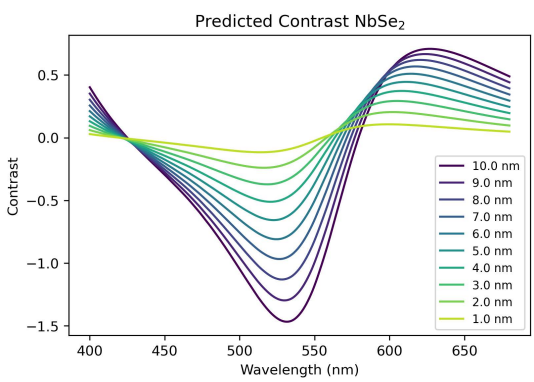


Figure 17: Measured colours $NbSe_3$

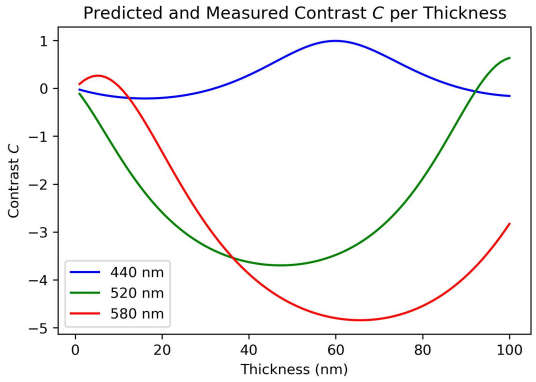
5.4 Computing the Theoretical Model for $NbSe_2$

The theory model for $NbSe_2$ has been computed in the same way as for $FePS_3$, adapting the refractive index (taken from `https://refractiveindex.info/`). Since the experimental data only goes up to 100 nm thickness, the computations were done for the same range.

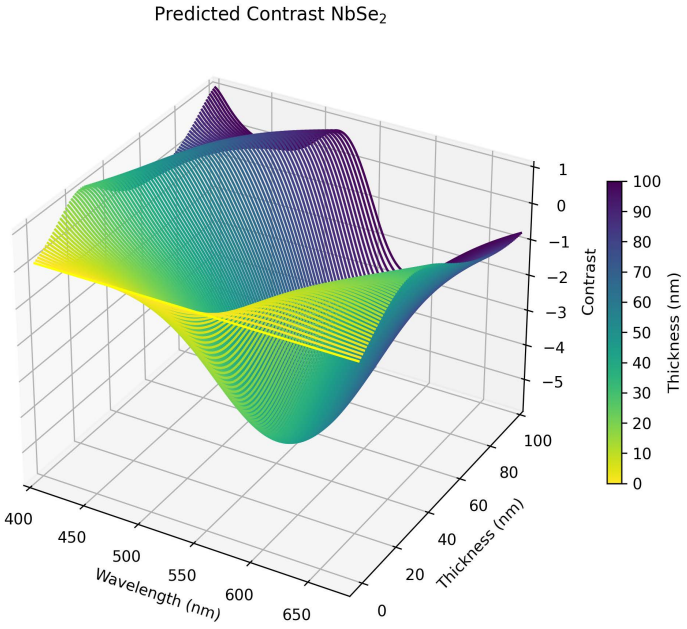
5.4.1 Predicted Contrast $NbSe_2$



(a) Predicted contrasts $NbSe_2$ (1-10 nm)



(b) Predicted contrast $NbSe_2$ (0-100 nm)



(c) Predicted contrasts $NbSe_2$ (0-100 nm; 101 lines)

Figure 18: Computing the predicted contrast

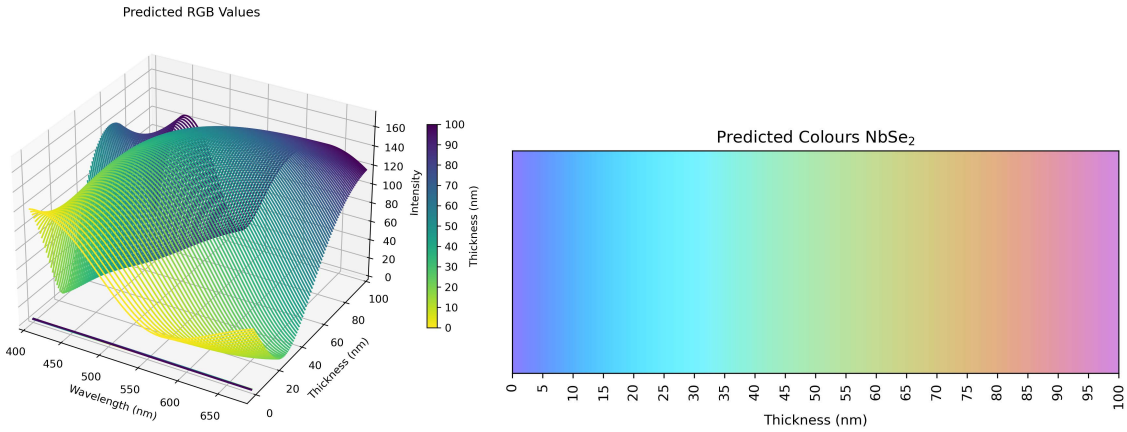
Figure 18 shows the results of the computations for contrast; once for the very thin flakes (Figure 18a), once with respect to the thickness of the flakes (Figure 18b), and finally once in 3 dimensions to show the evolution of the Contrast over thickness per wavelength (Figure 18c). Note: values close to 1 indicate low intensity, contrast minima correspond to peaks in light intensity.

Again, there is a linear decrease (increase respectively) in contrast for the very thin flakes, especially for wavelengths around 530 nm and 650 nm. In Figure 18b, the green wavelength

begins with exactly this linearly decreasing contrast up to around 20 nm, reaching a minimum of -3.6 just below 50 nm, then again increasing until 100 nm, even going above 0. The blue colours are strong until approximately 35 nm, then the contrast peaks at 60 nm, before decreasing again. Red colours are predicted to start with low intensity (0-10 nm), then increasing (again, almost linear between 20 nm and 35 nm), peaking at 65 nm thickness and decreasing again. Figure 18c completes Figure 18b by including the whole wavelength spectrum from 400 nm to 680 nm. Note that the theory model starts with 0 contrast at thickness 0 nm.

5.4.2 Predicted Colours and Reflected Intensity $NbSe_2$

From the predicted reflected intensity (Figure 19a, the predicted colours have been obtained and are shown in Figure 19b. From a slight purple, the colour develops to a light blue within the first 35 nm. It then turns gradually into orange at 70-80 nm, via green and yellow. Further, the colour develops to red and pink between 90 nm and 100 nm.



(a) Predicted RGB intensity $NbSe_2$ (b) Predicted colours $NbSe_2$ (0-100 nm)

Figure 19: Predicted colours and reflected intensity for $NbSe_2$

6 Discussion

6.1 Analyzing Results of $FePS_3$

In section 1, the research question was stated: "How does the optical contrast and the colour of a three-layer system, consisting of thin flakes of the insulator $FePS_3$ and of the metal $NbSe_2$ on SiO_2/Si substrate, change, depending on the flake thickness?" The goal of this chapter is to give an answer to the research question using the results as presented in section 5. Additionally, the results are compared to literature, and the data is tested for utility in an applications.

6.1.1 Verification of Theoretical Model

In section 3, we have derived the following equation for the reflected intensity R, depending on $\Delta\phi_i = \frac{2\pi n_i d_i}{\lambda}$ and $r_{i,i+1} = \frac{n_i - n_{i+1}}{n_i + n_{i+1}}$.

$$R = r_{tot} r_{tot}^* \text{ where } r_{tot} = \frac{r_{01} + r_{12}e^{-i2\Delta\phi_1} + r_{23}e^{-i2(\Delta\phi_1+\Delta\phi_2)} + r_{01}r_{12}r_{23}e^{-i2\Delta\phi_2}}{1 + r_{01}r_{12}e^{-i2\Delta\phi_1} + r_{01}r_{23}e^{-i2(\Delta\phi_1+\Delta\phi_2)} + r_{12}r_{23}e^{-i2\Delta\phi_2}}$$

We further defined the contrast $C = \frac{R_s - R}{R_s} = \frac{I_{R_s} - I_R}{I_{R_s}}$. Both values and the resulting colours were computed and displayed in appropriate graphs in section 5.

Colours

The colour stripes displayed in Figure 20 provide a way of comparing the results of the experiment and the model, which is not very precise, but they are very intuitive and aid understanding the subsequent Figures. In Figure 20, the blue channel of the theoretical model has been doubled and all channels have been amplified by a factor of 1.3 due to reasons explained in section 5.

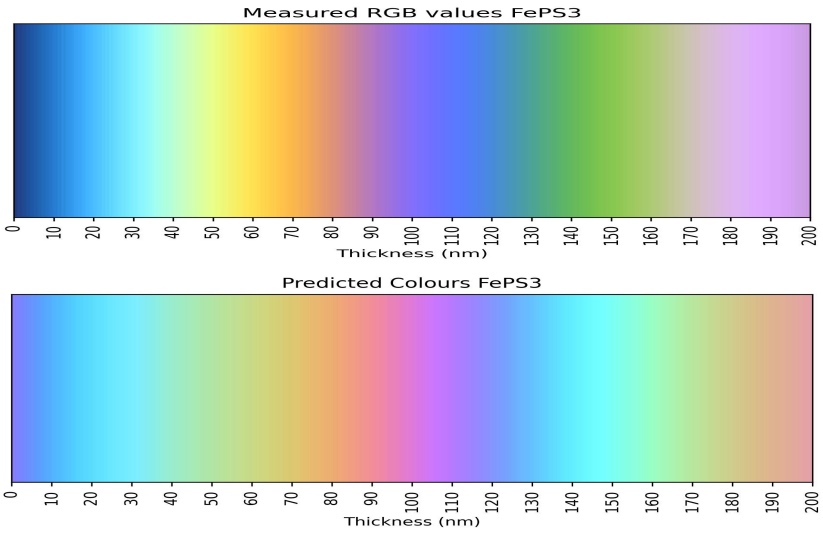


Figure 20: Measured and predicted colours per thickness for $FePS_3$

The key feature to notice on this graph is the thicknesses, to which a change in colour corresponds. These are the same for both measured and predicted colours. Around 10 nm, 45 nm, 125 nm, 165 nm, there are significant changes of colour which appear in both colour stripes. To a high degree, the colours are also similar; starting from a light blue between 10 nm, then turning yellow/redish around 45-80 nm. However, for bigger thicknesses, the colours start to deviate. They remain fairly similar until 125 nm, but then the prediction yields a light blue, whereas green was measured.
The comparison drawn from Figure 20 suggests that the theoretical model gives first predictions of which colours to expect, but they are not very accurate, especially from 125 nm onwards. However, there is a great positive message: The 'Measured RGB values $FePS_3$' are very diverse over thicknesses from 0 nm to 200 nm. Given a sample with flakes of many different heights, one should therefore be able to distinguish flakes of different heights quite accurately on the basis of our measured results (see section 6.1.3).

Contrast

Figure 21 allows a more accurate comparison between the measured and the predicted contrast values. 3 wavelengths (440 nm, 520 nm, 580 nm) have been selected from the continuous spectrum given by the theoretical model to represent the blue, green, and red colour channels of the camera. The wavelengths have been chosen according to the peak sensitivity of cone cells (which are responsible for colour detection) in the eye [6]. In this Figure, the blue channel has not been doubled, and no brightness correction was applied.

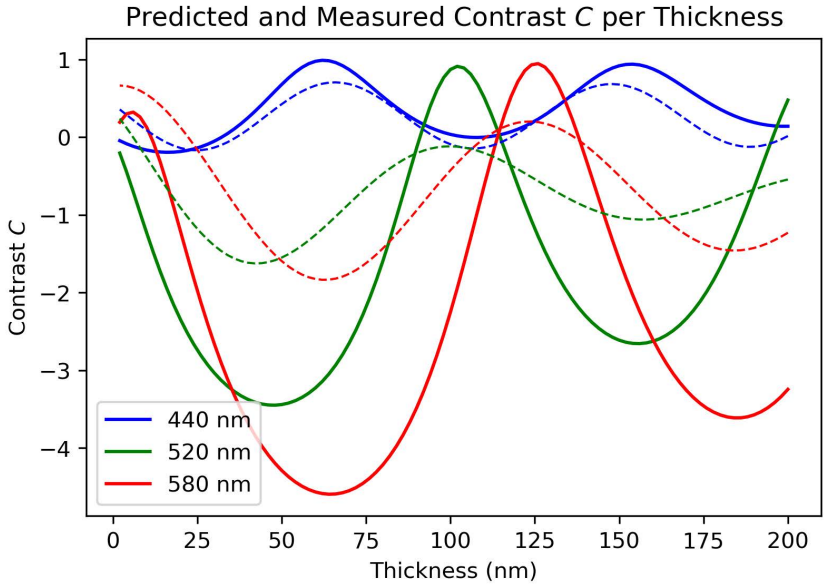


Figure 21: Measured and predicted contrast per thickness for $FePS_3$ (solid line: predicted; dashed line: measured)

At first glance, the difference in amplitude is striking, especially for the green and red channel. However, this is easily adjusted by the brightness calibration. What is far more interesting is the comparison between the thicknesses, at which the predicted and measured contrast values reach peaks or minima, and the period at which they are occurring: e.g. the minima just

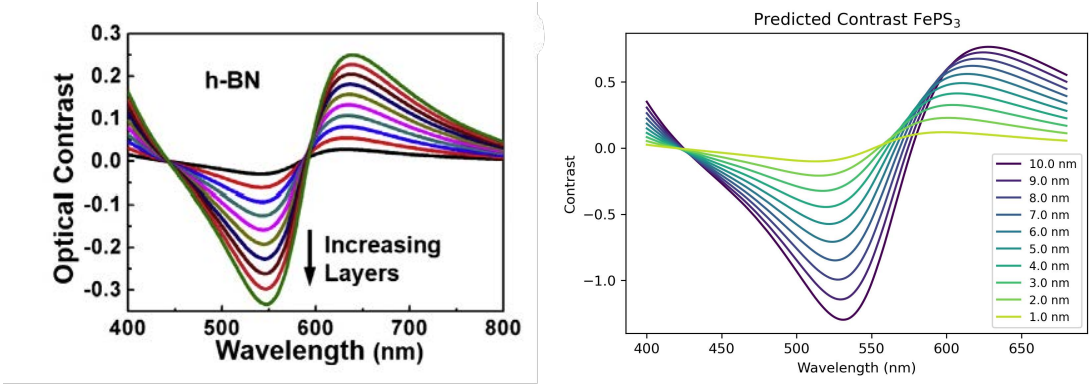
above 100 nm of the blue contrast channel coincides very nicely; however, because the period of the measured contrasts is smaller, the maxima at around 60 nm and 150 nm do not coincide anymore. Similarly for the green channel: the minimum at around 160 nm coincides, but the maximum and the preceding minimum of the predicted and measured contrasts do not coincide exactly. Only for the red wavelengths, the minima coincide quite precisely, but the maximum is shifted a tiny bit. Therefore this graph provides evidence that the theoretical model is valid only for a certain range of thicknesses, during which the peaks and troughs align. Note that the minima and maxima of the theoretical model highly depend on the extracted wavelengths. Varying them a tiny bit (around 5 nm) already results in a relatively big shift of the maxima and minima. Since the period of the measured and predicted functions is not exactly the same, using slight shifts can be used to configure the theoretical model such that it approximates the contrast for the thickness of interest very well.

6.1.2 Literature Comparison

In order to gain significance, this study should be supported by previous findings by other research groups, with the ultimate goal of extending their findings. B. Ma et al. [11] used a very similar method of determining the thickness of 2D materials by optical contrast for different materials (Graphene, hexagonal boron nitride (h-BN) and molybdenum disulfide (MoS_2)). They focused on very thin flakes, ranging from one to ten layers of atoms. Their theoretical prediction for h-BN was based on the following equation:

$$\boxed{R = \left(\frac{r_{01}e^{i(\Delta\phi_1+\Delta\phi_2)} + r_{12}e^{-i(\Delta\phi_1-\Delta\phi_2)} + r_{23}e^{-i(\Delta\phi_1+\Delta\phi_2)} + r_{01}r_{12}r_{23}e^{i(\Delta\phi_1-\Delta\phi_2)}}{e^{i(\Delta\phi_1+\Delta\phi_2)} + r_{01}r_{12}e^{-i(\Delta\phi_1-\Delta\phi_2)} + r_{01}r_{23}e^{-i(\Delta\phi_1+\Delta\phi_2)} + r_{12}r_{23}e^{i(\Delta\phi_1-\Delta\phi_2)}} \right)^2} \quad (72)$$

This equation is very similar to the one we developed, with some slight twists. Moreover, the SiO_2 layer in the SiO_2/Si substrate is 300 nm thick. In their paper, they show Figure 22a - the results of computing their equation for 10 h-BN layers (thickness per layer = 0.333 nm [11]). Figure 22 allows a comparison of their graph to the results of our investigation. Both graphs show a minimum and a maximum - the ones of h-BN occuring at 30-40 nm shorter wavelengths. This makes sense: h-BN has a refractive index slightly lower than the one of $FePS_3$. Light can therefore travel faster through h-BN. Hence the phase shift at the same flake thickness is less for h-BN than for $FePS_3$. If the phase shift is lower, shorter wavelengths are amplified/canceled out, since after the time interval, at which the ray, which went through the material, returns back to the surface, the ray being directly reflected has had less time to 'develop' (develop: imagine a dot travelling along a sine curve). For shorter wavelengths, this time interval is enough to 'develop' over one/multiple whole wavelength(s), which causes them to be amplified. For longer wavelengths, the duration of the interval is not enough to 'develop' over a whole wavelength, and therefore they are amplified less or even canceled (as long as the frequency/energy of the photons remains constant). The increased thickness of the SiO_2/Si layer counteracts this effect, but the ratio between the refractive indices dominates the ratio between the thicknesses within the substrate.



(a) Theoretical model by B. Ma et al. [11]

(b) Predicted contrasts $FePS_3$

Figure 22: Literature comparison

More importantly, both models predict a linear de-/increase of contrast for very thin flakes. This is very useful for applications, because it allows precise identification of thickness. But: *"Note that the largest layer numbers of [...] h-BN that can be identified through this method is [...] 10, respectively, since the changes of the image contrasts are less and less obvious with further increasing the layer number."* [11] B. Ma et al. faced problems: after a few layers (in their case 10 layers $\cong$ 3 nm), the linearity is not present anymore. We are lucky: for $FePS_3$ with a 285 nm layer of SiO_2, the linearity continues on for at least 30 nm - allowing for a slight bend in the curve, we can even get to around 45 nm, each layer having its own specific contrast. As described above, the theoretical model can be customised such that it fits the very thin flakes nicely.

6.1.3 Data Patterns for Application

As stated above, the theoretical model's precision is debatable concerning a broad spectrum of thicknesses. If an overview over a big range of thicknesses is needed, experimental data is preferable. Such an overview is needed for the core of the application: entering RGB values and returning the height. For this application, a map of thickness to a unique RGB value is needed. The value of one wavelength or colour itself is not enough: Figure 23 shows the thicknesses at which, e.g. for green colours, the RGB values computed by the fit-functions are similar and where they are distinguishable. Two distinct intensity values for a colour are separated by at least a difference of 10. This critical dimension is referred to as 'error factor'. It is chosen in such a way, that if the difference between two intensity values is bigger than the error factor, a confusion should not happen. For a difference smaller than the error factor, the probability of a confusion rises. This probability is indicated as a change in colour: a blue pixel indicates a very low chance of confusion; a yellow pixel indicates that a confusion is very probable. The error factor has been set to 10 because the standard deviation during the colour measurements were almost exclusively below 10.
Surely, there is a line going from the bottom left to the top right in each plot, since the same thickness is associated with a single RGB value (the map is bijective). The other yellow or orange lines in the red, green and blue channel graph indicate there are confuseable thicknesses, if only one colour is used. Hence the contrast alone is not enough to exactly determine the thickness.

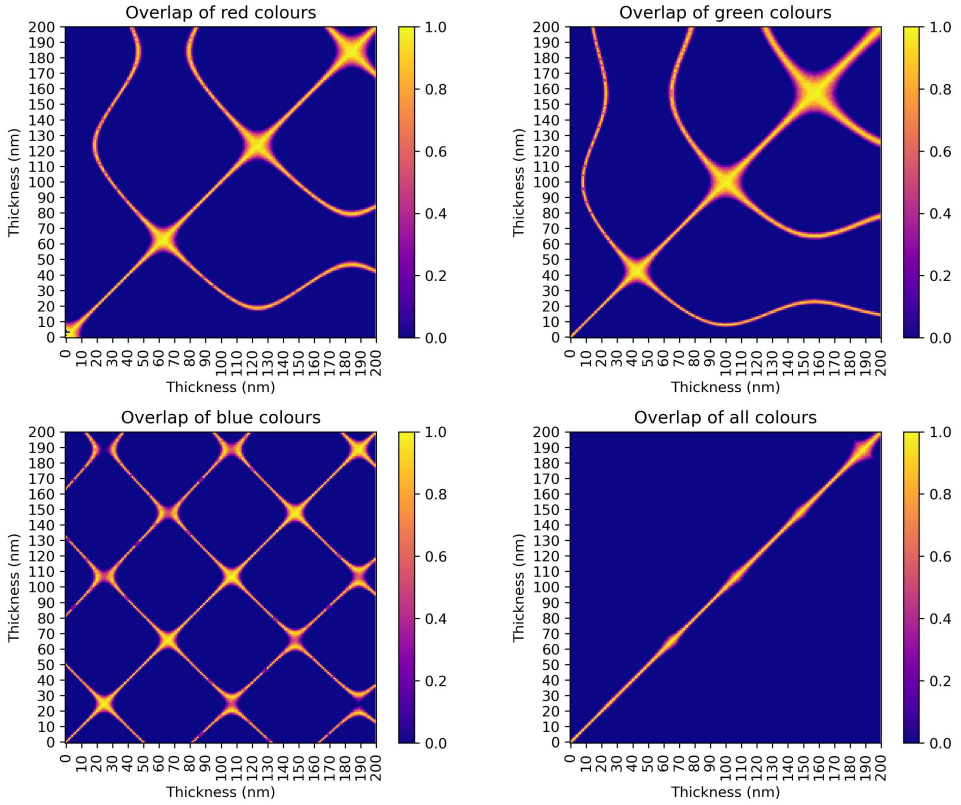


Figure 23: Confuseable thicknesses $FePS_3$

However, using the average of the differences between two RGB values, which correspond to to two thicknesses, for each colour channel, the probability for a confusion of any thickness with another is very low, as displayed in the bottom right graph: Because there are only blue pixels except for a thin diagonal, the RGB values for every thickness are unique and distinct. The graph therefore suggests, that confusions can be avoided for thicknesses up to 200 nm using the average difference between the RGB values for each colour channel.

6.2 Analyzing Results of $NbSe_2$ and Literature Comparison

The experiment described in section 4 was repeated for $NbSe_2$ to further examine our assumptions: Is there significant difference, if the top layer of our structure is a metal instead of an insulator? The following comparison between experimental and theoretical results suggests an answer.

6.2.1 Verification of Theoretical Model

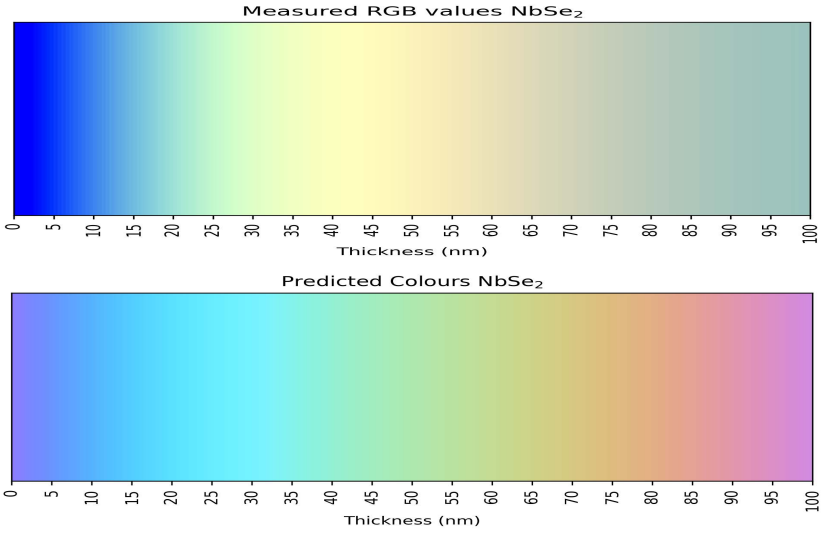


Figure 24: Colour comparison

The measured and predicted colours are shown in Figure 24. There is a strong difference between them. Although both start blueish (a very strong blue was measured, more purple was predicted), the light blue ends 20 nm before it was predicted to end. Instead, a slow change to a pale yellow occurs. This colour never even appears in the predicted colours - they continue from light blue to greenish to orange, red and pink. The key difference to the results of $FePS_3$ is that the colour changes do not occur at the same thicknesses. Therefore the divergence between the predicted and the measured colours is not due to differences in the light intensity spectrum.
Plotting the predicted and the measured contrast per thickness for certain wavelengths (corresponding to the colour channels) on the same axes, as shown in Figure 25, reveals the error again: While the predicted contrast for the short wavelength changes strongly, the measured contrast remains almost constant. The predicted and measured contrasts for the green and red wavelengths/channels are relatively similar up to about 40 nm, then they diverge.
Note that the amplitude difference is probably due to spectral differences. The distribution of contrast over thickness is not. Interestingly, these contrasts from the green and red channels have a y-intercept greater than one, corresponding to an intensity lower than 0. One would expect a contrast of 0 at a thickness of 0 nm, corresponding to the colour of the substrate, assuming that the transparency of the flakes increases. There is a chance, that this is a measurement error due to some edge/contrast enhancement of the camera used to take the images, especially since thin flakes are often very small, even though attention was paid to get as raw images as possible; As no flakes below 5 nm were found, this remains an open question for further investigations.

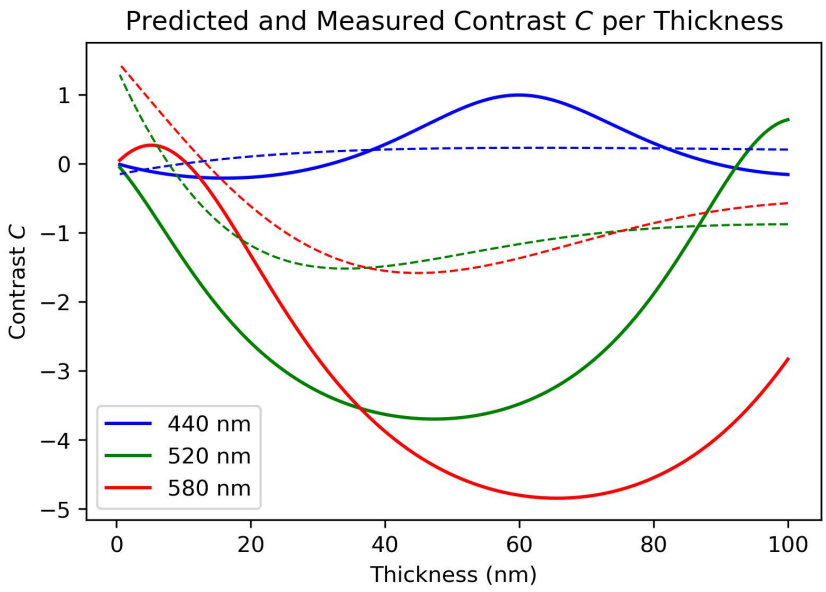


Figure 25: Measured and predicted contrast per thickness for $NbSe_2$ (solid line: prediction, dashed line: measured)

The fact that $NbSe_2$ does not behave as predicted by their theoretical model has also been found by M. M. Benameur et al. [2]. They compared experimental results to the theoretical model described by equation (72) (section 6.1.1). Therefore the theoretical model presented in our report does not contradict this previous finding.

6.2.2 Data Patterns for Application

Even though there is a difference between the theoretical model and the experimental findings, researchers are still interested in $NbSe_2$ for many other purposes. Whether or not the thickness $NbSe_2$ flakes can be determined by optical contrast has not been clarified in previous sections.

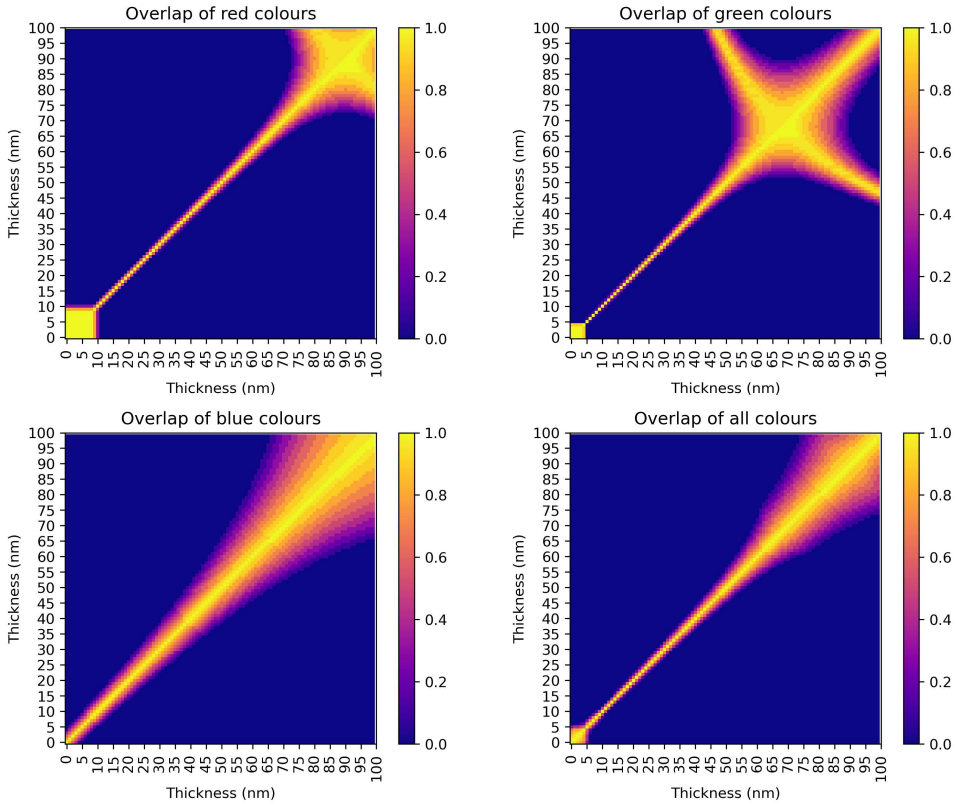


Figure 26: Indistinguishable thicknesses $NbSe_2$

Figure 26 shows the absolute value of the difference in reflected intensity between any two thicknesses and is based on our contrast fits. If two values are the same, they appear as yellow. The bigger the difference, the darker. If the difference is greater than 10 and smaller than -10, it is coloured dark blue. This cutoff has been set, since the standard deviation of the colour measurements was always less than 10. The difference, which codes for the colour in the bottom right graph, is the average value of the differences for each channel. Taking the average reduces the amount of confuseable thicknesses. However, from about 60 nm thickness upwards, the diagonal starts to diverge. This implies, that the RGB values for thicker flakes are not unique and there is a high chance of confusion between different thicknesses. Therefore the method of thickness identification by optical imaging can only work well for thin metallic 2D materials.

6.3 Error Analysis

Throughout any experiment, small errors and uncertainties appear. The goal of this section is to analyse the reliability of our measurements. There are three main kinds of errors, systematic errors, measurement uncertainties, and procedural errors:

6.3.1 Systematic Errors

Consistent biases, affecting all measurements of a kind, are called systematic errors. The main such error, which occurred during our experiment, were the factors, which changed the brightness of the microscope camera. Attention was paid to the settings of the camera, but the brightness changed nevertheless - these factors were beyond our control. Calculating the contrast with respect to the substrate brightness allowed a slight correction; Further studies should take care to control even more factors to create results, with are replicable on their microscope. Another systematic issue is caused by the differences in emitted and recorded light spectra. The theoretical model yields the ratio of reflected over incoming intensity. Light sources, such as the one in the microscope, do not emit the same intensity over all wavelengths - nor are cameras or eyes equally sensitive over the whole visible spectrum. By doubling the blue channel, and amplifying all channels by a factor of 1.3, this issue was more or less fixed - future studies could analyse the spectra and model them, such that they can be implemented into the theoretical model. Since this is a very tedious job and, because of its high specificity (only for one kind of microscope), it is perhaps more efficient to create a sample data set and customized contrast fit-functions. The same holds for colour distortion by the lens - in the case of our microscope, the 100X magnification lens was not used, because there was a significant magenta shift compared to the other lenses. It has to be expected that every lens has its own permeability spectrum; and thus increasing the advantage of a sample data set.

6.3.2 Measurement Uncertainty - Gaussian Error Propagation

Every measurement is subject to small errors, due to the finite precision of measuring instruments. For the height measurements with the AFM, the way in which errors add up has been examined properly and is implemented into Gwyddion. From the height data, the error of fits are computed automatically. For the measurements of contrast, the estimation of the error is more complex. First, the values from 25 pixels I_P were averaged to one RGB value I_R. The standard deviation of this RGB value is denoted as

$$\sigma_R = \frac{\sum (I_R - I_P)^2}{25}. \tag{73}$$

In a second step, a Gaussian Error Propagation was derived for the equation for the contrast C. Hence, the error for contrast is defined as

$$\sigma_C = \sqrt{\sigma_{S_{av}}^2 \left(\frac{\partial C}{\partial S_{av}}\right)^2 + \sigma_R^2 \left(\frac{\partial C}{\partial I_R}\right)^2} \qquad \text{where } C = \frac{S_{av} - I_R}{S_{av}} \text{ and } S_{av} = I_{R_s}. \tag{74}$$

Inserting the partial derivatives yields:

$$\sigma_C = \sqrt{\sigma_{S_{av}}^2 \left(-\frac{I_R}{S_{av}^2}\right)^2 + \sigma_R^2 \left(\frac{1}{S_{av}}\right)^2} = \sqrt{\frac{\sigma_{S_{av}}^2 I_R^2}{S_{av}^4} + \frac{\sigma_R^2}{S_{av}^2}} \tag{75}$$

This value was computed for every sample using Microsoft Excel and they are shown as the vertical bars in Figure 11 and 16.

6.3.3 Procedural Errors and Improvements

If this study shall be replicated or extended, considering the following points can enhance precision and efficiency of the data collection. Firstly, the number of folds of the blue tape strongly influences the size and thicknesses of the flakes. Preliminary experiments should be conducted to find out the optimal number. Secondly, further studies should aim for an overall low image brightness to prevent plateauing of the RGB values at 255. Therefore a camera with a high dynamic range and a better microscope are beneficial. Finally, a high AFM resolution values a lot and is worth the time: The higher the resolution, the lower the noise for thin flakes.

6.4 Significance

All in all, there have been three main insights gained by the experiment and the theoretical model. Firstly, the contrast of $FePS_3$ and $NbSe_2$ on SiO_2/Si substrate changes strongly, depending on the flake thickness. The results in section 5.1 and 5.3 describe, how the dependence looks like and thus answering the research question. Secondly, the theoretical model describes the colour and contrast of the insulator ($FePS_3$) well, particularly for small ranges of thickness. However, it is not applicable to the metal $NbSe_2$. This confirms that the assumption for the theoretical model, the material being insulating, plays a significant role in the interaction between light and matter. Finally, it has been shown for $FePS_3$ and for $NbSe_2$, that the combination of the red, green and blue channels suffices to create a specific map up to 200 nm and 60 nm respectively, assigning each thickness a unique colour.
With these results, the feasibility of the method to determine the thickness of $FePS_3$ and $NbSe_2$ flakes using optical imaging is demonstrated. Because of the differences in light spectra, lenses and camera sensor, calibration is necessary. Our fit-equations for the contrast C could be copied and multiply them by self-measured substrate values I_{R_s} in the equation for the reflected intensity $I_R = (1 - C)I_{R_s}$ for each channel. For more precise results, doing sample measurements is recommended. Since our results strongly suggest a sinusoidal behaviour, a few sample measurements are enough to create accurate fitting functions. For researchers, this helps to speed up the thickness determination of such flakes enormously and reduce the costs of a lab, since expensive instruments as the AFM are required only for calibration.
However, the determination of thickness by optical imaging by eye is not as precise as an AFM (or similar) measurement. Therefore in the next section, an application is created and its accuracy for $FePS_3$ is tested. If the accuracy were too low, it can still be used to identify flakes of a range of thicknesses - which reduces the number of exact measurements and thus the time required.

7 Application and further Ideas

7.1 Production of 2D materials

Current research is very interested in 2D materials because of their extraordinary properties. To conduct experiments with 2D materials, their thickness must be known, as it has a strong influence on the properties [3, 11]. The method of exfoliation used in this report is quick and simple, but it creates small flakes of various thicknesses. Measuring the thickness using expensive devices such as an AFM takes a lot of time because of the low output rate. The approach to determine the thickness of $FePS_3$ and $NbSe_2$ by optical imaging, which is discussed in this report, is much faster. For example, an AFM needs 8.5 min to scan a 30 µm × 30 µm square in good quality. With a camera and an optical microscope, a single picture can be used to estimate the thickness of hundreds of flakes (depending on the resolution of the camera and magnification of the microscope) within a few seconds. This approach has the potential to facilitate and speed up the production of 2D materials for research. This goal can be achieved through comparing the colours from the images to a reference colour stripe by hand. The aim of this section is to develop a more precise application, written on Python, which can convert RGB values into an estimate of thickness and assert its reliability for $FePS_3$.

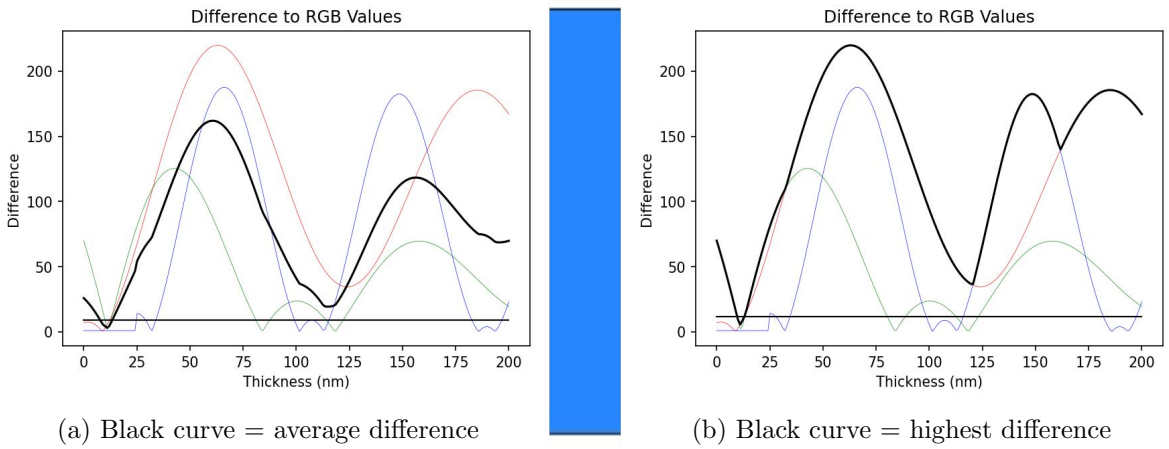
7.2 Working Principle of the Application

The purpose of the application is to find the corresponding thickness to an input RGB value. Firstly, reference RGB values are calculated based on some reference function, relating thickness to contrast for the red, green and blue colour channel (e.g. the fitted sinusoidal functions from above), for a certain range of thicknesses. For tests later on, this range has been set corresponding to the boundaries of the fitting functions (0-200 nm for $FePs_3$). Again, the equation $I_R = (1-C)I_{R_s}$ is applied, where C is a function of the thickness. For the application, I_{R_s} are measured substrate values. This is a very simple but effective configuration to account for differences in lighting, camera sensors and microscopes.

From the input value, the difference between the value of each channel and the corresponding reference red, green or blue value is calculated separately for each thickness. What results, is a list, which contains the difference for all colour channels for every thickness. The thicknesses associated with the lowest differences can now be identified. The amount of outputs can be customised by including a defined error range. This error, called cutoff, denotes the number of units on top of the minimal difference, up to which differences are accepted and thicknesses in this range are then indicated. If the cutoff is higher, the application yields more possible thicknesses, which increases the chance, that the correct thickness is in the output, but decreases the precision.

As shown in section 6.1.3 and 6.2.2, there may be many indistinguishable thicknesses if only one colour channel is considered. A solution is, that the application calculates the average of the differences of all channels for every thickness; thus similarities between thicknesses in one channel are counterbalanced by differences in other channels. This works only because the indistinguishable thicknesses vary greatly from channel to channel. There is another possible approach: instead of the average, the highest difference out of the three from the colour channels

for every thickness is used as critical difference when finding the minimal difference over all thicknesses. The benefits and problems of the two approaches are compared in section 7.3.
The principle underlying the application is summarised in Figure 27. The differences at each thickness between an input RGB value and the reference RGB values, which are determined by the contrast fit-functions, are displayed. In Figure 27a, the average difference is indicated as a black curve, whereas in Figure 27b, the maximal difference is highlighted. Between the two graphs, the input RGB value is shown (blue).



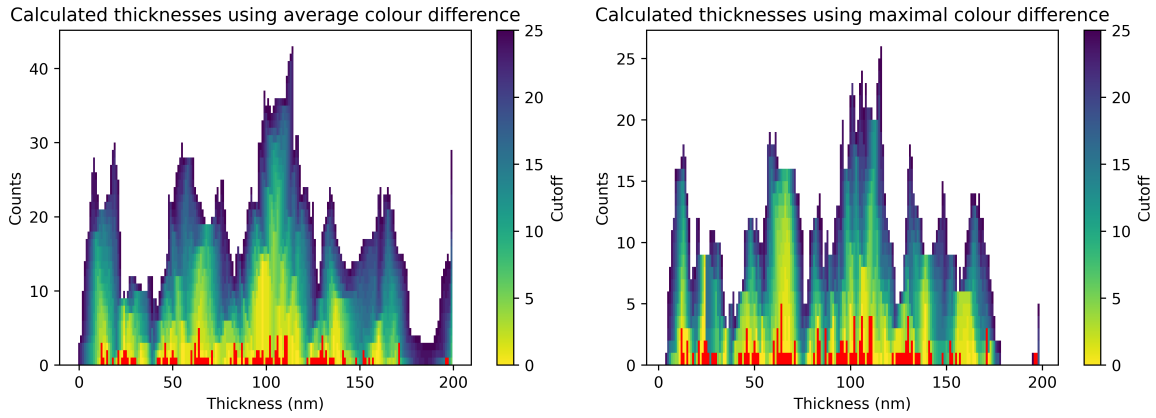
(a) Black curve = average difference

(b) Black curve = highest difference

Figure 27: Principle of the application; black horizontal line = cutoff

7.3 Evaluation of Software - Benefits and Limitations

In order to examine the advantages and disadvantages of taking the average or the maximal difference, Figure 28 and Figure 29 have been generated using the measured heights and RGB values of $FePS_3$ as inputs. A histogram of all the output thicknesses is shown in Figure 28. Since a higher cutoff always contains the output thicknesses of lower cutoff, the results for different cutoffs were stacked behind each other. The colours allow distinction between the cutoffs (0-25). The input thicknesses are coloured in red.



(a) Red = measured thickness,
other colours = output thickness

(b) Red = measured heights,
other colours = calculated heights

Figure 28: Histograms of calculated and measured thicknesses per cutoff.

The number of thicknesses yielded using the average colour difference is much greater than the number of outputs by the maximal difference (by a factor of approx. 1.5). The number of thicknesses, at which the difference is below a certain cutoff, is higher when using the average difference, since the average difference is always lower or equal to the maximal difference. Further, from Figure 28, it can be deduced, that the maximal difference is generally more specific, because the spread of yellow colours is less broadly distributed at low frequencies. Up to around 100 nm, the peaks of the low cutoffs coincide with the input thicknesses. Only around 150 nm, there is a significant distance between the indicated and the measured thicknesses. The inputs close to 200 nm are yielded only for very high cutoffs. There, the average difference has its benefit: Even at relatively low cutoffs, it yields the correct thickness of almost all inputs. As a trade-off, the amount of output thicknesses is high.

Figure 29 shows two kinds of ratios: the number of correctly identified samples divided by the number of inputs (here: 124). Ratios of this kind are denoted by 'id. samples', called the integrity of the output and are distinguished between average (av) and maximal (max) difference operating method. The indicated uncertainty range (e.g. ±2.5 nm) is used for the classification as identified sample: If the measured height is not contained exactly in the output thicknesses, but a thickness, which differs from the measured thickness only by the denoted amount of uncertainty, is included, then the sample is called identified. The second kind of ratio in Figure 29 is the number of output thicknesses which differ at most by the uncertainty range from the measured thickness divided by the total number of output thicknesses. Let us call this ratio the quality of the output. In the graph, it is denoted as 'av' or 'max'.

$$\text{integrity} = \frac{\#\text{correctly id. samples}}{\#\text{inputs}} \qquad \text{quality} = \frac{\#\text{output thicknesses in uncertainty range}}{\#\text{output thicknesses}} \tag{76}$$

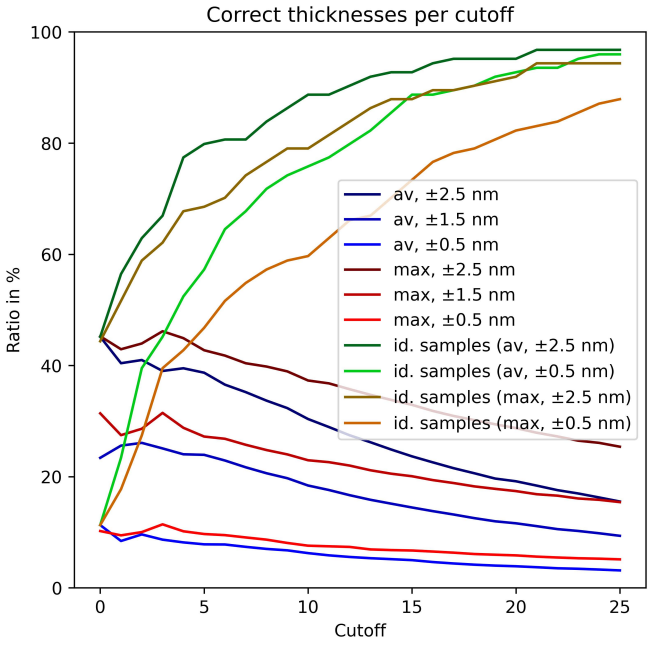


Figure 29: Integrity vs. quality of average (av) and maximal (max) differences

The Figure 29 shows that approximately 45% of the output thicknesses are at least as close as $\pm 2.5\,\text{nm}$ to the actual thickness. Further computations have shown that >80% (for max, >70% for av.) of the output thicknesses are at least as close as $\pm 5.5\,\text{nm}$. Overall, the maximal difference leads to a higher quality of the output than the maximal difference. However, the integrity of the output is better using the average difference. The two methods can therefore be recommended for various situations: If precise results ($\pm 0.5 - 2.5\,\text{nm}$) are needed, there is an easy accessible AFM and a lot of time available, using the average difference is suitable. This method is also beneficiary for very expensive materials, where missing out on flakes would be a big loss. If the highest precision is less important, the time short or the AFM less available, the maximal difference is the better option.
Finally, Figure 29 indicates, that, using the maximal difference, the quality is similar for cutoffs 0 and 2-3, while the integrity is much higher. There is no evidence that this is not an outlier due to our specific set of heights. However, if ratified by other research groups, this may increase the utility of the method of thickness identification by optical imaging to an even higher level. Overall, the cutoff serves as an balancing factor between the two methods. For example, if an AFM is easily accessible, the maximal difference can be applied using a higher cutoff than what would be useful for average difference.

7.4 Outlook

From a list of RGB values and a measured substrate RGB value, our application can generate lists of thicknesses, to which the RGB values are likely to correspond. The application can therefore be used to analyse images and indicate thicknesses of a desired height. To fully automatise the identification of flakes of a desired thickness, an optical microscope with a motorised sample platform would be required. Then, an application can be written, which, after the sample has been loaded, scans over it, producing a high resolution image of the sample, and then use our application to mark the positions of flakes of desired thickness. Thus hours of searching for the right flakes can be saved.
To achieve a more precise application, it has been suggested by K. Kitagawa [8] to use a camera, which is more sensitive and is able to distinguish more colours (10-bit). Subnanometre precision could be expected.
The precision of the application can probably be enhanced even more if a simple learning algorithm is included as an addition after the data fit: As training, a list of RGB values and the corresponding measured heights, a learning factor α and the fit-functions are required. From the fits, reference RGB values are computed for every thickness. The application then receives the RGB values as an input and yields some output thicknesses. The reference RGB value for the output thicknesses, which are close to the actual thickness, are changed by α times the difference between the reference and the input value, such that the difference decreases. the reference RGB values for the output thicknesses, which are far away form the actual thickness, are changed by the same value, but with the opposite sign, such that the differences increase. Thus, the precision of the reference RGB values can be enhanced, bringing the utility of thickness identification by optical imaging to a new level of precision.

8 Conclusion

In section 1, two central questions were asked: "How does the optical contrast and the colour of a three-layer system, consisting of thin flakes of the insulator $FePS_3$ and of the metal $NbSe_2$ on SiO_2/Si substrate, change as a function of flake thickness?" and "Can the results be used to predict thicknesses for exfoliated insulators?". To answer these two questions, a theoretical model has been derived from Maxwell's equations, experimental data was collected and evaluated, and a Python script has been written to enable the integration of thickness identification by optical imaging in the production workflow of researchers.

Our results confirm, that the theoretical model can describe the contrast and colour of an insulator on a two-layered substrate well. The correlation attests, that the underlying assumptions of the models are well chosen - especially, since there is a significant divergence between the experimental and theoretical results for the conductive material.

From the experimental results of $FePS_3$, an application was developed, which is able to convert a list of RGB values into a list of thicknesses, to which the colours correspond very likely. A precision, resulting in above 80% of the output thicknesses diverging less than $\pm 5.5\,\text{nm}$ from the actual height and even 45% of the output within a range of $\pm 2.5\,\text{nm}$, was achieved. This confirms the high practicality of thickness identification by optical imaging, providing a fast, non-invasive, large-scale and cheap method to determine the thickness of 2D materials.

9 Personal Motivation and Reflection

When I started to search for a topic for my matura paper, I had one crucial criterion: the work shall have a real benefit for any group of people. Even after talking to many people from both areas, I had some ideas developed, but non of them was fully convincing. Finally, a friend and leader from the Swiss Physics Olympiad enabled me to get in contact with a doctoral student at the Department of Materials of the ETH Zürich. Together we developed a rough aim for this study, trying to find the areas, where I could create significant benefit for his research. I was aware that the project was at the upper limit of effort for this particular assignment - I had already prepared a plan B. Looking at the results, I am very happy of my decision to pursue this aim, because I was able to create an application, which has an actual use.
For any future studies, if I shall do them, I need more endurance in the literature research, to figure out the detailed insights, which have been found already, and the problems, which occur often. Furthermore, a greater set of preliminary experiments and more in detail planning of the experiment could perhaps have prevented the differences between day one and two of the data collection for $FePS_3$. Apart from that, I really appreciated the back and forth between the theoretical model and the experimental results, which didn't match at all in the beginning because of errors. This forced me to revise and question every step in detail - and finally led to a much more in depth understanding of what is shown in this report.
In the process of satisfying my criterion, I learned a lot - programming and data analysis in Python, literature research for the theoretical model, planning and conducting experiments, and incredibly many smaller insights and experiences. On top of that, I developed a fascination for 2D materials and their countless applications, which will continue to flourish for a long time.

References

[1] H. Anders. *Thin Films in Optics*. The Focal Press, 1967. ISBN 978-0240447933.

[2] M. M. Benameur, B. Radisavljevic, J. S. Heron, S. Sahoo1, and H. Berger. Visibility of dichalcogenide nanolayers. *IOP Science*, 2011.

[3] D. Bing, Y. Wang, J. Bai, R. Du, G. Wu, and L. Liu. Optical contrast for identifying the thickness of two-dimensional materials. *Optics Communications*, 406:128–138, 2018. ISSN 0030-4018. doi: https://doi.org/10.1016/j.optcom.2017.06.012. URL `https://www.sciencedirect.com/science/article/pii/S0030401817304789`. Optoelectronics and Photonics Based on Two-dimensional Materials.

[4] M. Casimiro, L. Ferreira, J. Leal, C. Pereira, and B. Monteiro. Ionizing radiation for preparation and functionalization of membranes and their biomedical and environmental applications, 2019. URL `https://www.mdpi.com/2077-0375/9/12/163`. (Accessed on 9.12.2023).

[5] J. Edmunds. Thin-film interference using fresnel equations, 2019. URL `https://youtu.be/EOsWDeMuBgo`. (Accessed on 1.8.2023).

[6] E. B. Goldstein. *Wahrnehmungspsychologie, Der Grundkurs*. Hans Irtel, Spektrum, Akademischer Verlag, 2007. ISBN 978-3827417664. Translation by Guido Plata, Original english title: Sensation and Perception, Seventh Edition.

[7] P. D. E. Hecht. *Optik - 7. Auflage*. DE GRUYTER, 2018. ISBN 978-3-11-052664-6. Translation by Karen Lippert, Original english title: Optics - 5th Edition.

[8] K. Kitagawa. Thin-film thickness profile measurement by three-wavelength interference color analysis. *Appl. Opt.*, 52(10):1998–2007, Apr 2013. doi: 10.1364/AO.52.001998. URL `https://opg.optica.org/ao/abstract.cfm?URI=ao-52-10-1998`.

[9] P. Kumbhakar, J. S. Jayan, A. Sreedevi Madhavikutty, P. Sreeram, A. Saritha, T. Ito, and C. S. Tiwary. Prospective applications of two-dimensional materials beyond laboratory frontiers: A review. *iScience*, 26(5):106671, 2023. ISSN 2589-0042. doi: https://doi.org/10.1016/j.isci.2023.106671. URL `https://www.sciencedirect.com/science/article/pii/S2589004223007484`.

[10] D. J. Late, B. Liu, H. S. S. R. Matte, C. N. R. Rao, and V. P. Dravid. Rapid characterization of ultrathin layers of chalcogenides on sio2/si substrates. *Advanced Functional Materials*, 22(9):1894–1905, 2012. doi: https://doi.org/10.1002/adfm.201102913. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/adfm.201102913`.

[11] B. Ma, P. Wang, S. Ren, C. Jia, and X. Guo. Versatile optical determination of two-dimensional atomic crystal layers. *Elsevier*, 2016.

[12] K. S. Novoselov, A. Mishchenko, A. Carvalho, and A. H. C. Neto. 2d materials and van der waals heterostructures. *Science*, 353(6298):aac9439, 2016. doi: 10.1126/science.aac9439. URL `https://www.science.org/doi/abs/10.1126/science.aac9439`.

[13] B. Voigtländer. *Scanning Probe Microscopy, Atomic Force Microscopy and Scanning Tunneling Microscopy*. Springer, 2015. ISBN 978-3-662-45239-4.

[14] L. Wang and J. Yu. Chapter 1 - principles of photocatalysis. In J. Yu, L. Zhang, L. Wang, and B. Zhu, editors, *S-scheme Heterojunction Photocatalysts*, volume 35 of *Interface Science and Technology*, pages 1–52. Elsevier, 2023. doi: https://doi.org/10.1016/B978-0-443-18786-5.00002-0. URL `https://www.sciencedirect.com/science/article/pii/B9780443187865000020`. (Accessed on 9.12.23).

[15] W. Zinth and U. Zinth. *Optik, Lichtstrahlen - Wellen - Photonen*. Oldenbourg Verlag München, 2018. ISBN 978-3-11-049501-0.

Used Technologies

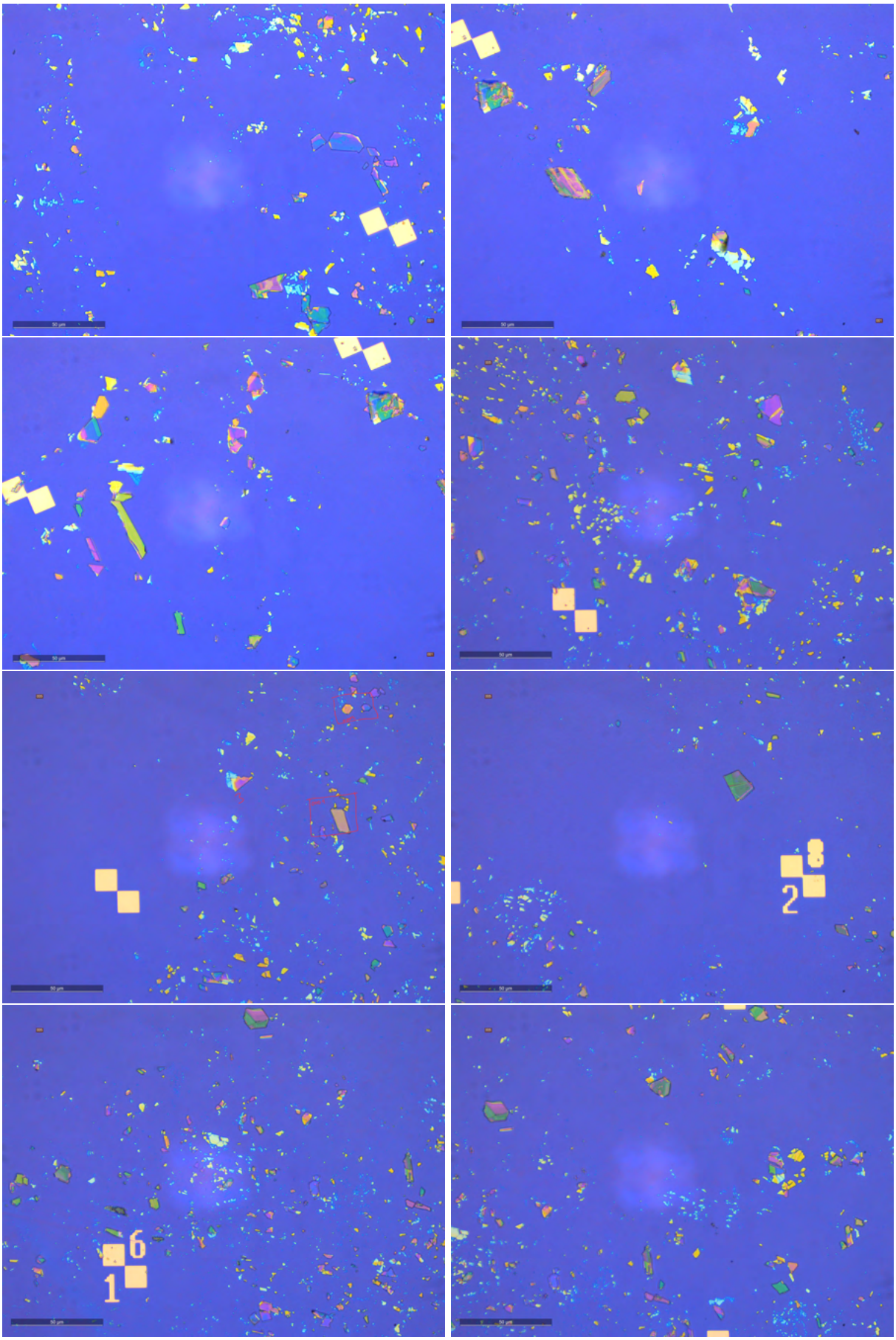
The data evaluation has been done using Gwyddion Version 2.63, Spyder (with Python 3.11.3), Microsoft Excel (Version 2308 Build 16.0.16731.20182), and Desmos Scientific Calculator. Small parts of the Python code, mostly concerning visual improvements for the graphs, are inspired by responses from ChatGPT Version 3.5.

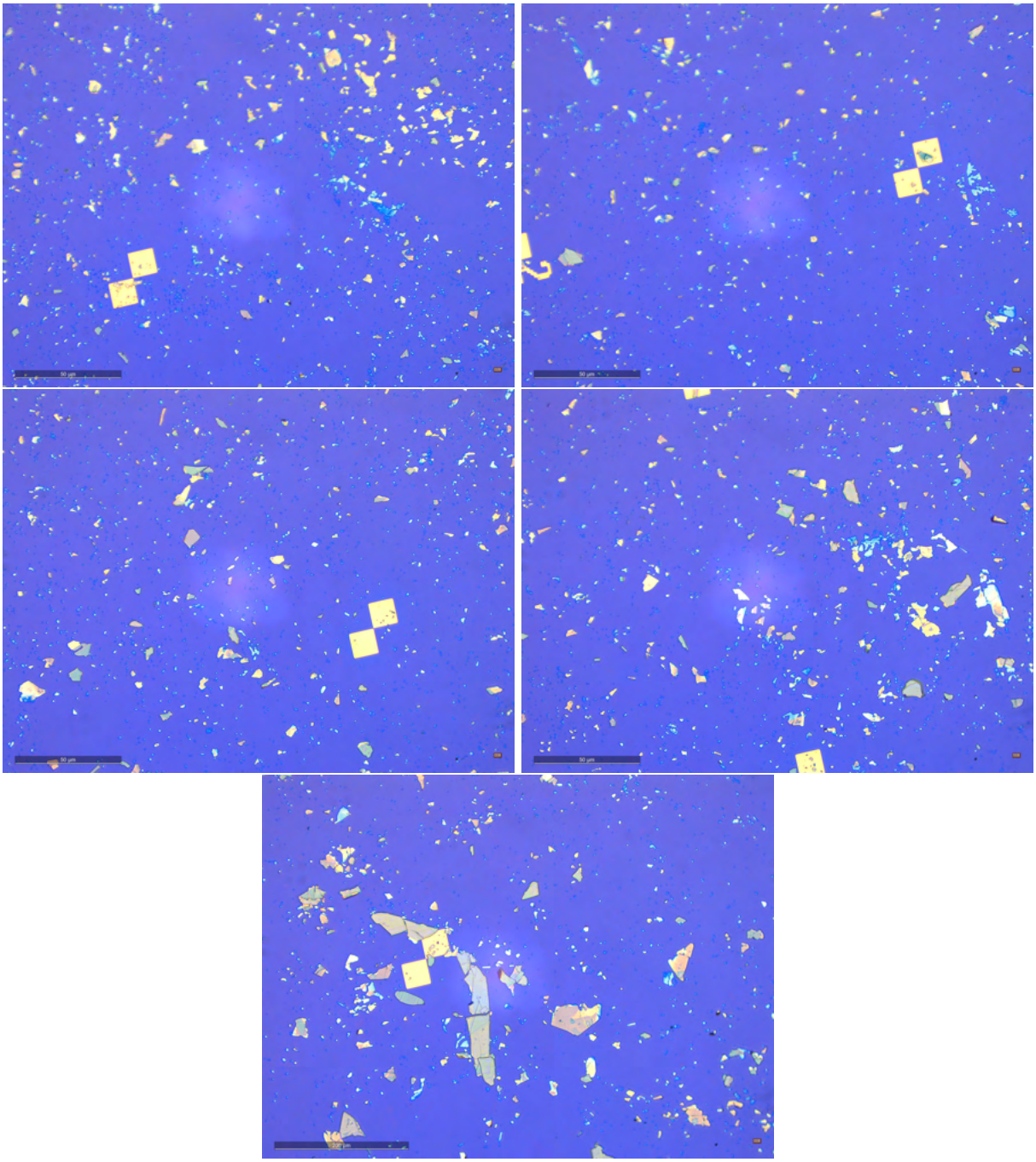
Appendix

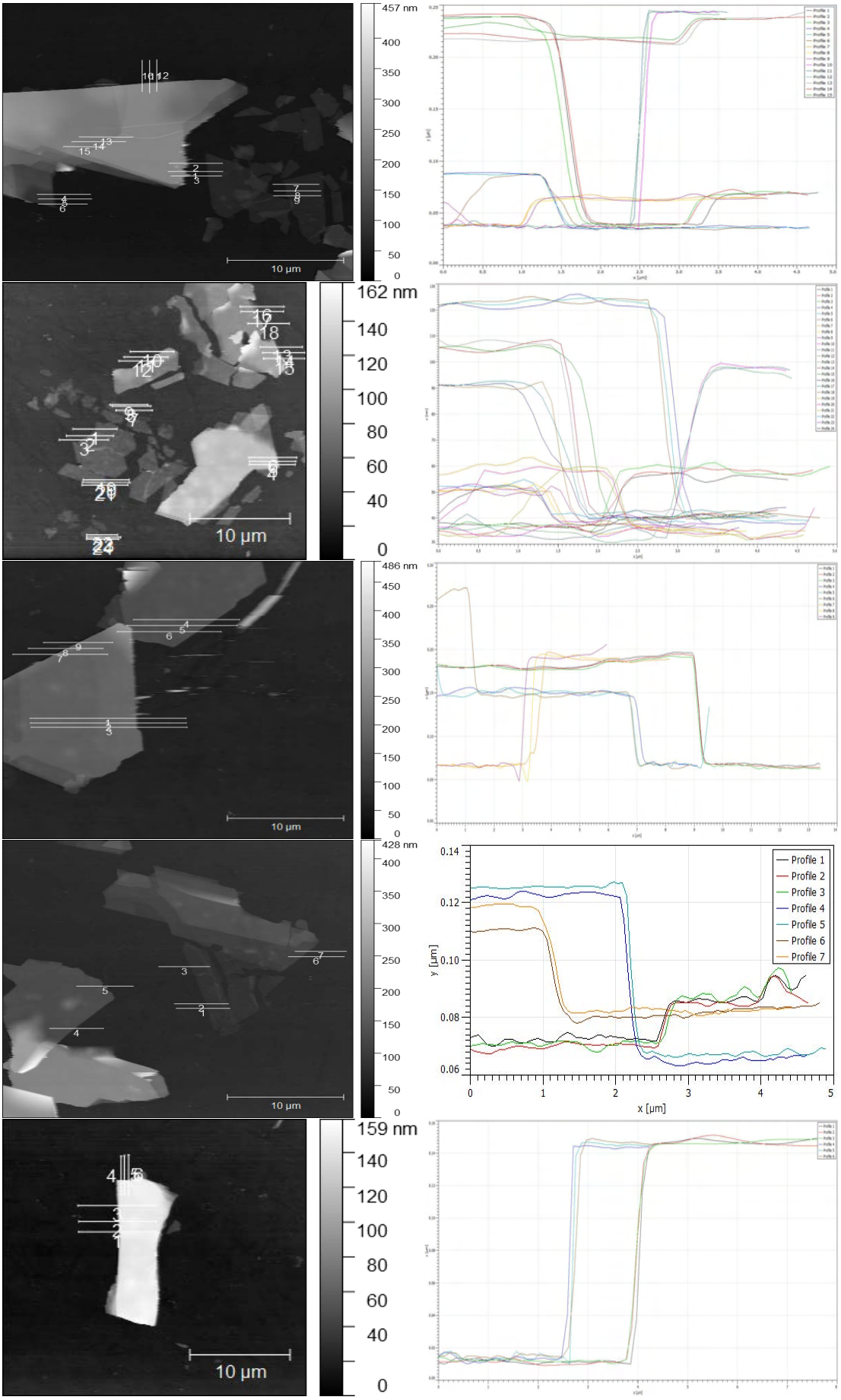
The following points provide a summary of the procedure:

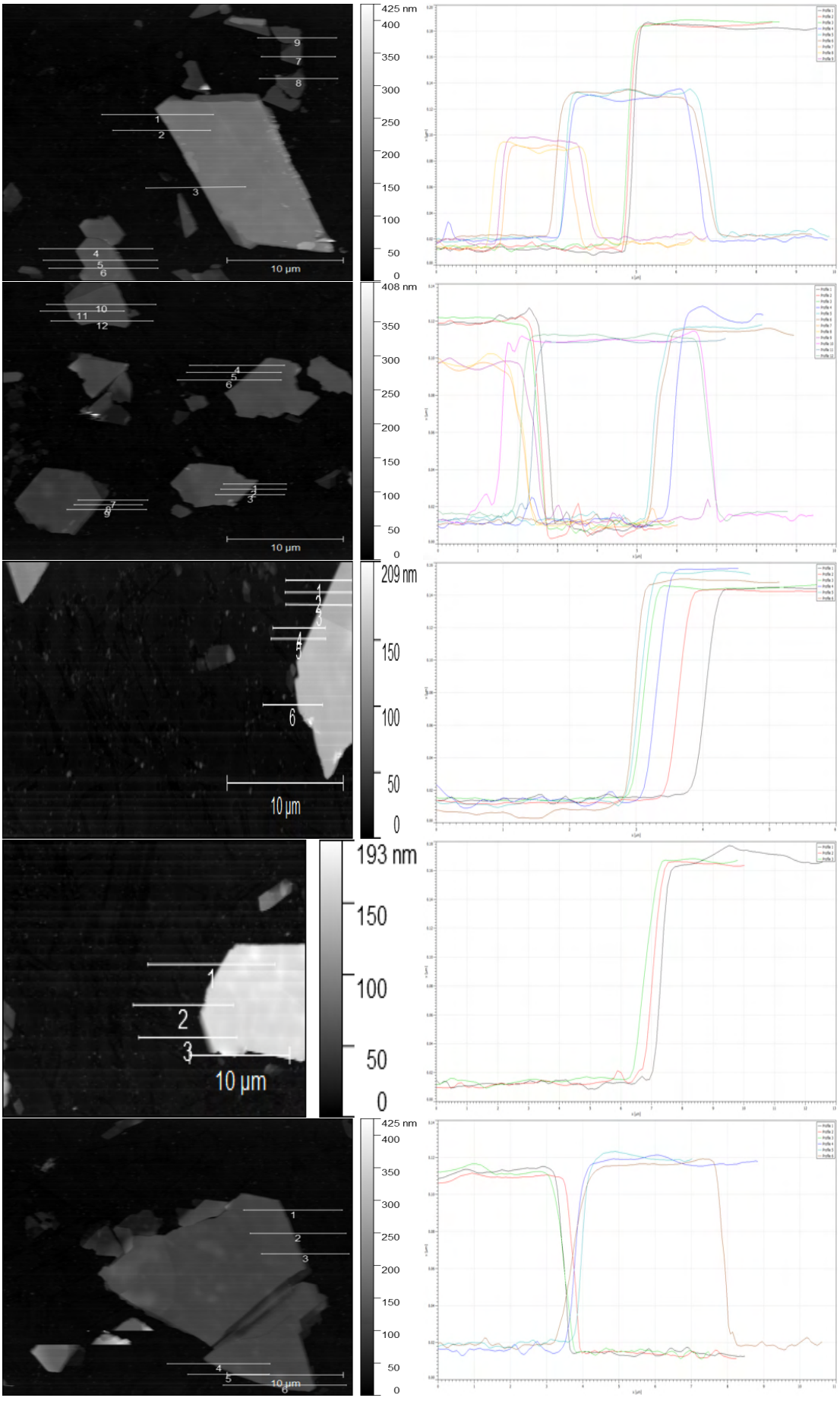
- Substrate Preparation and Exfoliation
- Optical and AFM Imaging
- Processing AFM images (Gwyddion commands: Level data by fitting a plane through points, Shift minimum data value to zero, Extracting profiles along arbitrary lines)
- Drawing and fitting profiles, recording height and error
- Obtaining RGB values from the same positions
- Levelling the RGB values, mean and std.n per 5×5 pixels
- Computation of contrast
- Graphs for measured values (intensity and contrast), curve-fit
- Reconstruction of colours from contrast fit-functions
- Colour stripes
- Theoretical model: computing reflectivity and contrast
- 3D plots of reflectivity and contrast generated.
- Theoretical model: computing colour and calibration; graphing
- Repetition of process for the other material

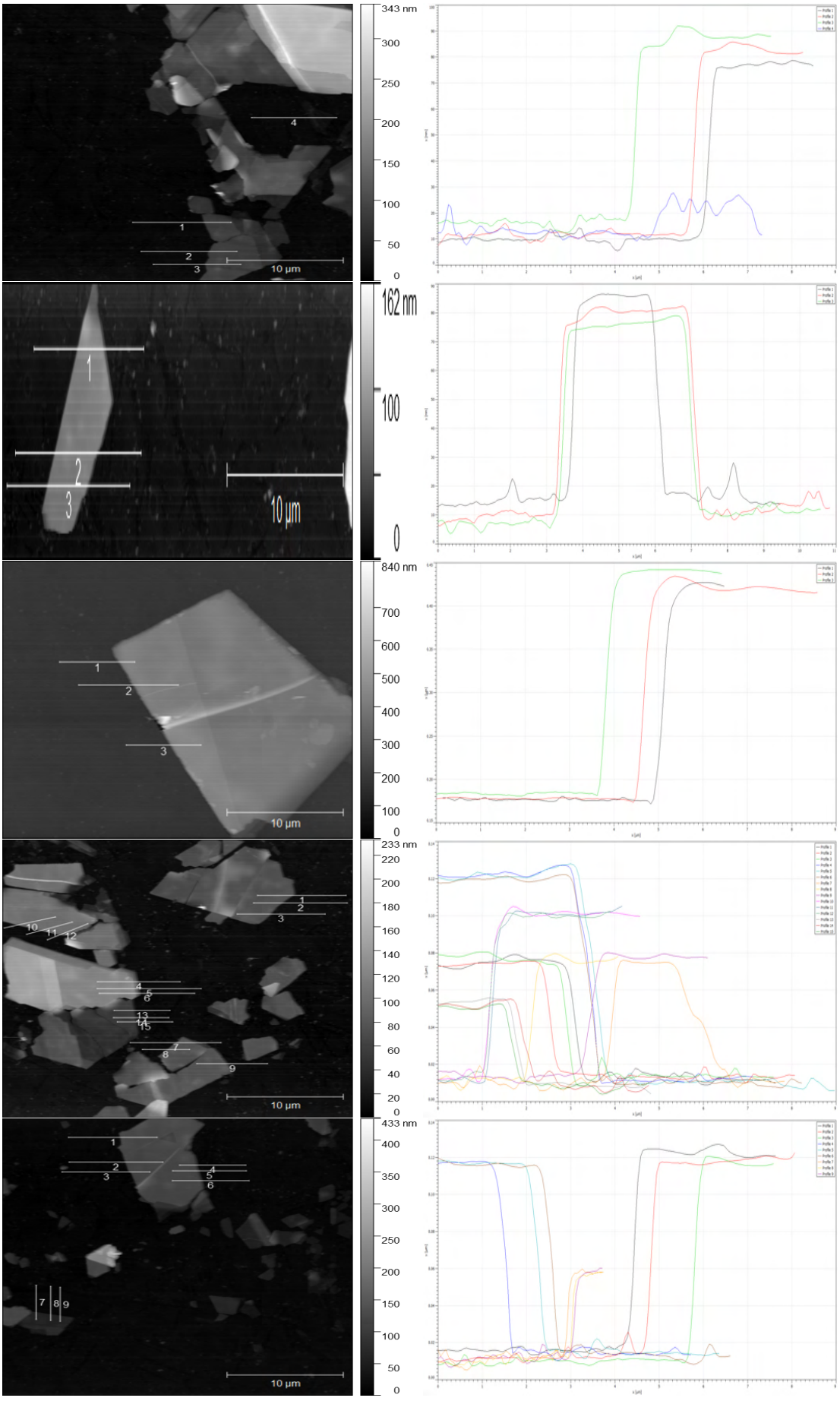
The following pages contain:

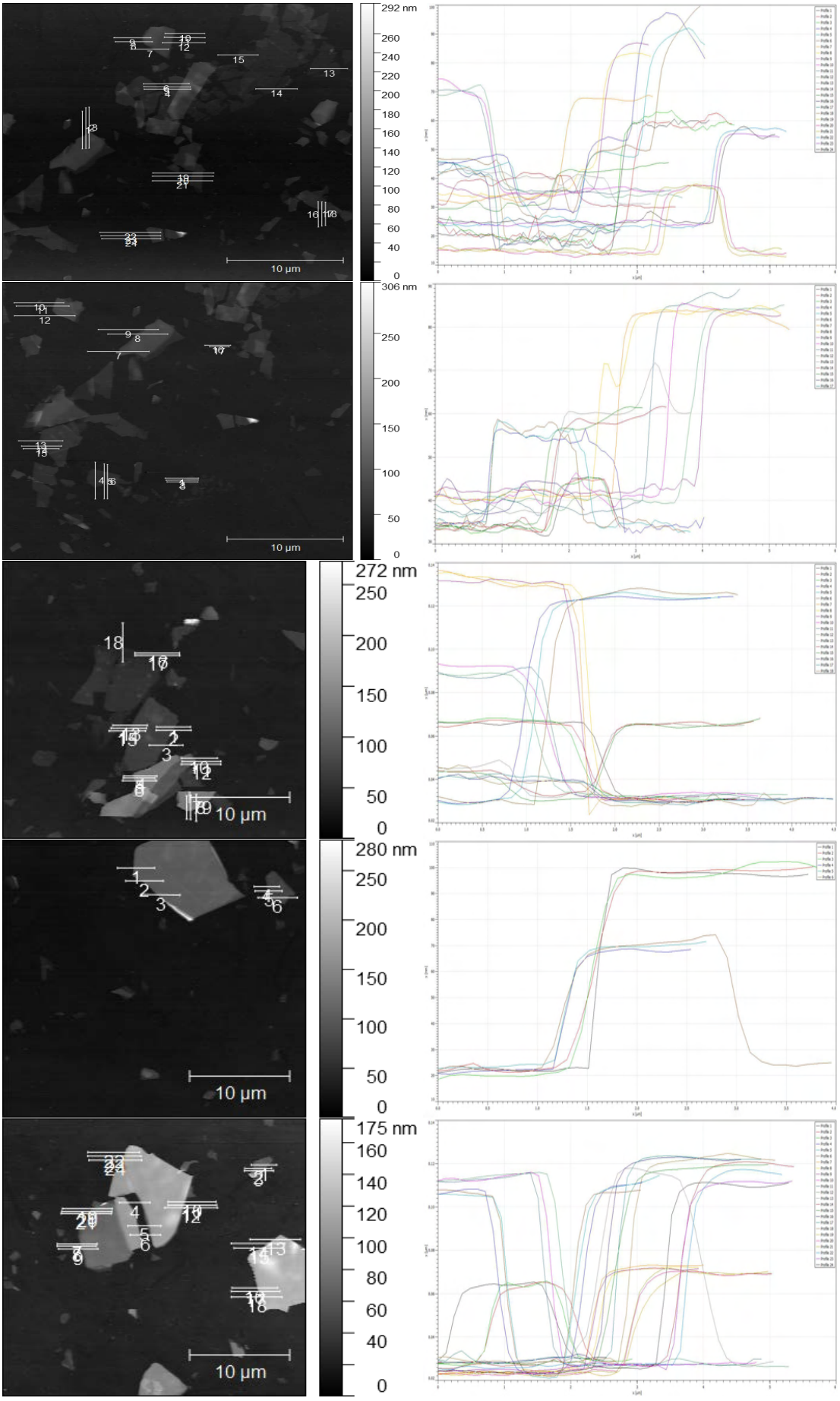

2
6
1




457 nm
400
350
300
250
200
150
100
50
0
10 µm
162 nm
140
120
100
80
60
40
0
10 µm
486 nm
450
400
350
300
250
200
150
100
50
0
10 µm
428 nm
400
350
300
250
200
150
100
50
0
10 µm
159 nm
140
120
100
80
60
40
0
10 µm
Profile 1
Profile 2
Profile 3
Profile 4
Profile 5
Profile 6
Profile 7
y [µm]
x [µm]


425 nm
400
350
300
250
200
150
100
50
0
10 µm
408 nm
209 nm
193 nm


343 nm
300
250
200
150
100
50
0
10 µm
162 nm
100
0
10 µm
840 nm
700
600
500
400
300
200
100
0
10 µm
233 nm
220
200
180
160
140
120
100
80
60
40
20
0
10 µm
433 nm
400
350
300
250
200
150
100
50
0
10 µm


292 nm
260
240
220
200
180
160
140
120
100
80
60
40
0
10 µm
306 nm
250
200
150
100
50
0
10 µm
272 nm
250
200
150
100
50
0
10 µm
280 nm
250
200
150
100
50
0
10 µm
175 nm
160
140
120
100
80
60
40
0
10 µm

```
import numpy as np
import cv2
import matplotlib.pyplot as plt

path = r'Used_Images\FePS3_514.png'
x=1820
y=1100
imag= cv2.imread(path, cv2.IMREAD_COLOR)
(b, g, r) = cv2.split(imag)
img= cv2.imread(path, cv2.IMREAD_GRAYSCALE)
L= [r,g,b,img]
La=['r','g', 'b','all', 'finish' ]
M=[0,0,0,0]
ST=[0,0,0,0]
for i in range (5):
    print(La[i])
    m=[]
    for p1 in range(5):
        for p2 in range(5):
            if i<4:
                print(L[i][p1+y][p2+x])
                m.append(L[i][p1+y][p2+x])
            else: img[p1+y][p2+x]=0
    if i<4:
        M[i]=sum(m)/len(m)
        ST[i]=np.std(m)
for j in range(len(M)):
    print (M[j])
    print (ST[j])
for i in range(int(350)):
    for j in range(450):
        img[i*5][j*5]=0
        if i<= 389/20 and j <=524/20:
            img[i*20][j*20]=255
print(x)
print(y)
plt.rcParams['figure.dpi']=700
plt.imshow(img, cmap='gray')
plt.show()
```

Listing 1: Collecting the RGB values from pictures

```
import matplotlib.pyplot as plt
import numpy as np
import cmath
from scipy import interpolate
from scipy.optimize import curve_fit
from matplotlib import cm
import pandas as pd

n0=1
n1=2.41+0.1j  #refractive index of the flakes; for NbSe2, the data was
    treated as for Si.
thinnest_layer=2E-9
max_layers= 100
lambdmax=680E-9
lambdmin=400E-9
excel_file = "FePS3 Contrasts.xlsx"
```

```
plt.rcParams['figure.dpi']=300
colormap = plt.cm.viridis

# Load refractive index data
data = np.genfromtxt('Si.txt')
x = data[:,0]*1E-9
data_n_Si = data[:,1]
data_k_Si = data[:,2]
f_n_Si = interpolate.interp1d(x, data_n_Si)
f_k_Si = interpolate.interp1d(x, data_k_Si)
data = np.genfromtxt('SiO2.txt')
x = data[:,0]*1E-9
data_n_SiO2 = data[:,1]
data_k_SiO2 = data[:,1]
f_n_SiO2 = interpolate.interp1d(x, data_n_SiO2)
f_k_SiO2 = interpolate.interp1d(x, data_k_SiO2)
df = pd.read_excel(excel_file)
x = df['Heighttt'].values
yr = df['ConR'].values
yg =df['ConG'].values
yb =df['ConB'].values

def damped_sinusoid(t, A, lambda_, omega, phi, C):
    return A * np.exp(-lambda_ * t) * np.cos(omega * t + phi) + C
popt_r, pcov = curve_fit(damped_sinusoid, x, yr, p0=[10, 1E-3, 1E-2, 0,
    0])
popt_g, pcov = curve_fit(damped_sinusoid, x, yg, p0=[10, 1E-3, 1E-2, 0,
    0])
popt_b, pcov = curve_fit(damped_sinusoid, x, yb, p0=[10, 1E-3, 1E-2, 0,
    0])
def Red_measured_FePs3 (x): #x=thickness
    r_conmes= damped_sinusoid(x, *popt_r)
    if r_conmes >1: r_conmes =1
    return (1-r_conmes)*91  #here, substrate values are included (91)
def Green_measured_FePs3 (x): #x=thickness
    r_conmes= damped_sinusoid(x, *popt_g)
    if r_conmes >1: r_conmes =1
    return  (1-r_conmes)*99 #here, substrate values are included (99)
def Blue_measured_FePs3 (x): #x=thickness
    r_conmes= damped_sinusoid(x, *popt_b)
    if r_conmes >1: r_conmes =1
    return (1-r_conmes)*231 #here, substrate values are included (231)
def getn_Si(lam):
    n_Si = f_n_Si(lam)
    k_Si = f_k_Si(lam)
    return n_Si + k_Si*1j
def getn_SiO2(lam):
    n_SiO2 = f_n_SiO2(lam)
    k_SiO2 = f_k_SiO2(lam)
    return n_SiO2 + 0*k_SiO2*1j
def phi (d, n1, lam):
    return (2*(np.pi)*n1*d)/(lam)
def phib (d, n2, lam):
    return (2*(np.pi)*n2*d)/(lam)
def calculate_RS(lam, r2, r3, d2, n2):
    numerator= r2+ r3*cmath.exp(2j*phib(d2,n2, lam))
    denominator= 1+r2*r3*cmath.exp(2j*phib(d2,n2, lam))
    R=(numerator/denominator)
```

```
    return R* np.conjugate(R)
def calculate_C(lam, d1, d2):
    n3 = getn_Si(lam)
    n2 = getn_SiO2(lam)
    r01=(n0-n1)/(n0+n1)
    r23= (n2 - n3) / (n2 + n3)
    r02= (n0 - n2) / (n0 + n2)
    r12= (n1 - n2) / (n1 + n2)
    numerator = r01 + r12*cmath.exp(2j*phi(d1,n1, lam)) + r23*cmath.exp
       (2j*(phi(d1, n1, lam)+phib(d2,n2,  lam))) +r01*r12*r23*cmath.exp
       (2j*phib(d2,n2, lam))
    denominator = 1+r01*r12*cmath.exp(2j*phi(d1,n1, lam)) + r01*r23*
       cmath.exp(2j*phi(d1,n1, lam) +2j*phib(d2,n2, lam)) + r12*r23*
       cmath.exp(2j*phib(d2, n2, lam))
    r = numerator/denominator
    R = r * np.conjugate(r)
    RS= calculate_RS(lam, r02, r23, d2, n2)
    C= (RS-R)/RS
    return C + R*1j
def legend (lay):
    L=[]
    for i in range(max_layers+1):
        L.append(str(lay*(max_layers-i)) + " nm")
    return L
def legendnormal (lay):
    L=[]
    for i in range(len(lay)):
        L.append(str(lay[i])+ " nm")
    return L
def wavelength_to_rgb(wavelength, contrast): #Approximation wavelength
   to RGB
    gamma = 1
    intensity_max = 255
    if 380 <= wavelength < 440:
        R = -(wavelength - 440) / (440 - 380)
        G = 0.0
        B = 1.0
    elif 440 <= wavelength < 490:
        R = 0.0
        G = (wavelength - 440) / (490 - 440)
        B = 1.0
    elif 490 <= wavelength < 510:
        R = 0.0
        G = 1.0
        B = -(wavelength - 510) / (510 - 490)
    elif 510 <= wavelength < 580:
        R = (wavelength - 510) / (580 - 510)
        G = 1.0
        B = 0.0
    elif 580 <= wavelength < 645:
        R = 1.0
        G = -(wavelength - 645) / (645 - 580)
        B = 0.0
    elif 645 <= wavelength <= 780:
        R = 1.0
        G = 0.0
        B = 0.0
    else:
```

```
        R = 0.0
        G = 0.0
        B = 0.0
    factor = 1 #contrast * (0.3 + 0.7 * (1.0 - np.exp(-((wavelength -
        480) / 80) ** 2)))
    R = (intensity_max * (R * factor) ** gamma)
    G = (intensity_max * (G * factor) ** gamma)
    B = (intensity_max * (B * factor) ** gamma)
    return R, G, B
def calculate_Contrasts(wavelengths, thickness):
    C_real = []
    R_real = []
    d_2D=thinnest_layer*thickness
    d_SiO2 = 285E-9
    for wavelength in wavelengths:
        C_real.append(calculate_C(wavelength, d_2D, d_SiO2).real)
        R_real.append(calculate_C(wavelength, d_2D, d_SiO2).imag)
    plt.figure(1)
    plt.plot(wavelengths*1E9, C_real, color=colormap((thickness/
        max_layers)))
    plt.figure(4)
    plt.plot(wavelengths*1E9, R_real, color=colormap((thickness/
        max_layers)))
    plt.title('Predicted RGB Values')
    return  R_real, C_real
def calculate_weighted_av_RGB (R,G,B,r,g,b, contrast,subcon):
    r+=R*contrast
    g+=G*contrast
    b+=B*contrast
    return r,g,b

Reds=[]
Greens=[]
Blues=[]
plt.figure(1)
plt.figure(2)
wavelengths = np.linspace(lambdmin, lambdmax, num=int((lambdmax -
    lambdmin) / 1E-9) + 1)
fig=plt.figure(figsize=(11,7))
ax= fig.add_subplot(111, projection='3d')
figu=plt.figure(figsize=(11,7))
axu= figu.add_subplot(111, projection='3d')
conthick = [[],[],[]]
extractedlam = [440,520,580]  #these Wavelengths are extracted from the
    model

for o in range(1,max_layers+1,1):
    Reflectivity, Contrast = calculate_Contrasts(wavelengths,
        max_layers+1-o)
    #3D plots:
    ax.plot(wavelengths*1E9,[(max_layers+1-o)*thinnest_layer*1e9 for _
        in range(len(wavelengths))], Contrast, label=int((max_layers+1-o
        )*thinnest_layer*1E9), color=colormap((o/max_layers)), linewidth
        =1.5)
    axu.plot(wavelengths*1E9,[(max_layers+1-o)*thinnest_layer*1e9 for _
         in range(len(wavelengths))], Reflectivity, label=int((
        max_layers+1-o)*thinnest_layer*1E9), color=colormap((o/
        max_layers)), linewidth=1.5)
```

```
    conthick[0].append(Contrast[int(extractedlam[0]-lambdmin*1e9)]*1*1)
    conthick[1].append(Contrast[int(extractedlam[1]-lambdmin*1e9)]*1*1)
    conthick[2].append(Contrast[int(extractedlam[2]-lambdmin*1e9)]*1*1)
    R, G, B = [],[],[]
    r, g, b = 0,0,0
    for i in range(len(wavelengths)):
        R, G, B = wavelength_to_rgb(wavelengths[i]*1e9, Reflectivity[i
            ])
        r,g,b=calculate_weighted_av_RGB(R, G, B, r, g, b, Reflectivity[
            i], Contrast[i])

    tot_contr= sum(Reflectivity[:])
    r=r / (tot_contr) *1*1.3 #CALIBRATION
    g=g / (tot_contr) *1*1.3
    b=b / (tot_contr)  *2*1.3
    Reds.append(r)
    Greens.append(g)
    Blues.append(b)

ax.plot(wavelengths*1E9,[0 for _ in range(len(wavelengths))], [0 for _
    in range(len(wavelengths))], color='yellow', linewidth=1.5)
ax.set_xlim(lambdmin*1e9,lambdmax*1e9)
ax.set_xlabel('Wavelength (nm)')
ax.set_ylim(None, o*thinnest_layer*1e9)
ax.set_ylabel('Thickness (nm)')
ax.set_zlim(None,1)
ax.set_zlabel('Contrast')
ax.set_title('Predicted Contrast FePS$_3$')
cbar = fig.colorbar(plt.cm.ScalarMappable(cmap=cm.viridis_r), ax=ax,
    shrink =0.5)
cbar.set_label('Thickness (nm)')
cbar.set_ticks(np.linspace(0, max_layers*thinnest_layer*1e6*5, 11))
cbar.set_ticklabels([f'{int(tick*thinnest_layer*1e9/2)}' for tick in np
    .linspace(0, max_layers*thinnest_layer*1e9, 11)])  # Set labels to Y
     values
ax.grid(True)
axu.set_xlim(lambdmin*1e9,lambdmax*1e9)
axu.set_xlabel('Wavelength (nm)')
axu.set_ylim(None, o*thinnest_layer*1e9)
axu.set_ylabel('Thickness (nm)')
axu.set_zlim(0, None)
axu.set_zlabel('Intensity')
cbaru = fig.colorbar(plt.cm.ScalarMappable(cmap=cm.viridis_r), ax=axu,
    shrink =0.5)
cbaru.set_label('Thickness (nm)')
cbaru.set_ticks(np.linspace(0, max_layers*thinnest_layer*1e6*5, 11))
cbaru.set_ticklabels([f'{int(tick*thinnest_layer*1e9/2)}' for tick in
    np.linspace(0, max_layers*thinnest_layer*1e9, 11)])  # Set labels to
     Y values

plt.figure(6)
plt.plot([1e9*thinnest_layer*(max_layers-_) for _ in range(max_layers)
    ], conthick[0], color= 'blue')
plt.plot([1e9*thinnest_layer*(max_layers-_) for _ in range(max_layers)
    ], conthick[1], color= 'green')
plt.plot([1e9*thinnest_layer*(max_layers-_) for _ in range(max_layers)
    ], conthick[2], color = 'red')
plt.plot([1e9*thinnest_layer*(max_layers-_) for _ in range(max_layers)
```

```
    ], [damped_sinusoid(1e9*thinnest_layer*(max_layers-_), *popt_b) for
    _ in range(max_layers)], color='blue', linestyle='dashed',linewidth
    =1)
plt.plot([1e9*thinnest_layer*(max_layers-_) for _ in range(max_layers)
    ], [damped_sinusoid(1e9*thinnest_layer*(max_layers-_), *popt_g)*1
    for _ in range(max_layers)], color='green', linestyle='dashed',
    linewidth=1)
plt.plot([1e9*thinnest_layer*(max_layers-_) for _ in range(max_layers)
    ], [damped_sinusoid(1e9*thinnest_layer*(max_layers-_), *popt_r)*1
    for _ in range(max_layers)], color='red', linestyle='dashed',
    linewidth=1)
plt.xlabel('Thickness (nm)')
plt.ylabel('Contrast $C$')
plt.legend(legendnormal(extractedlam))
plt.title('Predicted and Measured Contrast $C$ per Thickness')

plt.figure(1)
plt.legend(legend(thinnest_layer*1E9), loc='lower right',fontsize='
    small')
plt.title('Predicted Contrast FePS$_3$')
plt.ylabel('Contrast')
plt.xlabel('Wavelength (nm)')

plt.figure(2)
plt.figure(figsize=(12, 6))
num_values = max_layers
colors = [(Reds[num_values-_-1]/255, Greens[num_values-_-1]/255, Blues[
    num_values-_-1]/255) for _ in range(num_values)]
rgb_colors = np.array(colors)
fig, ax = plt.subplots(figsize=(8, 3))
ax.imshow(rgb_colors[np.newaxis, :, :], aspect='auto', extent=[0, len(
    rgb_colors), 0, 1])
plt.xlabel("Thickness (nm)")
num_values = max_layers
x_labels = [f"{int(i*thinnest_layer*1e9)}" for i in range(0, num_values
    + 1)]
x_indices_to_label = range(0, num_values+1, 10)
plt.xticks(x_indices_to_label, [x_labels[i] for i in x_indices_to_label
    ], rotation=90)
plt.yticks([])
plt.title("Predicted Colours FePS3")

#here we define the RGB Colors of the 'measured' graph
colors = [(Red_measured_FePs3(thinnest_layer*_*1e9)/255,
    Green_measured_FePs3(thinnest_layer*1e9*_)/255, Blue_measured_FePs3(
    thinnest_layer*1e9*_)/255) for _ in range(num_values)]
rgb_colors = np.array(colors)
fig, ax = plt.subplots(figsize=(6, 3.6))
ax.imshow(rgb_colors[np.newaxis, :, :], aspect='auto', extent=[0, len(
    rgb_colors), 0, 1])
plt.xlabel("Thickness (nm)")
num_values = max_layers
x_labels = [f"{int(i*thinnest_layer*1e9)}" for i in range(0, num_values
    + 1)]
x_indices_to_label = range(0, num_values+1, 10)
plt.xticks(x_indices_to_label, [x_labels[i] for i in x_indices_to_label
    ], rotation=90)
plt.yticks([])
```

```
plt.title("Measured RGB values FePS3")
plt.tight_layout()
plt.show()

Reds.reverse()
Greens.reverse()
Blues.reverse()
colorsmes = [(Red_measured_FePs3(thinnest_layer*_*1e9)/255,
    Green_measured_FePs3(thinnest_layer*1e9*_)/255, Blue_measured_FePs3(
    thinnest_layer*1e9*_)/255) for _ in range(num_values)]
colortheory = [(Reds[_]/255, Greens[_]/255, Blues[_]/255) for _ in
    range(num_values)]
colors = colorsmes #colortheory
plt.figure(figsize=(10,8))
col= ['red colours', 'green colours', 'blue colours' ,'all colours']
for k in range(4):
    data=[]
    for i in range(len(colors)):
        heat = []
        for j in range(len(colors)):
            if k <= 2:
                diff = abs(colors[i][k]*255-colors[j][k]*255)
            elif k==3:
                diffr = abs(colors[i][0]*255-colors[j][0]*255)
                diffg = abs(colors[i][1]*255-colors[j][1]*255)
                diffb = abs(colors[i][2]*255-colors[j][2]*255)
                diff = (diffr+diffg+diffb)/3
            if diff >10:  val = 0
            elif diff >9: val = 0.1
            elif diff >8: val = 0.2
            elif diff >7: val = 0.3
            elif diff >6: val = 0.4
            elif diff >5: val = 0.5
            elif diff >4: val = 0.6
            elif diff >3: val = 0.7
            elif diff >2: val = 0.8
            elif diff >1: val = 0.9
            elif diff >0.5: val = 0.95
            elif diff == 0: val = 1
            heat.append(val)
        data.append(heat)
    plt.subplot(2,2,k+1)
    heatmap = plt.imshow(data, cmap='plasma')
    plt.gca().invert_yaxis()
    cbar = plt.colorbar(heatmap)
    ticks = np.arange(0, max_layers+1, thinnest_layer*1e9*10)  # Adjust
        the range and step as needed
    plt.xticks(ticks, rotation = 90)
    plt.yticks(ticks)
    plt.ylabel('Measured Colour per Thickness (nm)')
    plt.xlabel('Predicted Colour Thickness (nm)')
    plt.title(f'Overlap of {col[k]}')
plt.tight_layout()
plt.show()
```

Listing 2: Creating theoretical model computations, colour stripes and confuseable thicknesses graph. Specific for $FePS_3$, similar for $NbSe_2$

```
import pandas as pd
import cv2
import matplotlib.pyplot as plt

data = pd.read_excel('XY values.xlsx')
path = r'FePS3_104.png'

imag= cv2.imread(path, cv2.IMREAD_COLOR)
column1_data = data['x value']
column2_data = data['y value']

plt.imshow(cv2.cvtColor(imag, cv2.COLOR_BGR2RGB))
plt.scatter(column1_data, column2_data, label='Data collection sites',
    color='white', marker='x', linewidth=0.7)
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.legend(loc = 'upper left')
plt.show()
```

Listing 3: Graph for sites of recording

```
import pandas as pd
import matplotlib.pyplot as plt

excel_file = "FePS3 RGB absolut.xlsx" #specific for FePS3

df = pd.read_excel(excel_file)
datacolor='Red','Green','Blue','All'
stdcolor='R','G','B','A'
xP = df['HeightP'].values
xQ = df['HeightQ'].values
cross_length_xP = df['stdH'].values
cross_length_xQ = df['qstdH'].values

plt.figure(figsize=(10,8), dpi=300)
for i in range(len(datacolor)):
    plt.subplot(2,2,i+1)
    yPr = df[datacolor[i]].values
    yQr = df['q'+datacolor[i]].values
    cross_length_yP = df['std'+stdcolor[i]].values
    cross_length_yQ = df['qstd'+stdcolor[i]].values
    plt.xlabel('Height (nm)')
    plt.ylabel('Intensity of Color1')
    plt.xlim(0, max(xP) + 10)
    plt.ylim(0, max(max(yPr), max(yQr)) + 10)
    plt.grid(True)

    colorP='black'
    if i<3: colorQ=datacolor[i].lower()
    elif i==3: colorQ='gray'
    for x_val, y_val, len_x, len_y in zip(xP, yPr, cross_length_xP,
        cross_length_yP):
        plt.plot([x_val - len_x, x_val + len_x], [y_val, y_val], color=
            colorP, linewidth=1)
        plt.plot([x_val, x_val], [y_val - len_y, y_val + len_y], color=
            colorP, linewidth=1)
    for x_val, y_val, len_x, len_y in zip(xQ, yQr, cross_length_xQ,
        cross_length_yQ):
```

```
        plt.plot([x_val - len_x, x_val + len_x], [y_val, y_val], color=
            colorQ, linewidth=1)
        plt.plot([x_val, x_val], [y_val - len_y, y_val + len_y], color=
            colorQ, linewidth=1)
    plt.title('Leveled reflected Intensity '+datacolor[i])
plt.tight_layout()
plt.show()
```

Listing 4: Dataplots RGB values, specific for $FePS_3$, similar for $NbSe_2$

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

material= 'NbSe2'
def damped_sinusoid(t, A, lambda_, omega, phi, C):
    return A * np.exp(-t/lambda_) * np.cos((2*np.pi)/omega * t + phi) +
        C
if material == 'FePS3':
    p00=[10,10,100,0,0]
else:
    p00=[10,5,10,0,0]
np.polynomial.set_default_printstyle('ascii')

excel_file = f"{material} Contrasts.xlsx"
df = pd.read_excel(excel_file)
datacolor = 'R','G','B', 'A'
colors= 'red','green','blue','gray'
x = df['Heighttt'].values

plt.figure(figsize=(10,8))
for i in range(len(datacolor)):
    plt.subplot(2,2,i+1)
    yr = df['Con'+datacolor[i]].values
    cross_length_x = df['cstdH'].values
    cross_length_y = df['cstd'+datacolor[i]].values
    popt, pcov = curve_fit(damped_sinusoid, x, yr, p0=p00)
    A_fit, lambda_fit, omega_fit, phi_fit, C_fit = popt
    print("Fit function"+str(datacolor[i])+":")
    print(f'Gefittete Parameter: A = {A_fit}, lambda = {lambda_fit},
        omega = {omega_fit}, phi = {phi_fit}, C = {C_fit}')
    print(f'{A_fit}e^-t/{lambda_fit} cos(2\pi/{omega_fit}t +{phi_fit})
        +{C_fit}')
    #Calculate R-squared (R^2)
    y_pred = damped_sinusoid(x, *popt)
    ss_total = ((yr - np.mean(yr)) ** 2).sum()
    ss_res = ((yr - y_pred) ** 2).sum()
    r_squared = 1 - (ss_res / ss_total)
    r_squared = int(r_squared*10000)/10000
    print(f'R^2 is: {r_squared}')

    x_fit = np.linspace(min(x), max(x), 500)
    y_fit =damped_sinusoid(x_fit, *popt)
    plt.title('Contrast for '+ colors[i] + ' channel')
    plt.plot(x_fit, y_fit, label='Sinusoidal Fit', color='red')
    plt.xlabel('Height (nm)')
    plt.ylabel('Contrast')
```

```
    plt.xlim(0, min(max(x)+10,217))
    plt.ylim(min(yr)-0.5,max(yr)+0.5)
    plt.grid(True)
    plt.plot([0,max(x) + 10],[0,0], color='black',linestyle='-')

    if material == 'FePS3':
        for x_val, y_val, len_x, len_y in zip(x[77:], yr[77:],
            cross_length_x[77:], cross_length_y[77:]):
            plt.plot([x_val - len_x, x_val + len_x], [y_val, y_val],
                color='black', linewidth=1)
            plt.plot([x_val, x_val], [y_val - len_y, y_val + len_y],
                color='black', linewidth=1)
        for x_val, y_val, len_x, len_y in zip(x[0:77], yr[0:77],
            cross_length_x[0:77], cross_length_y[0:77]):
            plt.plot([x_val - len_x, x_val + len_x], [y_val, y_val],
                color=colors[i], linewidth=1)
            plt.plot([x_val, x_val], [y_val - len_y, y_val + len_y],
                color=colors[i], linewidth=1)
    else:
        for x_val, y_val, len_x, len_y in zip(x, yr, cross_length_x,
            cross_length_y):
            plt.plot([x_val - len_x, x_val + len_x], [y_val, y_val],
                color=colors[i], linewidth=1)
            plt.plot([x_val, x_val], [y_val - len_y, y_val + len_y],
                color=colors[i], linewidth=1)

    plt.annotate(f'$r^2$={r_squared}', (0.05, 0.93), xycoords='axes
        fraction', fontsize=12, color='green')
plt.tight_layout()

for i in range(len(datacolor)):
    plt.figure(5)
    yr = df['Con'+datacolor[i]].values
    yg = df['ConG'].values
    yb = df['ConR'].values
    popt, pcov = curve_fit(damped_sinusoid, x, yr, p0=p00)
    A_fit, lambda_fit, omega_fit, phi_fit, C_fit = popt
    x_fit = np.linspace(5, max(x), 500)
    y_fit = damped_sinusoid(x_fit, *popt)
    plt.plot(x_fit, y_fit, label='Polynomial Fit', color=colors[i])
    plt.xlim(0,min(max(x)+10,220))
    plt.ylim(-2,1.5)
plt.show()
```

Listing 5: Dataplots Contrast values, for both materials, change line 6

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import curve_fit
import pandas as pd

R =38.18 #Input value Red
G= 134.42 #Input value Green
B=256 #Input value Blue
SubR=91  #Red channel Substrate
SubG=99   #Green channel Substrate
SubB=231  #Blue channel Substrate
mode = 'average' #or 'maximum'
```

```
cutoff=6  #how many units of light intensity above minimal difference
    will be accepted

excel_file = "FePS3 Contrasts.xlsx"
thinnest_layer=1E-9
max_layers= 200
df = pd.read_excel(excel_file)
x = df['Heighttt'].values
yr = df['ConR'].values
yg =df['ConG'].values
yb =df['ConB'].values

def damped_sinusoid(t, A, lambda_, omega, phi, C):
    return A * np.exp(-lambda_ * t) * np.cos(omega * t + phi) + C
#this script does the fitting aswell, can be replaced by reference
    function.
popt_r, pcov = curve_fit(damped_sinusoid, x, yr, p0=[10, 1E-3, 1E-2, 0,
    0])
popt_g, pcov = curve_fit(damped_sinusoid, x, yg, p0=[10, 1E-3, 1E-2, 0,
    0])
popt_b, pcov = curve_fit(damped_sinusoid, x, yb, p0=[10, 1E-3, 1E-2, 0,
    0])

def Red_measured_FePs3 (x):
    r_conmes= damped_sinusoid(x, *popt_r)
    if r_conmes >1: r_conmes =1
    return (1-r_conmes)*SubR
def Green_measured_FePs3 (x):
    r_conmes= damped_sinusoid(x, *popt_g)
    if r_conmes >1: r_conmes =1
    return  (1-r_conmes)*SubG
def Blue_measured_FePs3 (x):
    r_conmes= damped_sinusoid(x, *popt_b)
    if r_conmes >1: r_conmes =1
    if x<25: return 255
    else:
        return (1-r_conmes)*SubB

colorr=[]
colorg=[]
colorb=[]
colorav=[]
for i in range(max_layers):
    colorr.append(abs(Red_measured_FePs3(thinnest_layer*i*1e9)-R))
    colorg.append(abs(Green_measured_FePs3(thinnest_layer*i*1e9)-G))
    colorb.append(abs(Blue_measured_FePs3(thinnest_layer*i*1e9)-B))
    if mode =='average':
        colorav.append((colorr[i]+colorg[i]+colorb[i])/3)
    elif mode == 'maximum':
        colorav.append(max(colorr[i], colorg[i], colorb[i]))

minval = min(colorav)

indices = [i for i, x in enumerate(colorav) if minval + cutoff>= x >=
    minval]
print(f"The indices of {minval} are: {indices}")

if R>255: R=255
```

```
if G>255: G=255
if B>255: B=255
rgb_colors= np.array([[int(R),int(G),int(B)]])
fig, ax = plt.subplots(figsize=(6, 3.6))
ax.imshow(rgb_colors[np.newaxis, :, :], aspect='auto', extent=[0, len(
    rgb_colors), 0, 1])
ax.set_xticks([])
ax.set_yticks([])

plt.figure(dpi=150)
x=np.linspace(0, max_layers*thinnest_layer*1e9, max_layers)

plt.plot(x,colorr, color = 'red', linewidth = 0.3)
plt.plot(x,colorg, color = 'green', linewidth = 0.3)
plt.plot(x,colorb, color = 'blue', linewidth = 0.3)
plt.plot(x,colorav, color = 'black')
plt.plot(x, [minval+cutoff for _ in range(max_layers)], color= 'black',
     linewidth = 1)
plt.xlabel('Thickness (nm)')
plt.ylabel('Difference')
plt.title('Difference to RGB Values')
plt.show()
```

Listing 6: Application for single RGB value

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import curve_fit
import pandas as pd


thinnest_layer=1E-9
max_layers= 200
SubR=91  #Red channel Substrate
SubG=99   #Green channel Substrate
SubB=232  #Blue channel Substrate
cutoff=25  #units of light intensity above minimal difference accepted
excel_file = "FePS3 Contrasts.xlsx" #fitting is done here, can be
    replaced by functions
rgb_file = "FePS3 RGB absolut application leveled.xlsx" #for raw values
    , "FePS3 RGB absolut_raw.xlsx"

df = pd.read_excel(excel_file)
dRGB = pd.read_excel(rgb_file)
x = df['Heighttt'].values
yr = df['ConR'].values
yg =df['ConG'].values
yb =df['ConB'].values
tot_sampleH = dRGB['HeightQ'].values
tot_sampleR = dRGB['qRed'].values
tot_sampleG = dRGB ['qGreen'].values
tot_sampleB = dRGB ['qBlue'].values

def damped_sinusoid(t, A, lambda_, omega, phi, C):
    return A * np.exp(-lambda_ * t) * np.cos(omega * t + phi) + C
popt_r, pcov = curve_fit(damped_sinusoid, x, yr, p0=[10, 1E-3, 1E-2, 0,
     0])
popt_g, pcov = curve_fit(damped_sinusoid, x, yg, p0=[10, 1E-3, 1E-2, 0,
```

```
    0])
popt_b, pcov = curve_fit(damped_sinusoid, x, yb, p0=[10, 1E-3, 1E-2, 0,
    0])
def Red_measured_FePs3 (x):
    r_conmes= damped_sinusoid(x, *popt_r)
    if r_conmes >1: r_conmes =1
    return (1-r_conmes)*SubR
def Green_measured_FePs3 (x):
    r_conmes= damped_sinusoid(x, *popt_g)
    if r_conmes >1: r_conmes =1
    return  (1-r_conmes)*SubG
def Blue_measured_FePs3 (x):
    r_conmes= damped_sinusoid(x, *popt_b)
    if r_conmes >1: r_conmes =1
    if x<25: return 255
    else:
        return (1-r_conmes)*SubB

u_tot, umax_tot, e_tot, emax_tot, f_tot, fmax_tot = [],[],[],[],[],[]
ff_tot, ffmax_tot=[],[]
g_tot,gmax_tot=[],[]
guesses_tot,guessesmax_tot=[],[]
for cutof in range(cutoff+1):
    u, umax=0,0 #correctly guessed thicknesses
    e,emax=0,0 #number of correctly identified thicknesses
    f,fmax=0,0 #number of exactly identified thicknesses
    g,gmax=0,0
    ff,ffmax=0,0
    data=[]
    datamax=[]
    for y in range(len(tot_sampleH)):
        indices = []
        colorr=[]
        colorg=[]
        colorb=[]
        colorav=[]
        colormax=[]
        ffcheck=0
        ffcheckmax=0
        for i in range(max_layers):
            colorr.append(abs(Red_measured_FePs3(thinnest_layer*i*1e9)-
                tot_sampleR[y]))
            colorg.append(abs(Green_measured_FePs3(thinnest_layer*i*1e9
                )-tot_sampleG[y]))
            colorb.append(abs(Blue_measured_FePs3(thinnest_layer*i*1e9)
                -tot_sampleB[y]))
            colorav.append((colorr[i]+colorg[i]+colorb[i])/3)
            colormax.append(max(colorr[i], colorg[i], colorb[i]))
        minval = min(colorav)
        minmaxval = min(colormax)
        indices = [i for i, x in enumerate(colorav) if minval + cutof>=
            x >= minval]
        indicesmax = [i for i, x in enumerate(colormax) if minmaxval +
           cutof>= x >= minmaxval]

        for i in indices:
            data.append(i)
        for i in indicesmax:
```

```
82              datamax.append(i)
83          if int(tot_sampleH[y])-2 in indices or int(tot_sampleH[y])+2 in
                indices:
84              g=g+1
85              ffcheck=1
86          if int(tot_sampleH[y])-2 in indicesmax or int(tot_sampleH[y])+2
                in indicesmax:
87              gmax=gmax+1
88              ffcheckmax=1
89          '''     if int(tot_sampleH[y])-3 in indices or int(tot_sampleH[y
                ])+3 in indices:
90              g=g+1
91          if int(tot_sampleH[y])-3 in indicesmax or int(tot_sampleH[y])+3
                in indicesmax:
92              gmax=gmax+1
93
94          if int(tot_sampleH[y])-4 in indices or int(tot_sampleH[y])+4 in
                indices:
95              g=g+1
96          if int(tot_sampleH[y])-4 in indicesmax or int(tot_sampleH[y])+4
                in indicesmax:
97              gmax=gmax+1
98
99          if int(tot_sampleH[y])-5 in indices or int(tot_sampleH[y])+5 in
                indices:
100             g=g+1
101         if int(tot_sampleH[y])-5 in indicesmaxor int(tot_sampleH[y])+5
                in indicesmax:
102             gmax=gmax+1
103
104         if int(tot_sampleH[y])-6 in indices or int(tot_sampleH[y])+6 in
                indices:
105             g=g+1
106         if int(tot_sampleH[y])-6 in indicesmax or int(tot_sampleH[y])+6
                in indicesmax:
107             gmax=gmax+1         '''
108         if int(tot_sampleH[y]) in indices:
109             f=f+1
110             ffcheck=1
111             if int(tot_sampleH[y])-1 in indices:
112                 if int(tot_sampleH[y])+1 in indices :
113                     u= u+3
114                     e = e+1
115                 else:
116                     u= u+2
117                     e = e+1
118
119             elif int(tot_sampleH[y])+1 in indices:
120                 u=u+2
121                 e = e+1
122             else:
123                 u=u+1
124                 e = e+1
125         elif int(tot_sampleH[y])-1 in indices or int(tot_sampleH[y])+1
                in indices:
126             ffcheck=1
127             u=u+1
128             e = e+1
```

```
129         if int(tot_sampleH[y]) in indicesmax:
130             fmax=fmax+1
131             ffcheckmax=1
132             if int(tot_sampleH[y])-1 in indicesmax:
133                 if int(tot_sampleH[y])+1 in indicesmax :
134                     umax= umax+3
135                     emax = emax+1
136                 else:
137                     umax= umax+2
138                     emax = emax+1
139             elif int(tot_sampleH[y])+1 in indicesmax:
140                 umax=umax+2
141                 emax = emax+1
142             else:
143                 umax=umax+1
144                 emax = emax+1
145         elif int(tot_sampleH[y])-1 in indicesmax or int(tot_sampleH[y])
               +1 in indicesmax:
146             umax=umax+1
147             emax = emax+1
148             ffcheckmax=1
149 
150         if ffcheck ==1: ff=ff+1
151         if ffcheckmax ==1: ffmax=ffmax+1
152     g=g+u
153     gmax=gmax+umax
154         #if int(tot_sampleH[y]) in indicesmax or int(tot_sampleH[y])-1
               in indicesmax or int(tot_sampleH[y])+1 in indicesmax:
155         #    umax = umax+1
156     ''' #creates plots for every cutoff
157     plt.figure(dpi=100)
158     plt.hist(data, bins=range(min(data), max(data) + 1), align='left',
           edgecolor='black', label = 'Predicted Thickness')
159     plt.hist(tot_sampleH, bins=range(int(min(tot_sampleH)), int(max(
           tot_sampleH)) + 1), align='left', edgecolor='red', label = '
           Measured Thickness')
160     plt.legend()
161     plt.xlabel('Thickness (nm)')
162     plt.ylabel('Frequency')
163     plt.title(f'Using Average Colour Difference and Cutoff {cutof}')
164     plt.show()
165     plt.figure(dpi=100)
166     plt.hist(datamax, bins=range(min(datamax), max(datamax) + 1), align
           ='left', edgecolor='black', label = 'Predicted Thickness')
167     plt.hist(tot_sampleH, bins=range(int(min(tot_sampleH)), int(max(
           tot_sampleH)) + 1), align='left', edgecolor='red', label = '
           Measured Thickness')
168     plt.legend()
169     plt.xlabel('Thickness (nm)')
170     plt.ylabel('Frequency')
171     plt.title(f'Using Maximal Colour Difference and Cutoff {cutof}')
172     plt.show()    '''
173     percentage_av= u/len(data)
174     percentage_max= umax/len(datamax)
175     print(f'Cutoff = {cutof}')
176     print(f'The height of {e} out of {len(tot_sampleH)} samples were
           detected correctly (+-1nm), {f} flakes exactly, with average
           difference. That is {int(percentage_av*1000)/10}% and {int(f/len
```

```
        (data)*1000)/10}% of the indicated thicknesses.')
    print(f'{len(data)} guesses for {len(tot_sampleH)} samples')
    print(f'The height of {emax} out of {len(tot_sampleH)} samples were
        detected exactly (+-1nm), {fmax} flakes exactly,  with maximal
       difference. That is {int(percentage_max*1000)/10}% and {int(fmax
       /len(datamax)*1000)/10}% of the indicated thicknesses.')
    print(f'{len(datamax)} guesses for {len(tot_sampleH)} samples')

    u_tot.append(u) #correctly guessed thicknesses
    umax_tot.append(umax)
    e_tot.append(e) #number of correctly identified thicknesses
    emax_tot.append(emax)
    f_tot.append(f) #number of exactly identified thicknesses
    fmax_tot.append(fmax)
    g_tot.append(g)
    gmax_tot.append(gmax)
    ff_tot.append(ff)
    ffmax_tot.append(ffmax)
    guesses_tot.append(len(data))
    guessesmax_tot.append(len(datamax))

plt.figure(1)
plt.plot([i for i in range(cutoff+1)],e_tot, label='number of correctly
    identified thicknesses')
plt.plot([i for i in range(cutoff+1)],f_tot, label='number of exactly
   identified thicknesses guessed thicknesses')
plt.figure(2)
plt.plot([i for i in range(cutoff+1)],guesses_tot, label='number of
   correctly identified thicknesses')
plt.plot([i for i in range(cutoff+1)],guessesmax_tot, label='number of
   exactly identified thicknesses guessed thicknesses')
plt.figure(figsize=(6,6), dpi=600)
plt.title('Correct thicknesses per cutoff') #Comparison between average
    (av) and maximal (max) difference
plt.plot([i for i in range(cutoff+1)],[g_tot[i]/guesses_tot[i]*100 for
   i in range(len(u_tot))], label='av, $\pm2.5$ nm', color=[0,0,0.45])
plt.plot([i for i in range(cutoff+1)],[u_tot[i]/guesses_tot[i]*100 for
   i in range(len(u_tot))], label='av, $\pm1.5$ nm', color=[0,0,0.75])
plt.plot([i for i in range(cutoff+1)],[f_tot[i]/guesses_tot[i]*100 for
   i in range(len(f_tot))], label='av, $\pm0.5$ nm', color=[0,0,1])
plt.plot([i for i in range(cutoff+1)],[gmax_tot[i]/guessesmax_tot[i
   ]*100 for i in range(len(umax_tot))], label='max, $\pm2.5$ nm',
   color=[0.45,0,0])
plt.plot([i for i in range(cutoff+1)],[umax_tot[i]/guessesmax_tot[i
   ]*100 for i in range(len(umax_tot))], label='max, $\pm1.5$ nm',
   color=[0.75,0,0])
plt.plot([i for i in range(cutoff+1)],[fmax_tot[i]/guessesmax_tot[i
   ]*100 for i in range(len(fmax_tot))], label='max, $\pm0.5$ nm',
   color=[1,0,0])
plt.plot([i for i in range(cutoff+1)],[ff_tot[i]/124*100 for i in range
   (len(f_tot))], label='id. samples (av, $\pm2.5$ nm)', color =
   [0,0.4,0.1])
plt.plot([i for i in range(cutoff+1)],[f_tot[i]/124*100 for i in range(
   len(f_tot))], label='id. samples (av, $\pm0.5$ nm)', color =
   [0,0.8,0.1])
plt.plot([i for i in range(cutoff+1)],[ffmax_tot[i]/124*100 for i in
   range(len(f_tot))], label='id. samples (max, $\pm2.5$ nm)', color =
   [0.55,0.4,0])
```

```
plt.plot([i for i in range(cutoff+1)],[fmax_tot[i]/124*100 for i in
    range(len(f_tot))], label='id. samples (max, $\pm0.5$ nm)', color =
    [0.8,0.4,0])
plt.legend()
plt.ylim(0,100)
plt.xlabel('Cutoff')
plt.ylabel('Ratio in %')
```

Listing 7: Evaluation of application for lists of RGB values, part 1

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import curve_fit
import pandas as pd

thinnest_layer=1E-9
max_layers= 200
SubR=91  #Red channel Substrate
SubG=99   #Green channel Substrate
SubB=232  #Blue channel Substrate
cutoff=25  #units of light intensity above minimal difference accepted
excel_file = "FePS3 Contrasts.xlsx"
rgb_file = "FePS3 RGB absolut application leveled.xlsx"
colormap= plt.cm.viridis_r
norm=plt.Normalize(0,cutoff)

df = pd.read_excel(excel_file)
dRGB = pd.read_excel(rgb_file)
x = df['Heighttt'].values
yr = df['ConR'].values
yg =df['ConG'].values
yb =df['ConB'].values
tot_sampleH = dRGB['HeightQ'].values
tot_sampleR = dRGB['qRed'].values
tot_sampleG = dRGB ['qGreen'].values
tot_sampleB = dRGB ['qBlue'].values
def damped_sinusoid(t, A, lambda_, omega, phi, C):
    return A * np.exp(-lambda_ * t) * np.cos(omega * t + phi) + C
popt_r, pcov = curve_fit(damped_sinusoid, x, yr, p0=[10, 1E-3, 1E-2, 0,
    0])
popt_g, pcov = curve_fit(damped_sinusoid, x, yg, p0=[10, 1E-3, 1E-2, 0,
    0])
popt_b, pcov = curve_fit(damped_sinusoid, x, yb, p0=[10, 1E-3, 1E-2, 0,
    0])
def Red_measured_FePs3 (x):
    r_conmes= damped_sinusoid(x, *popt_r)
    if r_conmes >1: r_conmes =1
    return (1-r_conmes)*SubR
def Green_measured_FePs3 (x):
    r_conmes= damped_sinusoid(x, *popt_g)
    if r_conmes >1: r_conmes =1
    return  (1-r_conmes)*SubG
def Blue_measured_FePs3 (x):
    r_conmes= damped_sinusoid(x, *popt_b)
    if r_conmes >1: r_conmes =1
    if x<25: return 255
    else:
        return (1-r_conmes)*SubB
```

```
plt.figure(dpi=600)
for cutof in range(cutoff+1):
    cutof=cutoff-cutof
    u, umax=0,0 #correctly guessed thicknesses
    e,emax=0,0 #number of correctly identified thicknesses
    f,fmax=0,0 #number of exactly identified thicknesses
    data=[]
    datamax=[]
    for y in range(len(tot_sampleH)):
        indices = []
        colorr=[]
        colorg=[]
        colorb=[]
        colorav=[]
        colormax=[]
        for i in range(max_layers):
            colorr.append(abs(Red_measured_FePs3(thinnest_layer*i*1e9)-
                tot_sampleR[y]))
            colorg.append(abs(Green_measured_FePs3(thinnest_layer*i*1e9
                )-tot_sampleG[y]))
            colorb.append(abs(Blue_measured_FePs3(thinnest_layer*i*1e9)
                -tot_sampleB[y]))
            colorav.append((colorr[i]+colorg[i]+colorb[i])/3)
            colormax.append(max(colorr[i], colorg[i], colorb[i]))
        minval = min(colorav)
        minmaxval = min(colormax)
        indices = [i for i, x in enumerate(colorav) if minval + cutof>=
            x >= minval]
        indicesmax = [i for i, x in enumerate(colormax) if minmaxval +
            cutof>= x >= minmaxval]
        print(f"The indices of {minmaxval} are: {indices}. The actual
            Height is {tot_sampleH[y]}")
        for i in indices:
            data.append(i)
        for i in indicesmax:
            datamax.append(i)
        if int(tot_sampleH[y]) in indices:
            f=f+1
            if int(tot_sampleH[y])-1 in indices:
                if int(tot_sampleH[y])+1 in indices :
                    u= u+3
                    e = e+1
                else:
                    u= u+2
                    e = e+1
            elif int(tot_sampleH[y])+1 in indices:
                u=u+2
                e = e+1
            else:
                u=u+1
                e = e+1
        elif int(tot_sampleH[y])-1 in indices or int(tot_sampleH[y])+1
            in indices:
            u=u+1
            e = e+1
        if int(tot_sampleH[y]) in indicesmax:
            fmax=fmax+1
            if int(tot_sampleH[y])-1 in indicesmax:
```

```
                if int(tot_sampleH[y])+1 in indicesmax :
                    umax= umax+3
                    emax = emax+1
                else:
                    umax= umax+2
                    emax = emax+1
            elif int(tot_sampleH[y])+1 in indicesmax:
                umax=umax+2
                emax = emax+1
            else:
                umax=umax+1
                emax = emax+1
        elif int(tot_sampleH[y])-1 in indicesmax or int(tot_sampleH[y])
            +1 in indicesmax:
            umax=umax+1
            emax = emax+1
    plt.hist(datamax, bins=range(min(datamax), max(datamax) + 1), align
        ='left', label = 'Predicted Thickness', color=plt.cm.viridis_r(
        cutof/cutoff))
    percentage_av= u/len(data)
    percentage_max= umax/len(datamax)
    print(f'Cutoff = {cutof}')
    print(f'The height of {e} out of {len(tot_sampleH)} samples were
        detected correctly (+-1nm), {f} flakes exactly, with average
        difference. That is {int(percentage_av*1000)/10}% and {int(f/len
        (data)*1000)/10}% of the indicated thicknesses.')
    print(f'{len(data)} guesses for {len(tot_sampleH)} samples')
    print(f'The height of {emax} out of {len(tot_sampleH)} samples were
         detected exactly (+-1nm), {fmax} flakes exactly,  with maximal
        difference. That is {int(percentage_max*1000)/10}% and {int(fmax
        /len(datamax)*1000)/10}% of the indicated thicknesses.')
    print(f'{len(datamax)} guesses for {len(tot_sampleH)} samples')

plt.hist(tot_sampleH, bins=range(int(min(tot_sampleH)), int(max(
    tot_sampleH)) + 1), align='left', color='red', label = 'Measured
    Thickness')
plt.ylim(0, None)
plt.xlabel('Thickness (nm)')
plt.ylabel('Frequency')
plt.title('Calculated thicknesses using maximal colour difference')
sm = plt.cm.ScalarMappable(cmap=colormap, norm=norm)
sm.set_array([])
plt.colorbar(sm, label='Cutoff')
plt.show()
```

Listing 8: Evaluation of the application for lists of RGB values, part 2 (frequency diagrams). Specific for maximum difference, average difference very similar