UiT The Arctic University of Norway

Faculty of Science and Technology
Department of Physics and Technology

Development of a Multi-angle Illumination Microscope for 3D Refractive Index Reconstruction

Martin Nilsen Stave

FYS-3941 Master's thesis in applied physics and mathematics 30 SP
16th of December 2024

This thesis document was typeset using the *UiT Thesis LaTeX Template*.

Abstract

This thesis presents the design, implementation, and evaluation of a multi-angle illumination microscope tailored for 3D refractive index (RI) reconstruction using the Multi-Slice Beam Propagation (MSBP) method. The system combines custom-built hardware, including precision-aligned optical components and motorized kinematic mirrors, with computational algorithms developed in Python to achieve high-resolution imaging of complex biological samples. By calibrating illumination angles through Fourier ptychography and optimizing experimental parameters, the system unlocks the potential for reconstruction of 3D RI distributions from intensity measurements.

The developed methodology addresses challenges in adapting the MSBP method for real-world applications, such as alignment precision, system calibration, and data processing. Experimental validation is performed using bead samples and a biological specimen (Peace Lily pollen), demonstrating the system's capability to reconstruct detailed 3D RI maps. However, it was ultimately found that the iteration amount needed to be optimized to achieve better reconstruction quality. Thus the results underscore the potential of this approach for advancing label-free, high-resolution optical microscopy, while highlighting areas of improvement, paving the way for its application to complex, multiple-scattering biological samples.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisors, Krishna Agarwal, Farhad Niknam, Jean-Claude Tinguely, and Ankit Butola, for their invaluable guidance and support throughout this project. Your consistent feedback during our weekly discussions has greatly enhanced my ability to conduct rigorous scientific research and analysis, which has been essential to completing this work and writing this paper.

Personal Growth

After completing a course called Waves and Optics, my curiosity sparked and I decided to do a Master's thesis in Optics. Initially, as I approached the pre-project, I had little to no expectations. During this phase, I learned the importance of persistence: how to commit to a project, work diligently, and see it through to the end. After completing the pre-project, I began constructing the optical setup for my Master Thesis. I quickly realized that the process was far from straightforward. Each obstacle demanded its own solution, and tackling them one by one helped me develop practical skills and problem-solving techniques. In the end, I could rebuild the entire microscopic setup in a fraction of the time it initially took, a reflection of how much I had grown. The thesis also taught me an invaluable lesson about progress. It is tempting to imagine a clear linear path from start to finish, but the reality is much more intricate. Progress often feels like navigating a chaotic spiral, full of unexpected twists and challenges. Each step brings new hurdles, and the endpoint only becomes clear after overcoming them all.

This experience has reshaped my perspective. Meaningful projects rarely follow a predictable route; instead, they open doors to opportunities and insights that you never anticipated. By staying resilient and embracing uncertainty, you grow in ways you could simply not have imagined. Completing this thesis has not only made me a better engineer and problem-solver, but also strengthened my character. It has prepared me to face any challenge with confidence, no matter how complex the path ahead may seem.

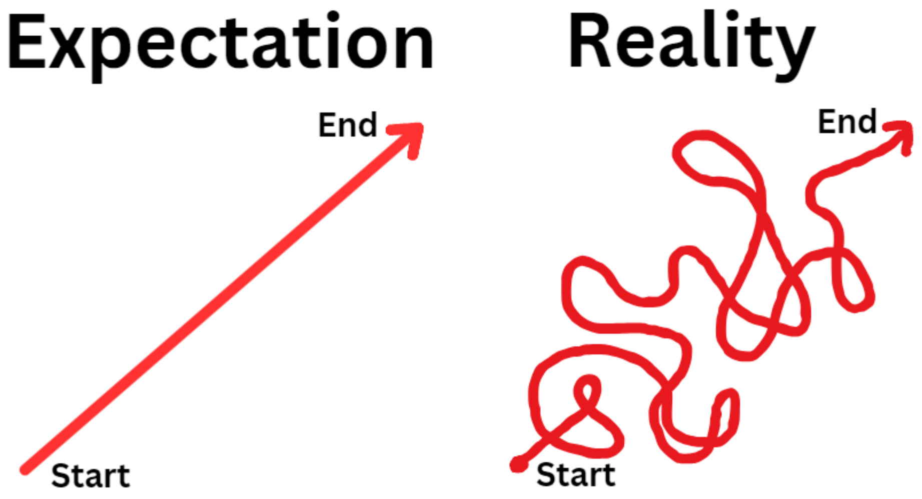


Figure 1: Expectation Vs. Reality.

Once again, thank you for your support. Without your encouragement and expertise, this achievement would not have been possible.

Statement on Prior Work and AI Assistance

ChatGPT was used as a tool throughout the process of writing this thesis to summarize articles, analyze code, and format tables. Given that ChatGPT is simply a language model, this tool has not been fully relied upon but has simply served as an introductory resource to aid the author's understanding and providing digestible summaries of complex concepts. All information from ChatGPT was further researched and verified using reliable sources.

This Master Thesis has been written as an extension of the author's *preproject paper* [Stave, Martin Nilsen. (2024). Computational 3D Reconstruction of the Refractive Index Distribution - UIT Prosjekt Oppgave]. Therefore, this thesis will have similar subject matter and structure. However, everything is rewritten/recontextualized for a bigger scope of reference. The preproject thesis can be found in its entirety in appendix C.

Figure 2: My workstation throughtout this project.

Contents

List of Figures

List of Listings

List of Abbreviations

API Application Programming Interface

DU Device Units

FWHM Full Width at Half Maximum

GLIM Gradient Light Interference microscopy

GPU Graphics Processing Unit

KM1 Kinematic Mirror 1

KM2 Kinematic Mirror 2 (Motorized)

KM3 Kinematic Mirror 3 (90°)

KM4 Kinematic Mirror 4 (90°)

L1 Collimation Lens

L2 Focusing Lens

LED Light Emitting Diode

MSBP Multi-Slice Beam Propagation

NA Numerical Aperture

ODT Optical Diffraction Tomography

QPI Quantitative Phase Imaging

RI Refractive Index

UI User Interface

/1

Introduction

Background, Motivation and Expectation

The field of optical microscopy allows us to visualize things that are invisible to the human eye. Using microscopy techniques, we can capture intricate details of microscopic structures at the nanometer scale. The ability to reconstruct three-dimensional refractive index distributions of biological samples is crucial to advance our understanding of cellular structures and dynamics. One method that has shown great promise for this purpose is the multi-slice beam propagation method, as developed by *Chowdhury et al.* [1]. This technique, based on multi-angle illumination, allows for high-resolution 3D refractive index reconstruction from intensity images, offering significant improvements over traditional imaging methods.

Although the method is well-documented, adapting it to a functional microscope system presents significant challenges. The original work provides a solid foundation, but leaves some practical issues unaddressed, as is often the case with proprietary "know-how." Thus, substantial challenges remain when it comes to adapting it to a functional real-world microscope system.

The goal of this thesis is to address these challenges by implementing the multi-slice beam propagation method into my very own practical experimental setup. In doing so, this work not only recreates the theoretical framework in Python but also builds a microscope capable of multi-angle illumination to capture and reconstruct 3D refractive index distributions. Along the way, various technical hurdles and practical questions emerge, such as hardware calibration, system optimization, and data processing, many of which are not explicitly covered in the original research. By tackling these issues, this thesis contributes to the field by providing a deeper understanding of how such methods can be applied in practice, offering solutions that could guide future implementations of similar systems.

2

Background

To address the challenges of implementing the multi-slice beam propagation method, it is essential to first understand the foundational principles of 3D refractive index reconstruction. Central to this is optical diffraction tomography (ODT), which provides the theoretical basis for reconstructing refractive-index distributions from intensity data. This section begins with an overview of ODT, explaining the interaction of light with biological samples. It then covers the multi-slice beam propagation method, detailing its key steps and calculations.

2.1 Optical Diffraction Tomography

Optical diffraction tomography is a three-dimensional (3D) imaging technique that reconstructs the 3D refractive index (RI) distribution of a sample by illuminating it with light from multiple angles and analyzing the resulting scattered waves [1][2]. Unlike conventional microscopy, which relies on on-axis light propagation, ODT leverages the diffraction of light as it interacts with the sample from multiple directions. This approach improves optical sectioning and enables visualization of subcellular structures that are difficult to observe with on-axis propagation alone [3].

ODT utilizes *Quantitative Phase Imaging* (QPI) principles to capture 2D intensity images at various illumination angles, which are later processed to extract both amplitude and phase information essential for accurate 3D reconstruction. Because light behaves as an electromagnetic wave, it can be described by amplitude (brightness) and phase, where the phase represents the position of a point within a wave cycle [4]. When light passes through a transparent object, its speed changes because of the object's refractive index (RI), which in turn causes a shift in the light wave's phase. A phase image is a 2D representation of these phase shifts across the field of view, with brighter regions typically corresponding to areas where the light wave has experienced a larger phase shift. These shifts often indicate a higher refractive index or a thicker part of the sample. Using the information collected from the quantitative phase imaging, we can tomographically reconstruct a map detailing the RI distribution of the sample [1][4].

2.1.1 Scattering Approximations

In optical diffraction tomography, the way light scatters as it interacts with a sample can be categorized into two main types: weak scattering and multiple scattering. Weak scattering refers to an approximation where incident light is assumed to scatter only once as it passes through the sample, simplifying the complex interactions that actually occur [4].

This simplifies the analysis of the scattered field and allows the use of approximations, such as the Born or Rytov approximations [3], to reconstruct the sample's refractive index (RI) distribution. These approximations linearize the scattering problem, making it computationally easier to solve [1][2][3][5]. However, these approximations are only valid when the sample is weakly scattering and the phase variations introduced by the sample are small [3]. For thicker, denser samples, where light undergoes multiple scattering events, these approximations break down, leading to inaccurate reconstructions. In such cases, more computationally intensive techniques that account for multiple scattering, such as the aforementioned Multi-slice Beam Propagation (MSBP) method are required.

2.2 Multi-Slice Beam Propagation Forward Model

Section 2.2 *and* 2.3 *refer to similar sections of the candidate's pre-project paper [6], which can also be found in Appendix C of this thesis.*

The high resolution 3D RI microscopy technique utilizes MSBP to reconstruct the 3D RI of multiple scattering samples, allowing it to image a wider range of biological samples, including those that are dense or have complex structures [1]. This approach acquires intensity-only measurements from different illumination angles and solves a non-linear optimization problem to recover the 3D RI distribution. By not relying on weak scattering approximations, this technique can achieve high-resolution imaging even in samples that exhibit significant multiple scattering.

The MSBP method simulates the propagation of light through a 3D object by dividing it into a series of thin layers, each of uniform thickness Δz. The surrounding medium is assumed to be homogeneous with a constant RI (n_m) to simplify the calculations [1].

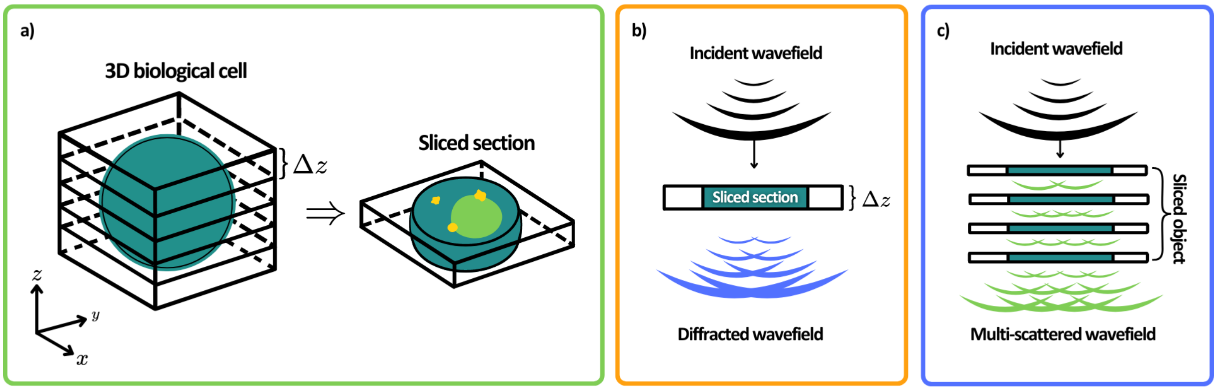


Figure 2.1: a) A section of the discretized 3D biological cell. b) Diffraction of an incident wave-field passing through a single slice. c) Multi-scattering of the incident wave-field passing from layer to layer.

The propagation of light through the sample is modeled as a sequential process. The incident wavefront interacts with the first layer and is diffracted. The diffracted field then becomes the incident field for the next layer. Each layer acts as a complex transmittance function determined by its RI distribution. The transmitted electric field at each layer is calculated recursively as the light propagates through the object. The amplitude of the electric field at the kth layer, denoted as $y_k(r)$ is defined as:

$$y_k(r) = t_k(\mathbf{r}) * P_{\Delta z}\{y_{k-1}(\mathbf{r})\}, \tag{2.1}$$

where $t_k(\mathbf{r})$ represents the complex transmittance of the kth layer, and $P_{\Delta z}$ is the propagation operator. The complex transmittance $t_k(r)$ is defined as:

$$t_k(\mathbf{r}) = \exp(j(\frac{2\pi}{\lambda})\Delta z(n_k(r) - n_m)),$$

where n_k is the complex-valued RI of the kth-layer and n_m is the homogeneous refractive index of the surrounding medium, λ is the wavelength of light and Δz is the thickness of each layer. The propagation operator $P_{\Delta z}$, which advances the electric field by a distance Δz for each layer, is defined as:

$$P_{\Delta z}\{\cdot\} = F^{-1}\left\{\exp\left(-j\Delta z\left(\left(\frac{2\pi}{\lambda}\right)^2 - |\mathbf{k}|^2\right)^{1/2}\right)F\{\cdot\}\right\}, \tag{2.2}$$

where $F\{\cdot\}$ and $F^{-1}\{\cdot\}$ are the 2D Fourier and inverse Fourier transforms, respectively. (For more details, see the Appendix C). To initialize Eq. (2.1), an incident planar electric field is applied to the sample at a specific angle. This incident field is defined as $y_0(\mathbf{r}) = \exp(j\mathbf{k}_0 \cdot \mathbf{r})$ where $\mathbf{k}_0$ is the incident wave vector of the 2D illumination on the first layer. As light propagates through the object, the electric field evolves, and at the final layer ($k = N$) the field $y_N(r)$ contains the accumulated diffraction effects from multi-scattering within the object. The electric field is then backpropagated to the sensor plane, located in the center of the object. The forward model predicts the electric field distribution at the sensor plane as:

$$E(\mathbf{r}) = F^{-1}\{p(\mathbf{k}) * F\{P_{-z}\{y_N(\mathbf{r})\}\}\}, \tag{2.3}$$

where p($\mathbf{k}$) represents the pupil function of the system, which mathematically models how light propagates through the optical system. For an ideal microscope, the pupil function is approximated as a circular aperture with a radius equal to the cutoff frequency of the system, which defines its resolution. The parameter z denotes the distance from the final object layer $y_N(r)$ to the plane within the object volume that is conjugate to the focal plane of the imaging systems. When the system is focused on the center of the object, $z = \Delta z N/2$.

The complete forward model can be represented by the operator $G\{n\}$, which transforms a 3D complex-valued refractive index distribution into a 2D complex-valued electric field, $\mathbb{C}^3 \rightarrow \mathbb{C}^2$. This operator encapsulates the entire propagation process, predicting the 2D field at the sensor plane after illuminating $n(r)$ with the incident field $y_0^l(\mathbf{r})$, as described in Eqs. (2.1) and (2.3). Thus, it captures the cumulative effect of all layers in producing the observed electric field in the microscope. The final intensity can then be expressed as

$$I(\mathbf{r}) = |E(\mathbf{r})|^2, \tag{2.4}$$

where $I(r)$ is the intensity distribution and $E(r)$ represents the complex electric field amplitude in the sensor plane.

2.3 Inverse Reconstruction Model

The inverse model reconstructs the 3D refractive index distribution of the sample through an iterative process that refines an estimate of the sample's RI based on the difference between the diffraction patterns acquired by the microscope system and the diffraction pattern created by the reconstructed sample object.

Initially, the forward model simulates the light-scattering process for a given estimate of the sample's 3D RI distribution. To achieve this, it applies a series of incident electric fields, denoted $y_0^l(r) = \exp(j\mathbf{k}_0^l \cdot r)$ for l = 1, 2, ..., L, each corresponding to a specific illumination angle determined by its wave vector $\mathbf{k}_0^l$. The forward model predicts the diffraction patterns that would be observed in the detector plane for each illumination angle, providing a set of expected intensity measurements denoted as $I^l(\mathbf{r})$.

The inverse model utilizes the measured amplitudes (square root of intensity measurements, $\sqrt{I^l(r)}$) and the predictions of the forward model $G^l\{\mathbf{r}\}$ to iteratively refine its estimate of the 3D RI distribution of the sample. This refinement process involves minimizing the least squares difference between the measured and predicted amplitudes. By adjusting the RI distribution, the inverse model seeks to align the diffraction patterns of the reconstructed sample with those obtained from the actual sample. Thus, the estimated RI distribution of the 3D sample is defined as:

$$n^*(\mathbf{r}_{3D}) = \arg \min_{n(\mathbf{r}_{3D})} \sum_{l=1}^{L} \sum_{r} |\sqrt{I^l(\mathbf{r})} - |G^l\{n(\mathbf{r}_{3D})\}||^2, \tag{2.5}$$

where $\mathbf{r}_{3D} = \langle \mathbf{r}, k \rangle$ is the 3D spatial position vector defining a specific point within the 3D space of the sample, where $\mathbf{r}$ represents the 2D lateral coordinates and k refers to the layer index along the axial direction. The term $n(\mathbf{r}_{3D}) = n_k(\mathbf{r})$ for layers $k = 1, 2, 3, ..., N$, is essentially a collection of 2D RI distributions $n_k(\mathbf{r})$ that highlights how the reconstruction process builds the 3D RI map by combining information from each layer. And $G^l\{\mathbf{r}\}$ is the forward model mentioned above, taking the 3D RI distribution as input and outputs the predicted amplitude distribution in the detector plane.

Finally, a complex-valued quantity $n^*(\mathbf{r}_{3D})$ is obtained which represents the 3D complex transmittance of the objective. This transmittance can be used to derive the refractive index distribution of the sample, capturing both the phase and absorption characteristics. The primary objective of the inverse model is to reconstruct a 3D RI distribution that accurately represents the structure of the sample. Once the iterative process converges, the resulting reconstructed RI distribution provides a quantitative 3D map of the optical properties of the sample. This map can then be visualized and analyzed to gain insight into the morphology, composition, and internal features of the sample.

Part I

Methodology

3

Illumination System and Alignment

The methodology of this thesis is centered around the design, implementation and alignment of a microscope system tailored for 3D refractive index microscopy, utilizing ODT. This section provides an overview of the design of the system, with more detailed discussions on each component and the steps that were taken to complete the experimental setup. Figure 3.1 shows the experimental setup in its entirety. The system was designed, constructed, and automated during the fall semester of 2024 in a laboratory located in the basement of the Teknologibygget at UiT.

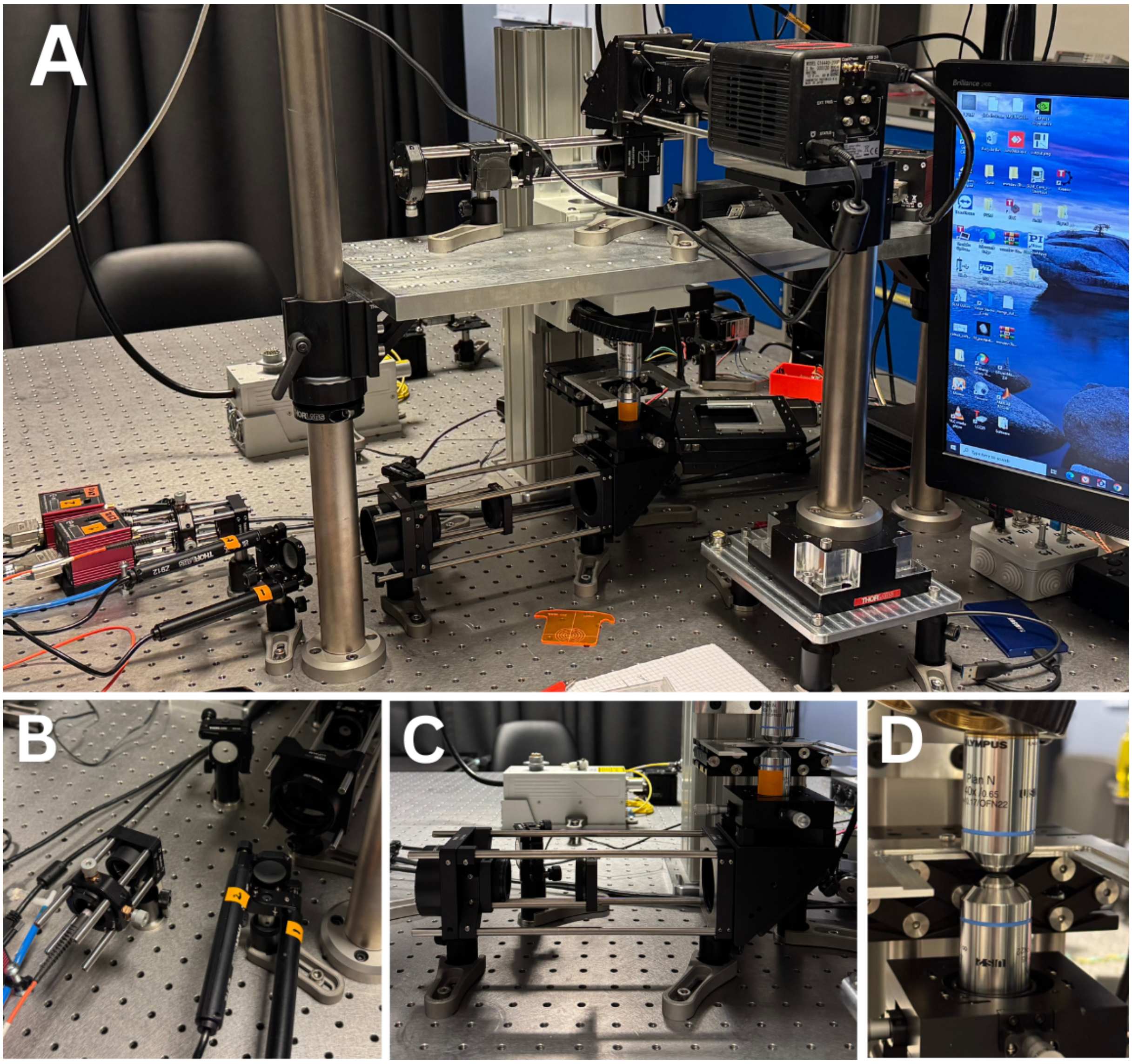


Figure 3.1: A) Experimental Setup, B) Light source and kinematic mirrors, C) Focusing lens and pinholes, D) Objective lenses.

3.1 Overview of System Design

Figure 3.2 details the system design of my experimental setup. Each component has been marked for reference throughout the text. My experimental setup is divided into three parts. First, the illumination system, described in Section 3, details how light from the source is guided to the sample using both theoretical and experimental approaches. Next, Section 4 explains how the system achieves precise light angling in the sample plane and captures images. Finally, Section 5 covers the post-processing steps used to reconstruct the refractive index distribution from the acquired images. In the following section, each of these components will be discussed in detail, including the rationale behind their selection and the methods used to determine their optimal placement within the system.

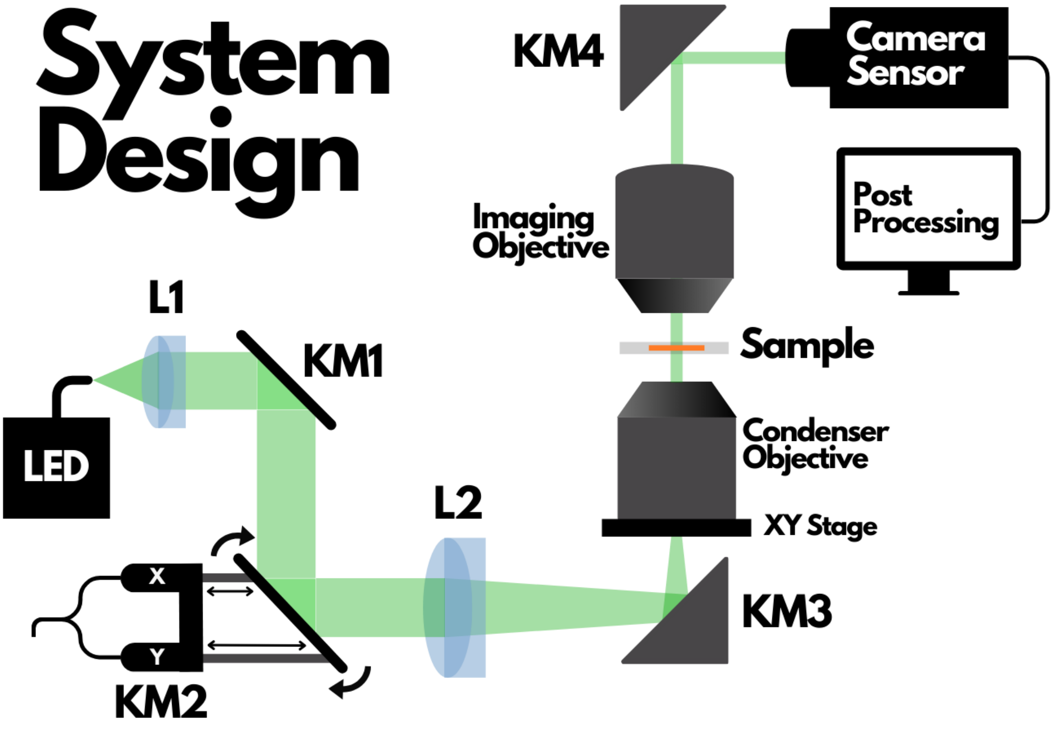


Figure 3.2: System Design Components: LED, Collimation lens (L1), Kinematic mirror (KM1), Motorized kinematic mirror (KM2), Focusing lens (L2), Kinematic mirror (KM3), Condenser objective on XY-axis stage, Sample placed on z-axis stage, Imaging objective on z-axis stage, Kinematic mirror (KM4), Camera sensor.

3.2 Light Source

The starting point of the experimental setup is the illumination source. A fiber-coupled LED light source (λ = 530nm, Thorlabs M530F2) is used to illuminate the sample. The LED source on its own is very divergent. Therefore, I coupled the LED light to a fiber cable with a 50µm fiber core (Thorlabs M14L01). Given the nature of this setup, it is important to have efficient control over the illumination. Additionally, the beam size should be broad enough to ensure uniform illumination on the back focal plane of the condenser objective lens. Initially, this configuration was tested using a 400µm wide fiber core. However, I quickly determined that the resulting beam size was too large for the intended application. Consequently, I opted for the 50µm fiber core, which proved to be more suitable.

Choosing an LED light source over a laser primarily comes down to the low coherence of the LED. Although laser sources are often more cost-effective, their high coherence can introduce speckle and noise into the images. By opting for an LED, it is possible to achieve a better balance between coherence and noise. Having less potential for speckle and noise will ultimately result in a cleaner image [7].

3.3 Beam Collimation

The output beam of the fiber is quite divergent (ideally, the entire intensity of the LED should reach the sample). Therefore, it is necessary to collimate the beam using a lens (LED→ L1).

To effectively collimate the divergent beam emitted from the fiber output, the lens was placed on a post, positioned at a distance from the fiber equivalent to the focal distance of the lens f = 30mm. This ensures that the divergent rays are redirected to become parallel, achieving collimation. By placing the lens approximately 30mm away from the fiber output, the beam becomes less divergent, further adjustments can be made experimentally to ensure correct collimation. Collimation can be observed by placing "measuring paper" close to the lens and noting the beam size, then placing the paper some distance away and noting the beam size again. If the beam width remains the same, the light is collimated.

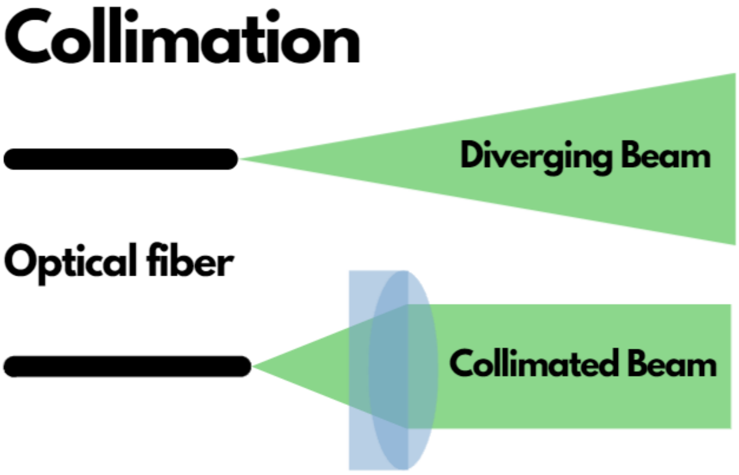


Figure 3.3: Comparison between diverging and collimated beam.

3.4 Kinematic Mirrors

The next part on the light beam's path are two kinematic mirrors (Thorlabs BB2-E02). The first mirror, placed in front of the converging lens, deflects the light onto a second mirror (KM1⟶ KM2), which then deflects the light onto a focusing lens (KM2 ⟶ L2). The mirrors are placed on kinematic mirror mounts to allow for tilting of the face of the mirrors. In this way, the direction at which the light is deflected can be precisely controlled.

The second kinematic mirror mount (KM2) is equipped with two 12mm motorized actuators (Thorlabs KS1-Z9) that will provide automatic tilting capabilities through scripting. This solution is what allows for the illumination angle-scanning of the sample; exactly how this is implemented will be discussed in Section 4.1. The reason why the setup needs two kinematic mirrors and not just one scanning mirror is to be able to adjust the beam height. It is important that the setup has calibration methods, ensuring that the light follows a precise optical path.

3.4.1 Aligning Beam Height

To align the beam height, two adjustable pinholes are placed in line (using Thorlabs cage rods) after the motorized kinematic mirror. By manually tilting KM1 the beam deflected from KM2 can be directed into the center of the first pinhole. Opening the aperture allows light to pass through to the second pinhole. If the beam height is consistent, then the light would also land in the center of the second pinhole. If this is not the case, KM2 can be adjusted to direct the light to the center of the second pinhole. Thus, it is evident that the tilt of KM1 determines the height at which the beam strikes the first pinhole, while KM2 then controls the beam height at the second pinhole. By iteratively going back and forth between these, the beam height can be aligned so as to be parallel to the optical axis as shown in Figure 3.4.

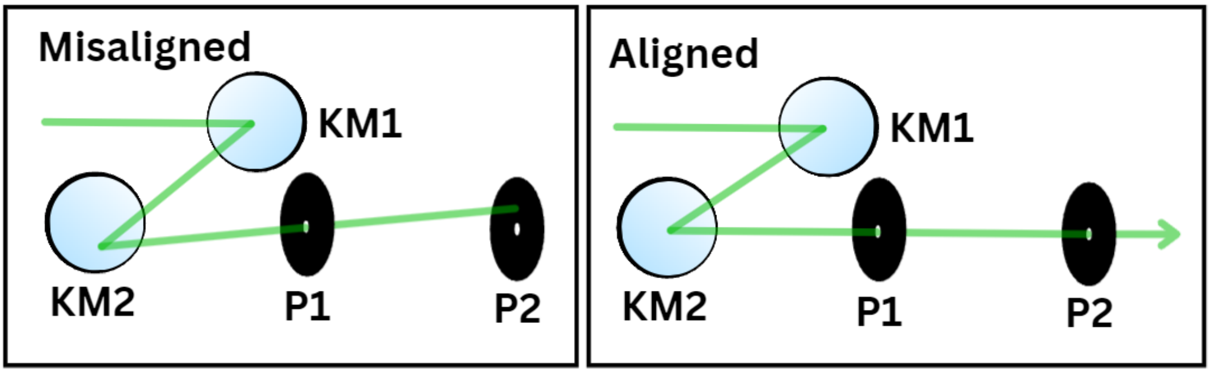


Figure 3.4: Misaligned vs Aligned beam height using aligned pinholes.

3.5 Focusing Lens

Once the beam height is aligned, the next step involves the focusing lens. The purpose of this lens is to focus the collimated light onto the back focal plane of the condenser objective lens, which is technically very close to the back aperture. This ensures that a collimated beam illuminates the sample.

3.5.1 Distance From Motorized Kinematic Mirror to Focusing Lens

The ideal place to position the scanning mirror (KM2) in relation to the focusing lens can be found theoretically through simple geometry. As the mirror tilts, it creates an angle of deflection ϕ with the optical axis. The maximum tilt achievable of the scanning mirror is ±4° (Thorlabs KS1-Z9), which means that the maximum angle of deflection from the optical axis will be $\phi_{max} = \pm 2^\circ$, as shown in Figure 3.5. Here I seek to find the distance p where the focusing lens keeps the beam within the radius of the back-aperture. If the beam is focused outside of the back-aperture, the sample will receive less intensity, and information will be lost.

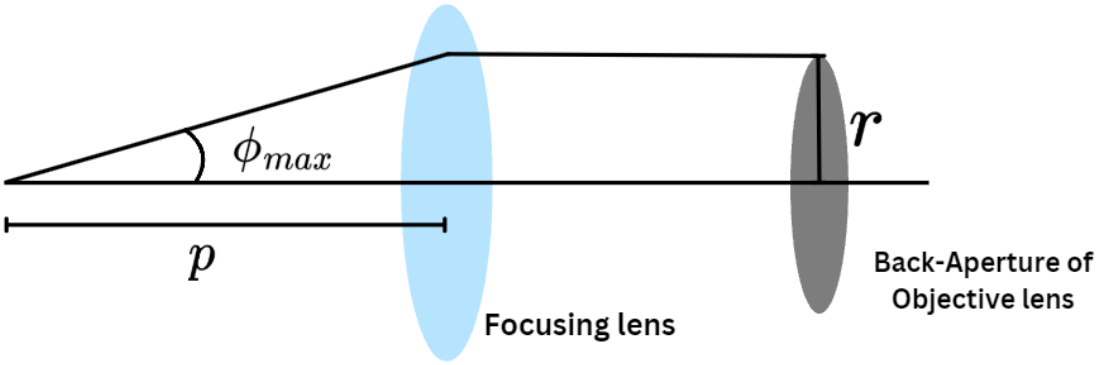


Figure 3.5: Geometry of angled illumination

The condenser objective lens (RMS40X) has a magnification of 40X and an entrance pupil with a diameter of $EP = 5.9$mm. That means that the maximum radius we can illuminate the back aperture is $r = 2.95$mm. Thus, to find the distance p between the scanning mirror and the focusing lens, we only need to solve a simple geometry, which results in the following:

$$p = \frac{r}{\tan(\phi_{max})} = 84.48mm \tag{3.1}$$

As such, the scanning mirror must be placed approximately 84.48mm away from the focusing lens to maximize the illumination on the back-aperture of the condenser lens. The next step is then to determine where to place the focusing lens in relation to the back-aperture of the condenser objective itself.

3.5.2 Distance From Focusing Lens to Condenser Objective

To determine the ideal distance from the focusing lens to the condenser objective back-aperture, I mounted the focusing lens on ER rods attached to a 60mm cage (Thorlabs LCP33). This allowed me to slide the focusing lens back and forth along the optical axis, adjusting its position relative to the condenser objective. During this process I kept the scanning mirror deflected beam on-axis, by aligning the aforementioned beam height. Now, moving the focusing lens simply shifts the position of the focal point.

To identify the precise location where the beam converges, I used an alignment disk (Thorlabs ADF4) with a crosshair as the target, positioning it where the back-aperture of the condenser would approximately be placed. Then when shifting the focusing lens on the ER rods along the optical axis, the beam spot on the crosshair expanded and contracted as in Figure 3.6. Once the beam appeared tightly focused on the crosshair acting as the back-aperture, the focusing lens was in the right place.

When replacing the alignment disk with the condenser objective, I noticed that the back-aperture ended up approximately 10mm closer to the focusing lens. So, I accounted for this offset by shifting the focusing lens away from the back-aperture by 10mm. This ensured that the collimated light was now accurately focused onto the back aperture.

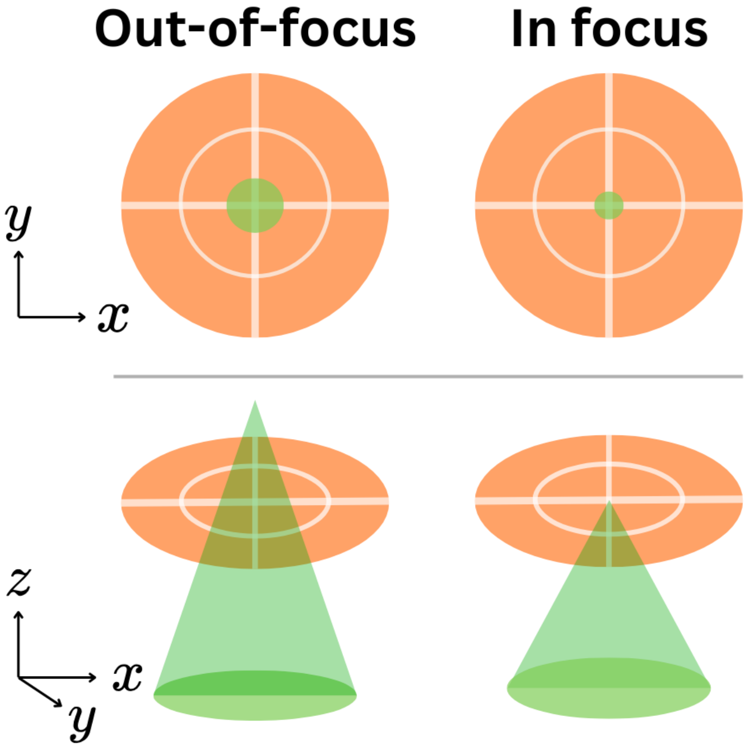


Figure 3.6: Beam spot on out-of-focus and in-focus alignment disk.

3.6 Determining Spatial Cutoff Frequency

The spatial cut-off frequency of the system determines the maximum range of angles that can pass through the back aperture of the condenser objective lens without a significant loss in intensity. This sets a practical limit for the maximum angle at which the focusing lens can focus the light within the back-aperture. Thus, the cutoff frequency of the illumination angles is equivalent to the diameter of the back aperture. To determine this limit, the alignment disk can again be used. By tilting the scanning mirror the angle of deflection increases; this increases the radius at which the light is incident on the focusing lens, which results in the beam being focused at a certain radius on the alignment disk where the back aperture would be. Fortunately, the alignment disk has a circle which is approximately the same in size as the back aperture of the condenser objective. By noting down the maximum scanning angle for which the beam still lands within the radius of this circle, I can determine the spatial cutoff frequency for my system.

I found that the beam remained within the radius of the back-aperture as long as I did not extend the motorized actuators more than 0.2mm. Figure 3.7 illustrates what this looks like once the alignment disk has been replaced by the condenser objective lens, showing both on- and off-axis illumination of the back aperture.

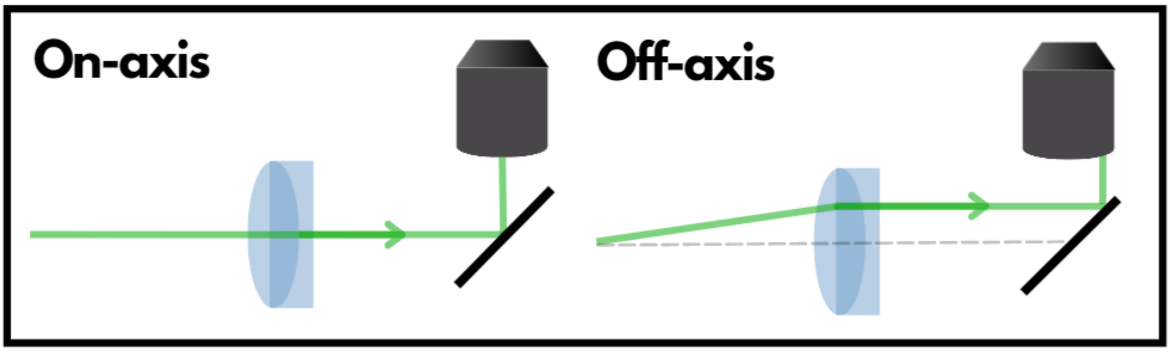


Figure 3.7: Determining the off-axis angle at which the beam remains within the back-aperture area of the condenser objective lens.

Now that the beam height and the focusing lens have been aligned and the spatial cutoff frequency has been determined, the next step is to align the condenser objective and the imaging objective.

3.7 Aligning The Objectives in The XY-plane

The beam gets focused on the back-aperture of the condenser lens and in turn the condenser lens illuminates the sample with an approximately collimated beam. Here, it is important that the condenser objective and the imaging objective are in line, as the imaging objective should be able to capture all of the illumination. To determine whether the objectives are aligned, an experimental adjustment of the objectives is necessary. The imaging lens is attached to a single z-axis stage, allowing moving the imaging objective up and down to focus the objective on the sample plane. The condenser objective, on the other hand, is screwed onto an XY stage for which the position of the condenser objective can be manually moved. By manual adjustment of both the angle of the kinematic mirror (KM3) and the XY position of the condenser objective, as seen in Figure 3.8, the condenser objective can be aligned with the imaging objective, ensuring that the illumination travels from the condenser objective, in a straight line through the sample, and is collected by the imaging lens. Its worth mentioning that during this alignment it is necessary that the beam is kept on-axis. To fine-tune the setup, start by positioning the detection lens as close as possible to the illumination lens (position D). Use the XY stage to center the circular light pattern on the camera sensor. Next, gradually increase the distance between the detection and illumination lenses (position B). If the light projection shifts off-center as the distance increases, adjust the beam angle at the back aperture of the illumination lens. This back and forth process continues until the circular light pattern remains centered in both near and far positions, ensuring precise alignment of the objective lenses and maintaining the beam alignment with the optical axis. It is worth noting that the beam appears quite divergent in the transition from Figure 3.8B to D, as achieving perfect collimation is challenging. However, within the distance at which the objective lenses will be positioned relative to each other, the beam can reasonably be approximated as collimated.

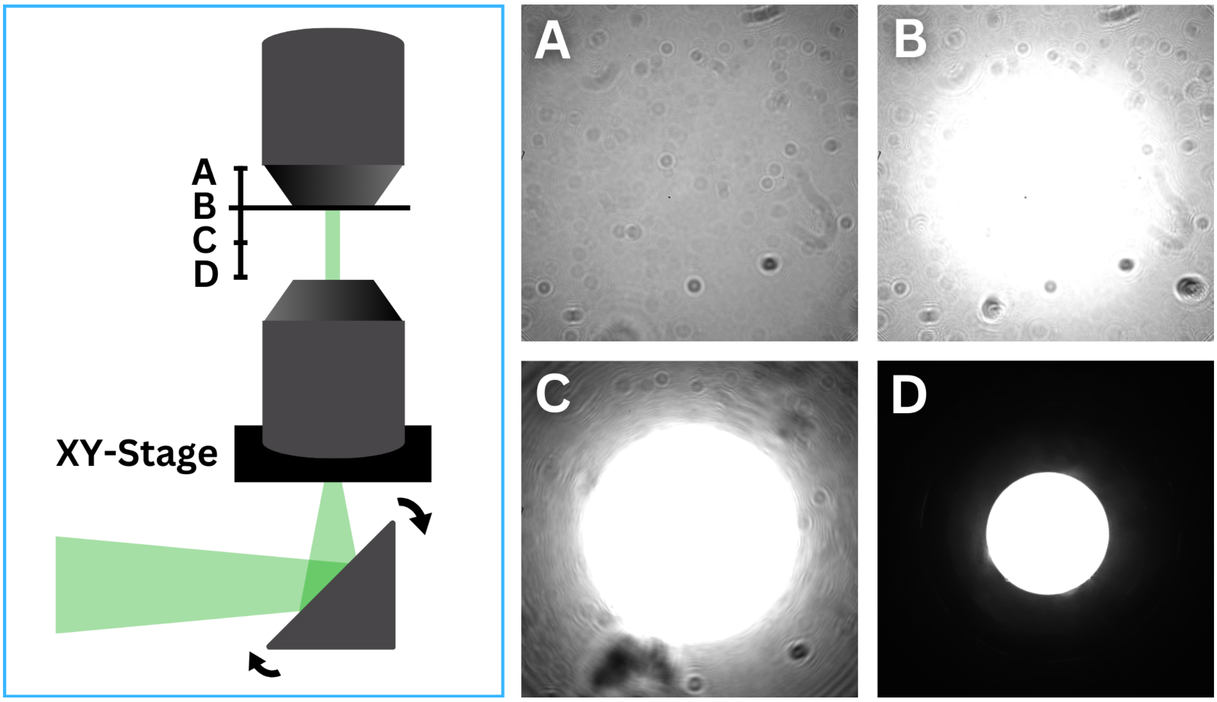


Figure 3.8: Alignment of objectives. Moving the imaging lens up and down along the z-axis we can align our condenser lens to be in-line with the imaging lens.

4

Illumination Angle Scanning & Data Acquisition

Now that my illumination system has been aligned with the optical axis, I move on to utilize the scanning capabilities of the motorized kinetic mirror (KM2). To begin with, I had to figure out how to control the kinematic mirror, automating the illumination angle scanning using Python scripts.

4.1 Controlling The Motorized Kinematic Mirror

The motorized kinematic mirror (Thorlabs KS1-Z9) is equipped with two actuator arms that can be extended by applying direct current DC. To control these arms I used two K-Cube DC Servo motor controller drivers (KDC101), which plug right into my lab computer. To begin with, the K-Cube drivers, which drive the actuator arms, can be controlled using the *Thorlabs Kinesis (TK)* application.

4.1.1 Thorlabs Kinesis

This program has user interface (UI) elements that allow extending and retracting the actuator arms from their home position at 0.0mm to a maximum extended distance of 12.0mm for both arms. The control of actuator arms using TK is completely manual and cannot be automated. However, TK can be used to find the parameters required to properly automate actuator arms. To be able to move the actuators using Python, I need to know the Device Units (DU) of the actuator arms to apply the arm distance. Each device has a set amount of device units which correlates to the position of the arms. In TK, I can enter the distance in millimeters for which the arms extend. To begin with, I ensured that the actuator devices were properly configured, displaying the same offset in the TK UI and also on the display hud of the K-cube drivers. I also ensured that there was no mechanical collision occurring through the calibration, which means that I disconnected the actuator arms from the kinematic mirror. I first "home" the arms, setting their position to 0.0mm. Then I moved the arms to their maximum position, noting it as $P = 12.0$mm, ensuring the position was far enough from zero to get good precision. Now that my device was at its maximum position, I changed the setting to show device units instead of millimeters. Now, the program displays the position in device units as D=411600000.

Finally, to find the parameter required to programmatically use the actuator arms, I simply calculate the device units per millimeter of the device as follows:

$$\texttt{DeviceUnitsPerMillimeter} = D/P, \tag{4.1}$$

where D is the position in units and P is the position in millimeters. This parameter can be used to move the actuator arms using scripting. If I want to move the device by a distance of 10.0 mm, I need to give the program DU * (10.0mm) for it to drive the actuator arms to the correct position.

4.1.2 Micro-Manager

Running the Micro-Manager program, I can also manually control the actuators. However, what is beneficial about Micro-Manager is that it allows me to write scripts to automate the control of the actuator arms. This is done by running the kinesis program through Micro-Manager. To add Thorlabs kinesis support to Micro-Manager, I needed to download the dll drivers for Thorlabs kinesis and import the drivers into Micro-Manager using the built-in hardware wizard. The kinesis drivers were downloaded from the github module [8] using version 71. Here, I needed to make sure that Micro-Manager was up-to-date so that it also supports version 71. Once the kinesis drivers was set up in Micro-Manager, I could finally move on to writing scripts to control the actuator arms in Python.

4.1.3 Pycro-Manager

To control Micro-Manager using Python, I needed to import Micro-Manager controllability into Python using what is known as "Headless mode" [9], where the Micro-Manager Core is launched programmatically without the desktop application. To achieve the headless solution, in which you do not need Micro-Manager to run, I utilized a Python package called Pycro-Manager.

Listing 4.1: Initializing Headless Mode in Python using Pycro-Manager

```
def initialize_core():
    pycromanager.start_headless(mm_app_path, config_file,
                                python_backend=True, java_loc=None,
                                core_log_path, buffer_size_mb=1024,
                                max_memory_mb=2000, port=4827,
                                debug=False)

    core = pycromanager.Core()
    core.set_timeout_ms(100000)
    return core

```

Pycro-Manager is a low-level application programming interface (API) which provides direct access to the fundamental features of Micro-Manager with minimal abstractions. Having direct access to the Micro-Manager "Core" allows direct hardware control from the Python script, allowing me to control the necessary components of the microscopic setup. This allows me to directly set the actuator X and actuator Y positions, send devices to their original position through "homing", but most importantly write user-specific functions for controlling the precise position of actuators.

I wrote a script which allows the user to move the actuators by using the arrow keys (up) and (down) increased and decreased the y-actuator position while (left) and (right) arrow keys increased and decreased the x-actuator position. I also wrote a function for cycling through different step sizes, from 0.1mm to 0.001mm, allowing for precise control of the arm’s extent. I wrote a function to send the device `home()` meaning that both actuators were retracted to their 0.0mm positions.

Listing 4.2: Manual Actuator Control Script

```
def control_actuators(self):
    directions = {
    ’up’: (self.device_2, self.step_size_mm),
    ’down’: (self.device_2, -self.step_size_mm),
    ’left’: (self.device_1, -self.step_size_mm),
    ’right’: (self.device_1, self.step_size_mm),}
    for key_name, (device, step) in directions.items():
    if key.is_pressed(key_name) and not self.state[key_name]:
        self.state[key_name] = True
        move_device(self.core, device, step)
    else:
        self.state[key_name] = False

```

Aligning the illumination to be on-axis, as described in Section 3.4.1, could now be done by tilting the kinematic mirror. Directing the angle of deflection using the arrow keys and aligning the beam using the alignment pinholes.

Once the alignment was in place, the x, y coordinates of the kinematic mirror were noted as `origin`= (9.15, 7.80) and a function was written to drive the actuators to this position by pressing "o" on the keyboard. I called this position the origin. This is where the mirror should be for the on-axis illumination. From the origin position, increasing each actuator by some distance, the illumination moves off-axis.

In the end, I had the following functions. `send_devices_home()` for resetting the coordinate system. `send_devices_to_origin()` for moving the mirror in the on-axis position. `find_devices()` for displaying the actuator current position. And `Wait()` for waiting until the actuators stop moving before attempting the next action.

Listing 4.3: Functions Script

```
    def handle_input(self):
        if key.is_pressed('h'):
            send_devices_home(self.core, self.device_1, self.device_2)
        if key.is_pressed('o'):
            send_devices_to_origin(self.core, self.device_1, self.device_2)
        if key.is_pressed('w') and not self.state['find_devices']:
            self.state['find_devices'] = True
            x, y = find_devices(self.core, self.device_1, self.device_2)
            print(f'\n Devices position @ x={x} y={y}')
        elif not key.is_pressed('w'):
            self.state['find_devices'] = False
        if key.is_pressed('r') and not self.state['change_step_size']:
            self.change_step_size()
        else:
            self.state['change_step_size'] = False

```

Camera Sensor

The camera sensor in my setup is a Hamamatsu ORCA-Fusion (C14440-20UP). It has an image size (2304x2304 pixels) that is captured at 16bit, allowing for high-quality imaging. Since I already have made the actuator arms work headlessly through Micro-Manager and Python, setting up the camera sensor to take images was pretty straightforward. I simply added the Hamamatsu driver to the Micro-Manager hardware wizard and could then access the camera controls, take images, and change the exposure time using the same `Core` as the actuator arms. This setup enabled me to operate the camera while simultaneously moving the actuator arms, allowing me to observe the effect of the deflection angle on the intensity captured by the camera, as demonstrated by the main function in Listing 4.4.

Listing 4.4: Main Function

```

    def main():
        run_main = True
        core = initialize_core()
        actuator = ActuatorInputController(core)
        camera = CameraController(core)
        camera_thread = threading.Thread(target=camera.live_view)
        camera_thread.start()
        while run_main:
            actuator.handle_input()
            if actuator.state['quit']:
                run_main = False
                camera.running = False
        camera_thread.join()
        quit_core(core)

```

I observed that if the actuators were set to specific positions, such as $x = 10.0$mm, and the Thorlabs drivers were turned off and then turned on, the driver would reset the actuator positions to $x = 0.0$mm. However, the actuators themselves did not physically move from their original positions before the power cycle. This meant that if I then moved the actuators, say, by 5.0mm, the program would think the position was $x = 5.0$mm, but in reality it would be $x = 15.0$mm. To solve this inconsistency, I wrote the "homing" function to reset the position of the actuators, so that when the devices are turned back on, I can reset the position back to 0.0mm before moving the actuators into the desired position.

Illumination Pattern

Once the controller functions were in place, I wrote a script to run through different illumination angles and take pictures at each angle as shown in Listing 4.5.

Listing 4.5: Pattern Function

```
    def run_pattern(self):
        self.core.snap_image()
        image = self.core.get_image()
        height, width = image.shape
        num_positions = len(self.positions)
        image_stack = np.zeros((height, width, num_positions), dtype=np.uint16)
        for idx, (pos_x, pos_y) in enumerate(self.positions):
            self.core.set_position(self.device_1, pos_x * DEVICE_UNITS_PER_MM)
            self.core.set_position(self.device_2, pos_y * DEVICE_UNITS_PER_MM)
            wait_for_device_to_finish_moving(self.core, self.device_1)
            wait_for_device_to_finish_moving(self.core, self.device_2)
            self.core.snap_image()
            image_stack[:, :, idx] = self.core.get_image()
        return image_stack

```

4.2 Determining The Angle of Incident On The Sample

Once I have control over the positioning of the motorized kinematic mirror (KM2), the next step was to determine the relationship between the angle of deflection (ϕ) determined by KM2 and the resulting angle of incidence (θ) of the illumination on the sample. Knowing this relationship, I can directly control the illumination angle on the sample through the input of the motorized kinematic mirror, which is an essential step for accurate multiangle imaging.

The thin lens equation is defined as:

$$\frac{1}{p} + \frac{1}{q} = \frac{1}{f} \tag{4.2}$$

where p is the distance from the object (KM2) to the focusing lens, q is the distance from the focusing lens to the virtual image and f is the focal length of the focusing lens. This relationship is illustrated in Figure 4.1, which shows a two-lens system where the focusing lens forms a virtual image, magnifying the real inverted image created by the objective lens.

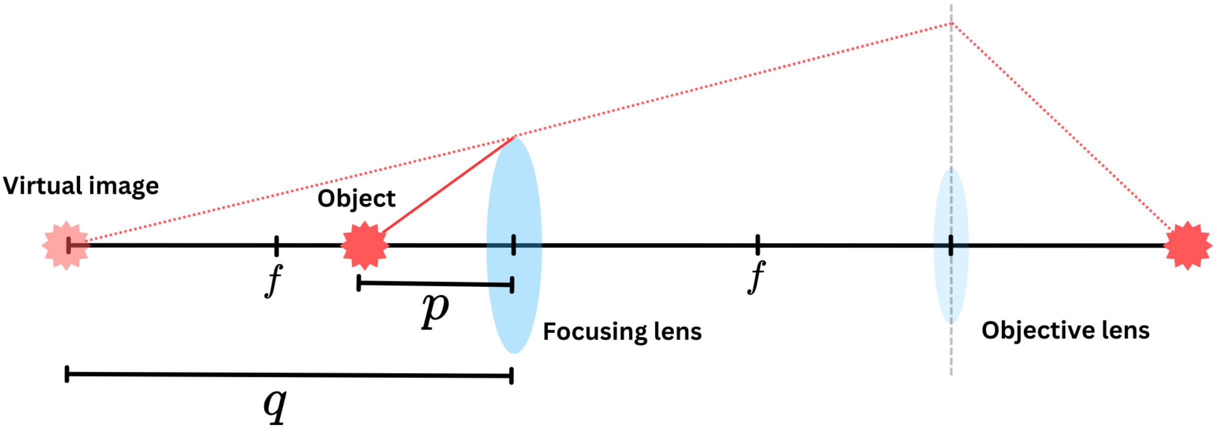


Figure 4.1: Ray-optics diagram illustrating thin-lens approximation. Showing relationship between focusing lens and image formation.

Knowing p from Section 3.5 and $f = 300$ mm, the imaging distance q can be calculated as:

$$q = \frac{1}{(\frac{1}{f} - \frac{1}{p})} \tag{4.3}$$

However, the system's optical path includes a condenser objective, which can also be approximated as a thin lens. By incorporating this additional lens, we model the system as shown in Figure 4.2.

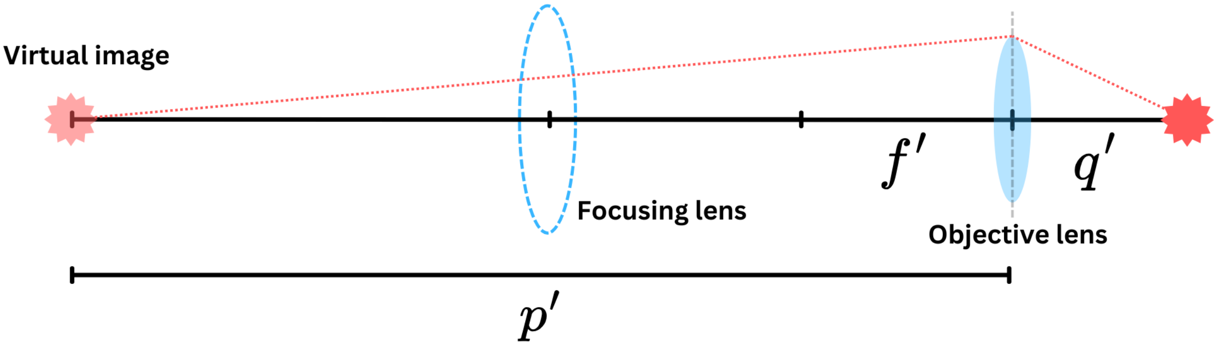


Figure 4.2: Ray-optics diagram illustrating showing the relationship between objective lens and image formation.

Thus, what I really need to find is q'. The compounded thin-lens equation can then be defined as:

$$\frac{1}{p'} + \frac{1}{q'} = \frac{1}{f'} \tag{4.4}$$

where p' is the distance from the objective lens to the image, q' is the distance from the condenser objective to the sample, and f' is the focal length of the condenser objective.

Using the relationship $p' = q + f + f'$ I can determine q' as:

$$q' = \frac{1}{\frac{1}{f'} - \frac{1}{p'}} \tag{4.5}$$

Next, the radius r at which the light strikes the focusing lens is defined as:

$$r = p \tan \phi, \tag{4.6}$$

where p is the distance from the kinematic mirror to the focusing lens and ϕ is the angle of deflection of the kinematic mirror. Then using r, the incident angle θ on the sample is given by:

$$\theta = \arctan\left(\frac{r}{q'}\right) = \arctan\left(\frac{p \tan \phi (p' - f')}{p' f'}\right) = \arctan\left(\frac{p \tan \phi (q + f)}{f'(q + f + f')}\right) \tag{4.7}$$

where q' is the distance from the condenser objective to the sample. Now I have a direct relationship between the input angle (angle of deflection) and the output angle (incident angle on the sample). The final step is to relate the angle of deflection to the kinematic tilt of the arm. The angle of deflection can be defined as:

$$\phi = \arctan\left(\frac{x}{L}\right), \tag{4.8}$$

where L is the length of the kinematic mirror and x is the extended position of the actuator arm.

4.3 Defining The Pattern of k-Vectors

This code generates a spiral pattern of k-vectors by producing x and y coordinates that trace a continuous and evenly spaced spiral path. The pattern is defined by the number of points N and the number of revolutions revs. Initially, the code parametrizes the x and y coordinates as trigonometric functions of t, representing the angular position along the spiral. To ensure uniform spacing, it calculates the cumulative distance between consecutive points and adjusts the angular increments (tInc) accordingly. This results in a well-defined spiral where each point corresponds to a unique k-vector. The spiral provides a structured sequence of illumination angles, ideal for experiments that require precise control over the angular diversity of the illumination pattern.

It is important to make sure that the spiral pattern k-vectors are within the microscope cutoff frequency. In section 3.6 I found that the maximum offset my actuator arms could be at during illumination was approximately 0.2mm. Therefore, it is important that when I now convert my spiral pattern k-vectors to x and y actuator offsets that these offsets from origin do not go beyond 0.2mm. This constraint ensures that the illumination angle produces bright-field images, as the algorithm does not account for dark-field scattering. This requirement helps maintain the accuracy and precision of the experimental setup by keeping the spiral pattern within the spatial cutoff frequency of the microscope.

Listing 4.6: Spiral Path Generation Script

```
def generate_spiral_path(N, revs):
    t = np.linspace(0, 2*np.pi, N)
    x = t * np.cos(revs*t)/2/np.pi
    y = t * np.sin(revs*t)/2/np.pi
    dist = np.zeros(len(x))
    for k in np.arange(2,len(dist)+1):
        dist[k-1] = np.sqrt((x[k-1]-x[k-2])**2
                            + (y[k-1]-y[k-2])**2) + dist[k-2]
    coef = np.mean(t[N-1]**2/dist[N-1])
    distPerAcq = dist[-1]/N  # 0.187979
    tInc = np.sqrt(np.arange(0, dist[-1]
                   + distPerAcq, distPerAcq) * coef)
    x = tInc*np.cos(revs*tInc)/2/np.pi
    y = tInc*np.sin(revs*tInc)/2/np.pi
    return x,y

```

4.4 Determining Actuator Offsets Based on k-Vectors

In the code, the theory of Section 4.2 is implemented to compute the required offsets for the actuator arms based on the incoming wave vectors (kx, ky). The function `convert_k_vectors` calculates the incident angles on the sample (θ_x and θ_y) using the input wave vectors and the wavelength of light. The focal length of the objective lens is factored in and the distances from the focusing lens and the condenser objective are used to find the position of the image. Using these distances, the radius at which the light hits the tub lens is determined, from which the deflection angles (ϕ_x, ϕ_y) are calculated. Finally, the required offsets for the actuator arms are determined, enabling precise control over the illumination angles, and the code outputs the maximum arm offset and its relative size compared to the back aperture diameter. This allows for a direct mapping between theoretical light path predictions and the practical control of illumination angles in the optical setup. With the pattern I defined, I made sure that once the k vectors were converted, that the offsets of the actuator arms x and y were within the spatial cutoff frequency. The maximum arm offset is 0.2mm, but to ensure optimal performance, it is advisable to stay somewhat within this value. The chosen maximum offset of 0.175mm is within 87.79% of the back aperture diameter, which falls comfortably within this range, leading to the selection of k-vectors that adhere to this restriction.

4.5 Angle Scanning Pattern

The system utilizes a motorized kinematic mirror to sequentially illuminate a sample at various angles by running through a predefined set of offset positions. Using a `pattern_control` class, the system moves two actuators to the specified x and y positions, converting these into precise angular offsets. At each position, the system captures an image using a connected camera, ensuring that the devices have reached their target positions before snapping the image. These images are then compiled into a 3D image stack, enabling the capture of illumination patterns across all positions for subsequent analysis. To obtain a complete reconstruction, sufficient coverage of this 3D Fourier space is needed. Increasing the number of illumination angles provides denser sampling of the reciprocal space, leading to better coverage and more accurate reconstructions. However, increasing the number of acquisitions, which corresponds to the number of illumination angles, increases the computational burden of the reconstruction process. Therefore, a balance must be struck between the quality of reconstruction and the computational feasibility [1].

4.6 Data Acquisition

For each pattern position, the sample is illuminated from a different angle. The illumination is diffracted by the RI distribution of the samples and is propagated to the imaging sensor. Once each illumination angle position is reached, the camera takes an image of the diffraction pattern with a 10ms exposure time, before moving on to the next position in the pattern. Subsequently, the acquired images were processed to extract quantitative information regarding the diffraction patterns, enabling detailed analysis of the samples' refractive index and optical properties, as will be discussed in the next section.

5

Post Processing

The final stage of the multi-angle illumination setup is post-processing, where the captured images are analyzed and calibrated for accurate reconstruction.

In Section 4.2, the approximate incident angles on the sample were theoretically determined. However, these values may be inaccurate due to potential misalignment or human error in the experimental setup, necessitating further calibration.

5.1 Illumination Angle Calibration

To achieve accurate reconstruction in ODT, it is essential to have a precise knowledge of the illumination angles. Calibration procedures are crucial for determining these angles with high accuracy, thereby minimizing artifacts and distortions in the 3D RI map. One such approach involves using known reference samples, such as microbeads with well-defined dimensions and refractive indices, to estimate the illumination angles [7]. By imaging these samples and comparing the acquired data with their expected properties, the angles can be determined. Alternatively, self-calibration algorithms can be employed to directly extract the required information from the acquired diffraction pattern measurements, eliminating the need for separate calibration samples [1]. These methods often involve analyzing the Fourier spectrum of acquired images [10]. Fourier ptychography, which is a quantitative phase imaging technique, employs a multi-angle illumination setup similar to mine. In my setup, I am utilizing calibration techniques commonly found in Fourier ptychography to improve image acquisition, specifically to enhance the accuracy of the phase recovery process. This method takes advantage of the distinct patterns that emerge in the Fourier spectrum when a sample is illuminated at different angles. For instance, when the sample is illuminated off-axis, the Fourier spectrum typically shows circular patterns, where the centers of these circles correspond to the angle of illumination. Identifying these circles, the angles can be accurately determined [10].

5.1.1 Efficient Self Calibration Method

For weakly scattering samples illuminated within the pupil function of the system, the Fourier transform of the intensity image displays symmetric circular regions, commonly referred to as the "bright-field" region [10][11]. However, in the case of multiple scattering samples or dark-field illumination, where direct transmitted light is weak or absent, additional spatial frequency signals may appear outside these bright-field circles [11]. To accurately determine the illumination angle, I employ a circular edge detection method to locate the centers of these circles, thereby identifying the corresponding illumination angles.

The illumination calibration algorithm in this paper [10] is open source and can be used. The algorithm refines the estimated illumination angles found in Section 4.2 through post-acquisition self-calibration of the acquired diffraction images. Raw image data and system parameters, such as numerical aperture, magnification, pixel size, and wavelength, are used to calculate the expected spatial frequencies of illumination. Fourier transforms are applied to the image stack, allowing analysis in the frequency domain, where circular patterns representing illumination wave vectors are identified.

A circle-finding algorithm is used to locate these patterns, refining the incident illumination angles. To account for system imperfections, an outlier removal process applies a transformation model to correct misalignment. The calibrated angles are converted back to k-space and saved for comparison. Visualization tools display the uncalibrated and calibrated angles as seen in Figure 5.1. This calibration step ensures precise angle estimation, critical for accurate 3D refractive index reconstruction.

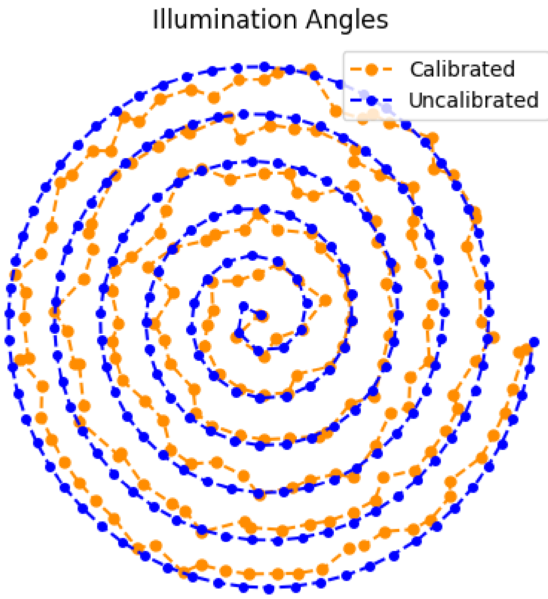


Figure 5.1: Calibrated and Uncalibrated illumination k-vectors for the spiral pattern overlaid on top of each other. This figure shows the deviation of the calibrated spiral pattern from the original spiral pattern.

5.1.2 Challenges with Image Cropping

During the early stage of the development process, I experimented with cropping the acquired diffraction images from their original size of 2304x2304 pixels down to 100x100 pixels to reduce algorithmic processing time. This approach aimed to significantly accelerate computations, potentially reducing the processing time by a factor of 20. However, this method proved problematic when applied before calibrating the illumination angles, as the calibration algorithm relies on full-scale images for accurate results. Consequently, the cropped images led to incorrect calibration outputs. At this early stage, the sample images consisted of both bright-field and dark-field regions, which contributed to the calibration errors. Cropping the images often resulted in a region that was either entirely bright-field or entirely dark-field, skewing the Fourier spectrum and compromising angle recovery. The Fourier transform used in the calibration algorithm depends on the complete frequency-space structure present in the full images to ensure accuracy. Cropping not only reduced the spatial resolution in the frequency space but also disrupted the necessary symmetry and centering of the data, which are critical for successful calibration. Later in the development process, I realized that the images used for calibration must all be bright-field, meaning that most of the light must reach the sample. This adjustment eliminated the issue caused by mixed bright-field and dark-field regions and ensured consistent and reliable calibration results.

5.2 Finding Total Magnification Using Bead Samples

To utilize the aforementioned calibration algorithm, I needed to figure out what the sample size was on the sensor. This can be done by taking images of particles with a known size. For this approach, I imaged a sample comprising 7µm polystyrene beads in immersion medium with glycerol. Because I know the real size of the beads, I can determine how much the image is magnified, revealing the total magnification of my system. By taking an image of the bead sample in focus and analyzing a line profile across one of the beads (Figure 5.2), the Full Width at Half Maximum (FWHM) can be determined, as shown in Figure 5.3.

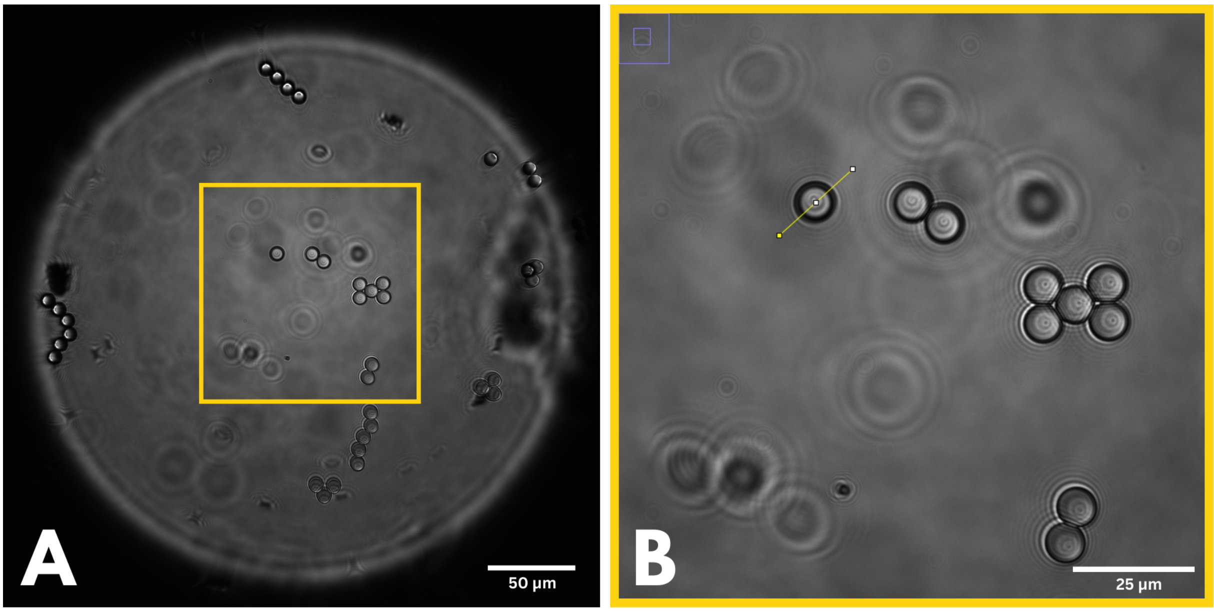


Figure 5.2: A) Full FOV image of 7µm beads and B) shows the acquiring process for line profile across bead structure using *ImageJ*.

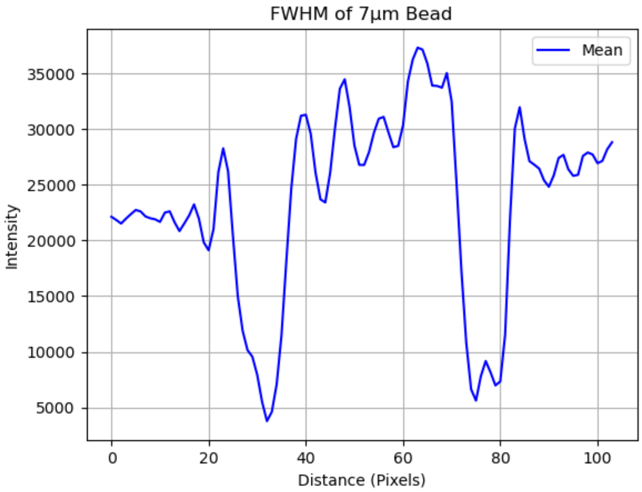


Figure 5.3: Intensity line-profile of 7µm bead structure. The intensity profile shows significant peaks and troughs, suggesting variations in the bead's light distribution.

Now, calculating the total system magnification is pretty straightforward. The only issue I face here is that the FWHM region of the bead sample is quite wide as seen in Figure 5.3. It ranges from a distance of 43 pixels to 56 pixels. Ideally, the FWHM should be very clear, but due to the range of possible values, it is a bit hard to tell exactly what the FWHM is. This issue stems from the bead's outer radius appearing to have a thick border, probably due to diffraction or other optical artifacts. Misfocus can exacerbate this effect, causing halos or diffraction rings to appear thicker. Using larger beads may help reduce the prominence of these surrounding artifacts, depending on the setup. The total magnification can then be calculated as follows:

$$\text{Total Magnification} = \frac{px_{\text{camera}}}{px_{\text{sample}}} \tag{5.1}$$

where px_{camera} is the pixel size of the camera (6.5µm) and px_{sample} = sample size/FWHM. Therefore, the total magnification is between 39X and 52X, depending on the FWHM range. I decided that somewhere in the middle should be sufficient, so I let the total magnification be 40X.

There is a visible discrepancy between the beads in the center of Figure 5.2A and those located at the edge of the image. This effect is likely due to the curvature of the illumination field that occurs on the edges. The middle beads look as expected because the field in the middle is approximately planar (Fig. 5.4C), but as you go towards the edge, it is clear that the field curves. Figure 5.4A shows the effect of the curved incident e-fields on the beads at the edge of the field of view, while Figure 5.4B shows the approximately planar incident e-fields on the beads in the center of the field of view.

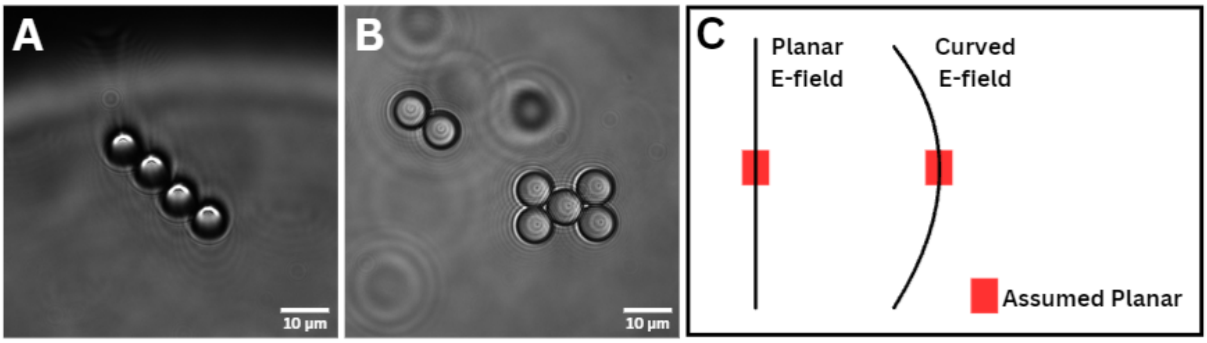


Figure 5.4: A) Curved incident e-field at the FOV edge, B) planar incident e-field on 7 µm beads at the FOV center, and C) planar e-field and Curved e-field with assumed planar midsection in red.

This field curvature comes from the misalignment of the illumination system. Ideally, the illumination should produce a field that is planar, but this is difficult to manage in a custom-built setup. The data is still usable as the middle section of the image seems to be axially aligned, one must simply crop out the edge of the image.

6

3D Refractive Index Reconstruction

Having calibrated the illumination k-vectors, I proceeded to reconstruct the 3D refractive index from the acquired intensity diffraction images. This section describes the Python-implemented MSBP algorithm used for reconstruction, initially developed during the master thesis preproject [6]. The required libraries are listed in Appendix A.

6.1 Data Loading and Preprocessing

The algorithm begins by loading the calibrated data from the illumination calibration algorithm. The data includes intensity measurements and calibrated spatial frequencies, as well as metadata such as the numerical aperture (NA), and the pixel size px_sample. The wavelength of the illumination source is set to 530nm, matching the wavelength used in the experimental setup. The experimental setup ran through 200 different illumination angles comprised in the illumination pattern of section 4.5.

Intensity measurements are normalized by dividing by the maximum intensity value to obtain the amplitude of the electric field. The data is then cropped to a specific region of interest (ROI) for further processing as seen in Figure 6.1. The sample I imaged was that of a Peace Lilly Pollen immersed in a glycerol-based medium called ProLong Glass Antifade Mountant. The peace lilly pollen has a refractive index of RI=1.55 − 1.58 [12] and the immersion medium has a RI of 1.51. Using this sample came down to the fact that the refractive index difference between the sample and the medium it is immersed in is $\Delta RI \approx 0.5$, which makes the sample weakly scattering, ultimately reducing the reconstruction artifacts. Peace lilly pollen cells (Spathiphyllum cannifolium) are typically ca. 25µm in size [12] so the scale of Figure 6.1 seems accurate.

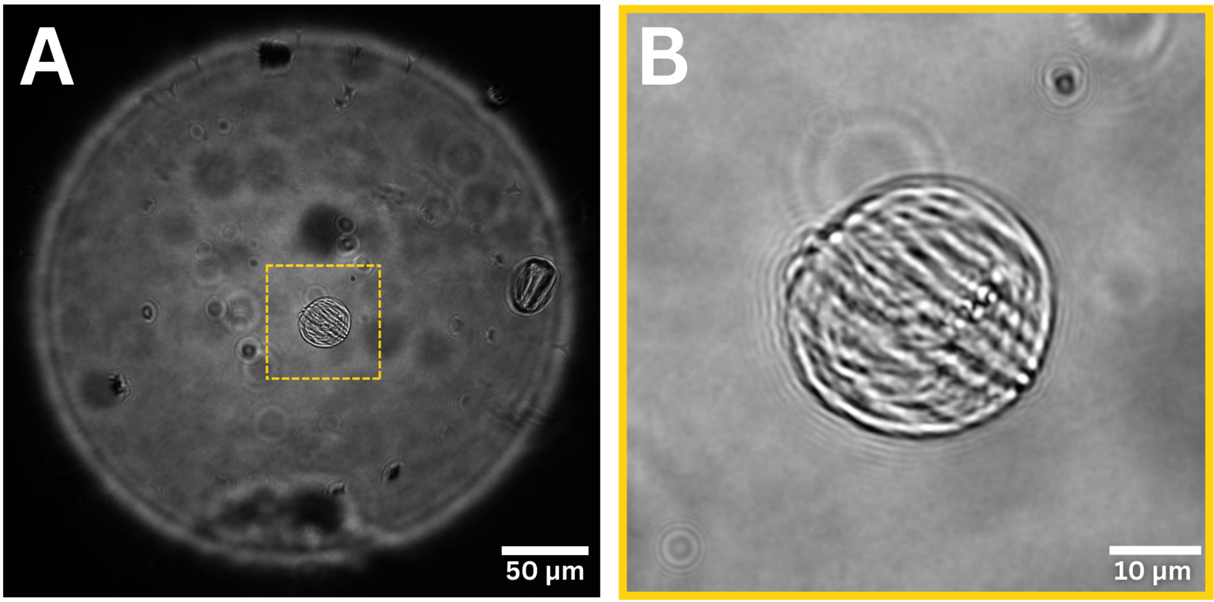


Figure 6.1: A) Full FOV of Peace Lilly Pollen Sample, B) Cropped ROI on Pollen Cell for reconstruction.

6.2 Propagation and Numerical Aperture Masking

The Multi-Slice Beam Propagation method relies on a propagation kernel and NA masks to model the behavior of light as it interacts with the sample. These components simulate the physical constraints of the optical system and are essential for accurate reconstruction of the 3D RI distribution.

The propagation kernel (Fig. 6.2A) models phase and amplitude changes in electromagnetic waves as they propagate through a medium. It accounts for the refractive index, wavelength, and spatial frequencies in the Fourier domain, allowing the simulation of light scattering and refraction within the sample. This is key to forward modeling the optical scattering process and predicting intensity distributions at the detector plane.

The NA mask (Fig.6.2B) restricts the spatial frequencies beyond the imaging system's resolving capability, defined as:

$$\nu_x^2 + \nu_y^2 \leq \left(\frac{NA}{\lambda}\right)^2 \tag{6.1}$$

where (ν_x, ν_y) are spatial frequencies, NA is the numerical aperture of the imaging objective, and λ is the wavelength. These filters ensure that only physically propagating waves, consistent with the NA of the system, contribute to the reconstruction. By incorporating these elements, the MSBP algorithm provides a forward model aligned with the constraints of my custom-built microscope, facilitating accurate 3D refractive index imaging from experimental data.

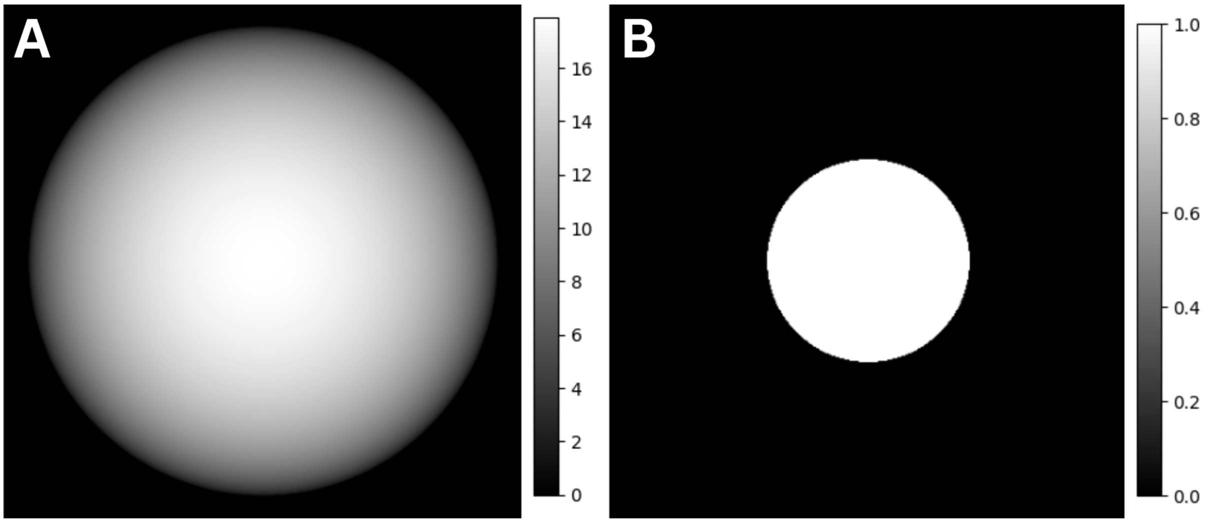


Figure 6.2: A) Propagation Kernel: The propagation kernel represents the phase shift introduced by wave propagation in free space. The grayscale intensity indicates the kernel values, which depend on the spatial frequency components of the light. This kernel is critical for modeling and simulating light propagation in numerical reconstruction methods. B) The NA mask defines the spatial frequency cutoff determined by the optical system's aperture. Values of 1 (white) represent the allowed spatial frequencies within the NA limit, while values of 0 (black) represent frequencies beyond the system's resolution. This mask is used to filter spatial frequency components and plays a crucial role in restricting light propagation to realistic optical conditions.

6.3 Multi-Slice Beam Propagation (MSBP) Method

The reconstruction algorithm is built around the MSBP method, which iteratively refines the refractive index distribution of the sample. This process relies on two key functions: `multi_slice_forward` and `BPM_update`.

The `multi_slice_forward` function simulates the propagation of incident light through the sample. Calculate the electric field at each slice and propagate it to the focal plane, accounting for spatial frequencies of the illumination, the propagation phase, and the NA mask.

The `BPM_update` function handles the backprojection gradient update. It computes the difference between the measured and simulated electric fields, back-propagates this discrepancy through the sample, and updates the refractive index at each layer. In addition, it evaluates the cost function to monitor the convergence of the reconstruction process.

6.4 Dimensions and Pixel Size

In the reconstruction of a pollen cell from a peace lily, the axial size of the pixels ΔZ and the number of slices (O) were evaluated to balance resolution and computational efficiency. Abbe's diffraction formula provides a theoretical limit for axial resolution defined as:

$$\Delta Z = \frac{2\lambda}{NA^2} = 2.51\mu\text{m}, \tag{6.2}$$

given the parameters $\lambda = 530$nm and NA = 0.65. For Nyquist sampling, the axial resolution should be $\Delta Z/2 = 1.25\mu$m. As the cell I am imaging is approximately 25μm it will be required that the axial dimension of the reconstruction space O is equal to

$$O = \frac{25\mu\text{m}}{1.25\mu\text{m}} = 20, \tag{6.3}$$

representing the number of slices along the z axis in the 3D refractive index reconstruction. This choice of axial dimension and pixel size is crucial to balance the trade-off between computational efficiency and the desired resolution in the reconstructed 3D refractive index distribution. A larger axial dimension would enhance the resolution but would also significantly increase the computational cost, while a smaller dimension would reduce the computational cost but potentially compromise the accuracy of the reconstruction. The selected values ensure that the reconstruction captures the fine details of the sample's refractive index while remaining computationally feasible.

6.5 Iterative Reconstruction

The iterative reconstruction process runs the forward model and the back-projection update for a set number of iterations. To enhance the robustness of the reconstruction, the algorithm randomly scrambles the illumination angles at each iteration. The cost function is monitored to track convergence and the algorithm includes a predefined maximum number of iterations. The stopping point is determined by observing when the cost function converges as in Figure 6.3. After iterative reconstruction, the algorithm simulates the forward model for all illumination angles to compare the reconstructed electric fields with the measured data.

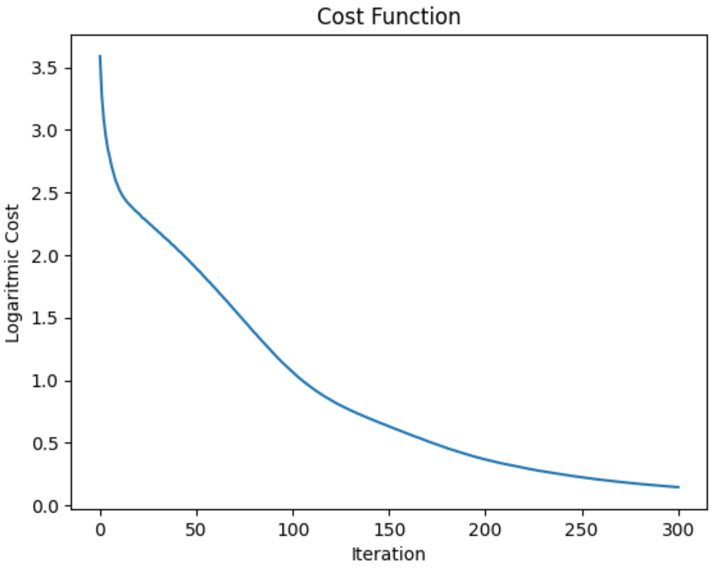


Figure 6.3: Cost function, showing convergence after around 300 iterations.

In summary, the algorithm employs the MSBP method to iteratively reconstruct the 3D refractive index of the sample by simulating the forward propagation of light through the sample and updating the refractive index based on the difference between the measured and simulated electric fields. The iterative process is guided by a cost function that evaluates the convergence of the reconstruction. The computational algorithm defines the maximum iteration through which the algorithm runs. The maximum iterations before stopping can be determined by checking how many iterations it takes before the cost function converges.

Part II

Results

7

Results & Discussion

The results section begins with the validation of the reconstruction algorithm using simulated phantom data as a benchmark in Section 7.1, establishing a foundation for evaluating the reconstruction of the experimental data in later sections. Then, the reconstructed amplitude and phase images of the Peace lily pollen cell will be presented in Section 7.2, highlighting the captured morphology and internal features. Reconstruction problems will then be identified and hypothesis for future improvements to the reconstruction will be presented in Section 7.3. Lastly, the reconstruction performance will be compared with that of commercial microscopy setups to assess its potential within the field of microscopy in Section 7.4.

Due to the time limitations, the subsequent discussion is largely hypothetical, hinting toward the potential cause of the presented reconstruction problems and steps that can be taken in solving these problems in the future. I also highlight some areas of improvement that can be made in future iterations of this microscopic setup.

7.1 Reconstruction Benchmark

The reconstruction algorithm used in this work was developed for my preproject paper. In this project I applied multi-angle beam propagation on a simulated phantom cell. Phantom cells are commonly used in simulations during the development of reconstruction algorithms because their expected results provide a reliable benchmark for testing and validation. Therefore, the phantom reconstruction from my pre-project, and by extension this section, serves as a reference point before discussing the reconstructed results of the Peace Lily sample data obtained from my setup.

Figure 7.1 shows the reconstruction results of the MSBP method on a simulated Phantom cell. Panels (a,b,c) display the true phantom's cross-sectional planes (x,y), (x,z), and (z,y), while (d,e,f) show the corresponding reconstructed planes. The true and reconstructed phantom match well.

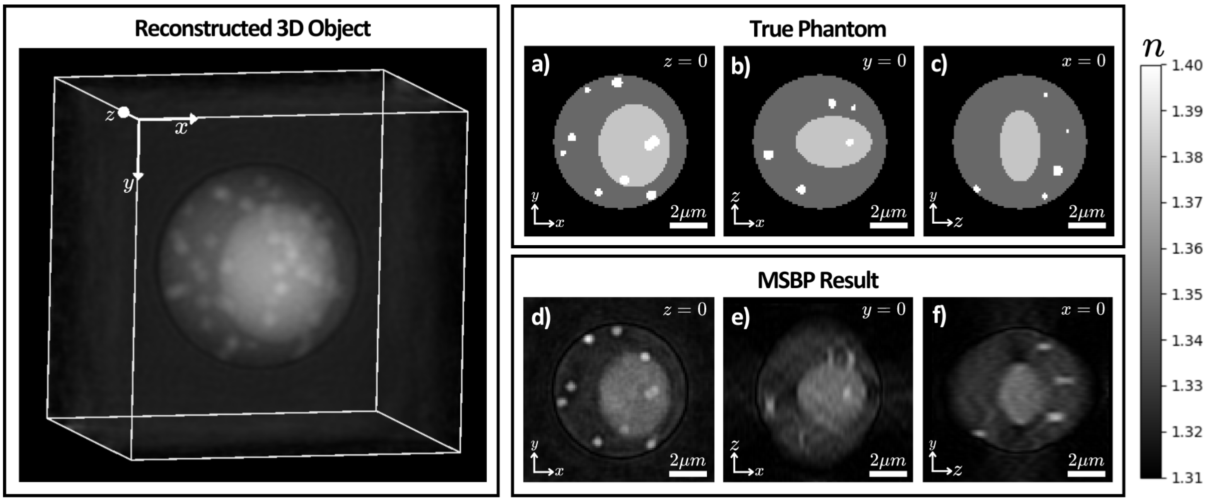


Figure 7.1: Reconstructed results of the 3D phantom object. Cross-section of the sample from the (a,d) front, (b,e) top, and (c,f) side. This Figure is from my preproject paper

By applying the MSBP forward model to the reconstructed phantom object, I can validate its precision. Figure 7.2 shows the wave field propagated on-axis through the reconstructed phantom, demonstrating that the measurements are virtually identical.

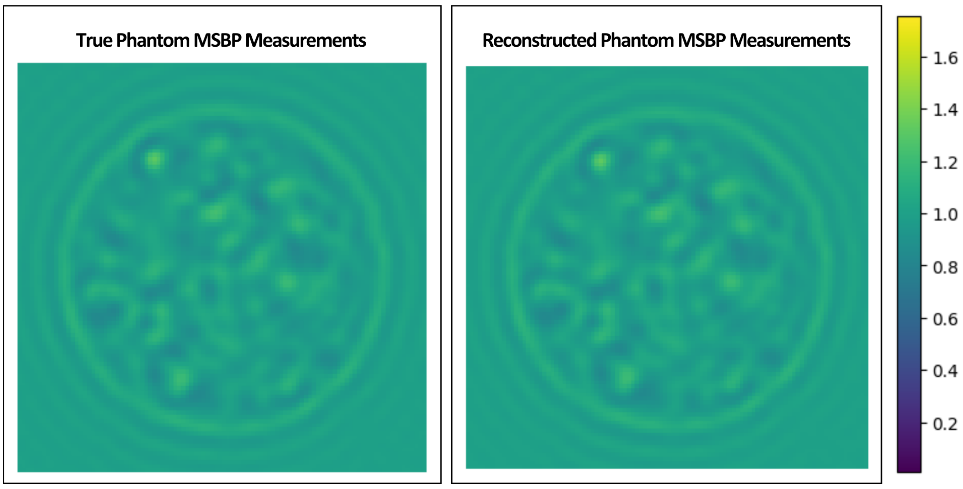


Figure 7.2: Comparison between true phantom diffraction measurement and reconstructed phantom diffraction measurement from the on-axis illumination angle.

This benchmark serves as a ground truth, confirming that the Python algorithm functions as expected with the correct parameter considerations. By establishing this benchmark, I have a clear reference to evaluate the performance of the reconstruction algorithm on real experimental data. Although the results in the following sections did not meet my expectations, this benchmark confirms that the algorithm works correctly. Therefore, any discrepancies in the results are likely due to issues with the reconstruction parameters or the experimental data itself, rather than errors in the algorithm's calculations.

7.2 3D Refractive Index Reconstruction

Figure 7.3 presents the reconstructed amplitude and phase in the center of the Peace Lily pollen cell, providing insight to the performance of the 3D refractive index reconstruction algorithm. Fig. 7.3A shows the reconstructed intensity, revealing a circular structure corresponding to the cross section of the sample. The circular boundary suggests that the algorithm has captured the general morphology of the sample to some extent. However, several artifacts are visible, indicating potential issues within the reconstruction process.

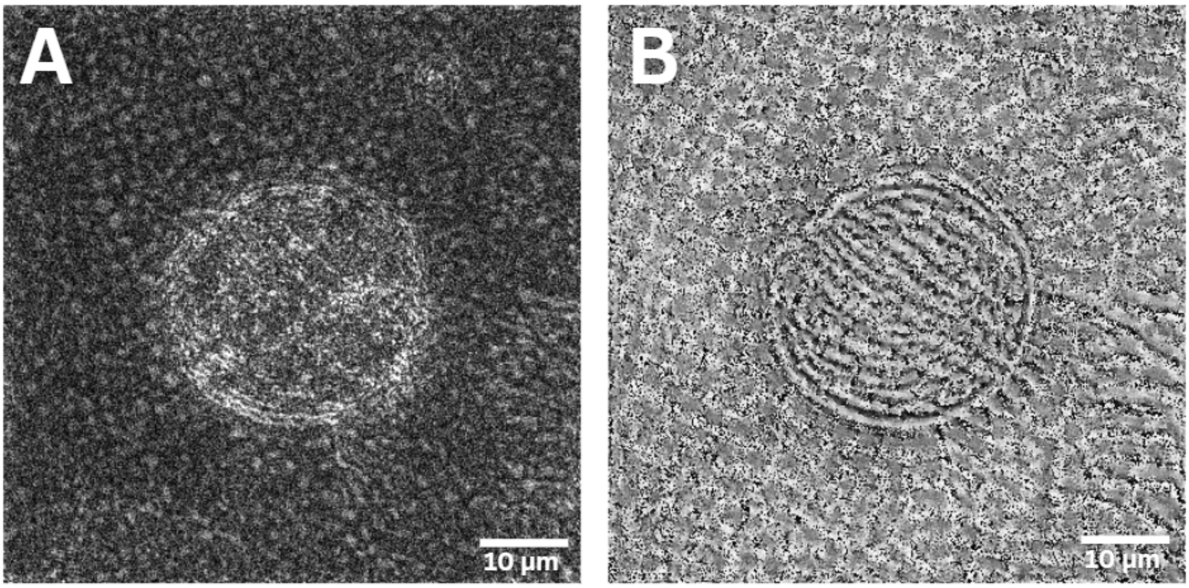


Figure 7.3: Reconstructed: Amplitude (A) and phase (B) at center of reconstructed cell.

Fig. 7.3B shows the reconstructed phase image, which, while also affected by artifacts, provides a more detailed view of the internal features of the sample. The concentric rings observed in the phase image suggest variations in the refractive index of the cell, possibly indicating internal structures. Although some structural details are visible, the results fall short of expectations, signaling issues in the reconstruction process. Although the scale of the reconstruction matches the expected size of the cell, implying that the setup can resolve micrometer-scale features, the accuracy of these features remains uncertain. Further investigation is required to identify and address the sources of artifacts, as discussed in Section 7.3. Additional work is needed to fully harness the potential of the MSBP algorithm and the imaging system.

Figure 7.4, presents both orthogonal (left panel) and three-dimensional (right panel) views of the reconstructed Peace Lily pollen cell, offering detailed 3D representations of the cell's internal structure. The left panel displays orthogonal views along the XY, YZ, and XZ planes. The XY view provides a top-down projection, revealing intricate internal details and shapes, suggesting a circular or spherical structure with varying intensity regions. These variations indicate differences in refractive index that highlight the complex texture of the cell and its internal organization. In contrast, the YZ and XZ views provide side projections, showing the cell's depth and density distribution. Areas of brighter intensity suggest higher density or refractive index variations, while the vertical perspectives reveal the cell's thickness and layered structure.

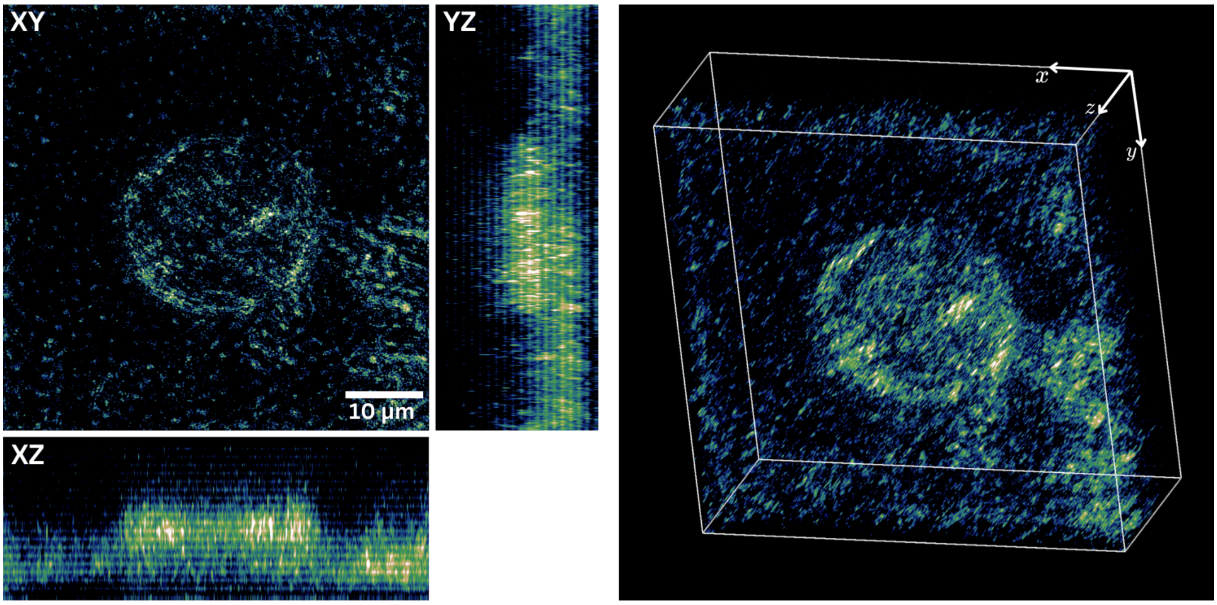


Figure 7.4: Orthogonal Views (Left panel) and 3-dimensional View (right panel) of Reconstructed Peace Lily Pollen cell. Acquired using Napari.

The right panel of Figure 7.4 shows a 3D representation of the reconstruction, which captures the complex internal structure of the cell with varying intensity, indicating regions of different refractive indices. The reconstruction provides insight into the depth and spatial distribution of the internal characteristics of the cell. However, the presence of noise and artifacts significantly impacts the clarity and accuracy of the reconstruction, pointing to the need for further refinement and optimization, as discussed in the following section.

7.3 Sources and Mitigation of Reconstruction Artifacts

Artifacts in the reconstructed refractive index distribution can arise from a combination of factors related to experimental setup, data quality, and the parameters of the MSBP algorithm, as well as sample preparation. Addressing these challenges is essential for improving the accuracy of the 3D refractive index reconstruction and reducing artifacts. The following subsections detail my hypothesis of what may be wrong with the reconstruction and possible solutions for solving these problems in the future, starting with the calibration of illumination angles.

7.3.1 Illumination Angle Calibration

The illumination angles play an important role in the reconstruction process. Precise knowledge of the illumination angles at the sample is crucial for accurate results. Although the illumination angles outputted by the motorized kinematic mount were calibrated, it appears that their accuracy may not have been sufficient for satisfactory 3D RI reconstruction. This could be due to factors such as system imperfections, sample-induced changes, or misalignment, as suggested by previous work [1]. I believe that the deviation of the calibrated illumination angles from the uncalibrated illumination angles, as shown in Figure 7.5, may have a large impact on the quality of the reconstruction. If the expected illumination angles on the sample are not accurate, they could introduce errors that affect the overall reconstruction, though further investigation would be required to confirm this.

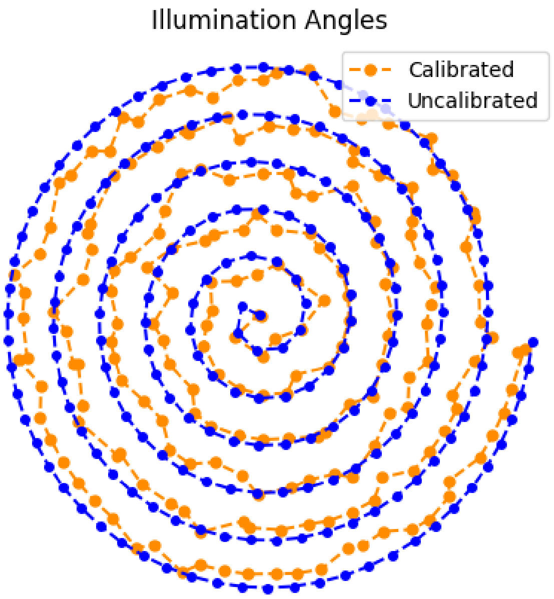


Figure 7.5: Calibrated and Uncalibrated illumination k-vectors for the spiral pattern overlaid on top of each other. This figure shows the heavy deviation of the calibrated spiral pattern from the expected spiral pattern.

In addition, having a small maximum illumination angle due to the NA may limit the range of possible illumination angles, potentially restricting the angular information available for reconstruction. I used a 40X which has a NA of 0.65, while the paper [1] on which this setup is based used a 100X with a NA of 1.45, allowing for steeper illumination angles. It is possible that increasing the NA of my setup could help improve the quality of the reconstructed data, but this also needs to be investigated further.

7.3.2 The Impact of Iteration Count on Reconstruction Quality

Parameter optimization plays a crucial role in ensuring accurate and stable 3D reconstructions. Poorly tuned parameters, such as the number of iterations, reconstruction dimensions, and regularization terms, can introduce artifacts and degrade reconstruction quality. One particularly important parameter in my algorithm is `maxiter`, which controls the number of iterations in the reconstruction process. Too few iterations can prevent convergence, resulting in inaccurate reconstructions, while too many can lead to overfitting, where the algorithm captures noise instead of the true signal. Striking a balance is essential to achieve both efficient convergence and high-quality results.

In retrospective, the reconstruction was probably overfitted as the algorithm was run for a total of 200 iterations. The convergence of the reconstruction can be seen in Figure 7.6. This reconstruction, as presented in Section 7.2, has visible artifacts that obscure finer details of the structure. The cost function plot shows a continuous decline over 200 iterations with no clear stabilization. While the algorithm continues to minimize the cost function, the resulting image quality suggests that the algorithm may have started to fit irrelevant noise or details in the data. This behavior is characteristic of overfitting, where excessive iterations can degrade the quality of the reconstruction rather than improve it.

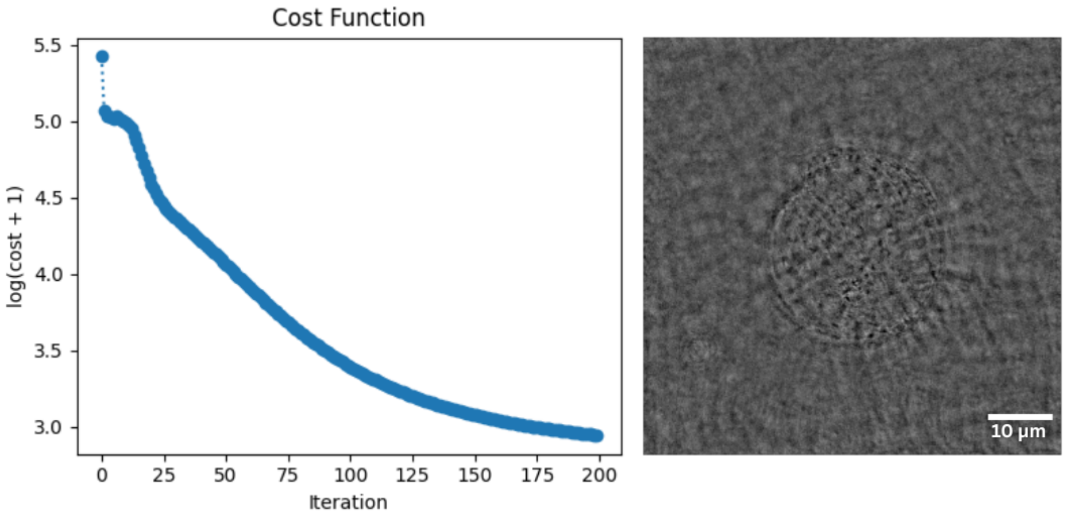


Figure 7.6: Unstable cost function and poor reconstruction from 200 Iterations. Cost function seems to steadily decline.

In contrast, applying the algorithm through only 10 iterations, the result is shown in Figure 7.7. In this reconstruction, the image appears clearer, with well-defined structures and minimal artifacts. The corresponding cost function plot shows a sharp decrease within the first few iterations, stabilizing around iteration 4. This indicates that the reconstruction algorithm converged quickly, capturing the relevant features. Rapid stabilization of the cost function suggests a good balance between fitting the data and avoiding overfitting, leading to higher quality reconstruction.

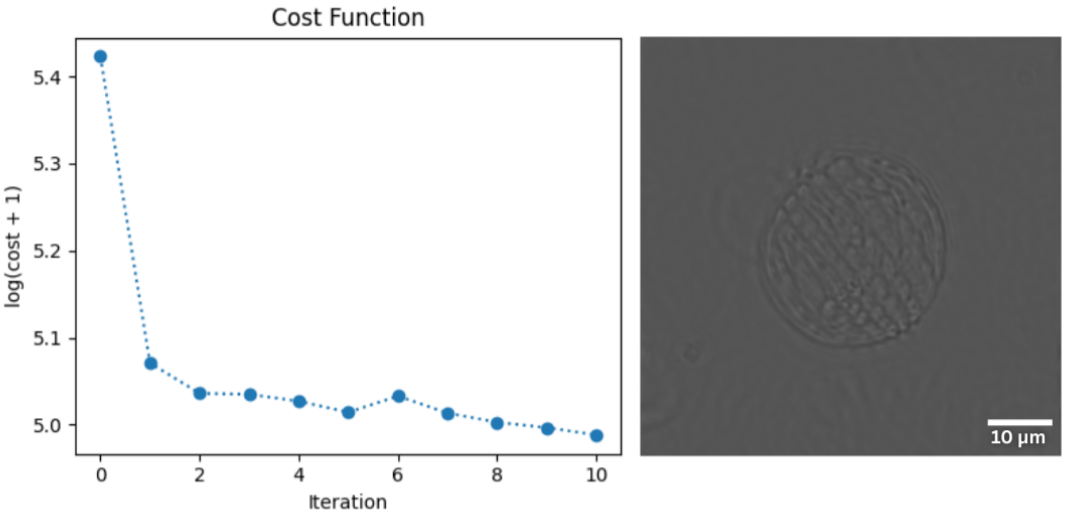


Figure 7.7: Stable cost function and good reconstruction from 10 Iterations. Cost function seems to stabilize around 4 iterations indicating fast convergence of the reconstruction algorithm.

These observations highlight the importance of controlling the number of iterations during optimization to achieve a reliable and high-quality reconstruction. Reflecting on the poor results presented in Section 7.2, it becomes clear to me where I went wrong. As the lack of convergence control likely contributed to the degraded image quality, as the algorithm overfit the data instead of capturing meaningful structures. With more time, I would explore this issue further and fine-tune the optimization process to strike the proper balance. By carefully limiting the number of iterations and analyzing the convergence behavior, the reconstruction could potentially achieve much better results. However, due to time constraints, this realization remains an afterthought, one that offers valuable insight for future work.

7.3.3 Other Relevant Reconstruction Parameters

Defining accurate values for the axial dimension of the reconstruction space (`o`) and the pixel size in the z-direction (`psz`) is fundamental as they define the axial resolution and volume size of the reconstructed object. These parameters must be carefully selected to match the characteristics of the sample being imaged as I found in Section 6.4.

The parameter `step_size` controls the magnitude of updates in gradient-based optimization and plays a critical role in the convergence behavior of the algorithm. If `step_size` is set too large, the reconstruction process may diverge, resulting in instability and potentially failing to find a solution. Conversely, a `step_size` that is too small can excessively slow down convergence, rendering the process inefficient. In addition to stability and efficiency, `step_size` also affects the algorithm's susceptibility to converging to local minima. When the step size is improperly tuned, the reconstruction process may become trapped in local minima, especially in complex reconstruction problems with non-convex cost functions. Carefully balancing `step_size` ensures that the algorithm makes stable, efficient updates while reducing the risk of stagnation in local minima. Building on the insights from Section 7.3.2, optimizing the maximum number of iterations in conjunction with the step size parameter will likely yield improved results. A careful balance between step size and iteration count is essential, as adjustments to one can influence the other. While this approach shows promise, further investigation is required to fully understand their combined impact on reconstruction quality.

Regularization techniques stabilize iterative convergence in reconstruction algorithms, particularly under noisy or ill-conditioned conditions [1]. However, their effectiveness depends on the balance of over and underregularization. Overregularization can blur fine details, such as smoothing sharp edges in the reconstructed RI distribution, while underregularization amplifies noise, producing artifact-laden reconstructions [11], as shown in Figure 7.3. Optimal regularization requires careful adjustment of the `regParam` parameter, which mitigates overfitting by penalizing the complexity of the solution. If `regParam` is too high, it oversmooths the reconstruction, erasing details. If too low, it amplifies the noise. Proper calibration, often achieved through experimental fine-tuning, ensures reconstructions that are smooth and detailed, preserving sample features while minimizing artifacts.

In conclusion, I believe that careful optimization of the parameters discussed will significantly improve the reconstruction process. Since the reconstruction algorithm itself is functional, as demonstrated by the benchmark in Section 7.1, the suboptimal results are probably due to unoptimized parameters. For future work, a more thorough exploration of how these parameters interact and affect the reconstruction will lead to improved results and better overall performance of the reconstruction algorithm.

7.4 Comparison of Reconstruction Performance in Commercial Microscopic Setups

To validate my reconstruction and make some comparisons with commercial systems, I took the Peace Lily pollen sample to a commercial Nikon eclipse Ti2 system. This allowed me to utilize both autofluorescent and gradient light interference methods for commercial-level reconstruction of the pollen cell. Thus, with this modality available, I can make a comparison between the reconstruction of my custom-built setup and that of a commercial microscope system.

7.4.1 Autofluorescence Imaging

The auto-fluorescence provides natural emission data from the sample, which can be captured and utilized for reconstruction, as shown in Figure 7.8A. Compared to my reconstruction in Figure 7.3, it is evident that the autofluorescence image exhibits significantly lower background noise. However, the sample itself lacks any discernible features, making it difficult to even identify it as a pollen cell. To address this, another fluorescent marker could be introduced, but that would make this method no longer label-free [13].

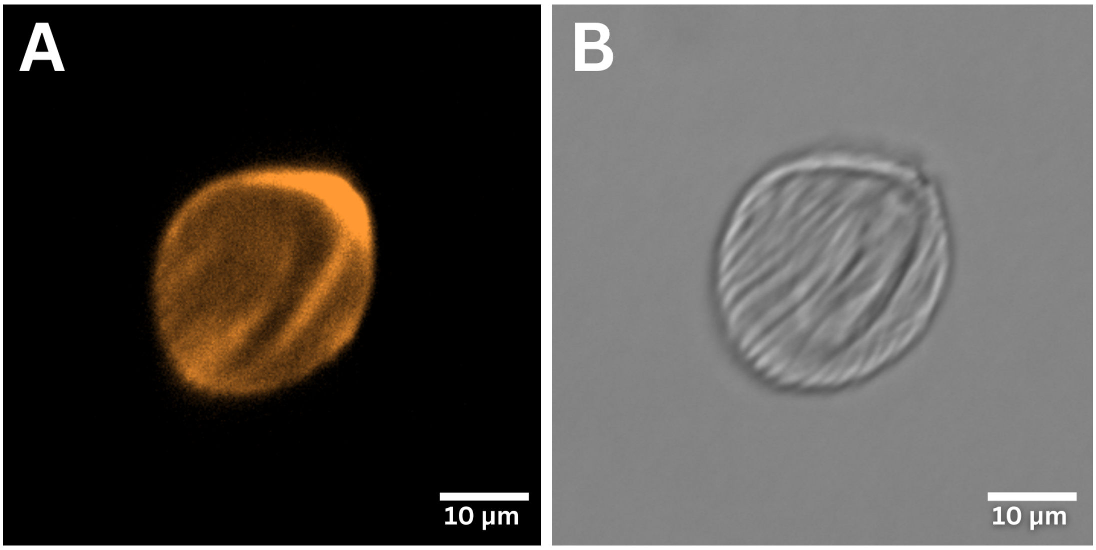


Figure 7.8: Reconstructed: Autofluorescence image (Panel A) and GLIM image (Panel B) of Peace Lily pollen cell. Acquired using a commercial Nikon eclipse Ti2 system with a 40X 0.95NA objective lens.

7.4.2 Gradient Light Interference Microscopy

The Gradient Light Interference Microscopy (GLIM) method offers good resolution and contrast with reduced background noise, as seen in Figure 7.8B. GLIM uses physical filtering techniques like high-pass filtering and Wiener deconvolution to suppress noise and improve optical sectioning. In contrast, MSBP relies on computational regularization, such as total variation minimization, to stabilize reconstruction, requiring precise illumination calibration for optimal performance. GLIM excels in label-free, high-contrast quantitative phase mapping of live cells, while MSBP is better for reconstructing 3D refractive index distributions of complex scattering samples, though it demands more computational resources.

8

Conclusion

In conclusion, after much *blood, sweat and tears*, this thesis has demonstrated the potential of the Multi-Angle Beam Propagation (MSBP) method as a powerful tool for high-resolution, label-free imaging of complex biological samples. The custom-built multi-angle illumination microscope was not able to completely reconstruct the 3D refractive index distributions of a Peace Lily pollen sample, but through this work I have highlighted several areas for improvement, pointing to challenges that need to be addressed in order to fully realize the system's potential.

8.1 Key findings

A key finding from this work was the importance of balancing the number of iterations and the step size. Both of these factors played a critical role in the quality of the reconstruction. Overfitting due to excessive iterations led to degraded image quality and obscured finer structural details. By carefully tuning these parameters, the reconstruction quality improved significantly, as demonstrated in the comparison of 200 iterations versus 10 iterations in Figure 7.7.

In addition, the accuracy of the illumination angle calibration and algorithm regularization could be crucial for achieving high-quality results. Inaccurate calibration of the illumination angles might contribute to increased noise and artifacts, since the algorithm relies on precise angle definitions for accurate modeling of the light-sample interaction. Similarly, improper regularization—whether over or underregularization could introduce distortions and artifacts, suggesting that further optimization of these parameters may be necessary to improve the results.

Other factors affecting the reconstruction may include field curvature and imperfections in the experimental setup. The presence of spherical aberration, exacerbated by the lack of an immersion medium, introduced non-uniform phase variations that were not accounted for in the reconstruction. Addressing these issues, such as incorporating an immersion medium, for both objectives, to reduce aberrations, could significantly enhance the quality of the reconstructions.

Despite these challenges, the benchmark validation of the reconstruction algorithm using simulated phantom data confirmed its functionality and accuracy, indicating that the suboptimal results were likely due to unoptimized parameters or experimental data quality issues rather than errors in the algorithm itself. Comparison with commercial microscopy systems, such as autofluorescence imaging and Gradient Light Interference Microscopy (GLIM), highlighted the strengths and limitations of the MSBP method.

In summary, while the current setup and reconstruction process have shown promising results, significant improvements are needed to address the identified challenges. Future work should focus on refining the alignment of the experimental setup as well as optimizing the highlighted reconstruction parameters to improve the accuracy of the 3D refractive index reconstruction. By addressing these areas, this custom-built microscopic setup utilizing the MSBP method has the potential to become a robust and versatile tool for label-free high-resolution imaging, with significant applications in biological and materials science research.

8.2 Future Work

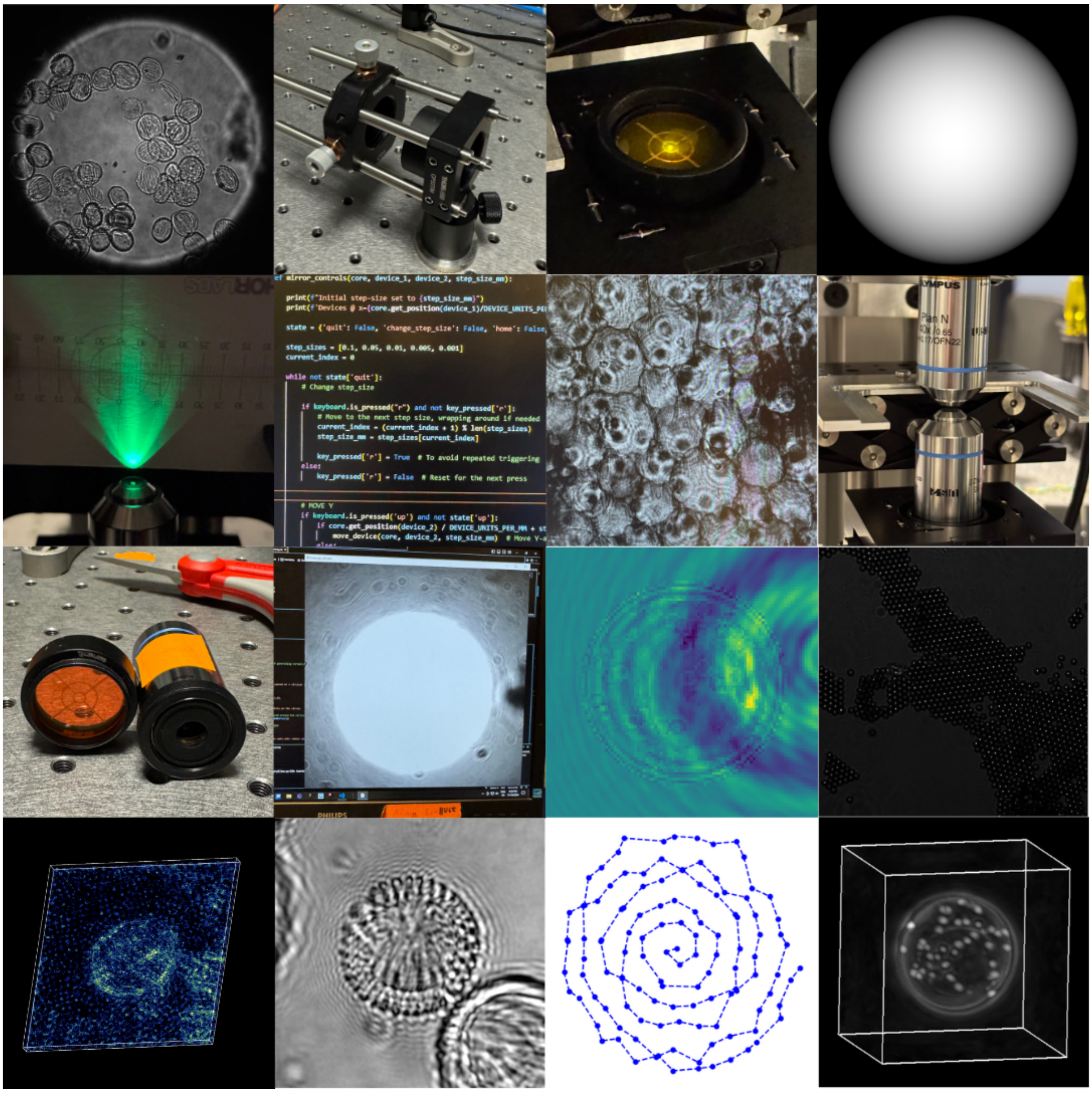
Given more time to dedicate to this project, I would focus on improving both the user-friendliness of the microscopic setup and the efficiency of the reconstruction process. This would involve refining the interface and optimizing the workflow to ensure smoother operation, as well as fine-tuning the reconstruction algorithms to enhance accuracy and reduce processing time. Future work on the experimental setup should consider the following improvements:

1. Add motorized XY stage to sample holder, so that sample field of view can be more precisely scanned
2. Implement illumination calibration algorithm in Python to improve workflow from acquisition to reconstruction
3. Create easy-to-install library and software documentation
4. Graphical interface to allow user to control parameters for reconstruction
5. Experiment with reduction of necessary illumination angle amount needed for quality reconstruction
6. Explore other possible reconstruction methods for the custom-built multi-angle illumination system

Bibliography

1. Chowdhury, S. *et al.* High-resolution 3D refractive index microscopy of multiple-scattering samples from intensity images. *Optica* **6,** 1211–1219 (2019).
2. Faucher, F., Kirisits, C., Quellmalz, M., Scherzer, O. & Setterqvist, E. in *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging* 273–312 (Springer International Publishing, 2023). ISBN: 9783030986612. http://dx.doi.org/10.1007/978-3-030-98661-2_115.
3. Müller, P., Schürmann, M. & Guck, J. *The Theory of Diffraction Tomography* 2016. arXiv: 1507.00466 [q-bio.QM]. https://arxiv.org/abs/1507.00466.
4. Park, Y., Depeursinge, C. & Popescu, G. Quantitative phase imaging in biomedicine. *Nature Photonics* **12,** 578–589. https://doi.org/10.1038/s41566-018-0253-x (2018).
5. Devaney, A. J. A Filtered Backpropagation Algorithm for Diffraction Tomography. *Ultrasound Imaging* **4,** 336–350 (1982).
6. Stave, M. N. *Computational 3D Reconstruction of Refractive Index Distribution - UIT Prosjekt Oppgave* UIT Prosjekt Oppgave. 2024.
7. Su, T.-W. *et al.* Multi-angle lensless digital holography for depth resolved imaging on a chip. *Optics Express* **18,** 9690–9711 (2010).
8. Team, M.-M. *Micro-Manager Thorlabs Kinesis Device Adapter Releases* Accessed: 2024-12-02. 2024. https://github.com/micro-manager/mmdev-ThorlabsKinesis/releases.
9. Team, M.-M. *Pycro-Manager Documentation: Headless Mode* Accessed: 2024-12-02. 2024. https://pycro-manager.readthedocs.io/en/latest/backends.html#headless-mode.
10. Eckert, R., Phillips, Z. F. & Waller, L. Efficient illumination angle self-calibration in Fourier ptychography. *Applied Optics* **57,** 5434–5442 (2018).
11. Chowdhury, S. *et al.* *Supplementary document for High-resolution 3D refractive index microscopy of multiple-scattering samples from intensity images - 4044047.pdf* Journal contribution. 2019. https://doi.org/10.6084/m9.figshare.9227360.v1.
12. Kim, G. *et al.* Three-dimensional label-free imaging and analysis of Pinus pollen grains using optical diffraction tomography. *Scientific Reports* **8,** 1782. https://doi.org/10.1038/s41598-018-20113-w (2018).
13. Croce, A. Light and Autofluorescence, Multitasking Features in Living Organisms. *Photochem* **1,** 67–124 (2021).

Interesting Images Taken Throughout the Development Process



```
ef mirror_controls(core, device_1, device_2, step_size_mm):

    print(f"Initial step-size set to {step_size_mm}")
    print(f'Devices @ x={core.get_position(device_1)/DEVICE_UNITS_PE

    state = {'quit': False, 'change_step_size': False, 'home': False,

    step_sizes = [0.1, 0.05, 0.01, 0.005, 0.001]
    current_index = 0

    while not state['quit']:
        # Change step_size

        if keyboard.is_pressed("r") and not key_pressed['r']:
            # Move to the next step size, wrapping around if needed
            current_index = (current_index + 1) % len(step_sizes)
            step_size_mm = step_sizes[current_index]

            key_pressed['r'] = True  # To avoid repeated triggering
        else:
            key_pressed['r'] = False  # Reset for the next press

        # MOVE Y
        if keyboard.is_pressed('up') and not state['up']:
            if core.get_position(device_2) / DEVICE_UNITS_PER_MM + st
                move_device(core, device_2, step_size_mm)  # Move Y-a
```

Figure 8.1: Some interesting images that were taken through the development process.

/A
Programming Libraries

A.1 Numpy

NumPy is a fundamental library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. NumPy is widely used in scientific computing, data analysis, and machine learning due to its performance and ease of use.

A.2 Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib provides a MATLAB-like interface for plotting graphs, histograms, bar charts, and other types of visualization. Matplotlib is commonly used in data analysis, research, and education to visualize data and communicate results effectively.

A.3 PycroManager

PycroManager is a Python library that provides a bridge between Python and Micro-Manager, a software platform for controlling microscope hardware. It allows users to write Python scripts to control microscope components, acquire images, and process data in real-time. PycroManager is essential for researchers and developers working on automated microscopy and high-throughput imaging.

A.4 Threading

The Python threading library provides a high-level interface for creating and managing threads. Threads allow concurrent execution of multiple tasks within a single program, which can be useful for improving performance in I/O-bound or parallelizable tasks. The threading library is commonly used in applications that require background tasks, such as network servers or GUI applications.

A.5 Keyboard

The keyboard library is a Python package that allows for the detection and control of keyboard events. It can be used to capture key presses, simulate key presses, and create keyboard shortcuts or macros. This library is particularly useful for automation scripts, gaming, and applications that require custom keyboard input handling.

A.6 Cupy

CuPy is a GPU-accelerated library for numerical computing in Python. It provides a NumPy-compatible interface, allowing users to take advantage of the power of GPUs for faster computations. CuPy is particularly useful in machine learning, deep learning, and other computationally intensive tasks where GPU acceleration can significantly improve performance.

A.7 Napari

Napari is an open-source, Python-based tool for visualizing and analyzing multidimensional scientific data, such as microscopy images. It offers an intuitive interface for viewing, annotating, and processing large datasets, with support for various data formats and seamless integration into Python workflows. Its plugin system allows customization, making it adaptable for diverse research needs, while its performance is optimized for handling large-scale data efficiently.

/B

Programs used

B.1 ImageJ

ImageJ is a public domain Java-based image processing program. It is widely used in scientific research for image analysis, including tasks such as image filtering, measurement, and segmentation. ImageJ supports a wide range of plugins and macros, making it highly customizable for various applications in biology, medicine, and materials science.

B.2 Micromanager

Micro-Manager is an open source software platform for controlling automated microscopy hardware. It provides a user-friendly interface for configuring and controlling microscope components, such as cameras, stages, and light sources. Micro-Manager is commonly used in life sciences research for high-throughput imaging, time-lapse experiments, and multi-dimensional imaging.

B.3 Thorlabs Kinesis

Thorlabs Kinesis is a software suite provided by Thorlabs to control their range of scientific instruments, such as motorized stages and other types of precision equipment. Kinesis offers a graphical user interface and a programming interface to control these devices, making it a valuable tool for researchers in fields such as optics, photonics, and precision engineering.

B.4 Visual Studio Code

Visual Studio Code (VS Code) is a lightweight but powerful source code editor developed by Microsoft. It supports a wide range of programming languages and includes features such as debugging, task running, and version control integration. VS Code is highly extensible, with a large ecosystem of plugins and extensions, making it a popular choice for developers in various domains, including web development, data science, and scientific computing.

C

Pre-Project Paper

The following is my pre-project paper (June 2024). This paper served as preparatory work for the Master Thesis presented above. I have included it to provide an essential context and to ensure the completeness of the thesis.

Computational 3D Reconstruction of Refractive Index Distribution

Prosjektoppgave - Anvendt Matematikk og Fysikk
Candidate #8

UIT - December 9, 2024

abstract

This paper details a computational reconstruction of 3D refractive index distribution using optical diffraction tomography. The study replicates the work of Chowdhury et al. [1], implementing a Multi-Slice Beam Propagation (MSBP) forward model in Python to simulate light propagation through biological samples. The model utilizes Fourier optics principles to reconstruct 3D refractive index distributions from holographic microscopy data. The Python implementation is validated against the original MATLAB code, demonstrating accurate reconstruction of a phantom object's refractive index. This paper also discusses the limitations and potential improvements of the model, highlighting the importance of efficient data acquisition and noise reduction in achieving high-resolution 3D imaging of biological samples.

Preface

To the reader

In this project thesis, I will detail the preliminary work I have done so far for my Master's project, as well as future improvements and areas that can be explored.

This project entails the application of Wave optics and Fourier optics, which I have studied, analyzed, and built upon to replicate earlier work done in the microscopy field and ultimately expand on my ideas and vision for my Master's project.

To familiarize myself with current-day approaches in the field of computational diffraction tomography, I have replicated the work of a paper [1] for which I have carried out a theoretical analysis and translated their MATLAB code into my own Python algorithm for further development. Leveraging their implementation, the Master's project aims to adapt and refine their algorithm for its own unique experimental setup and objectives.

Contents

1 Introduction

If you have ever been by the ocean on a sunny day, you have probably noticed the mesmerizing pattern the sunlight casts on the ocean floor. This optical phenomenon is caused by the rough ocean surface refracting the sun rays, scattering them onto the ocean floor as light and dark patterns. A physicist on vacation might then ask themselves, how exactly does this happen?

In the realm of physics, we seek to create models to help us understand and explain the underlying mechanisms of phenomena we see in the real world. This pursuit frequently entails solving inverse problems, in which the cause is deduced from the observed effects. In our example we might want to create a model of the ocean surface (cause) from the patterns projected on the ocean floor (effect).

Advances in light-matter interaction models have led to innovations like digital holographic microscopy, enabling detailed reconstruction and imaging of microscopic sample structures. While providing the ability to acquire phase information, which is often unavailable through traditional microscopic setups.

Building upon this foundation, this thesis serves as the groundwork for the development of a multi-angle holographic microscope for 3D imaging of biological samples. Drawing from the computational work of Chowdhury et al.[1], this project entails the implementation of the multi-slice beam-propagation forward model (MSBP) to simulate propagation and diffraction, as well as an inverse model to reconstruct the samples' refractive index distribution.

2 Principles of Optical Imaging

To study the diffraction phenomenon, we need to consider light as a wave. In wave optics, light is characterized by the scalar wave function, representing the electric field strength at each point in space ($\mathbf{r}$) and time (t). The majority of the content within this chapter is in reference to Saleh et al. [2].

The Wave Function is expressed as:

$$U(\mathbf{r}, t) = U(\mathbf{r}) \exp(j2\pi\nu t), \tag{1}$$

where $U(\mathbf{r})$ denotes a time-independent complex amplitude at position $\mathbf{r}$ and $\exp(j2\pi\nu t)$ represents the time-dependent oscillatory component of the wave function. The complex amplitude term, defined as $U(\mathbf{r}) = a(\mathbf{r}) \exp[j\varphi(\mathbf{r})]$, encapsulates both the **amplitude** $|U(\mathbf{r})| = a(\mathbf{r})$ and the **phase** $\arg\{U(\mathbf{r})\} = \varphi(\mathbf{r})$ of the wave, offering crucial insights into the fundamental properties of waves. The amplitude measures the magnitude of the wave (intensity of light), and the phase indicates the wave's position within its cycle.

The Plane Wave and Wave-fronts

A special case of the wave function is the **plane wave**. The plane wave has a constant amplitude and frequency at which it propagates in a direction defined by a constant wave-vector. The plane wave can be described by the following wave function:

$$U(\mathbf{r}, t) = A \exp(-j(\mathbf{k} \cdot \mathbf{r} - \omega t)), \tag{2}$$

where A is a complex constant representing the strength of the wave, $\mathbf{k}$ is the wave-vector, $\mathbf{r}$ is the position vector, ω is the angular frequency, and t is time. The wave-vector $\mathbf{k}$ points in the direction of wave propagation and its magnitude is the wave-number $k = 2\pi/\lambda$, where λ is the wavelength of the wave.

Wave-fronts are surfaces of constant phase in a wave field. For a plane wave, these wave-fronts are planes that are parallel to each other and perpendicular to the direction of wave propagation. The concept of wave-fronts is essential for understanding wave phenomena such as interference and diffraction, as they represent the direction at which the rate of change of the phase is at its maximum. When the peaks and valleys of two waves coincide, they interfere constructively. Conversely, if two wave have a phase difference of half wavelength they interfere destructively, as the peak of one wave coincide with the valley of another.

Amplitude and phase together provide a comprehensive view of a wave's variation across space and time, making them crucial in microscopy. In traditional microscopy, we can only measure light intensity. Without staining, we miss the phase information of the light wave. In contrast, holographic microscopy uses coherent light to illuminate a sample, capturing an interference pattern between the sample's light wave and a reference wave, thus recording both amplitude and phase information. This recorded interference pattern is called a hologram. However, the hologram doesn't directly show any details. To reveal the sample's information, we must decode this pattern. This is achieved through computational reconstruction of the hologram, which then reveals details about the sample's thickness, density, and refractive index distribution.

2.1 The Wave Equation

The Wave Function mentioned above provides information about a wave at a specific **point** in space and time. Namely where it is and how strong it might be. Taking this a step further we can also describe how a wave can **move** and **change** over space and time. This is where the wave equation comes in.

The Wave Equation governs how waves are allowed to propagate and evolve over time. It is defined as the partial differential equation:

$$\nabla^2 U - \frac{1}{c^2}\frac{\partial^2 U}{\partial t^2} = 0, \tag{3}$$

where the left side describes the spatial variation of the wave, through the Laplace operator ∇^2, and the right side encapsulates how the wave field evolves in time. The Laplace Operator describes how the wave function changes as function of position and is defined as:

$$\nabla^2 U = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} \tag{4}$$

The Helmholtz equation, which is derived from the wave equation, separates the space and time components. The Helmholtz equation is defined as:

$$\nabla^2 U + k^2 U = 0, \tag{5}$$

where k is the wave-number defined as $k = \frac{2\pi n}{\lambda}$ for which n is the refractive index of the material the wave is propagating through. This decoupling, of the space and time components of the wave equation, allows us to focus primarily on the spatial aspects of the wave function, simplifying the analysis of how waves behave in space. The Helmholtz equation describes the wave-fronts by only using the complex amplitude, encapsulating both the magnitude and the phase of the wave in space.

2.2 Transmittance and Diffraction

Now that we have an understanding of how wave fields evolve in space and time we move on to describe the **transmission** of waves through media. In terms of our introduction example, the transmittance describes what the ocean does to the sun-light as it passes through to the ocean floor.

Similarly, and in theme of this thesis, lets imagine we apply an incident field of light on a translucent sample. In this case the transmittance refers to the modification of the light wave as it passes through the sample. This includes changes in both the amplitude and phase of the wave due to the sample's refractive index and thickness variations. The complex amplitude of the light wave after interaction with the sample can be represented as:

$$U_s(x,y) = t(x,y) \cdot U_i(x,y), \tag{6}$$

where U_s is the complex amplitude of the light wave at a specific plane of z. This is a result of the transmittance function of the sample $t(x,y)$ modifying the incident light wave U_i. The transmittance function here captures both the amplitude and phase changes caused by the sample's refractive index distribution and thickness variation across the x-y plane and along the z-axis.

Diffraction, on the other hand, refers to the process that occurs after the light wave has passed through the sample. The now modified wave field U_s start to interfere with a reference wave as they propagate outwards, away from the sample. This interference causes the light to constructively add together in some parts of the wave field and destructively subtract in other parts. The intensity distribution of the diffraction pattern can be defined as:

$$H(x,y) = |U_s(x,y) + U_r(x,y)|^2, \tag{7}$$

where U_r is the reference wave-field. The result of this is the intensity pattern we see on the ocean floor, namely the diffraction pattern.

Understanding wave optics lays the foundation for exploring the transmittance of light through material, but it alone cannot explain how diffraction works. To thoroughly understand diffraction, we need to proceed to climb another step on the optical ladder, namely to Fourier Optics.

2.3 Fourier Optics

According to the **Fourier theorem**, a two-dimensional function of space $f(x,y)$ can be decomposed into a linear superposition of harmonic functions with different frequencies and complex amplitudes. Considering a harmonic function $F(\nu_x,\nu_y)\exp(j2\pi(\nu_x x + \nu_y y))$ having frequency components (ν_x,ν_y), amplitudes $F(\nu_x,\nu_y)$ and phase $\arg\{F\}$, the $f(x,y)$ can be defined as the **inverse Fourier transform**:

$$f(x,y) = \iint_{-\infty}^{\infty} F(\nu_x,\nu_y)\exp[-j2\pi(\nu_x x + \nu_y y)]d\nu_x d\nu_y, \tag{8}$$

where the coefficients $F(\nu_x,\nu_y)$ are determined by finding the **Fourier transform**:

$$F(\nu_x,\nu_y) = \iint_{-\infty}^{\infty} f(x,y)\exp[j2\pi(v_x x + v_y y)]dxdy \tag{9}$$

Essentially, the Fourier transforms, converts our function from spatial domain (x,y) into the frequency domain (ν_x,ν_y). This conversion can be utilized to our advantage as we will see in the next section.

2.4 Linear and Nonlinear Systems

In linear systems theory, we treat optical systems as linear operators that transform input signals, like a light wave, into output signals, like diffraction pattern. This framework allows us to analyze the behavior of optical systems using mathematical tools such as convolution and Fourier transforms. The linear systems theory facilitates the design and analysis of complex optical systems for various applications, both in the field of imaging as well as communication.

Linear Systems

A system is considered linear when it adheres to the principle of superposition. This means that the system's response to the sum of two inputs is equivalent to the sum of its responses to each input individually. For example, a two-dimensional system establishes a relationship between a two-dimensional input function $f_1(x, y)$ and output function $f_2(x, y)$. Within the scope of this paper, these functions can represent optical fields at two parallel planes separated by some distance of free space. The output function of our linear system is defined as convolution integral:

$$f_2(x, y) = \iint_{-\infty}^{\infty} h(x - x', y - y') f_1(x', y') dx' dy', \tag{10}$$

where the function $h(x - x', y - y')$ is the **impulse response** function of the system (also known as the point-spread function). The impulse response function represents a weighting function that applies the effect of the input at point (x', y') on the output at the point (x, y). Additionally, if the system responds to two spatially separated points of the input field similarly, we regard the system as shift-invariant.

If the system is shift-invariant, we can utilizing the Fourier theorem and introduce the **convolution theorem**, which greatly simplifies the computation of the convolution integral. Instead of solving the convolution analytically through a large matrix-vector multiplication, the convolution theorem allows us to perform the operation efficiently in the frequency domain by transforming the convolution integral into a simple point-wise multiplication.

The convolution theorem is defined as:

$$f(x, y) * g(x, y) = F^{-1}\{F\{f(x, y)\} \cdot F\{g(x, y)\}\}, \tag{11}$$

where $f(x, y)$ and $g(x, y)$ are functions in the spatial domain, $F\{\cdot\}$ denotes the Fourier transform, $F^{-1}\{\cdot\}$ denotes the inverse Fourier transform, ($*$) represents convolution and ($\cdot$) represents multiplication.

By applying the convolution theorem, one can compute the diffraction pattern by multiplying the Fourier transforms of the wave field and the aperture function, rather than performing the convolution directly, which can be computationally intensive. The convolution theorem [3] is widely used in optical systems analysis, including the design of lenses, filters, and other optical components. It is also the basis for techniques such as coherent optical processing and digital holography. Convolution is deeply rooted in the mathematical framework of linear systems theory, as we will discuss in the next section.

Applying the convolution theorem, we can simplify equation (10) as:

$$F_2(\nu_x, \nu_y) = H(\nu_x, \nu_y) \cdot F_1(\nu_x, \nu_y), \tag{12}$$

where $F_2(\nu_x, \nu_y)$, $H(\nu_x, \nu_y)$, and $F_1(\nu_x, \nu_y)$ are the Fourier transforms of $f_2(x, y)$, $h(x, y)$ and $f_1(x, y)$, respectively. In Fig.1 we see an example of such a system, which simulates diffraction patterns. The input field (object function) is a amplitude distribution of the light field at the input plane. This is defined as two circular objects. Our transfer function $h(x, y)$ is the propagated field of a point source. By taking an inverse Fourier transform of F_2, we find the output field f_2.

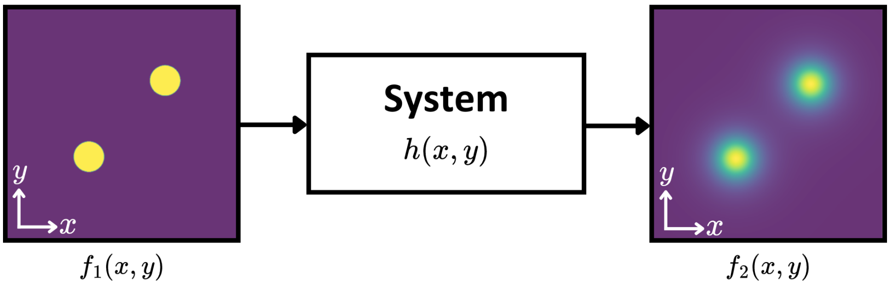


Figure 1: Response of a 2D linear shift-invariant system.

Nonlinear systems

A nonlinear system [4], on the other hand, is a system in which the output is not directly proportional to the input. In contrast to a linear system, the response of a nonlinear system does not scale linearly with the input, and the superposition principle does not hold. Thus, the system's response to multiple inputs is not simply the sum of its responses to the individual inputs.

In wave optics, a nonlinear system can, for example, be characterized by a medium that has a refractive index distribution that is dependent on the intensity of the light. This means that the speed at which light travels through the medium changes according to the local intensity of the light wave. In linear optics, the refractive index is assumed to be independent of light intensity.

3 Advanced Theoretical Frameworks

A **simulation** represents a process or system, allowing us to predict and understand the behavior of systems that are too complex or costly to study directly through experiments or analytical methods. In this project, we are simulating the propagation, transmittance, and interference of light through an approximated model of a biological cell. Thus in this chapter, I will demonstrate how we can simulate the interaction of light with a transparent object using scalar diffraction theory.

3.1 Numerical Foundations: Computational Mathematics

A **mathematical model** is an abstract representation of a process, system, or phenomenon [5]. It is a tool scientists and engineers can use to analyze, predict, and understand how complex systems work by breaking them down into simple, quantifiable components. These models can take various forms, including equations, graphs, and computer simulations. To make accurate models, we make predictions, perform experiments, and compare and improve our model with the help of the already-established theory, experimental measurements, and simulation results.

A model is composed of mathematical equations that encapsulate the dynamics of a given system. Simulation involves running software that executes algorithms to solve these complex equations. Simulation has become an important tool in science, enabling the implementation of highly intricate models beyond the scope of manual calculations.

In Fig.2 we observe the process of creating a computational model which requires a collaborative effort between experimentation, simulation, and theoretical analysis. This involves crafting a digital representation that reflects the behaviour of a system, based on empirical data, experimentation through simulation, and applying well-established theoretical concepts. The synergy between these components is essential, as it facilitates the refinement and validation of the model, ensuring its accuracy and effectiveness in predicting and interpreting real-world phenomena.

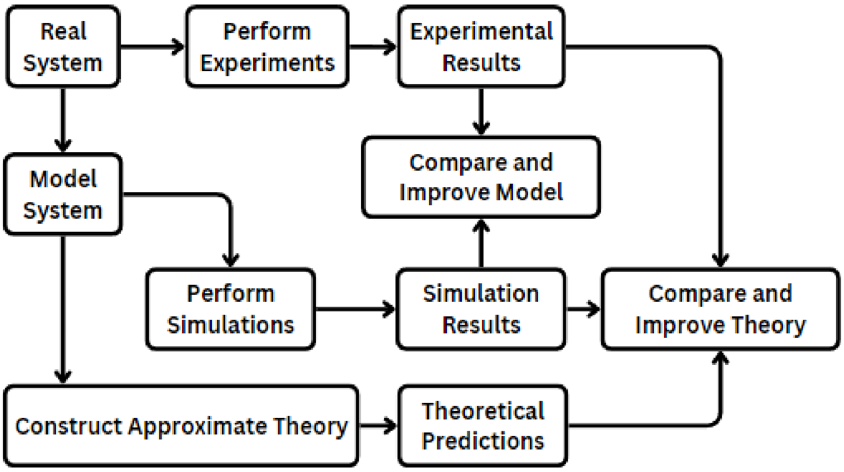


Figure 2: Process of building a computational model and the interplay between experimentation, simulation, and theory.

Discretization

Discretization is a fundamental process in numerical methods, involving the sampling of continuous functions, variables, or systems into discrete parts. This approximation allows computers to process continuous data, which is crucial for developing numerical models and algorithms to solve complex mathematical problems that cannot be addressed analytically.

In imaging, discretization means capturing discrete samples of a continuous light intensity variation across a biological cell, resulting in a digital image composed of pixels. According to the Nyquist-Shannon sampling theorem, if the sampling rate is more than twice the highest frequency of the signal, the continuous signal can be accurately reconstructed from its samples.

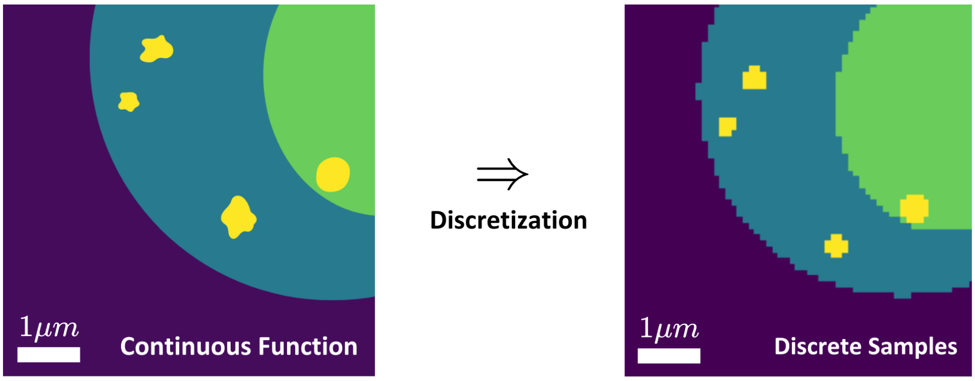


Figure 3: Example of the discretization or pixelation of a biological cell.

In this paper, we simulate biological samples by discretizing the data to facilitate numerical computations. Our model approximates the refractive index distribution of a cell's structure, representing each pixel as 0.1 microns. Unlike real biological cells with continuous layers and smooth boundaries, our simulated cell is pixelated. A high enough sampling rate ensures that the digital image accurately represents the structure and variations of the cell, capturing sufficient detail to prevent information loss, as shown in Fig.3. If the pixel size were too large, we might miss important details, leading to a loss of information.

3.2 Scalar Diffraction Theorem

The **scalar diffraction theorem** (SDT) is a fundamental principle in wave optics, which describes how light propagating through space interacts with apertures and obstacles. Light is originally described as a electromagnetic vector field composed of electric and magnetic vector components, which are explained through the well known Maxwell's equations. The SDT allows us to approximate the wave field of light as a **scalar field**. By ignoring the vector nature of electromagnetic waves we simplify the analysis of wave propagation while still capturing the essential parts of the diffraction phenomenon.

In scalar approximation, the electric field $E(x, y, z)$ can be approximated as a scalar field $U(x, y, z)$. This simplifies the wave equation allowing us to utilize the Helmholtz equation (Eq.5) in our derivation of the diffraction pattern. The Helmholtz equation describes how a scalar field propagates in a homogeneous medium.

Huygens-Fresnel Principle

The Huygens-Fresnel principle is a fundamental concept in wave theory. It proposes that every point on a wavefront acts as a secondary source of spherical wavelets. We can therefore approximate the next wavefront, at some distance away, as the envelope/superposition of these secondary wavelets.

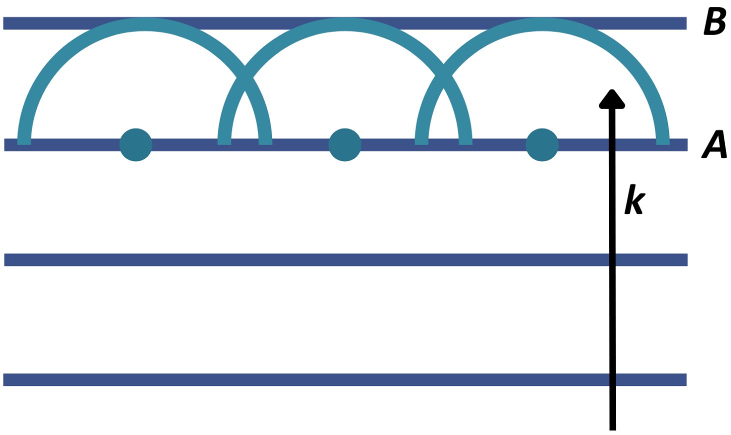


Figure 4: Wavefront propagation as superposition of wavelets. Wavelets from wavefront A constructively interfere into wavefront B.

If we approximate the light wave to be paraxial and weakly scattering, we can utilize **the Huygens-Fresnel diffraction integral**. This integral is a practical approximation derived from the more general Helmholtz equation through the application of Kirchhoff's integral theorem. It allows for the computation of diffraction patterns through an aperture or object by applying the Huygens-Fresnel principle. We can define the Huygens-Fresnel diffraction integral as:

$$U(r) = \frac{A}{i\lambda} \iint_{aperture} \frac{e^{ik|r-r'|}}{|r-r'|} \cos(\theta) U(r') dS' \tag{13}$$

where $U(r')$ is the wave amplitude at the aperture, θ is the angle between the incident wave and the line connecting r and r', A is a constant, λ is the wavelength and dS' is an element of the aperture area. This integral encapsulates the aforementioned Huygens-Fresnel principle by describing how the wavelets of the wave field emerging from an aperture combine to form the observed diffraction pattern. By considering the contributions from all points in the aperture, it provides a complete description of the diffraction phenomenon.

3.3 Free Space Propagation Using The Angular Spectrum Method

Free space propagation refers to the phenomenon where light propagates through free space, defined as a homogeneous medium with some uniform refractive index, like air or vacuum. **The transfer function** of free space describes how different spatial frequency components of an optical field propagate through free space and is defined as:

$$H(\nu_x, \nu_y, z) = \exp\left(i2\pi z\sqrt{\frac{1}{\lambda^2} - \nu_x^2 - \nu_y^2}\right), \tag{14}$$

where λ is the wavelength of the light, z is the propagation distance and (ν_x, ν_y) are the spatial frequencies. The transfer function $H(\nu_x, \nu_y)$ can be applied to the a wave field using convolution, to modify the phase of each spatial frequency component, depending on how far it travels in free space.

The angular spectrum method [2] can be used to propagate optical fields over free space, by applying a convolution formula with a shift-invariant free-space transfer function. First we can decompose a wave field $U(x, y : z = 0)$ situated at the initial plane $z = 0$, using the Fourier transform:

$$U(\nu_x, \nu_y : z = 0) = F\{U(x, y : z = 0)\} \tag{15}$$

Then each spatial frequency component of $U(\nu_x, \nu_y : z = 0)$ can be propagated separately in free space by applying the free space transfer function $H(\nu_x, \nu_y)$ using convolution. This is defined as:

$$U(\nu_x, \nu_y : z) = H(\nu_x, \nu_y : z) \cdot U(\nu_x, \nu_y : z = 0) \tag{16}$$

Finally, the propagated field $U(x, y : z)$ can be found by taking the inverse Fourier transform of the propagated spectrum:

$$U(x, y : z) = F^{-1}\{U(\nu_x, \nu_y : z)\} \tag{17}$$

By applying the angular spectrum method, we can compute how a wave develops over a certain distance in free space. This method will prove useful in later sections.

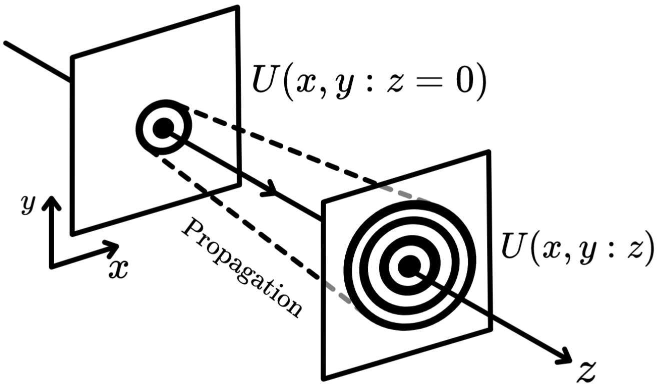


Figure 5: Optical field at $U(x, y : z = 0)$ is propagated in the positive z-direction, until sensor-plane $U(x, y : z)$.

3.4 Reconstruction: The Forward problem and The Inverse Problem

Most of the problems we encounter in physics and engineering entail solving systems of linear equations. Such systems can be represented using the matrix form, as:

$$Ax = b, \tag{18}$$

where A is a known transformation matrix, x a unknown vector we wish to determine and b is the vector of measured data. In a **Forward problem**, the solution is simple. We know A and x and can thus determine b. In reality however, systems are not always as straight forward.

In the **Inverse problem** [6], on the other hand, we know A and b, and need to determine x. One would imagine that we could simply express x by taking the inverse of A, defined as:

$$x = A^{-1}b, \tag{19}$$

but, this is not the case. There are several practical issues with the inverse problem. For one, A might not be invertible, meaning we cannot simply compute it's inverse. The measured data of b can have small errors which will lead to larger errors in x. Further more, in real life, the measurements of vector b often includes noise. Meaning that b won't be exact. Thus taking a direct inverse would amplify the differences, leading to inaccurate results. In the context of this thesis, which applies the holographic approach to sample imaging, x would be defined as the 3D refractive index distribution of the sample that we want to recover. A is the transmittance of light through the sample, thus A decodes the RI distribution x, as intensity images defined as b.

To solve this inverse problem we can apply the **Gradient descent method**. We make an initial guess of the RI distribution x and perform comparisons. We are interested in the difference between the resulting measurements b of our guess compared to the ground truth. We can formulate this difference as the **objective function**:

$$J(x) = \frac{1}{2}||Ax - b||^2 \tag{20}$$

We seek to minimize the objective function by finding the value of x that leads to the predicted value Ax to be as close as possible to b. Taking the gradient of the objective function, gives the direction of steepest ascent. If we then move in the opposite direction we can efficiently decrease the value of the objective function by adjusting x until we reach a minimum difference. The gradient of the objective function can then be defined as:

$$\nabla J(x) = A^T(Ax - b), \tag{21}$$

where A^T is the transpose of A. In the gradient descent algorithm we iteratively update the estimate of x using the gradient, which can be defined as:

$$x_{k+1} = x_k - \alpha \nabla J(x_k) \tag{22}$$

where x_k is the estimate of x at iteration k, and α is the learning rate, which controls the step size. The inverse problem can thus be solved by making a initial guess x_0, computing the gradient, updating our estimate using gradient descent and checking for convergence.

4 Review of the literature: Optical diffraction tomography

Before diving into the methodology of this paper, I want to preface why this method is relevant and how it can improve the field of microscopy.

4.1 Conventional methods and their drawbacks

Conventional optical imaging methods, such as confocal and bright-field microscopy has been widely used to visualize biological samples. However, these conventional methods suffer from limitations such out-of-focus blur, limited depth penetration and most importantly the inability to capture phase information without the use of staining. Further more, these techniques cannot provide 3D information, which is important for understanding the spatial structure of samples.

4.2 Angled Illumination and 3D reconstruction

To overcome the limitations of conventional microscopy, the development of tomographic reconstruction techniques has been groundbreaking. One of which, is Optical diffraction tomography (ODT), which entails illuminating the sample from multiple angles and collecting the resulting diffraction patterns. Through this process it is possible to reconstruct the 3D refractive index distribution of the sample and recover the phase information. The diffraction patterns are typically captured using a high-numerical-aperture objective lens and a coherent light source, such as a laser.

4.3 The proposed method: High-resolution 3D refractive index microscopy

The method I have implemented for this paper is a realization of the optical diffraction tomography technique, particularly the Multi-Slice Beam Propagation (MSBP) method. This approach enables the reconstruction of high-resolution 3D refractive index distributions of biological samples by simulating the propagation of light through the sample approximated as series of thin slices. The MSBP method relies on Fourier optics and employs the angular spectrum technique to represent the light's journey through each section.

In previous chapters, we have explored the fundamental components of this method. We have discussed light propagation and interaction through transmittance and diffraction. We discussed the optical equivalence of using Fourier transforms to model optical objects. We talked about nonlinear systems and the inverse model. As well as the process of simulating experiments. In the proceeding section we will apply these concepts to explain how the proposed algorithm by Chowdhury et al. [1] works. Detailing the specifications of the model, the approximations that need to be made. In section 6.5 I will show the results of my own implementation of this algorithm done in Python as a preparation for further development for my Masters thesis.

5 Methodology

5.1 Modeling Light - Multi-Slice Beam-Propagation Forward Model

In the Multi-slice beam propagation algorithm, we consider our sample as a series of thinly sliced layers. By discretizing the sample in this way we can approximate the light's changing phase and amplitude as it propagates from layer to layer throughout our sample. The incident field transmitted through one layer, produces a diffracted field which becomes the incident field of the next layer. Each layer of the sample thus acts as a complex transmission function determined by its refractive index (RI) distribution. This cascade ultimately leads to the diffraction pattern we can measure some distance away from the sample structure. These light intensity (diffraction) patterns can then be utilized in the reconstruction of the 3D refractive index distribution of our sample. Fig.6 details the multi-slice beam propagation process.

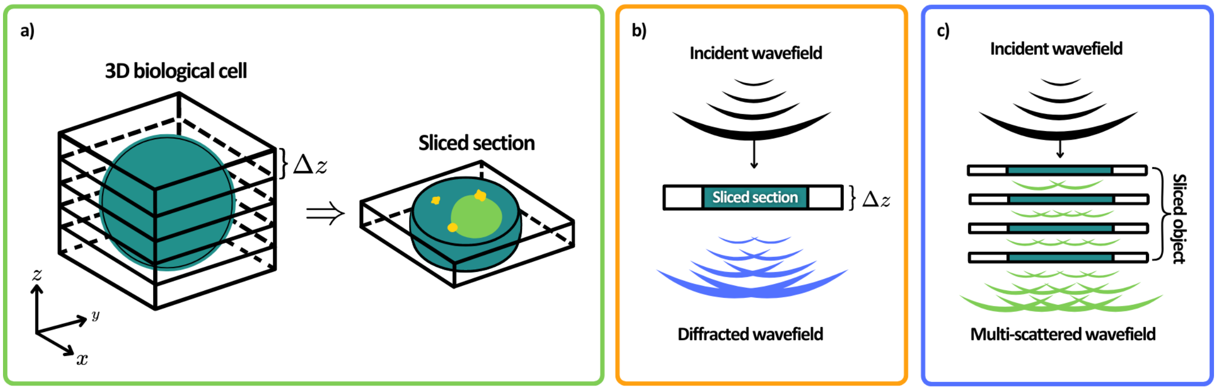


Figure 6: a) A section of the discretized 3D biological cell. b) Diffraction of an incident wave-field passing through a single slice. c) Multi-scattering of the incident wave-field passing from layer to layer.

Here we are making a few assumptions. First, to simplify the complex 3D structure of our object we are approximating it as a series of thin layers. The electric field at each layer is then calculated recursively as a approximation of the forward propagating light passes through the object. The surrounding media is assumed to be homogeneous (constant RI n_m) to simplifying our calculations. Further more, refractive index is assumed to be slowly varying compared to the distance between each layer, so that our sampling of the object is sufficient.

The amplitude of the electric field $y_k(\mathbf{r})$ at the *k*th layer is defined as:

$$y_k(r) = t_k(\mathbf{r}) * P_{\Delta z}\{y_{k-1}(\mathbf{r})\}, \tag{23}$$

where the objects *k*th-layer complex transmittance is defined by $t_k(\mathbf{r}) = \exp(j(\frac{2\pi}{\lambda})\Delta z(n_k(r) - n_m))$ for which n_k is the objects *k*th-layer complex-valued refractive index and n_m is the homogeneous refractive index of the surrounding media.

Further more the mathematical operator $P_{\Delta z}$, defined as:

$$P_{\Delta z}\{\cdot\} = F^{-1}\left\{\exp\left(-j\Delta z\left(\left(\frac{2\pi}{\lambda}\right)^2 - |\mathbf{k}|^2\right)^{1/2}\right) F\{\cdot\}\right\} \tag{24}$$

propagates the e-field by a distance Δz for each N equally-spaced layer in accordance with the angular spectrum propagation method, for which we utilize the 2D Fourier and inverse Fourier transforms $F\{\cdot\}$ and $F^{-1}\{\cdot\}$ respectively, as mentioned in section 2.3.

To initialize Eq. (23) we apply a incident planar electric field onto the sample from a particular angle. The incident e-field can be written as $y_0(\mathbf{r}) = \exp(j\mathbf{k}_0 \cdot \mathbf{r})$ for which $\mathbf{k}_0$ is the 2D illumination wave-vector applied to the first layer. The light is thus propagated through the object. At the last slice of the object ($k = N$), the e-field defined as $y_N(\mathbf{r})$ will contain the accumulated diffraction pattern caused by the multi-scattered wave field.

The electric-field is then back-propagated to a sensor plane at the objects center. The Forward model predicted electric field distribution at the sensor plane is defined as:

$$E(\mathbf{r}) = F^{-1}\{p(\mathbf{k}) * F\{P_{-z}\{y_N(\mathbf{r})\}\}\}, \tag{25}$$

where p($\mathbf{k}$) is the systems pupil function. The pupil function is a mathematical description of how the light propagates through an optical system. For a perfect microscope the pupil function is approximated as a circle with a radius equal to the cutoff frequency describing the microscopes resolution. z refers to the distance from the last object layer $y_N(\mathbf{r})$ to the layer/plane within the object's volume at conjugate focus to the imaging plane of the system. Thus, if the imaging system is focused at the center of the object we will have $z = \Delta z N/2$.

The forward model in its entirety can be denoted as the operator $G\{n\}$, which transforms a 3D complex-valued input into a 2D complex-valued output $\mathbb{C}^3 \to \mathbb{C}^2$. The operator $G\{\cdot\}$ thus predicts the 2D e-field distribution measured at the sensor plane after illuminating $n(\mathbf{r})$ with the incident electric-field $y_0^l(\mathbf{r})$, described by Eqs. (23) and (25). The forward model operator $G\{n\}$ thus encapsulates propagation of all layers to the final electric field visible in a microscope.

The intensity of the measurements is defined as:

$$I(\mathbf{r}) = |E(\mathbf{r})|^2 \tag{26}$$

5.2 Reconstructing RI distribution - Inverse model

The inverse model of this method involves a dynamic interplay between the forward model, sample diffraction patterns and the reconstructed object RI distribution. Initially, the forward model generates diffraction pattern by applying L incident e-fields defined as $y_0^l(r) = \exp(j\mathbf{k}_0^l \cdot r)$ with their corresponding wave vector $\mathbf{k}_0^l$ for $l = 1, 2, ..., L$ for multiple illumination angles. Each angle's intensity measurement is denoted as $I^l(\mathbf{r})$. The inverse model then iteratively refines its estimate of the sample's 3D RI distribution by minimizing the least squares difference between the measured amplitude $\sqrt{I^l(r)}$ and the forward model's predictions, aiming to align the reconstructed sample's diffraction patterns with those of the original sample [7].

We thus have the estimated RI distribution of the 3D sample as:

$$n^*(\mathbf{r}_{3D}) = \arg\min_{n(\mathbf{r}_{3D})} \sum_{l=1}^{L} \sum_{r} |\sqrt{I^l(\mathbf{r})} - |G^l\{n(\mathbf{r}_{3D})\}||^2, \tag{27}$$

where $\mathbf{r}_{3D} = \langle \mathbf{r}, k \rangle$ is the 3D spatial position vector, such that $n(\mathbf{r}_{3D}) = n_k(\mathbf{r})$ for layers $k = 1, 2, 3, ..., N$. And $G^l\{\mathbf{r}\}$ is the forward model mentioned above. Finally, we end up with a complex-valued quantity $n^*(\mathbf{r}_{3D})$ which has both real and imaginary components describing the object's 3D refractive index distribution detailing both phase and absorption distribution.

6 Implementation & Results

The computational simulation algorithm I am replicating using Python, was originally written in MATLAB. I validated the consistency of my implementation using the provided MATLAB dataset and code outputs. The program implements the Multi-Slice Beam Propagation (MSBP) method to simulate and reconstruct the 3D refractive index of a phantom object. It initializes a 3D RI distribution from a stacked .TIF file and sets physical parameters such as refractive indices, pixel size, wavelength, and numerical aperture. The code defines a spiral scan pattern for illumination angles, calculates propagation kernels and pupil support, simulating the optical scattering process for each angle, ultimately producing amplitude measurements. The reconstruction involves the refinement of the estimated 3D RI distribution using gradient-based optimization. The forward model then is re-run on the reconstructed object to compare its measurements to the original measurements as a validation check.

In this section, I will break down the implementation components, present the outcomes of my implementation, and compare them with those from the MATLAB code.

6.1 Phantom Object

Initially, the algorithm sets up a phantom object, with a simplified 3D refractive index distribution. A "phantom" is used as a basic substitute for complex real-world samples, facilitating algorithm development by providing a clear refractive index pattern. This allows us to easily observe and correlate the relationship between the sample's features and the resulting diffraction pattern, aiding the validation of the simulation's accuracy compared to the expected outcomes.

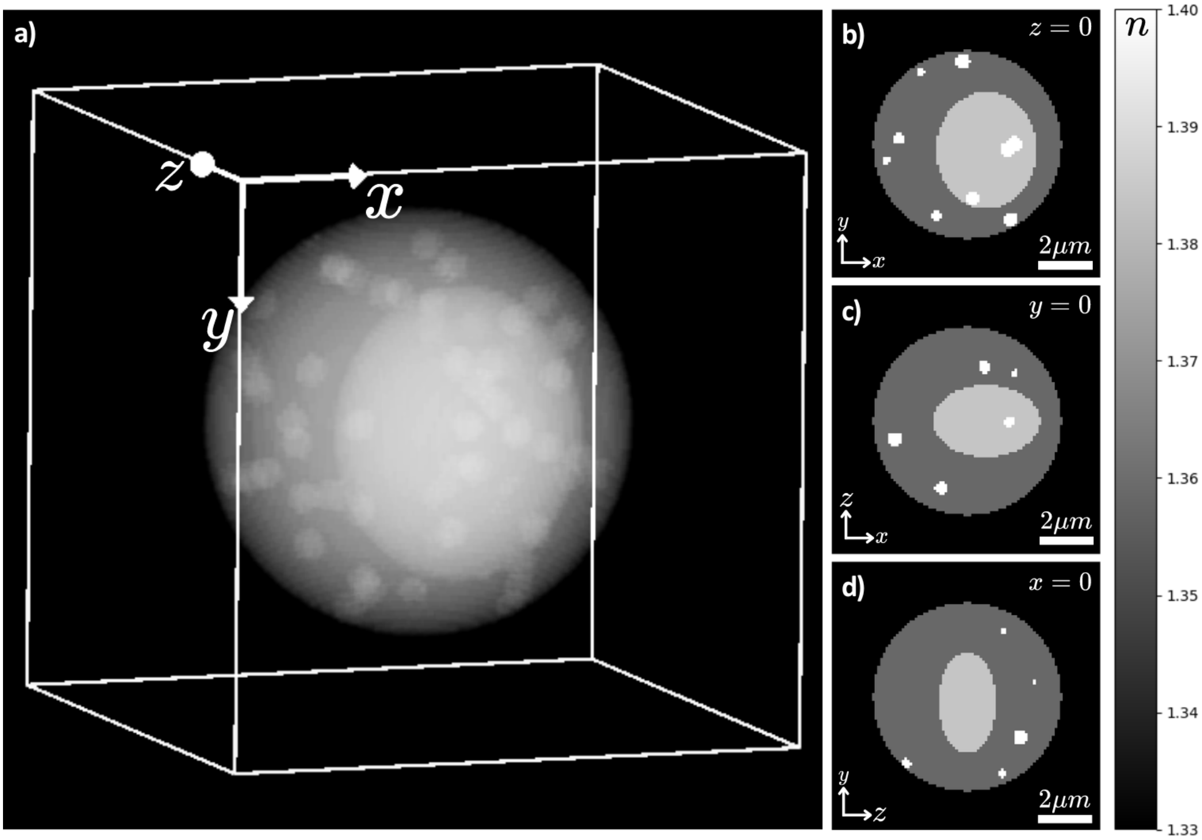


Figure 7: a) 3D representation of the phantom object's refractive index distribution, while (b,c,d) are cross sections of the samples center.

The phantom object, detailed in Fig.7, includes a central body, a nucleus, and interspersed vesicles with refractive indices of 1.4, 1.38, 1.45, and 1.52, all enclosed in a medium with a refractive index of 1.33, replicating water. The significant variations in refractive index within this cell make it a strongly scattering sample, which is essential because the degree of scattering affects the amount of information we can extract from the diffraction pattern.

The phantom object measure roughly 71 pixels in diameter, with each pixel representing 0.1 microns, resulting in an approximate diameter of 7.1 microns for the simulated sample. This size is reasonable, considering that typical cells used in light microscopy, range from 0.1 to 5.0 microns for prokaryotic cells and 10 to 100 microns for eukaryotic cells [8].

The phantom object RI distribution gets subtracted from the immersive media so that our simulated scattering effect is focused on the internal features of the object itself. This also allows us to simplify the calculations of the propagation model, making the computations more efficient. The reconstruction we will be discussing in the proceeding sections aims to calculate the delta RI (Δn), which is the difference between the RI of a specific section of the sample, like the nucleus, and the RI of the surrounding medium. This ensures that the reconstructed RI directly represent the structure of our sample. From now on, I'll be refering to this phantom object as the "true phantom object".

6.2 Propagation kernel and pupil support

The **propagation kernel** in optical simulations is a mathematical function that models how the phase and amplitude of an electromagnetic wave change as it propagates through a medium. It is used to calculate the phase advance of the wave per unit distance, which is crucial for simulating the behavior of light in complex optical systems. The propagation kernel (Fig.8a) is a complex-valued function, where the real part represents amplitude changes and the imaginary part represents phase shifts. The phase advance is calculated based on the refractive index of the medium, the wavelength of the light, and the spatial frequencies in the Fourier domain. This calculation allows the simulation to account for the effects of refraction that occur when light interacts with the medium.

In the context of the Multi-Slice Beam Propagation (MSBP) method, the propagation kernel is essential for forward modeling the optical scattering process. It helps determine how light waves will evolve as they pass through the simulated object, providing the basis for predicting the resulting intensity distributions at the detector plane. This information is then used for iterative reconstruction of the RI distribution within the object, contributing to the overall accuracy of the 3D refractive index imaging.

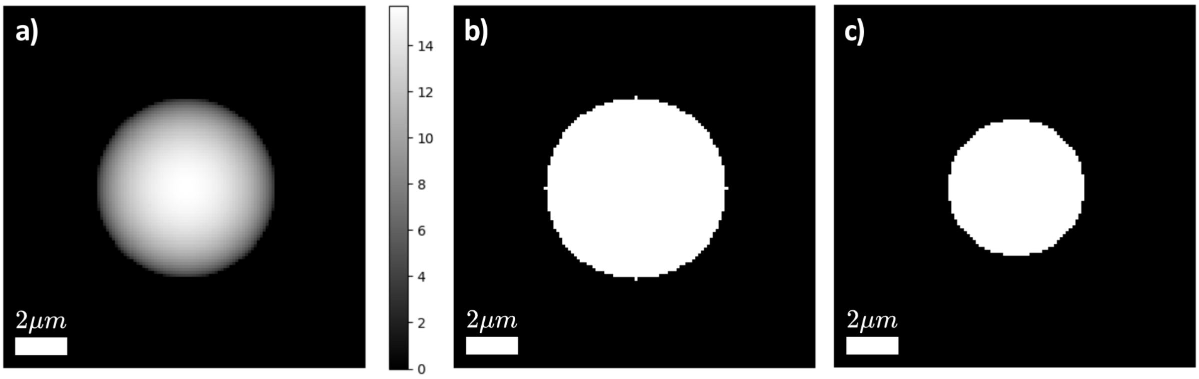


Figure 8: a) Propagation kernel, b) Propagation cutoff binary mask, c) Numerical aperture binary mask. The binary masks in b) and c) have values 1 for the accepted area and 0 for the cutoff area.

The **propagation cutoff** binary mask (Fig.8b) filters out spatial frequencies that cannot propagate through the medium. This can be defined as:

$$\nu_x^2 + \nu_y^2 \leq \left(\frac{n_{imm}}{\lambda}\right)^2, \tag{28}$$

where (ν_x, ν_y) are the spatial frequencies, λ is the wavelength and n_{imm} is the refractive index of the immersion medium. If the spatial frequencies are too high, corresponding to evanescent waves, the mask sets these frequencies to 0, indicating exclusion from the simulation. This ensures that only propagating waves are modeled.

The **numerical aperture** binary mask (Fig.8c) filters out spatial frequencies beyond the resolving capability of the imaging system, defined by the numerical aperture (NA) of the system the simulation represents. This can be defined as:

$$\nu_x^2 + \nu_y^2 \leq \left(\frac{NA}{\lambda}\right)^2, \tag{29}$$

where (ν_x, ν_y) are spatial frequencies and λ is the wavelength.

These masks and kernels are crucial for the forward model of the optical system, as they determine which spatial frequencies are allowed to propagate and how they change as they do so. This ensures that the simulation accurately represents the physical constraints of a real optical setup.

6.3 Illumination angles

Another factor of the implemented simulation is the angles at which the sample is illuminated. The algorithm initializes a spiral scan pattern of about 100 illumination angles, as seen in Fig.9.

This number of illumination angles is a trade-off between the level of detail desired in the simulation and the computational resources available. A larger number of measurements taken at different angles can lead to a more accurate representation of the light-object interaction but also requires more computational power. This is an important issue in solving inverse problems. If a problem is too difficult to solve with few measurements it is called under-determined, and in this case the angular information is considered to be under-sampled. Thus a multiple of varied measurements are need to accurately solve the inverse problem.

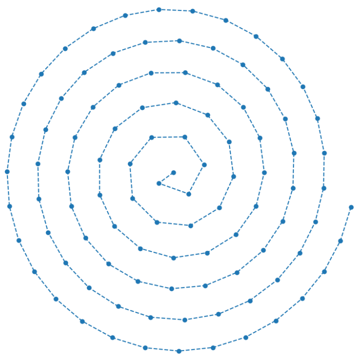


Figure 9: Spiral scan pattern

Ideally we can find a reasonable trade-off between the necessary number of measurements and the resolution we want to achieve. Once the spiral coordinates are prepared, they are scaled by the ratio of the numerical aperture NA to the wavelength, to ensure that the illumination angles are within the range that the imaging system can physically accommodate. The numerical aperture is a measure of the system's ability to gather light and resolve fine detail, and scaling the coordinates accordingly ensures that the angles are physically realistic. Finally we transform the components of the spiral coordinates to the Fourier domain allowing us to apply the propagation of light through the object from different angles.

6.4 Implemented Forward Model: The multi-slice beam propagation (MSBP)

Once the spiral pattern has been determined, we implement our forward model. The forward model involves the optical scattering process as the simulated light propagates through the phantom object for each illumination angle. This is done according to the multi-slice framework mentioned in section 5.1. Fig.10 details the forward model procedure.

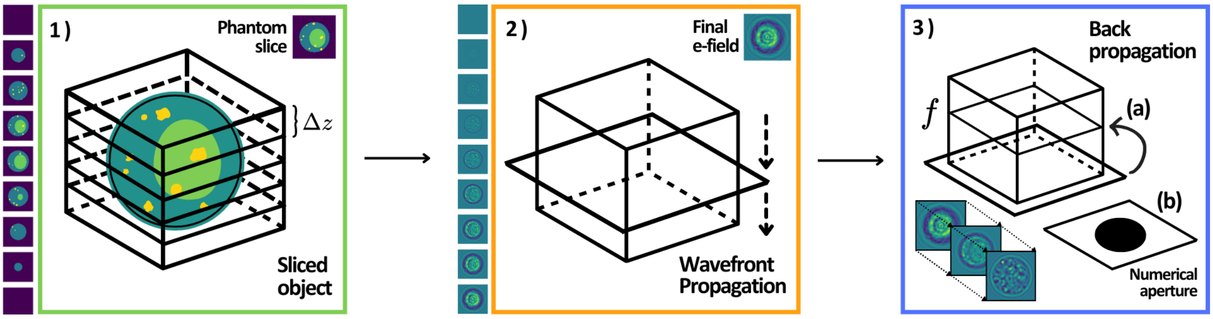


Figure 10: MSBP Forward Model Procedure:
1) Slicing, 2) Wave propagation, 3) Back-propagation to focal plane, applying numerical aperture

Single Axis Forward Propagation

The light wave field propagates through the object layer by layer. As it passes through each layer, its phase and amplitude change due to variations in the refractive index from one layer to the other. This process is illustrated in the forward model applied to a single angle in Fig.11.

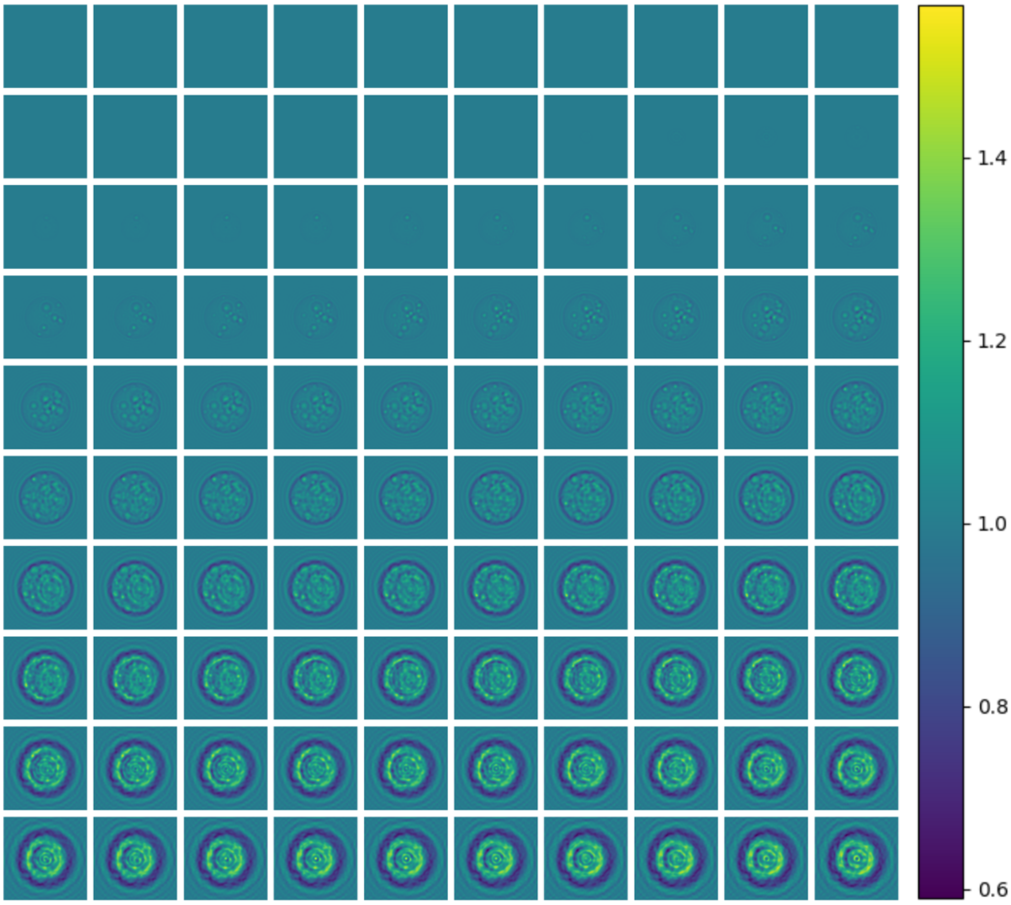


Figure 11: Propagated electric field through the phantom object.

Initially, when there is no material present in the first layers of the object. The incident light has a uniform intensity across its wavefront, meaning its phase and amplitude are constant. As this uniform light travels through the first layer of the sample, its phase and amplitude are altered by the layer's refractive index distribution and thickness. The modified wave field then moves on to the next layer, where it undergoes further changes. Consequently, the wave field continuously evolves as it interacts with each subsequent layer. Essentially, the wave field accumulates information about the sample RI distribution one layer at a time. The incident wavefront can be imagined as an empty blueprint, to which more information is added after propagating through each layer of the sample. By the end, we have a complex wave field encoded with the sample's structural information along a single axis.

Back-propagation

Once the wave field has been propagated through all the layers, the complex wave field gets back-propagated to the center of the sample. The algorithm essentially reverses the direction of the light propagation, simulating the accumulated object information of all the layers to a intermediate layer, defined by the focal point of the system. The back-propagation to the sample's center is a aspect of the simulation which result is equivalent to the focal point of a microscope. Through back-propagation, essentially we also include the contribution of the layers after the middle plane in the diffraction pattern.

Multiple illumination angle wave field propagation

To reconstruct the 3D structure of the sample's refractive index, we need information from multiple different angles. Applying the spiral pattern from section 6.3, we illuminate the sample from different angles by rotating the light source above the sample. Each illumination angle provides a 2D projection of the sample's RI distribution along that specific direction. For each angle, we also apply back-propagation. Illuminating the sample at 100 different illumination angles yields the wave fields in Fig.12. These measurements serve as a foundation for the reconstruction process, acting as the measurements.

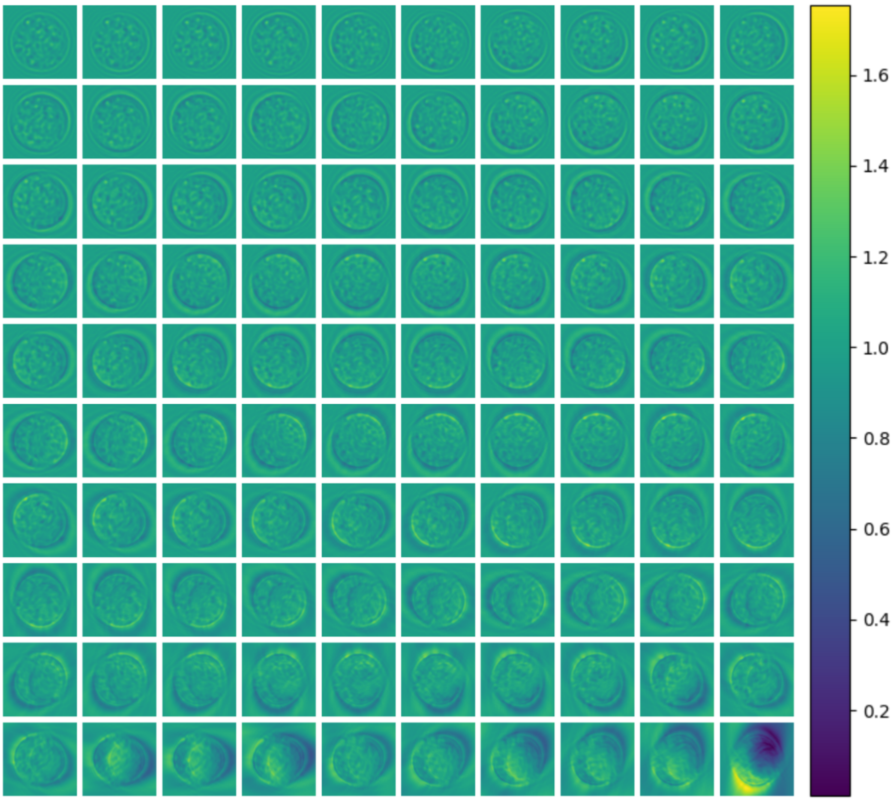


Figure 12: Wave-fields obtained with different illumination angles at the center of the sample

6.5 Implemented Inverse model: 3D refractive index reconstruction

The inverse model, as discussed in section 3.4, provides an estimate of the sample's RI distribution. We apply the multi-angled forward model onto this estimate, acquiring a set of 2D projections. We then compare these projections to the projections we measured from the true phantom object in Fig.12, evaluating the accuracy of the inverse model's estimated RI distribution. Through gradient descent, the algorithm iteratively adjusts its estimate to produce an RI distribution that can produce projections as similar to the true projections as possible. After enough iterations, we can acquire an accurate reconstruction of the phantom object's 3D refractive index.

The Results

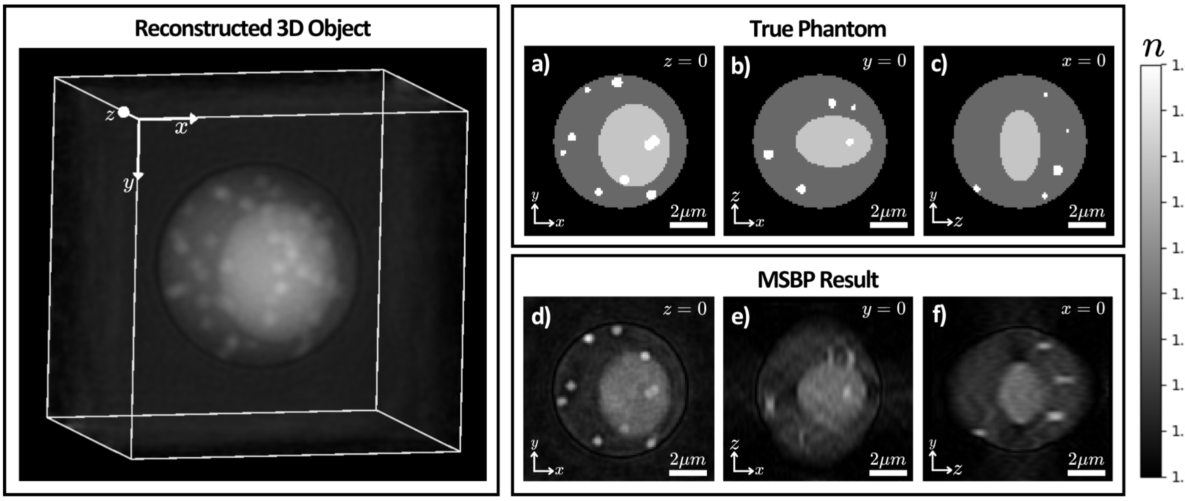


Figure 13: Reconstructed results of the 3D phantom object. Cross-section of the sample from the (a,d) front, (b,e) top, and (c,f) side.

The reconstruction of the phantom object was successful, demonstrating 3D refractive index imaging. Fig.13 demonstrates the reconstruction results of the multi-slice beam-propagation (MSBP) method. For which Fig.13(a,b,c) show the true phantoms cross-sectional planes (x,y), (x,z) and (z,y), within the true phantom's volume. While Fig.13(d,e,f) shows the corresponding reconstructed cross-sectional planes. The true and reconstructed phantom objects shows good matching of their corresponding planes. Running the algorithm took approximately 146 minutes, and I was able to acquire a minimized cost of 1.48.

Noise & Inaccuracies

The visible **noise** in our reconstructed 3D object can arise from several different sources in our iterative algorithm. The numerical methods and approaches used in this algorithm may have inherent limitations in numerical precision which can iteratively introduce tiny errors that gets amplified over each iteration. Such numerical errors can for example arise from **round-off errors**, for which mathematical operations perform miscalculation because of inaccuracy in representing real numbers.

The reconstructed object's **shape** is most accurate through the (x,y) plane. This is due to the fact that the angled illumination occurs primarily through this plane. Subsequently, the side and top view (x,z) and (z,y) demonstrate what seems to be "stretching" or "skewing" of the sample's body. This phenomenon occurs due to incomplete sampling along these planes, as a result of limited or incomplete coverage of the object from certain perspectives. Thus the algorithm can have some trouble in accurately capturing the shape and features of the object along these directions. This is an important aspect of imaging to consider. To reconstruct the object in its entirety, with minimal difference, will ultimately require an abundance of information about the sample. In this case, we would need to illuminate the sample from multiple different directions, at multiple different angles. This is a trade-off between amount of measurements and accuracy, which is commonplace in microscopic imaging. The stretching along the z-axis, which is fairly visible in Fig.13(e,f), illustrates the limited z-axis resolution of our reconstruction algorithm.

There is a fine line between how accurate our model needs to be and how many measurements we need to perform. Ideally we want a accurate reconstruction with as little data as possible, maximizing the efficiency of imaging. In terms of spatial inaccuracies we can explore strategies to improve angular coverage or develop algorithms that are more robust to limited-angle data. In addition we can apply post processing techniques to help mitigate these distortions such as artifact correction methods or regularization.

Cost function

The **cost function** in Fig.14 details the convergence of the model. The model seems to converge at around 300 iterations, reducing the cost from 24.53 to 1.48.

The amount of iterations one chooses to perform, depends on a few factors. It depends on how quickly the cost function of the model converges and how efficient one needs the algorithm to be. In a lab, were someone works with multiple samples throughout the day, it would be ideal to be able to image a sample efficiently within minutes. For the sake of this paper and producing accurate results, I decided to run the algorithm for 300 iterations. The convergence rate can be improved by optimizing the learning rate of the model.

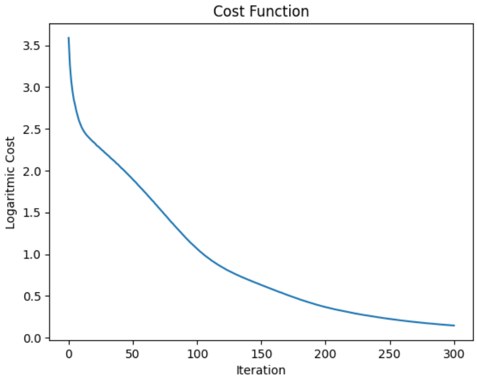


Figure 14: Cost function convergence through 300 iterations.

Validating the reconstructed phantom object

By applying the multi-slice beam propagation (MSBP) forward model on the reconstructed phantom object we can validate its accuracy. Fig.15a shows the wave fields of different illumination angles propagated through the reconstructed phantom object. This result is very similar to what we see in Fig.12. A more detailed comparison can be seen in Fig.15b,c, for which the measurements look identical.

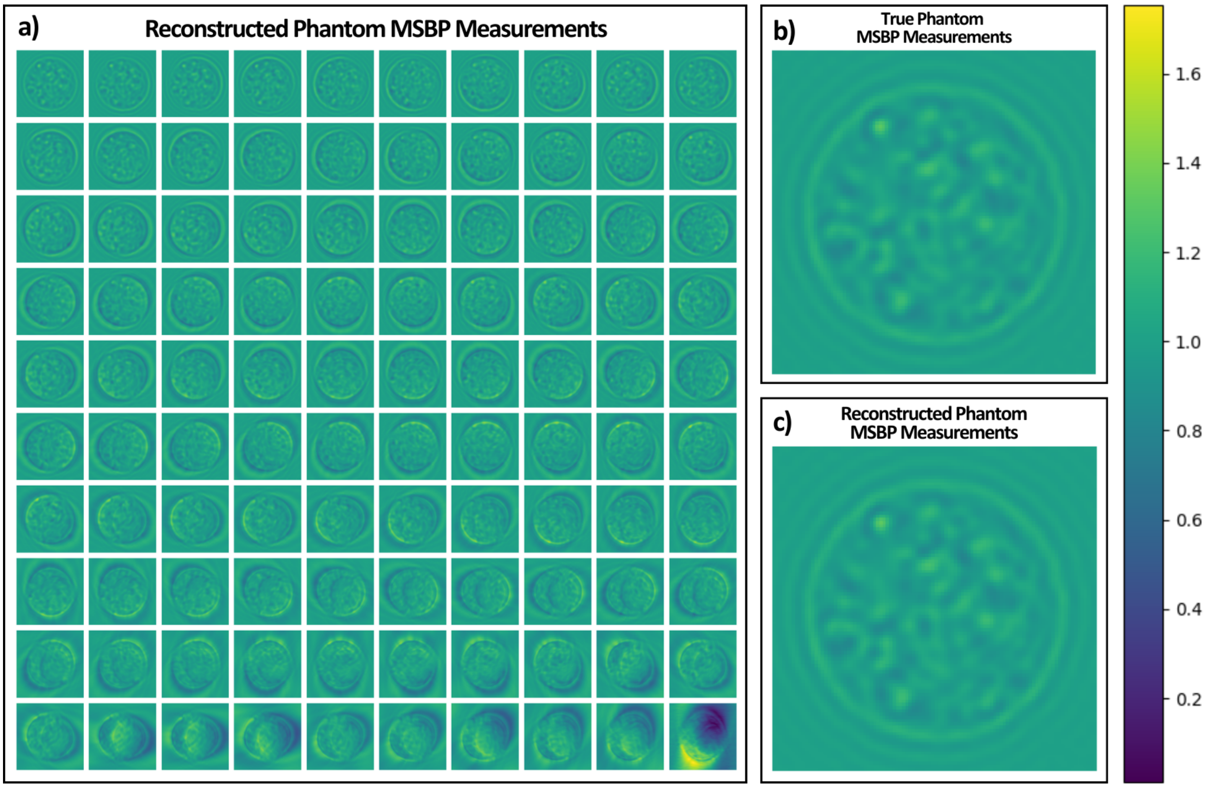


Figure 15: a) Multi-angle forward model applied to the reconstructed phantom object. (b,c) Comparison between true phantom measurement and reconstructed phantom measurement from the first illumination angle.

6.6 Validating The Python Implementation

While replicating the work of ref.[1] I compared the data matrices between my results and the results of the MATLAB code. Doing this I made sure that my data was as similar to the MATLAB data as possible throughout the whole algorithm. Ultimately there are some differences between the performance of the MATLAB code and my code, but overall my replication yielded a accurate reconstruction of the sample's refractive index distribution. Comparison between data matrices can be done by computing the maximum difference between the data values. It is however worth mentioning that while the result is similar there will be discrepancies between the MATLAB and Python data values. For example, the formatted .MAT files, imported into python, will have small errors simply stemming from the way the data is formatted. There is also a noticeable difference in the way MATLAB computes the Fourier transforms compared to Numpy in Python. Although these differences are apparent, I was still able to produce results in my Python code that closely resembles the results of the MATLAB implementation.

The MATLAB code took 96 minutes to run, while the python code took 146 minutes, which is mainly due to the fact that MATLAB have inbuilt operations while python utilizes extended libraries. In 300 iterations my implementation was able to acquire a reduced error of 1.48, while the MATLAB code had a reduced error of 1.35. Fig.16 shows the multi-angle measurements from the MATLAB code and the difference between the MATLAB and Python measurements.

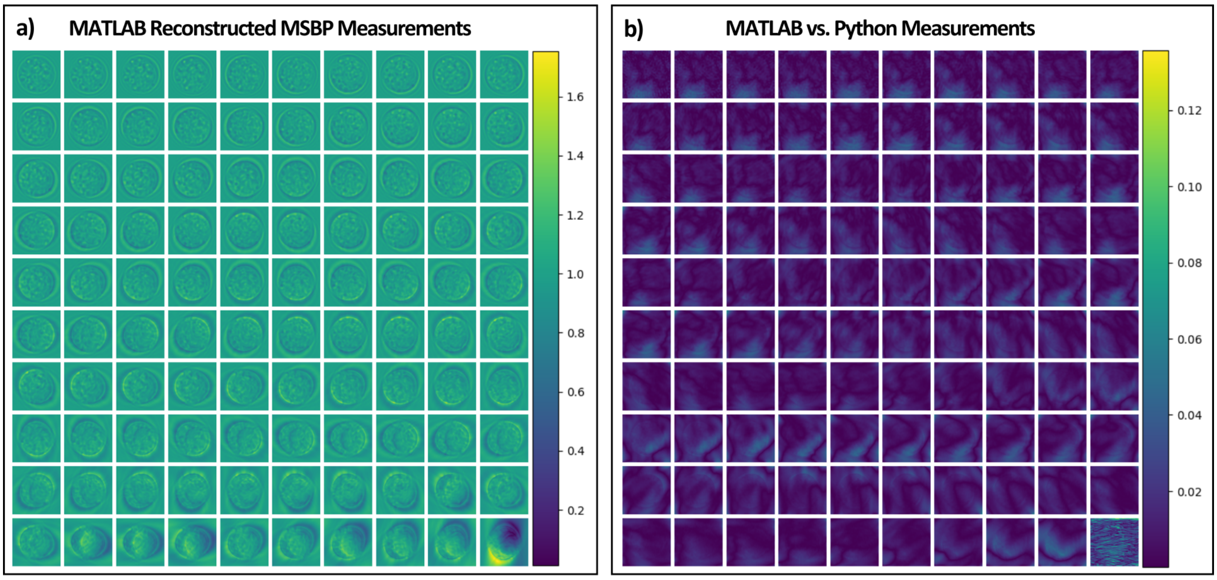


Figure 16: a) MATLAB reconstructed MSBP measurements. b) The difference between the MATLAB and Python data arrays.

7 Conclusion

In conclusion, this paper has detailed the replication of a computational model for reconstructing the refractive index distribution of biological samples by applying methods of optical diffraction tomography. The implementation of the model in Python was validated by comparing its results with those obtained using the original MATLAB implementation.

The Python implementation was able to accurately reconstruct the refractive index distribution of a phantom object, demonstrating the feasibility of the approach. However, there are some limitations to the current model, such as the need for a large number of illumination angles to achieve accurate results and the presence of noise in the reconstructed images.

Future work could focus on improving the efficiency of the model by reducing the number of required illumination angles or developing more robust algorithms that can handle limited-angle data. Additionally, post-processing techniques such as artifact correction or regularization could be applied to mitigate the effects of noise and improve the quality of the reconstructed images.

Overall, this computational model represents a significant step forward in the field microscopy and through further development and experimentation, it has the potential to greatly improve our ability to study the structure and function of biological samples at the microscopic level.

References

[1] Chen M. Eckert R. Ren D. Wu F. Repina N. Waller L. (2019). High-resolution 3D refractive index microscopy of multiple-scattering samples from intensity images. Optica 6(9) 1211-1219. Chowdhury, S.

[2] Teich M. C. (2019). Fundamentals of photonics. john Wiley sons. Saleh, B. E.

[3] J. W. (2005). Introduction to Fourier Optics. Roberts Goodman and Company Publishers.

[4] Y. R. (2003). The Principles of Nonlinear Optics. Wiley-Interscience. Shen.

[5] A. C. (1997). Mathematical models in the applied sciences (Vol. 17). Cambridge University Press. Fowler.

[6] A. (2005). Inverse problem theory Tarantola, methods for model parameter estimation. Society for industrial, and applied mathematics.

[7] J. Zhong L. Tian M. Chen G. Tang M. Soltanolkotabi L.-H. Yeh, J. Dong and " Opt. Express 23 33214–33240 (2015). L. Waller, "Experimental robustness of Fourier ptychography phase retrieval algorithms.

[8] 2024 LibreTexts. (n.d.). Studying cells cell size. In General Biology (Boundless). Retrieved May 24.