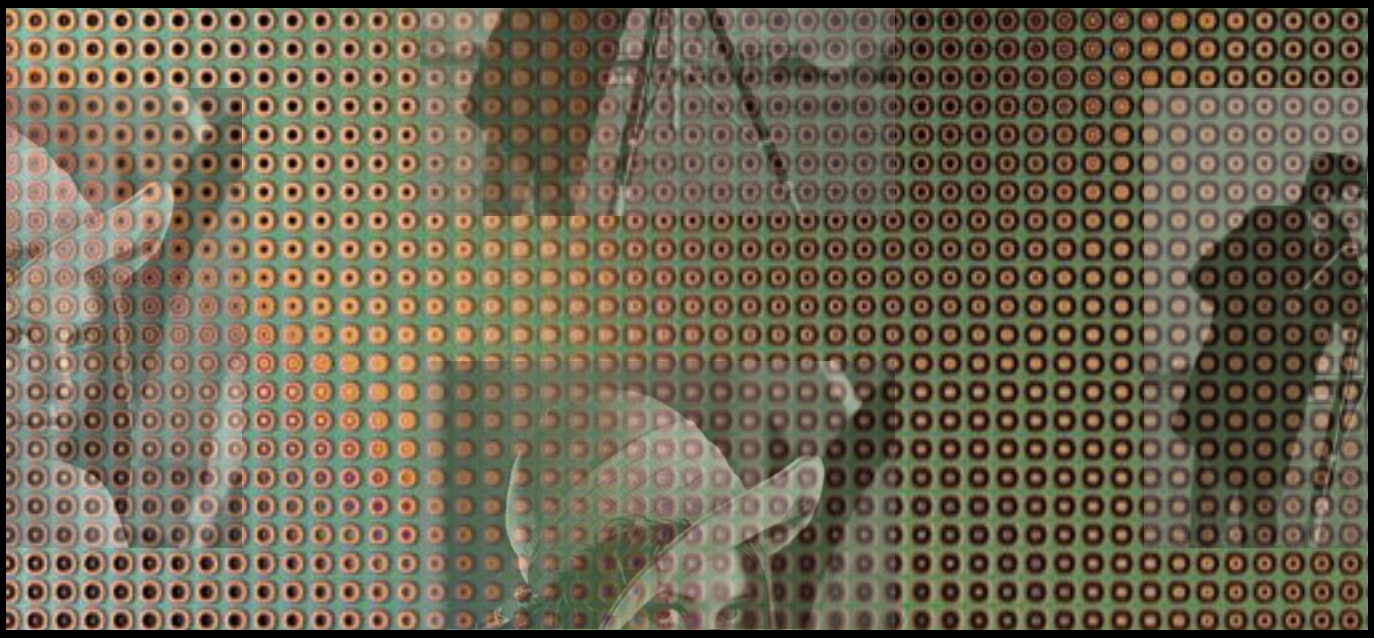
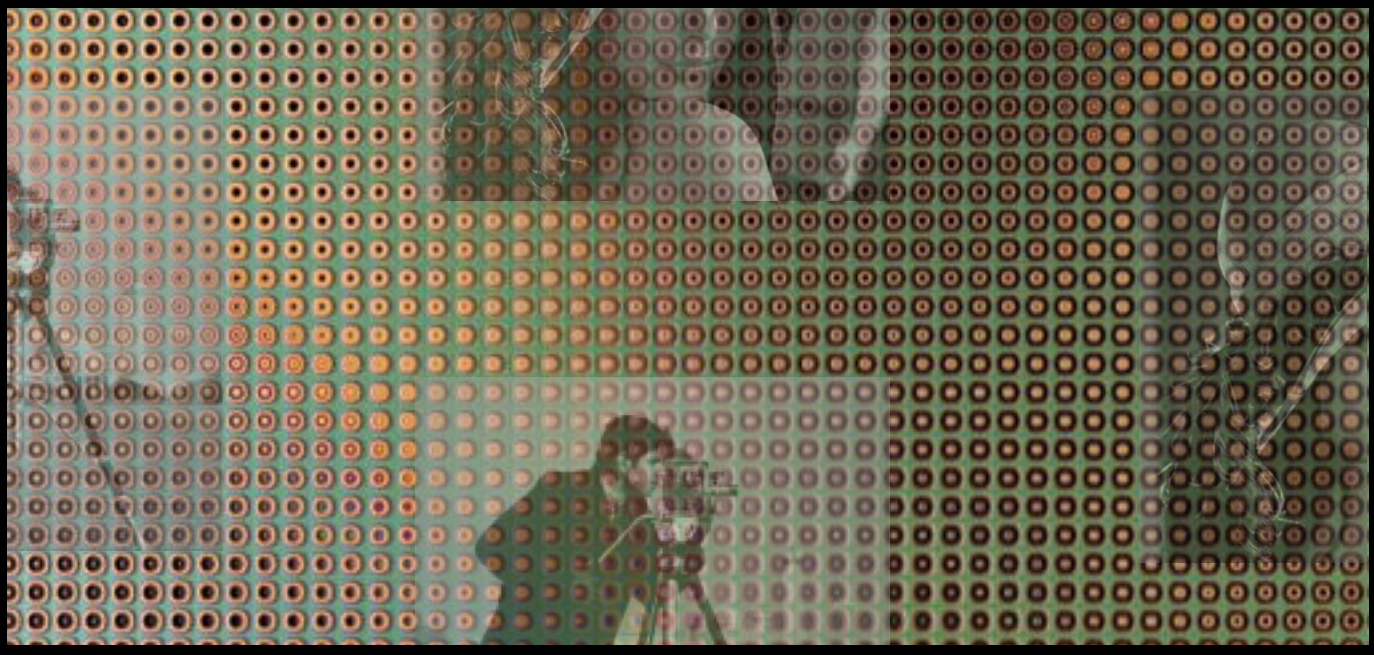


Project Report

Iterative phase retrieval using the HIO Algorithm

By Ansal Kanu, Hansraj Saxena and Yashraj Sood

Team Number 3



Iterative Phase Retrieval Using the Hybrid Input-Output (HIO) Algorithm

Ansal Kanu[1], Hansraj Saxena[2], and Yash Raj Sood[3]

[1]B.Sc.(H) Physics, Second Year, Ramjas College, University of Delhi, Delhi, India
[2]B.Sc.(H) Physics, Second Year, Hindu College, University of Delhi, Delhi, India
[3]B.Sc.(H) Physics, Graduated (2024), Shivaji College, University of Delhi, Delhi, India

March 8, 2025

Contents

1 Introduction **4**

2 Background Theory **5**

3 The Fourier Transform **10**

4 The Evolution and Importance of Algorithmic Phase Retrieval **13**
4.1 Gerchberg-Saxton Algorithm and the Hybrid Input-Output Algorithm . . . 14
4.2 Iterative Phase Retrieval using Vortex Illumination . . . 15

5 Methodology **17**

6 Results **19**
6.1 Phase Object Generation . . . 19
6.2 Fourier Magnitude Extraction . . . 19
6.3 Phase Retrieval using HIO Algorithm (Plane Wave Illumination) . . . 19
6.4 Phase Retrieval using HIO Algorithm (Vortex Illumination) . . . 20
6.5 Comparison of Plane Wave and Vortex Illumination Reconstructions . . . 20
6.6 Discussion and Conclusion . . . 21

7 Applications of Iterative Phase Retrieval **22**
7.1 Coherent Diffractive Imaging (CDI) and the Phase Retrieval Problem . . . 22
7.2 Phase Retrieval in Astronomy . . . 23
7.2.1 Phase Retrieval and Optical Aberrations . . . 23
7.2.2 Interferometry and Phase Retrieval . . . 24
7.2.3 Phase Retrieval in Adaptive Optics . . . 24
7.2.4 Phase Retrieval for Space Telescopes . . . 25
7.2.5 Exoplanet Imaging and Coronagraphy . . . 30
7.2.6 Beyond Astronomy: Broader Applications of Phase Diversity . . . 31
7.2.7 Future Perspectives . . . 31
7.3 Modern Developments: Phase Retrieval Through Machine Learning . . . 31
7.3.1 Mathematical Formulation: Fourier Phase Retrieval . . . 32
7.3.2 Neural Networks for Phase Retrieval . . . 33
7.3.3 Learning in Neural Networks and Its Relevance to Phase Retrieval . . . 34
7.3.4 Deep Learning-Based Phase Retrieval . . . 34
7.3.5 Hybrid Classical-ML Approaches . . . 35
7.3.6 Future Challenges and Research Directions . . . 36

8 Limitations of Phase Retrieval **37**

Appendix: HIO Algorithm for Phase Retrieval **40**

1. Introduction

Since the late 1930s, the venerable branch of physics known as optics has gradually developed ever-closer ties with the communication and information sciences of electrical engineering. The trend is understandable, for both communication systems and imaging systems are designed to collect or convey information. In the former case, the information is generally of a temporal nature (e.g. a modulated voltage or current waveform), while in the latter case it is of a spatial nature (e.g. a light amplitude or intensity distribution over space), but from an abstract point of view, this difference is a rather superficial one — both fields share common mathematical frameworks and analytical techniques, making their intersection a natural evolution. One problem of particular significance that emerges at this intersection is the problem of phase retrieval.

Phase retrieval, which involves reconstructing a function given only the magnitude of its Fourier transform, is a fundamental challenge in various scientific and engineering fields, including electron microscopy, crystallography, astronomy, and optical imaging. This issue is particularly relevant in optics, where phase information is often lost due to the limitations of conventional detection systems. Optical measurement devices, such as charge-coupled device (CCD) cameras, photosensitive films, and even the human eye, are sensitive only to the intensity of light rather than its phase. This limitation arises because these detectors convert incoming photons into electrical signals but cannot directly track the rapid oscillations of an electromagnetic wave, which typically occur at frequencies of the order of 10^{15} Hz. As a result, they register only the photon flux, proportional to the squared magnitude of the optical field, without retaining any phase information. In order to reconstruct an object from its diffraction pattern, the inverse Fourier transform must be computed. This is possible only given the full complex-valued diffraction data, i.e. the magnitude and phase. However, in diffractive imaging, generally only magnitudes can be directly measured while the phase needs to be estimated.

Recovering the phase of an optical wave, therefore, requires additional techniques beyond direct detection. Traditionally, interferometric methods, such as holography, have been employed to extract phase information by combining an unknown wave with a reference wave of known characteristics. However, in cases where no reference wave is available or when working with non-interferometric imaging systems, computational approaches become necessary.

In this project, we explore iterative phase retrieval, a computational technique that reconstructs the phase from intensity-only measurements through an iterative refinement process. Specifically, we implement the Hybrid Input-Output (HIO) algorithm, a widely used method that has been instrumental in applications such as coherent diffractive imaging, adaptive optics, and astronomical imaging. By employing this approach, we aim to demonstrate how phase retrieval can recover lost phase information, enabling accurate reconstructions of optical fields and expanding the capabilities of imaging systems beyond conventional limitations, which is important not only to understand the nature of imaging systems but also because of its extensive usage in other fields of physics.

However, before jumping directly into the process of Hybrid Input Output(HIO) algorithm or iterative phase recovery, for that matter, we first would like to discuss some basic principles so that the readers of the report can understand the objective behind our work.

2. Background Theory

It is natural to begin from the very beginning in order to understand the motive behind this report. Therefore, we will first begin with a small revision of the basic theory of Fourier Series and Transforms.

Fourier Series: To understand what Fourier Optics is and how and why does it work, we first must go through the beautiful mathematical essence of the theory given by Jean-Baptiste Joseph Fourier. First, we'll begin with the famous Fourier series. The Fourier series is a mathematical tool used to represent periodic functions as an infinite sum of sines and cosines[**?**]. It is fundamental in fields like signal processing, physics, and engineering, as it helps us analyse wave-like phenomena.

Mathematically speaking, the most general form of the Fourier Series of a periodic function $f(x)$ with period T is written as:

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{2\pi n x}{T} + b_n \sin \frac{2\pi n x}{T} \right)$$

where the Fourier coefficients a_0, a_n, and b_n are defined as:

$$a_0 = \frac{1}{T} \int_0^T f(x)\, dx$$

$$a_n = \frac{2}{T} \int_0^T f(x) \cos \frac{2\pi n x}{T}\, dx, \quad n \geq 1$$

$$b_n = \frac{2}{T} \int_0^T f(x) \sin \frac{2\pi n x}{T}\, dx, \quad n \geq 1$$

The coefficients are analogous to those in a power series expansion, and the determination of their numerical values is the essential step in wiring a function as a Fourier series.

Dirichlet Conditions: A function $f(x)$ can be represented as a Fourier series if it satisfies the following Dirichlet conditions:

1. $f(x)$ is periodic with period T, i.e., $f(x + T) = f(x)$ for all x.
2. $f(x)$ is piecewise continuous, which means it has a finite number of discontinuities.
3. $f(x)$ has a finite number of extrema (maxima and minima) in any given period.
4. $f(x)$ is absolutely integrable over one period:

$$\int_0^T |f(x)|\, dx < \infty.$$

As shown in the figure on the next page, we have a random periodic function. Obviously, it can be observed that the function representing this figure is not of sinusoidal nature. However, the function representing this function can be deciphered using the principles of Fourier Series.

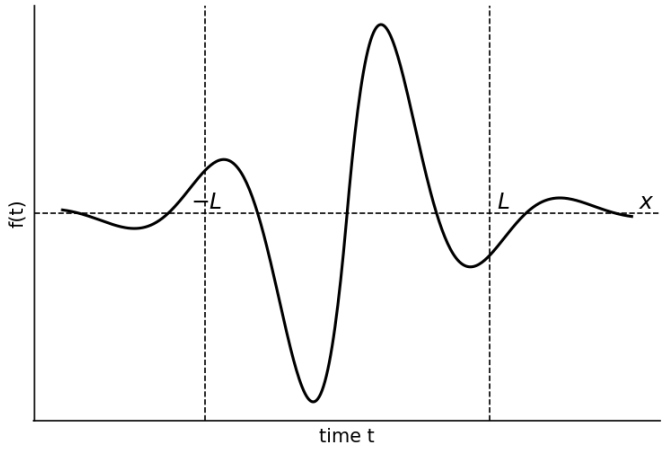


Figure 1: An example of a function that may be represented as Fourier series without modification.

In order to explain such process, we have also provided the Fourier series representation of some day-to-day functions in order to illustrate the importance of the Fourier series.

1. **Square Wave Waveform**

   A standard square wave with period T and amplitude A is defined as:

$$f(t) = \begin{cases} A, & 0 \leq t \leq \frac{T}{2} \\ -A, & \frac{T}{2} \leq t \leq T \end{cases}$$

   This is an odd function, meaning it contains only sine terms in its Fourier series.

   The Fourier Series of a Square Waveform is given as:

$$f(t) = \sum_{n=1,odd}^{\infty} \frac{4A}{n\pi} \sin(n\omega_o t)$$

   The Fourier coefficients decay as $\frac{1}{n}$, meaning the higher harmonics contribute significantly to the sharp transitions.Additionally, only the odd harmonics $(n = 1, 2, 3, ...odd)$ are present.

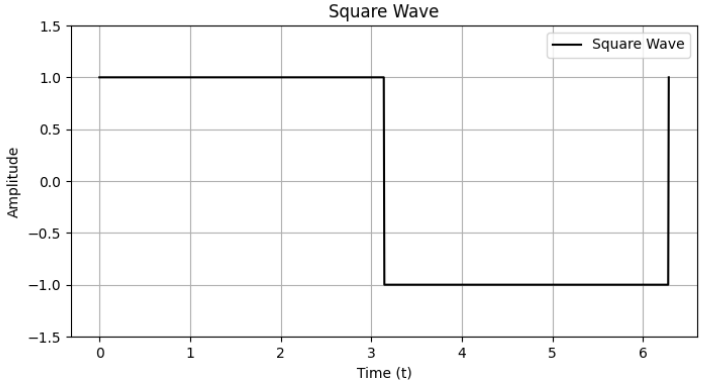


Figure 2: A simple square wave.

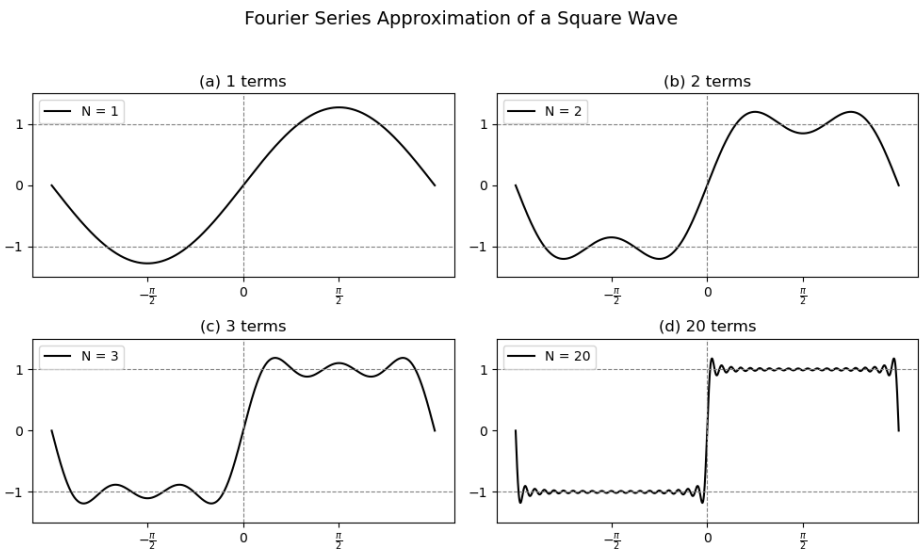


Figure 3: The convergence of a Fourier Series Expansion of a Square waveform, including a) one term, b) two terms, c) three terms and d) 20 terms. The overshoot δ is shown in d.

2. **Triangular Wave Waveform:** The Fourier series representation of a triangular wave can be derived using its odd symmetry and periodicity. A triangular wave is a continuous, piecewise linear function that oscillates between a maximum and a minimum value.

   A standard triangular wave of period T (or equivalently, fundamental frequency $f_0 = \frac{1}{T}$ is typically defined as:

$$f(t) = \begin{cases} t, & 0 \leq t \leq T \\ 2T - t, & T \leq t \leq 2T \end{cases}$$

   Since the function is odd, it contains only sine terms, meaning there are no cosine terms $(a_n = 0)$. The Fourier series of a periodic function, with amplitude A is:

$$f(t) = \sum_{n=1,odd}^{\infty} \frac{8A}{n^2\pi^2} \sin(n\omega_o t)$$

   The Fourier coefficients decay as $\frac{1}{n^2}$, which is faster than the Fourier series of a square wave $(\frac{1}{n})$. This leads to a smoother approximation. Naturally, in a triangular wave only the odd harmonics appear in the expansion.

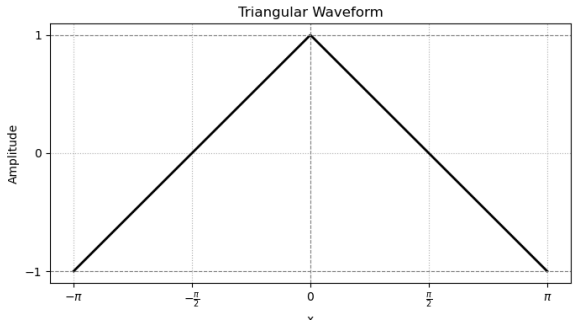


Figure 4: A simple triangular wave.

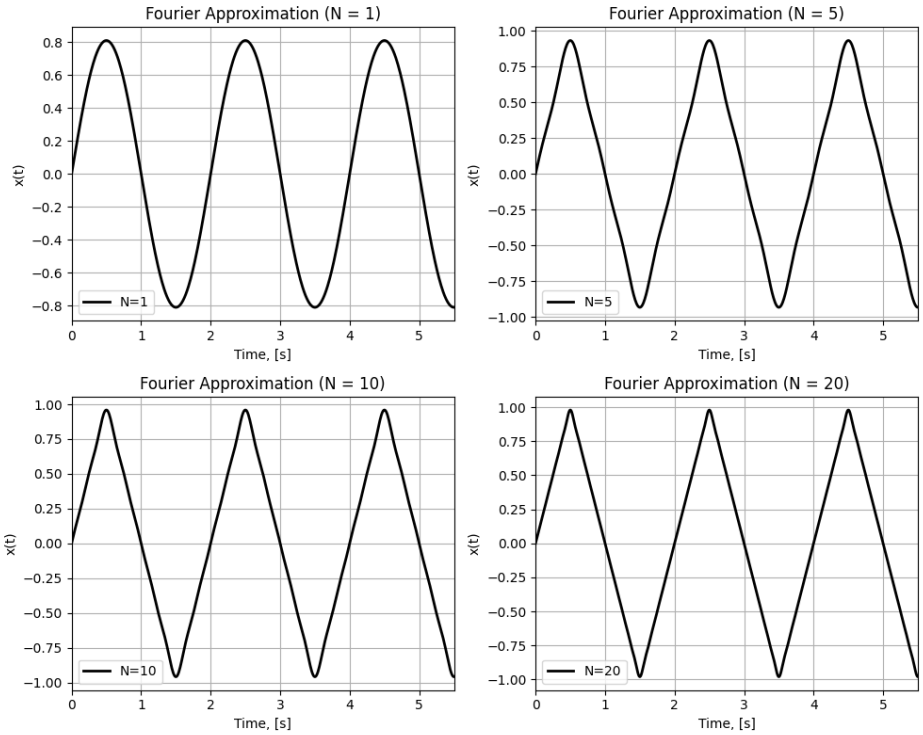


Figure 5: The convergence of a Fourier Series Expansion of a Triangular waveform, including a) one term, b) two terms, c) three terms and d) 20 terms.

3. **Sawtooth Wave Waveform**

The Fourier series of a sawtooth wave can be derived based on its periodicity and symmetry properties. A sawtooth wave is a periodic function that increases linearly over one period and then sharply drops.

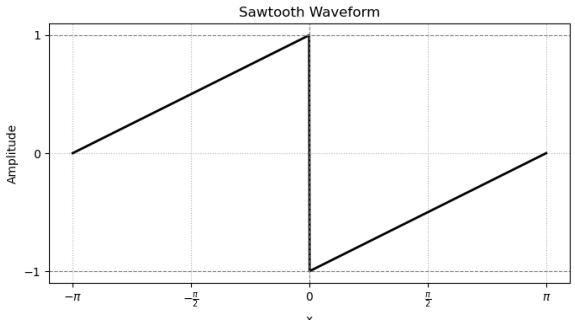


Figure 6: A simple sawtooth wave.

Mathematically, speaking this waveform can be defined with a time period T can be defined as:

$$f(t) = \begin{cases} t, & 0 \leq t \leq T \\ 0, & t \leq 0 \end{cases}$$

Similarly, the Fourier series of a sawtooth waveform, with amplitude A can be represented as:

$$f(t) = \frac{-2A}{\pi} \sum_{n=1}^{\infty} \frac{1}{n} \sin(n\omega_o t)$$

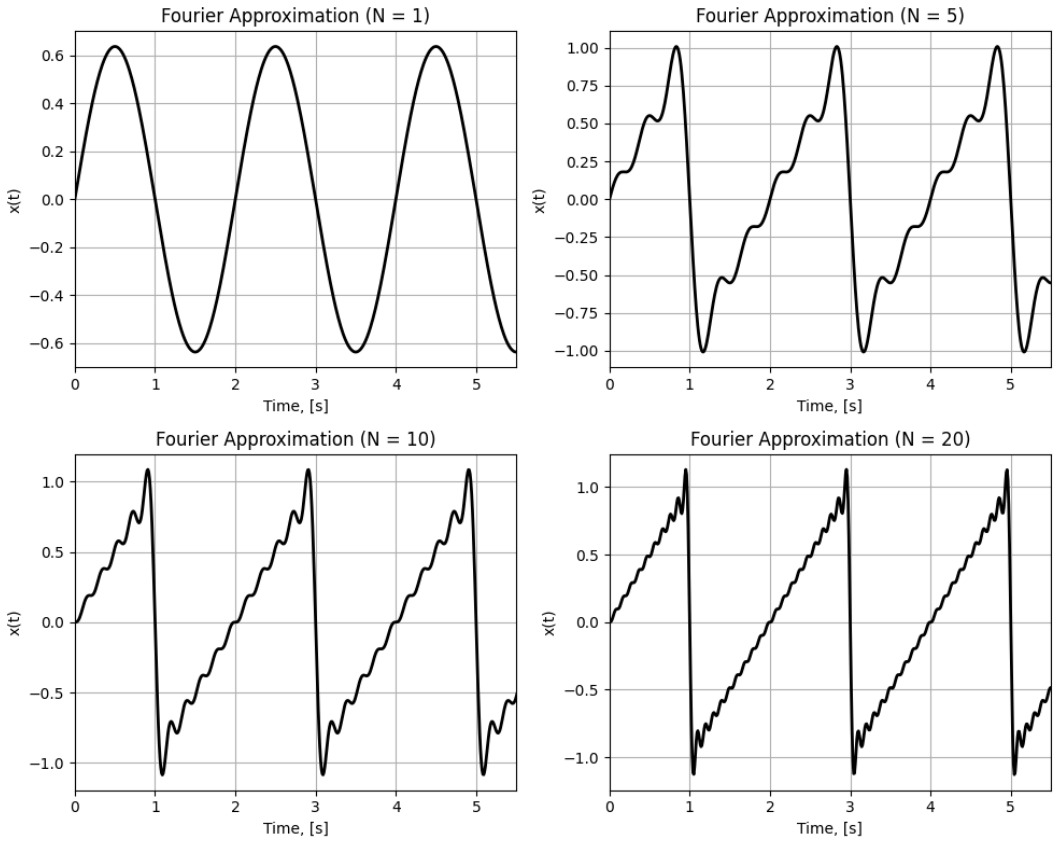


Figure 7: The convergence of a Fourier Series Expansion of a sawtooth waveform, including a) one term, b) two terms, c) three terms and d) 20 terms. The overshoots can be seen in figure c and d

The Fourier series of waveforms like triangular, sawtooth, and square waves is essential in many fields of physics, engineering, and signal processing. For example, the Fourier series these functions are used in decomposing Complex Signals. Many real-world signals are non-sinusoidal but periodic. Fourier series allows us to express them as sums of sine and cosine functions. For filtering and transmission purposes, knowing the Fourier series helps in designing low-pass, high-pass, and band-pass filters, which are essential in communication systems (e.g., radio, TV, and mobile networks).Additionally, the knowledge of Harmonic Analysis can be used in audio processing to analyse and synthesize sounds.

Now that we have seen how Fourier series can approximate different periodic waveforms, it is important to understand its broader applications in various domains:

(a) Signal Processing: Fourier series helps analyse and filter signals in audio processing, telecommunications, and image compression (e.g., MP3, JPEG, and video encoding formats like MPEG).

(b) Electrical Engineering: It is used to study AC circuits, power distribution, and harmonic analysis in electrical signals.

(c) Vibrations and Mechanical Systems: Engineers use Fourier techniques to analyse structural vibrations and mechanical oscillations in buildings, bridges, and vehicles.

(d) Quantum Mechanics and Optics: In Fourier Optics, lenses perform Fourier transforms of optical wavefronts, which is crucial in imaging systems. In quantum mechanics, wave functions can be decomposed into momentum-space representations using Fourier transforms.

3. The Fourier Transform

In the previous section we encountered the Fourier series representation of a periodic function in a fixed interval as superposition of sinusoidal functions. However, it is often desirable, to obtain such a representation even for functions defined over an infinite interval and with no particular periodicity. Such a representation is called a Fourier transform and is one of a class of representations called integral transforms.

We begin by considering Fourier transforms as a generalisation of the Fourier series. The Fourier transform provides a representation of functions defined over an infinite interval and having no particular periodicity, in terms of a superposition of sinusoidal functions. It may thus be considered as a generalisation of the Fourier series representation of periodic functions.

The Fourier transform is a powerful mathematical tool as it allows us to represent a function as a superposition of sinusoidal functions of varying frequencies, thereby capturing its frequency components in continuous form. Unlike the Fourier series, which is applicable only to periodic functions, the Fourier transform accommodates functions without periodicity constraints, making it a versatile tool in signal processing, physics, and engineering. One of the key properties of the Fourier transform is its ability to convert a function from the time or spatial domain to the frequency domain, revealing the frequency spectrum inherent in the original signal. This transformation facilitates analysis and manipulation based on frequency characteristics, such as filtering, modulation, and compression, essential in fields such as telecommunications, image processing, and quantum mechanics [1].

Below, we define the 2D Fourier transform, used to analyse spatial frequency components in images, signals, and physics applications (e.g., wave propagation, optics, and quantum mechanics). It extends the 1D Fourier transform to two dimensions.

$$\mathcal{F}(f) = \iint_{-\infty}^{\infty} f(x,y) \exp\left(-2\pi j(f_x x + f_y y)\right) dx$$

It is evident from the above equation that the above defined function is complex in nature, a fact that will play an important role later in this report.

The inverse of this function, known as the inverse Fourier transform, is also widely used in Fourier optics. The formula is given as follows:

$$\mathcal{F}^{-1}(F) = \iint_{-\infty}^{\infty} F(f_x,f_y) \exp\left(2\pi(f_x x + f_y y)\right) df_x df_y$$

Note that as mathematical operations, the transform and inverse transform are very similar, differing only in the sign of the exponent appearing in the integrand. The inverse Fourier transform is sometimes referred to as the Fourier integral representation of a function $f(x,y)$.

One of the most important results of the Fourier Transform is the Fast Fourier Transform (FFT). FFT is an algorithmic implementation of the discrete Fourier transform (DFT) that dramatically accelerates the computation of Fourier transforms for discrete, finite sequences of data points. Developed by Cooley and Tukey [2] in the 1965, FFT efficiently computes DFT by reducing the number of operations from $O(N^2)$ to $O(N \cdot logN)$, where N is the number of data points. This computational efficiency has made FFT ubiquitous in digital signal processing, real-time analysis, and scientific computing applications. The FFT finds wide application in various domains such as spectral analysis, where it helps analyse the frequency content of signals, from audio signals in music processing to seismic data in geophysics.

Moreover, FFT plays a crucial role in modern technologies such as Magnetic Resonance Imaging (MRI) imaging in medicine and in the simulation of physical systems governed by differential equations in computational physics. Its speed and versatility have made it a cornerstone of digital signal processing, enabling complex analysis and manipulation of signals across diverse fields of science and technology.

In order to illustrate how FFT works, we have provided some FFT of some basic waveforms below.

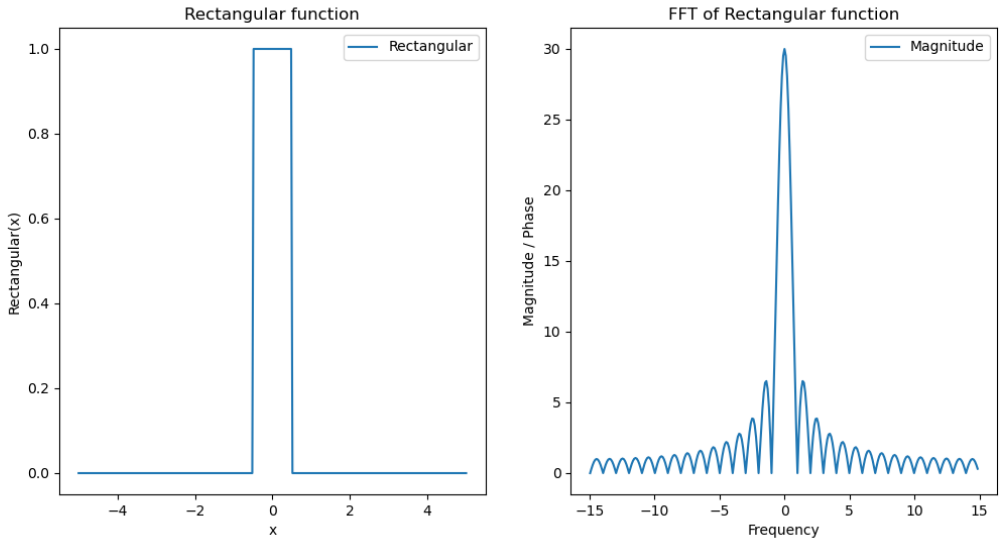


Figure 8: Fast Fourier Transform of a Rectangular Function.

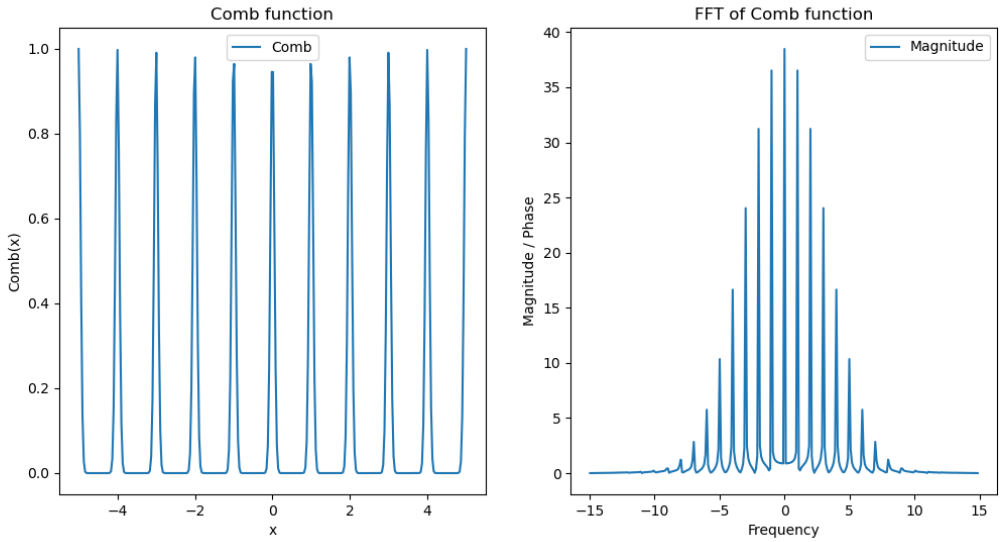


Figure 9: Fast Fourier Transform of a Comb Function.

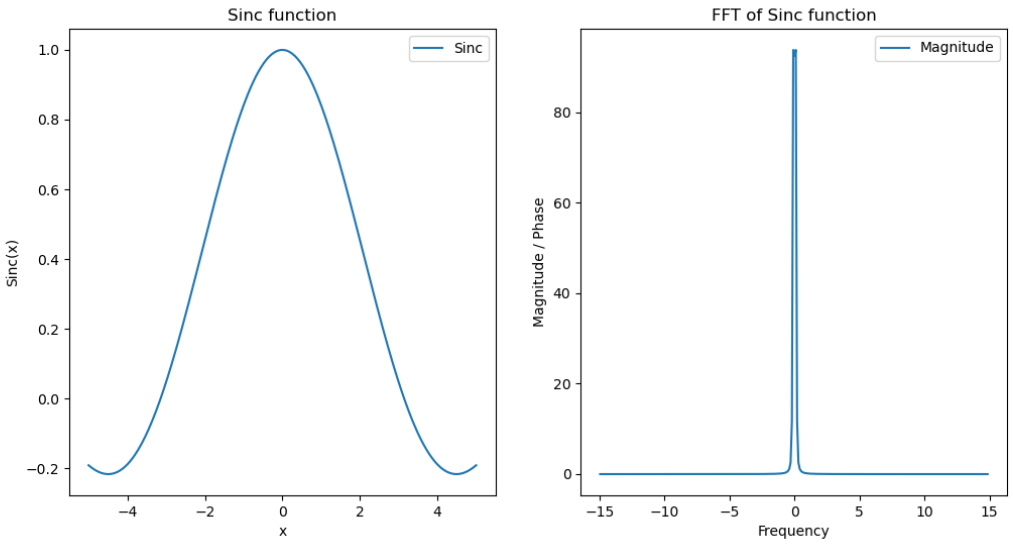


Figure 10: Fast Fourier Transform of a Sinc Function.

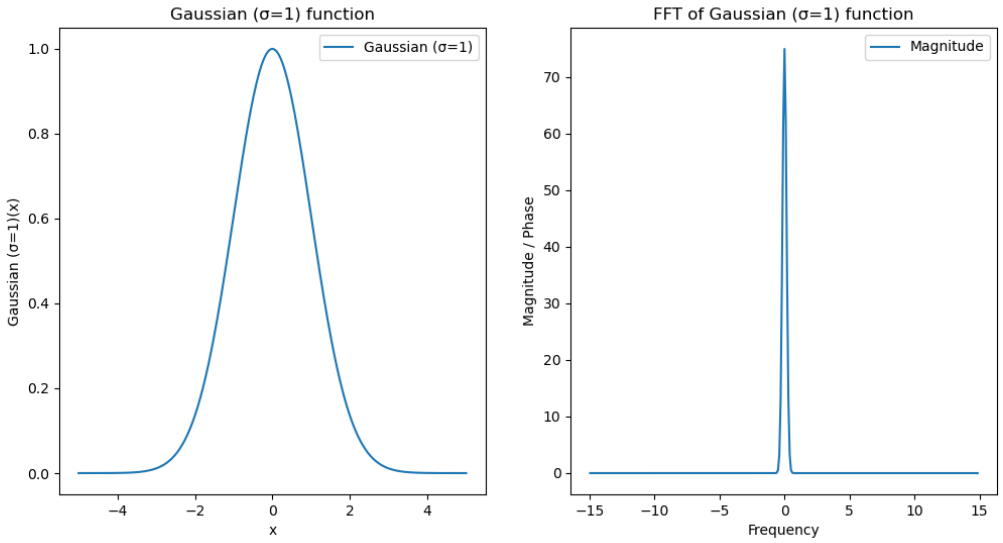


Figure 11: Fast Fourier Transform of a Gaussian Function.

4. The Evolution and Importance of Algorithmic Phase Retrieval

Interferometric phase retrieval exploits interference patterns to reconstruct wavefront phase information with high precision. By capturing intensity measurements from multiple interferograms typically obtained by varying a reference wave, these methods can retrieve phase distributions without requiring direct phase-sensitive detectors. However, they require highly stable setups and precise calibration to avoid noise and systematic errors. Algorithmic phase retrieval, in contrast, is a computational approach that overcomes these experimental limitations, making it a powerful alternative for high-resolution imaging and other precision-demanding applications. It provides an alternative to traditional interferometric methods, enabling phase recovery without requiring complex optical setups. These computational techniques reconstruct phase information from intensity measurements by leveraging prior knowledge or constraints, making them very helpful in applications such as coherent diffraction imaging (CDI), adaptive optics, and lensless microscopy.

In 1952, David Sayre proposed that the phase information of a scattered wave could be recovered if the intensity pattern between the Bragg peaks of a diffraction pattern was finely sampled. This is because, in crystallography, where the atomic arrangement is periodic, diffraction patterns exhibit sharp peaks that represent the Fourier transform of the underlying structure. Sayre's insight suggested that detailed intensity measurements could, in certain cases, enable phase recovery. However, practical computational phase-retrieval methods only emerged decades later. The first iterative algorithm for the retrieval of the phase of an optical wavefront was demonstrated by Gerchberg and Saxton in 1972 [3]. The Gerchberg-Saxton (GS) algorithm is an iterative phase retrieval technique used to reconstruct the phase of a complex wavefront from two intensity measurements, typically one in the object (spatial) domain and another in the Fourier domain. It is widely used in optical imaging, holography, and adaptive optics, as well as in applications like X-ray crystallography and lensless imaging. GS-based algorithms recover complex-valued wavefronts by iteratively propagating back and forth between two planes while applying constraints at each step. Iterative phase retrieval allows quantitatively correct and twin image-free reconstructions of the amplitude and phase distributions of the object from its inline hologram.

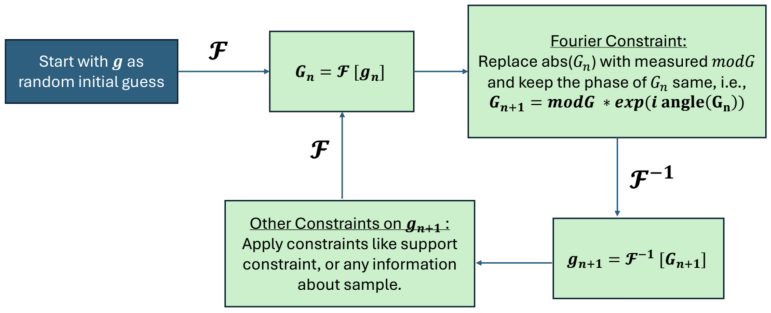


Figure 12: A basic block diagram of a Gerchberg-Saxton Algorithm

In 1978, James R. Fienup developed iterative algorithms namely the Error Reduction(ER) Algorithm and the Hybrid Input-Output(HIO) Algorithm that could retrieve phase information from a two-dimensional Fourier modulus, incorporating constraints like non-negativity and known support regions of the image through prior knowledge of the object [4]. This development spurred significant research activity, particularly in the 1980s, when optical computing was envisioned as a transformative technology. Unfortunately, as the limitations of optical computing became apparent, interest in algorithmic phase retrieval declined during the late 1980s and early 1990s.

The field experienced a resurgence in the late 1990s, largely driven by advances in X-ray imaging. The emergence of high-intensity X-ray sources, such as synchrotrons and undulators, rekindled interest in phase-retrieval techniques. In 1999, Miao et al. successfully recorded and reconstructed a continuous diffraction pattern from a non-crystalline (nonperiodic) test object, demonstrating the feasibility of phase retrieval in high-resolution imaging applications.

4.1. Gerchberg-Saxton Algorithm and the Hybrid Input-Output Algorithm

Since then, the phase retrieval problem has been under investigation for more than 50 years and continues to attract research efforts from all over the world. One of the most widely used methodologies for phase retrieval is the iterative Fourier transform (IFT) algorithm. The earliest version of the IFT is known as the error reduction (ER) algorithm, which is derived from the Gerchberg and Saxton (GS) algorithm. ER can be considered as an alternating projection method, where the Fourier magnitude constraint and the object domain constraints (e.g., support, positivity, etc.) are applied sequentially to an initial guess for the solution $g(x, y)$ in a repeated manner [5].

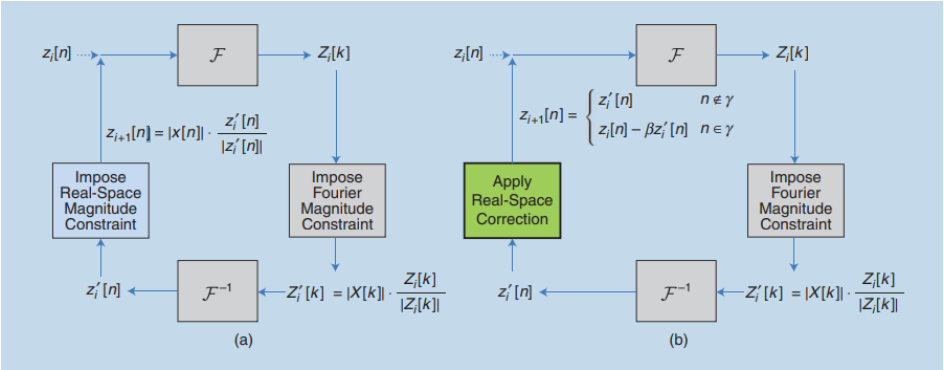


Figure 13: Block diagrams of (a) the Gerchberg-Saxton (GS) algorithm and (b) the Fienup Hybrid Input-Output (HIO) algorithm. The algorithms differ in their fourth (coloured) step.

The GS algorithm requires two intensity measurements: one in the sample plane and the second one in the detector plane. The algorithm assumes that the complex-valued wavefronts in the sample and the detector planes are connected through a Fourier transform (FT) with each other. The result of the algorithm is the recovered complex-valued wavefront distributions in the sample and the diffraction planes.

One of the key inferences from phase-retrieval research is the importance of Fourier phase in reconstructing images. The following figure demonstrates this concept:

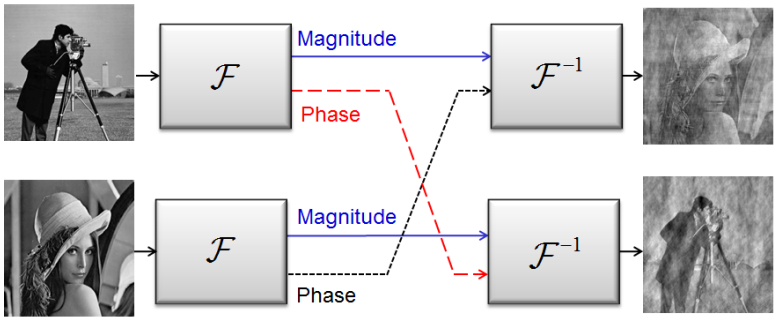


Figure 14: The importance of Fourier phase. Two images, a cameraman and Lena, are Fourier transformed. After swapping their phases, they are inverse Fourier transformed. The result clearly demonstrates the importance of phase information for image recovery.

Fienup's Hybrid Input-Output (HIO) algorithm improved upon the GS approach by introducing relaxation parameters, allowing the algorithm to escape local minima. This modification significantly enhanced the sturdiness of phase retrieval in practical applications. The HIO Algorithm is built from definition(one of many), and demonstrated in this report.

4.2. Iterative Phase Retrieval using Vortex Illumination

Iterative phase retrieval techniques play a significant role in reconstructing phase information from intensity-only measurements. However, one of the major challenges in phase retrieval algorithms is the twin-image stagnation problem, where the reconstructed phase converges to two possible solutions: the original image and its flipped, complex-conjugated twin. This ambiguity arises because Fourier magnitude constraints alone are insufficient to uniquely determine the phase, since different phase distributions can produce the same magnitude spectrum. As a result, twin-image artifacts can hinder convergence and degrade reconstruction accuracy.

The vortex phase illumination method offers a novel solution by introducing a controlled phase structure in the Fourier domain, effectively breaking the **symmetry** that leads to twin stagnation. By incorporating a phase vortex in the illumination as opposed to a uniform plane wave in its plane illumination counterpart, this approach introduces a structured zero-intensity point in the Fourier domain, which suppresses twin-image artifacts and improves convergence in iterative phase retrieval [6].

Need for the Vortex Illumination Method: In Fourier phase retrieval, we attempt to recover an object $g(x, y)$ from the magnitude of its Fourier transform:

$$G(f_x, f_y) = |G(f_x, f_y)|e^{i\phi(f_x, f_y)}$$

where $|G(f_x, f_y)|$ is the Fourier magnitude and $\phi(f_x, f_y)$ is the unknown Fourier phase.

Now, standard iterative methods like the Hybrid Input-Output (HIO) or Error Reduction (ER) algorithm struggle because the phase $\phi(f_x, f_y)$ is unknown and must be estimated iteratively. The struggle is because the function $g(x, y)$ and its flipped complex conjugate $g^*(-x, -y)$ have the same Fourier magnitude but differ in phase. Naturally, the computational complexity increases because now two phase values must be calculated simultaneously in order to know about the object.

The key idea behind vortex phase illumination is to intentionally introduce a strategical zero in the Fourier domain at low frequencies. This controlled phase structure breaks the mirror symmetry that causes twin stagnation [6]. A charge-1 vortex phase illumination (an optical field with a helical phase structure characterized by a phase singularity at its center) is introduced in the input plane as:

$$g_o(x, y) = e^{i\theta(x,y)}$$

where

$$\theta(x, y) = tan^{-1}\frac{y}{x}$$

This spiral phase creates a central phase singularity where the intensity is zero in the Fourier domain. In other words, it can be stated that the Fourier transform of a vortex phase object contains a zero-intensity point at the centre, given by:

$$G_o(f_x, f_y) \approx e^{i\phi(f_x, f_y)} \cdot J_1(kr)$$

where $J_1(kr)$ is a Bessel function of the first kind, which describes a describing an annular diffraction pattern with a dark central core [7]. This structured illumination modifies the Fourier domain representation of the

object, ensuring that no twin-image solutions satisfy the Fourier magnitude constraints. After retrieving the phase using HIO, the reconstructed object will still contain vortex phase terms:

$$g_{retrieved}(x, y) = g_{true} \cdot e^{i\theta(x,y)}$$

To obtain the original phase object $g_{true}(x, y)$, we multiply the retrieved image with the conjugate vortex function, i.e.,

$$g_{true}(x, y) = g_{retrieved} \cdot e^{-i\theta(x,y)}$$

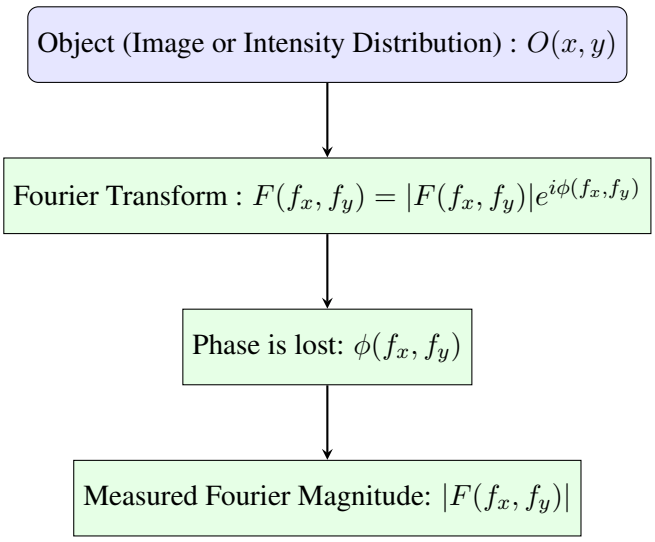
This correction restores the original phase distribution, thus accurately reconstructing the image.

Please turn to the next page.

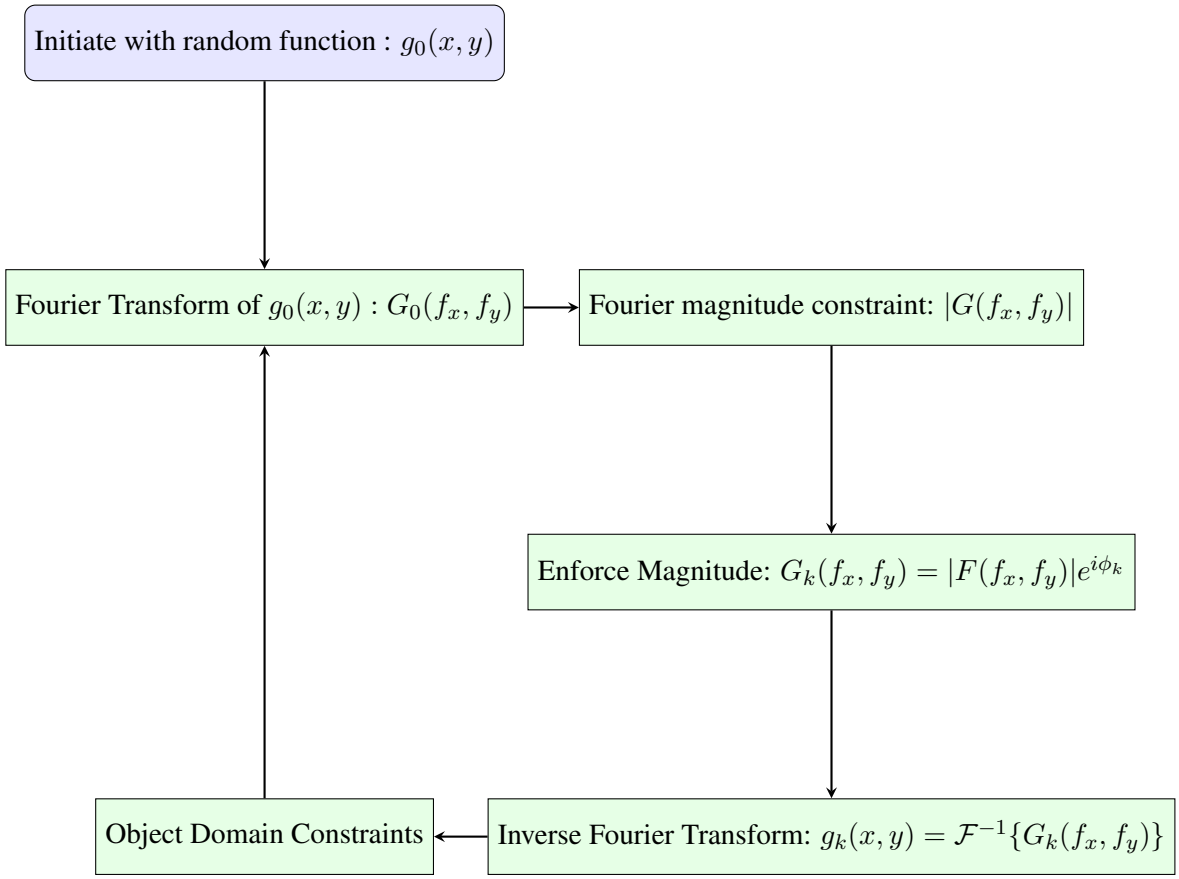
5. Methodology

Algorithm: In order to understand the algorithm visually, we have provided a pictorial representation below.

Step 1: Calculation of the experimental values of the Fourier Magnitude of the Object:



Step 2: Initialisation and Phase Determination:



The following steps explain the process theoretically:

Step 1: Fourier Transform and Phase-Retrieval Problem:

(a) If $O(x, y)$ represents an object (such as an image or an intensity distribution), its Fourier transform is given by

$$F(f_x, f_y) = |F(f_x, f_y)|e^{i\phi(f_x, f_y)}$$

where $|F(f_x, f_y)|$ is the modulus (magnitude) of the Fourier transform and $\phi(f_x, f_y)$ is the phase.

(b) In many practical cases (such as astronomical imaging), only $|F(f_x, f_y)|$ can be measured due to limitations in optical systems, while $\phi(f_x, f_y)$ is lost. However, reconstructing $O(x, y)$ requires knowledge of both the modulus and phase. The challenge is to infer $\phi(f_x, f_y)$ from the available data.

Step 2: Iterative Phase Retrieval using Hybrid Input-Output (HIO) Algorithm:

(a) **Initialization**: The reconstruction starts with a random or uniform phase distribution in the spatial domain, constructing a complex initial field:

$$U_0(x, y) = \sqrt{I_0(x, y)}e^{i\phi_0(x,y)}$$

(b) **Forward Fourier Transform**: Compute the Fourier transform:

$$F_0(f_x, f_y) = \mathcal{F}\{U_0(x, y)\}$$

(c) **Fourier Magnitude Constraint**: Replace the magnitude of $F_0(f_x, f_y)$ with the known Fourier magnitude $|F(f_x, f_y)|$ while keeping the computed phase:

$$G_k(f_x, f_y) = |F(f_x, f_y)|e^{i\phi_k(f_x, f_y)}$$

(d) **Inverse Fourier Transform**: Compute the inverse transform to return to the spatial domain:

$$g_k(x, y) = \mathcal{F}^{-1}\{G_k(f_x, f_y)\}$$

(e) **Object Domain Constraints**: Apply the Hybrid Input-Output (HIO) update rule to enforce constraints in the spatial domain:

$$O^{(k+1)}(x, y) = \begin{cases} O'^{(k)}(x, y), & \text{if } (x, y) \in \text{Support}, \\ O^{(k)}(x, y) - \beta\big(O^{(k)}(x, y) - O'^{(k)}(x, y)\big), & \text{otherwise.} \end{cases}$$

Here,

- $O^{(k+1)}(x, y)$ is the updated object estimate in iteration $k + 1$.
- $O'^{(k)}(x, y)$ is the newly obtained function after the inverse Fourier transform in iteration k.
- The **support** refers to the known region where the object is expected to exist.
- β is a relaxation parameter, typically set between 0.5 and 1, which helps control the update strength and prevent stagnation. Here, $\beta = 0.8$ is used.

(f) The above steps are repeated for several iterations until convergence is achieved, yielding the reconstructed phase.

6. Results

This section presents the simulation and recovery of a phase object using the Hybrid Input-Output (HIO) algorithm for two types of illumination: **plane wave illumination** and **vortex illumination**. The quality of reconstruction in both cases is analysed and compared.

6.1. Phase Object Generation

A 256×256 grayscale image (Lena) was used as the phase function. The image was normalized to the range $[0, 1]$ and converted into a pure phase object:

$$O(x, y) = e^{i\phi(x,y)}$$

Zero padding was added around the image to enforce support constraints in the reconstruction process. The visualization of the phase object with unit amplitude is shown in Figure 15.

Figure 15: Original Phase Object

6.2. Fourier Magnitude Extraction

The phase object was illuminated with a plane wave (amplitude = 1). The Fourier transform of the object was computed, and only the Fourier magnitude was retained while discarding the phase. This magnitude-only information served as the input for the phase retrieval process.

6.3. Phase Retrieval using HIO Algorithm (Plane Wave Illumination)

The HIO algorithm was implemented with an initial random phase guess. The algorithm was run for 500 iterations with a feedback parameter of $\beta = 0.8$. Over iterations, the reconstruction gradually improved, showing increasing similarity to the original phase object. The final retrieved phase object is shown in Figure 16a.

(a) Reconstructed Phase Object (Plane Illumination)

(b) Reconstructed Phase Object (Vortex Illumination)

Figure 16: Comparison of reconstructed phase objects using plane wave and vortex illumination.

6.4. Phase Retrieval using HIO Algorithm (Vortex Illumination)

Instead of a plane wave, the object was illuminated with a vortex beam characterized by a spiral phase:

$$O_v(x, y) = O(x, y) \cdot e^{i \tan^{-1}(y/x)}$$

This vortex illumination introduced a phase singularity, modifying the Fourier transform and breaking the twin-image stagnation problem. The HIO algorithm was run for 400 iterations under these conditions, and the final reconstruction is presented in Figure 16b.

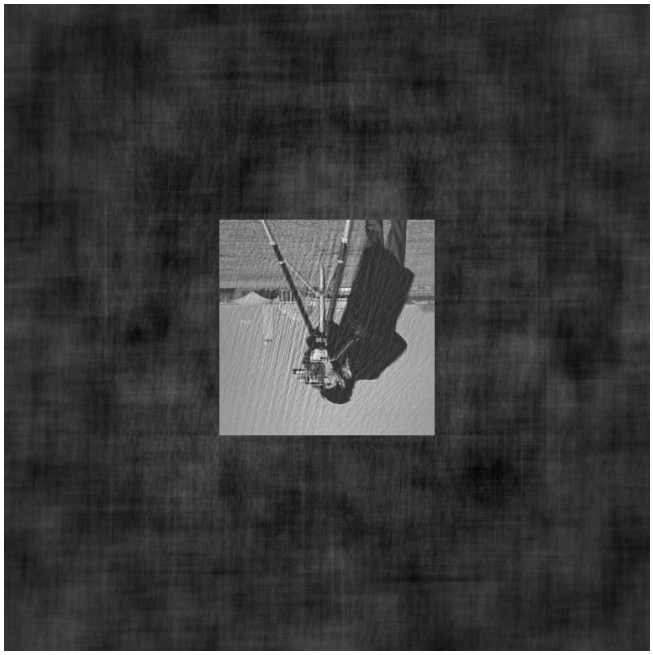
6.5. Comparison of Plane Wave and Vortex Illumination Reconstructions

The results of the phase retrieval process using both illumination methods are summarized in Table 1, and the reconstructed phase objects are presented side-by-side in Fig 16. The table presents a comparative analysis of reconstruction quality, convergence behavior, and the presence of twin-image artifacts. Notably, vortex illumination helps mitigate the twin-image problem, leading to a more accurate retrieval of phase information. These observations are further supported by the reconstructed images shown in Figures 17a and 17b.

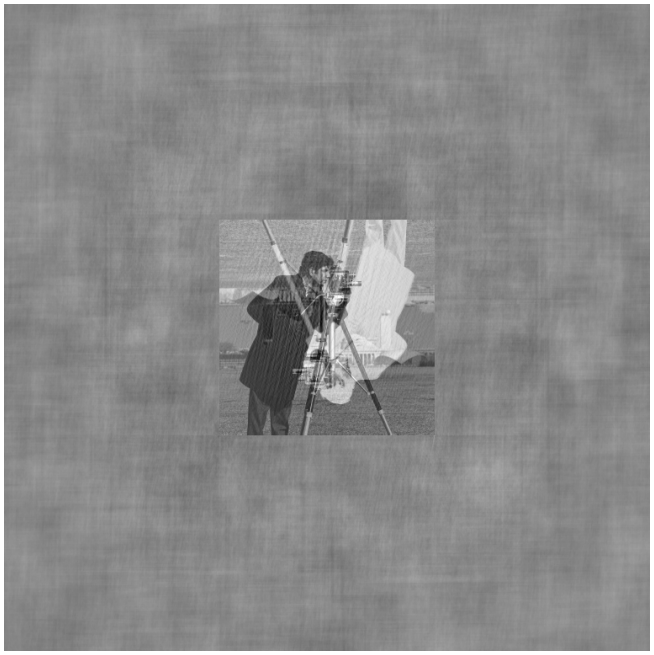
Illumination Type	Reconstruction Quality	Artifacts Present	Convergence Speed
Plane Wave	Satisfactory, but with phase ambiguities	Twin-image artifacts observed	Slower
Vortex Beam	Improved accuracy, sharper phase retrieval	No twin-image artifacts	Faster

Table 1: Comparison of Phase Retrieval Results for Different Illumination Conditions

The reconstructed phase object using plane wave illumination exhibited noticeable twin-image artifacts, characterized by a mirrored duplication of the original phase structure. This is a well-known issue in phase retrieval due to the ambiguity in Fourier magnitude constraints. In contrast, the vortex illumination method significantly suppressed these artifacts by introducing a controlled phase structure, effectively breaking the symmetry responsible for twin-image stagnation.



(a) Inverted cameraman image obtained during reconstruction



(b) After taking difference between the original and reconstruction

Figure 17: The twin image problem encountered during the algorithm execution with plane illumination for the cameraman image.

6.6. Discussion and Conclusion

The results of the phase retrieval process highlight the advantages of using vortex illumination over standard plane wave illumination. In the case of plane wave illumination, the reconstructed phase object exhibited noticeable twin-image artifacts during arbitrary program executions, a common challenge in iterative phase retrieval due to the ambiguity in Fourier magnitude constraints. These artifacts degrade the reconstruction quality and slow down convergence. Also, introducing minor tweaks in the handling of certain variables in the program environment unpredictably led to this phenomena sometimes, which should otherwise be at all unrelated to the reconstruction problem at hand.

On the other hand, vortex illumination significantly mitigated these issues by introducing a controlled phase singularity in the Fourier domain. This structured illumination altered the symmetry properties of the retrieved phase, effectively reducing twin-image stagnation. As a result, the convergence rate of the HIO algorithm improved, leading to a more accurate reconstruction with better phase fidelity. The twin image problem was never encountered while implementing vortex illumination.

These findings demonstrate that structured illumination techniques, such as vortex beams, can enhance the reliability of iterative phase retrieval algorithms. By carefully designing the illumination phase structure, it is possible to achieve faster convergence and higher reconstruction accuracy, which has important implications for applications in computational imaging and optical phase retrieval. Please refer the Appendix **??** for a detailed report on the program code, results and necessary comparison.

7. Applications of Iterative Phase Retrieval

The Hybrid Input-Output (HIO) algorithm offers an effective way to solving the phase retrieval problem, particularly in fields where direct phase measurement is infeasible. By iteratively refining an estimated phase solution under known constraints, the HIO algorithm enables accurate reconstruction of complex wavefronts from intensity-only measurements. This capability has made it a widely adopted tool in diverse scientific and technological applications, ranging from optical imaging to crystallography and even machine learning.

One of the most significant applications of the HIO algorithm is in coherent diffractive imaging (CDI), where it facilitates high-resolution imaging without the need for high-quality lenses.

7.1. Coherent Diffractive Imaging (CDI) and the Phase Retrieval Problem

Coherent Diffractive Imaging (CDI) is a technique that enables high-resolution imaging of nanoscale structures by utilizing the principles of diffraction and phase retrieval. As discussed by [8], in a basic CDI setup, an object is illuminated by a quasi-monochromatic coherent wave, and the diffracted intensity pattern is recorded at a detector placed at a sufficiently large distance from the object. Since conventional optical detectors can only measure intensity and not phase, due to the fact that detectors respond to photon energy, which is proportional to intensity rather than the full complex wave amplitude. So, retrieving the complete complex wavefront requires solving the phase retrieval problem. One of the key reasons phase retrieval algorithms work well in optics is due to a fundamental property of electromagnetic waves: under appropriate conditions, the far-field diffraction pattern of an object is related to its near-field distribution by a Fourier transform. This means that by measuring the intensity pattern in the far field and with some additional prior knowledge or constraints, it is possible to algorithmically reconstruct the missing phase information.

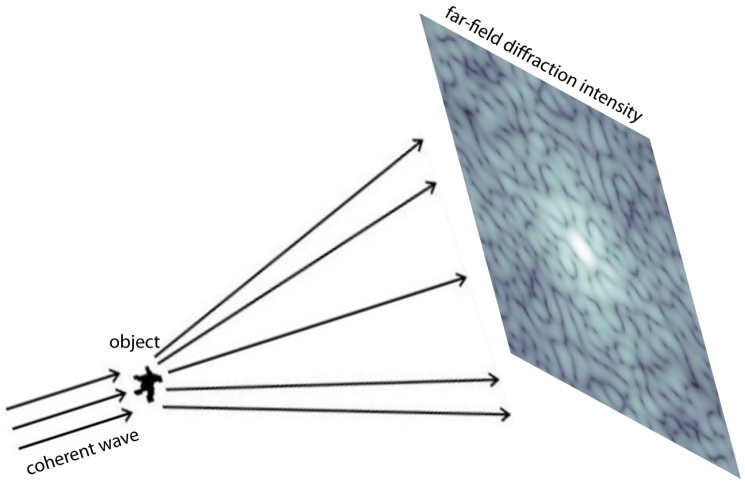


Figure 18: A forward-scattering Coherent Diffractive Imaging (CDI) setup. A coherent wave illuminates the object, causing diffraction. The resulting far-field intensity pattern, recorded at the detector, corresponds to the magnitude of the Fourier transform of the object, enabling phase retrieval for image reconstruction.

When the object is sufficiently small and the detector is positioned in the far field, the measured intensity is proportional to the magnitude of the Fourier transform of the wave at the object plane, subject to an appropriate spatial scaling factor.

Mathematically, this condition is satisfied when the Fresnel number N_{F} is small, given by:

$$N_{\mathrm{F}} = \frac{a^2}{\lambda d} \ll 1$$

where, a is the radius confining the object in the object plane, d is the distance between the object and the detector and λ is the wavelength of the illuminating wave.

In this regime, the relationship between the measured intensity I_{out} at the detector and the input field E_{in} at the object plane is given by:

$$I_{\text{out}}(x, y) \propto \left| \hat{E}_{\text{in}} \left(\frac{x}{\lambda d}, \frac{y}{\lambda d} \right) \right|^2$$

where, $\hat{E}_{\text{in}}$ is related to the measured intensity via the Fourier transform:

$$\hat{E}_{\text{in}} = \mathcal{F}\{E_{\text{in}}\}$$

Here, $\mathcal{F}$ represents the Fourier transform operation. However, due to the loss of phase information during intensity measurement, directly inverting the Fourier transform to reconstruct E_{in} is not possible. This makes CDI an ideal application for iterative phase retrieval algorithms, such as the Hybrid Input-Output (HIO) algorithm and the Gerchberg-Saxton (GS) algorithm, which iteratively refine an estimate of the missing phase to reconstruct the object.

CDI has revolutionized high-resolution imaging in various scientific domains, including materials science, biology, and nanotechnology. Its ability to bypass traditional lens-based imaging limitations makes it particularly valuable for imaging structures at the nanoscale, where high-resolution lenses are difficult to fabricate and often introduce aberrations. By leveraging algorithmic phase retrieval, CDI continues to push the boundaries of diffraction-limited imaging, enabling researchers to study complex structures with unprecedented clarity.

7.2. Phase Retrieval in Astronomy

In astronomy, images are often degraded because of optical aberrations, atmospheric turbulence, and instrumental limitations. This is especially critical in radio interferometers like ALMA (Atacama Large Millimeter/submillimeter Array) and VLA (Very Large Array), which inherently measure only the Fourier magnitude of an image, while the phase information is lost due to observational limitations. The Hybrid Input-Output (HIO) algorithm and Iterative Phase Retrieval (IPR) methods are powerful techniques used to reconstruct high-resolution images of celestial objects from incomplete or noisy data. These methods prove crucial in recovering lost information from the source, significantly enhancing imaging across various fields, such as adaptive optics, radio interferometry, exoplanet imaging, and diffraction-limited imaging.

Phase retrieval is an important computational technique used in astronomical imaging to reconstruct wavefronts from intensity measurements. The method plays an essential role in various astronomical applications, including active optics, adaptive optics, interferometry, and space-based telescopes. Recovering phase information allows the correction of distortions caused by atmospheric turbulence and instrumental aberrations, ultimately leading to sharper and more accurate images of celestial objects [9] [10] [11].

7.2.1. Phase Retrieval and Optical Aberrations

One of the primary applications of phase retrieval is in wavefront sensing, where phase distortions at the entrance pupil of an imaging system are reconstructed from intensity measurements in the image plane. In

telescopic or optical imaging systems, these distortions arise due to optical imperfections, misalignments, or atmospheric turbulence that affects the final image quality.

Since phase cannot be directly measured using conventional detectors, it must be inferred from intensity measurements in the image plane. The goal of phase retrieval is to reconstruct these distortions using the point spread function (PSF) $p(x)$, which describes how a point source appears in the image. The PSF is related to the coherent spread function $h(x)$, which describes how a point source of light propagates through an optical system when the system is perfectly coherent. It represents the response of the imaging system to a single point source, including both amplitude and phase information, as follows:

$$p(x) = |h(x)|^2.$$

Using the Fourier transform, the coherent transfer function $H(f)$ can be expressed as:

$$H(f) = \int h(x) e^{i2\pi f x} dx.$$

This function describes how different spatial frequency components of the image are transmitted through the system. Since conventional detectors only measure intensity ($|H(f)|^2$), phase retrieval algorithms reconstruct the missing phase $\theta(f)$ in:

$$H(f) = A(f) e^{i\theta(f)},$$

where $A(f)$ is the aperture function and $\theta(f)$ is the phase information that needs to be retrieved. Once retrieved, this phase information can be used to correct aberrations in telescopic imaging systems [9].

7.2.2. Interferometry and Phase Retrieval

In astronomical interferometry, phase retrieval techniques reconstruct object phase information from intensity measurements taken at different baselines. This is crucial for imaging faint celestial objects with high angular resolution. The error metric for interferometric phase retrieval is often minimized using:

$$E_c = \int |z(f) - P_c(f) D(f) O(f)|^2 \, df.$$

where $z(f)$ is the observed signal, $P_c(f)$ is the system's coherent transfer function, $D(f)$ represents detector response, and $O(f)$ is the object spectrum. Using multiple measurements at different focal planes enhances the phase reconstruction accuracy [10][12].

7.2.3. Phase Retrieval in Adaptive Optics

Adaptive optics (AO) actively corrects for wavefront distortions in real-time, significantly improving astronomical image resolution. Phase retrieval algorithms assist in AO systems by estimating the wavefront errors that need correction. The correction process follows:

(a) **Measurement of the Distorted PSF:** The point spread function (PSF) characterizes how an optical system images a point source. In an ideal diffraction-limited system, the PSF is sharply concentrated. However, due to wavefront distortions caused by atmospheric turbulence, optical misalignments, or instrumental aberrations, the observed PSF deviates from this ideal form. To correct for these distortions, an initial measurement of the distorted PSF is required. This can be done using a bright reference star or an artificial guide star created by a laser, allowing the system to quantify the deviations from the ideal PSF.

(b) **Estimation of the Wavefront Phase $\theta(f)$ using Phase Retrieval:** Directly measuring the phase of incoming wavefronts is challenging because optical detectors typically record only intensity information. Phase retrieval techniques address this issue by inferring the missing phase information from multiple intensity images taken under different conditions (e.g., with intentional defocus). Algorithms such as the Hybrid Input-Output (HIO) method, Gerchberg-Saxton algorithm, or phase diversity techniques iteratively reconstruct the phase $\theta(f)$, allowing a detailed understanding of wavefront aberrations.

(c) **Application of Corrections via Deformable Mirrors:** Once the wavefront errors are estimated, corrections are applied using adaptive optics elements such as deformable mirrors (DMs). A DM consists of an array of actuators that can dynamically adjust the mirror surface to compensate for the detected distortions. The required correction is computed from the retrieved phase information, and the actuators adjust the mirror shape in real time to counteract the aberrations. This step is crucial for restoring sharp imaging capabilities, particularly in astronomical observations where atmospheric turbulence significantly degrades image quality.

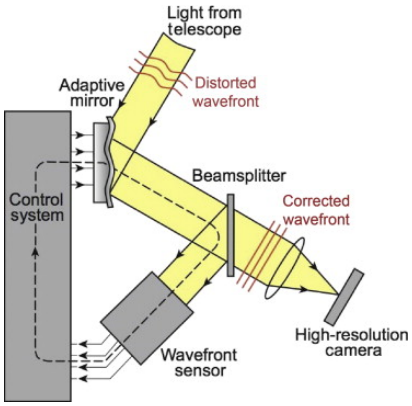


Figure 19: Simplified diagram of an AO system. Light from the telescope is collimated and sent to an adaptive or deformable mirror. If there were no atmospheric turbulence, the wavefront of the light would be perfectly straight and parallel. The light is then reflected to a beam splitter, where part of the light is reflected to the wavefront sensor. The wavefront sensor measures the distortion of the wavefront and sends a correction signal to the adaptive mirror. The adaptive mirror is capable of changing its shape to remove the deformations in the light wave caused by the atmospheric turbulence. Light with a corrected wavefront thus reaches the high-resolution camera, where a diffraction-limited image is formed. Image credit: Science Direct; C. Max.

(d) **Continuous Iteration Until Diffraction-Limited Performance is Achieved:** The correction process is not a one-time adjustment but an iterative procedure. Each correction improves the system's performance, but residual errors may persist due to dynamic changes in atmospheric conditions or instrumental imperfections. By continuously monitoring the PSF, refining the phase retrieval process, and updating the deformable mirror adjustments, the system gradually approaches diffraction-limited imaging. This iterative loop is essential for achieving high-contrast, high-resolution observations, such as those required for exoplanet imaging or deep-space surveys.

This technique has been successfully used in ground-based observatories such as Keck and the Very Large Telescope (VLT), enabling sharper images by mitigating atmospheric turbulence [13] [14].

7.2.4. Phase Retrieval for Space Telescopes

Phase retrieval has been instrumental in space-based observatories, providing precise wavefront sensing and optical correction. A notable example is the **Hubble Space Telescope (HST)** and the infamous spherical aberration in its primary mirror. Investigation uncovered that Hubble's primary mirror had the wrong curve

and it was too flat near its outer edge so light landing on that outer edge had different focal points than the other parts of the mirror. This flaw, called spherical aberration, gave Hubble's primary mirror more than one focal point, which made its images blurry. The error was 10 times larger than the specified tolerance needed for its design to work properly. Phase retrieval techniques came to the rescue and were used to diagnose it. By analyzing defocused images, these algorithms identified the exact nature of the defect, leading to the implementation of the **Corrective Optics Space Telescope Axial Replacement (COSTAR)**, which restored HST's imaging capabilities (See figures 20 and 21).

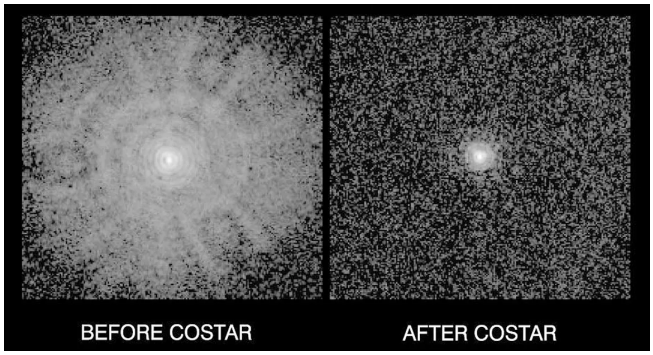


Figure 20: This pair of images of a single star, called Melnick 34, taken with the Hubble's Faint Object Camera, helped demonstrate that Hubble was fully restored to its planned optical performance.

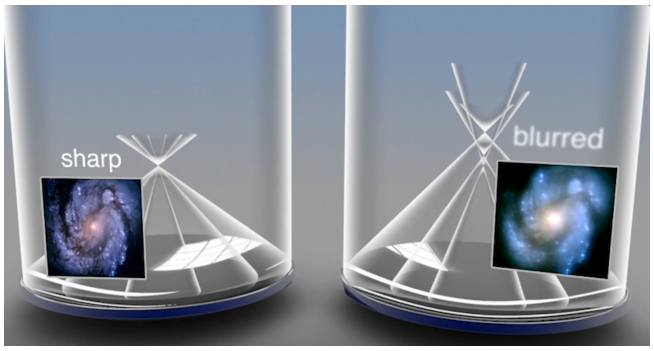


Figure 21: The spherical aberration (right) in Hubble's mirror created multiple focal points, which made its images blurry. This was corrected by COSTAR hence drastically improving performance.

Correction of Hubble Space Telescope's primary mirror defect using COSTAR optics. Credit: NASA, ESA, and the COSTAR Team

The **James Webb Space Telescope (JWST)** also relies extensively on phase retrieval, but with added complexity due to its **segmented primary mirror**. Unlike HST, which has a monolithic mirror, JWST's 18 hexagonal segments must be aligned to nanometer precision to achieve diffraction-limited performance. JWST employs phase retrieval algorithms in an **iterative wavefront sensing and control (WFSC) process** to correct segment misalignments and optimize optical performance.

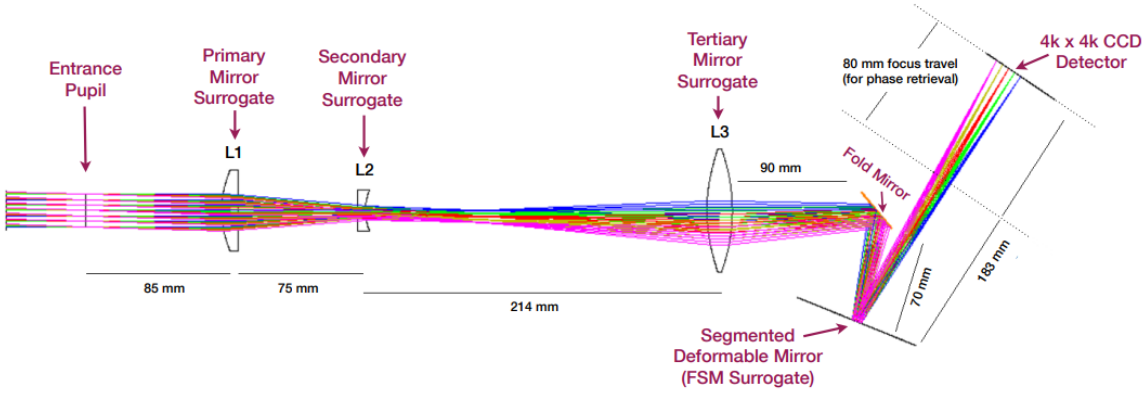


Figure 22: Optical layout for JOST, with components labeled. A collimated beam enters the system from left, passes through the three refractive optics and then reflects off the segmented deformable mirror to the detector. A flat fold mirror is included before the deformable mirror for packaging reasons to provide sufficient clearance between the detector and the L3 lens. The detector can be translated up to 8 cm to obtain defocus for phase retrieval. Credit: Marshall D. et al., JWST OST I: Overview and First Results (arXiv:1407.0591v1).

A key component of JWST's **wavefront sensing system** is its ability to capture multiple defocused images using its **Near-Infrared Camera (NIRCam)**. These images are then processed to determine phase errors, allowing precise adjustments to be made to the mirror segments. This process is critical because JWST operates at **Lagrange Point 2 (L2)**, nearly 1.5 million km from Earth, where no direct physical realignment is possible after launch.

The JWST Optical Simulator Testbed (JOST) is designed as a three-lens anastigmat system, analogous to JWST's three-mirror anastigmat design. As illustrated in Figure 22, a collimated beam enters the system, passing through three refractive optics before reflecting off the segmented deformable mirror (FSM surrogate) and reaching the 4k × 4k CCD detector. The primary mirror surrogate is a deformable mirror with 37 small adjustable segments, but only 18 are used. The secondary mirror surrogate is a diverging lens mounted on a motorized tip/tilt and translation stage, while the tertiary mirror surrogate is a converging aspherical lens ensuring the correct focal ratio. The detector's translation allows focus-diverse phase retrieval, and a Fizeau interferometer (4D AccuFiz) can be integrated for direct wavefront sensing validation. This setup operates at visible wavelengths (HeNe laser, 632.8 nm) and is optimized for diffraction-limited imaging with wavefront error tolerances as low as 10–20 nm RMS across a wide field of view.

A major advantage of **phase diversity** (refer the section on Phase Diversity) over conventional interferometric wavefront sensing is that it does not require a separate reference beam or additional hardware. This makes it particularly suited for space telescopes, where mass, power, and system complexity are limited. JWST's **phase diversity wavefront sensing** allows for robust phase error measurement and correction, ensuring that the segmented mirror system remains properly aligned throughout its mission.

Beyond space telescopes, **phase diversity** has also been widely adopted in **ground-based adaptive optics (AO)**. By continuously capturing **defocused images** and correcting for **atmospheric turbulence** in real time, AO-equipped telescopes—such as the **Very Large Telescope (VLT)** and **Keck Observatory**—achieve sharper images of celestial objects, rivaling the resolution of space telescopes.

Mathematically, phase diversity reconstructs phase information from multiple defocused images by solving the equation:

$$Z(f) = P(f)D(f)O(f) + N(f),$$

where $P(f)$ represents the **pupil function**, $D(f)$ accounts for the **defocus**, $O(f)$ is the **object being imaged**, and $N(f)$ represents **noise**. This technique enhances phase reconstruction accuracy, enabling space observatories to maintain optimal imaging performance [12] [15].

Wavefront Sensing and Correction in Space Telescopes: Maintaining diffraction-limited imaging performance over an extended mission lifetime requires continuous monitoring and correction of optical errors. Phase retrieval techniques contribute significantly to this active wavefront sensing and correction process.

- **Hubble Space Telescope (HST):** After the discovery of spherical aberration in HST's primary mirror, phase retrieval methods, including phase diversity, were used to diagnose the issue and guide the design of the **Corrective Optics Space Telescope Axial Replacement (COSTAR)** system.
- **James Webb Space Telescope (JWST):** Phase diversity plays a crucial role in JWST's **wavefront sensing and control (WFSC)** system. The telescope captures multiple defocused images and applies a **modified Gerchberg-Saxton algorithm** to estimate and correct phase aberrations in its segmented mirror system, ensuring **diffraction-limited imaging**.

Mathematical Representation of Wavefront Reconstruction: In the **spatial frequency domain**, images recorded by a space telescope can be represented as:

$$I_1 = P_1 O, \quad I_2 = P_2 O$$

where I_1 and I_2 are intensity measurements from different focal planes (e.g., in-focus and defocused images), P_1 and P_2 represent the corresponding point spread functions (PSFs), and O is the unknown object (or wavefront phase) to be retrieved. Multiplying each equation by the **complex conjugate** of its corresponding PSF and solving for O gives:

$$O = \frac{I_1 \cdot \text{conj}(P_1) + I_2 \cdot \text{conj}(P_2)}{\text{Abs}(P_1)^2 + \text{Abs}(P_2)^2}$$

This formulation enables **object-independent phase retrieval**, meaning that the wavefront phase can be reconstructed without prior knowledge of the observed astronomical source. This is particularly useful for space telescopes that observe a variety of celestial objects with unknown structures [16].

While conventional phase retrieval techniques effectively reconstruct wavefronts, they often suffer from ambiguities and slow convergence. To address these challenges, phase diversity introduces controlled optical variations to improve phase estimation accuracy. This technique has been widely adopted in space telescopes, adaptive optics, and high-contrast imaging.

Phase Diversity: Phase diversity (PD) is a technique that enhances **phase retrieval (PR)** by introducing controlled **phase variations (diversities)** into an imaging system. Unlike traditional PR methods, which require prior knowledge of the object for deconvolution, PD jointly estimates both the **unknown phase** T and the **object** o by capturing multiple images with strategically introduced phase aberrations. This additional information eliminates ambiguities in phase retrieval and improves convergence, making PD particularly useful in high-precision optical systems like **adaptive optics (AO), space telescopes, and microscopy**.

When a telescope observes an object o, the measured image i is given by:

$$i = p \otimes o$$

where p is the **point spread function (PSF)**, which describes how the optical system blurs the object, and $\otimes$ denotes convolution. If the object o is known, the PSF p can be determined through deconvolution, and standard phase retrieval methods can reconstruct the missing phase information T. However, in practicality, the object o is unknown, making the retrieval of T an inconvenient inverse problem with multiple possible solutions. To overcome this, PD captures multiple images $i_1, i_2, \ldots$, each recorded with a different intentional phase diversity $D_1, D_2, \ldots$, applied to the optical system, such that:

$$T_2 = T + D_2, \quad T_3 = T + D_3, \ldots$$

Here, T_i is the total phase in presence of the applied phase diversity D_i. D_i quantifies a deliberate, known phase aberration intentionally introduced into an optical system when capturing multiple images. These aberrations modify the wavefront in a controlled manner, altering the recorded intensity patterns in a predictable way. The most common type of phase diversity is defocus, but other wavefront modifications such as astigmatism or spherical aberrations can also be used.

By incorporating phase diversity, the problem at hand becomes **overdetermined** (since we have a system of multiple equations with a single unknown), significantly enhancing phase retrieval accuracy. The estimated phase T can then be used to correct aberrations in **adaptive optics (AO)**, enabling nearly **diffraction-limited imaging**. This approach—sometimes called **Phase-Diverse Phase Retrieval (PDPR)**—is significantly more robust than conventional PR methods because it resolves **twin-image ambiguity** and improves algorithmic stability [16].

While phase diversity has proven to be an effective enhancement to phase retrieval, it is not the only method used for extracting phase information from intensity images. Several alternative techniques exist, each with its own strengths and limitations. The next section compares phase diversity with other focal-plane phase retrieval approaches, such as the Gerchberg-Saxton algorithm and Shack-Hartmann sensing.

Comparison with Other Focal Plane Phase Retrieval Methods: Phase diversity (PD) belongs to a broader class of **focal-plane phase retrieval methods**, which estimate phase directly from intensity measurements. While several alternative techniques exist, each comes with its own advantages and limitations:

- **Gerchberg-Saxton Algorithm (GSA) and Hybrid Input-Output (HIO):** These iterative algorithms refine phase information using Fourier constraints but often suffer from **twin-image ambiguity** and slow convergence, particularly in the presence of noise [3].
- **Multidither Technique:** This method applies high-frequency phase modulations to optimize an intensity-based metric. However, it requires **bright reference sources**, making it impractical for faint astronomical objects [17].
- **Shack-Hartmann Wavefront Sensing:** A widely used technique that measures local wavefront slopes using a microlens array. As discussed by [18], each microlens focuses a portion of the incoming wavefront onto a detector, forming a grid of focal spots. By analyzing the displacement of these spots relative to their expected positions in an undistorted wavefront, the local wavefront gradients can be computed.

  However, Shack-Hartmann sensing only provides **wavefront slopes**, not the absolute phase. Recovering the full wavefront requires an additional **phase reconstruction step**, typically using numerical integration or least-squares fitting. This reconstruction process can introduce artifacts due to noise, missing data, or computational approximations. Additionally, Shack-Hartmann sensors struggle in **high-contrast imaging scenarios**, such as exoplanet detection, because they are sensitive to scattered light and may not resolve extremely fine wavefront aberrations needed for deep contrast suppression [14].

Among these, **phase diversity stands out** due to its ability to directly retrieve phase from intensity images without requiring an external reference beam or additional hardware. This makes it highly suitable for **real-time adaptive optics (AO) applications** and space-based telescopes, where precision wavefront control is essential.

High-precision phase retrieval techniques are particularly crucial in applications demanding **extreme contrast sensitivity**, such as **exoplanet imaging**. The challenge lies in detecting faint planetary signals buried within the overwhelming glare of a host star. To overcome this, **phase retrieval is integrated with coronagraphy and adaptive optics**, enabling direct imaging of exoplanets. The following section explores how these advanced wavefront sensing methods enhance our ability to detect and characterize exoplanetary atmospheres, bringing us closer to identifying potentially habitable worlds.

7.2.5. Exoplanet Imaging and Coronagraphy

One of the most challenging problems in observational astronomy is the direct detection and characterization of exoplanets. Since exoplanets are orders of magnitude fainter than their host stars, traditional imaging techniques struggle due to overwhelming stellar glare. Direct imaging requires advanced wavefront control methods to suppress scattered starlight and enhance planetary signals. Some of the key techniques include:

- **Coronagraphy:** A coronagraph is an optical instrument designed to block out the intense light from a star while allowing light from nearby exoplanets to be detected. This is achieved through a combination of an opaque mask at the focal plane and apodized pupil designs that reduce diffraction effects. By suppressing stellar glare, coronagraphs significantly improve contrast, making it possible to detect faint exoplanets that would otherwise remain hidden.

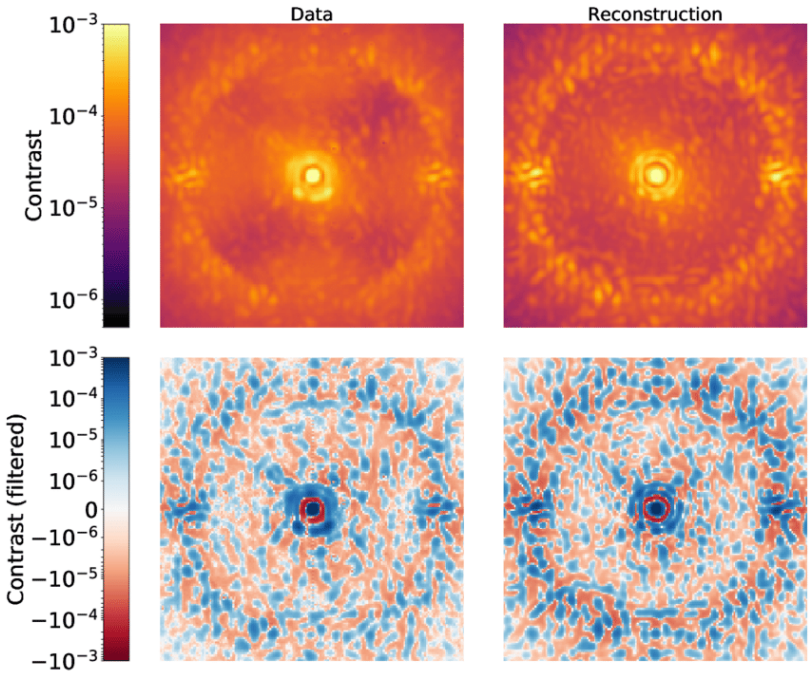


Figure 23: Comparison of the coronagraphic data obtained on sky on 2018-04-03 (left) with the reconstructed coronagraphic image (right). Top row: complete data and model, while bottom row: high-passfiltered version of the images to remove the low-spatial frequencies corresponding to the averaged ExAO residuals. Credit: A. Vigan et al., Calibration of quasi-static aberrations in exoplanet direct-imaging instruments with a Zernike phase-mask sensor. III. On-sky validation in VLT/SPHERE.

- **Phase Diversity with High-Contrast Imaging:** High-contrast imaging (HCI) relies on precise wavefront control to minimize residual optical aberrations. One method to achieve this is phase diversity (PD), which involves capturing multiple images with intentional defocus to infer both amplitude and phase variations in the wavefront. By iteratively correcting these aberrations, PD helps suppress residual stellar speckles—unwanted light patterns that mimic exoplanet signals—thereby enhancing the detectability of faint planetary companions.
- **Hybrid Wavefront Sensing:** Traditional wavefront sensing techniques, such as Shack-Hartmann sensors, often fall short in achieving the extreme precision required for exoplanet imaging. Hybrid wavefront sensing combines phase diversity with electric field conjugation (EFC), an iterative process that actively refines phase estimates to further suppress residual speckles. EFC optimizes the wavefront correction in the focal plane, allowing for deeper suppression of stellar noise and improved contrast ratios necessary for detecting exoplanets.

Space-based missions such as the *Nancy Grace Roman Space Telescope* and ground-based observatories equipped with adaptive optics (e.g., the Extremely Large Telescope) leverage these wavefront control methods to push the limits of exoplanet imaging. By integrating phase diversity with advanced coronagraphs and real-time correction algorithms, these instruments aim to capture direct images of exoplanetary atmospheres, providing information about their composition, weather patterns, and potential habitability. Success in exoplanet imaging relies on both theoretical advancements and technological innovations. As instrumentation advances, these methods will continue to refine observational capabilities, bringing us closer to definitely characterizing exoplanetary atmospheres and identifying potential signs of habitability beyond our solar system [19][20].

7.2.6. Beyond Astronomy: Broader Applications of Phase Diversity

Phase diversity significantly enhances **phase retrieval accuracy** by introducing intentional phase modulations, resolving ambiguities, and improving convergence. Originally developed for astronomical imaging, phase diversity (PD) has since found applications in various fields:

- **Adaptive Optics (AO):** Enhances ground-based astronomy by compensating for **atmospheric turbulence in real time**.
- **Microscopy & Biomedical Imaging:** Used in **fluorescence microscopy and optical coherence tomography** to compensate for aberrations in biological samples.
- **Surveillance & Defense Systems:** Improves imaging through turbulent atmospheres, crucial for **reconnaissance and tracking**.
- **X-ray & Electron Microscopy:** Enables **high-resolution phase contrast imaging** in materials science.

As phase retrieval techniques continue to evolve, their impact extends far beyond astronomy, influencing fields such as biomedical imaging and even defense technologies.

7.2.7. Future Perspectives

Advancements in phase retrieval methods continue to impact astronomical imaging, enabling higher precision wavefront sensing for upcoming telescopes such as the Thirty Meter Telescope (TMT) and the European Extremely Large Telescope (E-ELT). The development of machine learning-based phase retrieval algorithms is also a promising direction, allowing for faster and more accurate phase estimation.

7.3. Modern Developments: Phase Retrieval Through Machine Learning

Phase retrieval is the process of reconstructing a complex-valued signal from intensity-only measurements, a fundamental problem in computational imaging. Traditional approaches to phase retrieval include gradient-based optimization, spectral methods, and iterative algorithms such as the Hybrid Input-Output (HIO) method. However, despite decades of research, phase retrieval remains computationally intensive and highly sensitive to noise, limiting its effectiveness in real-world scenarios.

Recent advances in *machine learning* (ML) have significantly impacted phase retrieval by improving both theoretical understanding and practical performance. Machine learning approaches have contributed in two key ways:

(a) **Theoretical insights**: Researchers have drawn analogies between phase retrieval and single-layer neural networks, leading to a deeper mathematical understanding of the problem.

(b) **Practical breakthroughs**: Deep learning-based techniques have demonstrated superior accuracy and robustness in reconstructing phase information, even in challenging imaging conditions.

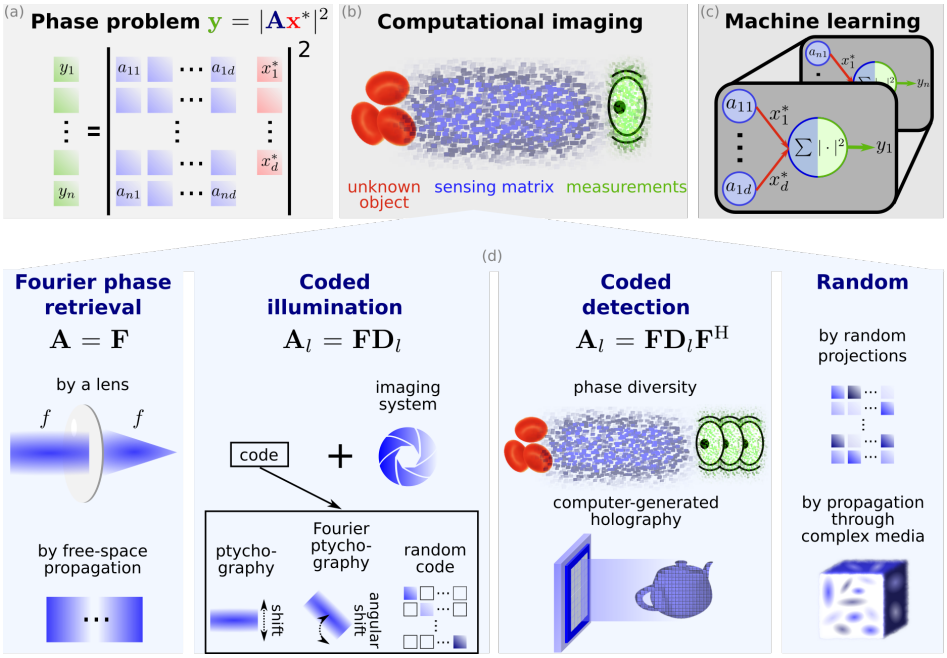


Figure 24: Illustration of various phase retrieval techniques and their connections to computational imaging and machine learning. The phase retrieval problem (a) involves recovering the lost phase information from intensity-only measurements. Computational imaging (b) employs sensing matrices and measurement processes to reconstruct unknown objects. Machine learning approaches (c) model phase retrieval as a nonlinear transformation, learning mappings between intensity and phase. Different retrieval strategies (d) include Fourier phase retrieval, coded illumination, coded detection, and random projections, each offering unique advantages for improving phase recovery. Credit: Jonathan Dong et al., (arXiv:2204.03554v2 [physics.optics] 14 Nov 2022)

Phase retrieval is a fundamental problem in computational imaging, where the goal is to reconstruct an object's complex-valued wavefront from intensity-only measurements. The top-left section (a) of the figure 24 mathematically represents this problem, showing how the intensity data alone is insufficient for direct phase recovery. To address this, computational imaging methods (b) employ sensing matrices and measurement models to extract meaningful information about the object. The challenge of phase retrieval has also attracted interest from machine learning (c), where neural networks and other algorithms attempt to approximate phase information by learning from data distributions. The bottom section (d) categorizes different phase retrieval approaches: Fourier phase retrieval relies on far-field propagation constraints; coded illumination techniques such as ptychography introduce structured illumination patterns to enhance reconstruction; coded detection methods leverage phase diversity and holography for improved measurement diversity; and random projection techniques exploit scattering media or random phase encoding to impose additional constraints on the reconstruction process. By integrating physics-based models with computational and machine learning-driven techniques, phase retrieval continues to evolve as a crucial tool for high-resolution imaging in fields such as astronomy, microscopy, and optical metrology.

7.3.1. Mathematical Formulation: Fourier Phase Retrieval

Phase retrieval can be formulated as a nonlinear inverse problem. Given an unknown complex signal $x^* \in C^d$, the measurement process can be written as:

$$y = |Ax^*|^2$$

where:

- $y \in R^n$ represents the intensity measurements obtained from the detector,

- $A \in C^{n \times d}$ is a known measurement matrix, which models how the signal is transformed before measurement,
- $|\cdot|$ denotes the element-wise modulus operator, which removes the phase information.

Unlike standard linear systems of the form $y = Ax^*$, the missing phase information in the above equation makes the problem significantly more challenging, as it introduces nonlinearity [21].

A particularly important case is **Fourier phase retrieval**, where the measurement operator is given by the Fourier transform matrix F. Since the Fourier transform is **unitary** (i.e., $F^H = F^{-1}$, where F^H denotes the Hermitian transpose), we can efficiently backpropagate information in learning-based phase retrieval methods and the equation remains structurally similar. In phase retrieval, this property is crucial because the Fourier transform does not amplify or distort energy — it merely redistributes it.

In Fourier phase retrieval, the measurements are taken in the frequency domain, and the challenge lies in recovering the lost phase.

Fourier phase retrieval has found applications in:

- **Speckle correlation imaging**, used for imaging through complex media.
- **Non-line-of-sight imaging**, which reconstructs hidden objects from scattered light.

7.3.2. Neural Networks for Phase Retrieval

A neural network is a computational model inspired by the structure of the human brain, consisting of layers of interconnected neurons. Each neuron applies a mathematical transformation to its inputs and passes the result to the next layer.

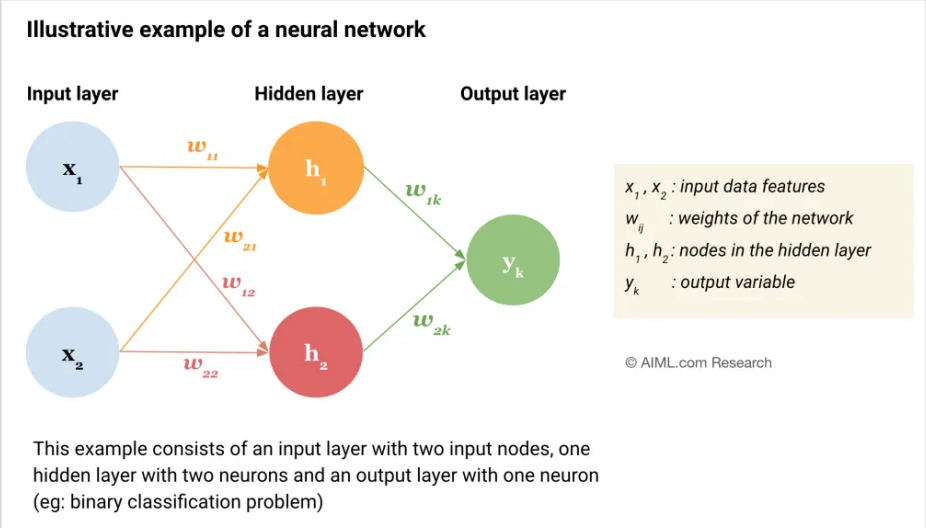


Figure 25: Basic structure of a neural network with an input layer, hidden layers, and an output layer.

For phase retrieval, neural networks typically consist of:

- **Input Layer:** Receives the intensity measurements from the detector.
- **Hidden Layers:** Composed of multiple fully connected layers, convolutional layers (in CNNs), or self-attention mechanisms (in transformers).
- **Output Layer:** Produces an estimate of the recovered phase distribution.

The network is trained using a loss function that compares the reconstructed phase with ground truth data, adjusting the weights of the network through backpropagation. A commonly used loss functions is:

$$\mathcal{L} = ||\hat{x} - x^*||^2$$

where $\hat{x}$ is the predicted phase, and x^* is the true phase [21].

7.3.3. Learning in Neural Networks and Its Relevance to Phase Retrieval

Neural networks learn by adjusting their internal parameters (weights and biases) to minimize the error in predictions. Each neuron computes a weighted sum of its inputs:

$$z_j^l = \sum_i w_{ij}^l a_i^{l-1} + b_j^l$$

where, a_i^{l-1} are the activations (outputs) from the previous layer, w_{ij}^l are the weights connecting neurons, and b_j^l is the bias term.

In phase retrieval, the mapping from intensity measurements to phase is highly non-linear. Traditional methods like the Hybrid Input-Output (HIO) algorithm iteratively enforce constraints in Fourier space and real space, which inherently involves non-linearity. Neural networks, by incorporating activation functions, can better approximate such mappings, improving phase reconstruction, especially in cases where iterative algorithms struggle due to noise or missing data. The activation function σ introduces non-linearity as:

$$a_j^l = \sigma(z_j^l)$$

which represents the activation of neuron j in layer l of a neural network.

The network is trained using **back propagation**, which computes the gradient of the loss function with respect to each weight, and **gradient descent**, which updates the weights in the direction that reduces the error. This allows the network to progressively improve its phase predictions [22].

In the context of phase retrieval, neural networks serve as function approximators that learn to map intensity measurements to phase distributions. The ability to capture complex, non-linear mappings makes them particularly effective in overcoming limitations of traditional iterative methods. By leveraging large datasets, neural networks can generalize well to unseen data, making them a powerful tool for computational imaging applications.

7.3.4. Deep Learning-Based Phase Retrieval

Deep learning, particularly neural networks, has shown significant potential in solving the phase retrieval problem by learning complex mappings from intensity measurements to their corresponding phase distributions. Unlike traditional iterative phase retrieval methods, which rely on physical models and optimization techniques, deep learning-based approaches can leverage large datasets and learn data-driven priors to improve accuracy, robustness, and computational efficiency. Some notable approaches include [21][22] :

Supervised Learning Approaches:

- **Convolutional Neural Networks (CNNs)**: CNNs are trained on large datasets of intensity-phase pairs to learn the direct mapping between input intensity measurements and the corresponding phase information.
- **Generative Adversarial Networks (GANs)**: GAN-based methods generate high-quality phase reconstructions by learning the statistical distribution of phases from observed data, often producing more realistic results than traditional techniques.

Physics-Informed Neural Networks (PINNs) These models integrate physical laws—such as wave propagation equations—directly into neural network architectures. By embedding known physics, PINNs improve generalization and reduce dependence on large labeled datasets. Unlike purely data-driven models, PINNs integrate physical laws—such as wave propagation and diffraction equations—directly into the learning process. By embedding these constraints into the neural network architecture or loss function, PINNs enhance generalization across different imaging conditions and improve robustness to noise and experimental variations.

Transformer-Based Models Originally developed for natural language processing, *transformers* are now being explored for phase retrieval. Their self-attention mechanisms enable **global feature extraction**, making them effective in processing complex imaging data. Inspired by their success in natural language processing, transformer models are being adapted for phase retrieval. These architectures leverage self-attention mechanisms to capture global dependencies in intensity measurements, making them particularly effective in scenarios where long-range correlations are important for phase reconstruction.

Unsupervised and Self-Supervised Learning Unsupervised and self-supervised learning approaches have emerged as powerful tools for phase retrieval, particularly in scenarios where obtaining large labeled datasets is challenging or impractical. These methods allow models to learn meaningful representations of phase information without explicit supervision, making them well-suited for real-world applications where ground truth phase data is often unavailable.

- **Autoencoders**: These networks learn a compact, low-dimensional representation of phase distributions without requiring labeled data. These networks consist of an encoder-decoder architecture that compresses phase distributions into a lower-dimensional latent space before reconstructing them. By training on intensity-only measurements, autoencoders can capture underlying structures and statistical properties of phase data. Variational autoencoders (VAEs) and denoising autoencoders (DAEs) have been particularly useful in improving reconstruction quality and robustness to noise.
- **Self-Supervised Learning**: Instead of relying on explicit phase labels, self-supervised learning techniques leverage inherent patterns and relationships in the data to train models. Methods such as contrastive learning and consistency-based objectives allow networks to infer phase information by distinguishing between different intensity patterns or learning invariant representations across multiple measurements. These techniques improve generalization across diverse imaging conditions and can enhance the adaptability of phase retrieval models to experimental imperfections.

7.3.5. Hybrid Classical-ML Approaches

To combine the strengths of both classical and deep learning based techniques, hybrid approaches have been developed. These typically follow a two-step process:

- A classical iterative phase retrieval algorithm, such as the Hybrid Input-Output (HIO) or Gerchberg-Saxton (GS) algorithm, produces an initial estimate of the phase. These algorithms are grounded in well-established physical principles and provide a structured approach to phase recovery. However, they often suffer from stagnation at local minima, sensitivity to noise, and the requirement for carefully tuned initialization.
- A machine learning model then refines this phase estimate, correcting errors and improving robustness against noise. By learning from data distributions, the neural network corrects systematic errors introduced by the iterative method, enhances convergence speed, and improves robustness to experimental noise and incomplete measurements. This step can be implemented using convolutional neural networks (CNNs), autoencoders, or generative models trained to recognize and refine realistic phase distributions. [21]

7.3.6. Future Challenges and Research Directions

Despite its rapid progress, ML-based phase retrieval still faces several challenges:

- **Data Efficiency:** Reducing reliance on large labeled datasets through self-supervised and unsupervised learning.
- **Robustness:** Ensuring that phase retrieval models perform well in the presence of noise, aberrations, and experimental imperfections.
- **Generalization:** Developing models that can adapt to different imaging conditions, wavelengths, and optical setups.

Machine learning is rapidly transforming phase retrieval, enabling faster and more accurate reconstructions. As computational power continues to increase and new learning paradigms emerge, ML-driven phase retrieval is poised to become an essential tool in high-resolution imaging, adaptive optics, and next-generation telescopes.

8. Limitations of Phase Retrieval

Algorithmic phase retrieval has emerged as a powerful tool for high-resolution imaging, overcoming the limitations of conventional optical systems that rely on lenses and direct phase-sensitive measurements. However, it is not without its challenges. One fundamental limitation arises from the **ill-posed nature** of the inverse problem. Since phase information is lost in intensity measurements, retrieving it requires additional constraints, assumptions, or iterative methods, which may not always converge to the correct solution. The presence of multiple possible solutions (ambiguity) and sensitivity to initialization make the reconstruction process non-trivial.

Another challenge is **measurement noise and system instability**. Real-world imaging systems are subject to photon shot noise, detector readout noise, and environmental fluctuations, all of which can degrade the accuracy of retrieved phase information. Small errors in intensity measurements can lead to significant phase reconstruction artifacts, especially in high-resolution or low-light scenarios. Moreover, as **spatial resolution increases**, the phase retrieval process becomes increasingly demanding. In imaging, resolution is often characterized by the voxel size, where a **voxel** (short for "volumetric pixel") represents the smallest distinguishable unit in a three-dimensional image. As voxel sizes decrease, the number of photons per unit area must increase to maintain a sufficient signal-to-noise ratio (SNR). This follows a scaling law where the required exposure time increases as $1/d^4$, with d being the resolution length. This poses practical limitations in applications requiring fast imaging, such as live biological imaging or real-time adaptive optics. In extreme cases, prolonged exposure times can lead to motion artifacts, reducing the reliability of phase retrieval techniques in dynamic environments [12].

Despite these challenges, phase retrieval techniques continue to play a crucial role in modern imaging science. Advances in computational algorithms, machine learning, and hardware-assisted phase retrieval have significantly improved robustness and efficiency. From **super-resolution microscopy** and **X-ray crystallography** to **adaptive optics** and **astronomical imaging**, iterative phase retrieval methods have enabled groundbreaking advances in fields that rely on high-precision wavefront reconstruction.

A phase retrieval algorithm usually involves a non-linear inverse problem which is generally computationally complex and must be solved iteratively. Therefore, in order to obtain a real-time phase estimation based on non-linear phase retrieval, a high performance computing system has to be employed to decrease the computational time required for the phase aberration estimation. When such hardware is not available, proper approximations of the phase retrieval model and efficient control algorithms needs to be developed in order to obtain a satisfying performance on a simple hardware bench. Ongoing research aims to address these limitations by integrating physics-informed priors, deep learning-based reconstructions, and hybrid classical-ML approaches to further enhance accuracy and computational efficiency [10] [23].

References

[1] K.F. Riley, M.P. Hobson, and S.j. Bence. Mathematical Methods for Physics and Engineering. Cambridge University Press, 3rd edition, 2006.

[2] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex Fourier series. Mathematics of computation, 19(90):297–301, 1965.

[3] Ralph W Gerchberg. A practical algorithm for the determination of plane from image and diffraction pictures. Optik, 35(2):237–246, 1972.

[4] James R Fienup. Reconstruction of an object from the modulus of its Fourier transform. Optics letters, 3(1):27–29, 1978.

[5] J.R. Fienup. Phase retrieval algorithms: a comparison. Applied Optics, 21(15):2758–2769, 1982.

[6] Muskan Kularia, Manidipa Banerjee, and Kedar Khare. Twin-stagnation-free phase retrieval with vortex phase illumination. JOSA A, 41(6):1166–1174, 2024.

[7] Joseph W. Goodman. Introduction to Fourier Optics. McGraw-Hill, 3rd edition, 2005.

[8] Yoav Shechtman, Yonina C Eldar, Oren Cohen, Henry Nicholas Chapman, Jianwei Miao, and Mordechai Segev. Phase retrieval with application to optical imaging: a contemporary overview. IEEE signal processing magazine, 32(3):87–109, 2015.

[9] R. A. Gonsalves and R. Chidlaw. Wavefront sensing by phase retrieval. In A. G. Tescher, editor, Applications of Digital Image Processing III, volume 207 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, pages 32–39, December 1979.

[10] J. R. Fienup. Phase retrieval algorithms: a comparison. Applied Optics, 21(15):2758–2769, 1982.

[11] Richard M. Clare. Wavefront Sensing and Phase Retrieval for Astronomical Imaging. PhD thesis, University of Canterbury, Christchurch, New Zealand, September 2004. Doctoral Thesis, Department of Electrical and Computer Engineering.

[12] Robert A. Gonsalves. Phase retrieval and diversity in adaptive optics. Optical Engineering, 21(5):215829, October 1982.

[13] Claire Max. Adaptive optics for astronomy: Principles, performance, and applications. Science Direct, 2000.

[14] F. Roddier. Adaptive Optics in Astronomy. Cambridge University Press, 1999.

[15] Bruce H. Dean, David L. Aronstein, J. Scott Smith, Ron Shiri, and D. Scott Acton. Phase retrieval algorithm for JWST Flight and Testbed Telescope. In John C. Mather, Howard A. MacEwen, and Mattheus W. M. de Graauw, editors, Space Telescopes and Instrumentation I: Optical, Infrared, and Millimeter, volume 6265 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, page 626511, June 2006.

[16] Robert A Gonsalves. Perspectives on phase retrieval and phase diversity in astronomy. In Adaptive Optics Systems IV, volume 9148, pages 942–951. SPIE, 2014.

[17] R. G. Paxman, T. J. Schulz, and J. R. Fienup. Joint estimation of object and aberrations by using phase diversity. JOSA A, 9(7):1072–1085, 1992.

[18] B. C. Platt and R. Shack. History and principles of Shack-Hartmann wavefront sensing. Journal of Refractive Surgery, 17(5):S573–S577, 2001.

[19] A. Vigan, M. N'Diaye, K. Dohlen, J.-F. Sauvage, J. Milli, G. Zins, C. Petit, Z. Wahhaj, F. Cantalloube, A. Caillat, A. Costille, J. Le Merrer, A. Carlotti, J.-L. Beuzit, and D. Mouillet. Calibration of quasistatic aberrations in exoplanet direct-imaging instruments with a zernike phase-mask sensor. iii. on-sky validation in vlt/sphere. Astronomy Astrophysics, 629:A11, 2019.

[20] Michael Perryman. The Exoplanet Handbook. Cambridge University Press, Cambridge, UK, 2nd edition, 2018.

[21] Jonathan Dong, Lorenzo Valzania, Antoine Maillard, Thanh-an Pham, Sylvain Gigan, and Michael Unser. Phase retrieval: From computational imaging to machine learning: A tutorial. IEEE Signal Processing Magazine, 40(1):45–57, January 2023.

[22] Rujia Li, Giancarlo Pedrini, Zhengzhong Huang, Stephan Reichelt, and Liangcai Cao. Physics enhanced neural network for phase retrieval from two diffraction patterns. Optics Express, 30(18):32680–32692, 2022.

[23] Polo, A Haber, SF Pereira, M Verhaegen, and HP Urbach. Linear phase retrieval for real-time adaptive optics. Journal of the European Optical Society-Rapid publications, 8:13070, 2013.

APPENDIX: HIO Algorithm for Phase Retrieval

Introduction

This notebook implements the **Hybrid Input-Output (HIO) Algorithm** for iterative phase retrieval. The goal is to reconstruct an image using only its Fourier magnitude while imposing real-space constraints. This approach is widely used in **optical imaging, X-ray crystallography, and astronomy**.

The algorithm works as follows:

- Compute the **Fourier transform** of a known image.
- Retain only the **magnitude information** and discard the phase.
- Use an iterative approach to **recover the phase** by enforcing constraints in both Fourier and spatial domains.

This method is particularly useful in **lensless imaging** applications where the phase of the optical wavefront is lost.

In [205…

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from numpy.fft import fft2, ifft2
import os
```

Loading and Preprocessing the Image

A test image is loaded, converted to grayscale, and **zero-padded** to simulate an extended support region. This ensures that the Fourier transform does not introduce artificial boundaries that affect phase retrieval.

In [207…

```
# Read in source image
source = Image.open(r'C:\Users\asus\Desktop\Image manipulation\Lena.png')
         .convert("L")

# Get image dimensions
width, height = source.size

# Image is padded to simulate oversampling
pad_len = max(width, height)
padded = np.pad(np.array(source), ((pad_len, pad_len), (pad_len, pad_len))
                , 'constant')
```

Computing Fourier Magnitude

The **Fourier Transform (FT)** of the padded image is computed using the Fast Fourier Transform (FFT). Since phase information is missing in certain imaging techniques, we retain only the **Fourier magnitude** (absolute values of the FFT).

In [209…
```
ft = fft2(padded)

# Simulate diffraction pattern
diffract = np.abs(ft)
```

Defining the Image Region Mask

A **binary mask** is created to differentiate the object region from the background in the reconstructed image. This mask will be used to apply constraints during iterative updates.

In [211…
```
l = len(padded)

# Create mask for image region
mask = np.zeros_like(padded, dtype=bool)
mask[pad_len:pad_len + height, pad_len:pad_len + width] = True
```

Initializing with a Random Phase

Since phase information is unavailable, we generate an **initial guess** by combining the measured Fourier magnitude with a **random phase distribution**.

In [213…
```
# Initial guess using random phase
guess = diffract * np.exp(1j * np.random.rand(l, l) * 2 * np.pi)
```

Algorithm Implementation

The reconstruction process follows an iterative approach, where an initial estimate undergoes continuous refinement over **501 iterations**. The steps involved are detailed below:

1. Initialization

- The variable `prev` is initialized as a zero-valued array of the same shape as the padded image.
- The number of iterations, `r`, is set to **501**.
- A step-size parameter `β = 0.8` is chosen to control the update rate.

2. Iterative Phase Retrieval Process

At each iteration, the algorithm applies constraints in both **Fourier space** and **real space** to refine the image estimate.

Fourier Domain Constraint:

- The Fourier amplitude of the estimate is replaced with the known diffraction amplitude (`diffract`), while preserving the phase from the previous iteration. This results in a new estimate:

  $$\text{update} = \text{diffract} \cdot e^{i \cdot \theta}$$

  where θ is the phase from the previous estimate.

- The inverse Fourier transform (`ifft2(update)`) is computed to bring the estimate back into the real domain.

Real-Space Constraint:

- The recovered real-space image is refined using **non-negativity and support constraints** as follows:
  - Pixels within the object region (`mask`) that have **negative values** are modified using:

    $$\text{inv}[\text{neg\_mask}] = \text{prev}[\text{neg\_mask}] - \beta \cdot \text{inv}[\text{neg\_mask}]$$

This ensures that the recovered image respects non-negativity constraints, which is essential in phase retrieval problems where intensity values should be positive or zero.

- Pixels **outside the object region** are similarly updated:

  $$\text{inv}[\text{support\_mask}] = \text{prev}[\text{support\_mask}] - \beta \cdot \text{inv}[\text{support\_mask}]$$

The term $\beta \cdot \text{inv}[\text{support\_mask}]$ acts as a damping factor, reducing the magnitude of values outside the support. Over iterations, this gradually pushes unwanted artifacts toward zero. This step thus improves reconstruction accuracy by ensuring that the algorithm does not introduce artifacts where the object should not exist.

These steps ensures that the reconstructed image satisfies both **non-negativity** and **support constraints**.

Iteration Update:

- The updated image replaces the previous estimate (`prev = inv.copy()`).
- The **Fourier transform** (`fft2(inv)`) is computed to update the phase information.

3. Saving Progress

To visualize the reconstruction process, the algorithm saves the intermediate results every **100 iterations**. The saved images are normalized to grayscale (0–255) and stored in `save_dir`.

$$\text{Image} = \frac{\text{prev} - \min(\text{prev})}{\max(\text{prev}) - \min(\text{prev})} \times 255$$

These saved images help track how the reconstruction improves over time.

In [215…

```
save_dir = r"C:\Users\asus\Desktop\Image manipulation\save5"
os.makedirs(save_dir, exist_ok=True)

# Number of iterations
r = 501

# Step size parameter
beta = 0.8

# Previous result
prev = np.zeros_like(padded, dtype=float)

for s in range(r):
    # Fourier domain constraints
    update = diffract * np.exp(1j * np.angle(guess))
    inv = np.real(ifft2(update))

    # Apply real-space constraints using vectorized operations
    neg_mask = (inv < 0) & mask  # Negative values in the image region
    support_mask = ~mask         # Outside the image region

    inv[neg_mask] = prev[neg_mask] - beta * inv[neg_mask]
    inv[support_mask] = prev[support_mask] - beta * inv[support_mask]

    prev = inv.copy()
    guess = fft2(inv)

    # Save progress every 100 iterations
    if s % 100 == 0:
        Image.fromarray((prev - prev.min()) / (prev.max() - prev.min()) * 255)
             .convert("L").save(os.path.join(save_dir, f"progress{s}.png"))
        print(f"Iteration {s} saved.")
```

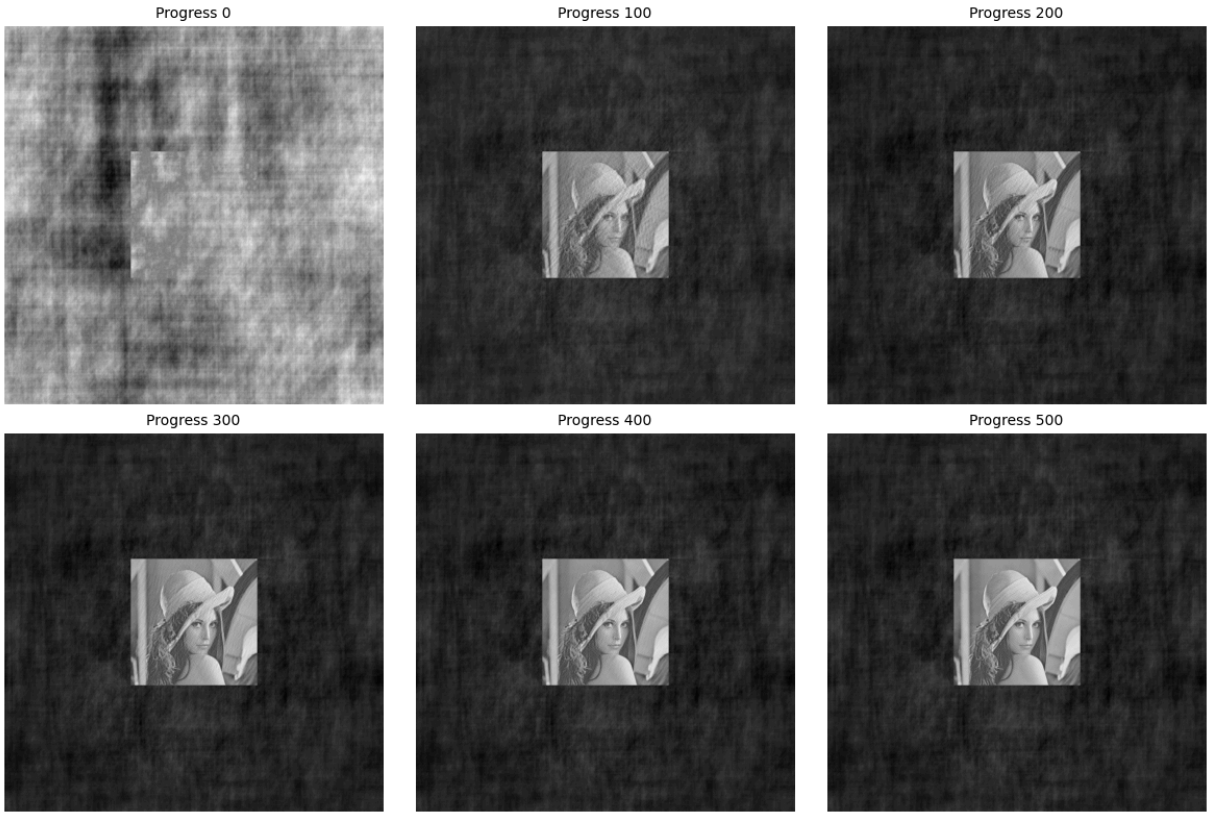
```
Iteration 0 saved.
Iteration 100 saved.
Iteration 200 saved.
Iteration 300 saved.
Iteration 400 saved.
Iteration 500 saved.
```

The saved images are shown below for reference

In [217…

```
# List of progress image filenames
image_indices = range(0, 501, 100)  # 0, 100, 200, ..., 500
```

```python
# Define grid dimensions (2 rows, 3 columns)
fig, axes = plt.subplots(2, 3, figsize=(12, 8))

for ax, i in zip(axes.flat, image_indices): #axes are flattened for easy iterations
    img_path = os.path.join(save_dir, f"progress{i}.png")
    if os.path.exists(img_path):
        img = mpimg.imread(img_path)
        ax.imshow(img, cmap='gray')
        ax.set_title(f"Progress {i}", fontsize=10)
        ax.axis("off")  # looks good when off
    else:
        ax.axis("off")  # This hides empty plots if file is missing

plt.tight_layout()
plt.show()
```



Computing the Difference Image

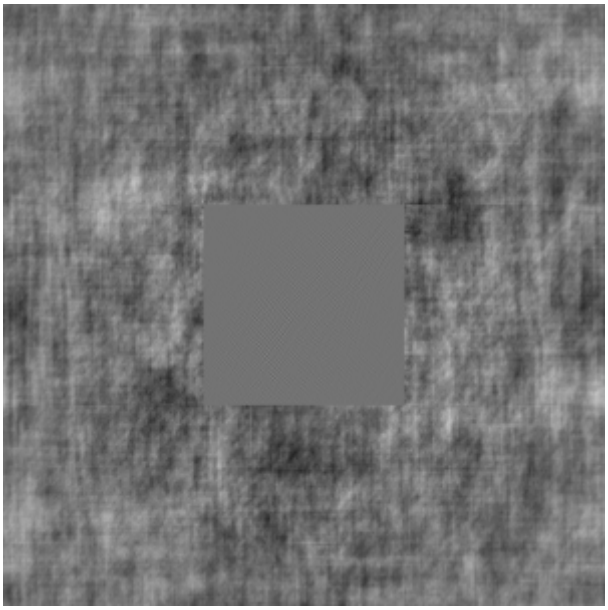
To assess reconstruction accuracy, the **difference between the recovered image and the original image** is computed and saved.

In [219…
```python
# Compute difference between recovered image and padded original
difference = padded - prev

# Save difference image
Image.fromarray((difference - difference.min()) / (difference.max()
                - difference.min()) * 255).convert("L")
                 .save(os.path.join(save_dir, "difference.png"))
print("Difference image saved.")
```

```
Difference image saved.
```

The difference image is added too for reference.

In [269…
```python
save_dir = r"C:\Users\asus\Desktop\Image manipulation\save5"
# Load the saved difference image
diff_image_path = os.path.join(save_dir, "difference.png")
diff_image = Image.open(diff_image_path)

display(diff_image.resize((300, 300)))
```



Analysis of the Difference Image

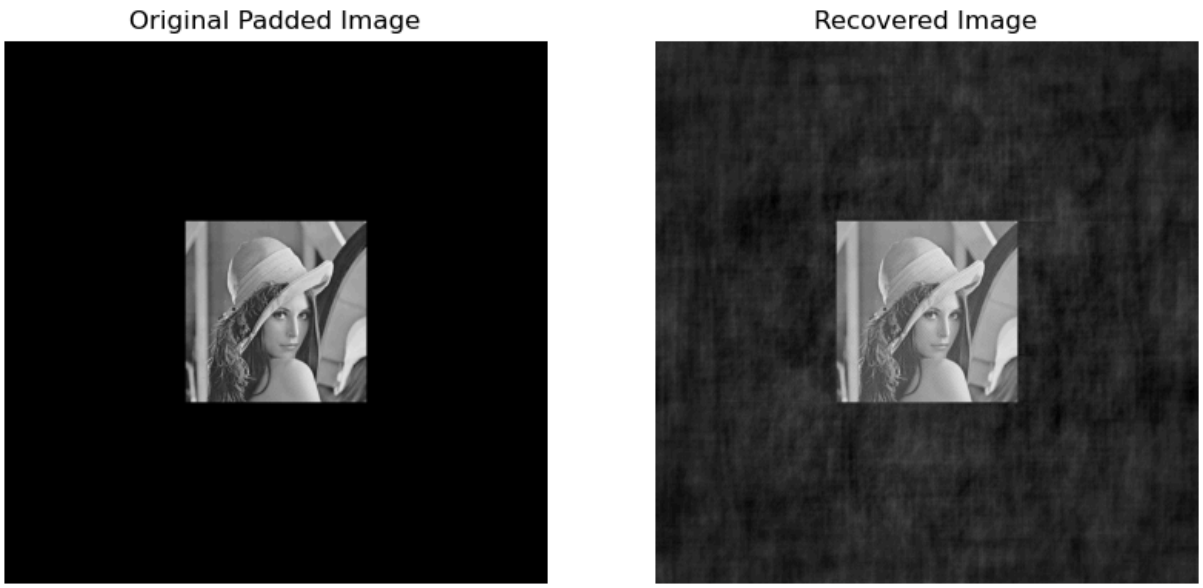
The difference image represents the residual error between the original (padded) image and the reconstructed image obtained using the algorithm. The central region retains structural information, indicating that the algorithm successfully recovered major features of the object. Some high-frequency details and edges are visible, suggesting that the reconstruction preserves key elements despite phase ambiguity. The surrounding noise-like artifacts mainly appear in the padded regions, which do not contribute to the actual object reconstruction. Overall, the phase retrieval process demonstrates effective image recovery, with potential for further refinement through additional iterations or even better constraints.

Visualization of Results

The **original padded image** and the **recovered image** are displayed side by side for comparison.

In [224…

```
plane_reconstruction = prev.copy()  # Save result for comparison later
# Plot original and final images side by side
fig, ax = plt.subplots(1, 2, figsize=(10, 5))
ax[0].imshow(padded, cmap='gray')
ax[0].set_title("Original Padded Image")
ax[0].axis("off")

prev_copy = prev.copy()
prev_normalized = (prev_copy - prev_copy.min()) / (prev_copy.max()
                                                   - prev_copy.min()) * 255
ax[1].imshow(prev_normalized.astype(np.uint8), cmap='gray')
ax[1].set_title("Recovered Image")
ax[1].axis("off")
plt.show()
```



Thus, our algorithm is able to reconstruct the original image with all significant features identifiable, within a few hundred iterations.

To have a better look at the reconstructuion, let's remove the padding and get the images side by side.

In [226…

```
# Plot original and reconstructed images side by side (without padding)
fig, ax = plt.subplots(1, 2, figsize=(10, 5))

# Show the original image
ax[0].imshow(source, cmap='gray')
ax[0].set_title("Original Image")
ax[0].axis("off")

# Crop the reconstructed image to match the original dimensions
recovered_cropped = prev[pad_len:pad_len + height, pad_len:pad_len + width]
recovered_normalized = (recovered_cropped - recovered_cropped.min()) /
```

```
                    (recovered_cropped.max() - recovered_cropped.min()) * 255

# Show the reconstructed image without padding
ax[1].imshow(recovered_normalized.astype(np.uint8), cmap='gray')
ax[1].set_title("Recovered Image (after 500 iterations)")
ax[1].axis("off")

plt.show()
```

The **Hybrid Input-Output (HIO) Algorithm** effectively reconstructs an image by iteratively refining the phase information. Despite some artifacts, the recovered image closely resembles the original, within a few hundred iterations.

This method is widely used in applications such as:

- **Lensless imaging and holography**
- **X-ray crystallography**
- **Astronomical imaging (e.g., wavefront sensing)**

Further improvements can be made by using **alternative phase retrieval algorithms**, or incorporating **deep learning techniques** for faster convergence. One method to mitigate stagnation problems like twin image is to use an different illumination for the object at hand. Here, we discuss votex illumination and show how it reconstructs the image more efficiently in fewer number of iterations comapred to plane illumination, in addition to mitigating stagnation effects.

Vortex Illumination Reconstruction Using HIO Algorithm

In this section, we implement **iterative phase retrieval** using the **Hybrid Input-Output (HIO) algorithm**.

Instead of **plane illumination**, we now use a **vortex phase illumination** defined as:

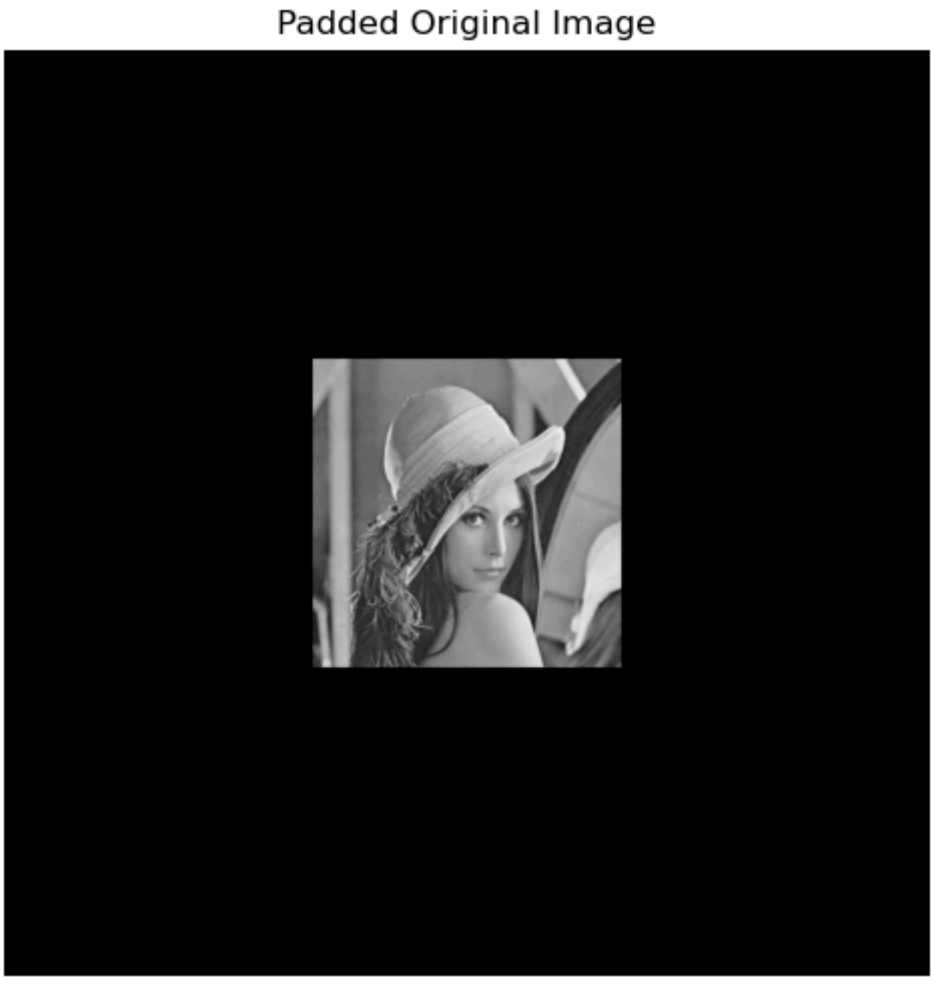
$$\theta(x, y) = \arctan\left(\frac{y}{x}\right)$$

This introduces a **spatial phase variation** that alters the diffraction pattern, making phase retrieval more complex, however more rewarding. For this, we reconstruct the image and then analyze the difference between the original and retrieved images.

For this, the image is first loaded and padded and a save directory is created, as before.

In [229…

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from numpy.fft import fft2, ifft2
import os
# Read source image
source = Image.open(r'C:\Users\asus\Desktop\Image manipulation\Lena.png')
              .convert("L")

# Get image dimensions
width, height = source.size

# Pad image to simulate oversampling
pad_len = max(width, height)
padded = np.pad(np.array(source), ((pad_len, pad_len), (pad_len, pad_len))
                , 'constant')

# Create mask for image region
mask = np.zeros_like(padded, dtype=bool)
mask[pad_len:pad_len + height, pad_len:pad_len + width] = True

# Create directory for saving results
save_dir = r"C:\Users\asus\Desktop\Image manipulation\save_vortex1"
os.makedirs(save_dir, exist_ok=True)

plt.figure(figsize=(6, 6))
plt.imshow(padded, cmap='gray')
plt.title("Padded Original Image")
plt.axis("off")
plt.show()
```



Vortex Phase Pattern

We apply **vortex illumination** by introducing a **phase mask** using the equation:

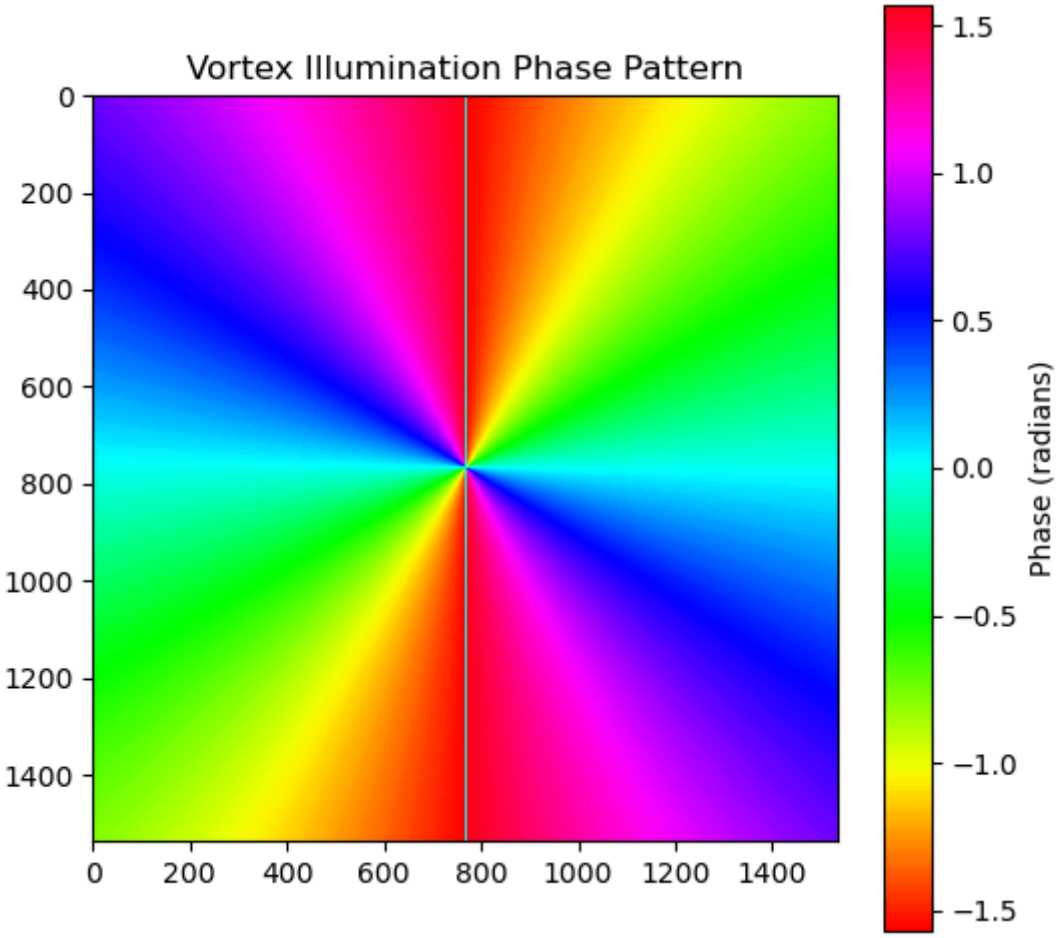
$$\theta(x, y) = \arctan\left(\frac{y}{x}\right)$$

The phase mask is applied to the original image to alter its Fourier domain properties, influencing the reconstructed image. istics.

In [231…

```python
# Function to create vortex phase pattern
def create_vortex_phase(shape):
    y, x = np.indices(shape)
    y = y - shape[0] // 2
    x = x - shape[1] // 2

    # Avoid division by zero
    epsilon = 1e-10
    safe_x = np.where(np.abs(x) < epsilon, epsilon, x)

    # Use arctan(y/x)
    angle = np.arctan(y / safe_x)
    return np.exp(1j * angle)
```

```
# Create vortex phase
vortex_phase = create_vortex_phase(padded.shape)

# Apply vortex illumination
object_with_vortex = padded * vortex_phase

# Display vortex phase pattern
plt.figure(figsize=(6, 6))
plt.imshow(np.angle(vortex_phase), cmap='hsv')
plt.colorbar(label="Phase (radians)")
plt.title("Vortex Illumination Phase Pattern")
plt.savefig(os.path.join(save_dir, "vortex_phase.png"))
plt.show()
```



HIO Algorithm Implementation

We apply the **Hybrid Input-Output (HIO) algorithm** to recover the lost phase information. The algorithm progresses as folleps:

1. Compute the **Fourier Transform** of the vortex-illuminated object.
2. Replace the amplitude with the known **diffraction pattern** while preserving phase.
3. Apply **Inverse Fourier Transform** to obtain an estimate.
4. Use the **support constraint** to refine the estimate.

5. Iterate the process to gradually recover the correct phase.

We set **β = 0.8** for the HIO update rule.

In [233…

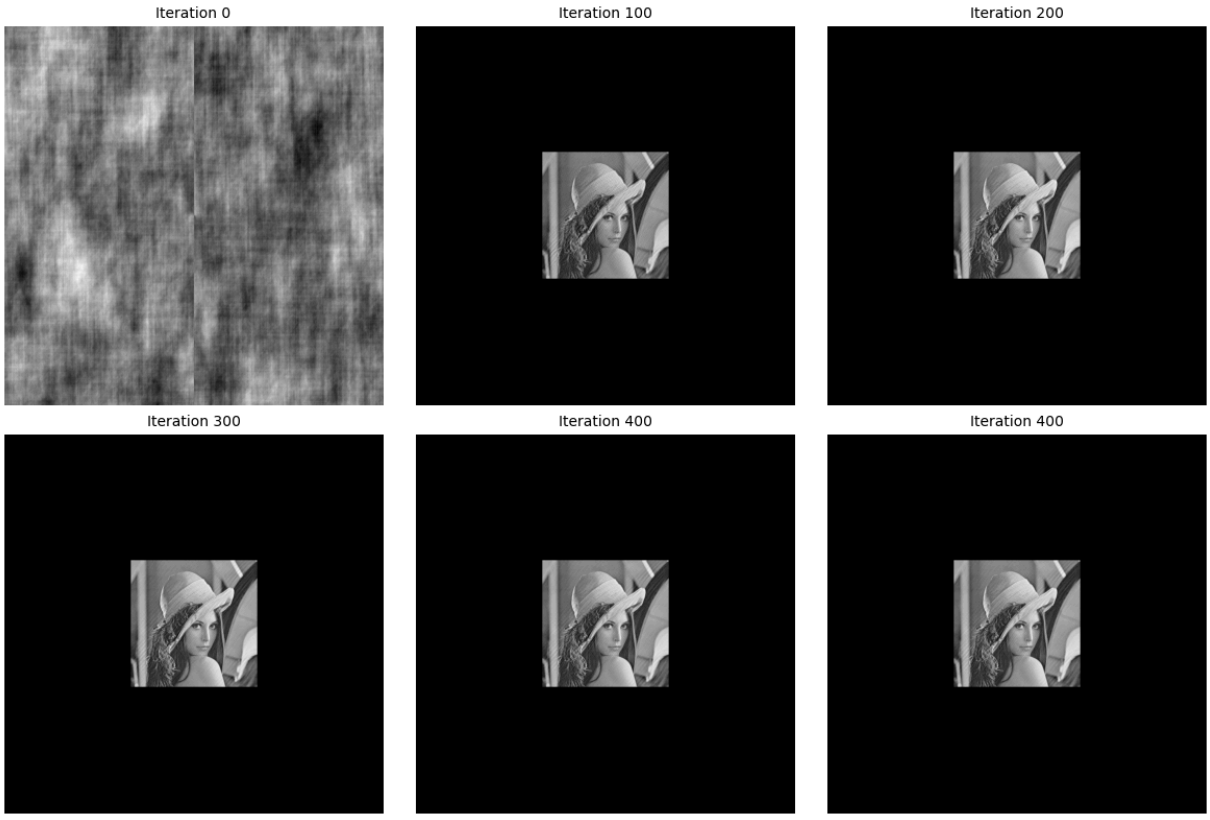
```
# Calculate diffraction pattern
ft = fft2(object_with_vortex)
diffract = np.abs(ft)

# Initial random phase guess
l = len(padded)
guess = diffract * np.exp(1j * np.random.rand(l, l) * 2 * np.pi)

# HIO parameters
iterations = 401
beta = 0.8
prev = np.zeros_like(padded, dtype=complex)

# Iterative Phase Retrieval
for s in range(iterations):
    update = diffract * np.exp(1j * np.angle(guess))
    inv = ifft2(update)

    # Remove vortex phase for constraint application
    inv_no_vortex = inv / vortex_phase
    real_part = np.real(inv_no_vortex)

    # Apply HIO constraints
    next_iteration = np.zeros_like(real_part, dtype=complex)
    positive_support = mask & (real_part >= 0)
    negative_support = mask & (real_part < 0)
    non_support = ~mask

    next_iteration[positive_support] = real_part[positive_support]
    next_iteration[negative_support] = np.real(prev[negative_support])
                                        - beta * real_part[negative_support]
    next_iteration[non_support] = np.real(prev[non_support])
                                        - beta * real_part[non_support]

    prev = next_iteration.copy()

    # Re-apply vortex phase
    next_with_vortex = next_iteration * vortex_phase
    guess = fft2(next_with_vortex)

    # Save progress images
    if s % 100 == 0 or s == iterations - 1:
        display_img = (real_part - real_part.min()) / (real_part.max()
                                                       - real_part.min()) * 255
        Image.fromarray(display_img.astype(np.uint8)).convert("L").save(
            os.path.join(save_dir, f"vortex_progress_{s}.png"))
        print(f"Iteration {s} saved.")

# Final reconstructed image
final_result = np.real(prev)
```

```
Iteration 0 saved.
Iteration 100 saved.
Iteration 200 saved.
Iteration 300 saved.
Iteration 400 saved.
```

The saved images are shown for reference.

In [235…

```python
save_dir = r"C:\Users\asus\Desktop\Image manipulation\save_vortex1"
# List of saved progress images
image_indices = list(range(0, 401, 100)) + [400]  # Includes 0,100,...,upto 400

# Define grid dimensions (2 rows, 3 columns)
fig, axes = plt.subplots(2, 3, figsize=(12, 8))

for ax, i in zip(axes.flat, image_indices):
    img_path = os.path.join(save_dir, f"vortex_progress_{i}.png")
    if os.path.exists(img_path):
        img = mpimg.imread(img_path)
        ax.imshow(img, cmap='gray')
        ax.set_title(f"Iteration {i}", fontsize=10)
        ax.axis("off")
    else:
        ax.axis("off")

plt.tight_layout()
plt.show()
```



In [236…

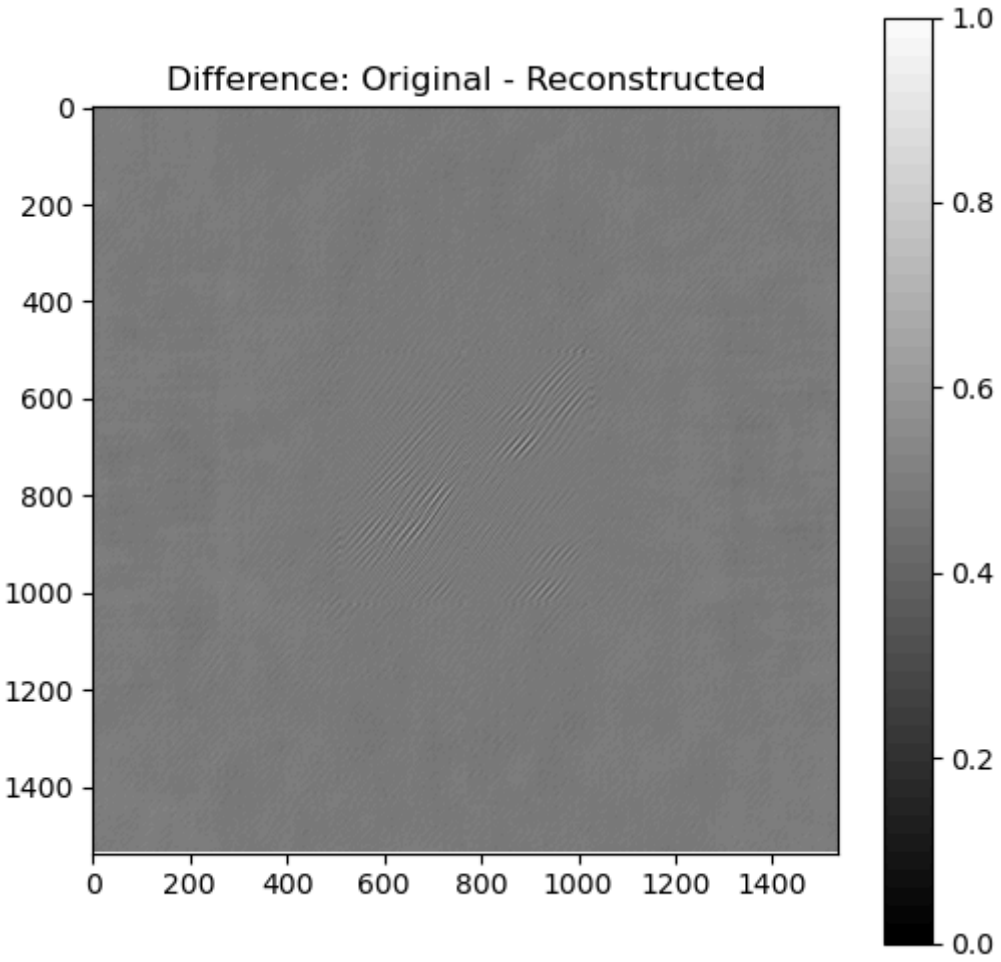
```python
fig, ax = plt.subplots(1, 2, figsize=(8, 8))

# Original Padded Image
```

```
ax[0].imshow(padded, cmap='gray')
ax[0].set_title("Original Padded Image")
ax[0].axis("off")

# Reconstructed Image
final_display = (final_result - final_result.min()) / (final_result.max()
                                                       - final_result.min())
ax[1].imshow(final_display, cmap='gray')
ax[1].set_title("Vortex Illumination Reconstruction")
ax[1].axis("off")

plt.tight_layout()
plt.savefig(os.path.join(save_dir, "vortex_final_comparison.png"))
plt.show()
```

The images are subtracted and the difference is obatained as outuput:

In [238…

```
# Compute difference
difference = padded - final_result

plt.figure(figsize=(6, 6))
diff_norm = (difference - difference.min()) / (difference.max()
                                               - difference.min())
plt.imshow(diff_norm, cmap='gray')
plt.title("Difference: Original - Reconstructed")
plt.colorbar()
plt.savefig(os.path.join(save_dir, "vortex_difference.png"))
plt.show()
```



Original vs. Reconstructed Image (Without Padding)

This section compares the original input image with the reconstructed image after iterative phase retrieval, **without any padding**.

In [240…

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

# Crop the reconstructed image to match the original size
cropped_reconstructed = final_result[pad_len:pad_len + height, pad_len:pad_len
                                     + width]

# Plot original and reconstructed images side by side
fig, axes = plt.subplots(1, 2, figsize=(10, 5))

# Original image
axes[0].imshow(source, cmap='gray')
axes[0].set_title("Original Image")
axes[0].axis("off")

# Reconstructed image (cropped)
axes[1].imshow(cropped_reconstructed, cmap='gray')
axes[1].set_title("Reconstructed Image (Cropped)")
axes[1].axis("off")
```

```
plt.tight_layout()
plt.show()
```

Plane vs. Vortex Illumination Reconstruction

Here, we compare the reconstructed images obtained using **plane wave illumination** and **vortex illumination**.

In [242…

```
# Ensures that the plane reconstruction was stored separately before applying
#vortex illumination!
if 'plane_reconstruction' not in globals():
    raise ValueError("Plane reconstruction not stored before vortex illumination!")

# Crop both reconstructions to original image size
cropped_plane = plane_reconstruction[pad_len:pad_len + height, pad_len:pad_len
                                     + width]
cropped_vortex = prev[pad_len:pad_len + height, pad_len:pad_len + width]

# Normalize both images for display
plane_normalized = (cropped_plane - cropped_plane.min()) / (cropped_plane.max()
                                                            - cropped_plane.min())
vortex_normalized = (cropped_vortex - cropped_vortex.min()) / (cropped_vortex.max()
                                                             - cropped_vortex.min()

# Plot Plane vs. Vortex Reconstructions
fig, axes = plt.subplots(1, 2, figsize=(10, 5))

# Plane Illumination Reconstruction
axes[0].imshow(plane_normalized.astype(np.uint8), cmap='gray')
axes[0].set_title("Plane Illumination Reconstruction")
axes[0].axis("off")

# Vortex Illumination Reconstruction
axes[1].imshow(vortex_normalized.astype(np.uint8), cmap='gray')
```

```
axes[1].set_title("Vortex Illumination Reconstruction")
axes[1].axis("off")

plt.tight_layout()
plt.show()
```

```
C:\Users\asus\AppData\Local\Temp\ipykernel_18244\4186126807.py:22: ComplexWarning: C
asting complex values to real discards the imaginary part
  axes[1].imshow(vortex_normalized.astype(np.uint8), cmap='gray')
```

This comparison shows that vortex illumination achieves better reconstruction in fewer number of iterations, apart from helping avoid stagnation phenomena.