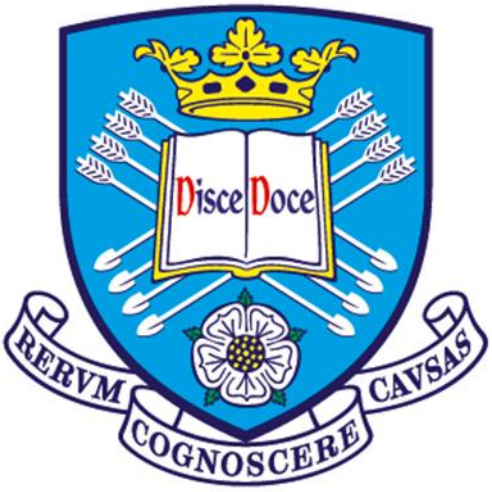
Analysis and development of phase retrieval algorithms for ptychography



Author: Zhuoqun Zhang

First supervisor: Andrew M. Maiden

Second supervisor: John M. Rodenburg

Department: Electrical and electronic engineering

Contents

Abstract

Ptychography, a relatively new form of phase retrieval, can reconstruct both intensity and phase images of a sample from a group of diffraction patterns, which are recorded as the sample is translated through a grid of positions. To recover the phase information lost in the recording of these diffraction patterns, iterative algorithms must optimise an objective function full of local minima, in a huge multidimensional space. Many such algorithms have been developed, each aiming to converge rapidly whilst avoiding stagnation. This thesis aims to set a standard error metric for comparing some of the more popular algorithms, to determine their advantages and disadvantages under a range of different conditions, and hence develop a more adaptive algorithm that combines the advantages of these ancestors.

In this thesis, different algorithms are explained together with their reconstruction results from both simulated and practical data. Modifications for mPIE, ADMM and RAAR are suggested to either reducing the number of parameters or improving their computation efficiency. An improved spatial error metric, which can evaluate the reconstruction quality by removing inherent ambiguities, is introduced to compare these algorithms. Based on the explained phase retrieval algorithms, a new algorithm, i.e., adaptive PIE, is developed. It has a faster converging speed and better accuracy comparing to its ancestors.

1. Introduction

Microscopy, as a fundamental tool for observing microstructure, plays an important role in biology, solid state physics, material science and integrated circuit etc[1,2]. From Antony van Leeuwenhoek observing the bacteria under optical microscopes with manually manufactured lens in 1683[3], to the modern electron and x-ray microscopes[4], the evolution of this technology has overcome many difficulties to expand the limitation of human vision from 1 µm (about the size of bacteria) to 0.2nm[5] and helped researchers to visually approve their discovery. *Figure 1. 1.* demonstrates dimensions of some common specimens and limitations of different microscopy technologies. As shown in the figure, even the most powerful modern microscopy technology is not flawless, and most of its limitation arises from a crucial physical component of microscopes since the first day it was developed: the lens.

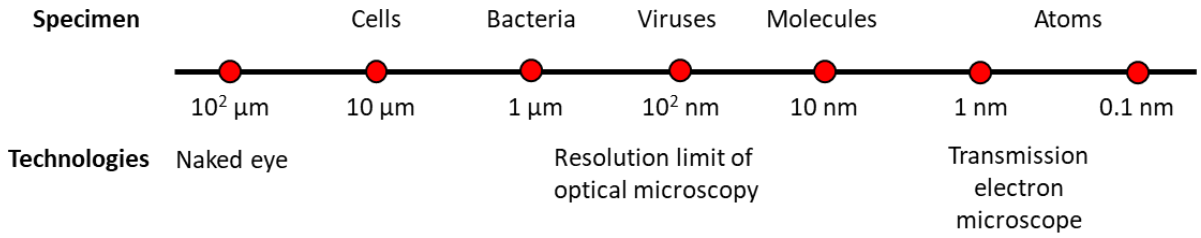


Figure 1. 1. Dimensions of some common specimens and limitations of different observing methods[1,3,5,6,7].

The lens is an important component in various kinds of optical devices. It is utilised in focusing illumination onto a specimen, and forming a magnified virtual image on the image plane in optical microscopy[5,7]. Nevertheless, a lens without defects is not easy to manufacture. Besides the transparency and coating requirement, a well-designed microscope lens must correct the spherical aberration[7]. Even with modern manufacturing technology, an optical microscope with good quality lenses is still expensive and its resolution is limited up to 200nm due to the diffraction limitation of visible light[5]. Such an accuracy is acceptable for cytology and histology but is still miles away comparing with the scale of atoms, which is 0.3nm[5,8]. For obtaining a decent image on atoms, a short illumination wavelength is compulsory to overcome the diffraction limit. However, the shorter a wavelength is, the higher quality lens is required[9,10]. When wavelength comes down to 10nm or less, which is utilised in most of the

x-ray microscopy, a good quality lens is too expensive to afford, if indeed it is possible to produce[11]. This drawback does not only limit image quality, but also slows down the spread of these microscope technologies due to their enormous investments[12].

Many solutions have been attempted since the late 20th century. An idea gradually attracts the attention of researchers: what if there is no lens[13,14]? It was mind breaking to think of how to form an image without a lens, since no image can be detected directly without it[2,14,15]. Instead, other information can be measured, for example a diffraction pattern (e.g. *Figure 1. 2 (b)*), which may contain sufficient information to reversely figure out the image of a specimen. This new approach bypasses the limitation caused by lenses and makes the image solution regardless of lens quality[1]. Due to this special characteristic, the new concept is known as lens-less imaging[13,14]. One should notice that lens-less imaging does not mean no lens is allowed in this technology. As shown in *Figure 1. 3*, lenses may still be utilised for converging illumination in the upper stream. They are just not compulsory in imaging and hence do not determine the image quality[1,13,14]. Lens defects and aberrations have no significant impact on the final images. Besides improving image quality, lens-less imaging also brings other benefits, including a quantitative phase image together with the intensity image and the possibility of 3D microscopy[16,17,18].

Nevertheless, all these benefits come with a question that must be solved first, which is the phase problem[2,14]. This problem is caused by the lost phase information during taking intensity measurements in lens-less microscopy: when a diffraction pattern is recorded, the phase of the wavefront that caused it cannot be detected. Without this lost phase it is not possible to reconstruct an image of the specimen, but if somehow the phase can be figured out and added to the diffraction recording, the specimen can be revealed by digital propagation of the recovered wavefront.

Many efforts have been made for solving this problem[13,15]. Ptychography, as one of the most successful solutions up to date, is developed together with various phase retrieving algorithms. It does not only provide a satisfied solution to phase problem[19], but also adapts to various types of illuminations and pushes the image quality to a higher level. As a price, ptychography suffers from its complexity in retrieving phase iteratively. This process is time consuming and might lead to a fruitless end due to improperly tuned algorithms or noisy measurements.

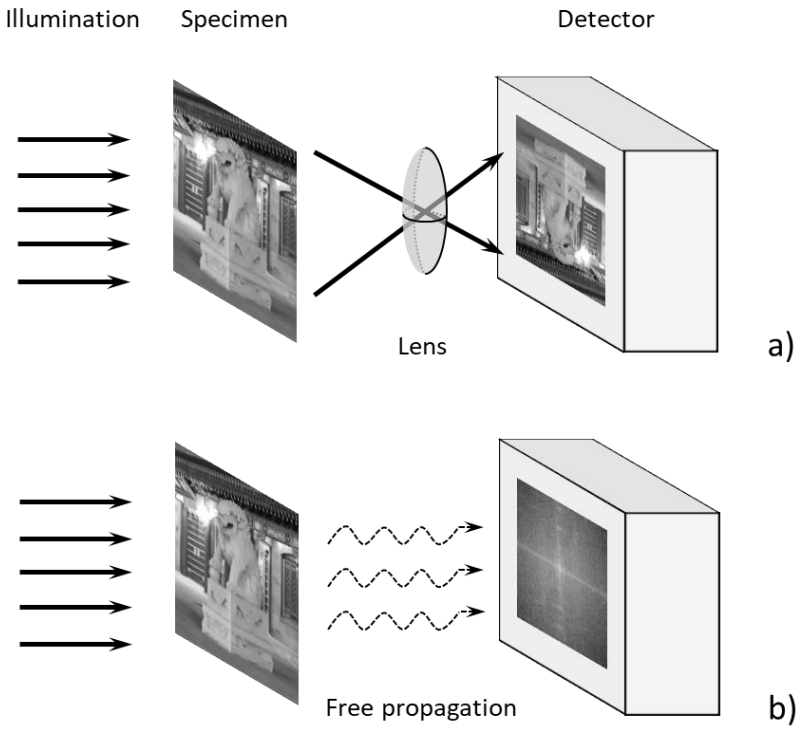


Figure 1. 2. A general demonstration of the detected image with and without lens. An illumination goes through a specimen from left to right. With the existence of lens (a), the light is focused onto the detector plane. Hence the intensity image is detected directly. Without the lens (b), waves propagate freely and form a diffracted pattern (i.e. diffraction pattern) on the detector. The detector only records the intensity of diffraction pattern, which is nothing like the intensity of image. Since the phase information of diffracted wave is lost during the measurement, it is impossible to obtain the intensity image with inversely propagating the diffracted wave.

Ptychography operates as shown in A beam of illumination ($\mathbf{P}_{\vec{r}}$) made up of coherent waves shines on a specimen ($\mathbf{O}_{\vec{r}}$), which has negligible thickness. This illumination is absorbed and diffracted while propagating through this specimen and is detected by a detector sitting at the downstream as an intensity measurement ($\mathbf{I}_{\vec{u}}$). Once a diffraction pattern is fully recorded, a relative shift is introduced between the illumination and specimen, hence their contact area changes slightly but still shares more than half (e.g. 60~70%[20]) overlapping area with the previous illuminated area. As a result, a new diffraction pattern appears on the detector. It is recorded together with the corresponding shifts. This 'shift-and-detect' process is repeated until the whole area of interest is covered. These diffraction patterns together with the shifting positions are the data collected by ptychography in experiments and will be used for phase retrieving later.

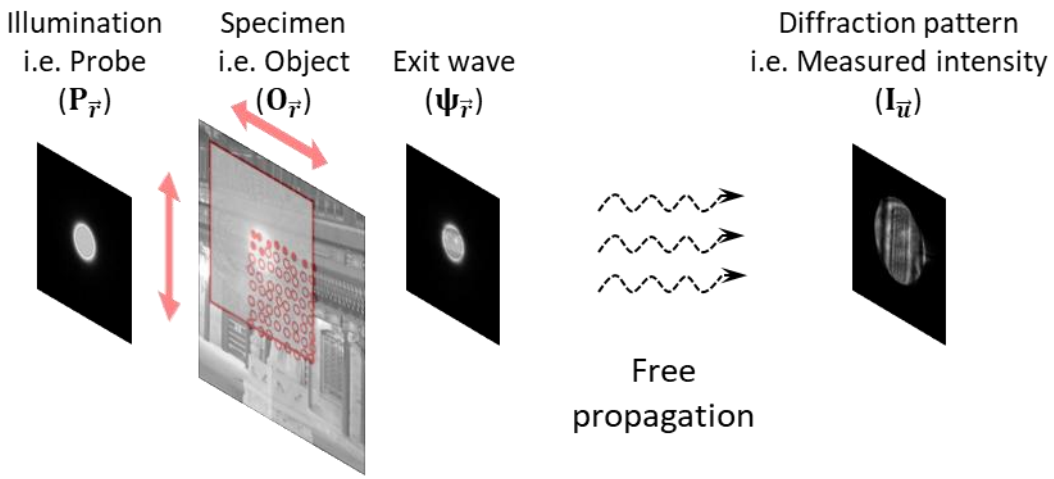


Figure 1. 3. A simplified demonstration of ptychography. From left to right, an illumination ($\boldsymbol{P}_{\vec{r}}$) shines onto the specimen ($\boldsymbol{O}_{\vec{r}}$) and turns into an exit wave ($\boldsymbol{\psi}_{\vec{r}}$). The contact area is highlighted by the red square on the specimen. After a far-field free propagation, the exit wave is detected as a diffraction pattern ($\boldsymbol{I}_{\vec{u}}$) at the downstream.

An improvement on the phase retrieving algorithms is crucial to make ptychography produce promising results under various test scenarios. Therefore, this research focuses on evaluation and development of algorithms for ptychography. Developing a new algorithm requires a good understanding on the existed ones, hence the most famous phase retrieval algorithms are explained and tested with different simulated scenarios including practical data sets. The evaluation of reconstructed images requires an accurate error metric, which prevents the interference of ambiguities. Several error metrics are explained together with a standard process of reducing the inherent ambiguities. Based on the solid understanding of existed algorithms, a more adaptive and promising phase retrieval algorithm is introduced. This new algorithm (i.e. adaPIE) is inspired by the methods utilised in deep learning and is highly competitive in various test scenarios.

This thesis is organised as follows. In Chapter 2, the phase problem of less-less imaging is explained from both physics and mathematical point of view. Several existed device set-ups for lens-less imaging are also generally introduced based on their order of development. Ptychography, as one of the most attractive candidates[1,21,22], are explained in depth. This chapter also includes the conventions and definitions utilised through the whole thesis.

Chapter 3 introduced the well-known existed phase retrieval algorithms. This chapter starts with the mathematics background for understanding the phase retrieval algorithms. Then these algorithms are separated in two categories and explained with pseudo code, flowchart,

modification and hardware usage. A noiseless simulated data set is utilised to evaluate the effectiveness of these algorithms. Some related knowledge of computer hardware is given in the end of this chapter.

Chapter 4 explains the error metrics for evaluating the quality of reconstructed images. Since computing error metric requires minimising inherent ambiguities in advance, all inherent ambiguities of ptychography are explained together with corresponding removing methodologies. The performance of explained algorithms are tested with simulated data.

Chapter 5 introduces a new phase retrieval algorithm: adaptive PIE, which is inspired by the algorithms for training neural networks. A new regularisation approach, which adapts to the over-all illumination intensity, is explained. Various tests with different difficulties are applied to test this new phase retrieval algorithms. The test results reveal this algorithm has a good converging speed with a decent reconstruction quality.

Chapter 6 talks about dealing with practical data that is collected with a scanning transmission electron microscope (STEM). Methods for checking the collected data before reconstruction are explained and applied to calibrate the practical data. Then different phase retrieval algorithms are utilised to reconstruct this data set.

Chapter 7 introduces some optional constraints that can be added into phase retrieval algorithms. The ‘probe calling map’ can visualise how a pixel of probe is related to itself by a given scanning grid, while other constraints are developed to prevent stagnation or accumulation of ambiguities. Some of these ideas show a strong potential and can be expanded with further research.

Chapter 8 is the conclusion chapter, which summaries the development mentioned in this thesis and suggests some topics for future research.

Glossary

Variables	
$\mathbf{A}_{\vec{r}}$	Aperture
$\mathbf{S}_{\vec{r}}$	Support matrix
$\mathbf{I}_{\vec{u}}$	Diffraction pattern of single intensity measurement
$\mathbf{I}_{\vec{u},k}$	The k^{th} diffraction pattern of multiple intensity measurements
$\mathbf{I}_{img_{\vec{r}}}$	Intensity measurement of specimen
$\mathbf{L}_{\vec{r}}$	Lens
$\widehat{\mathbf{O}}_{\vec{r}}$	Specimen, true object
$\widehat{\mathbf{P}}_{\vec{r}}$	Illumination, true probe
$\mathbf{O}_{\vec{r}}$	Guessed object
$\mathbf{P}_{\vec{r}}$	Guessed probe
$\mathbf{O}_{\vec{r}}{}'$	Revised object
$\mathbf{P}_{\vec{r}}{}'$	Revised probe
$\boldsymbol{\psi}_{\vec{r}}$	Exit wave in spatial domain
$\boldsymbol{\psi}_{\vec{r},k}$	The k^{th} exit wave
$\boldsymbol{\psi}_{\vec{r}}{}'$	Revised exit wave
$\mathbf{f}_{\vec{r}}$	Modulus of exit wave
$\mathbf{g}_{\vec{r}}$	Phase of exit wave
$\boldsymbol{\Psi}_{\vec{u}}$	Exit wave in frequency domain, i.e. Fourier transformed exit wave
$\boldsymbol{\Psi}_{\vec{u},k}$	The k^{th} Fourier transformed exit wave
$\boldsymbol{\Psi}'_{\vec{u}}$	Revised Fourier transformed exit wave
$\boldsymbol{\lambda}_{\vec{r}}$	Multiplier
$\mathbf{F}_{\vec{u}}$	Modulus of Fourier transformed exit wave
$\mathbf{G}_{\vec{u}}$	Phase of Fourier transformed exit wave
$\mathbf{0}$	An 'all-zero' matrix
$\mathbf{1}$	An 'all-one' matrix
$\mathbb{M}$	Modulus set formed by diffraction intensity measurement
$\mathbb{M}_{img}$	Modulus set formed by specimen intensity measurement
$\mathbb{S}$	Support set

$\mathbb{O}$	Consistency set of ptychography
$\vec{r}$	Referring vector in spatial domain
$\vec{u}$	Referring vector in frequency domain
$\vec{r}_k$	The k^{th} shifting vector
dxy	Conversion ratio (unit: meter/pixel)
d_{cam}	Detector dimension or camera dimension
l_{cam}	Camera length
λ	Wavelength (unit: m)
G	Detector gain
η	Quantum efficiency (of detector)
K	Detector sensitivity
$M \times N$	Size of intensity measurement, i.e. M rows and N columns
K	Total No. of intensity measurements
D	A complex space, defined by the size and number of intensity measurements: $D = M \times N \times K$
$\boldsymbol{x}$ and $\boldsymbol{y}$	Two example vectors in space D
θ	Rotating angle (in degree)
k_{scale}	Scaling factor for scan positions
$\overrightarrow{r_{\Delta,k}}$	Random shift on the k^{th} scan position
$\theta_{span\ of\ detector}$	The span angle of detector
$\mathbf{R}$	Rotation matrix for coordinates in 2-dimensional space
$Energy_k$	Energy of k^{th} diffraction pattern, the sum of squared of modulus for a complex matrix
Err_{SS}	The summed squared error
$s-domain$	Real-space[23], imaging plane[24] or spatial domain
$f-domain$	Reciprocal-space, momentum-space[23], diffraction plane[24], Fourier domain or frequency domain
$s-constraint$	Constraint formed by the priori information in s-domain
$f-constraint$	Constraint formed by the diffraction patterns
a	s-domain ambiguity: scaling constant
e^{jc}	s-domain ambiguity: phase offset

$e^{jb\cdot\vec{r}}$	s-domain ambiguity: phase ramp
$+\vec{d}$	s-domain: ambiguity: global shift
Super/sub script	
k	Variable at the k^{th} scanning position
n	Variable at the start of the n^{th} iteration
$\vec{\mathrm{r}}$	Variables in real space
$\vec{u}$	Variables in reciprocal space
Operator	
*	Complex conjugate
$\|\ \ \|$	Modulus of complex number
$\cdot$	Element-wise multiplication
$\angle(\ \ )$	Put following terms onto the phase, e.g. $A\angle\theta = A\cdot exp(j\cdot\theta)$
$(\ \ )_{max}$	The maximum value of in the matrix
$\boldsymbol{\mathcal{F}}$	2-dimensional Fourier transformation
$\boldsymbol{\mathcal{F}}^{-1}$	2-dimentional inverse Fourier transformation
$\boldsymbol{Cut}$	Cut out a part of matrix
$\boldsymbol{Add}$	Add a part onto a larger matrix
$\boldsymbol{shuffle}$	Shuffle a sequence
$\boldsymbol{\mathcal{P}}_f$	Projection to the modulus constraint to *f-domain* variables
$\boldsymbol{\mathcal{r}}_f$	Reflection to the modulus constraint to *f-domain* variables
$\boldsymbol{\mathcal{P}}_f$	Projection to the modulus constraint to s-domain variables
$\boldsymbol{\mathcal{P}}_{img}$	Projection to the image intensity constraint
$\boldsymbol{\mathcal{P}}_{\mathrm{support}}$	Projection to the support constraint
$\boldsymbol{\mathcal{P}}_{\mathrm{s}}$	Projection to consistency set
$\boldsymbol{\mathcal{R}}_f$	Reflection to modulus constraint
$\boldsymbol{\mathcal{P}}_f^{\alpha}$	Relaxed f-constraint projection with relaxing coefficient $\boldsymbol{\alpha}$
$\boldsymbol{\mathcal{P}}_s^{\alpha}$	Relaxed s-constraint projection with relaxing coefficient $\boldsymbol{\alpha}$
$\boldsymbol{\mathcal{I}}$	Identity (i.e. 'Unchanged') operator
$\boldsymbol{\mathcal{L}}$	Cost function

2. The phase problem and ptychography

This chapter explains the significance of the phase problem (section 2.1) and details its causes (section 2.2). Different methodologies have been attempted to solve this problem by applying constraints that reflect knowledge about the lens-less experimental process. These methodologies and constraints are explained in section 2.3. Among them ptychography is the most attractive one and highly relates to this thesis. Thus, its detailed description is given from both physics and mathematics point of view (section 2.4).

2.1. The meaning of phase problem

The phase problem is caused by two facts. First, all waves have both amplitude and phase property no matter they are electromagnetic wave[2] or matter wave[9] (e.g. electron beam). Second, detectors can record the intensity of a contact wave, which equals the square of its amplitude, but cannot record the phase simultaneously[13]. Such an unideal situation is due to the extremely short response time required for measuring phase in this circumstance. For instance, the illumination beam utilised in ptychography usually has wavelength in micrometre scales (e.g. $10^{-6}m$) with propagating speed equal to speed of light (e.g. $3 \times 10^{8} m\, \boldsymbol{s^{-1}}$)[9]. As a result, this wave can propagate through a distance equal to its wavelength within femtosecond (e.g. $10^{-15}s$). Taking the Nyquist sampling theorem into account, the response time of a detector must be on the scale of fractional of femtosecond to be able to collect the phase information. In other words, an effective measurement on the wave phase requires the detector having response frequency as multiple of petahertz (e.g. $10^{15}Hz$), which is impractical for nowadays detectors that usually have 2-10MHz[23].

Using optical microscopy as an example, the brightness of an illumination is affected by the transparency of a specimen during the propagation[2,7]. This brightness relates to the content of specimen and is focused into an image with intensity variation and recorded by a detector. Hence researchers can observe the intensity image directly on the detector, though the phase information is lost[5,7].

Without a lens, the wave will diffract during propagation rather than focusing into an image. An at least partially coherent light source is required to produce an analysable diffraction pattern onto the detector[13]. If the full information of this diffracted pattern is recorded (i.e.

record both its intensity and phase), a clear image still can be obtained by inversely propagating this detected wave[2]. However, as only its intensity is recorded, the lost phase makes a directly reverse propagation impossible. The difference between these two set-ups has been demonstrated in *Figure 1. 2*.

The phase problem happens when only intensity is recorded but a reverse propagation is desired. This is not a trivial problem that merely exists in lens-less microscopy. Its variation is also noticed in other imaging technologies. For instance, in the optical astronomy, solving phase problem is helpful for extracting useful information from a foggy background[25]. Besides that, this sort of problem is also known as the inverse problems in mathematics and have been studied in a more general form[26,27]. In return for solving phase problem, lens-less imaging does not only provide an intensity image but also a phase image of the specimen. Although phase image has shorter history comparing to the intensity image, it has significant meaning on observation[21,28]. The most considerable advantage is even a completely transparent specimen can generate a phase image, as long as it is made with a material that has a different refractive index with its surroundings[1]. Do not even mention solving phase problem provides the phase image together with the traditional intensity image[29]. Such a combination offers diverse information of a specimen. Some astonishing improvements brought by phasing imaging, especially ptychography, are listed below:

2.1.1. Imaging living cells

A typical usage of phase imaging is the study of living cells. Since living cells are nearly transparent to illuminations, they are almost invisible under an optical microscopy besides their edges where a significant transition of reflection index happens[21,30,31]. A traditional solution is changing their transparency by staining, though this process usually involves killing cells to let pigment pass through their membrane[30]. With this manner, it is impossible to observe how a living cell response to the stimulation in real time, which increases the uncertainty and difficulty of research. Methods like fluorescence microscopy[32,33] is developed but requires a long-term preparation on specific samples. As a comparison, due to their different refractive index, cells can provide clear, high-contrast phase image while staying alive[34,35] as shown in *Figure 2. 1*. In addition, the characteristic of phase image is also helpful on distinguishing different materials, which is convenient for material science[36].

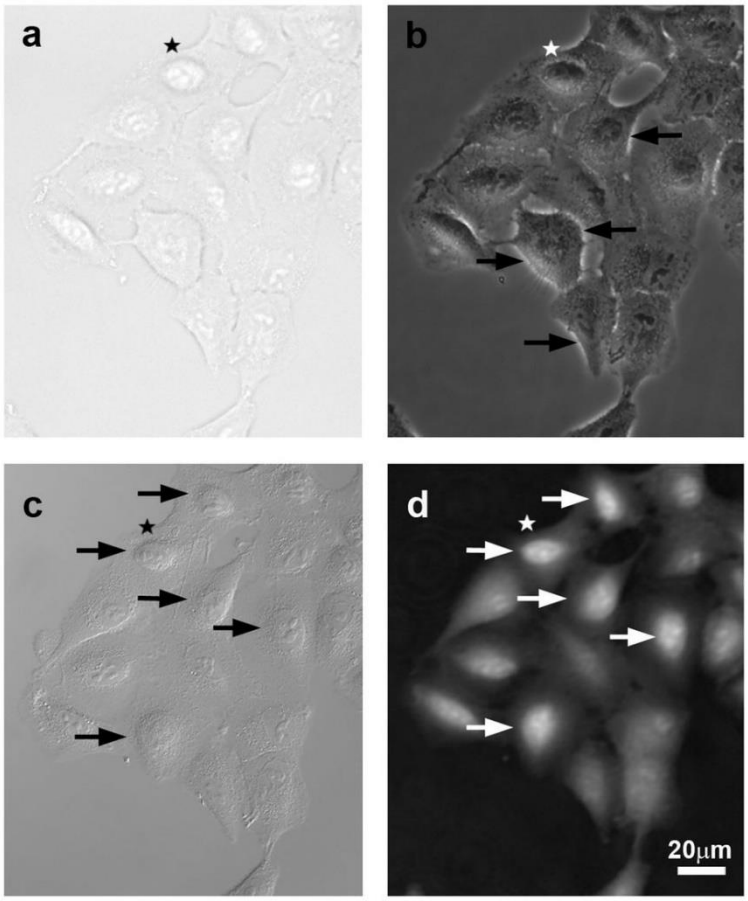


Figure 2. 1. The comparison between the modulus and phase images of a group of cells. Figure 2a demonstrates the modulus image produced by the brightfield microscopy. Since the cells are semi-transparent, the only visible details are the regions with significant transition of reflective index, e.g. the edge of membrane. Figure 2b and 2c are results of the phase contrast image and the differential interference contrast (DIC) image respectively. They have a better contrast than the modulus image. Figure 2d is the phase image provided by ptychography. It uses grey level to represent the phase shift caused by the different reflective index and thickness of the cells. It has the strongest brightness contrast among these four images. The arrows in the above images indicate the cells that are probably into the G2/M state. Three phase images provide a better contrast on these cells, while ptychography has the best contrast among them. The star shows an example point, at which ptychography gives an enhanced contrast[21].

2.1.2. Resolution improvement on electron microscopy

Ptychography also brings new potential to many mature microscope technologies. A recent example is the new world record in resolution achieved by the combination of ptychography and electron microscope[23]. Previously, the main approaches for improving the resolution of electron microscopy was either increasing the beam energy or the numerical aperture[9,23], though the former one has potential damage to the specimen, while the later one could

introduce aberrations[23]. However, by introducing the ptychography, a better resolution (e.g. 0.39 ångström) has been achieved with significantly smaller beam energy (80keV) on a MoS_2 sample[23]. As a comparison, the traditional technology can only achieve 0.98 ångström under the same circumstance. Such a resolution makes observing single atom defect possible as shown in *Figure 2. 2*.

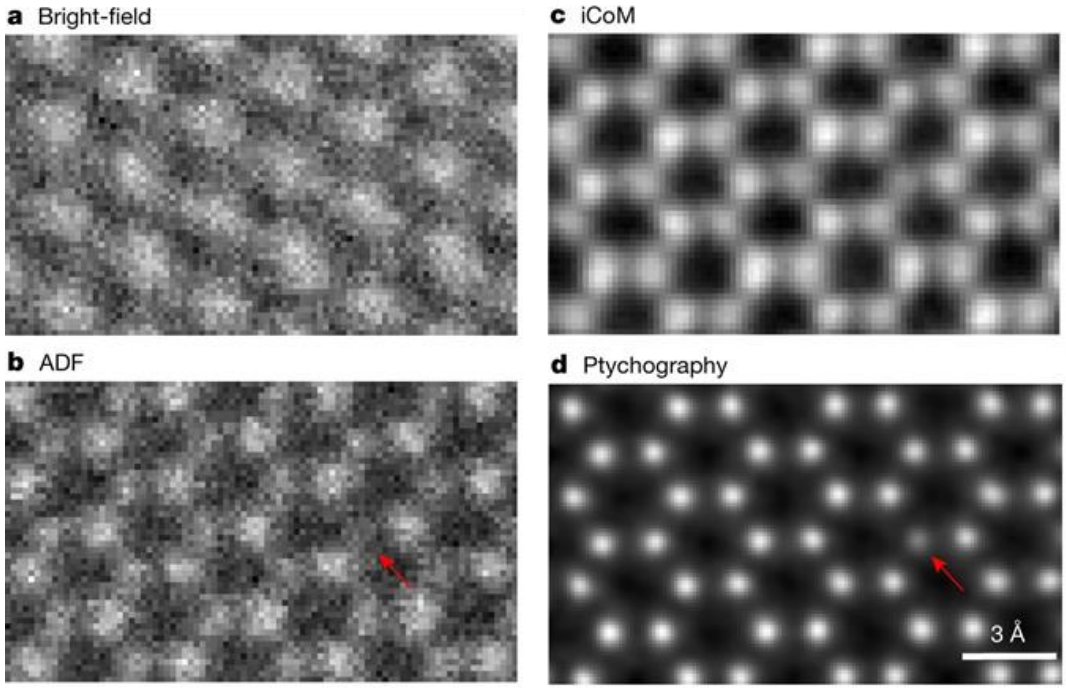


Figure 2. 2. A comparison of reconstructed images from different electron microscopy technologies. From (a) to (d) are the MoS_2 images obtained by bright-field, angular dark-field (ADF), integrated centre of mass (iCoM) and full-field ptychography. A 3 ångström scale bar is shown in (d). The red arrows indicate a single atom defects in the specimen[23].

2.1.3. 3-dimentional microscopy

The development of phase imaging also makes 3D microscopy possible[37]. To image a 3D structure, a light source that can penetrate through a specimen freely is a must. However, if the illumination can pass through the sample without significant absorption, its intensity image will have weak contrast. On the other hand, its phase accumulates linearly during passing through different materials. With a properly retrieved phase, one can figure out the material contacted by the illumination with an inverse propagation, hence reconstruct the internal structure without disassembling it. *Figure 2. 3* shows an example of observing the 3D transistor structure inside a chip under the help of phase retrieving.

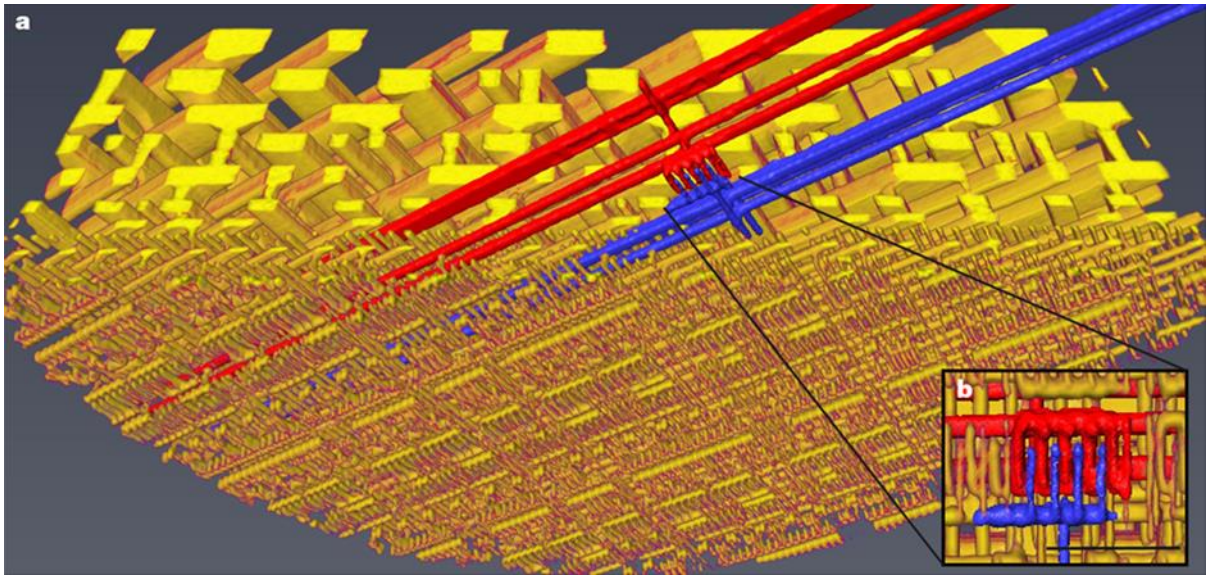


Figure 2. 3. A 3D image of a 10μm diameter chip produced by the ptychography X-ray computed tomography (PXCT). The finest structure of inside transistor is obtained without broken the sample. a, whole active structure of the chip. b, the fine structure of a single transistor[37].

2.1.4. Fourier ptychography

The idea of gaining extra priori information by collecting multiple intensity measurements with overlapping positions also inspired other methodologies. One of the variants of ptychography, Fourier ptychography, also gradually attracts attention of the researches. Like ptychography, Fourier ptychography also relies on a set of intensity measurements to recover the lost phase, hence synthesis all measurements into a single complex-valued image. However, its real and reciprocal space constraints are swapped due to the existence of lens in Fourier ptychography[38]. Such a similarity with conventual microscope system allows the Fourier ptychography being applied with minor modification on the existed microscope platform. Besides preventing the conflict between the resolution and the field of view, which widely exists in conventional microscope systems, Fourier ptychography is capable for aberration correction and refocus images during the reconstruction, which significantly reduces both the expense on high quality lens system and the difficulty caused by device calibration.

As a quick summary, phase problem is a widely existed problem and have general meaning in various aspects. Solving it will not only bring the benefits of lens-less imaging to microscopy technology, but also inspire other research topics.

2.2. The cause of phase problem

To begin with, a model of wave propagation needs to be stated together with an important approximation, 'Fraunhofer approximation', on which the rest of thesis is based. The basic propagation of coherent light from source plane to an observation plane is described by *eq 2. 1*. This expression assumes the source is formed by infinite number of fractional point light source, and the wave detected at any point of observation plane is the combination of emitted wave from all these light sources. λ is the wavelength, k is the wavenumber, which equals $2\pi/\lambda$. Other related variables are demonstrated in *Figure 2. 4*[39].

$$U_2(u_1,u_2) = \frac{z}{j\lambda}\iint U_1(r_1,r_2)\frac{\exp(jkd_{12})}{d_{12}^2}dr_1dr_2 \qquad \textit{eq 2. 1}$$

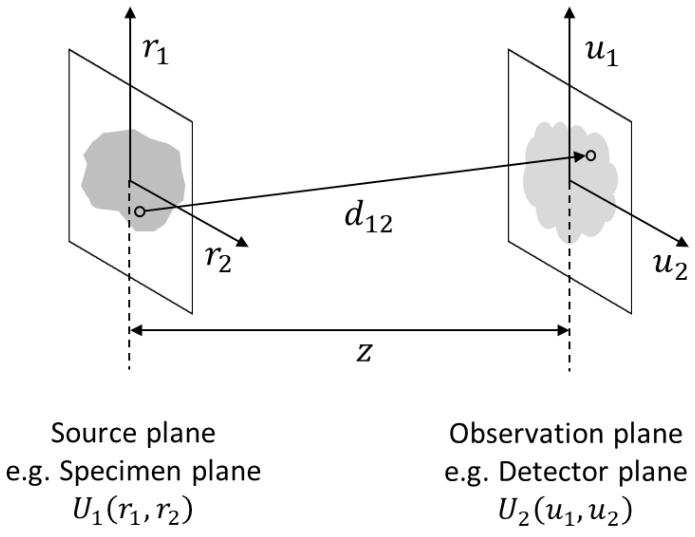


Figure 2. 4. A model for wave propagation. In this simplified model, the source plane (U_1) and the observation plane (U_2) are in parallel and separated by distance z. Each of them has their own coordinates as marked in the figure. The distance from a random point on source plane to the observation plane is labelled as d_{12}.

Although *eq 2. 1* gives an accurate expression on the propagating wave, its complexity, especially the square root term, increases the time spend on simulation, also make solving it for a solution a difficult task[39]. To make the expression less computationally expensive, two famous approximations are developed: the Fresnel approximation and Fraunhofer approximation, while the later one is highly related to this thesis. The Fraunhofer approximation happens when the propagation distance is significant comparing with the source size. Such a priori condition is named as 'far-field' and expressed as:

$$z \gg \left(\frac{k({r_1}^2 + {r_2}^2)}{2}\right)_{max} \qquad \textit{eq 2. 2}$$

Under this assumption, the diffraction effect can be derived as:

$$U_2(u_1, u_2) = \frac{\exp(jkz)}{j\lambda z}\exp\left[j\frac{k}{2z}({u_1}^2 + {u_2}^2)\right] \times \iint U_1(r_1, r_2)\exp\left[-jk\left(\frac{u_1}{2z}r_1 + \frac{u_2}{2z}r_2\right)\right]dr_1 dr_2 \qquad \textit{eq 2. 3}$$

What makes the Fraunhofer approximation widely applied is its relationship with Fourier transformation. By substitute the $f_\xi = \frac{x}{2z}$ and $f_\eta = \frac{y}{2z}$, one can see the *eq 2. 3* can be simplified as a scaled Fourier transformation of the source wave ($U_1(\xi, \eta)$), which considerably reduces the difficulty of analysing the resultant wave.

$$U_2(u_1, u_2) = \frac{\exp(jkz)}{j\lambda z}\exp\left[j\frac{k}{2z}({u_1}^2 + {u_2}^2)\right] \cdot \boldsymbol{\mathcal{F}}\big(U_1(r_1, r_2)\big) \qquad \textit{eq 2. 4}$$

All diffraction patterns mentioned in this thesis are all collected based on the Fraunhofer approximation, which implies the 'far-field' assumption should be applied carefully.

Now let us consider a basic lens-less imaging system without any imperfect issue. A specimen ($\widehat{\mathbf{O}}_{\vec{r}}$) is illuminated by a coherent plane wave. As the illumination passes through the specimen, it is diffracted and turns into an exit wave ($\boldsymbol{\psi}_{\vec{r}}$), which is expressed as a complex matrix. Its modulus is affected by the transparency of the specimen, while its phase is affected by the refractive index and thickness variations of the specimen[2]. Referring to *Figure 1. 2 (b)*, since

there is no lens in the set-up, the exit wave propagates freely, interferes with itself and turns into a diffracted wave ($\mathbf{\Psi}_{\vec{u}}$) in the far field.

A detector is placed at downstream to detect the diffracted wave ($\mathbf{\Psi}_{\vec{u}}$). As explained previously, the detector only records the square of the modulus of the diffracted wave (i.e. the diffraction pattern ($\mathbf{I}_{\vec{u}}$)), while the phase is lost. The detector samples the diffraction pattern and records the intensity on each sampling point[10]. These sampling points are considered as pixel from now on. The final measurement is a matrix, whose size equals the pixel-wise dimension of the detector, filling with real numbers that indicate the light intensity detected at each pixel.

The relationship between these variables is demonstrated in *Figure 2. 5*. These variables belong to two different spaces (domains) that are related by Fourier transformation in this scenario. For the sake of simplicity, the space holding variables before propagating onto the detector is named as "real space" or the spatial domain (*s-domain*). Another space holding variables after the propagation is named "reciprocal space" or the frequency domain (*f-domain*).

The eventual aim of solving the phase problem is to determine a matrix (the object matrix) that represents the transmission characteristics of the specimen under investigation in the lens-less microscope. Without any constraints, this matrix could be populated with any of an infinite set of complex values. The measured diffraction pattern reduces this set of values drastically by introducing the following ***f-domain constraint:***

The amplitude of the Fourier transform of the object matrix must match the amplitude of the recorded diffraction pattern (i.e. the square root of the recorded intensity).

$$|\mathbf{\Psi}_{\vec{u}}| = \sqrt{\mathbf{I}_{\vec{u}}} \qquad \textit{eq 2. 5}$$

Nevertheless, $\mathbf{I}_{\vec{u}}$ only holds the modulus information. Due to the lost phase, it is mathematically impossible to find a unique diffracted wave from a given intensity based only on the f-domain constraint. Therefore, a phase problem with only f-constraint is not well-conditioned and insoluble.

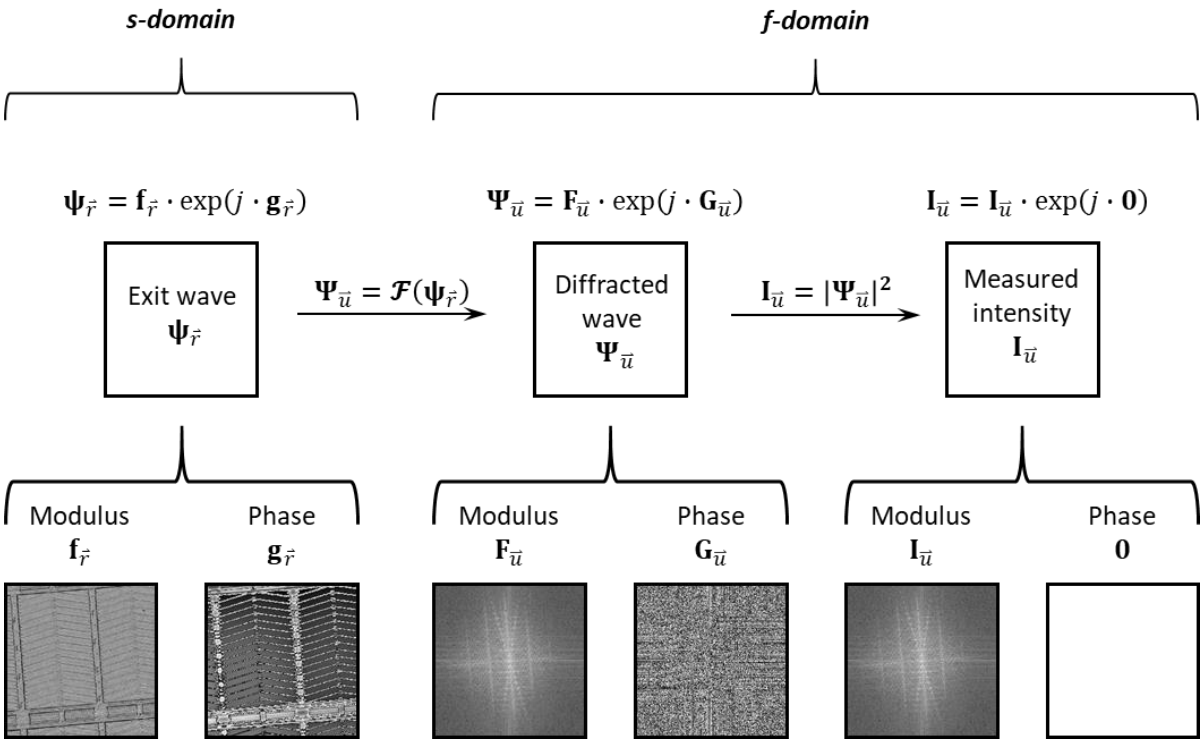


Figure 2. 5. The mathematic expression of the exit wave ($\boldsymbol{\psi}_{\vec{r}}$), diffracted wave ($\boldsymbol{\Psi}_{\vec{u}}$), measured intensity ($\boldsymbol{I}_{\vec{u}}$) and their relationship. All these variables are in equal size of matrices, which is expressed as a square in above figure. Variables in real space and reciprocal space are distinguished by sub-script $\vec{r}$ and $\vec{u}$ respectively (more explanations on real and reciprocal space variables in section 2.4). The exit wave and diffracted wave are made of complex elements; hence they have both modulus and phase images. As measured intensity is filled with real numbers, it only has modulus image, while its phase image is matrix filled up with zero. With this figure, it is obvious that inversely finding a unique diffracted wave with only intensity measurement is not possible.

2.3. The solution of phase problem

The f-domain constraint by itself is not adequate for solving the phase problem. Other constraints must be involved to further confine it[13]. Therefore, the history of lens-less imaging is about looking for other constraints. Various desired constraints are generated by different device set-ups and affect development of the solving approach slightly, though they all share one thing in common: they are all constraints in the real space (i.e. ***s-constraints***). A quick glance on these constraints can demonstrate how ptychography is inspired by its ancestors. More importantly, the advantages brought by ptychography are revealed in this comparison. In the following section, these existed phase imaging methods are explained together with the constraints that they provided.

2.3.1. Intensity image of specimen

Back to the early stage of lens-less imaging, the available information and experience are very limited. Instead of stepping into a new topic without any directions, combining with the lens imaging system can produce more information for reconstruction. After all, lens is already a mature device with hundreds of years of development and capable of providing an excellent image in most optical experiments[7]. Hence a constraint formed by the intensity image of the specimen is suggested.

This method[24, 40] requires an adjustment on the device set-up during experiment, which is shown in *Figure 1. 2*. *(a) and (b)*. An intensity image of the specimen is acquired with the help of lens, then a diffraction pattern is measured without the lens. The measured image intensity (i.e. $\mathbf{I}_{\boldsymbol{img}_{\vec{\mathbf{r}}}}$) forms a ***s-domain constraint***, which is also named as ***specimen intensity constraint*** as shown below:

$$|\boldsymbol{\psi}_{\vec{\mathbf{r}}}| = \sqrt{\mathbf{I}_{\boldsymbol{img}_{\vec{\mathbf{r}}}}} \qquad \textit{eq 2. 6}$$

Although this constraint makes the phase problem theoretically soluble, it has some drawbacks. The most significant one is the existence of lens in a technology that is designed to be lens-less. Secondly, this new constraint formed by the intensity of the specimen is barely enough for solving the unknowns [40]. Research indicates that it takes intensive computations to get a correct reconstruction[40]. Nevertheless, this method proves the possibility of solving phase problem by introducing another constraint. The Gechberg-Saxton algorithm is developed together with this method[29]. More details of this algorithm are given in section 3.2.1.

2.3.2. A known support

Since forming an intensity image with a lens is not preferred, another idea of adding constraint onto the specimen was suggested by Fienup[41,42], which is given a known boundary on the illuminated area. In this set-up, the specimen is covered with a mask, which has a transparent centre and opaque edges. This mask, also known as a support ($\mathbf{S}_{\vec{r}}$), only allows illumination that passes through its centre to contact the specimen; hence it forms an exit

wave whose edges are completely dark. A demonstration of this set-up is shown in *Figure 2. 6*.

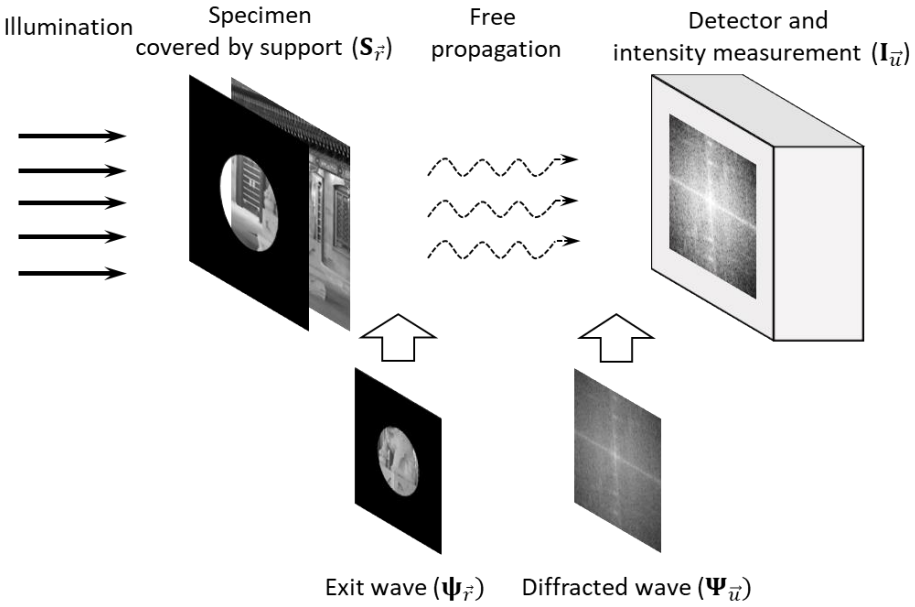


Figure 2. 6. The set-up of involving support constraint to lens-less imaging. From left to right, a coherent light source illuminates a specimen through a mask (i.e. the support ($\boldsymbol{S}_{\vec{r}}$)). The exit wave ($\boldsymbol{\psi}_{\vec{r}}$) propagates freely and turns into a diffracted wave ($\boldsymbol{\Psi}_{\vec{u}}$) at far field. Finally, the intensity ($\boldsymbol{I}_{\vec{u}}$) of this diffracted wave is measured by a detector at downstream. All the pictures showing above is only the intensity images.

Due to the existence of this support, a set of exit waves that have pixel values equal zero outside support area is formed. This set is named as support set ($\mathbb{S}$) and expressed as *eq 2. 7*. With such a constraint, the retrieving process is looking for a phase that can inversely propagate an exit wave whose pixels outside the support all equals zero. This ***support constraint*** is also a kind of ***s-domain constrain***, which can be expressed by following equation:

$$\boldsymbol{\psi}_{\vec{r}} = 0, \qquad if\ \vec{r} \notin \mathbb{S} \qquad \textit{eq 2. 7}$$

This support constraint can be applied without adjusting set-up during collecting data. It also completely removes lenses from the imaging system. However, this constraint also has some drawbacks. First, making a support with sharp edges and a known area is not an easy task. Diffraction pattern is usually observed at the edges of support due to its physical thickness[43]. This phenomenon becomes more significant with the decrease of wavelength[13]. One way to

solve this problem is soften the constraint in the beginning[44]. In another words, a support with larger diameter is applied as an initial guess and its size is gradually reduced as the reconstruction proceeds. Second, a centrosymmetric support can lead to twin images effect[45]. Such an inherent ambiguity is caused by the characteristic of Fourier transformation, as the flipped conjugate of a centrosymmetric function provides the same energy distribution in the frequency domain as the original one. Moreover, as the Fourier transformation is a linear operator, all linear combinations of these two functions satisfy the f-domain constraint. The only way of preventing this effect is to avoid using a centrosymmetric support, which increase the difficulty of guessing the support function. Third, the application of a support shrinks the observed area of the sample, so the observer has to compromise between view area and magnification[13].

2.3.3. Overlapped scanning positions

The intensity image and support constraints are utilised in phase imaging for a while, but they cannot always provide efficient and promising reconstructions[46], not to even mention their limited view area. An alternative s-constraint that circumvents these issues is latterly recommended by scanning the specimen with a localised structured illumination at multiple, partially overlapping positions: this is ptychography[47].

The device set-up for ptychography is demonstrated in *Figure 2. 7*. Starting from the left side of this figure, a coherent light passes through aperture ($\mathbf{A}_{\vec{r}}$) and lens ($\mathbf{L}_{\vec{r}}$) and forms a defocussed "probe" beam ($\widehat{\mathbf{P}}_{\vec{r}}$) on a localised region of the specimen plane. The probe is absorbed and diffracted when penetrating through the specimen, which is also known as object ($\widehat{\mathbf{O}}_{\vec{r}}$) in ptychography. At the back of the specimen, this wave is encoded by the information of both the probe and object and is named as the "k^{th} exit wave" ($\boldsymbol{\psi}_{\vec{r},k}$). Finally, the exit wave propagates over a long distance and is detected by the detector as the k^{th} diffraction pattern ($\mathbf{I}_{\vec{u},k}$).

Once a diffraction pattern is collected, a relative movement ($\vec{r}_{k+1}$) is introduced between the object and probe, indicated by the red arrows in *Figure 2. 7*. Hence a new diffraction pattern is obtained. This process is repeated until the whole area of interest has been scanned through by the probe. The relative movement can be achieved either by shifting the probe or shifting the object, though a shifting object is more common.

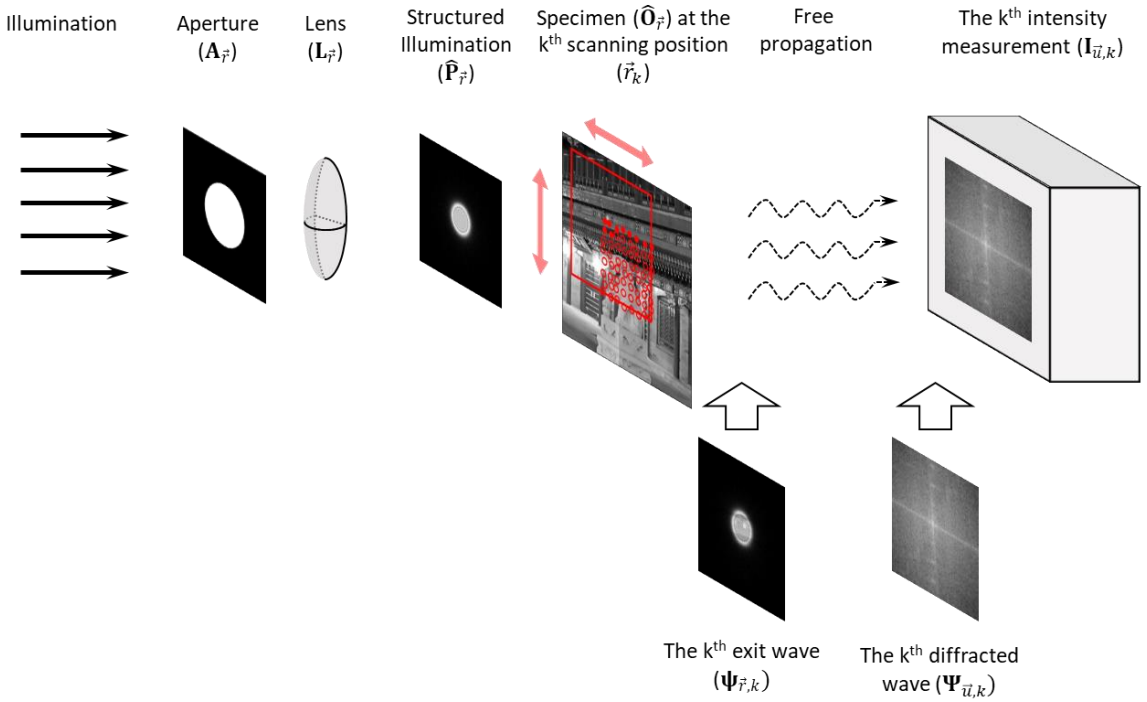


Figure 2. 7. A demonstration of ptychography with some key variables. From left to right, a coherent light passes through an aperture ($\boldsymbol{A}_{\vec{r}}$) and lens ($\boldsymbol{L}_{\vec{r}}$), then propagate through a distance forms a probe ($\widehat{\boldsymbol{P}}_{\vec{r}}$) onto the object plane ($\widehat{\boldsymbol{O}}_{\vec{r}}$). (In an experiment, the true probe is on the same plane of object. The gap between probe and object in the above picture is exaggerated to give a clear view.) The red outline on the object highlights the area contacted by the probe at the current scanning position ($\vec{r}_{k+1}$), while the red dots indicate the centre of all scanning position. Once the wave front leaves the object, it carries the information of both probe and object and named as the exit wave ($\boldsymbol{\psi}_{\vec{r},k}$). This exit wave propagates through a far field and is recorded by the detector as a diffraction pattern ($\boldsymbol{I}_{\vec{u},k}$). After that, the object is shifted to the next scanning position. The shifting directions are demonstrated by the arrows next to the object. This process is repeated until diffraction patterns at all scanning positions have been obtained.

The s-constraint in ptychography is formed by the overlapping area at different scanning positions. As one area of the specimen needs to satisfy multiple intensity measurements, it is well confined. Such a redundancy in the collected data also makes refining scan positions[48] and recover missing data[49] possible. Another advantage of ptychography is that it can achieve any desired view area by adjusting the scanning grid[47]. Due to its adjustable step size and scanning grid, ptychography is highly adaptive to various kinds of circumstance. On the other hand, ptychography also suffers problems brought by scanning positions. Although the shifting specimen is done by specifically designed platform, a rotated or scaled scanning grid still happen occasionally[50]. This sort of inaccuracy significantly increases the difficulty of reconstruction. Another potential difficulty is caused by dealing a large amount of data, which

imposes restrictions onto the reconstruction time and computation hardware. But, after all, the constraint provided by ptychography provides the most information for solving phase problem among all the s-constraints[47]. Its adaptivity and topology variations (e.g. Fourier ptychography, 3D ptychography and rotating ptychography) let it outperform other competitors and become one of the mainstreams in nowadays phase imaging.

2.3.4. Other constraints

Besides the constraints that are obtained by physically adjusting the arrangement of devices, other ***s-domain constraints*** have also served for solving phase problem for a long time. ***Nonnegativity***[29] is a constraint that can be applied to real-valued object. It forces all negative pixels of a guessed object to zero. When the intensity of certain amount of pixels of object is known, ***Histogram***[29] can be applied by replacing the value of the most brightest pixel in the current reconstruction by the known histogram values. Assuming the modulus of all pixels of object are available without knowing their corresponding coordinates, these values are sorted to produce an array, or a histogram. Hence, every time a guessed real-valued object is obtained, its pixels can be sorted based on their modulus, then replace their modulus by the corresponding histogram array values.

Atomicity[29] is a constraint requires prior knowledge of the number of non-overlapping individuals in the field of view[29]. This constraint is common in crystallography and astronomy, as the individuals are atoms or stars, which are sort of countable in advance. Applying this constraint requires defining an area that one individual can occupy. In the simplest case, each individual only occupies one pixel. Hence, for any guessed object, the pixels with the most significant values are considered as the 'present of individuals', while other pixels are replaced by zeros.

2.4. Mathematical model of ptychography

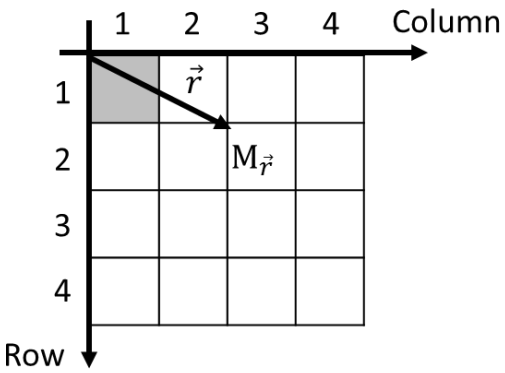
To deepen the understanding on ptychography, this section explains it with a mathematical model together with the approximations assumed and some of the possible modifications. Such a model is also the basis for the software simulations detailed in later chapters.

2.4.1. The background

To make a mathematical model for ptychography, there are three questions that need to be answered: how the involved variables are expressed, how their units are related to metric units in the physical model and what kinds of operators are applied between them.

The first step is representing these variables mathematically. Most of the variables shown in *Figure 2. 7* can be considered as images contained in a certain area and have both modulus and phase information, hence can be expressed as matrices filled with complex numbers. The size of matrix is explicitly determined by the detector. For instance, if the sensors of detector are binned into a structure of 512×512 pixels, then the diffraction pattern ($\mathbf{I}_{\vec{u}}$) will be recorded as a 512×512 matrix. As all variables are related to the diffraction pattern either explicitly or implicitly, they all have the same matrix size as the diffraction pattern. Therefore, nearly all the variables in ptychography appear as, for example, 512×512 matrices, filled with complex numbers. The only exception is the object ($\widehat{\mathbf{O}}_{\vec{r}}$), which has a larger size in most cases. During the simulation, the area of the object that is illuminated at a given scan position will be "cut out" and turns into a matrix with the same size as the detector size (e.g. 512x512 pixels). In this thesis all the matrix-type variables are denoted by bold capital letters.

Since all the images are expressed in matrix style, their pixels are equivalent to elements in the matrix. These elements can be referred to by setting up an upside-down Cartesian coordinate sat at the top-left corner, which in MATLAB starts from 1 rather than 0. After that, any pixel in this matrix can be referenced by a vector, which is noted by a lower-case letter with a vector hat ($\overrightarrow{\ }$). Here, $\vec{r}$ and $\vec{u}$ are utilised to distinguish vectors in real and reciprocal spaces. With this set-up, one can refer to one pixel of a matrix ($\mathbf{M}$) at the position $\vec{r}$ as $\mathrm{M}_{\vec{r}}$. The element of a matrix is noted by the same letter but not bold font. Examples are shown in *Figure 2. 8.*



For matrix $\mathbf{M}$ and referring vector $\vec{r} = (2,3)$

$$M_{\vec{r}} = \mathbf{M}(2,3)$$

Figure 2. 8. A demonstration of referring to a pixel when the image is represented as a matrix. The origin of coordinate is set at the top-left corner of this matrix. The row and column axes are labelled with indexing numbers. The first element in this setup is referred as (1,1), which is highlighted by grey colour. An example of referring to a specific pixel ($\vec{r} = (2,3)$) is shown in the picture as well.

The second problem is converting the scanning positions from metric units into pixels. Such a conversion is based on the concept of angular spectrum. In far-field ptychography, the conversion ratio (dxy) is expressed in *eq 2. 8* with unit as meter/pixel[2,39].

$$dxy = \frac{\lambda}{\theta_{span\ of\ detector}} \qquad \textit{eq 2. 8}$$

The λ is the wavelength of the illuminating wave in meters. The detector span angle is usually estimated as the ratio of detector dimension (d_{cam}) and the distance between source and detector (l_{cam}) as shown in *eq 2. 9*, which is under the assumption of small angle approximation. The subscript ' $_{cam}$' stands for 'camera', which is the detector in this context.

$$\theta_{span\ of\ detector} \approx \tan\left(\theta_{span\ of\ detector}\right) = \frac{d_{cam}}{l_{cam}} \qquad \textit{eq 2. 9}$$

Nevertheless, these two parameters might not be acquired directly and precisely. So other routines of computing the detector span angle are also common. One example is given in *Figure 2. 9*. One can derived their own equation following this logic based on the available data and the device set-up.

Scenario: calibration with heavy atom

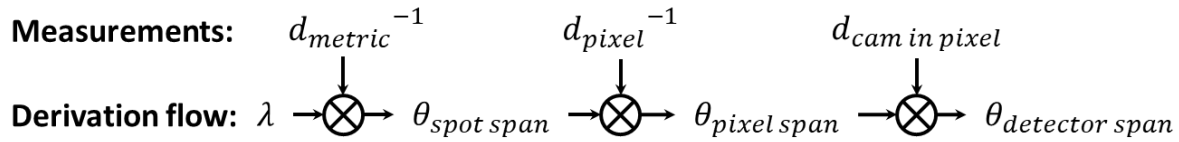


Figure 2. 9. A demonstration of an optional routine for computing the detector span angle ($\theta_{detector\ span}$). In some experiment, the detector set-up is calibrated by detecting the scattering pattern from heavy atoms, e.g. gold atoms. In this scenario, one can obtain the span angle of a picked diffraction ring (e.g. 3rd diffraction ring of gold atoms) by dividing the wavelength (λ) with the diameter of the chosen ring in metric unit (d_{metric}). Then the span angle of each pixel ($\theta_{pixel\ span}$) can be obtained by further dividing this spot span angle by the diameter of the spot in number of pixels (d_{pixel}). Finally, a multiplication of the pixel span angle and the number of pixels along the side of detector gives the detector span angle ($\theta_{detector\ span}$).

The scanning position hence can be converted from metric unit to the pixel unit by dividing the dxy. Since only the relative positions matter during reconstruction, an offset is often adapted to the scanning grid to minimise the unscanned area in the reconstructed object. Such an adjustment is helpful for reducing the memory occupation and computing time. A typical converting formula is given as *eq 2. 10*. This equation is prone to fractional pixel positions, which is normally solved by rounding its outcomes to the nearest integers.

$$position_{pixel} = \frac{position_{metric}}{dxy} + offset \qquad \textit{eq 2. 10}$$

The last problem is the operators between these matrix-style variables. Most of the operators act pixel-wise (e.g. summation, multiplication, and division etc.), hence variables related by them have the same size. All operators are noted in curly bold letters in this thesis.

As one of the important operators in this thesis, Fourier ($\boldsymbol{\mathcal{F}}$) and inverse Fourier ($\boldsymbol{\mathcal{F}^{-1}}$) transformation in this thesis imply their 2-dimensional application as shown in *eq 2. 11* and *eq 2. 12* respectively, which are normally done by DFT as a more efficient fashion[24]. These transformations are not elementwise operation, though they produce outputs that have the same size as the inputs.

$$\begin{aligned}\boldsymbol{\Psi}_{\vec{u}} &= \boldsymbol{\mathcal{F}}(\boldsymbol{\psi}_{\vec{\mathrm{r}}}) \\ &= \iint \boldsymbol{\psi}_{\vec{\mathrm{r}}} \exp\left(-\frac{j2\pi u_1 r_1}{M}\right) \exp\left(-\frac{j2\pi u_2 r_2}{N}\right) dr_1 dr_2\end{aligned} \qquad \textit{eq 2. 11}$$

$$\boldsymbol{\psi}_{\vec{r}} = \boldsymbol{\mathcal{F}}^{-1}(\boldsymbol{\Psi}_{\vec{u}}) = \frac{1}{N^2}\iint \boldsymbol{\Psi}_{\vec{u}} \exp\left(-\frac{j2\pi u_1 r_1}{M}\right) \exp\left(\frac{j2\pi u_2 r_2}{N}\right) du_1 du_2 \qquad \textit{eq 2. 12}$$

There are two special operators that return an output with different size: the ***Cut*** and ***Add***. These two operators are utilised to relate the object to the rest variables. Since the object ($\widehat{\mathbf{O}}_{\vec{r}}$) usually has a larger size than others, the illuminated area at each scan position needs to be specified before performing any element-wise operation. The ***Cut*** operator extracts the illuminated area of object at a given scanning position, while the ***Add*** operator add a given matrix to the illuminated area. Their pseudo codes are given in ***Pseudocode 2. 1*** and ***Pseudocode 2. 2***.

Pseudocode 2. 1: *Cut out part of matrix (**Cut**)*

Input:	The matrix ($matrix$), the coordinate of the centre of chosen area ($[r_1, r_2]$), the size of chosen area ($[row, col]$)
Output:	A chosen part of matrix ($part$)
Format:	$part = \boldsymbol{Cut}(matrix,\ [r_1, r_2],\ [row, col])$

1:	$row\ range = \left(r_1 - \frac{row}{2}\right) : \left(r_1 + \frac{row}{2} - 1\right)$
2:	$col\ range = \left(r_2 - \frac{col}{2}\right) : \left(r_2 + \frac{col}{2} - 1\right)$
3:	$part = matrix(row\ range, col\ range)$

Pseudocode 2. 2: *Add a smaller matrix to a part of larger one (**Add**)*

Input:	A large matrix ($matrix$), a smaller matrix ($part$), the coordinate of the centre of chosen area ($[r_1, r_2]$), the size of the chosen area ($[row, col]$)
Output:	The summed matrix ($matrix$)
Format:	$matrix = \boldsymbol{Add}(matrix,\ part,\ [x, y],\ [row, column])$

1:	$row\ range = \left(r_1 - \frac{row}{2}\right) : \left(r_1 + \frac{row}{2} - 1\right)$
2:	$col\ range = \left(r_2 - \frac{col}{2}\right) : \left(r_2 + \frac{col}{2} - 1\right)$
3:	$matrix(row\ range, col\ range) = matrix(row\ range, col\ range) + part$

2.4.2. The mathematical model

With above conventions, ptychography can be transferred into a mathematical model. To make it clear, this process is explained in two separate stages: the formation of probe and the formation of diffraction pattern.

There are many ways to form the probe ($\widehat{\mathbf{P}}_{\vec{r}}$) in ptychography, but the most common method focuses a light source using a lens. Modelling the formation of such a probe starts from a coherent light source, which is simulated as matrix filling with constants. This light source passes through an aperture ($\mathbf{A}_{\vec{r}}$), which is expressed as matrix. Its central part is transparent, expressed as 1, while its dark edges are expressed as 0. Hence any wave front can pass freely from the centre but be stopped outside that area. This process is expressed as a multiplication between the light source and aperture. Following the aperture is a condensing lens to form a localised illumination. This step adds a quadratic phase to the passing wave, which is also simulated as a multiplication. Finally, this wave propagates and turns into a probe, which is approximated by a Fourier transformation. This process is demonstrated with 4x4 matrices as a simplified example in *Figure 2. 10.*

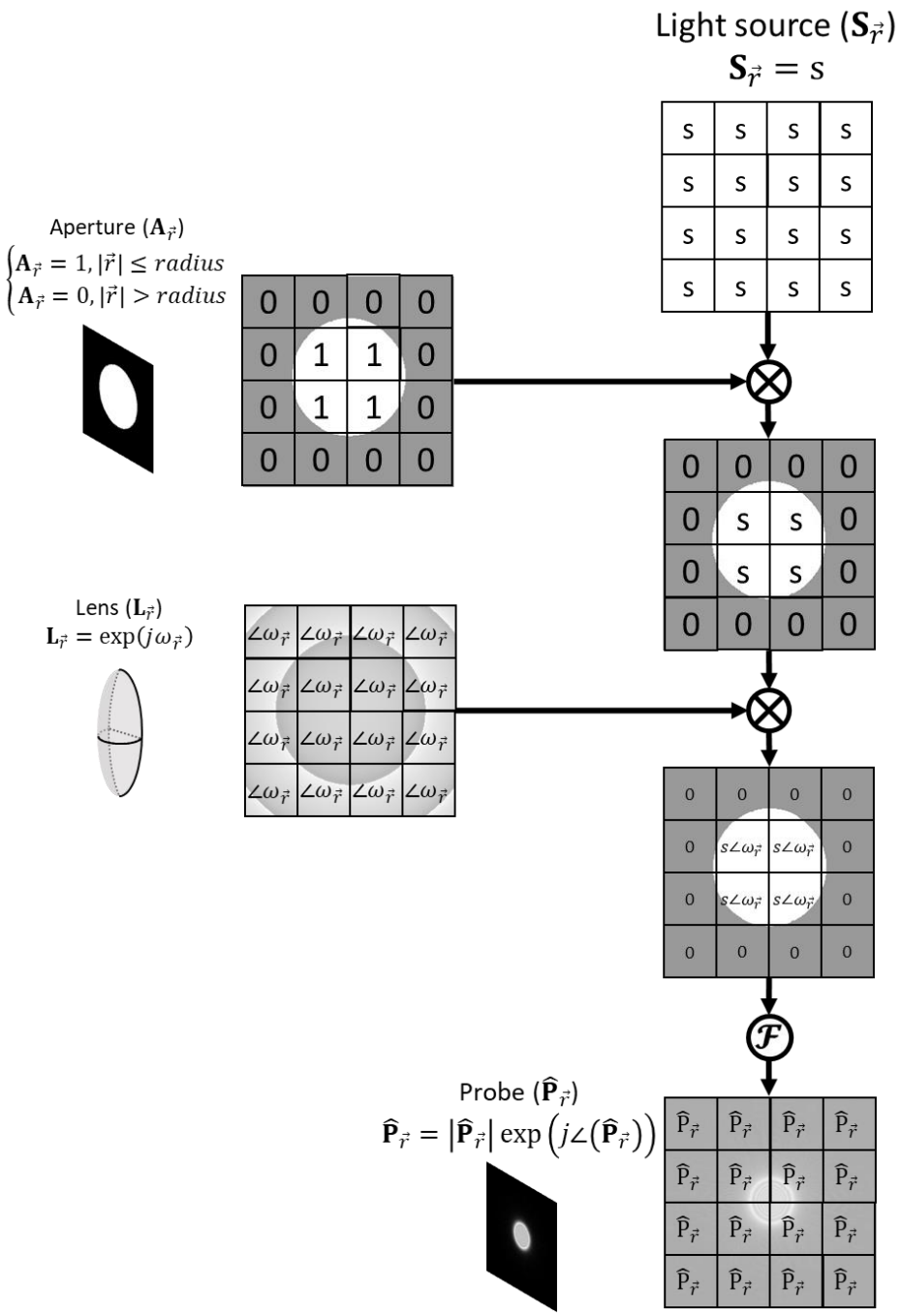


Figure 2. 10. A mathematical model of the formation of probe with 4x4 matrices as an example. To visualise the content of each variable, each matrix is combined with a corresponding picture in its background. All matrices use the modulus image as background picture, except the lens. Since the modulus of lens equals one everywhere, its phase image is chosen as background picture instead. $s\angle\omega_{\vec{r}}$ is a shorthand for $s \cdot exp(j\omega_{\vec{r}})$. As the aperture and lens are not in the same space as probe, their referring vectors are noted as $\vec{v}$. From top to the bottom, a coherent light source simulated as a matrix filling with constant (e.g. s) passes through an aperture ($\boldsymbol{A}_{\vec{r}}$), which is expressed as a matrix filled with 0s and 1s. When the passing light is converged by lens ($\boldsymbol{L}_{\vec{r}}$), a phase information is added. Finally, a localised illumination, so called "probe ($\widehat{\boldsymbol{P}}_{\vec{r}}$)", is formed after a far-field propagation, which is simulated by a 2D Fourier transformation.

This probe ($\widehat{\mathbf{P}}_{\vec{r}}$) falls onto a specimen ($\widehat{\mathbf{O}}_{\vec{r}}$) and is absorbed and diffracted during penetration. The interaction between the probe and object can be simulated as a multiplication, as long as the depth of field does not go beyond the thickness limit[51], which is defined as follows:

$$\delta = 4.88 \frac{{d_{pixel}}^2}{\lambda} \qquad \textit{eq 2. 13}$$

In order to performing elementwise multiplication, a part of object ($\widehat{\mathbf{O}}_{\vec{r},k}$) with the same size as the probe is cut out from the object matrix with the ***Cut*** operator. The propagation of exit wave ($\boldsymbol{\psi}_{\vec{r},k}$) at far-field is simulated by Fourier transformation, which turns the exit wave into a diffraction pattern ($\mathbf{I}_{\vec{u},k}$) that falls onto the detector. Only the modulus of this complex matrix is maintained by the detector. This process is demonstrated in *Figure 2. 11* together with ***Pseudocode 2. 3***. A hint of minimising memory occupation by sharing a same variable between several intermediate variables is given as a note in the bottom of the pseudo code. The energy of a matrix is defined as the sum of the squared modulus of each element. Since the energy stays the same in both spatial and frequency domain, the energy of $\boldsymbol{\psi}_{\vec{r},k}$ (i.e. $Energy_k$) can be gauged by summing up all the element of $\mathbf{I}_{\vec{u},k}$. The energy of matrix as defined by *eq 2. 14* is a useful property for limiting the probe energy during the phase retrieval.

$$Energy_k = \sum_{\vec{r}} \left|\boldsymbol{\psi}_{\vec{r},k}\right|^2 = \sum_{\vec{u}} \mathbf{I}_{\vec{u},k} \qquad \textit{eq 2. 14}$$

Pseudocode 2. 3: *The formation of diffraction pattern*

Input: specimen ($object$), illumination ($probe$) and scanning positions ($positions$)

Output: Measured intensities ($intensities$)

For (k=1: total number of $positions$) **do**

  $the\ k_{th}\ part = \boldsymbol{Cut}(object,\ the\ k_{th}\ position,\ size\ of\ probe)$

  $exit\ wave = the\ k_{th}\ part\ \cdot probe$

  $modulus = |\boldsymbol{\mathcal{F}}(\ exit\ wave\ )|$

  $the\ k_{th}\ intensity = modulus^2$

End(k)

Note [1]: Temporary variable: $the\ k_{th}\ part$, $exit\ wave$ and $modulus$

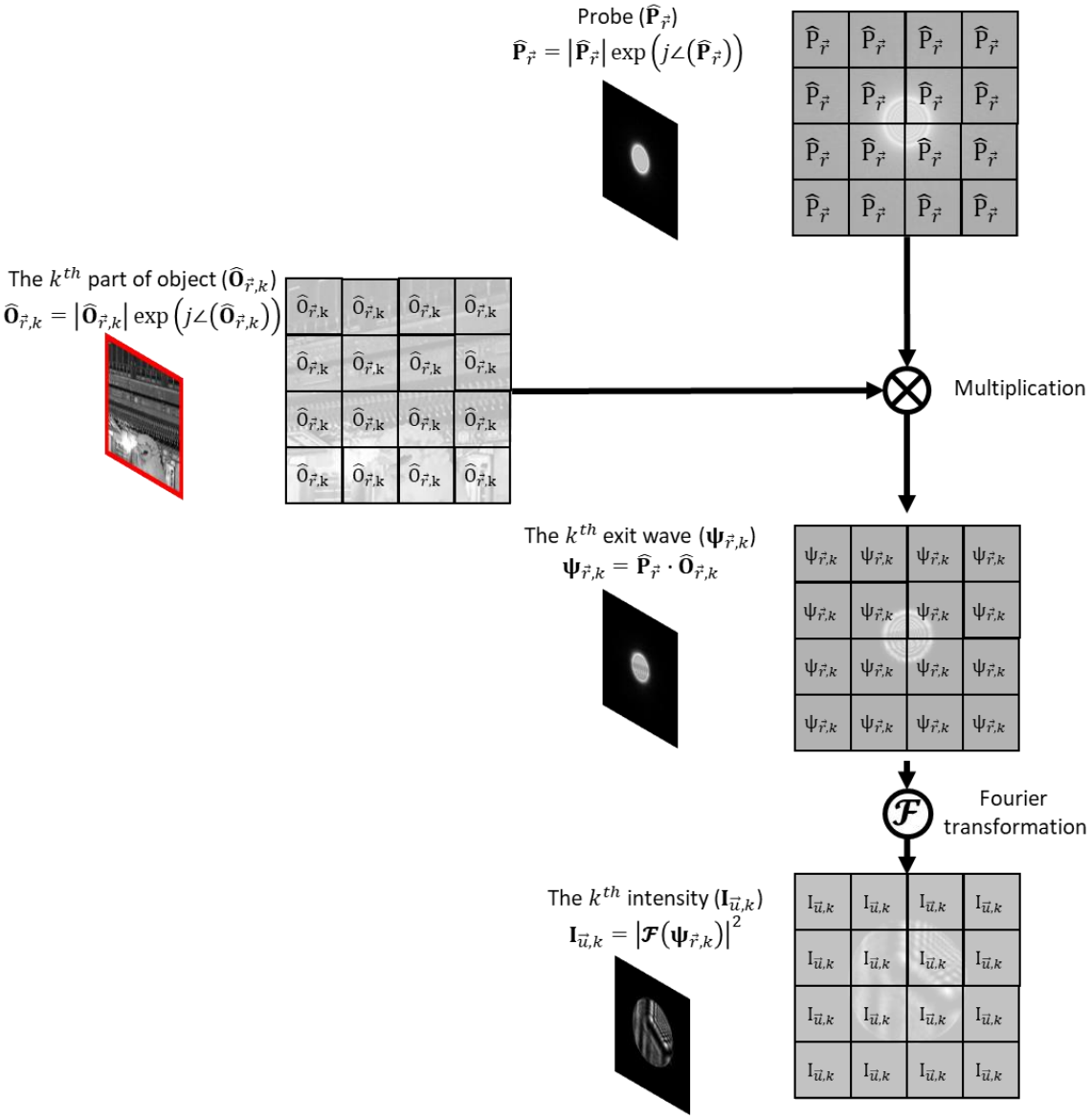


Figure 2. 11. The formation of the kth diffraction pattern with 4x4 matrices as examples. A probe ($\widehat{\boldsymbol{P}}_{\vec{r}}$) contact the k^{th} part of object ($\widehat{\boldsymbol{O}}_{\vec{r},k}$), which is the part covered at the k^{th} scanning position that is highlighted by a red outline in Figure 2. 7. These two matrices generate the kth exit wave ($\boldsymbol{\psi}_{\vec{r},k}$) by multiplication. All these variables are matrices filled with complex numbers. Finally, the kth diffraction pattern ($\boldsymbol{I}_{\vec{u},k}$) is calculated by taking the square of the modulus of the Fourier transformed exit wave. One should notice the diffraction pattern is a pure real matrix, unlike any other matrix-style variables.

2.4.3. Simulating noise

The process explained above models ptychography under ideal circumstances. However, the data collected from an experiment contains various kinds of noise. Some of them are inevitable and have influence on the outcomes. To test the robustness of different phase retrieval methods, some artificial noise can be added to the flawless data during simulation.

Since the data utilised in reconstruction is made up of the diffraction patterns and the scanning positions, the noise and error that might exist within them are explained respectively.

2.4.3.1. Detector noise in diffraction patterns

This detector noise is introduced during recording diffraction patterns. It includes the Poisson shot noise, dark current noise and readout noise. The Poisson noise is an inevitable error caused by a physical phenomenon when counting random episodes with insufficient samples. Although increasing the counts of incident particles (e.g. photons or electrons) can minimise its influence, it is not suitable for samples that can be damaged by an over-dose of illumination. The dark current noise is caused by randomly excited electrons who are stochastically triggered by the thermal energy from their surroundings. As it follows the normal distribution, it is also called Gaussian noise. The last one, readout noise, is caused by rounding-up and clamping the readings to a specific dynamic range that is limited by the design of detector.

The process of adding detector noise is shown in *Figure 2. 12*. First, a noiseless simulated diffraction pattern is divided by a detector gain (G), which is estimated as the ratio between the probe energy and the count of photons. This converts the intensity measurements into a photon matrix. Then this photon matrix is contaminated by Poisson noise and converted into an electron matrix by multiplying with the quantum efficiency (η) and rounding up. After that, a dark noise following a Gaussian distribution is added into the electron matrix. This electron matrix is scaled by the sensitivity of detector (K) and rounded to integer. Finally, a base line is added to make sure the minimum readings are higher than a specific value and any number larger than the maximum value is clamped by the bit-depth of the detector[52].

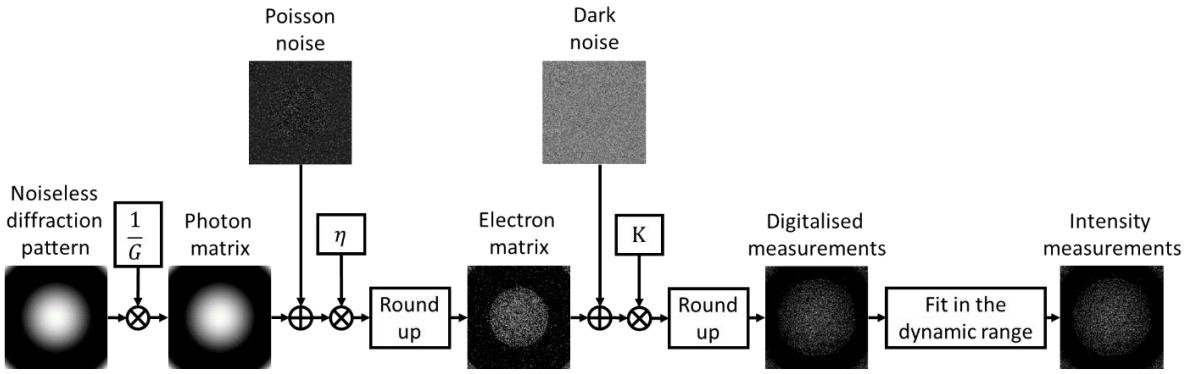


Figure 2. 12. The process of simulating detector noise. From left to right, a photon matrix is gauged by dividing the noiseless diffraction pattern with the detector gain (G). Then a Poisson noise is generated based on the noiseless photon matrix and summed up together. This combined matrix is scaled by the quantum efficiency (η) and rounded up to obtain an electron matrix. A dark noise following normal distribution is composed onto the electron matrix. After that, this noisy electron matrix is scaled by the sensitivity (K) and round up again. Finally, a read noise is introduced when converting this matrix into the final readings. Since detectors have dynamic range limited by the base line and bit-depth, a constant is added to the whole matrix to make sure its minimum value is not smaller than the base line. Then any value beyond the range of bit-depth is limited to the maximum possible value.

2.4.3.2. Errors in scanning positions

The scanning positions collected from experiment might also come with errors, which causes the actual scanning grid to deviate from the desired one. The transformation is a combination of rotation, scaling and random offsets. A rotated scanning grid is usually caused by an improperly positioned detector as shown in *Figure 2. 13*.

With a poorly calibrated detector, an angle (θ) is introduced between the scanning grid and detector. By taking the detector coordinate as the standard, one can identify the scanning positions is angled by θ. This rotation effect can be simulated by multiplying the position coordinate with a rotation matrix (**R**) as shown in *eq 2. 15* and *eq 2. 16*. The multiplication here follows the inner product rule of matrix.

$$\overrightarrow{r_k'} = \mathbf{R}(\theta) \cdot \overrightarrow{r_k} \qquad \textit{eq 2. 15}$$

$$\begin{bmatrix} r'_{k_1} \\ r'_{k_2} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} r_{k_1} \\ r_{k_2} \end{bmatrix} \qquad \textit{eq 2. 16}$$

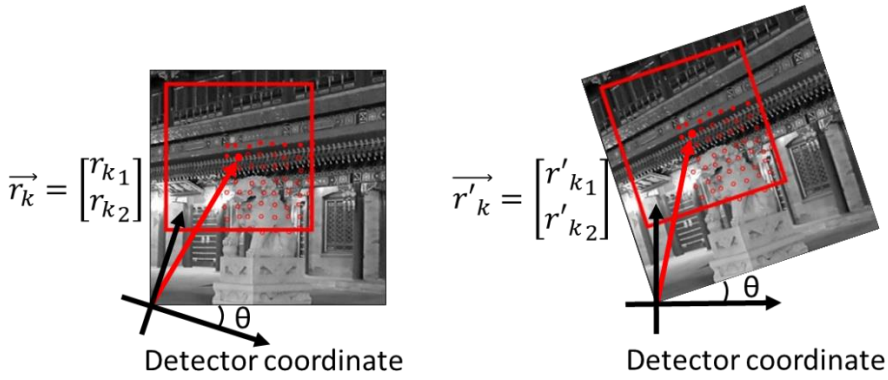


Figure 2. 13. The rotation effect caused by an angle between the scanning grid and the detector. The coordinates indicate the horizontal and vertical direction of the detector. The left picture shows that an angle (ϑ) between the coordinates of detector and scanning grid. If the detector coordinate is treated as the standard, the angled scanning positions can be calculated by multiplying the original coordinate (r_{k_1}, r_{k_2}) with a rotation matrix (R). The k^{th} scanning position ($\overrightarrow{r_k}$) is given in the picture as an example, which is highlighted by a red arrow.

Scaling is another factor affecting the actual scanning positions. A coarse measurement on the device set-up or an inaccurate shifting platform can easily lead to an inaccurate converting ratio between metric and pixels, hence a scaled scanning position. This effect can be simulated by multiplying the position coordinate with a constant as shown in *eq 2. 17* and *eq 2. 18.*

$$\overrightarrow{r'_k} = k_{scale} \cdot \overrightarrow{r_k} \qquad \textit{eq 2. 17}$$

$$\begin{bmatrix} r'_{k_2} \\ r'_{k_2} \end{bmatrix} = k_{scale} \begin{bmatrix} r_{k_1} \\ r_{k_2} \end{bmatrix} \qquad \textit{eq 2. 18}$$

Last but not the least, each scanning position could contain a random shift ($\overrightarrow{r_{\Delta,k}}$) from its desired position[48]. The final k^{th} scanning position after taking all the possible errors into account can be expressed by *eq 2. 19*.

$$\overrightarrow{r'_k} = k_{scale} \cdot \boldsymbol{R}(\theta) \cdot \overrightarrow{r_k} + \overrightarrow{r_{\Delta,k}} \qquad \textit{eq 2. 19}$$

3. Algorithms for solving phase problems

The key concepts of lens-less imaging are passing the task of forming images from lens systems to phase retrieval algorithms, hence the defects in lens will not affect the image quality. As an essential part of ptychography, phase retrieval algorithm plays an important role in recovering images from measured data and evolve together with imaging methods. In this chapter, some of the most representative algorithms in the history of lens-less imaging are explained. Although algorithms applied in early-stage imaging methods are different with those utilised in ptychography due to their different constraints, they are helpful to unveil the common logic in solving phase problem. This section begins with introducing the related background knowledge in mathematics (section 3.1) and follows by the descriptions of algorithms before (section 3.2) and after (section 3.2) the development of ptychography. Some computer background, which is related to ptychography, is given in section 3.4.

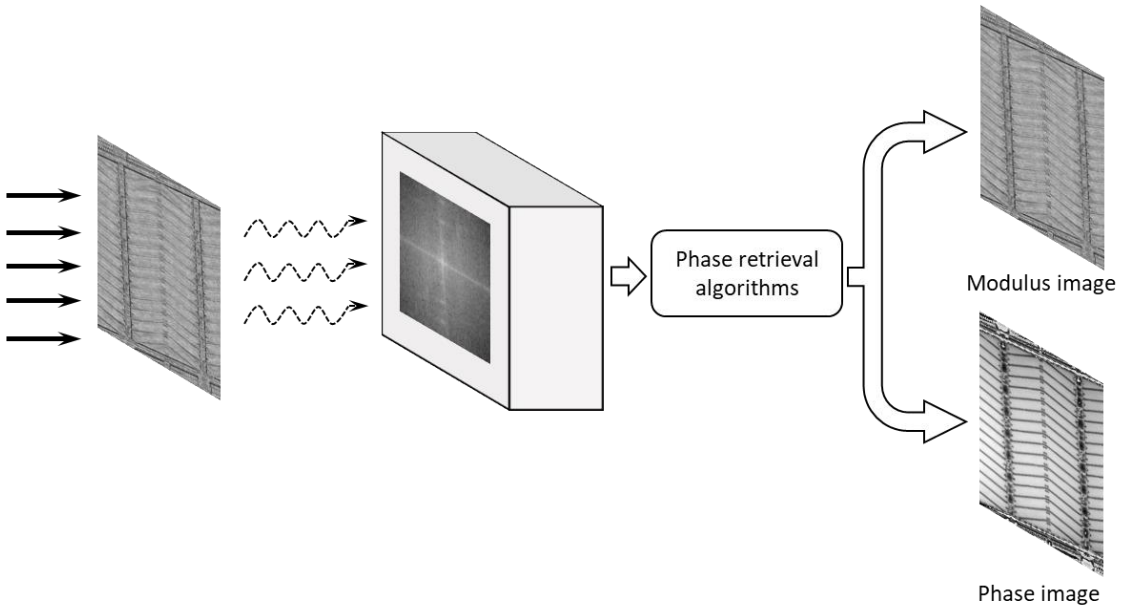


Figure 3. 1. Phase retrieval algorithms act as compulsory post-processing tools in diffraction imaging. They reconstruct both the modulus and phase images from the collected data.

3.1. Mathematics background

Due to the complexity of the phase problem, most algorithms are based on iterative optimisation rather than figuring out solutions explicitly. They all start with a reasonable guess on the unknown terms and gradually approach a solution by repeatedly modifying unknowns

with measured data. These algorithms can be separated into two categories based on their methodologies of optimisation: set-projection methods and steepest descent methods. The math knowledge of these two methodologies is explained in the following sub sections.

3.1.1. Set projection and reflection

Set projection is a widely used method for finding intersection points between several sets that represent different constraints in an optimization problem, as shown in *Figure 3. 2*. Beside the application of in phase imaging, the set-projection methods have also been tested in wide range of problems, including graph colouring, logical satisfiability, spin glass ground states, bit retrieval and sudoku[46].

In phase retrieval, each constraint forms a set and their intersection points are considered as solutions, as an ideal solution should satisfy all the constraints at the same time. With such a model, the set-projection method inspired many phase retrieval algorithms. Some of them is well-known for its capability of preventing stagnation. This section uses ptychography as an example, and explains how the concept of set-projection is applied in solving phase problem.

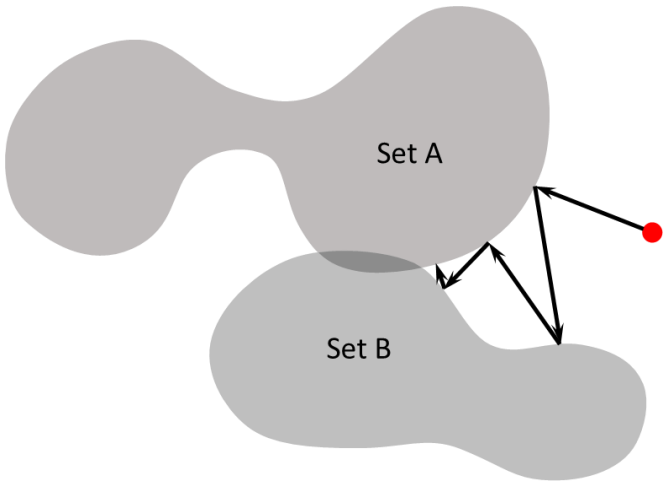


Figure 3. 2. A demonstration of how set projection method approaches a solution. There are two given sets in the above picture, i.e. set A and B. Their intersection, which is the shadow area, contains the solutions. An initial guess, which is marked by red dot, approaches the solution area by alternatively projecting between two sets. The movement of each projection is indicated by arrows

3.1.1.1. f- and s- sets in ptychography

All advantages achieved by ptychography come from two constraints: the measured intensities and the overlapping areas. As explained in Chapter 2, if the sampling grid is organised with M rows and N columns, the specimen ($\widehat{\mathbf{O}}_{\vec{r}}$) can be expressed as a $M \times N$ complex matrix. In ptychography, K diffraction patterns ($\mathbf{I}_{\vec{u},k}$) are obtained by shining illuminations onto this specimen with K diverse positions, each diffraction pattern is expressed as a $M \times N$ matrix with reciprocal sampling rate in *f-domain*. The illumination is denoted as $\widehat{\mathbf{P}}_{\vec{r}}$, while the specimen with *K* diverse translations is denoted as $\widehat{\mathbf{O}}_{\vec{r},k} \in \mathbb{C}^D$, $D = M \times N \times K$. Such a relationship is expressed by *eq 3. 1*, where the subscript $\overrightarrow{r_k}$ represents the relative shift.

$$\mathbf{I}_{\vec{u},k} = \left|\boldsymbol{\mathcal{F}}\left(\widehat{\mathbf{O}}_{\vec{r},k} \cdot \widehat{\mathbf{P}}_{\vec{r}}\right)\right|^2 \qquad \textit{eq 3. 1}$$

Therefore, all guessed objects ($\mathbf{O}_{\vec{r}}$) and probes ($\mathbf{P}_{(\vec{r}-\overrightarrow{r_k})}$) satisfying the intensity measurements form a modulus set ($\mathbb{M}$), which is expressed by *eq 3. 2*. This set constrains the moduli of the set of viable exit waves in reciprocal space, which is also referred to as the *f-domain*. Hence, the set $\mathbb{M}$ is also referred as the ***f-constraint*** in the thesis.

$$\left\{\mathbb{M} \subseteq \mathbb{C}^D : \left|\boldsymbol{\mathcal{F}}\left(\mathbf{O}_{\vec{r},k} \cdot \mathbf{P}_{\vec{r}}\right)\right| = \sqrt{\mathbf{I}_{\vec{u},k}}, \qquad \forall k \epsilon [1,2,\dots,K]\right\} \qquad \textit{eq 3. 2}$$

Due to lack of phase information, the *f-constraint* alone is insufficient for phase retrieving (as explained in Chapter 2). Thus, another constraint is required, which is a constraint in the spatial domain (i.e. the ***s-constraint***). The *s-constraint* is formed by the overlapping area between different scanning positions in ptychography. When exit waves are generated from the guessed object and probe, they are sharing the overlapping area and related to each other in the spatial domain[47]. The set formed by the overlapped area is the consistency set ($\mathbb{O}$), which is expressed by *eq 3. 3*. This consistency set is due to the fact that the content of specimen is not changed during the recording of the diffraction patterns. Therefore, a successful reconstruction should produce $\mathbf{O}_{\vec{r}_k} = \mathbf{O}_{\vec{r}_{fixed}}, \forall k \epsilon [1,2,\dots,K]$.

$$\left\{\mathbb{O} \subseteq \mathbb{C}^D : \mathbf{O}_{\vec{r}_k} = \mathbf{O}_{\vec{r}_l}, \qquad \forall k, l \epsilon [1,2,\dots,K]\right\} \qquad \textit{eq 3. 3}$$

With these two sets, the lost phase can be iteratively retrieved by propagating the guessed object and probe between *s*- and *f*- domains and applying these constraints alternatively. Such a flow is demonstrated in *Figure 3. 3*. The process of manipulating exit waves with respect to these constraints is known as projection and reflection[46,53], which are explained in the following sections.

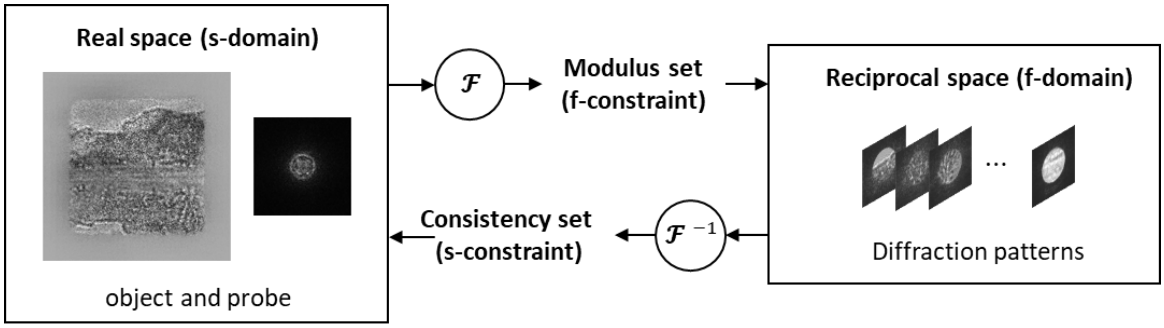


Figure 3. 3. The flow chart of updating object and probe with the f- and s-constraint. Starting from the left, the consistency of overlapping area forms the s-constraint. A group of guessed exit waves are generated by taking part of object out and multiplying with probe. These exit waves satisfy the s-constraint. After that, they are revised by the f-constraint and turn into the revised exit waves. Eventually, these revised exit waves are utilised to update the guessed object and probe.

3.1.1.2. Projection ($\mathcal{P}$)

Projecting a vector onto a set is equivalent to finding a new vector that lies within the set, which has a minimum "distance" from the original vector[29,46,54]. To give a mathematical description, we start from the weighted inner product in a complex vector space $\mathbb{C}^D$:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\boldsymbol{\omega}} = \sum_{d=1}^{D} \omega_d x_d {y_d}^* \qquad \textit{eq 3. 4}$$

Where the subscript $\boldsymbol{\omega}$ is a vector of weights in space $\mathbb{C}^D$, hence each dimension of this complex space is correspondingly weighted by $\omega_{\boldsymbol{d}} \in \mathbb{R}^+$. Therefore, $\langle \square, \square \rangle_\omega$ represents the inner product weighted by $\boldsymbol{\omega}$ and in particular $\langle \square, \square \rangle_{\mathbf{1}}$ is the conventional complex Euclidean inner product. A weighted norm can be expressed by *eq 3. 5*.

$$\|\boldsymbol{x}\|^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\boldsymbol{\omega}} = \sum_{d=1}^{D} \omega_d |x_d|^2 \qquad \textit{eq 3. 5}$$

Meanwhile, the related distance metric between two vectors, $\boldsymbol{x}$ and $\boldsymbol{y}$ in a complex space $D = M \times N \times K$, is defined as *eq 3. 6*.

$$d_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_{\boldsymbol{\omega}} = \sqrt{\sum_{d=1}^{D} \omega_d |x_d - y_d|^2} \qquad \textit{eq 3. 6}$$

For $\boldsymbol{\omega} = \mathbf{1}$, the above norm (*eq 3. 5*) and distance (*eq 3. 6*) expressions are their Euclidean equivalents. The distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ along each dimension of $\mathbb{C}^D$ can be weighted unequally by assigning different values to different dimensions of $\boldsymbol{\omega}$. To be more specific, a large ω_d magnifies the distance between x_d and y_d, while a small ω_d shrinks their distance. Such an influence is illustrated in *Figure 3. 4*. Based on this definition of distance, projecting a vector ($\boldsymbol{x}$) to a set ($\mathbb{M}$) can be expressed as *eq 3. 7*.

$$\mathcal{P}_{\mathbb{M}}(\boldsymbol{x}) = \arg\min_{\boldsymbol{z} \in \mathbb{M}} d_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{z}) \qquad \textit{eq 3. 7}$$

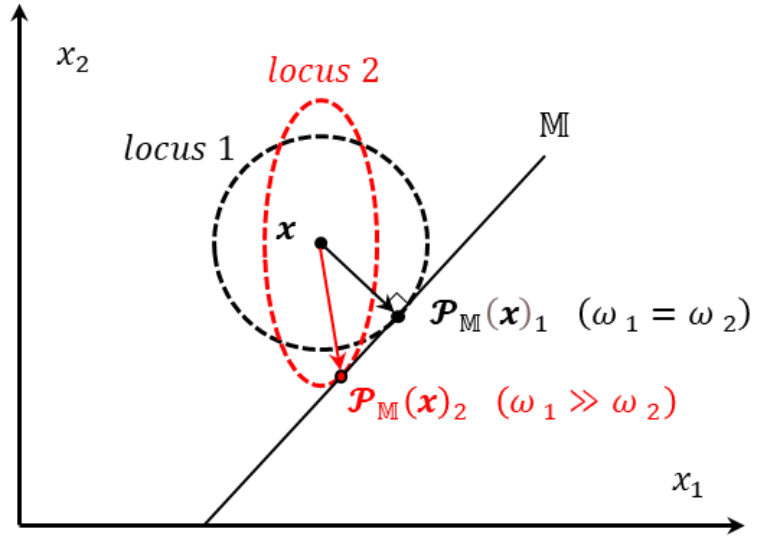


Figure 3. 4. A 2D example of how the $\boldsymbol{\omega}$ affects projections. In this example, vector $\boldsymbol{x}$ is projected onto the set $\mathbb{M}$ under the influence of $\boldsymbol{\omega} = (\omega_1, \omega_2)$. 2 equal-distance loci from vector $\boldsymbol{x}$ (locus 1 and 2) under different circumstance are shown as dotted lines. The corresponding projection points are marked as $z_1$and z_2. When ω_1 and ω_2 are equal, the horizontal and vertical dimensions are equally weighted. Hence a circle locus is formed as locus 1, which touches set $\mathbb{M}$ at point z_1. When ω_1 is much larger than ω_2, the variation along the horizontal dimension has more influence on the distance. Hence the locus has less horizontal variation, and acts as a horizontally squeezed eclipse that touches set $\mathbb{M}$ at z_2.

- **f-constraint: intensity measurement of diffraction pattern**

Projection with respect to the *f-constraint* (i.e. the measured intensities) is a common operator in diffractive imaging. To perform the *f-projection* ($\boldsymbol{\mathcal{P}}_f$), the exit wave ($\boldsymbol{\psi}_{\vec{r}}$) is transformed into the reciprocal space (*f-domain*) by Fourier transformation. Since this exit wave is calculated based on the guessed object and probe, neither of its amplitude and phase is correct. Based on the definition of distance, projecting a guessed exit wave ($\boldsymbol{\Psi}_{\vec{u}}$) to the f-constraint is done by replacing its modulus with the square root of the measured intensities($\mathbf{I}_{\vec{u}}$), while leaving its phase unchanged. The process of replacing modulus in *f-domain* is noted as $\boldsymbol{\varphi}_f$ and demonstrated by *eq 3. 8*. A simplified demonstration of its geometry meaning is shown in *Figure 3. 5*.

$$\boldsymbol{\Psi}_{\vec{u}}{}' = \boldsymbol{\varphi}_f(\boldsymbol{\Psi}_{\vec{u}}) = \sqrt{\mathbf{I}_{\vec{u}}}\frac{\boldsymbol{\Psi}_{\vec{u}}}{|\boldsymbol{\Psi}_{\vec{u}}|} \qquad \textit{eq 3. 8}$$

This is a 'distance-minimising' operation and can be proved with the help of *Figure 3. 5*. For a single pixel of $\boldsymbol{\Psi}_{\vec{u}}$ at position $d\epsilon D$, its value can be expressed in polar coordinate form as:

$$\Psi_{\vec{u}} = |\Psi_{\vec{u}}|\angle\tau_{\vec{u}} \qquad \textit{eq 3. 9}$$

Where $\tau_{\vec{u}}$ stands for its phase angle. Similarly, a pixel that satisfies the measured intensity at the same pixel position (i.e. $\Psi_{\vec{u}}{}'$) must have modulus equals $\sqrt{\mathrm{I}_{\vec{u}}}$, while its phase angle is unknown. We express the phase difference between the $\Psi_{\vec{u}}$ and $\Psi_{\vec{u}}{}'$ by $\Delta\tau_{\vec{u}}$, and get:

$$\Psi_{\vec{u}}{}' = \sqrt{\mathrm{I}_{\vec{u}}}\angle(\tau_{\vec{u}} + \Delta\tau_{\vec{u}}) \qquad \textit{eq 3. 10}$$

Hence their complex Euclidean distance can be expressed as:

$$d\left(\Psi_{\vec{u}}, \Psi'_{\vec{u}}\right) = \sqrt{|\Psi_{\vec{u}}|^{2} + \mathrm{I}_{\vec{u}} - 2|\Psi_{\vec{u}}|\sqrt{\mathrm{I}_{\vec{u}}}\cos(\Delta\tau_{\vec{u}})} \qquad \textit{eq 3. 11}$$

Which achieves the minimum value when $\Delta\tau_{\vec{u}} = 0$. In other word, the distance reaches the minimum value when the pixels of guessed and revised exit waves have the same phase, which is exactly the effect of $\boldsymbol{\varphi}_f$. Since each pixel is fitted to the constraint with the minimum variation, the complex Euclidean distance between $\boldsymbol{\Psi}_{\vec{u}}$ and $\boldsymbol{\Psi}_{\vec{u}}{}'$ is also minimised. Hence the

operator $\boldsymbol{\mathcal{p}}_f$, which makes the Fourier transformed guessed exit wave satisfies its *f-constraint* with minimum adjustment, is a '*f-projection*' operator.

After applying the *f-projection* ($\boldsymbol{\mathcal{p}}_f$), this revised exit waves ($\boldsymbol{\Psi}_{\vec{u}}^{'}$) are transformed back to the real space (*s-domain*) with inverse transformation. This whole procedure is combined as $\boldsymbol{\mathcal{P}}_f$ in *eq 3. 12* for the sake of simplicity. A pseudo code of *f-projection* is given in **Pseudocode 3. *1***.

$$\boldsymbol{\psi}'_{\vec{r}} = \boldsymbol{\mathcal{F}}^{-1}\left(\boldsymbol{\mathcal{p}}_f\left(\boldsymbol{\mathcal{F}}(\boldsymbol{\psi}_{\vec{r}})\right)\right) = \boldsymbol{\mathcal{P}}_{\boldsymbol{f}}(\boldsymbol{\psi}_{\vec{r}}) \qquad \textit{eq 3. 12}$$

Pseudocode 3. 1: *The f-projection ($\boldsymbol{\mathcal{P}}_f$)*

Input: guessed exit wave ($exit\ wave$), measured diffraction pattern ($intensity$)

Output: revised exit wave ($revised\ exit\ wave$)

Format: $revised\ exit\ wave = \boldsymbol{\mathcal{P}}_f(exit\ wave, intensity)$

1:	$FT\ exit\ wave = \boldsymbol{\mathcal{F}}\ (exit\ wave)$
2:	$FT\ exit\ wave = \sqrt{intensity} \cdot \exp\left(j \cdot angle(FT\ exit\ wave)\right)$
3:	$revised\ exit\ wave = \boldsymbol{\mathcal{F}}^{-1}(FT\ exit\ wave)$

Note [1]: Temporary variable: $FT\ exit\ wave$

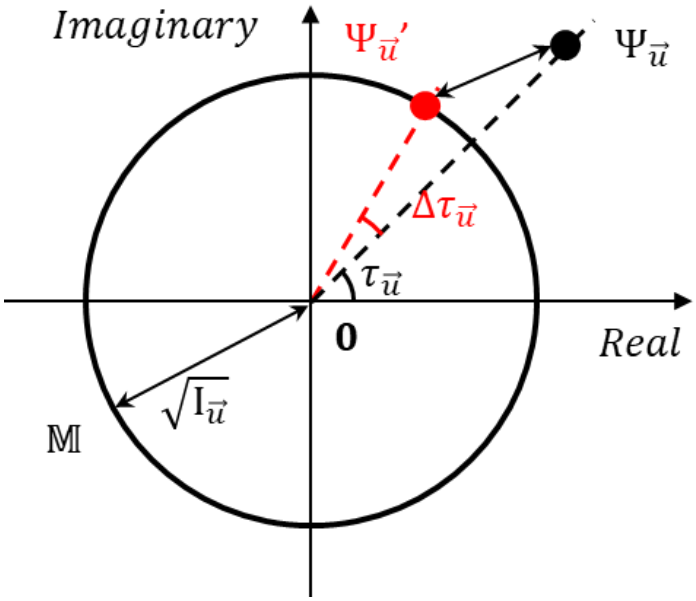


Figure 3. 5. A demonstration of f-projection for a single pixel of $\boldsymbol{\Psi}_{\vec{u}}$ at position $d\epsilon D$ in $f-domain$. The pixel value of Fourier transformed guessed exit wave is expressed with polar coordinate as $\Psi_{\vec{u}} = |\Psi_{\vec{u}}|\angle\tau_{\vec{u}}$ and denoted as a black dot in the figure. The constraint formed by the measured intensity at the same pixel is expressed as a circle with radius equals $\sqrt{I_{\vec{u}}}$. Any pixel that satisfies f-constraint must sits on this circle, though its phase angle may vary. An example of pixel fitting f-constraint is highlighted by a red dot, whose coordinate is $\Psi_{\vec{u}}' = \sqrt{I_{\vec{u}}}\,\angle(\tau_{\vec{u}} + \Delta\tau_{\vec{u}})$.

Unlike the *f-constraint*, the *s-constraint* used by different diffractive imaging methods varies from one to another. Referring to the order in Chapter 2, the approaches of applying these s-constraint are explained one by one.

- **s-constraint #1: known image intensity**

When the image intensity of specimen ($\mathbf{I}_{\boldsymbol{img}_{\vec{r}}}$) is known, projecting the guessed image (i.e. $\mathbf{O}_{\vec{r}}$ in this case) to the s-constraint is similar to the *f-projection*, which is replacing the modulus by the square root of measured image intensity while leaving its phase unchanged as shown by *eq 3. 13*. Its pseudocode is given in ***Pseudocode 3. 2***.

$$\mathbf{O}'_{\vec{r}} = \boldsymbol{\mathcal{P}}_{\mathrm{img}}(\mathbf{O}_{\vec{r}}) = \sqrt{\mathbf{I}_{\boldsymbol{img}_{\vec{r}}}}\,\frac{\mathbf{O}_{\vec{r}}}{|\mathbf{O}_{\vec{r}}|} \qquad \textit{eq 3. 13}$$

Pseudocode 3. 2: *The s-projection for image intensity constraint ($\boldsymbol{\mathcal{P}}_{img}$)*

Input: guessed object ($object$), measured image intensity ($image\ intensity$)

Output: revised object ($revised\ object$)

Format: $revised\ exit\ wave = \boldsymbol{\mathcal{P}}_{img}(exit\ wave, support)$

1: $revised\ object = \sqrt{image\ intensity} \cdot angle(object)$

- **s-constraint #2: known support**

Another common s-constraint is the area defined by a support ($\mathbb{S}$). Projecting to this kind of constraint forces all pixels beyond support area to zero, while leave other pixels unchanged. This process is shown by *eq 3. 14* and ***Pseudocode 3. 3***.

$$\boldsymbol{\psi}_{\vec{r}}{}' = \begin{cases} 0 & \vec{r} \notin \mathbb{S} \\ \boldsymbol{\psi}_{\vec{r}} & \vec{r} \in \mathbb{S} \end{cases} \qquad \textit{eq 3. 14}$$

Pseudocode 3. 3**:** The s-projection for support constraint ($\boldsymbol{\mathcal{P}}_{support}$)

Input: guessed exit wave ($exit\ wave$), support ($support$)

Output: revised exit wave ($revised\ exit\ wave$)

Format: $revised\ exit\ wave = \boldsymbol{\mathcal{P}}_{support}(exit\ wave, support)$

```
1: For (r⃗=all pixels of exit wave) do
2:     If support(r⃗) == 0
3:         revised exit wave(r⃗) = 0
4:     Else
5:         revised exit wave(r⃗) = exit wave(r⃗)
6:     End
7: End
```

- **s-constraint #3: object consistency**

In ptychography, the derivation of projection with respect to the consistency set ($\mathbb{O}$) requires another characteristic of projection, which is that the variation caused by projection to a set

($\boldsymbol{\mathcal{P}_s}(\boldsymbol{x}) - \boldsymbol{x}$) should be orthogonal to any vector ($\boldsymbol{y}$) that belongs to that set ($\mathbb{O}$)[54]. This relationship is usually expressed by inner product as shown in *eq 3. 15*.

$$\langle \boldsymbol{\mathcal{P}_s}(\boldsymbol{x}) - \boldsymbol{x}, \boldsymbol{y} \rangle_{\boldsymbol{\omega}} = \sum_{d=1}^{D} \omega_d \cdot (\boldsymbol{\mathcal{P}_s}(x_d) - x_d) \cdot y_d^* = 0, \qquad \forall \boldsymbol{y} \in \mathbb{O} \qquad \textit{eq 3. 15}$$

Similar to *eq 3. 6*, a weighting factor $\boldsymbol{\omega}_k$ is involved to assign different weights to different dimension of the vector space. Combining with the definition of the consistency set, this equation can be rewritten as *eq 3. 16*.

$$\sum_{k=1}^{K} \omega_k \cdot \left(\boldsymbol{\mathcal{P}_s}\left(\mathbf{O}_{\vec{r},k}\right) - \mathbf{O}_{\vec{r},k}\right) \cdot {\mathbf{O}_{\vec{r}_{fixed}}}^* = 0, \qquad \forall \boldsymbol{y} \in \mathbb{O} \qquad \textit{eq 3. 16}$$

As the ${\mathbf{O}_{\vec{r}_{fixed}}}^*$ and $\boldsymbol{\mathcal{P}_s}\left(\mathbf{O}_{\vec{r},k}\right)$ does not vary with k, it can be moved out from the summation. With the further derivation shown below, the projection with respect to the consistence set (i.e. s-constraint) can be found as eq 3. 17.

$$\begin{aligned}
{\mathbf{O}_{\vec{r}_{fixed}}}^* \cdot \sum_{k=1}^{K} \omega_k \cdot \left(\boldsymbol{\mathcal{P}_s}\left(\mathbf{O}_{\vec{r},k}\right) - \mathbf{O}_{\vec{r},k}\right) &= 0 \\
\sum_{k=1}^{K} \omega_k \cdot \boldsymbol{\mathcal{P}_s}\left(\mathbf{O}_{\vec{r},k}\right) - \sum_{k=1}^{K} \omega_k \cdot \mathbf{O}_{\vec{r},k} &= 0 \\
\boldsymbol{\mathcal{P}_s}(\mathbf{O}_{\vec{r}}) \cdot \sum_{k=1}^{K} \omega_k &= \sum_{k=1}^{K} \omega_k \cdot \mathbf{O}_{\vec{r},k} \\
\boldsymbol{\mathcal{P}_s}(\mathbf{O}_{\vec{r}}) &= \frac{\sum_{k=1}^{K} \omega_k \cdot \mathbf{O}_{\vec{r},k}}{\sum_{k=1}^{K} \omega_k}
\end{aligned} \qquad \textit{eq 3. 17}$$

The intensity of reconstructed probes ($|\mathbf{P}_{\vec{r}}|^2$) are usually considered as the main components of weighting factor. This approach does not only relate object and probe to the exit wave (as $\mathbf{P}_{\vec{r}} \cdot \mathbf{O}_{\vec{r}} = \boldsymbol{\psi}_{\vec{r}}$), but also gives a reasonable physical meaning to the weighting factor. As a larger ω_k, which is brighter part of the probe, often implies a better illumination hence higher signal to noise ratio and more confidence on the projection outcome.

Replacing ω_k in *eq 3. 17* by $|\mathbf{P}_{\vec{r}}|^2$ and applying $|\mathbf{P}_{\vec{r}}|^2 \cdot \mathbf{O}_{\vec{r},k} = \mathbf{P}_{\vec{r}}{}^{*}\boldsymbol{\psi}_{\vec{r},k}$, a standard *s-projection* for ptychography is found as *eq 3. 21* with corresponding ***Pseudocode 3. 4***. Other projection methods also exist by using normalised probe or other weighting factors.

$$\boldsymbol{\mathcal{P}_s}(\mathbf{O}_{\vec{r}}) = \frac{\sum_{k=1}^{K} \mathbf{P}_{\vec{r}}{}^{*}\boldsymbol{\psi}_{\vec{r},k}}{\sum_{k=1}^{K}|\mathbf{P}_{\vec{r}}|^2} \qquad \textit{eq 3. 18}$$

The process of a standard s-projection is demonstrated by *Figure 3. 6*. All guessed exit waves are multiplied with the guessed illumination and added up follow the corresponding scanning positions. This summation is divided by the position-wise add-up of the intensities of the guessed illumination. This process can be considered as taking the weighted average value for the overlapping area.

With proper constraints, alternative projection gives an error that never increases during the progress[40]. This becomes the main concept of error reduction algorithm, which is proved in *Section 3.2.1*. One should notice projection is not a linear operator and be careful when moving variables around it. Last but not the least, repeat projecting to the same constraint will not affect the outcome, which is shown in *eq 3. 19*.

$$\boldsymbol{\psi}_{\vec{r}} = \boldsymbol{\mathcal{P}_f}(\boldsymbol{\psi}_{\vec{r}}), \qquad \forall \boldsymbol{\psi}_{\vec{r}} \in \mathbb{M} \qquad \textit{eq 3. 19}$$

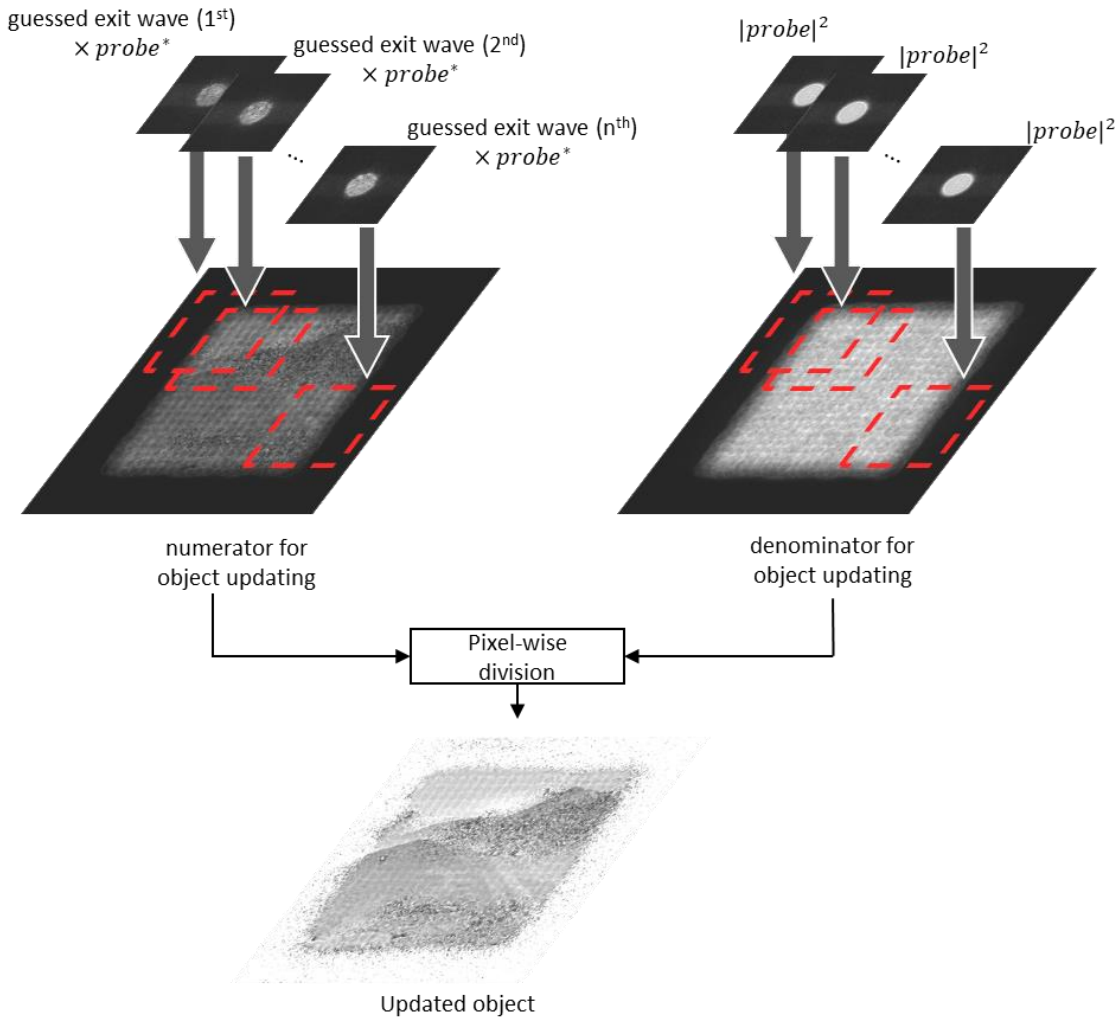


Figure 3. 6. A basic consistency set projection. Guessed exit waves are multiplied with the conjugate of probe and added up following the scanning positions. Three exit waves are given as examples in the figure, while the outlines of their covering areas are highlighted by red dotted line. A similar add-up process is repeated with the squared modulus of probe. Finally, an updated object is generated by a pixel-wise division between these two matrices. The probe is updated in a similar way, except the probe existed in the above computation is replaced by the corresponding object part and the position-wise add up is replaced by a normal matrix add up. The adding up process can involve different weighting factors[55].

Pseudocode 3. 4:*The s-projection for consistency constraint ($\mathcal{P}_s$)*

Input: exit waves ($exit\ wave$), scanning positions ($positions$), guessed probe ($probe$)

Output: revised object ($revised\ object$), revised probe ($revised\ probe$), revised exit waves ($revised\ exit\ wave$)

Format: $[revised\ exit\ wave, revised\ probe, revised\ object] = \mathcal{P}_s(exit\ wave, positions, probe)$

1: $object\ numerator = \mathbf{0}$

2: $object\ denominator = \mathbf{0}$

3: **For** (k=1: total number of $positions$) **do**

4: $object\ numerator = \boldsymbol{Add}(object\ numerator, the\ kth\ exit\ wave \cdot probe^*, the\ k_{th}\ positions,\ size\ of\ probe)$

5: $object\ denominator = \boldsymbol{Add}(object\ numerator, |probe|^2,\ the\ k_{th}\ positions, size\ of\ probe)$

6: **End**

7: $revised\ object = object\ numerator / object\ denominator$

8: $revised\ object = \boldsymbol{ReplaceInf}(revised\ object, object)$

9: **For** (k=1: total number of $positions$) **do**

10: $the\ k_{th}\ part = \boldsymbol{Cut}(revised\ object,\ the\ k_{th}\ positions,\ size\ of\ probe)$

11: $probe\ numerator = probe\ numerator + kth\ exit\ wave \cdot the\ k_{th}\ part^*$

12: $probe\ denominator = probe\ denominator + |the\ k_{th}\ part|^2$

13: **End**

14: $revised\ probe = probe\ numerator / probe\ denominator$

15: $revised\ probe = \boldsymbol{ReplaceInf}(revised\ probe, probe)$

16: **For** (k=1: total number of $positions$) **do**

17: $the\ k_{th}\ part = \boldsymbol{Cut}(revised\ object,\ the\ k_{th}\ positions,\ size\ of\ probe)$

18: $the\ k_{th}\ revised\ exit\ wave = the\ k_{th}\ part \cdot revised\ probe$

19: **End**

Note [1]: The object and probe are always revised while projecting exit waves to the s-constraint.

Note [2]: The order of updating object and probe is interchangeable, though it will converge to a different solution during iterations[46].

Note [3]: Updating object first brings 2 advantages in application: 1. The original object is not required; 2. All of $the\ k_{th}\ part$ produced in revising probe can be saved and utilised in revising exit wave

Note [4]: $\boldsymbol{ReplaceInf}(matrix, backup)$ replaces any 'not a number' element in $matrix$ by the element at the same place of $backup$.

3.1.1.3. **Reflection ($\boldsymbol{\mathcal{R}}$)**

Reflection doubles the correction of projection as shown in *eq 3. 20*. Applying reflection normally requires applying projection first, then doubles the variation. As an aside, such a progress often requires an extra memory with the same size as the variable that is being reflected, as it needs to keep both the original variable (e.g. $\boldsymbol{\psi}_{\vec{r}}$) and its projection (e.g. $\boldsymbol{\mathcal{P}}(\boldsymbol{\psi}_{\vec{r}})$) until a result is reached. This characteristic makes reflection algorithms usually demand more computer memory than others.

$$\begin{aligned}\boldsymbol{\psi}_{\vec{r}}{}' &= \boldsymbol{\mathcal{R}}(\boldsymbol{\psi}_{\vec{r}}) \\ &= \boldsymbol{\psi}_{\vec{r}} + 2(\boldsymbol{\mathcal{P}}(\boldsymbol{\psi}_{\vec{r}}) - \boldsymbol{\psi}_{\vec{r}}) \\ &= \boldsymbol{\psi}_{\vec{r}} + 2\boldsymbol{\mathcal{P}}(\boldsymbol{\psi}_{\vec{r}}) - 2\boldsymbol{\psi}_{\vec{r}} \\ &= (2\boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{I}})\,\boldsymbol{\psi}_{\vec{r}}\end{aligned} \qquad \textit{eq 3. 20}$$

Given the *f-reflection* as an example. As shown in *eq 3. 21*, both the original and projected exit waves are demanded to compute the reflection with a corresponding ***Pseudocode 3. 5***. A modified version, which can reduce its memory occupation, is explained in ***Pseudocode 3. 6***. The geometry meaning of *f-reflection* is illustrated by *Figure 3. 7*.

$$\boldsymbol{\mathcal{R}}_{\boldsymbol{f}}(\boldsymbol{\psi}_{\vec{r}}) = (2 \cdot \boldsymbol{\mathcal{P}}_{\boldsymbol{f}} - \boldsymbol{\mathcal{I}})\boldsymbol{\psi}_{\vec{r}} \qquad \textit{eq 3. 21}$$

Reflection add diversity to algorithms based on set-projection concept. Phase retrieval algorithms with different combination of projection and reflection can have different performance and robustness. However, as reflection-based algorithms do not have to match any constraint, their error may increase during the optimisation, hence making convergence harder to prove. Reflection is not a linear operator and it can be proved the reflection of a reflection equals the original vector as shown in *eq 3. 22*.

$$\begin{aligned}\boldsymbol{\psi}_{\vec{r}} &= \boldsymbol{\mathcal{R}}\big(\boldsymbol{\mathcal{R}}(\boldsymbol{\psi}_{\vec{r}})\big) \\ &= \boldsymbol{\mathcal{R}}(2\boldsymbol{\mathcal{P}}(\boldsymbol{\psi}_{\vec{r}}) - \boldsymbol{\psi}_{\vec{r}}) \\ &= 2\boldsymbol{\mathcal{P}}(2\boldsymbol{\mathcal{P}}(\boldsymbol{\psi}_{\vec{r}}) - \boldsymbol{\psi}_{\vec{r}}) - (2\boldsymbol{\mathcal{P}}(\boldsymbol{\psi}_{\vec{r}}) - \boldsymbol{\psi}_{\vec{r}}) \\ &= 2\boldsymbol{\mathcal{P}}\big(\boldsymbol{\mathcal{P}}(\boldsymbol{\psi}_{\vec{r}}) + (\boldsymbol{\mathcal{P}}(\boldsymbol{\psi}_{\vec{r}}) - \boldsymbol{\psi}_{\vec{r}})\big) - 2\boldsymbol{\mathcal{P}}(\boldsymbol{\psi}_{\vec{r}}) + \boldsymbol{\psi}_{\vec{r}} \\ &= 2\boldsymbol{\mathcal{P}}(\boldsymbol{\psi}_{\vec{r}}) - 2\boldsymbol{\mathcal{P}}(\boldsymbol{\psi}_{\vec{r}}) + \boldsymbol{\psi}_{\vec{r}} \\ &= \boldsymbol{\psi}_{\vec{r}}\end{aligned} \qquad \textit{eq 3. 22}$$

Pseudocode 3. 5: *The f-reflection ($\mathcal{R}_f$)*

Input: guessed exit wave ($exit\ wave$), measured diffraction pattern ($intensity$)

Output: revised exit wave ($revised\ exit\ wave$)

Format: $revised\ exit\ wave = \mathcal{R}_f(exit\ wave, intensity)$

1:	$revised\ exit\ wave = \mathcal{P}_f(exit\ wave, intensity)$
2:	$revised\ exit\ wave = 2 \cdot revised\ exit\ wave - exit\ wave$

Pseudocode 3. 6: *The 'memory-saving' version of f-reflection ($\mathcal{R}_f$)*

Input: guessed exit wave ($exit\ wave$), measured diffraction pattern ($intensity$)

Output: revised exit wave ($revised\ exit\ wave$)

Format: $revised\ exit\ wave = \mathcal{R}_f(exit\ wave, intensity)$

1:	$FT\ exit\ wave = \mathcal{F}\ (exit\ wave)$
2:	$FT\ exit\ wave = \left(2 \cdot \sqrt{intensity} - |FT\ exit\ wave|\right) \cdot \exp\left(j \cdot angle(FT\ exit\ wave)\right)$
3:	$exit\ wave = \mathcal{F}^{-1}(FT\ exit\ wave)$

Note [1]: $exit\ wave$ and $FT\ exit\ wave$ can share the same memory space

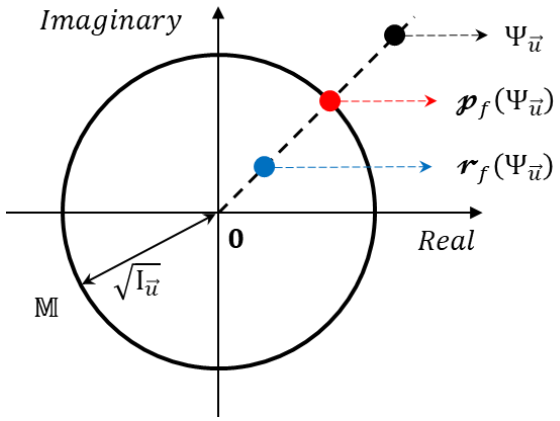


Figure 3. 7. A demonstration of geometry relationship between f-projection $(\mathcal{P}_f)$ and f-reflection ($\mathcal{R}_f$) on the same pixel of $\Psi_{\vec{u}}$ (i.e. $\Psi_{\vec{u},d}$). The modulus set ($\mathbb{M}$) for this pixel is expressed as a circle, whose radius is the square root of measured intensity $\left(\sqrt{I_{\vec{u}}}\right)$. The resultant of projection is expressed by the red dot, while the reflection is expressed by the blue dot.

3.1.1.4. **Relaxed projection ($\boldsymbol{\mathcal{P}^{\alpha}}$)**

The relaxed projection is utilised in several phase retrieval algorithms[54,55]. They come with relaxation parameters (e.g. α) to mediate the projection outcome with original value as demonstrated in *eq 3. 23*. The main feature of this operator is that it is a linear combination between the original value ($\boldsymbol{\mathcal{I}}$) and the projection outcome ($\boldsymbol{\mathcal{P}}$) with the sum of their coefficients equal to one. Although involving a parameter allows the algorithms to be tuned for different scenarios, it puts challenge to the tuning process and leave its results to the experience of users. The ***Pseudocode 3. 7*** and ***Pseudocode 3. 8*** give examples of relaxed *f-constraint* projection ($\boldsymbol{\mathcal{P}_f^{\alpha}}$) and relaxed s-*constraint* projection ($\boldsymbol{\mathcal{P}_s^{\alpha}}$).

$$\boldsymbol{\mathcal{P}}^{\alpha}\left(\boldsymbol{\psi}_{\vec{r}}\right) = \left((1-\alpha)\boldsymbol{\mathcal{I}} + \alpha\boldsymbol{\mathcal{P}}\right)\boldsymbol{\psi}_{\vec{r}} \qquad \textit{eq 3. 23}$$

***Pseudocode 3. 7**:* The relaxed f-projection ($\boldsymbol{\mathcal{P}_f^{\alpha}}$)

Input: guessed exit wave ($exit\ wave$), measured diffraction pattern ($intensity$)

Output: revised exit wave ($revised\ exit\ wave$)

Format: $revised\ exit\ wave = \boldsymbol{\mathcal{P}}_f^{\alpha}(exit\ wave, intensity, \alpha)$

1:	$FT\ exit\ wave = \boldsymbol{\mathcal{F}}\ (exit\ wave)$
2:	$FT\ exit\ wave = \left((1-\alpha)\sqrt{intensity} + \alpha \cdot \lvert FT\ exit\ wave\rvert\right) \cdot \exp\left(j \cdot angle(FT\ exit\ wave)\right)$
3:	$revised\ exit\ wave = \boldsymbol{\mathcal{F}}^{-1}(FT\ exit\ wave)$

Note [1]: Temporary variable: $FT\ exit\ wave$

***Pseudocode 3. 8**: The relaxed s-projection ($\boldsymbol{\mathcal{P}}_s^{\alpha}$)*

Input:	exit waves ($exit\ wave$), scanning positions ($positions$), guessed probe ($probe$), parameter (α)
Output:	revised object ($revised\ object$), revised probe ($revised\ probe$) and revised exit waves ($revised\ exit\ wave$)
Format:	$revised\ exit\ wave = \boldsymbol{\mathcal{P}}_s^{\alpha}\ (exit\ wave, positions, probe, \alpha)$

1:	$revised\ exit\ wave = \boldsymbol{\mathcal{P}}_{\mathrm{s}}(exit\ wave, positions, probe)$
2:	$revised\ exit\ wave = (1-\alpha)\cdot revised\ exit\ wave - \alpha \cdot exit\ wave$

3.1.1.5. Relaxed reflection ($\boldsymbol{\mathcal{R}^{\alpha}}$)

The reflection can also be generalised by involving a mediate parameter. However, since the reflection is already a linear combination of the projection and original value (i.e. *eq 3. 20*), its generalised form overlaps with the relaxed projection ($\boldsymbol{\mathcal{P}^{\alpha}}$). Every result produced by relaxed reflection can be obtained from the relaxed projection just with a proper chosen parameter. Therefore, the relaxed reflection is combined with relaxed projection in this thesis, and shares the same expression as shown in *eq 3. 23*.

3.1.2. Gradient descent

The gradient descent method is another common optimisation methodology. All PIE-related algorithms[20,56], ADMM[55] and conjugate gradient[57] belong to this category. These algorithms usually take a random guess in the searching space ($D = M \times N \times K$) as a start. They evaluate the gradient at the current guess with a properly designed error metric, then move towards the negative gradient direction with a certain step size. The latter two steps are repeated until a minimum of the error metric is found, which is considered as a solution. The concept of gradient descent method is commonly viewed as letting a ball roll on a curvature surface and expecting the ball keeps descending until a pit is reached. Such an optimisation approach is widely used in training neural networks. The gradient descent method unveils the similarity between training neural networks and solving phase problem iteratively[58], unsurprisingly, some useful concept in the former one can be transferred to the latter one. This section

explains some main concepts within the gradient descent method and describes how it solves phase problem.

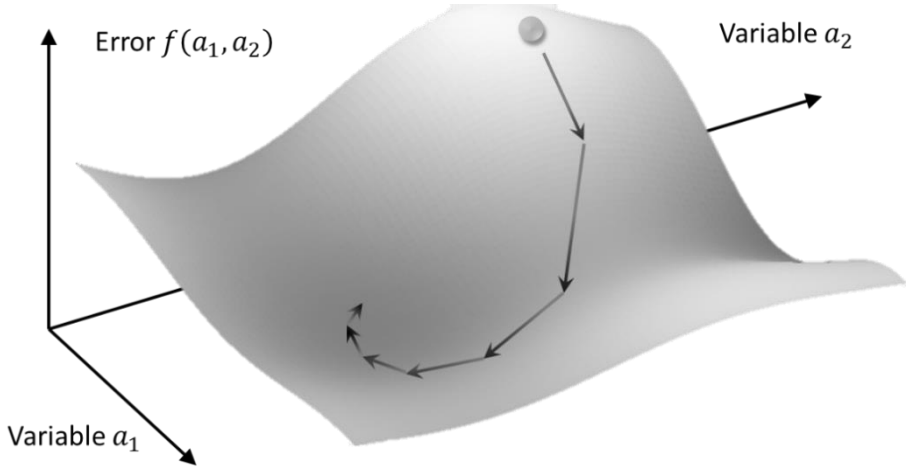


Figure 3. 8. A simplified demonstration of steepest descent method in 2-dimensional searching space. The initial guess (a_1, a_2) is noted by a ball sitting on the curvature surface, which is defined by a cost function ($f(a_1, a_2)$). The ball tracks the descending direction and reaches the bottom of this surface. Iterative optimisation can be considered as viewing this process with equal time interval, the ball coordinate at each observing time point (i.e. the outcome at the end of each iteration) can be referred to the next arrow tip. As the height of surface is proportional to error, its bottom indicates a variable combination that gives the least error. Hence the error is minimised, and a solution is found.

3.1.2.1. Cost function ($\mathcal{L}$)

A *loss function* ($\mathcal{L}$), also known as a *cost function*, is the core of gradient descent method. It is usually an error metric that reflects the difference between the guesses and experimental measurements. Its value should decrease when approaching to a solution and, ideally, equals zero when a solution is found. A properly designed cost function usually contains variables that are under optimisation and be differentiable with respect to that interested variable.

The basic cost function is demonstrated as *eq 3. 24*. It evaluates the Euclidean distance of exit waves before and after *f-projection*, and this evaluation can be done either in *s-domain* or *f-domain*. These two functions share a lot of common structure: they all include the exit waves satisfying one constraint and compute the difference to the other constraint; they all take the square of the modulus; and they all sum up every pixel for every scanned position. The development of PIE family starts from reducing the distance in *s-domain*, while alternating direction method of multipliers and conjugate gradient are based on the distance in *f-domain*.

$$\mathcal{L} = \sum_k \sum_{\vec{r}} \left| \mathbf{O}_{\vec{r},k} \mathbf{P}_{\vec{r}} - \boldsymbol{\psi}_{\vec{r},k} \right|^2 = \sum_k \sum_{\vec{u}} \left| \sqrt{\mathbf{I}_{\vec{u},k}} - \left| \boldsymbol{\Psi}_{\vec{u},k} \right| \right|^2 \qquad \textit{eq 3. 24}$$

Such a similarity come with a reason. As the two main constraints in ptychography are the *s-* and *f- constraints*, any accepted guess must satisfy both, which implies "zero-difference" to any of these constraints at the same time. Unfortunately, these two constraints are separated by Fourier transformation and reciprocal to each other. For evaluating the difference, guessed exit waves act as carriers, bring the value satisfying *s-domain* constraint to the reciprocal space, hence the difference can be computed. Due to the Parseval's theorem, the difference computed in *s-domain* equals the difference in *f-domain*[24]. The square of modulus is utilised as it is a standard way of evaluating the Euclidean distance, which is the geometry meaning of the difference. Finally, the distance for all pixels (i.e. $\sum_{\vec{r}}$ or $\sum_{\vec{u}}$) and all positions (i.e. $\sum_k$) is summed up to give an overall distance in the whole searching space. Using this logic, a cost function for a variable ($\boldsymbol{\psi}_{\vec{r}}$) belongs to space $\mathbb{C}^D$ and needs to satisfy constraint $\mathbb{M}$ can be written as *eq 3. 25*.

$$\mathcal{L}_{\mathbb{M}} = \sum_k \sum_{\vec{r}} |\boldsymbol{\psi}_{\vec{r}} - \boldsymbol{\mathcal{P}}_{\mathbb{M}}(\boldsymbol{\psi}_{\vec{r}})|^2 \qquad \textit{eq 3. 25}$$

3.1.2.2. Regularization

Other terms can be added into cost function to emphasize other properties. One typical example is the regularisation term. Regularisation is widely used in training neural networks to prevent overfitting[58]. This term evaluates the variation on the object under optimisation within one iteration and returns high penalties on dramatic changes. Two examples are given in *eq 3. 26* and *eq 3. 27*, where the super script $\mathbf{O}'_{\vec{r}}$ and $\boldsymbol{\Psi}'_{\vec{u}}$ indicates the updated object and exit wave correspondingly. *eq 3. 26* is utilised by rPIE and *eq 3. 27* is utilised by alternating direction method of multipliers. More explanation on regularisation is given in Chapter 5.

$$\mathcal{L}_{s-regulation} = \sum_{\vec{r}} \left| \mathbf{O}'_{\vec{r}} - \mathbf{O}_{\vec{r}} \right|^2 \qquad \textit{eq 3. 26}$$

$$\mathcal{L}_{f-regulation} = \sum_k \sum_{\vec{u}} \left| \boldsymbol{\Psi}'_{\vec{u}} - \boldsymbol{\Psi}_{\vec{u}} \right|^2 \qquad \textit{eq 3. 27}$$

3.1.2.3. $\mathbb{CR}$- or Wirtinger calculus

Once a cost function is defined, its gradient can be calculated by partially differentiating this function with respect to the interested variable. Differentiating a function made up of complex matrices requires different math skills, the implicit relationship between the involved variables can also cause confusion. Therefore, some basic rules of these partial differentiations are given in ***Table 3. 1***. One can derivate a cost function by applying chain rule.

Table 3. 1. Rules of Wirtinger calculus

Relationship	Differentiation	Examples
Complex modulus	$\frac{\partial\lvert\mathbf{M}\rvert}{\partial\mathbf{M}}=\frac{1}{2\lvert\mathbf{M}\rvert}\frac{\partial\lvert\mathbf{M}\rvert^2}{\partial\mathbf{M}}$	$\frac{\partial\lvert\boldsymbol{\Psi}_{\vec{u}}\rvert}{\partial\boldsymbol{\Psi}_{\vec{u}}}=\frac{1}{2\lvert\boldsymbol{\Psi}_{\vec{u}}\rvert}\frac{\partial\lvert\boldsymbol{\Psi}_{\vec{u}}\rvert^2}{\partial\boldsymbol{\Psi}_{\vec{u}}}$
	$\frac{\partial\lvert\mathbf{M}\rvert^2}{\partial\mathbf{M}}=\frac{\partial\mathbf{M}\cdot\mathbf{M}^*}{\partial\mathbf{M}}$	$\frac{\partial\lvert\mathbf{P}_{\vec{r}}\mathbf{O}_{\vec{r}}-\boldsymbol{\psi}_{\vec{r}}\rvert^2}{\partial\mathbf{P}_{\vec{r}}{}^*}=\frac{\partial(\mathbf{P}_{\vec{r}}\mathbf{O}_{\vec{r}}-\boldsymbol{\psi}_{\vec{r}})\cdot(\mathbf{P}_{\vec{r}}\mathbf{O}_{\vec{r}}-\boldsymbol{\psi}_{\vec{r}})^*}{\partial\mathbf{P}_{\vec{r}}{}^*}$
Conjugate function	$\frac{\partial(\mathrm{k}\mathbf{M}+\mathrm{b})}{\partial\mathbf{M}}=\mathbf{M}$	$\frac{\partial\left(\mathbf{P}_{\vec{r}}\mathbf{O}_{\vec{r}}-\boldsymbol{\psi}'_{\vec{r}}\right)}{\partial\mathbf{P}_{\vec{r}}{}^*}=\mathbf{0}$
	$\frac{\partial(\mathrm{k}\mathbf{M}+\mathrm{b})^*}{\partial\mathbf{M}}=\mathbf{0}$	$\frac{\partial\left(\mathbf{P}_{\vec{r}}\mathbf{O}_{\vec{r}}-\boldsymbol{\psi}'_{\vec{r}}\right)^*}{\partial\mathbf{P}_{\vec{r}}{}^*}=\mathbf{O}_{\vec{r}}{}^*$
Projection	$\frac{\partial\boldsymbol{\mathcal{P}}(\mathbf{M})}{\partial\mathbf{M}}=\mathbf{0}$	$\frac{\partial\mathbf{O}'_{\vec{r}}}{\partial\mathbf{O}_{\vec{r}}}=\mathbf{0}$
Fourier transformation	$\frac{\partial\boldsymbol{\mathcal{F}}(\mathbf{M})}{\partial\mathbf{M}}=\exp(-2\pi j\vec{u}\vec{r})$	$\frac{\partial\boldsymbol{\Psi}_{\vec{u}}}{\partial\boldsymbol{\psi}_{\vec{r}}}=\exp(-2\pi j\vec{u}\vec{r})$

3.1.2.4. Step size

As the gradient indicates the fastest error-increasing direction, a better guess can be achieved by moving in the opposite direction with a proper step size. As most of gradient-descent methods use the first order derivative of the cost function, the approximate step size is estimated by finding the intersection point by solving a first order equation.

One of the most significant characteristic of gradient descent method is its error never increase during iteration. Such a characteristics gives it a fast converging speed. This seems a good characteristic in the first thought. However, the searching space of phase problem is full of local minima. Any algorithms only focus on minimising the error metric can easily get stuck. Therefore, the variants of gradient descent methods, e.g. stochastic gradient descent[59], regularisation[56] and descent with momentum[56], are more preferred.

3.2. Algorithms before ptychography

The phase problem has existed in coherent diffraction imaging microscopy before the invention of ptychography[13] and variety approaches have been developed to tackle this problem. Although they all utilise the intensity of diffraction patterns ($\mathbf{I}_{\vec{u}}$) as one of their constraints, the other constraint was under developing hence varied from one to another. This promoted the diversity of algorithms. Some of these ideas and algorithms inspire the development of ptychography, hence they are explained to give a better understanding of the diffractive imaging problem.

3.2.1. Gerchberg-Saxton (Error reduction)

Back in the early 1970's, there was no promising and efficient way of retrieving phase until the development of Gerchberg -Saxton algorithm in 1972[24]. This algorithm was not only a decent solution to the phase problem in electron microscopy during that period, but also provided a new idea of involving extra constraint to make phase retrieval practical.

The Gerchberg-Saxton method requires only one scanning position, but two intensity measurements of both the specimen ($\mathbf{I}_{\boldsymbol{img}_{\vec{r}}}$) and diffraction pattern ($\mathbf{I}_{\vec{u}}$). With these *s*- and *f*-*constraints*, one can replace the modulus of the guessed image with the square root of the specimen intensity, then transfer this revised wave to its reciprocal space and apply the diffraction pattern intensity constraint. This concept is demonstrated by flowchart in *Figure 3. 9* and explained with ***Pseudocode 3.* 9**.

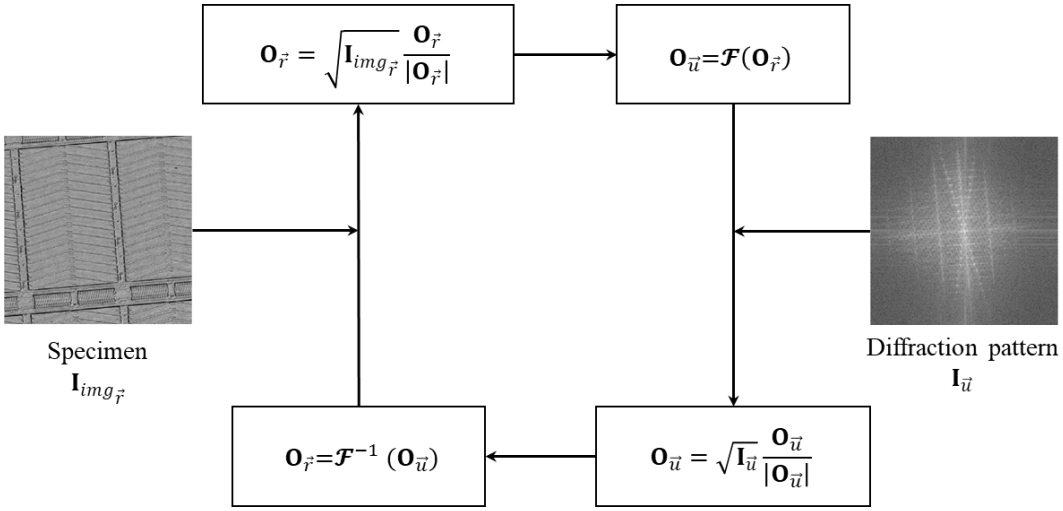


Figure 3. 9. The flowchart of Gerchberg-Saxton algorithm. This algorithm is alternatively projecting the guessed image ($\boldsymbol{O}_{\vec{r}}$) to the intensity measurements in s- and f- domains.

Pseudocode 3. 9: *Gerchberg-Saxton (Error reduction) with image intensity*

Input: measured image intensity $specimen\ intensity$), measured diffraction pattern ($intensity$), No. of iterations (N)

Output: revised object ($revised\ object$)

1: $object = \sqrt{specimen\ intensity}$

2: **For** (n=1: N) **do**

3: $object = \boldsymbol{\mathcal{P}}_f(object, intensity)$

4: $revised\ object = \boldsymbol{\mathcal{P}}_{img}(object, specimen\ intensity)$

5: **End**

The influence of this alternative-projection is demonstrated by arbitrary pixels of the guessed object in *s-* and *f- domains* by *Figure 3. 10 (a) and (b)* correspondingly. The circles in *Figure 3. 10 (a) and (b)* represent the solution sets formed by the measured specimen intensity ($\mathbb{M}_{img}$) and diffraction pattern intensity ($\mathbb{M}$). An arbitrary pixel ($\mathrm{O}_{\vec{r},n}$) of guessed object, who satisfies the *s-constraint*, is represented as a black dot in *Figure 3. 10 (a)*. This guessed object is transformed into its reciprocal space, yields $\mathbf{O}_{\vec{u},n}$. An arbitrary pixel in the reciprocal space is represented as a black dot ($\mathrm{O}_{\vec{u},n}$) in *Figure 3. 10 (b)*. After $\boldsymbol{\mathcal{P}}_f$, the value at this pixel becomes $\mathrm{O}_{\vec{u},n}{}'$. Then the whole revised object is transformed back to the real space, becomes $\mathbf{O}_{\vec{r},n}{}'$.

Unless a solution is found, this revised object ($\mathbf{O}_{\vec{r},n}{}'$) will not satisfy the s-constraint. As an example, which is shown in *Figure 3. 10 (a)*, the pixel after revision ($\mathrm{O}_{\vec{r},n}{}'$) does not belong to the set formed by s-constraint. Hence the revised object needs to be projected ($\boldsymbol{\mathcal{P}}_{img}$) to the s-constraint and turned into $\mathbf{O}_{\vec{r},k+1}$, which leads to a new pixel value ($\mathrm{O}_{\vec{r},k+1}$) that fits the s-constraint. This is the end of k^{th} iteration and the start of $k+1^{th}$h iteration of Gerchberg-Saxton algorithm.

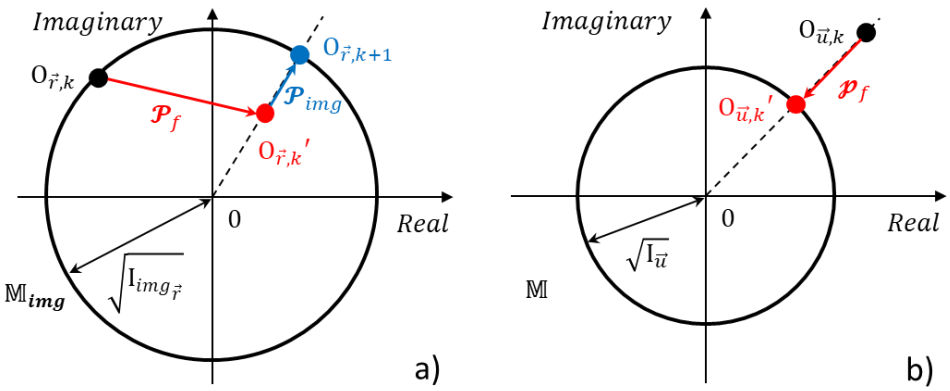


Figure 3. 10. A demonstration of Gerchberg-Saxton algorithm with an arbitrary pixel of guessed object in s-domain (i.e. figure (a)) and f-domain (i.e. figure (b)). The sets formed by s-constraint ($\mathbb{M}_{img}$) and f-constraint ($\mathbb{M}$) for the chosen pixel are represented as circles in (a) and (b) respectively. The initial value of this arbitrary pixel at the start of kth iteration satisfies the s-constraint and demonstrated as black dot ($O_{\vec{r},k}$) in figure (a). Fourier transform this guessed object $\boldsymbol{O}_{\vec{r},n}$ yields $\boldsymbol{O}_{\vec{u},n}$ in the f-domain. An arbitrary pixel $O_{\vec{u},k}$ is chosen out of this transformed guessed object. After $\boldsymbol{\mathcal{P}}_f$, this pixel falls onto the set formed by f-constraint and is highlighted as a red dot ($O_{\vec{u},n}{}'$). The whole revised object is then inversely transformed back to the s-constraint, gives $\boldsymbol{O}_{\vec{r},n}{}'$, and the s-domain chosen pixel value now becomes $O_{\vec{r},k}{}'$, which does not satisfy the s-constraint. Applying $\boldsymbol{\mathcal{P}}_{img}$ leads to $O_{\vec{r},n+1}$, which is illustrated as a blue dot, who satisfies the s-constraint again.

The recursive formula of Gerchberg-Saxton algorithm is given in *eq 3. 28*, where $\boldsymbol{\mathcal{P}}_{img}$ represents the projection to the *s-constraint* for Gerchberg-Saxton.

$$\mathbf{O}_{\vec{r},n+1} = \boldsymbol{\mathcal{P}}_f\left(\boldsymbol{\mathcal{P}}_{img}\left(\mathbf{O}_{\vec{r},n}\right)\right) \qquad \textit{eq 3. 28}$$

The summed squared error (Err_{SS}) evaluates the difference between the estimated and measured intensities either in s-domain (*eq 3. 29*) or in f-domain (*eq 3. 30*). Gerchberg-Saxton algorithm can be proved as a gradient descent method with this error metric, since its error never increases during the reconstruction[40].

$$Err_{SS\,s} = \sum_{\vec{r}} \left(\sqrt{\mathbf{I}_{img_{\vec{r}}}} - |\mathbf{O}_{\vec{r}}| \right)^2 \quad \textit{eq 3. 29}$$

$$Err_{SS\,f} = \sum_{\vec{u}} \left(\sqrt{\mathbf{I}_{\vec{u}}} - |\mathbf{O}_{\vec{u}}| \right)^2 \quad \textit{eq 3. 30}$$

First, using the revised object in real space ($\mathbf{O}_{\vec{r},n}{}'$) as a reference, computing the error of object at the beginning and end of the n^{th} iteration.

$$Err_{SS\,s,n} = \sum_{\vec{r}} \left(|\mathbf{O}_{\vec{r},n}| - |\mathbf{O}'_{\vec{r},n}| \right)^2 \quad \textit{eq 3. 31}$$

$$Err_{SS\,s,n\prime} = \sum_{\vec{r}} \left(|\mathbf{O}_{\vec{r},n+1}| - |\mathbf{O}'_{\vec{r},n}| \right)^2 \quad \textit{eq 3. 32}$$

From the definition of projection, we have:

$$\left| |\mathbf{O}_{\vec{r},n}| - |\mathbf{O}'_{\vec{r},n}| \right| \geq \left| |\mathbf{O}_{\vec{r},n+1}| - |\mathbf{O}'_{\vec{r},n}| \right| \quad \textit{eq 3. 33}$$

Hence:

$$Err_{SS\,s,n} \geq Err_{SS\,s,n\prime} \quad \textit{eq 3. 34}$$

Assuming $\mathbf{O}'_{\vec{u},n+1} = \boldsymbol{\mathcal{P}}_f\left(\mathbf{O}_{\vec{u},n+1}\right)$, then using the $\mathbf{O}_{\vec{u},n+1}$ as a reference, we can compute the error of $\mathbf{O}'_{\vec{u},n}$ and $\mathbf{O}'_{\vec{u},n+1}$ as:

$$Err_{SS\,f,n\prime} = \sum_{\vec{u}} \left(|\mathbf{O}'_{\vec{u},n}| - |\mathbf{O}_{\vec{u},n+1}| \right)^2 \quad \textit{eq 3. 35}$$

$$Err_{SS\,f,n+1} = \sum_{\vec{u}} \left(|\mathbf{O}'_{\vec{u},n+1}| - |\mathbf{O}_{\vec{u},n+1}| \right)^2 \quad \textit{eq 3. 36}$$

Again, with the definition of projection, we have:

$$\left| |\mathbf{O}'_{\vec{u},n}| - |\mathbf{O}_{\vec{u},n+1}| \right| \geq \left| |\mathbf{O}'_{\vec{u},n+1}| - |\mathbf{O}_{\vec{u},n+1}| \right| \quad \textit{eq 3. 37}$$

Hence, we have:

$$Err_{SS\,f,n'} \geq Err_{SS\,f,n+1} \qquad \textit{eq 3. 38}$$

Due to the Parseval's rule:

$$Err_{SS\,s,n'} = Err_{SS\,f,n'} \qquad \textit{eq 3. 39}$$

Combining *eq 3. 53*, *eq 3. 62* and *eq 3. 39*, we have:

$$Err_{SS\,s,n} \geq Err_{SS\,s,n'} = Err_{SS\,f,n'} \geq Err_{SS\,f,n+1} \qquad \textit{eq 3. 40}$$

which indicates the error should always decrease or stay the same during the iteration. For this reason, Gerchberg-Saxton algorithm is also known as the error reduction (ER) algorithm[40]. Gerchberg-Saxton algorithm was one of the earliest phase retrieval algorithms. It outperformed other algorithms in reconstructing the whole phase of an interested wave front in electron microscopy [24]. As a derivation of gradient descent method, its error drops quickly in the beginning[40]. However, this algorithm also suffers several drawbacks. First, since its error can never increase, it is not possible to escape from a stagnation. This is a poor characteristic for phase retrieving, which contains multiple local minima[54]. Secondly, as Gerchberg-Saxton algorithm cannot distinguish any existed phase offset in the reconstruction, its outcomes have an inherent ambiguity on the phase bias, though this ambiguity is not fatal if only the relative phase is interested[40].

3.2.2. Hybrid input output (HIO)

The HIO algorithm is developed for a support ($\mathbf{S}_{\vec{r}}$) constraint in s-domain. As the support only lets waves passes through its centre ($\vec{r} \in \mathbf{S}_{\vec{r}}$), any non-zero element outside this range ($\vec{r} \notin \mathbf{S}_{\vec{r}}$) is considered as violating the constraint. The concept of HIO algorithm is similar to the feedback system[41], which is modifying the input proportional to the error in its output.

Due to the existence of the support, the variable under reconstruction becomes the exit wave ($\boldsymbol{\psi}_{\vec{r}}$) rather than the image of the specimen. HIO projects a guessed exit wave to the modulus

constraint (i.e. $\boldsymbol{\psi}'_{\vec{r}} = \boldsymbol{\mathcal{P}}_f(\boldsymbol{\psi}_{\vec{r}})$) and compares the output with the support. For the area inside the support ($\vec{r} \in \mathbf{S}_{\vec{r}}$), all updates are accepted. For areas beyond the support ($\vec{r} \notin \mathbf{S}_{\vec{r}}$), they are scaled by a coefficient (β) and taken away from the guessed exit wave as negative feedback. The flowchart of HIO is given in *Figure 3. 11* with pseudo code given in ***Pseudocode 3. 10***.

$$\boldsymbol{\psi}_{\vec{r}} = \begin{cases} \boldsymbol{\psi}'_{\vec{r}}, & \boldsymbol{r} \in \boldsymbol{S} \\ \boldsymbol{\psi}_{\vec{r}} - \beta \boldsymbol{\psi}'_{\vec{r}}, & \boldsymbol{r} \notin \boldsymbol{S} \end{cases} \qquad \textit{eq 3. 41}$$

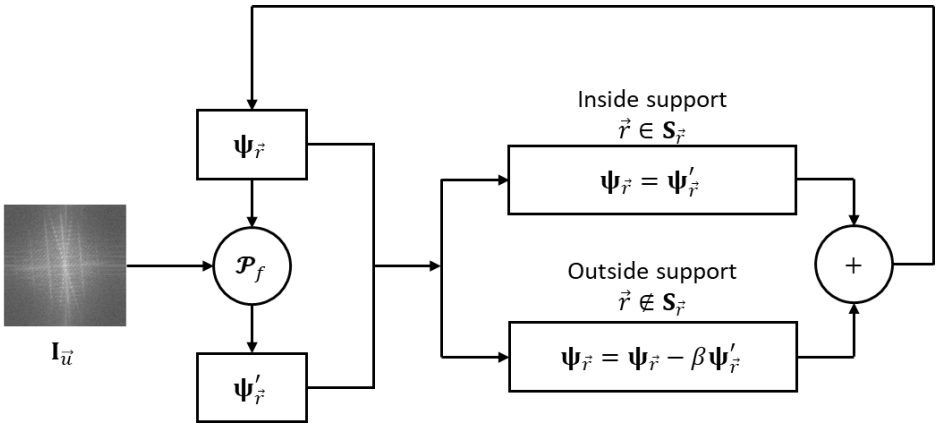


Figure 3. 11. A demonstration of HIO. From top left, a guessed exit wave ($\boldsymbol{\psi}_{\vec{r}}$) is projected to the f-constraint to produce a revised exit wave ($\boldsymbol{\psi}'_{\vec{r}}$). The final updated exit wave is obtained by tuning the input based on the part of output that violates the support constraint.

Pseudocode 3. 10: *Hybrid Input and Output (HIO) with support*

Input: support ($support$), measured diffraction pattern ($intensity$), No. of iterations (N), parameter (β)

Output: revised exit wave ($revised\ \ exit\ wave$)

1: $exit\ wave = support$

2: **For** (n=1: N) **do**

3: $\quad revised\ exit\ wave = \boldsymbol{\mathcal{P}}_f(exit\ wave, intensity)$

4: $\quad$ **If** $support(\vec{r}) == 0$

5: $\quad\quad revised\ exit\ wave(\vec{r}) = exit\ wave(\vec{r}) - \beta \cdot revised\ exit\ wave(\vec{r})$

6: $\quad$ **Else**

7: $\quad\quad revised\ exit\ wave(\vec{r}) = exit\ wave(\vec{r})$

8: $\quad$ **End**

9: **End**

Many variations have been developed to further improve the behaviour of HIO. It has been used for ptychography [60] and modified to accommodate an uncertain support[43]. However, the existence of a support limits the development of this method. Not only because one needs to leave the limited field of view to the non-information support area, but also the accuracy of the support has influence to the quality of reconstruction. Meanwhile, the support constraint may lead to an inherent ambiguity, which is called as 'twin-image'. This ambiguity can happen when the support is centrosymmetric[45]. To prove this, assume $\boldsymbol{\psi}^*_{-\vec{r}}$ is the complex conjugate of $\boldsymbol{\psi}_{\vec{r}}$. If the Fourier transformed $\boldsymbol{\psi}_{\vec{r}}$ is satisfies the modulus constraint (i.e. $|\boldsymbol{\Psi}_{\vec{u}}| = \sqrt{\mathbf{I}_{\vec{u}}}$),

$$\boldsymbol{\Psi}_{\vec{u}} = \boldsymbol{\mathcal{F}}(\boldsymbol{\psi}_{\vec{r}}) = |\boldsymbol{\Psi}_{\vec{u}}| \exp(j\mathbf{G}_{\vec{u}}) \qquad \textit{eq 3. 42}$$

then the Fourier transformation of $\boldsymbol{\psi}^*_{-\vec{r}}$ is:

$$\boldsymbol{\Psi}^*_{\vec{u}} = |\boldsymbol{\Psi}_{\vec{u}}| \exp(-j\mathbf{G}_{\vec{u}}) \qquad \textit{eq 3. 43}$$

Which has the same modulus with $\boldsymbol{\Psi}_{\vec{u}}$, hence also satisfies the modulus constraint. This implies the $\boldsymbol{\psi}^{*}_{-\vec{r}}$ is also an allowed solution. Using image as an example, then the appearance of $\boldsymbol{\psi}^{*}_{-\vec{r}}$ will be centrosymmetric with $\boldsymbol{\psi}_{\vec{r}}$ with a negative phase component. This leads to the reconstruction result stagnates as two centrosymmetric images overlapping with each other, hence this problem is known as 'twin-images'. This problem may even happen to non-centrosymmetric support, when the twin images can fit into a loose support constraint[45].

Besides the twin-image ambiguity, support constraint also cannot find the absolute transverse position of the reconstructed exit wave, as the shifting appears as phase ramp in the *f-domai*n and loses during taking intensity measurement. Though the influence of this ambiguity is less significant and can be limited by fixing the support position.

Besides the ER and HIO, many other algorithms are applied for solving phase problem before the development of ptychography, for instance the difference mapping (DM), the averaged successive reflections (ASR), the hybrid projection reflection (HPR) and relaxed averaged alternating reflectors (RAAR) are also utilised for support-based phase retrieval[61]. Instead of explaining them twice (e.g. before and after ptychography), they are explained with the ptychography together in the following section to give a more specific description, which is more related to this thesis.

3.3. Algorithms for ptychography

As explained in *Chapter 2.3*, ptychography provides a different constraint with its ancestors: the overlapped area between multiple scanning positions. Since the adjacent diffraction patterns share part of common area of specimen, the reconstructed image must satisfy all of them at the same time. This consistency set forms the *s-constraint* in ptychography. However, during the reconstruction, ptychography requires separating the guessed specimen from revised exit waves. This requires the exact knowledge of the illumination function, which is not available for most of the time. The good news is that ptychography offers redundant information, which is more than enough to retrieve both the specimen and illumination[59,47]. Such a characteristic attracts the attention of many researchers and variety algorithms are exploited to take the advantage of this new constraint.

In the following sections, each introduced algorithm is applied to reconstruct a simulated noiseless data set, to give a general idea of its performance. A complete analysis of the algorithms will be undertaken in section 4.3, once an appropriate error metric has been developed in section 4.1 and 4.2. The modulus and phase images of the simulated specimen is given in *Figure 3. 12* together with the true and estimated illumination functions. One can compare the reconstructed images with these true object and probe. Since the edges of the reconstructed object is usually noisy and causes images to lose contrast due to their large dynamic range, only the centre of the reconstructed object is displayed as examples in the later sections. This area is marked with a red dotted line in *Figure 3. 12*. The simulated diffraction patterns are generated with a 20×20 scanning grid, which has about 60% overlap area. A random offset equals 20% of the step size is introduced to minimise the influence of raster grid ambiguity. Until a more accurate error metric has been introduced in Chapter 0, the error is evaluated as the normalised difference between the modulus of Fourier transformed exit waves ($\mathbf{\Psi}_{\vec{u},k}$) and the square root of diffraction patterns ($\mathbf{I}_{\vec{u},k}$) as shown by *eq 3. 44*. The memory occupation of the algorithms is listed in ***Table 3. 3***

$$Err_f = \frac{\sum_k \sum_{\vec{u}} \left| \left| \mathbf{\Psi}_{\vec{u},k} \right| - \sqrt{\mathbf{I}_{\vec{u},k}} \right|^2}{\sum_k \sum_{\vec{u}} \mathbf{I}_{\vec{u},k}} \qquad \textit{eq 3. 44}$$

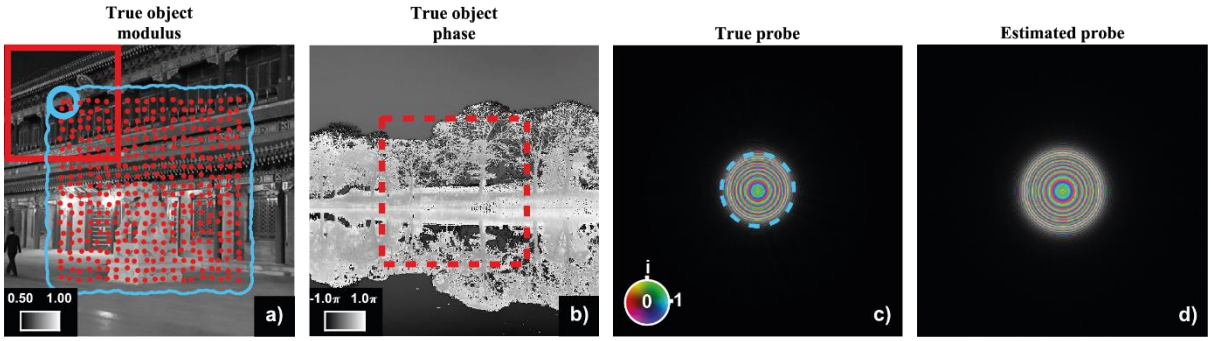


Figure 3. 12. The object and probes used for testing algorithms. From left to right, (a) the modulus of object together with scanning positions (denoted as red dots) and the area covered by probe spot (denoted by cyan outline). The size of probe and its spot at the first scanning position are demonstrated by red square and cyan circle respectively; (b) the phase of object together with the selected area for error computation which is highlighted by red dotted line; (c) the true probe with the outline of spot and (d) the estimated probe. These two probes are plotted in colour wheel format, where the brightness and phase are expressed by the intensity and colour respectively.

3.3.1. Ptychography iterative engine (PIE)

As its name would suggest, PIE was one of the earliest algorithms that specifically designed for solving the phase problem with data collected from ptychography[62]. Without the ability to reconstruct the illumination function, an approximate probe ($\mathbf{P}_{\vec{r}}$) is utilised by the PIE algorithm. As a result, the outcome significantly depends on the accuracy of the estimation. The initial object ($\mathbf{O}_{\vec{r}}$) is usually guessed as a free space (i.e. an all-one matrix) and optimised iteratively during the reconstruction.

Following the scanning sequence, a part of object ($\mathbf{O}_{\vec{r},k}$), which corresponds to the area covered by the probe at the chosen scanning position ($\vec{r}_k$), is cut out from the guessed object and forms an exit wave ($\boldsymbol{\psi}_{\vec{r},k}$) by multiplying with the estimated probe. This guessed exit wave is revised by the corresponding diffraction pattern ($\mathbf{I}_{\vec{u},k}$) and utilised to modify that part of object with following updating function.

$$\mathbf{O}'_{\vec{r},k} = \mathbf{O}_{\vec{r},k} + \frac{|\mathbf{P}_{\vec{r}}|}{|\mathbf{P}_{\vec{r}}|_{max}} \cdot \frac{\mathbf{P}_{\vec{r}}^{*}}{(|\mathbf{P}_{\vec{r}}|^2 + \alpha)} \cdot \left(\mathcal{P}_{\boldsymbol{f}}\left(\boldsymbol{\psi}_{\vec{r},k}\right) - \boldsymbol{\psi}_{\vec{r},k}\right) \qquad \textit{eq 3. 45}$$

Where $\mathbf{O}'_{\vec{r},k}$ is the updated part of object and α is a parameter preventing dividing by zero at the dim part of probe. Then PIE replace the corresponding part of object with the updated one and move to the next scanning position. The origin of this updating function is described in detail in Chapter 5.

In this mechanism, part of the previously updated object appears in the later updating process of adjacent scanning positions. Those revised area improves the quality of following guessed exit wave. Such a mechanism is shown in *Figure 3. 14*. This iterative phase retrieving process is repeated until a desired error level is achieved or a stagnation has been reached. A flow chart and pseudo code are given below with more details of PIE.

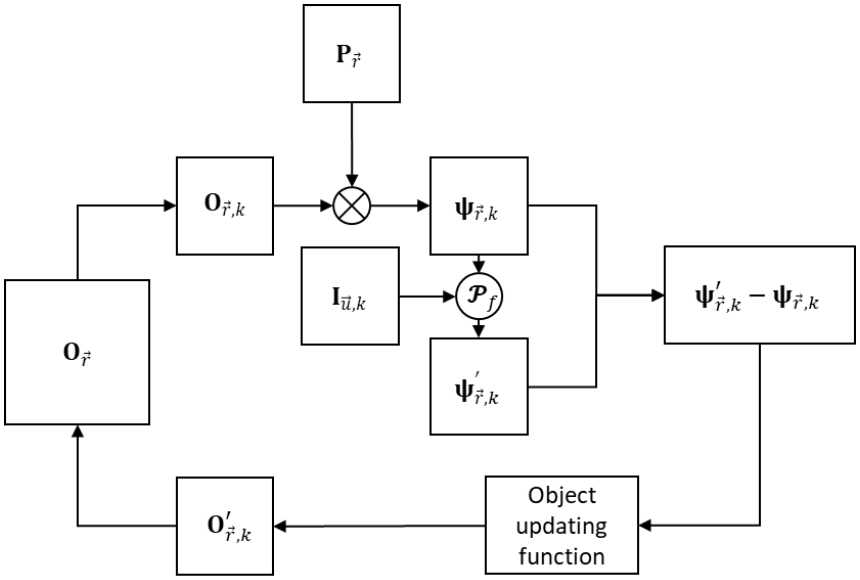


Figure 3. 13. A flow chart for PIE algorithm. A part of object ($\boldsymbol{O}_{\vec{r},k}$) cover by the probe at the $\boldsymbol{r}_k$ scanning position is cut out form the guessed object ($\boldsymbol{O}_{\vec{r}}$) and multiplies with the estimated probe ($\boldsymbol{P}_{\vec{r}}$) to give an guessed exit wave ($\boldsymbol{\psi}_{\vec{r},k}$). This exit wave is projected to f-constraint and applied to update the object part. Finally, the updated object part replaces the corresponding area on the guessed object.

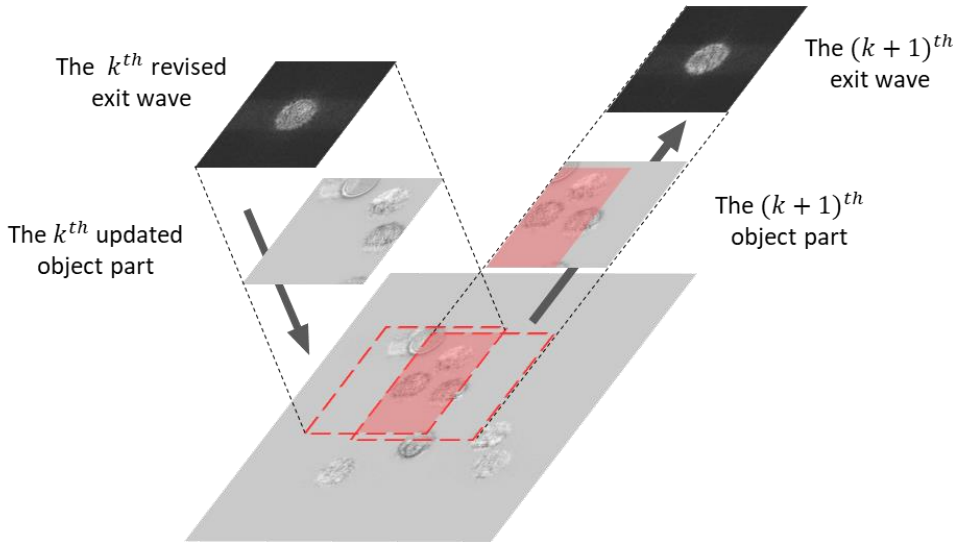


Figure 3. 14. This figure demonstrates how PIE updates object part by part. The k^{th} and $(k+1)^{th}$ scanning positions share an overlapped area, which is coloured as red in the figure. This area is updated by the revised exit wave (k^{th}) and improve the quality of the next estimated exit wave ($(k+1)^{th}$).

To demonstrate the performance of PIE strongly depending on the accuracy of guessed probe, the simulated data is reconstructed with correct and estimated probes respectively. The error during reconstruction is shown in *Figure 3. 15*. As shown in the figure, PIE fails when the estimated probe is utilised, though it gives decent reconstruction with a correct probe. Its

error decreases quickly in the beginning (e.g. drops more than 10 magnitudes in the first 500 iteration), then slows down (e.g. decreases about 3 magnitudes in the last 3000 iterations). This is a typical performance for most of the PIE algorithms as shown in the following sections.

Pseudocode 3. 11: *Ptychographical Iterative Engine (PIE)*

Input: measured diffraction pattern ($intensity$), scanning positions ($positions$), guessed object ($object$), guessed probe ($probe$), No. of iterations (N), Parameter (α)

Output: revised object ($revised\ object$), revised probe ($revised\ probe$)

For (n=1: N) **do**
  $positions = \boldsymbol{shuffle}(positions)$
  For (k=1: total number of $positions$) **do**
    $the\ k_{th}\ part = \boldsymbol{Cut}(revised\ object,\ the\ k_{th}\ positions,\ size\ of\ probe)$
    $exit\ wave = the\ k_{th}\ part\ \cdot probe$
    $revised\ exit\ wave = \boldsymbol{\mathcal{P}}_f(exit\ wave, intensity)$
    $difference = revised\ exit\ wave - exit\ wave$
    $modification = \frac{probe}{|probe|_{max}} \frac{probe^*}{(|probe|^2+\alpha)} \times difference$
    $revised\ object =$
    $\boldsymbol{Add}(object, modification, the\ k_{th}\ positions, size\ of\ probe)$
  End
End

Note [1]: Temporary variable: $the\ k_{th}\ part$, $exit\ wave$, $difference$, $modification$ and $revised\ exit\ wave$

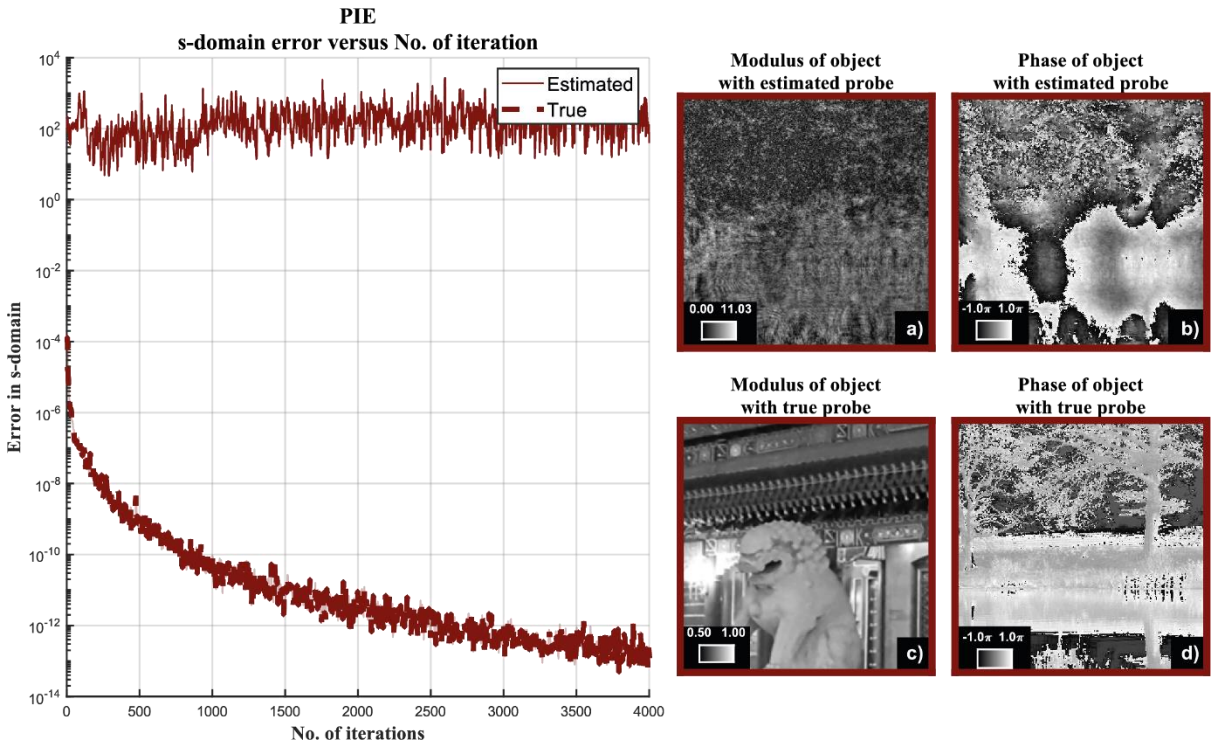


Figure 3. 15. The reconstruction results of PIE with estimated and true probes. The outcome of reconstruction with estimated probe is shown in (a) and (b), while the result from true probe is shown in (c) and (d). As shown in the figure, PIE's results highly depend on the quality of estimated probe. An accurate probe helps PIE bringing the error down to 10^{-12} *magnitude, which is significantly better than the result of an estimated probe.*

3.3.2. Extended PIE (ePIE)

Extended PIE (ePIE) was developed in 2009 by Andrew Maiden and John Rodenburg[59]. It extends the updating approach to the illumination function ($\mathbf{P}_{\vec{r}}$). Since the illumination function is also retrieved during the reconstruction, the accuracy of estimated illumination does not significantly affect the reconstruction quality.

To update the part of object covered by the probe at the k^{th} position ($\mathbf{O}_{\vec{r},k}{}'$) with the corresponding exit wave revised by the *f-constraint* ($\boldsymbol{\psi}'_{\vec{r},k}$), a naïve updating equation is written as:

$$\mathbf{O}'_{\vec{r},k} = \frac{\boldsymbol{\psi}'_{\vec{r},k}}{\mathbf{P}_{\vec{r}}} = \frac{\mathbf{P}_{\vec{r}}{}^{*}\boldsymbol{\psi}'_{\vec{r},k}}{|\mathbf{P}_{\vec{r}}|^2} \qquad \textit{eq 3. 46}$$

This updating function assumes the probe is correct, hence all variation on the exit wave should be all adapted to the $\mathbf{O}_{\vec{r},k}{}'$. However, this function is poorly conditioned at the area covered by the dim part of probe[56]. Any pixel of probe with small modulus can magnify the variation of the exit wave and lead to fluctuation or even instable performance. To prevent this from happening, this updated result is mitigated by the original object part ($\mathbf{O}_{\vec{r},k}$), gives:

$$\mathbf{O}'_{\vec{r},k} = (1 - \boldsymbol{\mu}_{\vec{r}}) \cdot \mathbf{O}_{\vec{r},k} + \boldsymbol{\mu}_{\vec{r}} \cdot \frac{\mathbf{P}_{\vec{r}}{}^{*} \boldsymbol{\psi}'_{\vec{r},k}}{|\mathbf{P}_{\vec{r}}|^2} \qquad \textit{eq 3. 47}$$

Which can be written as:

$$\mathbf{O}'_{\vec{r},k} = \mathbf{O}_{\vec{r},k} + \boldsymbol{\mu}_{\vec{r}} \cdot \left(\frac{\mathbf{P}_{\vec{r}}{}^{*} \boldsymbol{\psi}'_{\vec{r},k}}{|\mathbf{P}_{\vec{r}}|^2} - \mathbf{O}_{\vec{r},k} \right) \qquad \textit{eq 3. 48}$$

Substitute $\mathbf{O}_{\vec{r},k} \cdot |\mathbf{P}_{\vec{r}}|^2 = \boldsymbol{\psi}_{\vec{r},k} \cdot \mathbf{P}_{\vec{r}}{}^{*}$, the updating equation is obtained as follows:

$$\mathbf{O}'_{\vec{r},k} = \mathbf{O}_{\vec{r},k} + \boldsymbol{\mu}_{\vec{r}} \cdot \frac{\mathbf{P}_{\vec{r}}{}^{*}}{|\mathbf{P}_{\vec{r}}|^2} \cdot \left(\boldsymbol{\psi}'_{\vec{r},k} - \boldsymbol{\psi}_{\vec{r},k} \right) \qquad \textit{eq 3. 49}$$

A well-designed weighting factor $\boldsymbol{\mu}_{\vec{r}}$ should approach to 1 for the pixels that are well illuminated by the probe and approach to 0 for the poorly illuminated pixels. In ePIE, this weighting factor is set as a scaled intensity of probe, which is normalised by its brightest pixel.

$$\boldsymbol{\mu}_{\vec{r}} = \alpha \frac{|\mathbf{P}_{\vec{r}}|^2}{|\mathbf{P}_{\vec{r}}|^2_{max}} \qquad \textit{eq 3. 50}$$

Hence the standard k^{th} object part updating function of ePIE is:

$$\mathbf{O}'_{\vec{r},k} = \mathbf{O}_{\vec{r},k} + \alpha \frac{\mathbf{P}_{\vec{r}}{}^{*}}{|\mathbf{P}_{\vec{r}}|^2_{max}} \cdot \left(\boldsymbol{\psi}'_{\vec{r},k} - \boldsymbol{\psi}_{\vec{r},k} \right) \qquad \textit{eq 3. 51}$$

Comparing with PIE, the biggest improvement of ePIE is that it also reconstructs the illumination function ($\mathbf{P}_{\vec{r}}$). The derivation of probe updating function is similar as above, hence is omitted here. The standard probe updating function of ePIE is:

$$\mathbf{P}'_{\vec{r}} = \mathbf{P}_{\vec{r}} + \beta \frac{\mathbf{O}_{\vec{r},k}{}^{*}}{\left|\mathbf{O}_{\vec{r},k}\right|^2_{max}} \cdot \left(\boldsymbol{\psi}'_{\vec{r},k} - \boldsymbol{\psi}_{\vec{r},k}\right) \qquad \textit{eq 3. 52}$$

Another difference of ePIE is a shuffle on the scanning sequence at the beginning of each iteration. Such a modification is recommended to prevent a drifting probe during the reconstruction[59]. A flow chart and pseudo code for ePIE is given to help understanding this process. The reconstruction results of ePIE are given in *Figure 3. 17*.

Pseudocode 3. 12: *Extended Ptychography Iterative Engine (ePIE)*

Input: measured diffraction pattern ($intensity$), scanning positions ($positions$), guessed object ($object$), guessed probe ($probe$), No. of iterations (N), Parameter (α, β)

Output: revised object ($revised\ object$), revised probe ($revised\ probe$)

For (n=1: N) **do**
  $positions = \boldsymbol{shuffle}(positions)$
  For (k=1: total number of $positions$) **do**
    $the\ k_{th}\ part = \boldsymbol{Cut}(revised\ object,\ the\ k_{th}\ positions,\ size\ of\ probe)$
    $exit\ wave = the\ k_{th}\ part\ \cdot probe$
    $revised\ exit\ wave = \boldsymbol{\mathcal{P}}_f(exit\ wave, intensity)$
    $difference = revised\ exit\ wave - exit\ wave$
    $modification = \alpha \times \frac{probe^*}{|probe|^2_{max}} \times difference$
    $revised\ object =$ $\boldsymbol{Add}(object, modification, the\ k_{th}\ positions, size\ of\ probe)$
    $revised\ probe = probe + \beta \times \frac{the\ k_{th}\ part^*}{|the\ k_{th}\ part|^2_{max}} \times difference$
  End
End

Note [1]: Temporary variable: $the\ k_{th}\ part, exit\ wave, revised\ exit\ wave, modification$

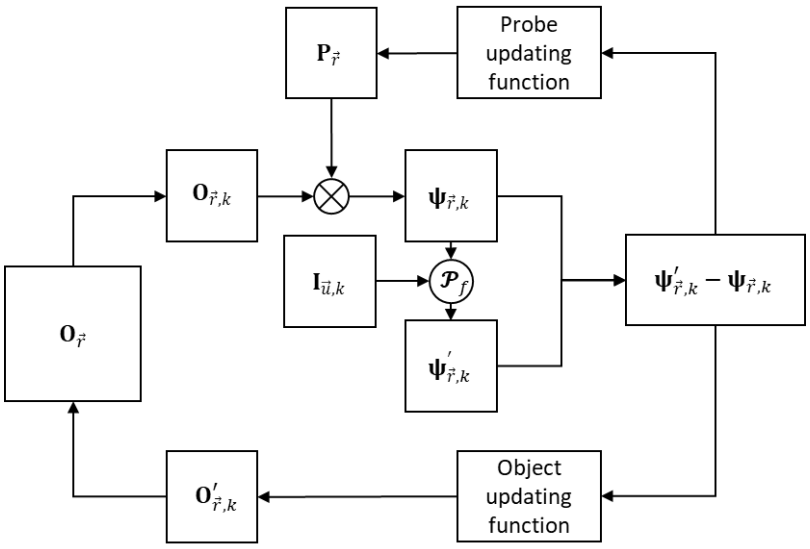


Figure 3. 16. The flow chart of ePIE. Comparing to Figure 3. 13*, the main difference between PIE and ePIE is an additional probe updating branch. The updating functions are given by eq 3. 51 and eq 3. 52.*

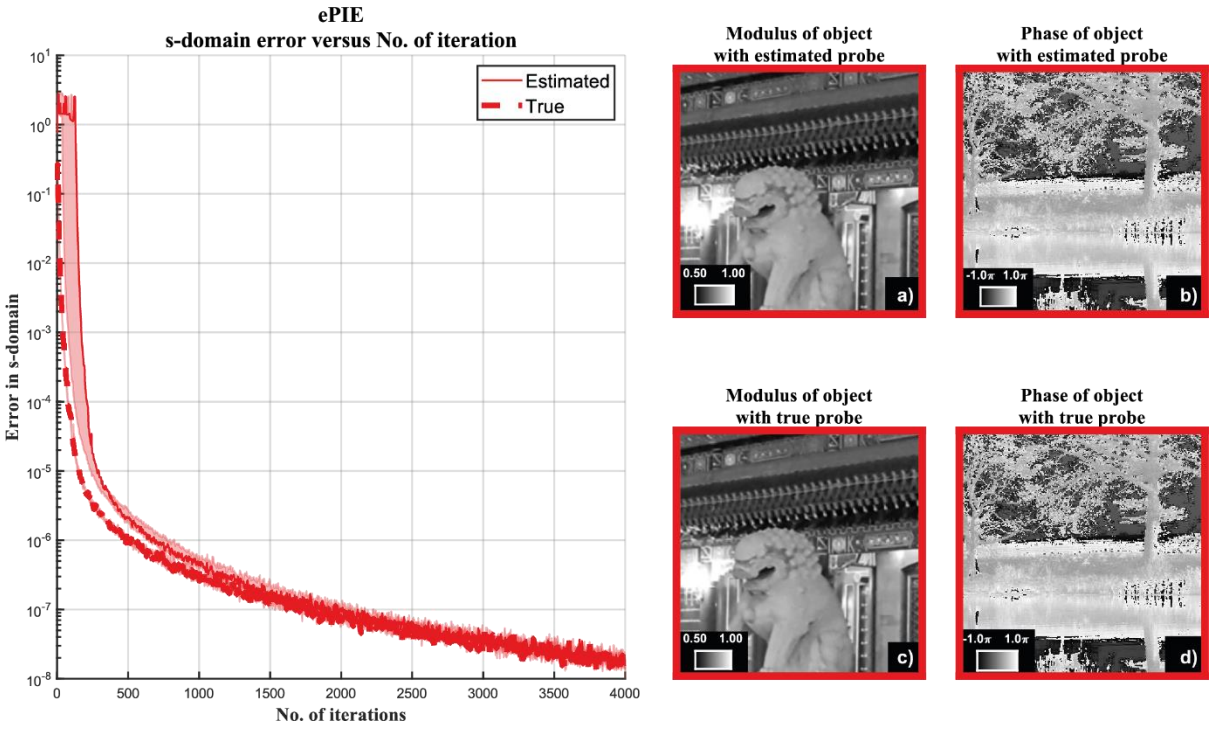


Figure 3. 17. The reconstruction result of ePIE with estimated and true probes. The outcome of reconstruction with estimated probe is shown in (a) and (b), while the result from true probe is shown in (c) and (d). Since the illumination is updated in ePIE, the accuracy of initial probe does not have significant influence on the result. Both initial probes lead to descent reconstructions. The error for true and estimated probe stays at similar magnitude during the reconstruction.

3.3.3. Regularised PIE (rPIE)

In 2017, another modified version of PIE is developed as regularised PIE (rPIE). Regularisation is a common concept in machine learning[63,64] to prevent a neural network getting overfit to the training data. In the reconstruction of PIE family, the mechanism of the fluctuation of poorly illuminated pixels is similar to the overfitting, hence can be supressed by introducing a regularised term to the cost function. The derivation of rPIE updating functions is explained below.

All PIE algorithms come from a cost function as given by *eq 3. 53*, where $\mathbf{O}_{\vec{r},k}{}'$ denotes the updated object. The first term of this equation evaluates the difference between the present guessed object and probe pair with the exit wave revised by *f-constraint* ($\boldsymbol{\psi}_{\vec{r},k}$), while the second term evaluates the variation of object during the present updating process.

$$\mathcal{L}_O = \sum_{\vec{r}} \left|\mathbf{O}'_{\vec{r},k}\mathbf{P}_{\vec{r}} - \boldsymbol{\psi}'_{\vec{r},k}\right|^2 + \sum_{\vec{r}} \boldsymbol{\omega}_{O,\vec{r}} \left|\mathbf{O}'_{\vec{r},k} - \mathbf{O}_{\vec{r},k}\right|^2 \qquad \textit{eq 3. 53}$$

When a solution is found for a noiseless data set, the exit wave satisfying the s-constraint (i.e. $\mathbf{O}_{\vec{r},k}{}'\mathbf{P}_{\vec{r}}$) should be the same with the exit wave satisfying f-constraint (i.e. $\boldsymbol{\psi}'_{\vec{r},k}$), which gives zero for the first term. Meanwhile, when the solution is reached, there is no difference between the present ($\mathbf{O}_{\vec{r},k}$) and updated object ($\mathbf{O}'_{\vec{r},k}$). Hence the second term also equals zero. As expected, the cost function reaches zero when the solution is found. Nevertheless, the second term also provides an approach for adaptively tuning the updating speed. By assigning large $\boldsymbol{\omega}_{\vec{r}}$ to the pixels that are not well illuminated, this cost function can penalise any dramatic variation on those poorly illuminated area of object. Different definition of $\boldsymbol{\omega}_{O,\vec{r}}$ leads to different updating functions.

To find an updated object part at the k^{th} scan position ($\mathbf{O}_{\vec{r},k}{}'$) that can minimise this cost function, a common approach is calculate the gradient of this function with respect to this variable and find a value that gives zero gradient. Thus, we differentiate the cost function for this object part with respect to $\left(\mathbf{O}'_{\vec{r},k}\right)^*$ to obtain the gradient:

$$\frac{\partial \mathcal{L}_{O,k}}{\partial\left(\mathbf{O}'_{\vec{r},k}\right)^*} = \mathbf{P}_{\vec{r}}{}^*\left(\mathbf{O}_{\vec{r},k}{}'\mathbf{P}_{\vec{r}} - \boldsymbol{\psi}'_{\vec{r},k}\right) + \boldsymbol{\omega}_{O,\vec{r}}\left(\mathbf{O}_{\vec{r},k}{}' - \mathbf{O}_{\vec{r},k}\right) \qquad \textit{eq 3. 54}$$

Set this equation of gradient to zero and re-arrange it with respect to $\mathbf{O}'_{\vec{r},k}$:

$$\mathbf{O}_{\vec{r},k}{}' = \frac{\mathbf{P}_{\vec{r}}{}^{*}\boldsymbol{\psi}'_{\vec{r},k} + \boldsymbol{\omega}_{O,\vec{r}}\mathbf{O}_{\vec{r},k}}{|\mathbf{P}_{\vec{r}}|^2 + \boldsymbol{\omega}_{O,\vec{r}}} \qquad \textit{eq 3. 55}$$

Or:

$$\begin{aligned}\mathbf{O}_{\vec{r},k}{}' &= \mathbf{O}_{\vec{r},k} + \frac{\mathbf{P}_{\vec{r}}{}^{*}\boldsymbol{\psi}'_{\vec{r},k} - |\mathbf{P}_{\vec{r}}|^2\mathbf{O}_{\vec{r},k}}{|\mathbf{P}_{\vec{r}}|^2 + \boldsymbol{\omega}_{O,\vec{r}}} \\ &= \mathbf{O}_{\vec{r},k} + \frac{\mathbf{P}_{\vec{r}}{}^{*}\left(\boldsymbol{\psi}'_{\vec{r},k} - \boldsymbol{\psi}_{\vec{r},k}\right)}{|\mathbf{P}_{\vec{r}}|^2 + \boldsymbol{\omega}_{O,\vec{r}}}\end{aligned} \qquad \textit{eq 3. 56}$$

All object updating functions used by PIE-algorithms can be derived from this function by choosing a suitable weighting factor ($\boldsymbol{\omega}_{O,\vec{r}}$). For instance, ePIE sets $\boldsymbol{\omega}_{O,\vec{r}} = \frac{1}{\alpha}|\mathbf{P}_{\vec{r}}|^2_{max} - |\mathbf{P}_{\vec{r}}|^2$, gives the object updating function as *eq 3. 57*.

$$\mathbf{O}'_{\vec{r},k} = \mathbf{O}_{\vec{r},k} + \alpha\frac{\mathbf{P}_{\vec{r}}{}^{*}\cdot\left(\boldsymbol{\psi}'_{\vec{r},k} - \boldsymbol{\psi}_{\vec{r},k}\right)}{|\mathbf{P}_{\vec{r}}|^2_{max}} \qquad \textit{eq 3. 57}$$

For rPIE, $\boldsymbol{\omega}_{O,\vec{r}} = \alpha(|\mathbf{P}_{\vec{r}}|^2_{max} - |\mathbf{P}_{\vec{r}}|^2)$ is utilised[56], which turns the denominator of *eq 3. 56* into a mediate value between the intensity of picked pixel ($|\mathbf{P}_{\vec{r}}|^2$) and the maximum intensity ($|\mathbf{P}_{\vec{r}}|^2_{max}$).

$$\mathbf{O}_{\vec{r},k}{}' = \mathbf{O}_{\vec{r},k} + \frac{\mathbf{P}_{\vec{r}}{}^{*}\left(\boldsymbol{\psi}'_{\vec{r},k} - \boldsymbol{\psi}_{\vec{r},k}\right)}{(1-\alpha)|\mathbf{P}_{\vec{r}}|^2 + \alpha|\mathbf{P}_{\vec{r}}|^2_{max}} \qquad \textit{eq 3. 58}$$

The illumination updating function is derived in a similar way. First, a cost function with respect to the probe is written as follows:

$$\mathcal{L}_P = \sum\nolimits_k \left|\mathbf{O}_{\vec{r},k}\mathbf{P}'_{\vec{r}} - \boldsymbol{\psi}'_{\vec{r},k}\right|^2 + \sum\nolimits_k \boldsymbol{\omega}_{P,\vec{r}}\left|\mathbf{P}'_{\vec{r}} - \mathbf{P}_{\vec{r}}\right|^2 \qquad \textit{eq 3. 59}$$

Then the gradient at current scan position with respect to the probe is obtained by differentiating it with respect to $(\mathbf{P}'_{\vec{r}})^{*}$,

$$\frac{\partial \mathcal{L}_{P,k}}{\partial \left(\mathbf{P}'_{\vec{r}}\right)^*} = \mathbf{O}_{\vec{r},k}{}^*\left(\mathbf{O}_{\vec{r},k}\mathbf{P}'_{\vec{r}} - \boldsymbol{\psi}'_{\vec{r},k}\right) + \boldsymbol{\omega}_{P,\vec{r}}\left(\mathbf{P}'_{\vec{r}} - \mathbf{P}_{\vec{r}}\right) \qquad \textit{eq 3. 60}$$

Set this gradient to zero, and rearrange it with respect to $\mathbf{P}'_{\vec{r}}$, gives:

$$\mathbf{P}'_{\vec{r}} = \frac{\mathbf{O}_{\vec{r},k}{}^*\boldsymbol{\psi}'_{\vec{r},k} + \boldsymbol{\omega}_{P,\vec{r}}\mathbf{O}_{\vec{r},k}}{|\mathbf{P}_{\vec{r}}|^2 + \boldsymbol{\omega}_{P,\vec{r}}} \qquad \textit{eq 3. 61}$$

Then an updating function for probe can be found by setting $\boldsymbol{\omega}_{P,\vec{r}} = \beta\left(\left|\mathbf{O}_{\vec{r},k}\right|^2_{max} - \left|\mathbf{O}_{\vec{r},k}\right|^2\right)$

$$\mathbf{P}'_{\vec{r}} = \mathbf{P}_{\vec{r}} + \beta\frac{\mathbf{O}^*_{\vec{r}_k}\left(\boldsymbol{\psi}'_{\vec{r},k} - \boldsymbol{\psi}_{\vec{r},k}\right)}{\left|\mathbf{O}_{\vec{r},k}\right|^2_{max}} \qquad \textit{eq 3. 62}$$

Besides the modified updating functions, rPIE reconstruction process is exactly the same as ePIE. One can refer to *Figure 3. 16* for rPIE flowchart. The pseudo code of rPIE is given in ***Pseudocode 3. 13***. The reconstruction result from rPIE is given in *Figure 3. 18*.

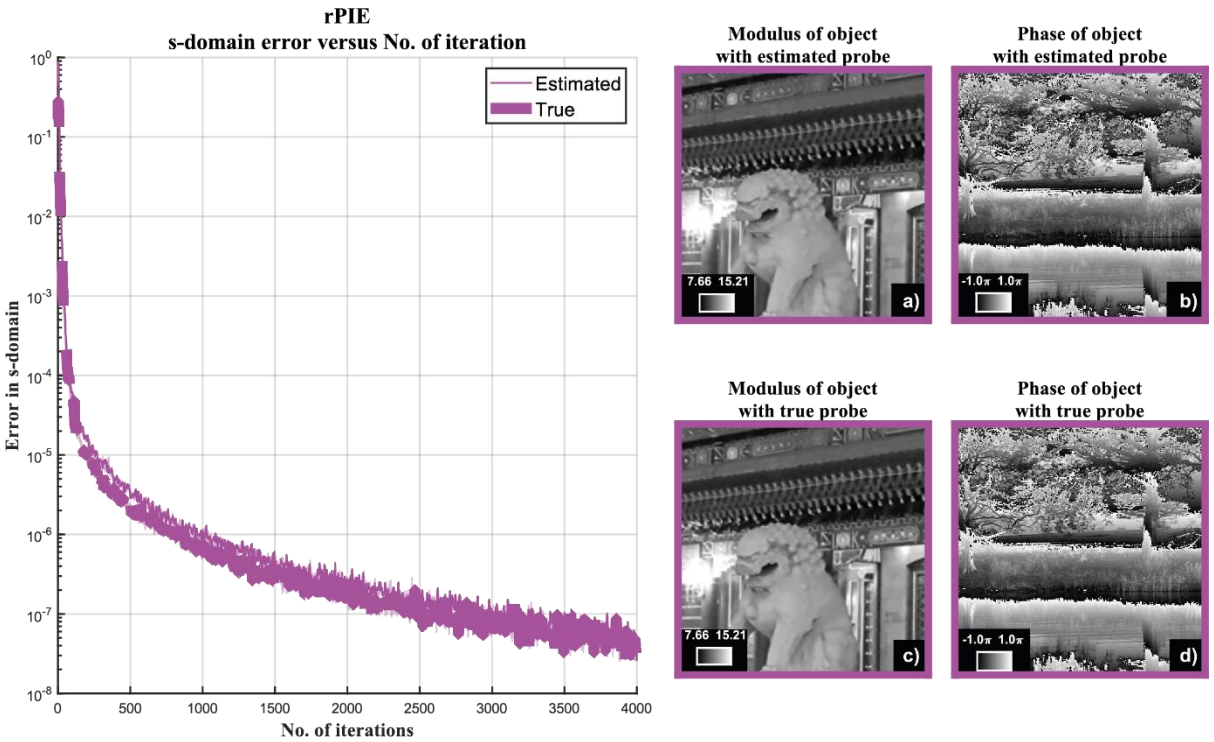


Figure 3. 18. The reconstruction result of rPIE with estimated and true probes. The outcome of reconstruction with estimated probe is shown in (a) and (b), while the result from true probe is shown in (c) and (d). Again, as the probe is simultaneously updated, its initial value does not affect the outcome significantly. rPIE achieves reasonable reconstructions in both cases.

Pseudocode 3. 13: *Regularised Ptychographical Iterative Engine (rPIE)*

Input: measured diffraction pattern ($intensity$), scanning positions ($positions$), guessed object ($object$), guessed probe ($probe$), No. of iterations (N), Parameter (α, β)

Output: revised object ($revised\ object$), revised probe ($revised\ probe$)

For (n=1: N) **do**

  $positions = \boldsymbol{shuffle}(positions)$

  For (k=1: total number of $positions$) **do**

    $the\ k_{th}\ part = \boldsymbol{Cut}(revised\ object,\ the\ k_{th}\ positions,\ size\ of\ probe)$

    $exit\ wave = the\ k_{th}\ part\ \cdot probe$

    $revised\ exit\ wave = \boldsymbol{\mathcal{P}}_f(exit\ wave, intensity)$

    $difference = revised\ exit\ wave - exit\ wave$

    $modification = \frac{probe^*}{(1-\alpha)\cdot|probe|^2+\alpha\cdot|probe|^2_{max}} \times difference$

    $revised\ object =$
    $\boldsymbol{Add}(object, modification, the\ k_{th}\ positions, size\ of\ probe)$

    $revised\ probe = probe + \frac{the\ k_{th}\ part^*}{(1-\beta)\cdot|the\ k_{th}\ part|^2+\beta\cdot|the\ k_{th}\ part|^2_{max}} \times$
    $difference$

  End

End

Note [1]: Temporary variable: $the\ k_{th}\ part$, $exit\ wave$, $revised\ exit\ wave$, $modification$

3.3.4. Momentum PIE (mPIE)

The latest member of PIE family is the momentum PIE (mPIE). Momentum is a concept in optimisation, which has been applied in training neural networks. Referring to the example in *Figure 3. 8*, a ball can overcome the pits with the accumulated momentum. The momentum (i.e. velocity) is reduced due to the friction and eventually lets the ball stay in the global minimum. During the reconstruction, mPIE estimates the current pixel-wise velocity by

comparing the updated object ($\mathbf{O}'_{\vec{r}}$) with its previous value ($\mathbf{O}_{\vec{r}}$). This current velocity is accumulated with a damped previous one ($\gamma \boldsymbol{v}_o$). Finally, the updated object is moved further along the direction defined by the over-all velocity as demonstrated by *eq 3. 63* and *eq 3. 64*, where the $\boldsymbol{v}_o$ is the previous object velocity and γ is the friction coefficient. The velocity ($\boldsymbol{v}_o$) is a matrix with the same size as object and initialise to zero in the beginning. The friction coefficient is a fractional number between 0 and 1. A high value indicates less damping effect; hence the velocity dissipates slower. A similar momentum updating is applied to the probe.

$$\boldsymbol{v}'_o = \gamma \boldsymbol{v}_o + \left(\mathbf{O}'_{\vec{r}} - \mathbf{O}_{\vec{r}}\right) \qquad \textit{eq 3. 63}$$

$$\mathbf{O}_{\vec{r}} = \mathbf{O}'_{\vec{r}} + \gamma \boldsymbol{v}'_o \qquad \textit{eq 3. 64}$$

As shown in *Figure 3. 19*, the workflow of mPIE is about the same with rPIE, except the momentum part existing in the end of iteration. This is different to the original paper, where momentum was applied periodically within the main update steps, which requires user choosing a parameter for it. We have demonstrated that applying momentum at the end of each update step is both more straightforward and more stable than the original scheme. Meanwhile, the original mPIE have 7 parameters, including two parameters for step size, two for regularisation, two for momentum updating and one for the period of applying momentum. We reduce the parameters to three, while still keep the most characteristics of mPIE. The pseudo code of momentum updating is given in ***Pseudocode 3. 14***. The result of mPIE is shown in *Figure 3. 20*.

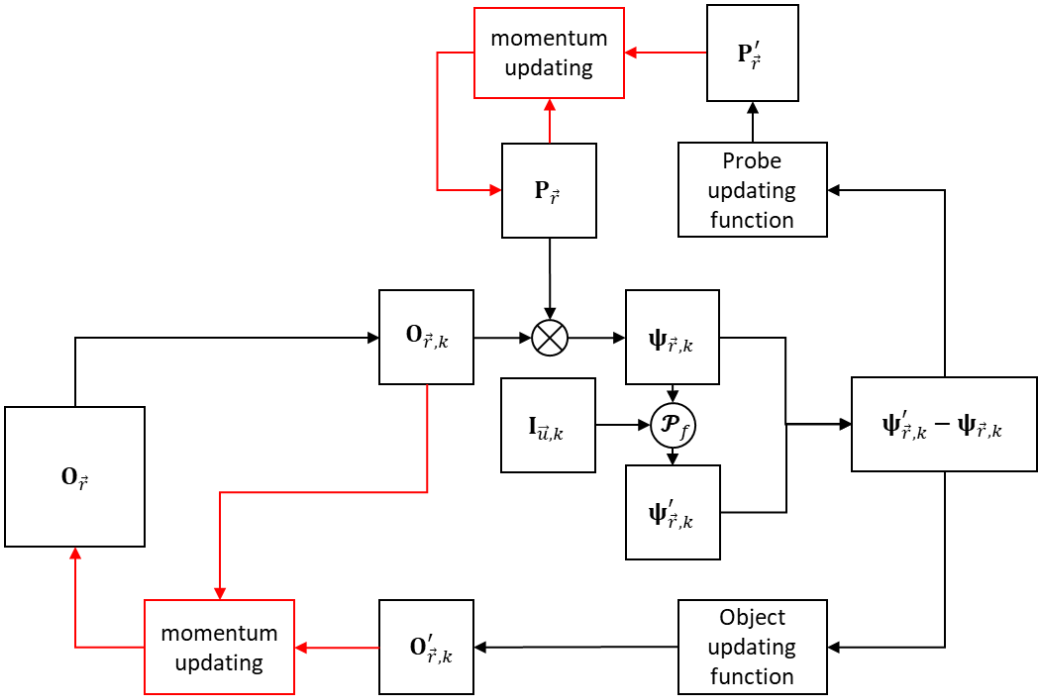


Figure 3. 19. The flow chart of mPIE. The difference between mPIE and ePIE (or rPIE) is the momentum updating modules, which are highlighted in red colour. In the original paper[56], this momentum module is activated after the object is updated with certain amount positions. we have demonstrated that applying momentum at the end of each update step is both more straightforward and more stable than the original scheme.

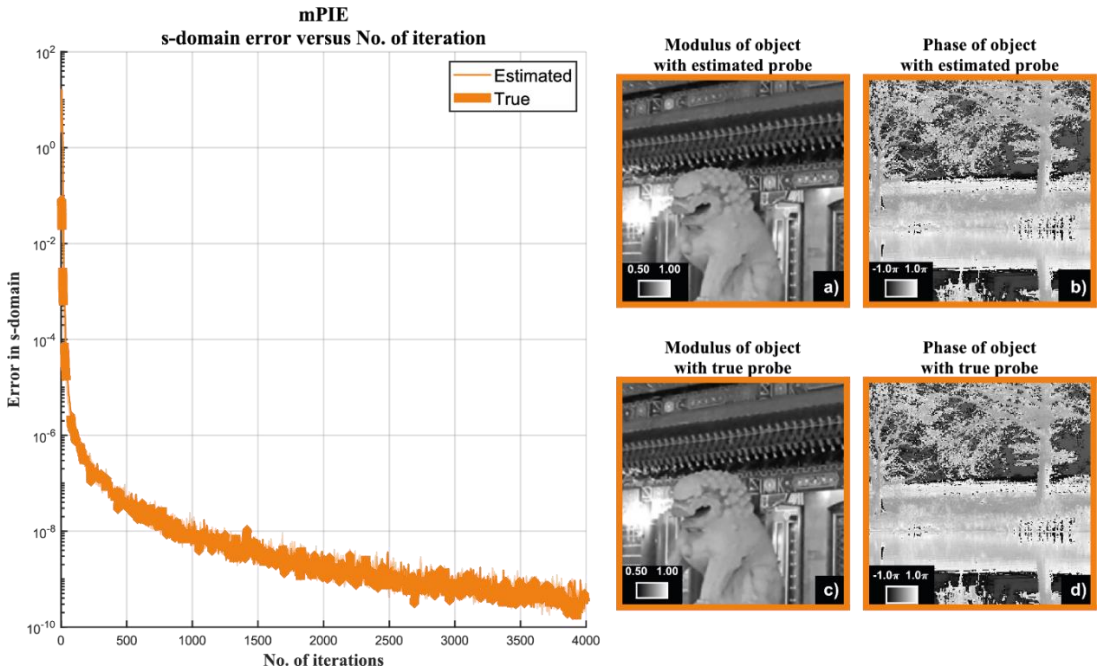


Figure 3. 20. The reconstruction result of mPIE with estimated and true probes. The outcome of reconstruction with estimated probe is shown in (a) and (b), while the result from true probe is shown in (c) and (d). The accuracy of initial probe does not affect the outcome, mPIE success with both initial guessed probes.

Pseudocode 3. 14: *Momentum Ptychographical Iterative Engine (mPIE)*

Input: measured diffraction pattern ($intensity$), scanning positions ($positions$), guessed object ($object$), guessed probe ($probe$), object velocity ($object\ velocity$), probe velocity($probe\ velocity$), No. of iterations (N), Parameter (α, β, γ)

Output: revised object ($revised\ object$), revised probe ($revised\ probe$)

1: $object\ velocity = \mathbf{0}$

2: $probe\ velocity = \mathbf{0}$

3: **For** (n=1: N) **do**

4: $positions = \boldsymbol{shuffle}(positions)$

5: $previous\ object = object$

6: $previous\ probe = probe$

7: **For** (k=1: total number of $positions$) **do**

8: $the\ k_{th}\ part = \boldsymbol{Cut}(revised\ object,\ the\ k_{th}\ positions,\ size\ of\ probe)$

9: $exit\ wave = the\ k_{th}\ part\ \cdot probe$

10: $revised\ exit\ wave = \boldsymbol{\mathcal{P}}_f(exit\ wave, intensity)$

11: $difference = revised\ exit\ wave - exit\ wave$

12: $modification = \frac{probe^*}{(1-\alpha)\cdot|probe|^2+\alpha\cdot|probe|^2_{max}} \times difference$

13: $object = \boldsymbol{Add}(object, modification, the\ k_{th}\ positions, size\ of\ probe)$

14: $probe = probe + \frac{the\ k_{th}\ part^*}{(1-\beta)\cdot|the\ k_{th}\ part|^2+\beta\cdot|the\ k_{th}\ part|^2_{max}} \times difference$

15: **End**

16: $object\ velocity = \gamma \times object\ velocity + (object - previous\ object)$

17: $probe\ velocity = \gamma \times probe\ velocity + (probe - previous\ probe)$

18: $revised\ object = object + \gamma \times object\ velocity$

19: $revised\ probe = probe + \gamma \times probe\ velocity$

20: **End**

Note [1]: Temporary variable: $the\ k_{th}\ part, exit\ wave, revised\ exit\ wave, modification$

3.3.5. Alternating direction method of multipliers (ADMM)

The alternating direction method of multipliers (ADMM) is a relatively new phase retrieval algorithm developed from the gradient descent concept[55, 65]. Just like any other ptychography algorithms, ADMM can be separated into two parts: the application of f-constraint and the application of s-constraint. The way of applying f-constraint in ADMM shares similarity with rPIE. The derivation starts with a cost function given below:

$$\mathcal{L}_{\Psi} = \sum\nolimits_{k} \left| \left|\boldsymbol{\Psi}'_{\vec{u},k}\right| - \sqrt{\mathbf{I}_{\vec{u},k}} \right|^2 + \beta \sum\nolimits_{k} \left| \left|\boldsymbol{\Psi}'_{\vec{u},k}\right| - \left|\boldsymbol{\Psi}_{\vec{u},k}\right| \right|^2 \qquad \textit{eq 3. 65}$$

The first term is an error metric evaluating the Euclidean distance in *f-domain*, while the second term is regularisation term based the difference of exit waves before and after updating of exit waves. Following the process of deriving updating function from cost function, we perform a partial differentiation with respect to $\boldsymbol{\Psi}_{\vec{u},k}$, gives:

$$\frac{\partial \mathcal{L}_{\Psi,\mathrm{k}}}{\partial\left(\left|\boldsymbol{\Psi}'_{\vec{u},k}\right|^{*}\right)} = 2\left(\left|\boldsymbol{\Psi}'_{\vec{u},k}\right| - \sqrt{\mathbf{I}_{\vec{u},k}}\right) + 2\beta\left(\left|\boldsymbol{\Psi}'_{\vec{u},k}\right| - \left|\boldsymbol{\Psi}_{\vec{u},k}\right|\right) \qquad \textit{eq 3. 66}$$

Set its value to zero and calculate $\boldsymbol{\Psi}'_{\vec{u},k}$, gives:

$$\left|\boldsymbol{\Psi}'_{\vec{u},k}\right| = \frac{\sqrt{\mathbf{I}_{\vec{u},k}} + \beta\left|\boldsymbol{\Psi}_{\vec{u},k}\right|}{1+\beta} = \frac{1}{1+\beta}\sqrt{\mathbf{I}_{\vec{u},k}} + \left(1 - \frac{1}{1+\beta}\right)\left|\boldsymbol{\Psi}_{\vec{u},k}\right| \qquad \textit{eq 3. 67}$$

Which has a similar format of the rPIE updating functions. Such an updating function equivalents to a relaxed f-projection and can be re-written into the standard format of relaxed projection ($\boldsymbol{\mathcal{P}}_f^{\alpha}$) as shown below. It equals projection when $\beta = 0$ and remains as unchanged when β goes to infinity.

$$\boldsymbol{\Psi}'_{\vec{u},k} = \boldsymbol{\mathcal{P}}_f^{\alpha}\left(\boldsymbol{\Psi}_{\vec{u},k}\right) = (1-\alpha)\left|\boldsymbol{\Psi}_{\vec{u},k}\right| + \alpha\sqrt{\mathbf{I}_{\vec{u},k}}, \qquad where\ \alpha = \frac{1}{1+\beta} \qquad \textit{eq 3. 68}$$

The *s-constraint* is applied onto the multiplier rather than exit waves, though they are strongly related to each other and even interchangeable under some assumptions as shown later. The

guessed exit waves are revised by a *relaxed s-reflection* ($\boldsymbol{\mathcal{R}}_s^{\beta}$) of multipliers. Its outcome is equivalent to projection when $\beta = 0$ and equals to reflection when $\beta = 1$.

Finally, the *f-* and *s-constraints* are connected to each other by interfaces that relate multipliers ($\boldsymbol{\lambda}_{\vec{r}}$) and exit waves ($\boldsymbol{\psi}_{\vec{r}}$). However, the original arrangement of ADMM, which is given in ***Pseudocode 3. 15***, is not easy to follow due to the successive converting between exit waves and multipliers. For revealing the logic beneath ADMM, its workflow is re-arranged as shown in *Figure 3. 21*. This rearranged format converges in a same rate as the original one under two assumptions. The first assumption is the initial value of multiplier is zero, which is common initialisation of ADMM. The second assumption is that swapping the order of projecting to s- and f- constraints has no significant influence on the reconstruction in a long run, which is also true in most of the applications[46].

Referring to the re-arranged format in *Figure 3. 21*, the "*interface s to f" module* is a self-updating for multiplier. It sets the multiplier to a vector pointing from the s-projection towards the original value. Then, this vector is added to the exit wave in "*interface f to s*" module and drives it away from the s-constraint. The multipliers work as memory terms that accumulate the previous error to the current exit wave. Such a mechanism perturbates the exit wave when it approaches to a stable point and prevent it from stagnation. To approve the re-arrangement does not affect the convergency of ADMM, both the original and the re-arranged ADMM are applied to reconstruct the simulated data with $\beta = 0.2$ with 3000 iterations with the estimated probe. The result is demonstrated in *Figure 3. 22.*

Pseudocode 3. 15: *original Alternating direction method of multipliers (ADMM)*

Input: measured diffraction pattern ($intensity$), scanning positions ($positions$), guessed object ($object$), guessed probe ($probe$), object velocity ($object\ velocity$), probe velocity($probe\ velocity$), No. of iterations (N), Parameter (β)

Output: revised object ($revised\ object$), revised probe ($revised\ probe$)

1: $multiplier = \mathbf{0}$

2: **For** (n=1: N) **do**

3: **For** (k=1: total number of $positions$) **do**

4: $the\ k_{th}\ part = \boldsymbol{Cut}(revised\ object,\ the\ k_{th}\ positions,\ size\ of\ probe)$

5: $the\ k_{th}\ exit\ wave = the\ k_{th}\ part\ \cdot probe$

6: $FT\ exit\ waves = \boldsymbol{\mathcal{F}}(\ the\ k_{th}\ exit\ waves)$

7: $FT\ exit\ waves = \frac{\sqrt{intensity}+\beta|FT\ exit\ waves|}{1+\beta} \cdot \exp\left(j \cdot angle(FT\ exit\ wave)\right)$

8: $the\ k_{th}\ exit\ waves = \boldsymbol{\mathcal{F}^{-1}}(FT\ exit\ waves)$

9: $the\ k_{th}\ multiplier =\ the\ k_{th}\ exit\ waves + \frac{multiplier}{\beta}$

10: **End**

11: $[exit\ wave, revised\ probe, revised\ object] = \boldsymbol{\mathcal{P}}_{\mathrm{s}}(multiplier, positions, probe)$

12: $multiplier =\ \beta \cdot (multiplier - exit\ wave)$

13: $exit\ waves = exit\ waves - multiplier$

14: **End**

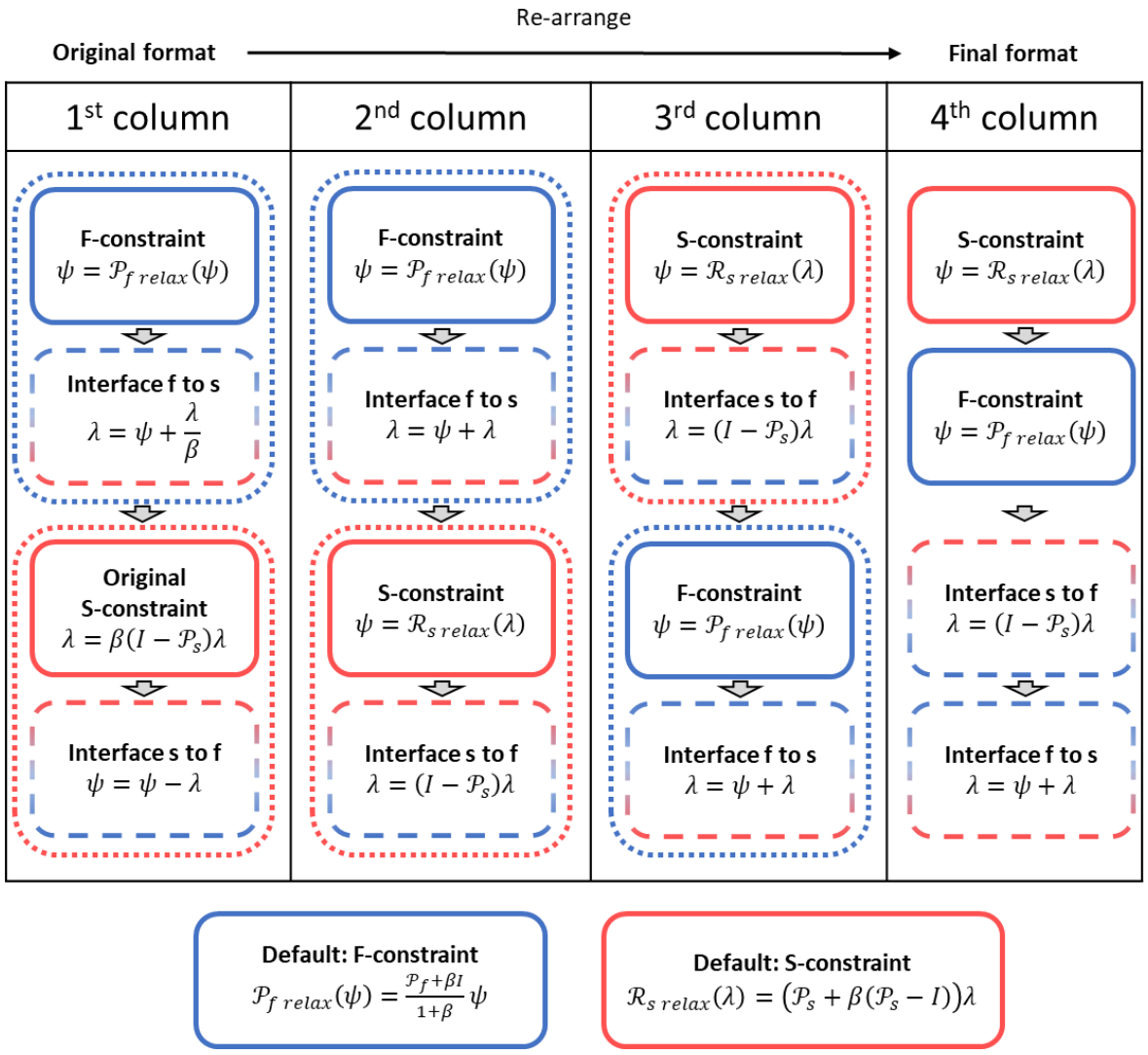


Figure 3. 21. A re-arrangement of ADMM algorithm. Each column (from top to bottom) represents a complete ADMM updating iteration, the updating process is gradually re-arranged from left to right. Modules relates to f-constraint is marked with blue colour and modules relates to s-constraint is marked with red colour. The expressions of relaxed f-projection ($\boldsymbol{\mathcal{P}}_f^{\beta}$) and relaxed s-reflection ($\boldsymbol{\mathcal{R}}_s^{\beta}$) are giving at the bottom of this figure.

The 1st column is the standard ADMM that matches ***Pseudocode 3. 15***. From the 1st to 2nd column, with assumption that initial value of multiplier is zero, the order of computing exit waves ($\boldsymbol{\psi}_{\vec{r}}$) and multipliers ($\boldsymbol{\lambda}_{\vec{r}}$) are swapped with corresponding adjustment on the updating functions to prevent any influence on the outcomes. From the 2nd to 3rd column, the order of applying f- and s-constraints are swapped together with their interfaces. For most of the iterative phase retrieval algorithms, such an adjustment has negligible effect in a long run. From 3rd to 4th column, as the f-constraint does not affect multiplier, it is brought to the front and leaves all functions of multiplier at the end of iteration. In this new format, exit waves

behaves more like a buffer, while the multiplier becomes the real exit wave and passed between iterations.

Pseudocode 3. 16: *modified Alternating Direction Method of Multipliers (ADMM)*

Input: measured diffraction pattern ($intensity$), scanning positions ($positions$), guessed object ($object$), guessed probe ($probe$), object velocity ($object\ velocity$), probe velocity($probe\ velocity$), No. of iterations (N), Parameter (β)

Output: revised object ($revised\ object$), revised probe ($revised\ probe$)

For (k=1: total number of $positions$) **do**
  $the\ k_{th}\ part = \boldsymbol{Cut}(object,\ the\ k_{th}\ positions,\ size\ of\ probe)$
  $the\ k_{th}\ multiplier = the\ k_{th}\ part\ \cdot probe$
End
For (n=1: N) **do**
  $[revised\ multiplier, revised\ probe, revised\ object] = \boldsymbol{\mathcal{P}}_s(multiplier, positions, probe)$
  For (k=1: total number of $positions$) **do**
    $the\ k_{th}\ exit\ wave = (1+\beta)\cdot revised\ multiplier - \beta \cdot multiplier$
    $FT\ exit\ waves = \boldsymbol{\mathcal{F}}(\ the\ k_{th}\ exit\ waves)$
    $FT\ exit\ waves = \frac{\sqrt{intensity}+\beta|FT\ exit\ waves|}{1+\beta}\cdot \exp\big(j\cdot angle(FT\ exit\ wave)\big)$
    $the\ k_{th} exit\ waves = \boldsymbol{\mathcal{F}^{-1}}(FT\ exit\ waves)$
  End
  $revised\ multiplier = muliplier - revised\ multiplier$
  $revised\ multiplier =\ exit\ waves +\ revised\ multiplier$
End

Moreover, the re-arranged format is also helpful to explain ADMM from set-projection point of view. In the final re-arranged format, one can combining the interfaces together and replace exit wave with an expression of multiplier. Hence, a recursive formula of multiplier is found as *eq 3. 69*.

$$\boldsymbol{\lambda}'_{\vec{r}} = \boldsymbol{\mathcal{P}}_f^{\beta}\boldsymbol{\mathcal{R}}_s^{\beta}\boldsymbol{\lambda}_{\vec{r}} + (I - \boldsymbol{\mathcal{P}}_s)\boldsymbol{\lambda}_{\vec{r}} \qquad \textit{eq 3. 69}$$

Derive the above equation to the most memory saving format.

$$
\begin{aligned}
\boldsymbol{\lambda}_{\vec{r},k+1} &= \left(\boldsymbol{\mathcal{P}}_f^{\beta}\boldsymbol{\mathcal{R}}_s^{\beta} + (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{P}}_{\mathrm{s}})\right)\boldsymbol{\lambda}_{\vec{r},k} \\
&= \left(\boldsymbol{\mathcal{P}}_f^{\beta}\boldsymbol{\mathcal{R}}_s^{\beta} - \boldsymbol{\mathcal{P}}_{\mathrm{s}} + \boldsymbol{\mathcal{I}}\right)\boldsymbol{\lambda}_{\vec{r},k} \\
&= \left(\boldsymbol{\mathcal{P}}_f^{\beta}\boldsymbol{\mathcal{R}}_s^{\beta} - \left((1+\beta)\boldsymbol{\mathcal{P}}_{\mathrm{s}} - \beta\boldsymbol{\mathcal{I}}\right) + \beta\boldsymbol{\mathcal{P}}_{\mathrm{s}} + (1-\beta)\boldsymbol{\mathcal{I}}\right)\boldsymbol{\lambda}_{\vec{r},k} \\
&= \left(\boldsymbol{\mathcal{P}}_f^{\beta}\boldsymbol{\mathcal{R}}_s^{\beta} - \boldsymbol{\mathcal{R}}_s^{\beta} + \beta\boldsymbol{\mathcal{P}}_{\mathrm{s}} + (1-\beta)\boldsymbol{\mathcal{I}}\right)\boldsymbol{\lambda}_{\vec{r},k} \\
&= \left(\left(\boldsymbol{\mathcal{P}}_f^{\beta} - I\right)\boldsymbol{\mathcal{R}}_s^{\beta} + \beta\boldsymbol{\mathcal{P}}_{\mathrm{s}} + (1-\beta)\boldsymbol{\mathcal{I}}\right)\boldsymbol{\lambda}_{\vec{r},k} \\
&= \left(\left(\boldsymbol{\mathcal{P}}_f^{\beta} - I\right)\boldsymbol{\mathcal{R}}_s^{\beta} + \boldsymbol{\mathcal{I}} + \beta(\boldsymbol{\mathcal{P}}_{\mathrm{s}} - \boldsymbol{\mathcal{I}})\right)\boldsymbol{\lambda}_{\vec{r},k}
\end{aligned}
$$

Last but not the least, the re-arranged format reduces the required memory. The original ADMM format is very memory intensive, as it requires storing both the exit waves and the multipliers, which are the largest variables in ptychography referring to ***Table 3. 3***. This drawback limits the implementation of ADMM onto large data sets. Since exit waves become intermediate variables in the re-arranged format, they can be wiped by the end of each iteration. In other words, the exit waves behave more like a buffer, and the multipliers become the true exit waves that is retrieved iteratively. The required memory size is nearly halved with this adjustment. The memory occupation can be found in ***Table 3. 3***.

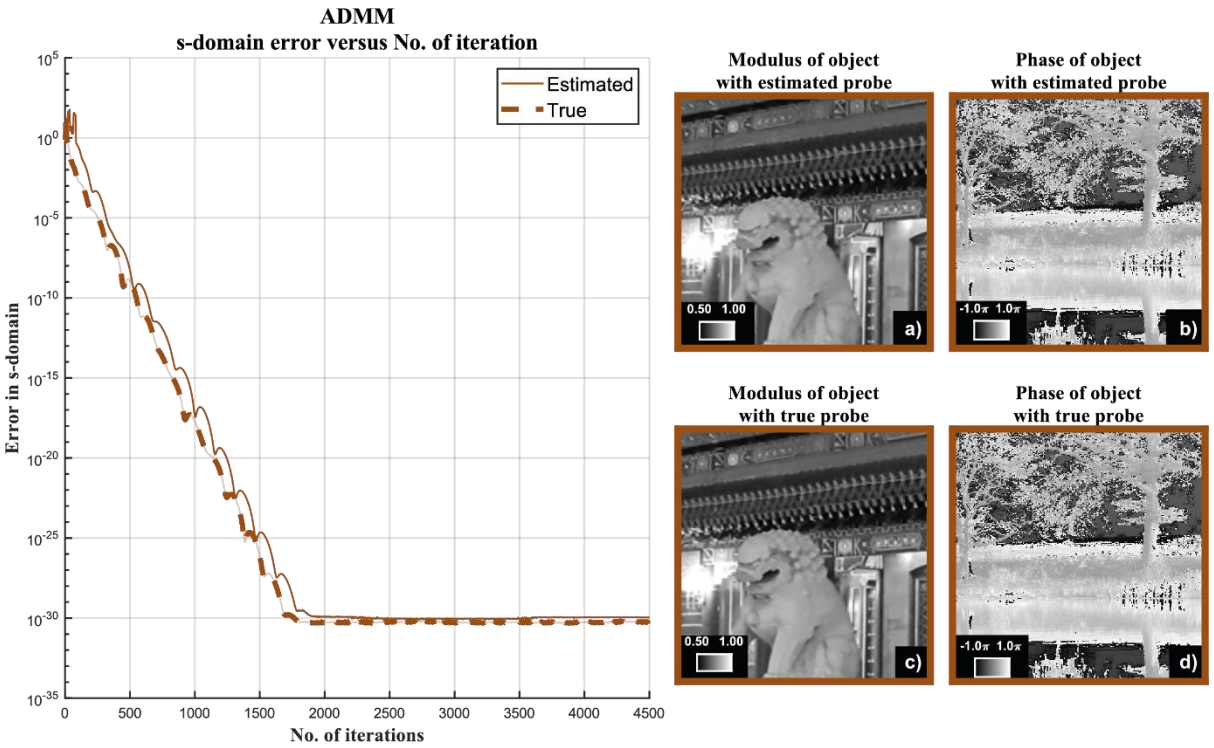


Figure 3. 22. The reconstruction result of ADMM with true and estimated probes. The outcome of reconstruction with true probe is shown in (a) and (b), while the result from guessed probe is shown in (c) and (d). The initial guess of probe has negligible influence on the reconstruction for ADMM. The error linearly decreases during the whole reconstruction. The modified version of ADMM (marked as dash-dotted line) has about the same trend as the original version.

3.3.6. Difference Mapping (DM)

Since the difference mapping was applied in ptychography in 2008[4], it has become one of the most competitive opponents of ePIE. It was the first algorithms tried to improve the reconstruction quality by retrieving illumination function during the reconstruction[59]. It is also considered as the ancestor of set-projection inspired algorithms[61]. The DM[4] recursive formula with respect to $\boldsymbol{\psi}_{\vec{r},k}$ is given as follow:

$$\boldsymbol{\psi}_{\vec{r},k+1} = \boldsymbol{\psi}_{\vec{r},k} + \boldsymbol{\mathcal{P}}_f\left(2\boldsymbol{\mathcal{P}}_s\left(\boldsymbol{\psi}_{\vec{r},k}\right) - \boldsymbol{\psi}_{\vec{r},k}\right) - \boldsymbol{\mathcal{P}}_s\left(\boldsymbol{\psi}_{\vec{r},k}\right) \qquad \textit{eq 3. 70}$$

Or:

$$\boldsymbol{\psi}_{\vec{r},k+1} = \left(\boldsymbol{\mathcal{I}} + \boldsymbol{\mathcal{P}}_f\boldsymbol{\mathcal{R}}_s - \boldsymbol{\mathcal{P}}_s\right)\boldsymbol{\psi}_{\vec{r},k} \qquad \textit{eq 3. 71}$$

Theoretically, DM should always converge to a true solution. To prove this, assume the $\boldsymbol{\psi}_{\vec{r},0}$ is a fixed point found by *eq 3. 70*, then we have:

$$\begin{aligned}\boldsymbol{\psi}_{\vec{r},0} &= \boldsymbol{\psi}_{\vec{r},0} + \boldsymbol{\mathcal{P}}_f\left(2\boldsymbol{\mathcal{P}}_s\left(\boldsymbol{\psi}_{\vec{r},0}\right) - \boldsymbol{\psi}_{\vec{r},0}\right) - \boldsymbol{\mathcal{P}}_s\left(\boldsymbol{\psi}_{\vec{r},0}\right)\\ 0 &= \boldsymbol{\mathcal{P}}_f\left(2\boldsymbol{\mathcal{P}}_s\left(\boldsymbol{\psi}_{\vec{r},0}\right) - \boldsymbol{\psi}_{\vec{r},0}\right) - \boldsymbol{\mathcal{P}}_s\left(\boldsymbol{\psi}_{\vec{r},0}\right)\\ \boldsymbol{\mathcal{P}}_s\left(\boldsymbol{\psi}_{\vec{r},0}\right) &= \boldsymbol{\mathcal{P}}_f\left(2\boldsymbol{\mathcal{P}}_s\left(\boldsymbol{\psi}_{\vec{r},0}\right) - \boldsymbol{\psi}_{\vec{r},0}\right)\end{aligned} \qquad \textit{eq 3. 72}$$

Which indicates the $\boldsymbol{\psi}_{\vec{r},0}$ satisfies both constraints at the same time, hence it is a solution. For later comparison, *eq 3. 71* is derived into a standard format as follows:

$$\begin{aligned}\boldsymbol{\psi}_{\vec{r},k+1} &= \left(\boldsymbol{\mathcal{I}} + \boldsymbol{\mathcal{P}}_f\boldsymbol{\mathcal{R}}_s - \boldsymbol{\mathcal{P}}_s\right)\boldsymbol{\psi}_{\vec{r},k}\\ &= \left(\boldsymbol{\mathcal{P}}_f\boldsymbol{\mathcal{R}}_s - \boldsymbol{\mathcal{P}}_s + \boldsymbol{\mathcal{I}}\right)\boldsymbol{\psi}_{\vec{r},k}\\ &= \left(\boldsymbol{\mathcal{P}}_f\boldsymbol{\mathcal{R}}_s - 2\boldsymbol{\mathcal{P}}_s + \boldsymbol{\mathcal{I}} + \boldsymbol{\mathcal{P}}_s\right)\boldsymbol{\psi}_{\vec{r},k}\\ &= \left(\boldsymbol{\mathcal{P}}_f\boldsymbol{\mathcal{R}}_s - (2\boldsymbol{\mathcal{P}}_s - \boldsymbol{\mathcal{I}}) + \boldsymbol{\mathcal{P}}_s\right)\boldsymbol{\psi}_{\vec{r},k}\\ &= \left(\boldsymbol{\mathcal{P}}_f\boldsymbol{\mathcal{R}}_s - \boldsymbol{\mathcal{R}}_s + \boldsymbol{\mathcal{P}}_s\right)\boldsymbol{\psi}_{\vec{r},k}\\ &= \left(\left(\boldsymbol{\mathcal{P}}_f - 1\right)\boldsymbol{\mathcal{R}}_s + \boldsymbol{\mathcal{P}}_s\right)\boldsymbol{\psi}_{\vec{r},k}\end{aligned} \qquad \textit{eq 3. 73}$$

The pseudo code of DM is given in ***Pseudocode 3. 17***.

Pseudocode 3. 17: *Difference Mapping (DM)*

Input: measured diffraction pattern ($intensity$), scanning positions ($positions$), guessed object ($object$), guessed probe ($probe$), No. of iterations (N)

Output: revised object ($revised\ object$), revised probe ($revised\ probe$)

1: **For** (k=1: total number of $positions$) **do**

2: $the\ k_{th}\ part = \boldsymbol{Cut}(object,\ the\ k_{th}\ positions,\ size\ of\ probe)$

3: $the\ k_{th}\ previous\ exit\ wave = the\ k_{th}\ part\ \cdot probe$

4: **End**

5: **For** (n=1: N) **do**

6: $[exit\ wave, revised\ probe, revised\ object] = \boldsymbol{\mathcal{P}}_s(previous\ exit\ wave, positions, probe)$

7: $revised\ exit\ wave = \boldsymbol{\mathcal{P}}_f(2 \cdot exit\ wave - previous\ exit\ wave, intensity)$

8: $previous\ exit\ wave = previous\ exit\ wave + revised\ exit\ wave - exit\ wave$

9: **End**

As shown in *eq 3. 70eq 3. 69*, all guessed exit waves from the previous iteration needed to be preserved for the next iteration. Moreover, some of the intermediate results, for instance the

outcome from $\mathcal{P}_s$ and $\mathcal{P}_f$, also need to be stored temporarily. This gives DM a relatively larger memory footprint than PIE-algorithms. Since the DM is not based on gradient descent concept, it has a relatively slow error reducing speed, and usually accompanied with fluctuations. The test results of DM are shown in *Figure 3. 23.*

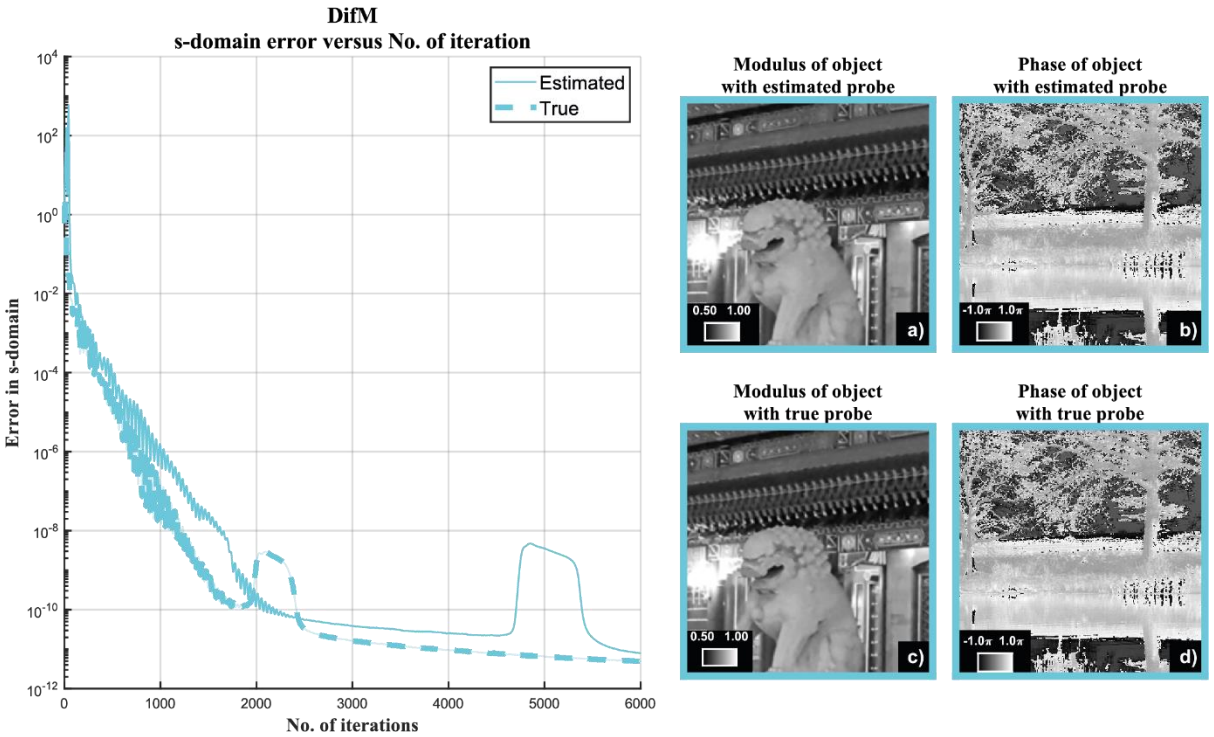


Figure 3. 23. The reconstruction result of DM with estimated and true probes. The outcome of reconstruction with estimated probe is shown in (a) and (b), while the result from true probe is shown in (c) and (d). The accuracy of initial probe does not affect the outcome of DM significantly. The error reduces in a linear trend for the first 2000 iterations. The fluctuation on the error plot is typical characteristic of DM.

3.3.7. Relaxed averaged alternating reflections (RAAR)

The relaxed average alternating reflections (RAAR) is based on set-projection concept. The pseudo code of RAAR is given in ***Pseudocode 3. 18***.

Pseudocode 3. 18: *Relaxed averaged alternative reflections (RAAR)*

Input: measured diffraction pattern ($intensity$), scanning positions ($positions$), guessed object ($object$), guessed probe ($probe$), No. of iterations (N), parameter (β)

Output: revised object ($revised\ object$), revised probe ($revised\ probe$)

For (k=1: total number of $positions$) **do**

  $the\ k_{th}\ part = \boldsymbol{Cut}(object,\ the\ k_{th}\ positions,\ size\ of\ probe)$

  $the\ k_{th}\ previous\ exit\ wave = the\ k_{th}\ part\ \cdot probe$

End

For (n=1: N) **do**

  $[exit\ wave, revised\ probe, revised\ object] = \boldsymbol{\mathcal{P}}_s(previous\ exit\ wave, positions, probe)$

  $revised\ exit\ wave = \boldsymbol{\mathcal{P}_f}(2 \cdot exit\ wave - previous\ exit\ wave, intensity)$

  $previous\ exit\ wave = \beta \cdot (previous\ exit\ wave + revised\ exit\ wave) + (1 - 2\beta) \cdot exit\ wave$

End

The updating function of exit wave in RAAR is given below:

$$\boldsymbol{\psi}_{\vec{r},k+1} = \left(\beta\left(\boldsymbol{\mathcal{I}} + \boldsymbol{\mathcal{P}}_f\boldsymbol{\mathcal{R}}_s\right) + (1-2\beta)\boldsymbol{\mathcal{P}}_s\right)\boldsymbol{\psi}_{\vec{r},k} \qquad \textit{eq 3. 74}$$

With some derivation works:

$$\begin{aligned}\boldsymbol{\psi}_{\vec{r},k+1} &= \left(\beta\boldsymbol{\mathcal{I}} + \beta\boldsymbol{\mathcal{P}}_f\boldsymbol{\mathcal{R}}_s + \boldsymbol{\mathcal{P}}_s - 2\beta\boldsymbol{\mathcal{P}}_s\right)\boldsymbol{\psi}_{\vec{r},k} \\ &= \left(\beta\boldsymbol{\mathcal{P}}_f\boldsymbol{\mathcal{R}}_s - \beta(2\boldsymbol{\mathcal{P}}_s - \boldsymbol{\mathcal{I}}) + \boldsymbol{\mathcal{P}}_s\right)\boldsymbol{\psi}_{\vec{r},k} \\ &= \left(\beta\boldsymbol{\mathcal{P}}_f\boldsymbol{\mathcal{R}}_s - \beta\boldsymbol{\mathcal{R}}_s + \boldsymbol{\mathcal{P}}_s\right)\boldsymbol{\psi}_{\vec{r},k} \\ &= \left(\beta\left(\boldsymbol{\mathcal{P}}_f - 1\right)\boldsymbol{\mathcal{R}}_s + \boldsymbol{\mathcal{P}}_s\right)\boldsymbol{\psi}_{\vec{r},k}\end{aligned} \qquad \textit{eq 3. 75}$$

Comparing to *eq 3. 73*, this equation almost identical to the DM updating function, except it has a relaxed parameter (β). This tiny variance, however, significantly improves the capability of preventing stagnation as shown in *Figure 3. 24*.

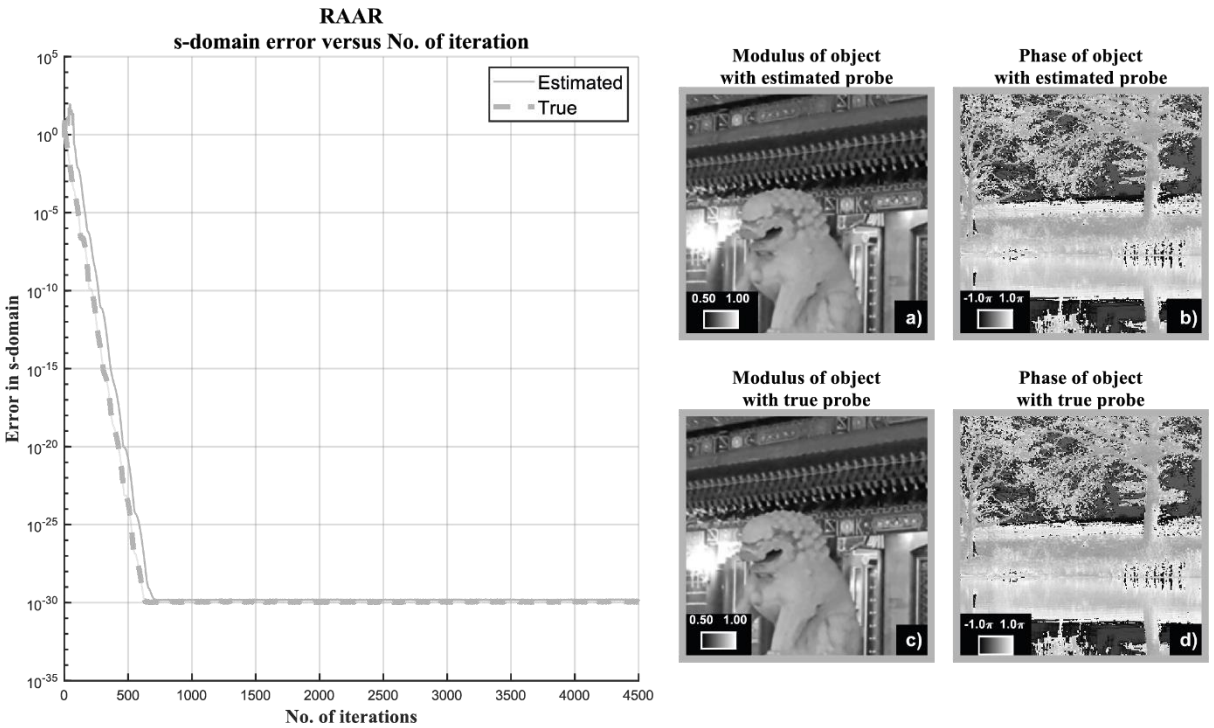


Figure 3. 24. The reconstruction result of RAAR with true and estimated probes. The outcome of reconstruction with true probe is shown in (a) and (b), while the result from guessed probe is shown in (c) and (d). RAAR has a linear error reduction until it hit the precision limit, which causes flat error trend at the end of reconstruction. The initial guess of probe has no observable influence on the output, which indicates RAAR reconstructs both the specimen and probe with good precision.

3.3.8. Hybrid projection and reflection (HPR)

The hybrid projection and reflection (HPR) is inspired by Fienup's basic input-output method (BIO) and hybrid input-output (HIO) method, which were developed in the early stage of computational microscopy[66]. The equation of HPR is shown in *eq 3. 76*. The β is a parameter, which give the algorithm more flexibility[53].

$$\boldsymbol{\psi}_{\vec{r},k+1} = \frac{1}{2}\left(\boldsymbol{\mathcal{R}}_s\left(\boldsymbol{\mathcal{R}}_f + (\beta-1)\boldsymbol{\mathcal{P}}_f\right) + \boldsymbol{I} + (1-\beta)\boldsymbol{\mathcal{P}}_f\right)\boldsymbol{\psi}_{\vec{r},k} \qquad \textit{eq 3. 76}$$

When the HPR is applied to ptychography, the current exit waves need to be computed first based on the guessed probe and object. Then their projection with respect to the *f-constraint* is denoted as $\boldsymbol{\mathcal{P}}_f\left(\boldsymbol{\psi}_{\vec{r},k}\right)$,. Meanwhile, their reflection ($\boldsymbol{\mathcal{R}}_f\left(\boldsymbol{\psi}_{\vec{r},k}\right)$) is computed as explained before. After that, a group of temporary exit waves are form with the following equation.

$$\boldsymbol{\psi}'_{\vec{r},k} = \boldsymbol{\mathcal{R}}_f \boldsymbol{\psi}_{\vec{r},k} - (1-\beta)\boldsymbol{\mathcal{P}}_f \boldsymbol{\psi}_{\vec{r},k} \qquad \textit{eq 3. 77}$$

These temporary exit waves are reflected with respect to the spatial domain constrain and update the probe and object at the same time by using equation (4). Finally, the s domain reflected exit waves are added up with the present exit waves and the adjusted frequency domain exit waves and divided by 2 to obtain the final updated exit waves, which will be used in the next iteration.

$$\boldsymbol{\psi}_{\vec{r},k+1} = \left(\boldsymbol{\mathcal{R}}_s \boldsymbol{\psi}'_{\vec{r},k}\right) + \boldsymbol{\psi}_{\vec{r},k} + (1-\beta)\boldsymbol{\mathcal{P}}_f \boldsymbol{\psi}_{\vec{r},k} \qquad \textit{eq 3. 78}$$

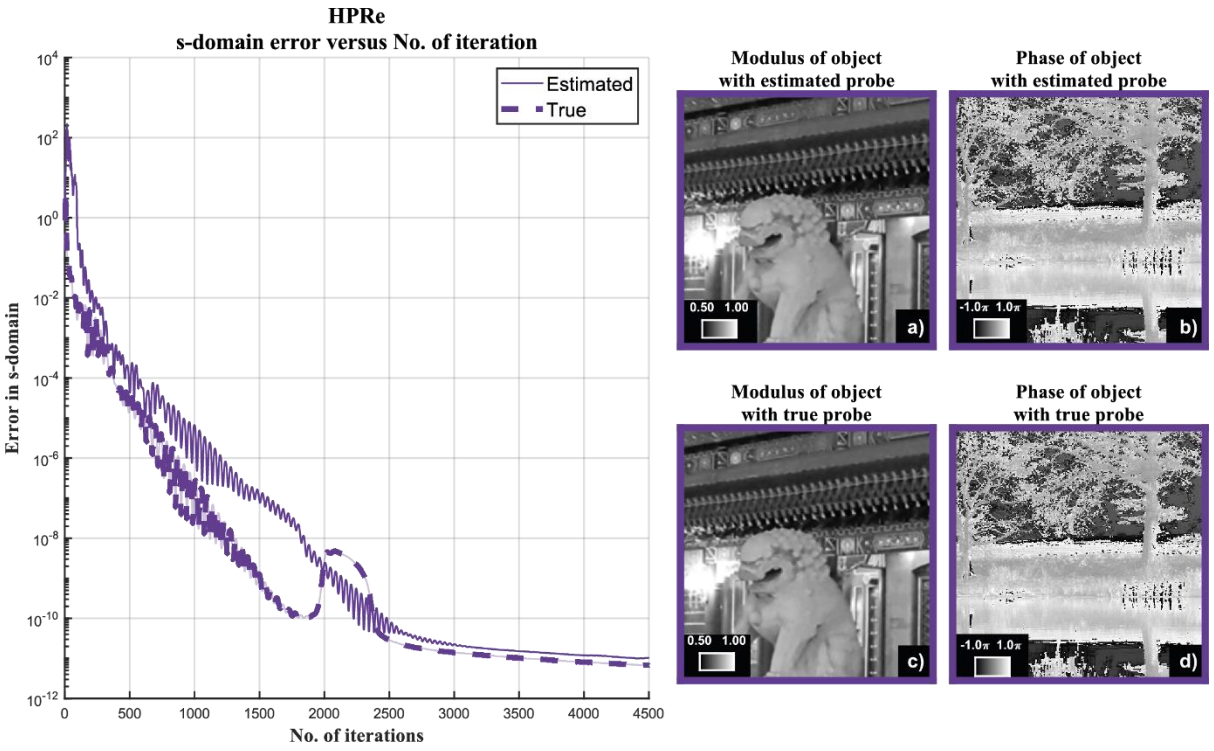


Figure 3. 25. The reconstruction result of HPR with estimated and true probes. The outcome of reconstruction with true probe is shown in (a) and (b), while the result from true probe is shown in (c) and (d). HPR has a linear error reduction until it hit the precision limit, which causes flat error trend at the end of reconstruction. The quality of initial probe has no observable influence when the reconstruction fully converges, which indicates HPR reconstructs both the specimen and probe with good precision.

3.3.9. Incremental accelerated proximal gradient (iAPG)

Proximal algorithm is an optimisation approach, which has been extensively studied[67,68]. It can produce a descent gradient for a constrained non-differentiable cost function, which makes it attractive for solving phase problem[67]. For ptychography, the phase retrieval can be expressed as optimising the following objective function[69]:

$$\min_{\boldsymbol{\Psi}_{\vec{r},k}} f\left(\boldsymbol{\Psi}_{\vec{r},k}\right) + g\left(\boldsymbol{\Psi}_{\vec{r},k}\right) \qquad \textit{eq 3. 79}$$

Where $f\left(\boldsymbol{\Psi}_{\vec{r},k}\right)$ represents the ***f-constraint*** formed by the measured intensities and $g\left(\boldsymbol{\Psi}_{\vec{r},k}\right)$ represents the ***s-constraint*** formed by the over lapping area. The gradient descent method can be applied to minimising $f\left(\boldsymbol{\Psi}_{\vec{r},k}\right)$ as explained in *section 3.3.3*.

$$\boldsymbol{\Psi}_{\vec{r},k}{}' = \boldsymbol{\Psi}_{\vec{r},k} - \gamma_k \nabla f\left(\boldsymbol{\Psi}_{\vec{r},k}\right) \qquad \textit{eq 3. 80}$$

On the other hand, $g\left(\boldsymbol{\Psi}_{\vec{r},k}\right)$ is not differentiable with non-convex constraint[67]. It can be written as an indicator function as follow:

$$g\left(\boldsymbol{\Psi}_{\vec{r},k}\right) = \begin{cases} 0 & \boldsymbol{\Psi}_{\vec{r},k} \in \mathbb{O} \\ \infty & \boldsymbol{\Psi}_{\vec{r},k} \notin \mathbb{O} \end{cases} \qquad \textit{eq 3. 81}$$

To minimise *eq 3. 79*, one must use proximal gradient method. Then the recursive equation of exit waves can be written as:

$$\begin{aligned} \boldsymbol{\Psi}_{\vec{r},k+1} &= \boldsymbol{prox}_g\left(\boldsymbol{\Psi}_{\vec{r},k}{}'\right) \\ &= \underset{\boldsymbol{\Psi}}{\operatorname{argmin}}\, g(\boldsymbol{\Psi}) + \left|\boldsymbol{\Psi} - \boldsymbol{\Psi}_{\vec{r},k}{}'\right|^2 \end{aligned} \qquad \textit{eq 3. 82}$$

To minimise *eq 3.82*, $\boldsymbol{\Psi}$ has to satisfy s-constraint ($\boldsymbol{\Psi} \in \mathbb{O}$), otherwise the first term becomes infinity. Moreover, $\boldsymbol{\Psi}$ needs to be close to the current guess (i.e. $\boldsymbol{\Psi}_{\vec{r},k}{}'$), hence the second term is also minimised. Combining *eq 3.80* and *eq 3.82*, the recursive equation for proximal gradient descent method is obtained:

$$\boldsymbol{\Psi}_{\vec{r},k+1} = \boldsymbol{prox}_g\left(\boldsymbol{\Psi}_{\vec{r},k} - \gamma_k \nabla f\left(\boldsymbol{\Psi}_{\vec{r},k}\right)\right) \qquad \textit{eq 3. 83}$$

This equation can be understood as minimising the differentiable function by gradient descent, then finding a new value that stays in the domain and is close to the present value. Researches[68] indicate the converging speed can be improved by introducing an adaptive parameter w, whose value (as defined in *eq 3.84*) approaches to unity as iteration goes. This gives the Incremental accelerated proximal gradient algorithm. Its pseudocode is given in ***Pseudocode 3.19***. Its simulation results are shown in *Figure 3.26.*

$$w = \frac{n-1}{n+2} \qquad \textit{eq 3. 84}$$

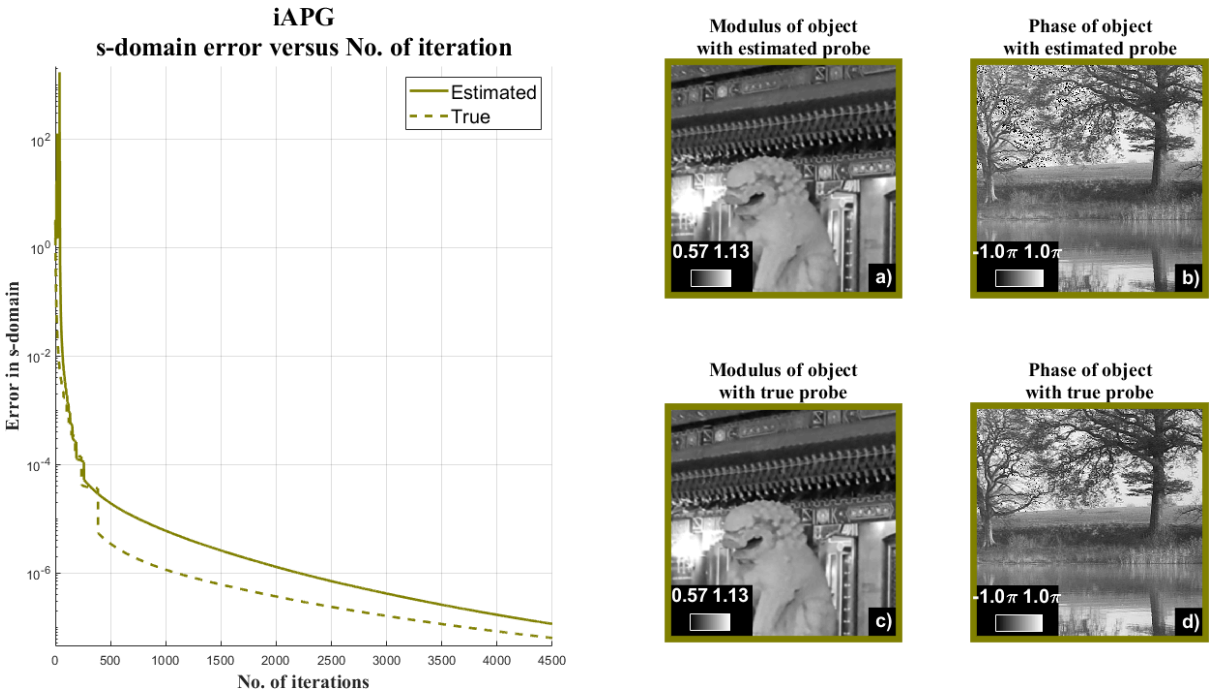


Figure 3. 26. The reconstruction result of iAPG with estimated and true probes. The outcome of reconstruction with true probe is shown in (a) and (b), while the result from true probe is shown in (c) and (d). iAPG converges fast in the beginning and gradually slows down as the iteration goes. It has a relatively slow over-all converging speed. The reconstruction quality is not significantly affected by the quality of initial guessed probe.

Pseudocode 3. 19: *Incremental accelerated proximal gradient (iAPG)*

Input: measured diffraction pattern ($intensity$), scanning positions ($positions$), guessed object ($object$), guessed probe ($probe$), No. of iterations (N), parameter (α, β, γ)

Output: revised object ($revised\ object$), revised probe ($revised\ probe$)

13: **For** (n=1: N) **do**

14: $w = \frac{n-1}{n+2}$

15: $revised\ object = object$

16: $revised\ probe = probe$

17: **For** (k=1: total number of $positions$) **do**

18: $the\ k_{th}\ part = \boldsymbol{Cut}(object,\ the\ k_{th}\ positions,\ size\ of\ probe)$

19: $the\ revised\ k_{th}\ part = \boldsymbol{Cut}(revised\ object,\ the\ k_{th}\ positions,\ size\ of\ probe)$

20: $exit\ wave = the\ k_{th}\ part\ \cdot probe$

21: $revised\ exit\ wave = the\ revised\ k_{th}\ part\ \cdot revised\ probe$

22: $FT\ exit\ wave = \boldsymbol{\mathcal{F}}\ (exit\ wave)$

23: $FT\ revised\ exit\ wave = \boldsymbol{\mathcal{F}}\ (revised\ exit\ wave)$

24: $FT\ combined\ exit\ wave = (1+w) \cdot FT\ revised\ exit\ wave - w \cdot FT\ exit\ wave$

25: $delta = FT\ combined\ exit\ wave - \boldsymbol{\mathcal{P}}_f(FT\ combined\ exit\ wave, intensity)$

26: $FT\ next\ exit\ wave = FT\ combined\ exit\ wave - \gamma \cdot delta$

27: $next\ exit\ wave = \boldsymbol{\mathcal{F}}^{-1}(FT\ next\ exit\ wave)$

28: $difference = next\ exit\ wave - revised\ exit\ wave$

29: $object = revised\ object$

30: $probe = revised\ probe$

31: $modification = \alpha \times \frac{probe^*}{|probe|^2_{max}} \times difference$

32: $revised\ object = \boldsymbol{Add}(object, modification, the\ k_{th}\ positions, size\ of\ probe)$

33: $revised\ probe = probe + \beta \times \frac{the\ k_{th}\ part^*}{|the\ k_{th}\ part|^2_{max}} \times difference$

34: **End**

35: **End**

3.3.10. Other existed algorithms

Besides the algorithms explained above, there are also some other ptychographic algorithms, such as proximal algorithms [70,67,71] , maximum likelihood optimization[72], convex relaxation to linearize the ptychography problem[73], momentum-accelerated Wirtinger flow[74] and flexible least-squares optimize[75]. However, they are either less influential as the algorithms explained in the previous sections, or still under developing, or not suitable for retrieving phase for ptychography, they are not discussed in this thesis.

3.4. Computer hardware background

All simulations and phase retrieval within this thesis are carried out using the MATLAB software. This software is well known for its efficiency in computing vectorised variables. Since most of the variables in diffraction imaging are in matrix style, they are benefited by the strength of MATLAB. Meanwhile, MATLAB offers various kinds of functions and toolboxes that assist with prototyping new algorithms. This section explains some important hardware limitations that strongly relates to the phase retrieving algorithms.

3.4.1. Benefits and limitations of parallel computation

The redundancy in collected data brings many advantages to ptychography, but it also puts a strict requirement on the hardware. Retrieving phase iteratively from a set of diffraction patterns is computation-intensive and demands enormous memory for storing necessary variables. Nowadays, the phase retrieving process are mostly done by Graphic Processing Unit (GPU), as they are designed for parallel computation hence faster than Central Processing Unit (CPU) in this application. However, unlike the CPU whose memory size can be increased by plugging more RAMs, the available memory for GPU (i.e. vRAM) is limited and cannot be modified. By the time this thesis is written, a CPU can access multiple 16G RAMs with a reasonable mother board comparing with only 12G vRAM in a top tier domestic GPU.

.

Although some specifically designed GPUs or cross fire technology are capable to provide more total memory[76], they are, firstly, not sharing the vRAM between different GPUs, and secondly, coming with a considerable price. Transferring variables between RAM and vRAM can prevent running out of memory, though it takes significant time and should be prevented for a good efficiency. Moreover, extra memory is implicitly demanded by other functions as buffers. The amount of implicit demand is usually proportional to the variable size in MATLAB. This leads to a conflict between the computing speed and available memory. As multiple variables need to be computed simultaneously to maximise the efficiency, but the memory for storing variables is limited by the finite vRAM. Such a conflict does not affect the outcome, but it varies the efficiency of a same code on different devices and data set.

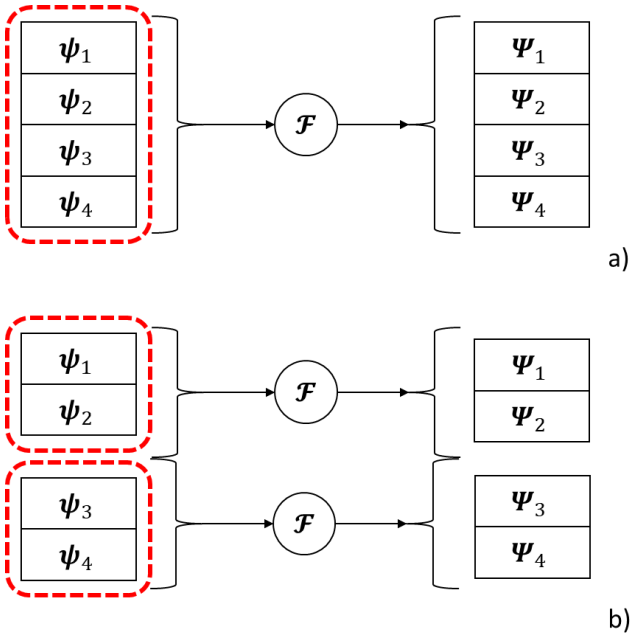


Figure 3. 27. The available memory size affects the efficiency of parallel computing. This figure uses Fourier transformation of guessed exit waves as an example. The available memory size is visualised as a red dotted outline. When the memory is sufficient (a), all exit waves are parallelly computed to maximise the efficiency. However, when the available memory is insufficient (b), the computation has to be applied onto several smaller batches. Giving the batch size has negligible effect on parallel computation in MATLAB, the number of computations dominate the consuming time. Therefore, a smaller memory size leads to more batches, hence consumes more time.

The required memory of different algorithms varies from one to another. One should put this into consideration as some algorithms may not be an option for a large data set, even though it had a promising outcome on smaller one. Therefore, a metric for assessing the required

memory is developed to illustrate this intrinsic characteristic of algorithms. The memory occupation is mainly caused by the matrix-type variables, including matrices to hold the specimen, illumination function, measured intensities, exit waves and their buffers. Their size is determined by both the size of the matrix and the data type. For two variables with equal matrix size, the one filled with complex numbers, e.g. the exit wave, occupies twice the size of the one filled with real numbers, e.g. the diffraction pattern. The smallest matrix within ptychography context is a matrix with probe size filled with real number. If the size of this matrix is defined as 1 unit ($unit_{pro}$), most of the variables in ptychography can be expressed as multiple of this unit as shown in *Table 3. 2*. To flesh out these variables, a data set generated by a 512 by 512 probe, 20 by 20 scanning grid and 30% step size is utilised as an example.

The only exception is the object, which can be related to this unit by the step size and scanning grid. For a step size equal to δ_s percent of the probe matrix size and a scanning grid with A by B positions, the object size ($unit_{obj}$) can be approximated by *eq 3. 85*. However, such a relationship complicates the expression. Hence $unit_{obj}$ is also considered as a basic unit. One can converting these units referring to *eq 3. 85*.

$$unit_{obj} = \left[AB{\delta_s}^2 - (A+B)\delta_s(\delta_s - 1) + (\delta_s - 1)^2\right] \cdot unit_{pro} \qquad \textit{eq 3. 85}$$

Table 3. 2: Variable size with examples

Variable	*Illumination*	*Specimen*	*Measured intensities*	*Exit waves*
Size	$2unit_{pro}$	$2unit_{obj}$	$400unit_{pro}$	$800unit_{pro}$
Elements	512×512 complex numbers	$\sim 34 \times 10^6$ complex numbers	$512 \times 512 \times 400$ real numbers	$512 \times 512 \times 400$ complex numbers
Example	4.2MB	67.1MB	838.9MB	1.68GB

As shown in the table, exit waves and measured intensities occupy most of the memory. Some algorithms requiring buffers for exit waves dramatically worsen the memory occupation. Without wisely spending the memory space, the limitation of vRAM is quickly reached in some scenarios

Table 3. 3. The memory occupation of different algorithms

Algorithm	Standard	With standard size unit	With given example
PIE, ePIE, rPIE	Least memory	$(4+K)unit_{pro}+2unit_{obj}$	914Mb
	Max efficiency	$(6+K)unit_{pro}+2unit_{obj}$	852Mb
mPIE	Least memory	$(8+K)unit_{pro}+6unit_{obj}$	868Mb
	Max efficiency	$(10+K)unit_{pro}+6unit_{obj}$	860Mb
ADMM original	Least memory	$(8+5K)unit_{pro}+6unit_{obj}$	4247Mb
	Max efficiency	$(6+7K)unit_{pro}+6unit_{obj}$	5885Mb
ADMM re-arranged, DM, RAAR	Least memory	$(8+3K)unit_{pro}+6unit_{obj}$	2570Mb
	Max efficiency	$(6+5K)unit_{pro}+6unit_{obj}$	4207Mb

3.4.2. Prevent running out of memory

There are several ways to prevent running out of memory. Firstly, the intermediate variables can share the same buffer and get updated once it has been used. Secondly, in some of the algorithms, the operation can be done in separate groups, though this will reduce the benefits brought by parallel computation. Thirdly, variables can be converted from double into single type when the accuracy is not dominated by the data precision. Fourthly, some variables can be removed and regenerated when it is necessary. Obviously, this will increase the computation time. Last but not the least, a careful re-arrangement of the computation order can also save memory space without significantly affecting results.

4. Error metric

After decades of development, various kinds of phase retrieval algorithms have been developed. Their unique characteristics give them different performance during the reconstruction. In this chapter the algorithms described in Chapter 3 are tested under a range of simulation scenarios. The chapter begins with a discussion of ambiguities that can occur in ptychographic reconstruction (Section 4.1) and of error metrics that can be used to assess the performance of ptychographic algorithms for real-world and simulated data. A new error metric we have developed is introduced (Section 4.2). Finally, the error metrics are used in a series of simulations to assess the performance of the algorithms detailed in Chapter 3.

4.1. Dealing with s-domain ambiguities

To quantitatively compare the quality of ptychographic reconstructions from simulated data, a direct comparison with the original ground truth object is required. However, inherent in the ptychographic process are ambiguities that satisfy all of the priori conditions on the object and probe reconstructions, but which deviate in a systematic way from the ground truth[56]. It is commonly seen that the difference between the guessed exit waves and their f-constraint has been reduced to a negligible level, while the resultant images are still not informative due to a combination of multiple kinds of ambiguities. As shown in *Figure 4. 1*, the pair of object and probe with ambiguity looks completely different with the true pair, though they can produce diffraction patterns those exactly fit the simulated ones. Unsurprisingly, a direct comparison between such an ambiguous image with the true one result a meaningless, large difference, which does not match with the comparison in f-domain. Therefore, these s-domain ambiguities must be removed before computing a s-domain error.

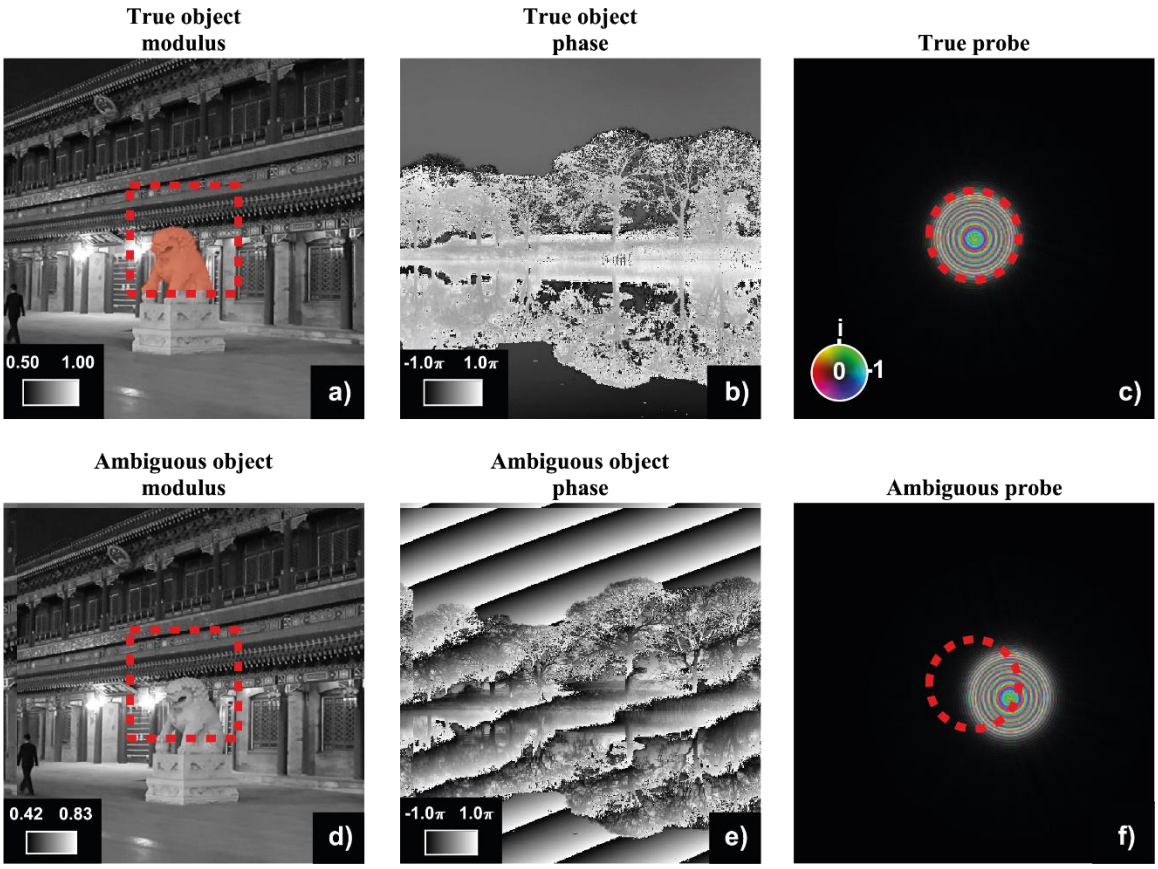


Figure 4. 1. An example of the influence of ambiguities. The top row shows the modulus (a), phase (b) of true object and true probe (c) utilised for simulating diffraction patterns, while the bottom row shows the corresponding properties of object and probe (e.g. (d) for modulus of object, (e) for phase of object and (f) for probe) after adding artificial ambiguities. The red dotted square indicates the area for computing s-domain difference, and the outline of an inside feature is highlighted for later use. The red dotted circles in (c) and (f) indicate the general spot location of true probe. In this example, one can identify the global shifting (e.g. the shift content inside red dotted square), the phase ramp (e.g. strips in (e)) and scaling factor (e.g. the difference in the colour bars of (a) and (c)).

Equation *eq 4. 1* sets out these ambiguities. The probe ($\mathbf{P}_{\vec{r}}$) and the k^{th} part of object ($\mathbf{O}_{\vec{r},k}$) reconstructed by any of the ptychographic algorithms together form a set of exit waves ($\boldsymbol{\psi}_{\vec{r},k}$), where $\vec{r}$ represents pixel coordinates and k represents the offset position of the object when the k^{th} diffraction pattern was recorded. The algorithm can be considered successful if the Fourier-transformed intensities of these exit waves match the recorded data. This condition is met by a range of reconstructed objects and probes related directly to the true probe and object: $\widehat{\mathbf{P}}_{\vec{r}}$ and $\widehat{\mathbf{O}}_{\vec{r}}$. The possible ambiguities include an *amplitude scaling constant* (a), a *constant phase offset* (e^{jc}), a *linear phase ramp* ($e^{jb\cdot\vec{r}}$) and a *global shift* ($+\vec{d}$ term in the subscript). Although these trivial ambiguities are neither determined by the quantity of the reconstructed exit waves, nor indicative of an unsuccessful reconstruction, they do affect the

appearance of images and the computation of an accurate error metric unless they are accounted for. (There is one more ambiguity, which is known as the “raster grid pathology”, caused by the regular scanning positions. This is usually prevented by adding some randomness to the scanning grid.) A process of automatically estimating and removing ambiguities in reconstructed images is explained in the following sections.

$$\boldsymbol{\psi}_{\vec{r},k} = \mathbf{P}_{\vec{r}}\mathbf{O}_{\vec{r},k} = \left(ae^{jc}e^{jb\cdot\vec{r}}\widehat{\mathbf{P}}_{\vec{r}+\vec{d}}\right)\left(a^{-1}e^{-jc}e^{-jb\cdot\vec{r}}\widehat{\mathbf{O}}_{\vec{r}+\vec{d},k}\right) \qquad \textit{eq 4. 1}$$

4.1.1. Global shifting ($+\vec{d}$)

The global shifting ambiguity appears as both the object and probe shift towards the same direction with the same distance as shown in *Figure 4. 2(d)-(f)*. During the reconstruction, as long as the bright spot of probe does not shift beyond the range defined by the matrix size of probe, this ambiguity only gives a shifting effect onto guessed exit waves, which appears as a phase ramp in its reciprocal space. Since applying the *f-constraint* only modifies the modulus, the phase ramp is not corrected in this process. Meanwhile, as both the object and probe shift with the same amount, the content in their contact area is not significantly affected by the global shifting. Hence the *s-constraint* also cannot correct this ambiguity.

To estimate the global shifting ambiguity accurately, the probe used for simulating diffraction patterns has to have zero boundaries. Then the ambiguity can be estimated through cross-correlation between the modulus of reconstructed and true probes[59,77]. Once the shifting vector is estimated, a counter shifting can be applied by introducing a phase ramp in its *f-domain* rather than shifting in the *s-domain*. In this way, the counter shift can be done in fraction of pixels, which gives a better accuracy. The counter shift is applied to both the object and probe. This ambiguity-eliminating process is visualised in *Figure 4. 2*. One should notice that this step usually requires a reasonably reconstructed probe, which may not be the case in the beginning of reconstruction. Without calibrating the shifting ambiguity, the rest ambiguity removing processes cannot act properly. This is the main cause of fluctuation in the *s-domain* error in the beginning of reconstruction.

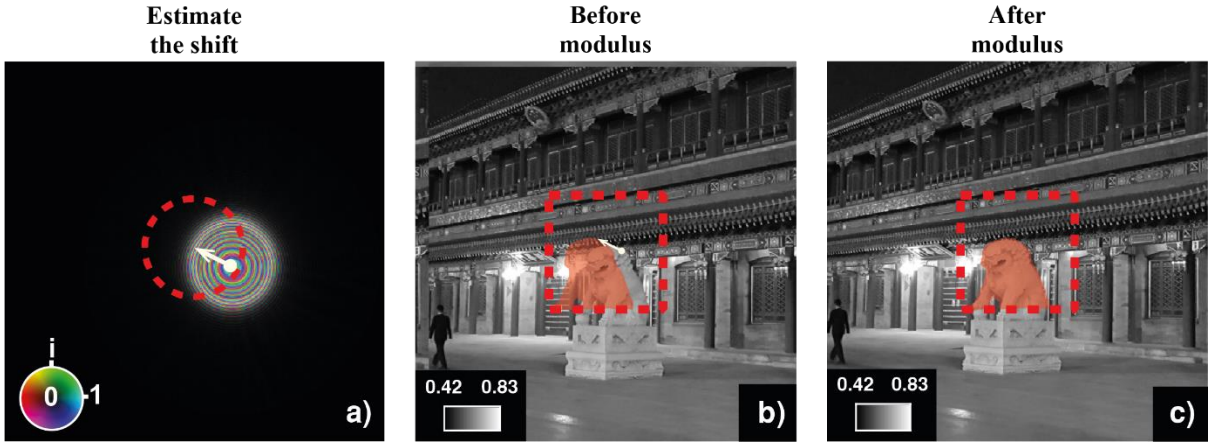


Figure 4. 2. A demonstration of removing global shifting ambiguity. The shifting ambiguity is estimated as the displacement from the centre of guessed probe to the centre of true probe. This displacement is highlighted by a white arrow in (a). The same counter shift should be applied to the guessed object at the same time, as shown in (b). After removing the global shifting, the feature inside the comparison area of guessed object should match to the true one. As shown in (c), now the feature is aligned.

4.1.2. Phase ramp ($e^{j\boldsymbol{b}\cdot\vec{r}}$)

Phase ramp ambiguity causes stripes on the phase images as shown in *Figure 4. 1(e)*. In ptychography, a phase ramp and its counter ramp may exist in the object and probe. This linear phase variation pair cancels out while producing exit waves, and hence has no influence on the modelled wavefronts. Once the influence of global shifting has been measured and removed from a reconstruction, any phase ramp error can be obtained by multiplying the reconstructed object with the conjugate of the true image. The phase of the resulting matrix is then the difference between the phase of the true object and the reconstruction as shown by *eq 4. 2*. The gradient of this linear expression is the phase ramp ambiguity needed to be removed.

$$\begin{aligned} phase\ difference &= angle\left(\left(a^{-1}e^{-jc}e^{-j\boldsymbol{b}\cdot\vec{r}}\widehat{\mathbf{O}}_{\vec{r}+\vec{d},k}\right)\cdot\left(\widehat{\mathbf{O}}_{\vec{r}+\vec{d},k}\right)^{*}\right) \\ &= angle\left(\left(a^{-1}e^{-jc}e^{-j\boldsymbol{b}\cdot\vec{r}}\right)\cdot\left|\widehat{\mathbf{O}}_{\vec{r}+\vec{d},k}\right|^{2}\right) \\ &= angle\left(\left(a^{-1}\cdot\left|\widehat{\mathbf{O}}_{\vec{r}+\vec{d},k}\right|^{2}\right)\cdot e^{-j(\boldsymbol{b}\cdot\vec{r}+\boldsymbol{c})}\right) \\ &= -\boldsymbol{b}\cdot\vec{r}-c \end{aligned} \qquad \textit{eq 4. 2}$$

Since the edges of object usually contain considerable amount of noise that causes unnecessary complexity, a centre area of the object is used for this step. To measure any

phase ramp in the phase error, a best fit is carried out. To do this fit, the most challenging part is first unwrapping the matrix of phase difference. There is no perfect way to unwrap a 2D data so far, and 2D unwrapping can be very time consuming[78,79,80]**.** One way to solve this problem, which is the method that utilised in this thesis, is applying 1D unwrap along the horizontal direction. Then take the gradient of the central row as the estimated phase gradient along the horizontal direction. Then repeat the same process on the central column to obtain the phase gradient along the vertical direction. The order of calculations is interchangeable. Once the gradient of the 2D phase ramp is known, a counter phase ramp will be generated and applied to both the object and probe. Last but not the least, a non-averaged zero phase ramp can introduce a phase offset ambiguity to the image. However, it does not affect the details in the phase image (as shown in Figure 4. 3 (b) and (d)), and it can be removed as apart of complex ambiguity with method explained in the next section.

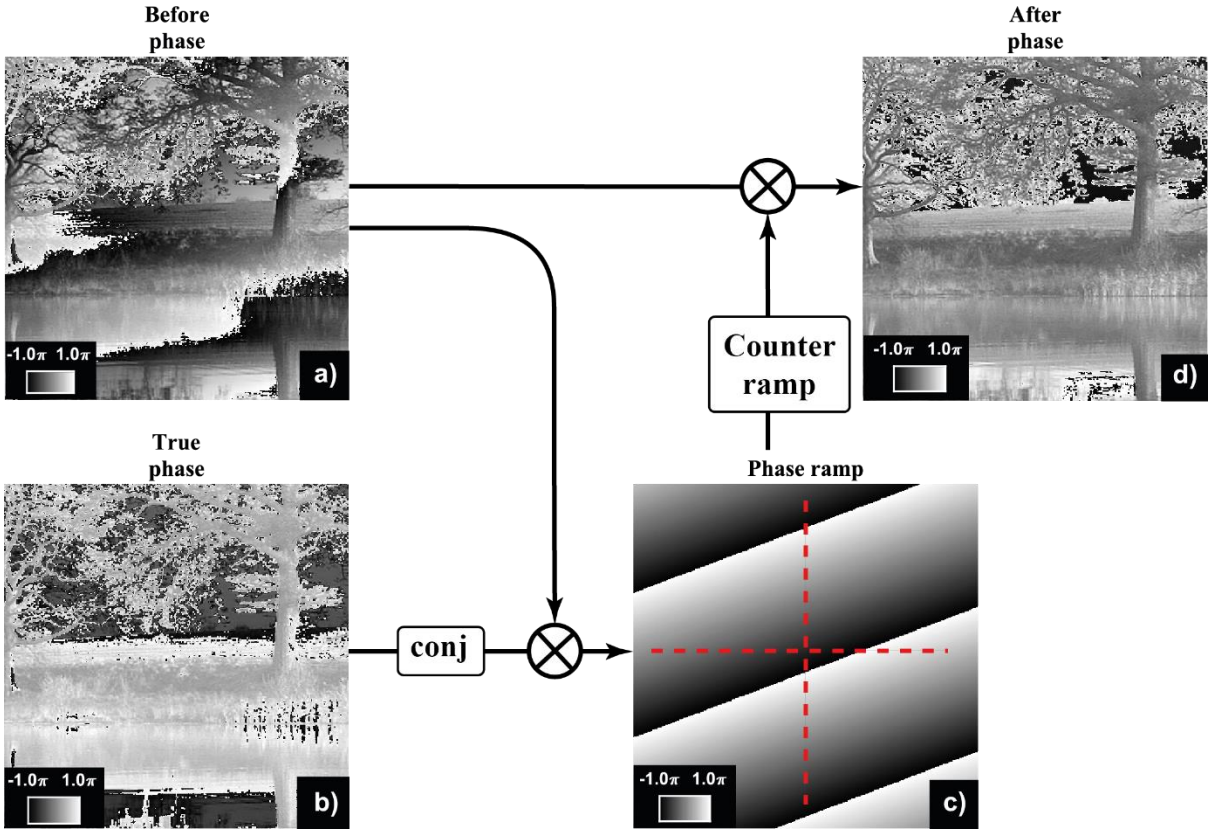


Figure 4. 3. The process of estimating and removing phase ramp ambiguity. A phase difference (c) can be found by multiplying the centre of ambiguous object (a) with the true one (b). The linear gradient of phase ramp along the horizontal and vertical direction can be estimated by unwrapping the centre row and column of the phase difference, as indicated by the 2 red lines in (c). With the estimated phase gradient, a counter phase ramp can be produced and applied to balance out the phase ramp in ambiguity in ambiguous object. As shown in (d), the 'strip appearance' in (a) is significantly improves in (d).

4.1.3. Complex scaling (ae^{jc})

Unlike the previous two, the complex scaling ambiguity has no significant influence on the appearance of reconstructed image. Its modulus part introduces a constant scaling factor to the modulus image. Although it changes the dynamic range of modulus image, its influence is negligible during display as shown in *Figure 4. 1 (a) and (d)*. Meanwhile, its phase part has no considerable impact on the contrast of phase image, as it only changes the phase offset and does not affect the relative phase. However, it can significantly affect the computing of s-domain error, hence must be removed for a more accurate error metric.

The complex scaling ambiguity is estimated as the averaged complex difference between the ambiguous object with the true one (*eq 4. 3*). Again, the poorly reconstructed edges should be excluded for better accuracy. The object centre obtained from previous step is a good option for this step. The influence of removing complex scaling ambiguity is shown in *Figure 4. 4*.

$$\begin{aligned} complex\ constant &= \frac{a^{-1}e^{-jc}\widehat{\mathbf{O}}_{\vec{r}+\vec{d},k} \cdot \left(\widehat{\mathbf{O}}_{\vec{r}+\vec{d},k}\right)^*}{\left|\widehat{\mathbf{O}}_{\vec{r}+\vec{d},k}\right|^2} \\ &= \frac{a^{-1}e^{-jc}\left|\widehat{\mathbf{O}}_{\vec{r}+\vec{d},k}\right|^2}{\left|\widehat{\mathbf{O}}_{\vec{r}+\vec{d},k}\right|^2} \\ &= a^{-1}e^{-jc} \end{aligned} \qquad \textit{eq 4. 3}$$

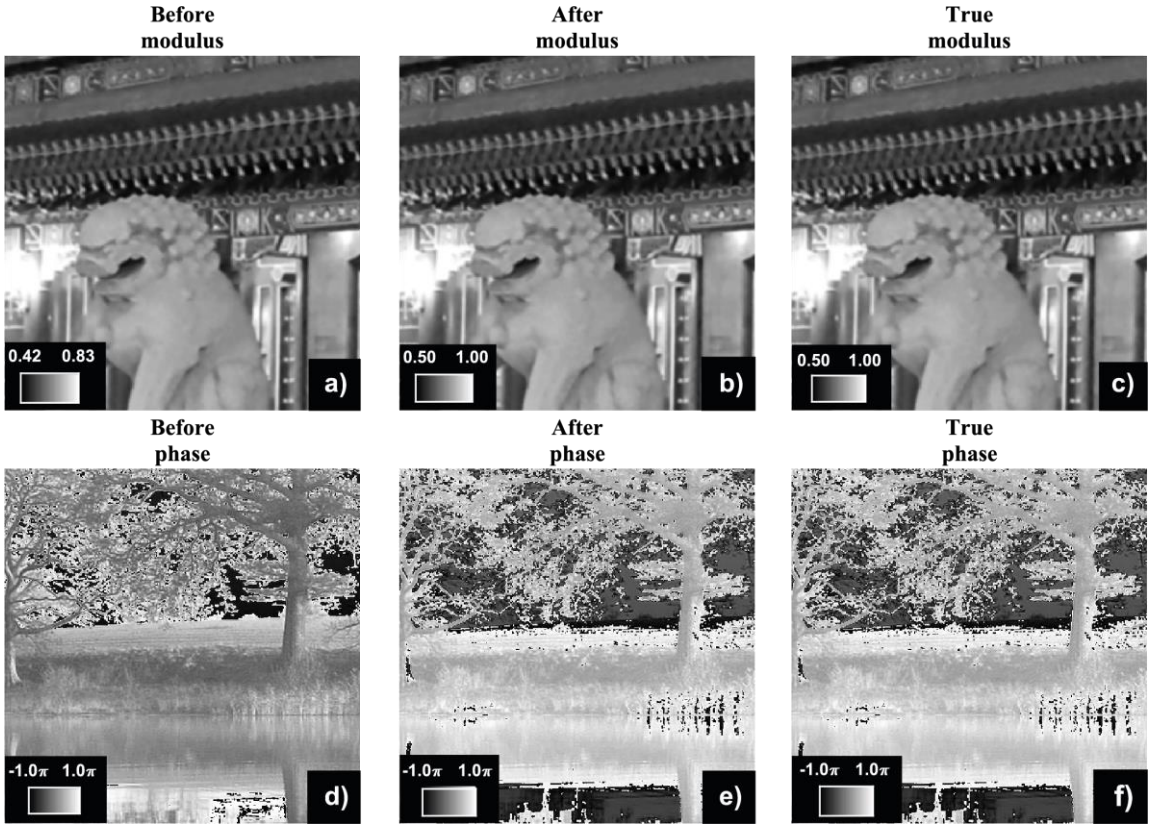


Figure 4. 4. The influence of removing complex scaling ambiguity. From left to right, each column represents the modulus (top, (a)-(c)) and phase (bottom, (d)-(f)) images of before, after and the true centre of object. After removing the complex ambiguity, both the modulus dynamic range (the colour bar in (b)) and the phase offset (e) fit the true images.

4.2. Error metric

Error metrics are designed to quantitatively evaluate the quality of reconstructed images by comparing them with the available true grounds. Since the true grounds are related to the two constraints, i.e. the ***f-constraint*** and ***s-constraint***, two types of error metrics were developed based on them. One extra error metric, which evaluates the variation before and after one iteration, is also explained below. These error metrics can be applied to evaluate the performance of algorithms or to demonstrate a stagnation has reached.

4.2.1. f-domain error (Err_f)

The ***f-domain*** error (i.e. ***f-error***) is established on the difference between the measured diffraction patterns ($\mathbf{I}_{\vec{u},k}$) and the guessed exit waves ($\boldsymbol{\psi}_{\vec{r},k}$) as shown in *eq 4. 4*. This error

metric is always an option for evaluating the reconstruction, as every data set of ptychography must have measured diffraction patterns. Meanwhile, this error metric can be computed while applying ***f-projection***, hence no extra variable needs to be computed during reconstruction. However, it also has two drawbacks. First, since the measured diffraction patterns do not have phase information, ambiguity can exist in the phase of Fourier transformed exit waves. This is evitable when evaluating a result formed by complex number while only its modulus is known. Second, the accuracy of this error metric is determined by the quality of measured intensities. If collected diffraction patterns are contaminated by noise, *eq 4. 4* will produce inaccurate error values. Last but not the least, a low Err_f indicates a good quality of reconstruction, though the content of images can still be distorted by the ambiguities that explained in section *4.1*.

$$Err_f = \frac{\sum_k \sum_{\vec{u}} \left(\left| \boldsymbol{\mathcal{F}}\left(\boldsymbol{\psi}_{\vec{r},k}\right) \right| - \sqrt{\mathbf{I}_{\vec{u},k}} \right)^2}{\sum_k \sum_{\vec{u}} \mathbf{I}_{\vec{u},k}} \qquad \textit{eq 4. 4}$$

4.2.2. s-domain error (Err_s)

Another approach for evaluating the reconstruction quality is comparing the reconstructed images with the true one. After removing *s-domain* ambiguities, a *s-domain* error (i.e. ***s-error***) can be computed by *eq 4. 5*[20,56]. The advantage of *s-domain* error metric is that its value directly relates to the appearance of reconstructed image. Both the modulus and phase difference between two complex values are directly under comparison, which gives no uncertainty for the error. On the other hand, this error metric is only appliable when the true image is available, which implies a simulation situation. Meanwhile, the ambiguity removing procedure demands extra computation, which slightly increase the memory footprint and computing time.

$$Err_s = \frac{\sum_{\vec{r}} \left| \mathbf{O}_{\vec{r}} - \widehat{\mathbf{O}}_{\vec{r}} \right|^2}{\sum_{\vec{r}} \left| \widehat{\mathbf{O}}_{\vec{r}} \right|^2} \qquad \textit{eq 4. 5}$$

4.2.3. self-variation (Err_{self})

As explained by its name, this self-variation error metric (Err_{self}) evaluates the difference of a variable before and after one complete reconstruction. Two examples are given in *eq 4. 6* and *eq 4. 7* respectively by using the exit wave and object as the comparing variables. This error metric adapts to both simulated and practical data. Since it is a comparison between two complex numbers, it is sensitive to variation. However, its value should be considered as an indicator for converging rather than the quality of reconstruction. A small Err_{self} indicates the reconstruction is stagnated, which can be either converging to a solution or getting stagnated.

$$Err_{self\ \boldsymbol{\Psi}} = \frac{\sum_k \sum_{\vec{r}} \left| \boldsymbol{\Psi}_{\vec{r},k,n} - \boldsymbol{\Psi}_{\vec{r},k,n+1} \right|^2}{\sum_k \sum_{\vec{r}} \left| \boldsymbol{\Psi}_{\vec{r},k,n+1} \right|^2} \qquad \textit{eq 4. 6}$$

$$Err_{self\ \mathbf{O}} = \frac{\sum_{\vec{r}} \left| \mathbf{O}_{\vec{r},n} - \mathbf{O}_{\vec{r},n+1} \right|^2}{\sum_{\vec{r}} \left| \widehat{\mathbf{O}}_{\vec{r},n+1} \right|^2} \qquad \textit{eq 4. 7}$$

4.3. Comparison of ptychographic algorithms

We have implemented and tested the well-known phase retrieval algorithms: the 'PIE' family of algorithms[20], the difference map (DM)[4,29] , relaxed averaged alternating reflections (RAAR)[81] and hybrid projection and reflection (HPR)[66]. The PIE-type algorithms are based on the stochastic gradient descent concept[56], whilst the rests are based on the set projection and reflection concept[82], and hence named the 'PR' algorithms in this section. The tests are begun by tuning algorithm parameters using multiple sets of simulated calibration data. Then, these tuned algorithms were tested on simulated data generated from a range of scenarios using either a randomised illumination function or convergent beam illumination, combined with either a weakly- or a strongly- scattering sample. We then used the ambiguity-invariant error measure detailed in Section 4.1 and 4.2 to evaluate the differences between the resulting images[56].

4.3.1. Description of the simulation

The ptychographic algorithms detailed in Chapter 3 all reconstruct the transmission characteristics of the object under examination whilst simultaneously recovering the complex-valued illumination wavefront incident on the object (commonly referred to as the 'probe'). To test this blind-deconvolution ability, a wide range of probe and object combinations were simulated based on four specific examples from the literature, each under a different wavelength regime: optical[56], electron[19], soft X-ray[83] and hard X-ray[4]. Experimental conditions and the probes in our tests were simulated to match as closely as possible those used in the respective references, whilst two simulated 'objects' were tested under each scenario. One had both wide intensity and phase dynamic range, what we will refer to as the "strong object" in this thesis. The other was fully transparent and only provided weak phase variation: the "weak object". Details of the four probes and two objects are given in *Figure 4. 5* and *Figure 4. 6* respectively. Further details of the experiment setups are listed in *Table 4. 1*. For each scenario, a reasonable approximation of the probe was produced beforehand and utilised as the initial guess in each algorithm. These initial probes are also shown in *Figure 4. 5*. The initial guess of the object in every case was an all-one matrix, which can be interpreted as a completely transparent specimen. Meanwhile, considering the PIE algorithms employ a random reconstruction sequence for each iteration, we pre-prepared this sequence in advance and shared it for all PIE reconstructions.

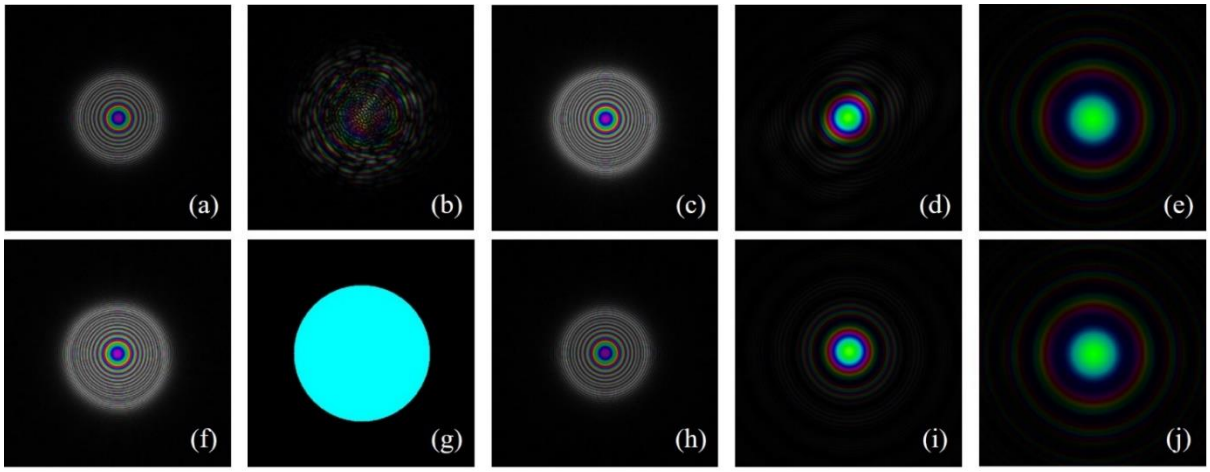


Figure 4. 5. The first row (from (a) to (e)) are the probes used for simulating diffraction patterns, while the second row (from (f) to (j)) are the corresponding initial probe for reconstruction. The columns (e.g. (a) and (f)), from left to right, represent the probe for respectively the parameter optimisation, then the optical, electron, soft x-ray and hard x-ray test cases. All probes are displayed on the colour wheel shown as an inset in (a). Further details of these probes are listed in Table 1.

Figure 4. 6. The modulus and phase image of tested objects are shown in this figure. Each column is a modulus and phase pair of a tested object. The first row (i.e. (a) to (c)) demonstrates the modulus images of these objects, whereas the second row (i.e. (d) to (f)) demonstrates their phase images. To be more specific: the modulus (a) and phase (d) images of object for parameter optimisation, the modulus (b) and phase (e) images of strong object and the modulus (c) and phase (f) images of weak object. Each image comes with a colour bar to indicate their dynamic range. All images have the same size (3800×3800 pixels), which is large enough for all testing scenarios.

Table 4. 1.Details of four test scenarios

Wavelength regime	**Optical**[56]	**Electron**[19]	**Soft x-ray**[83]	**Hard x-ray**[4]
Real space pixel width	1.01 μm	0.34nm	42.6nm	18.3nm
Probe size (pixels)	512×512	1024×1024	960×960	128×128
Probe spot size	150 μm	40nm	2 μm	300nm
Scan position (lateral × vertical)	15×15	8×8	9×9	40×40
Step size (w.r.t. the probe spot size)	20%	25%	40%	33%
Randomness (w.r.t the spot size)	33%	75%	0%	30%

4.3.2. Parameter optimisation

Although in its initial presentation, ePIE and DM included tuning parameters, they are almost always set to unity. This choice has firm theoretical foundations in both cases. The ePIE can be considered a stochastic gradient descent scheme whose step sizes, when the tuning parameters are unity, are the Lipschitz constants of the gradients of the associated cost functions[56], whilst the DM approach corresponds to the Douglas-Rachford method when its tuning parameter is unity[53 81]. The other tested ptychographic algorithms require users to determine some tuning parameters, which typically trade off convergence speed against stability. Thus, these algorithms were calibrated before the test. To avoid bias, a calibration object, with a dynamic range between the strong and weak object, was utilised for the tuning. Its modulus and phase images are shown in *Figure 4. 6 (a) and (d)* respectively. Similarly, the probe used for parameter optimisation is shown in *Figure 4. 5 (a) and (f)*. The test range for each parameter is given in *Table 4. 2*, where the variables relate to the algorithm descriptions provided in the references listed in the first column of the table. Equally spaced values within the range were tested and each algorithm was cycled through 90 iterations before assessing their performance. The average values of parameter combination with the smallest error (see *Table 4. 2*) was considered the most suitable and applied in the rest of tests.

Table 4. 2. Optimised parameters for each algorithm tested. Parameter descriptions can be found in the references listed.

Algorithm	Parameter test range	Chosen tuning parameters
rPIE/mPIE[56]	$\alpha \in [0.01, 1.1]$	$\alpha = 0.1$
	$\beta \in [0.01, 1.1]$	$\beta = 0.8$
HPR[53]	$\beta \in [0.01, 1.2]$	$\beta = 0.5$
RAAR[81]	$\beta \in [0.01, 1.2]$	$\beta = 0.8$

The mPIE algorithm applies Nesterov-type acceleration to rPIE[56]. In its original exposition, mPIE had a restrictively large parameter set (the α and β parameters from rPIE plus five additional parameters). However, we have found that most of these additional parameters can be avoided simply by applying the momentum only at the end of each iteration of rPIE. Referring to the original paper, this equates to setting T – the batch size – equal to the number of diffraction patterns in the data set. Under this condition, the rPIE update equations can be

employed directly in mPIE (without reduced step sizes, so with reference to Chapter 3, $\alpha = \beta = 1$), and the momentum learning rate can be fixed at $\gamma = 0.9$).

4.3.3. Test results

Having selected parameters for all the algorithms, we ran eight tests to assess their performance: using the strong and weak objects shown in *Figure 4. 6*, under each of the four experiment/wavelength scenarios (optical, electron, soft- and hard-x-ray). Each scenario has two error figures below, which are the outcomes from tests with the strong and weak objects respectively. The central part of the reconstructed object from each algorithm is also plotted next to the error graph with a coloured frame: these cut out areas show the modulus for the strong object and the phase for the weak object tests. The ground truth of these regions is shown in *Figure 4. 7* for comparison.

Figure 4. 7. The centre of modulus image of strong object (a) and the phase image of weak object (b) for later comparison.

4.3.3.1. Optical ptychography

The optics microscopy is a difficult scenario, as its probe is highly structured whilst the initial guessed probe is only an aperture without any phase information. This requires the algorithms to recover the probe effectively besides reconstructing the object. As shown in the error figures, only rPIE and mPIE reconstructed the centre of the strong object, while only mPIE can reconstruct the weak one, even after 1000 iterations. In unsuccessful reconstructions of all the algorithms, the probe either drifted to a corner or collapsed into a

single point. Based on the test results, only mPIE can provide promising results under this most demanding of tests. Note that the fluctuations in the plots at very low error values arise from our removal of the ambiguities before calculating Err_s, which requires image registration to small fractions of a pixel.

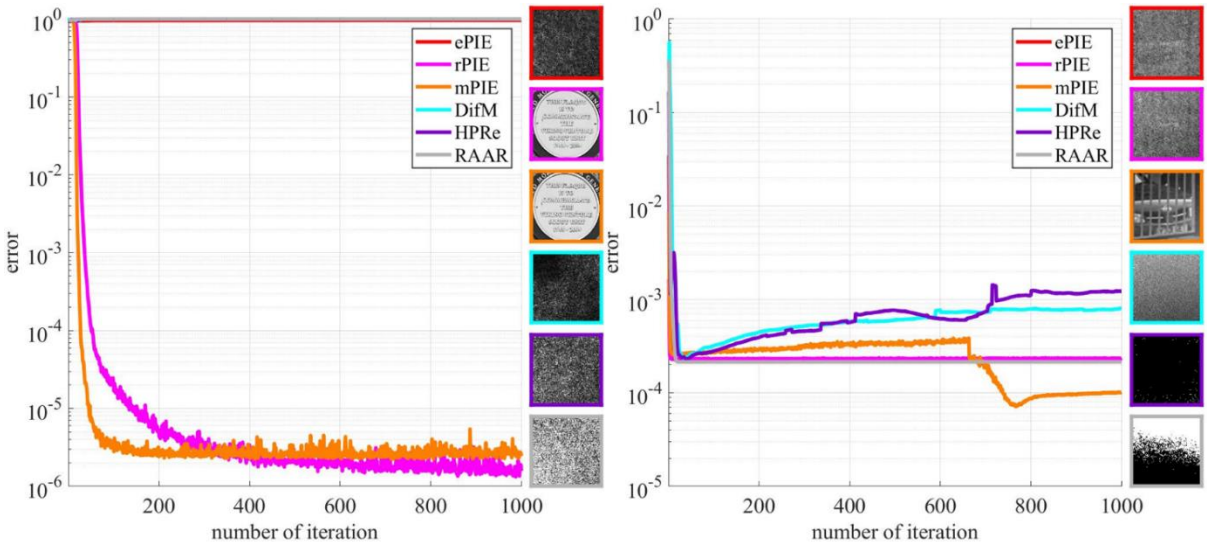


Figure 4. 8. The ambiguity-free real space error (Err_n) for simulations based on optical ptychography with a strong object (left) and a weak object (right). Each algorithm is marked by a unique colour, their legends are provided in the figure. Central parts reconstructed by different algorithms are shown to the right, contained in a frame with the corresponding colour. These images show the modulus of the strong object and the phase of weak object. Subsequent figures use the same structure.

4.3.3.2. Electron ptychography

In this scenario, all algorithms provided acceptable reconstructions of the strong object. The PR algorithms have generally the same gradient as ePIE, while rPIE and mPIE converge more quickly. The behaviour of all these algorithms are similar in the weak object test. Here, only mPIE is capable to further decrease its error as the iterations progress.

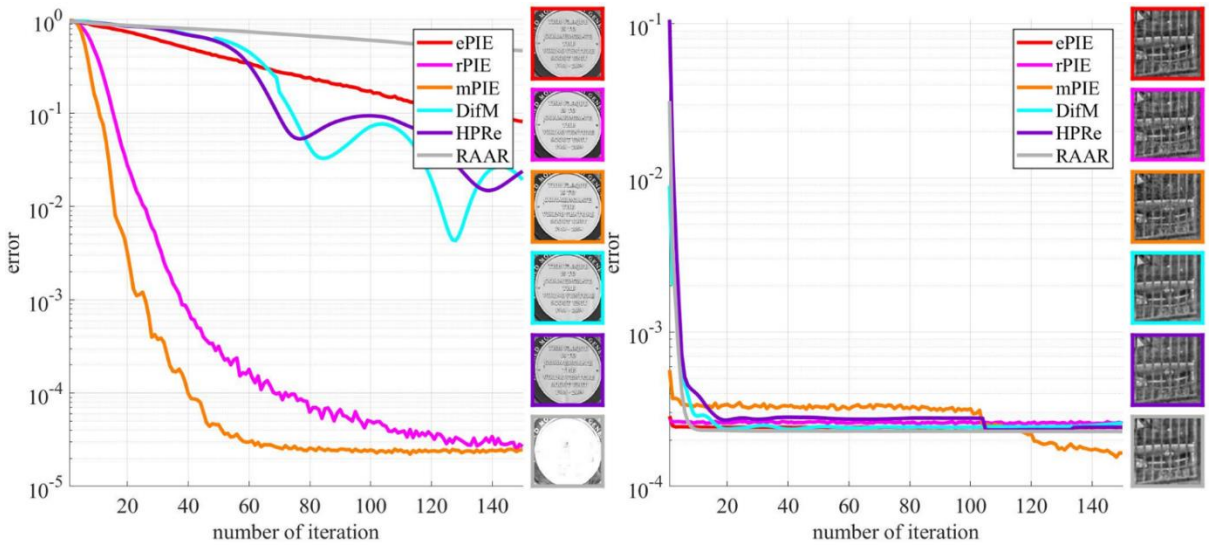


Figure 4. 9. The ambiguity-free real space error (Err_n) for electron microscopy with strong object (left) and weak object (right). The error plots and central part of the respective reconstructed objects share the same colour coding.

4.3.3.3. Soft x-ray ptychography

Since this test case used a regular scanning grid (as in the corresponding paper), its results are affected by the raster grid ambiguity, which appears as a periodic pattern on the reconstructed images. In the strong object test, all algorithms approach the exact object to differing extents, while the mPIE gives the least error among all these algorithms. DM and HPR performs well in this test, although some grid artefacts are apparent in the reconstructed image.

The PR algorithms are consistently outperformed by the PIE algorithms in the weak object test. All projection and reflection algorithms struggle to reconstruct the weak object under the influence of raster grid ambiguity. Significant grid pattern can be observed on their reconstructed images. Once again, mPIE gives the most successful reconstruction.

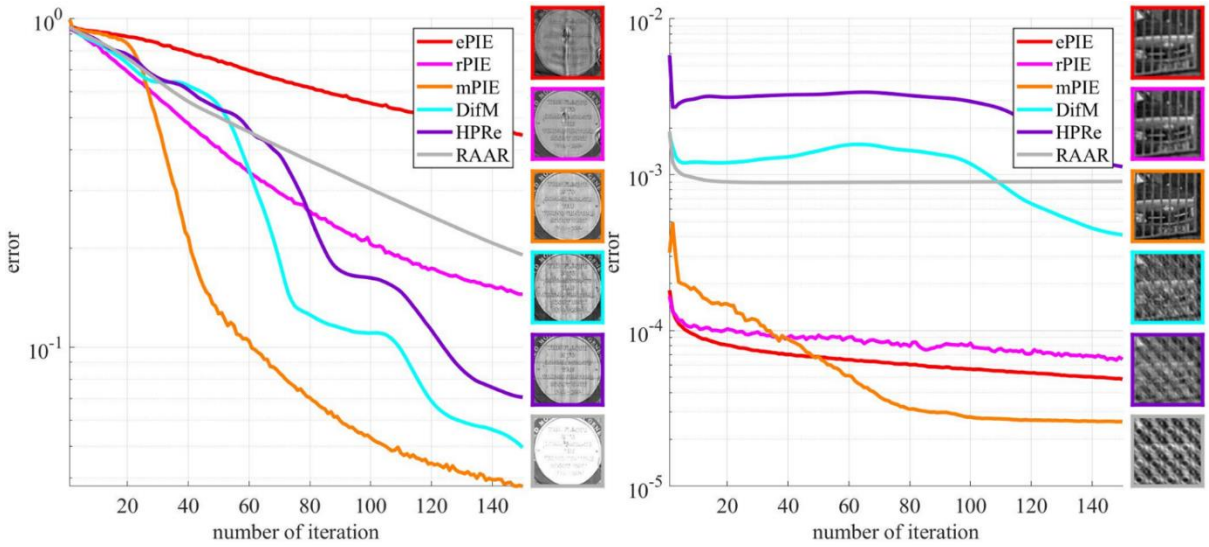


Figure 4. 10. The ambiguity-free real space error (Err_n) for soft x-ray microscopy with strong object (left) and weak object (right). The error plots and central part of the respective reconstructed objects share the same colour coding.

4.3.3.4. Hard x-ray ptychography

In the strong object test, all algorithms reconstruct the centre of the object successfully, although mPIE and rPIE give a smaller error than the others by at least two orders of magnitude. ePIE, DM and HPR stay about the same error level. The weak object test in this case shows a clear margin between the PIE algorithms and the PR algorithms, while the PIE algorithms give much smaller final errors.

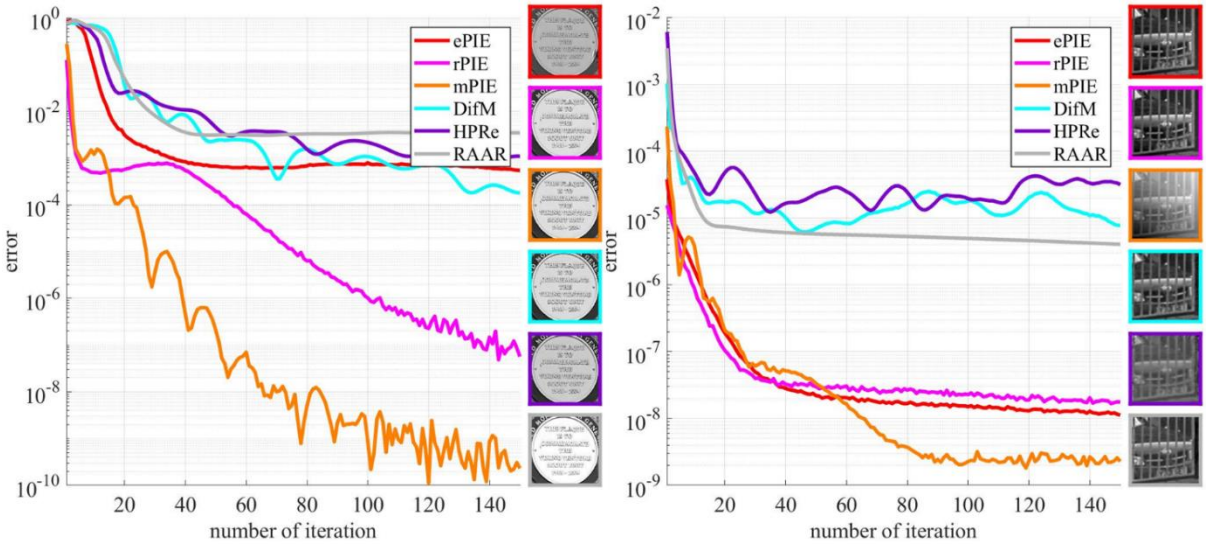


Figure 4. 11. The ambiguity-free real space error (Err_n) for hard x-ray microscopy with strong object (left) and weak object (right). The error plots and central part of the respective reconstructed objects share the same colour coding.

4.3.4. Noise resistance

Previous tests have demonstrated that all introduced algorithms can reconstruct noiseless data. However, noise is inevitable in practical scenarios and can bring difficulty to the reconstruction. As explained in *section 2.4.3*, noise could exist either in the measured intensities or the scanning positions. Hence some noisy data are simulated to test the robustness of these algorithms under the influence of various noise. The influence of the detector noise and scanning position noise are tested separately.

4.3.4.1. Noise in measured intensities

The simulation of detector noise is explained in section 2.4.3. Different noise level is simulated by modifying the incident photons. By decreasing the photon counts from 10^{10} to 10^{6}, the detector noise becomes increasingly significant on the diffraction pattens. Then these noisy diffraction patterns are applied with correct scanning positions for reconstruction. Due to the influence of noise, all algorithms reach to stagnations within less iterations. Test results indicate that 500 iterations are enough to reach the stagnation in this test. Since the measured intensities are noisy, they are not suitable for evaluating the f-domain error. To

demonstrate the influence of detector noise on different algorithms, the reconstructed modulus images with 10^9 and 10^6 photon dose are demonstrated in *Figure 4. 12*. As shown in the figure, mPIE and DM are more sensitive to the detector noise. The reconstruction quality of mPIE significantly drops at low photon dose, while the performance of DM is poor even at a relatively high photon dose (10^9).

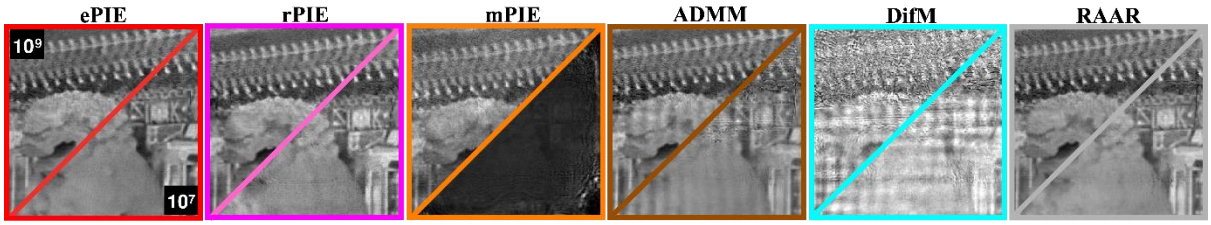


Figure 4. 12. A comparision of different algorithms under the influence of detector noise, which is determined by the incident photon dose. The result from 10^9 photon counts is plotted on the top left corner, while the results from 10^7 photon counts is plotted on the bottom right corner (as labeled on the figure). All reconstructed image are normailised to the same dynamic range to reveal their details.

4.3.4.2. Noise in scanning positions

As explained in section 2.4.3, the noise in scanning positions contains three parts: scaling, rotating and randomness. They are tested separately here. First, a group of noiseless measured intensities are simulated with an irregular scanning grid. After that, these three different types of position noise are added separately. Finally, the noiseless diffraction patterns and noisy scanning positions are provided together for phase retrieving. The noisy scanning positions cause the reconstructed images having distortion comparing with the ground truth, hence the s-domain error is not suitable for evaluating the quality of reconstruction. Error level, which gives the most contrast between these algorithms, is chosen, their reconstruction results are shown in *Figure 4.13*. Test results indicate that the scaling error has the most significant impact on the reconstruction. 105% scaling on the scanning grid considerably decreases the quality of reconstruction for all algorithms. ePIE, rPIE and RAAR demonstrate better tolerance on the rotating and randomness error.

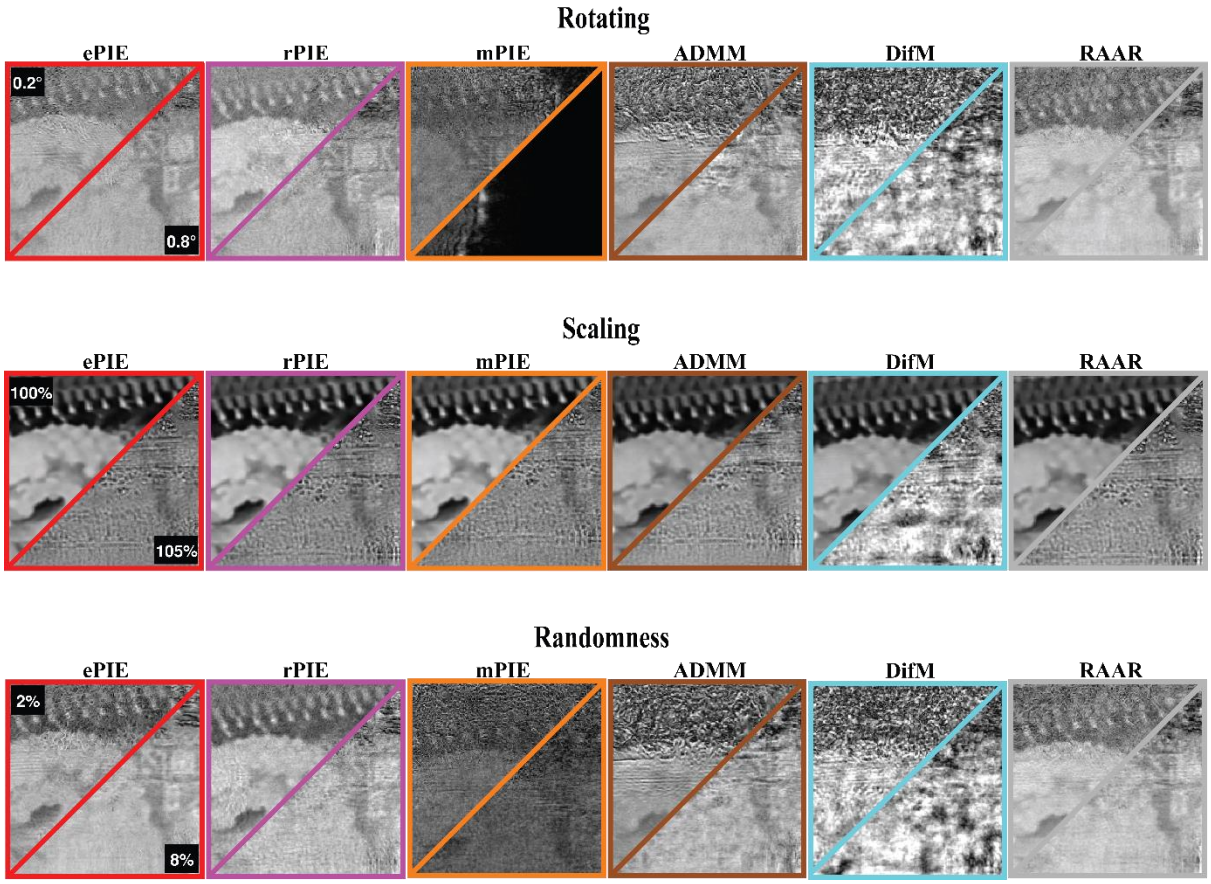


Figure 4. 13. A comparision of different algorithms under the influence of differetn types of scanning position noise. The first row compares the results with 0.2 and 0.8 degree rotating error. The second row compares the results from correct (100% scaling) and 105% scaling scanning grid. The third row compares the results from scanning grid with 2% and 8% of step size randomness error. Among all the scanning position error, the scaling error has a more significant influence on the reconstruction too all algorithms. mPIE and DM are sensitive to the noise, while ePIE, rPIE and RAAR show a better tolerance on the noise.

As shown in the above tests, ePIE, rPIE and RAAR have the best noise resistance among these tested algorithms. Although the momentum concept accelerates the converging speed in noiseless situation, it also makes mPIE become instable in noisy scenario. Updating with all exit waves or one by one give no significant difference on the noise tolerance.

4.3.5. Summary

As shown above, both PIE and PR algorithms can provide reasonable reconstruction when the specimen has large dynamic range, the guessed probe is not wildly different from the exact one, and the scanning positions do not lie on a regular grid. In this kind of scenario, the PIE

algorithms converge to the final solution faster than the projection and reflection algorithms. Meanwhile, PIE algorithms are more tolerant to the ambiguity caused by a regular scanning grid and the seemingly more difficult task of reconstructing a weak specimen, which appears as a significant challenge to the PR algorithms. When the probe is both highly structured and difficult to model accurately prior to the reconstruction, image reconstruction is a challenge for all the algorithms. Only mPIE can successfully reconstruct the specimen regardless to its dynamic range, although it takes significantly more iterations.

To evaluate phase retrieval algorithms for ptychography, we began by tuning their parameters with several groups of noise-free diffraction patterns, which simulated an ideal experimental data set. Different parameter values were evaluated by reconstructing these data, removing ambiguities in the reconstructed images, and calculating a spatial domain error value. The average values of the best-performing parameters were utilised in subsequent tests, whose data sets were generated from simulating realistic scenarios, including various combinations of strongly- and weakly-diffracting samples with focused and defocused/diffused illumination probes. These tests gave an insight into the differences between the algorithms and highlighted their robustness to a wide range of experimental geometries. Our results indicate that best performance (in terms of convergence rate and final error value) is realised by the mPIE algorithm with correctly tuned parameters.

Although there are already some publications in comparing different algorithms[84,85], this section tries to reveal their difference by increasing the diversity of the test scenarios. Some relatively new algorithms are also involved in the test (e.g. rPIE and mPIE). The improved s-domain error metric provides a new approach of observing the behaviour of algorithms.

5. Adaptive regularised PIE

In this chapter, the logic of the regularisation term in the existing PIE algorithms is fully explained (*section 5.1*). Then a new PIE-based algorithm is introduced together with pseudocode for implementation (*section 5.2*) and its performance is assessed using simulated data (*section 5.3*). Our work suggests this adaptive ptychographical iterative engine, or adaPIE, moulds together benefit from the different alternatives to give plug and play operation. Like ePIE, it is a stochastic gradient algorithm with a very small memory footprint and a rapid initial rate of convergence. Like RAAR and ADMM, it often converges to a global minimum when given perfect data (although in common with these approaches there are no convergence guarantees). And crucially, like DM it is essentially parameter-free and there is no need to tune the algorithm for different experiment scenarios. The algorithm and results in this chapter are adapted from a draft paper under preparation and due for submission to Optics Express.

5.1. The limitation of existed PIEs

As explained in the 'regularised PIE' section in *Chapter 3.3.3*, the updating functions utilised by all PIE-based algorithms are developed to minimise two cost functions. For a clear description, these two cost functions are re-written as *eq 5. 1* and *eq 5. 2*.

$$\mathcal{L}_{\mathbf{O},k} = \sum_{\vec{u}} \left| \boldsymbol{\mathcal{F}}\left(\mathbf{O}'_{\vec{r},k}\mathbf{P}_{\vec{r}}\right) - \sqrt{\mathbf{I}_{\vec{u},k}} \right|^2 + \sum_{\vec{r}} \boldsymbol{\omega}_{\mathbf{O},\vec{r}} \left| \mathbf{O}'_{\vec{r},k} - \mathbf{O}_{\vec{r},k} \right|^2 \qquad \textit{eq 5. 1}$$

$$\mathcal{L}_{\mathbf{P}} = \sum_{\vec{u}} \left| \boldsymbol{\mathcal{F}}\left(\mathbf{O}_{\vec{r},k}\mathbf{P}'_{\vec{r}}\right) - \sqrt{\mathbf{I}_{\vec{u},k}} \right|^2 + \sum_{\vec{r}} \boldsymbol{\omega}_{\mathbf{P},\vec{r}} \left| \mathbf{P}'_{\vec{r}} - \mathbf{P}_{\vec{r}} \right|^2 \qquad \textit{eq 5. 2}$$

The first one ($\mathcal{L}_{\mathbf{O},k}$) takes the updated object part at the k^{th} scan position ($\mathbf{O}'_{\vec{r},k}$) as the independent variable, while the second one ($\mathcal{L}_{\mathbf{P}}$) takes the updated probe ($\mathbf{P}'_{\vec{r}}$) as the independent variable. Both two functions are made up by two terms. The first term evaluates the error between the guessed exit wave produced by the updated object (or probe) and the corresponding intensity measurement ($\mathbf{I}_{\vec{u},k}$). The second term evaluates a weighted variation of this independent variable during the updating. The core of PIEs is searching for the updated

object and probe that can minimise these two cost functions. Such an optimisation process is done by estimating the current gradient and moving towards an opposite direction, which is also known as 'gradient descent' method[86].

Gradient descent method is well known for the converging speed, as its error can never increase during the optimisation[86]. However, such a characteristic is not desired for solving phase problem. The modulus constraint ($\mathbb{M}$) formed by the intensity measurements is a non-convex set[47], which cause local minima in the searching space. Since the gradient descent method does not allow the error increasing, it cannot escape from a local minimum, hence get stagnated.

Instead of gradient descent, PIEs use stochastic gradient descent method to prevent stagnation. One should notice that *eq 5. 1* and *eq 5. 2* only take one scan position (e.g. k^{th}) into consideration rather than summing across all scan positions. By doing this, the variable updated at one position is not the most optimised value for other positions unless a solution is found. Such a process is named as stochastic gradient descend method, which is utilised by PIEs for preventing stagnation.

After explaining the common parts of all PIE algorithms, let us talk about their difference. The development of various PIEs comes from the differently designed weighting factors: $\boldsymbol{\omega}_{\mathbf{O},\vec{r}}$ and $\boldsymbol{\omega}_{\mathbf{P},\vec{r}}$. Referring to these cost functions, their first terms evaluate the error with respect to the intensity measurements. When the noise is negligible, such an error directly indicates the incorrectness of a guessed exit wave, so called a '*hard-error*'. On the other hand, their second terms evaluate the variation of the independent variables. Although this second term equals zero when a solution is found, its value does not reflect the correctness of the current guess. As a comparison, this error is named as '*soft-error*'.

The purpose of this '*soft-error*' term is to penalise any significant change on the variable that is under optimisation. The updated variable is not the one giving the minimum *hard-error* after introducing the regularisation term. This is a desired behaviour when multiple constraints exist and the variable under optimisation is not expected to over-fit to any of these constraints[63]. As shown in *Figure 5. 1*, less fluctuation is one of the advantages brought by preventing over-fitting.

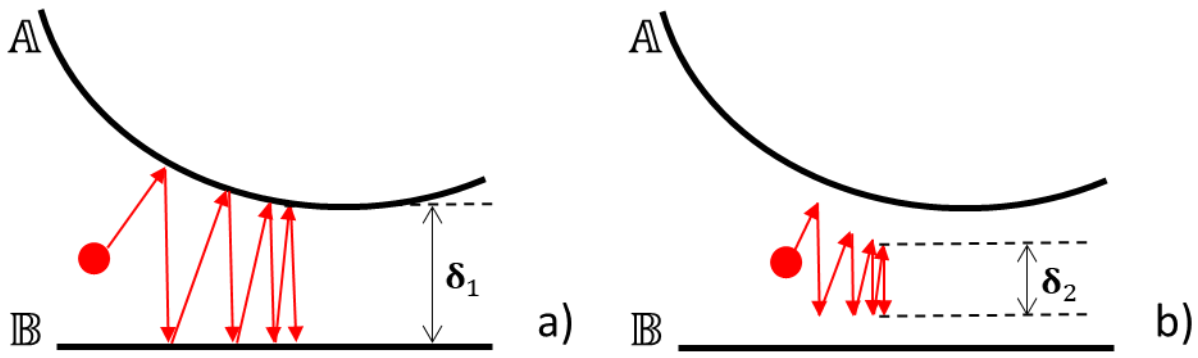


Figure 5. 1. A demonstration of how 'over-fitting' causes fluctuation. In this example, two non-interception sets (i.e. $\mathbb{A}$ and $\mathbb{B}$) are shown as the black curved line and black straight line respectively. The same starting points are marked as red dots in (a) and (b). In 'over-fitting' scenario (i.e. (a)), the red dot approaches the area that two sets are close to each other, then fluctuates by alternatively fitting these two sets. By penalising the variation (i.e. (b), the red dot also moves towards the same area, but moves less distance each time. The fluctuations in these two scenarios are labels as δ_1 and δ_2 respectively. The fluctuation is less by preventing over-fitting.

Rather than weighting the variation with a constant, a more flexible way is defining a matrix with the same size as the variable, hence a spatially varying penalties can be applied pixel-wisely. For example, we may think it sensible to increase the penalty for changing an object part ($\mathbf{O}_{\vec{r},k}$) in areas where the probe is dim, or we may penalize changes to the probe ($\mathbf{P}_{\vec{r}}$) when it passes through opaque regions of the object. As $\boldsymbol{\omega}_{\mathbf{O},\vec{r}}$ and $\boldsymbol{\omega}_{\mathbf{P},\vec{r}}$are in matrix form instead of constants and act on the regularisation term, they are named as 'regularisation maps' in the later context. Differently designed regularisation maps making the updating functions of PIEs diverge from its general form. The direviation from cost function to the general form of updating function has been shown by *eq 3. 54* to *eq 3. 56* in the rPIE section. From this general form, the updating functions utilised by each PIE algorithms can be obtained by substitute $\boldsymbol{\omega}_{\mathbf{O},\vec{r}}$ with suitable expression. The object updating functions utilised by existed PIE algorithms and corresponding definition of $\boldsymbol{\omega}_{\mathbf{O},\vec{r}}$ are listed in ***Table 5. 1***.

Table 5. 1. *The object updating functions of different PIE algorithms*

	The updating function	The regularisation map
General form	$\mathbf{O}'_{\vec{r},k} = \frac{\mathbf{P}^*_{\vec{r}}\boldsymbol{\psi}'_{\vec{r},k} + \boldsymbol{\omega}_{\mathbf{O},\vec{r}}\mathbf{O}_{\vec{r},k}}{\lvert\mathbf{P}_{\vec{r}}\rvert^2 + \boldsymbol{\omega}_{\mathbf{O},\vec{r}}}$	$\boldsymbol{\omega}_{\mathbf{O},\vec{r}}$
PIE	$\mathbf{O}'_{\vec{r},k} = \mathbf{O}_{\vec{r},k} + \frac{\lvert\mathbf{P}_{\vec{r}}\rvert\mathbf{P}^*_{\vec{r}}}{\lvert\mathbf{P}_{\vec{r}}\rvert_{max}(\lvert\mathbf{P}_{\vec{r}}\rvert^2 + \alpha\lvert\mathbf{P}_{\vec{r}}\rvert^2_{max})}(\boldsymbol{\psi}'_{\vec{r},k} - \boldsymbol{\psi}_{\vec{r},k})$	$\boldsymbol{\omega}_{\mathbf{O},\vec{r}} = \lvert\mathbf{P}_{\vec{r}}\rvert_{max}\left(\lvert\mathbf{P}_{\vec{r}}\rvert + \alpha\frac{\lvert\mathbf{P}_{\vec{r}}\rvert^2_{max}}{\lvert\mathbf{P}_{\vec{r}}\rvert}\right) - \lvert\mathbf{P}_{\vec{r}}\rvert^2$
ePIE	$\mathbf{O}'_{\vec{r},k} = \mathbf{O}_{\vec{r},k} + \alpha\frac{\mathbf{P}^*_{\vec{r}}}{\lvert\mathbf{P}_{\vec{r}}\rvert^2_{max}}(\boldsymbol{\psi}'_{\vec{r},k} - \boldsymbol{\psi}_{\vec{r},k})$	$\boldsymbol{\omega}_{\mathbf{O},\vec{r}} = \frac{\lvert\mathbf{P}_{\vec{r}}\rvert^2_{max}}{\alpha} - \lvert\mathbf{P}_{\vec{r}}\rvert^2$
rPIE (mPIE)	$\mathbf{O}'_{\vec{r},k} = \mathbf{O}_{\vec{r},k} + \frac{\mathbf{P}^*_{\vec{r}}}{(1-\alpha)\lvert\mathbf{P}_{\vec{r}}\rvert^2 + \alpha\lvert\mathbf{P}_{\vec{r}}\rvert^2_{max}}(\boldsymbol{\psi}'_{\vec{r},k} - \boldsymbol{\psi}_{\vec{r},k})$	$\boldsymbol{\omega}_{\mathbf{O},\vec{r}} = \alpha(\lvert\mathbf{P}_{\vec{r}}\rvert^2_{max} - \lvert\mathbf{P}_{\vec{r}}\rvert^2)$

The logic of $\boldsymbol{\omega}_{\mathbf{O},\vec{r}}$ is to apply a strong penalty to the object part update where the probe is dim, since these regions are susceptible to noise, and to apply a smaller penalty where the probe is bright, reflecting the higher signal-noise ratio there. Likewise, the weighting factor ($\boldsymbol{\omega}_{\mathbf{P},\vec{r}}$) for probe regularization assumes a low signal to noise (so a high penalty) where the object is relatively opaque, and a high signal to noise (low penalty) where the object is transparent. Such a trend can be seen in all of the definitions of $\boldsymbol{\omega}_{\mathbf{O},\vec{r}}$ in ***Table 5. 1***.

There are a couple of gaps in this logic. First, it ignores the interplay between the updates at different positions. For example, the object regularization map is exactly the same whether the object box is taken from the centre of the object reconstruction, which will have been illuminated many times, or if it is taken from the edge, where some areas of the object will have been illuminated only once. Second, it fails to account for the influence of errant bright pixels during the reconstruction, for example a single bright pixel in the object results in a large regularization penalty applied across every pixel in the probe.

5.2. Adaptive regularisation

A new regularisation approach is suggested, which takes the influence of overall-illumination condition into consideration when building a regularisation map for a single scan position. The new algorithm applying this concept is named as ‘adaPIE’, who solves the issues of previous PIEs by using as regularisation maps the average probe intensity that illuminates

each object pixel and the average object transparency through which each probe pixel passes. The update steps for the j^{th} object part and the probe are shown in *eq 5. 3* and *eq 5. 4* respectively.

$$\mathbf{O}'_{\vec{r},k} = \mathbf{O}_{\vec{r},k} + \frac{\mathbf{P}^*_{\vec{r}} \cdot \left(\boldsymbol{\psi}'_{\vec{r},k} - \boldsymbol{\psi}_{\vec{r},k}\right)}{|\mathbf{P}_{\vec{r}}|^2 + \alpha\langle|\mathbf{P}|^2\rangle_{\vec{r},k}} \qquad \textit{eq 5. 3}$$

$$\mathbf{P}'_{\vec{r}} = \mathbf{P}_{\vec{r}} + \frac{\mathbf{O}^*_{\vec{r},k} \cdot \left(\boldsymbol{\psi}'_{\vec{r},k} - \boldsymbol{\psi}_{\vec{r},k}\right)}{\left|\mathbf{O}_{\vec{r},k}\right|^2 + \beta\langle|\mathbf{O}|^2\rangle_{\vec{r}}} \qquad \textit{eq 5. 4}$$

Where $\langle|\mathbf{P}|^2\rangle_{\vec{r}}$ maps the average probe intensity illuminating each pixel over the whole object ($\mathbf{O}_{\vec{r}}$) and $\langle|\mathbf{O}|^2\rangle_{\vec{r}}$ maps the average object transparency encountered by each pixel of the probe. One should notice that $\langle|\mathbf{P}|^2\rangle_{\vec{r}}$ has the same size as the object. When it is applied to updating process (*eq 5. 3*), an area corresponds to the k^{th} scan position is cut out from $\langle|\mathbf{P}|^2\rangle_{\vec{r}}$. This k^{th} part of the regularisation map is denoted as $\langle|\mathbf{P}|^2\rangle_{\vec{r},k}$. Such a process is similar to take the k^{th} object part ($\mathbf{O}_{\vec{r},k}$) from the guessed object ($\mathbf{O}_{\vec{r}}$). The parameters α and β are fixed at $\alpha = \beta = 1$. We have included them because increasing their value in the final iterations of an adaPIE reconstruction is an effective means to halt the cyclical behaviour common to stochastic gradient descent algorithms.

Calculating the regularisation maps in *eq 5. 3* and *eq 5. 4* every time the probe and object are updated is computationally time-consuming and can be unstable at the beginning of the reconstruction process, where both object and probe can change significantly from update to update. A better way is to take rolling averages; these rolling averages are the 'adaptive' part of the algorithm. To Compute the rolling average probe intensity illuminating each object pixel, we make use of a 'visit' matrix whose entries contain the number of times each object pixel is updated (or visited) during an iteration of the algorithm. The number of visits to pixels near the edges of the object has values of 1 or 2, whilst visits to the centre may be over 100. The visit matrix $\boldsymbol{V}_{\vec{r}}$, is defined by *eq 5. 5*.

$$\boldsymbol{V}_{\vec{r}} = \sum\nolimits_k \mathbf{1}_{\vec{r},k} \qquad \textit{eq 5. 5}$$

Where $\mathbf{1}_{\vec{r},k}$ is an 'all-one' matrix with the same size as the probe. Such a matrix is accumulated according to the scanning positions. The purpose of doing this to count how many times of each pixel in the object is covered by the probe in the scanning process, hence they can be regulated later. The resultant matrix, i.e. $\boldsymbol{V}_{\vec{r}}$, has the same size as the object. The element value in this matrix equals the number of 'being-covered-by-probe' during a scan process, which is affected by both the scan positions and the size of probe. Using the visit matrix, the rolling average probe intensity is updated as follows:

$$\langle|\mathbf{P}|^2\rangle'_{\vec{r},k} = \langle|\mathbf{P}|^2\rangle_{\vec{r},k} + \frac{|\mathbf{P}_{\vec{r}}|^2 - \langle|\mathbf{P}|^2\rangle_{\vec{r},k}}{\boldsymbol{V}_{\vec{r},k}{}^2} \qquad \textit{eq 5. 6}$$

Or:

$$\langle|\mathbf{P}|^2\rangle'_{\vec{r},k} = \left(1 - \frac{1}{\boldsymbol{V}_{\vec{r},k}{}^2}\right)\langle|\mathbf{P}|^2\rangle_{\vec{r},k} + \frac{1}{\boldsymbol{V}_{\vec{r},k}{}^2}|\mathbf{P}_{\vec{r}}|^2 \qquad \textit{eq 5. 7}$$

Which is gradually update the regularisation map by removing its certain amount, then adding an equivalent portion produced by the current guessed probe. By expecting the guessed probe is getting closer to the ground-truth during the iteration, the regularisation map is converging to its 'true' value, which is a position-wise summation of the true-probe. For a given object pixel, the rolling average decays with the square of the number of visits to that pixel.

Regularization of the probe requires the rolling average object transparency through which each pixel of the probe passes. Unlike the object reconstruction, where central regions are updated many more times than the edges during each iteration, every pixel in the probe is updated K times per iteration. The equivalent to the visit matrix ($\boldsymbol{V}_{\vec{r}}$) for the probe is therefore simply equal to K at every point. Hence it is no need to compute it separately. The matching update to the rolling average for the probe regularization is therefore:

$$\langle|\mathbf{O}|^2\rangle'_{\vec{r}} = \langle|\mathbf{O}|^2\rangle_{\vec{r}} + \frac{\mathbf{O}_{\vec{r},k} - \langle|\mathbf{O}|^2\rangle_{\vec{r}}}{K^2} \qquad \textit{eq 5. 8}$$

The rolling average decays as the inverse square of the number of diffraction patterns, K. These choices of decay rate mean the speed with which the rolling averages forget old values decreases approximately linearly with visit map values. Regions of the object regularization map ($\langle|\mathbf{P}|^2\rangle'_{\vec{r},k}$) corresponding to the object edges have a short memory; the rolling average decays very quickly there. Regions corresponding to the object centre have a longer memory and it takes several iterations before the influence of previous probe estimates disappear. The probe regularization map decays even more slowly. This decay rate strategy introduces a sort of inertia to the probe and object updates. The relatively slow decay of the probe regularization compensates for the relatively large number of probe updates per iteration. The quick decay of the object regularization at the edges means the more erratic object updates there are quickly forgotten.

To initialize the two rolling averages, a safe choice is to set them as constants and equal to the maximum initial probe and object intensities, so that:

$$\langle|\mathbf{O}|^2\rangle_{\vec{r},n=0} = \max(|\mathbf{O}_{\vec{r}}|^2) \qquad \textit{eq 8. 1}$$

$$\langle|\mathbf{P}|^2\rangle_{\vec{r},n=0} = \max(|\mathbf{P}_{\vec{r}}|^2) \qquad \textit{eq 8. 2}$$

The pseudo code for the adaPIE is given in ***Pseudocode 5. 1***.

Pseudocode 5. 1: *Adaptive Ptychography Iterative Engine (adaPIE)*

Input: measured diffraction pattern ($intensity$), scanning positions ($positions$), guessed object ($object$), guessed probe ($probe$), No. of iterations (N), Parameter (α, β)

Output: revised object ($revised\ object$), revised probe ($revised\ probe$)

1: $avgeO = (|object|^2)_{max} \cdot \mathbf{1}(size\ of\ object)$

2: $avgeP = (|probe|^2)_{max} \cdot \mathbf{1}(size\ of\ probe)$

3: $visits = \mathbf{0}(size\ of\ object)$

4: **For** (k=1: total number of $positions$) **do**

5: $visits = \boldsymbol{Add}(visits, \mathbf{1}, the\ k_{th}\ positions, size\ of\ probe)$

6: **End**

7: **For** (n=1: N) **do**

8: $positions = \boldsymbol{shuffle}(positions)$

9: **For** (k=1: total number of $positions$) **do**

10: $the\ k_{th}\ part = \boldsymbol{Cut}(revised\ object,\ the\ k_{th}\ positions,\ size\ of\ probe)$

11: $objRegBox = \boldsymbol{Cut}(avgeP,\ the\ k_{th}\ positions,\ size\ of\ probe)$

12: $visit\ box = \boldsymbol{Cut}(visits,\ the\ k_{th}\ positions,\ size\ of\ probe)$

13: $exit\ wave = the\ k_{th}\ part\ \cdot probe$

14: $revised\ exit\ wave = \boldsymbol{\mathcal{P}}_f(exit\ wave, intensity)$

15: $difference = revised\ exit\ wave - exit\ wave$

16: $modification = \frac{probe^*}{|probe|^2 + \alpha \cdot objRegBox} \times difference$

17: $revised\ object = \boldsymbol{Add}(object, modification, the\ k_{th}\ positions, size\ of\ probe)$

18: $revised\ probe = probe + \frac{the\ k_{th}\ part^*}{|the\ k_{th}\ part|^2 + \beta \cdot avgeO} \times difference$

19: $avgeO = avgeO + \frac{|the\ k_{th}\ part + modification|^2 - avgeO}{K^2}$

20: $avgeP\ modification = \frac{|revised\ probe|^2 - objRegBox}{visit\ box^2}$

21: $avgeP = \boldsymbol{Add}(avgeP, avgeP\ modification, the\ k_{th}\ positions, size\ of\ probe)$

22: **End**

23: Apply additional constraints

24: **End**

Note [1]: Temporary variable: $the\ k_{th}\ part$, $exit\ wave$, $revised\ exit\ wave$, $modification$

5.3. Simulation scenarios

Our simulation scenarios are illustrated by *Figure 5. 2*. Every simulation uses a scan pattern comprising $K = 400$ positions arranged in a 20 × 20 grid, with random offsets from perfect uniformity to eliminate the possibility of periodic artefacts in the reconstructions[87]. The boundary traces in *Figure 5. 2 (a)* and *(b)* indicate the extent of the scan pattern. The probes are of size $[M, N] = [512,512]$, with a bright central area in each case of diameter 150 pixels (shown by the shading in *Figure 5. 2 (a)*. The average step size in the simulations is 35 pixels, ±20% random offsets. The object matrix has a dimension of 1350 × 1350 pixels. In all the simulations

the object matrix is initialised as free-space, i.e. $\mathbf{O}_{\vec{r},0} = \mathbf{1}$.

Simulation 1 is an unrealistic but illustrative example. It is designed as an easy test with no noise and a reasonably accurate estimation on the initial probe, so we expect every algorithm to give good results. The two photographs utilised as the modulus and phase images of the object are shown in *Figure 5. 2 (a)* and *(b)*. Images such as these, or the 'cameraman' or 'Lena' images, are often used as examples, although we will see that they seem to give an optimistic view of algorithm performance (perhaps because there is no correlation between the amplitude and phase parts, and because photographs are generally very rich in spatial frequency content). The probe for Simulation 1 is shown in *Figure 5. 2 (g)*. It results from Fourier-transforming an aperture (drawn in a paint package) with a quadratic phase curvature. This simulates the point spread function from a stopped-down lens, imaged at a defocus, and is the sort of probe that might arise in soft x-ray or electron ptychography. The initial probe estimate is simulated in the same way as this ground true probe, but a perfect disc is employed rather than a hand-drawn aperture, and a 7% error in the defocus is introduced.

Simulation 2 is a little more realistic. It uses a complex-valued image of frog's blood (*Figure 5. 2 (e)* and *(f)*), which we generated from the results of a real-world optical bench ptychography experiment. This is a much weaker phase object. The probe (*Figure 5. 2 (i))* simulates a small angle beam of illumination that might arise in hard X-ray experiments and the initial probe estimate is an Airy disc of approximately the right size.

Simulation 3 uses a complex-valued images of a cotton spider (*Figure 5. 2 (c)* and *(d)*), derived again from an optical bench ptychographic reconstruction. The probe (*Figure 5. 2 (h)*) models a random diffuser placed in the path of a laser beam. Such structuring of the probe has been

shown to aid the experimental process by reducing the dynamic range of the recorded diffraction data, and to improve resolution in the reconstructed image[88]. The initial probe estimate is a clear aperture of approximately the correct size, as it is difficult to estimate the random structure of the diffuser. This simulation is set as a challenge for the algorithms, because the initial probe is necessarily far from the true probe, and the interior of the spider in the object contains considerable fine detail.

Since these tests are simulations, algorithm performance can be measured by comparison of reconstructions with a known ground truth. A direct real-space simulation error metric (i.e. Err_s), comparing the reconstructed and ground truth object matrices, must account for various ambiguities s that can arise in ptychography. Although these ambiguities and their removing process have be described in Chapter 4, we will compare the difference between the reconstructed diffraction patterns with the ground-truth to avoid complication here.

$$Err_f = \frac{\sum_k \sum_{\vec{u}} \left(\left| \boldsymbol{\mathcal{F}}\left(\boldsymbol{\psi}_{\vec{r},k}\right) \right| - \sqrt{\mathbf{I}_{\vec{u},k}} \right)^2}{\sum_k \sum_{\vec{u}} \mathbf{I}_{\vec{u},k}} \qquad \textit{eq 5. 9}$$

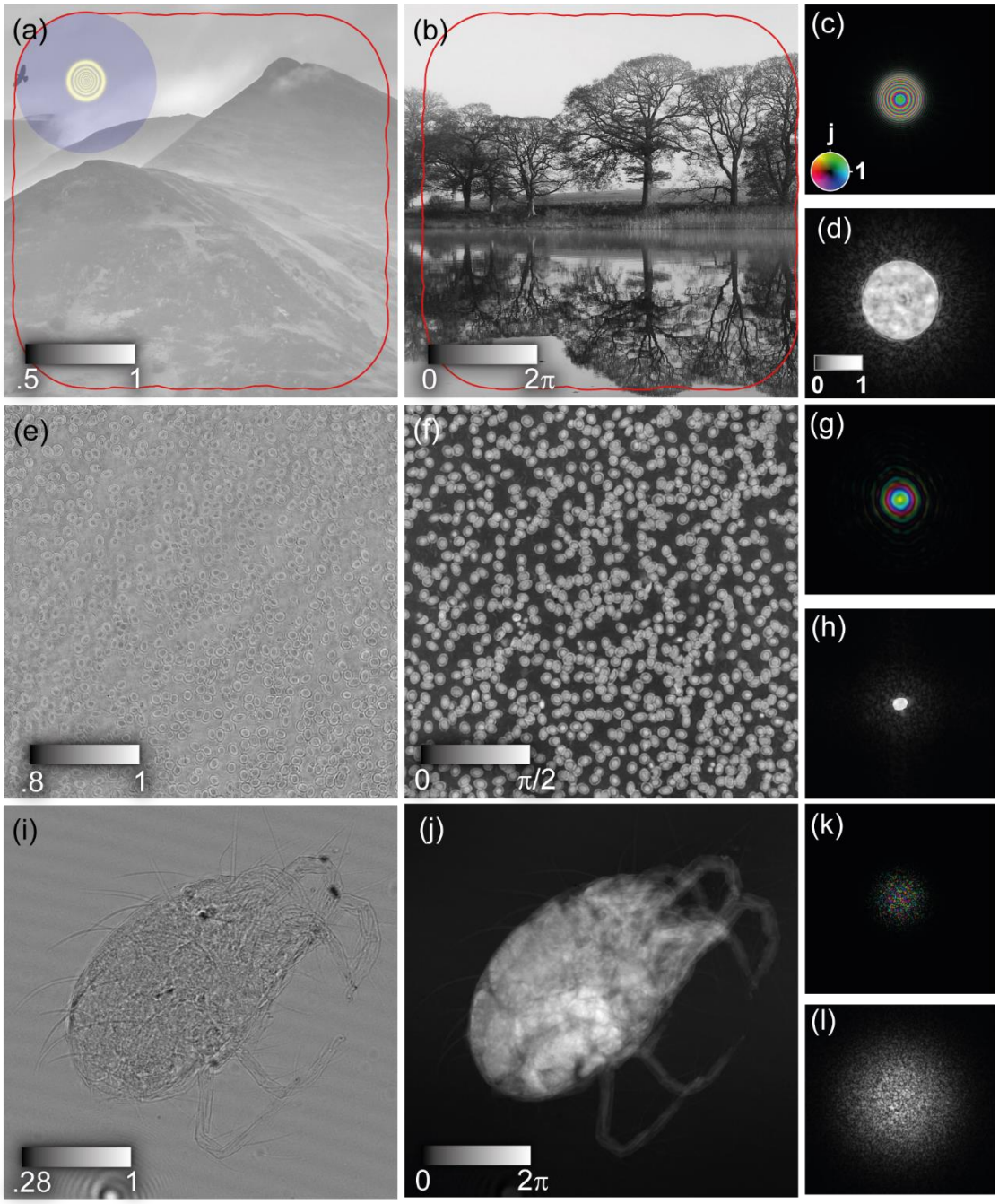


Figure 5. 2. Objects, probes and example diffraction patterns for three simulation scenarios. ***Simulation 1:*** *a) and b) show the object modulus and phase, c) shows the probe on a colorwheel scale and d) gives an example diffraction pattern (contrast enhanced to show detail). The boundary trace in a) and b) indicates the extent of the scan pattern and the highlighted region in a) illustrates the relative size of the probe and the extent of its non-zero values. This boundary and probe size apply to all three simulations.* ***Simulation 2:*** *e) and f) object modulus and phase, g) probe, h) example diffraction pattern.* ***Simulation 3:*** *i) and j) object modulus and phase, k) probe and l) example diffraction pattern.*

5.4. Results from noiseless data (Simulation 1 and 2)

Figure 5. 3 shows the progress of the simulation error metric for the six algorithms, reconstructing data from **Simulations 1** and **2**. In **Simulation 1**, all of the algorithms reached a threshold of $Err_f = 10^{-4}$ where the central region of the reconstructed object amplitude and phase appear visually very similar to the true object. rPIE converged to this point quickest (76 iterations), followed by adaPIE (138), DM (168), RAAR (181), ePIE (377) and ADMM (413). RAAR and adaPIE progressed beyond this threshold, reaching a global minimum equal to the working precision of our computer at which point they were terminated. ADMM we expect would reach this limit given sufficient further iterations. (To determine the working precision, the algorithms were seeded with the ground truth object and probe as initial estimates and allowed to run for a few iterations.) As *Figure 5. 3 (b)* shows, all of the algorithms found **Simulation 2** more challenging. After 5000 iterations the algorithms all passed the visual accuracy threshold, but the visual appearance of the results does not tell the whole story. Whilst the error level is in large part set by errors at the edges of the object reconstruction, these edge regions feed into the centres of the object reconstructions through the errors they impart to the probe. After removing all of the ambiguities between the reconstructed and true objects for each of the six algorithms, mean(maximum) phase errors in milliradians in the central regions of the object reconstructions reached: ADMM $3.7(26)$; DM $0.39(2.7)$; RAAR $1.0(5.3)$; ePIE $0.3(2.9)$; rPIE $0.066(0.86)$; and adaPIE $1.9 \times 10^{-10}(2.5 \times 10^{-9})$.

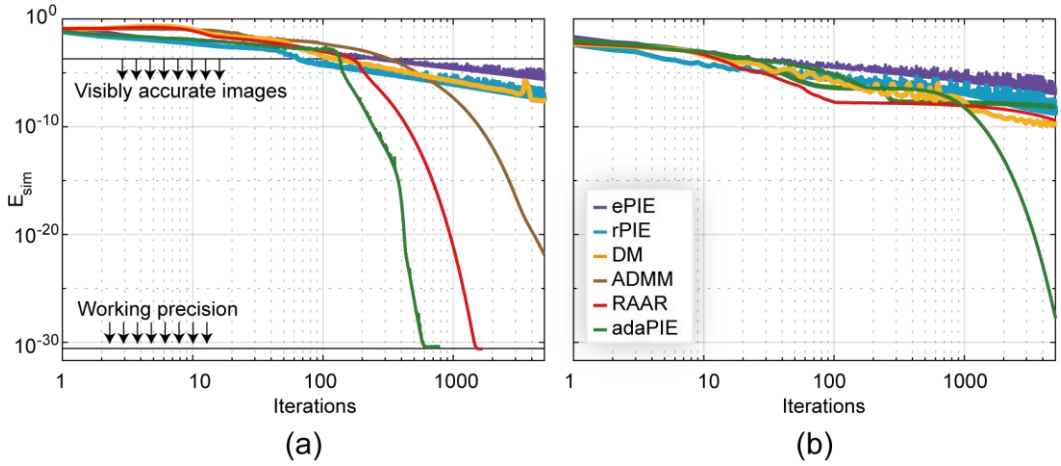


*Figure 5. 3. Results of **Simulation 1** and **2** showing the simulation error metric, Err_f, over 5000 iterations of six ptychographic algorithms. The plots are shown on a log-log scale to highlight the initial convergence rate and the final error level for the different algorithms. Below an error of $Err_f = 10^{-4}$ the images reconstructed by the algorithms are visually very similar to the ground truth. When provided with the ground truth probe and object as initial estimates, all the algorithms give the indicated lower bound on the error value of $Err_f = 4 \times 10^{-31}$, the limit of double-precision accuracy in our simulations.*

5.5. Results from noisy data (Simulation 3)

In **Simulation 3**, different levels of Poisson-distributed noise were introduced to the data. The noise was calibrated by setting the total power in the ground truth probe equal to a fixed number of counts (i.e. the probe's Fourier transform was scaled so that its summed intensity was equal to the required counts). *Figure 5. 4 (a)* shows extracts from the amplitudes of the reconstructions when total counts of photon in the probe was 10^6. This is a relatively low dose, which is supposed to provide slightly poor signal to noise ratio. To highlight the noise content, the gray levels in these images are scaled to $\pm 20\%$ of the mean pixel value.

When noise is present in the data, the cost functions (i.e. *eq 5. 1* and *eq 5. 2*) will have different, non-zero minima. The algorithms deal differently with this situation: batch approaches (i.e. DM, RAAR and ADMM) aim for a fixed point that is at least a local minimum of the sum of the cost functions for all scan positions, whereas stochastic gradient descent algorithms (i.e. ePIE, rPIE and adaPIE) will cycle through the minima. There are two ways to settle down these cycles and draw stochastic algorithms to a fixed point. One is to average the final few iterations of the reconstruction[4], as is often done when training neural networks[63]. Another, which we apply here, is to increase the regularization constants α and β in the final few

iterations. For ePIE, rPIE and adaPIE this second method gives a marked improvement in image clarity. The split panes in *Figure 5. 4 (a)* show the amplitudes of the reconstructions immediately before (bottom right) and after (top left) 50 iterations of the algorithms with α and β multiplied by $50\times$ their initial value. In the interest of fairness, the RAAR, ADMM and DM results in *Figure 5. 4 (a)* include averaging of their final 50 iterations, and the effect of this is shown in the split panes for their reconstructions. Whilst RAAR and ADMM perform well without this averaging, DM fares quite badly overall in this test. Averaging the object estimates over the final 50 iterations somewhat improves matters, but this process has only negligible influence on the final Err_f metric. *Figure 5. 4 (b)* shows how the algorithms deal with a range of noise levels. The error after 500 iterations is plotted, where in each case this includes either averaging (DM, RAAR and ADMM) or increased regularization (PIEs) over the final 50 iterations. The effect of the increased regularization on the adaPIE error level is indicated by the arrows adjacent to each column of the plot.

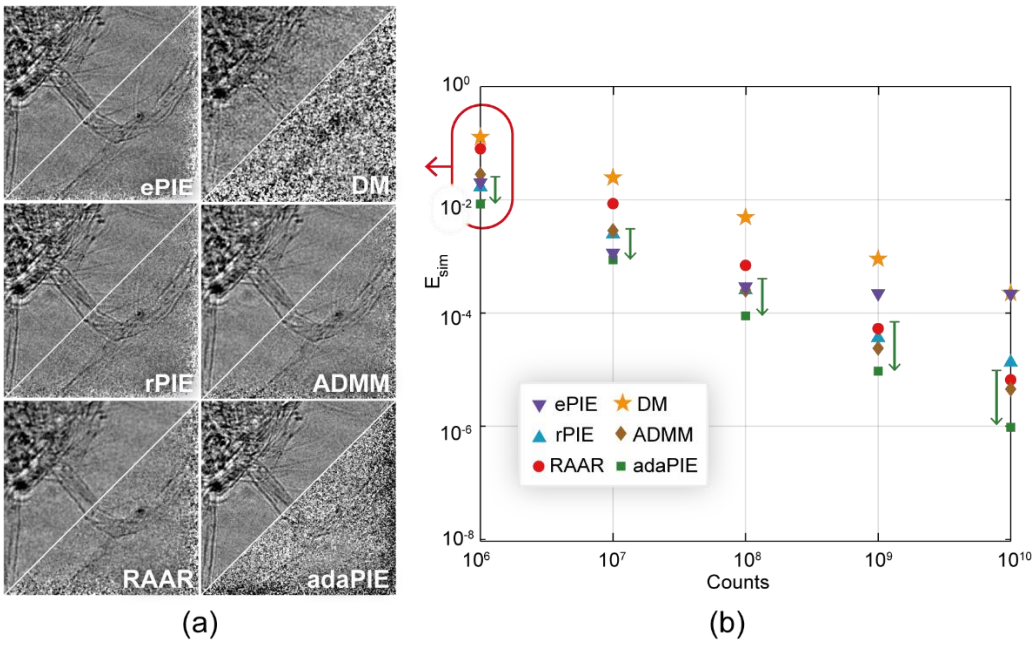


*Figure 5. 4. Results of **Simulation 3**. (a) Cutouts from the modulus of the reconstructions after 500 iterations of the different algorithms, when the total counts in the probe was 10^6. During the final 50 iterations of ePIE, rPIE and adaPIE the regularization constants, α and β, were increased by a factor of 50; the effect of this is shown by the split panes, which show reconstructions after 450 iterations (bottom right) and after 50 iterations with the increased regularization (top left). The DM, RAAR and ADMM split-panes show the effect of averaging over the final 50 iterations of those algorithms. (b) plots the final error after 500 iterations of the algorithms for different noise levels. The effect of the increased regularization on the adaPIE error level is indicated by the arrows adjacent to each column of the plot.*

6. Reconstruction of practical electron data

Data collected from experiment may include various defects. Some of them are inevitable, such as detector noise, while other are caused by improper setup or unknown conventions, for instance, slightly rotated detector, unmatched diffraction patterns with scanning sequence, unknown scanning direction and orientation of loading data from detector to storage. Although artificial error is preventable with careful manipulation and detailed experiment record, it is tedious to check every possible convention in advance. This kind of problem brings significant barrier when different research groups try to exchange their data. This sort of error usually does not get enough attention until all the reconstruction attempts are denied. To prevent fruitless efforts caused by this kind of error, a procedure of pre-check the collected data without reconstruction is suggested in this chapter. This process aims at detecting the potential error in a practical data without extra measurements and minimise their negative influence in the reconstruction. Some of these procedures are also helpful to check whether a desired feature is captured during the experiment in real-time or narrow down to a specific area before the reconstruction. To demonstrate the effectiveness, this procedure is applied on a practical data that collected with STEM.

6.1. Description of the experiment

A group of experimental data is collected with a scanning transmission electron microscope (STEM). The sample was a bilayer of Molybdenum Disulphide (MoS2). The electron microscope operated with a beam energy of 80kV, which is equivalent to a 4.18pm wavelength based on *eq 6. 1*. The influence of specimen thickness is neglected. The camera is calibrated by measuring the 3rd diffraction ring from a gold test sample. Based on the measurements, the approximate camera length is gauged as 183mm. The camera utilised in this experiment has 128 by 128 pixels with a pixel size of 150μm/pixel, which makes pixel angle equal 0.82 mrad/pixel. Each pixel of the reconstructed image corresponds to 0.04nm. A negative defocus (about 56nm) is applied to give the electron probe a diameter of 2nm, equivalent to approximately ⅓ of the probe reconstruction window (=128*0.04nm). The device set-up is demonstrated in *Figure 6. 1*.

$$\lambda = \frac{k}{\sqrt{2m_e E_{eV}\left(1 + \frac{E_{eV}}{2m_e c^2}\right)}} \qquad \textit{eq 6. 1}$$

In the data set considered here, the beam current is 99pA, where 1pA stands for 6242 electrons per millisecond. The step size of the scanning grid is 15% of the probe size, which is about 0.3nm (i.e. 3Å). Such a step size offers about 81% overlap area between two adjacent scanning positions. The scanning grid is a 95 by 95 regular grid. Therefore, the total electron dose can be calculated as 6.87ke/ Å^2 in this setup.

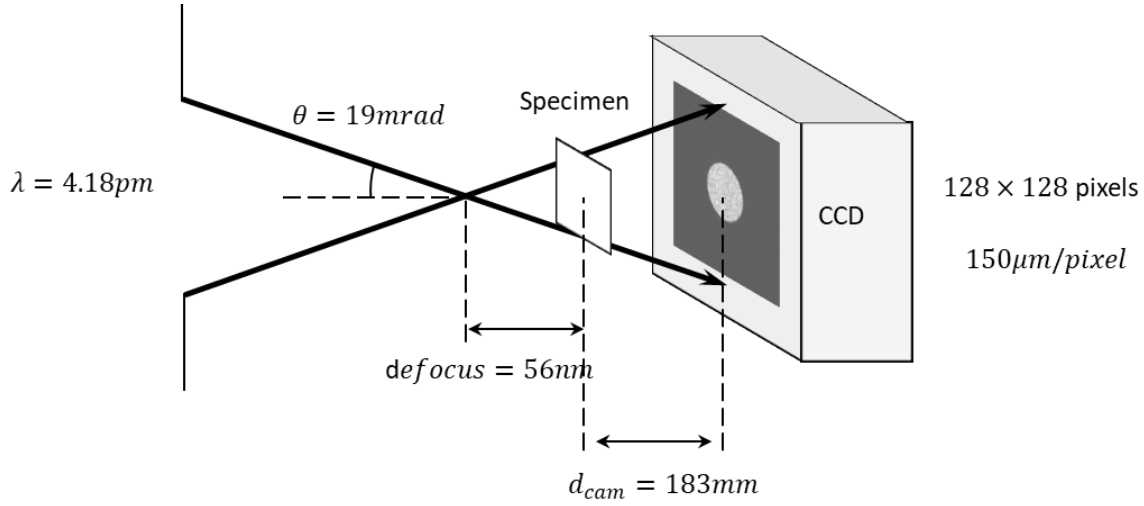


Figure 6. 1. The device set-up for collecting this data set. The camera utilised in this experiment has 128 by 128 pixels with a pixel size of 150μm/pixel. A negative defocus (about 56nm) is applied on the specimen plane. The approximate camera length is gauged as 183mm.

6.2. Match with scanning sequence

Every ptychography experiment data includes the scanning positions and measured intensities, but their order may not match due to improper data storage or various device setup. Such a mismatching, unsurprisingly, causes fruitless reconstruction, but can be prevented easily with the concept of '*General Modulus Image*' (*GMI*).

If each diffraction pattern is represented by its energy as a scalar and fill these values into adjacent blocks following the sequence that defined by the scanning sequence, a blurred image similar to the modulus of object is obtained. This effect is trivial to approve. Each block in the *GMI* represents the energy of a diffraction pattern, which equals the energy of exit

waves. As the probe is consistent and the phase of object does not affect the energy, the energy purely depends on the modulus of object that covered by probe at the corresponding scanning position. Moreover, as the adjacent scanning positions should share more than half of the area, their contrast is limited. Hence it is not a surprise that the *GMI* appears as a 'smoothed' modulus image of the specimen. The modulus image of specimen is unavailable in practical situation, but the smoothness of *GMI* is a helpful tool for checking whether diffraction patten sequence matches scanning positions. As if they are not matching, the GMI will be full of high frequency details and no general outline can be observed.

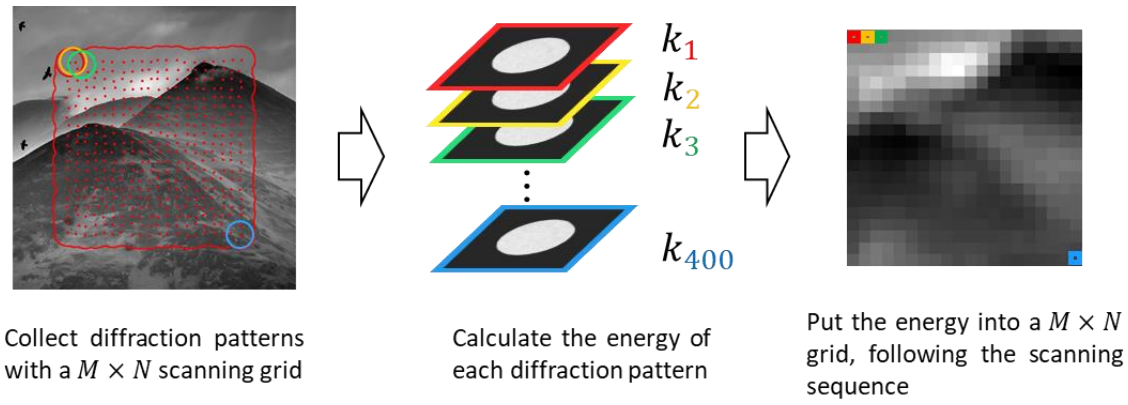


Figure 6. 2. Produce a General Modulus Image (GMI) with collected data. A group of collected diffraction patterns are collected with a $M \times N$ scanning grid. Each of them is represented by its energy as a constant 'k' with corresponding subscripts. These numbers are arranged following the direction of scan to form a general Modulus Image with $M \times N$ resolution. The correspondence is highlighted with different colours.

The blurry effect on *GMI* can also be explained with filter effect. The scanning process of ptychography is similar as the concept of 'Moving Average Filter (MAF)' in signal processing in 2D, where the object is the original signal and the probe is the smoothing kernel. Due to the similarity between ptychography and gaussian smoothing (*Figure 6. 3*), the *GMI* can be considered as filtering the object with probe to some extent. Theoretically, smaller the spot size of probe, less area of specimen is covered at each position, hence a general image with sharper edges will be. On the other hand, smaller the step size, the overlap area between two scanning positions increasing, hence less energy variation and smoother the general image will be. Moreover, this process is similar to the texture extraction that used in image processing. Every time the structure of probe matches the texture of specimen, a peak value in the general image will be detected. Such an effect is demonstrated by *Figure 6. 4*.

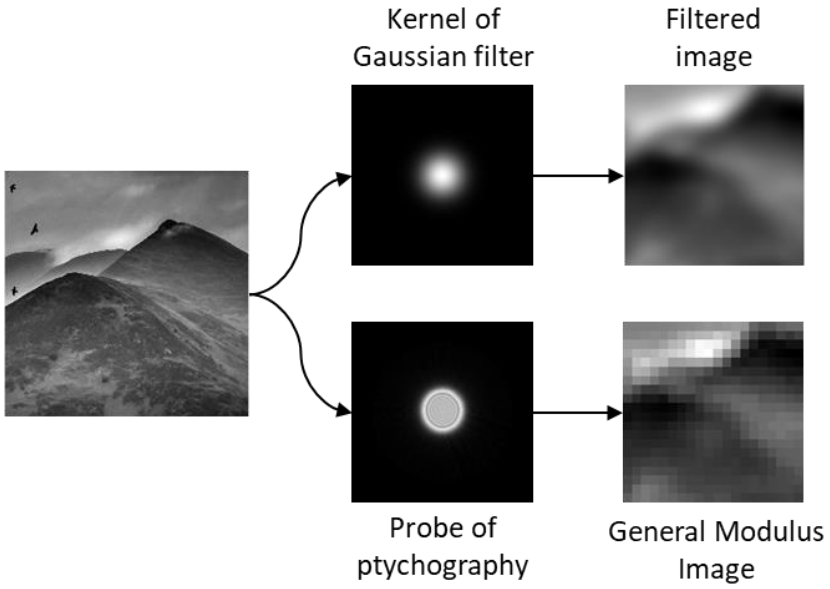


Figure 6. 3. A comparison between Gaussian filter and ptychography. A gaussian kernel with the same size and similar outline of the example probe is created for comparison. The diffraction patterns collected by ptychography are converted into energy values and put into the corresponding area to form a General Modulus Image (GMI). The GMI is similar to the filtered image, which is a smoothed modulus image of the object. When the probe becomes a real matrix, whose modulus follows Gaussian distribution, scanning step size decreases to one pixel and scanned area covers the whole object, the general image equals to the filtered image.

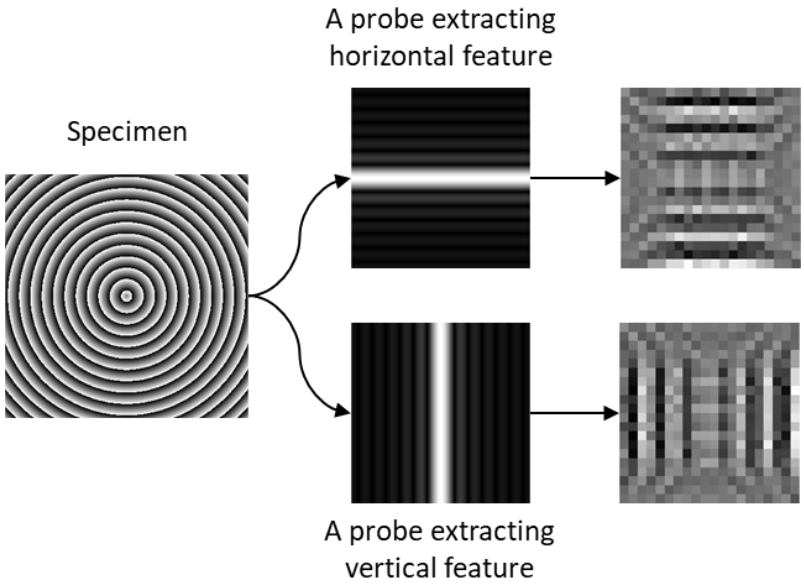


Figure 6. 4. The GMI outline is determined by both the modulus of specimen and probe. When the structure of probe shares similar structure with the covered area of specimen, peak values (the brightest or darkest pixels) appear on the GMI.

Generating a *GMI* only requires diffraction patterns and scanning coordinates. It is not affected by the orientation of diffraction patterns and always smooth when the diffraction patterns correspond to the recorded position. Therefore, this GMI concept provides an option for instantly checking the correspondence between the diffraction patterns and scanning sequence without being affected by the orientation of guessed scanning sequence. An example is given in *Figure 6. 5*. It also illustrates the content of collected data before reconstruction and help user to narrow down an interested area. However, hot pixels in the diffraction patterns sometime break the smoothness of general image. In that case, it is recommended to remove certain amount of high value pixels in diffraction patterns before generating *GMI*.

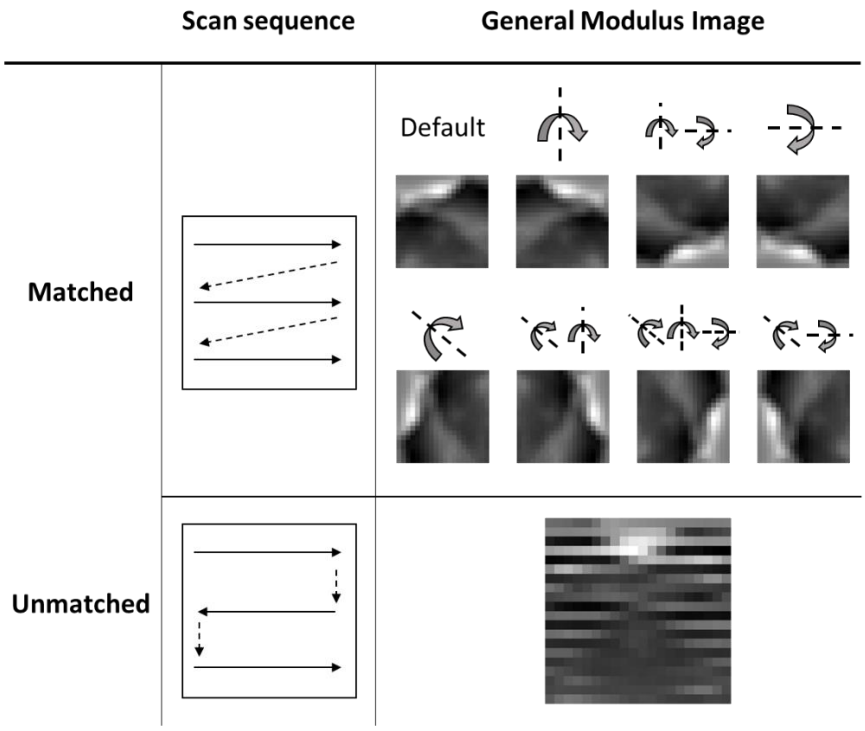


Figure 6. 5. Applying the GMI concept for checking the correspondence of scanning sequence and diffraction patterns. In the first row, the default scanning sequence, which is the exact sequence for collecting diffraction patterns, is demonstrated with arrows. The dotted arrows only represent the connection between scans, they are not a part of it. 8 possible scanning sequences can be obtained by flip and transpose the default sequence, such an effect is named as 'different orientation' in this thesis. They demonstrate the GMI is only determined by the scanning direction and does not affect by its orientation. In the second row, the outcome of an unmatched sequence is shown, which contains much more high frequency information comparing with correct ones. Only default orientation is used for producing GMI in the rest of the thesis, as others are the same picture with different orientations.

6.2.1. Scanning coordinates

For applying the *GMI* concept, we start from the most ideal scenario, which is the scanning positions are provided as regular coordinates. In this case, the scanning direction, grid size and step size can be easily determined by looking at the coordinates. Hence the scanning grid can be easily scaled to adjacent pixels by dividing by the step size. Then a *GMI* is produced by filling these pixels following the scan direction with the corresponding energy. For irregular scanning coordinates, difficulty lies in finding a proper scaling factor that converts the random coordinates into adjacent pixels. This problem has been solved under the help of 'probe calling map'. By dividing with step size and rounding to integers, scan grid with randomness can also be converted into adjacent pixel grids as shown in *Figure 6. 6*. Neglecting the randomness of coordinates does not affect the *GMI*.

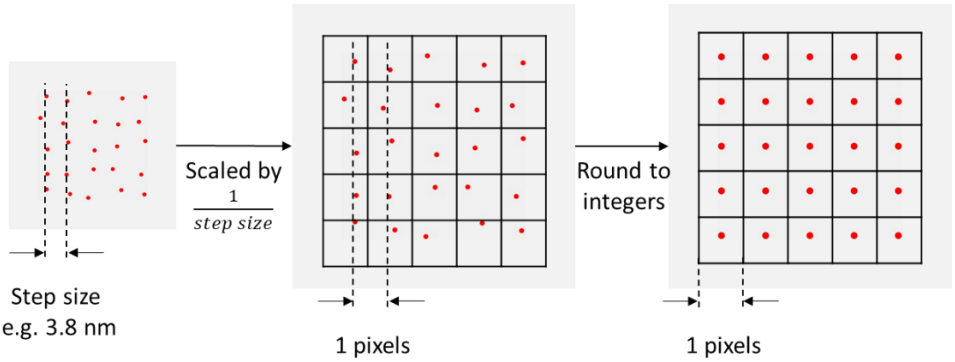


Figure 6. 6. An example of converting a group of scan positions with randomness into adjacent pixel grid for making GMI. Scanning positions are scaled by dividing by the step size and rounded to integers. A GMI can be produced based on these scaled positions.

6.2.2. Square scanning grid with step size

Recording the step size and duplicate a scanning grid is also common in the experiment, as it simplifies recorded data. No randomness is introduced in this circumstance. Generating a pixel grid for *GMI* is straightforward. As the pasting sequence of the diffraction pattern energy does not affect by the orientation of the scanning sequence for a square scanning grid.

However, the problem arises from an unknown scanning direction with a not square scanning grid, whose size is unknown. This situation is common when the collected data is too massive to reconstruct all together or only a part of it contains usable details.

In this case, the *GMI* can help to figure out the most likely arrangement. Since the grid size is unknown, a pair of factors can be guessed by factorising the primes from the number of diffraction patterns. Two possible grids are produced by alternating these two numbers on the row and columns. *GMI* is sensitive to the incorrect grid size. When the grid size is incorrect, multiple stripes appear on the GMI with the same gradient as shown in *Figure 6. 7*. Hence the most likely non-square grid size can be determined by observing the GMI.

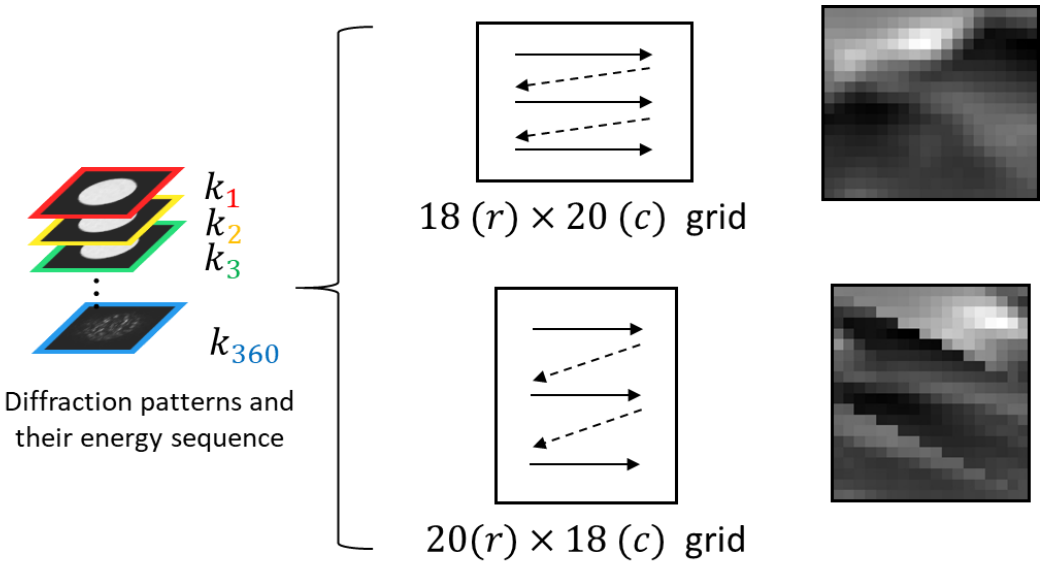


Figure 6. 7. Determine the unknown grid size from the number of diffraction patterns by GMI. In this example, 360 diffraction patterns are given without scanning grid size. By factorising the number of diffraction patterns into 2 close integers, the most possible grid size is either 18 rows by 20 columns or 20 rows by 18 columns. Two GMIs are generated correspondingly and one of them has significant diagonal stripes due to the incorrect row and column numbers. Therefore, a proper non-square grid size is deduced under the help of GMI.

In the collected data, all data are collected by a square scan grid with given grid size. A GMI is produced for each collected data with a default scan sequence. As shown in *Figure 6. 8*, these images demonstrate the guessed scan sequence matches the collected diffraction patterns. Hence a suitable scan sequence is obtained.

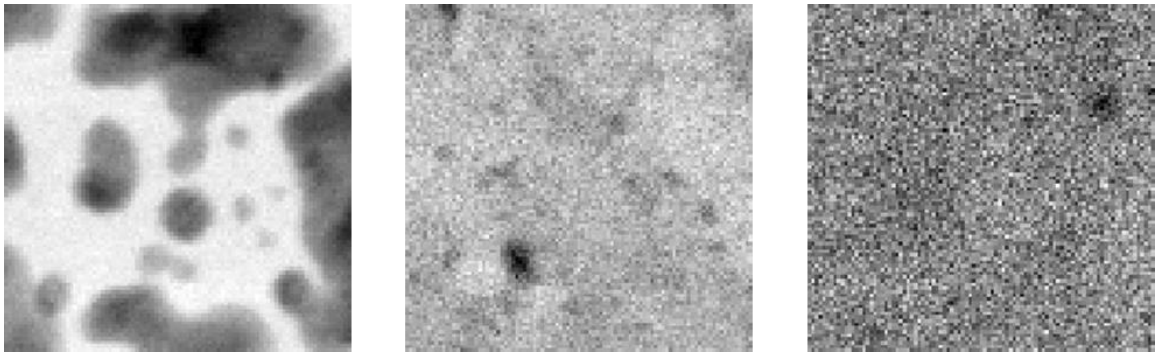


Figure 6. 8. Three examples of GMIs of collected data. From left to the right, the details of GMIs are less significant due to the reducing illumination strength. However, their continuous outline is generally kept. This implies the guessed scan sequence adapts to the collected diffraction pattern.

6.3. Match with scanning direction

General Modulus Image is handy in matching scanning positions with diffraction pattern sequence and checking a non-square grid size. However, it is not capable to identify the orientation of diffraction patterns. Since each diffraction pattern is represented by a scalar during making *GMI*, its orientation property is not considered. Like the 8 possible orientations of scanning directions in *Figure 6. 5*, a diffraction pattern also has 8 orientations without disturbing its content. Due to the device setup and unknown convention, recorded diffraction patterns may be transformed and not match with the direction of scan. A successful reconstruction is not possible without figuring out the correct orientation.

Although it seems 8 possible scan orientations combining with 8 possible diffraction orientations give 64 possibilities, most of them are related by rotation. In other words, one can consider a chosen scan orientation defines the base vectors in that 2D plane. Hence if the orientation of diffraction patterns matches this setup, they are correctly oriented and capable for a successful reconstruction. A quick simulation is conducted to demonstrates how a chosen scan direction re-defines base vectors. A group of object parts are cut out following the default scan sequence, then their orientation is adjusted and pasted back with a re-oriented scan sequence. Object parts are utilised here as they have clearer content than the diffraction patterns. Since the object parts and diffraction patterns share the same orientation, the outcome of this experiment has general meaning to the diffraction patterns as well. As demonstrated by *Figure 6. 9*, for each chosen scan orientation, there is one and only one

group of re-oriented object parts producing a seamless image. This confirms there are at most 8 combinations on the orientations of chosen scan sequence and collected diffraction patterns.

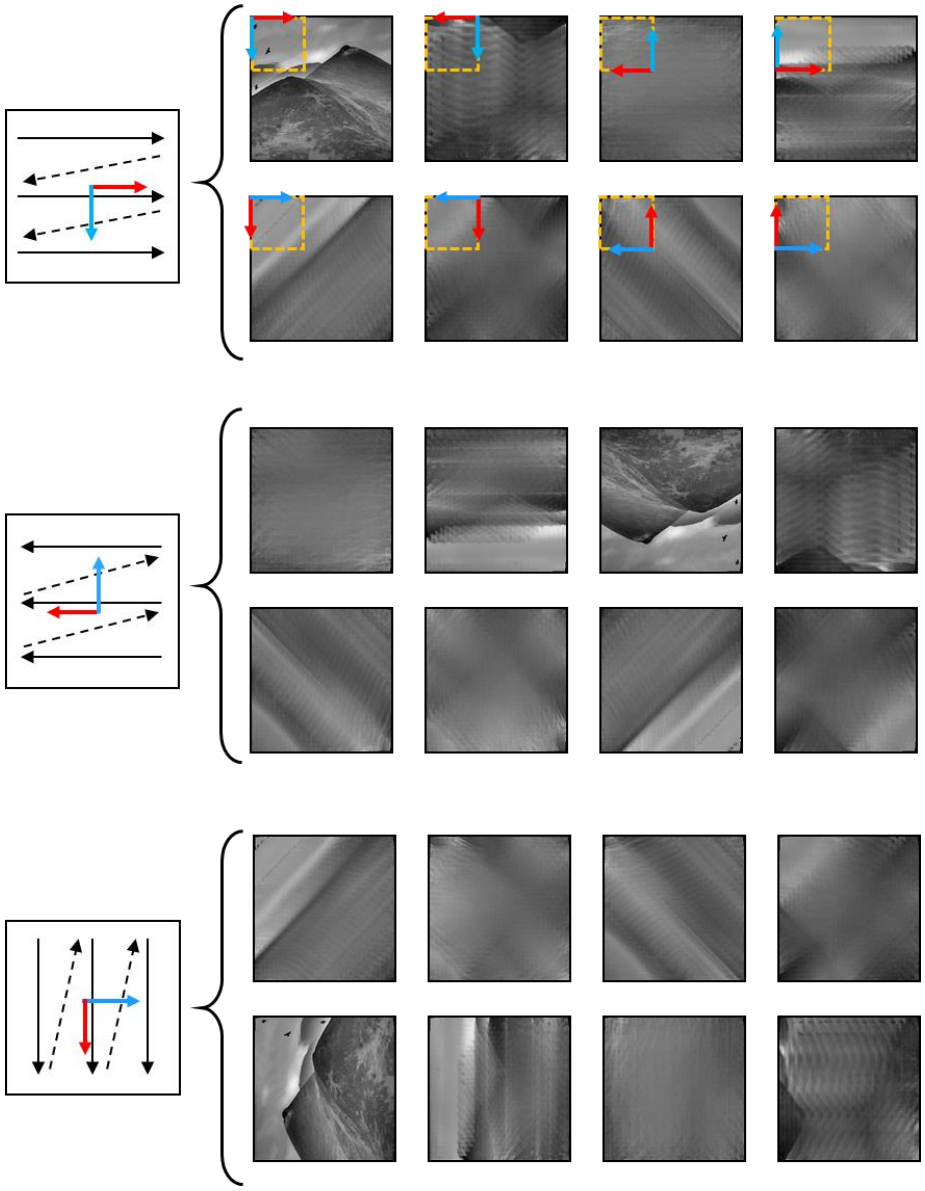


Figure 6. 9. There is one and only one correct orientation of diffraction patterns based on the chosen scan sequence. Three different scanning orientations are chosen as examples. The scan directions are demonstrated with black arrows. Each of them has a pair of red and blue arrows illustrating the primary and secondary scan dimensions. If images produced at each scanning position (e.g. exit wave or the cut-out part of object) are correctly oriented, their normalised, position-wise sum should form a seamless image as the object. Hence, 8 images are produced by 8 differently oriented cut-out parts of object for each scan orientation. The size of a single object part is shown as dotted square in the first overlapped image. In the first group, the orientation of the object part is denoted by the blue and red arrows. Rest groups follow this convention. As shown in the figure, there are 8 different overlapped images caused by the orientation of the parts of object, while the scanning orientation only affect their order. Only one of them has the matched orientation and provides a seamless image.

The next problem needs to be considered is how to relate the content of diffraction patterns with the scanning sequence. In an experiment with defocus probe, which is common in ptychography, the illuminated area of specimen affects the content in diffraction pattern, which varies together with the shifting specimen during an experiment. This leads to a thought: if two adjacent diffraction patterns have the correct orientation, adding them up based on the scaled displacement of their collected positions will cause the similar patterns overlap with each other as shown in *Figure 6. 10*. When all diffraction patterns are pasted with a correctly scaled scanning grid, a blurred image appears, which has a strong relationship with the content of specimen. To bring this coarse idea to practical, the main difficulty is finding the proper scaling factor for the scanning grid.

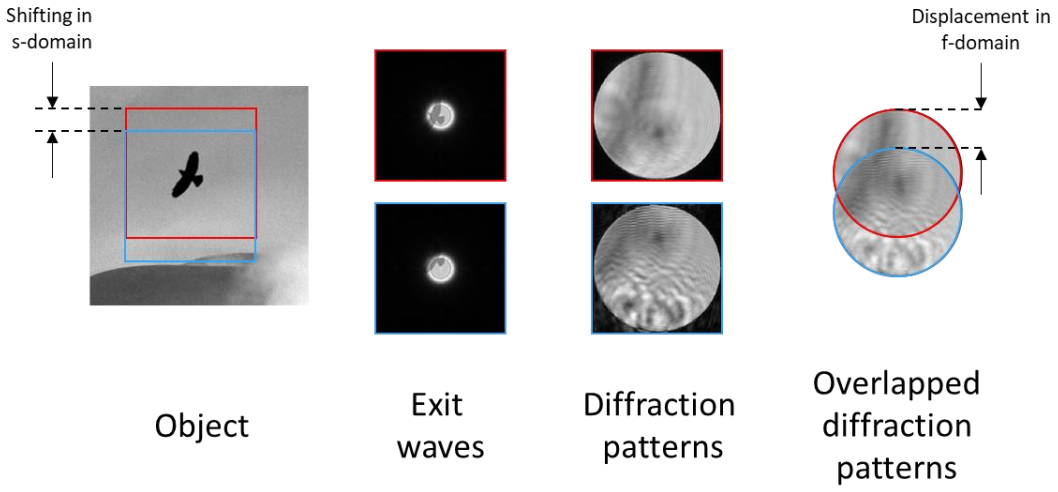


Figure 6. 10. The effect of overlapping diffraction patterns. Two exit waves are produced by illuminating the object at partially overlapped positions and transformed into diffraction patterns. Their correspondence is highlighted by different colours. As shown in the figure, the diffraction patterns contain similar structures. By overlapping them with correct displacement, their structure is enhanced (Their dark edges are removed in this example to give a better contrast).

So far, there are three different approaches for estimating this scaling factor. The most basic one raises from the cause of this phenomenon: the ratio between the diameter of defocused probe and diffraction pattern. Since any interference on the diffraction pattern happens when the probe interacts with specimen and vanishes when specimen moves away from probe, the ratio between the travel distance of specimen and the perturbation pattern should be similar with the ratio between the diameter of probe to the diffraction pattern. Thus, the displacement scaling factor can be obtained by comparing the diameter of a guessed probe

with the diameter of averaged diffraction patterns. However, there is no quantitative way to define 'diameter' for these two properties. In practice, their diameters are usually estimated as the diameter of the bright area. Moreover, the inaccurate guessed probe can lead to a wrong ratio. When the outline of well guessed probe is available, this straightforward method gives a good estimation for further tuning. It can also work oppositely to check whether the guessed probe has at least a similar outline with the true probe.

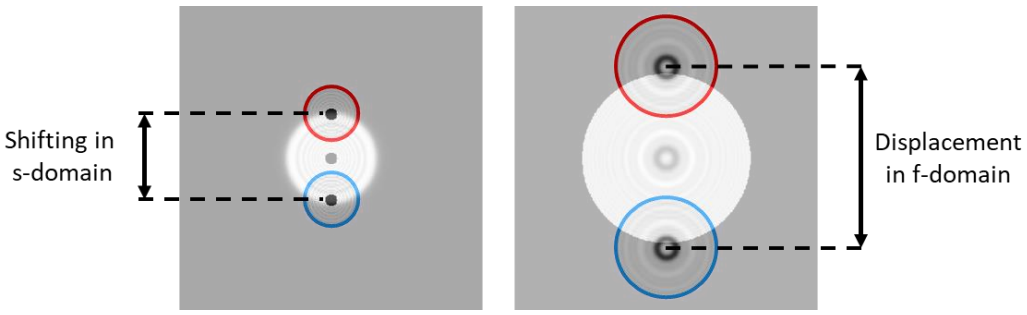


Figure 6. 11. The displacement relationship between the specimen (left) and its interference on the diffraction pattern (right). The background demonstrates the exit wave and diffraction pattern when the specimen sitting right at the centre of probe. An estimated image when the specimen sitting at the edge of probe is illustrated by overlapping the coloured circle and the background. The colour shows their correspondence. The perturbation on the diffraction pattern can be observed when the specimen contacts the bright area of probe. Hence their displacement ratio can be estimated by taking the ratio of a probe and its diffraction patterns diameters.

The second approach is estimating the shifts by fitting two adjacent diffraction patterns. This is done by superimposing 2 diffraction patterns with similar patterns and gradually adjust their displacement until their structure fits together. Then a ratio is deduced by dividing this displacement with their relative shifts during scanning. To get a good estimation, one needs to repeat this process with different diffraction patterns multiple times and takes their average. The good side of this method is that one can immediately notice the unmatched orientation when the estimated displacement does not in the same direction of shifting.

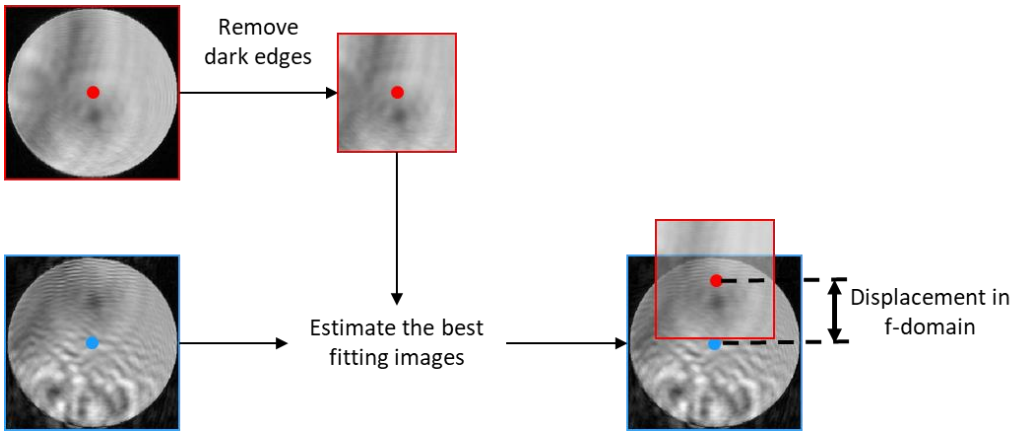


Figure 6. 12. Estimating the displacement in f-domain by fitting 2 diffraction patterns.

The third approach is pasting all diffraction patterns with a scaled scan position. When all diffraction patterns are correctly oriented and pasted with correctly scaled scanning positions, their features overlap with each other and form an image. As shown in the *Figure 6. 13*, one out of 8 possible diffraction pattern orientations produces the best image for a given scan sequence. To minimise the influence of dark edges, the overlapped image needs to be normalised by overlapping the average diffraction patterns. Hence the correct DP orientation is determined.

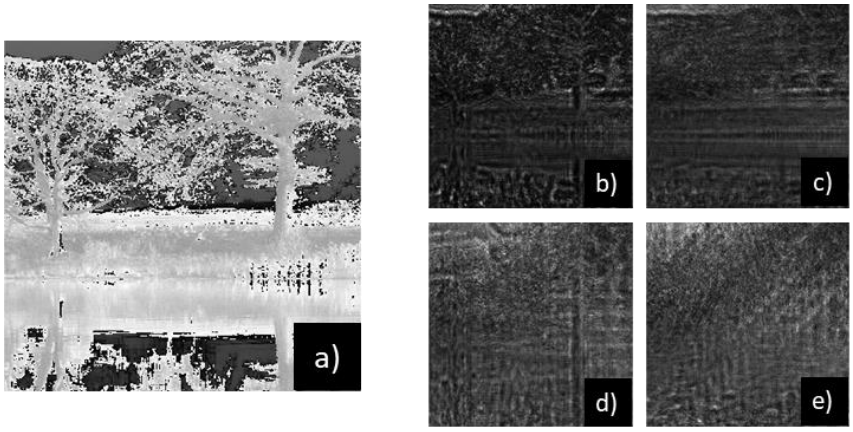


Figure 6. 13. The centre of specimen used for simulation (a) and images formed by overlapping diffraction patterns with different orientations (b, c, d and e). Among the overlapped images, (a) is the formed by correctly orientated diffraction patterns, while the rest 3 are examples formed by incorrect orientations. Hence a correct diffraction pattern orientation is determined for the selected scan sequence.

6.4. Fine adjustment on the rotating angle

The last common defect in the device setup is the rotating angle of detector. A slightly rotated detector is hard to notice during experiment until its data fails reconstructions. The question here is how to determine the rotating angle without access to the detector and also recover the data to its best without commencing a new experiment. A rotated detector dose not only affects the true scanning positions, but also rotates the diffraction patterns. A ptychography seen by a rotated detector is shown in *Figure 6. 14*.

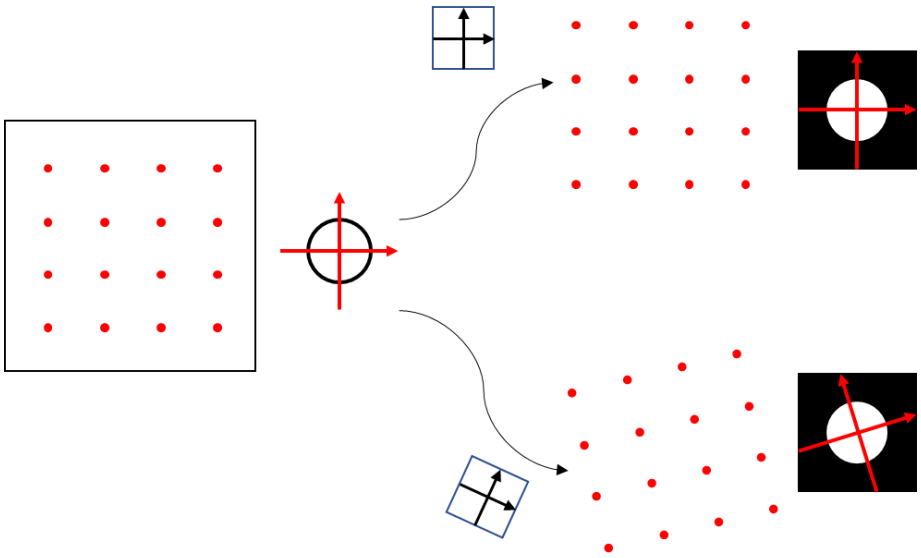


Figure 6. 14. The influence of a rotated detector. The true scanning grid and a sample diffraction pattern is shown in the left. When the detector is correctly calibrated, it observes the correct scan grid and collects diffraction pattern without rotating effect as shown on the top right branch. When an angle is introduced between the detector and scan grid, it collects rotated diffraction patterns with a rotated scan grid. These two data sets have the same difficulty on reconstruction. However, due to the unnoticed rotating angle, the unrotated grid is combined with rotated diffraction patterns, which cause the scan grid mismatches diffraction pattens.

To make the data useable, one need to estimate the rotating angle, which can be achieved by applying the overlapping diffraction patterns idea. This requires user manually testing different rotating angle (from -90 to 90 degrees) on the scan grid, and visually check the overlap images until the image has a good sharpness. For the practical data, a 11 degrees clockwise rotation on the detector is estimated with this method.

6.5. Reconstruction with tuned data

Once the collected data is calibrated following the process explained above, the reconstruction failure due to unmatched measured intensities and scan positions is minimised. These data can be applied to phase retrieval and further tuned to minimise the influence of noise.

6.5.1. Reconstruct the raw data

The first thing needs to be considered is the noise offset. The same data is reconstructed by all ptychographic algorithms explained in Chapter 3 with 100 iterations, which is enough to let them converge and settle down. Among all those algorithms, results from ePIE, rPIE, ADMM and RAAR are shown in *Figure 6. 15*. These results are chosen because they all produce images that reveal details to different extent, while others end up in noisy images. ADMM and RAAR reconstruct similar probes, which implies a decent reconstruction on the probe. Meanwhile, PIEs give poorer resolution and smaller view area than the ADMM and RAAR in this test. This is not a surprising result, as the PIEs are more sensitive to the noise on the intensity measurements.

Meanwhile, the energy of probes reconstructed by PIEs keeps increasing during the iteration as shown in *Figure 6. 16*. This is due to the accumulation of complex scaling ambiguity (ae^{jc}), which is explained in Chapter 4. At the same time, the modulus of objects gradually reduces and leads to images with poor contrast. A probe energy limitation constraint is added to PIEs in the later tests to prevent this problem. In the interest of fairness, the same energy constraint is added to other algorithms. Although aPIE does not have problem in the probe energy, it does not give a successful reconstruction as shown in *Figure 6. 17*.

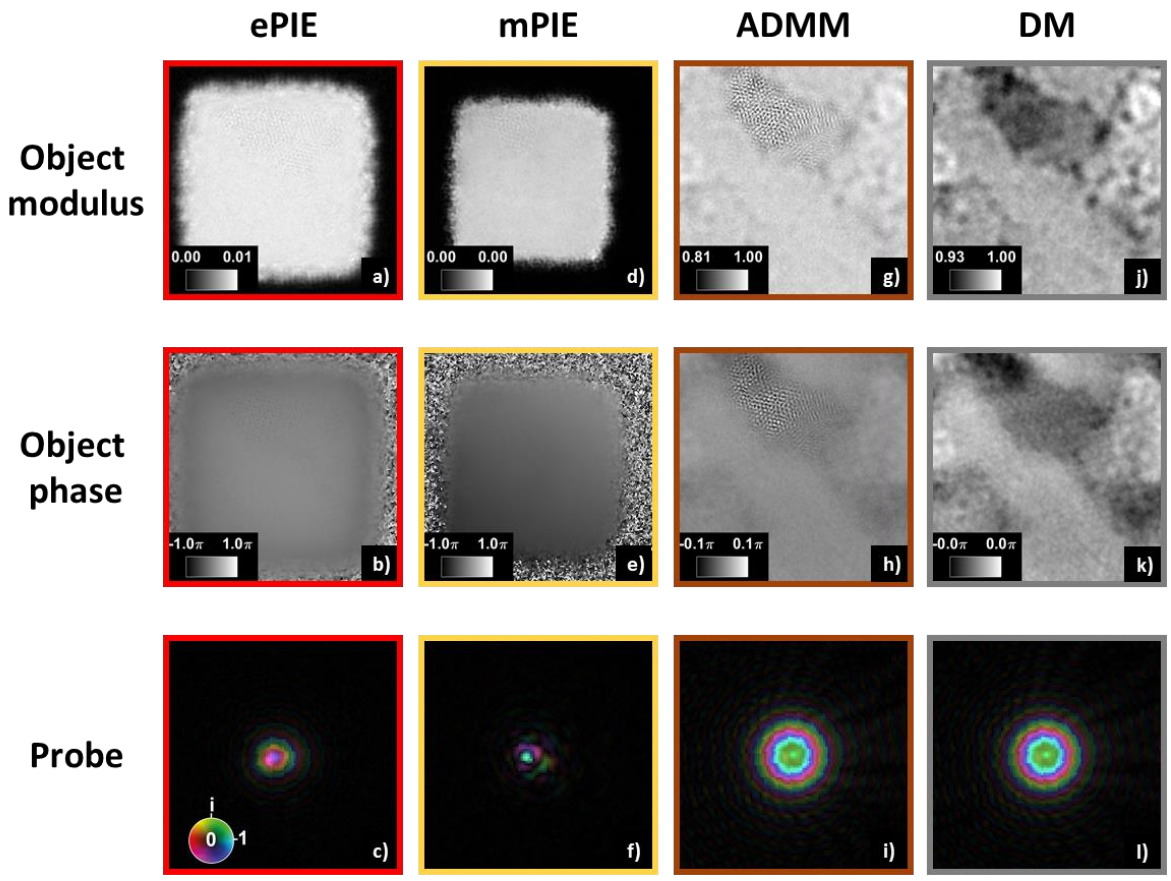


Figure 6. 15. Reconstructed images by different algorithms. Ambiguities has been calibrated to the same level. Each column represents the reconstructed images from one algorithm. From top to bottom, each row represents the modulus of object, phase of object and probe respectively. PIEs are less successful in this test. Both the resolution and the size of informative area are poorer than ADMM and RAAR.

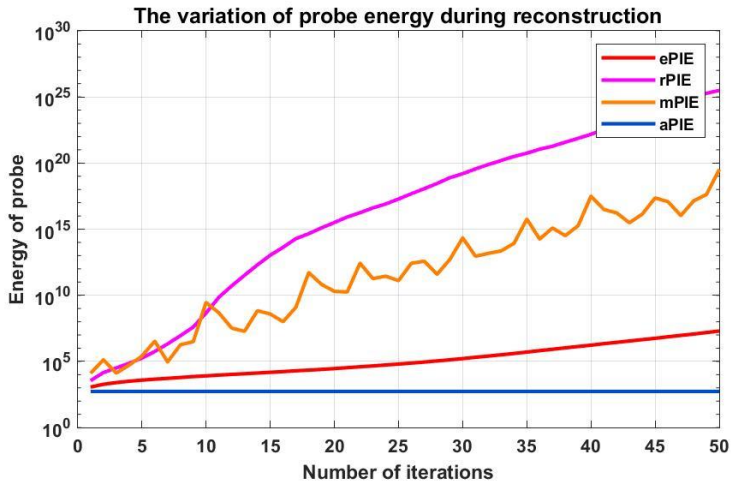


Figure 6. 16. The diverge of probe energy as the PIE iteration goes. This figure demonstrates how the energy of probe varies without setting a limit. This causes the modulus of reconstructed object losing contrast.

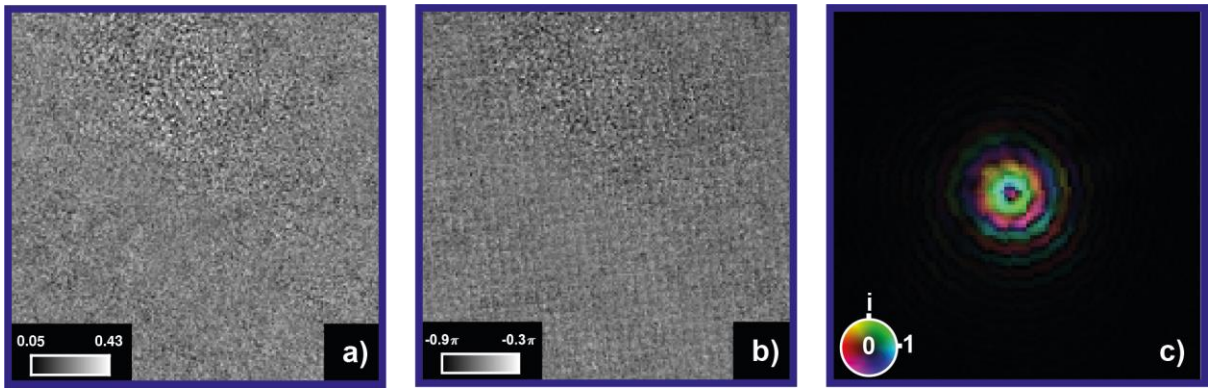


Figure 6. 17. The reconstructed images from adaPIE. From left to right, the modulus of central specimen (a), the phase image of central specimen (b) and the probe (c).

6.5.2. Estimate the noise offset

The previous test indicates the existence of background noise in the intensity measurements, though its level is unknown. To minimise the negative influence of the noise, various threshold levels are attempted. The range of possible threshold is estimated by observing the distribution of the readings. *Figure 6. 18* demonstrates the average of all measured intensities and sorted pixel values of these diffraction patterns. As shown in the plot, about 36% of the pixels have readings smaller than 0. These negative readings are caused by the detector noise and should be neglected. Then, by drawing a circle containing the spot of averaged diffraction patterns, the percentage of pixels dominated by the spot against all measured pixels can be estimated, which is about 10% in this test. Hence the 10% largest readings are considered as having good signal to noise ratio, and the noise threshold should not be larger than them, which is about 300 for this data set. Therefore, the background noise offset has high possibility falling between 0 and 300. For estimate the background noise offset, values within this range are tested with step size as 50. With each attempted noise threshold, all intensity measurements are subtracted by the threshold, and all negative pixels are set to zero.

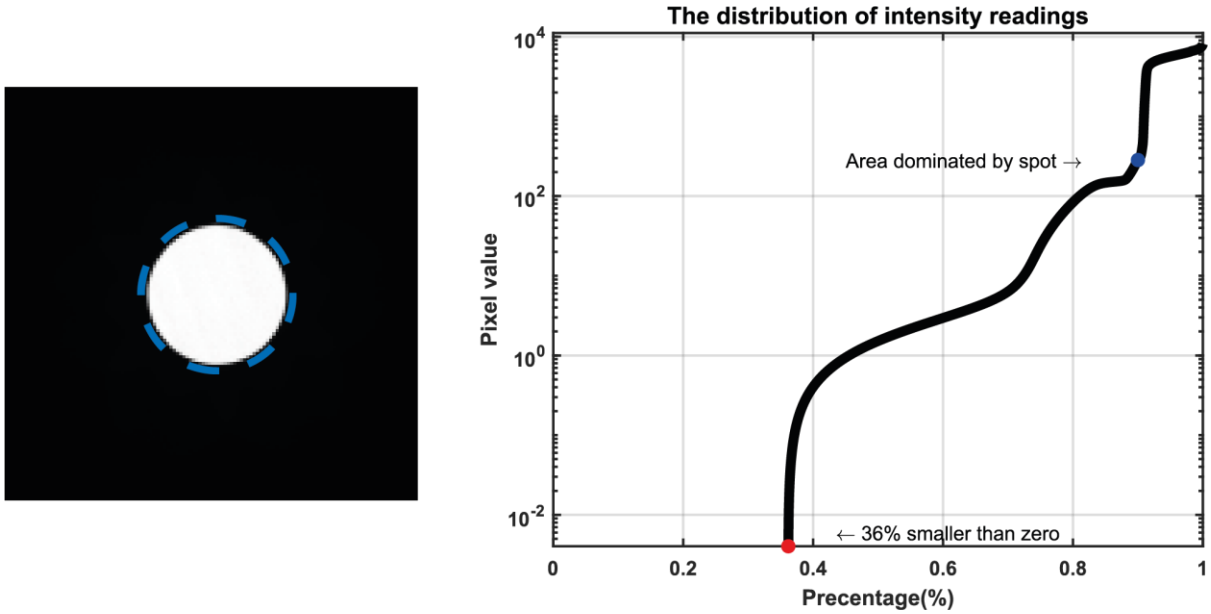


Figure 6. 18. The estimate the background noise offset. The left picture is the average of all diffraction patterns. The approximate spot of diffraction patterns is labelled by a blue dotted circle. Pixels inside this circle has relatively high signal to noise ratio due to more detected electrons. In this data set, 10% of pixels fit into the this estimated spot with minimum value about 300. This 'turning point' is highlighted as a blue dot on the right line plot. The right picture illustrates the sorted pixel value against the percentage over all pixels in measured intensities. As its y-axis is log-scaled, values not larger than zero are omitted. The first pixel larger than zero is highlighted with a red dot together with the percent of pixels had value smaller than zero.

To find out the influence of different noise threshold quickly, the object and probe reconstructed by ADMM from the previous run is utilised as the initial guess. Such well guessed object and probe makes the reconstruction quickly converge to stable state. Since the diffraction patterns are modified, the f-domain error metric cannot evaluate the quality. The proper noise threshold is estimated by observing the reconstructed images.

Some reconstructed images from ePIE and ADMM are demonstrated in *Figure 6. 19*. These two algorithms are chosen as they are highly representative for the gradient descent and set-projection methods. As shown in this figure, ePIE is more sensitive to the modification of diffraction patterns. As the noise threshold increases, more details are revealed on the phase image, though the modulus image gradually lose details. On the other hand, ADMM is less sensitive to the varying threshold. Their results barely change until a high threshold removes too much information and causes blurred reconstructions. To balance these effects, the noise threshold is chosen as 150 for the further reconstruction. The final reconstruction after removing the background noise offset is shown in *Figure 6. 20*. As a comparison, the crystal

structure of MoS2 is shown in *Figure 6. 21*. Only the ePIE and ADMM can reveal the hexagonal structures for the specimen in this experiment.

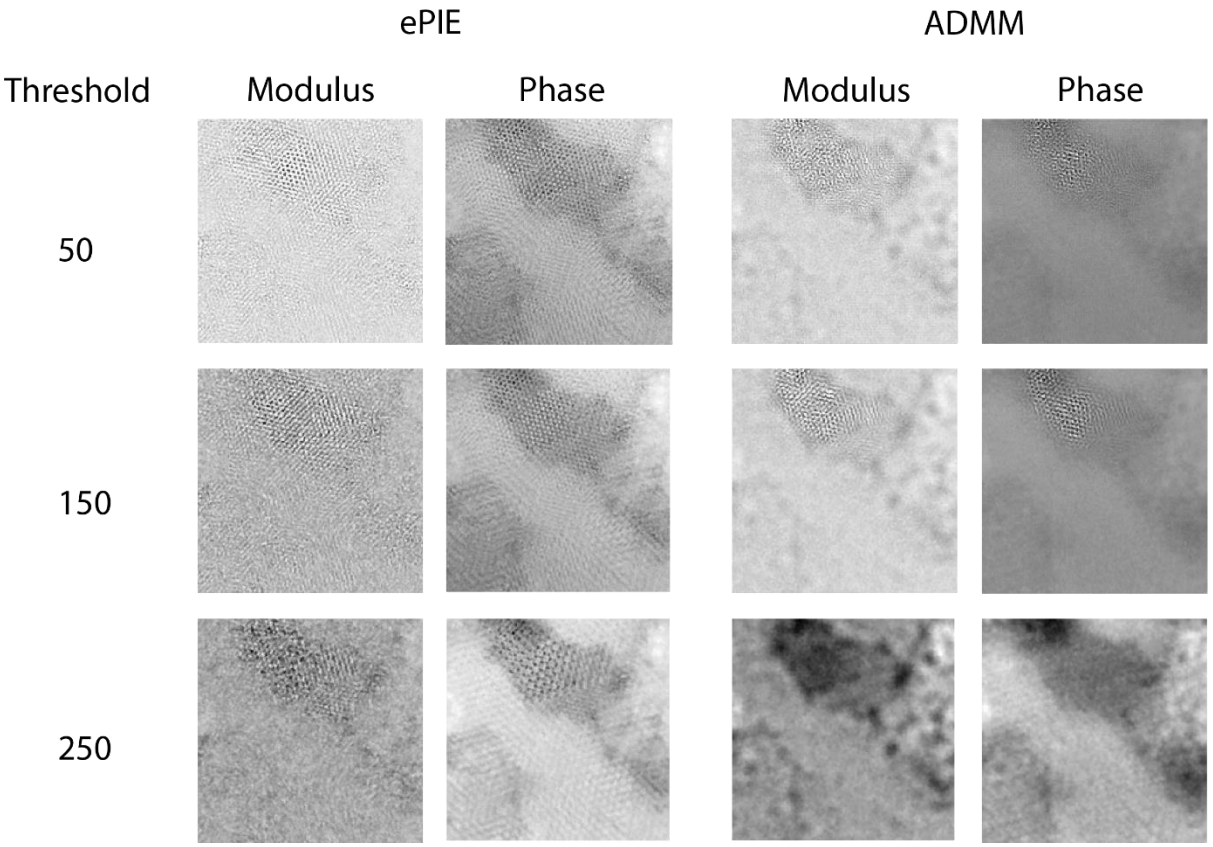


Figure 6. 19. Reconstruction results after applying different noise threshold. From the top row to the bottom, the threshold value of each row is given in the first column. The modulus and phase images of object reconstructed by ePIE and ADMM are shown in the figure. These two algorithms are chosen as they give the overall best reconstruction. The results of ePIE vary significantly as the increasing of noise threshold, while the results from ADMM are less sensitive to the modification.

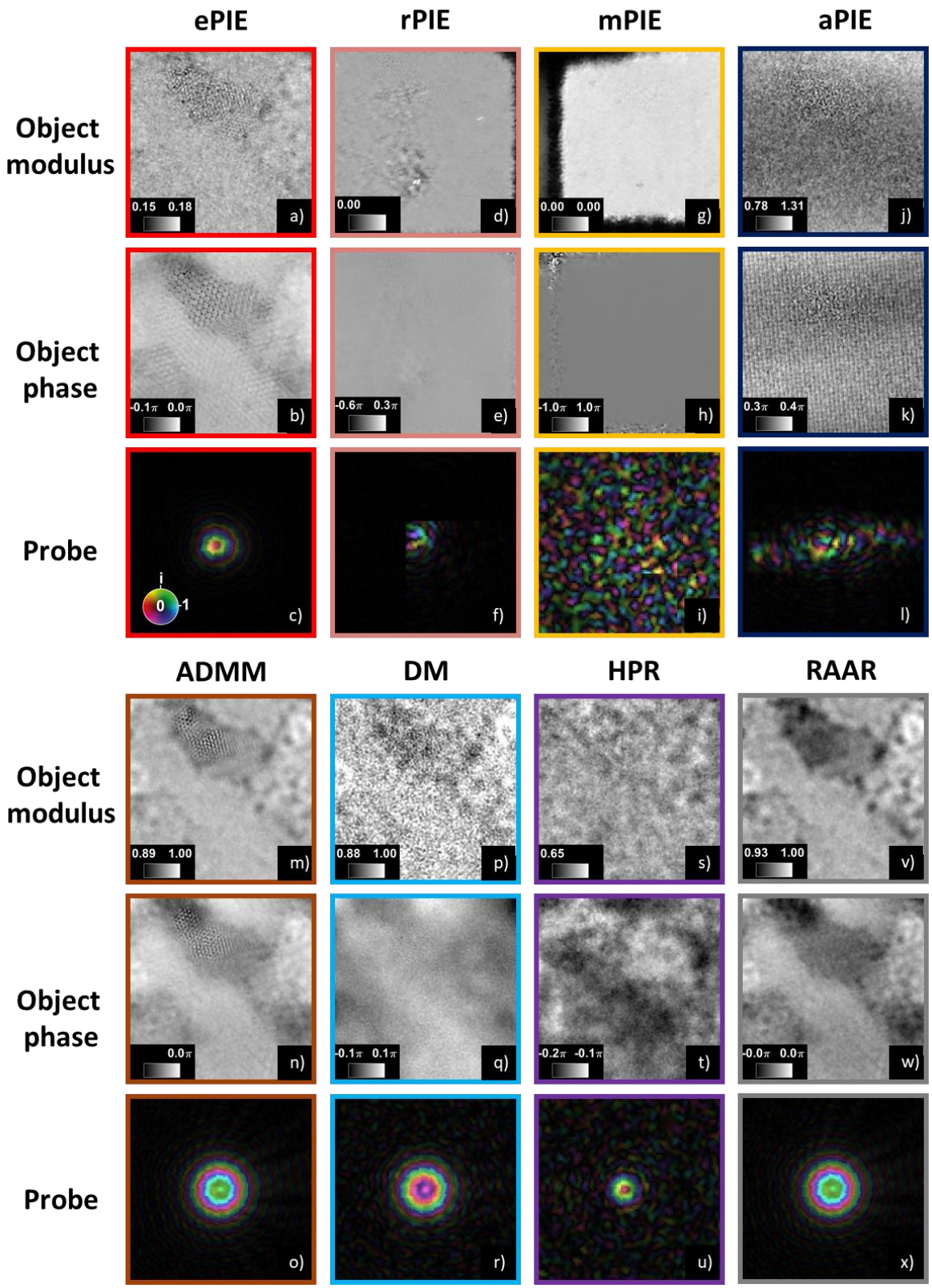


Figure 6. 20. The reconstruction results after removing the noise offset from the measured intensities. The object and probe images from the same algorithm is put in outlines with the same colour.

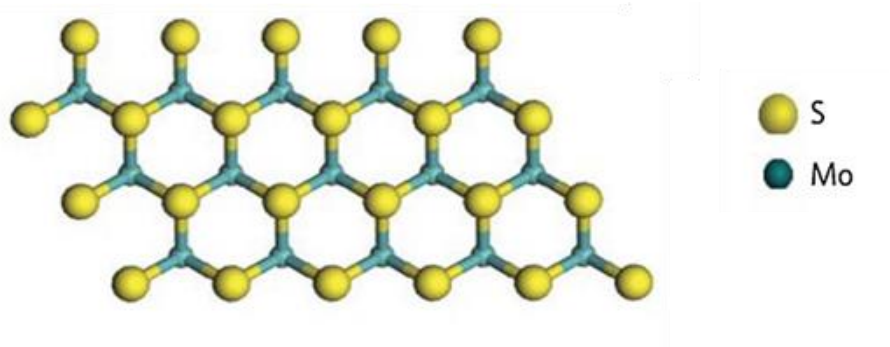


Figure 6. 21. The crystal structure of a bilayer MoS2 sample. The hexagonal structure is a typical geometry from its top view. The sulfur atoms are in yellow colour, while the molybdenum atoms are in green colour[68]*.*

7. Other tricks

Phase retrieving algorithms are not the only way of extracting information from the data collected by ptychography. Some tricks can be applied to improve the quality of initial guess or produce extra constraint to improve the robustness of phase retrieving. These concepts are explained respectively in each section.

7.1. Probe calling map (data analysis)

In ptychography, a well illuminated area of specimen is covered multiple times by different part of probe. Such a relationship makes one part of guessed object contributes to the updating of different parts of probe at different scan position and vice versa. This relationship can also be considered in another way—one part of probe is related to its other parts under the interference of guessed object. This self-referring relationship is determined by the utilised scan grid. To unveil this relationship in an understandable way, the reconstruction process is imagined as stamping a paper with a stained stamp. The probe is the stamp while the object is a blank paper. To trace the influence of a specific pixel, only that pixel is stained in the beginning. When the probe stamps on the object following the scan positions, the stained probe pixel contaminates the contacted object pixel. The previously stained object also affects other probe pixels at the same time. After one iteration, all stained probe pixels are caused by the first contaminated pixel and considered as directly related to the first pixel through a full reconstruction process. Therefore, if one pixel of probe is significantly poorly reconstructed and not updated in time, it will have directly impact on these related pixels within next updating. As this pattern illustrates how one-pixel contacts (calls) other individuals under the influence of scan positions, it is named as ‘calling map’.

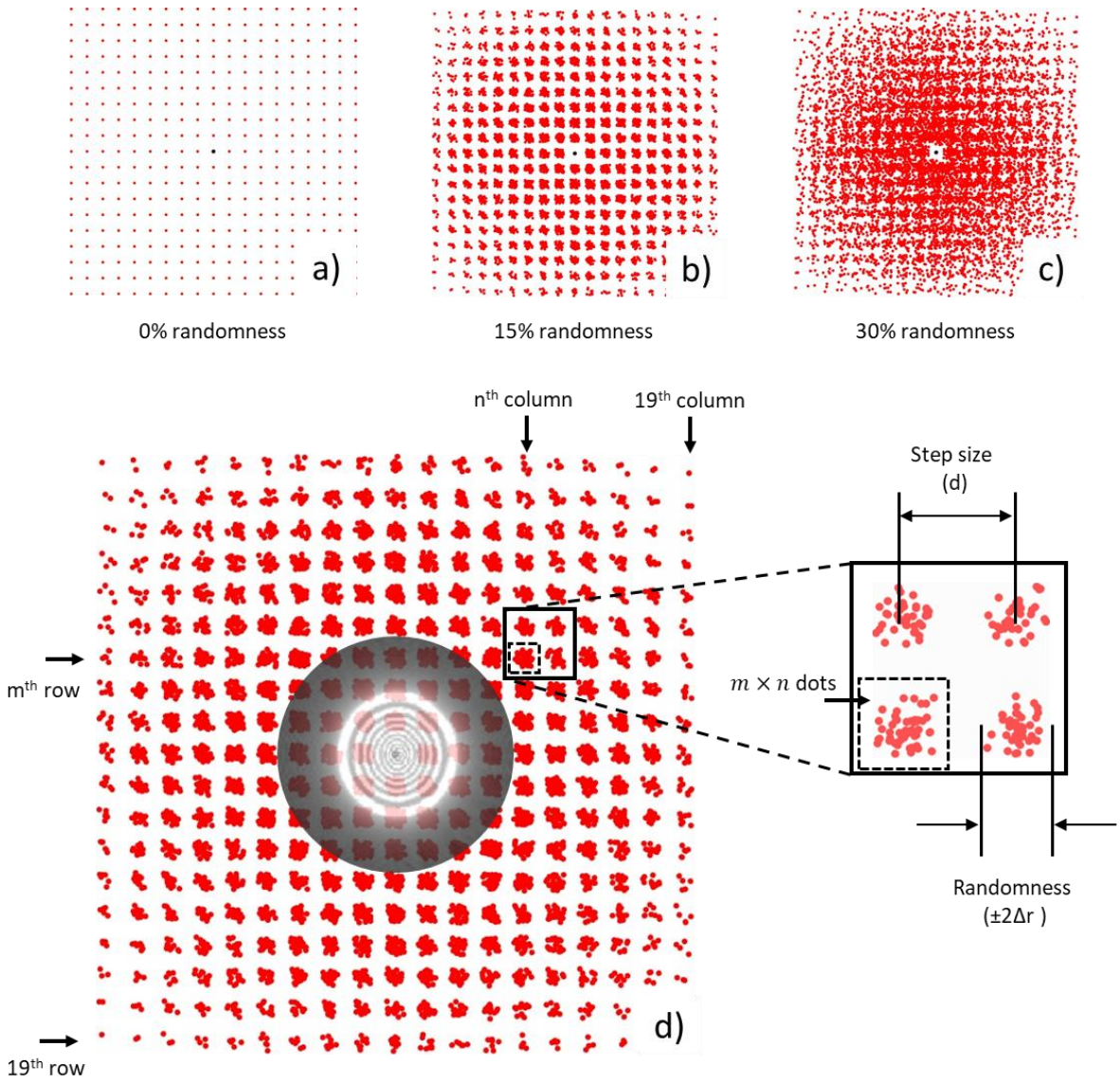


Figure 7. 1. Appearance of 19×19 *calling maps formed by* 10×10 *scan grids with increasing randomness. As shown on the top row of this figure, when randomness is 0% of the step size (a), each group of dots overlap with each other. As the randomness increasing, the groups gradually expand. The randomness of dots in each group (*$\pm 2\Delta r$*) is double of the randomness of scan grid (*$\pm \Delta r$*). When the grid randomness is larger than 25% (e.g. 30% in figure (c)), adjacent groups start to contact with each other. Figure (d) demonstrates how a probe is related to itself in s-domain under the influence of a 15% randomness scan grid. There are 8 groups completely covered by the probe spot. This implies a good self-referring relationship.*

A $M \times N$ scan grid with randomness $\pm \Delta r$ leads to a calling map with $(2M-1) \times (2N-1)$ groups of positions, which appear as dots on the calling map. As long as the randomness is less than 25% of the step size, there is clear gap between these groups. For the group on mth row and nth column, it contains $m \times n$ dots within it. The group closer to the centre of this map contains more dots. The geometry centre of each group can be estimated by computing

their average value. The step size hence can be estimated by finding the distance between the geometry centres of two adjacent groups. The randomness of each group is the double of scan grid randomness (*eq 7. 1* and *eq 7. 2*). This method uses the scan position to the max for estimating unknown step size and randomness with decent accuracy. The calling map provides acceptable step size estimation even with randomness slightly larger than 25%, although groups start to overlap in that case. As long as there are sufficient individuals in the group (a large enough scan grid).

$$\overrightarrow{r_n} = \overrightarrow{r_{n_{grid}}} \pm \Delta r \qquad \textit{eq 7. 1}$$

$$\overrightarrow{r_n} - \overrightarrow{r_{n-1}} = d \pm 2\Delta r \qquad \textit{eq 7. 2}$$

Some facts can be concluded by looking at the calling map. First, calling map only depends on the scan grid. As the scan grid shows how the pixel of object referring to others object pixels during reconstruction, the calling map is the counter part of scan grid for probe. Second, there is always a blank zone around the origin. This implies any pixel of probe cannot affect the pixels right next to it through s-constraint. They are not sharing any information, either good or bad reconstruction, unless f-constraint is applied. Third, each group collapse into a single dot when randomness equals zero. In that case, the communication between probe pixels is reduced to the minimum level. They can only communicate through the f-constraint.

7.2. Artificial randomness to the scan grid (reconstruction)

As demonstrated by the calling map, a regular scan grid gives weak s-domain relationship to the probe. Hence each group of probe pixels are not sharing information effectively by s-constraint and not benefit the converging in the beginning. To improve the s-domain connection of probe, some randomness can be artificially added into the scan grid in the beginning of reconstruction. In this way, poor probe pixels contact more object pixels that are modified by other revised probe pixels. As long as the specimen is not full of details in the scale of probe pixel, a tiny drift from the correct scan grid does not significant change the

values that each probe pixel contacted. Since the priority at the beginning of reconstruction is not refining details, this method is also acceptable and provides a good reconstruction on the coarse outline. This artificial randomness also supresses the raster grid ambiguity, though the ambiguity comes back once the randomness is removed. A dynamic randomness (that change from time to time between iterations) is preferred, as they also improve the s-domain communication of object pixels due to slightly varied scan grid.

Test results prove this concept has negative influence on the algorithms that updates the object with all exit waves together (mainly projection and reflection algorithms), as they are very sensitive to the mismatching scan grid. It has no observable influence on PIE families when the scan grid already has randomness. However, it gives better initial convergence with PIE family with regular scan grid, and the benefits increases when the initial probe is inaccurate.

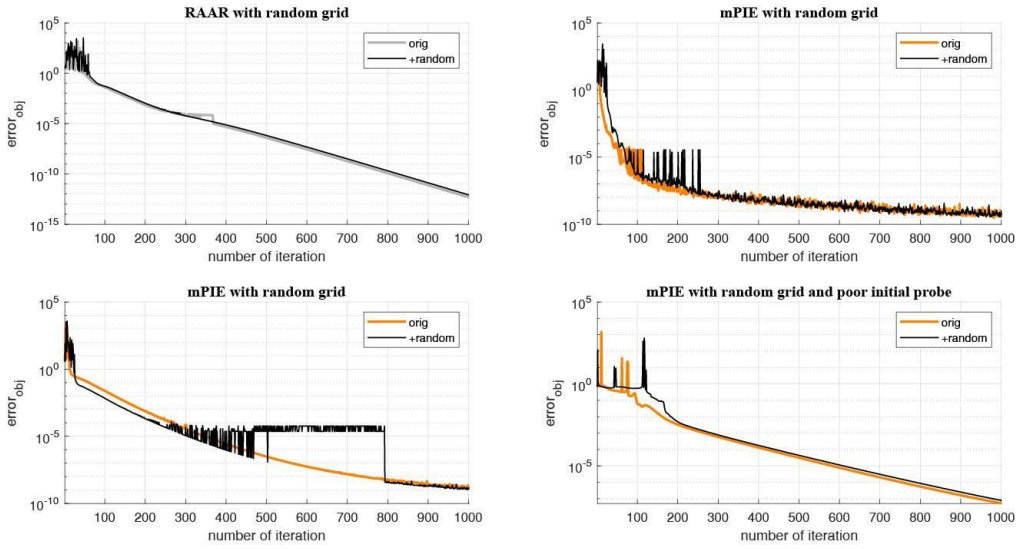


Figure 7. 2. The influence of adding artificial dynamic randomness to the scan grid in the beginning of reconstruction. The platform error metric is due to the data precision and can be ignored for this topic. In this test, ±2 pixels dynamic randomness is added onto the scan grid in the first 20 iterations. Between 21st to 25th iterations, the randomness is static. After that, the randomness is removed. In these tests, the added randomness does not have significant influence on the outcome.

7.3. Smash a collapsed probe (reconstruction)

Collapsed probe is a common cause of stagnation. When it happens, the energy of probe concentrates into an area that is significantly smaller than the true one. When a probe collapses, its increasing dark edges collect less information from object while producing guessed exit waves. This makes the mismatch in overlapped scan position less detectable, hence reduces the effectiveness of f-constraint onto the object. Meanwhile, dim probe pixels have trend of magnifying the variation caused by revised exit waves during updating object. Such a response lead to instability on the area covered by dark area. When the spot diameter is about right, the area covered by dark edges of probe at one position is corrected multiple times at other scan positions with better illumination. But this correction mechanism fails when the spot is too small. Moreover, as the energy of the revised exit wave is defined by the diffraction patterns, a probe with all energy concentrating on a small area gives a dark 'hole' on the updated object. This makes the reconstruction like 'punching-holes' onto the guessed object. As the two main constraints in ptychography are considerably suppressed in this scenario, a collapsed probe is hard to recover and causes stagnation eventually.

The collapsed probe is mainly caused by a poor phase structure in the reciprocal space, as its modulus in f-domain is usually in good shape thanks to the f-constraint. The phase of a complex matrix works as a 'frame' in the reciprocal space. A well-reconstructed phase in f-domain increases the spatial frequency content to the data, hence the entire matrix does not collapse during the inverse Fourier transformation.

Stagnation caused by a collapsed probe is solved in two steps: identification and intervention. In the Chapter 6, a method of estimating probe diameter from diffraction patterns and scan grid has been explained. That is considered as a priory condition for detecting a collapsing probe. The calling map concept is also helpful for observing how the current probe relates to itself. As step size should be no larger than 30% of the spot size[59], overlapping the origin of calling map at the centre of probe (e.g. *Figure 7. 1(d)*) illustrates whether the probe is effectively updated. A rule of thumb is at least 4 groups of dots stays inside the probe spot. Once the probe spot barely touches adjacent calling map groups, interference should be introduced to prevent it from further collapsing.

Since the cause of a collapsing probe is poor phase profile in the f-domain, one recommended solution is modifying or replacing its phase with a new one, which can provide a probe with spot size close to the desired diameter.

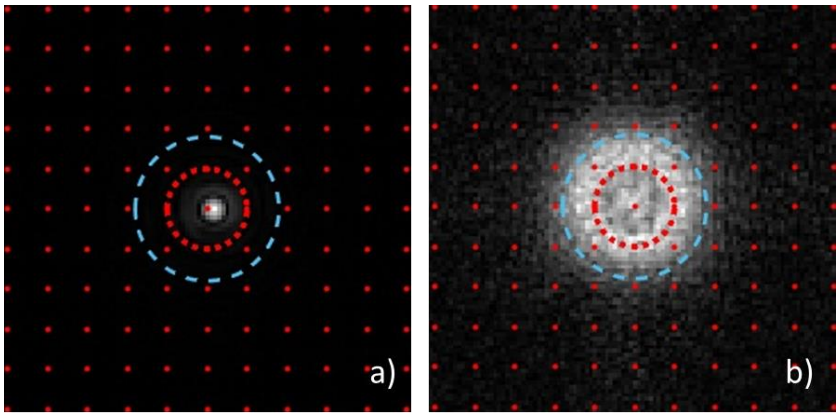


Figure 7. 3. An example of collapsed probe (a) and after 'smashed' (b). The outer blue circle indicates the estimated probe diameter, while the radius of inner red circle is step size. The collapsed probe(a) is significantly smaller than the desired diameter. Its radius is even smaller than the step size. Such a guessed probe cannot form effective overlapping area during reconstruction and cause stagnation. After replacing its f-domain phase with the initial one, the spot size matches the estimated probe size again (b).

7.4. Object hot pixel limit (constraint, result observing)

During the iteration, some pixels on the object become significantly brighter than others. They do not only make the modulus image lose contrast but also introduce instability to the iteration. This problem is solved by setting a limit on the maximum values of modulus. Theoretically, a non-illuminating object should not have modulus larger than 1. However, some algorithms prefer a tolerance to perform reflection more effectively. Therefore, a slightly higher limits, e.g. 2, is an overall better choice during the iterations.

7.5. Energy confinement (constraint)

As explained previously, scaling factor is an inherent ambiguity for ptychography and cannot be completely removed by f- or s- constraint. Although, as an ambiguity, the scaling factor should have no influence on the reconstruction, it sometimes accumulates during the iteration and eventually causes the values go beyond the data precision. Adjusting the energy

of probe can stop the positive feedback in scaling factor effectively. The energy of probe is adapted to the energy of brightest diffraction pattern at the end of each iteration. This fixed probe energy can effectively prevent the probe diverges from its correct energy level. One should notice that even the brightest diffraction pattern has less energy than the true probe due to the specimen absorption and scattering. But this energy confinement constraint still applies for most of the cases and one can scale the energy level based on the experiment scenario to give a better estimation on the true probe energy.

7.6. Blind recentre (constraint)

Besides the scaling factor, the global shifting is also a common ambiguity that can accumulate with iteration and leads to fail reconstruction without proper handling. For instance, if the probe continuously drifts during the reconstruction, its integrity will break when some parts of it go beyond the boundary. This could ruin the reconstructed object and lead to an instable reconstruction. Therefore, preventing the guessed probe from continuously drifting is crucial for a stable reconstruction. This “blind recentre” constraint is made to limit the guessed probe in a certain area during the reconstruction.

Without available true probe, a good approximation for finding the centre of probe is taking its ‘centre of mass’ as the centre. The ‘centre of mass’ is the position that provides similar summation value on its both sides. By applying this concept along the row and column direction respectively, the centre of reconstructed probe is found. Hence the object and probe can be shifted to the centre of its range. One should notice the global shifts must be applied to the probe and object (and other related variables, e.g. the exit waves) simultaneously to prevent disturbing the reconstruction. The target of this function is not locating the guessed probe to the same location as the true probe but preventing it from continuously drifting during the reconstruction. Since the guessed probe could change dramatically in the beginning of reconstruction, it is recommended to activate the ‘blind recentre’ after tens of iterations to let the probe structure settle down.

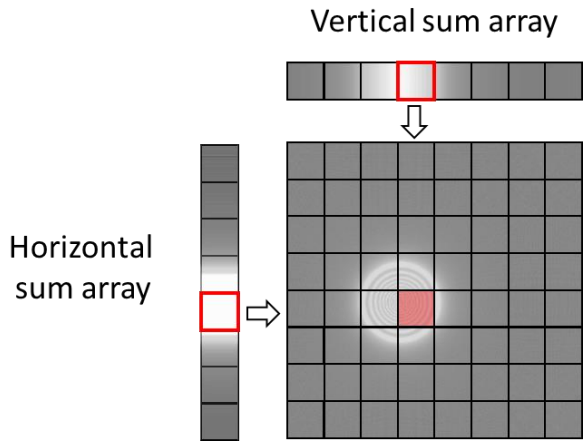


Figure 7. 4. A demonstration of estimating the centre of a probe by 'centre of mass' method. Summing the probe along the horizontal and vertical direction directions respectively. Then the centre of two 1-D array is estimated by finding the pixel that has about the same summation on its both sides. The mass centre is highlighted by re colour in this example.

To test this constraint under the influence of 'not spatially well confined' probes, two scenarios are simulated: one with highly diffused probe and one with double spots. These two types of probes are demonstrated in *Fig 7.5* together with the corresponding initial guessed probes. Diffraction patterns are produced with these probes and applied for reconstruction. The reconstructed probe is artificially shifted at the 200th iteration. The reconstruction results are shown in *Fig 7.6*. As shown in the figure, the 'not spatially confined' probes do not make the 'blind recentre' instable.

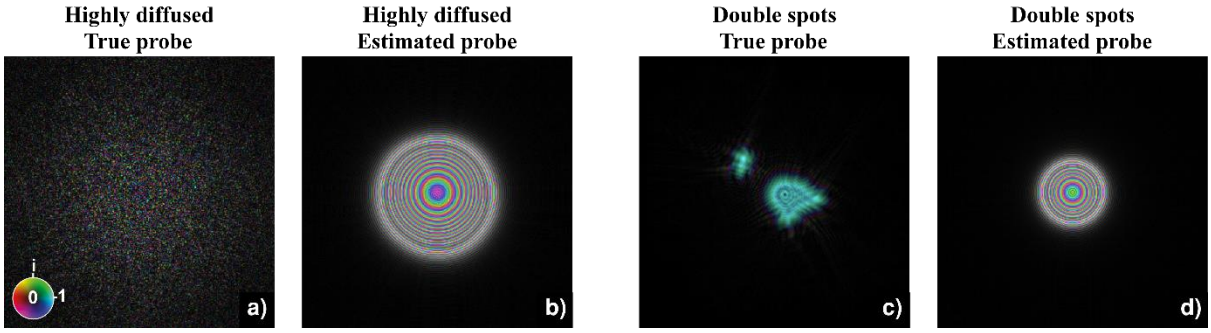


Figure 7. 5. Not spatially well confined probes for testing "blind recentre" constraint. (a) shows a probe, whose structure is significantly affected by a diffuser. (b) is the initial guessed probe for reconstruction. (c) is a probe having two spots on it, and (d) is the corresponding guessed probe. All these probes are plotted in colour wheel format.

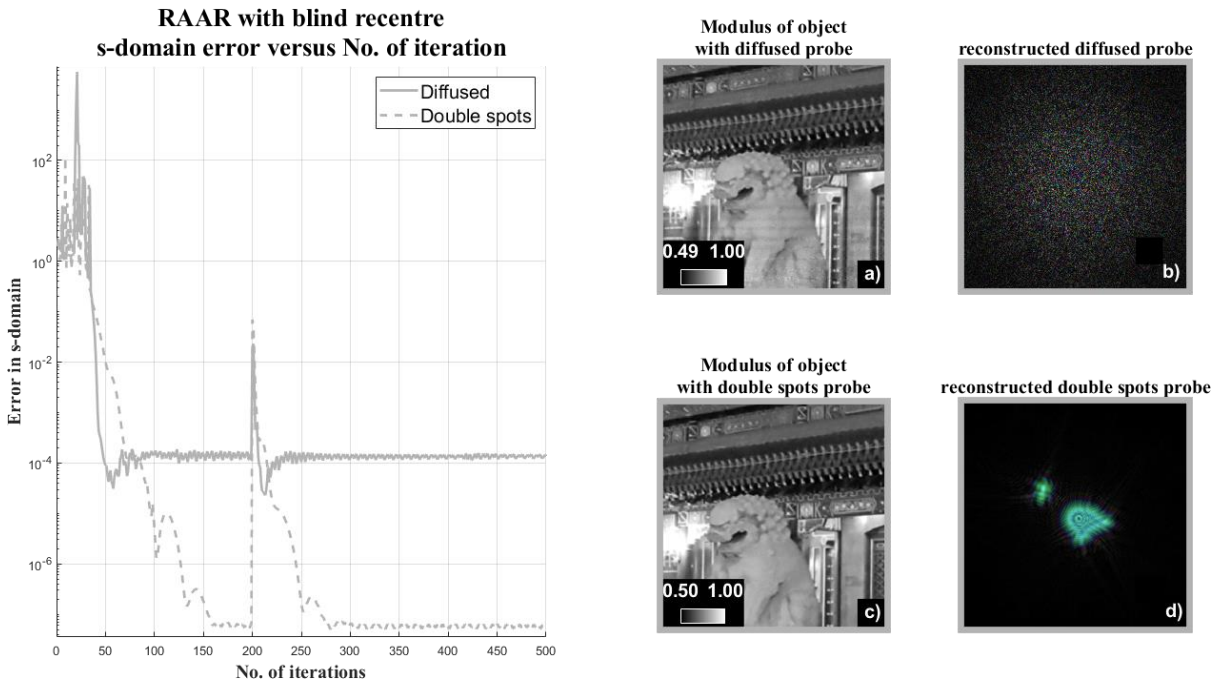


Figure 7. 6. As shown in the figure, decent reconstructions are obtained in both scenarios. The error spike caused by the artificially shifting the probe at the 200th iteration is quickly levelled out as the reconstruction progress. Eventually the error returns to the same level as it was before shifting the probe.

8. Conclusion

The development of lens-less imaging removes the limitation of lens quality from the traditional imaging system and brings fascinating potentials to various microscopy technologies. All these benefits are based on solving the phase problem, which is caused by losing phase information during recording diffraction intensities. Solving the phase problem requires more information than a single diffraction pattern provides; other constraints are required to fully confine the solution. Ptychography, as one of the competitors, forms a consistency constraint by scanning the specimen with a defocused probe at overlapping positions. With a sufficiently large overlapping area, this consistency constraint contains sufficient information to not only recover an image of the specimen, but also the illumination function. Many phase retrieving algorithms have been developed that take advantage of the rich data provided by ptychography to provide a robust and promising reconstruction. These algorithms are based on different concepts and perform differently with the same collected data set.

This thesis started from a study on the existed algorithms developed for ptychography. Several widely utilised algorithms and their variants are separated into two categories, their operation explained, and their implementation detailed using pseudocode. Besides representing the original form of these algorithms, some modifications are also suggested to maximise their efficiency and reduce their memory footprint. For the momentum PIE (mPIE) algorithm, the required parameters are reduced from 7 to 3, which makes its performance less dependent on user experience while maintaining an excellent converging speed. Its momentum updating functions are also moved to the end of a complete iteration for better computing speed and stability. For alternating direction method of multipliers (ADMM), a re-arrangement of the computation order under two assumptions significantly reduces its memory footprint without significantly affecting its results. The hybrid projection and reflection (HPR), which was developed for support constraint previously, is also translated to the ptychography environment and tested.

Once a reconstructed image is obtained, the next task is to evaluate its quality. Although an error metric comparing reconstructed and measured diffraction pattern intensities offers a straightforward route, this is not a good way to compare the performance of different algorithms in simulations, since it does not directly measure the accuracy of the image against

the known ground truth. In ptychography, there are three kinds of inherent ambiguities appearing in pairs on the reconstructed object and probe. They cannot be suppressed by the constraints, but significantly distort the appearance of the images. In this thesis, a working flow of estimating these ambiguities, removing them and computing an accurate s-domain error is explained. This improves on previous work by maximising efficiency and robustness.

With the understanding of these existed algorithms, a new algorithm is developed, which is named as adaptive PIE (adaPIE). This algorithm is inspired by methods used in the training of neural networks. By introducing an adaptive regularisation term to the cost function, the local minima in the searching space is flattened. This does not only reduce the chance of getting stagnated in the beginning, but also prevents ill-posed pixels fluctuating significantly during reconstruction. Both its converging speed and reconstruction quality outperforms other algorithms in simulated tests.

To test these algorithms within a practical scenario, several data sets are collected by observing a bilayer of Molybdenum Disulphide (MoS2) under a scanning transmission electron microscope (STEM). A series of manipulations on the collected data is explained to prevent fruitless reconstruction due to unmatched scan position and measured intensities. The test results indicate none of these algorithms has absolute advantages, though ePIE, ADMM and RAAR demonstrates a strong robustness for noisy data. On the other hand, the adaPIE does not show considerable advantages in these tests, which indicates further work is required to refine the approach and improve its robustness to the noise.

Other constraints and tricks that are developed during the research are listed in the last chapter. The ‘probe calling map’ can visualise how a pixel of probe is related to itself by a given scanning grid, while other constraints are developed to prevent stagnation or accumulation of ambiguities.

Many other ideas are still under development. First, the adaPIE demonstrates a strong potential by reconstructing the simulated data, though it is not the case for the practical data set. Such an unmatched performance indicates the impact of noise on the adaptive regulation map needs to be considered more seriously. We expect to bring the advantage of adaPIE in simulated data to a more practical scenario by adjusting its regulation components.

Another further research direction is mediating the updating strategies used by PIE and batch algorithms. The ‘one- by-one’ updating strategy used by PIE has less memory footprint but leads to the final reconstruction fluctuating around a solution. On the other hand, the batch

algorithms prevent the fluctuation by averaging the variation on all revised exit waves. The concept, named as ‘mini-batches’, is attempted to mediate their performance by separating the exit waves into several smaller groups, then updates the object and probe group by group. Some efforts have been made on this topic, but they give no advantages so far. A deeper look into the objective function is required for this algorithm.

As a help for researches interested in this topic, all the codes for algorithms and other useful tools for simulation and evaluation are available for requires.

Reference

1. Rodenburg, J. M. New microscopic-imaging method delivers novel capabilities. *SPIE Newsroom* 2–4 (2011) doi:10.1117/2.1201012.003414.
2. Pozzi, G. Fourier Optics. in *Advances in Imaging and Electron Physics* (2016). doi:10.1016/bs.aiep.2016.02.007.
3. Singer, C. Notes on the Early History of Microscopy. *Proc. R. Soc. Med.* (1914) doi:10.1177/003591571400701617.
4. Thibault, P. High-resolution scanning x-ray diffraction microscopy. *Science (80-.).* **379**, 379–383 (2008).
5. Mualla, F., Aubreville, M. & Maier, A. Microscopy. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2018). doi:10.1007/978-3-319-96520-8_5.
6. Physical principles of electron microscopy. *Mater. Today* (2005) doi:10.1016/s1369-7021(05)71290-6.
7. Mertz, J. *Introduction to Optical Microscopy*. *Introduction to Optical Microscopy* (2019). doi:10.1017/9781108552660.
8. Chen, C. J. *Introduction to Scanning Tunneling Microscopy: Second Edition*. *Introduction to Scanning Tunneling Microscopy: Second Edition* (2007). doi:10.1093/acprof:oso/9780199211500.001.0001.
9. Pulizzi, F. Electron and X-ray microscopy. *Nature Materials* (2009) doi:10.1038/nmat2424.
10. Jacobsen, C. *X-ray microscopy*. *X-ray Microscopy* (2019). doi:10.1017/9781139924542.
11. Cosslett, V. E. X-ray microscopy and microanalysis. *Metall. Rev.* (1960) doi:10.1179/mtlr.1960.5.1.225.
12. Pfeiffer, F. X-ray ptychography. *Nat. Photonics* (2018) doi:10.1038/s41566-017-0072-5.
13. Ozcan, A. & McLeod, E. Lensless Imaging and Sensing. *Annual Review of Biomedical Engineering* (2016) doi:10.1146/annurev-bioeng-092515-010849.
14. Wu, Y. & Ozcan, A. Lensless digital holographic microscopy and its applications in biomedicine and environmental monitoring. *Methods* (2018) doi:10.1016/j.ymeth.2017.08.013.
15. Ersoy, O. K. *Diffraction, Fourier Optics and Imaging*. *Diffraction, Fourier Optics and*

Imaging (2006). doi:10.1002/0470085002.

16. Li, P. & Maiden, A. Multi-slice ptychographic tomography. *Sci. Rep.* (2018) doi:10.1038/s41598-018-20530-x.
17. Tian, L. & Waller, L. 3D intensity and phase imaging from light field measurements in an LED array microscope. *Optica* (2015) doi:10.1364/optica.2.000104.
18. Zhang, B. *et al.* High contrast 3D imaging of surfaces near the wavelength limit using tabletop EUV ptychography. *Ultramicroscopy* (2015) doi:10.1016/j.ultramic.2015.07.006.
19. Humphry, M. J., Kraus, B., Hurst, A. C., Maiden, A. M. & Rodenburg, J. M. Ptychographic electron microscopy using high-angle dark-field scattering for sub-nanometre resolution imaging. *Nat. Commun.* **3**, 730–737 (2012).
20. Maiden, A. M. & Rodenburg, J. M. An improved ptychographical phase retrieval algorithm for diffractive imaging. *Ultramicroscopy* **109**, 1256–1262 (2009).
21. Marrison, J., Räty, L., Marriott, P. & O'Toole, P. Ptychography-a label free, high-contrast imaging technique for live cells using quantitative phase information. *Sci. Rep.* (2013) doi:10.1038/srep02369.
22. Li, P. & Maiden, A. M. Ten implementations of ptychography. *J. Microsc.* (2018) doi:10.1111/jmi.12614.
23. Jiang, Y. *et al.* Electron ptychography of 2D materials to deep sub-ångström resolution. *Nature* (2018) doi:10.1038/s41586-018-0298-5.
24. Gerchberg, R. W. & Saxton, W. O. PRACTICAL ALGORITHM FOR THE DETERMINATION OF PHASE FROM IMAGE AND DIFFRACTION PLANE PICTURES. *Opt.* (1972).
25. Dainty, J. C. & Fienup, J. R. Phase Retrieval and Image Reconstruction for Astronomy. *Image Recover. theory Appl.* (1987).
26. Bertero, M. & Boccacci, P. *Introduction to Inverse Problems in Imaging*. *Introduction to Inverse Problems in Imaging* (1998). doi:10.1887/0750304359.
27. Groetsch, C. Introduction to inverse problems. *Inverse Probl.* 1–24 (1999) doi:10.1090/clrm/012/01.
28. Maiden, A. M., Sarahan, M. C., Stagg, M. D., Schramm, S. M. & Humphry, M. J. Quantitative electron phase imaging with high sensitivity and an unlimited field of view. *Sci. Rep.* (2015) doi:10.1038/srep14690.
29. Elser, V. Phase retrieval by iterated projections. **20**, 40–55 (2001).

30. Stephens, D. J. & Allan, V. J. Light microscopy techniques for live cell imaging. *Science* (2003) doi:10.1126/science.1082160.

31. Dailey, M. E., Manders, E., Soll, D. R. & Terasaki, M. Confocal microscopy of living cells. in *Handbook of Biological Confocal Microscopy: Third Edition* (2006). doi:10.1007/978-0-387-45524-2_19.

32. Andersson, H., Baechi, T., Hoechl, M. & Richter, C. Autofluorescence of living cells. *J. Microsc.* (1998) doi:10.1046/j.1365-2818.1998.00347.x.

33. Huang, B., Bates, M. & Zhuang, X. Super-resolution fluorescence microscopy. *Annual Review of Biochemistry* (2009) doi:10.1146/annurev.biochem.77.061906.092014.

34. Park, Y. K., Depeursinge, C. & Popescu, G. Quantitative phase imaging in biomedicine. *Nature Photonics* (2018) doi:10.1038/s41566-018-0253-x.

35. Mir, M., Bhaduri, B., Wang, R., Zhu, R. & Popescu, G. Quantitative Phase Imaging. in *Progress in Optics* (2012). doi:10.1016/B978-0-44-459422-8.00003-5.

36. Fitzgerald, R. Phase-Sensitive X-ray Imaging. *Phys. Today* (2000) doi:10.1063/1.1292471.

37. Holler, M. *et al.* High-resolution non-destructive three-dimensional imaging of integrated circuits. *Nature* (2017) doi:10.1038/nature21698.

38. Guoan, Z., Cheng, S., Shaowei, J., Pengming, S. & Changhuei, Y. Concept, implementations and applications of Fourier ptychography. *Nat. Rev. Phys.* **0123456789**, (2021).

39. Voelz, D. G. *Computational Fourier Optics: A MATLAB Tutorial*. *Computational Fourier Optics: A MATLAB Tutorial* (2011). doi:10.1117/3.858456.

40. Fienup, J. R. Phase retrieval algorithms: a comparison. *Appl. Opt.* **21**, 2758–2769 (1982).

41. Fienup, J. R. Reconstruction of an object from the modulus of its Fourier transform. *Opt. Lett.* **3**, 27–29 (1978).

42. Fienup, J. R. Phase retrieval with continuous version of hybrid input-output. in (2014). doi:10.1364/fio.2003.thi3.

43. Marchesini, S. *et al.* X-ray image reconstruction from a diffraction pattern alone. *Phys. Rev. B - Condens. Matter Mater. Phys.* **68**, 1–4 (2003).

44. Konijnenberg, A. P., Coene, W. M. J., Pereira, S. F. & Urbach, H. P. Ptychographic phase retrieval by applying hybrid input-output (HIO) iterations sequentially. *Digit. Opt. Technol. 2017* **10335**, 103351I (2017).

45. Guizar-Sicairos, M. & Fienup, J. R. Understanding the twin-image problem in phase retrieval. *J. Opt. Soc. Am. A* (2012) doi:10.1364/josaa.29.002367.
46. Elser, V., Rankenburg, I. & Thibault, P. Searching with iterated maps. *Proc. Natl. Acad. Sci. U. S. A.* (2007) doi:10.1073/pnas.0606359104.
47. Hawkes, P. & Spence, J. *Springer Handbook of Microscopy*. *Neuroanatomy* (2019).
48. Maiden, A. M., Humphry, M. J., Sarahan, M. C., Kraus, B. & Rodenburg, J. M. An annealing algorithm to correct positioning errors in ptychography. *Ultramicroscopy* (2012) doi:10.1016/j.ultramic.2012.06.001.
49. Maiden, A. M., Humphry, M. J., Zhang, F. & Rodenburg, J. M. Superresolution imaging via ptychography. *J. Opt. Soc. Am. A* (2011) doi:10.1364/josaa.28.000604.
50. Tripathi, A., McNulty, I. & Shpyrko, O. G. Ptychographic overlap constraint errors and the limits of their numerical recovery using conjugate gradient descent methods. *Opt. Express* (2014) doi:10.1364/oe.22.001452.
51. Jacobsen, C. Relaxation of the Crowther criterion in multislice tomography. *Opt. Lett.* **43**, 4811 (2018).
52. Godard, P., Allain, M., Chamard, V. & Rodenburg, J. Noise models for low counting rate coherent diffraction imaging. *Opt. Express* (2012) doi:10.1364/oe.20.025914.
53. Bauschke, H. H., Combettes, P. L. & Luke, D. R. Hybrid projection–reflection method for phase retrieval. *J. Opt. Soc. Am. A* **20**, 1025 (2003).
54. Marchesini, S. A unified evaluation of iterative projection algorithms for phase retrieval. *Rev. Sci. Instrum.* (2007) doi:10.1063/1.2403783.
55. Chang, H., Enfedaque, P. & Marchesini, S. Blind ptychographic phase retrieval via convergent alternating direction method of multipliers. *SIAM J. Imaging Sci.* **12**, 153–185 (2019).
56. Maiden, A., Johnson, D. & Li, P. Further improvements to the ptychographical iterative engine. *Optica* **4**, 736 (2017).
57. Guizar-Sicairos, M. & Fienup, J. R. Phase retrieval with transverse translation diversity: a nonlinear optimization approach. *Opt. Express* **16**, 7264 (2008).
58. Ruder, S. An overview of gradient descent optimization algorithms. 1–14 (2016).
59. Maiden, A. M. & Rodenburg, J. M. An improved ptychographical phase retrieval algorithm for diffractive imaging. *Ultramicroscopy* **109**, 1256–1262 (2009).
60. Konijnenberg, A. P., Coene, W. M. J., Pereira, S. F. & Urbach, H. P. Combining

ptychographical algorithms with the Hybrid Input-Output (HIO) algorithm. *Ultramicroscopy* **171**, 43–54 (2016).

61. Marchesini, S. A unified evaluation of iterative projection algorithms for phase retrieval. *Rev. Sci. Instrum.* **78**, (2007).
62. Rodenburg, J. M. & Faulkner, H. M. L. A phase retrieval algorithm for shifting illumination. *Appl. Phys. Lett.* **85**, 4795–4797 (2004).
63. Murphy, K. P. *Machine learning: a probabilistic perspective (adaptive computation and machine learning series)*. *Mit Press. ISBN* (2012).
64. Kukačka, J., Golkov, V. & Cremers, D. Regularization for deep learning: A taxonomy. *arXiv* (2017).
65. Enfedaque, P., Chang, H., Krishnan, H. & Marchesini, S. GPU-based implementation of ptycho-ADMM for high performance x-ray imaging. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2018). doi:10.1007/978-3-319-93698-7_41.
66. Bauschke, H. H., Combettes, P. L. & Luke, D. R. Hybrid projection–reflection method for phase retrieval. *J. Opt. Soc. Am. A* **20**, 1025 (2003).
67. Yan, H. Ptychographic phase retrieval by proximal algorithms. *New J. Phys.* (2020) doi:10.1088/1367-2630/ab704e.
68. Wu, M. hong *et al.* Molybdenum disulfide (MoS2) as a co-catalyst for photocatalytic degradation of organic contaminants: A review. *Process Saf. Environ. Prot.* **118**, 40–58 (2018).
69. Huang, Y. *et al.* Ptychography-based high-throughput lensless on-chip microscopy via incremental proximal algorithms. *Opt. Express* **29**, 37892 (2021).
70. Hesse, R., Luke, D. R., Sabach, S. & Tam, M. K. Proximal Heterogeneous Block Input-Output Method and application to Blind Ptychographic Diffraction Imaging. 1–32 (2014) doi:10.1137/14098168X.
71. Bertsekas, D. P. Incremental Gradient, Subgradient, and Proximal Methods for Convex Optimization: A Survey. *Optim. Mach. Learn.* **2010**, (2019).
72. Thibault, P. & Guizar-Sicairos, M. Maximum-likelihood refinement for coherent diffractive imaging. *New J. Phys.* **14**, (2012).
73. Horstmeyer, R. *et al.* Solving ptychography with a convex relaxation. *New J. Phys.* (2015) doi:10.1088/1367-2630/17/5/053044.

74. Bostan, E., Soltanolkotabi, M., Ren, D. & Waller, L. Accelerated Wirtinger Flow for Multiplexed Fourier Ptychographic Microscopy. in *Proceedings - International Conference on Image Processing, ICIP* (2018). doi:10.1109/ICIP.2018.8451437.

75. Odstrčil, M., Menzel, A. & Guizar-Sicairos, M. Iterative least-squares solver for generalized maximum-likelihood ptychography. *Opt. Express* (2018) doi:10.1364/oe.26.003108.

76. Marchesini, S. *et al.* SHARP: A distributed GPU-based ptychographic solver. *J. Appl. Crystallogr.* (2016) doi:10.1107/S1600576716008074.

77. Guizar-sicairos, M., Thurman, S. T. & Fienup, J. R. Efficient subpixel image registration algorithms. **33**, 156–158 (2008).

78. Zhang, T. *et al.* Rapid and robust two-dimensional phase unwrapping via deep learning. *Opt. Express* **27**, 23173 (2019).

79. Navarro, M. A., Estrada, J. C., Servin, M., Quiroga, J. A. & Vargas, J. Fast two-dimensional simultaneous phase unwrapping and low-pass filtering. *Opt. Express* **20**, 2556 (2012).

80. PIJEWSKA, E., GORCZYNSKA, I. & SZKULMOWSKI, M. Computationally effective 2D and 3D fast phase unwrapping algorithms and their applications to Doppler optical coherence tomography. *Biomed. express* **10**, 1365–1382 (2019).

81. Luke, D. R. Relaxed averaged alternating reflections for diffraction imaging. *Inverse Probl.* **21**, 37–50 (2005).

82. Marchesini, S. A unified evaluation of iterative projection algorithms for phase retrieval. *Rev. Sci. Instrum.* **78**, 1–12 (2007).

83. Giewekemeyer, K. *et al.* Ptychographic coherent x-ray diffractive imaging in the water window. *Opt. Express* (2011) doi:10.1364/OE.19.001037.

84. Qian, J., Yang, C., Schirotzek, A., Maia, F. & Marchesini, S. Efficient Algorithms for Ptychographic Phase Retrieval. **0**, 261–279 (2014).

85. Yang, C., Qian, J., Schirotzek, A., Maia, F. & Marchesini, S. Iterative Algorithms for Ptychographic Phase Retrieval. 630–632 (2011).

86. Nocedal, J. & Wright, S. J. Numerical optimization. in *Springer Series in Operations Research and Financial Engineering* (2006). doi:10.1201/b19115-11.

87. Fannjiang, A. Raster grid pathology and the cure. *Multiscale Model. Simul.* (2019) doi:10.1137/18M1227354.

88. Odstrčil, M., Lebugle, M., Guizar-Sicairos, M., David, C. & Holler, M. Towards optimized

illumination for high-resolution ptychography. *Opt. Express* (2019) doi:10.1364/oe.27.014981.