Algorithms for iterative phase recovery for photoemission tomography data

BACHELOR THESIS

by

Daniel Khodachenko

written at the Department of Physics
University of Graz

under the supervision of
Assoc.-Prof. Dr. Peter Puschnig

Graz, August 2020

Abstract

Photoemission tomography is a powerful new tool in science, which allows the reconstruction of electron orbitals from experimental photoemission data, through the use of iterative algorithms. It is based on Fermi's golden rule for a photoemission process, and the use of a plane wave approximation for the final state. With this assumption the experimental data of angle-resolved photoemission spectroscopy (ARPES) becomes proportional to the absolute value Fourier transform (FT) of the initial state. At the same time, the information on the phase of the wave function is lost in the process. However, with the use of the iterative algorithms discussed in this thesis, this phase can be recovered.

This work makes use of the ARPES measurement data of complex molecules like perylene-3,4,9,10-
tetracarboxylic dianhydride (PTCDA) and pentacene (5A), and reconstructs their electron orbitals using two different iterative methods, the Gerchberg-Saxton (GS) algorithm and a combination of error reduction (ER) and phase-constrained hybrid input-output (PC-HIO) algorithms. The latter algorithm was applied to this ARPES measurement data for the first time.

Using these methods, the real space and Fourier space images of the electron orbitals of the above mentioned molecules could be reconstructed. Thereby, advantages and disadvantages of the various methods have been compared and assessed for future applications.

Kurzzusammenfassung

Photoemissions-Tomographie ist ein mächtiges neues Werkzeug in der Oberflächenphysik, welches die Rekonstruktion von Elektronenorbitalen aus experimentellen Messdaten, durch die Verwendung von iterativen Algorithmen, ermöglicht. Sie basiert auf Fermis goldener Regel für den Photoemissionsprozess und einer Näherung des Endzustands durch eine ebene Welle. Unter dieser Annahme kann gezeigt werden, dass die experimentellen Messdaten einer winkelaufgelösten Photoelektronenspektroskopie (ARPES) zum Absolutbetrag der Fouriertransformation (FT) vom Anfangszustand proportional werden. Da es sich um den Absolutbetrag der FT handelt, geht die Phase der Wellenfunktion verloren. Mithilfe von iterativen Algorithmen kann diese jedoch rekonstruiert werden.

Diese Arbeit verwendet die ARPES Messdaten von komplexen Molekülen wie zum Beispiel 3,4,9,10-Perylentetracarbonsuredianhydrid (PTCDA) und Pentacen (5A), und rekonstruiert ihre Elektronenorbitale mit zwei unterschiedlichen iterativen Methoden, dem Gerchberg-Saxton (GS) Algorithmus und einer Kombination aus dem *error reduction* (ER) Algorithmus und dem *phase-constrained hybrid input-output* (PC-HIO) Algorithmus. Die zweite Methode wurde zum ersten mal auf diese APRES Messdaten angewendet.

Mit diesen Methoden knnen die Elektronenorbitale dieser Moleküle im Realraum und im Fourierraum rekonstruiert und abgebildet werden. Anschließend wurden Vor- und Nachteile der unterschiedlichen Methoden für zukünftige Anwendungen verglichen und beurteilt.

Contents

List of Abbreviations

5A	pentacene
ARPES	angle-resolved photoemission spectroscopy
DFT	density functional theory
ER	error reduction (algorithm)
FFT	fast Fourier transform
FT	Fourier transform
GS	Gerchberg-Saxton (algorithm)
HIO	hybrid input-output (algorithm)
HOMO	highest occupied molecular orbital
LUMO	lowest unoccupied molecular orbital
PC-HIO	phase-constrained hybrid input-output (algorithm)
PTCDA	perylene-3,4,9,10-tetracarboxylic dianhydride

1 Introduction

In quantum mechanics the spatial probability distributions of electrons in atoms are called orbitals. Atomic orbitals used to be a concept of theoretical physics, with no way to create images from experimental data. However, in the past couple of decades scientists have been able to make use of different methods to reconstruct orbitals from experimental observation, ranging from X-ray and electron diffraction data [1] to scanning tunneling microscopy, as well as electron momentum spectroscopy and photoemission spectroscopy [2]. This work mainly focuses on the method of angle-resolved photoemission spectroscopy (ARPES).

In 1887 Hertz discovered, that electrons can be emitted from a sample when electromagnetic radiation hits the surface. This effect was later explained by Einstein in 1905, who proposed the concept of wave-particle duality. In the case of the photoelectric effect this meant, that light itself is also part particle, quantized into discrete wave packets called photons. When these photons have the appropriate energy, they are able to kick out the electrons in the sample.

ARPES is based on the photoelectric effect and is the most direct technique to determine the electronic band structure of a surface. The probability for the electron to be emitted can be calculated from Fermi's golden rule of perturbation theory. Furthermore, by assuming that the photoemitted electron is no longer influenced by the potential of the sample, the final state can be approximated with a plane wave. With this approach the ARPES measurement data becomes proportional to the absolute value Fourier transform (FT) of the initial state, in this case the electron orbitals. One might assume, that a simple inverse FT on the measured data would be sufficient to reconstruct the orbitals, but the problem lies in the fact, that it is the absolute value of said FT, which results in the phase of the electron wave function to be lost.

This anomaly is called the phase problem, which will be described in more detail in Chapter 2.4. In fact, an inverse FT of the ARPES measurement data produces a real-space wave function with twice the spatial extent and wrong phase. To correct this, we introduced two different iterative reconstruction methods, the Gerchberg-Saxton (GS) algorithm and a combination of error reduction (ER) and phase-constrained hybrid input-output (PC-HIO) algorithms. Both of these methods require a known spatial extent of the molecules in real space, called the domain constraint.

With the GS algorithm, the value outside of the domain constraint is reduced by a constant factor of its initial value with each iteration, therefore reducing the error at each step and slowly iterating to the desired real-space wave function. This idea is also used in the second method with the ER algorithm, which sets the values outside of the spatial extent to zero with each iteration. This alone would not converge to a useful result, which is why we have to combine it with the PC-HIO algorithm. This algorithm

in a way recycles the wave function and calculates the difference between itself outside of the domain constraint in each iteration.

With these methods it is possible to reconstruct the electron orbitals of complex molecules. With the use of the ARPES measurement data of perylene-3,4,9,10-tetracarboxylic dianhydride (PTCDA) and pentacene (5A), this work is dedicated to the software implementation of these phase recovery algorithms, as well as their comparison in performance. The results are depicted and described in more details in the Chapters 4.2 and 4.3.

2 Theory

2.1 Angle-resolved photoemission spectroscopy (ARPES)

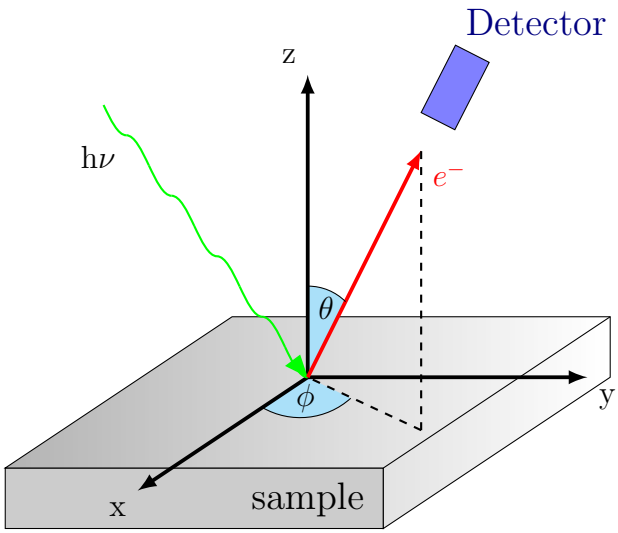

Figure 1: Schematic layout of the ARPES experiment: From the monochromatic light source a photon with energy $h\nu$ enters the sample and kicks out a photoelectron. The kinetic energy of the electron is measured as a function of the angles θ and ϕ.

Angle resolved photoemission spectroscopy (ARPES) is the most direct experimental technique to determine the electronic (band) structure of a surface. As the name suggests, the angles, at which photoelectrons are emitted, are of much importance in this kind of setup.

ARPES is based on the photoelectric effect, that was discovered in 1887 by Hertz, and later explained by Einstein in 1905, for which he famously received the Noble prize. Einstein introduced the wave-particle duality of light, and therefore the concept of a photon. The usual ARPES experiment uses monochromatic UV light as a source, so that the energy of those photons $h\nu$ is constant. The kinetic energies E_{kin} of the photoelectrons are measured as a function of their angle (see Fig. 1), which, by conservation of energy, is given by [3]:

$$E_{kin} = h\nu - \Phi - E_B. \tag{1}$$

Here Φ is the work function, which is given by the energy difference between vacuum level and Fermi energy $\Phi = E_{vac} - E_F$, and E_B is the electrons binding energy with respect to the Fermi level, which is characteristic for a given electronic state [4]. Under the assumption, that the emitted electron is no longer influenced by the potential of

its solid, the final state of the electron can be approximated as a free particle with the energy:

$$E_{kin} = \frac{\hbar^2 (k^{ex})^2}{2m_e} \tag{2}$$

Here k^{ex} is the wave number of the emitted photoelectron, and m_e is its mass. From Eq. (2) it follows that

$$k^{ex} = \frac{\sqrt{2m_e E_{kin}}}{\hbar}. \tag{3}$$

The x and y components of the wave vector can now be expressed using the measured angles θ and ϕ like [3, 5]

$$k_x^{ex} = \frac{\sqrt{2m_e E_{kin}}}{\hbar} \sin\theta \, \cos\phi \tag{4}$$

$$k_y^{ex} = \frac{\sqrt{2m_e E_{kin}}}{\hbar} \sin\theta \, \sin\phi. \tag{5}$$

Due to the geometry of the ARPES experiment, the electrons wave vector component parallel to the surface before and after its exit from the sample is conserved (Fig. 2), and can be described using the reciprocal lattice vector $\mathbf{G}$ like [4]

$$\mathbf{k}_{||}^{\mathbf{ex}} = \mathbf{k}_{||} + \mathbf{G}_{||}. \tag{6}$$

As $\mathbf{k}_{||}^{\mathbf{ex}}$ is known by measurement, $\mathbf{k}_{||}$ can also be calculated using this equation.

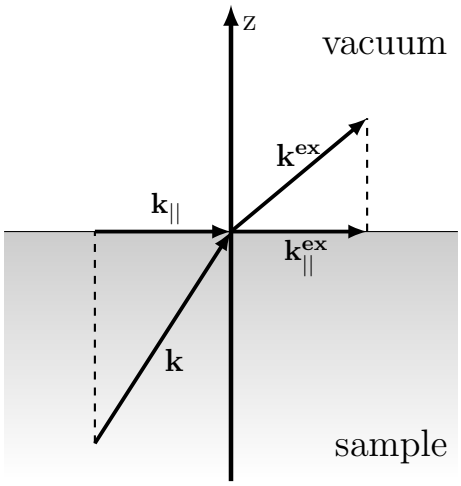

Figure 2: Geometry of the ARPES experiment: Parallel component of the wave vector is conserved when exiting the surface.

2.2 Quantum Mechanics of the photoemission process

The photoemission process can be explained using either the three-step model, or the one-step model. The three-step model divides the process into optical excitation of the electron from an initial to a final state by absorption of a photon, travel of the photoelectron towards the surface, and transmission of the electron into the vacuum.

The one-step model, on the other hand, describes the photoemission in a single step and makes use of Fermi's golden rule from perturbation theory, which describes the transition rate from an initial state ψ_i to a final state ψ_f under a certain perturbation, in this case the absorption of a photon [3, 4]. Thus, the photoemission intensity I is proportional to the transition rate:

$$I \propto |\langle \psi_f | H_{int} | \psi_i \rangle|^2 \delta(E_{kin} + \Phi - E_B - h\nu). \tag{7}$$

Here, the δ function accounts for energy conservation, as stated in Eq. (1), and H_{int} is the interacting or perturbing Hamiltonian. The unperturbed Hamiltonian H_0 has the form:

$$H_0 = \frac{\mathbf{p}^2}{2m_e} + eV(\mathbf{r}), \tag{8}$$

where $V(\mathbf{r})$ stands for the potential. For a particle with charge e, in presence of an electromagnetic field, the momentum operator must be replaced by $\mathbf{p} \to \mathbf{p} - \frac{e}{c}\mathbf{A}$, known as the minimal coupling principle, and the Hamiltonian becomes

$$\begin{aligned} H &= \frac{1}{2m_e}[\mathbf{p} - \frac{e}{c}\mathbf{A}]^2 + eV(\mathbf{r}) = \\ &= \underbrace{\frac{\mathbf{p}^2}{2m_e} + eV(\mathbf{r})}_{H_0} + \underbrace{\frac{e}{2m_e c}(\mathbf{Ap} + \mathbf{pA}) + \frac{e}{2m_e c}\mathbf{A}^2}_{H_{int}} \\ &= H_0 + H_{int} \end{aligned} \tag{9}$$

Assuming the vector potential $\mathbf{A}$ is weak, one can neglect its quadratic term, i.e. using the approximation $\mathbf{A}^2 \approx 0$. The Eq. (9) can then be further simplified by employing Coulomb gauge $\nabla\mathbf{A} = 0$, and the commutator relation of momentum operator and vector potential $[\mathbf{p}, \mathbf{A}] = i\hbar\nabla\mathbf{A}$ which in this gauge becomes zero [3]. Now with the use of $\mathbf{Ap} + \mathbf{pA} = 2\mathbf{Ap}$ the second term in Eq. (9) can be rewritten, and the interaction Hamilton simplifies to

$$H_{int} = \frac{e}{m_e c}\mathbf{Ap}, \tag{10}$$

which can be inserted back into Eq. (7) to get the final result:

$$I \propto |\langle \psi_f | \mathbf{Ap} | \psi_i \rangle|^2 \delta(E_{kin} + \Phi - E_B - h\nu). \tag{11}$$

2.3 Plane wave approximation

Under the simplifying assumption that the photoemitted electron is no longer influenced by the potential produced from the solid, it makes sense to approximate the final state $|\psi_f\rangle$ with a plane wave of the form $|\psi_f\rangle \propto e^{i\mathbf{k_f}\mathbf{r}}$. Inserting this, and the momentum operator $\mathbf{p} = -i\hbar\nabla$, into Eq. (11) for a single photoelectron gives us [7]

$$I \propto |\langle e^{i\mathbf{k_f}\mathbf{r}}|\mathbf{A}(-i\hbar\nabla)|\psi_i\rangle|^2 = |i\hbar\langle\psi_i|\mathbf{A}\nabla|e^{i\mathbf{k_f}\mathbf{r}}\rangle|^2 = \left|i\hbar\int\psi_i^*(\mathbf{r})\mathbf{A}\nabla e^{i\mathbf{k_f}\mathbf{r}}\,\mathrm{d}r\right|^2. \tag{12}$$

Here $\psi_i(\mathbf{r})$ is the wave function in real space. Applying the derivative on $e^{i\mathbf{k_f}\mathbf{r}}$ and pulling the position-independent terms to the front leads to:

$$I \propto \left|i\hbar\int\psi_i^*(\mathbf{r})\mathbf{A}i\mathbf{k_f}e^{i\mathbf{k_f}\mathbf{r}}\,\mathrm{d}r\right|^2 = \hbar^2|\mathbf{Ak_f}|^2\left|\int\psi_i^*(\mathbf{r})e^{i\mathbf{k_f}\mathbf{r}}\,\mathrm{d}r\right|^2 \tag{13}$$

$$= \hbar^2|\mathbf{Ak_f}|^2\left|\int\psi_i(\mathbf{r})e^{-i\mathbf{k_f}\mathbf{r}}\,\mathrm{d}r\right|^2. \tag{14}$$

Here we have used the fact, that $|z|^2 = zz^* = (zz^*)^* = |z^*|^2$ holds for an arbitrary complex number z. The resulting integral can be identified as the Fourier transform (FT) of $\psi_i(\mathbf{r})$, therefore the expression for the intensity becomes

$$I \propto \hbar^2|\mathbf{Ak_f}|^2|\tilde{\psi}_i(\mathbf{k_f})|^2, \tag{15}$$

where $\tilde{\psi}_i(\mathbf{k_f})$ is the wave function in Fourier space. The problem is, that it is the squared absolute value of $\tilde{\psi}_i(\mathbf{k_f})$, thus its imaginary part and therefore its phase is lost during an ARPES measurement. This is called the phase problem, which will be explored in more detail in Chapter 2.4.

2.4 The phase problem

During an ARPES measurement one can only directly observe the photoemission intensity (see Eq. (15)), while the phase of the wave function is lost. This loss of information is called the phase problem, further illustrated in Fig. 3 [6] for a one-dimensional sine function localized to the interval $[-\frac{L}{2}, \frac{L}{2}]$. The model wave function in Fig. 3A represents the desired electron orbital wave function $\Psi(x)$ in real space, constrained to a certain length L. Its Fourier transform $\Psi(k)$, depicted in Fig. 1B, is not what can actually be measured in an ARPES experiment. What is measured is the squared absolute value of the Fourier transformed wave function $|\Psi(k)|^2$ in Fig. 3D, the intensity distribution of the photoelectrons in momentum space. Therefore, when one tries to simply compute the inverse Fourier transform (FT^{-1}) of the measured data, the result in Fig. 3C will be a wave function with wrong spatial extent (twice the length $2L$) and wrong phase (now symmetric). [6]

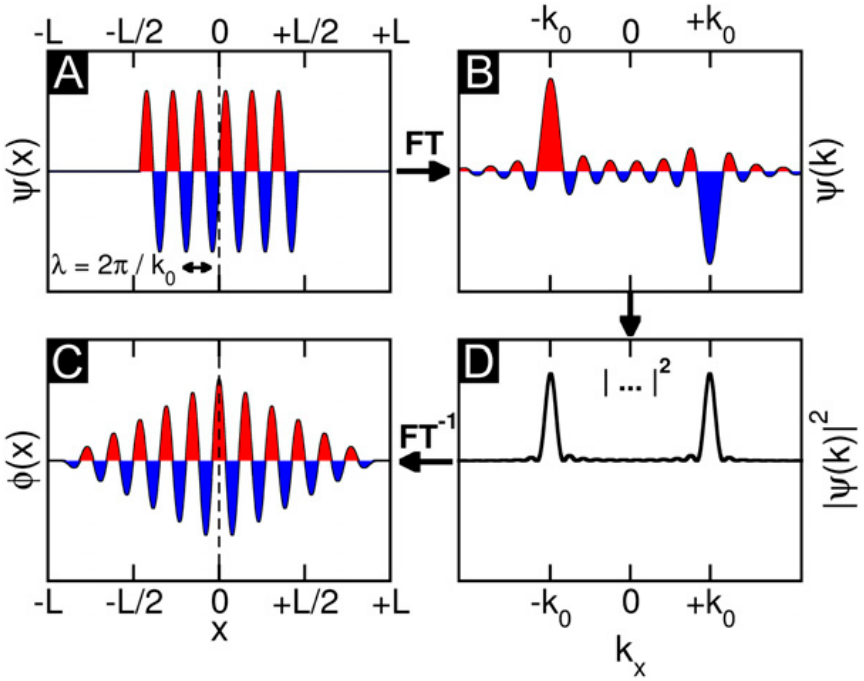

Figure 3: The phase problem of a 1D wave function. (A) 1D model wave function $\Psi(x)$: sine function limited to length L with wave number k_0. (B) Fourier transform (FT) of the model function $\Psi(k)$. (C) Squared absolute value of the function in Fourier space $|\Psi(k)|^2$, where the phase is lost. (D) Inverse Fourier transform (FT^{-1}) back into real space, with wrong spatial extent $2L$ and wrong phase. Image taken from [6], p. 606.

The goal of the following chapters will be to recover the lost phase in Fig. 3C and find the desired wave function like in Fig. 3A, using the known spatial constraint L.

The fact, that the spatial extent becomes twice the length $2L$ can be also demonstrated analytically. Let us define the auto-correlation function $R(x)$ of the real-space wave function $\psi(x)$, which is given by the following integral

$$R(x) = \int_{-\infty}^{\infty} \psi(x')\,\psi(x'-x)\,\mathrm{d}x'. \tag{16}$$

Note that $\psi(x')$ is a function, that vanishes outside of a spatial region of length L. $\psi(x'-x)$ is the same function but shifted by x. With this definition it follows, that $R(x)$ has to be zero outside of the range $2L$, as $R(x > L)$ and $R(x < -L)$ become zero. If we now calculate the Fourier transform $\widetilde{R}(k)$ of $R(x)$ we get

$$\widetilde{R}(k) = \int_{-\infty}^{\infty} R(x)\,e^{ikx}\,\mathrm{d}x \tag{17}$$

$$= \int_{-\infty}^{\infty} \mathrm{d}x \int_{-\infty}^{\infty} \mathrm{d}x'\,\psi(x')\,\psi(x'-x)\,e^{ikx}. \tag{18}$$

Inserting a one in the shape of $e^{ikx'}\,e^{-ikx'} = 1$ into this equation leads to

$$\widetilde{R}(k) = \int_{-\infty}^{\infty} \mathrm{d}x \int_{-\infty}^{\infty} \mathrm{d}x'\,\psi(x')\,\psi(x'-x)\,e^{ikx}\,e^{ikx'}\,e^{-ikx'} \tag{19}$$

$$= \int_{-\infty}^{\infty} \mathrm{d}x'\,\psi(x')\,e^{ikx'} \int_{-\infty}^{\infty} \mathrm{d}x\,\psi(x'-x)\,e^{-ik(x'-x)}. \tag{20}$$

Using the variable substitution $u = x' - x$ in the second integral, we get $du = -dx$, and $\widetilde{R}(k)$ becomes

$$\widetilde{R}(k) = \int_{-\infty}^{\infty} \mathrm{d}x'\,\psi(x')\,e^{ikx'} \int_{x'+\infty}^{x'-\infty} (-\mathrm{d}u)\,\psi(u)\,e^{-iku}. \tag{21}$$

The x' in the second integral boundaries are irrelevant, as we are dealing with infinities. Thus, we can change the boundaries back to $-\infty$ and ∞, and inverse them to remove the minus inside the integral. Finally we find, that the integrals in Eq. (21) can be identified as the Fourier transform $\tilde{\psi}(k)$ and the complex conjugate Fourier transform $\tilde{\psi}^*(k)$ respectively:

$$\widetilde{R}(k) = \underbrace{\int_{-\infty}^{\infty} \mathrm{d}x'\,\psi(x')\,e^{ikx'}}_{\tilde{\psi}^*(k)} \underbrace{\int_{-\infty}^{\infty} \mathrm{d}u\,\psi(u)\,e^{-iku}}_{\tilde{\psi}(k)} \tag{22}$$

$$= \tilde{\psi}^*(k)\,\tilde{\psi}(k) = |\tilde{\psi}(k)|^2. \tag{23}$$

Therefore, the Fourier transform of the auto-correlation function $R(x)$ is just the squared absolute value of $\tilde{\psi}(k)$, which as shown before, is exactly what is measured in an ARPES experiment. If we where to reverse the recent steps we can then conclude, that the inverse Fourier transform of $\widetilde{R}(k) = |\tilde{\psi}(k)|^2$ will have a spatial extent of twice the length $2L$.

3 Method

3.1 Simple reconstruction algorithm (Gerchberg-Saxton (GS) algorithm [7])

To recover the lost phase information of the orbital wave function, we will be applying an iterative algorithm. The first step of the algorithm is to make a guess for the missing phase, which in general means to either randomly generate one (between $-\pi$ and π), or to just leave it zero (Fig. 4A [6]). Using that phase we can compute the first inverse Fourier transform to obtain the real-space function, like that in Fig. 4B. Because we know the true spatial extent of the wave function in real space, usually called the domain constraint, we can now correct the function by multiplying it with a so-called box-function. It has the value of one inside of the box, and $\beta < 1$ outside of the box. Therefore, the wave function is unchanged inside of the box, and reduced by a certain percentage outside of the box. The feedback Parameter β is usually set to $\beta = 0.1$, so that the wave function is reduced to 10% of its original value. After a Fourier transform back to momentum space in Fig. 4D we now have a slightly better guess for the lost phase than at the start. As the final step, we use this wave function, and replace its amplitude with the initially measured data from Fig. 4A, which had no phase. By repeating this process many times, one can now slowly recover the phase (Fig. 4E&4F after 15 iterations, and 4G&4H after 250 iterations.) [6]

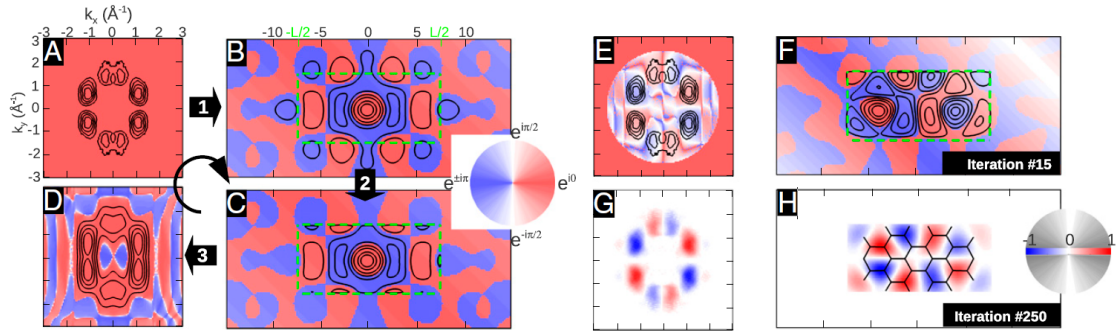

Figure 4: Iterative phase recovery algorithm, using PTCDA HOMO as an example. (A) ARPES measurement of Fourier space, where the color represents the phase and the black isolines illustrate the square root of the intensities. After an inverse Fourier transform we get (B). (C) Corrected wave function, reduced to 10% of its value outside of the confinement box, with its Fourier transform in (D). After 15 iterations leads to (E)&(F), and after 250 iterations to (G)&(H). Image taken from [6], p. 607.

A simple conceptual implementation of this algorithm in python is shown in the following code excerpt:

```
while i < imax:
        f = np.fft.ifft(F) * np.sqrt(N)
        f *= box
        F = np.fft.fft(f) / np.sqrt(N)
        phi = np.angle(F)
        F = F_data * np.exp(1j*phi)
        i += 1
```

Here the small `f` denotes the wave function in real space, and the capital `F` denotes the wave function in Fourier space. First the inverse fast Fourier transform (FFT^{-1}) of `F` is calculated with `np.fft.ifft()`, to get its corresponding real-space function. This example also includes the normalizations of the functions, using the square root `np.sqrt()`. The domain constraint is applied by multiplying the function with a box function, in the code denoted by `box`, which has a value of $\beta = 1$ inside of the domain, and $\beta = 0.1$ outside of the domain. After another FFT back into Fourier space, we calculate the phase of the new wave function, and multiply it with the initial data `F_data`. For the 2-dimensional problem the FFT commands become `np.fft.fft2()` and `np.fft.ifft2()`.

This code can be extended with a termination condition, by implementing an error criterion into the while loop, which was also done in this work, in Section 4.1 for the one-dimensional problem.

3.2 Error reduction (ER) and hybrid input-output (HIO) algorithm

This is an improved method of the Gerchberg-Saxton (GS) algorithm described above, suggested by James Fienup in 1978 [8]. It divides the simple reconstruction algorithm into two separate ones, the error reduction (ER) algorithm, and the hybrid input-output (HIO) algorithm, which was extended by Ross John Harder in 2010 to the phase-constrained hybrid input-output (PC-HIO) algorithm [9]. The basic concept of all these algorithms is the same, with the only difference being in the approach to the object domain constraint (see Fig. 5). [7]

The first step is again to make an initial guess for the phase of the wave function and multiply it with the measured Amplitude $|F(k_x, k_y)| = \sqrt{I(k_x, k_y)}$, the square root of the Intensity as we have seen in Chapter 2.2 and 2.4. After an inverse Fourier transform we get the first initial estimate for our wave function in real space in Fig. 5 step (1), denoted by a small $g_k(x, y)$, usually called the input [8]. In step (2) we obtain the wave function in Fourier space $G_k(k_x, k_y)$. Next, the amplitude of $G_k(k_x, k_y)$ gets replaced by the measured data $|G_k(k_x, k_y)| = |F(k_x, k_y)|$ (Fourier domain constraint) in step (3), to get $G'_k(k_x, k_y)$, which is identical to the final step in the GS algorithm. Finally, we

obtain the wave function in real space $g'_k(x, y)$ in step (4), called the output [8]. The main difference of this method comes in step (5), the object domain constraint.

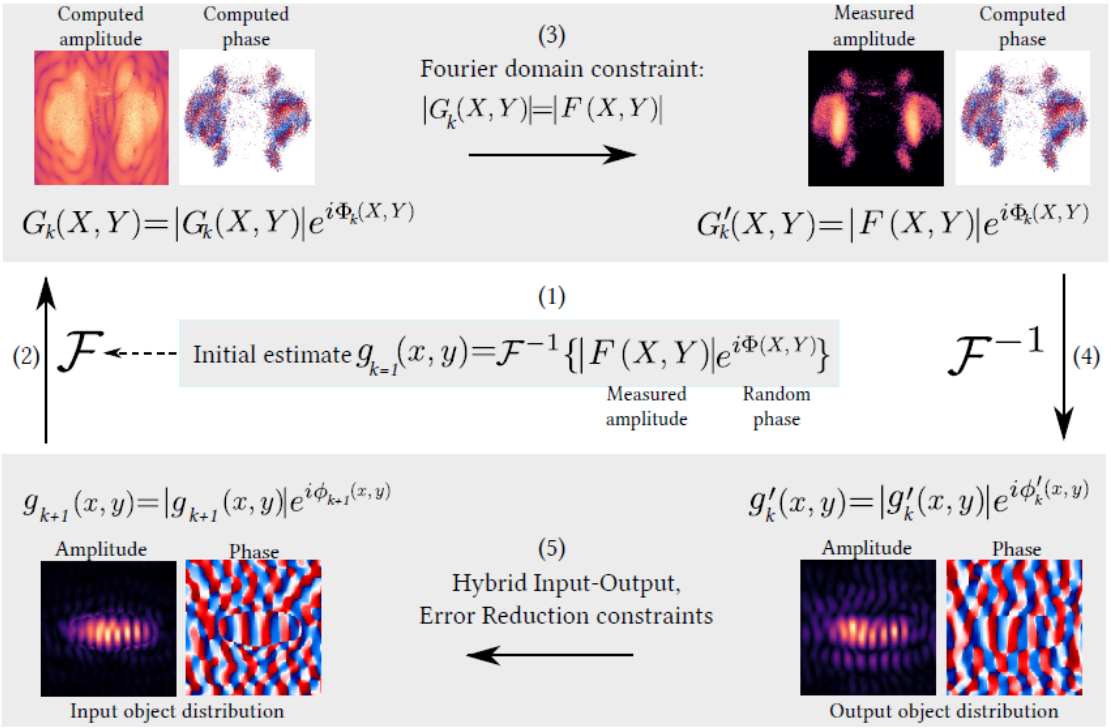

Figure 5: Iterative phase recovery algorithm, using ER and PC-HIO. Here $X \equiv k_x$ and $Y \equiv k_y$. (1) Multiply measured data $|F(k_x, k_y)| = \sqrt{I(k_x, k_y)}$ with a random phase, and calculate the FT, to get an initial estimate for the real-space function $g_{k=1}(x, y)$. (2) FT of the wave function in real space, to get the Fourier space function $G_k(k_x, k_y)$. (3) Replace the calculated amplitude $|G_k(k_x, k_y)|$ with the experimental amplitude $|F(k_x, k_y)|$ (4) inverse FT back into real space (5) Alternate between the object domain constraint of ER and HIO algorithm. Image taken from [7], p. 16.

The object domain constraint of the GS algorithm from Chapter 3.1 was a simple multiplication of the wave function with a box-function, with a value of β=1 inside the box and β=0.1 outside of the box. The domain constraints of the ER and PC-HIO are as follows:

$$ER: \qquad g_{k+1}(x,y) = \begin{cases} g'_k(x,y), & \text{if } (x,y) \in \gamma \\ 0, & \text{if } (x,y) \notin \gamma \end{cases} \tag{24}$$

$$PC-HIO: \qquad g_{k+1}(x,y) = \begin{cases} g'_k(x,y), & \text{if } (x,y) \in \gamma \\ g_k(x,y) - \beta g'_k(x,y), & \text{if } (x,y) \notin \gamma \end{cases} \tag{25}$$

where γ are all the points inside of the domain constraint, $\beta = 0.9$ is the feedback

parameter, and $g_k(x, y)$ and $g'_k(x, y)$ is the wave function before and after applying the Fourier domain constraint. As seen in these equations the ER algorithm sets the wave function outside of the box to zero during an iteration, simply reducing the error as the name suggests, while the PC-HIO algorithm subtracts the output $g'_k(x, y)$, multiplied by a feedback parameter β, from the input $g_k(x, y)$. In both cases the wave function inside the domain stays the same, and gets reduced outside of the domain, similar to the GS algorithm.

Usually, the reconstruction process consists of a combination of the ER and PC-HIO algorithms in an alternating sequence, to achieve a faster convergence. In particular we start out with 20 iterations of the ER algorithm, followed by 10-30 iterations of the PC-HIO algorithm and finished with about 5-10 ER algorithm iterations. [7]

To implement the ER and PC-HIO algorithms in python we can again use a while loop:

```
while i < imax:
        G = np.fft.fft(g)                              #step (2)
        phiG = np.angle(G)
        G_prime = F_data * np.exp(1j*phiG)             #step (3)
        g_prime = np.fft.ifft(G_prime)                 #step (4)

        #step (5): Switch between ER and PC-HIO
        if i <= nER or i > (nER+nHIO):                 #ER
                g = g_prime * box0
        if i > nER and i <= (nER+nHIO):                #PC-HIO
                gnew = g - 0.9*g_prime
                g = np.add(gnew*box0inv, g_prime*box0)
        i += 1
```

Where `g`, `g_prime`, `G` and `G_prime` correspond to $g_k(x, y)$, $g'_k(x, y)$, $G_k(k_x, k_y)$ and $G'_k(k_x, k_y)$ from before. `phiG` is the phase of $G_k(k_x, k_y)$ and `F_data` are again the measured data from the experiment. In step (5) we alternate between the ER and the PC-HIO domain constraint, with `nER` and `nHIO` being the number of ER and PC-HIO iterations. `box0` is a box function with value $\beta = 1$ inside the box and $\beta = 0$ outside, while `box0inv` is the opposite of that, with $\beta = 0$ inside the box and $\beta = 1$ outside.

The number of PC-HIO steps has the most impact on the resulting image, and can be the difference between a useful image or not, as can be seen in the highly varying number of PC-HIO iterations in the results of Chapter 4.3. Therefore, the code was implemented in a way, such that we could compare the images with different number of PC-HIO iterations, and choose the ones of interest.

4 Results

4.1 The one-dimensional problem

To demonstrate the effectiveness of the algorithm described in Chapter 3.1, we will first tackle the problem of a simpler, one-dimensional phase retrieval. To achieve that, the first task was to create a one-dimensional sine function localized to the interval $[-\frac{L}{2}, \frac{L}{2}]$, similar to the model wave function in Fig. 3A, and then compute the absolute value Fourier transform of it, to simulate the measured data with lost phase (see Fig. 6). As the domain constraint is fulfilled, because of given L, we can apply the iterative algorithm on the absolute value of the function, and after some iterations get back the initial sine function.

In Fig. 6A we have the initial model function in blue, with a confinement length of L=12 (note that some lines overlap). Analytically this wave function has the form

$$\psi(x) = \begin{cases} \sin(k_0 x), & \text{if } |x| \leq \frac{L}{2} \\ 0, & \text{if } |x| \geq \frac{L}{2}. \end{cases} \tag{26}$$

To obtain its Fourier transform $\tilde{\psi}(k)$, we need to solve the integral

$$\tilde{\psi}(k) = \int_{-\infty}^{\infty} \psi(x)\, e^{-ikx}\, \mathrm{d}x. \tag{27}$$

As the wave function in Eq. (26) is only nonzero for $|x| \leq \frac{L}{2}$, the integral outside of this boundary vanishes, and we are left to calculate

$$\tilde{\psi}(k) = \int_{-L/2}^{L/2} \sin(k_0 x)\, e^{-ikx}\, \mathrm{d}x. \tag{28}$$

This integral can be solved by integration by parts twice in a row, to get an equation, which has the integral on both sides:

$$\int_{-L/2}^{L/2} \sin(k_0 x)\, e^{-ikx}\, \mathrm{d}x = \left[(i\, \sin(k_0 x) + \frac{k_0}{k} \cos(k_0 x)) \right]_{-L/2}^{L/2} + \frac{k_0^2}{k^2} \int_{-L/2}^{L/2} \sin(k_0 x)\, e^{-ikx}\, \mathrm{d}x. \tag{29}$$

leading to:

$$\tilde{\psi}(k) = \int_{-L/2}^{L/2} \sin(k_0 x)\, e^{-ikx}\, \mathrm{d}x = \left[\frac{(ik \sin(k_0 x) + k_0 \cos(k_0 x)) e^{-ikx}}{k^2 - k_0^2} \right]_{-L/2}^{L/2} \tag{30}$$

Inserting the boundary values and using the trigonometric identities

$$2\sin\theta\cos\phi = \sin(\theta+\phi) + \sin(\theta-\phi) \tag{31}$$
$$2\cos\theta\sin\phi = \sin(\theta+\phi) - \sin(\theta-\phi), \tag{32}$$

leads to the final solution of the Fourier transform of $\psi(x)$:

$$\tilde{\psi}(k) = \frac{i\sin(\frac{(k+k_0)L}{2}}{k+k_0} - \frac{i\sin(\frac{(k-k_0)L}{2}}{k-k_0}. \tag{33}$$

Fig. 6B shows the result of Eq. (33) by numerical integration, using the numpy FFT command in python. Note that the wave function $\tilde{\psi}(k)$ is imaginary, therefore the dashed blue line in Fig. 6B (real part) is zero, besides tiny peaks at $+k_0$ and $-k_0$, which are most likely a result of discrete Fourier transform instead of the continuous Fourier transform in the analytic calculation.

What is measured though is the absolute value of the Fourier transformation, i.e. the green curve in Fig. 6B, which after an inverse Fourier transform yields the solid blue line in Fig. 7A, with a confinement length of $2L$.

Then the procedure of the GS algorithm is applied. After five iterations the algorithm is starting to take effect, as seen in Fig. 8A and 8B, having a small dip in the amplitude outside of the spatial confinement. An image even closer to the initial one we can find after 40 iterations in Fig. 9, which becomes almost identical to the initial wave function after 250 iterations, in Fig. 10.

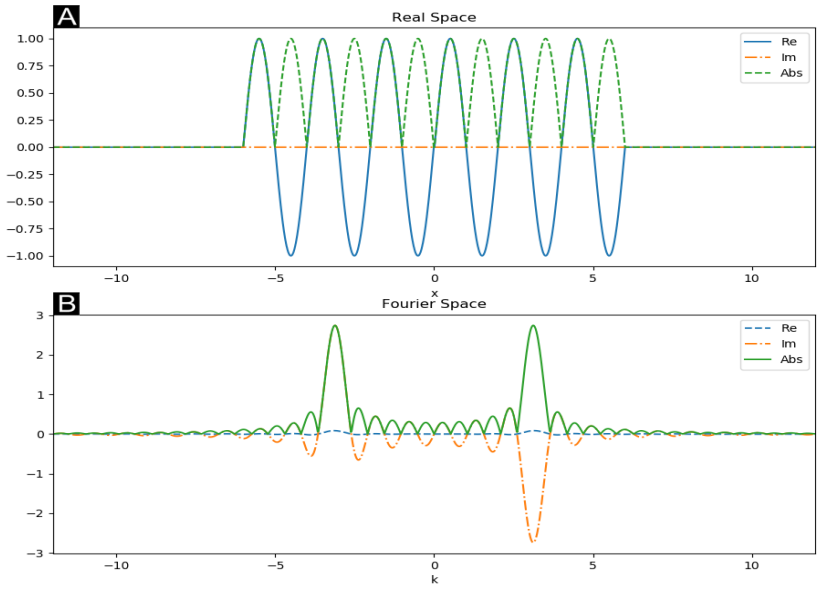

Figure 6: Initial model sine function with real part, imaginary part, and absolute value in real space (A) and Fourier space (B).

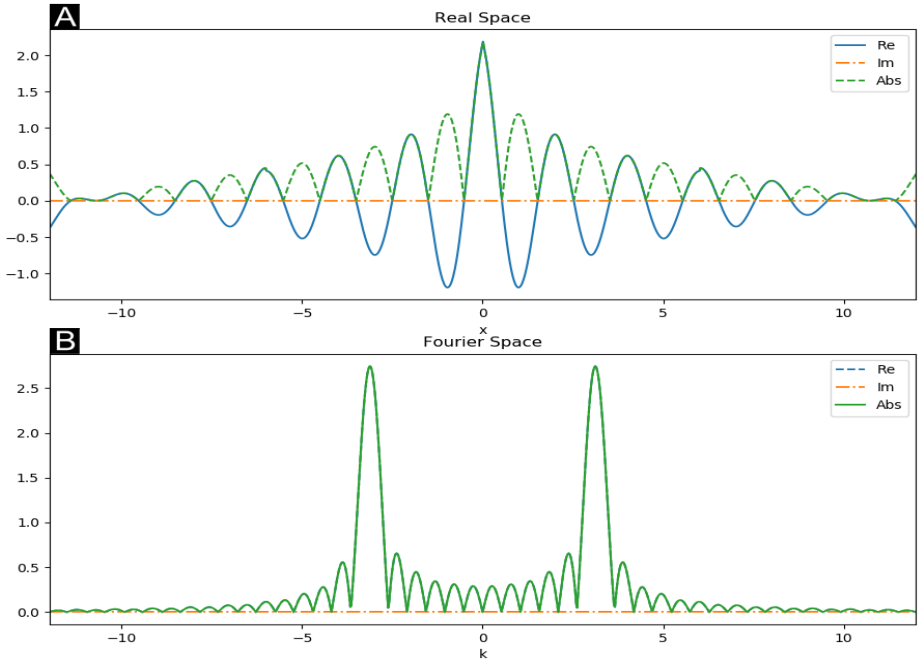

Figure 7: Iteration N=0 of the algorithm: real part, imaginary part, and absolute value in real space (A) and Fourier space (B).

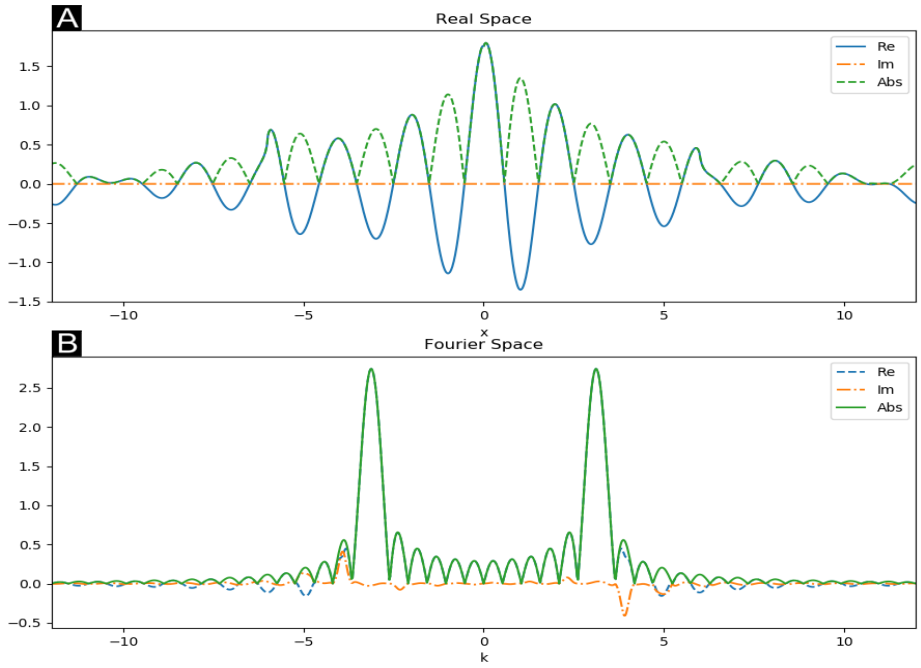

Figure 8: Iteration N=5 of the algorithm: real part, imaginary part, and absolute value in real space (A) and Fourier space (B).

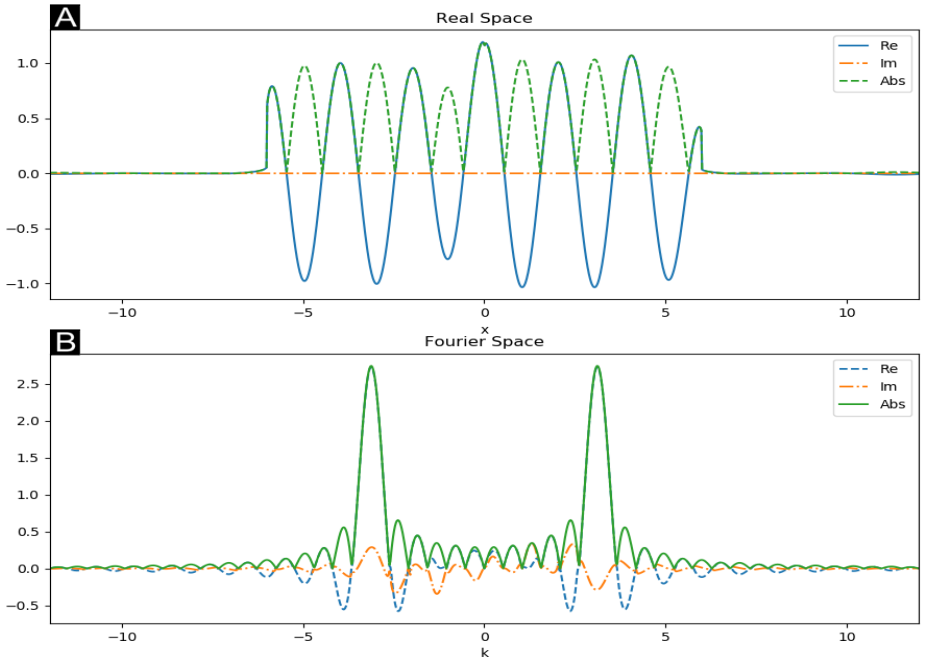

Figure 9: Iteration N=40 of the algorithm: real part, imaginary part, and absolute value in real space and Fourier space.

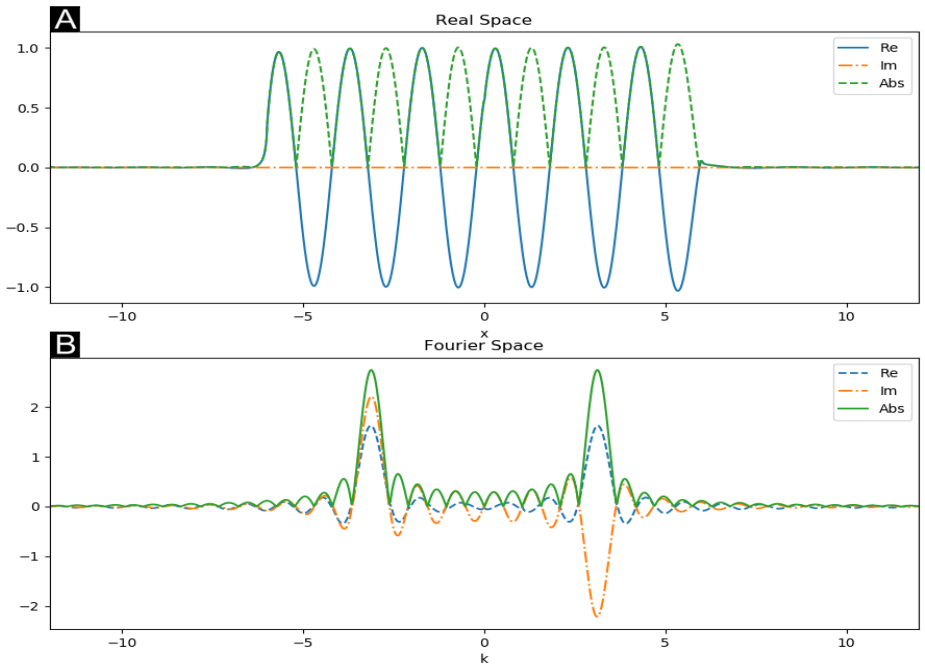

Figure 10: Iteration N=250 of the algorithm: real part, imaginary part, and absolute value in real space (A) and Fourier space (B).

Here we implement a termination condition, which stops the program, when the difference of phases between wave functions inside the box becomes small enough. This is the so-called cost function:

$$\sum_{x,y}^{box} |\mathrm{Arg}(f_n(x,y)) - \mathrm{Arg}(f_{n+1}(x,y))| \to \mathrm{Min}, \tag{34}$$

which is minimized until a certain accuracy is achieved. To obtain an accuracy of 10^{-15} for the phase difference inside of the spatial domain, only $N = 66$ iterations are needed in our example. As one can see in Fig. 11, the phase difference fluctuates between each step, which could be avoided by averaging over a reasonably small number of iterations (eg. 3 to 5) to make the curve more smooth.

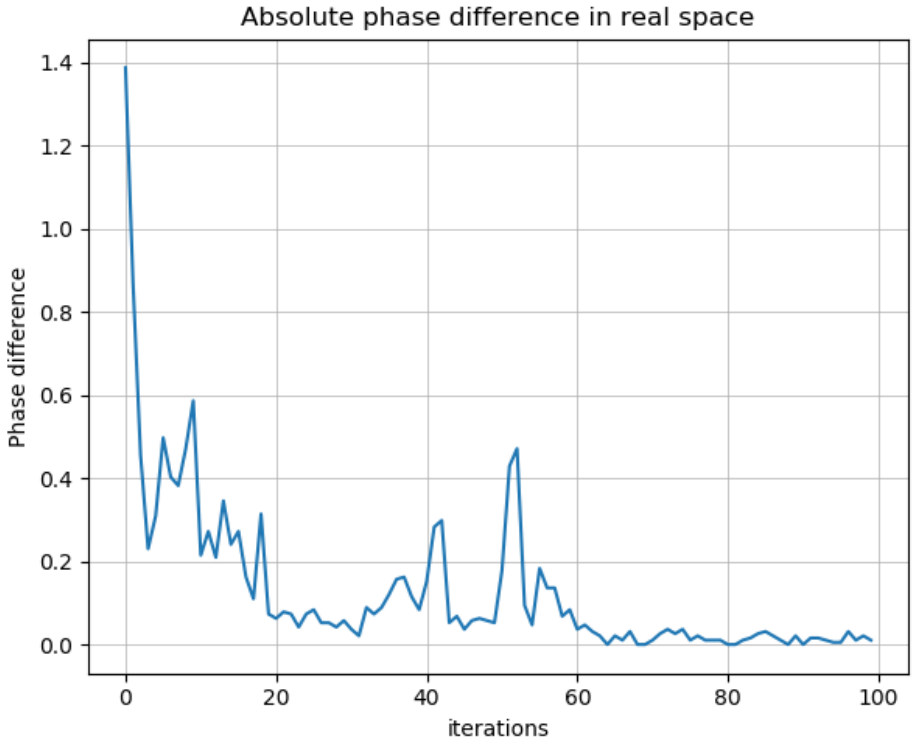

Figure 11: Convergence of the absolute value phase difference.

4.2 The two-dimensional problem: GS algorithm

This work uses the ARPES measurement data of perylene-3,4,9,10-tetracarboxylic dianhydride (PTCDA) ($C_{24}H_8O_6$) for its highest occupied molecular orbital (HOMO) and its lowest unoccupied molecular orbital (LUMO), as well as the data of pentacene (5A) ($C_{22}H_{14}$) for its LUMO, HOMO, and HOMO-1 (one below the HOMO). Here we assumed, that the spatial constraint of the molecules are defined by their van der Waals size, which is 14.8×7.2 Å^2 for PTCDA and 14.8×5.6 Å^2 for 5A. [6]

Due to the fact, that the measurement data used in this work is centered around the coordinate origin, it includes negative values (see Fig. 12A). Applying the inverse (fast) Fourier transform on this measurement data, using the python (numpy) command `numpy.fft.ifft2()`, will result in a wave function with every second element in the grid having a wrong sign (see Fig. 12C, on the example of PTCDA HOMO).

To avoid this one needs to either create a chessboard-like matrix with alternating sign and multiply the wave function with this matrix, or to shift the measurement data to start in the origin of the coordinate system, at the values $x = 0$ and $y = 0$, to avoid negative values. This way we can get the corrected image in Fig. 13C.

As can be seen in Fig. 14B, after 60 iterations the phase of the wave function is starting to take on shape, but the wave function itself has almost no change in its real space in Fig. 14C. After 80 iterations the wave function in real space is also starting to change (see Fig. 15C). The final result of PTCDA HOMO is depicted in Fig. 16, which is in agreement with its one-electron wave function from density functional theory (DFT) [3].

The results of PTCDA LUMO, and 5A HOMO, HOMO-1 and LUMO are shown in the Fig. 17, 18, 19 and 20. The GS algorithm only failed for the data of 5A HOMO-1 (Fig. 19), where the real part in real space does not correspond to expectations, but the imaginary part does. Physically, this does not make any sense, so it is most likely just a coincidence. For a correct result see Fig. 26, where we successfully applied the ER/PC-HIO algorithm on the same data.

One notable fact is that the wave function of PTCDA LUMO in real space is slightly oscillating between two vertically symmetric solutions with each iteration (Fig. 21), which comes from an asymmetry in the measurement data and can be an issue when trying to define a termination condition using the wave function in real space. To fix this, we can modify the measurement data to be symmetric along the x axis, by replacing the values $y > 0$ above the x axis with the values $y < 0$ below, and get a result like in Fig. 22, which stops oscillating, but also has its confinement space shifted along the x axis.

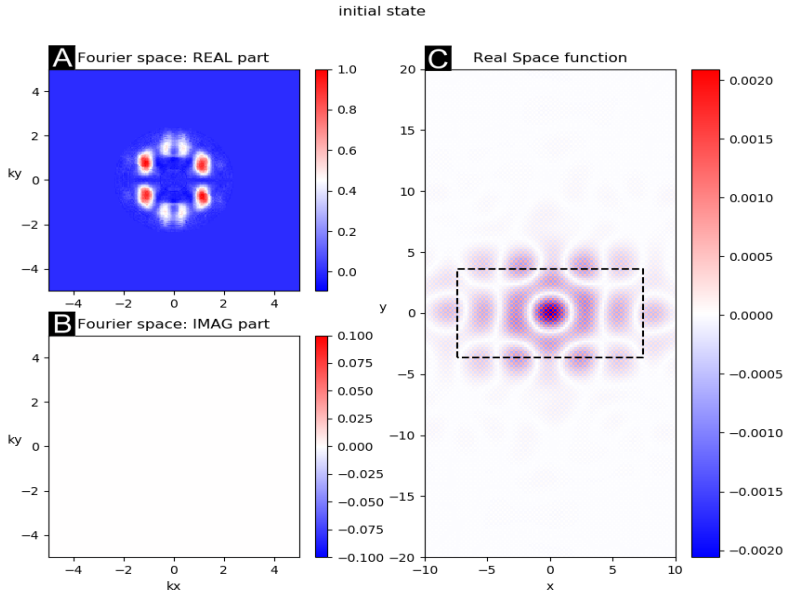

Figure 12: GS algorithm: Initial uncorrected wave function of PTCDA HOMO, with the ARPES measurement data in (A): (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real space

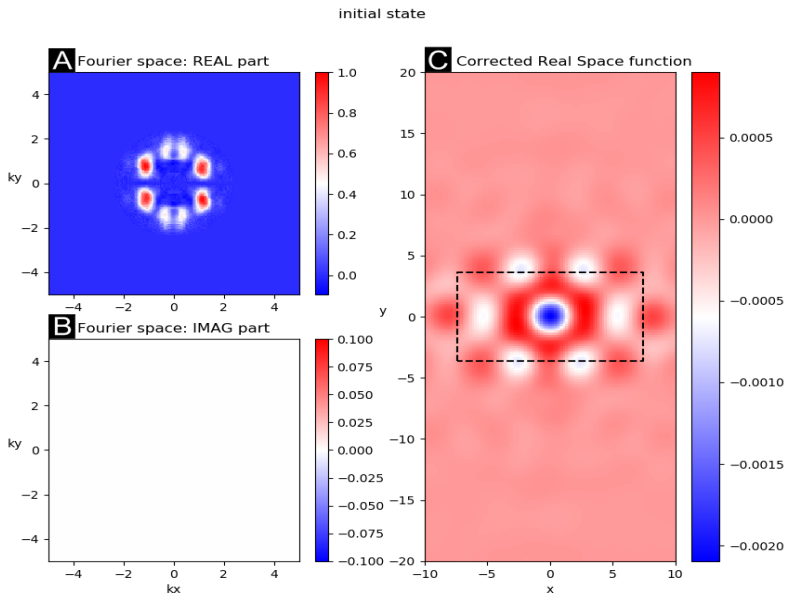

Figure 13: GS algorithm; Initial corrected wave function of PTCDA HOMO, with the ARPES measurement data in (A): (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real space

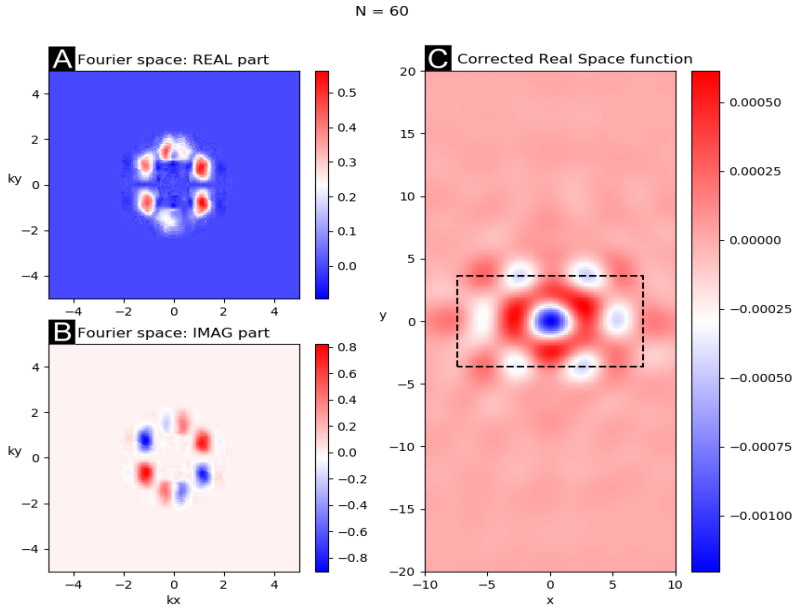

Figure 14: GS algorithm; PTCDA HOMO, after 60 iterations: (A) real part in Fourier space. (B) imaginary part in Fourier space (B). (C) real space, corrected

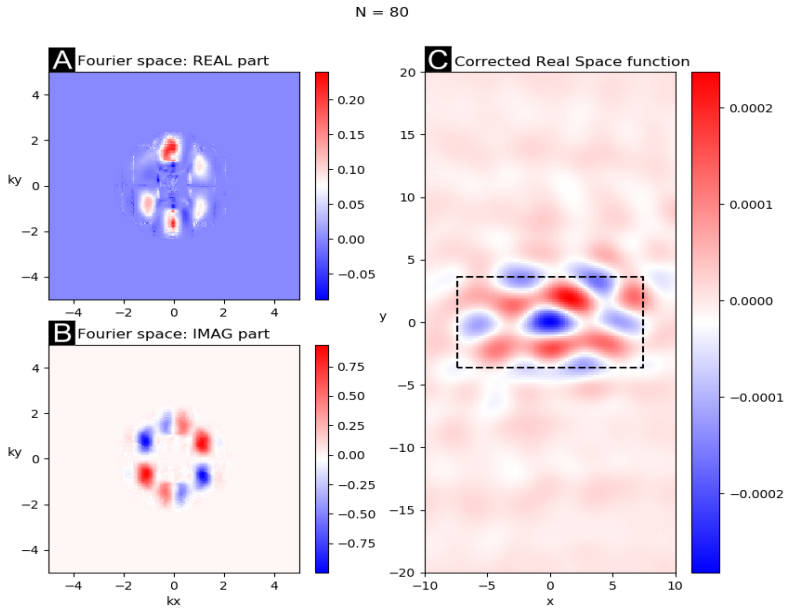

Figure 15: GS algorithm; PTCDA HOMO, after 80 iterations: (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real space, corrected

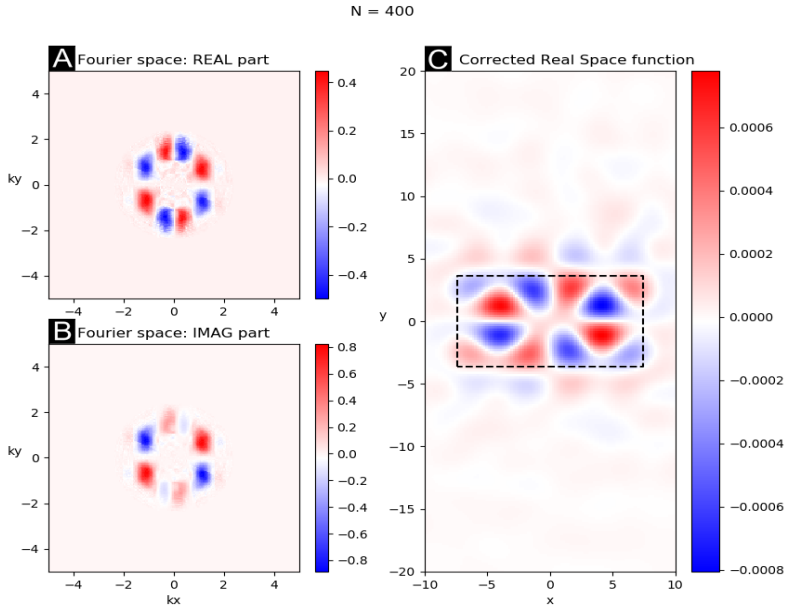

Figure 16: GS algorithm; Result of PTCDA HOMO, after 400 iterations: (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real space, corrected

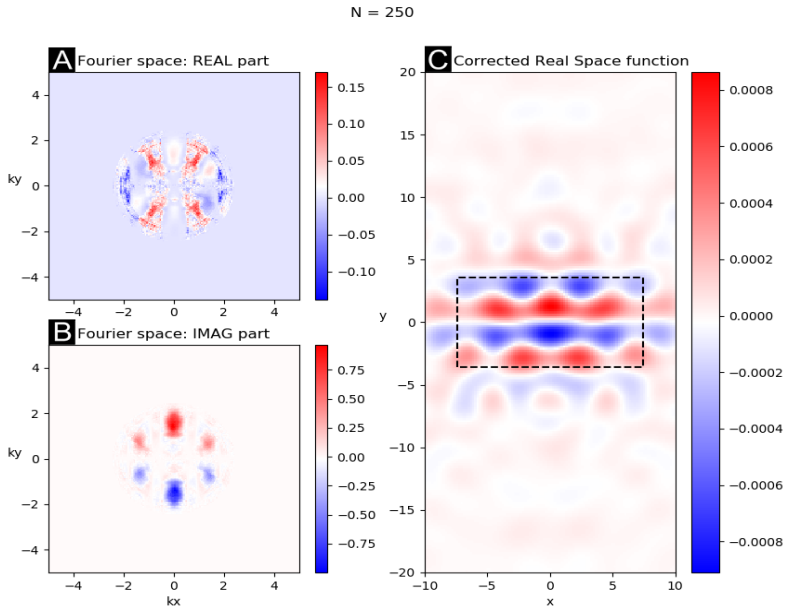

Figure 17: GS algorithm; Result of PTCDA LUMO, after 250 iterations: (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real space, corrected

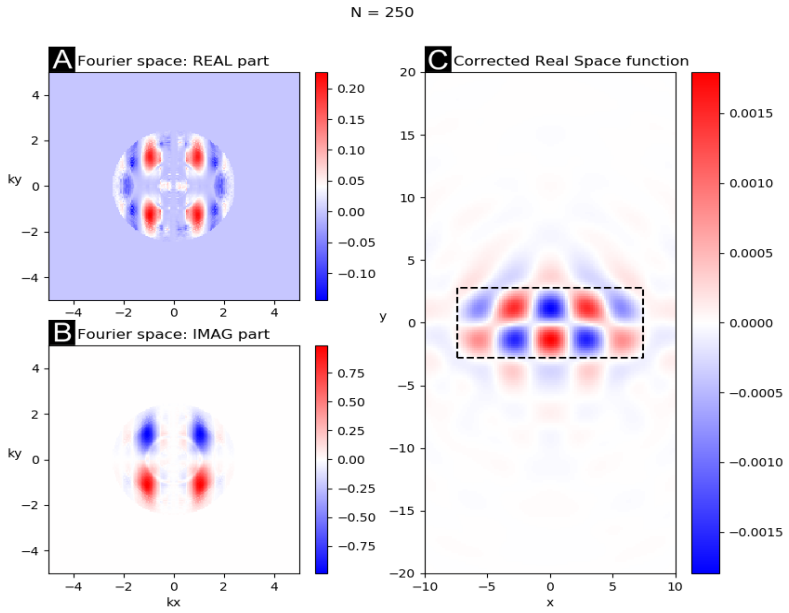

Figure 18: GS algorithm; Result of 5A HOMO, after 250 iterations: (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real space, corrected

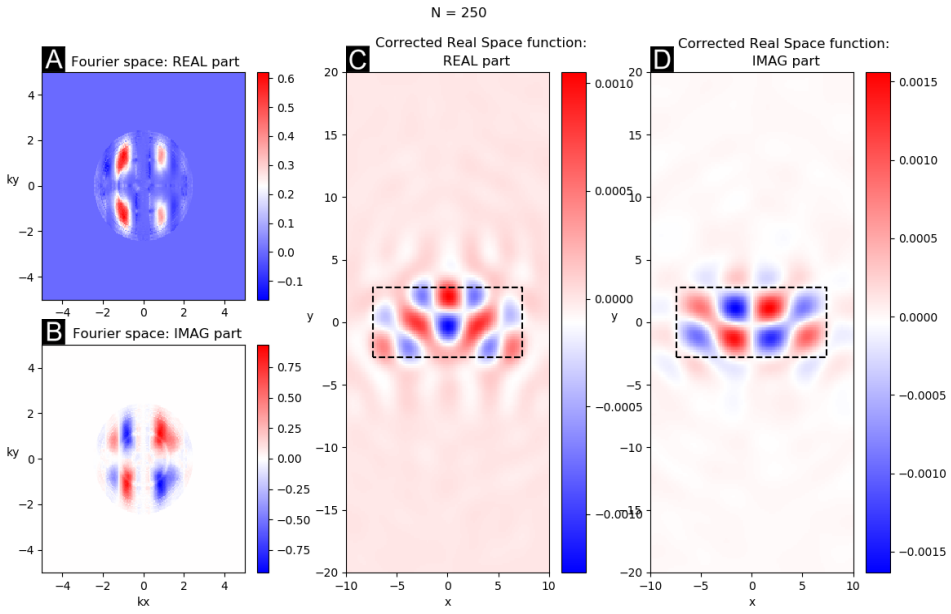

Figure 19: GS algorithm; (Failed) Result of 5A HOMO-1, after 250 iterations: (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real part in real space (D) imaginary part in real space

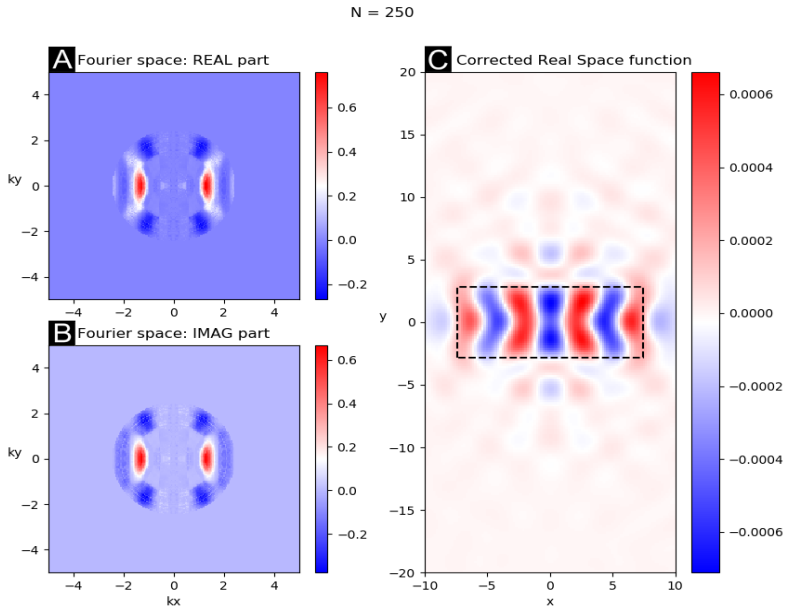

Figure 20: GS algorithm; Result of 5A LUMO, after 250 iterations: (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real space, corrected

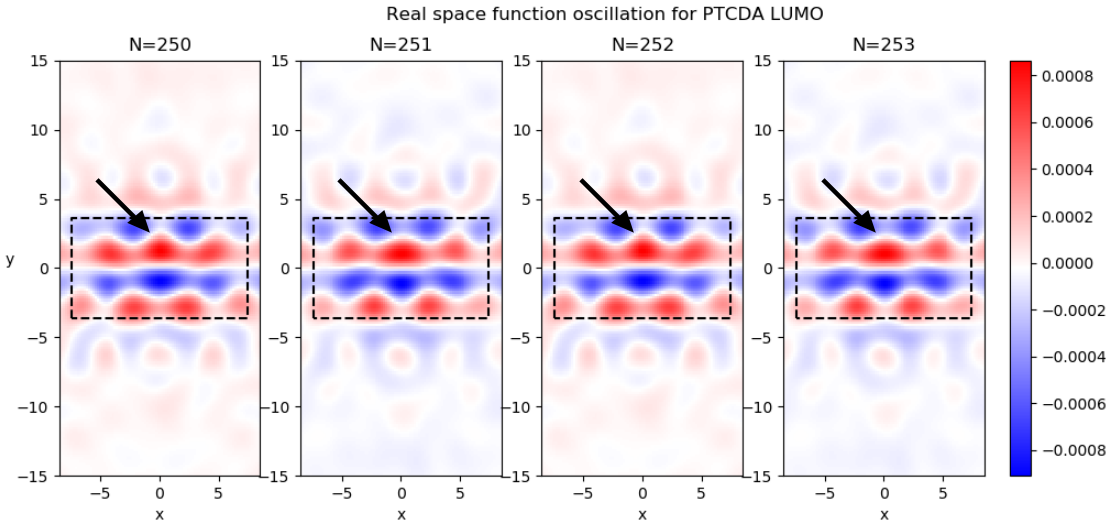

Figure 21: Real-space function oscillation for PTCDA LUMO, depicted for the iterations 250 to 253. Center peaks change shape for every second iteration (marked with arrows).

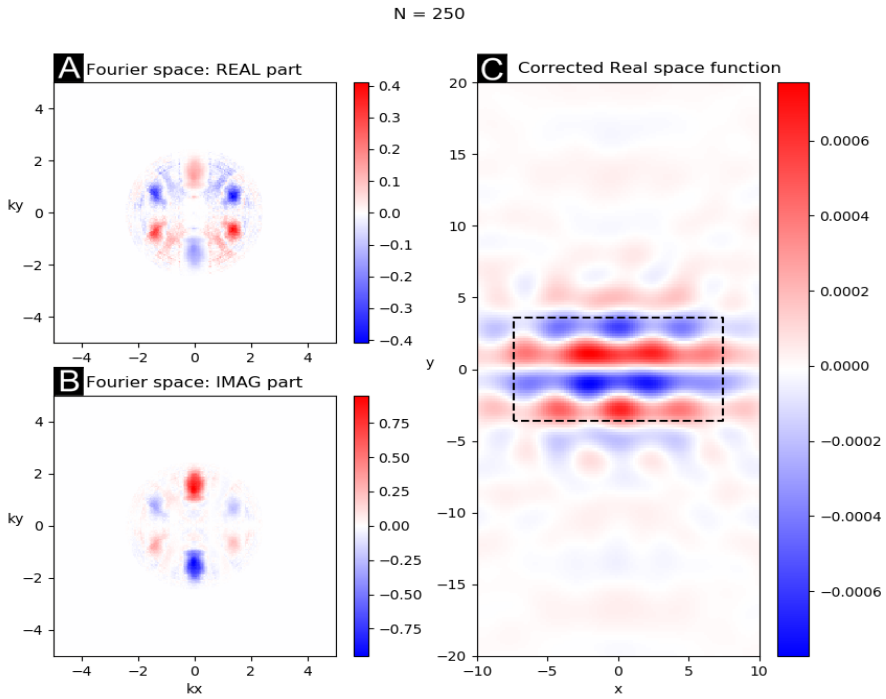

Figure 22: GS algorithm; Symmetritzed result of PTCDA LUMO, after 250 iterations: (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real space, corrected

4.3 The two-dimensional problem: ER and PC-HIO algorithm

Here we again deal with the ARPES measurement data of PTCDA and 5A for the HOMO and LUMO, as well as the HOMO-1 for 5A only, with the van der Waals sizes 14.8×7.2 Å^2 for PTCDA and 14.8×5.6 Å^2 for 5A. [6] The results are depicted in Fig. 23 to 27, all in agreement with DFT, with the one in Fig. 26 being of particular interest, as it failed using the GS algorithm in the previous chapter.

The phase-recovery process was realized using a random initial phase, therefore repeating runs can lead to slightly different outcomes. The following images where all obtained using 20 ER iterations in the beginning and 10 ER iterations at the end, while the number of PC-HIO iterations varies between 8-30, depending on the data. Having a different number of PC-HIO iterations can also lead to results which are shifted in real space, or do not seem to agree with DFT at all. Therefore, the algorithm was set up in a way that enabled the direct comparison of the final figure with different PC-HIO steps between 8-30, to find the best image. It is also again necessary to multiply the final real-space function with a chessboard-like matrix, for the reasons discussed in chapter 4.2.

This method requires a lot less iterations than the GS algorithm, making it much faster. However, this comes with a disadvantage of the algorithm being very dependent on the number of PC-HIO steps for different ARPES measurement data. It can be very effective for measurement data, where the number of PC-HIO iterations barely changes the final image. But in other cases it requires so many cycles with different PC-HIO steps, that this method is even slower than the GS algorithm.

Also noticeable is that due to the nature of the PC-HIO algorithm, the wave function in Fourier space takes on a more squarish shape for all the results, showing that this algorithm works better for objects with sharper edges [7].

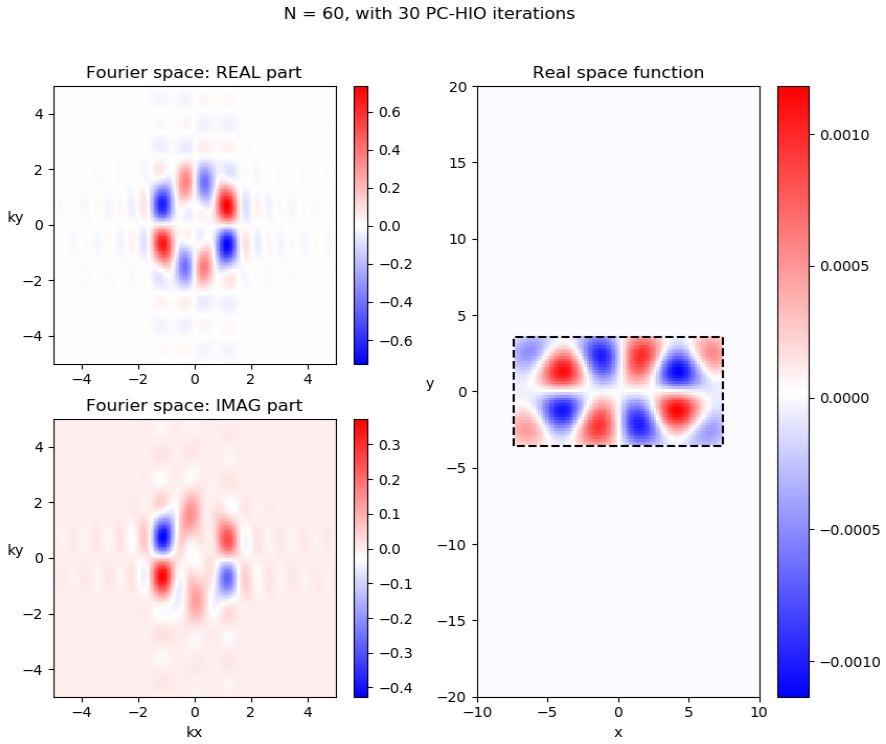

Figure 23: Result of PTCDA HOMO, using ER/PC-HIO method with 30 PC-HIO iterations. (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real space, corrected

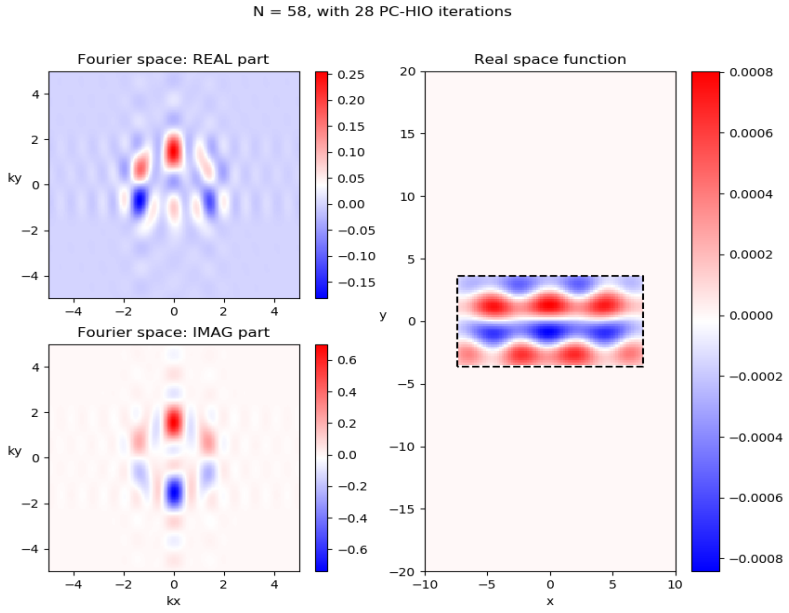

Figure 24: Result of PTCDA LUMO, using ER/PC-HIO method with 28 PC-HIO iterations. (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real space, corrected

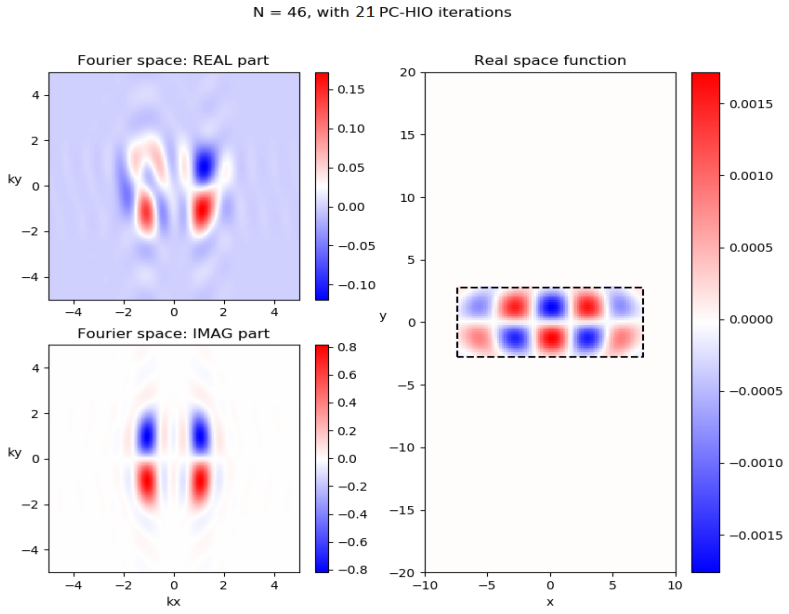

Figure 25: Result of 5A HOMO, using ER/PC-HIO method with 21 PC-HIO iterations. (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real space, corrected

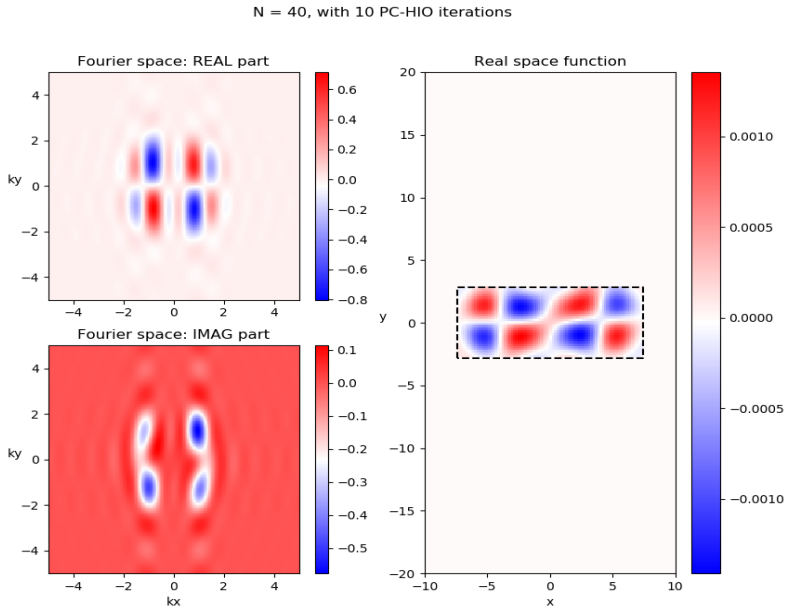

Figure 26: Result of 5A HOMO-1, using ER/PC-HIO method with 10 PC-HIO iterations. Corrected result of Fig. 19. (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real space, corrected

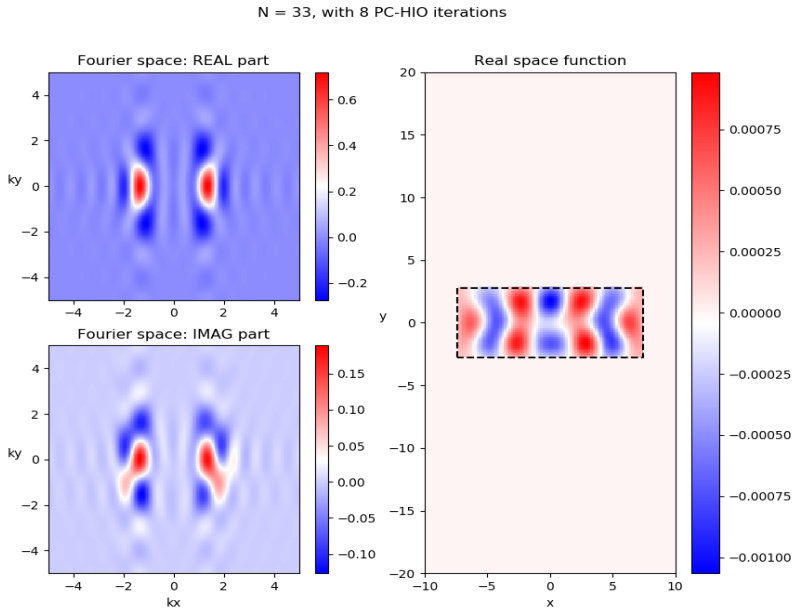

Figure 27: Result of 5A LUMO, using ER/PC-HIO method with 8 PC-HIO iterations. (A) real part in Fourier space. (B) imaginary part in Fourier space. (C) real space, corrected

5 Conclusions and Outlook

In this thesis the electron wave functions of PTCDA and 5A for there HOMO, LUMO and HOMO-1 for 5A where reconstructed from angle-resolved photoemission data. We used two different iterative methods, the GS algorithm and the ER/PC-HIO algorithms, to recover the orbitals in real space and Fourier space.

The GS algorithm succeeded for the most part, with only one failed result for 5A HOMO-1, all the other images are in good agreement with DFT. The measurement data of PTCDA LUMO turned out to be asymmetrical, resulting in real-space images which oscillate between two solutions. One can fix this problem by modifying the measurement data, but then the final real-space wave function has its confinement space shifted along the x axis.

The results obtained with the ER and PC-HIO algorithm are all in agreement with DFT. This method requires a lot less iterations than the GS algorithm, but due to its dependence on the number of PC-HIO steps, one needs to run the algorithm multiple times to find the correct number of iterations for each particular measurement data. The wave functions in Fourier space also take on an squarish shape.

In this thesis we determined the quality of an image from the PC-HIO algorithm by comparing to DFT. In future works, this process can be automated using normalized root-mean-square metric in Fourier space E_R [7]. By applying this metric at the end of each ER cycle one can determine the quality of each image. At the end one can also average over multiple images with different randomized initial phases, to make the results more consistent and easier to recreate. Overall the PC-HIO algorithm has a lot of ways for improvement, making it a very exciting method to explore and optimize.

References

[1] J. M. Zuo, M. Kim, M. O'Keeffe, J. C. H. Spence. *Direct observation of d-orbital holes and Cu-Cu bonding in Cu_2O.* Nature 401, p. 49-52, 1999

[2] D. G. Truhlar, P. C. Hiberty, S. Shaik, M. S. Gordon, D. Danovich. *Orbitals and the Interpretation of Photoelectron Spectroscopy and (e,2e) Ionization Experiments.* Angew. Chem. Int. Ed. 58, p. 2-9, 2019

[3] Daniel Lüftner. *Doctoral Thesis: Orbital tomography: Understanding photoemission of organic molecular films.* University Graz, 2015.

[4] E. W. Plummer, and W. Eberhardt. *Angle-Resolved Photoemission as a Tool for the Study of Surfaces.* Advance in Chemical Physics, vol. XLIX, p. 533-565, University of Pennsylvania, 1982.

[5] Georg Koller, Peter Puschnig, Alexander Gottwald, and F. Stefan Tautz. *Photoelektronen-tomographische Bilder von Moleklorbitalen: Elektronenorbitale in 3D.* Phys. Unserer Zeit vol. 47, p. 192-198, 4/2016.

[6] Daniel Lüftner, Thomas Ules, Eva Maria Reinisch, Georg Koller, Serguei Soubatch, Stefan Tautz, Michael G. Ramsey, and Peter Puschnig. *Imaging the wave functions of adsorbed molecules.* PNAS Vol. 111, p. 605-610, 2014.

[7] Pavlo Kliuiev. *Doctoral Thesis: Reconstruction of Molecular Orbitals from Photoemission Data with Iterative Phase Retrieval Algorithms.* University Zrich, 2018.

[8] J. R. Fienup. *Reconstruction of an object from the modulus of its Fourier transform.* Optics Letters Vol. 3 No. 1, p. 27-29, Environmental Research Institute of Michigan, 1978.

[9] R. Harder, M. Liang, Y. Sun, Y. Xia, I. K. Robinson. *Imaging of complex density in silver nanocubes by coherent x-ray diffraction.* New J. Phys. Vol. 12 No. 3, 035019, 2010.