Dominik Brandstetter

Photoemission tomography: experimental data evaluation by fitting to simulated momentum maps

Bachelor thesis

written at the Department of Physics
University of Graz

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

Graz, August 2019

Abstract

Orbital tomography is a new method of analysis in surface science, developed in the last decade. It uses the data from angle-resolved photoemission spectroscopy (ARPES) and density functional theory (DFT) to clarify or even reconstruct orbitals of more complex organic molecules like bisanthene. The conservation of momentum parallel to the surface for an escaping photoelectron provides a basis for this. Using Fermi's golden rule, it can be shown that the differential cross-section is proportional to the Fourier transformation of the initial state of the electron.
Thus, comparing experimental data and the data of a DFT-simulation, applying a "plane wave" approximation for the final state of the electron, information about the initial state can be gained.

The practical part of this thesis was concerned with the extension of the functionality provided by a program written for said comparison. Besides implementing a "brute-force" algorithm to find the optimal orientation of the molecule in the simulation, multiple features to manipulate the data were added.

Kurzzusammenfassung

Orbital-Tomographie ist eine neue Analysemethode der Oberflächenphysik, die im letzten Jahrzehnt entwickelt wurde. Sie nutzt die Daten aus der winkelaufgelösten Photoelektronenspektroskopie (ARPES) und die Dichtefunktionaltheorie (DFT) zur Aufklärung oder gar Rekonstruktion der Orbitale auch komplexerer organischer Moleküle wie Bisanthen. Grundlage bildet die Impulserhaltung der Parallelkomponenten der im photoelektrischen Prozess ausgelösten Elektronen im Übergang an der Oberfläche. Es kann über Fermis goldene Regel gezeigt werden, dass der differentielle Wirkungsquerschnitt proportional zur Fouriertransformation des Ausgangszustands des Elektrons ist.
Aus dem Vergleich der experimentellen Daten mit denen aus einer DFT-Simulation, unter der Approximation des Endzustands des Elektron als ebene Welle, können Informationen über den Ausgangszustand gewonnen werden.

Diese Arbeit beschäftigt sich mit der Erweiterung an Funktionalität eines für eben diesen Vergleich geschriebenen Programmes. Neben eines "brute-force" Algorithmus, zum Anpassen der Orientierung des Moleküls in der Simulation an die Messdaten, wurden einige Features zur Manipulation der Daten implementiert.

Ehrenwörtliche Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen inländischen oder ausländischen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht. Die vorliegende Fassung entspricht der eingereichten elektronischen Version.

Acronyms

ARPES angle-resolved photoemission spectroscopy

DFT density functional theory

GUI graphical user interface

HOMO highest occupied molecular orbital

IUPAC international union of pure and applied chemistry

LUMO lowest unoccupied molecular orbital

PES photoemission spectroscopy

UPS ultraviolet photoelectron spectroscopy

VCS version control systems

XPS X-ray photoelectron spectroscopy

Contents

Chapter 1

Introduction

For this thesis, I was allowed to contribute to a project working on an emerging technology labelled *orbital tomography*. Orbital tomography aims to utilise the data from angle-resolved photoemission spectroscopy (ARPES) experiments to clarify or even reconstruct orbitals of more complex organic molecules.

This thesis is split into two major parts.
In chapter "Theoretical Background" first, some background information and the basics of orbital tomography will be clarified. Starting with a more general summary of the *photoelectric effect*, the central process in question, we will work our way up to a more detailed description for the special case of a hydrogen atom using quantum mechanics and perturbation theory. In doing so, we will derive the famous equation "Golden Rule No. 2". This chapter will close with a quick overview of the experimental side, namely ARPES.
In the second chapter "Practical Work", a close look on what I was able to contribute to this project will be given. This work revolves around extending a program written to compare experimental and simulated momentum maps. This part starts with a short description of the data used and general information on the program itself. The principal portion, however, will focus on individual features implemented during this thesis. As closure, we will briefly discuss what key issues should be addressed in the future of this program.

To limit the scope, this thesis only aims to give a brief introduction to this currently developing technology as far as is necessary to understand the work done for this thesis. It does not claim to be a complete or even basic description of the material. If one seeks more information on various subjects, please refer to the following sources [1–6].

Chapter 2

Theoretical Background

2.1 Photoelectric effect

In the centre of orbital tomography lies what is now known as the *photoelectric effect*. First observed by Heinrich Hertz (1857-1894) 1887 and Wilhelm Hallwachs (1859-1922) 1895 [7], it was Albert Einstein (1879-1955) who was awarded the Nobel Prize in physics in 1921 for his "discovery of the law of the photoelectric effect" [8]. He lay down his theory 1905 in one of his Annus mirabilis papers titled "Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt" [9].

Based on the work of Max Planck (1858-1947) just five years earlier [10, 11], in his essay, Einstein proposed the quantisation of the electromagnetic field. He introduced the idea that light consists of packets of energy, photons, which he called quanta.
According to Einstein, the energy E of a photon, therefore, can be characterised solely by its frequency ν using the Planck–Einstein relation

$$E = h\nu \tag{2.1}$$

with h being a constant of proportionality called Planck's constant [11].
Because a photon carries momentum, when it gets absorbed, it will transfer its entire energy to the electron in the form of kinetic energy. Assuming a photon hits an electron in a bound state inside a molecule (see figure 2.1), we can express the maximum kinetic energy E_{kin} of the electron escaping a material by applying the law of conservation of energy as such

$$E_{kin} = h\nu - E_B - \Phi \tag{2.2}$$

The kinetic energy E_{kin} is reduced by the binding energy E_B and what is called the work function Φ. It represents the energy it takes the electron to escape the surface of the material [1].

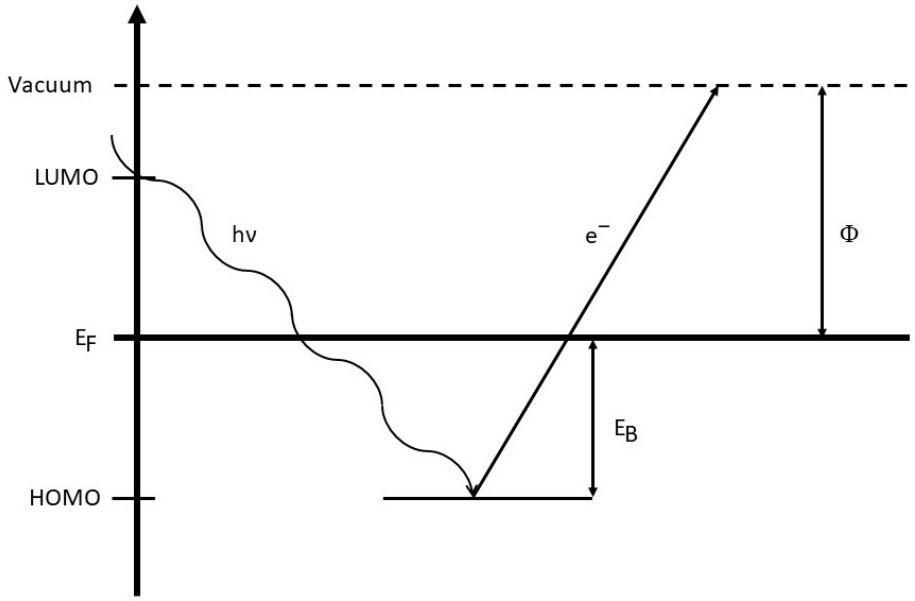

Figure 2.1: A schematic depiction of the photoelectric effect in terms of energy levels. A bound electron bound in the HOMO (***H**ighest **O**ccupied **M**olecular **O**rbital*) escapes past the Fermi energy E_F and LUMO (***L**owest **U**noccupied **M**olecular **O**rbital*) to a free vacuum state. The energy $h\nu$ from the incidental photon, fully transferred to the electron in the form of kinetic energy by its absorption, needs to exceed the binding energy E_B and the work function Φ of the material.

2.2 Transition rate between electronic states

In this section, we want to determine the transition rate of an electron between a bound eigenstate ($|b\rangle$, ϵ_b) and a quasi-continuum of states $\{|f\rangle\}$, whose energy range encloses ϵ_b using time-dependent perturbation theory [2, 12]. As a periodic perturbation potential $\hat{V}$, we will state an electromagnetic wave $\boldsymbol{A}(\boldsymbol{r}, t)$ to determine the simplified case of the photoelectric effect for a hydrogen atom in the ground state in section 2.3.

Starting with a Hamiltonian $\hat{H}$ for our system

$$\hat{H} = \hat{H}_0 + \hat{V}(t) \tag{2.3}$$

consisting of a Hamiltonian $\hat{H}_0$ for an unperturbed, known system

$$\hat{H}_0 |n\rangle = \epsilon_n |n\rangle \quad (n = 1, 2, ...) \tag{2.4}$$

and a in some sense "small", time-dependent potential $\hat{V}(t)$ we want to approximate a solution for $\hat{H}$. Because $\hat{V}(t)$ is an operator, we introduce a parameter λ in (2.3)

$$\hat{H} = \hat{H}_0 + \lambda\hat{V}(t) \tag{2.5}$$

which allows us to expand the solution as a power series in λ. Setting $\lambda = 1$ later will give us back the system $\hat{H}$ to be solved (2.3), while $\lambda = 0$ would equal the unperturbed system $\hat{H}_0$.
Because the solution $|\psi(\boldsymbol{r}, t; \lambda)\rangle$ of the time-dependent Schrödinger equation (2.6) [12]

$$i\hbar\frac{\partial}{\partial t} |\psi(\boldsymbol{r}, t; \lambda)\rangle = \hat{H} |\psi(\boldsymbol{r}, t; \lambda)\rangle \tag{2.6}$$

now also depends on λ we can expand it into a power series

$$|\psi(\boldsymbol{r}, t; \lambda)\rangle = \sum_{s=0}^{\infty} \lambda^s |\psi_s(\boldsymbol{r}, t)\rangle \tag{2.7}$$

Plugging (2.7) into (2.6) and comparing coefficients of equal powers of λ on both sides we find an iterative solvable system of equations as a solution of (2.3)

$$i\hbar\frac{\partial}{\partial t} |\psi_0(\boldsymbol{r}, t)\rangle = \hat{H}_0 |\psi_0(\boldsymbol{r}, t)\rangle \tag{2.8}$$

$$i\hbar\frac{\partial}{\partial t} |\psi_{s'}(\boldsymbol{r}, t)\rangle = \hat{H}_0 |\psi_{s'}(\boldsymbol{r}, t)\rangle + \hat{V}(t) |\psi_{s'-1}(\boldsymbol{r}, t)\rangle \quad (s' = 1, 2, 3, ...) \tag{2.9}$$

Using the zeroth-order (2.8), the unperturbed system (2.4), we can impose our initial condition by assuming the system to be in an eigenstate $|b\rangle$ of $\hat{H}_0$ at $t = 0$. Because we assumed a "small" perturbing potential $\hat{V}$, we restrict ourselves to a first-order approximation $s' = 1$.

$$\begin{aligned} |\psi_0(\boldsymbol{r}, 0)\rangle &= |b\rangle \\ |\psi_{s'=1}(\boldsymbol{r}, 0)\rangle &= 0 \end{aligned} \tag{2.10}$$

Because the eigenstates $|n\rangle$ of $\hat{H}_0$ form a complete orthonormal system (2.4), using the phase factor $\exp(-[i\epsilon_n t]/\hbar)$, we can express $|\psi_1(\boldsymbol{r}, t)\rangle$ as a linear

combination of time-evolving eigenstates. Using this we also find the time-evolution of $|b\rangle$

$$|\psi_0(\boldsymbol{r},t)\rangle = \exp(-\frac{i\epsilon_b t}{\hbar})\,|b\rangle \tag{2.11}$$

$$|\psi_1(\boldsymbol{r},t)\rangle = \sum_{n=1}^{\infty} c_n(t)\exp(-\frac{i\epsilon_n t}{\hbar})\,|n\rangle \tag{2.12}$$

We plug (2.11) and (2.12) into the first-order approximation (2.9) and simplify by evaluating the time derivative on the left side of the equation as well as letting $\hat{H}_0$ act on its eigenstates on the right side. In the former, we have to apply the product rule, since the coefficients $c_n(t)$ now also experience a time dependence due to the time-dependent perturbation potential $\hat{V}(t)$ [12, 13]. In the latter $\hat{H}_0\,|n\rangle = \epsilon_n\,|n\rangle$ and therefore cancels out the additional term on the left.

$$\begin{aligned}
&i\hbar\frac{\partial}{\partial t}\sum_{n=1}^{\infty} c_n(t)\exp(-\frac{i\epsilon_n t}{\hbar})\,|n\rangle = \\
&\hat{H}_0\sum_{n=1}^{\infty} c_n(t)\exp(-\frac{i\epsilon_n t}{\hbar})\,|n\rangle + \hat{V}(t)\exp(-\frac{i\epsilon_b t}{\hbar})\,|b\rangle \\
&i\hbar\sum_{n=1}^{\infty}\frac{\partial c_n(t)}{\partial t}\exp(-\frac{i\epsilon_n t}{\hbar})\,|n\rangle = \hat{V}(t)\exp(-\frac{i\epsilon_b t}{\hbar})\,|b\rangle
\end{aligned} \tag{2.13}$$

Projecting one state $|m\rangle$ of the quasi-continuum of states $\{|f\rangle\}$, the electron could end up on (2.13)

$$i\hbar\frac{\partial c_m(t)}{\partial t} = \exp(i\omega_{mb}t)\,\langle m|\,\hat{V}(t)\,|b\rangle \tag{2.14}$$

with

$$\omega_{mb} = \frac{\epsilon_m - \epsilon_b}{\hbar} \tag{2.15}$$

results in a linear inhomogeneous differential equation in $c_m(t)$ with an easily obtainable solution

$$c_m(t) = \frac{1}{i\hbar}\int_0^t \exp(i\omega_{mb}t')\,\langle m|\,\hat{V}(t')\,|b\rangle\,dt' \tag{2.16}$$

To evaluate (2.16) further, we need to specify the potential $\hat{V}(t)$ in order to calculate the matrix element $\langle m|\,\hat{V}(t')\,|b\rangle$. Let $\hat{V}(t)$ be a periodic potential (e.g. an electromagnetic wave) [12]

$$\hat{V}(t) = \hat{V}_0^-\,\exp(-i\omega t) + \hat{V}_0^+\,\exp(i\omega t),\ \hat{V}_0^+ \coloneqq (V_0^-)^\dagger \tag{2.17}$$

Here the complex conjugated term will ensure that $\hat{V}(t)$ is hermitian. We plug (2.17) into (2.16) and evaluate the integral

$$\begin{aligned}
c_m(t) &= \frac{1}{i\hbar}\,\langle m|\,\hat{V}_0^{\pm}\,|b\rangle\int_0^t \exp(i\,[\omega_{mb}\pm\omega]\,t')dt' \\
c_m(t) &= \frac{\langle m|\,\hat{V}_0^{\pm}\,|b\rangle}{\hbar(\omega_{mb}\pm\omega)}\,\{1-\exp(i\,[\omega_{mb}\pm\omega]\,t)\}
\end{aligned} \tag{2.18}$$

Here we neglected one of the integrals because of the quotient only $\omega_{mb} = \omega$ or $\omega_{mb} = -\omega$ will contribute significantly. The absolute square of the coefficients $c_m(t)$ (2.18) gives the probability for a transition $|b\rangle \to |m\rangle$ during a particular time t. Dividing the probability by t results in the transition rate $W_{b\to m}$ between $|b\rangle$ and $|m\rangle$ [12]

$$W_{b\to m} = \frac{|c_m(t)|^2}{t} = \frac{\left|\langle m|\, \hat{V}_0^{\pm}\, |b\rangle\right|^2}{\hbar^2} \times \frac{4\,\sin^2\left(\frac{t(\omega_{mb}\pm\omega)}{2}\right)}{t\,(\omega_{mb}\pm\omega)^2} \tag{2.19}$$

In the limit $t \to \infty$ we can approximate the second term as a delta distribution [12]

$$\lim_{t\to\infty} \frac{4\,\sin^2\left(\frac{t(\omega_{mb}\pm\omega)}{2}\right)}{t\,(\omega_{mb}\pm\omega)^2} = 2\pi\delta(\omega_{mb}\pm\omega) = 2\hbar\pi\delta(\epsilon_{mb}\pm\epsilon) \tag{2.20}$$

$$\Rightarrow W_{b\to m} = \frac{2\pi}{\hbar} * \left|\langle m|\, \hat{V}_0^{\pm}\, |b\rangle\right|^2 \times \delta(\epsilon_{mb}\pm\epsilon) \tag{2.21}$$

We arrived at what is known as *Fermi's Golden Rule* (2.21). First derived by Paul Dirac (1902-1984) 1927 [14, 15] and later dubbed "Golden Rule No. 2" by Enrico Fermi (1901-1954) [16, 17], this equation is an indispensable tool in quantum mechanics. It enables the estimation of transition rates between states, initiated by a perturbation potential, by calculating the matrix elements of the potential. The delta distribution ensures the law of conservation of energy [1].

2.3 Photoelectric effect for the hydrogen atom

Applying Fermi's Golden Rule (2.21), we now want to show that the photoemission cross-section/differential cross-section $\frac{d\sigma}{d\Omega}$, the infinitesimal probability $d\sigma$ of photoelectron emission occurring in the direction of an infinitesimal angle element $d\Omega$, is proportional to the Fourier transformation of the initial state of the electron. For this, we will assume a photon of energy $\hbar\omega_\gamma$ freeing a hydrogen electron in the $|1s\rangle$ ground state [18].

An electromagnetic wave can be characterised by the vector potential $\boldsymbol{A}(\hat{\boldsymbol{r}}, t)$

$$\boldsymbol{A}(\hat{\boldsymbol{r}}, t) = A_0 \boldsymbol{\epsilon} \cos(\boldsymbol{k}_\gamma \hat{\boldsymbol{r}} - \omega t) \tag{2.22}$$

with a polarisation vector $\boldsymbol{\epsilon}$ for which $\boldsymbol{\epsilon k} = 0$ holds [12].
Using the Lagrange function for a charged particle in an electromagnetic field, one can show that the Hamiltonian operator can be obtained by a substitution $\hat{\boldsymbol{p}} \to \hat{\boldsymbol{p}} - \frac{q}{c}\boldsymbol{A}(\hat{\boldsymbol{r}}, t)$ for the unperturbed Hamiltonian [1, 12, 19]

$$\hat{H} = \frac{1}{2m_e}\left[\hat{\boldsymbol{p}} + \frac{e}{c}\boldsymbol{A}(\hat{\boldsymbol{r}}, t)\right]^2 - \frac{e^2}{\hat{\boldsymbol{r}}} \tag{2.23}$$

The last term in (2.23) conforms to the Coulomb potential of a single electron in a hydrogen atom. Expanding the brackets results in a term $\mathcal{O}(\boldsymbol{A}(\hat{\boldsymbol{r}}, t)^2)$ which is small enough to be neglected in our case [1, 12]. Utilising the Coulomb gauge $\nabla \boldsymbol{A}(\hat{\boldsymbol{r}}, t) = 0$ the cross terms in (2.23) can be simplified. One needs to be careful though; an operator always acts on some function Ψ

$$\begin{aligned}
\{\hat{\boldsymbol{p}},\ \boldsymbol{A}(\hat{\boldsymbol{r}}, t)\}\,\Psi &= \left[(-i\hbar\boldsymbol{\nabla}\boldsymbol{A}(\hat{\boldsymbol{r}}, t) + \boldsymbol{A}(\hat{\boldsymbol{r}}, t)(-i\hbar\boldsymbol{\nabla})\right]\Psi \\
&= -i\hbar\left[\boldsymbol{\nabla}(\boldsymbol{A}(\hat{\boldsymbol{r}}, t)\Psi) + \boldsymbol{A}(\hat{\boldsymbol{r}}, t)\boldsymbol{\nabla}(\Psi)\right] \\
&= -2i\hbar\boldsymbol{A}(\hat{\boldsymbol{r}}, t)\boldsymbol{\nabla}(\Psi) = 2\boldsymbol{A}(\hat{\boldsymbol{r}}, t)\hat{\boldsymbol{p}}
\end{aligned} \tag{2.24}$$

$$\begin{aligned}
\Rightarrow \hat{H} &= \frac{1}{2m_e}\left[\hat{\boldsymbol{p}}^2 + 2\frac{e}{c}\boldsymbol{A}(\hat{\boldsymbol{r}}, t)\hat{\boldsymbol{p}}\right] - \frac{e^2}{\hat{\boldsymbol{r}}} \\
&= \frac{1}{2m_e}\hat{\boldsymbol{p}}^2 - \frac{e^2}{\hat{\boldsymbol{r}}} + \frac{e}{m_e c}\boldsymbol{A}(\hat{\boldsymbol{r}}, t)\hat{\boldsymbol{p}}
\end{aligned} \tag{2.25}$$

We identify the unperturbed Hamiltonian $\hat{H}_0$ and the perturbation potential $\hat{V}$ in (2.25), by plugging in (2.22), as

$$\begin{aligned}
\hat{H}_0 &= \frac{1}{2m_e}\hat{\boldsymbol{p}}^2 - \frac{e^2}{\hat{\boldsymbol{r}}} \\
\hat{V}(t) = \frac{e}{m_e c}\boldsymbol{A}(\hat{\boldsymbol{r}}, t)\hat{\boldsymbol{p}} &= \frac{A_0 e}{2m_e c}\left[e^{i(\boldsymbol{k}_\gamma \hat{\boldsymbol{r}} - \omega t)} + e^{-i(\boldsymbol{k}_\gamma \hat{\boldsymbol{r}} - \omega t)}\right]\boldsymbol{\epsilon}\hat{\boldsymbol{p}}
\end{aligned} \tag{2.26}$$

When comparing (2.26) and (2.17) we recognise

$$\hat{V}_0^- = \frac{A_0 e}{2m_e c} e^{i\boldsymbol{k}_\gamma \hat{\boldsymbol{r}}} \boldsymbol{\epsilon}\hat{\boldsymbol{p}} \tag{2.27}$$

To find the transition rate of an electron from an initial state $|b\rangle = |1s\rangle$ to a free state initiated by the excitation of a photon $\hbar\omega_\gamma$, we will plug (2.27)

into (2.21). Additionally, we approximate the photoelectron as a plane wave $|m\rangle = |\boldsymbol{k}\rangle = \exp(i\left[\boldsymbol{k}\hat{\boldsymbol{r}} - \omega t\right])$ [12]

$$\begin{aligned} W_{b\to m} &= \frac{|c_m(t)|^2}{t} = \frac{|A_0|^2 \pi e^2}{2\hbar m_e^2 c^2} \times \left| e^{i\boldsymbol{k}_\gamma \hat{\boldsymbol{r}}} \boldsymbol{\epsilon} \langle \boldsymbol{k}| \hat{\boldsymbol{p}} |1s\rangle \right|^2 \times \delta(\epsilon_k - \epsilon_{1s} - \epsilon) \\ &= \frac{|c_m(t)|^2}{t} = \frac{|A_0|^2 \pi \hbar e^2}{2m_e^2 c^2} \times \left| e^{i\boldsymbol{k}_\gamma \hat{\boldsymbol{r}}} \boldsymbol{\epsilon}\boldsymbol{k} \langle \boldsymbol{k}|1s\rangle \right|^2 \times \delta(\epsilon_k - \epsilon_{1s} - \epsilon) \\ &= \frac{|c_m(t)|^2}{t} = \frac{|A_0|^2 \pi \hbar e^2}{2m_e^2 c^2} (\boldsymbol{\epsilon}\boldsymbol{k})^2 \left|\langle \boldsymbol{q}|1s\rangle\right|^2 \times \delta(\epsilon_k - \epsilon_{1s} - \epsilon) \qquad (2.28) \\ \langle \boldsymbol{q}| &= \exp(-i\boldsymbol{q}\hat{\boldsymbol{r}}),\ \boldsymbol{q} = \boldsymbol{k} - \boldsymbol{k}_\gamma \qquad (2.29) \end{aligned}$$

Next, we will integrate over all possible final momentum states d^3k'

$$P_{b\to m} = \int W_{b\to m} d^3k' \qquad (2.30)$$

The Delta distribution in (2.28) ensures the conservation of energy

$$\epsilon_{k'} \stackrel{!}{=} \epsilon_k = \frac{\hbar^2 k^2}{2m_e} = \epsilon_{1s} + \hbar\omega_\gamma \qquad (2.31)$$

and we can use it to get rid of one integration. Doing so requires a conversion of our Cartesian coordinate system to a spherical one and another transformation $k' \to \epsilon_{k'} \Rightarrow dk' \to (m_e)/(\hbar^2 k')d\epsilon_{k'}$

$$\begin{aligned} P_{b\to m} &= \int W_{b\to m}(\boldsymbol{k}') k'^2 \sin(\vartheta') d\varphi' d\vartheta' dk' \\ &= \int W_{b\to m}(\boldsymbol{k}') k'^2 \sin(\vartheta') \frac{m_e}{\hbar^2 k'} d\varphi' d\vartheta' d\epsilon_{k'} \\ &= \int \frac{|A_0|^2 \pi e^2 k}{2m_e c^2 \hbar} (\boldsymbol{\epsilon}\boldsymbol{k})^2 \left|\langle \boldsymbol{q}|1s\rangle\right|^2 \sin(\vartheta') d\varphi' d\vartheta' \\ &= \frac{(2\pi)^2 e^2 k}{V m_e \omega_\gamma} \int (\boldsymbol{\epsilon}\boldsymbol{k})^2 \left|\langle \boldsymbol{q}|1s\rangle\right|^2 d\Omega \qquad (2.32) \end{aligned}$$

In the last step, we have substituted for $|A_0|^2 \to (8\pi\hbar c^2)/(V\omega_\gamma)$ using the fact that inside the volume V is a photon with an energy $\hbar\omega_\gamma$. Moreover, we identify $\sin(\vartheta')d\varphi' d\vartheta' \to d\Omega$ as the infinitesimal solid angle element [20]. Shifting from a single photon inside the volume V to a photon density c/V, in the derivation of $P_{b\to m}$ with respect to $d\Omega$, we find the differential cross-section.

$$\frac{d\sigma}{d\Omega} = \left(\frac{dP_{b\to m}}{d\Omega}\right) / \left(\frac{c}{V}\right) = \frac{(2\pi)^2 e^2 k}{m_e c \omega_\gamma} (\boldsymbol{\epsilon}\boldsymbol{k})^2 \left|\langle \boldsymbol{q}|1s\rangle\right|^2 \qquad (2.33)$$

What we have found in (2.33) is, that the differential cross-section is $\frac{d\sigma}{d\Omega}$, describing the photoelectron intensity we expect to observe inside an angle element $d\Omega$ during the photoelectric process, is proportional to the Fourier transformation $\langle \boldsymbol{q}|1s\rangle$ of the initial state $|1s\rangle$ of the electron.
Employing this helpful relation theoretically estimated orbitals can be matched by transformation into momentum space and comparison with experimentally measured photoemission intensities. During this derivation, we assumed a hydrogen atom in ground state. However, this proportionality holds for more complex organic molecules as well [1, 12].

2.4 Methods of measurement

2.4.1 Photoemission spectroscopy

Photoemission spectroscopy (PES) is an umbrella term for several long-serving methods of analysis based on the photoelectric effect (see section 2.1) [21, 22]. Photoelectrons which are escaping from a surface sample are detected energy-resolved when irradiated by light resulting in a photoemission spectra as a function of the electron's kinetic energy. Using the spectra, there is information to be gained about the structure of electronic bands and their probability of occupation. Various types of PES are classified, among other factors, by the wavelength of the light employed, which in return determines the electrons that can be freed by it. Examples are X-ray photoelectron spectroscopy (XPS) and ultraviolet photoelectron spectroscopy (UPS). While the latter is used for valence electrons only, the former is capable of freeing inner shell electrons [7]. 1981 Kai Siegbahn (1918-2007) received half of the Nobel prize in physics for his work on high-resolution electron spectroscopy in 1957 [23, 24].

2.4.2 Angle-resolved photoemission spectroscopy

Based on PES, the so-called ***A****ngle-****R****esolved* ***P****hoto****E****mission* ***S****pectroscopy* developed during the 1970s [25–27]. It allows the mapping of Fermi surfaces by angle- and energy-resolved detection of photoelectrons. This is done by moving an energy analyser in a semicircle above the surface sample [28].

Beyond determining the orientation of thin layers, a quantitative evaluation of angle-dependent data resulting from ARPES measurements was considered to be too complicated until now. Photoemission tomography (see section 2.4.3) set out to solve this problem by matching experimental data with simulated data resulting from calculations using density functional theory (DFT) in a "plane wave" approximation [1, 29].

2.4.3 Photoemission Tomography

In the last few years an analysis method named *orbital tomography* emerged using the previously difficult to interpret data of ARPES.

A requirement for this method is the conservation of momentum components $\boldsymbol{k}_{||}$ parallel to the surface. Because of translational symmetry, those momentum components are preserved during the escaping process of the electron from the surface. Changing to spherical coordinates, we can describe the momentum of the electron inside the material as

$$\begin{aligned} |\boldsymbol{k}| &= \sqrt{\frac{2\epsilon_{kin} m_e}{\hbar^2}} \\ \boldsymbol{k}_x &= \sqrt{\frac{2\epsilon_{kin} m_e}{\hbar^2}} \sin(\theta)\cos(\varphi) \\ \boldsymbol{k}_y &= \sqrt{\frac{2\epsilon_{kin} m_e}{\hbar^2}} \sin(\theta)\sin(\varphi) \end{aligned} \tag{2.34}$$

Due to the potential step at the transition between surface and free vacuum state, the normal component $\boldsymbol{k}_\perp$ is not conserved. The intensity distribution $I(\epsilon_B,\ \{\boldsymbol{k}_x, \boldsymbol{k}_y\})$ measured at a constant binding energy ϵ_B, called momentum maps, is therefore a function of the binding energy ϵ_B and the parallel momentum components $\{\boldsymbol{k}_x,\ \boldsymbol{k}_y\}$ [1, 29].
First, the orbitals of the to-be analysed molecule will be calculated using DFT. The electron state after escaping the surface is approximated as a plane wave [1, 3]. DFT reduces the computation time for a many-body system significantly by representing all electrons as a single electron density $n(\boldsymbol{r})$ [1]. For his work on DFT, Walter Kohn (1923-2016) received 1998 part of the Nobel prize in chemistry [30].
As was shown in section 2.3, momentum maps (2.34) are proportional to the Fourier transform of the initial wavefunction of the electron before the photoelectric process. However, the experimental maps are measured for a specific energy at a time. Each map, therefore, represents a spherical "slice" in momentum space, hence the name orbital tomography. Likewise slicing through simulated momentum maps allows a comparison between these two. Such a comparison already yields valuable knowledge about the photoelectric process as well as the material and its electronic structure itself [3].
Reconstructing orbitals from experimental data alone presents another challenge. From (2.33) we learn that the intensity measured is proportional to the absolute square of the Fourier transformation. For an inverse Fourier transformation, which would enable us to switch back to position space, we lack a critical piece of information lost by absolute squaring the wavefunction, namely its phase.

To objectively match simulated and experimental data, we define a least-square cost function

$$\chi^2 = \int dk_x dk_y \left[I(\epsilon_B, k_x, k_y) - \sum_i a_i(\epsilon_B)\phi_i(k_x, k_y) \right]^2 \tag{2.35}$$

For the parameter $a_i(\epsilon_B)$, expressing the individual participation of each orbital $\phi_i(k_x, k_y)$, this ansatz leads to a solvable system of linear equation with a unique solution [3].
An additional set of parameters to be optimised gets introduced by the orientation of the molecules, characterised by Euler angles (φ, θ, ψ). There is no analytical solution of (2.35) for those parameters. Part of this work was implemented a simple algorithm finding the angles for which simulation and experiment agree best.

Chapter 3

Practical Work

The practical part of this work is concerned with the extension of functionality, improving the existing feature set as well as maintenance and troubleshooting a Python program (see figure 3.2) used for comparison of experimentally measured and simulated momentum maps.
In this chapter, we want to discuss various aspects of the practical side of this work. This ranges from the data used (see section 3.1), information about the already existing program (see section 3.2) and implementation done during this thesis (see section 3.3).

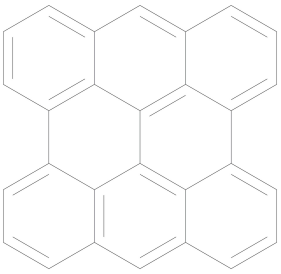

Figure 3.1: Structural formula of bisanthene (IUPAC: phenanthro(1,10,9,8-opqra)perylene).

3.1 Data

3.1.1 Simulation Data

The simulated data comes from a calculation using DFT (see section 2.4.3). For more information see the following sources [1, 5].

3.1.2 Experimental Data

The experimental data used in this in work originates from a measurement at the Metrology Light Source Insertion device beamline of the Physikalisch-Technische Bundesanstalt (Berlin, Germany). 10,10'-dibromo-9,9'-bianthracene has been polymerised to bisanthene (see figure 3.1) on a Cu(110) surface and excited by a photon energy of 35 eV. Escaping photoelectrons were detected by a toroidal energy analyser across a polar angle range of $\pm 85°$. The analyser was moved in $1°$ steps across an azimuthal angle range of $0°$ to $85°$. For more information see the following source for this paragraph [5].

3.2 Program Introduction

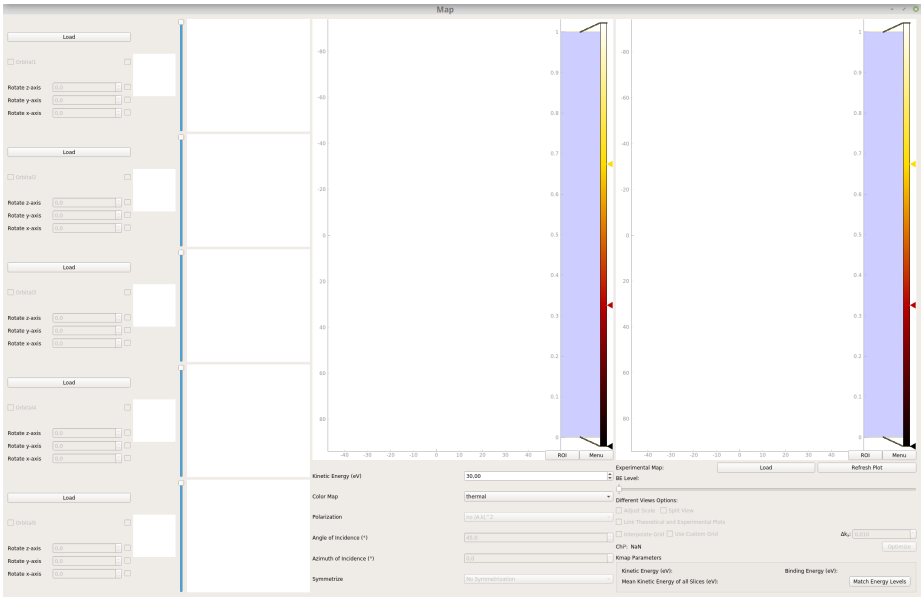

Figure 3.2: Screenshot of the Python program Map.py running under Linux (Mint 19) right after startup with no data loaded.
To the left five panes allow loading simulated momentum maps from different orbitals. Small ball and stick plots display the molecules loaded. QSpinBoxes and QSlider adjust the rotational orientation and orbital participation. To the right simulated and experimental momentum maps are displayed. Below are various settings and features for either one.

3.2.1 General information

In 2018, Peter Puschnig and Daniel Lüftner originally developed an Open-Source Python program (see section 3.2.2 and figure 3.2) under the GNU General Public License at the University of Graz and this work predominantly revolves around extending its functionality beyond the existing feature set (see section 3.3).
Its purpose lies in the presentation for and simplification of comparison between simulated and experimental momentum maps.

3.2.2 Setup

The program has initially been developed in Linux (Ubuntu), and Mint 19 (Tara) has exclusively been used during this work; however, it should work under different operating systems as well. It is entirely written in Python 3.7.1 utilising different third-party libraries (for a complete list see table 3.1). We used Git 2.17.1 as a version control systems (VCS) with an already employed remoted server hosted by GitLab. For the creation of graphical user interface (GUI) elements, we used Qt Creator 4.5.2 based on Qt 5.9.5.

Table 3.1: A complete list of third-party libraries used in the development process (see section 3.2). Python libraries are not listed. The first column lists the library name, the second column its version number and last column a short description for what we used the library.

Library	**Version**	**Describtion**
NumPy	1.15.4	Wrapper for dealing with numbers in N-dimensionals arrays.
SciPy	1.1.0	For interpolating momentum maps (see section 3.3.5).
PyQt	5.9.2	Provides everything regarding GUI.
PyQtGraph	0.10.0	
H5py	2.8.0	Used for interaction with the HDF5 file format.

3.3 Added Functionality

Hereafter functionality that has been added to the program within the scope of this work will be presented — beginning with section 3.3.1, a general overview of how the code is structured and the reasons behind it. Following sections "Binding energy slider" to "Fitting" address specific functionality in more detail. Rationale and benefits behind every feature will be handled under the subsection "Objective", a thorough description of what it does and how to utilise it is given under "Description and Functionality". Each section concludes with an illustration of how it was implemented code-wise under "Implementation". Section 3.3.9 addresses minor changes in less detail.

3.3.1 Structuring

At first, each method has been implemented by itself being called by, providing the corresponding functionality for and enabling each GUI element respectively. This simple approach came with hindrances as the feature set grew, and more complex interactions between methods arose. Following is a brief outline of the three most prominent difficulties this approach entailed:

- **Enablement of GUI elements**
  The program relies heavily on GUI elements, and for ease of use, those elements need to be disabled if their corresponding feature is unavailable for any reason (e.g. necessary data has not been loaded yet). This "state of enablement" can change many times during a session. Each method handling the state of its own GUI element, if it has one, is not only a clear violation of the single responsibility principle but also redundant.

- **Order of Execution**
  Different methods can influence each other and the order of execution matters when using more than one

  $$g\left(f\left(x\right)\right) \neq f\left(g\left(x\right)\right)$$

  As an example: if one changes the experimental momentum map by using the binding energy slider (see section 3.3.2) after adjusting the scale using the checkbox (see section 3.3.3), the slider will override the adjustment of scale. Either the checkbox will remain falsely checked, throwing the program into an unattainable state, or the user needs to redo his previous actions. The latter can cause additional problems when more functions are in use, and only one "correct" sequence works.
  Up to this point, it is the user's responsibility to use the methods in the correct order.

- **Undoing of Changes**
  Suppose a user is applying a function $f\left(x\right)$ and $g\left(x\right)$ in this order but undoes them in the reverse order. It cannot be guaranteed that the momentum map will be same afterwards

  $$y = g\left(f\left(x\right)\right) \rightarrow g^{-1}\left(f^{-1}\left(y\right)\right) \neq x$$

  Within this approach, it is the user's burden to remember the sequence of features performed and undoing them correctly.

Central to the features developed in this work are, therefore, the saving of data on a class level and two meta-functions "plot_manager"/"enable_manager" responsible for handling the order of execution as well as different cases (="states").

Saving Data

During this work, several new class member variables were introduced, mainly to address the third problem mentioned above. Grouped by their belonging to either the simulated or the experimental momentum maps, these variables store specific information/data of or about the momentum maps to be used by various methods. Besides the k_x range and the Δk step size needed by different functions, the unchanged data is saved as well. This allows returning to the unaltered maps anytime without running the risk to introduce artefacts by undoing changes. There are additional benefits specific to certain functionality. E.g. the binding energy slider does not need to reload the data file every time, or the optimisation can change the variable holding the image without changing the plot to save computation time.

States

Some methods newly introduced into the program during this work can be passed an additional parameter when called. This parameter labelled "state" is optional and defaults to zero. Its primary purpose is to lower computation time in the optimising process because specific steps can be skipped or altered to speed up the process (e.g. simulated momentum map does not need to be subject to any changes whatsoever). This was necessary because PyQt GUI elements are not capable of distinguishing the interaction done by the computer from "normal" interaction. Furthermore, this approach is very versatile as it allows for a multiple of "states" denoted by an integer.

Plot Manager

To address the second issue mentioned above, this "central hub" method was implemented. It manages every time either the experimental or simulated maps need to be refreshed by calling the responsible methods in order according to the state the program is in (see above). In the default state, it will call all functions in the default order:
The first method resets the simulated momentum map. Resetting the data ensures no artefacts get introduced by repeatedly applying certain features. This is not necessary for the experimental maps, because the second call, the binding energy slider, freshly loads the data corresponding to the position of the slider regardless.
Following are methods that will change the momentum maps (e.g. scale adjustment; see section 3.3.3) and methods performing miscellaneous task (e.g. linking plots; see section 3.3.4).
At this point the plot itself has not been updated. The last method to be called displays the new image data, resulting in changes visible for the user.

While at first, this may seem like an unnecessary complication, this approach offers significant benefits and, in fact, poses a simplification. As mentioned, it

was implemented as a solution to the second problem of handling the order of execution (see above). On top of that, it is a very future-proof way to organize multiple interacting methods. New features added in the future can be easily integrated by inserting a simple method call at the correct position in the order. If needed, a new state can be introduced to reorder the current sequence or make a new one altogether.

Enable Manager

To resolve the first problem listed above, a method named "enable_manager" was implemented. Called by the "plot_manager", it goes through all GUI elements and checks the requirements for each function individually. Using several if-statements, it enables GUI elements for which the necessary prerequisites are met, while it deactivates all others. These prerequisites are different for different elements and are grouped if possible.

While this approach did remove the additional responsibility of managing their respective GUI elements from each individual method, as is demanded by the principle of single responsibility, it is not a perfect solution. The elaborate lengthy tree of nested if-statements is "messy" and somewhat tricky to maintain. For every new function, from possibly different programmers, this method needs to be changed, not only added upon, as well.
Furthermore, within the implementation as it now stands, every function checks its requirements to be performed as well. This is an artefact of the prior implementation as well as a safeguard against in theory now impossible circumstances (function called without its requirements met). As is discussed in section 3.4, this redundancy is to be avoided in the future.

3.3.2 Binding Energy Slider

Objective

Experimental data in the project is organized in individual momentum maps of constant binding energy ("slices") due to the nature of the measurement (see section 2.4.2). To be able to go through all slices, a QSlider labelled "BE Level" has been added right beneath the loading button for the experimental data.

Description and Functionality

Loading (new) experimental data will update the slider to have as many discrete possible positions as there are slices in the data file. The slider will be placed to the far left, usually corresponding to the highest available binding energy, and pulling the slider to the right decreases the binding energy up to a minimum on the far right. The binding energy of the currently selected slice gets displayed at the bottom (see section 3.3.9).

Implementation

After checking if an experimental map has been loaded, the current position of the slider will be obtained by using a method provided by the QSlider class. Using this value, the data for the corresponding slice gets extracted, and the

variables (see section 3.3.1 get updated. Additionally, the kinetic energy of the current slice and the mean kinetic energy of all slice are calculated and displayed by subtracting the binding energy from the Fermi-level.

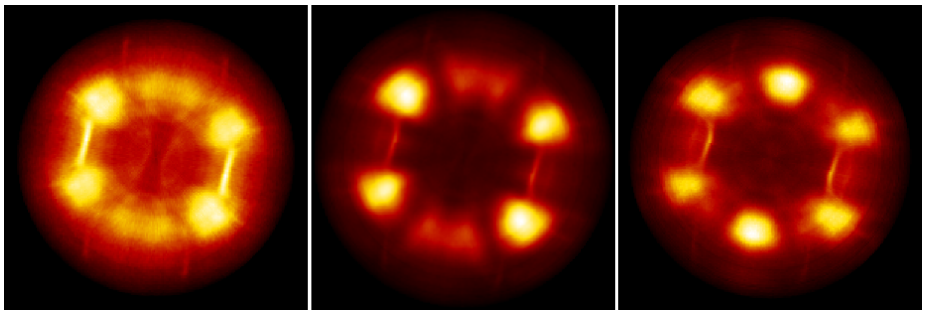

Figure 3.3: Example experimental momentum maps at different positions of the binding energy slider (see section 3.3.2). Part of the bisanthene data set (see section 3.1.2) for binding energies 0.99 eV (left), 0.50 eV (middle), 0.11 eV (right). Each momentum map equals a slice through momentum space for a specific wave number $|\boldsymbol{k}|$.

3.3.3 Rescaling and Background

Objective

Both the simulated data $I_{sim}(k_x, k_y)$ as well as the experimental data $I_{exp}(k_x, k_y)$ have arbitrary units. However, for a reasonable comparison between the two, one needs to adjust their scale. For this purpose, a QCheckBox labelled "Adjust Scale" has been added below the binding energy slider (see section 3.3.2).

Description and Functionality

The checkbox will be greyed out until both a simulated and an experimental data set has been loaded. Checking it will first calculate a quotient of the maximum intensity of both data sets

$$f = \frac{\max\left(I_{sim}(k_x, k_y)\right)}{\max\left(I_{exp}(k_x, k_y)\right)} \tag{3.1}$$

and then scale all experimental data by f

$$I_{exp}(k_x, k_y) \leftarrow f \times I_{exp}(k_x, k_y) \tag{3.2}$$

resulting in a changed experimental data image which highest intensity now matches the simulated highest intensity (still in arbitrary units). Additionally, the experimental data gets stripped of background noise by subtracting the now lowest intensity in the experimental data set from all values in this set

$$I_{exp}(k_x, k_y) \leftarrow I_{exp}(k_x, k_y) - \min\left(I_{exp}(k_x, k_y)\right) \tag{3.3}$$

Implementation

Subtracting the background and rescaling the experimental data has been implemented as two separate methods called by the plot manager (see section 3.3.1). This division of functionality, following the single responsibility principle, has been made because while both methods verify that experimental data has been loaded and the "Adjust Scale"-checkbox has been checked, only the rescale method checks additionally for loaded simulated data since subtracting the background is possible without (see equation (3.3)). Currently, it is not possible to subtract the background without also rescaling, and therefore, also loading simulated data. However implementing this feature now requires little effort, besides adding a new or changed QCheckBox.
Afterwards they apply (3.2) and (3.3) respectively.

3.3.4 Plot Linking

Objective

When looking at data sets, one may want to zoom in on finer details as well panning around the image while being close up. Meaningfully comparing details on both data sets requires to be on the same part of the image and at the same zoom. Therefore, as a convenience feature, a QCheckBox "Link Theoretical and Experimental Plots" has been added.

Description and Functionality

Zooming and panning in one of both plots will automatically be done as well in the other one when the checkbox is enabled. Enabling the checkbox is only possible if both simulated as well as experimental data have been loaded beforehand. Unchecking it will disconnect the plots again.

Implementation

It calls, when checked, the "setXLink"/"setYLink" method provided by the "ViewBox" class. Unchecking it will again call the same methods, however with an argument "None", this will undo the linking.

3.3.5 Interpolation

Objective

An essential step for core functionality implemented thereafter (see sections 3.3.6, 3.3.7 and 3.3.8) is the interpolation ("regridding") of data sets. This is necessary because for optimising the angle, one needs to calculate the difference of intensities between individual pixels when evaluating the cost function χ^2 (see section 3.3.7). Equation (2.35) necessitates an identical grid in both data sets, which is not necessarily the case.

Description and Functionality

If either one or both data sets get loaded the previously greyed out QCheckbox "Interpolate Grid" activates itself. Checking it will interpolate all loaded data

to a common grid. To make sure no information is lost, data sets get upscaled meaning as a grid size Δk the finer one is used

$$\Delta k \leftarrow \min(\Delta k_{exp}, \Delta k_{sim}) \tag{3.4}$$

Furthermore both data sets need to be cropped to the smaller of both ranges

$$\begin{aligned} k_x &\leftarrow \min(k_{exp;x}, k_{sim;x}) \\ k_y &\leftarrow \min(k_{exp;y}, k_{sim;y}) \\ \{[-k_{exp;x}, k_{exp;x}], [-k_{exp;y}, k_{exp;y}]\} &\leftarrow \{[-k_x, k_x], [-k_y, k_y]\} \\ \{[-k_{sim;x}, k_{sim;x}], [-k_{sim;y}, k_{sim;y}]\} &\leftarrow \{[-k_x, k_x], [-k_y, k_y]\} \end{aligned} \tag{3.5}$$

If only one data set has been loaded when activating this function, the program will not change the data set, since it already has the finer grid and smaller range. However this is a requirement for activating the "Use Custom Grid" QCheckBox to right. Checking this QCheckBox will enable the user to input a custom resolution Δk by using the QDoubleSpinBox to right. Until this point this greyed out Spinbox displayed the resolution Δk of the experimental data. Changing it's value overrides the previous behaviour of using the finer grid and forces all loaded maps to be regridded using the value entered by the user

$$\Delta k \leftarrow \Delta k_{custom} \tag{3.6}$$

Implementation

The SciPy package (see table 3.1) provides a method called "RegularGridInterpolator" used to implement this functionality. First, an interpolation object for each loaded map gets instanced by calling the method with the existing grids. Cropping the data sets has been separated into a different method, following the single responsibility principle. Using nested if statements, the method determines the smaller range in each direction separately and the finer grid, as well as adjusting the corresponding variables. If a custom value has been entered, the determination part will be skipped and the variables changed directly. With the NumPy package, a set of new grid points is generated using the new ranges and step sizes. Using this new grid and the interpolation objects created at the beginning, the data is interpolated at those grid points, and the images are updated.

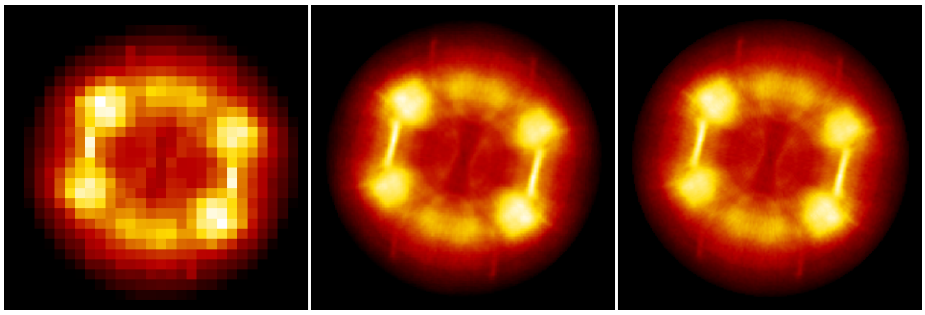

Figure 3.4: Example experimental momentum maps from the bisanthene data set (see section 3.1.2) for different interpolation scales (see section 3.3.5). Using the custom grid QSpinBox, data can be upscaled (right) or downscaled (left). The unaltered data (middle) is saved separately (see section 3.3.1) and can, therefore, be restored anytime without artefacts.

3.3.6 Split View

Objective

When comparing data sets or composing illustrative images, it can be valuable to view the data side by side.

Description and Functionality

Loading both a simulated and an experimental data set a QDoubleSpinBox called " Split View" becomes enabled. Checking this SpinBox will override the left half of the experimental data with the left half of the simulated data for side by side comparison (see figure 3.5).

Implementation

According to the "Plot Manager" (see section 3.3.1) this change will be performed after all necessary changes to the experimental data have been performed (i.e. adjusting scale; see section 3.3.3). Therefore changing the experimental data after activating split view will not impact the now left half (the simulated data). Moreover, the number of columns and rows of each data set need to be identical because the displaying method "setImage" provided by the "ImageView" class does not allow different sized images. Another possible solution would be padding the smaller image with "NaN" values. Code-wise, after checking that both momentum maps exist, the middle column is calculated by casting the division of the number of columns by two into an integer. Every column of the experiemental data set up to resulting index gets replaced by the simulated data's corresponding columns.

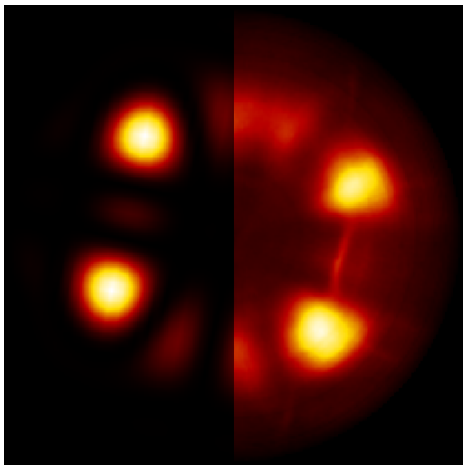

Figure 3.5: Example momentum maps from the bisanthene data set (see section 3.1.2) in split view (see section 3.3.6). Left half displays the simulated momentum map, on the right the experimental one. Using the automatic fitting function (see section 3.3.8) the optimal angle around the z-axes has been found to be 78° (map already turned; for unrotated data see first image in figure 3.6).

3.3.7 Least Square Function

Objective

An essential step in fitting the simulated data to experimentally determined momentum maps is comparing them objectively. The method of least squares is the standard approach to such problems.

Description and Functionality

Because the data is not continuous but consists of discrete steps Δk in the range of $\{[-k_x, k_x], [-k_y, k_y]\}$, the cost function (2.35) for comparing both sets needs to be discretized

$$\chi^2 = \sum_{k_y} \sum_{k_y} \left[I(\epsilon_B, k_x, k_y) - \sum_i a_i(\epsilon_B) \phi_i(k_x, k_y) \right]^2 \tag{3.7}$$

Equation (3.7) can only be meaningfully evaluated when both sets are of equal size and resolution, therefore only after checking the "Interpolate Grid" QCheckBox (see section 3.3.5) χ^2 will be calculated and displayed to the left of the "Optimize" button.

Implementation

Applying (3.7) is a straightforward matter using NumPy methods like "square" and "subtract" to perform standard operations on data matrices. Two minor issues quickly solved arise when doing so. First, one needs to catch potentially occurring "NaN" values in the data sets to avoid the entire sum becoming

"NaN". Again using a NumPy method called "nansum" resolves this issue by treating "NaN" values as zero. Second, from (3.7) it follows that χ^2 increases when interpolating data sets to a finer grid. This is an artefact of discretizing (2.35) into (3.7). It is impossible to fully avoid χ^2 changing when changing the step size Δk; however, normalizing χ^2 by dividing through the number of grid points is sufficient.

3.3.8 Fitting

Objective

Minimizing χ^2 (see section 3.3.7) now provides a straightforward method of tweaking various simulation parameters in such a way, that the simulated data maps agree with the experimental ones most. In principle, there are two sets of parameters open for fitting this way: orientation of the molecule(s) and participation of different orbitals. Only the former was part of this work and will be this discussed here. For details on the latter, known as deconvolution, see [3].

Description and Functionality

A fitting process triggers when clicking the QPushButton "Optimize" added for this reason to right of the χ^2 value. Until both simulated and experimental data have been loaded, this button will stay greyed out. During the calculation, the button's face text will change to "Wait", indicating that the process did not finish yet. Once it terminates, the face text will go back to "Optimize". The experimental data plot will only update once the optimal angle has been found to save unnecessary computation time. In theory, rotation around all three axes for the first orbital is available. However, to save currently unneeded computation time the code for rotating around the x and y axes have been disabled.

Implementation

To allow a colleague to work on the deconvolution problem separately, and to keep a maintainable code, three methods were introduced. Pushing the "Optimize" button will call a more general optimise method, responsible for controlling which parameters are fitted and in which order. It calls both methods for actually fitting the deconvolution as well as the orientation. However, as mentioned above, for this work, the deconvolution and two of three angles were commented out.

Due to this work coming to an end, only a simple, brute force algorithm for optimising the rotation has been implemented. Using a for loop, the cost function will be evaluated for each fixed step size between in a fixed range of values. Each loop an if statement compares the result of (3.7) with a variable that persists the for loop. If smaller, the variable will be overwritten with the result, otherwise nothing happens. Saving the corresponding angle every time a smaller χ^2 was found makes sure the for loop ends with the optimal angle.

This approach has benefits and drawbacks.
Because this work was close to an end, this straightforward approach easily

implemented was chosen to provide working functionality quickly and show a proof of concept. Furthermore, this algorithm is very robust as it will always terminate and has no issues with local minima.
Those benefits come at a cost. This strategy provides minimal accuracy for a high computational cost and scales very steep with the number of evaluation points. Furthermore, because (3.7) gets only evaluated at fixed, predefined step sizes, there is no guarantee to find the absolute minima within a $\pm\epsilon$ range.

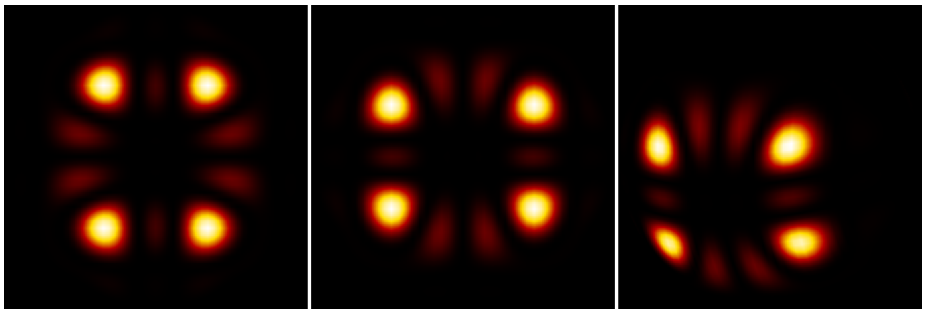

Figure 3.6: Example experimental momentum maps from the bisanthene data set (see section 3.1.2) at different orientations. The original data (left) was rotated 90° around the z-axes (middle) and additionally 25° around the x- and y-axes respectively (right). To automatize the finding of angles for which the simulation fits the experiment best was the central part of this thesis (see section 3.3.8; see figure 3.5).

3.3.9 Miscellaneous

Additionally, to the implemented features mentioned above, some smaller changes and fixes were done. Those shall be listed here briefly:

- **Displaying experimental data values**
  When loading experimental data the binding energy ϵ_B of the current slice, the kinetic energy ϵ_{kin} of the current, as well as the mean of all slices in [eV] will be displayed at the bottom right. The kinetic energy is calculated by subtracting the binding energy ϵ_B from the fermi level ϵ_F.

- **Matching energy levels**
  At the bottom right, a QPushButton labelled "Match Energy Levels" will, when experimental data has been loaded, take the mean kinetic energy of all slices as input for the simulated data in the corresponding QDoubleSpinBox.

- **Implementation of various checkboxes for future use**
  A couple of currently unused QCheckBox were implemented for future use. Checking them will select which values should be fixed in case of automatic optimisation. Furthermore, tooltips for those QCheckBox were implemented.

- **Prohibited loading wrong file types to avoid crashing**

3.4 Future Tasks

In this section, a short preview regarding this program's future shall serve us as a conclusion of this thesis.

A problem already mentioned in section 3.3.1, regarding the redundancy of requirement checks, is to be tackled. However, the expected gain of computational time by eliminating this redundancy should not be overestimated. Those checks generally only consist of inexpensive if-statements.
The removal of the enable manager serves a different purpose nevertheless. It is, as it stands now, a suboptimal solution violating core principles of clean code (e.g. encapsulation). A reasonable approach would be to outsource the requirement checks for each feature into a separate method returning a boolean whether the requirements are met. This would increase the number of methods drastically; however, essential design principles for clean code would be met.

As was mentioned in section 3.3.8, only a rudimentary algorithm with hard-coded angle range and angle steps was implemented up to this point. This approach is not only costly and inefficient but also challenging to use.
More advanced algorithms (e.g. brent's algorithm), a variable step size (automatically decreasing the step size when near a minimum) or iterative approaches (multiple searches with decreasing step size in regions containing a minimum) could address the former issue.
To improve usability, a couple of QCheckBoxes were already implemented. They can be used to fix individual values, dramatically decreasing the computational time. Additionally, QSpinBoxes to specify search ranges, as well as step sizes, are needed.

A core issue arising from all solutions mentioned above is that the entire GUI necessitates a design overhaul. After its initial creation, several features have been subject to change or have been newly implemented. To accommodate those features, as well as the ones mentioned above (e.g. new QSpinBoxes for each rotation) in a user-friendly and space-saving manner, a redesign would be beneficial.

List of Figures

List of Tables

Bibliography

[1] D. Lüftner, *Orbital tomography: Understanding photoemission of organic molecular films*. PhD thesis, University of Graz, 2015.

[2] J. M. Zhang and Y. Liu, "Fermi's golden rule: its derivation and breakdown by an ideal model," *European Journal of Physics*, vol. 37, no. 6, p. 065406, 2016.

[3] P. Puschnig, E. M. Reinisch, T. Ules, G. Koller, S. Soubatch, M. Ostler, L. Romaner, F. S. Tautz, C. Ambrosch-Draxl, and M. G. Ramsey, "Orbital tomography: Deconvoluting photoemission spectra of organic molecules," *Phys. Rev. B*, vol. 84, DEC 15 2011.

[4] L. Broekman, A. Tadich, E. Huwald, J. Riley, R. Leckey, T. Seyller, K. Emtsev, and L. Ley, "First results from a second generation toroidal electron spectrometer," *Journal of Electron Spectroscopy and Related Phenomena*, vol. 144-147, pp. 1001 – 1004, 2005. Proceeding of the Fourteenth International Conference on Vacuum Ultraviolet Radiation Physics.

[5] X. Yang, L. Egger, P. Hurdax, H. Kaser, D. Lüftner, F. C. Bocquet, G. Koller, A. Gottwald, P. Tegeder, M. Richter, M. G. Ramsey, P. Puschnig, S. Soubatch, and F. S. Tautz, "Identifying surface reaction intermediates with photoemission tomography," *Nature Communications*, vol. 10, July 2019.

[6] P. Puschnig, S. Berkebile, A. J. Fleming, G. Koller, K. Emtsev, T. Seyller, J. D. Riley, C. Ambrosch-Draxl, F. P. Netzer, and M. G. Ramsey, "Reconstruction of molecular orbital densities from photoemission data," *Science*, vol. 326, no. 5953, pp. 702–706, 2009.

[7] W. Demtröder, *Experimentalphysik 3 Atome, Moleküle und Festkörper*. Springer Spektrum, 5 ed., 2016.

[8] "The Nobel Prize in Physics 1921." https://www.nobelprize.org/prizes/physics/1921/einstein/biographical/, 2019. Online; accessed 30-July-2019.

[9] A. Einstein, "Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt," *Annalen der Physik*, vol. 322, no. 6, pp. 132–148, 1905.

[10] "Max Planck Facts." https://www.nobelprize.org/prizes/physics/1918/planck/facts/, 2019. Online; accessed 30-July-2019.

[11] K. Krane, *Atom- Kern- und Teilchenphysik an den Grazer Universitaeten*. Wiley Custom, 3 ed., 2012.

[12] T. Fließbach, *Quantenmechanik Lehrbuch zur Theoretischen Physik III*. Springer Spektrum, 5 ed., 2008.

[13] D. J. Griffiths, *Introduction to Quantum Mechanics*. Prentice Hall, 1994.

[14] "Paul A.M. Dirac Biographical." https://www.nobelprize.org/prizes /physics/1933/dirac/biographical/, 2019. Online; accessed 30-July-2019.

[15] P. A. M. Dirac, "The quantum theory of the emission and absorbtion of radiation," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 114, pp. 243–265, 3 1927.

[16] E. Fermi, *Nuclear Physics: A Course Given*. University of Chicago, 1949-1950.

[17] "Enrico Fermi Facts." https://www.nobelprize.org/prizes/physics/1938 /fermi/facts/, 2019. Online; accessed 30-July-2019.

[18] T. Fließbach and H. Walliser, *Arbeitsbuch zur Theoretischen Physik Repetitorium und Übungsbuch*. Springer Spektrum, 3 ed., 2012.

[19] P. Hadley, "The Hamiltonian of a charged particle in a magnetic field." https://lampx.tugraz.at/ hadley/ss1/IQHE/cpimf.php. Online; accessed 30-July-2019.

[20] T. Fließbach, *Mechanik Lehrbuch zur Theoretischen Physik I*. Springer Spektrum, 7 ed., 2015.

[21] A. Damascelli, Z. Hussain, and Z.-X. Shen, "Angle-resolved photoemission studies of the cuprate superconductors," *Rev. Mod. Phys.*, vol. 75, pp. 473–541, Apr 2003.

[22] T. A. Carlson, "Photoelectron spectroscopy," *Annual Review of Physical Chemistry*, vol. 26, no. 1, pp. 211–234, 1975.

[23] C. Nordling, E. Sokolowski, and K. Siegbahn, "Precision method for obtaining absolute values of atomic binding energies," *Phys. Rev.*, vol. 105, pp. 1676–1677, Mar 1957.

[24] "The Nobel Prize in Physics 1981." https://www.nobelprize.org/prizes /physics/1981/summary/, 2019. Online; accessed 30-July-2019.

[25] K. Shen, "Ultraviolet photoemission spectroscopy (ups) for in situ characterization of thin film growth," *In Situ Characterization of Thin Film Growth*, pp. 55–74, 10 2011.

[26] E. O. Kane, "Implications of crystal momentum conservation in photoelectric emission for band-structure measurements," *Physical Review Letters - PHYS REV LETT*, vol. 12, pp. 97–98, 01 1964.

[27] M. M. Traum, N. V. Smith, and F. J. Di Salvo, "Angular dependence of photoemission and atomic orbitals in the layer compound $1T - TaSe_2$," *Phys. Rev. Lett.*, vol. 32, pp. 1241–1244, Jun 1974.

[28] A. Stampfl, J. Foo, R. Leckey, J. Riley, R. Denecke, and L. Ley, "Mapping the Fermi surface of Cu using ARUPS," *Surface Science*, vol. 331-333, pp. 1272 – 1276, 1995. Proceedings of the 14th European Conference on Surface Science.

[29] G. Koller, P. Puschnig, A. Gottwald, and F. S. Tautz, "Elektronenorbitale in 3d," *Physik in unserer Zeit*, vol. 47, no. 4, pp. 192–198, 2016.

[30] "Walter Kohn Biographical." https://www.nobelprize.org/prizes/chemistry/1998/kohn/biographical/, 2019. Online; accessed 31-July-2019.