

UNIVERSITY OF GRAZ

Imaging molecular orbitals for ARPES simulations

by

Emil Parth

01510345

A thesis submitted in partial fulfillment for the
degree of Bachelor Of Science

to the
Institute of Physics

Supervisor: Assoc.-Prof. Dr. Peter Puschnig

January 2019

Contents

1	Introduction	1
2	Theory	2
2.1	Experiment	2
2.2	Density Functional Theory	3
2.3	XSF files	6
3	Method and Results	8
3.1	The benzene ring	8
3.2	Imaging atoms and bonds	9
3.3	Plotting molecular orbitals in real space	12
3.4	Plotting the orbitals in momentum space	14
4	Conclusion	18

Chapter 1

Introduction

Angle resolved photoemission spectroscopy (ARPES) is used to study the electronic structure of surfaces and is based on the photoelectric effect which has been first explained by Einstein 1905[1]. The quantum mechanical description of the photoemission process can be achieved within the one step model of photoemission. A particularly simple approach is to approximate the final state as a plane wave [2][3]. Under illumination of monochromatic light, the electrons detected as function of angles and energy correspond to the Fourier transformed initial state wavefunction of the observed atomic structure. By this method is possible to directly measure ground state orbitals of molecules. The wavefunction can be calculated theoretically via the so called *density functional theory* [4].

This work describes the implementation of a *python* script to calculate orbitals in momentum space and imaging these as an ARPES simulation. The program generates images of the molecular structure, the wavefunction and ground state orbitals in momentum space for a pre defined experimental setup. The input is a XSF file (internal *XCrySDen* structure format) [5] in txt format used to store atomic data of molecules and crystals.

Chapter 2

Theory

2.1 Experiment

With angle-resolved photoemission spectroscopy (ARPES), the electronic structure of surfaces can be directly measured. It is based on the photoelectric effect [1]. The sample is therefore illuminated with monochromatic light of the energy $h\nu$. The kinetic energy of an electron escaping a material by absorbing a photon is given by [2]:

$$E_{kin} = h\nu - \Phi - E_B \quad (2.1)$$

E_{kin} is the maximum kinetic energy of the electron [eV], Φ is the workfunction [eV] and denotes the binding energy E_B [eV]. The energy of the free electron approximated by a plane wave is given by:

$$E_{kin} = \frac{\hbar^2 k^2}{2m_e} \quad (2.2)$$

where m_e is the electron mass [eV].

In ARPES experiments, the emitted electrons are detected in terms of their kinetic energy as well as their emission angle [2].

The parallel momenta components are obtained by the following equations [2], where Θ and Φ are the polar and azimuthal emission angles as indicated in 2.1:

$$k_x = \sqrt{\frac{2m_e \cdot E_{kin}}{\hbar}} \sin \Theta \cos \Phi \quad (2.3)$$

$$k_y = \sqrt{\frac{2m_e \cdot E_{kin}}{\hbar}} \sin \Theta \sin \Phi \quad (2.4)$$

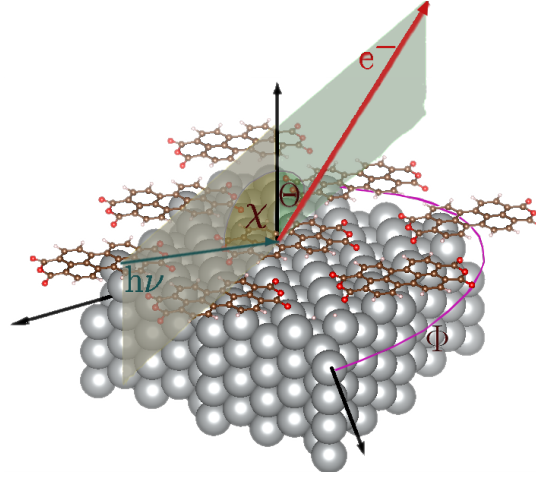


FIGURE 2.1: experiment

Within the one step model of photoemission, the photo-excitation is treated as a single coherent process, from bound state $|\psi_i\rangle$ with energy E_i , to the free state $|\psi_f\rangle$ with energy E_f and parallel momentum components k_x and k_y ([3]). When assuming a dipole transition, the intensity distribution $I(E, k_x, k_y)$ of the photoemitted electrons is given by Fermi's golden rule expression [2]

$$I(E, k_x, k_y) \propto \sum_i \left| \langle \psi_f(E, k_x, k_y) | \vec{A} \cdot \vec{\nabla} | \psi_i \rangle \right|^2 \times \delta(E_i + \Phi + E - \hbar\omega) \quad (2.5)$$

Where \vec{A} is the polarization vector of the incoming photon. By assuming a plane wave with vector \vec{k} as the final state $|\psi_f\rangle$ the photocurrent I_i becomes proportional to the Fourier transform of the initial wave function $\tilde{\psi}_i$ corrected by a polarization factor [3].

$$I_i(E, k_x, k_y) \propto |\vec{A} \cdot \vec{k}|^2 \times |\tilde{\psi}_i(k_x, k_y)|^2 \quad (2.6)$$

From a certain constant binding energy E_B , $|\vec{k}|$ is a constant. Therefore the Fourier transformed wavefunction is $\tilde{\psi}_i$ evaluated on a sphere with radius $|\vec{k}|$. This can be imaged as a cut of a sphere through momentum space representation of the initial state orbital (see figure 2.2).

2.2 Density Functional Theory

To calculate the energies and eigenfunctions of a molecule, the *Schroedinger Equation* is solved.

$$\hat{H} |\Psi(\vec{R}_M, \vec{r}_n)\rangle = E |\Psi(\vec{R}_M, \vec{r}_n)\rangle \quad (2.7)$$

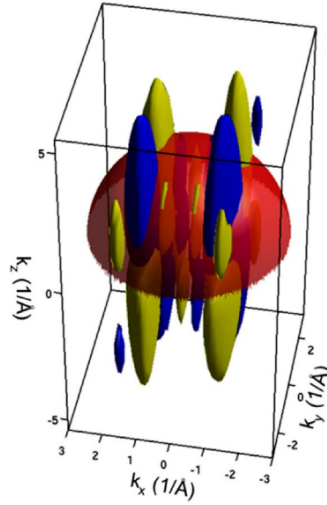


FIGURE 2.2: sphere cut trough orbitals [3]

We consider the Hamiltonian of a molecule with n electrons and M nuclei of Z_J protons:

$$H = -\frac{\hbar^2}{2m_e} \sum_{j=1}^n \nabla_j^2 - \frac{\hbar^2}{2} \sum_{J=1}^M \frac{\nabla_J^2}{m_p} - \sum_{j=1}^n \sum_{K=1}^M \frac{Z_K e^2}{4\pi\epsilon_0 |\vec{r}_j - \vec{R}_K|} + \sum_{j=1}^n \sum_{k>j}^n \frac{e^2}{4\pi\epsilon_0 |\vec{r}_{jk}|} + \sum_{J=1}^M \sum_{K>J}^M \frac{Z_J Z_K e^2}{4\pi\epsilon_0 |\vec{R}_{JK}|} \quad (2.8)$$

The first term describes the kinetic energy of the electrons, the second one the kinetic energy of the nuclei. The third sum describes the Coulomb interaction of nuclei and electrons, the fourth the electron-electron and the third the nuclei-nuclei interaction.

The length of this expression gives an impression on the difficulty of solving the corresponding *Schroödinger Equation*. A first step in solving it, is to make a few assumptions. 'Atomic nuclei are much heavier than individual electrons. Each proton or neutron in a nucleus has more than 1800 times the mass of an electron. This means, roughly speaking, that electrons respond much more rapidly to changes in their surroundings than nuclei can.'[4].

The so called *Born-Oppenheimer approximation* uses this observation to split the equation into two pieces. First it is solved for the groundstate energy $\vec{E}(\vec{R}_1 \dots \vec{R}_M)$ of fixed nuclei positions via the *Schödinger Equation*. This decouples the equation into one only dependent on the electron positions.

$$\left[\frac{\hbar^2}{2m} \sum_{i=1}^N \nabla_i^2 + \sum_{i=1}^N V(\vec{r}_i) + \sum_{i=1}^N \sum_{j<i} U(\vec{r}_i, \vec{r}_j) \right] |\Psi(\vec{r}_n)\rangle = E |\Psi(\vec{r}_n)\rangle \quad (2.9)$$

This Hamiltonian describes a set of electrons moving in a fixed potential from the nuclei, the first term describes the kinetic, the second the electron-nuclei and the third the electron-electron energy.

The n-electron wavefunction $\psi(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N)$ is a function of the $3n$ spatial coordinates. The numerical complexity of this problem scales exponentially with the number of electrons n [6]. A theory based on the electron density $n(\vec{r})$ would reduce this to a numerical complexity to n^3 . The so-called density functional theory takes on to this approach.

One of the fundamental theorems of the DFT proven by Kohn and Hohenberg states that: *The ground-state energy from Schrödinger equation is a unique functional of the electron density* [4].

This can be described as followed:

$$E[\{\Psi_i\}] = E_{known}[\{\Psi_i\}] + E_{XC}[\{\Psi_i\}] \quad (2.10)$$

where

$$E_{known}[\{\Psi_i\}] = -\frac{\hbar^2}{2m} \sum_i \int \Psi_i^* \nabla^2 \Psi_i d^3r + \int V(\vec{r}) n(\vec{r}) d^3r + \frac{e^2}{2} \int \frac{n(\vec{r}) n(\vec{r}')}{|\vec{r} - \vec{r}'|} d^3r d^3r' + E_{ion} \quad (2.11)$$

The terms describe in order the kinetic energies, electron nuclei Coulomb interaction and interaction of the pairs of electrons. The other functional E_{XC} describes the exchange-correlation function [4].

The second fundamental theorem of Hohenberg and Kohn states that: *The electron density that minimizes the energy of the overall functional is the true electron density corresponding to the full solution of the Schrödinger equation.*

Kohn and Sham showed that the electron density can be calculated by solving a set of decoupled single electron equations, the so called *Kohn-Sham equations*:

$$\left[-\frac{\hbar^2}{2m} \nabla^2 + V(\vec{r}) + V_H(\vec{r}) + V_{XC}(\vec{r}) \right] \phi_i(\vec{r}) = \epsilon_i \phi_i(\vec{r}) \quad (2.12)$$

It contains an effective potential comprised of three contributions V , V_H and V_{XC} . The first of these also appeared in the E_{known} part of the total energy functional given above (Eq. (13)). It describes the interaction of an electron and a fixed set of nuclei.

The so called Hartree potential V_H describes the Coulomb repulsion between the electron

of the Kohn–Sham equations and the total electron density.

$$V_H(\vec{r}) = e^2 \int \frac{n(\vec{r}')}{|\vec{r} - \vec{r}'|} d^3r' \quad (2.13)$$

V_{EX} defines exchange and correlation contribution to the single electron equations. It can be formally be defined as a functional derivative of the exchange correlation energy [4].

$$V_{EX} = \frac{\delta E_{EX}(\vec{r})}{\delta n(\vec{r})} \quad (2.14)$$

To solve these equations, an iterative process needs to be implemented.

1. An initial electron density $n(\vec{r})$ is guessed (e.g. via superposition of atomic densities).
2. The Kohn-Sham equations are solved to find the single particle wave functions $\phi_i(\vec{r})$.
3. The single particle wave functions are used to recalculate the electron density $n_{KS}(\vec{r}) = 2 \sum_i \phi_i^*(\vec{r}) \phi_i(\vec{r})$.
4. the so obtained new electron density is the reused in 2. This self-consistent circle ends, when the obtained electron density matches the used one.

2.3 XSF files

The xsf file (internal *XCrySDen* structure format) [5] is a txt format to store atomic data of molecules and crystals. The main attributes are:

- all records are in free format
- the XSF formatted file is composed from various sections
- each sections begins with the keyword
- there are two types of keywords: (i) single keywords, and (ii) sandwich keywords, which are defined as:
 - single keyword: section begins with a single keyword and ends without an end-keyword
 - sandwich keyword: section begins with a begin-keyword (i.e. **BEGIN_keyword**) and ends with an end-keyword (i.e. **END_keyword**), where keyword is one among keywords.
- all coordinates are in ANGSTROMS units
- all forces are in Hartree/ANGSTROM units

- the comment-lines start with the “#” character (see below)

It contains the crucial informations of the molecule such as the positions of all atoms, their atomic number, the wave-function etc. The text is therefore marked with keyword to indicate the subsequently listed information. The keywords consist of single words only. Commented lines are marked with # . The sequence of information after each keyword is unique and must be in exact order. An example is given below.

```
# -0.0069489594686865 0.0000000000000000
```

```
DIM-GROUP
```

```
3 1
```

```
PRIMVEC
```

```
1.0371873288E+01 0.0000000000E+00 0.0000000000E+00
```

```
0.0000000000E+00 1.1006885939E+01 0.0000000000E+00
```

```
0.0000000000E+00 0.0000000000E+00 6.0008695454E+00
```

```
PRIMCOORD
```

```
12 1
```

```
6 6.4011395272E+00 6.2052261853E+00 3.0004347727E+00
```

```
6 5.1859366442E+00 6.9073479094E+00 3.0004347727E+00
```

```
6 6.4011395272E+00 4.8016597533E+00 3.0004347727E+00
```

```
6 5.1859366442E+00 4.0995380293E+00 3.0004347727E+00
```

```
...
```

Chapter 3

Method and Results

3.1 The benzene ring

In this work the molecule benzene is used as an example. It is a planar molecule consisting of 6 carbon and 6 hydrogen atoms arranged in a ring with equivalent distance between the carbon atoms. Each carbon atom additionally binds 1 hydrogen atom, resulting in sp^2 hybridization.

The carbon atoms have 4 valence electrons in the $2s^2, 2p^2$ shells. hydrogen has one valence electron in the 1s shell. This corresponds to $6 \cdot 4 + 6 = 30$ valence electrons for the entire molecule. Due to spin occupation there are 15 orbitals to consider, 3 of which are of π symmetry.

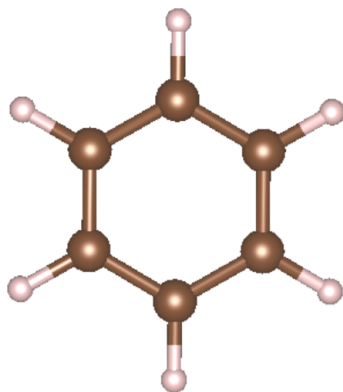


FIGURE 3.1: C_6H_6

All-together there are 4 valence electrons per carbon atom. 3 of there electrons (+ the 1s electron 4) make the sigma $s(\sigma)$ -bonds. The fourth electron is in a p_z orbital and forms π -bonds. So there are 6 π -electrons and 6 corresponding π -orbitals (3 of which occupied).

The σ bonds between each atom are of little interest here.

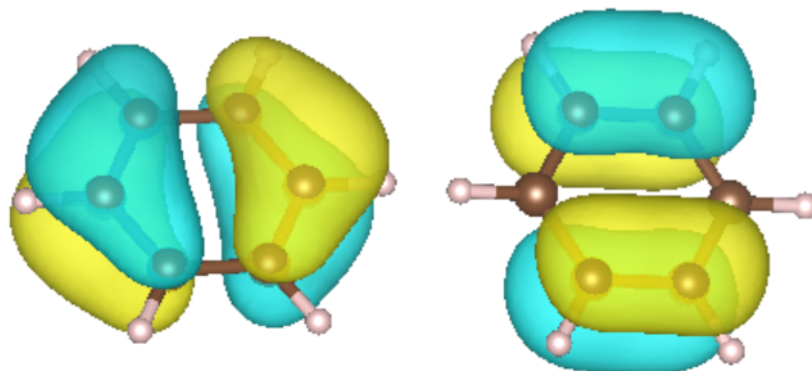


FIGURE 3.2: π_2 (left) and π_1 (righth) orbital of C_6H_6

3.2 Imaging atoms and bonds

The input of our program is an xsf (internal *XCrySDen* structure format) file. The file is accessible as txt-file and read line by line. After a keyword shows up, the following content is assigned to specific variables. Due to the size of these files, this may take a while.

The file is accessed via the open function:

```
with open(fname3,'r') as inFile:
    #enter reading routine
    data = inFile.readline()
    for i, data in enumerate(inFile):
```

where '*data*' is a character array with the current line. First the atoms have to be plotted.

However the definition of the size of atoms is quite problematic since its *effective size* changes for binding types and according to quantum-mechanics has no limit in space. For atoms forming covalent bindings, which is the case here, typically the *covalent radius* is used. It is listed in tables for the atoms used [7].

The positions of the atoms and their atomic number is headed with the keyword **ATOMS**. The following lines consist of the atomic number, and the x,y,z positions of the atoms similar to the content listed after the keyword 'PRIMCOORD' shown in the example of 2.3.

```

r=np.zeros(len(atomData[0]))
for i in range(0,len(atomData[0])):
    r[i]=coreSizeFac*kovRad(int(atomData[0][i]))*0.01

# Make data
u = np.linspace(0, 2 * np.pi, 20)
v = np.linspace(0, np.pi, 20)
x = 1 * np.outer(np.cos(u), np.sin(v))
y = 1 * np.outer(np.sin(u), np.sin(v))
z = 1 * np.outer(np.ones(np.size(u)), np.cos(v))

# Plot the surface
for i in range(0,len(atomData[0])):
    ax0.plot_surface(x*r[i]+atomData[1][i], y*r[i]+atomData[2][i], z*r[i]+atomData[3][i], color=

```

'atomData' 1-3 stores the x,y,z positions of the atoms in Angstrom, *atomData*[0] contains the atomic number. The atomic number is compared to a table of covalent radii corresponding to the atomic number which is realized in the function *kovRad()*. The covalent radius is given in the range of (32[H]-272[Cs]) 10^{-12} m. Since this definition is useful to understand the sizes of atoms inside the molecule, a plot with pure covalent radii would leave very few space to visualize orbitals:

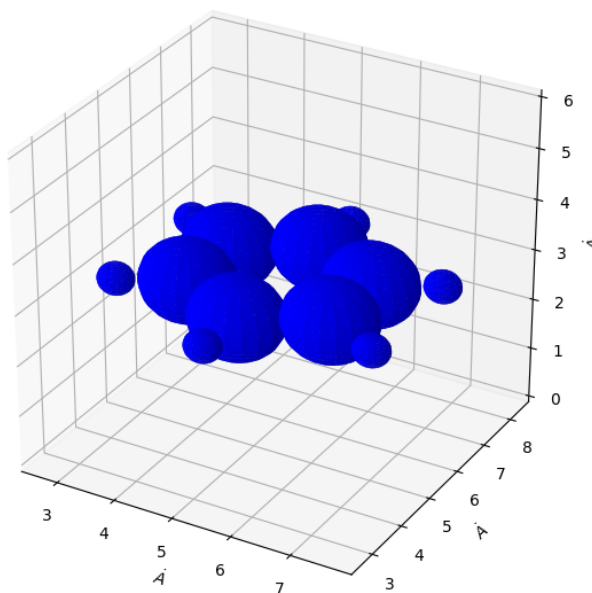


FIGURE 3.3: C_6H_6 with covalent radii

Because of the relatively large covalent radii the radius is scaled with a factor (*coreSizeFac*) $\in [0, 1]$. Relations between the volumes of the different atoms stay proper. The x,y,z axis of figure 3.3 are scaled in *Angstrom*.

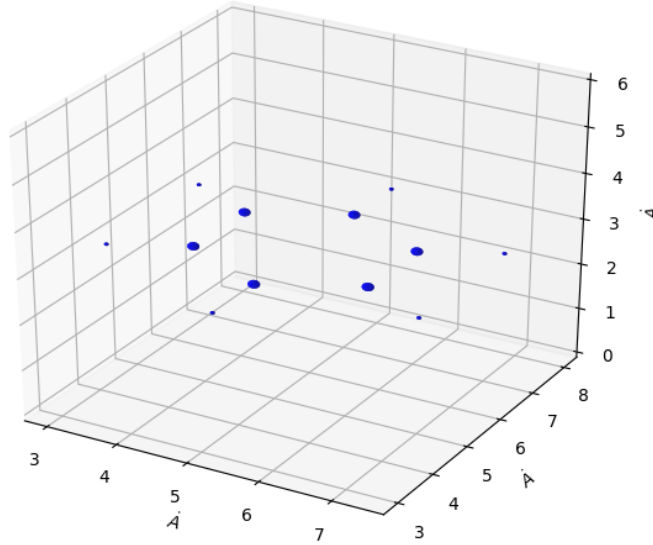


FIGURE 3.4: Figure of atoms, with covalent radii-0.1

To indicate the molecular bonds, the nuclei are connected via lines. Therefore the shortest distance d_{min} between one atom and any other atom is found. Every atom in the range of $d_{min} \cdot f$ (where f is a preset factor) is considered a binding partner of this atom.

$$d_n = \{d \in D : d < \min(D) \cdot f\} \quad (3.1)$$

d_n with distance of partner in [m], f being the preset factor and D being the amount of all atomic distances in [m]

This is implemented in the following routine:

```
def findNeighbours(atomPos, allAtomPos):
    neighbours=[]
    minDist=dist(atomPos, allAtomPos[0])
    for i in range(0, len(allAtomPos)):
        if (minDist > dist(atomPos, allAtomPos[i]) and dist(atomPos, allAtomPos[i]) > 0):
            minDist=dist(atomPos, allAtomPos[i])
    for i in range(0, len(allAtomPos)):
        if (minDist*boundDist >= dist(atomPos, allAtomPos[i])):
            neighbours.append(allAtomPos[i])
    return neighbours
```

Here the variable '*boundDist*' is the preset factor and set to 1.3.

Subsequently lines are drawn to connect the nuclei. In the implementation below a 2- or 3-dimensional molecule was considered. Later the sourcecode was only adapted to 3-dimensional molecules.

```

for i in range(0,len(atomPos)):
    pos=(findNeighbours(atomPos[i],atomPos))
    for j in range(0,len(pos)):
        if(len(pos[j])<3):
            ax0.plot([atomPos[i][0],pos[j][0]],[atomPos[i][1],pos[j][1]],
                    ,[0,0],c='k',linewidth=1.0)

        if(len(pos[j])==3):
            ax0.plot([atomPos[i][0],pos[j][0]],[atomPos[i][1],pos[j][1]],
                    [atomPos[i][2],pos[j][2]],c='k',linewidth=boundThickness)

```

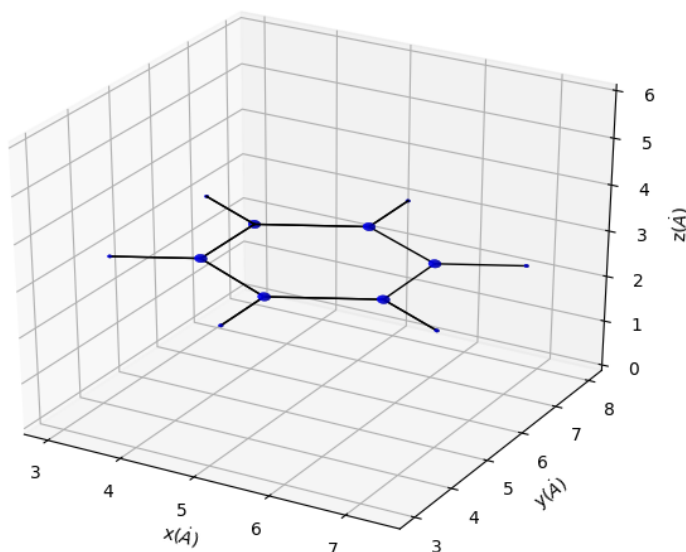


FIGURE 3.5: Figure of molecule

3.3 Plotting molecular orbitals in real space

The wavefunction is stored in a table of probabilities in the xsf-file. The keyword '*DATAGRID_3D_DENSITY*' indicates this section. The first line afterward shows how many variables there are for each direction (n_x, n_y, n_z), followed by one line with the origin of the datagrid and three lines dedicated to the direction (lattice vectors). An example is given below:

DATAGRID_3D_DENSITY

80 90 48

0.0 0.0 0.0

1.0371873288E+01 0.0000000000E+00 0.0000000000E+00

0.0000000000E+00 1.1006885939E+01 0.0000000000E+00

0.0000000000E+00 0.0000000000E+00 6.0008695454E+00

-1.5971783003E-10

-1.9233087226E-10

-2.2758253051E-10

-2.3961219969E-10

...

The first probability is of the gridpoint (0,0,0). The gridpoints are sorted so that x variables are counted first (a,0,0). After the x tuple reaches $((n_x-1),0,0)$ the y-tuple is increased by 1 and the x tuple is reset to 0.

$$((n_x - 1), 0, 0) \rightarrow (0, 1, 0) \quad (3.2)$$

The same rule continues for the z tuple.

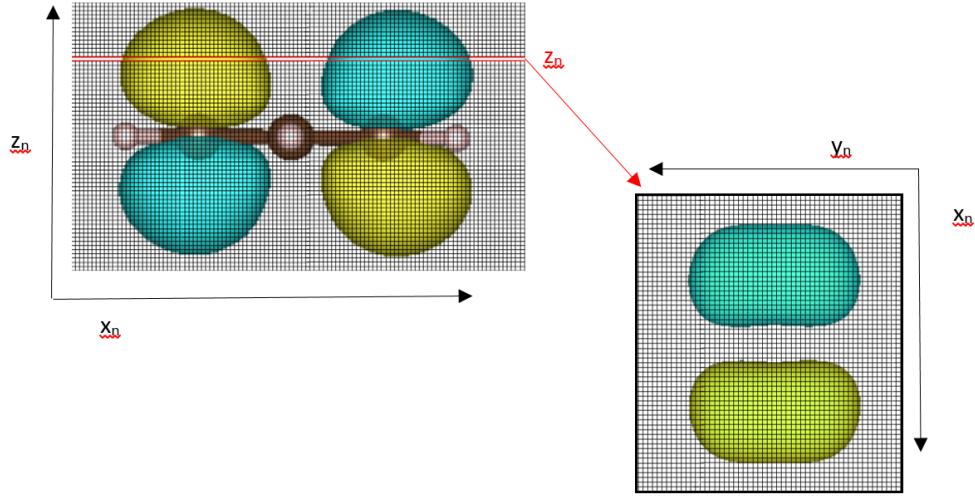
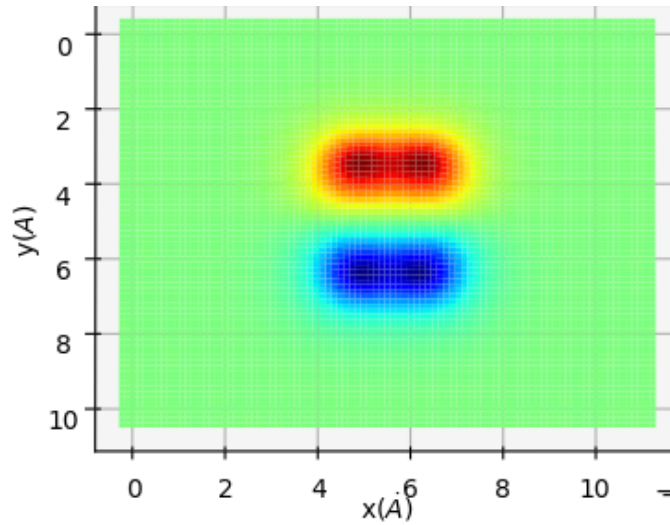
$$((n_x - 1), (n_y - 1), 0) \rightarrow (0, 0, 1) \quad (3.3)$$

This very intuitive sequence can be easily read in with 3-loops over the x,y and z variables. (arrays in python start at 0, thus n_x-1)

The orbital is then plotted via the 'plot_surface' function by choosing a z-plane to plot a 2-dimensional cut of the wavefunction. A reasonable choice would therefore be somewhere around $(\frac{n_z}{2} \pm \frac{n_z}{4})$, since the plane of the molecule is located at $\frac{n_z}{2}$ for this example file.

This can be manually changed for different cuts later. The origin of the coordinate system is at (0,0,0) and has the maximum lengths ($x_{max}=10.3$, $y_{max}=11$, $z_{max}=6$) Å, so for the subsequent cut the plane was chosen to $z=3.75\text{Å}$ (0.75Å above the benzene ring).

Plotting the orbital in real space shows the subsequent figure, it corresponds to the π_1 orbital.

FIGURE 3.6: cut through the π_1 orbitals of C_6H_6 FIGURE 3.7: figure of π_1 orbital

3.4 Plotting the orbitals in momentum space

To obtain images of the orbitals in momentum space, the orbitals are Fourier-transformed (see eq. 1.5-1.6). This is done via the '*np.fft.fftn()*' function. In the figures 3.4 to 3.6 the Fourier transformed wavefunction is plotted 2-dimensionally to illustrate the processes, although the subsequent operations (zero-frequency shift, adding zeros and interpolation) are done 3-dimensionally.

```
orbitals=np.fft.fftn(densData,s=zeropad)
orbitals =np.fft.fftshift(orbitals)
```

In the source code above *densData* is the wavefunction. The parameter *s* is an array of 3 numbers (x,y,z) which extends the x,y,z components of the given 3-dim array (wavefunction) with n zeros in the given direction. This increases the resolution of the

Fourier transformation (see figure 3.9).

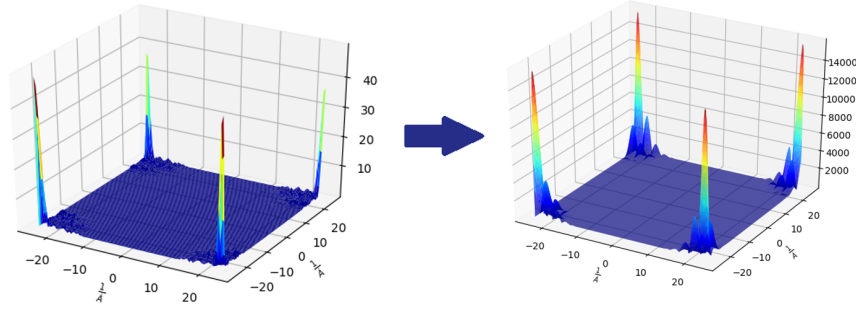


FIGURE 3.8: increase resolution

Afterward the zero frequency is shifted to the center of the spectrum. This is done via the `np.fft.fftshift()` function (see figure 3.9).

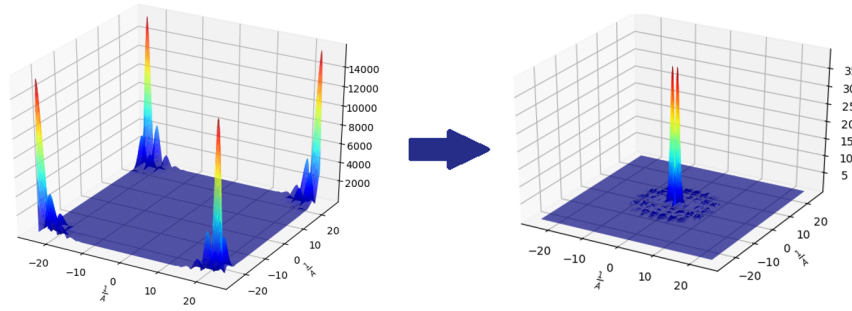


FIGURE 3.9: shift of the zero frequency to the middle

By switching the point of view normal to the x-y plane, the orbitals can be seen (see figure 3.10).

To resemble the experiment the orbitals are not cut in a plane (3.10), but in a sphere where the radius k matches the wavevektor of the free electrons used in the experiment.

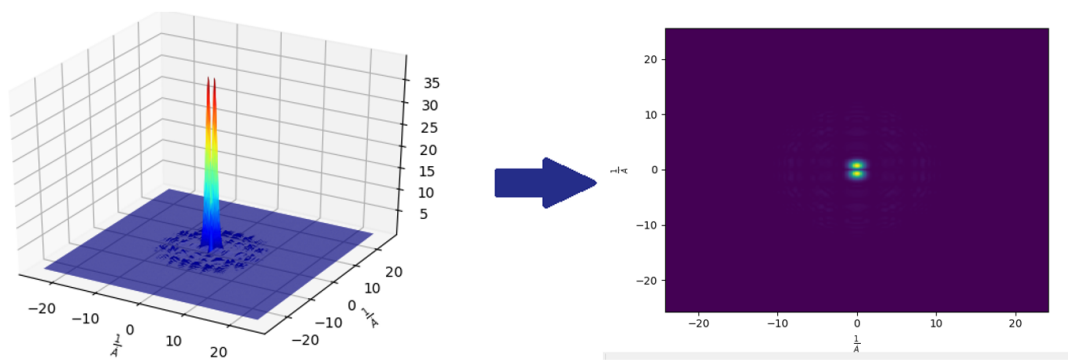
$$k = \frac{\sqrt{E_{kin} \cdot 2 \cdot m_e}}{\hbar} \quad (3.4)$$

E_{kin} ...kinetic energy of electron [eV], m_eMass of electron [kg], k ... wavevektor [1/m]

To interpolate the points on the sphere, all gridpoints of the x-y axis are determined first.

$$P_{xy} = \{(x, y) \in (X \times Y) : k^2 > x^2 + y^2\} \quad (3.5)$$

$$P_{xyz} = P_{xy} \times \sqrt{k^2 - x^2 - y^2} \quad (3.6)$$

FIGURE 3.10: π_1 orbital in x-y plane

This is implemented as shown below:

```
def spherePoints(xl,yl):
    grid=np.array([[0.0,0.0,0.0]])
    indices=np.array([[0,0,0]])
    index=0
    for x in range(len(xl)):
        for y in range(len(yl)):
            if((k**2)>(xl[x]**2+yl[y]**2)):
                grid=np.vstack([grid,[0.0,0.0,0.0]])
                indices=np.vstack([indices,[0,0,0]])
                indices[index][0]=x
                indices[index][1]=y
                grid[index][0]=xl[x]
                grid[index][1]=yl[y]
                grid[index][2]=kzsphere(xl[x],yl[y])
                index=index+1
    return grid,indices
```

To interpolate the points on the sphere, the instance *RegularGridInterpolator* is used. The first argument is a tuple of lists, containing the x,y,z spacings. The second argument contains the values of the regular grid in which the interpolation is done. The first two arguments must match in size. The third argument is sets the interpolation method.

```
intergrid,indices=spherePoints(kx,ky)
sphereCutData=RegularGridInterpolator((kx, ky, kz),
                                     orbitals,method='linear')(intergrid)
```

The function *RegularGridInterpolator* of the instance of the same name is then used by adding the argument of the coordinates which are to be interpolated. This function returns an array of values. To plot these values, they have to be refilled into an array.

```
interpolatedData=np.zeros((len(kx),len(ky)))
#fill in interpolated data in matrix
for i in range(len(sphereCutData)):
    if(intergrid[i][0]**2+intergrid[i][1]**2<k**2):
        #if datapoint in circle
        interpolatedData[indices[i][0]][indices[i][1]]=
            np.abs(sphereCutData[i])
```

[h] The resulting array is then plotted via the *pcolormesh* function. Here E_{kin} was set to 34.3 eV.

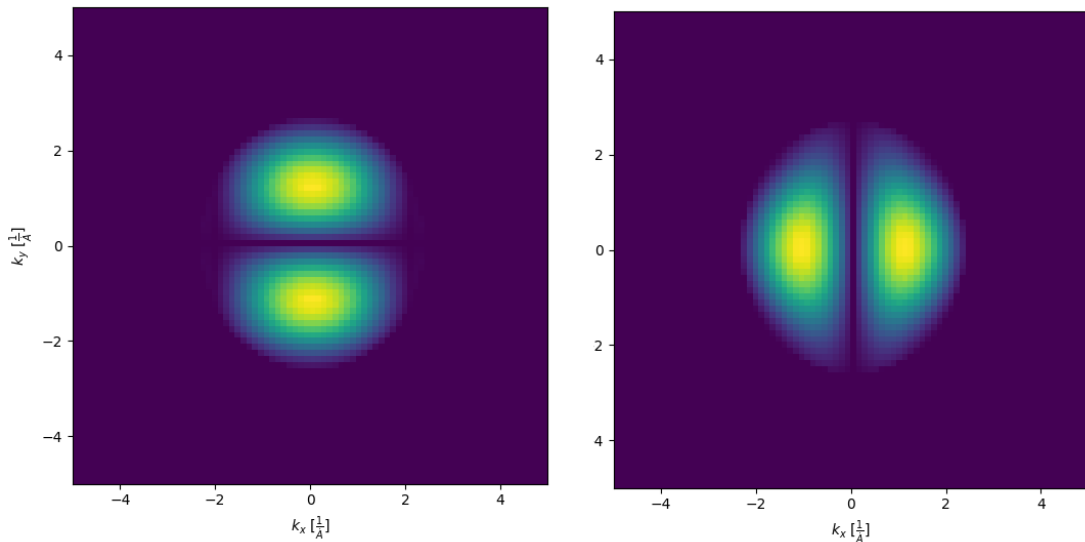


FIGURE 3.11: interpolated orbital-sphere cuts of π_1 (left) and π_2 (right)

Chapter 4

Conclusion

The aim of this work was to simulate the photoemission angular distribution of electrons emitted from molecular orbitals. By assuming a plane wave as the final state one can show that the photocurrent is proportional to the Fourier transform of the initial wave function. The first step for this simulation was to visualize the wavefunction. A commonly used program to visualize molecules and their orbitals from xsf-files is VESTA. For a benzene ring and the π_1 orbital, the following plots were made, for an energy of $E_{kin}=24.0$ eV.

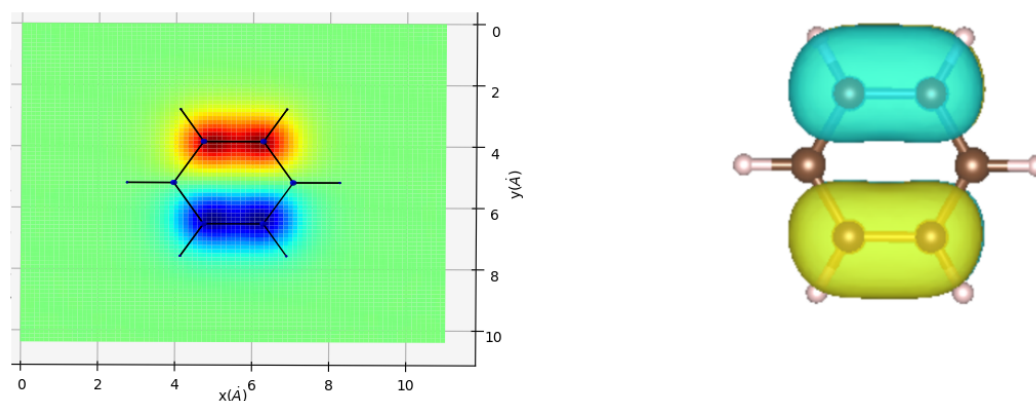


FIGURE 4.1: Molecule and Orbitals in real space

Figure 4.1. Plot of a benzene molecule with its π_1 orbital in real space. The left panel shows the output of the program described in this work and the right image shows the same file visualized via VESTA.

Subsequently the wavefunction was Fourier transformed and interpolated on a sphere with its radius corresponding to the kinetic energy of the emitted electrons. The orbital plot of the momentum space for benzens π_1 and π_2 orbitals as output of the program described in this work is shown below for a kinetic energy of $E_{kin} = 32.4$ eV corresponding to the energies of the experiments in [8]:

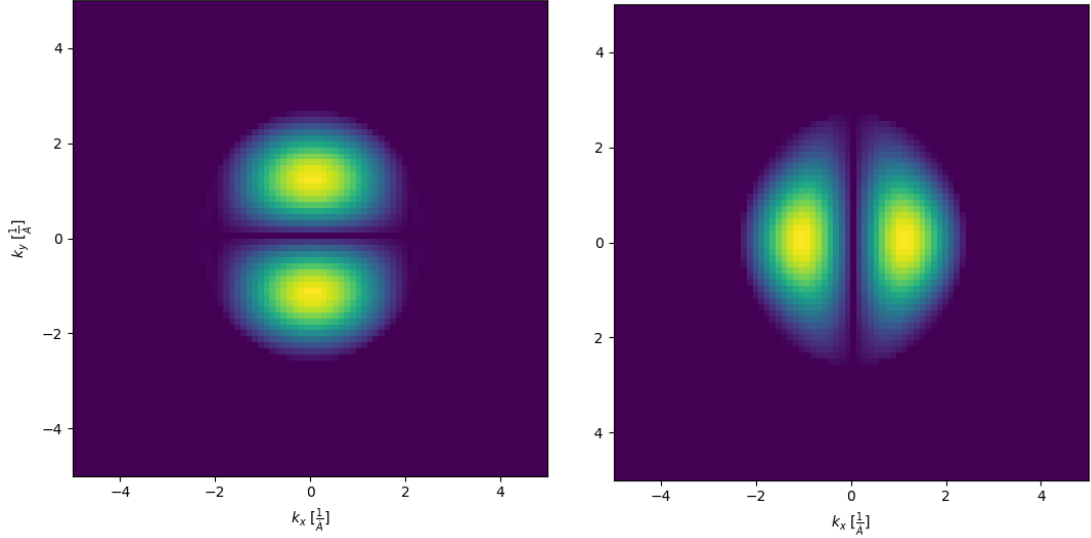


FIGURE 4.2: π_1 and π_2 orbital in momentum space for $E_{kin} = 32.4$ eV

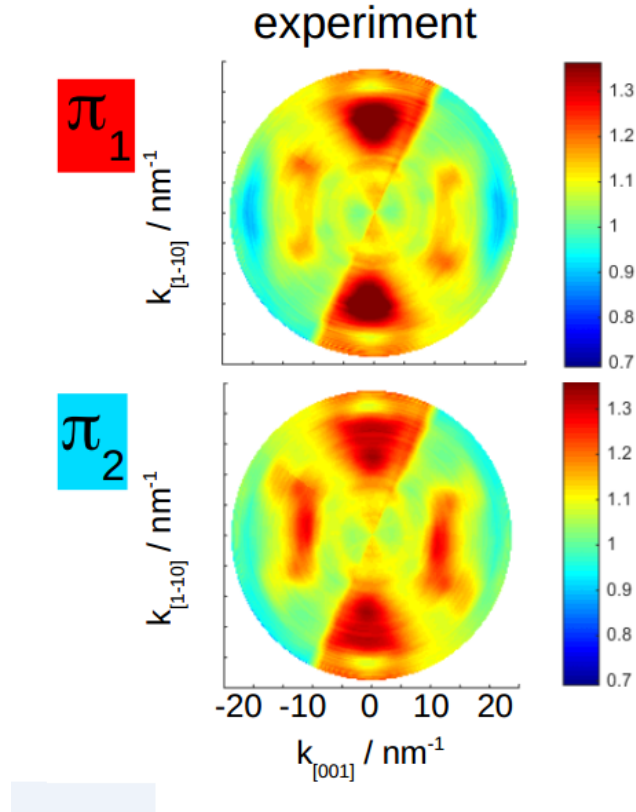


FIGURE 4.3: Photoemission angular distributions from [8]

Figure 4.3. Experimental momentum maps for energies of -8.3 eV (π_1) and -8.6 eV (π_2) integrated over an energy window of 0.25 eV.

In [8] the photoemission angular distributions for emission of the π orbitals from benzene are shown. The analysis of the photoemission momentum maps (MM) in [8] for benzene suggests a breaking of degeneracy of the topmost π state into two different orbitals (π_1 and π_2) with emission peaks around -8.3 eV and -8.6 eV (energy with respect to vacuum level). Due to the overlap of the density of states for energies from -8.3 to -8.6 eV for these two orbitals, both are visible in the experimental MM of 4.3. The characteristics of the corresponding orbital are more distinct at their emission peak. The momentum map for an energy of -8.3 eV shows a higher correspondence to the π_1 orbital, and the one for an energy of -8.6 eV to the π_2 orbital.

Bibliography

- [1] Albert Einstein. *Über einen die Erzeugung und Verwandlung des Lichts betreffenden heuristischen Gesichtspunkt*. Ann. Physik., 1905.
- [2] Daniel Lüftner. *PhD. Thesis: Orbital tomography: Understanding photoemission of organic molecular films*. Daniel Lüftner, 2015.
- [3] P Puschnig and MG Ramsey. *Photoemission Tomography: Valence Band Photoemission as a Quantitative Method for Investigating Molecular Films*. Elsevier Inc., 2017.
- [4] David Sholl and Janice Steckel. *Density Functional Theory - A Practical Introduction*. John Wiley & Sons, Inc., 2009.
- [5] A. Kokalj. *XCrySDen*. Comp. Mater. Sci, 2003. URL <http://www.xcrysden.org/XCrySDen.html>.
- [6] Peter Puschnig. *Fundamentals of electronic structure theory*. Peter Puschnig, 2018.
- [7] Beatriz Coredero, Verónica Gómez, Ana E. Platero-Prats, Marc Revés, Jorge Echevarría, Eduard Cremades, Flavia Barragán, and Santiago Alvarez. *Covalent radii revisited*. Dalton Transactions, 2008.
- [8] Larissa Egger, Bernd Kollmann, Philipp Hurdax, Daniel Lüftner, Xiaosheng Yang, Simon Weiss, Alexander Gottwald, Mathias Richter, Georg Koller, Serguei Soubatch, F. Stefan Tautz, Peter Puschnig, and Michael G. Ramsey. *Can photoemission tomography be useful for small, strongly-interacting adsorbate systems?* New J. Phys., 2018.