Simulate refraction of photoelectrons: `Photoelectron_refraction`

Hiroaki Tanaka (ISSP/School of Science, UTokyo)

February 3, 2022

Contents

1 Principles of refraction

1.1 Atomic units

The following arguments use **the atomic unit**, in which the following physical constants are omitted.

- **Electron mass** $m = 9.109 \times 10^{-31}$ kg
- **Bohr radius** $a_0 = 0.5292$ Å
- **Elementary charge** $e = 1.602 \times 10^{-19}$ C
- **Dirac constants** $\hbar = 1.054 \times 10^{-34}$ J · s

Values in the SI unit system are from Ref. [1]. As a result, the unit of the energy and wavevector becomes $E_h = 27.2114$ eV and $1/a_0 = 1.890\ \text{Å}^{-1}$, respectively. In addition, the dispersion relation of photoelectron is

$$\text{Inside of a crystal}\quad E = \frac{1}{2}|\mathbf{k}|^2 - V_0 \text{ and} \tag{1}$$

$$\text{Vacuum}\quad E = \frac{1}{2}|\mathbf{K}|^2, \tag{2}$$

where V_0 is the inner potential and $\mathbf{k}$ and $\mathbf{K}$ represent wavevectors in and out of a crystal. We take the vacuum level as the zero point of the energy. In the following, we use $k = |\mathbf{k}|,\ K = |\mathbf{K}|$.

1.2 Refraction by the xy plane

We consider the case where $z < 0$ is a crystal, $z > 0$ is vacuum, and they are separated by the xy plane (Fig. 1). When a photoelectron with wavevector $\mathbf{k}$ goes out of a crystal, the energy and in-plane wavevector are conserved. However, out-of-plane wavevector is not due to the inner potential V_0. Given $\mathbf{k} = (k_x,\ k_y,\ k_z)$, we can obtain $\mathbf{K} = (K_x,\ K_y,\ K_z)$ from the following equations.

$$K = \sqrt{k^2 - 2V_0} \tag{3}$$
$$K_x = k_x \tag{4}$$
$$K_y = k_y \tag{5}$$
$$K_z = \sqrt{K^2 - K_x^2 - K_y^2} \tag{6}$$

Since $K < k$ holds, there may be no solution of K_z. In this case, photoelectrons do not go out and total reflection happens.

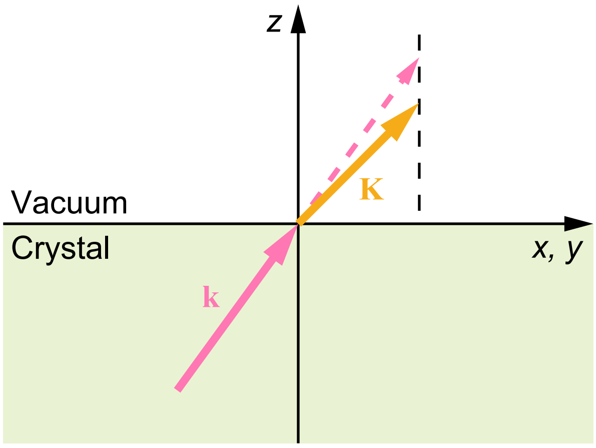


Figure 1: Refraction by the xy plane.

1.3 Refraction by a general flat plane

Next, we take a unit vector $\mathbf{n}$ and a plane which is perpendicular to $\mathbf{n}$ and includes the origin, and put a crystal in the negative z region and vacuum in the positive z region (Fig. 2(a)). Furthermore, we restrict the normal vector $\mathbf{n}$ so that it points to the vacuum, in other words the z component is positive. In this case, wavevector component perpendicular to $\mathbf{n}$ is conserved, so the following equations hold.

$$K = \sqrt{k^2 - 2V_0} \tag{7}$$
$$k_{//} = \mathbf{k} \cdot \mathbf{n} \tag{8}$$
$$\mathbf{k}_\perp = \mathbf{k} - k_{//} \cdot \mathbf{n} \tag{9}$$
$$\mathbf{K}_\perp = \mathbf{k}_\perp \tag{10}$$
$$K_{//} = \sqrt{K^2 - \mathbf{K}_\perp \cdot \mathbf{K}_\perp} \tag{11}$$
$$\mathbf{K} = \mathbf{K}_\perp + K_{//} \cdot \mathbf{n} \tag{12}$$

Since we consider photoelectrons to go out of a crystal, $k_{//} = \mathbf{k} \cdot \mathbf{n} > 0$ holds, which is used in the derivation of Eq. (12). If there is no solution of $K_{//}$ or $\mathbf{k} \cdot \mathbf{n} < 0$ holds (Fig. 2(b)), the photoelectron is not emitted to the vacuum.

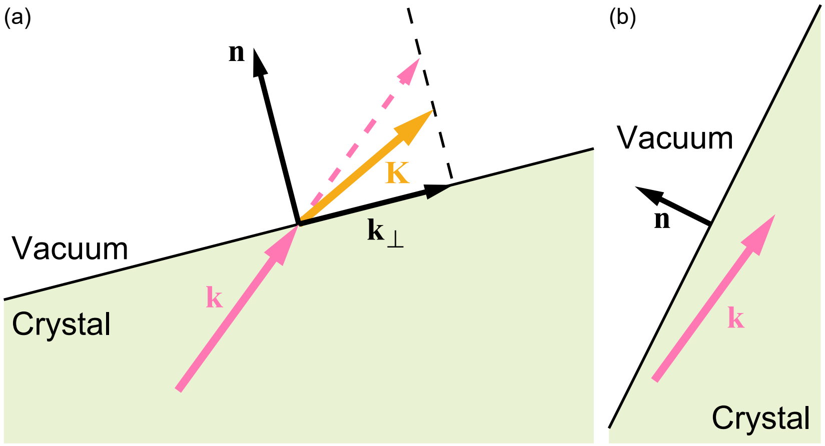


Figure 2: Refraction by a general flat plane. (a) $\mathbf{k} \cdot \mathbf{n} < 0$ case (photoelectron emitted). (b) $\mathbf{k} \cdot \mathbf{n} < 0$ case.

2 Simulation

2.1 Dispersion of initial states

In this software, we use the parabolic band dispersion like Fig. 3;

$$E_{\rm s}(\mathbf{k}) = \frac{a}{2}|\mathbf{k} - \mathbf{k}_0|^2 - W + V_1, \tag{13}$$

where a is the parameter to determine the parabola size, $\mathbf{k}_0$ is the center of the dispersion, V_1 is the energy level of the band top (or bottom), and W is the work function. We take the periodic reciprocal space for initial states, and consider photoelectrons with wavevector $\mathbf{k}$.

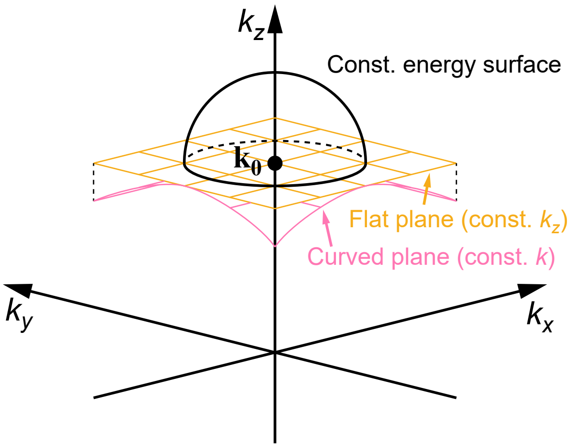


Figure 3: Constant energy surface of initial states and two-dimensional plane on which initial states are considered.

Taking into account the energy conservation of photoelectron excitation, we consider initial states on a two-dimensional plane; Ranges of k_x and k_y coordinates should be specified, and k_z or k is kept constant. If we keep k constant, the initial states are on a curved plane, but this situation is closer to the photoemission because the energy difference between initial and final states becomes the same[1].

2.2 Simulating the refracted dispersion

We take a discrete grid along the k_x and k_y directions, and put points `(kx[i], ky[j], kz, E)` by the rules determined above. Then we calculate the wavevector `(Kx, Ky, Kz, E)` refracted by the plane

[1] However, the energy conservation law is incomplete because the difference of the binding energy is neglected

specified by the normal vector $\mathbf{n}$, and put the dispersion intensity at `(Kx, Ky, E)`. The dispersion intensity is slightly broadened by the Gauss function to remove the effect of discretization. Performing this process for all points on the grid, finally we get the refracted dispersion (Fig. 4).

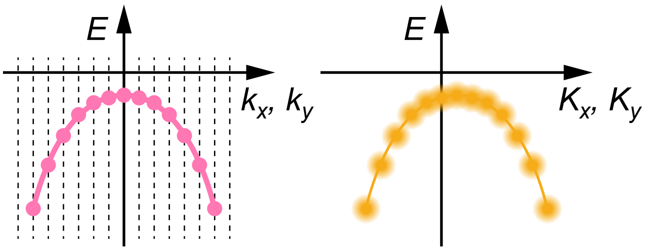


Figure 4: Schematic of the refraction simulation. The left panel is the original dispersion of initial states, and the right is the refracted dispersion. In actual calculations, we take the grid and broadening width so that the dispersion looks continuous.

2.3 Simulating the refraction by a rough surface

A rough surface is simulated by a set of isotropically distributed normal vectors $\mathbf{n}$. We perform the above calculations for the all normal vectors and take average of the intensity distributions.

2.4 Examples

Figure 5 shows examples of refraction simulation. We use default parameters in the Python GUI except for those of surfaces. Five samples is not enough to represent a rough surface, so we need much more samples to correctly simulate it.

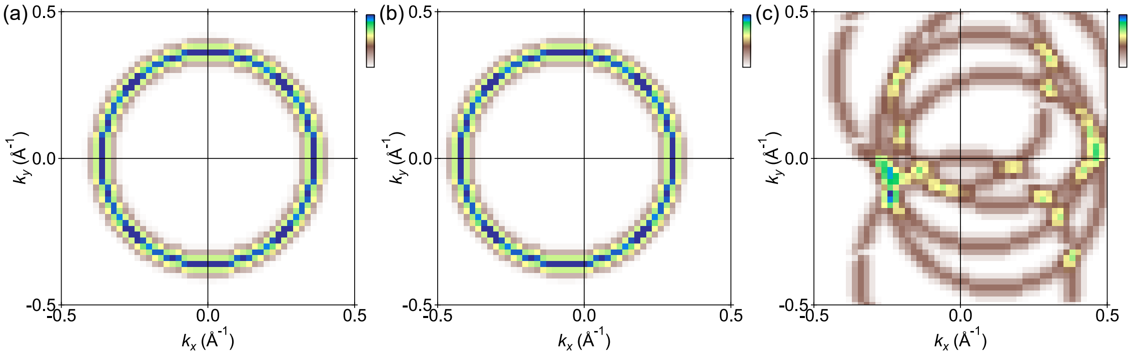


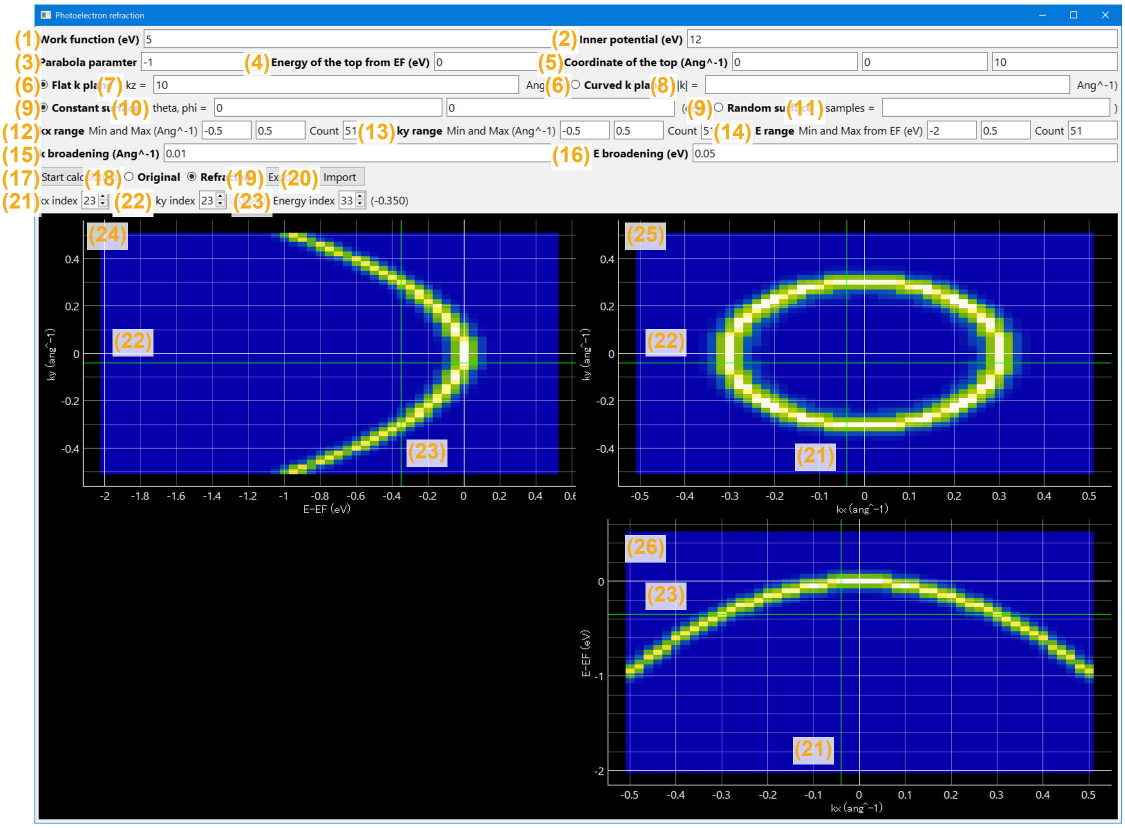
Figure 5: Examples of refraction simulation. (a) Original dispersion before refraction. (b) Refracted dispersion by the plane with the normal vector of $\theta = 20$ °, $\phi = 0$ °. (c) Refracted dispersion by five random surfaces.

3 Software

3.1 Python version

The Python version of our simulation software serves GUI based on PyQt5 and pyqtgraph (Fig. 6) Table 1 describes components. In addition, indices (21)-(23) can be changed by left/right, up/down, and PageUp/PageDn keys with the focus on the graph.

The development of the software is based on Python 3.8 and libraries in Table 2. All libraries can be installed by pip command.



Photoelectron refraction

Work function (eV) 5 · Inner potential (eV) 12

Parabola paramter -1 · Energy of the top from EF (eV) 0 · Coordinate of the top (Ang^-1) 0 · 0 · 10

● Flat k pla kz = 10 Ang · ○ Curved k pla |k| = Ang^-1)

● Constant su theta, phi = 0 · 0 (· ○ Random su samples =)

kx range Min and Max (Ang^-1) -0.5 · 0.5 · Count 51 · ky range Min and Max (Ang^-1) -0.5 · 0.5 · Count 5 · E range Min and Max from EF (eV) -2 · 0.5 · Count 51

k broadening (Ang^-1) 0.01 · E broadening (eV) 0.05

Start calc · ○ Original · ● Refra · Ex · Import

kx index 23 · ky index 23 · Energy index 33 (-0.350)

Figure 6: GUI of the software (Python version).

Table 1: Description of the components in FIg. 6.

No.	Description
(1)	Work function W
(2)	Inner potential V_0
(3)	Parabola size parameter a
(4)	Energy level of the parabola top/bottom V_1
(5)	Center of the parabola dispersion $\mathbf{k}_0$
(6)	Specify to two-dimensional surface for initial states
(7)	k_z value when the initial states are on a flat plane
(8)	k value when the initial states are on a curved plane
(9)	Surface(s) to simulate the refraction process
(10)	Polar coordinates of the normal vector when a constant plane is used
(11)	Number of samples when random planes are used
(12)	Calculation range along the k_x direction
(13)	Calculation range along the k_y direction
(14)	Calculation range along the energy direction
(15)	Broadening width along the wavevector directions
(16)	Broadening width along the eneryg direction
(17)	Execution button
(18)	Select which dispersion is displayed
(19)	Export the result to a HDF5 file
(20)	Load parameters and dispersions from a HDF5 file
(21)	k_x value to specify the cross section perpendicular to the k_x axis (24)
(22)	k_y value to specify the cross section perpendicular to the k_y axis (26)
(23)	Energy value to specify the cross section perpendicular to the energy axis (25)
(24)	Dispersion along the energy and k_x directions
(25)	Dispersion along the k_x and k_y directions
(26)	Dispersion along the energy and k_y directions

Table 2: Necessary libraries for the Python version and versions of them in our development environment.

Library	Version
`PyQt5`	5.15.4
`pyqtgraph`	0.12.3
`numpy`	1.20.3
`h5py`	3.5.0

3.2 C++ version

The C++ version is facter than the Python version, and the calculation result from the C++ version can be checked using the Python GUI.

You need a computer with a C+ compiler and `make` command, such as Linux, and the following preparations are necessary.

- Install HDF5[2] with `--enable-cxx` option.
- Enable OpenMP parallelization.

You can compile the C++ version by `make` command after adequate revision of `Makefile`. The executable `Refraction.o` runs with one argument of the input file.

The following is an example of input files. Line number determines which parameter to be read and you can leave comments after parameter(s).

```
5             # work function (eV)
12            # inner potential (eV)
0 0 4.07      # k0 (x, y, z, Ang^-1)
-1.385        # a
0             # V1 (eV)
1 4           # curved(0)/flat(1), k/kz
0 10000       # random(0)/const(1), number of samples/theta, phi (deg)
-1 1 201      # kx min, max, count
-1 1 201      # ky min, max, count
-1 0.1 56     # energy  min, max, count
0.01          # broadening width along k
0.02          # broadening width along energy
data.hdf5     # output file name
```

References

[1] https://physics.nist.gov/cuu/Constants/index.html

[2] https://www.hdfgroup.org/downloads/hdf5