

# 光電子屈折シミュレーションソフト

## Photoelectron\_refraction

田中 宏明 (東京大学 物性研究所/理学系研究科物理学専攻)

2022 年 2 月 3 日

### 目次

1	屈折の原理	1
1.1	原子単位系 . . . . .	1
1.2	$xy$ 面による屈折 . . . . .	2
1.3	一般の面による屈折 . . . . .	2
2	シミュレーション	3
2.1	始状態の分散関係 . . . . .	3
2.2	屈折を受けた分散のシミュレーション . . . . .	4
2.3	凹凸のある表面による屈折のシミュレーション . . . . .	4
2.4	例 . . . . .	4
3	ソフトウェア	5
3.1	Python 版 . . . . .	5
3.2	C++ 版 . . . . .	5

## 1 屈折の原理

### 1.1 原子単位系

以下の議論は、次の 4 つの物理定数を省略した**原子単位系**で行う。

- 電子の質量  $m = 9.109 \times 10^{-31}$  kg
- Bohr 半径  $a_0 = 0.5292 \text{ \AA}$
- 電気素量  $e = 1.602 \times 10^{-19}$  C
- Dirac 定数  $\hbar = 1.054 \times 10^{-34}$  J · s

SI 単位系における値は文献 [1] よりとった。この結果、エネルギーの単位は  $E_h = 27.2114$  eV、波数の単位は  $1/a_0 = 1.890 \text{ \AA}^{-1}$  となる。また、光電子の分散関係は

$$\text{結晶内部 } E = \frac{1}{2}|\mathbf{k}|^2 - V_0 \quad (1)$$

$$\text{結晶外部 (真空中) } E = \frac{1}{2}|\mathbf{K}|^2 \quad (2)$$

となる。ここで、 $V_0$  は内部ポテンシャルであり、 $\mathbf{k}$  および  $\mathbf{K}$  は結晶内・結晶外での波数ベクトルである。真

空準位をエネルギーの原点にとっている。また、以下では  $k = |\mathbf{k}|$ ,  $K = |\mathbf{K}|$  の表記を用いる。

## 1.2 $xy$ 面による屈折

$z < 0$  の領域が結晶、 $z > 0$  の領域が真空であり、 $xy$  面によって隔てられている場合（図 1）を考える。波数  $\mathbf{k}$  を持つ光電子が結晶内から結晶外に出るとき、エネルギーと面内波数は保存する。しかし、内部ポテンシャル  $V_0$  の存在により面直波数は保存しない。 $\mathbf{k} = (k_x, k_y, k_z)$  が与えられているとき、 $\mathbf{K} = (K_x, K_y, K_z)$  は以下の関係式から求められる。

$$K = \sqrt{k^2 - 2V_0} \quad (3)$$

$$K_x = k_x \quad (4)$$

$$K_y = k_y \quad (5)$$

$$K_z = \sqrt{K^2 - K_x^2 - K_y^2} \quad (6)$$

$K < k$  の関係があるため、条件を満たす  $K_z$  が存在しない可能性がある。この場合、光電子は結晶外へ出ず全反射する。

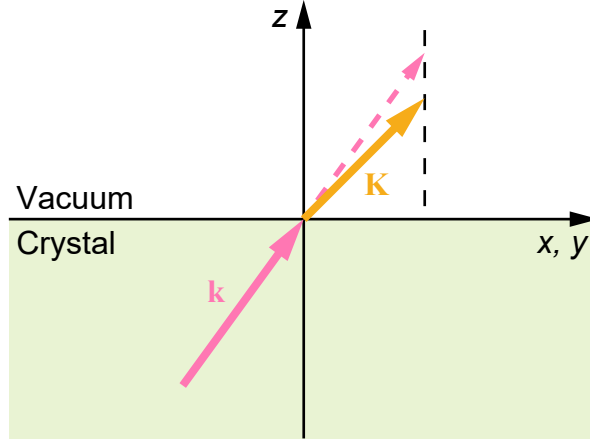


図 1  $xy$  面による屈折。

## 1.3 一般の面による屈折

原点を通り  $\mathbf{n}$  を単位法線ベクトルとして持つ面をとり、この面より  $z$  軸負の側が結晶、正の側が真空である場合を考える（図 2(a)）。ただし、 $\mathbf{n}$  が真空側を指すような配置、すなわち  $z$  成分を正にとる。このときは、 $\mathbf{n}$  に垂直な成分が保存されるように屈折が起きるので以下の関係式が得られる。

$$K = \sqrt{k^2 - 2V_0} \quad (7)$$

$$k_{//} = \mathbf{k} \cdot \mathbf{n} \quad (8)$$

$$\mathbf{k}_{\perp} = \mathbf{k} - k_{//} \cdot \mathbf{n} \quad (9)$$

$$\mathbf{K}_{\perp} = \mathbf{k}_{\perp} \quad (10)$$

$$K_{//} = \sqrt{K^2 - \mathbf{K}_{\perp} \cdot \mathbf{K}_{\perp}} \quad (11)$$

$$\mathbf{K} = \mathbf{K}_{\perp} + K_{//} \cdot \mathbf{n} \quad (12)$$

光電子が結晶側から真空側に出る運動を考えているので  $k_{//} = \mathbf{k} \cdot \mathbf{n} > 0$  が成り立ち、式 (12) においてはこのことを前提としている。条件を満たす  $K_{//}$  がない時、 $\mathbf{k} \cdot \mathbf{n} < 0$  の時（図 2(b)）は、光電子は真空中へ放出されない。

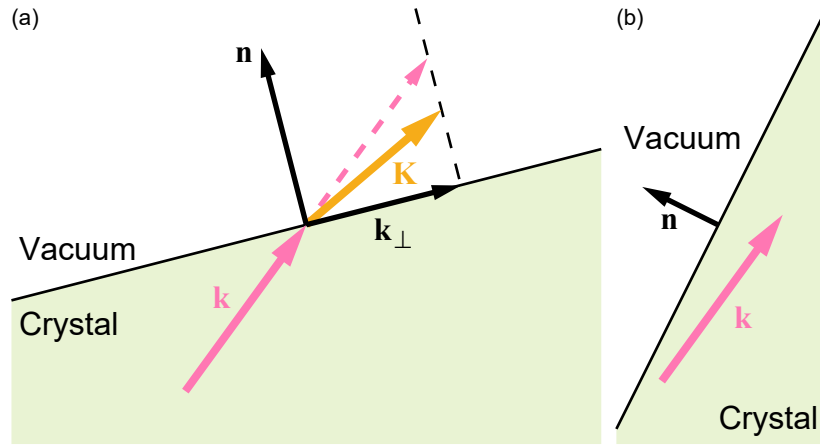


図2 一般の面による屈折。(a)  $\mathbf{k} \cdot \mathbf{n} < 0$  であり光電子が真空中へ放出される場合。(b)  $\mathbf{k} \cdot \mathbf{n} < 0$  の場合。

## 2 シミュレーション

### 2.1 始状態の分散関係

本ソフトウェアでは、始状態の分散関係は放物線バンド

$$E_s(\mathbf{k}) = \frac{a}{2} |\mathbf{k} - \mathbf{k}_0|^2 - W + V_1 \quad (13)$$

としてシミュレーションを行う (図 3)。 $a$  は放物線のサイズを決めるパラメータ、 $\mathbf{k}_0$  は放物線バンドの中心、 $V_1$  は Fermi 準位から見たバンドトップ (あるいは底) のエネルギー、 $W$  は仕事関数である。始状態の波数空間を周期ゾーン形式でとり、波数  $\mathbf{k}$  の光電子に励起されるとする。

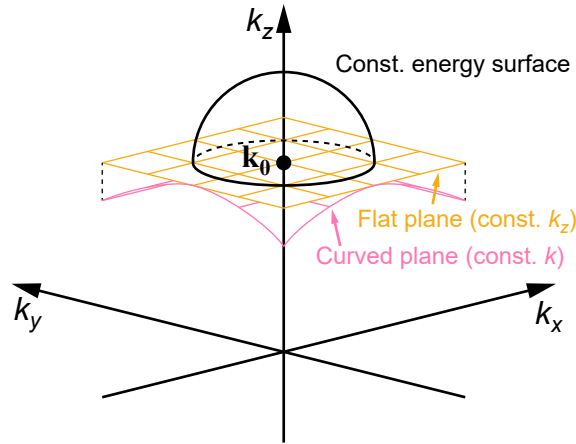


図3 始状態のバンド分散に対応する等エネルギー面およびシミュレーションを行う二次元面の取り方。

光電子励起におけるエネルギー保存則を考慮し、始状態は二次元面上にとる。 $k_x \cdot k_y$  座標は範囲指定し、 $k_z$  座標または  $k$  の値を定数で指定する。 $k$  の値を指定すると始状態は曲面上に載ってしまうが、終状態とのエネルギー差が一定になるので光電子励起現象に近くなる<sup>\*1</sup>。

<sup>\*1</sup> バンド分散による結合エネルギーの違いは考慮していないため、エネルギー保存則を完全に満たしているというわけではない。

## 2.2 屈折を受けた分散のシミュレーション

$k_x \cdot k_y$  方向に離散化したグリッドをとり、グリッドの各点 ( $k_x[i]$ ,  $k_y[j]$ ,  $k_z$ ,  $E$ ) を前節で指定した分散関係および面の取り方で定める。法線ベクトル  $\mathbf{n}$  によって指定される面での屈折を受けた後の（真空中）波数ベクトル ( $K_x$ ,  $K_y$ ,  $K_z$ ,  $E$ ) を計算し、( $K_x$ ,  $K_y$ ,  $E$ ) の位置にバンド分散の強度を置く。離散化の影響を消すために、強度は十分小さい幅で Gauss 関数による広がりを持たせておく。この行程をグリッド上の全ての点について行うことで、屈折を受けた分散を得ることができる（図 4）。

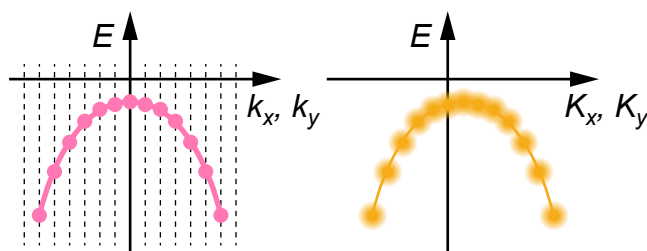


図 4 屈折を受けた分散シミュレーションの模式図。左が屈折前の始状態、右が屈折後の分散。実際には、Gauss 関数による広がり度で分散が連続的になるようなグリッドと広がりの幅に設定する。

## 2.3 凹凸のある表面による屈折のシミュレーション

凹凸のある表面は、等方的に分布したランダムな法線ベクトル  $\mathbf{n}$  の集合によってシミュレーションする。前節の行程を全ての法線ベクトルについて行って強度分布の平均をとる。

## 2.4 例

図 5 は屈折シミュレーションの例である。表面の指定以外は Python の GUI における既定値を使用している。凹凸のある表面を表現するのにサンプル数 5 は不十分であり、実際にはもっと多いサンプル数でシミュレーションする必要がある。

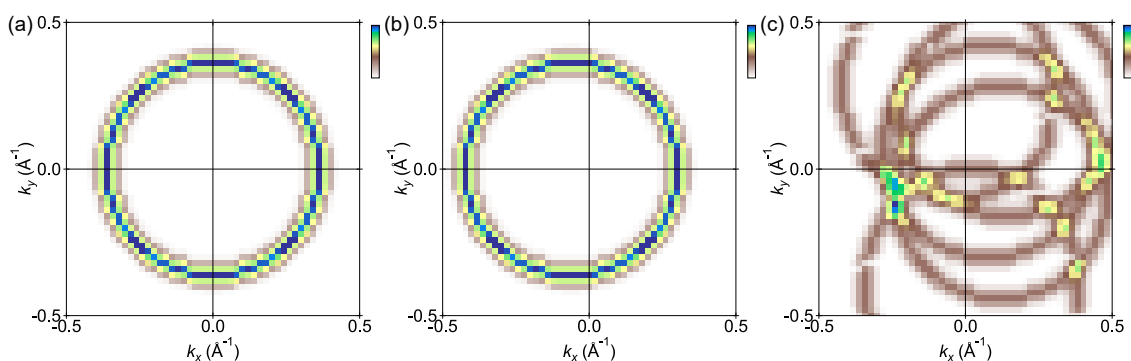


図 5 分散シミュレーションの例。(a) 屈折前の分散。(b)  $\theta = 20^\circ$ ,  $\phi = 0^\circ$  の平面による屈折後の分散。(c) サンプル数 5 のランダムな表面による屈折後の分散。

### 3 ソフトウェア

#### 3.1 Python 版

Python 版のシミュレーションソフトウェアは、PyQt5・pyqtgraph をベースとした GUI (図 6) になっている。各項目の役割は表 1 の通り。(21)-(23) のインデックスは、グラフにフォーカスした状態でカーソル左右・カーソル上下・PageUp/PageDn キーによって増減させられる。

開発は Python 3.8 上で行い、表 2 のライブラリを用いた。ライブラリは全て pip コマンドで入手することができる。

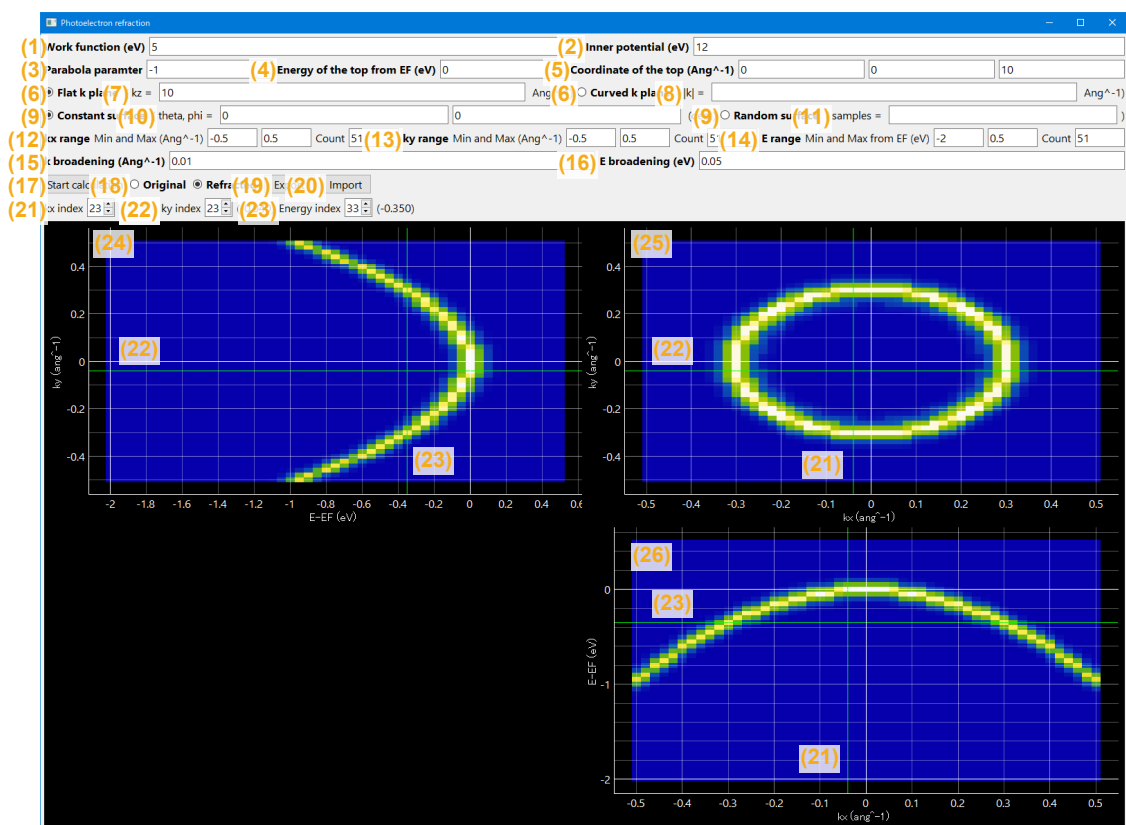


図 6 Python 版シミュレーションソフトウェアの外観。

表 1 図 6 に示した GUI の項目説明。

No.	説明
(1)	仕事関数 $W$
(2)	内部ポテンシャル $V_0$
(3)	放物線バンドの幅パラメータ $a$
(4)	バンドトップのエネルギー $V_1$
(5)	放物線バンドの中心 $\mathbf{k}_0$
(6)	始状態をとる二次元面の指定
(7)	始状態を平面でとるときの $k_z$
(8)	始状態を曲面でとるときの $k$
(9)	屈折を計算する面の指定
(10)	一定の面で屈折を計算する際の法線ベクトルを極座標で指定
(11)	ランダムな面で屈折を計算する際のサンプル数
(12)	分散を計算する範囲 ( $k_x$ 方向)
(13)	分散を計算する範囲 ( $k_y$ 方向)
(14)	分散を計算する範囲 (エネルギー方向)
(15)	Gauss 関数による広がり幅 (波数方向)
(16)	Gauss 関数による広がり幅 (エネルギー方向)
(17)	計算実行ボタン
(18)	グラフに表示する分散の選択
(19)	計算結果を HDF5 ファイルに出力
(20)	計算結果ファイルからパラメータ・分散を読み取る
(21)	$k_x$ 軸に垂直な切断面 (24) を指定する $k_x$ の値
(22)	$k_y$ 軸に垂直な切断面 (26) を指定する $k_y$ の値
(23)	エネルギー軸に垂直な切断面 (25) を指定するエネルギーの値
(24)	エネルギー- $k_y$ 方向の分散
(25)	$k_x$ - $k_y$ 方向の分散
(26)	エネルギー- $k_x$ 方向の分散

表 2 Python 版の実行に必要なライブラリと開発環境でのバージョン。

ライブラリ名	バージョン
PyQt5	5.15.4
pyqtgraph	0.12.3
numpy	1.20.3
h5py	3.5.0

## 3.2 C++ 版

C++ 版は Python 版に比べより高速な計算が可能である。C++ 版の計算結果は Python 版の GUI で読み込んで表示することもできる。

C++ 版を使用するためには、Linux などの C++ コンパイラ・`make` コマンドが使用できる環境が必要である。さらに、以下 2 つの準備が必要である。

- HDF5[2] を `--enable-cxx` オプション付きでインストールする。
- OpenMP 並列化を行えるようにする。

C++ 版のコンパイルは、`Makefile` を環境に応じて修正したのち `make` コマンドで行うことができる。実行ファイル `Refraction.o` は入力ファイル名を引数に実行する。

以下に入力ファイルの例を示す。行番号と読み取るパラメータが対応しており、パラメータの後ろには自由にコメントを残すことができる。

```
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
```

## 参考文献

- [1] <https://physics.nist.gov/cuu/Constants/index.html>
- [2] <https://www.hdfgroup.org/downloads/hdf5>