結晶工学セミナー

X線回折計算 xrayutilities入門

名古屋大学 シンクロトロン光研究センター
田渕雅夫

応用物理学会 結晶工学分科会 結晶工学セミナー
於: 産総研 臨海副都心センター
2026年3月9日(月) 13:00～14:30 (10:00～17:45)

xrayutilities :

python のライブラリ

- X線回折測定のデータ解析
- X線回折に関わるシミュレーション

python 初心者(田渕)が、xrayutilities を使って
シミュレーション計算等を行い、
初心者ならではの視点でその導入方法や使い方
注意点などを伝える

1. 環境構築

    xrayutilities のインストール

2. 構造因子の計算
3. 逆格子マップの計算
4. ω-2θ スペクトルの計算

環境構築: xrayutilities のインストール

0. 以降「xrayutilities」を「xu」と表記します。

1. python をどうにかしてインストールする
   例えば : https://www.python.org/

2. テキストエディタを準備する
   Windows 標準の note でも構わない
   vscode 等もおすすめ
   (アプリストアから入れない方が良さげ)

3. 作業は CUI ベースになるので、
   CMD / PowerShell / bash 等
   コマンド入力できるコンソールを選ぶ

4. 選んだコンソールを起動

python

vscode

bash/git bash

環境構築: xrayutilities のインストール

5. 作業場所(Windowsの言葉で「ホルダ」)を選び、
   そこに移動 (今後 xu 関連の作業を行う場所を決めて作るのがお勧め)
   例えば
   > mkdir xu　　個人フォルダの下に xu フォルダを作る
   > cd xu　　そこに「移動」

6. xu 実行用の python の仮想環境を作る

- python はライブラリも含めてみると、相互のバージョン依存性が高い。
- xu の為に入れたライブラリが、他のライブラリとはケンカして動かない
  というようなことがよく起こる。
- 各目的ごとに「仮想環境」を作り、特定の目的に必要なライブラリは
  その中でインストールするのがトラブル回避の最善策。

```
> python.exe  -m  venv  xuEnv
```

python.exe : もしかすると python3, python3.11 等ちょっと違う名前かも
venv : これはキーワードなので必ず venv
xuTest : 自分で決めた好きな名前(仮想環境の名前になる. 目的に応じて複数作れる)

環境構築: xrayutilities のインストール

7. 仮想環境の起動(仮想環境内への移動)

7-1. 起動用スクリプトがどこにあるか確認

- その場所で > start . としてエクスプローラ起動
- 今作った仮想環境の名前のフォルダを覗く
  bin もしくは Scripts とい名前のフォルダの中を覗く
- Activate, activate... という名前のファイルがあれば正解

7-2. CUI に戻って(使っている CUI に合わせて)、

CMD : > xuEnv¥bin¥activate.bat

PowerShell : > xuEnv¥bin¥Activate.ps1

bash : > source xuEnv¥bin¥activate (拡張子なし)

- bin か Scripts かは適宜
- ファイル名の区切りに使っている '¥' (バックスラッシュ)は、もしかすると'/' かも
- ps1 ファイルが動かないと言われたら...
  一時的に許可 : > Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
  ずっと許可 : > Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned

環境構築: xrayutilities のインストール

8. ここまでこれたらだいたい成功 !!

> pip install xrayutilities

このあと、xrayutilities を使う際に python の
モジュールが足りないと言われるかも。
その際には、仮想環境が起動している状態で

> pip install [無いと言われたモジュール名(複数可)]

とすれば良い。例えば

> pip install numpy scipy　…等々

必要なものはたいてい、pip install xrayutilities の段階で一緒に入るが、
言われる可能性があるとしたら、

h5py, scipy, numpy, lmfit, matplotlib

なのであらかじめinstall しておくのも良いかも
(> pip install numpy 等として、もし入ってたら、
入ってると言われるだけなので害はない)

1. 環境構築

    xrayutilities のインストール

2. 構造因子の計算
3. 逆格子マップの計算
4. ω-2θ スペクトルの計算

2. 構造因子

X線散乱：試料内各点で散乱したX線の総和

$$F = \int n(r) \exp -i\Delta k \cdot r \, dr$$

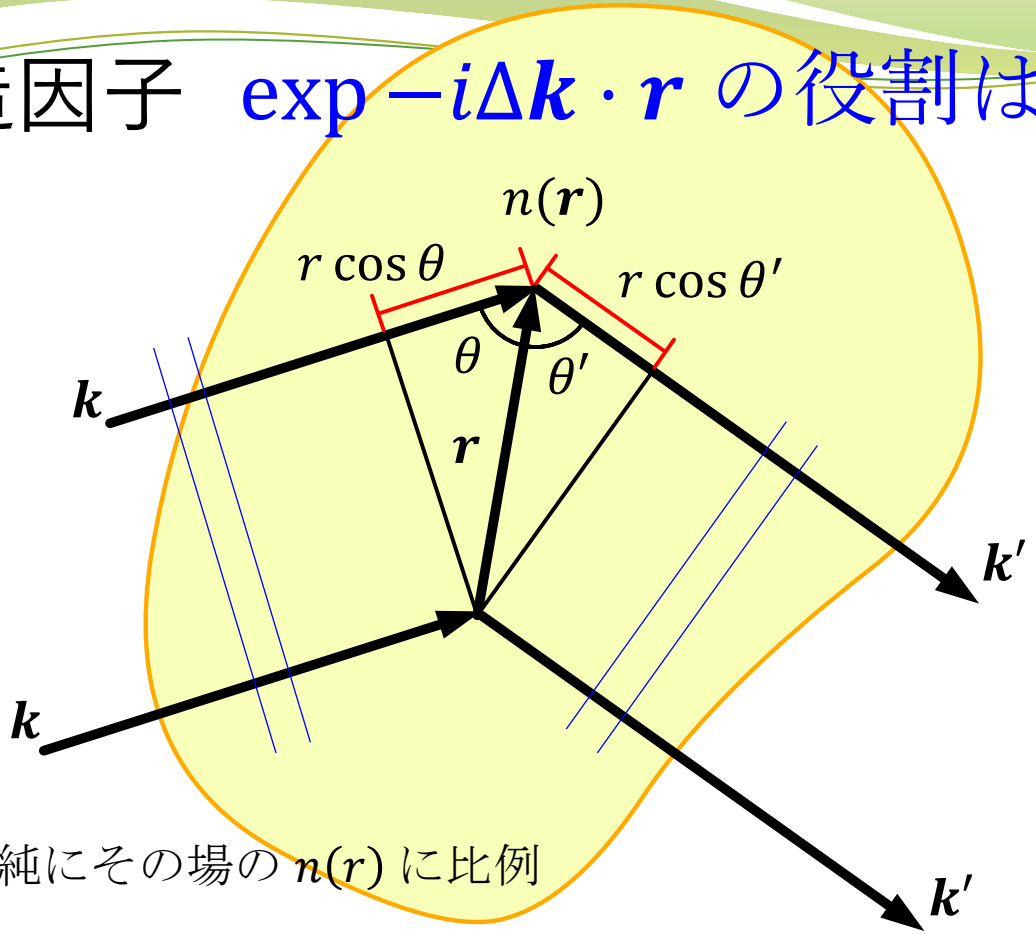
各点での散乱の強さは
その場所の電子密度に
比例する

r: 位置ベクトル
$n(r)$: 電子密度
Δk : 散乱ベクトル $\Delta k = k' - k$

$n(r)$ に周期性があると
回折が起こる

$\exp -i\Delta k \cdot r$ の役割は？

2. 構造因子 $\exp -i\Delta \boldsymbol{k}\cdot\boldsymbol{r}$ の役割は？



散乱強度は単純にその場の $n(r)$ に比例

光路差 : $\Delta L = r\cos\theta + r\cos\theta$

位相差 : $2\pi\dfrac{\Delta L}{\lambda} = \dfrac{2\pi}{\lambda}r\cos\theta + \dfrac{2\pi}{\lambda}r\cos\theta' = -\boldsymbol{k}\cdot\boldsymbol{r} + \boldsymbol{k}'\cdot\boldsymbol{r} = \Delta\boldsymbol{k}\cdot\boldsymbol{r}$

場所が違うと、波の山谷がどれだけずれるか

$$\exp i(k\cdot r - \omega t)\times \exp -i\Delta k\cdot r = \exp i(k\cdot r - \omega t - \Delta k\cdot r)$$

2. 構造因子

X線散乱：試料内各点で散乱したX線の総和

$$F = \int n(\boldsymbol{r}) \exp -i\Delta\boldsymbol{k} \cdot \boldsymbol{r}\, d\boldsymbol{r}$$

$\boldsymbol{r}$: 位置ベクトル
$n(\boldsymbol{r})$: 電子密度
$\Delta\boldsymbol{k}$: 散乱ベクトル $\Delta\boldsymbol{k} = \boldsymbol{k}' - \boldsymbol{k}$

$n(\boldsymbol{r})$ に周期性があると
回折が起こる

具体的に計算してみる

2. 構造因子

$$F = \int n(\boldsymbol{r}) \exp -i\Delta\boldsymbol{k} \cdot \boldsymbol{r}\, d\boldsymbol{r}$$

ルート1: フーリエ級数展開を代入してみる

フーリエ係数

$$n_G = \frac{1}{V_{\text{cell}}} \int_{\text{単位構造}} n(r) \exp -iG \cdot r\ dr$$

$$n(\boldsymbol{r}) = \Sigma_{\boldsymbol{G}} n_{\boldsymbol{G}} \exp i\boldsymbol{G} \cdot \boldsymbol{r}$$

$$F = \int_{\text{結晶全体}} (\Sigma_{\boldsymbol{G}} n_{\boldsymbol{G}} \exp i\boldsymbol{G} \cdot \boldsymbol{r}) \exp -i\Delta\boldsymbol{k} \cdot \boldsymbol{r}\, d\boldsymbol{r}$$

$$= \int_{\text{結晶全体}} \Sigma_{\boldsymbol{G}} n_{\boldsymbol{G}} \exp i(\boldsymbol{G} - \Delta\boldsymbol{k}) \cdot \boldsymbol{r}\ d\boldsymbol{r}$$

$$= \Sigma_{\boldsymbol{G}} n_{\boldsymbol{G}} \int_{\text{結晶全体}} \exp i(\boldsymbol{G} - \Delta\boldsymbol{k}) \cdot \boldsymbol{r}\, d\boldsymbol{r}$$

積分: $\boldsymbol{G} - \Delta\boldsymbol{k} = 0$: 結晶の体積

$\neq 0$: 打ち消しあった小さな値

回折条件($\boldsymbol{G} = \Delta\boldsymbol{k}$) を満たす : 強度は $n_{\boldsymbol{G}} \cdot V_{\text{Body}}$ (構造因子)

2. 構造因子

$$F = \int n(\boldsymbol{r}) \exp -i\Delta\boldsymbol{k}\cdot\boldsymbol{r}\, d\boldsymbol{r}$$

ルート2: 単位構造に分割してみる　並進ベクトル：$\boldsymbol{T} = n_1\boldsymbol{a} + n_2\boldsymbol{b} + n_3\boldsymbol{c}$

$$F = \int n(\boldsymbol{r}) \exp -i\Delta\boldsymbol{k}\cdot\boldsymbol{r}\, d\boldsymbol{r}$$

結晶全体 $= \Sigma_T$それぞれの場所の単位構造の体積

$$= \Sigma_T \int n(\boldsymbol{r}) \exp -i\Delta\boldsymbol{k}\cdot\boldsymbol{r}\, d\boldsymbol{r}$$

それぞれの場所の単位構造の体積

さっきは $\boldsymbol{r}$ だった

$$= \Sigma_T \left\{ \int n(\boldsymbol{r}) \exp -i\Delta\boldsymbol{k}\cdot\boldsymbol{r}\, d\boldsymbol{r} \right\} \exp -i\Delta\boldsymbol{k}\cdot\boldsymbol{T}$$

原点の単位構造の体積

積分範囲を原点に移すための因子

$$= \left\{ \int n(\boldsymbol{r}) \exp -i\Delta\boldsymbol{k}\cdot\boldsymbol{r}\, d\boldsymbol{r} \right\} \Sigma_T \exp -i\Delta\boldsymbol{k}\cdot\boldsymbol{T}$$

原点の単位構造の体積

場所による位相の変化

2. 構造因子

$$F = \left\{ \int n(\boldsymbol{r}) \exp -i\Delta\boldsymbol{k}\cdot\boldsymbol{r}\, d\boldsymbol{r} \right\} \Sigma_{\boldsymbol{T}} \exp -i\Delta\boldsymbol{k}\cdot\boldsymbol{T}$$

原点の単位構造の体積

一般の Δk :
～1 の正負様々な値の和 => 小さい

$\Delta\boldsymbol{k} = \boldsymbol{G}$ の時

逆格子ベクトル : $\boldsymbol{G} = h\boldsymbol{a}^* + k\boldsymbol{b}^* + l\boldsymbol{c}^*$

並進ベクトル : $\boldsymbol{T} = n_1\boldsymbol{a} + n_2\boldsymbol{b} + n_3\boldsymbol{c}$

$$\Delta\boldsymbol{k}\cdot\boldsymbol{T} = \boldsymbol{G}\cdot\boldsymbol{T} = hn_1\boldsymbol{a}\boldsymbol{a}^* + kn_2\boldsymbol{b}\boldsymbol{b}^* + ln_3\boldsymbol{c}\boldsymbol{c}^*$$
$$= 2\pi(hn_1 + kn_2 + ln_3) = 2\pi \times \text{整数}$$

$$\Sigma_{\boldsymbol{T}} \exp -i\Delta\boldsymbol{k}\cdot\boldsymbol{T} = \boldsymbol{\Sigma_T} \exp -2\pi i \text{ 整数} = N$$

格子点の数

$$F = N \int n(\boldsymbol{r}) \exp -i\boldsymbol{G}\cdot\boldsymbol{r}\, d\boldsymbol{r}$$

原点の単位構造の体積

2. 構造因子

回折条件($\boldsymbol{G} = \Delta \boldsymbol{k}$) を満たす：強度は $n_{\boldsymbol{G}} \cdot V_{\mathrm{Body}}$

ルート1:

$$F = n_{\boldsymbol{G}} V_{\mathrm{Body}}$$

フーリエ係数

$$n_{\boldsymbol{G}} = \frac{1}{V_{\mathrm{cell}}} \int_{\text{単位構造}} n(\boldsymbol{r}) \exp -i\boldsymbol{G} \cdot \boldsymbol{r}\ d\boldsymbol{r}$$

$\boldsymbol{n}$を平面波に分解して
表現するとき
波数 $\boldsymbol{G}$ の成分の大きさ

ルート2:

$$F = N \int_{\text{原点の単位構造の体積}} n(\boldsymbol{r}) \exp i\boldsymbol{G} \cdot \boldsymbol{r}\, d\boldsymbol{r}$$

$$= NV_{\mathrm{cell}} \frac{1}{V_{\mathrm{cell}}} \int_{\text{原点の単位構造の体積}} n(\boldsymbol{r}) \exp i\boldsymbol{G} \cdot \boldsymbol{r}\, d\boldsymbol{r} = V_{\mathrm{Body}}\, n_{\boldsymbol{G}}$$

単位格子のみの結晶が
どれだけX線を散乱するか

2. 構造因子

n_G　理論値 ➡ 測定値(回折強度)

測定すべき回折をあらかじめ知る
- 強い / 弱い
- 禁制

理論値 ⬅ 測定値(回折強度)

- 測定したものが「予想した結晶」であることの確認
- どう違うかという情報
- 単位構造そのものの推測

2. 構造因子 n_G の値は？ xrayUtilities 登場 !!

- 結晶構造情報を渡す どうやって？

```
import xrayutilities as xu
import matplotlib.pyplot as plt
```

この後いつもプログラムの先頭に書く

xu は、ある程度知ってる example-01-01.py

```
GaAs = xu.materials.GaAs
InP  = xu.materials.InP
```

xu が知ってる原子を使って組み立てる example-01.py

```
In = xu.materials.elements.In
P = xu.materials.elements.P
elastictensor = xu.materials.CubicElasticTensor( 10.11e+10, 5.61e+10, 4.56e+10 )
InP = xu.materials.Crystal( "InP",
            xu.materials.SGLattice( 216,   5.8687,
                                    atoms=[In, P], pos=['4a', '4c']), elastictensor )
```

cif ファイルから作らせる

```
GaAs = xu.materials.Crystal.fromCIF( "GaAs.cif" )
InAs = xu.materials.Crystal.fromCIF( "InAs.cif" )
InP = xu.materials.Crystal.fromCIF( "InP.cif" )
SiC4H = xu.materials.Crystal.fromCIF( "SiC-4H.cif" )
GaN = xu.materials.Crystal.fromCIF( "GaN-Hex.cif" )
AlN = xu.materials.Crystal.fromCIF( "AlN-Hex.cif" )
```

example-01-01.py

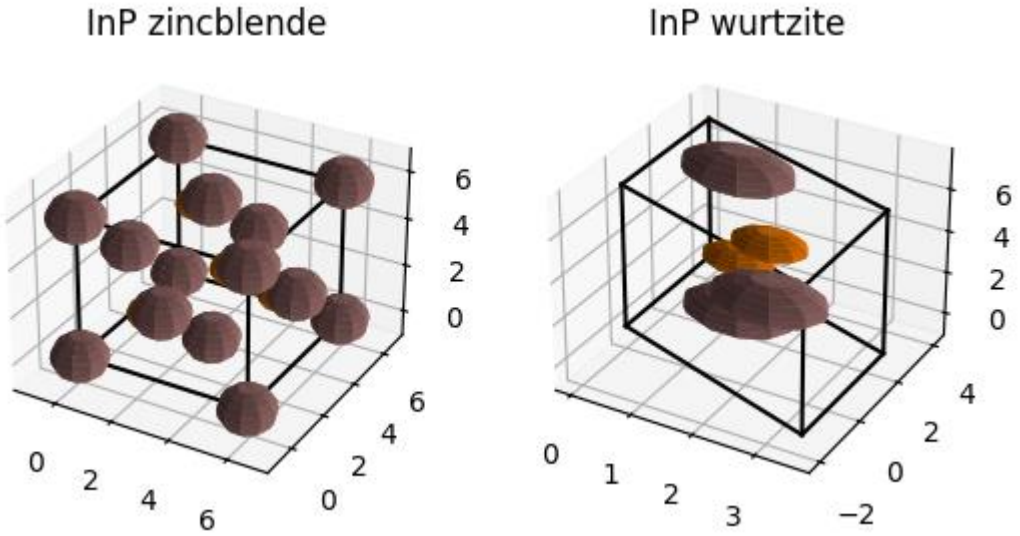
2. 構造因子　xrayUtilities 登場 !!

- どんな構造になったか見てみる

```
f = plt.figure()
InP.show_unitcell( fig=f, subplot=121 )
plt.title( 'InP zincblende' )
InPWZ.show_unitcell( fig=f, subplot=122 )
plt.title( 'InP wurtzite' )
plt.show()
```

f という名前で描画キャンパスを作る

さっき作った InP に依頼して
「単位格子」を f に描画してもらう



2. 構造因子　xrayUtilities 登場 !!

- 肝心の構造因子は？

xu は Δk を Q と呼びます: $Q \equiv \Delta k$

```
import xrayutilities as xu

GaAs = xu.materials.GaAs
InP  = xu.materials.InP

energy= 8048 # eV
hkllist = [[0,0,0], [1, 1, 1], [2, 2, 2], [3, 3, 3], [0, 0, 0], [1, 0, 0], [2, 0, 0], [4, 0, 0]]

print( f"Structual Factors of InP at {energy}[eV]" )
for hkl in hkllist:
    qvec = InP.Q( hkl )
    F = InP.StructureFactor( qvec, energy )
    print( f"|F|{hkl} = {abs(F):8.3f}" )

print( f"Structual Factors of GaAs at {energy}[eV]" )
for hkl in hkllist:
    qvec = GaAs.Q( hkl )
    F = GaAs.StructureFactor(qvec, energy)
    print(f"|F|{hkl} = {abs(F):8.3f}")
```

ちょっとさぼって xu が知ってる物質を使う

計算してほしい逆格子点のセット

(指数で)指定された逆格子点の散乱ベクトル

その散乱ベクトルの時の構造因子

なぜエネルギーを指定？

2. 構造因子

Structual Factors of InP at 8048[eV]

|F|[0, 0, 0] = 258.476
|F|[1, 1, 1] = 182.826
|F|[2, 2, 2] = 103.439
|F|[3, 3, 3] = 112.501
|F|[0, 0, 0] = 258.476
|F|[1, 0, 0] = 0.000
|F|[2, 0, 0] = 121.886
|F|[4, 0, 0] = 163.484

Structual Factors of GaAs at 8048[eV]

|F|[0, 0, 0] = 247.148
|F|[1, 1, 1] = 148.498
|F|[2, 2, 2] = 6.527
|F|[3, 3, 3] = 92.194
|F|[0, 0, 0] = 247.148
|F|[1, 0, 0] = 0.000
|F|[2, 0, 0] = 7.106
|F|[4, 0, 0] = 154.631

3. 逆格子マップの計算

・プロットしてみる

```
# X線のエネルギーが何も書かれていないが、
# デフォルトで Cu Ka1 になってるはず。
import xrayutilities as xu
import matplotlib.pyplot as plt

GaAs   = xu.materials.GaAs
hGaAs = xu.HXRD( GaAs.Q( 1, 0, 0 ), GaAs.Q( 0, 0, 1 ) )
ax, h    = xu.materials.show_reciprocal_space_plane( GaAs, hGaAs )
```

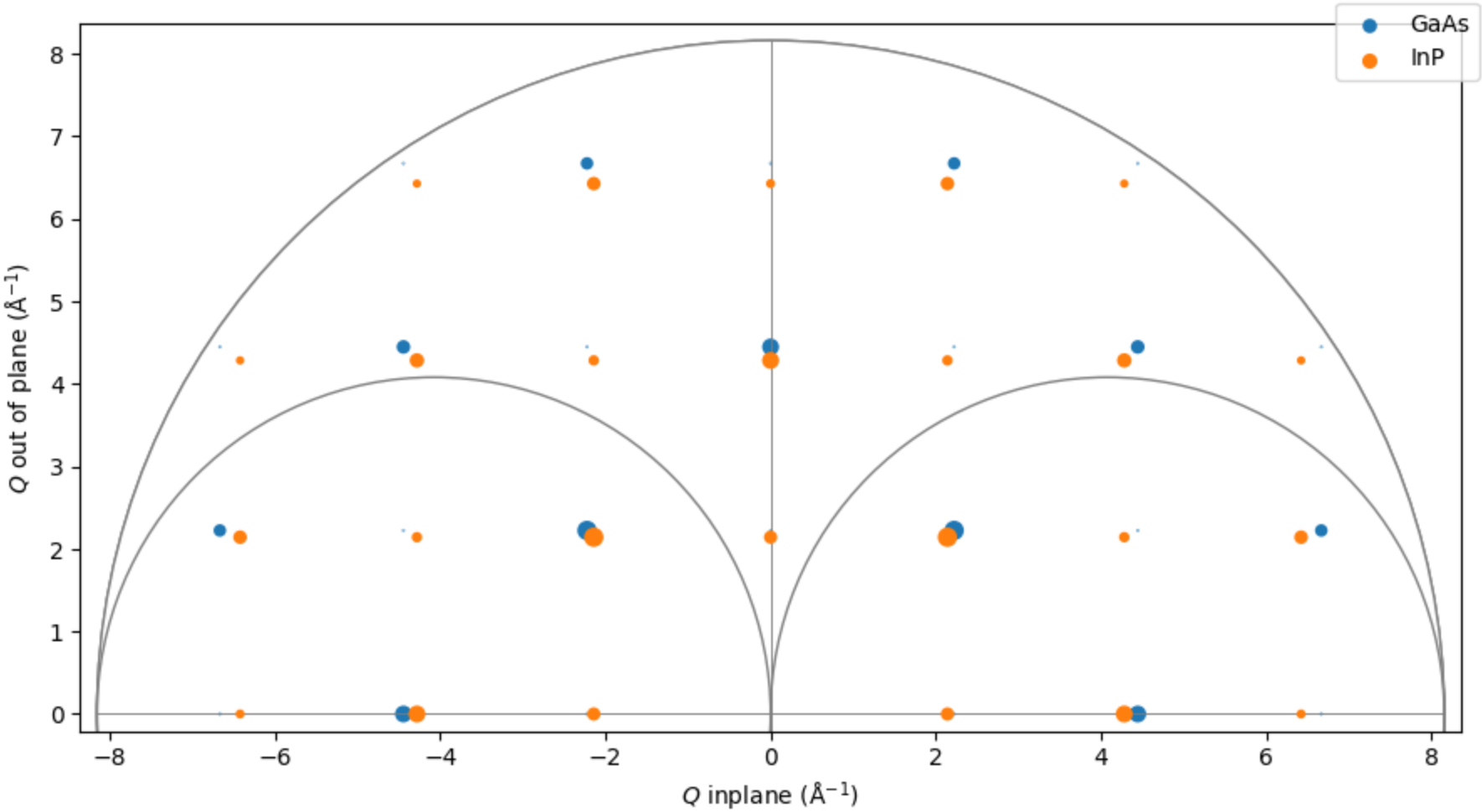
GaAs の x=(1,0,0), y=(0,0,1) 断面のプロット

```
InP   = xu.materials.InP
hInP = xu.HXRD( InP.Q( 1, 0, 0 ), InP.Q( 0, 0, 1 ) )
ax, h = xu.materials.show_reciprocal_space_plane( InP, hInP, ax=ax )
```

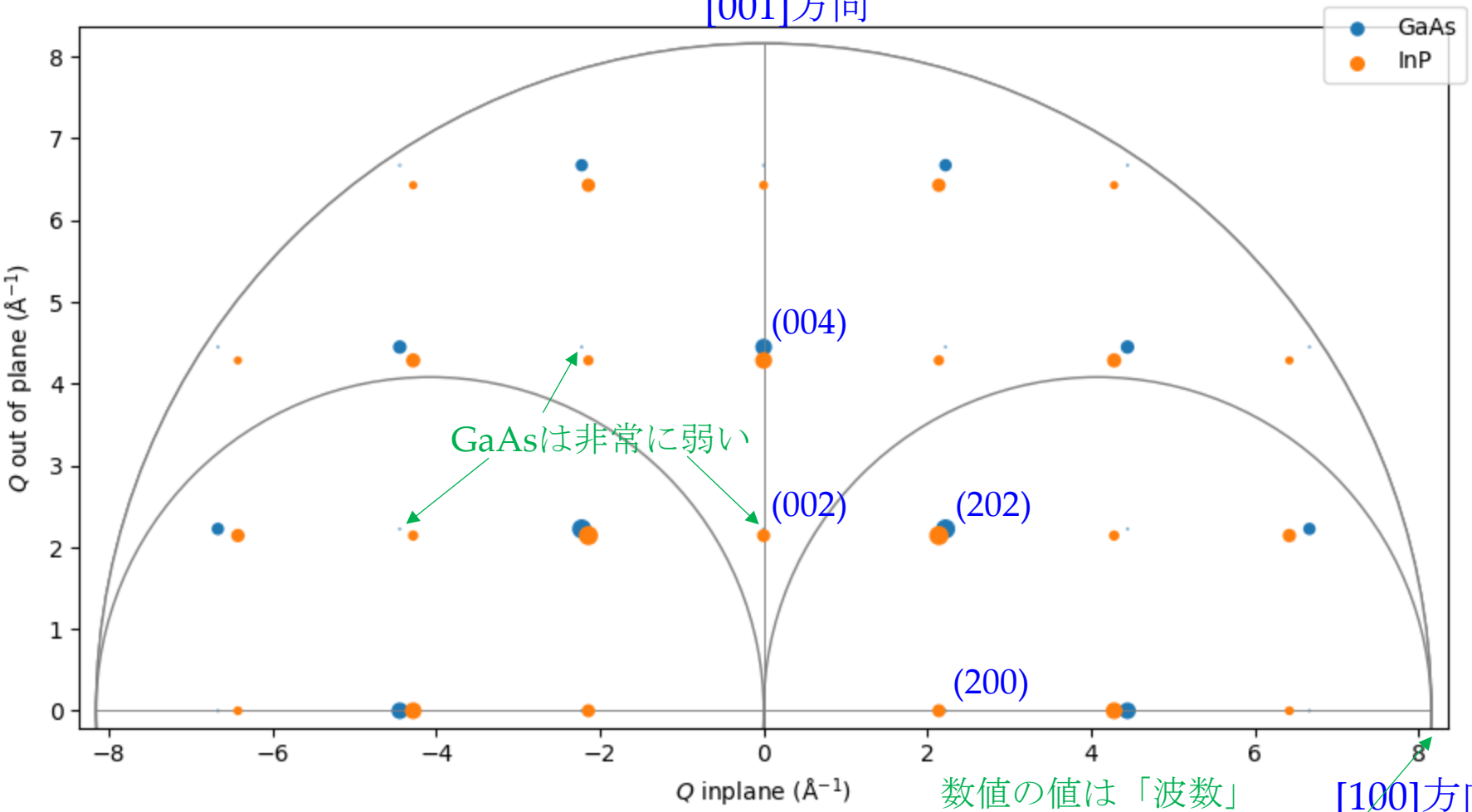
InP のプロットを重ねる

```
plt.show()
```

これがないと「図」が表示されない


GaAs
InP
Q out of plane (Å⁻¹)
Q inplane (Å⁻¹)
8
7
6
5
4
3
2
1
0
−8
−6
−4
−2
0
2
4
6
8

[001]方向



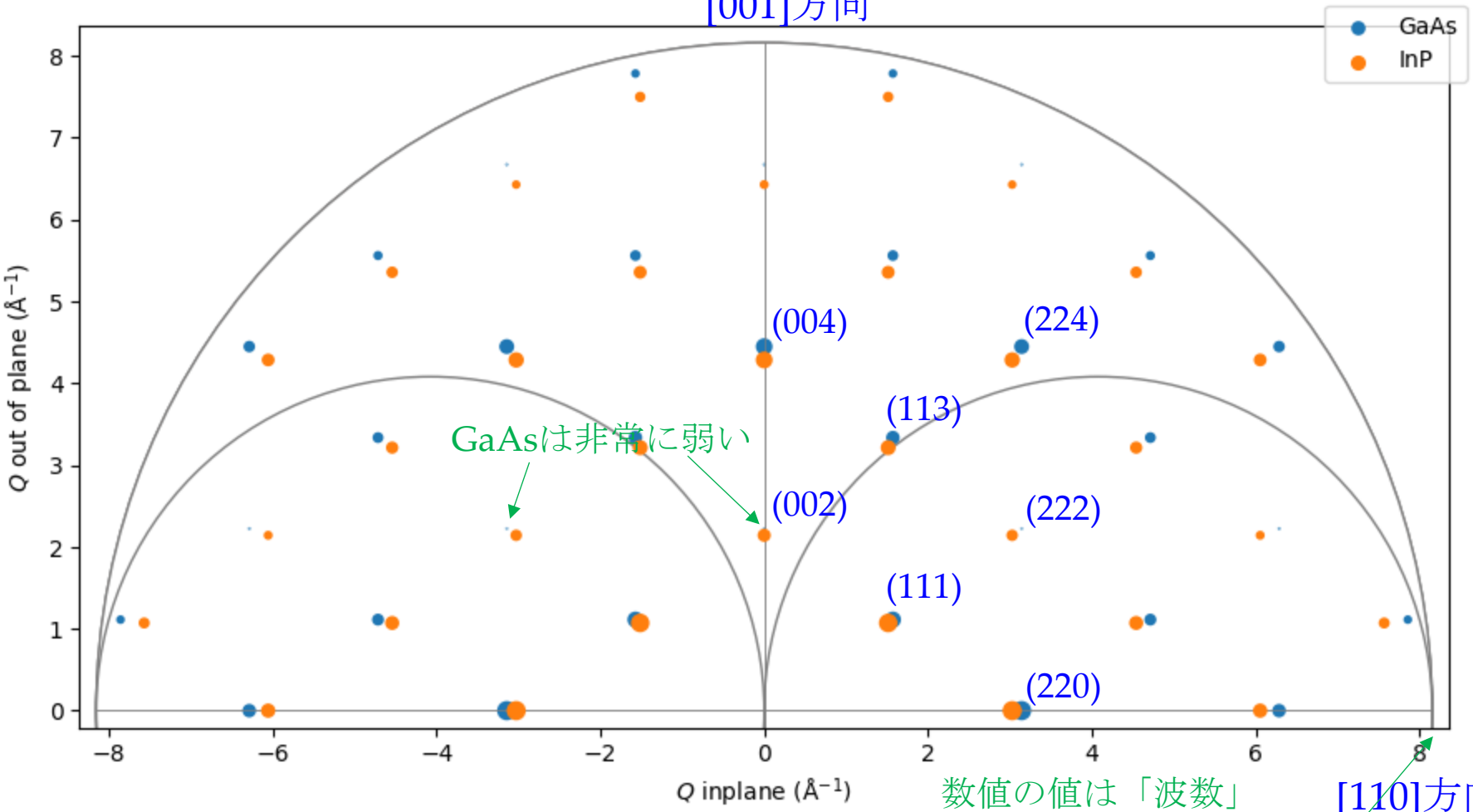
数値の値は「波数」

[100]方向

$E_{\mathrm{X-ray}} = 8048$ [eV] なので $|\boldsymbol{k}| = |\boldsymbol{k}'| = 4.076$ [Å^{-1}]

$|\Delta\boldsymbol{k}| = |\boldsymbol{k}' - \boldsymbol{k}| = Q$ の最大値は $\simeq 8.152$ [Å^{-1}]

[001]方向



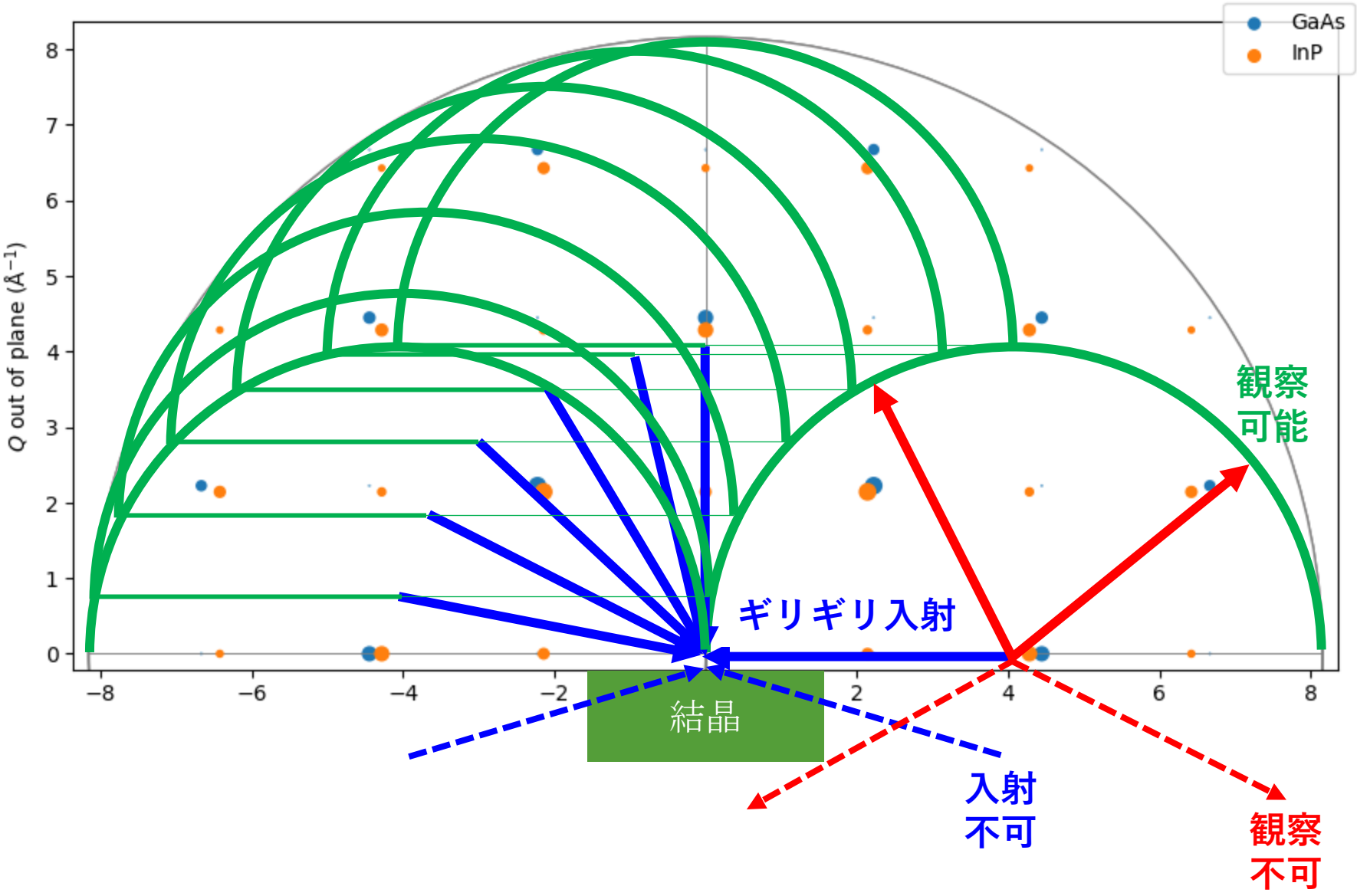
GaAsは非常に弱い

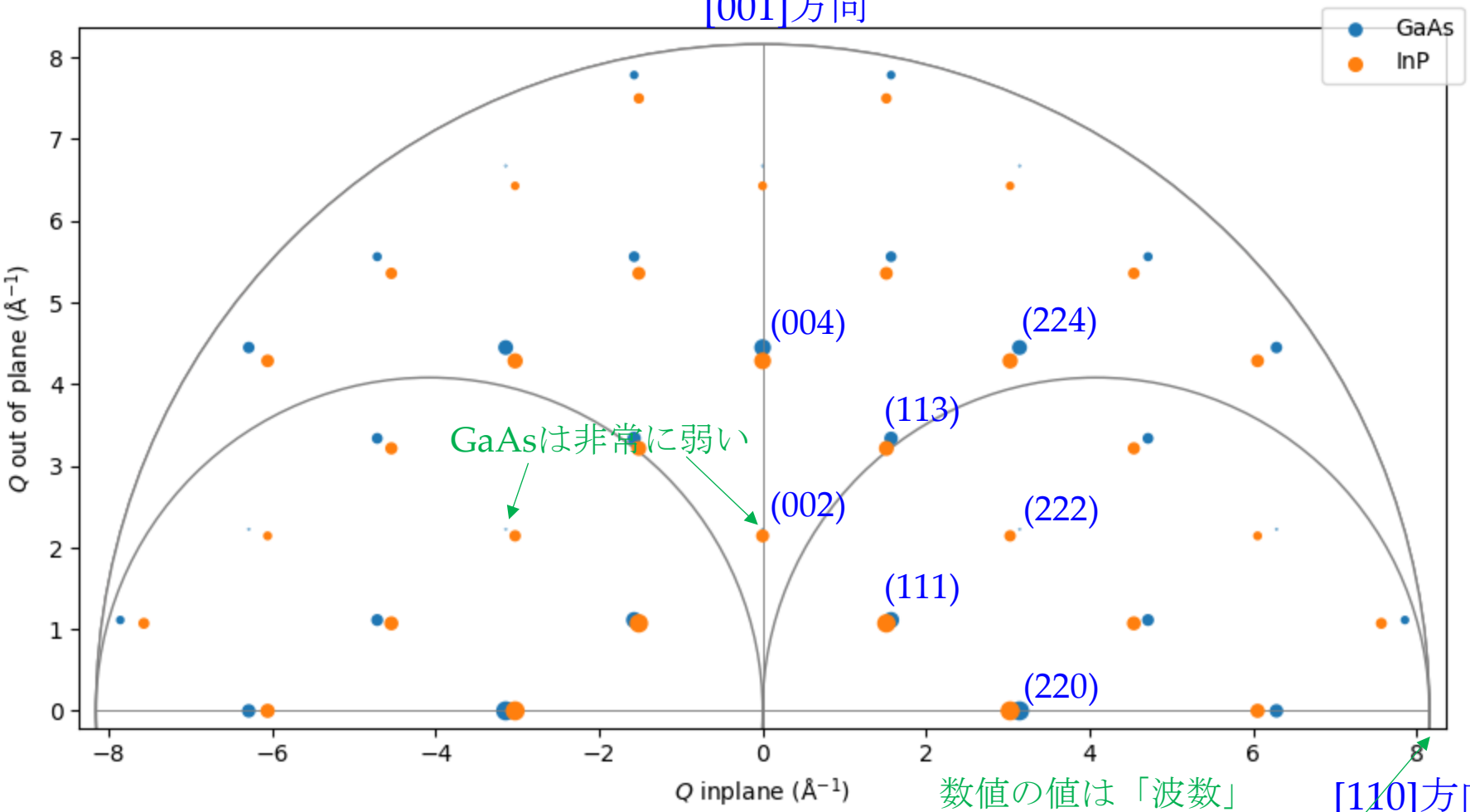
数値の値は「波数」

[110]方向

$E_{\mathrm{X-ray}} = 8048$ [eV] なので $|\boldsymbol{k}| = |\boldsymbol{k}'| = 4.076$ [Å^{-1}]

$|\Delta\boldsymbol{k}| = |\boldsymbol{k}' - \boldsymbol{k}| = Q$ の最大値は $\simeq 8.152$ [Å^{-1}]


GaAs
InP
Q out of plane (Å⁻¹)
8
7
6
5
4
3
2
1
0
−8
−6
−4
−2
2
4
6
8
観察
可能
ギリギリ入射
結晶
入射
不可
観察
不可



$E_{\mathrm{X-ray}} = 8048$ [eV] なので $|\boldsymbol{k}| = |\boldsymbol{k}'| = 4.076$ [Å^{-1}]

$|\Delta\boldsymbol{k}| = |\boldsymbol{k}' - \boldsymbol{k}| = Q$ の最大値は $\simeq 8.152$ [Å^{-1}]

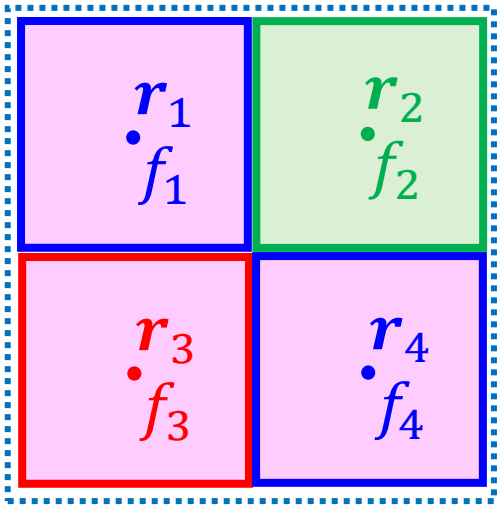
2. 構造因子 n_G の値は？ 人がやるなら？

$$n_G = \frac{1}{V_{\mathrm{cell}}} \int n(\boldsymbol{r}) \exp -i\boldsymbol{G}\cdot\boldsymbol{r}\ d\boldsymbol{r}$$

「ルート2」と同じ論法、戦略が使える

- 単位構造の中を分割する
- 分割したそれぞれの領域 j が原点にあった時の積分 $f_j = \int n(\boldsymbol{r}) \exp -i\boldsymbol{G}\cdot\boldsymbol{r}\ d\boldsymbol{r}$ を計算する
- 位相差を考慮して全部足す

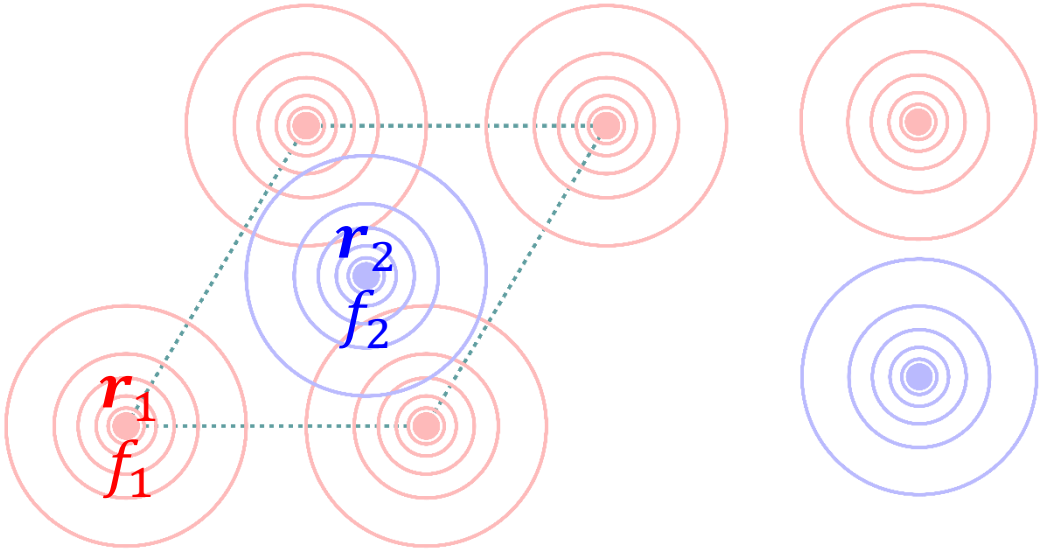
  $n_G = \Sigma_j\ f_j \exp -i\boldsymbol{G}\cdot\boldsymbol{r}_j$



2. 構造因子 n_G の値は？ 人がやるなら？

$n(\boldsymbol{r})$ を、原子が持つ電子密度分布に分解して考える。

$$n(\boldsymbol{r}) = \sum_{j\in\text{単位構造中の原子}} n_j(\boldsymbol{r}-\boldsymbol{r}_j)$$



$$f_1 = \int n_1(\boldsymbol{r}) \exp -i\boldsymbol{G}\cdot\boldsymbol{r}\ d\boldsymbol{r}$$

$$f_2 = \int n_2(\boldsymbol{r}) \exp -i\boldsymbol{G}\cdot\boldsymbol{r}\ d\boldsymbol{r}$$

$$n_{\boldsymbol{G}} = \sum_j f_j \exp -i\boldsymbol{G}\cdot\boldsymbol{r}_j = f_1 \exp -i\boldsymbol{G}\cdot\boldsymbol{r}_1 + f_2 \exp -i\boldsymbol{G}\cdot\boldsymbol{r}_2$$

2. 構造因子 n_G の値は？ 人がやるなら？

$$n_{\boldsymbol{G}} = \sum_j f_j \exp -i\boldsymbol{G}\cdot\boldsymbol{r}_j$$

f_i はどうする？

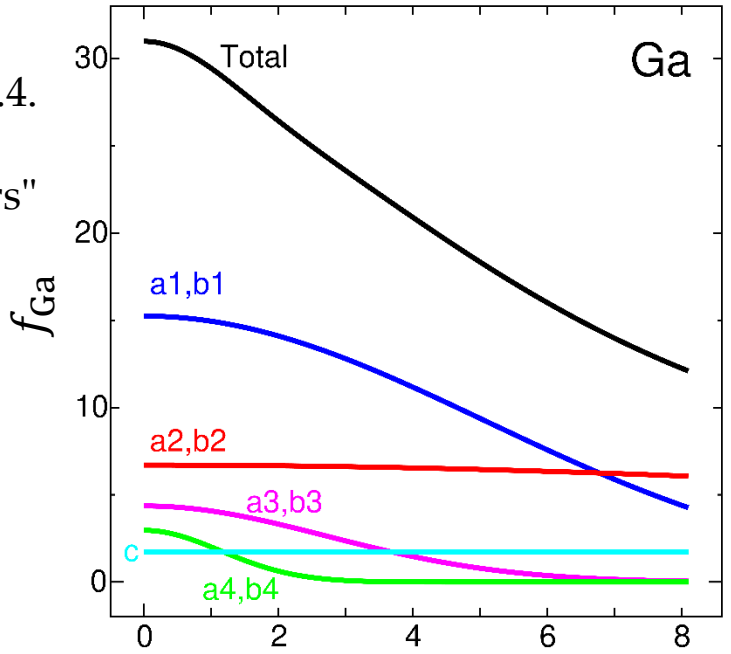
Ans. 1: $f_{原子} = \int n_{原子}(\boldsymbol{r}) \exp -i\boldsymbol{G}\cdot\boldsymbol{r}\ d\boldsymbol{r}$ は、もし $\exp -i\,\boldsymbol{G}\cdot\boldsymbol{r}$ が 1 なら
原子番号 Z (原子が持つ電子の数) に等しい

Ans. 2: f_i は高々原子の種類数(100個とか)しかないので数表が利用できる

International Tables for Crystallography, Vol. C. Table 6.1.1.4.
"Coefficients for analytical approximation
to the scattering factors"

$$f_{原子}(\sin\theta/\lambda) = \left(\sum_{i=1}^{4} a_i \exp -b_i \frac{\sin^2\theta}{\lambda^2}\right) + c$$

と近似して、その $a_i, b_i, c\ (i = 1 \sim 4)$ を
原子ごとに与えてくれる表。



($|\Delta\boldsymbol{k}| = 2|\boldsymbol{k}|\sin\theta = \frac{4\pi}{\lambda}\sin\theta$ なので $\frac{\sin\theta}{\lambda} = \frac{|\Delta\boldsymbol{k}|}{4\pi}$ だということに注意 ($\boldsymbol{G} = \Delta\boldsymbol{k}$))

2. 構造因子　具体的に $n_{\boldsymbol{G}} = \sum_j f_j \exp -i\boldsymbol{G}\cdot\boldsymbol{r}_j$

$$\boldsymbol{G} = h\boldsymbol{a}^* + k\boldsymbol{b}^* + l\boldsymbol{c}^*$$
$$\boldsymbol{r}_j = \alpha_j\boldsymbol{a} + \beta_j\boldsymbol{b} + \gamma_j\boldsymbol{c}$$
$$\boldsymbol{G}\cdot\boldsymbol{r}_j = 2\pi(h\alpha_j + k\beta_j + l\gamma_j)$$

$$\boldsymbol{a}\boldsymbol{a}^* = \boldsymbol{b}\boldsymbol{b}^* = \boldsymbol{c}\boldsymbol{c}^* = 2\pi$$
$$\boldsymbol{a}\boldsymbol{b}^* = \boldsymbol{a}\boldsymbol{c}^* = \boldsymbol{b}\boldsymbol{c}^* = \boldsymbol{b}\boldsymbol{a}^* = \boldsymbol{c}\boldsymbol{a}^* = \boldsymbol{c}\boldsymbol{b}^* = 0$$

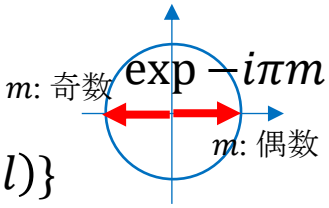
GaAs 単位格子中の原子位置(α, β, γ)

Ga : (0,0,0), (1/2, 1/2, 0), (1/2, 0, 1/2), (0, 1/2, 1/2)

As : Ga + (1/4, 1/4, 1/4)

⇒ 単位構造 Ga(0,0,0), As(1/4,1/4,1/4)が (0,0,0), (1/2, 1/2, 0), (1/2, 0, 1/2), (0, 1/2, 1/2) にある、と見る。

$$f_{\mathrm{Unit}} = f_{\mathrm{Ga}} + f_{\mathrm{As}} \exp -i\frac{\pi}{2}(h+k+l)$$
$$n_{\boldsymbol{G}} = f_{\mathrm{Unit}}\{1 + \exp -i\pi(h+k) + \exp -i\pi(h+l) + \exp -i\pi(k+l)\}$$

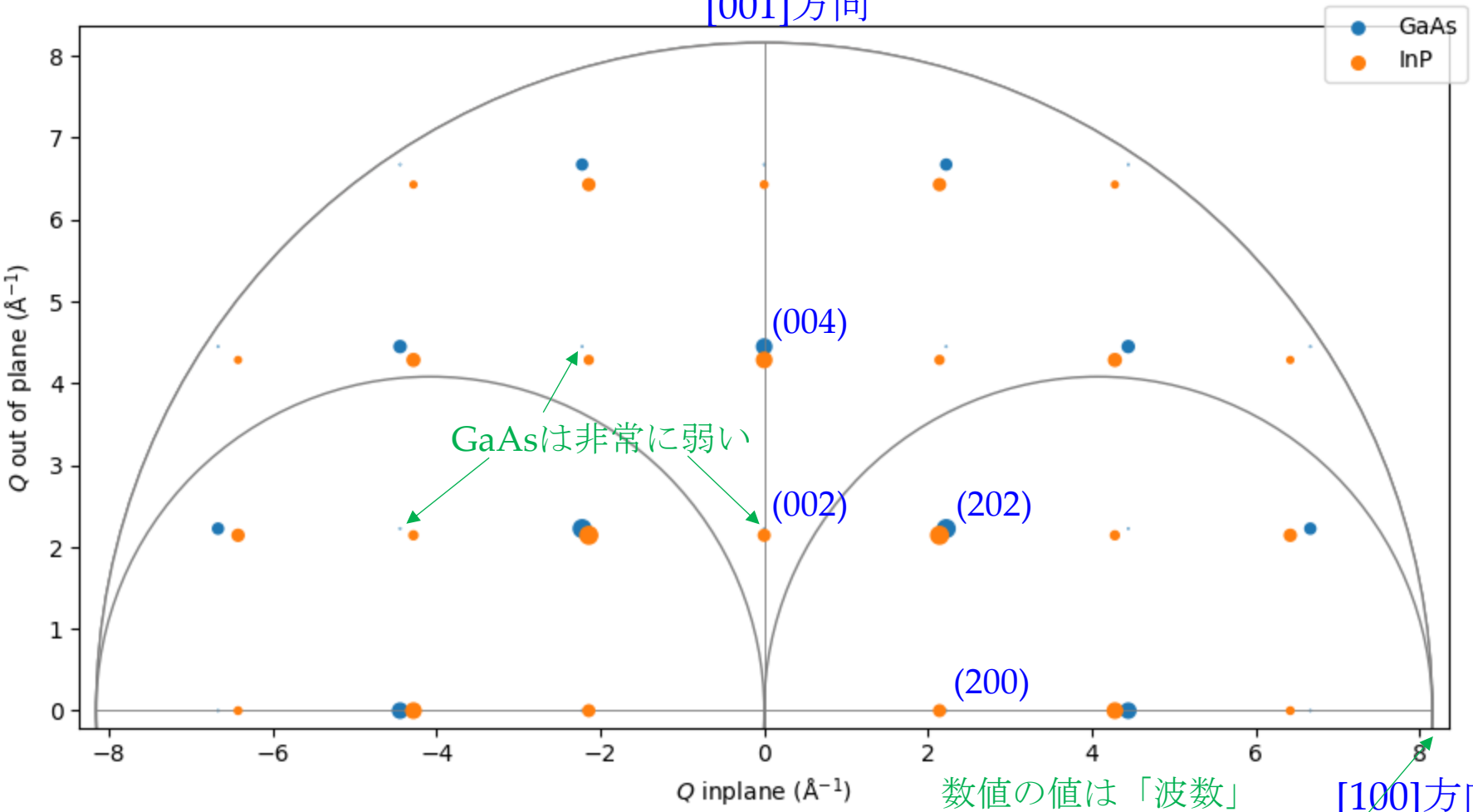


h, k, l : 偶奇混合　⇒ $n_G = 0$　: 面心立方由来の禁制。中々崩れない

全て奇数 or 全て偶数 ⇒ $n_G = 4f_{\mathrm{Unit}}$

h, k, l : 全て奇数	$f_{\mathrm{Unit}} = f_{\mathrm{Ga}} \pm if_{\mathrm{As}}$		中ぐらい
全て偶数	$f_{\mathrm{Unit}} = f_{\mathrm{Ga}} + f_{\mathrm{As}}$	$(h+k+l = 4m)$	強い
又は	$f_{\mathrm{Ga}} - f_{\mathrm{As}}$	$(h+k+l = 4m \pm 2)$	弱い

[001]方向



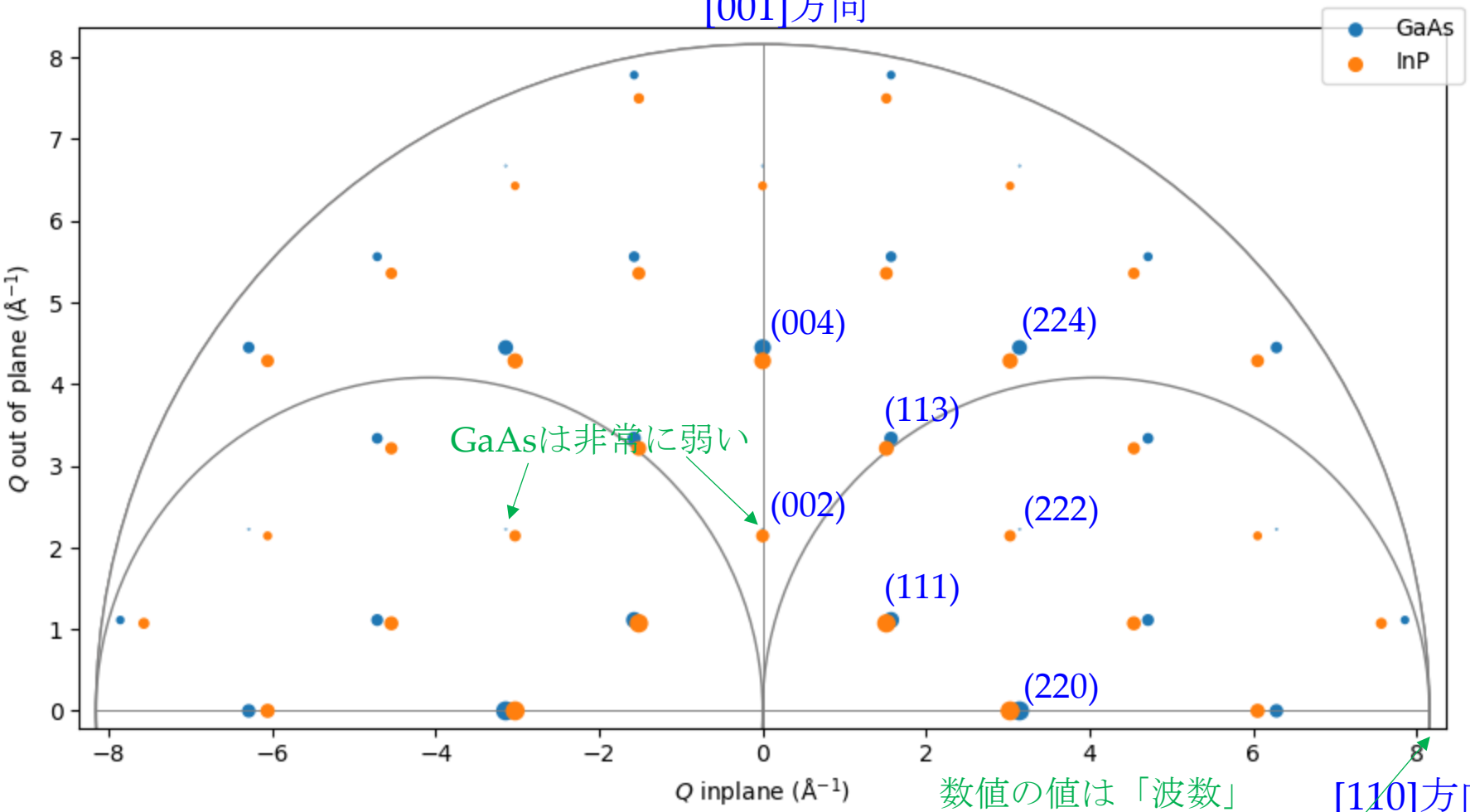
GaAsは非常に弱い

数値の値は「波数」

[100]方向

$E_{\mathrm{X-ray}} = 8048$ [eV] なので $|\boldsymbol{k}| = |\boldsymbol{k}'| = 4.076$ [Å^{-1}]

$|\Delta\boldsymbol{k}| = |\boldsymbol{k}' - \boldsymbol{k}| = Q$ の最大値は $\simeq 8.152$ [Å^{-1}]



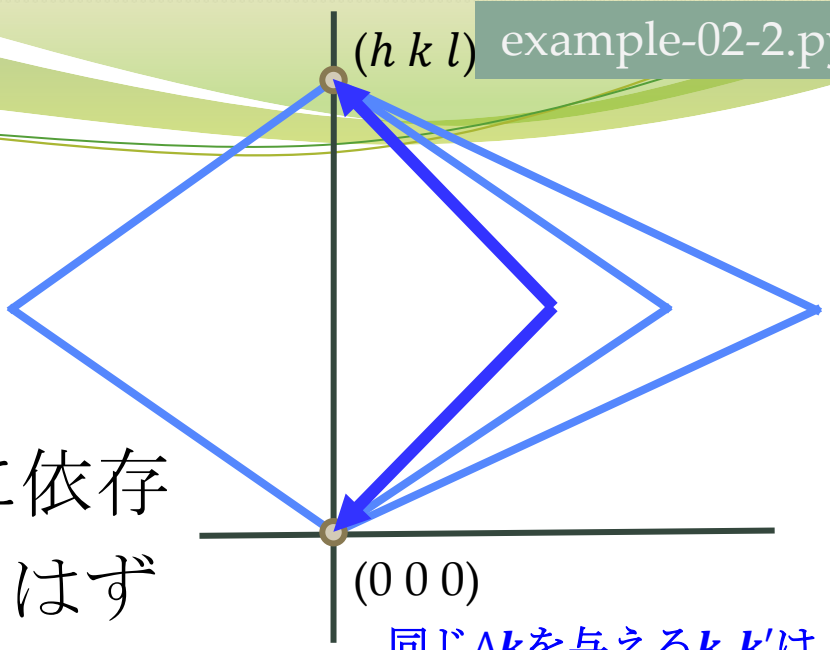
$E_{\mathrm{X-ray}} = 8048$ [eV] なので $|\boldsymbol{k}| = |\boldsymbol{k}'| = 4.076$ [Å^{-1}]

$|\Delta\boldsymbol{k}| = |\boldsymbol{k}' - \boldsymbol{k}| = Q$ の最大値は $\simeq 8.152$ [Å^{-1}]

example-02-2.py

2. 構造因子

$$\frac{1}{V_{\mathrm{cell}}}\int n(\boldsymbol{r})\exp i\boldsymbol{G}\cdot\boldsymbol{r}\,d\boldsymbol{r}$$



同じ$\Delta\boldsymbol{k}$を与える$\boldsymbol{k},\boldsymbol{k}'$は
いくらでもあり得る

構造因子は $\boldsymbol{G}$ $(=\Delta\boldsymbol{k}\equiv\boldsymbol{Q})$ に依存

$\boldsymbol{k},\boldsymbol{k}'$ には依存していない… はず

($\boldsymbol{k}$ なら $k=\frac{2\pi}{\lambda}=2\pi\frac{E}{hc}$)

X線(電磁波) :

原子核のポテンシャルに束縛された電子を揺らす

- そのままエネルギーを励起に使われる = 吸収
- 振動した電子からの再放射 = 位相変化

構造因子(慣用的な記号) $f=f_0+f'+if''$

電子密度の積分項

吸収による強度変化 (実部)
再放射による位相変化 (虚部)

2. 構造因子

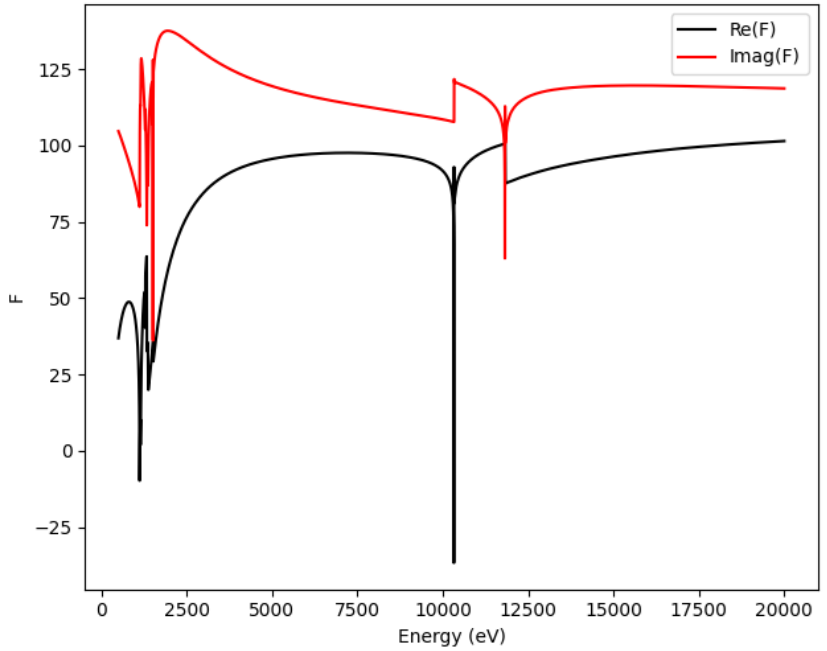
原子散乱因子 $f = f_0 + f' + if''$

f_0: 電子密度の積分項

f': 吸収による強度変化 (実部)

if'': 再放射による位相変化 (虚部)

```
import xrayutilities as xu
import numpy
import matplotlib.pyplot as plt

GaAs = xu.materials.GaAs

energy= numpy.linspace(500, 20000, 5000) # 500 - 20000 eV
F = GaAs.StructureFactorForEnergy(GaAs.Q(1, 1, 1), energy)

plt.figure(); plt.clf()
plt.plot(energy, F.real, '-k', label='Re(F)')
plt.plot(energy, F.imag, '-r', label='Imag(F)')
plt.xlabel("Energy (eV)"); plt.ylabel("F"); plt.legend()
plt.show()
```



AlN(epi)/GaN(sub)を計算してみる

4. $\omega - 2\theta$スペクトルの計算

ここまでは xu.materials の

- Element (Ga, In, N,...)
- Crystal (GaAs, InP, GaN, AlN,...)

を使って、計算を行ってきた。

回折スペクトルの計算は xu.simpack の

- Layer (AlN/GaN,...)

を使う。

```
GaN = xu.materials.Crystal.fromCIF("CIFs/GaN-Hex.cif")
                                          + 弾性に関する情報
```

```
sub = xu.simpack.Layer( GaN, float('inf') )              # 基板 (GaN)
lAlN = xu.simpack.Layer( AlN, 1500, relaxation=1.0 )     # 膜 (AlN)
```

```
Epi_Sub = xu.simpack.PseudomorphicStack001( "AlN/GaN", sub+lAlN )
```

Layer を構成する物質の「弾性」が分かっているので、「歪」が計算に入る

AlN(epi)/GaN(sub)を計算してみる

4. $\omega - 2\theta$スペクトルの計算

```
import numpy as np
import matplotlib.pyplot as plt
import xrayutilities as xu
```

```
# GaN, AlN の結晶構造をCIFから読む
GaN = xu.materials.Crystal.fromCIF("CIFs/GaN-Hex.cif")
AlN = xu.materials.Crystal.fromCIF("CIFs/AlN-Hex.cif")

# 今作った結晶(GaN, AlN)は、結晶構造の情報しか持ってないので
# 積層構造にした時の歪の計算ができるように、弾性に関わる情報を追加する
## どちらも Hexagonal
GaN.symm_class_name = 'Hexagonal'
AlN.symm_class_name = 'Hexagonal'

## GaN, AlN の弾性定数 Cij の値を、名前のついた値として準備
C11_GaN, C12_GaN, C13_GaN, C33_GaN, C44_GaN = 390, 145, 106, 398, 105
C11_AlN, C12_AlN, C13_AlN, C33_AlN, C44_AlN = 396, 137, 108, 373, 116

## xu にサポートしてもらって 2階、4階テンソルの形に組み上げる
cij_GaN = xu.materials.material.HexagonalElasticTensor( C11_GaN, C12_GaN, C13_GaN, C33_GaN, C44_GaN )
cij_AlN = xu.materials.material.HexagonalElasticTensor( C11_AlN, C12_AlN, C13_AlN, C33_AlN, C44_AlN )

cijkl_GaN = xu.materials.material.Cij2Cijkl( cij_GaN )
cijkl_AlN = xu.materials.material.Cij2Cijkl( cij_AlN )

## Crystal オブジェクト(GaN, AlN)に2階、4階テンソルを属性として設定
setattr(GaN, 'cijkl', cijkl_GaN)
setattr(AlN, 'cijkl', cijkl_AlN)
GaN.elastic = cij_GaN
AlN.elastic = cij_AlN
```

- このページの内容は全て「Layer」を作る準備
- 素材になる GaN, AlN という「Crystal」を用意

AlN(epi)/GaN(sub)を計算してみる

4. $\omega - 2\theta$スペクトルの計算

```
# ここまでで、弾性定数の情報も持った GaN, AlN という結晶が準備できた。
# 次は、この物質のを積み上げていくためにその部品として Layer を作り、
# それを積み上げて積層構造にする
```

sub, lAlN: GaN, AlN の Layer

```
## まずは 基板(厚さ無限のGaN Layer)、AlN層を準備
sub = xu.simpack.Layer( GaN, float('inf') )          # 基板 (GaN)
lAlN = xu.simpack.Layer( AlN, 1500, relaxation=1.0 )  # 膜 (AlN)
## 準備した Layer を積み上げて AlN層 / GaN 基板という積層構造を構築
Epi_Sub = xu.simpack.PseudomorphicStack001( "AlN/GaN", sub+lAlN )
```

Epi_sub: lAlN/sub の積層構造

```
# ω-2θ 強度計算
## 波長、エネルギー、計算する角度範囲指定
wavelength = xu.wavelength('CuKa1')   # 波長は Cu Ka1 の波長(1.5405...A)
energy     = 12390 / wavelength       # エネルギーは波長から計算
twotheta   = np.linspace(32,38,200)   # 横軸: 角度範囲のベクトルを作っておく
# 動力学モデルの設定とシミュレーション
Epi_Sub_Model = xu.simpack.DynamicalModel(
                                    Epi_Sub, energy=energy, resolution_width=0.0001 )
Int_Epi_Sub_Model = Epi_Sub_Model.simulate( twotheta/2, hkl=(0, 0, 2) )
```

指定が無ければ$\omega - 2\theta$スキャン

AlN(epi)/GaN(sub)を計算してみる

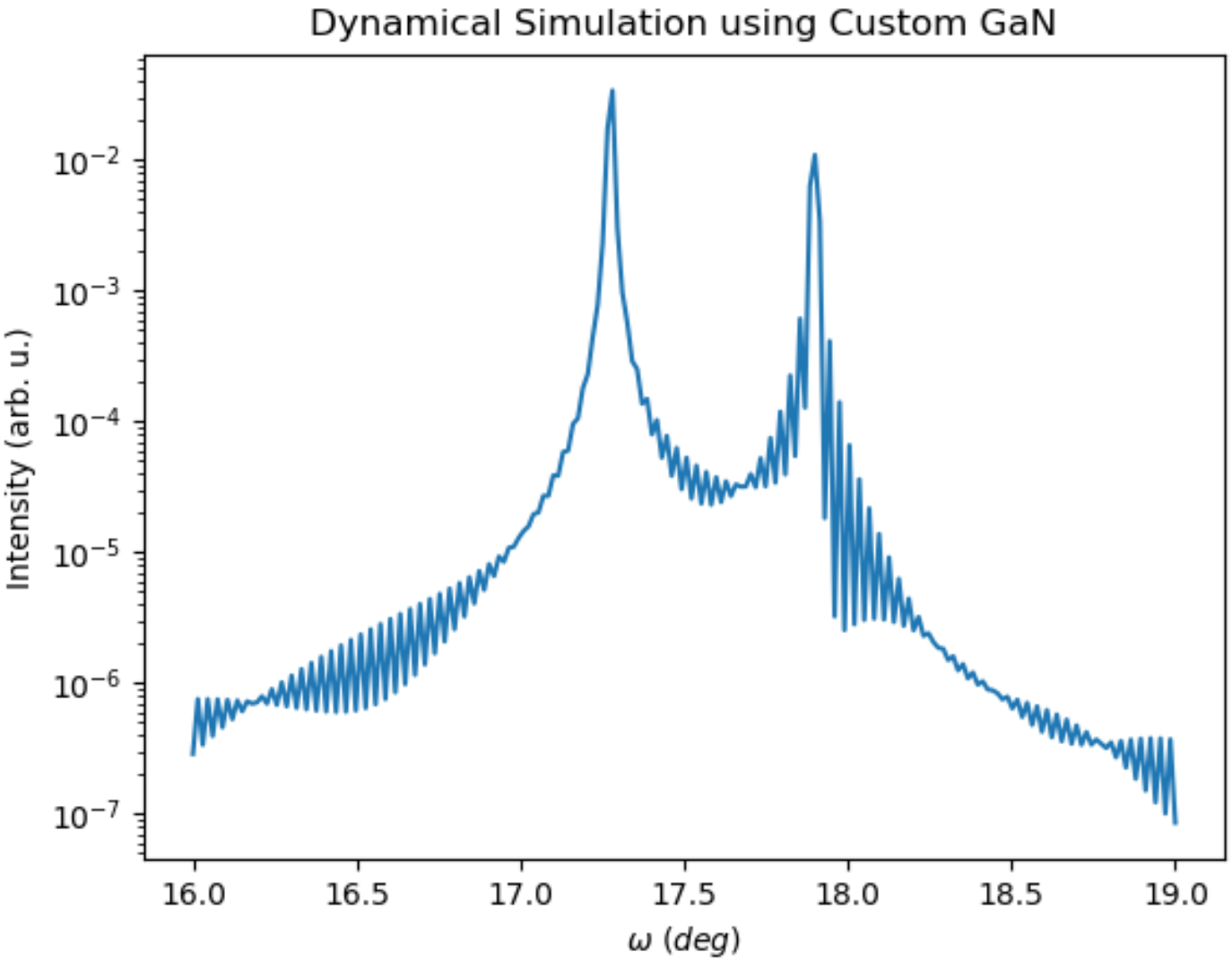
4. $\omega - 2\theta$スペクトルの計算

計算された回折強度 Int_Ep_Sub_Mode を twotheta/2 に対してプロットして表示

```
# 結果のプロット
plt.figure()
plt.plot( twotheta / 2, Int_Epi_Sub_Model )
plt.title( 'Dynamical Simulation of AlN(1500nm)/GaN Sub' )
plt.yscale( 'log' )
plt.xlabel( r'$¥omega¥ (deg)$' )
plt.ylabel( 'Intensity (arb. u.)' )
plt.show()
```

AlN(epi)/GaN(sub)を計算してみる

4. $\omega - 2\theta$スペクトルの計算



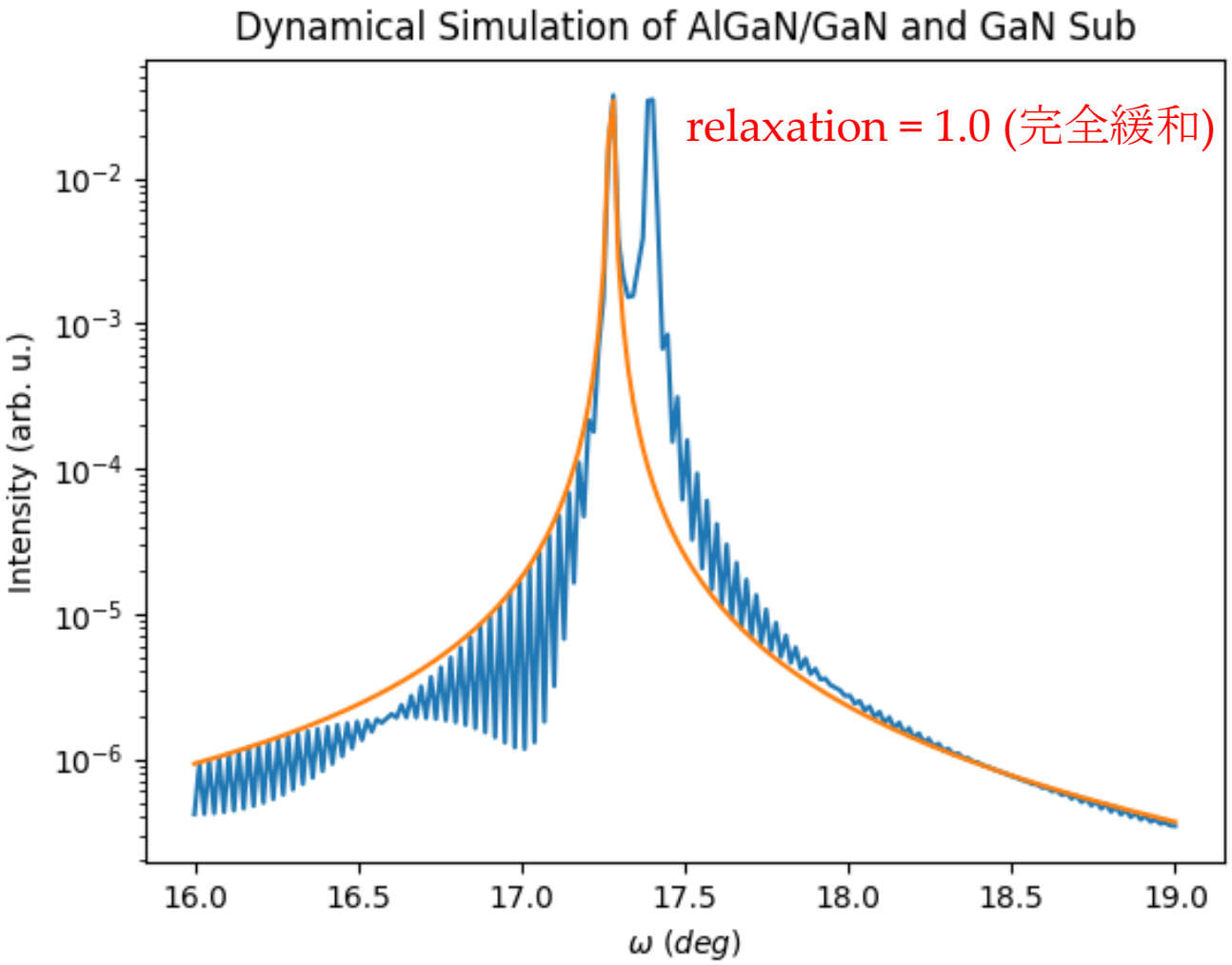
$Al_xGa_{1-x}N$(epi)/GaN(sub)を計算してみる

4. $\omega - 2\theta$スペクトルの計算

```
GaN.elastic = cij_array_GaN
AlN.elastic = cij_array_AlN
# ここまでは、先の例と全く同じ

# さらに混晶も準備する
Alx = 0.2                                                    # Al組成 x = 0.2
AlGaN = xu.materials.material.Alloy(GaN, AlN, Alx);

#これ以降は AlN => AlGaN 以外はほぼ同じ(それに合わせて、変数名を少し変えた)
## まずは 基板(厚さ無限のGaN Layer)、AlGaN層を準備
sub    = xu.simpack.Layer( GaN, float('inf')
lAlGaN = xu.simpack.Layer( AlGaN, 1500, relaxation=1.0 )
                        # relaxation 1.0(完全緩和) => 0.0 回折ピークは広角側へ

## 準備した Layer を積み上げて AlGaN層 / GaN 基板という積層構造を構築
Epi_Sub = xu.simpack.PseudomorphicStack001( "AlGaN/GaN", sub+lAlGaN )

# 以下(シミュレーションとその結果の表示)はまた AlN/GaN の例と同じ
```

$Al_xGa_{1-x}N$(epi)/GaN(sub)を計算してみる

4. $\omega - 2\theta$スペクトルの計算

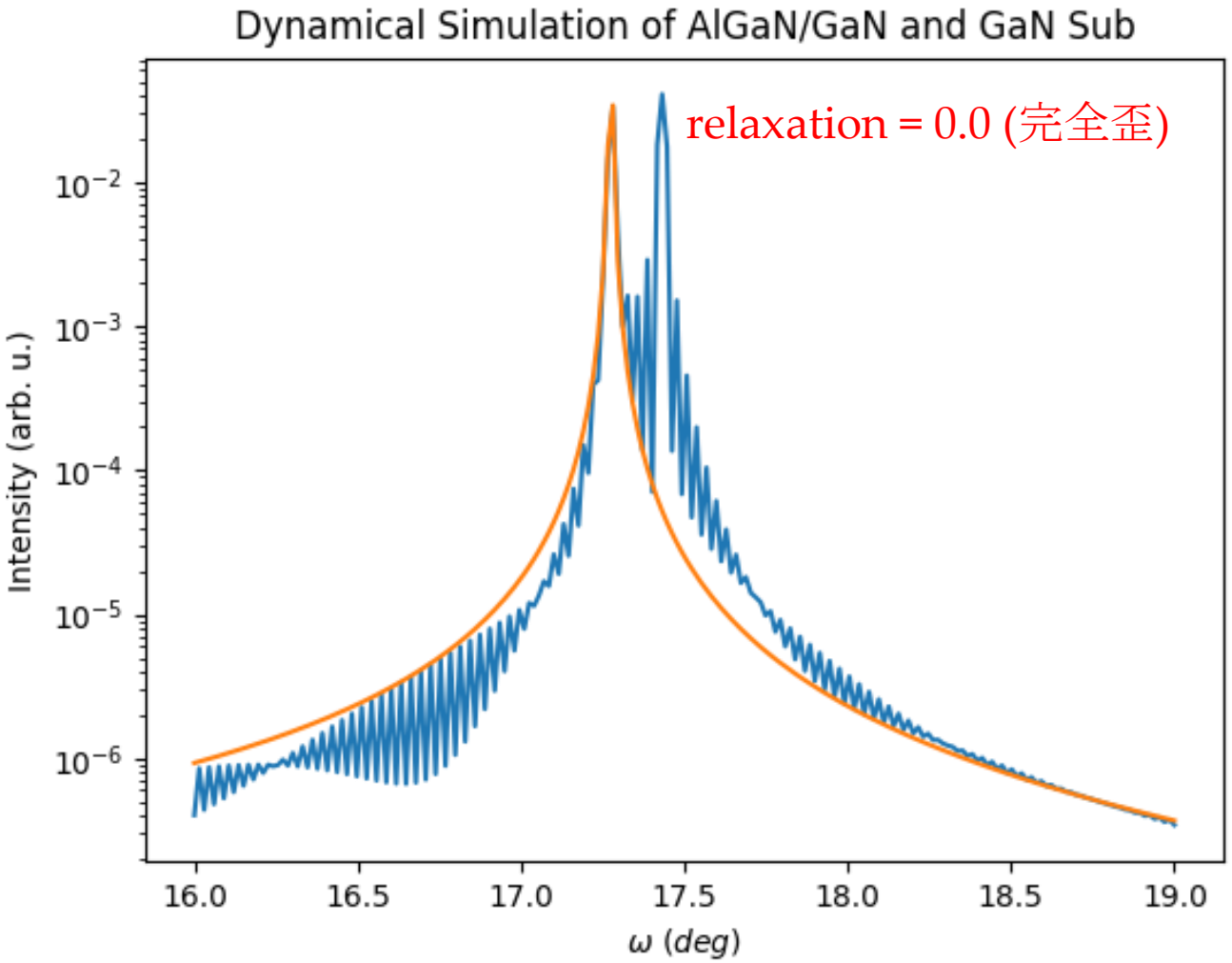


$Al_xGa_{1-x}N$(epi)/GaN(sub)を計算してみる

4. $\omega - 2\theta$スペクトルの計算

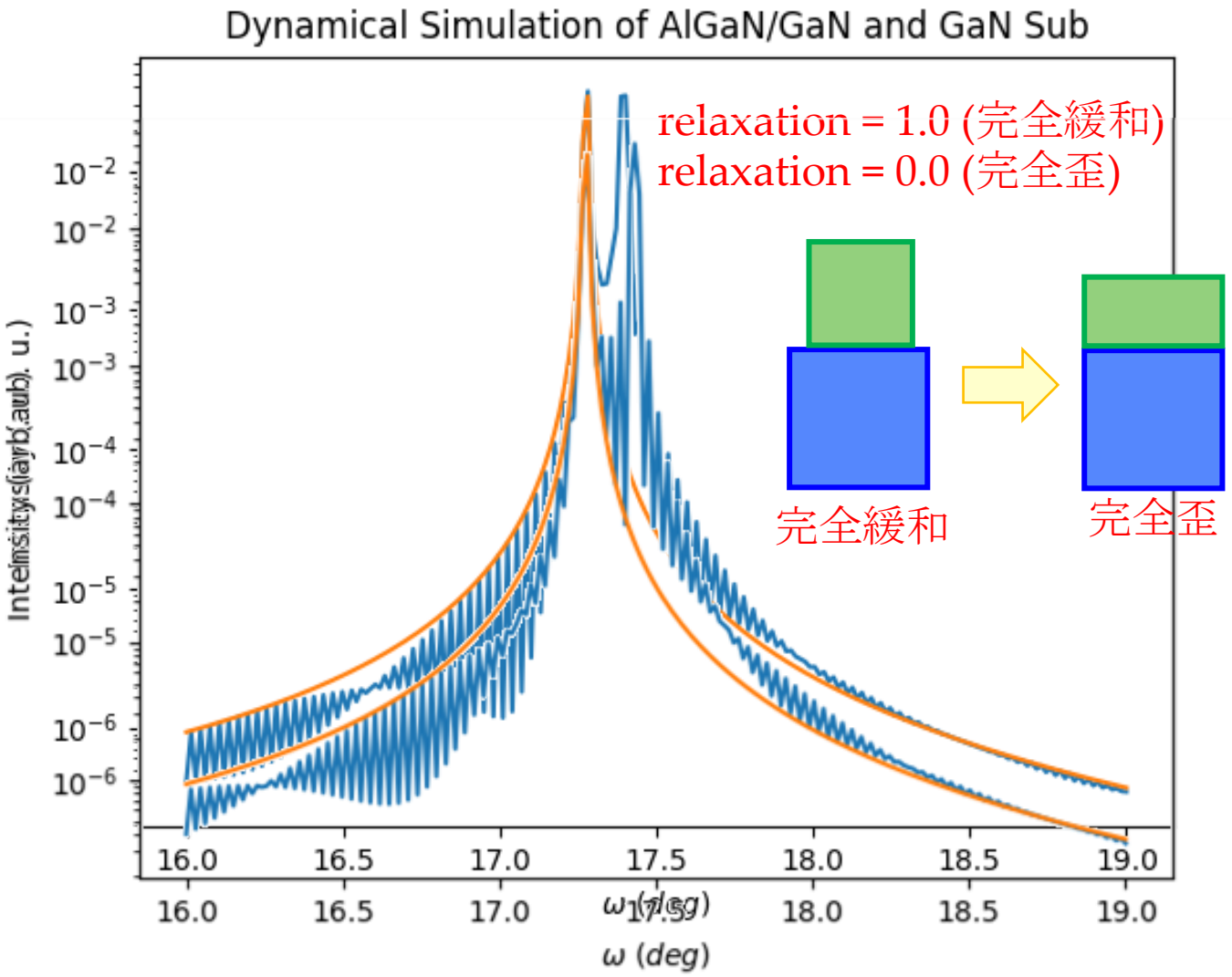
ここだけ変更!!

```
lAlGaN = xu.simpack.Layer( AlGaN, 1500, relaxation=0.0 )
```



$Al_xGa_{1-x}N$(epi)/GaN(sub)を計算してみる

4. $\omega - 2\theta$スペクトルの計算



どうせならフィッティング(のシミュレーション)

4. $\omega - 2\theta$スペクトルの計算

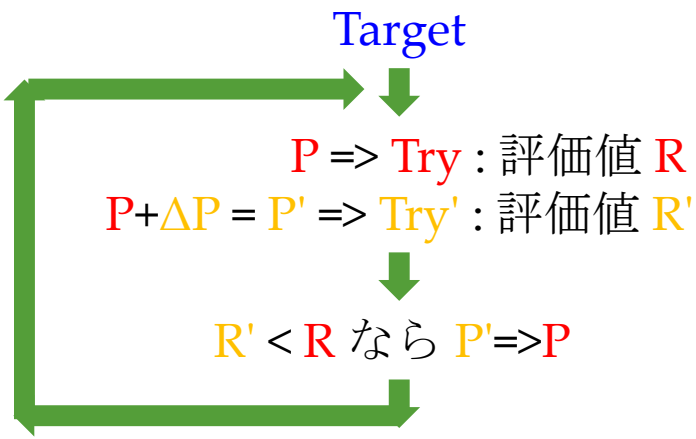
- xu で多層膜試料の回折スペクトルを一つ生成(Target)
- Target を仮想「測定スペクトル」として、
  Epi 層の組成、膜厚、緩和度をパラメーたにしたフィッティング
- お手軽にRMC(reverse monte-carlo)でフィッティング

  (多くの非線形最小二乗や降下法の類の様に、
  差分を使う方法は、細かな振動構造を持つ
  回折スペクトルの解析に使いにくい)

どうせならフィッティング(のシミュレーション)

4. $\omega - 2\theta$スペクトルの計算

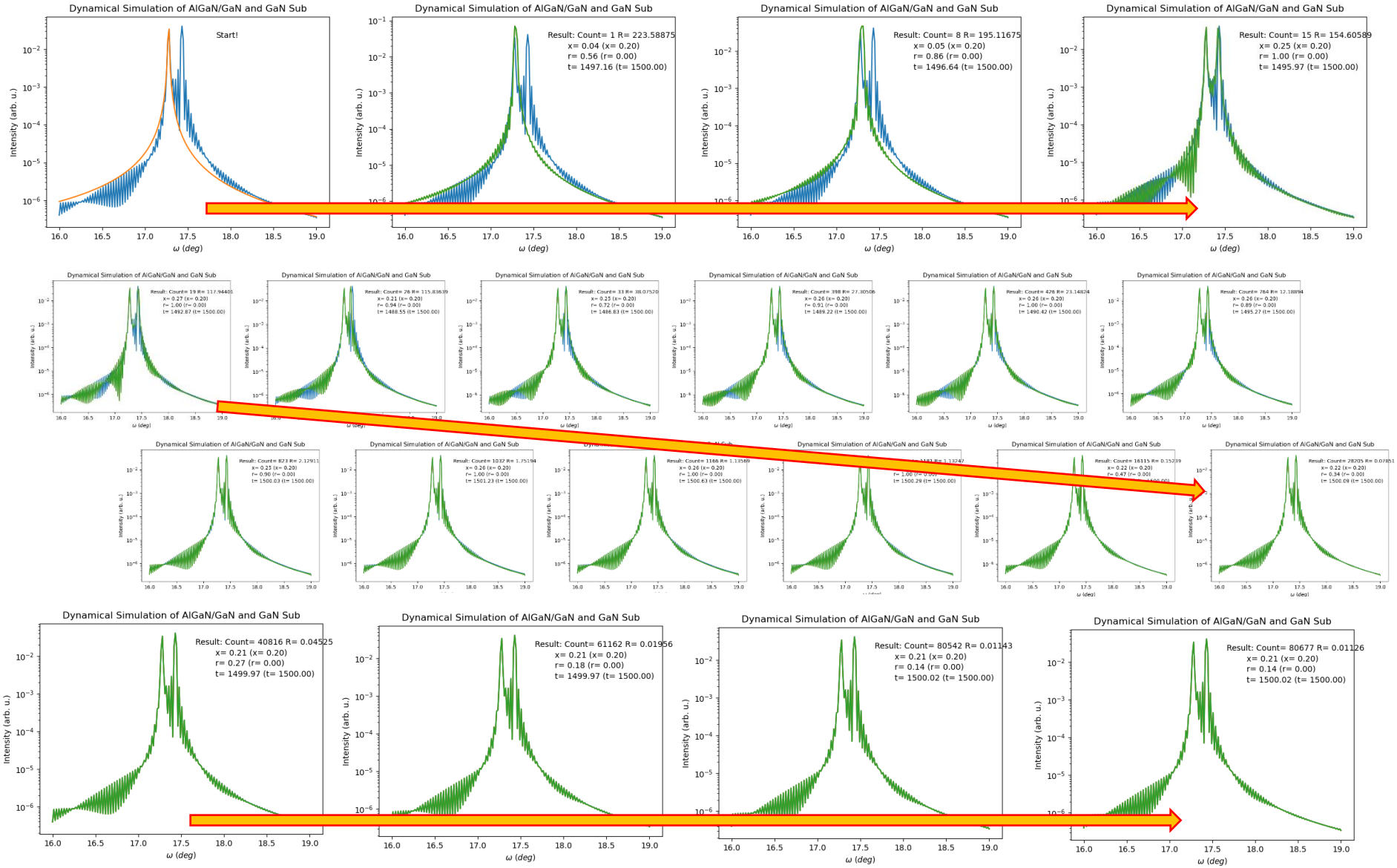
1. パラメータ P から xu でスペクトル Try を計算、Target との残差を R とする
2. P を乱数で更新し P' を生成
3. パラメータ P' から xu でスペクトル Try' を計算、Target との残差を R' とする
4. 二つの残差 R と R' を比較し、小さい残差を与えたパラメータを新たに P とする
5. 終了条件を満たすまで 1. に戻って繰り返す
   終了条件 : R < R' (P' が採用されない) が一定回数以上連続



example-04-3.py

どうせならフィッティング(のシミュレーション)

4. $\omega - 2\theta$スペクトルの計算



4. ロッキングカーブの計算

```
## 波長、エネルギー、計算する角度範囲指定
...
#### ここまでは先にやった omega-2theta のシミュレーションの時と同一 ####

# 動的モデルの設定とシミュレーション
## リファレンスに基板(GaN)だけの計算もする
# まずは計算の為にモデルを作って
sub_model          = xu.simpack.DynamicalModel( sub,
                                                energy=energy, resolution_width=0.0001 )
# 一旦、ω－2θ の計算をする
int_w2th_sub       = sub_model.simulate( two_theta/2, hkl=(0, 0, 2) )

# 計算結果を見て、ピーク位置(配列の指数)とその角度を決める
peak_index_sub     = np.argmax( int_w2th_sub )
peak_sub           = two_theta[ peak_index_sub ]

# 決まった角度を中心に計算する ω の範囲を決めて
omega_sub          = np.linspace( peak_sub/2-0.5, peak_sub/2+0.5, 400 )
# ロッキングカーブの計算
int_RC_sub         = sub_model.simulate( omega_sub, hkl=(0,0,2),
                                                    geometry="omega" )
```

ω－2θの計算時にはなかった
geometry="omega" を入れて
ロッキングカーブの計算

4. ロッキングカーブの計算

```
## こちらは Epi/Sub = AlGaN/GaN の計算
# まずは計算の為にモデルを作って
epi_sub_model       = xu.simpack.DynamicalModel( Epi_Sub, energy=energy,
                                                 resolution_width=0.0001 )
# 一旦、ω－2θ の計算をする
int_w2th_epi_sub    = epi_sub_model.simulate( two_theta/2, hkl=(0, 0, 2) )

# 計算結果を見て、ピーク位置(配列の指数)とその角度を決める
peak_index_epi_sub = np.argmax( int_w2th_epi_sub )
peak_epi_sub       = two_theta[ peak_index_epi_sub ]

# 決まった角度を中心に計算する ω の範囲を決めて
omega_epi_sub       = np.linspace( peak_epi_sub/2-0.5, peak_epi_sub/2+0.5, 400 )
# ロッキングカーブの計算
int_RC_epi_sub      = epi_sub_model.simulate( omega_epi_sub, hkl=(0,0,2),
                                              geometry="omega" )
```

$\omega - 2\theta$の計算時にはなかった
geometry="omega" を入れて
ロッキングカーブの計算

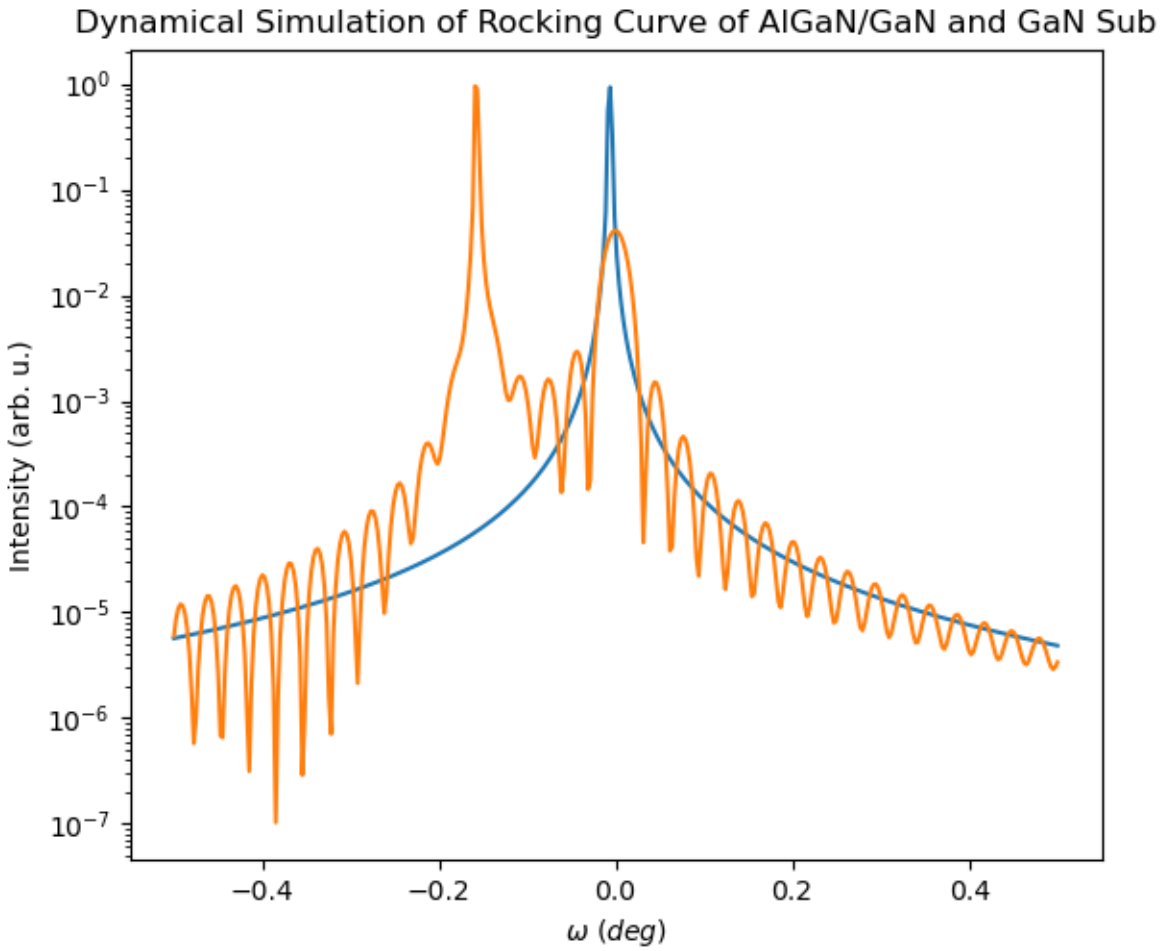
4. ロッキングカーブの計算

横軸の角度範囲は
それぞれ違う

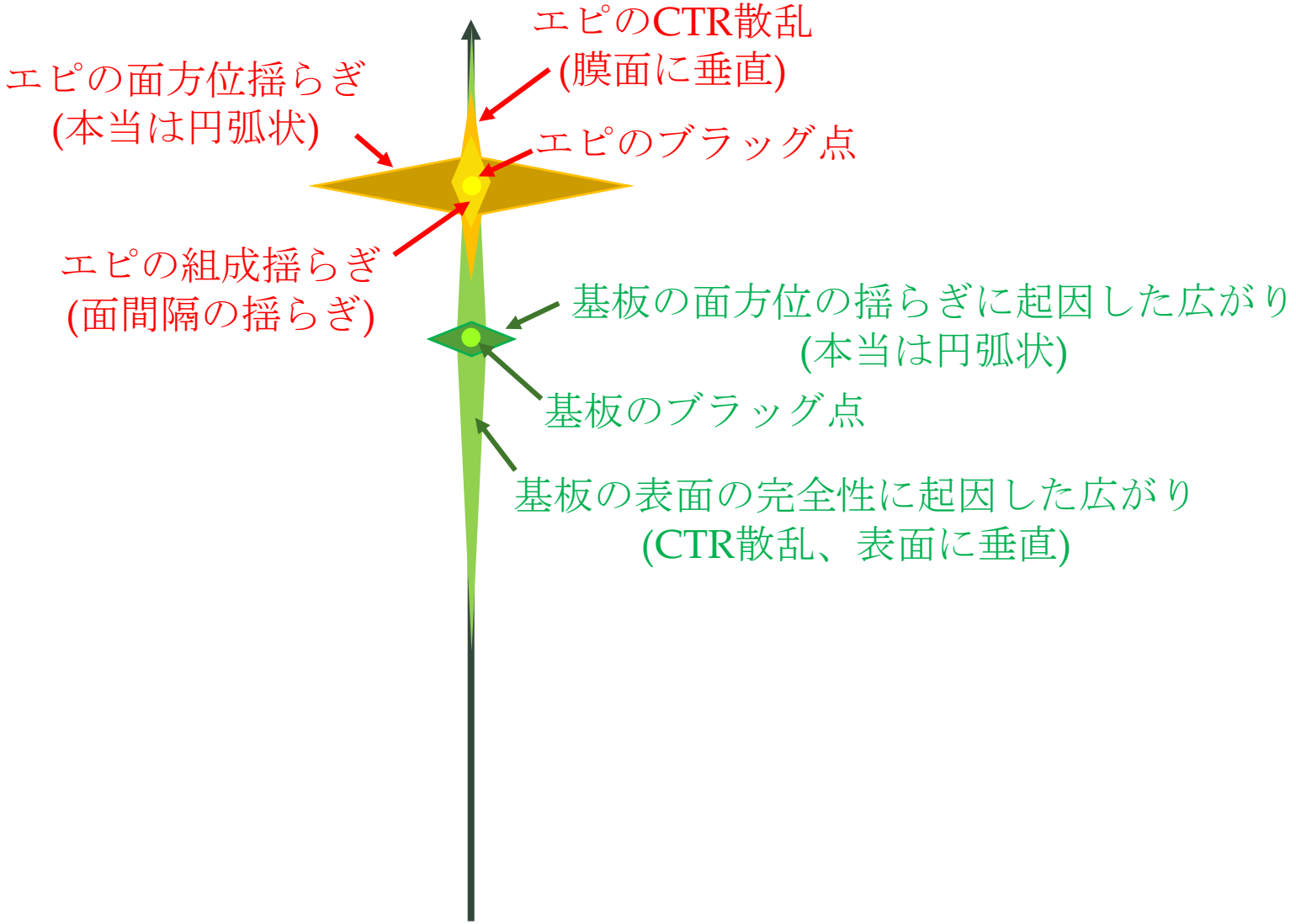
ピークが 0 度になるよう
シフトさせておく

```
# 結果のプロット
plt.figure()
plt.plot( omega_sub     - peak_sub/2,     int_RC_sub )
plt.plot( omega_epi_sub - peak_epi_sub/2, int_RC_epi_sub )
plt.title( 'Dynamical Simulation of Rocking Curve of
AlGaN/GaN and GaN Sub')
plt.yscale('log')
plt.xlabel(r'$¥omega¥ (deg)$')
plt.ylabel('Intensity (arb. u.)')
plt.show()
```

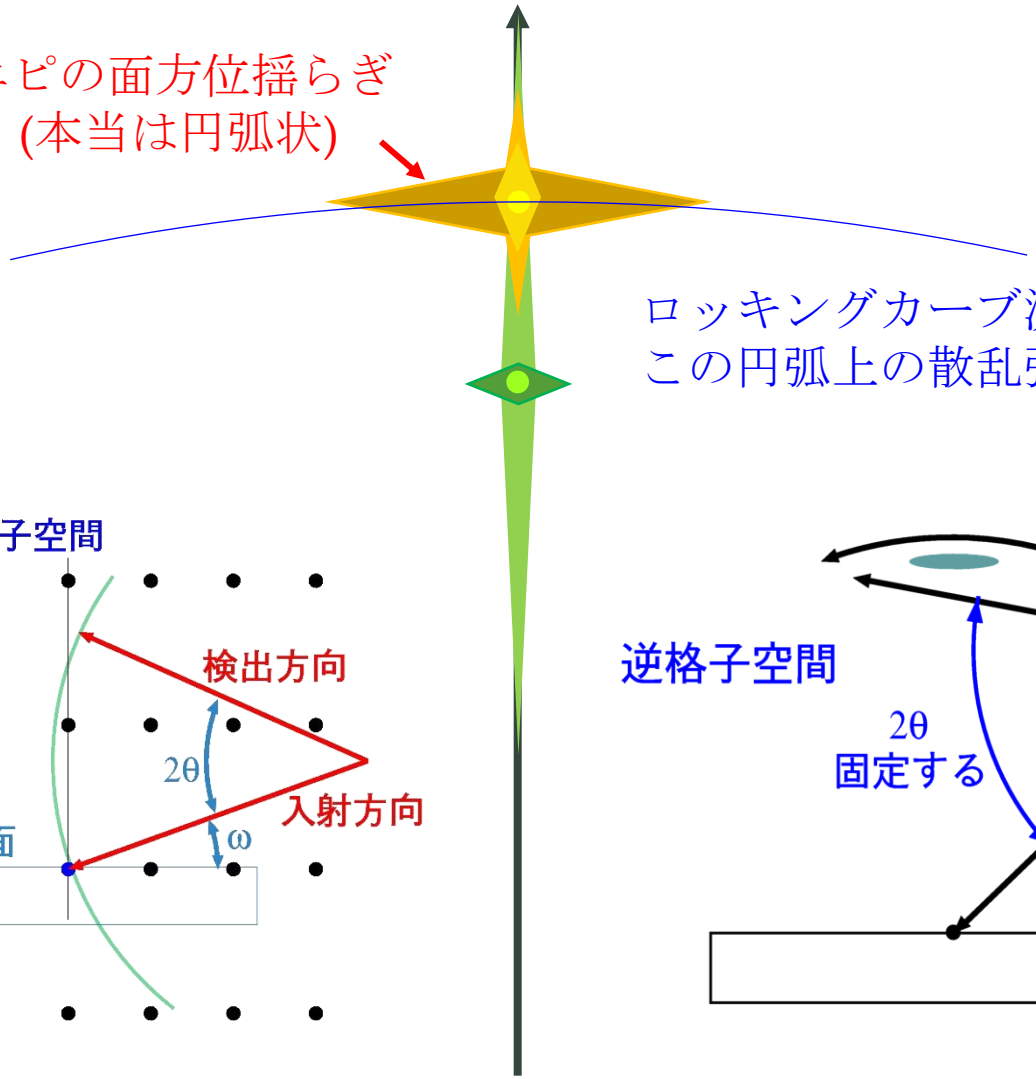
4. ロッキングカーブの計算



4. ロッキングカーブ？

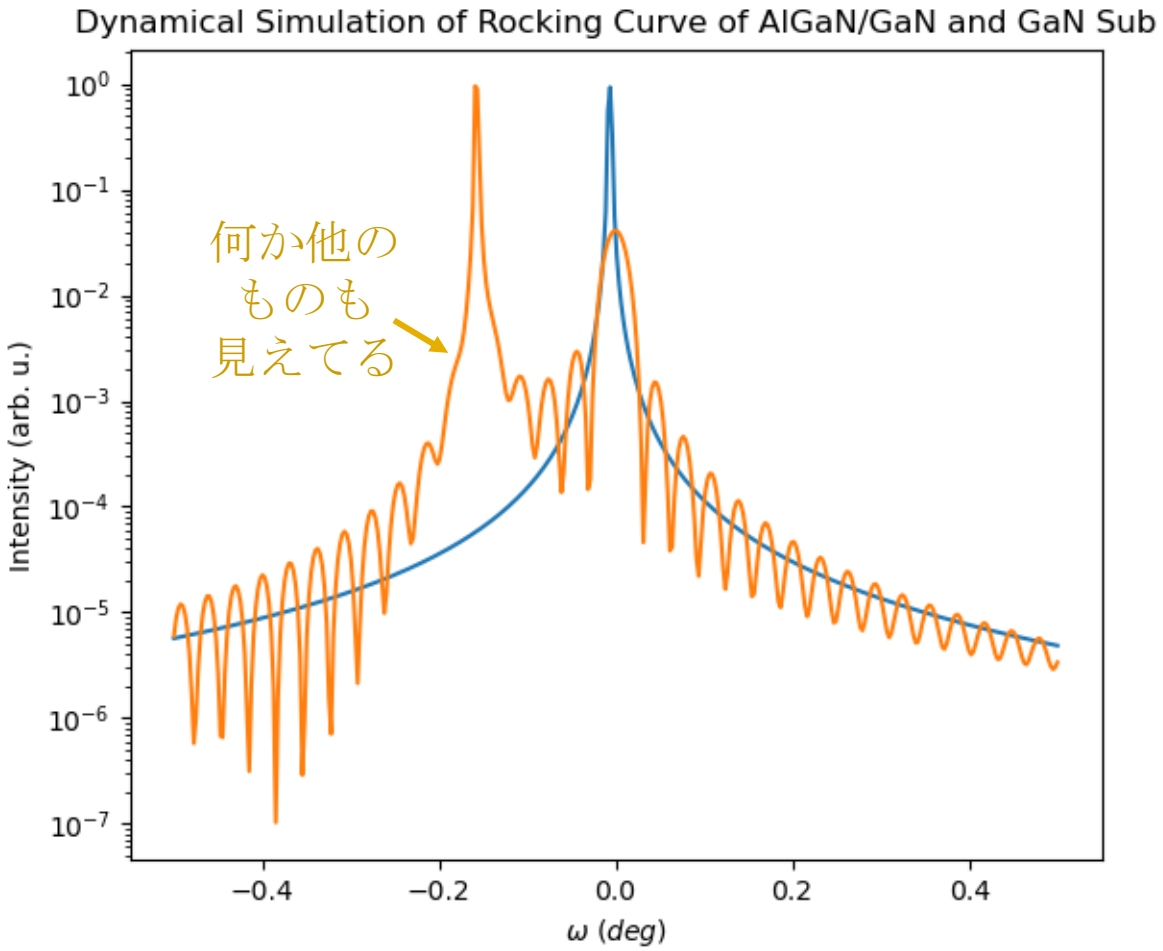

エピのCTR散乱
(膜面に垂直)
エピの面方位揺らぎ
(本当は円弧状)
エピのブラッグ点
エピの組成揺らぎ
(面間隔の揺らぎ)
基板の面方位の揺らぎに起因した広がり
(本当は円弧状)
基板のブラッグ点
基板の表面の完全性に起因した広がり
(CTR散乱、表面に垂直)

4. ロッキングカーブ？



ロッキングカーブ測定に期待するのは
この円弧上の散乱強度分布

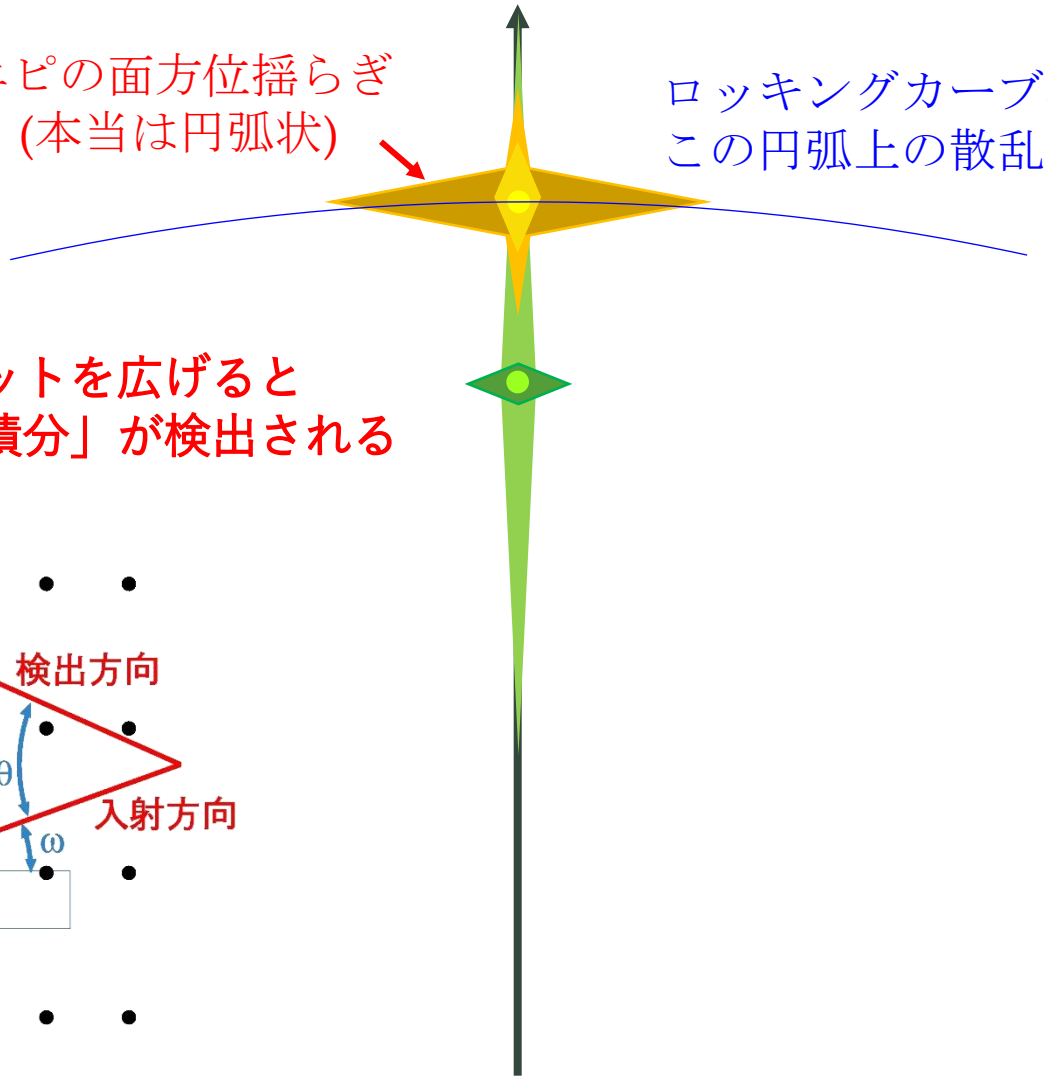
4. ロッキングカーブの計算



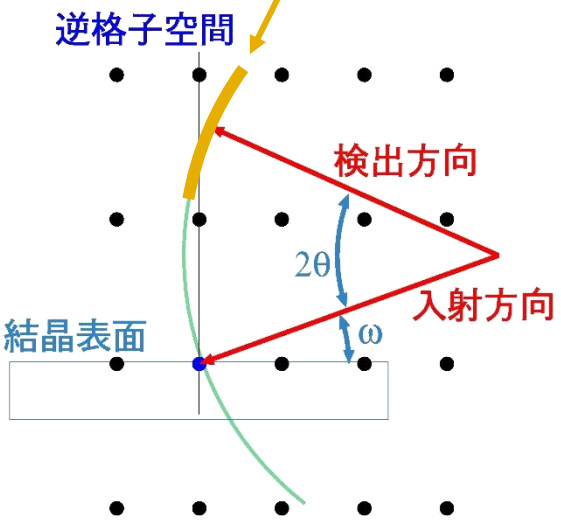
4. ロッキングカーブ？

エピの面方位揺らぎ
(本当は円弧状)

ロッキングカーブ測定に期待するのは
この円弧上の散乱強度分布



受光側のスリットを広げると
この範囲の「積分」が検出される



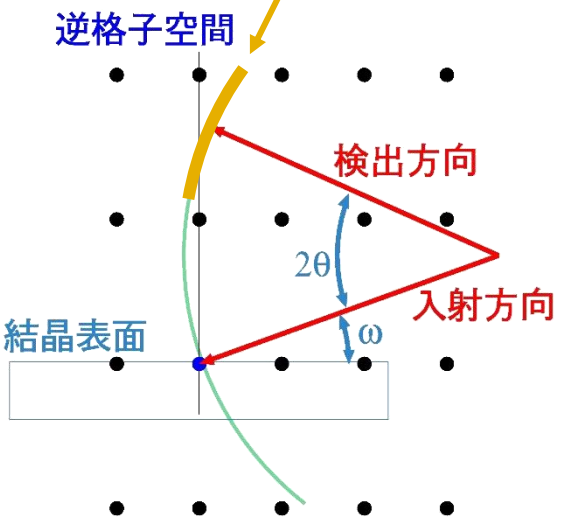
4. ロッキングカーブ？

エピの面方位揺らぎ
(本当は円弧状)

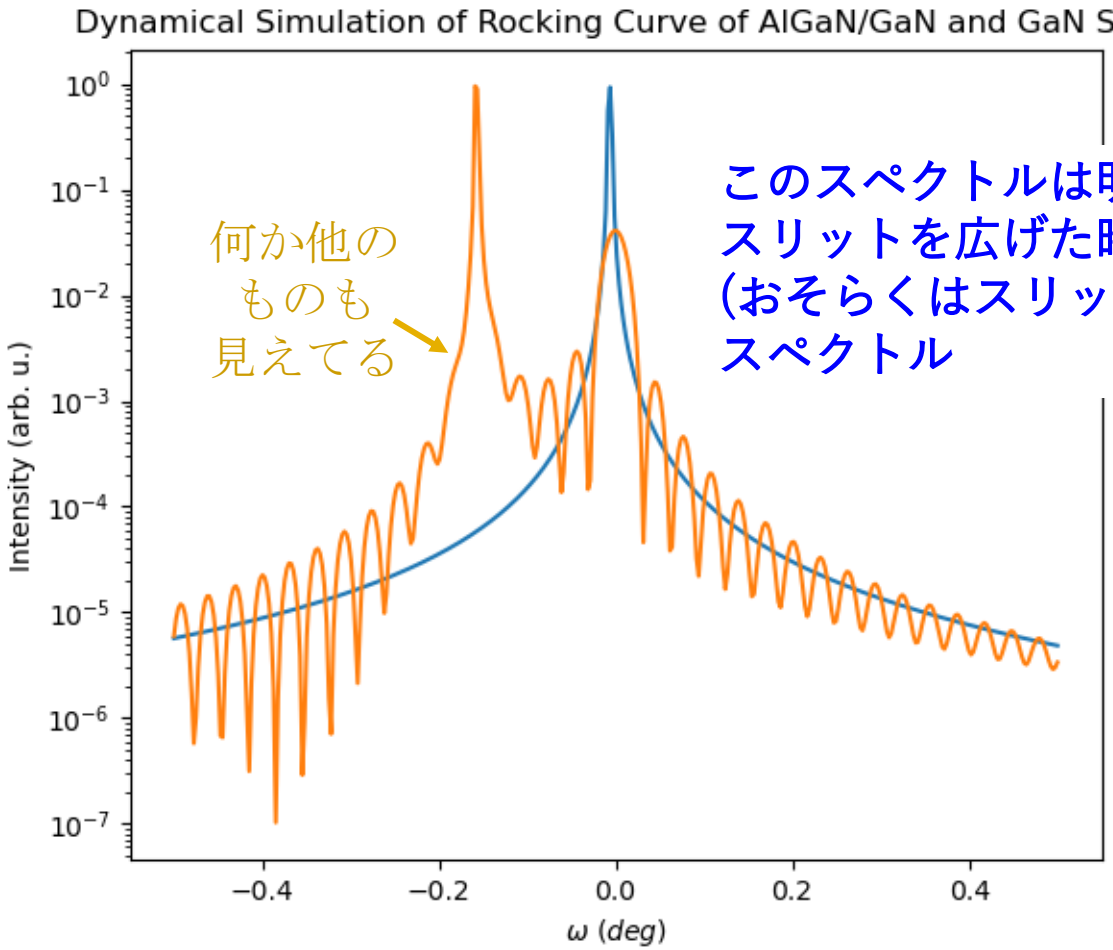
この散乱強度分布を見るつもり
なのに

この散乱強度分布も重なってしまっている

受光側のスリットを広げると
この範囲の「積分」が検出される



4. ロッキングカーブの計算



このスペクトルは明らかに
スリットを広げた時の
(おそらくはスリット制限がない時の)
スペクトル

知見：「xu はこの様な計算をする」このスペクトルは明らかに
学び：「自分の測定は見たいものを見る測定になっているか?」