

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  
田淵雅夫



xrayutilities :

python のライブラリ

- X線回折測定 of データ解析 <https://xrayutilities.sourceforge.io/>
- X線回折に関す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.  $\omega$ - $2\theta$  スペクトルの計算

ロックキングカーブの計算

## 環境構築: xrayutilities のインストール

0. 以降「xrayutilities」を「**xu**」と表記します。

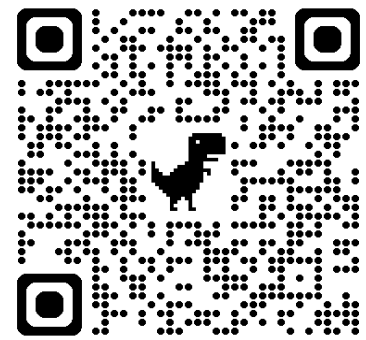
1. **python** をどうにかしてインストールする  
例えば : <https://www.python.org/>

2. テキストエディタを準備する

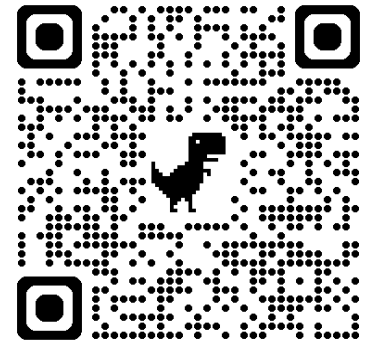
Windows 標準の **note** でも構わない  
**vscode** 等もおすすめ  
(アプリストアから入れない方が良さげ)

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

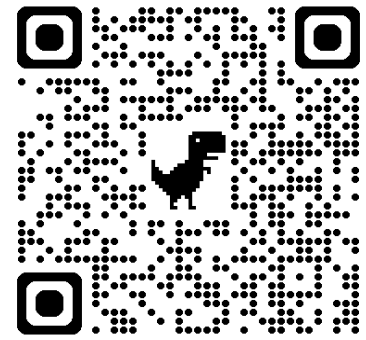
4. 選んだコンソールを起動



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. 起動用スクリプトの場所を確認

- その場で > start. (エクスプローラ起動)
- 今作った仮想環境のフォルダ(xuEnv)を覗く  
bin もしくは Scripts というフォルダを探す
- Activate, activate... があれば正解

```
xu -- xuEnv -- bin -- activate.bat
|           |           |- Activate.ps1
|           |...      |- activate
|           |           |- activate.csh
|- xuWork -- example-01.py
|           |- example-02.py
|...
```

### 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

**sh (bash, zsh) :** > 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

プロンプトに仮想環境の名前が入る (xuEnv)...>

## 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、

入っているとさ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. $\omega$ - $2\theta$ スペクトルの計算

ロックキングカーブの計算

## 2. 構造因子

X線散乱：試料内各点で散乱したX線の総和

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

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

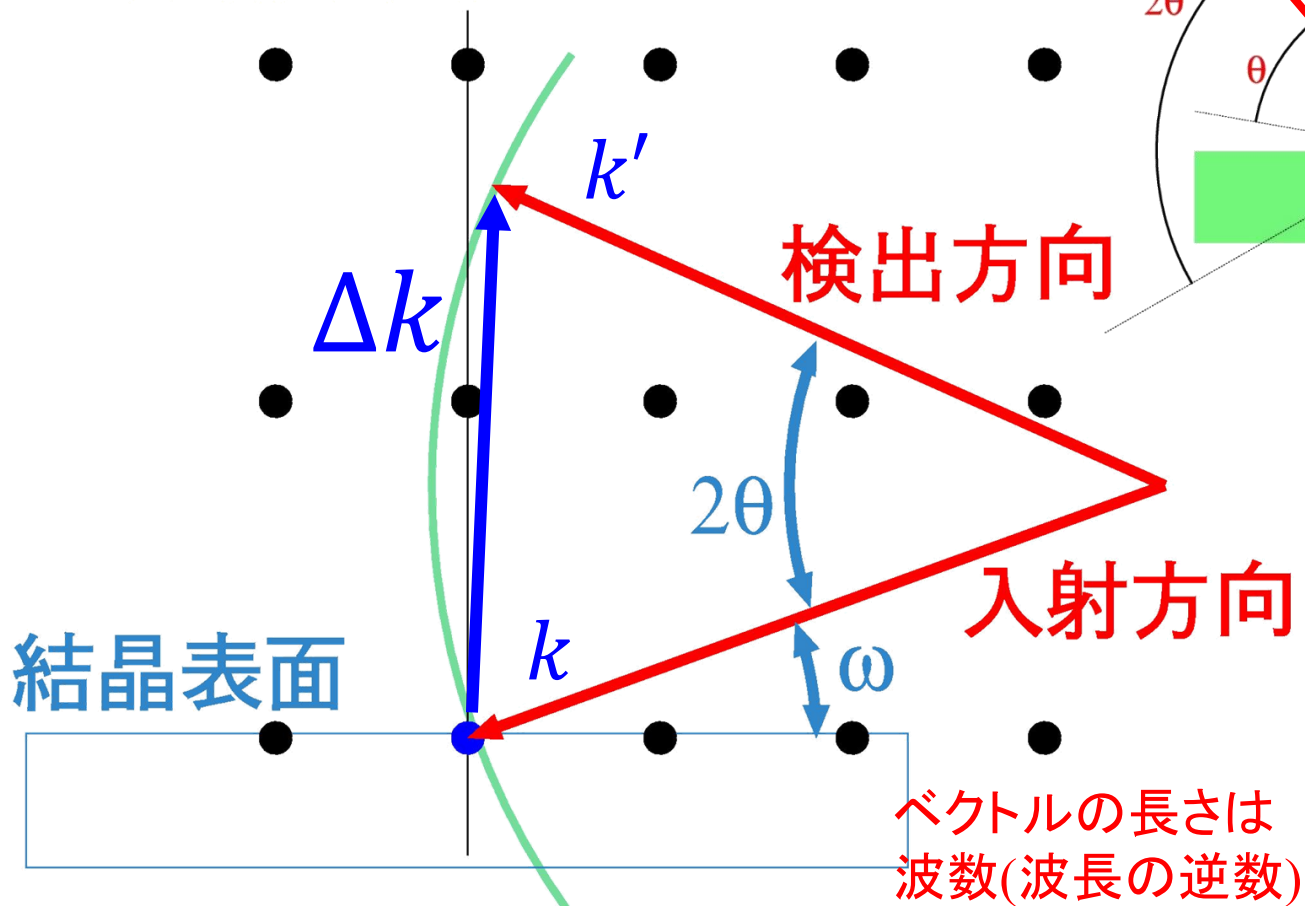
$\Delta k$ : 散乱ベクトル  $\Delta k = k' - k$

$n(r)$  に周期性があると  
回折が起こる

$\exp -i\Delta k \cdot r$  の役割は？

# エバルトの作図

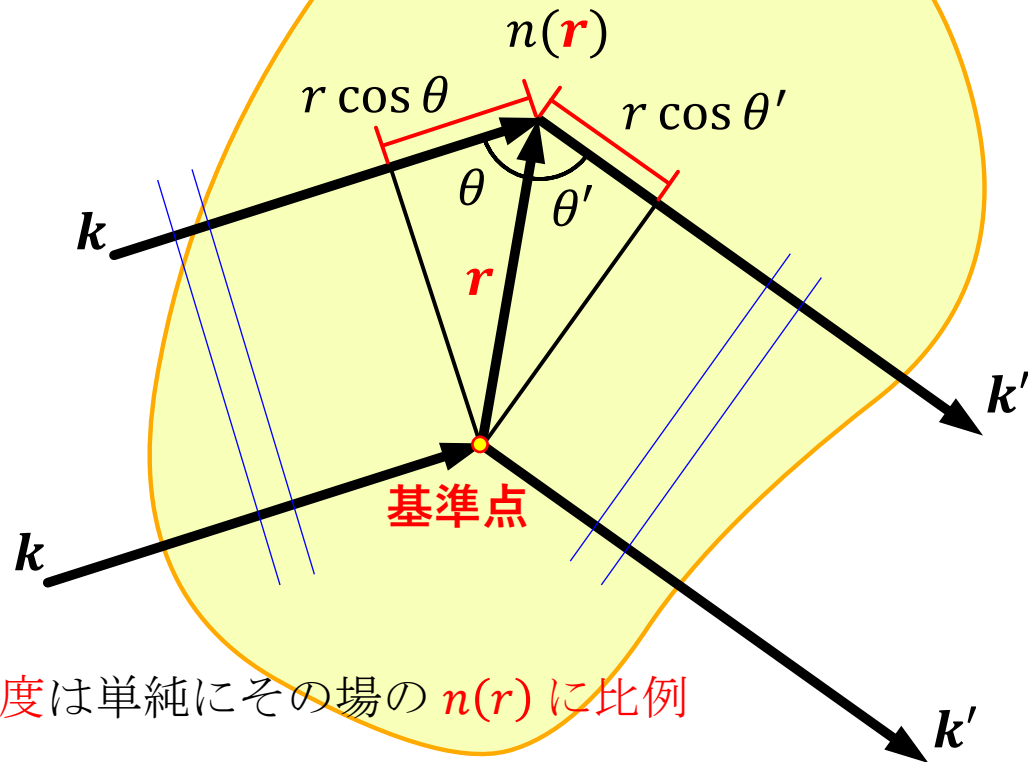
## 逆格子空間



ベクトルの長さは  
波数(波長の逆数)

$\Delta k = k' - k$  : 逆格子空間中の任意の点を  
指さすベクトル

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



各点の散乱強度は単純にその場の  $n(\mathbf{r})$  に比例

$$\text{光路差} : \Delta L = r \cos \theta + r \cos \theta'$$

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

場所が違うと、波の山谷がどれだけずれるか

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

## 2. 構造因子

X線散乱：試料内各点で散乱したX線の総和

散乱振幅

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

$\mathbf{r}$ : 位置ベクトル

$n(\mathbf{r})$ : 電子密度

$\Delta\mathbf{k}$ : 散乱ベクトル  $\Delta\mathbf{k} = \mathbf{k}' - \mathbf{k}$

$n(\mathbf{r})$  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

散乱振幅

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

ルート1: フーリエ級数展開を代入してみる

$$n(\mathbf{r}) = \sum_{\mathbf{G}} n_{\mathbf{G}} \exp i\mathbf{G} \cdot \mathbf{r}$$

フーリエ係数

$$n_{\mathbf{G}} = \frac{1}{V_{\text{cell}}} \int n(\mathbf{r}) \exp -i\mathbf{G} \cdot \mathbf{r} d\mathbf{r}$$

単位構造

$\mathbf{G}$ : フーリエ変換のindex

逆格子ベクトル =

「逆格子点」を指さすベクトル

散乱振幅

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

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

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

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

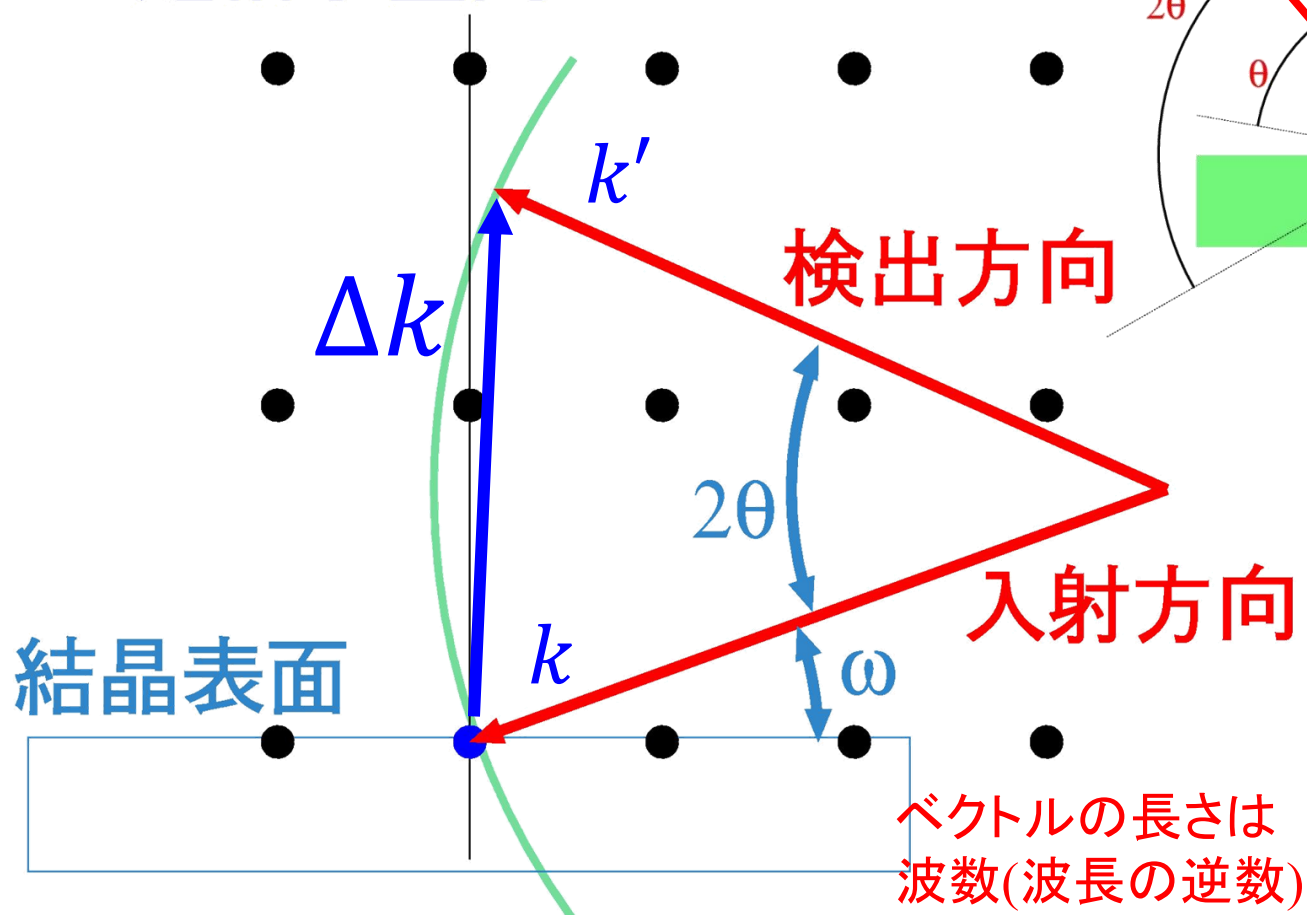
$\neq 0$ : 打ち消しあった小さな値

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

散乱振幅

# エバルトの作図

## 逆格子空間



ベクトルの長さは  
波数(波長の逆数)

$\Delta k = k' - k$  : 逆格子空間中の任意の点を  
指さすベクトル

## 2. 構造因子

散乱振幅  $F = \int n(\mathbf{r}) \exp -i\Delta\mathbf{k} \cdot \mathbf{r} d\mathbf{r}$

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

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

結晶全体 =  $\Sigma_T$  それぞれの場所の単位構造の体積

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

それぞれの場所の単位構造の体積

$$= \Sigma_T \left[ \left\{ \int n(\mathbf{r}') \exp -i\Delta\mathbf{k} \cdot \mathbf{r}' d\mathbf{r}' \right\} \exp -i\Delta\mathbf{k} \cdot \mathbf{T} \right]$$

原点の単位構造の体積

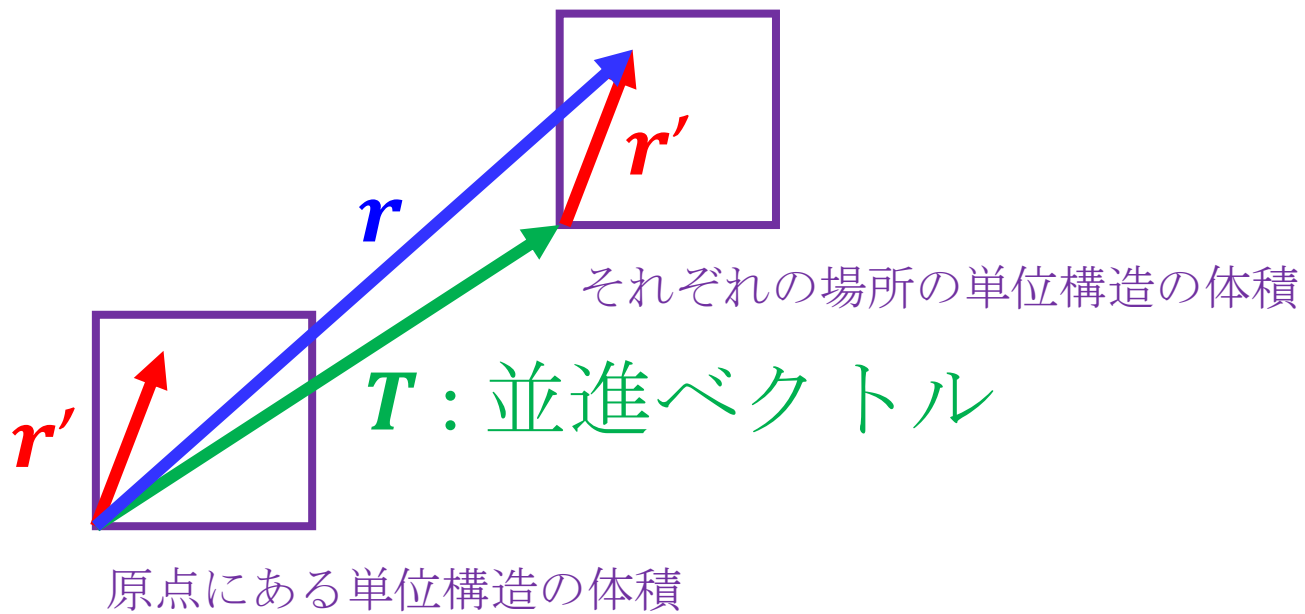
積分範囲を原点に移すための因子

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

原点の単位構造の体積

場所による位相の変化

さっきは  $\mathbf{r}$  だった



## 2. 構造因子

散乱振幅

$$F = \left\{ \int_{\text{原点の単位構造の体積}} n(\mathbf{r}) \exp -i\Delta\mathbf{k} \cdot \mathbf{r} d\mathbf{r} \right\} \underline{\Sigma_T \exp -i\Delta\mathbf{k} \cdot \mathbf{T}}$$

一般の  $\Delta\mathbf{k}$  :

$\sim 1$  の正負様々な値の和  $\Rightarrow$  小さい

$\Delta\mathbf{k} = \mathbf{G}$  の時

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

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

$$\Delta\mathbf{k} \cdot \mathbf{T} = \mathbf{G} \cdot \mathbf{T} = hn_1\mathbf{a}\mathbf{a}^* + kn_2\mathbf{b}\mathbf{b}^* + ln_3\mathbf{c}\mathbf{c}^*$$

$$= 2\pi(hn_1 + kn_2 + ln_3) = 2\pi \times \text{整数}$$

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

 = 1

格子点の数

散乱振幅

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

## 2. 構造因子

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

ルート1:

$$F = n_{\mathbf{G}} V_{\text{Body}} \quad n_{\mathbf{G}} = \frac{1}{V_{\text{cell}}} \int_{\text{単位構造}} n(\mathbf{r}) \exp -i\mathbf{G} \cdot \mathbf{r} \, d\mathbf{r}$$

散乱振幅

フーリエ係数



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

ルート2:

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

散乱振幅

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



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
```

xu は、ある程度知ってる `example-01-01.py`

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

この後いつもプログラムの先頭に書く

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 )
```

空間群の番号(International Tables for Crystallography 準拠)

ワイコフ座標

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.cif" )
AlN = xu.materials.Crystal.fromCIF( "AlN.cif" )
```

`example-01-01.py`

# 大阪公立大学 瀬戸雄介 先生のwebページ： <https://yseto.net/>

Seto's Page Search Switch to [English](#) Last update: 2024-11-14

TOP 自己紹介 ソフト 結晶学のいろは 空間群データ 結晶プロトタイプ 元素情報 プログラミング 結晶学近辺の数理と物理 結晶光学の基礎

## TOP

[自己紹介](#)

ここは瀬戸雄介のHP(yseto.net)です。このページでは結晶学・鉱物学・元素の物性・プログラミングなどに関連した情報を掲載しております。リンクはフリーです。予告なく閉鎖する可能性もありますのであらかじめご了承ください。ご意見、ご要望等はメールにてご連絡ください。mail: seto.y[at]omu.ac.jp ([at]を@に変更してください)

---

このホームページのコンテンツ一覧です。

- [TOP](#)
- [自己紹介](#)
  - [研究/教育活動](#)
- [ソフト](#)
  - [ReciPro](#)
  - [PDIndexer](#)
  - [IPAnalyzer](#)
  - [CSManager](#)
  - [共通コンポーネント](#)
- [結晶学のいろは](#)
  - [1章. 結晶と対称性](#)
  - [2章. 群・点群・空間群](#)
  - [3章. 対称要素、点群、空間群に関する実践的知識](#)
  - [4章. 空間群の超群/部分群](#)
  - [5章. 回折現象](#)
- [空間群データ](#)
  - [空間群一覧](#)
  - [ワイクフ位置一覧](#)
  - [結晶プロトタイプ](#)

<https://yseto.net/soft/pdi>

空間群、ワイクフ座標、原子散乱因子(X線、電子線、中性子線)

## 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()      f という名前で描画キャンパスを作る
```

```
InP.show_unitcell( fig=f, subplot=121 )
```

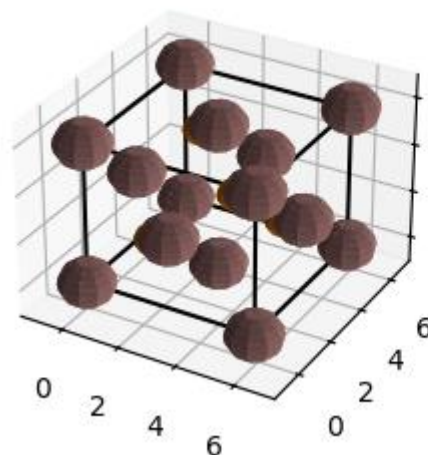
```
plt.title( 'InP zinoblende' )      さっき作った InP に依頼して  
                                   「単位格子」を f に描画してもらう
```

```
InPWZ.show_unitcell( fig=f, subplot=122 )
```

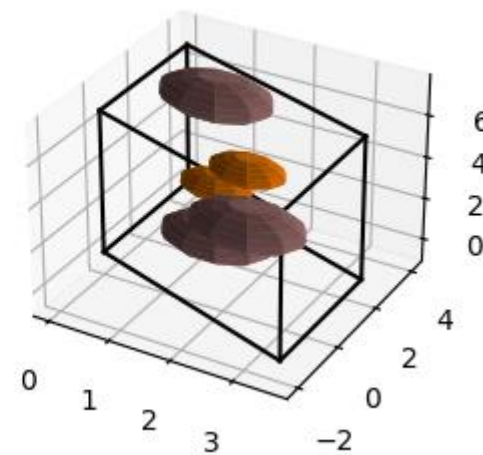
```
plt.title( 'InP wurtzite' )
```

```
plt.show()
```

InP zinoblende



InP wurtzite



## 2. 構造因子 xrayUtilities 登場 !!

- 肝心の構造因子は？

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

```
import xrayutilities as xu
```

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

ちよつとさぼって  $xu$  が知ってる物質を使う

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

```
energy = 8048 # [eV] Cu-K $\alpha$ 
```

計算してほしい逆格子点のセット

```
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 )
```

(指数で)指定された逆格子点の散乱ベクトル

```
    F = InP.StructureFactor( qvec, energy )
```

その散乱ベクトルの時の構造因子

```
    print( f"|F| {hkl} = {abs(F):8.3f}" )
```

なぜエネルギーを指定？

(後で考えます)

```
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}")
```

## 2. 構造因子

Structural Factors of InP at 8048[eV]

$$|F|[0, 0, 0] = 258.476$$

$$|F|[1, 1, 1] = 182.826$$

$$|F|[2, 2, 2] = 103.439$$

$$|F|[3, 3, 3] = 112.501$$

$$|F|[0, 0, 0] = 258.476$$

$$|F|[1, 0, 0] = 0.000$$

$$|F|[2, 0, 0] = 121.886$$

$$|F|[4, 0, 0] = 163.484$$

Structural Factors of GaAs at 8048[eV]

$$|F|[0, 0, 0] = 247.148$$

$$|F|[1, 1, 1] = 148.498$$

$$|F|[2, 2, 2] = 6.527$$

$$|F|[3, 3, 3] = 92.194$$

$$|F|[0, 0, 0] = 247.148$$

$$|F|[1, 0, 0] = 0.000$$

$$|F|[2, 0, 0] = 7.106$$

$$|F|[4, 0, 0] = 154.631$$

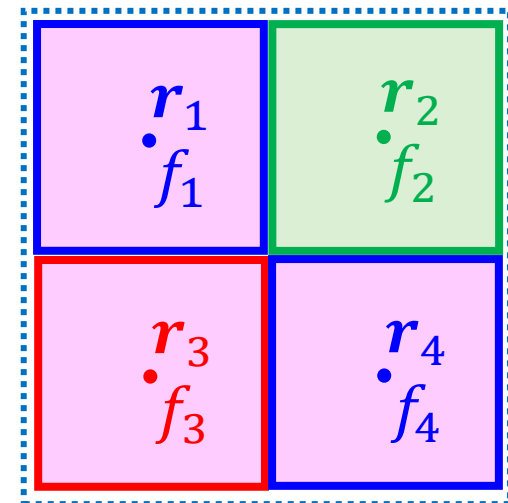
## 2. 構造因子 $n_G$ の値は？ 人がやるなら？

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

「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  $f_j = \int n(\mathbf{r}) \exp -i\mathbf{G} \cdot \mathbf{r} \, d\mathbf{r}$  を計算する
- 位相差を考慮して全部足す

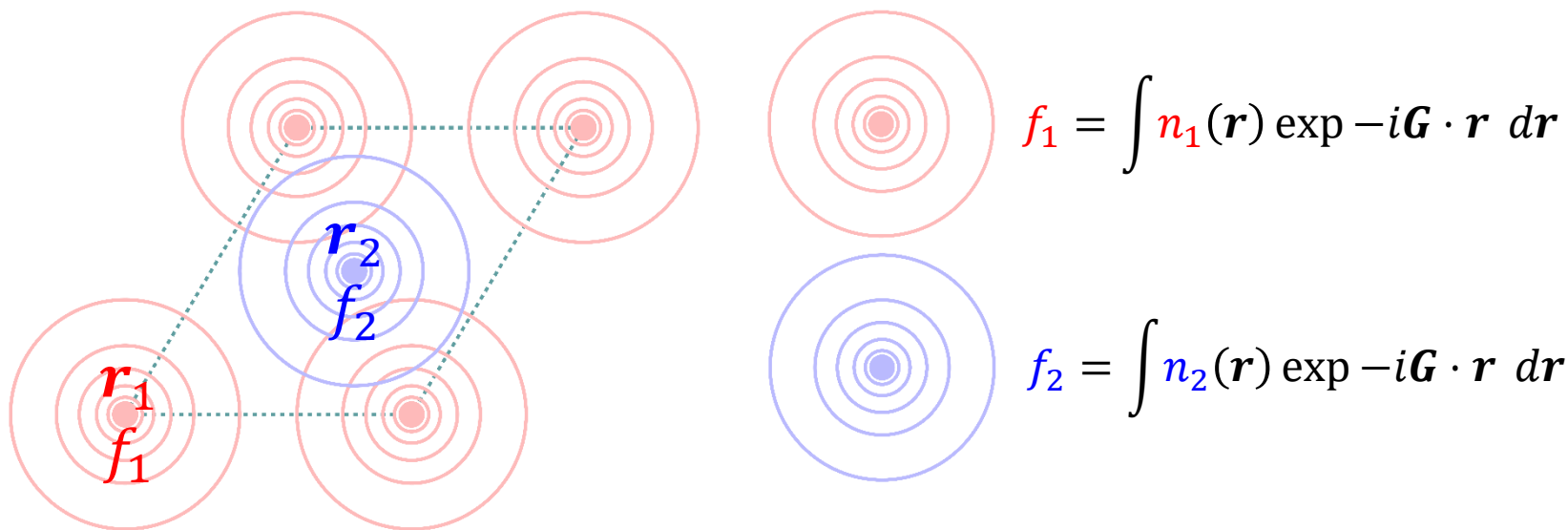
$$n_G = \sum_j f_j \exp -i\mathbf{G} \cdot \mathbf{r}_j$$



## 2. 構造因子 $n_G$ の値は？ 人がやるなら？

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

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



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

## 2. 構造因子 $n_G$ の値は？ 人がやるなら？

$$n_G = \sum_j f_j \exp -i\mathbf{G} \cdot \mathbf{r}_j \quad f_i \text{ はどうする？}$$

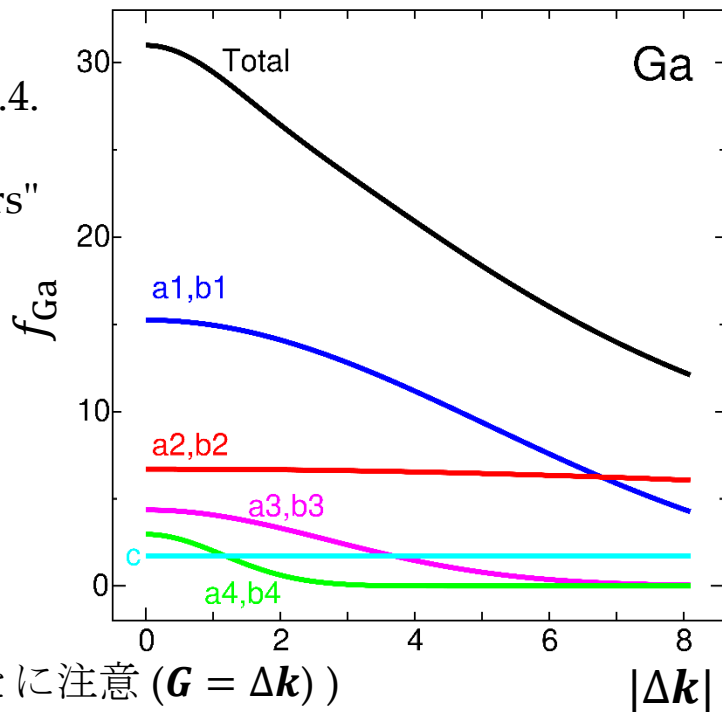
Ans. 1:  $f_{\text{原子}} = \int n_{\text{原子}}(\mathbf{r}) \exp -i\mathbf{G} \cdot \mathbf{r} d\mathbf{r}$  は、もし  $\exp -i\mathbf{G} \cdot \mathbf{r}$  が 1 なら  
原子番号  $Z$  (原子が持つ電子の数) に等しい In: 49, P: 15, Ga: 31, As: 33

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"

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

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。



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

$|\Delta \mathbf{k}|$

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

$$\mathbf{G} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$$

$$\mathbf{r}_j = \alpha_j\mathbf{a} + \beta_j\mathbf{b} + \gamma_j\mathbf{c}$$

$$\mathbf{G} \cdot \mathbf{r}_j = 2\pi(h\alpha_j + k\beta_j + l\gamma_j)$$

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

$$a\mathbf{b}^* = a\mathbf{c}^* = b\mathbf{c}^* = b\mathbf{a}^* = c\mathbf{a}^* = c\mathbf{b}^* = 0$$

GaAs 単位格子中の原子位置( $\alpha, \beta, \gamma$ )

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)

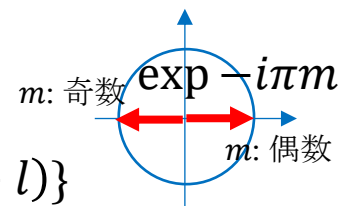
単位構造 Ga(0,0,0), As(1/4,1/4,1/4)が

(0,0,0), (1/2, 1/2, 0), (1/2, 0, 1/2), (0, 1/2, 1/2)

にある、と見る。

$$f_{\text{Unit}} = f_{\text{Ga}} + f_{\text{As}} \exp -i\frac{\pi}{2}(h+k+l)$$

$$n_G = f_{\text{Unit}} \{1 + \exp -i\pi(h+k) + \exp -i\pi(h+l) + \exp -i\pi(k+l)\}$$



$h, k, l$  : 偶奇混合

$$\Rightarrow n_G = 0$$

: 面心立方由来の禁制。中々崩れない

$$\text{全て奇数 or 全て偶数} \Rightarrow n_G = 4f_{\text{Unit}}$$

$$h, k, l : \text{全て奇数} \quad f_{\text{Unit}} = f_{\text{Ga}} \pm if_{\text{As}}$$

中ぐらい

$$\text{全て偶数} \quad f_{\text{Unit}} = f_{\text{Ga}} + f_{\text{As}} \quad (h+k+l = 4m)$$

強い

$$\text{又は} \quad f_{\text{Unit}} = f_{\text{Ga}} - f_{\text{As}} \quad (h+k+l = 4m \pm 2)$$

弱い

## 2. 構造因子

Structural Factors of InP at 8048[eV]

$$|F|[0, 0, 0] = 258.476$$

$$|F|[1, 1, 1] = 182.826$$

$$|F|[2, 2, 2] = 103.439$$

$$|F|[3, 3, 3] = 112.501$$

$$|F|[0, 0, 0] = 258.476$$

$$|F|[1, 0, 0] = 0.000$$

$$|F|[2, 0, 0] = 121.886$$

$$|F|[4, 0, 0] = 163.484$$

Structural Factors of GaAs at 8048[eV]

$$|F|[0, 0, 0] = 247.148$$

$$|F|[1, 1, 1] = 148.498$$

$$|F|[2, 2, 2] = 6.527$$

$$|F|[3, 3, 3] = 92.194$$

$$|F|[0, 0, 0] = 247.148 \quad 255.836$$

$$|F|[1, 0, 0] = 0.000 \quad 0.000$$

$$|F|[2, 0, 0] = 7.106 \quad 5.588$$

$$|F|[4, 0, 0] = 154.631 \quad 163.359$$

## 2. 構造因子 xrayUtilities 登場 !!

- 肝心の構造因子は？

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

```
import xrayutilities as xu
```

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

ちよつとさぼって  $xu$  が知ってる物質を使う

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

```
energy= 8048 # eV
```

計算してほしい逆格子点のセット

```
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 )
```

(指数で)指定された逆格子点の散乱ベクトル

```
    F = InP.StructureFactor( qvec, energy )
```

その散乱ベクトルの時の構造因子

```
    print( f"|F| {hkl} = {abs(F):8.3f}" )
```

**なぜエネルギーを指定？**

```
print( f"Structual Factors of GaAs at {energy}[eV]" )
```

```
for hkl in hkllist:
```

```
    qvec = GaAs.Q( hkl )
```

```
    F = GaAs.StructureFactor(qvec, energy)
```

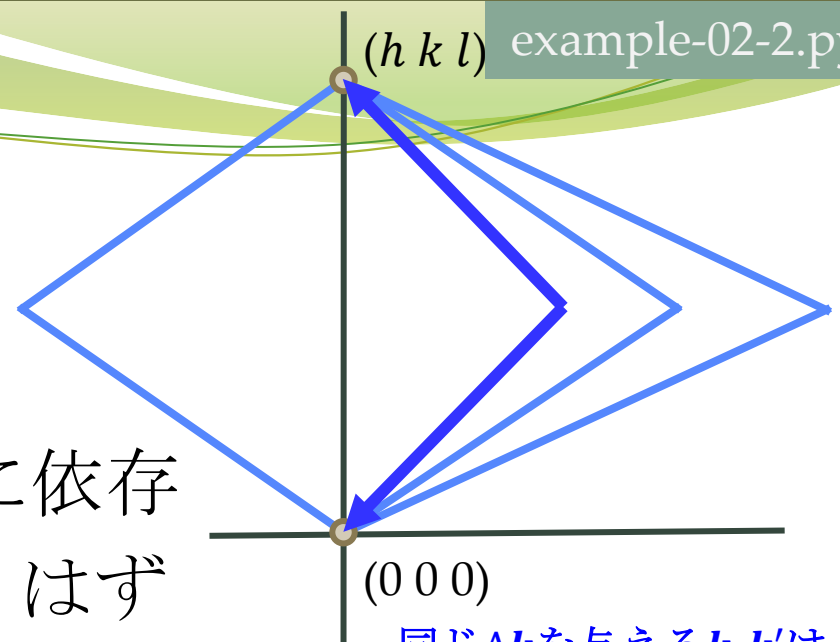
```
    print(f"|F| {hkl} = {abs(F):8.3f}")
```

## 2. 構造因子

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

構造因子は  $\mathbf{G}$  ( $= \Delta\mathbf{k} \equiv \mathbf{Q}$ ) に依存  
 $\mathbf{k}, \mathbf{k}'$  には依存していない... はず

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



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

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

$$f = f_0 + f' + if''$$

構造因子(慣用的な記号)

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

構造因子  $f = f_0 + f' + if''$

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)

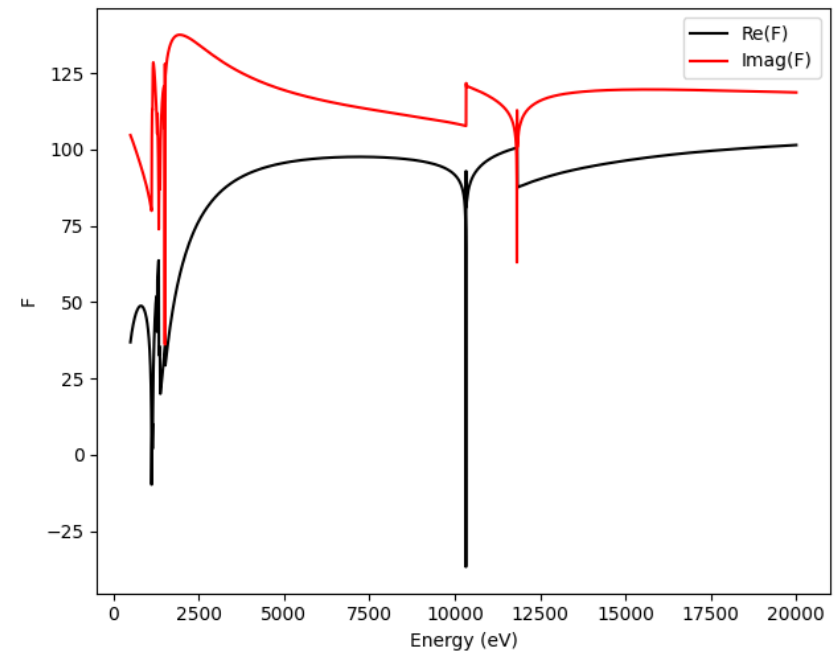
```
import xrayutilities as xu
import numpy
import matplotlib.pyplot as plt
```

```
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()
```



## 2. 構造因子

Structural Factors of InP at 8048[eV]

$$|F|[0, 0, 0] = 258.476$$

$$|F|[1, 1, 1] = 182.826$$

$$|F|[2, 2, 2] = 103.439$$

$$|F|[3, 3, 3] = 112.501$$

$$|F|[0, 0, 0] = 258.476$$

$$|F|[1, 0, 0] = 0.000$$

$$|F|[2, 0, 0] = 121.886$$

$$|F|[4, 0, 0] = 163.484$$

Structural Factors of GaAs at 8048[eV]

$$|F|[0, 0, 0] = 247.148$$

$$|F|[1, 1, 1] = 148.498$$

$$|F|[2, 2, 2] = 6.527$$

$$|F|[3, 3, 3] = 92.194$$

$$|F|[0, 0, 0] = 247.148 \quad 255.836$$

$$|F|[1, 0, 0] = 0.000 \quad 0.000$$

$$|F|[2, 0, 0] = 7.106 \quad 5.588$$

$$|F|[4, 0, 0] = 154.631 \quad 163.359$$

# 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.  $\omega$ - $2\theta$  スペクトルの計算

ロックキングカーブの計算

### 3. 逆格子マップの計算

- ・プロットしてみる

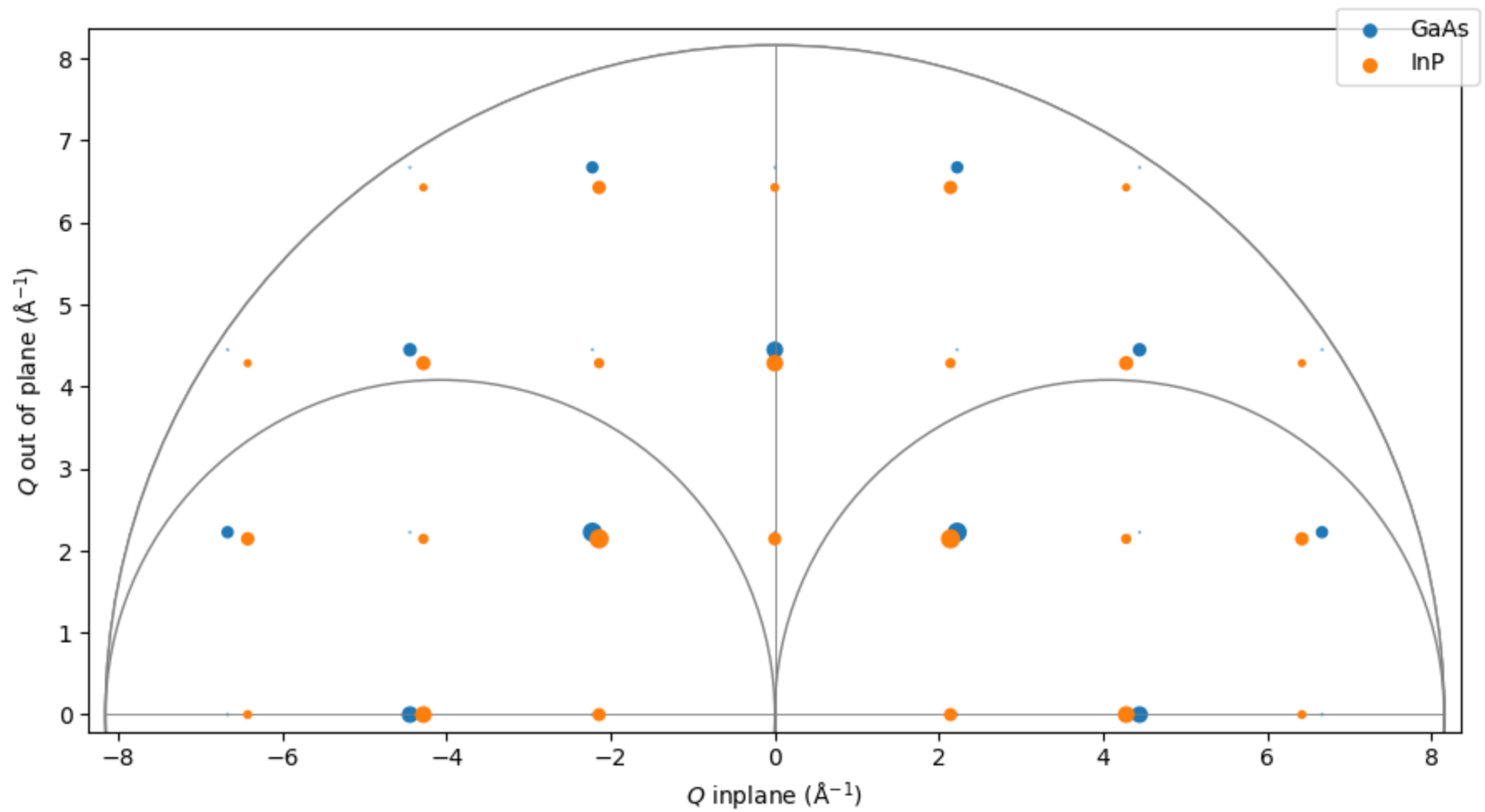
```
import xrayutilities as xu
import matplotlib.pyplot as plt
```

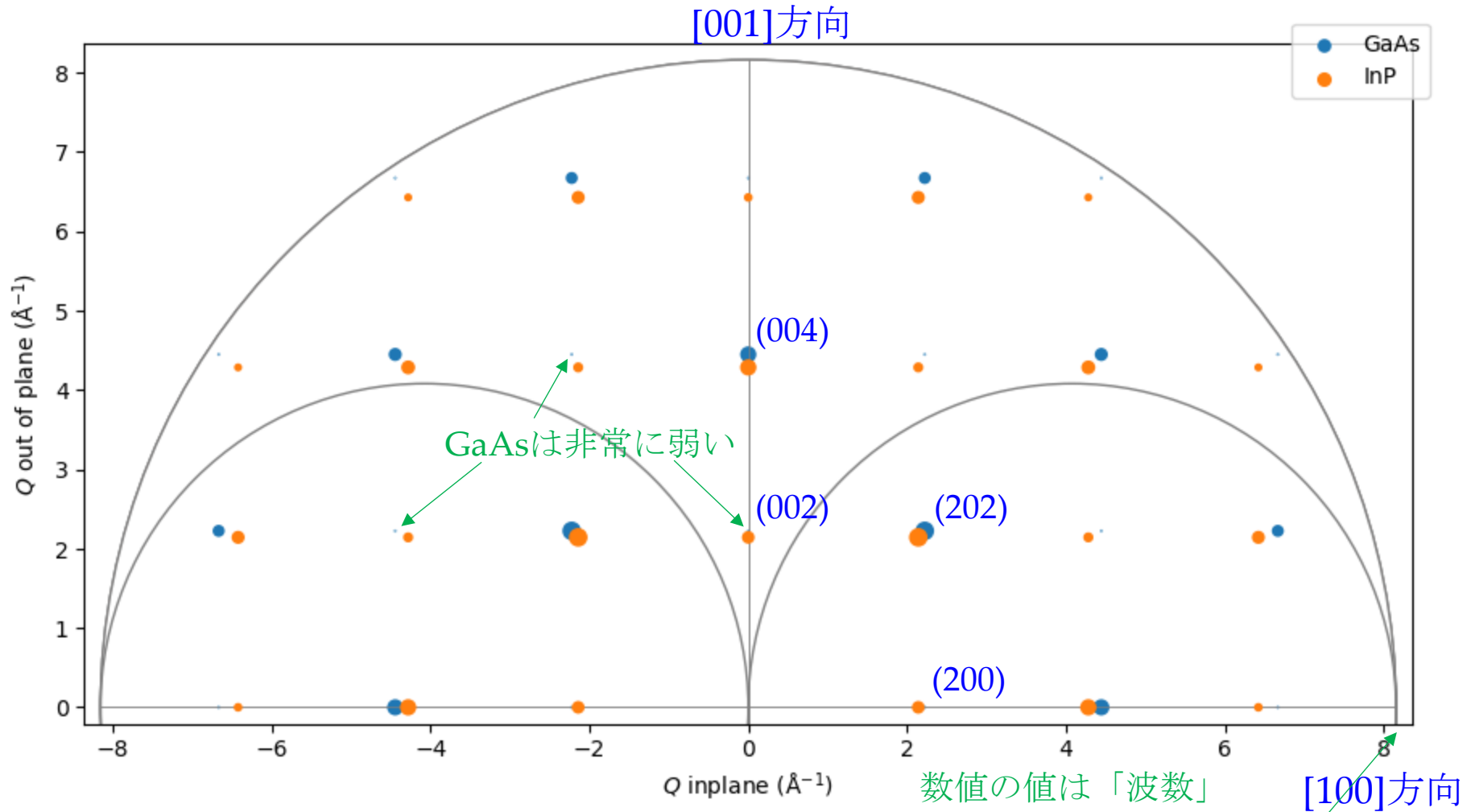
```
GaAs = xu.materials.GaAs      GaAs の x=(1,0,0), y=(0,0,1) 断面のプロット
hGaAs = xu.HXRD( GaAs.Q( 1, 0, 0 ), GaAs.Q( 0, 0, 1 ) )
ax, h = xu.materials.show_reciprocal_space_plane( GaAs, hGaAs )
```

```
InP = xu.materials.InP      InP のプロットを重ねる
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 )
```

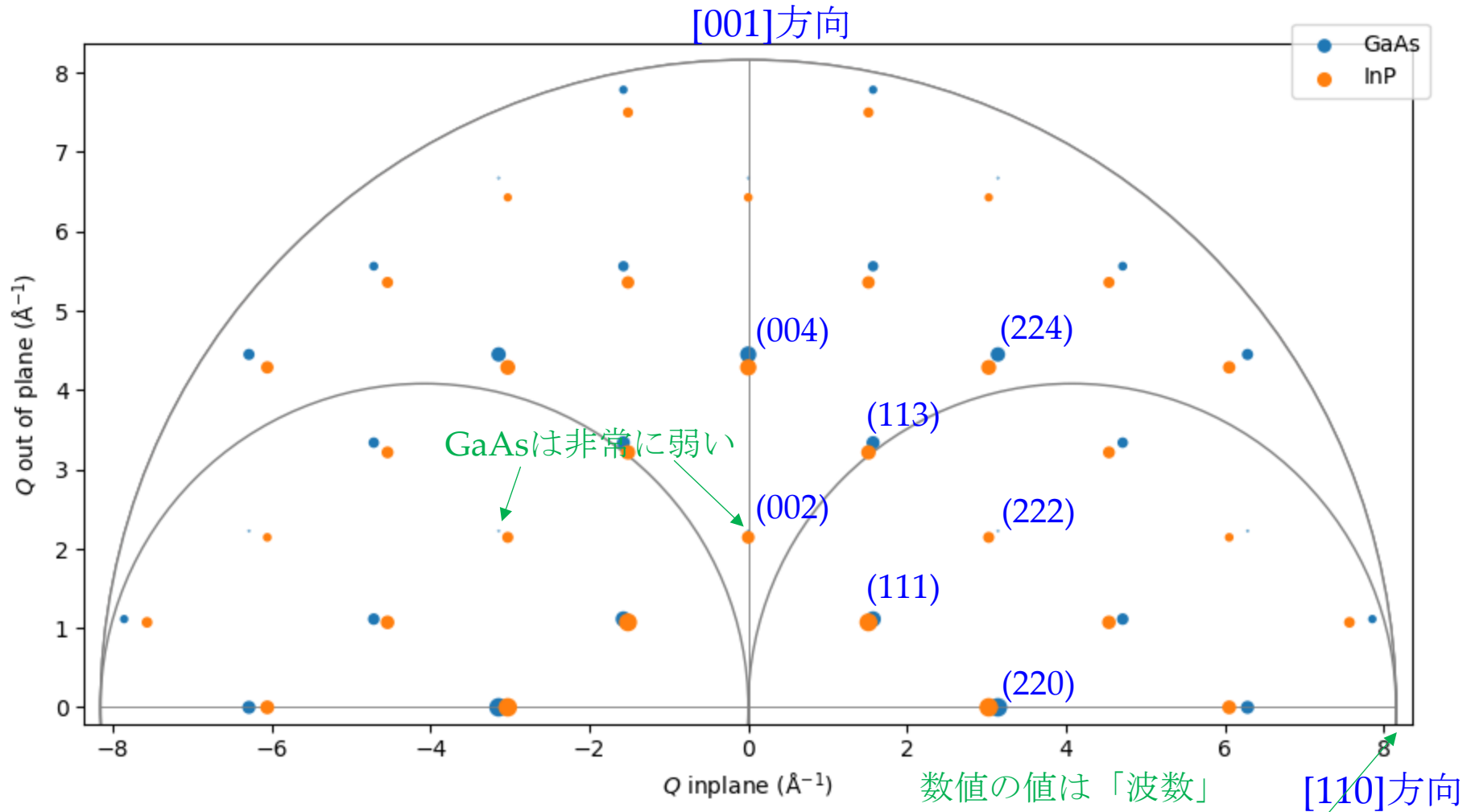
```
plt.show()   これがないと「図」が表示されない
```

# 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 になっているはず。

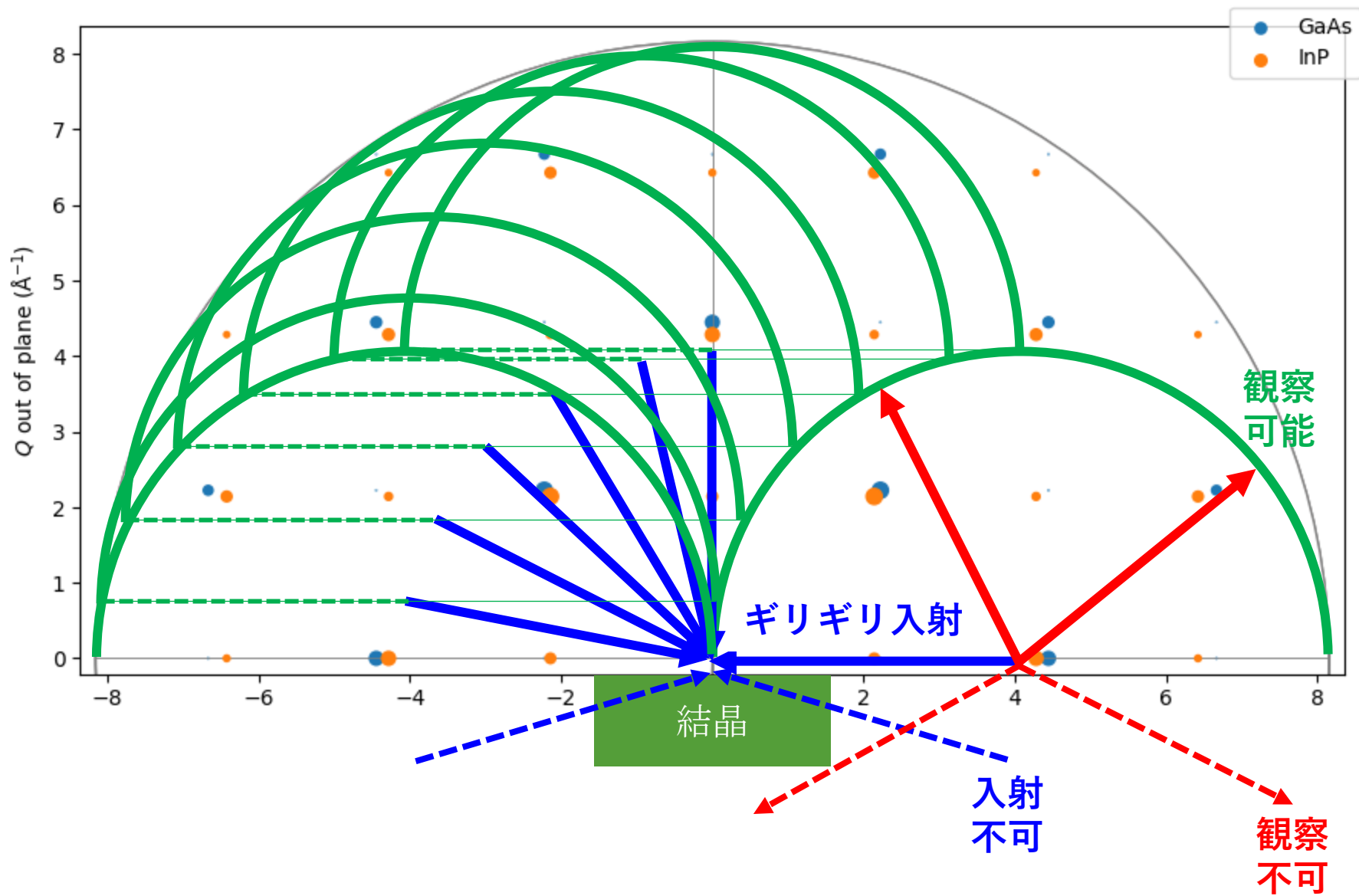


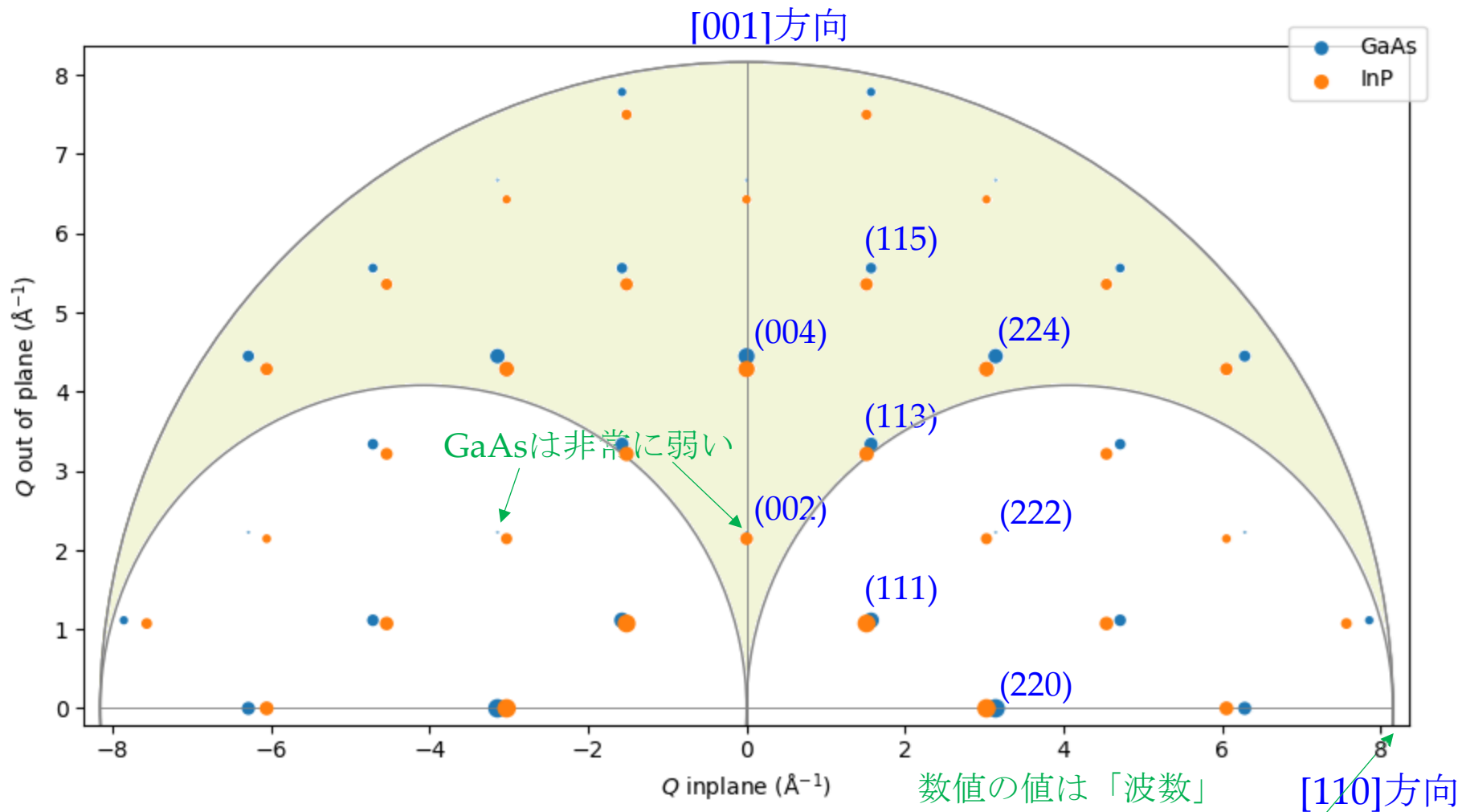


$E_{X\text{-ray}} = 8048 \text{ [eV]}$  なので  $|\mathbf{k}| = |\mathbf{k}'| = 4.076 \text{ [\AA}^{-1}]$   
 $|\Delta\mathbf{k}| = |\mathbf{k}' - \mathbf{k}| = Q$  の最大値は  $\approx 8.152 \text{ [\AA}^{-1}]$



$E_{X\text{-ray}} = 8048 \text{ [eV]}$  なので  $|\mathbf{k}| = |\mathbf{k}'| = 4.076 \text{ [Å}^{-1}\text{]}$   
 $|\Delta\mathbf{k}| = |\mathbf{k}' - \mathbf{k}| = Q$  の最大値は  $\approx 8.152 \text{ [Å}^{-1}\text{]}$





$E_{X\text{-ray}} = 8048 \text{ [eV]}$  なので  $|\mathbf{k}| = |\mathbf{k}'| = 4.076 \text{ [\AA}^{-1}]$   
 $|\Delta\mathbf{k}| = |\mathbf{k}' - \mathbf{k}| = Q$  の最大値は  $\approx 8.152 \text{ [\AA}^{-1}]$

# 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. $\omega-2\theta$ スペクトルの計算

ロックイングカーブの計算

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("GaN.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

```
# GaN, AlN の結晶構造をCIFから読む
```

```
GaN = xu.materials.Crystal.fromCIF("GaN.cif")
```

```
AlN = xu.materials.Crystal.fromCIF("AlN.cif")
```

```
# 今作った結晶(GaN, AlN)は、結晶構造の情報しか持っていないので
```

```
# 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
```

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

- 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 )
#  $\omega$ - $2\theta$  強度計算
Epi_sub: lAlN/sub の積層構造
## 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...Å)
energy      = 12390 / wavelength     # 波長からEに(計算に使うのはこちら)
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
```

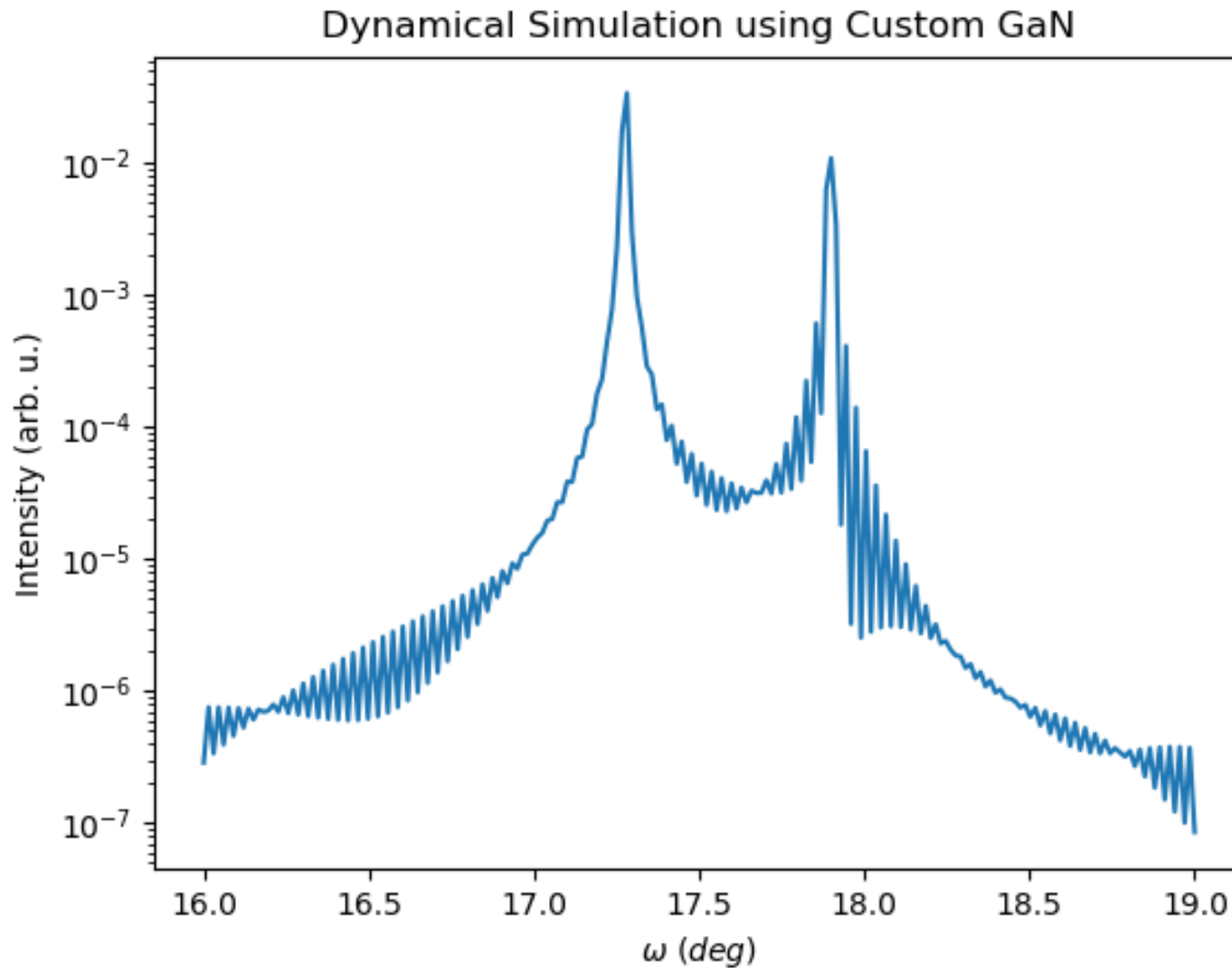
AlN(epi)/GaN(sub)を計算してみる

## 4. $\omega - 2\theta$ スペクトルの計算

```
# 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()
```

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

AlN(epi)/GaN(sub)を計算してみる

4.  $\omega - 2\theta$  スペクトルの計算

$\text{Al}_x\text{Ga}_{1-x}\text{N}(\text{epi})/\text{GaN}(\text{sub})$ 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

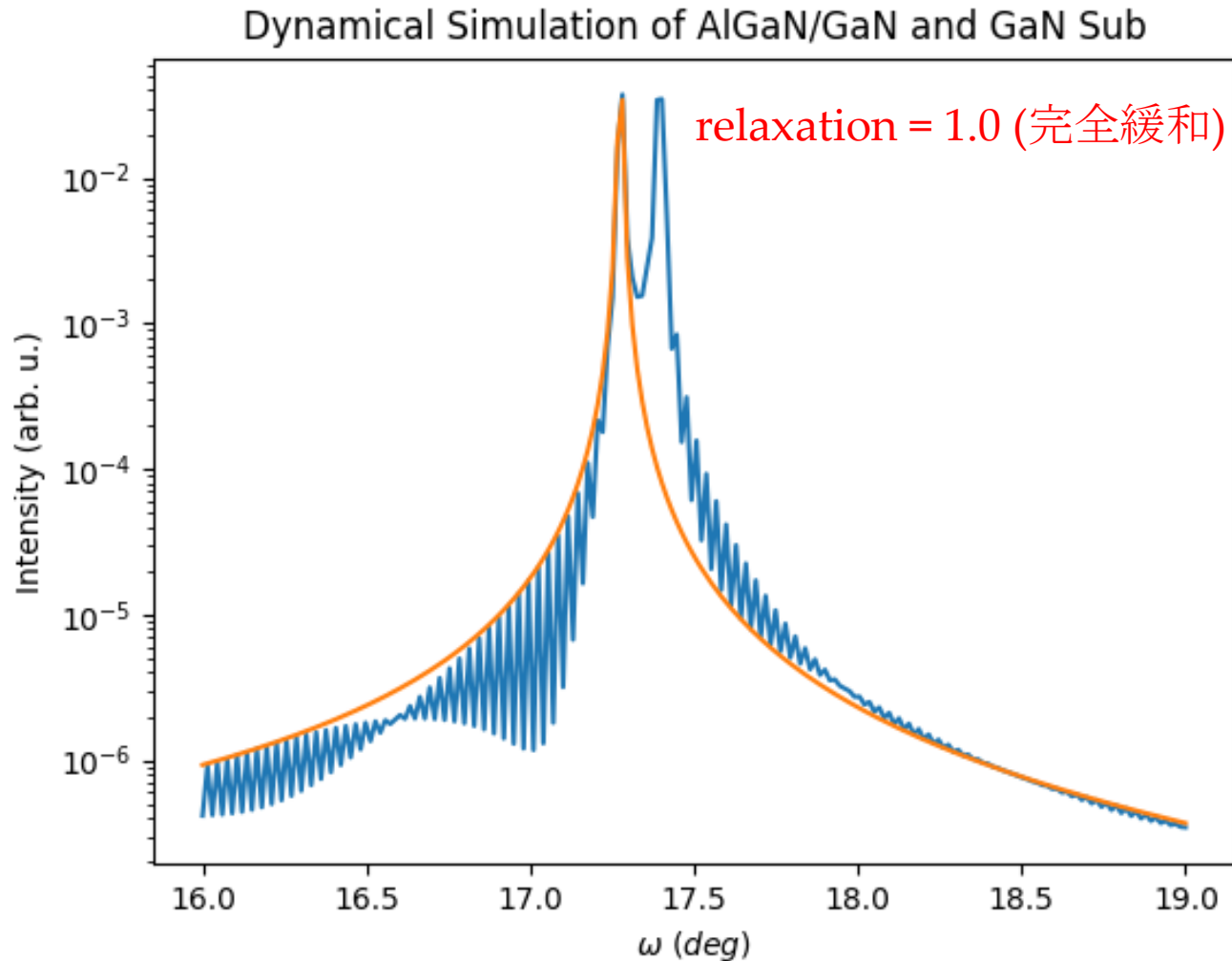
## 準備した Layer を積み上げて AlGaN層 / GaN 基板という積層構造を構築
Epi_Sub = xu.simpack.PseudomorphicStack001( "AlGaN/GaN", sub+lAlGaN )

# 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
# 変数名が少し違うのに注意

```

$\text{Al}_x\text{Ga}_{1-x}\text{N}(\text{epi})/\text{GaN}(\text{sub})$ を計算してみる

## 4. $\omega - 2\theta$ スペクトルの計算



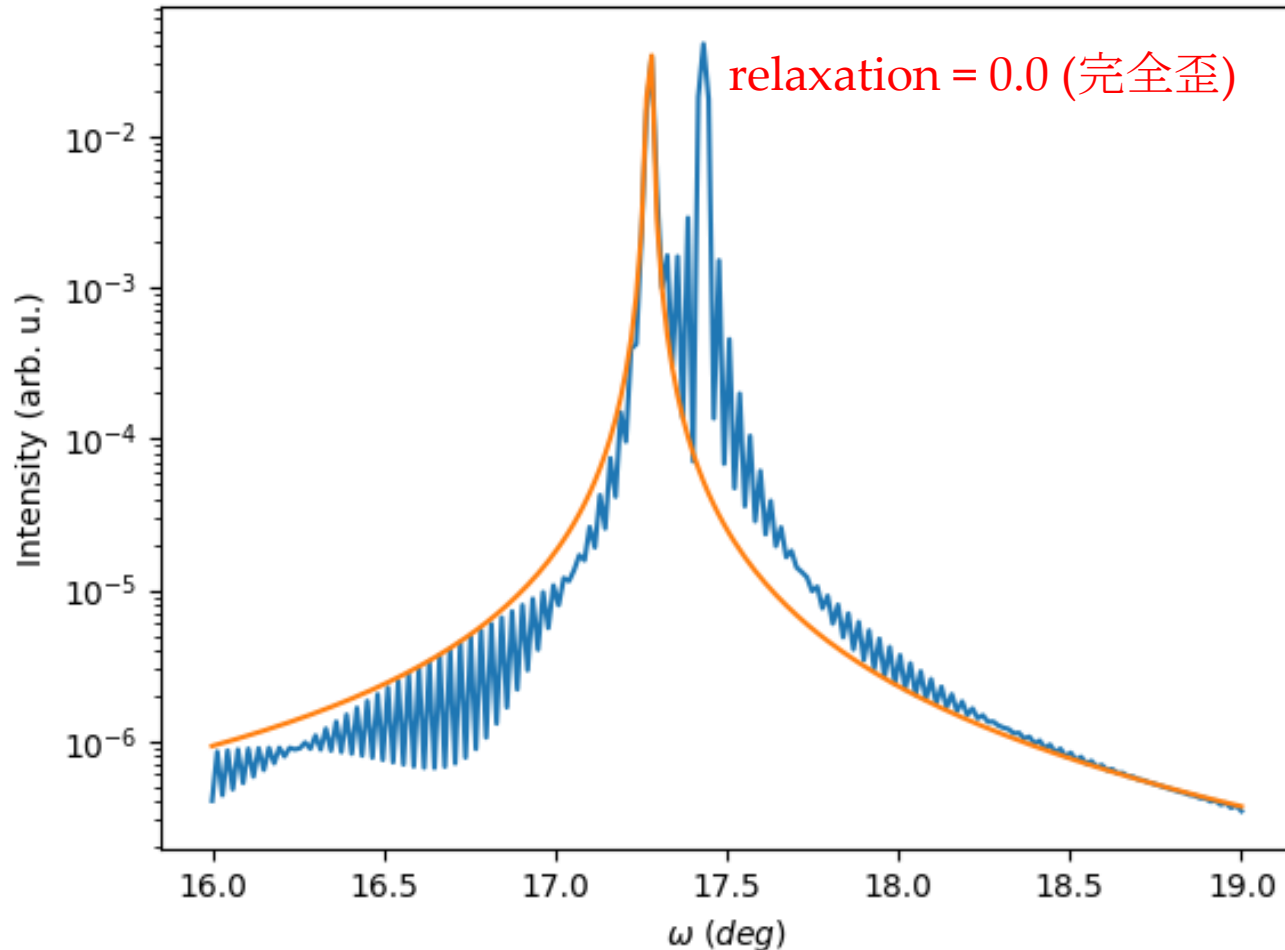
$\text{Al}_x\text{Ga}_{1-x}\text{N}(\text{epi})/\text{GaN}(\text{sub})$ を計算してみる

## 4. $\omega - 2\theta$ スペクトルの計算

ここだけ変更!!

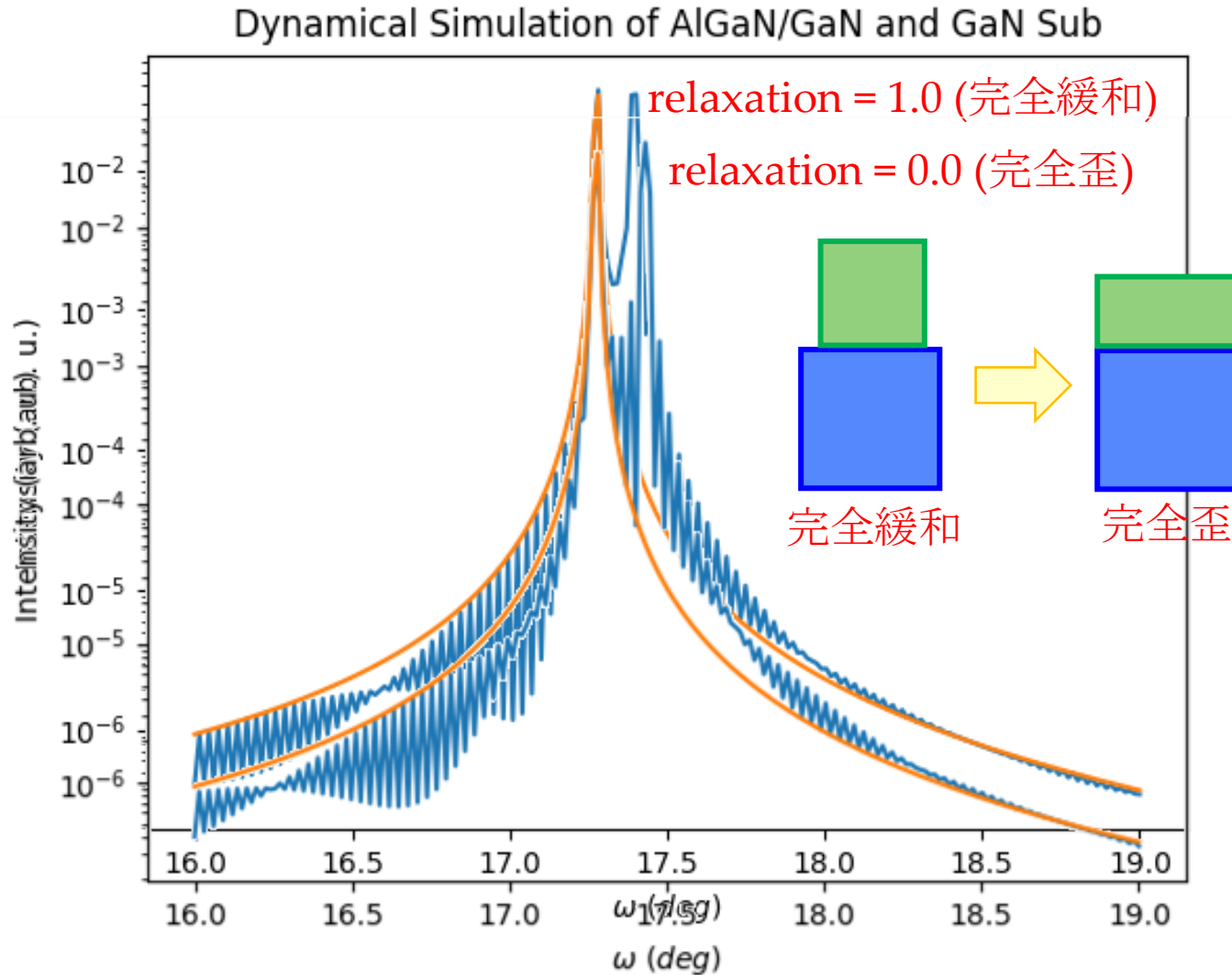
```
lAlGaN = xu.simpack.Layer( AlGaN, 1500, relaxation=0.0 )
```

Dynamical Simulation of AlGaN/GaN and GaN Sub



$\text{Al}_x\text{Ga}_{1-x}\text{N}(\text{epi})/\text{GaN}(\text{sub})$ を計算してみる

## 4. $\omega - 2\theta$ スペクトルの計算



# 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. $\omega-2\theta$ スペクトルの計算 フィッティング

ロッキングカーブの計算

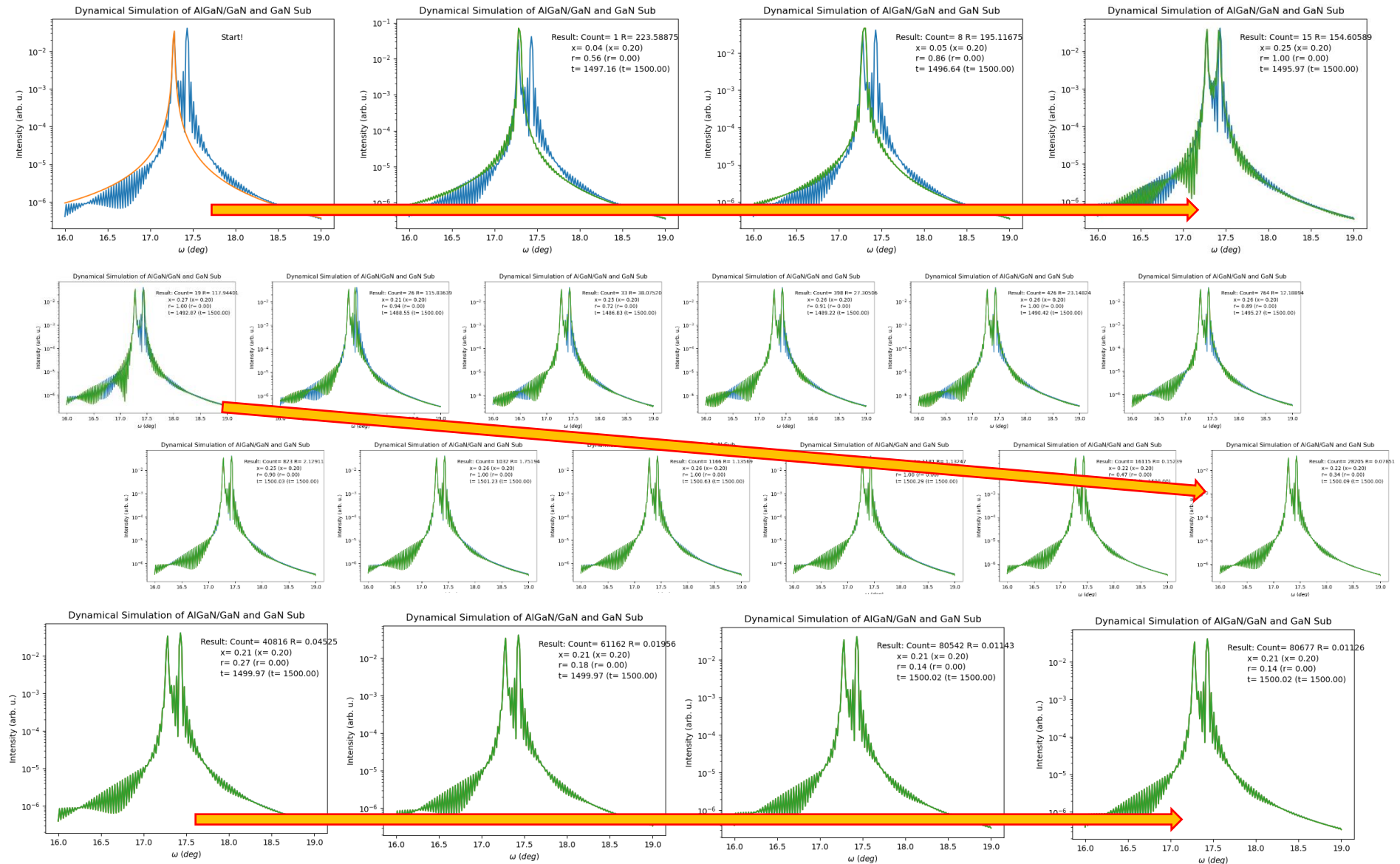
どうせならフィッティング(のシミュレーション)

## 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、緩和度をパラメータ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. 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.  $\omega$ - $2\theta$  スペクトルの計算

ロッキングカーブの計算

## 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 )
```

### $\omega$ の中心角をきちんと決めるため

```
# 一旦、 $\omega - 2\theta$  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 ]
```

$\omega - 2\theta$ 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

```
# 決まった角度を中心に計算する  $\omega$  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" )
```

## 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 )
```

**$\omega$ の中心角をきちんと決めるため**

```
# 一旦、 $\omega - 2\theta$  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  $\omega$  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

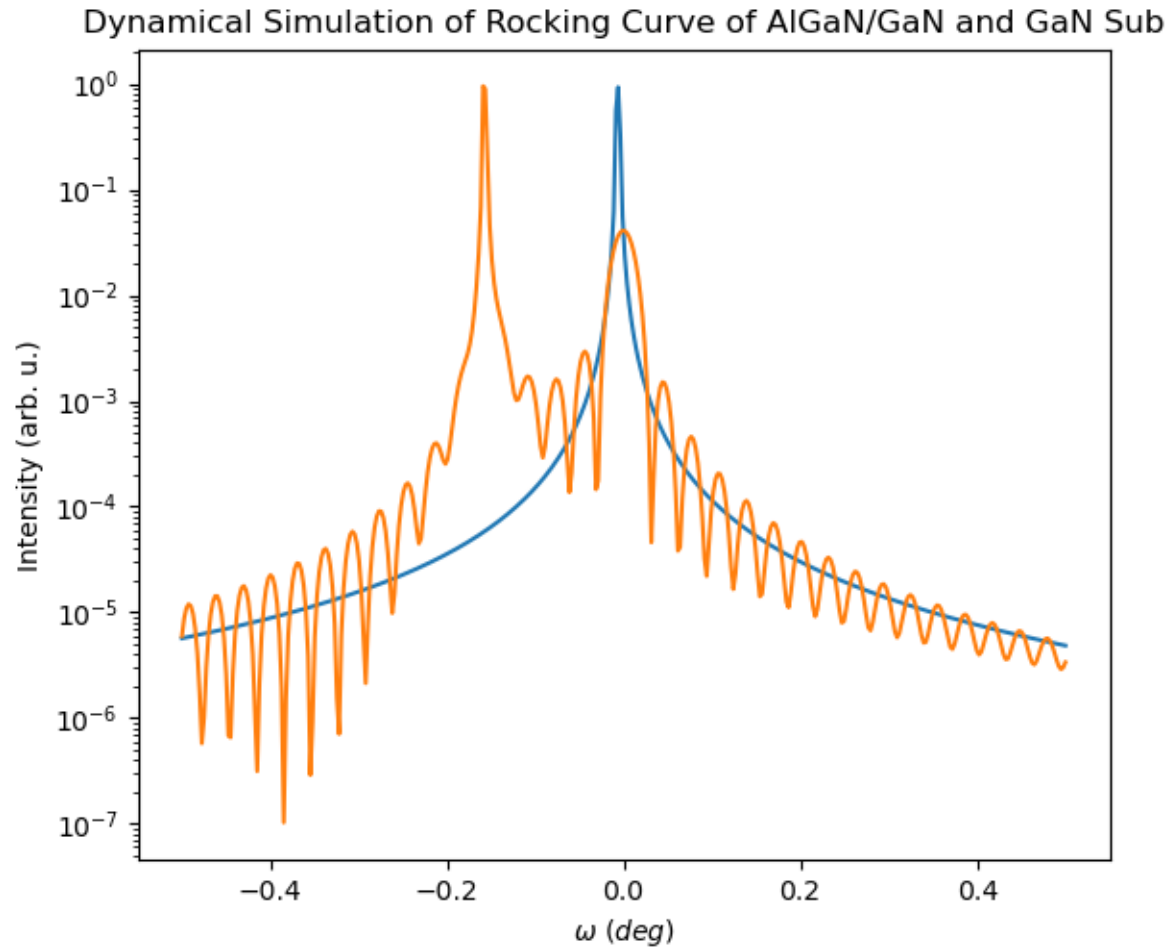
横軸の角度範囲は  
それぞれ違う

ピークが0度になるよう  
シフトさせておく

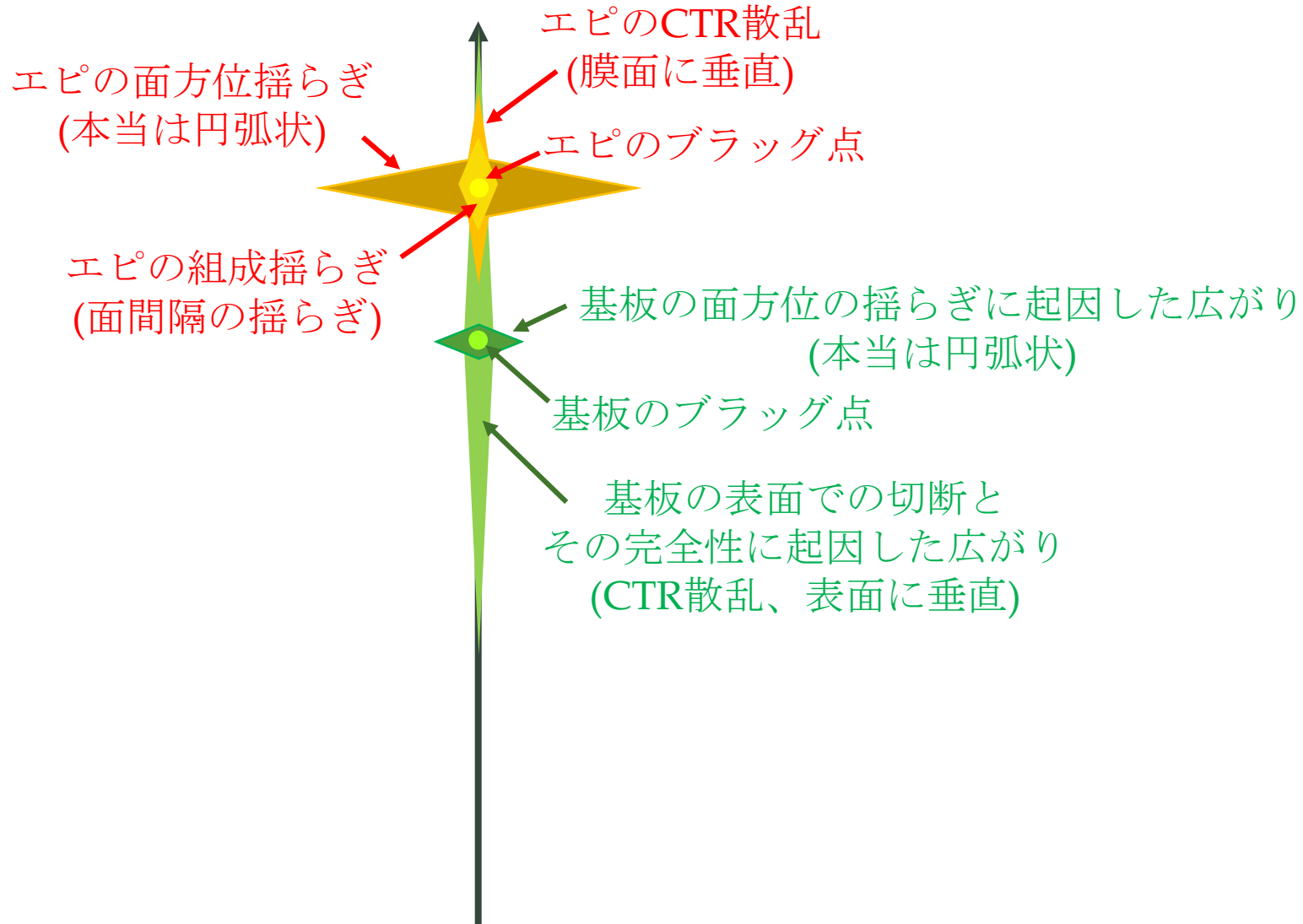
# 結果のプロット

```
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



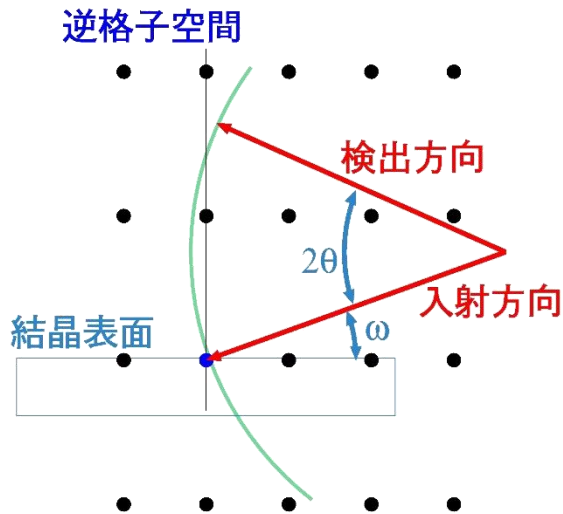
# 4. ロッキングカーブ？



# 4. ロッキングカーブ？

エピの面方位揺らぎ  
(本当は円弧状)

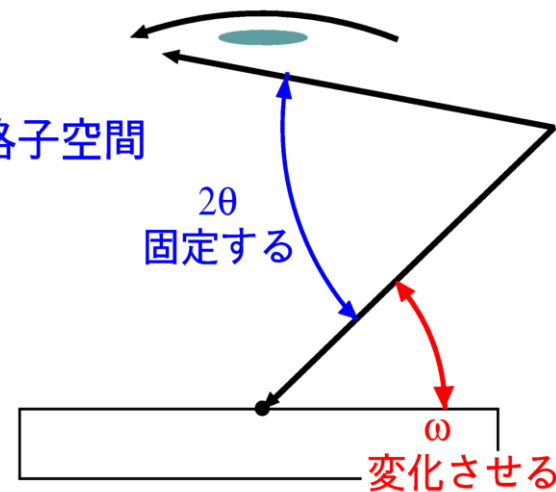
ロッキングカーブ測定に期待するのは  
この円弧上の散乱強度分布



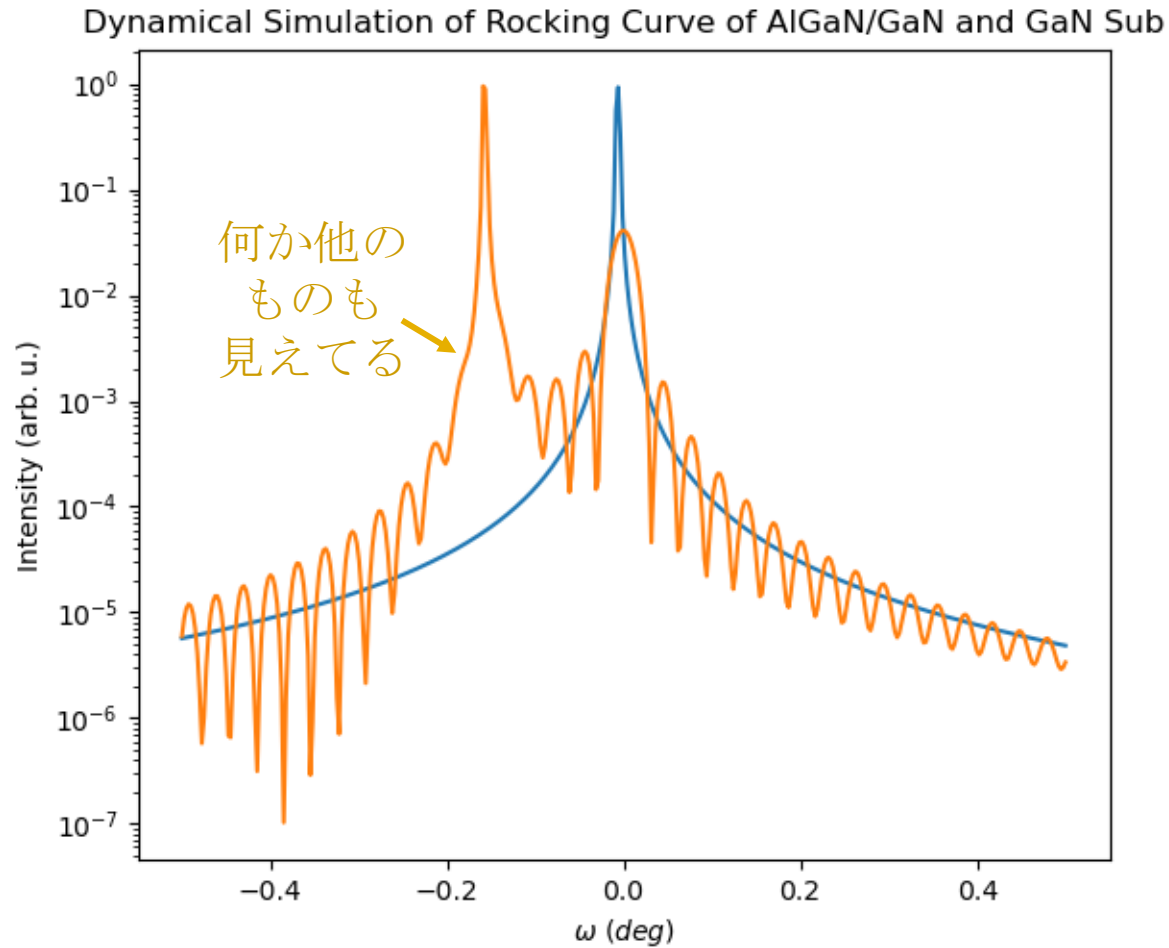
逆格子空間

$2\theta$   
固定する

$\omega$   
変化させる



# 4. ロッキングカーブの計算

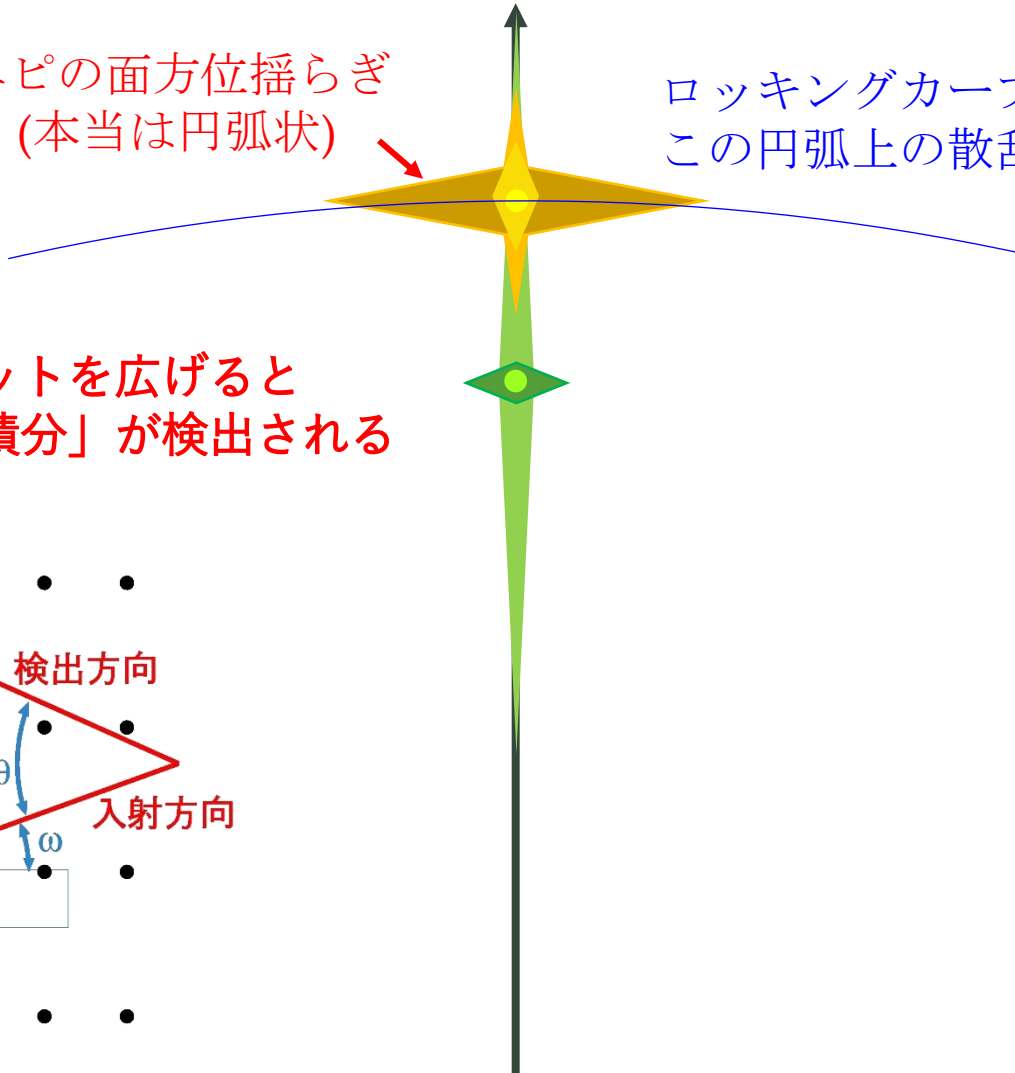
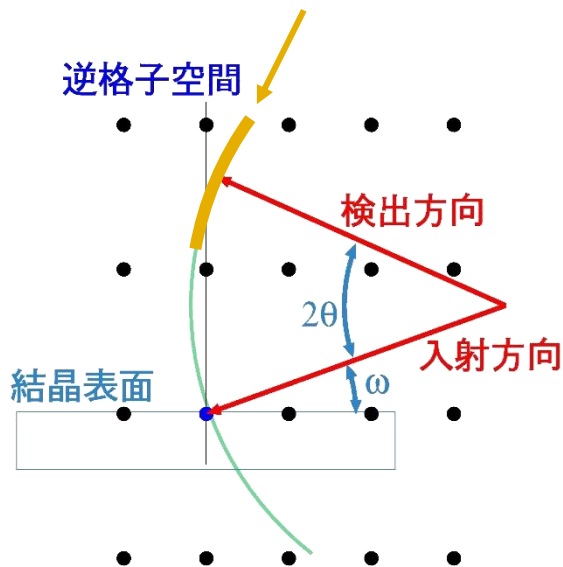


# 4. ロッキングカーブ？

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

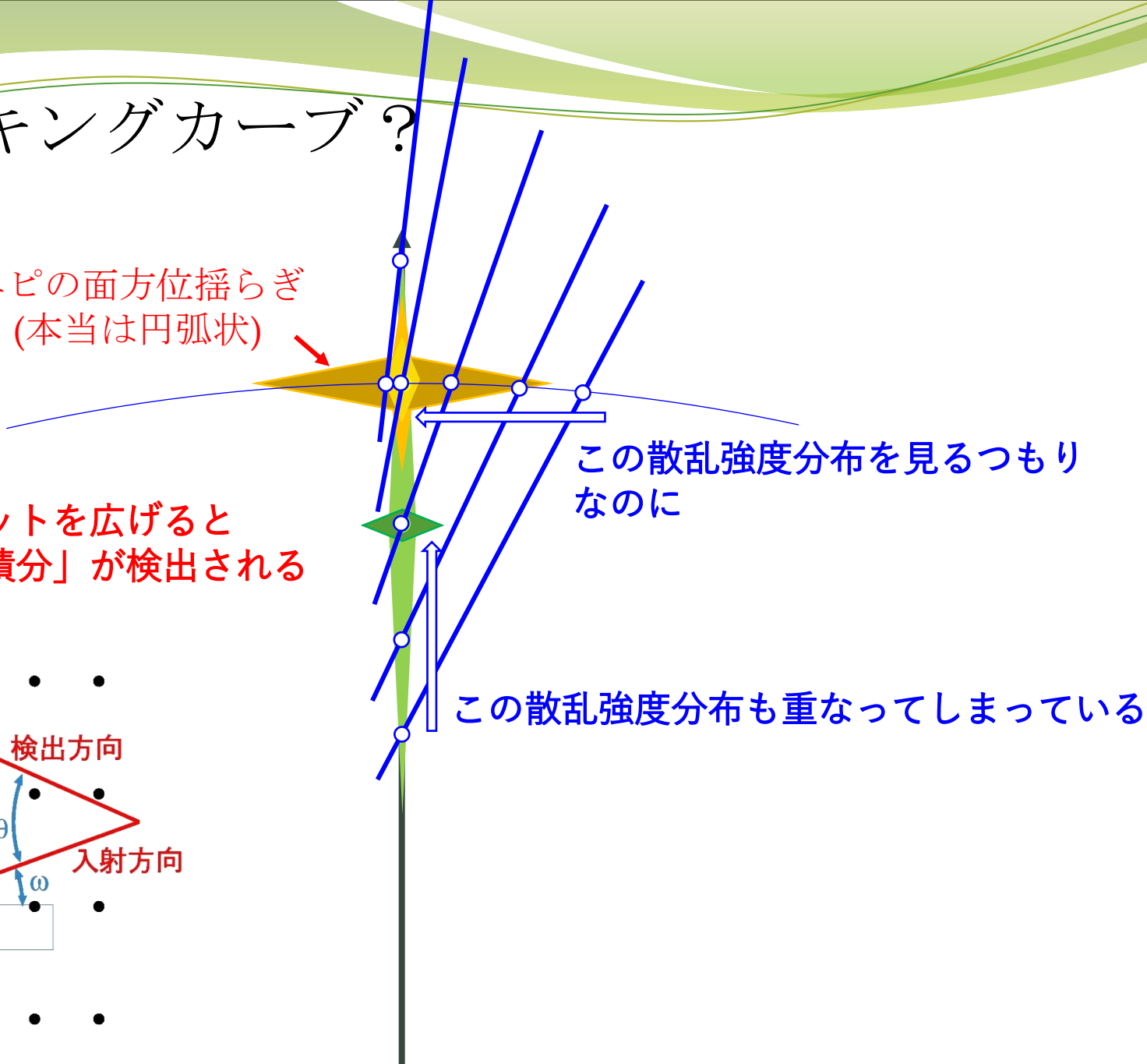
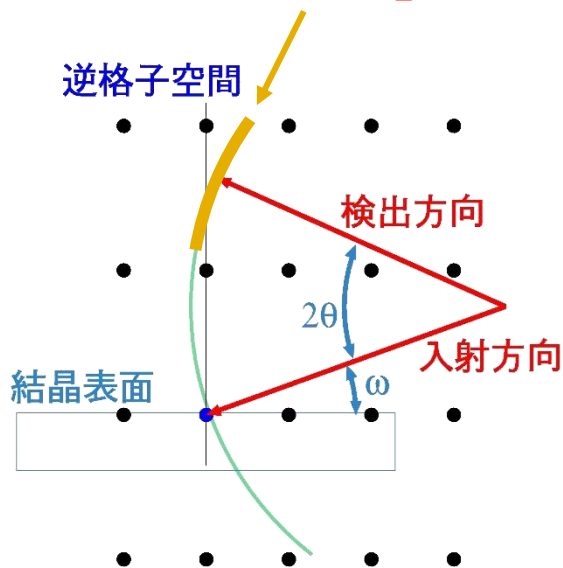
受光側のスリットを広げると  
この範囲の「積分」が検出される



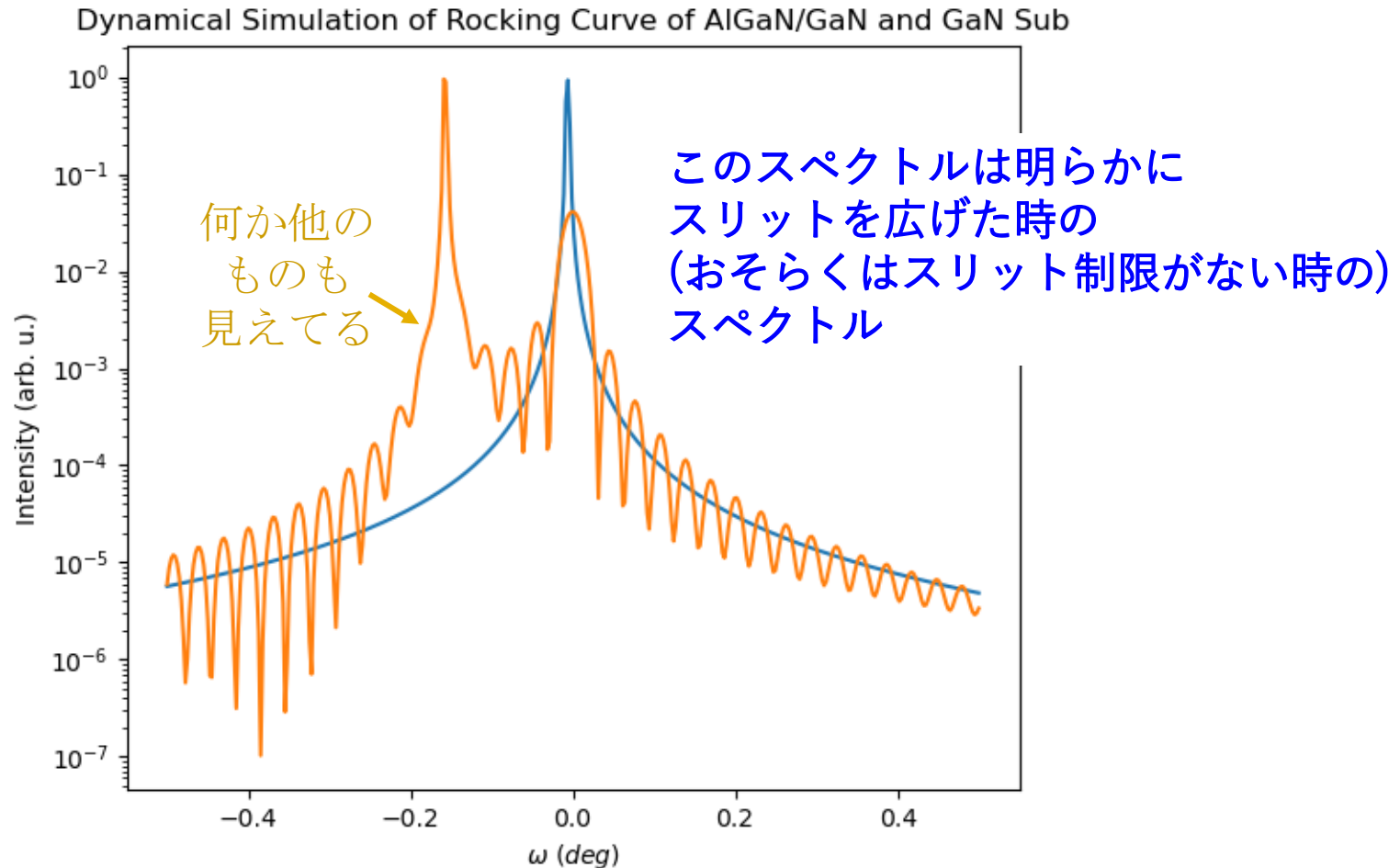
# 4. ロッキングカーブ？

エピの面方位揺らぎ  
(本当は円弧状)

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



知見: 「xuはこの様な計算をする」

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？」

- xu は確かに便利
- 本当のところ何をやっているのかわからない部分も多いので無条件に信じるのは危険そう
- やりたいことが、「すぐに」、「簡単に」、「便利に」行えることも多いが、違うことをしようとするともっともできるかどうかも含めて調べるのは大変
- xu は過渡期らしく、web 上のドキュメントも正しいものを探すのは少し難しい。
- 近い将来 xu が安定し、ドキュメントが充実することを祈っています。