

# energy.mac

マクロのpoint

## 1. calcM $\lambda$ に対するモノクロメータの角度計算

BL17では、 $\lambda$  をpulseに変換する計算を行う。

```
pulse = (hc_over_e/x+35.7257863)/0.0000549722779  
A[mono] = pulse
```

originalでは、 $\lambda$  をからモノクロの角度を計算する。

```
x = hc_over_e / (($1) * 2 * g_mod_d, x = asin(x), A[Mono] = deg(x)  
g_mod_d は モノクロのd値
```

## 2. calcE モノクロメータの位置から $\lambda$ を計算する

BL17では、mono(pulseの値から) $\lambda$ を計算する。

```
LAMBDA = 0.0000549722779*A[Mono]-35.7257863
```

originalでは、 $\lambda$  をからモノクロの角度を計算する。

```
LAMBDA = 2 * g_mod_d * sin(rad(A[Mono]))  
g_mod_d は モノクロのd値
```

## 3. getE エネルギーの値を求める

originalとBL17は、同じ。

モノクロの位置を得てから、 $\lambda$ を計算する。

```
waitmove; get_angles; calcE
```

## 4. moveE エネルギーを変える

originalとBL17は、同じ。

モノクロの位置を得てから、モノクロメータの角度を計算、  
実際に動かして、move\_em, 位置を得た後、エネルギーを求める

```
waitmove; get_angles; calcM $1  
move_em; waitmove; get_angles; calcE
```

## 5. BL17のマクロ (energy\_bl17su.mac) とオリジナル (energy\_original\_spec\_version5.mac) の違い

1. calcMとcalcEの違いだけ
2. eV単位とkeV単位 (hc\_over\_e=12398.42, hc\_over\_e=12.39842)
3. motor nameは A[Mono]ではなくA[mono]を使用している。

## 6. Original macrosの場所

~/Sync/documents/Applications/spec/spec\_6.08.05-1\_Beta\_osx-mojave\_ytanaka/macros

## 7. BL29用のマクロ (energy\_bl29.mac) の作成

### 方針

サーバにはenergyに関して、E [keV],  $\lambda$  [Å], Angle [deg]の三つのパラメータがある。これをspecから制御するとき、一つのパラメータでやりとりするのがいいと思う。基本的にAngle [deg]の値をサーバから拾ってきてそれをspecの中でE [keV],  $\lambda$  [Å]に変換することにする。このAngle [deg]の値をsudo motor A[Mono]とする。

calcM: spec  $\lambda$ を角度に変換するマクロ

```
(x = hc_over_e / (($1) * 2 * g_mo_d) > 1) {  
  x = asin(x)  
  A[mono] = deg(x)  
}
```

g\_mo\_dの値を決めておかないといけない。

setmonoを実行すると、自動的にg\_mo\_d=3.13542 [Å]が得られる。

ほとんどoriginalからの変更点がなかった。A[Mono]をA[mono]としただけoriginalからの変更点は、IDGAPの表示をgetE moveEに加えたこと。

## 8. サーバとの通信

getE, moveEは”get angles”を通してA[Mono]の値をサーバに読みに行く。

この時、A[Mono]をsudo motorとするためその定義として、mono\_bl29.macが必要になる。

BL17の場合は、grating.mac 中の”move\_grP2”と”get\_grP”で定義

```
def bl17_add_energy_pseudo '{  
  cdef ("user_moveall",sprintf("move_grP2;"),"mono", 0x1)  
  cdef ("user_getpangles",sprintf("get_grP;"),"mono", 0x1)  
}'
```

BL29の場合は、”mono\_bl29.mac” 中の”move\_EE”と”get\_EE”で定義

```
def bl29_add_energy_pseudo '{  
  cdef ("user_moveall",sprintf("move_EE;"),"Mono", 0x1)  
  cdef ("user_getpangles",sprintf("get_EE;"),"Mono", 0x1)  
}'
```

さらにBL29の場合は、Energy [keV]とID gap [mm]を連動させる。

move\_EEの中身

```
get_EE  
ENERGY=hc_over_e/LAMBDA  
IDGAP = 2.674 + 1.26*ENERGY  
moveID IDGAP
```