

spec 簡易マニュアル (BL-4C)

2005.2.01

spec 関連部分:若林裕助

C-plot 関連部分:若林裕助, 小山一郎¹

1 困った時は

キーボードが入力を受け付けなくなった

spec に `cntrl-q` を入力して見て下さい。

パルスがずれたと言ってきた

パルスモーターコントローラが信用できる時は No, 信用できない時は Yes を入れて下さい。

ビームモニターに計数されなくても測定を続けたい

`chk_thresh=-1` して下さい。monitor の計数が `chk_thresh` より大きくなると、ビームが来ていないと判断して測定を止めます。

何故か波長を 1.54\AA と思っているみたいだ

`getE` して下さい。

エネルギーを動かしている最中にモノクロのハードリミットに当たったという

無視してもう一度動かして下さい。

温調と通信できない

`_LTC_is_on_ = 1` として下さい。

spec が計算する四軸角がおかしい

1. phi の回転方向が正しいかチェック
2. chi の回転方向, 原点が正しいかチェック
3. th の原点を架台調整後動かしていないかチェック (データファイル中の “reset” という文字列を検索)
4. spec を抜けて, `fourc -f` で新規に起動しなおす。(or0 などの情報を全て忘れます)

架台調整でバックラッシュが変に大きい

1. 止めねじを緩めたか確認
2. th や tth の原点がずれていないか, 回折計の目盛りを見て確認
3. ビームライン担当者を呼ぶ

¹ Present address:東大工

2 起動

```
huber6:~$ fourc | tee -ai filename
```

と書くことで、*filename* に画面表示のログファイルを作っていきます。これがないと、scan の結果だけのようファイルが残るだけになりますので、トラブルが起こったときの対処が難しくなります。意味は、fourc を起動し、その出力を filename のファイルにも書き込み、という程度です。tee というのが、出力の分岐を行っています。ログファイルは帰る前に削除してってください。このファイルは非常に大きくなり、ハードディスクの容量を圧迫します。

spec は前回終了時の情報を記憶しています。読み込んだマクロや格子定数などの記憶を保持しています。前回の記憶を無くさせたい時には、fourc に -f オプションをつけて起動して下さい。

fourc を起動すると、

```
Welcome to "fourc" Release 4.03.11
```

```
      :
```

```
Opening "SPEC/CD/site.mac" at input nest level 1.
Opening "SPEC/CD/stage.mac" at input nest level 2.
Opening "SPEC/CD/monochro.mac" at input nest level 2.
Opening "SPEC/CD/slit.mac" at input nest level 2.
Opening "SPEC/CD/atten.mac" at input nest level 1.
```

```
1.FOURC>
```

と出て、コマンドを受け付けるようになります。

手持ちのパソコンをネットワークに繋いで、グラフを描くなどの作業を行いたい場合、spec の画面表示がパソコンにも出ていると、測定結果をコピーしてカレイダグラフ等にペーストできるので便利です。パソコンの方に spec の画面表示を出すには、huber6.kek.jp に telnet して、ログファイルを tail コマンドで見る、という操作をします。

```
huber6:~$ tail -f filename
```

お分かりだと思いますが、huber6.kek.jp というのは BL-4C の制御用のコンピュータです。最初に起動したときは、startup というマクロを実行する必要があります。

```
1.FOURC> startup
```

で、startup マクロを実行してください。大部分の項目はただ return を押すだけで問題ありません。入れる必要があるところは、データファイルの名前と、エラーバーを表示するかどうか程度です。

データファイル名入力例：

```
(newsample)
```

```
Title for scan headers (fourc)?
```

```
(newfile)
```

```
Data file (/dev/null)? LSM0327.dat      <---ここにデータファイル名
Last scan number (0)?                   ~~~~~
Using "LSM0327.dat".  Next scan is number 1.
```

エラーバーを表示するかどうか：

(setplot)

```
1) Do real-time screen plots during scans (YES)?
2) Do screen plot after scan (YES)?
8) Range x axis with scan min and max (NO)?
16) Force y-axis minimum to zero (NO)?
32) Use logarithmic y-axis (NO)?
64) Do background subtraction for plots (NO)?
128) Use high resolution plotting device (YES)?
    What kind of high-res graphics terminal (x11)?
256) Draw large dots at points (YES)?
512) Connect points with lines (YES)?
1024) Draw error bars (NO)? y      <---ここだけ注意
    Sum of selections is 131.      ~~~~~
```

1024) という欄に y と答えます (エラーバーなんか表示されなくても良い、と言う人は別にそのままリターンでも良いです。)。

格子定数などは、後からでも足りしますのでさほど重要ではありません。後から入れるときは、setlat コマンドを使います。もちろん、ここで入れてもいいです。

ここまでで、最初の準備は終わりました。

spec を終了するには quit と入力します。

3 架台調整

3.1 このセクションで使うコマンド, キーワード

<code>moveE energy</code>	<code>energy</code> keV の X 線にあわせてモノクロを設定する。
<code>GO arg</code>	<code>arg</code> が 2.5 より小さかったら, <code>arg</code> Å の波長にモノクロを設定する。それより大きな引き数の場合, モノクロの θ を <code>arg</code> ° に設定する。
<code>cvslit1 center width</code>	入射側スリットの縦を, 中心を <code>center</code> に取って, 幅 <code>width</code> に設定する。2 文字目を h にすると横。最後の 1 を 2 にすると下流側スリットを動かす事になる。
<code>dscan motor_name start stop interval time</code>	<code>motor_name</code> のモーターを <code>start</code> から <code>stop</code> まで, <code>interval</code> +1 点測定します。一点あたりの露光時間は <code>time</code> 秒です。 <code>start</code> , <code>stop</code> を, 現在地からの相対値で表します。
DTH	モノクロの $\Delta\theta$ を動かすモーターの名前。
DET	<code>spec</code> が画面にグラフを書く時に, 何番目のチャンネルでグラフを書くかを表すパラメタ。通常, 1 がビームモニター, 3 が後ろのイオンチェンバー, 2 がシンチである (2005.2.1 現在)。
<code>plotsselect</code>	DET とほぼ同様。 <code>plotsselect</code> [Enter] の後, <code>mon</code> (or) <code>det</code> (or) <code>c3</code> のどれかを入力することで, モニター, シンチ, 下流 IC のどれで測定するかを決める。
<code>chk_thresh</code>	ビームモニターの値がこれより小さくなったら測定を中止するという閾値を表す。
<code>mvr_st direction length</code>	ステージの移動。 <code>direction</code> は <code>rx,y,ry,z,rz</code> のどれかで, <code>length</code> は移動量もしくは移動角度。 <code>rx</code> はビーム方向を軸にした回転, <code>y</code> はビームに垂直な水平方向の移動, <code>ry</code> は <code>y</code> を軸にした回転, <code>z</code> は鉛直方向, <code>rz</code> は <code>z</code> を軸にした回転。
<code>scan_st direction start stop interval time</code>	ステージスキャンを行う。 <code>direction</code> に使えるのは <code>mvr_st</code> と同じ文字。 <code>start</code> , <code>stop</code> は現在地からの相対値で指定する
CEN	直前のスキャンの半値幅の中心を表す変数。

3.2 手順

現在, 波長をどこに設定しても, 1mm とビーム位置はずれません。そのため, ビームを見失うということとはまずありません。

まずは自分の使いたい波長を出します。moveE が, あるいは GO を使って, 波長変更をします。続いて, 上流, 下流ともスリットを開いておきましょう。cvslit1 0 10;chslit1 0 10;cvslit2 0 10;chslit2 0 10[改行] として下さい。これで上流, 下流の縦横とも 10mm に開きます。

次に, 必要ならばモノクロのデチューンを行います。² デチューンする人もしない人も次のような操作を行って下さい。

```
34.FOURC> DET=1
```

```
35.FOURC> chk_thresh=-1
```

² デチューンは高調波 ($\lambda/2$) をカットするために行います。4C では 21keV 以上の光はミラーでカットされますので, 11keV 以上の光を使うユーザーはデチューンする必要はありません。デチューンしない場合はビームサイズが 0.5mm×0.5mm 程度, デチューンした場合は 0.8mm×1.2mm 程度になったと記憶していますが, もしかしたらエネルギーに依存する部分もあるかもしれません。

```
36.FOURC> dscan DTH -0.02 0.02 30 1
```

```
37.FOURC> mv DTH *** <--ここに DTH の行き先を書く
```

と行います。ビームモニタでグラフを書くようにし、ビームモニタに何も入らなくても測定を続けるように設定し、DTH スキャンを行う。その後、DTH を好きな値に設定する。デチューンする人は半分くらいの強度になるように、しない人はピークトップに、それぞれ持って行って下さい。

次に DET=3, plotselect c3 として後ろのイオンチェンバーでグラフを書くように設定し、架台調整に使うピンホールを回折計に取り付けます。最初からダブルピンホールで、1mm 程度の物から始めても大抵見えます。架台の止めネジ (4 箇所) をゆるめることを忘れないように注意して下さい。

架台は 4 軸動くようになっています。y, z, ry, rz です。それぞれ、水平方向、鉛直方向、y 軸を中心とした回転、z 軸を中心とした回転、を表します。単位は mm で、ry, rz は架台の前の方を +xmm, 後ろを -xmm 動かす、という表現になっています。

架台スキャンは、scan_st というコマンドで行います。現在位置からの相対値で測定範囲を指定します。scan 終了後は最初の位置に戻ります。架台の移動は mvr_st で行います。これも現在位置からの相対値で表します。

ピークの形がきちんとしている時、便利な変数 CEN を使う事ができます。CEN という変数には、直前のスキャンの半値幅の中心が代入されています。ピーク位置は pl_xMAX という変数に代入されています。これを使って、次のようにしてピークをリファインしていくことができます。

```
57.FOURC> scan_st y -2 2 30 1
```

```
:  
:
```

```
58.FOURC> mvr_st y CEN
```

```
59.FOURC> scan_st rz -2 2 30 1
```

```
:  
:
```

ただし、CEN はピークトップの半分のカウント数までピークの両側で落ちていないと正しい値を返しません。ここは人間が判断してやる必要があります。

どうやら、ry だけ最近バックラッシュがあるようです。目標地点に移動した後、mvr_st ry -1;mvr_st ry 1 とすると思っている位置に動くようです。

架台調整が終わったら、最初にゆるめたネジを締めて下さい。

4 スリット調整

自動化されましたので、架台調整が終わった状態で slit1;slit2[改行] と入力してください。重要なのは、下流イオンチェンバーでスキャンのグラフ表示をするようになっていること、及びピンホールを通った光が全部下流イオンチェンバーに入っていることです。slit1;slit2 が終わった後は、スリットは開いた状態になっています。

以下は、手動で調整をやる時のやり方です。

4.1 このセクションで使うコマンド, キーワード

`ascan motor_name start stop interval time` `motor_name` のモーターを `start` から `stop` まで, `interval+1` 点測定します。一点あたりの露光時間は `time` 秒です。

`s1t, s1b, s1l, s1r, s2t, s2b, s2l, s2r` それぞれ前スリット上, 下, 左, 右, 後スリット上, 下, 左, 右。左右は下流から見てのもの。

4.2 手順

流儀はいくつか有り得ますが, 私が PF に来た時に習った手法を示します。

回折計にピンホールを入れたままの状態, 後ろのイオンチェンバーをカウンターとして使います。後ろのイオンチェンバーの計数が, スリット開放時の半分になるようにスリットのブレードを閉じ, そこをそのブレードの 0 と定義します。

```
98.FOURC> ascan s1t -1 1 40 1
      :
      :
99.FOURC> mv s1t ***          <--ここに半分になった s1t の座標を書く
100.FOURC> set s1t 0
101.FOURC> mv s1t -5
102.FOURC> ascan s1b 1 -1 40 1
      :
      :
```

スリットの座標は, 下流側から見て上から下がプラス, 左から右がプラスです。スリットは閉じる側にバックラッシュを取る設定になっていますので, スキャンの方向によって時間がかかる方, かからない方が存在します。8枚のブレード全部で数分の違いになると思いますが, これを気にしないなら適当にスキャンをかければいいです。

ここでも, 前回の人が 0 と定義していたところから 1mm とずれない場合がほとんどだと思います。この作業が終わったら, ピンホールを外します。

5 センター出し, サンプルマウント

5.1 このセクションで使うコマンド, キーワード

`config`

5.2 手順

センター出しは省略します。サンプルマウントも好きなようにやるといいでしょう。chi と phi は皆さんネジをゆるめて適当にぐるぐる回しているでしょうから, 設定を間違える可能性があります。chi の正の方向は下流から見て反時計回りで, chi=0 にした時, th と phi が回転方向も含めて一致する様にセットして下さい。

phi の回転方向を逆にしたい場合があると思います。その場合, `config` で phi の Sign of user * dial を 1, あるいは -1(今の設定と逆) にして下さい。また, どちらの場合も chi の設定をいいかげんにしないで下さ

い。chi 軸が鉛直な状態が $\pm 90^\circ$, 水平な状態が 0° , 180° , chi の正の方向は下流から見て反時計回りです。回折計に書いてある目盛りを見て、 $\pm 0.1^\circ$ の桁まで合わせて下さい (ここをいいかげんにすると ϕ 軸が曲がって回転することになるので、軸立てが困難になります)。

config はいいいかげんな書き換えをしないで下さい。場合によっては修復に数日を要す可能性もありますし、嘘のデータを吐き出している事に気づかない可能性も 0 ではありません！わからなかったらスタッフを呼んで下さい。

6 軸立て

6.1 このセクションで使うコマンド, キーワード

setlat	格子定数入力
or0	UB の計算に使う第一のブラッグ位置設定
or1	UB の計算に使う第二のブラッグ位置設定
reflex_beg	reflex モード開始
reflex	reflex ファイルに書き込み
reflex_end	reflex モード終了
qdo <i>filename</i>	マクロファイル <i>filename</i> 実行
calcL	reflex 後の格子定数の計算
pa	or0,1, 格子定数の表示
set_lm <i>motor_name lower upper</i>	リミッタ設定

6.2 手順

シンチレーションカウンタの設定を行います。これは SPEC とは無関係なのでここでは省略します³。必ずこの段階で調整を行ってください。カウンターを壊す恐れがあります。

setlat で格子定数を入れます。

Bragg を見つけます。

or0 h k l (h, k, l はそのブラッグの指数) と入力します。

もう一つ, 独立な Bragg を見つけます。

or1 h k l (h, k, l はそのブラッグの指数) と入力します。

これで, 一応は UB が計算されます。この場合, or0 と格子定数を信用して, or1 は参考につかわれるだけです。また, or0, or1 は方向だけを見て, 長さを見ません。つまり, 例えば or0 0 2 0 とした場合, その位置は $(0\eta 0)$ と設定されますが, η が何になるかは格子定数と波長から計算され, or0 に合わせて格子定数を計算し直したりはしません。次に or1 1 2 0 とした場合について考えます。これによって a 軸の方向が決められるだけで, or1 1 2 0 を実行した場所が $(\xi 2\xi 0)$ になる保証はありません。こうしようとしても, 格子定数を書き換えないと一般にはあいませので, or0, or1 は格子定数を優先するという都合上, 普通はどうしてもずれます。

いくつかの反射から格子定数, UB を最小二乗で計算するには reflex_beg reflex reflex_end という 3 つの一連のコマンドを使用します。まず reflex_beg コマンドを使用して各反射の四軸角を一時的なファイルに記録します。

20.FOURC> reflex_beg

³ これは SPEC のマニュアルであり, 装置のマニュアルではありません

Reflections file (reflex)?

デフォルトのファイル名は reflex です。次にこのファイルに四軸角を順に記録していきます。

21.FOURC> ここで 2 2 0 反射を見つける

22.FOURC> reflex 2 2 0

23.FOURC> ここで 2 0 2 反射を見つける

24.FOURC> reflex 2 0 2

25.FOURC> これらの手順を繰り返す

26.FOURC> reflex_end

Type "qdo reflex" to recalculate orientation matrix

reflex_end コマンドは入力された反射から reflex ファイルを実際に作成させるために用います。最後に reflex ファイルを実行します。

27.FOURC> qdo reflex

Opened command file 'reflex' at level 1.

これで UB が最小二乗 fit されます。格子定数は、calcL を実行後、pa を実行すると見ることができます。or0, or1 などを実行すると、UB の値を上書きしてしまうので、その場合には qdo reflex を再度実行して下さい。

7 コマンドリスト

使用頻度が高い、もしくは必ず使わなくてはならないコマンドとその説明をならべました。

7.1 モーター駆動

mv motor_name angle

motor_name の名前のモーターを *angle* に回す。角度は絶対値。スリットなどのモーターの場合、*angle* は角度ではなく位置を表す。

例：mv tth 30

mvr motor_name angle

mv の角度設定を現在地からの相対値で表したものの。

例：mvr th -1

umv motor_name angle

mv の移動中、現在地を表示するもの。

例：umv phi 20

umvr motor_name angle

mvr の移動中，現在地を表示するもの。

例：umvr chi -2

an tth th

tth と th を一度に動かす。角度指定は絶対値。

mvr_st direction length

ステージの移動。*direction* は rx,y,ry,z,rz のどれかで，*length* は移動量もしくは移動角度。
rx はビーム方向を軸にした回転，y はビームに垂直な水平方向の移動，ry は y を軸にした回転，z は鉛直方向，rz は z を軸にした回転。

例：mvr_st z 1

atten attenuator

アッテネーターを変える。atten *n* で，*n* 番のアッテネーターに設定する。

set motor_name value

motor_name のモーターの値の原点を変え，現在の値を *value* にする。

例：set th 0

set_lm motor_name low_limit high_limit

motor_name のモーターのソフトリミットの設定を行う。

例：set_lm th 0 40

config

リミッタの設定などを行います。カーソルキーで変更したいものにカーソルを持っていき，書き換えます。書き換えを決定するには，w を押します。メニューから抜け出るには ctrl+c で抜けます。リミッタ以外の部分の書き換えは，通常は行わないでください。

wa

モーターの位置を表示します。二段に表示されますが，上段の数字が set で 0 点を設定した後の値で，通常使うのは上段です。

wh

wa の縮小版

7.2 モノクロメータ

GO arg

arg が 2.5 より小さかったら，*arg* Å の波長にモノクロを設定する。それより大きな引き数の場合，モノクロの θ を arg° に設定する。

例：GO 1.8905

moveE energy

energy keV の X 線にあわせてモノクロを設定する。

例：moveE 6.655

getE

現在の X 線のエネルギー，波長を表示する。

例：

getE

E = 12.3985 keV, 1.00001 Angstroms

7.3 四軸関連

or0 *h k l*

現在地を (*h k l*) の逆格子点であると設定し，UB の計算に用いる。

or1 *h k l*

or0 と同じ。格子定数がわかっている場合で，UB を立てるのに最低二つは点がいるため。

ca *h k l*

逆格子の *h k l* の位置の角度を計算する。

例：

```
7.FOURC> ca 1 0 0
```

Calculated Positions:

H K L = 1 0 0

Alpha = 0 Beta = 0 Azimuth = 90

Omega = 0 Lambda = 1.54

Two Theta	Theta	Chi	Phi
60.0000	30.0000	0.0000	0.0000

br *h k l*

h k l に回折計をあわせる。

ubr *h k l*

br の u 版。

setlat

startup の所で出てきた、格子定数を入力するコマンドです。これが既に入力してあることが四軸関連のコマンドが動作する前提条件となります。

reflex

3つ以上の Bragg から UB と格子定数を決定する。詳細は section 軸立て を参照。

onsim/offsim

四軸回折計を実際に動作はさせたくないが、特定の四軸角を or0,or1 に入力したい、という場合があります。この場合、spec をシミュレーションモードに切り替え、回折計を反射位置に移動させて or0/or1 を実行し、その後シミュレーションモードを解除する、という操作を行います。これを行うのが onsim offsim です。onsim で spec はシミュレーションモードになり、offsim でそれが解除されます。

7.4 スリット系

cvslit1 *center width*

入射側スリットの縦を、中心を *center* に取って、幅 *width* に設定する。

例：

```
13.FOURC> set slit 0
      :
      :
39.FOURC> cvslit1 0 1
```

スリットで半割を出して、半割の位置を set で 0 に設定しておく、このコマンドは第一引き数が 0 で使えて便利です。

cvslit2, chslit1, chslit2

最後の 1,2 が上流、下流のスリットを表し、二文字目の v,h が垂直、水平を表すとして、それぞれ対応した方向の cvslit1 と考えたもの。

7.5 測定

ct *time*

カウントして、結果を画面に表示する。*time* は省略可能。

uct

ct の u 版

ascan *motor_name start stop interval time*

motor_name のモーターを *start* から *stop* まで、*interval*+1 点測定します。一点あたりの露光時間は *time* 秒です。例：ascan th 30 32 30 1

dscan *motor_name start stop interval time*

ascan と同じですが、*start*, *stop* を、現在地からの相対値で表します。

th2th *start stop interval time*

$\theta - 2\theta$ スキャンを行います。角度は相対値で、 2θ で表します。

scan_st direction start stop interval time

ステージスキャンを行います。 *direction* に使えるのは *mvr_st* と同じ文字です。 *start*, *stop* は現在地からの相対値で指定します。

center width_th2th width_th width_chi width_phi interval time

自動ピークリファインマクロです。 *width_** の二倍の幅にわたって測定し、 θ - 2θ , ω , χ , ϕ スキャンを行います。3周イテレーションを行って、センターを決定します。

hkscan Hstart Hstop Kstart Kstop Lstart Lstop interval time

逆格子空間でのスキャンを行います。意味は分かるかと思います。 *start*, *stop* は絶対値であり、相対値ではありません。

hscan Hstart Hstop interval time

現在 ($\xi\eta\zeta$) にいるとした場合、($\Xi\eta\zeta$) の Ξ をスキャンします。類似品に *kscan*, *lscan* があります。 *Hstart* *Hstop* は絶対値です。

Escan start stop interval time (fixQ)

エネルギースキャンを行います。 *fixQ* オプションをつけると、エネルギー変化に伴って 2θ と ω を動かして同じ逆空間の点のエネルギースキャンを行います。エネルギーの単位は keV です。

a2scan motor_name1 start stop motor_name2 start stop interval time

motor_name1 と *motor_name2* を同時に動かしてスキャンします。 *th2th* の正体はこれです。

7.6 温度調整

LTC20 関連のマクロは `/usr/local/lib/spec.d/CRYO-CON.mac` です。通常は最初に読み込まれているはずですが、もし読み込まれていなかったらこれを *qdo* で読み込んで下さい。温調を始める時には、`_LTC_is_on_ = 1` として下さい。温調装置の電源を切った時には、必ず `_LTC_is_on_ = 0` として下さい。GPIB の応答が無いというエラーで、*scan* ができなくなります。

te set_temp

温度コントローラの *set temp* を *set_temp* にセットする。

showtemp

温度を読み取る。

7.7 マクロ実行

qdo file_name

file_name のファイルの中身を実行する。

マクロファイルの作り方

マクロファイルは通常のテキストファイルであり、なんらかのエディタでこれを書く必要があります。テキストエディタはいくつかありますが、これらの使用法は省略します。UNIXの参考書を見て下さい。1999年4月26日現在、BL-4Cで普通に使われているエディタはviとemacsです。muleはありません。Mac等で書いてftpしても構いませんが、specのマクロは改行コードが間違っていると動作しません。改行コードをLFのみにして下さい。

マクロファイルはユーザーが必要なコマンドを自分で作ったり、一連のコマンドを順に自動で実行させるために使われます。一連のコマンドを順に自動で実行させるためには、ただ、そのコマンドを一行ずつ書き並べるだけでOKです。例えば θ 軸を $\pm 1\text{deg}$ の範囲でスキャンし、その半値幅の中心に移動させるためには次のようなマクロを作ります。

```
dscan th -1 1 40 1
mv th CEN
```

これをtest.macという名前で保存したとするとqdo test.macと入力することによって実行させることができます。ここでCENという変数は直前のスキャンの半値幅の中心が代入されています。ピーク位置はpl_xMAXという変数に代入されています。

新しいコマンドを作成するには次のようにします。

```
def refine_th '
dscan th -1 1 40 1
mv th CEN
'
```

このようなマクロファイルを作成し、これを実行することによってrefine_thという新しいコマンドが作成されます。これらのコマンドにパラメータが必要であれば、以下のように\$1 \$2などの変数を使用します。

```
def refine_th '
if ($# != 3) {
print "Usage: refine_th range interval time"
exit
}
dscan th -$1 $1 $2 $3
mv th CEN
'
```

\$#という変数はコマンドにいくつパラメータが入力されたかを表し、これが3でないときは使い方を表示しています。printやexitはそれぞれご想像通りの動作です。

マクロの中で、その時点でのh,k,lの値を使いたい場合があると思います。その時には、whを行った後にH,K,Lの値を参照する事で、それらの値を使う事ができます。

7.8 その他

newfile

データファイルを新しく作り、次の測定からはその新しいファイルに書き出すようにする。

8 四軸関連の手法

四軸回折計に詳しい人にはあまり必要無いかも知れませんが、一応書いておきます。

四軸で、 h, k, l を決定すると、一つ軸が余分なので、何か条件をつけてやらないと回折計の角度が決まりません。そこで、いくつかの束縛条件の種類があります。spec では、setmode というコマンドで設定する、以下のモードがあります。

- 0.Omega Equals Zero
- 1.Omega Fixed
- 2.Zone or χ and ϕ Fixed
- 3.Phi Fixed or Three Circle
- 4.Azimuth Fixed
- 5.Alpha Fixed
- 6.Beta Fixed

* Fixed というモードの場合、freeze というコマンドで、Fix する角度を設定できます。 ϕ を回しながら同じ逆格子点を測定したい場合、次のようにします。

```
289.FOURC> setmode 3
290.FOURC> freeze 0
291.FOURC> br 1 0 0
292.FOURC> dscan th -.5 .5 30 1
293.FOURC> freeze 30
294.FOURC> br 1 0 0
295.FOURC> dscan th -.5 .5 30 1
296.FOURC> freeze 60
297.FOURC> br 1 0 0
298.FOURC> dscan th -.5 .5 30 1
      :
      :
```

9 管理者覚え書き

Serial port は/dev/cua0

spec.d は/usr/local/lib/spec.d

```
huber2:/usr/local/lib/spec.d/fourc$ more config
# ID @(#)getinfo.c      4.18  12/16/98  CSS
# Device nodes
PC_GPIBPC_L      = /dev/gpib0/master    @gpib_0
PC_BIT3PCI      = /dev/bit3 0x0
SDEV_0          = /dev/cua0 9600 cooked
PC_OMSV58       = 0xf000 8 POLL
PC_OMSV58       = 0xe000 8 POLL
PC_OMSV58       = 0xd000 8 POLL
GP_TSUJI        = 6 16
PC_VSC16T       = 0x1000 8 POLL
# Slot Assignments
# Motor  cntrl  steps  sign  slew  base  backl  accel  nada  flags  mne  name
MOT000 =   OMS_E   5000  -1   5000   300 -1000 2000    0 0x003   tth  Two Theta
MOT001 =   OMS_E   5000   1   5000   300 1000 2500    0 0x003    th  Theta
```

MOT002 =	OMS_E	20000	-1	15000	300	-2000	3000	0	0x003	chi	Chi
MOT003 =	OMS_E	250	1	1000	300	500	2500	0	0x003	phi	phi
MOT004 =	OMS_E	500	-1	1000	200	50	300	0	0x003	m4	Motor 4
MOT005 =	OMS_E	5000	1	2000	200	200	125	0	0x003	st1	stage 1
MOT006 =	OMS_E	5000	1	2000	200	200	125	0	0x003	st2	stage 2
MOT007 =	OMS_E	5000	1	2000	200	200	125	0	0x003	st3	stage 3
MOT008 =	OMS_E	5000	1	2000	200	200	125	0	0x003	st4	stage 4
MOT009 =	OMS_E	5000	1	2000	200	200	125	0	0x003	st5	stage 5
MOT010 =	OMS_E	200	-1	2000	200	0	125	0	0x003	att	Atten
MOT011 =	OMS_E	5000	1	2000	200	200	125	0	0x003	m11	Motor 11
MOT012 =	OMS_E	4000	1	2000	200	2000	125	0	0x003	s1l	Slit1 left
MOT013 =	OMS_E	4000	1	2000	200	-2000	125	0	0x003	s1r	Slit1 right
MOT014 =	OMS_E	4000	1	2000	200	2000	125	0	0x003	s1t	Slit1 top
MOT015 =	OMS_E	4000	1	2000	200	-2000	125	0	0x003	s1b	Slit1 bottom
MOT016 =	OMS_E	4000	-1	2000	200	-2000	125	0	0x003	s2l	Slit2 left
MOT017 =	OMS_E	4000	-1	2000	200	2000	125	0	0x003	s2r	Slit2 right
MOT018 =	OMS_E	4000	-1	2000	200	-2000	125	0	0x003	s2t	Slit2 top
MOT019 =	OMS_E	4000	-1	2000	200	2000	125	0	0x003	s2b	Slit2 bottom
MOT020 =	OMS_E	5000	-1	5000	100	-500	300	0	0x003	ttha	tth_a
MOT021 =	OMS_E	5000	-1	5000	200	-500	300	0	0x003	tha	Theta_a
MOT022 =	OMS_E	4000	-1	2000	200	2000	125	0	0x003	m22	Motor 22
MOT023 =	OMS_E	500	1	500	50	50	125	0	0x003	m23	Motor23
MOT024 =	TSUJI	36000	1	2000	200	50	125	0	0x003	mono	Mono Theta
MOT025 =	TSUJI	-36000	1	2000	200	50	125	0	0x003	PH2	PH2
MOT026 =	TSUJI	36000	1	2000	200	50	125	0	0x003	DTH	DTH
MOT027 =	TSUJI	-40000	1	2000	200	50	125	0	0x003	D1	D1
MOT028 =	TSUJI	-5000	1	200	100	50	500	0	0x003	MFV	MFV
MOT029 =	TSUJI	5000	1	200	100	50	500	0	0x003	MRV	MRV
MOT030 =	TSUJI	2000	1	200	100	50	500	0	0x003	MFH	MFH
MOT031 =	TSUJI	2000	1	200	100	50	500	0	0x003	MRH	MRH
MOT032 =	TSUJI	8000	1	200	100	50	500	0	0x003	BNT	BNT
MOT033 =	TSUJI	2000	1	5000	200	2	125	0	0x003	yp	yp
MOT034 =	TSUJI	2000	1	5000	200	2	125	0	0x003	zp	zp
MOT035 =	TSUJI	595	1	1000	200	2	125	0	0x003	rzp	rzp
MOT036 =	TSUJI	144000	1	10000	200	500	500	0	0x003	m36	m36
MOT037 =	TSUJI	2000	1	5000	200	1	125	0	0x003	m37	m37
MOT038 =	TSUJI	500	1	500	50	50	125	0	0x003	pol	Polarimator
MOT039 =	TSUJI	144000	1	10000	200	500	500	0	0x003	pth	Pol_omega
# Counter	ctrl	unit	chan	scale	flags	mne	name				
CNT000 =	VSC16	0	0	1e+07	0x001	sec	Seconds				
CNT001 =	VSC16	0	1	1	0x002	mon	Monitor				
CNT002 =	VSC16	0	2	1	0x000	det	Detector				
CNT003 =	VSC16	0	3	1	0x000	c3	Counter 3				
CNT004 =	VSC16	0	4	1	0x000	c4	Counter 4				
CNT005 =	VSC16	0	5	1	0x000	c5	Counter 5				

10 C-PLOT

spec でデータテキングされたものをプロット (あるいは印刷) するには C-PLOT というプログラムを使用します。起動するにはシェルのコマンドラインから cplot と入力します。測定データを C-PLOT に読みこませるには以下のようにします。

```
PLOT-> f1 scans.4
(Using public version)
```

```
This is scans.4, version 4.6, 12/11/95.
For instructions, type "h scans" in cplot.
```

```
Name of file (data)? xray/setup.1      ここでファイル名を入力します。
Using index file which shows 57 scans.
Print contents (NO)?
Normalize points (YES)?
Use #I (re)normalization value (NO)?
Calculate error bars (YES)?
Get MCA data if present (NO)?
Sort and merge points (YES)?
Rescale axis each time (NO)?
Retrieve by scan number (YES)?
Verbose mode (NO)?
Scans/options (1)
? 3      ここでスキャン番号を入力します。
```

```
Scan number 3
Thu Oct 1 00:21:26 1998
Counted to 1 second.
Two Theta H K L Epoch Seconds Counter 3 Counter 4 Counter 5 Counter 6
Counter 7 Monitor Detector
```

```
Column for x (1)?
Column for y (-1)?
Column for monitor normalization (-2)?
Found 51 points.
```

```
Returning 51 points to plot.
(Counts per monitor count.)
```

```
PLOT->
```

データをファイルから読み込んだら、フィルタを設定します。フィルタとはデータを各デバイスに表示させるプログラムで、例えば画面 (X11 デバイス) に表示させるモードにするには

```
PLOT-> zi x11
```

Started graphics filter "x11".

PLOT->

とします。プリントアウトのためには、`zi psfilter filename` としておいて、その後、UNIX のコマンドラインから `lpr filename` として下さい。

最後に各軸の表示範囲を設定して表示させます。各軸の表示範囲を自動で指定するには `np` コマンドを使います。グラフを表示させるには `zz` というコマンドを使用します。

PLOT-> np

PLOT-> zz

もし手動で各軸の表示範囲を指定したければ `ra` コマンドを用います。以下のようになります。

PLOT-> ra

```
xmin (-2.5) = -2          xmax (0.5) = 0
ymin (0) = -1           ymax (47.8824) = 50
```

PLOT->

グラフの装飾に関して、少しだけ言及します。

グラフのタイトル、軸のタイトルを表示するには `tx` コマンドを使います。`tx[リターン]` の後、聞かれたとおりに入力していきます。その装飾の結果は次に `zz` を行ったときに反映されます。

プロットのシンボルをかえるには、`sy` コマンドを用います。`sy L` は実線、`sy 0` は open circle など、いろいろなシンボルが用意されています。

11 マクロ

C-PLOT は非常に強力なマクロ機能を持っています。要するにコマンドラインに流し込むだけなのですが、旨く使うと、ピークの温度変化を測定したとき、大量のデータを一度にフィッティングして、そのパラメタの温度変化のグラフを出力するようなマクロを作ることができます。あるいは、測定したデータを一枚の紙に 8 つずつ、コメントつきで出力する、というものも私 (若林) はよく使っています。ファイルの中身をコマンドラインに流し込むだけ、という仕様のため、空行に意味がある場合があります。マクロを見る場合には、それも意識する必要があります。

実際のところ、C-PLOT のマクロは汎用性を持たせるのが極めて困難であり、ちょこちょこ書き換えながら使うものです (どのファイルの何番目のスキャンを、何列目のデータを `x`, `y`, モニタの値として使って、軸のタイトルはこれこれ、コメントとしてこれを書き込み、この関数でフィッティング (初期値はこれこれのファイルを参照) しなさい、などという一続きのマクロを作ったら、引き数だらけで何のことかさっぱりわからないものになり、さらには便利でもないでしょう)。

マクロの例 (と、簡単な解説) を以下に添付します。これらのファイルは、`huber2` の `/home/wakabays/` の下の `plotting_macros/`, `fitting_macros/` に置いています。フィッティングのためには、自分のホームディレクトリの下に `functions` というディレクトリを作って、そこに `/home/wakabays/functions` の中身をコピーする必要があります。

```
.....:
alignfig.cpm
```

.....

re
zqw
ze

tu 1

```
*****
# onefig: 1.datafile 2.Num 3.x 4.xunit 5.title 6.pos_x 7.pos_y 8.x_col
do onefig.cpm ../113/990529/990530.dat 20 h rlu 10K(200) -9 -2 1
do onefig.cpm ../113/990529/990530.dat 31 h rlu (300) -9 -8 1
do onefig.cpm ../113/990529/990530.dat 36 h rlu (100) -9 -14 1
#do onefig.cpm ../113/990529/990530.dat 43 h rlu (110) -9 -20 1
#
do onefig.cpm ../113/990529/990530.dat 46 h rlu 170K -.5 -2 1
do onefig.cpm ../113/990529/990530.dat 47 h rlu 170K -.5 -8 1
do onefig.cpm ../113/990529/990530.dat 45 h rlu 220K -.5 -14 1
#do onefig.cpm ../113/990529/990530.dat 44 h rlu 220K_phi=90 -.5 -20 1
#ray 0 100000
#zaltpb
*****
#
zx
```

.....

onefig.cpm

.....

```
ft 2                フォントの設定
cs                  フォントの大きさの設定
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0
```

eb 1 エラーバー表示

```
f1 scans.4 -f $1 +n x=$8 y=-1 m=-2 $2   データ読み込み
#f1 scans.4 -f $1 -n x=1 y=-1 $2
```

```
wi $6 $7 6 4        図の位置, 大きさの設定
np                  図の表示範囲を自動設定
ty 0 0 0            軸の取り方の設定。対数グラフは ty 0 8 0 .
#rax 66 70
#ray 0 4000
```

```
tx                  タイトル, 軸のラベルなどの定義
$1                  図のタイトル。$1,$2... は引き数
$3                  x 軸のラベル
$4                  x 軸の単位
Intensity           y 軸のラベル
arb. units          y 軸の単位 tx は, その下 5 行の内容を使う
```

zn 0.5 0.5 コメント書き込み。引き数は位置。

\$2

\$5

~D

sy 0

zaltpb

```
zn 終わり
プロットのシンボル設定
図の描画
```



```

cs
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0

eb 1

u rm par_all.tmp
u cat Par.*>par_all.tmp
f4 fitpar.4 par_all.tmp 5 2

wi -9 -2 6 4
np
ty 0 0 0
tx

Temperature
K
Intensity
arb. units

sy 0
zaltpb

#Width
ft 2
cs
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0

eb 1

f4 fitpar.4 par_all.tmp 5 4

wi -9 -8 6 4
np
ty 0 0 0
tx

Temperature
K
FWHM
rlu

sy 0
#ray 0 .01
zaltpb

#Peak position
ft 2
cs
2.25 2 0
2.25 2 0

```

```

2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0

eb 1

f4 fitpar.4 par_all.tmp 5 3

wi -9 -14 6 4
np
ty 0 0 0
tx

Temperature
K
Peak center
rlu

sy 0
#ray 1.25 1.26
zaltpb

#Back Ground
ft 2
cs
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0

eb 1

f4 fitpar.4 par_all.tmp 5 0

wi -.5 -2 6 4
np
ty 0 0 0
tx

Temperature
K
Back Ground
arb. units

sy 0
zaltpb

#Back Ground(slope)
ft 2
cs
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0
2.25 2 0

```

```
eb 1

f4 fitpar.4 par_all.tmp 5 1

wi -.5 -8 6 4
np
ty 0 0 0
tx

Temperature
K
Back(slope)
arb. units

sy 0
zaltpb

#end
zx
}
```