

Day 2 X光散射儀控軟體介紹

SPEC fourc
speaker : 楊松

<https://certif.com/content/spec/>

Certified Scientific Software

[Welcome](#) [SPEC](#) [C-PLOT](#) [Users](#) [Support](#) [Contact](#)

A horizontal banner with a blue-to-green gradient background. On the left, the word 'spec' is written in white. Below it, 'Software for Diffraction' is written in a smaller, green font. The right side of the banner features an abstract, glowing green and blue pattern resembling light trails or particle tracks.

spec

Software for Diffraction

spec is a UNIX-based software package for instrument control and data acquisition widely used for X-ray diffraction at synchrotrons around the world and in university, national and industrial laboratories.

Manual Doownload

https://certif.com/spec_manual/idx.html

OUTLINE

- 查詢資訊
- 馬達控制
- CSS SPEC FourC 倒晶格空間系統
 - Sector 設定說明
 - Mode 設定說明
- 尋找繞射點
- Macro 撰寫
- FourC 檔案內容格式說明

SPEC fourc 啟始畫面

```

Welcome to "spec" Release 6.10.02
Copyright (c) 1987-2021 Certified Scientific Software
All rights reserved
[2021-06-26-102343]

(Portions derived from a program developed at Harvard University.)
(Linked with BSD libedit library for command line editing.)

Using "/usr/local/lib/spec.d" for auxiliary file directory (SPEC.D).
Restored state from "SPEC.D/fourc/userfiles/root_ttyp#S".

Getting configuration parameters from "SPEC.D/fourc/config".

Using four-circle configuration.

=
Type h changes for info on latest changes.
Type mstartup for menu-based general setup.
Use setshow, setplot, counters for menu-based configuration.
Browse to http://www.certif.com for complete documentation.
=

298.FOURC> 
```

左圖為使用 fourc 指令啟動 CSS SPEC 系統的 Four Circle Geometry 設定時的畫面。該系統為傳統的文字互動介面，使用者需鍵入已定義好的指令或程式碼來控制繞射儀進行實驗。

該系統的指令語法對有撰寫 C 或 Java 或 javascript 語言的人而言，應容易上手。

CSS SPEC – command

指令語法簡易運算

```
> print 100+200
```

下指令時語法注意事項

- 沒有使用過的變數內容預設為 0
- 大小寫有差
- 使用自定變數前，確認是否已被系統使用
- 使用算式在指令內時，算式寫法不能有空白(SPACE)

查詢指令

wh: 顯示主要馬達位置 **tth**, **th**, **chi**, **phi**。

ct: 顯示目前各個偵檢器讀值，預設積分時間一秒。

wm: 查詢指定馬達目前位置與上下限。

pa: 查詢倒空間相關設定訊息。

sectors : 計算指定 (**H K L**) 位置所有 **sector** 的馬達角度。

SPEC Command : wh

顯示目前馬達的位置

倒晶格空間位置。需完成設定
orientation matrix 參數。

H K L = 0 0 0

Alpha = 0 Beta = 0 Azimuth = 90
Omega = 0 Lambda = 1.033

tth	th	chi	phi
0.0000	0.0000	0.0000	0.0000

當下馬達角度

93.FOURC> █

SPEC Command : ct

顯示偵檢器量測值

下圖為指令執行的結果。系統除了顯示日期時間外，還有三個基本項目：Second, Monitor, Detector。在 TPS09A 實驗站上裝有各式功能的偵測器，所以會顯示除了三個基本項目之外的其他偵檢器的量測值。

```
289.FOURC> ct
Tue Feb 15 11:48:30 2022
Seconds = 1
Monitor = 0 (0/s)
Detector = 0 (0/s)
290.FOURC> 
```

執行指令

量測時的日期時間

積分時間(單位秒)

光源強度 = 測得強度 (每秒測得強度)

訊號強度 = 測得強度 (每秒測得強度)

在取數據的時候會把 Detector 值除以 Monitor 值作為 Intensity 以去除光源強度隨時間變化的因素。

SPEC command : wm

```
84.FOURC> wm phi
```

執行指令

```
Phi  
phi
```

```
User  
High      180.0000  
Current   -90.0010  
Low       -180.0000  
Dial  
High      180.0000  
Current   -90.0010  
Low       -180.0000
```

High 為上限, Low 為下限。這些為“軟體上下限”, 繞射儀系統硬體通常會再安裝一組“硬體上下限”。

在執行移動或 scan 馬達的指令時, 馬達移動前系統會進行“軟體上下限”檢查。檢查不過, 系統會顯示警告訊息並停止執行指令。

馬達移動中觸發硬體上下限, 馬達驅動器會在電路上強制停止馬達動作。此時系統會顯示有關“**hard limit**”的訊息

```
85.FOURC> umv phi 181
```

```
Move to 181 exceeds high limit of 180 on motor 3, unit 0, "phi".  
Motion canceled.
```

```
86.FOURC> dscan phi -200 200 100 1
```

```
Phi will hit low limit at -180 (dial).
```

馬達控制 移動與 scan

CSS SPEC : 角度(馬達)控制指令

Motor Moving Command :

`umv` <馬達名稱> <絕對位置>

`umvr` <馬達名稱> <相對位置>

Example:

`umv tth 25` ← 移動馬達 `tth` 到 `25` 度

`umvr tth 0.2` ← 移動馬達 `tth` 從目前的位置移動 `+0.2` 度

CSS SPEC : 角度(馬達)控制指令

Motor Scan Command :

`ascan` <馬達名稱> <起點 (絕對位置)> <終點 (絕對位置)> <等份點數> <積分時間 (每個點幾秒)>

`dscan` <馬達名稱> <起點 (相對位置)> <終點 (相對位置)> <等份點數> <積分時間 (每個點幾秒)>

`a2scan` <馬達名稱> <起點 (絕對位置)> <終點 (絕對位置)>

<馬達名稱> <起點 (絕對位置)> <終點 (絕對位置)> <等份點數> <積分時間 (每個點幾秒)>

`d2scan` <馬達名稱> <起點 (相對位置)> <終點 (相對位置)>

<馬達名稱> <起點 (相對位置)> <終點 (相對位置)> <等份點數> <積分時間 (每個點幾秒)>

!!! 相對位置的 scan 在結束後會回到 "開始 scan 前的位置"

!!! 絕對位置的 scan 在結束後會停留在 "結束的位置"

馬達二維 scan

Motor Scan Command :

mesh <馬達名稱> <起點 (絕對位置)> <終點 (絕對位置)> <等份點數>

<馬達名稱> <起點 (絕對位置)> <終點 (絕對位置)> <等份點數> <積分時間 (每個點幾秒)>

CSS SPEC : 角度(馬達)控制指令

Example:

ascan tth 20 25 50 1 ← 結束後 tth 停在 25 度

dscan tth -2.5 2.5 50 1 ← 結束後 tth 回到scan 前的位置。

a2scan tth 10 45 th 5 22.5 3000 1

← 結束後 tth 停在 45 度, th 停在 22.5 度

d2scan tth -1 1 th -0.5 0.5 50 1

← 結束後 tth,th 回到 scan 前的位置。

mesh th 12-0.1 12+0.1 20 chi -90-1 -90+1 20 1

← 結束後不會回到起始點。調整 chi 去掃 th 。

scan 的其他要點

1. scan 結束後，系統會儘可能的自動 fitting 出該 scan 的半高寬值和中心位置。
2. fitting 出來的中心位置會存放於系統變數 CEN 之中。

衰減片系統: TPS09A 特有功能

Command :

att <第幾片衰減片> ;

wp ; 加一片衰減片

wn ; 減一片衰減片

attDisk ; 顯示目標衰減片倍率設定

attDisk <第幾片衰減片> <相對衰減倍率> ; 設定指定衰減片倍率

auto_attn_on ; 開啟自動調整衰減片功能

auto_attn_on1; 開啟自動調整衰減片功能, count 前會先多加一片衰減片

auto_attn_off ; 關閉自動調整衰減片功能

auto_attn_ifno ; 顯示自動衰減片功能參數值

自動衰減片功能: 倍率

```
33.FOURC> attDisk
FileName : ~/macros/attDiskRatio/12keV.Mo2
Attn      Real Ratio      Relative Ratio
-----
Attn[0] 1.000e+00      1.000
Attn[1] 1.781e+01      17.814
Attn[2] 3.111e+02      17.465
Attn[3] 5.947e+03      19.116
Attn[4] 1.140e+05      19.172
Attn[5] 1.975e+06      17.318
Attn[6] 1.779e+07      9.011
Attn[7] 1.797e+08      10.100
Attn[8] 1.797e+09      10.000
Attn[9] 1.797e+10      10.000
Attn[10] 1.797e+11      10.000
Attn[11] 1.797e+12      10.000
```

count = **det** * Attn[cnt_att]

```
36.FOURC> ct
Wed Aug 10 09:01:15 2022

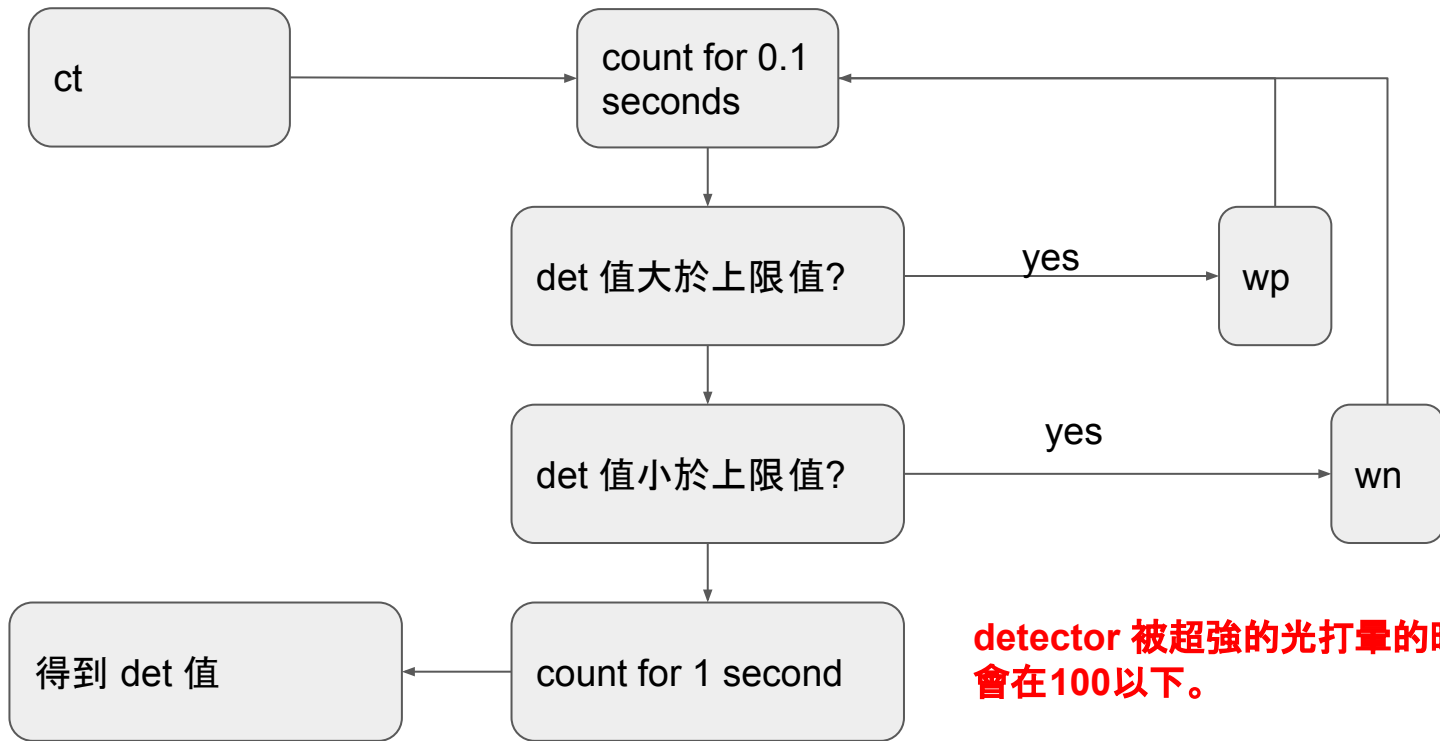
Second = 1
Monitor = 28 (28/s)
Detector = 1 (1/s)
IC1 = 344 (344/s)
IC2 = 793 (793/s)
IC3 = 97 (97/s)
Igate = 0 (0/s)
GAP1 = 0 (0/s)
CURRENT = 0 (0/s)
c400_t = 0 (0/s)
C400B = 0 (0/s)
C400A = 0 (0/s)
cnt_att = 7 (7/s)
count = 1.79714e+08 (1.79714e+08/s)
TempA = 0 (0/s)
TempB = 0 (0/s)
```

自動衰減片功能: 參數

```
37.FOURC> auto_attn_info  
=====
```

AUTO ATTN	:	DISABLE
AUTO ATTN MAX:		6
PROTECT PLUS	:	0
DET UPPER LIM:		23000
DET LOWER LIM:		1000

自動衰減片功能: 自動調整機制



detector 被超強的光打暈的時候, 量測到的 det 值約會在 100 以下。

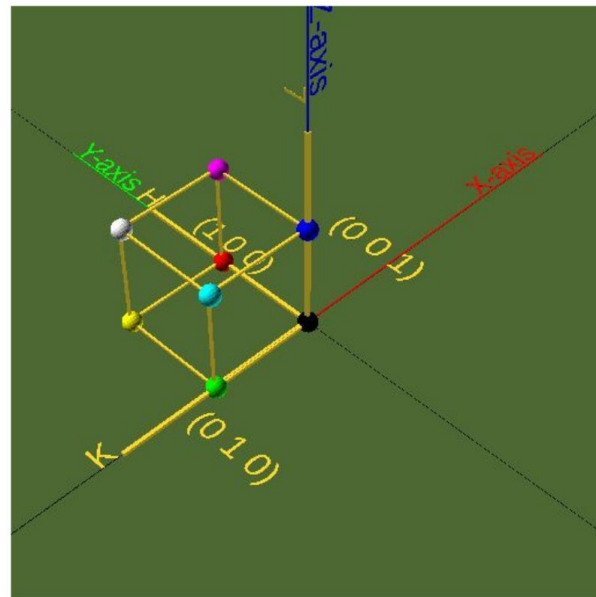
CSS SPEC 倒空間系統

在 CSS SPEC 系統中顯示倒空間座標是以 Crystal Frame 上的 G vector : (H K L) 形式呈現。

CSS SPEC 系統內會利用一個固定的轉換矩陣 B matrix 將晶體的 G vector 從 Crystal Frame (晶體座標軸) 轉換為 Lab. Frame (實驗室座標), 再計算出對應的四個角度位置 (tth, th, chi, phi) 。或是從指定的角度位置算出對應的 G vector。

CSS SPEC 系統會依據當下的三個主要設定 值(**晶格常數、主要 G vector 馬達位置、次要 G vector 馬達位置**) 自動計算 B matrix。

簡單的說, 預使用 CSS SPEC 倒空間計算功能的話就需要先設定好三個主要設定。



Orientation Matrix

$$\Theta X \Phi B \begin{pmatrix} H \\ K \\ L \end{pmatrix}_c = q = k_f - k_i = k \left(R_{tth} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right)$$

Transformation Matrix : B

$$\begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = B \begin{pmatrix} H \\ K \\ L \end{pmatrix}$$

Orientation Matrix 參數設定與倒空間相關指令

Information Command :

pa, sectors

Configure Command :

setlat, or0, or1, set_or0, set_ro1

Calculation Command :

ca, ci, setmode, setsector, freeze

Moving Command :

ubr

Scan Command :

hscan, kscan, lscan, hklscan, hklmesh

pa : 顯示倒空間參數設定

顯示倒空間設定資訊 – 指令 pa

```
294.FOURC> pa

Four-Circle Geometry, Omega equals zero (mode 0)
Sector 0

Primary Reflection (at lambda 1.033):
    tth th chi phi = 21.9335 10.967 -90 0
      H K L = 0 0 2

Secondary Reflection (at lambda 1.033):
    tth th chi phi = 15.4615 7.731 -45 45.5
      H K L = 1 0 1

Lattice Constants (lengths / angles):
    real space = 5.43 5.43 5.43 / 90 90 90
    reciprocal space = 1.157 1.157 1.157 / 90 90 90

Azimuthal Reference:
      H K L = 0 0 1

      Lambda = 1.033

Cut Points:
    tth th chi phi
    -180 -180 -180 -180

295.FOURC> █
```

pa 指令執行後，會顯示目前有關倒空間相關設定值。如左圖紅色線框起來的部份，有五個部份的設定值。

顯示倒空間設定資訊 – 指令 pa

```
294.FOURC> pa
```

```
Four-Circle Geometry, Omega equals zero (mode 0)  
Sector 0
```

```
Primary Reflection (at lambda 1.033):  
  tth th chi phi = 21.9335 10.967 -90 0  
    H K L = 0 0 2
```

```
Secondary Reflection (at lambda 1.033):  
  tth th chi phi = 15.4615 7.731 -45 45.5  
    H K L = 1 0 1
```

```
Lattice Constants (lengths / angles):  
  real space = 5.43 5.43 5.43 / 90 90 90  
  reciprocal space = 1.157 1.157 1.157 / 90 90 90
```

```
Azimuthal Reference:  
  H K L = 0 0 1
```

```
Lambda = 1.033
```

```
Cut Points:  
  tth th chi phi  
 -180 -180 -180 -180
```

```
295.FOURC> █
```

pa 指令執行後，會顯示目前有關倒空間相關設定值。如左圖紅色線框起來的部份，有五個部份的設定值。

mode & sector, 對應指令: **setmode**, **setsector**

主要與次要 G vector 的對應角度位置, 對應指令: **or0**, **or1**

lattice constants 設定, 對應指令: **setlat**

光源波長, 管理員會設定

計算角度時範圍邊界設定, 對應指令: **cuts**
通常只會設定 phi

倒空間設定項目與對應指令

1. Lattice constants : **setlat**
2. Primary reflection 0 : **or0**
3. Secondary reflection 1 : **or1**
4. Mode : **setmode**
5. Sector : **sectors, setsector**
6. 存檔與讀檔 : **save, do**

CSS SPEC 系統在使用倒空間之前所需完成的三個主要設定：**晶格常數、主要 G vector 馬達位置、次要 G vector 馬達位置**。在三個主要設定完成之後，可以透過設定 **mode** 和 **sector** 來設定計算模式與計算象限。

選用的主要 G vector 和 次要 G vector 不可以互相平行。

晶格常數使用指令 **setlat** 設定

主要 G vector 的馬達位置使用指令 **or0** 設定

次要 G vector 的馬達位置使用指令 **or1** 設定

pa 主要設定

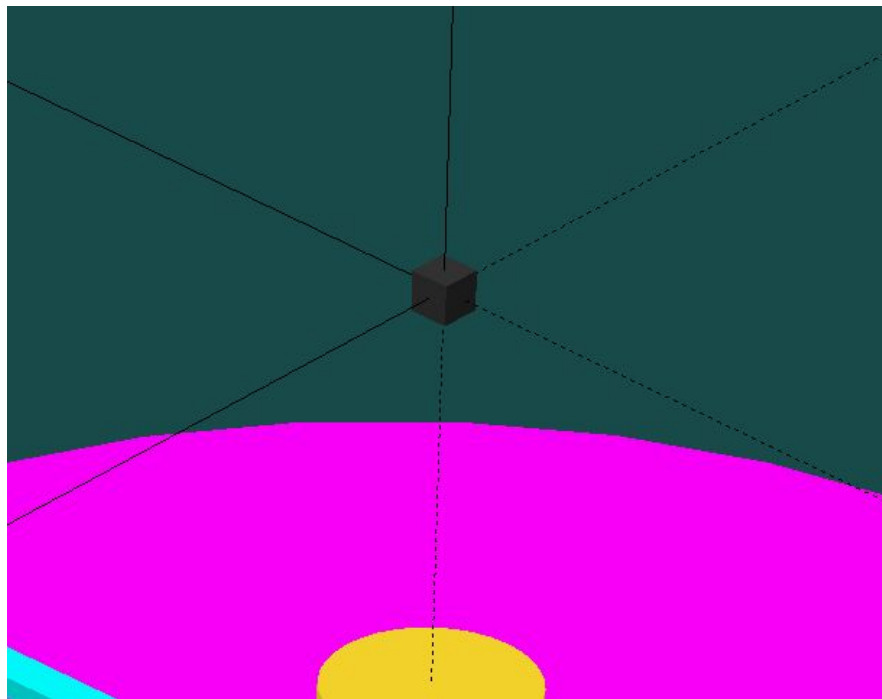
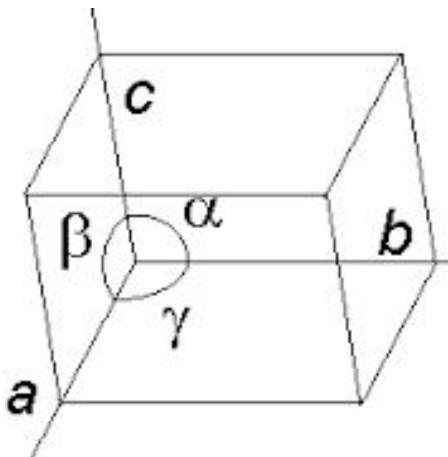
setlat：設定晶格常數

開始倒空間實驗之前的第一件事

Example : Si Crystal (晶格常數)

以 Si 的晶格常數為例進行設定說明。
下面顯示 Si 的 lattice constants 值。

$$a = b = c = 5.43 \text{ \AA}$$
$$\alpha = \beta = \gamma = 90^\circ$$



設定晶格常數 – 指令 **setlat**

下面的例子為設定 Si 的晶格常數：

指令執行

```
295.FOURC> setlat
```

```
Enter real space lattice parameters:
```

```
Lattice a (5.43)? 5.43
```

```
Lattice b (5.43)? 5.43
```

```
Lattice c (5.43)? 5.43
```

```
Lattice alpha (90)? 90
```

```
Lattice beta (90)? 90
```

```
Lattice gamma (90)? 90
```

```
(UB recalculated from orientation reflections and lattice.)
```

依序輸入 $a, b, c, \alpha, \beta, \gamma$ 的值。

設定晶格常數 – 指令 **setlat**

執行 pa 確認

```
296.FOURC> pa

Four-Circle Geometry, Omega equals zero (mode 0)
Sector 0

Primary Reflection (at lambda 1.033):
  tth th chi phi = 21.9335 10.967 -90 0
    H K L = 0 0 2

Secondary Reflection (at lambda 1.033):
  tth th chi phi = 15.4615 7.731 -45 45.5
    H K L = 1 0 1

Lattice Constants (lengths / angles):
  real space = 5.43 5.43 5.43 / 90 90 90
  reciprocal space = 1.157 1.157 1.157 / 90 90 90

Azimuthal Reference:
  H K L = 0 0 1

Lambda = 1.033

Cut Points:
  tth th chi phi
-180 -180 -180 -180
```

上一頁輸入的 lattice constants 值會顯示在這區塊。

pa 主要設定

or0 :

設定主要 G vector 的角度位置

範例設定 Orientation Matrix – 指令 or0

假設已經透過一系列的角度 scan 後，找到並位於 Si(002) 繞射點。
使用 **or0** 指令，將當下的四個角度位置和 **(0 0 2)** 進行綁定。

```
152.FOURC> umv tth 21.9335 th 10.9670 chi -90.0000
```

```
153.FOURC> wh
```

```
H K L = 0 0 2
```

```
Alpha = 10.967 Beta = 10.967 Azimuth = 90
```

```
Omega = 0 Lambda = 1.033
```

tth	th	chi	phi
21.9335	10.9670	-90.0000	10.0000

找到 Si(002)

```
154.FOURC> or0 0 0 2
```

```
(UB recalculated from orientation reflections and lattice.)
```

```
155.FOURC> pa
```

設定此處為 Si(002)，定為 **or0**

完成 or0 指令設定之後，用指令 pa 確認。

```
155.FOURC> pa
Four-Circle Geometry, Omega equals zero (mode 0)
Sector 0

Primary Reflection (at lambda 1.033):
  tth th chi phi = 21.9335 10.967 -90 10
    H K L = 0 0 2

Secondary Reflection (at lambda 1.033):
  tth th chi phi = 16.709 8.3545 -40.8935 10
    H K L = 1 0 1

Lattice Constants (lengths / angles):
  real space = 5.43 5.43 5.43 / 90 90 90
  reciprocal space = 1.157 1.157 1.157 / 90 90 90

Azimuthal Reference:
  H K L = 0 0 1

  Lambda = 1.033

Cut Points:
  tth th chi phi
  -180 -180 -180 -180

156.FOURC>
```

設定此處為 Si (002) , 足為 or0

確認寫入 or0
的設定值

主要設定

or1 :

設定次要 G vector 的角度位置

範例設定 Orientation Matrix – 指令 or1

透過一系列的角度 scan 後, 找到並位於 Si(101) 的繞射點後。使用 or1 指令告知系統該位置為 Si(101)。

```
157.FOURC> umv tth 15.4615 th 7.7310 chl -45.0000 phi 45.5
```

```
158.FOURC> wh
```

```
H K L = 0.8141 -0.5807 0.99999  
Alpha = 5.4581 Beta = 5.4582 Azimuth = 90  
Omega = 0 Lambda = 1.033
```

tth	th	chl	phi
15.4615	7.7310	-45.0000	45.5000

找到 Si(101)

```
159.FOURC> or1 1 0 1  
(UB recalculated from orientation reflections and lattice.)
```

```
160.FOURC> pa
```

設定此處為 Si(101), 定為 or1

完成 or0 指令設定之後, 用指令 pa 確認。

```
160.FOURC> pa
Four-Circle Geometry, Omega equals zero (mode 0)
Sector 0

Primary Reflection (at lambda 1.033):
  tth th chi phi = 21.9335 10.967 -90 10
    H K L = 0 0 2

Secondary Reflection (at lambda 1.033):
  tth th chi phi = 15.4615 7.731 -45 45.5
    H K L = 1 0 1

Lattice Constants (lengths / angles):
  real space = 5.43 5.43 5.43 / 90 90 90
  reciprocal space = 1.157 1.157 1.157 / 90 90 90

Azimuthal Reference:
  H K L = 0 0 1

  Lambda = 1.033

Cut Points:
  tth th chi phi
  -180 -180 -180 -180

61.FOURC>
```

設定此處為 Si(101), 定為 or1

確認寫入 or1
的設定值

```
157.FOURC> umv tth 15.4615 th 7.7310 chi -45.0000 phi 45.5
158.FOURC> wh
H K L = 0.8141 -0.5807 0.99999
Alpha = 5.4581 Beta = 5.4582 Azimuth = 90
Omega = 0 Lambda = 1.033

  tth th chi phi
  15.4615 7.7310 -45.0000 45.5000

159.FOURC> or1 1 0 1
(UB recalculated from orientation reflections and lattice.)
160.FOURC> pa
```

設定此處為 Si(1

pa 系統存檔/讀檔 (**save** and **do** file)

```
277.FOURC> save ST0.pa  
Type "do ST0.pa" to recover.
```

存檔

```
278.FOURC>
```

```
278.FOURC> do ST0.pa
```

讀檔

```
FOURC.1> Reading file "/home/admin/Data/S.Yang/ST0.pa".
```

```
FOURC.1> H=0
```

計算倒晶格空間位置對應角度 － 指令 `ca`

使用前先設定 `mode` 和 `sector`

```
29.FOURC> setmode 0  
Still using Omega equals zero mode.
```

```
30.FOURC> unfreeze  
Frozen mode already off.
```

設定 mode

```
31.FOURC> setsector 4  
Still using ...
```

設定 sector

```
Omega -> -Omega  
Chi -> 180 - Chi  
Phi -> Phi - 180
```

```
32.FOURC> ca 0 2 2
```

計算馬達位置

Calculated Positions:

```
H K L = 1.5231e-10 2 2  
Alpha = 17.939 Beta = 17.939 Azimuth = 90  
Omega = 0 Lambda = 1.54
```

tth	th	chi	phi
51.6440	25.8220	-45.0005	-90.0005

```
33.FOURC> █
```

參數設定：mode

倒空間位置對應馬達位置計算模式設定。

mode 0 (ω equals 0)

mode 3 (ϕ fixed)

mode 0

omega equals 0

tth/th = 2/1

在 off-normal 區域通常使用 mode 0

Mode : omega equal zero

```
184.FOURC> setmode
```

```
Choose a four-circle geometry mode:
```

- 0 Omega equals zero
- 1 Omega fixed
- 2 Zone (Chi and Phi fixed)
- 3 Phi fixed (Three-circle)
- 4 Azimuth fixed
- 5 Alpha fixed
- 6 Beta fixed
- 7 Theta fixed

```
Your choice (0)? 0
```

```
Still using Omega equals zero mode.
```

```
185.FOURC> unfreeze  
Frozen mode already off.
```

執行 setmode 指令後顯示 7 種角度計算模式。基本常用的兩種模式為：

0 Omega equals zero

3 Phi fixed (Three-circle)

而 Omega equals zero 模式即為之前提過的 tth:th 維持在 2:1 的模式。所以下面的互動式問答則輸入“0”。

打算使用 Omega equals zero

輸入 0

如果前一個模式是 Phi fixed 模式，記得解凍 phi 馬達。

Mode : omega equal zero

```
216.FOURC> ca 1 1 1
```

```
Calculated Positions:
```

```
H K L = 1 0.99998 0.99999
```

```
Alpha = 5.4581 Beta = 5.4583 Azimuth = 90
```

```
Omega = 0 Lambda = 1.033
```

tth	th	chi	phi
18.9655	9.4830	-35.2645	0.5005

```
216.FOURC> ca 0 1 2
```

```
Calculated Positions:
```

```
H K L = -1.0367e-05 1 2
```

```
Alpha = 10.966 Beta = 10.967 Azimuth = 89.999
```

```
Omega = 0 Lambda = 1.033
```

tth	th	chi	phi
24.5605	12.2800	-63.4350	-44.4995

```
217.FOURC> █
```

在該模式下, 用 ca 計算指令繞射點的馬達位置, 會維持在 $tth:th = 2:1$ 。

在 off-normal 區域使用倒空間相關指令的時候, 如無特殊需求該模式為常用的模式。

mode 設定 : 3

phi fixed

在 surface-normal 區域通常使用 mode 3

Mode : phi fixed

```
217.FOURC> setmode
```

```
Choose a four-circle geometry mode:
```

- 0 Omega equals zero
- 1 Omega fixed
- 2 Zone (Chi and Phi fixed)
- 3 Phi fixed (Three-circle)
- 4 Azimuth fixed
- 5 Alpha fixed
- 6 Beta fixed
- 7 Theta fixed

```
Your choice (0)? 3
```

```
Now using Phi fixed (Three-circle) mode
```

```
218.FOURC> freeze
```

```
Tue Feb 08 00:27:15 2022. Freezing Phi at 0.
```

tth:th 不固定在 2:1, **phi 固定**的模式

將 phi 馬達固定(凍)在目前的位置

Mode : phi fixed

ca 0 0 2 (phi-fixed mode)

```
H K L = 0 0 2
Alpha = 10.967 Beta = 10.967 Azimuth = 90
Omega = 0.00025 Lambda = 1.033

      tth      th      chi      phi
      21.9335   10.9670  -90.0000   0.0000

225.FOURC> ca 0.001 0 2

Calculated Positions:

H K L = 0.00099972 -3.6143e-06 2
Alpha = 10.987 Beta = 10.946 Azimuth = 135.71
Omega = 0.020683 Lambda = 1.033

      tth      th      chi      phi
      21.9335   10.9875  -89.9800   0.0000

226.FOURC> ca 0 0.001 2

Calculated Positions:

H K L = 3.6143e-06 0.00099972 2
Alpha = 10.947 Beta = 10.987 Azimuth = 45.707
Omega = -0.01983 Lambda = 1.033

      tth      th      chi      phi
      21.9335   10.9470  -89.9795   0.0000
```

同樣計算 surface-normal
附近的繞射點馬達位置。

比較左圖(phi fixed)與右圖
(omega equals zero) 的計
算結果。

phi-fixed mode 適用於在
surface-normal 上的繞射
點計算與量測: ubr, hscan,
kscan, lscan, hklscan,
hklmesh...

ca 0 0 2 (Omega equals zero mode)

```
230.FOURC> ca 0 0 2

Calculated Positions:

H K L = 0 0 2
Alpha = 10.967 Beta = 10.967 Azimuth = 90
Omega = 0 Lambda = 1.033

      tth      th      chi      phi
      21.9335   10.9670  -90.0000   0.0000

231.FOURC> ca 0.001 0 2

Calculated Positions:

H K L = 0.001006 -8.764e-06 2
Alpha = 10.967 Beta = 10.967 Azimuth = 90.151
Omega = 0 Lambda = 1.033

      tth      th      chi      phi
      21.9335   10.9670  -89.9710  45.8485

232.FOURC> ca 0 0.001 2

Calculated Positions:

H K L = 8.7324e-06 0.0010009 2
Alpha = 10.967 Beta = 10.967 Azimuth = 90.141
Omega = 0 Lambda = 1.033

      tth      th      chi      phi
      21.9335   10.9670  -89.9715 -44.1410
```

sector 設定

0 or 4 : 選擇適當 chi 角度

Mode : sector

```
247.FOURC> ca -2 0 2
```

```
Calculated Positions:
```

```
H K L = -2 -1.0821e-07 2
```

```
Alpha = 10.966 Beta = 10.967 Azimuth = -90
```

```
Omega = 0 Lambda = 1.033
```

tth	th	chi	phi
31.2140	15.6070	-135.0000	45.5000

```
248.FOURC> █
```

因為 TPS09A 繞射儀的硬體設制, chi 的移動範圍為 $-90^{\circ}\sim 0^{\circ}$ 。但是有時候會發生如左圖, 計算繞射點的 chi 角度位置超出移動範圍。

並非計算錯誤, 只是這個象限 (**sector**) 不適用於 TPS09A 繞射儀的設定。

此時可以更改象限 (**sector**) 設定後再進行計算。(next page)

```
249.FOURC> pa
```

```
Four-Circle Geometry, Omega equals zero (mode 0)  
Sector 0
```

```
Primary Reflection (at lambda 1.033):
```

```
tth th chi phi = 21.9335 10.967 -90 0
```

```
H K L = 0 0 2
```

```
Secondary Reflection (at lambda 1.033):
```


Mode : sector

```
250.FOURC> sectors -2 0 2
# Alpha Beta Azimuth Omega tth th chi phi
0 10.966 10.967 -90 0 31.214 15.607 -135 45.5
1 10.966 10.967 -90 0 31.214 15.607 -135 45.5
2 10.966 10.967 -90 0 -31.214 -15.607 45 45.5
3 10.966 10.967 -90 0 -31.214 -15.607 45 45.5
4 10.967 10.966 90 0 31.214 15.607 -45 -134.5
5 10.967 10.966 90 0 31.214 15.607 -45 -134.5
6 10.967 10.966 90 0 -31.214 -15.607 135 -134.5
7 10.967 10.966 90 0 -31.214 -15.607 135 -134.5
8 10.967 10.966 90 180 31.214 -164.393 45 45.5

251.FOURC> setsector 4
Now using ...

Omega -> -Omega
Chi -> 180 - Chi
Phi -> Phi - 180

252.FOURC> ca -2 0 2

Calculated Positions:

H K L = -2 -1.0821e-07 2
Alpha = 10.967 Beta = 10.966 Azimuth = 90
Omega = 0 Lambda = 1.033

tth th chi phi
31.2140 15.6070 -45.0000 -134.5000

253.FOURC>
```

如左圖。使用指令 sectors H K L 可列出指定繞射面所有象限的角度計算結果。依照 TPS09A 的硬體設制，通常適用 sector 設定為 0 或 4。

可看到 sector 4 的計算 chi 值在 -90~0 之間。

使用指令 “setsector #” 將 sector 設為 4，再使用 ca 指令確認計算結果。

ubr, hscan, kscan, lscan, hklscan, hklmesh 的馬達計算等同於 ca 的計算結果。故，在使用這些指令之前先用 ca 指令驗算以防止非預期的馬達動作發生。經常發生在 mode 與 sector 設定到不適合的值時。

倒晶格空間計算：**ca**

倒晶格空間移動：**ubr**

倒晶格空間 scan：**hscan, kscan, lscan, hklscan,**
hklmesh

倒晶格空間移動

進行倒晶格空間操作前確認:

1. **mode** 設定是否正確
2. 馬達設定為 **freeze** 或是 **unfreeze**
3. **sector** 設定是否正確
4. **ca** 驗算倒晶格空間

! **tth:th** 沒有為 **2:1** , 查看 **mode** 和 **freeze**

! **chi** 計算出來的角度超出範圍, 查看 **sector**

再看一次 pa 指令顯示的訊息

```
325.FOURC> pa
```

```
Four-Circle Geometry, Phi fixed (Three-circle) (mode 3)
```

```
Frozen values: Phi = -135
```

```
Sector 0
```

```
Primary Reflection (at lambda 1.033):
```

```
tth th chi phi = 63.8845 31.942 90 0
```

```
H K L = 0 0 4
```

```
Secondary Reflection (at lambda 1.033):
```

```
tth th chi phi = 15.201 7.6005 0 0
```

```
H K L = 1 0 0
```

```
Lattice Constants (lengths / angles):
```

```
real space = 3.905 3.905 3.905 / 90 90 90
```

```
reciprocal space = 1.609 1.609 1.609 / 90 90 90
```

```
Azimuthal Reference:
```

```
H K L = 0 0 1
```

```
Lambda = 1.033
```

```
Cut Points:
```

```
tth th chi phi
```

```
-180 -180 -180 -180
```

mode 設定顯示在這裡

freeze 的設定顯示在這裡, 沒有 **freeze** 則不顯示

sector 設定顯示在這裡

倒晶格空間計算狀況一: tth/th 比例非 2/1

```
268.FOURC> wh  
  
H K L = 6.5356e-06 -6.4224e-06 2.1  
Alpha = 11.523 Beta = 11.522 Azimuth = -180  
Omega = 0.0005 Lambda = 1.033  
  
      tth      th      chi      phi  
23.0450  11.5230 -90.0000  0.0000  
  
269.FOURC> ca 1 0 1  
  
Calculated Positions:  
  
H K L = 0.99999 -1.8565e-06 0.99999  
Alpha = 30.289 Beta = -18.308 Azimuth = 125.74  
Omega = 30.288 Lambda = 1.033  
  
      tth      th      chi      phi  
15.4615  38.0190 -54.9730  0.0000  
  
270.FOURC> █
```

假設目前位置如左圖, 位於 surface normal Si(002) 附近。

計算 Si(101) 後發現馬達位置異常。

倒晶格空間計算狀況一：tth/th 比例非 2/1→

```
270.FOURC> setmode  
  
Choose a four-circle geometry mode:  
  
0 Omega equals zero  
1 Omega fixed  
2 Zone (Chi and Phi fixed)  
3 Phi fixed (Three-circle)  
4 Azimuth fixed  
5 Alpha fixed  
6 Beta fixed  
7 Theta fixed  
  
Your choice (3)? 0  
  
Now using Omega equals zero mode.  
  
271.FOURC> unfreeze  
  
Tue Feb 08 01:56:16 2022. Frozen mode off.  
  
272.FOURC> ca 1 0 1  
  
Calculated Positions:  
  
H K L = 0.99999 3.6147e-06 0.99999  
Alpha = 5.4585 Beta = 5.4578 Azimuth = -90  
Omega = 0 Lambda = 1.033  
  
tth      th      chi      phi  
15.4615  7.7310 -135.0000 -134.4995
```

此時先確認並設定 mode。

設定完之後，再一次 ca 計算。
發現 chi 位置超出範圍。
...(next page)

倒晶格空間計算狀況：chi 超出範圍

```
276.FOURC> sectors 1 0 1
#   Alpha   Beta   Azimuth   Omega   tth   th   chi   phi
0   5.4582   5.4582   90       0     15.4615   7.731   -45   45.5005
1   5.4582   5.4582   90       0     15.4615   7.731   -45   45.5005
2   5.4582   5.4582   90       0    -15.4615  -7.731   135   45.5005
3   5.4582   5.4582   90       0    -15.4615  -7.731   135   45.5005
4   5.4585   5.4578  -90       0     15.4615   7.731  -135  -134.5
5   5.4585   5.4578  -90       0     15.4615   7.731  -135  -134.5
6   5.4585   5.4578  -90       0    -15.4615  -7.731   45   -134.5
7   5.4585   5.4578  -90       0    -15.4615  -7.731   45   -134.5
8   5.4585   5.4578  -90      180     15.4615 -172.269  135   45.5005
```

```
277.FOURC> setsector 0
Still using no sector transformations.
```

```
278.FOURC> ca 1 0 1
```

Calculated Positions:

```
H K L = 0.99999 -8.7265e-06 0.99999
Alpha = 5.4582 Beta = 5.4582 Azimuth = 90
Omega = 0 Lambda = 1.033
```

tth	th	chi	phi
15.4615	7.7310	-45.0000	45.5005

```
279.FOURC> ubr 1 0 1
```

```
280.FOURC> wh
```

```
H K L = 0.99999 -8.7265e-06 0.99999
Alpha = 5.4582 Beta = 5.4582 Azimuth = 90
Omega = 0 Lambda = 1.033
```

tth	th	chi	phi
15.4615	7.7310	-45.0000	45.5005

如左圖，使用指令 `sectors` 列出所有 sector 的計算結果。

使用 `setsector` 指令設定適當值後，使用 `ca` 指令再計算一次。確認計算結果後，方執行倒晶格相關指令 (`ubr`, `hscan`, `hklscan`, `hklmesh`...)

`ubr H K L`：移動到指定的 (H K L) 到晶格空間位置。

倒晶格空間 scan : line scan

hscan	<H起點位置 (絕對)>	<H終點位置 (絕對)>	<等分點數>	<每點積分時間 (秒)>
kscan	<K起點位置 (絕對)>	<K終點位置 (絕對)>	<等分點數>	<每點積分時間 (秒)>
lscan	<L起點位置 (絕對)>	<L終點位置 (絕對)>	<等分點數>	<每點積分時間 (秒)>

!!! 建議掃描前--先用 ca 確認起點和終點的馬達位置計算結果 --。

!!! 如要使用系統預設變數 H, K, L, 請先執行 wh 指令, 該指令會重新計算當下 (tth, th, chi, phi) 位置對應的 (H, K, L) 值

倒晶格空間 scan : line scan

Example: (H K L) 在 (1 0 1)

wh					
ca	1	0	1		
hscan	1-0.001	1+0.001	100	1	
hscan	H-0.001	H+0.001	100	1	
ubr	CEN	K	L		
kscan	-0.001	0.001	100	1	
kscan	K-0.001	K+0.001	100	1	
ubr	H	CEN	L		
lscan	1-0.001	1+0.001	100	1	
lscan	L-0.001	L+0.001	100	1	
ubr	H	K	CEN		

倒晶格空間計算在 0 值的正負交界處常常會出現算出來的 phi 與 chi 位置跑掉。
一個方法是使用 phi fix mode
另一個方法是分兩段不同 sector 分開 scan.

← 通常在這裡會出事

另一個常見錯誤就是 sector 不對。

事先用 ca 驗算可以防止這些問題發生。

倒晶格空間 scan : line scan

從 起點 (H_1 K_1 L_1) 直線 掃到 終點 (H_2 K_2 L_2)

```
hklscan  <H1 起點位置 (絕對)>    <H2 終點位置 (絕對)>  
          <K1 起點位置 (絕對)>    <K2 終點位置 (絕對)>  
          <L1 起點位置 (絕對)>    <L2 終點位置 (絕對)>    <等分點數>    <每點積分時間 (秒)>
```

!!! 建議掃描前--先用 ca 確認起點和終點的馬達位置計算結果 --。

!!! 如要使用系統預設變數 H , K , L , 請先執行 wh 指令, 該指令會重新計算當下 (tth, th, chi, phi) 位置對應的 (H, K, L) 值

倒晶格空間 scan : mesh scan

```
hklmesh <馬達1名稱> <起點位置 (絕對)> <終點位置 (絕對)> <等分點數>  
        <馬達2名稱> <起點位置 (絕對)> <終點位置 (絕對)> <等分點數> <每點積分時間 (秒)>
```

!!! 建議掃描前--先用 ca 確認 "四個端點" 的馬達位置計算結果 --。

!!! 如要使用系統預設變數 H, K, L, 請先執行 wh 指令, 該指令會重新計算當下 (tth, th, chi, phi) 位置對應的 (H, K, L) 值

倒晶格空間 scan : mesh scan

Example: (H K L) 在 (2 0 1) 進行 H-L mesh scan

wh

ca 2+0.001 0 1+0.001

ca 2+0.001 0 1-0.001

ca 2-0.001 0 1+0.001

ca 2-0.001 0 1-0.001

倒晶格空間的 mesh scan 常常會見的狀況是掃描的範圍超出馬達的 limit 值。特別是在 (H K L) 值很大的時候 tth 和 th 值會超出極限設值值。

建議使用 ca 驗算四個端點。

hklmesh H 2+0.001 2-0.001 20 L 1-0.001 1+0.001 20 1

wh

ca H+0.001 0 L+0.001

ca H+0.001 0 L-0.001

ca H-0.001 0 L+0.001

ca H-0.001 0 L-0.001

hklmesh H H+0.001 H-0.001 20 L L-0.001 L+0.001 20 1

倒晶格空間 scan : mesh scan

Example: (H K L) 在 (2 0 1) 進行 H-L mesh scan

```
setsector 0 或 4
```

```
wh
```

```
Hs = H
```

```
Ks = K
```

```
Ls = L
```

```
ca      Hs+0.001      0      Ls+0.001
```

```
ca      Hs+0.001      0      Ls-0.001
```

```
ca      Hs-0.001      0      Ls+0.001
```

```
ca      Hs-0.001      0      Ls-0.001
```

```
hklmesh  H  Hs+0.001  Hs-0.001  20  L  Ls-0.001  Ls+0.001  20  1
```

倒晶格空間 scan : mesh scan

Example: (H K L) 在 (0 2 1) 進行 K-L mesh scan

```
setsector 0 或 4
```

```
wh
```

```
Hs = H
```

```
Ks = K
```

```
Ls = L
```

```
ca      0      Ks-0.001      Ls+0.001
```

```
ca      0      Ks-0.001      Ls-0.001
```

```
ca      0      Ks-0.001      Ls+0.001
```

```
ca      0      Ks-0.001      Ls-0.001
```

```
hklmesh  K  Ks+0.001  Ks-0.001  20  L  Ls-0.001  Ls+0.001  20  1
```

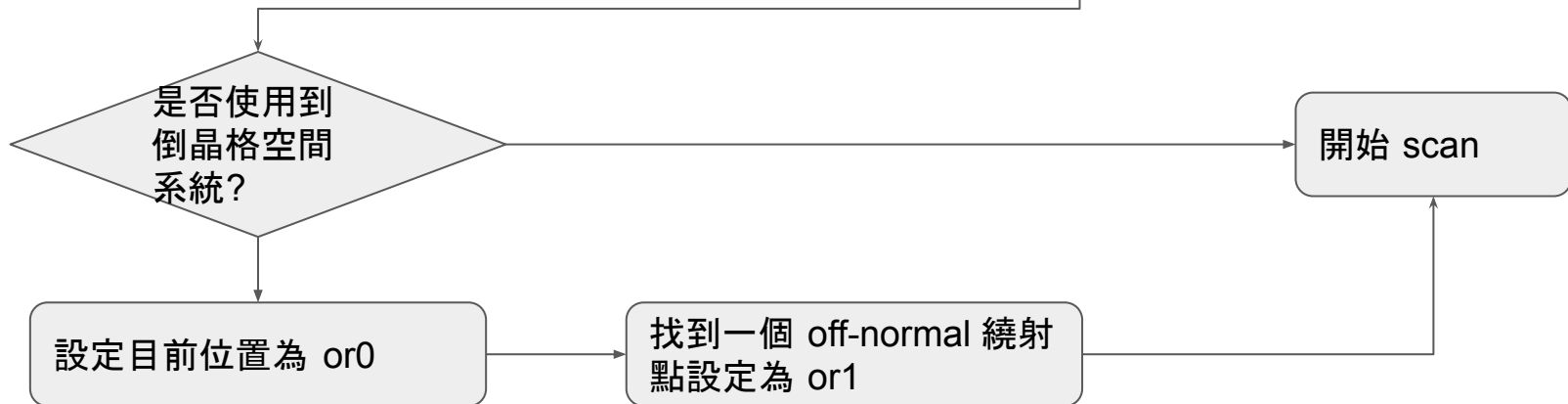
實驗流程

實驗開始:

1. 放置樣品
2. 砍光一半 (ϕ 和 θ)
3. 設定晶格常數 (setlat)

晶格調平:

1. 找到 surface normal 繞射點
2. 分別在 $\phi = 0, 180$ 找點 (tth, θ and χ)
3. 校正 θ 和 χ , 並用 arc 馬達把晶軸修正到平行 ϕ 盤的旋轉軸。
4. 分別在 $\phi = 90, 270$ 找點 (tth, θ and χ)
5. 用 arc 馬達把晶軸修正到平行 ϕ 盤的旋轉軸。



尋找繞射點

surface normal

off-normal

in-plane

How: 尋找繞射點

1. 計算目標繞射點位置 (ca)
2. 移動到可能的位置
3. 微調各個馬達找到繞射強度。(使用 umvr)
4. 重複交錯 scan 各個馬達。直到繞射點 CEN 位置固定。(使用 dscan)
5. wh; ct 記下位置資訊、繞射強度資訊。

EX: 尋找 Surface Normal 繞射點

```
291.FOURC> ca 0 0 2
```

計算馬達位置

```
Calculated Positions:
```

```
H K L = 0 8.7267e-06 2
```

```
Alpha = 11.923 Beta = 11.924 Azimuth = 2.9157e-09
```

```
Omega = 0 Lambda = 1.033
```

Two Theta	Theta	Chi	Phi
23.8465	11.9230	-90.0000	0.0000

```
292.FOURC> umv tth 23.8465 th 11.9230
```

移動到計算出來的馬達位置

EX: 尋找 Surface Normal 繞射點

```
# -- Surface Normal  
ca 0 0 2
```

```
umvr th 0.01 ; ct  
umvr chi 0.01; ct  
umvr tth 0.01; ct  
# --- change att
```

微調 tth, th, chi 找到繞射點的強度並慢慢接近繞射點位置。並適時調整衰减片。

```
dscan th -0.05 0.05 20 1  
umv th CEN
```

反覆 scan tth, th, chi 找到繞射點 peak 位置。

```
dscan tth -0.05 0.05 20 1  
umv tth CEN
```

```
dsca chi -0.1 0.1 20 1  
umv chi CEN
```

EX: 尋找 off-normal 繞射點

```
293.FOURC> ca 2 0 2
```

計算馬達位置

Calculated Positions:

H K L = 2 1.2341e-05 2

Alpha = 11.923 Beta = 11.923 Azimuth = 90

Omega = 0 Lambda = 1.033

Two Theta	Theta	Chi	Phi
33.9765	16.9880	-45.0000	0.0000

```
294.FOURC> umv tth 33.9765 th 16.9880 chi -45
```

移動到計算出來的馬達位置

EX: 尋找 off-normal 繞射點

微調 chi, phi, tth/th 找到繞射點的強度並慢慢接近繞射點位置。並適時調整衰減片。

```
umvr phi 0.01; ct
umvr chi 0.01; ct
umvr tth 0.01; umvr th 0.01/2 ; ct
# --- change att
```

反覆 scan chi, phi, tth/th 找到繞射點 peak 位置。

```
dscan chi -0.1 0.1 20 1
umv chi CEN
```

```
dscan phi -0.05 0.05 20 1
umv phi CEN
```

```
d2scan tth -0.05 0.05 th -0.025 0.025 20 1
umv tth CEN th CEN/2
```

EX : 尋找 in-plane 繞射點

```
295.FOURC> ca 2 0 0
```

計算馬達位置

```
Calculated Positions:
```

```
H K L = 2 8.7267e-06 0
```

```
Alpha = 0 Beta = 0 Azimuth = 90
```

```
Omega = 0 Lambda = 1.033
```

Two Theta	Theta	Chi	Phi
23.8465	11.9230	0.0000	0.0000

```
296.FOURC> umv tth 23.8465 th 11.9230 chi 0.005
```

移動到計算出來的馬達位置

EX : 尋找 in-plane 繞射點

```
umvr chi 0.01; ct
```

微調 chi, phi, tth/th 找到繞射點的強度並慢慢接近繞射點位置。並適時調整衰減片。

```
umvr phi 0.01; ct
```

```
umvr tth 0.01; umvr th 0.01/2 ; ct
```

```
# --- change att
```

```
dscan chi -0.1 0.1 20 1
```

```
umv chi CEN
```

反覆 scan chi, phi, tth/th 找到繞射點 peak 位置。

```
dscan phi -0.05 0.05 20 1
```

```
umv phi CEN
```

```
d2scan tth -0.05 0.05 th -0.025 0.025 20 1
```

```
umv tth CEN th CEN/2
```

EX: miscut 量測 & XRR 量測

```
uan 2 1  
dscan th -0.5 0.5 100
```

尋找光學全反射面, 得到 miscut 角度
 $1 + \text{miscut} = \text{th}$ 最強的位置

```
# -- XRR --  
auto_attn_on  
_miscut = ?
```

```
uan 0 0.0+_miscut
```

XRR 量測開始

```
a2scan tth 0      0.2      th 0.0+_miscut  0.1+_miscut 40  1  
a2scan tth 0.2    0.6      th 0.1+_miscut  0.3+_miscut 40  1  
a2scan tth 0.6    12      th 0.3+_miscut    6+_miscut 114 3
```

```
att 6
```


Macro 撰寫

1. macro 檔的副檔名以 .do 做為結尾
2. 執行 macro 檔的指令為 “do <macro 檔案>”
3. 使用自定變數的時候注意不要用到系統內已有的變數。先用指令 “syms” 確認。
4. 執行倒晶格空間 scan 前先使用 ca 計算，馬達不會跑出極限值之外。
5. 建議分段，並且在每一段開始前確認 mode, sector 設定值以及起始位置。
6. 勤勞寫註解。註解的開頭為 “#”。
7. 預估 macro 可能花費時間。粗估方式: 積分時間總和x2

EX: Surface Normal L-scan

```
setmode 3
freeze

# ---- 0 0 L scan -----
# -- YSZ (0 0 1)c -- ITO (0 0 2)c
# -----
Hs=0.0019519
Ks=0.00102
Ls=1.9973/2
ubr Hs Ks Ls
fw = 0.01
lscan Ls-0.5 Ls-0.2 60 3
lscan Ls-0.2 Ls-fw 50 3
lscan Ls-fw Ls+fw 40 3
lscan Ls+fw Ls+0.2 50 3
lscan Ls+0.2 Ls+0.5 60 3
att8

# -- YSZ (0 0 2)c -- ITO (0 0 4)c
Hs=0.0019519
Ks=0.00102
Ls=1.9973
ubr Hs Ks Ls
fw = 0.005
lscan Ls-0.5 Ls-0.2 60 3
lscan Ls-0.2 Ls-fw 100 3
lscan Ls-fw Ls+fw 40 3
lscan Ls+fw Ls+0.2 100 3
lscan Ls+0.2 Ls+0.5 60 3
att8
```

macro example:

```
auto_attn_off
```

```
# -----  
# 0 SRO(2 2 -1) A :  
# -----
```

```
setsector 4
```

```
Hs = 0
```

```
Ks = -0.5
```

```
Ls = 2
```

```
ubr Hs Ks Ls
```

```
att 0
```

```
lscan Ls-0.3 Ls+0.3 60 60
```

```
ubr Hs Ks Ls
```

```
dscan phi -2 2 40 60
```

```
hscan Hs-0.03 Hs+0.03 60 60
```

```
ubr Hs Ks Ls
```

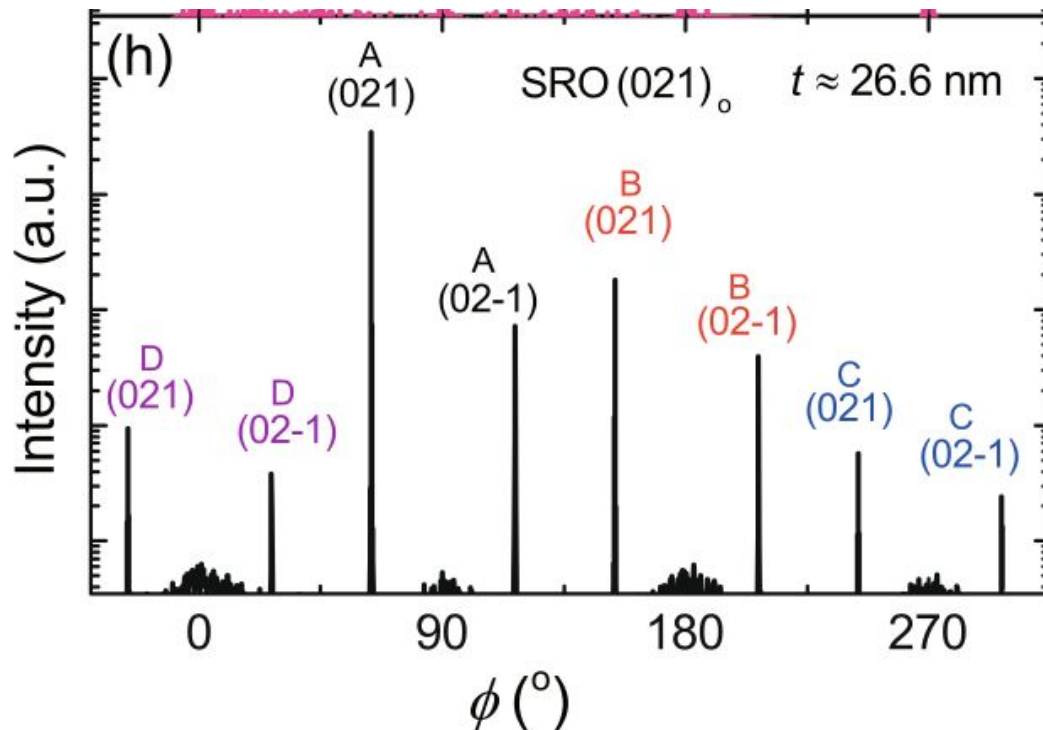
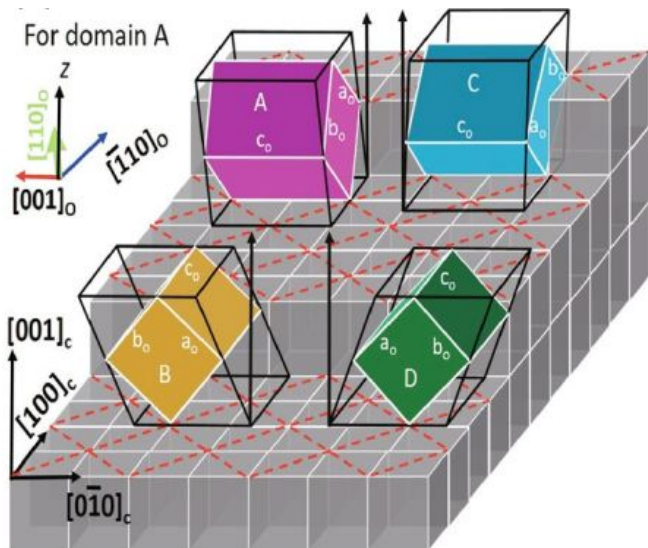
```
kscan Ks-0.03 Ks+0.03 60 60
```

每行開頭用 “#” 做為註解開頭。

每段起始的初始化：設定 sector,
mode, 起始位置。

量測開始

EX: Large phi scan



EX: Large phi Scan Marco

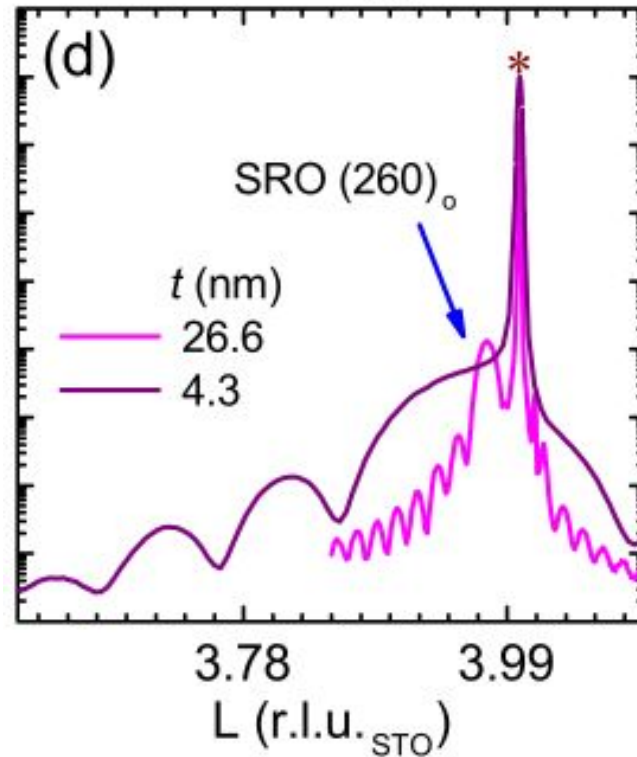
```
# -----  
# PHI : SR0(0 2 1) _ (110) surface normal  
# -----  
  
TTHs = 22.8523  
THs = 11.42615  
CHIs = -41.59685  
PHIs = 60.4598  
umv tth TTHs th THs chi CHIs phi PHIs  
auto_attn_off  
att 0  
  
# ----- A ----  
_dm=0.0  
_dp=53.14  
ascan phi PHIs+_dm-4 PHIs+_dm-0.1 20 8  
ascan phi PHIs+_dm-0.1 PHIs+_dm+0.1 100 8  
ascan phi PHIs+_dm+0.1 PHIs+_dm+_dp-0.1 100 8  
ascan phi PHIs+_dm+_dp-0.1 PHIs+_dm+_dp+0.1 100 8  
ascan phi PHIs+_dm+_dp+0.1 PHIs+_dm+90-4 80 8  
  
# ----- B ----  
_dm=90.0  
_dp=53.14  
ascan phi PHIs+_dm-4 PHIs+_dm-0.1 20 8  
ascan phi PHIs+_dm-0.1 PHIs+_dm+0.1 100 8  
ascan phi PHIs+_dm+0.1 PHIs+_dm+_dp-0.1 100 8  
ascan phi PHIs+_dm+_dp-0.1 PHIs+_dm+_dp+0.1 100 8  
ascan phi PHIs+_dm+_dp+0.1 PHIs+_dm+90-4 80 8
```

```
# ----- C ----  
_dm=180.0  
_dp=53.14  
ascan phi PHIs+_dm-4 PHIs+_dm-0.1 20 8  
ascan phi PHIs+_dm-0.1 PHIs+_dm+0.1 100 8  
ascan phi PHIs+_dm+0.1 PHIs+_dm+_dp-0.1 100 8  
ascan phi PHIs+_dm+_dp-0.1 PHIs+_dm+_dp+0.1 100 8  
ascan phi PHIs+_dm+_dp+0.1 PHIs+_dm+90-4 80 8  
  
# ----- D ----  
_dm=-90  
_dp=53.14  
ascan phi PHIs+_dm-4 PHIs+_dm-0.1 20 8  
ascan phi PHIs+_dm-0.1 PHIs+_dm+0.1 100 8  
ascan phi PHIs+_dm+0.1 PHIs+_dm+_dp-0.1 100 8  
ascan phi PHIs+_dm+_dp-0.1 PHIs+_dm+_dp+0.1 100 8  
ascan phi PHIs+_dm+_dp+0.1 PHIs+_dm+_dp+4 20 8
```

EX: off-normal L-scan

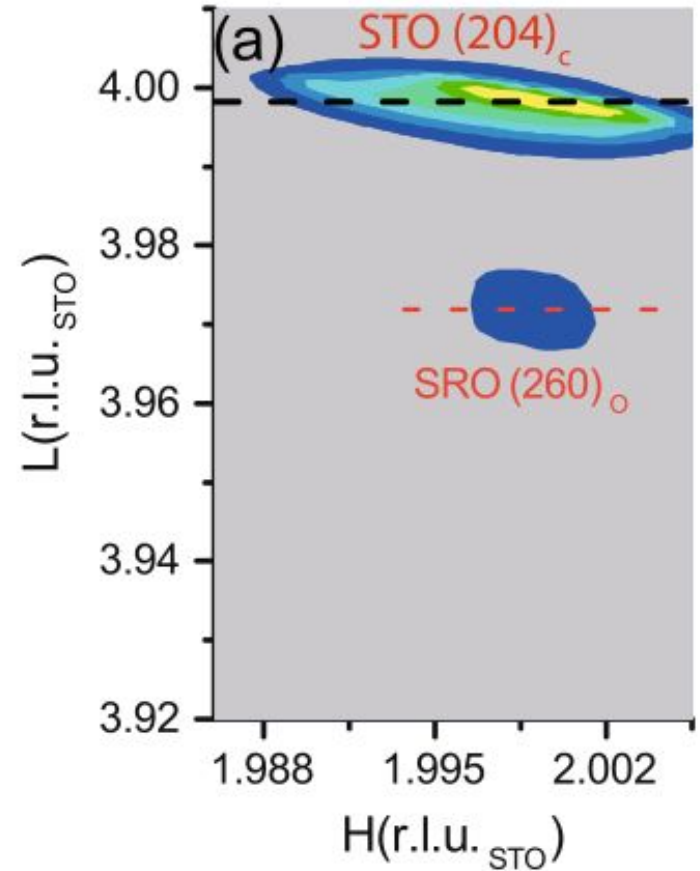
```
auto_attn_on
# -----
# ST0(2 0 4) -- L-scan
# -----

auto_attn_on
setsector 0
Hs = 1.9989
Ks = -0.001977
Ls = 3.9993
ubr Hs Ks 3.5
lscan 3.1      3.5      40  10
lscan 3.5      3.8      60  10
lscan 3.8      3.98     36  10
lscan 3.98     Ls-0.005  5   5
lscan Ls-0.005 Ls+0.005 40  1
lscan Ls+0.005 Ls+0.02  6   5
lscan Ls+0.02  Ls+0.2   36  10
```



EX: hkl mesh scan

```
# -----  
# MESH STO(2 0 4)  
# -----  
Hs = 2.0022  
Ks = -0.0016919  
Ls = 3.9979  
auto_attn_on1  
hklmesh H Hs-0.015 Hs+0.015 30 L 3.8 3.99 76 2  
auto_attn_on2  
hklmesh H Hs-0.015 Hs+0.015 30 L 3.9925 4.005 10 2  
auto_attn_on1  
hklmesh H Hs-0.015 Hs+0.015 30 L 4.0075 4.1 37 2
```



FourC 系統內已存在的變數 (查詢指令: **syms**)

82.FOURC> syms

```
( Built-In/Global/Local Array/Data Number String Constant/Immutable )
624 A (BA...) 120 NO_MV_MOTNUM (G.NS.) 144 UPDATE
144 ADMIN (G..S.) 136 NPTS (G.N..) 144 USER
128 BG (G.NS.) 1024 OUTFILES (BA..I) 120 USER_CHK_ACQ
112 BG_OXEL (G.NS.) 104 PI (B.N.I) 120 USER_CHK_COUNT
112 BG_OYELS (G.NS.) 112 PLOT_CNTRS (G.NS.) 120 USER_CHK_MOVE
136 CCDS (B.N.I) 112 PLOT_FILT (GA...) 144 VERSION
136 COLS (B.N..) 112 PLOT_MODE (G.N..) 112 VFMT
136 COUNT (G.NS.) 112 PLOT_MOTS (G.N..) 144 VHEAD
136 COUNTERS (B.N.I) 136 PLOT_NUM (G.N..) 120 VHKLSCAN_MINSTEP
144 COUNT_FMT_COLS (G.NS.) 240 PLOT_SEL (GA...) 144 VPRNT
144 COUNT_FMT_COMPACT (G.NS.) 120 PLOT_WINDOW_MAX (G.N.C) 112 VSCAN_EXPON
144 COUNT_FMT_DIGITS (G.NS.) 144 PL_A (G..S.) 120 VSCAN_MINSTEP
120 COUNT_FMT_MAXCH (G.N..) 120 PL_AFTER_SCAN (G.N.C) 112 WAITING_ACQ
120 COUNT_FMT_SHOW_MNE (G.N..) 112 PL_ASCII (G.N.C) 120 WAITING_COUNT
136 COUNT_TIME (G.N..) 112 PL_BG_SUB (G.N.C) 120 WAITING_MOVE
144 CP_FILTER (G..S.) 112 PL_DOTSIZE (G.N..) 136 X_L
144 CP_FILTER_CMD (G..S.) 120 PL_FULL_RANGE (G.N.C) 112 X_LM
144 CSCAN_TIMER_PRESET (G.NS.) 112 PL_GRID (G.N.C) 136 Y_L
152 CWD (B..SI) 112 PL_IN_SCAN (G.N.C) 728 Z
```


實驗過程記錄要點

1. 找到繞射點後。記下倒晶格空間位置 (H, K, L) 、馬達位置(tth th chi phi)、det 和 mon 的偵得強度值、衰減片位置。
2. 記下重要 scan 的 scan no 和指令。scan 完後記下半高寬和中心點位置。

補充教材

如有巨集、程式碼之類的教材會放置到下面這個網址

<https://github.com/onionys/2022-TPS09A-Training-Course>

(不定期更新)