



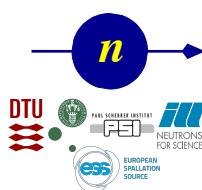
Realistic Laue cameras

- using advanced McStas keywords and a look at polycrystal-samples

2019 CSNS

McStas
School

McStas



Agenda

- ★ *Polycrystals - how to describe them?*
- ★ *Combining a powder sample with a Single Crystal*
- ★ *Tricking the Single_crystal...*
- ★ *A reminder of the GROUP and SPLIT keywords*
- ★ *The dangerous JUMP keyword*
- ★ *A at look at the TOPAZ and SENJU instruments*

From single crystal / crystallites to powder....

中
國
散
裂
中
子
源

Single crystal



Polycrystal with a little disorder,
i.e. a *preferred orientation, texture*

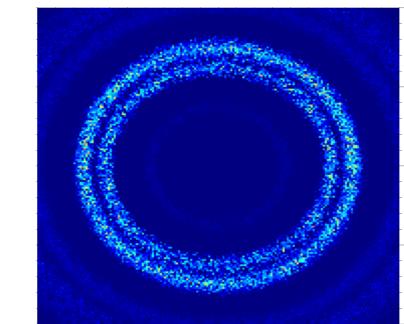
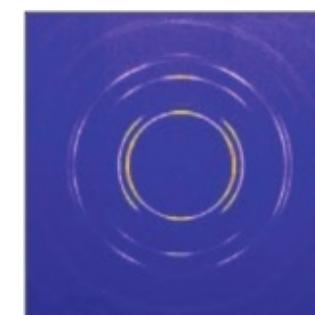
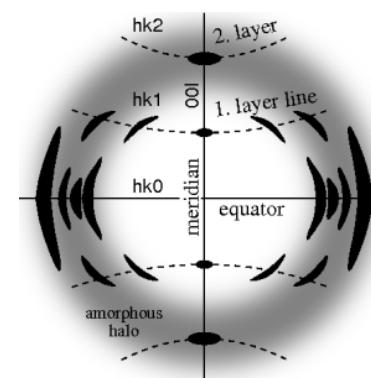
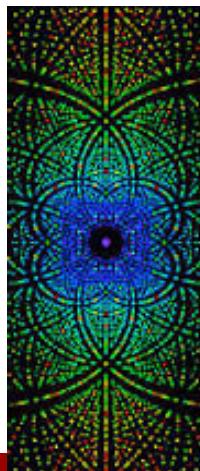


Powder with complete disorder



2019 CSNS
McStas
School

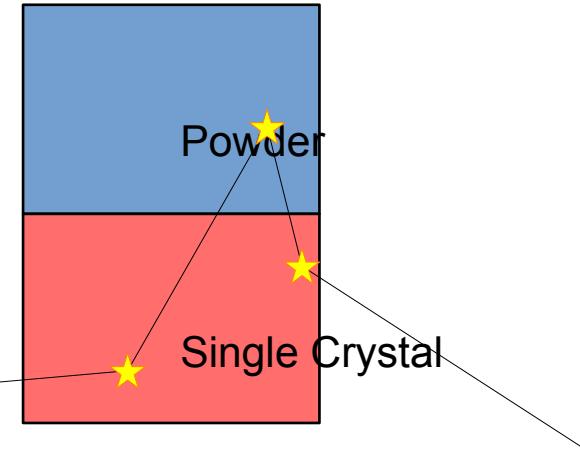
McStas



Powder SX combination

- Use a **GROUP!**
- ... but also allow reentry – How?

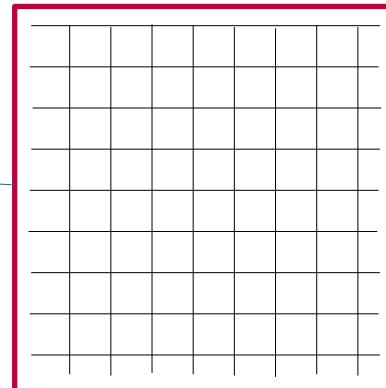
```
COMPONENT a1 = Arm()  
  
AT(0,0,0) RELATIVE sample_pos  
  
EXTEND  
  
%{  
    sample_scatter=0;  
}  
  
COMPONENT pow = PowderN(...)  
  
AT(0,h/2,0) RELATIVE a1  
  
GROUP sample  
  
EXTEND  
  
%{  
    if(SCATTERED) sample_scatter=1;  
}  
  
COMPONENT sx = Single_crystal(...)  
  
AT(0,-h/2,0) RELATIVE a1  
  
GROUP sample
```



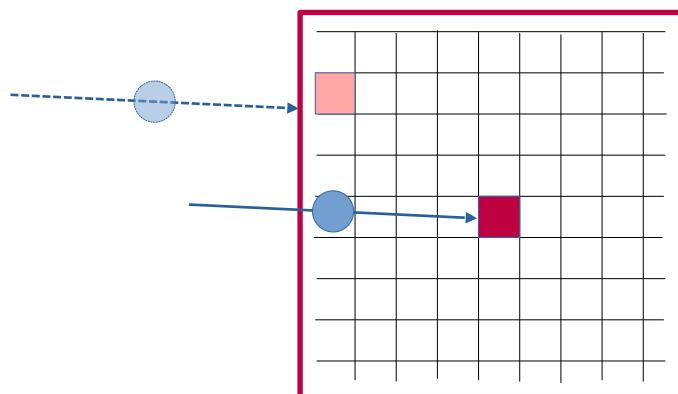
Extending that idea

A polycrystal as many Single crystals

The idea: Cheat a single Single_crystal into believing
it is many...

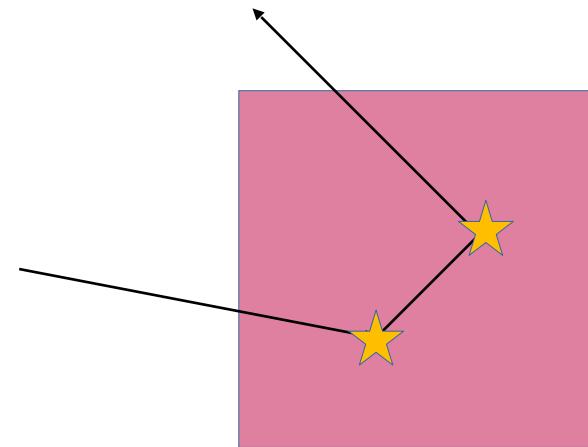
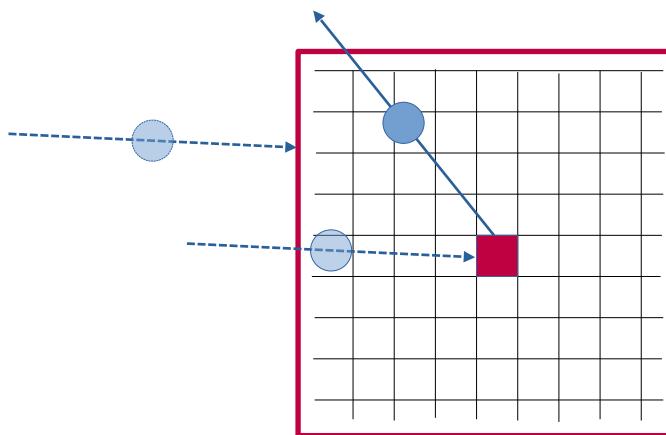


Apply translation



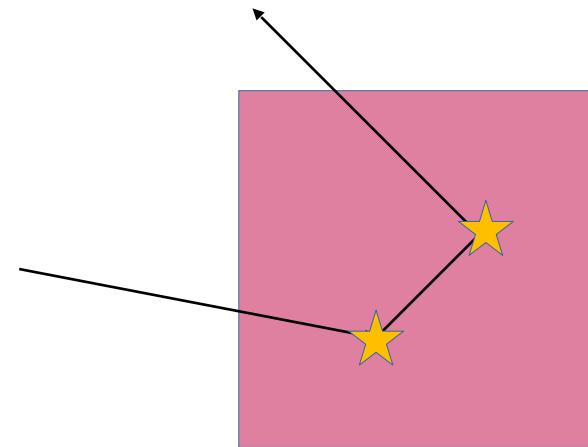
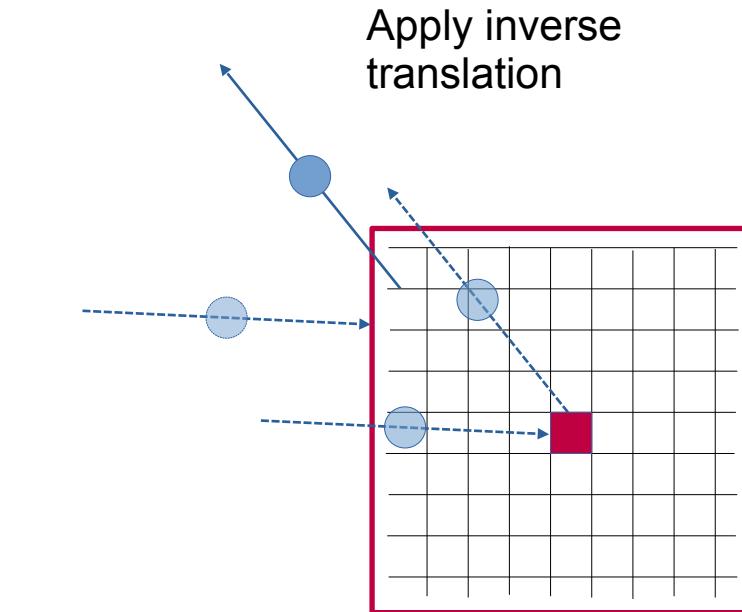
Apply translation

Single_crystal as polycrystal



Scattering events happen
in the crystal
neutron exits.

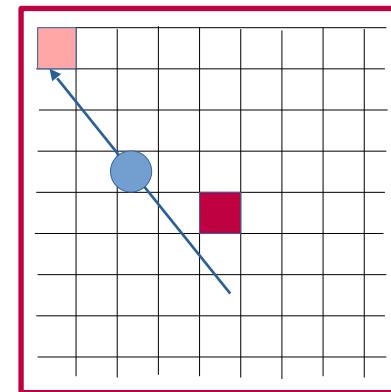
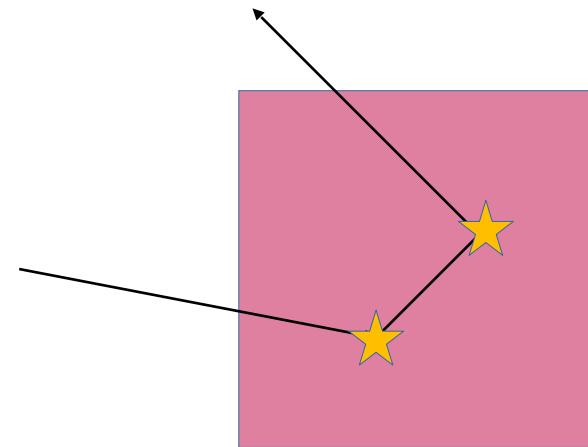
Apply translation



A pink square represents the crystal. Two yellow stars are located within the square, representing scattering events. A black arrow points from the center of the square to the top right, indicating the direction of the scattered neutrons.

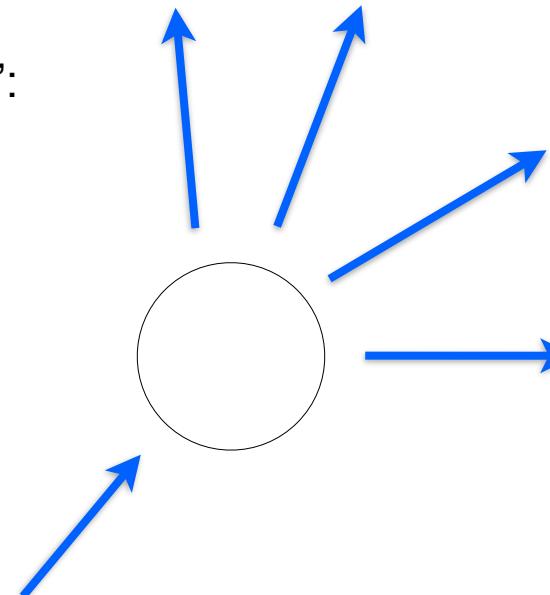
Scattering events happen
in the crystal
neutron exits.

Apply translation

Neutron may
enter a new
subcrystal...Apply inverse
translationScattering events happen
in the crystal
neutron exits.

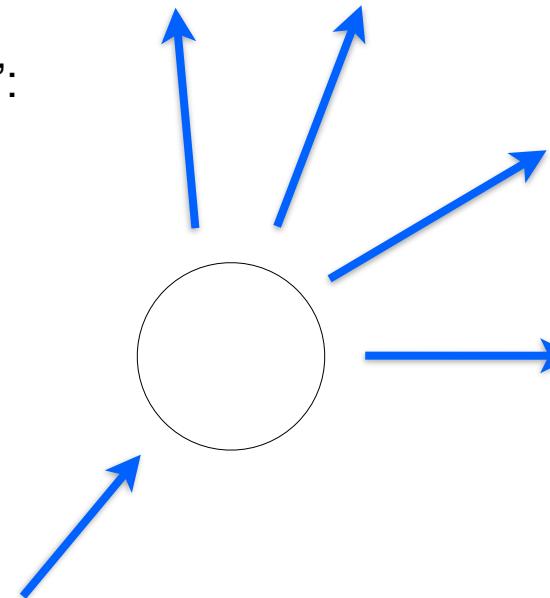
SPLIT

- Increase statistics beyond this point in the instrumentfile
- SPLIT n MyArm = Arm()
- AT somewhere
- will “formulate an if-statement”:
 - for j=1:n
 - comp1
 - comp2
 - comp3
 - ...
 - end (of instrument)
- ONLY meaningful in case of Monte Carlo choices after SPLIT point...



SPLIT

- Increase statistics beyond this point in the instrumentfile
- SPLIT n MyArm = Arm()
- AT somewhere
- will “formulate an if-statement”:
 - for j=1:n
 - comp1
 - comp2
 - comp3
 - ...
 - end (of instrument)
- ONLY meaningful in case of Monte Carlo choices after SPLIT point...



slight sidetrack....

Problem: McStas Single_crystal.comp “slow” for large unit cell diffraction studies

- Example: Rubredoxin

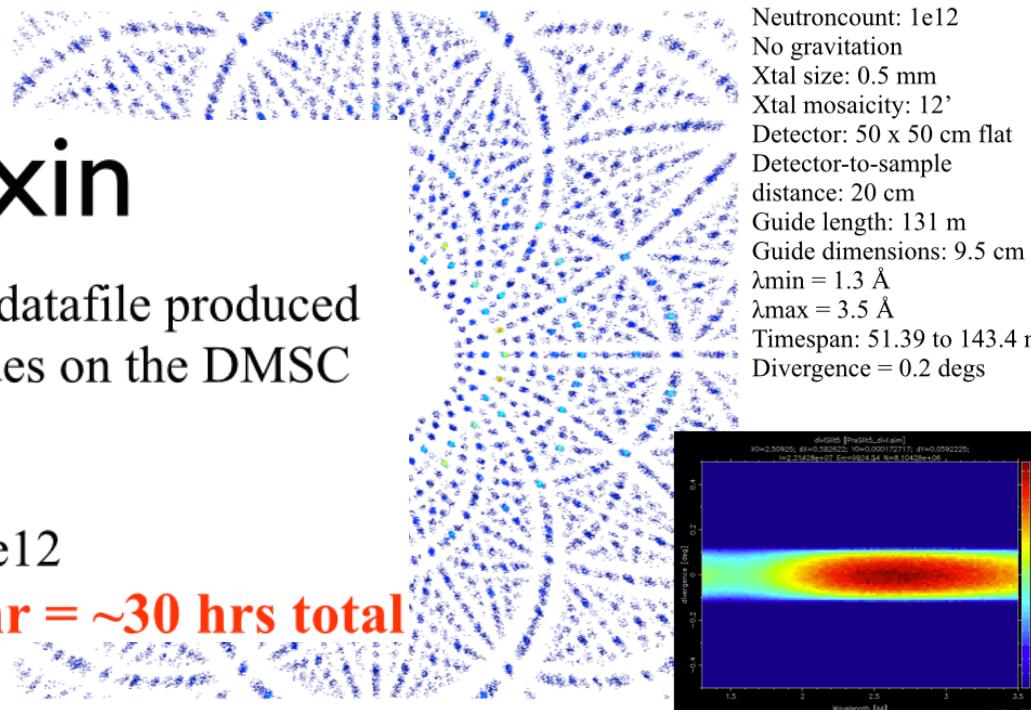
1 timebin, 1000 x,y-bins

Rubredoxin

Images created from simulated datafile produced August 20th 2012 using 25 nodes on the DMSC cluster.

Neutron count: 1e12

Simulation time: ~10 + ~20 hr = ~30 hrs total

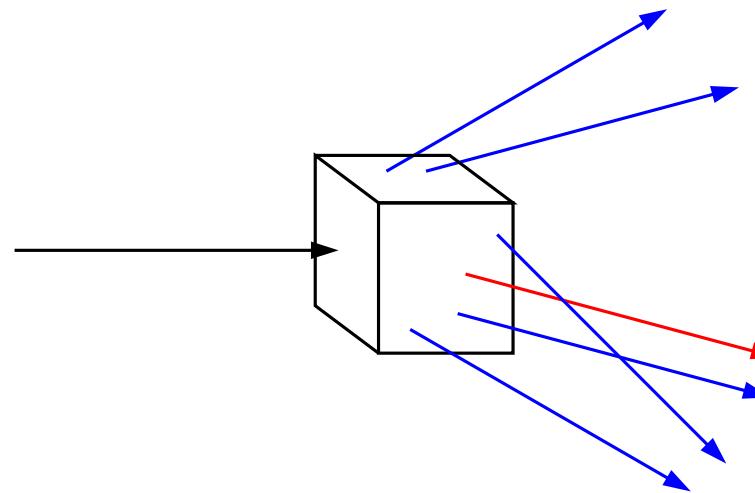


Neutroncount: 1e12
No gravitation
Xtal size: 0.5 mm
Xtal mosaicity: 12'
Detector: 50 x 50 cm flat
Detector-to-sample
distance: 20 cm
Guide length: 131 m
Guide dimensions: 9.5 cm
 $\lambda_{\min} = 1.3 \text{ \AA}$
 $\lambda_{\max} = 3.5 \text{ \AA}$
Timespan: 51.39 to 143.4 ms
Divergence = 0.2 degs

slight sidetrack....

Algorithm improvement: Use incoming neutrons more efficiently - scatter each one on all possible reflections

- **Red:** Original algorithm, one incoming neutron used only once
- **Blue:** Improved algorithm, each incoming neutron scattered (via SPLIT keyword) all possible times
- Component makes **estimate on average number of “active” diffraction spots** - in the case Rubredoxin this is around **50!**



GROUP - components working in parallel



*AT (0,0,-LMM) RELATIVE Cradle ROTATED (0,A1/2,0) RELATIVE Cradle
GROUP IN6Monoks*

*AT (0,0,0) RELATIVE Cradle ROTATED (0,A2/2,0) RELATIVE Cradle
GROUP IN6Monoks*

- One comp after the other is “tried” in sequential order until the neutron was SCATTERED.

EXTEND

- Enrich component behaviour using EXTEND:

```
COMPONENT Mono1 = Monochromator_curved(...)
AT (0,0, -LMM) RELATIVE Cradle ROTATED (0,A1/2,0) RELATIVE Cradle
GROUP IN6Monoks
```

```
EXTEND
```

```
%{
  if (SCATTERED) { myvar = 1; }
%}
```

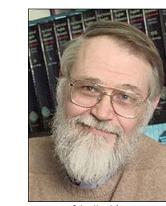
```
...
```

```
COMPONENT Mono2 = Monochromator_curved(...)
AT (0,0, 0) RELATIVE Cradle ROTATED (0,A2/2,0) RELATIVE Cradle
GROUP IN6Monoks
```

```
%{
  if (SCATTERED) { myvar = 2 ;}
%}
```



K & R. / GNU



SEGUNDA EDICIÓN
 en ANSI C
 EL
 LENGUAJE DE
 PROGRAMACIÓN

C

BRIAN W. KERNIGHAN
 DENNIS M. RITCHIE

WHEN

- Syntax:

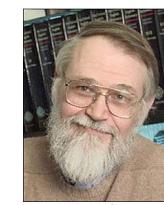
```
COMPONENT Mine = Yours(blah, blah)
WHEN (c-expression) AT (....)
```

- Is very powerful when combined with EXTEND and user variables, or as a method to let input parameters select if certain components are active.
- Example: Use EXTEND to flag if neutron was scattered on one monochromator blade or another. Then later use WHEN to only show contribution from blade N at sample position?

```
COMPONENT Mon = PSD_monitor(...)
WHEN (myvar==1) AT (0,0,0) RELATIVE Sample
```



K & R. / GNU



SEGUNDA EDICIÓN
 EL
 LENGUAJE DE
 PROGRAMACIÓN

C

BRIAN W. KERNIGHAN
 DENNIS M. RITCHIE

JUMP

- A goto. Be careful. Can be used in two situations:
- JUMP to myself
- JUMP to an Arm
- No coordinate transformations are applied... (Meaning that if the Arms you JUMP between do not coincide you will “move” / “reorient” the neutrons...)
- Syntaxes:
- COMPONENT a=b(...)
- WHEN (expr) AT (...) JUMP somewhere
- COMPONENT a=b(...)
- WHEN (expr) AT (...) JUMP myself



JUMP

- A goto. Be careful. Can be used in two situations:
- ***BEWARE - This IS a GOTO!***
- No coordinate transformations are applied... (Meaning that if the Arms you JUMP between do not coincide you will “move” / “reorient” the neutrons...)
- Syntaxes:
- COMPONENT a=b(...)
- WHEN (expr) AT (...) JUMP somewhere
- COMPONENT a=b(...)
- WHEN (expr) AT (...) JUMP myself



JUMP

- A goto. Be
- **BEW**
- No coordin
- between d
- Syntaxes:
- COMPONI
- WHEN (ex
- COMPONI
- WHEN (ex



that if the Arms you JUMP
 the neutrons...)



COPY- inside instruments

- In instruments: (see ILL_H25.instr)
 - COMPONENT H25_1 = Guide_gravity(
 - w1=0.03, h1=0.2, w2=0.03, h2=0.2, l=L_H25_1,
 - R0=gR0, Qc=gQc, alpha=gAlpha, m=m, W=gW)
 - AT (0,0,AI_Thickness+gGap) RELATIVE PREVIOUS
 - ROTATED (0,Rh_H25_1,0) RELATIVE PREVIOUS
 - COMPONENT COPY(H25_1) = COPY(H25_1)
 - AT (0,0,L_H25_1+gGap) RELATIVE PREVIOUS
 - ROTATED (0,Rh_H25_1,0) RELATIVE PREVIOUS
 - COMPONENT COPY(H25_1) = COPY(H25_1)(W=2*gW)
 - AT (0,0,L_H25_1+gGap) RELATIVE PREVIOUS
 - ROTATED (0,Rh_H25_1,0) RELATIVE PREVIOUS



TOPAZ and SENJU

Take a look at a couple of “real” instrument simulations of Laue Cameras:

they both resides in GitHub under today /
[3_Laue_diffractometers_TOPAZ\&SENJU/](https://github.com/3_Laue_diffractometers_TOPAZ\&SENJU/)

1. SNS TOPAZ
2. J.PARC SENJU (aka. BL18)

Here you need to also copy a datafile for the J-PARC source
“source_BL18.txt”



Surgeon General's Warning:
There are lots of COMPONENTS here