

# McStas introduction

ICNX 2009 pre-workshop on McStas

**Peter Willendrup**<sup>1</sup>

Emmanuel Farhi<sup>2</sup>

Erik Knudsen<sup>1</sup>

Kim Lefmann<sup>3</sup>

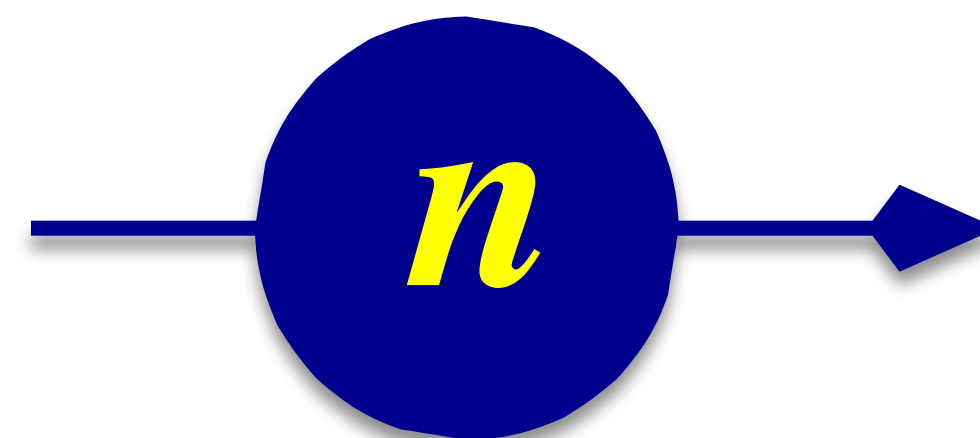
<sup>1</sup>Materials Research Division, RISØ DTU, Roskilde, Denmark

<sup>2</sup>Scientific Computing, Institut Laue-Langevein (ILL), Grenoble, France

<sup>3</sup>Niels Bohr Institute, University of Copenhagen, Copenhagen, Denmark.



# *McStas*



McStas project <http://www.mcstas.org> [mcstas-users@mcstas.org](mailto:mcstas-users@mcstas.org)  
Risø DTU, Niels Bohr Institute, Institut Laue-Langevin

# Agenda

- McStas project
- Applications of McStas
- Reliability
- Implementation and usage

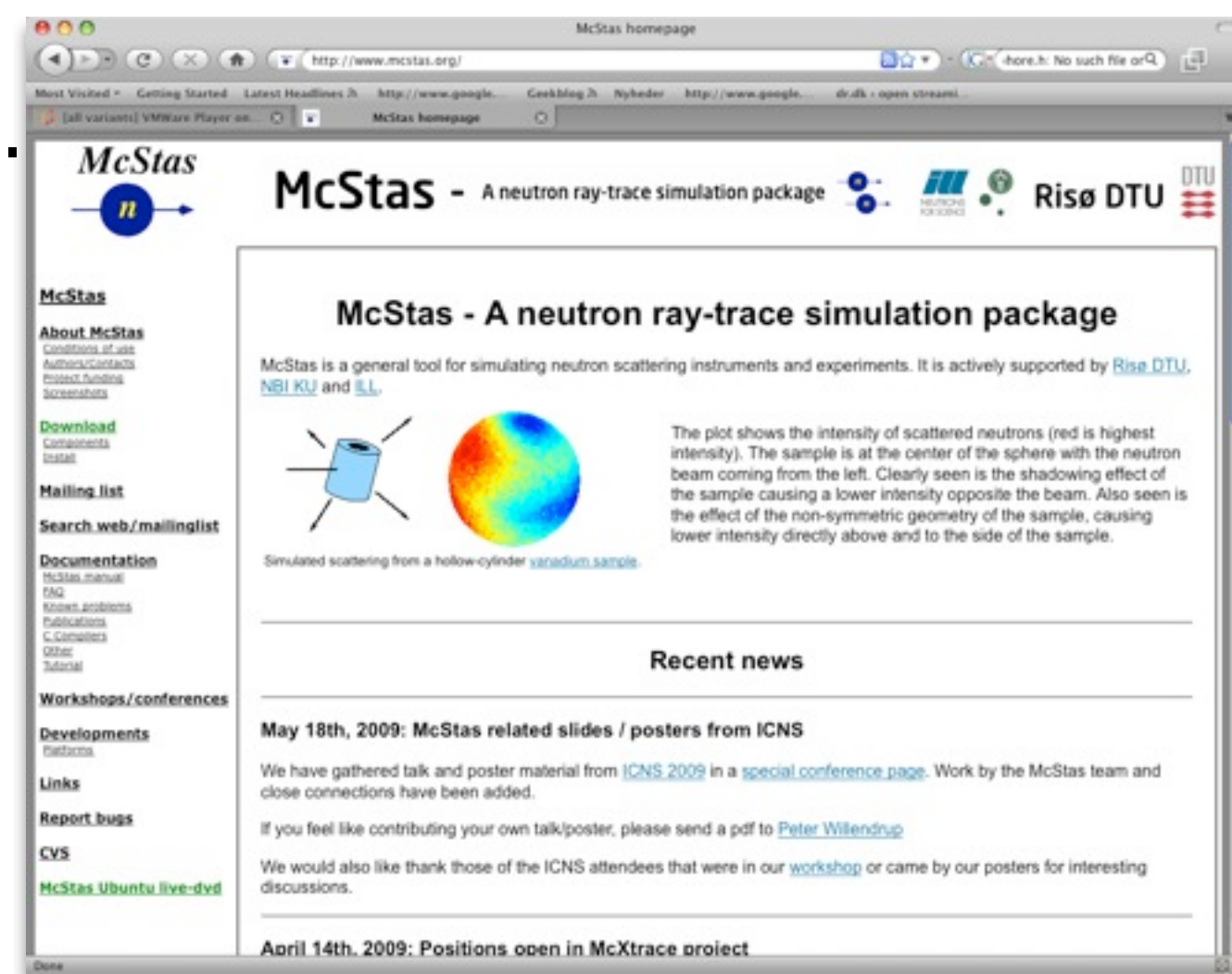
# McStas Introduction

- Flexible, general simulation utility for neutron scattering experiments.
- Original design for Monte carlo Simulation of triple axis spectrometers
- Developed at RISØ DTU, KU and ILL, Grenoble.
- V. 1.0 by K Nielsen & K Lefmann (1998)
- Currently 2.5+1 people full time plus students
- International users/contributors

GNU GPL license  
Open Source

Project website at  
<http://www.mcstas.org>

[neutron-mc@risoe.dk](mailto:neutron-mc@risoe.dk) mailinglist



# McStas Introduction

## McXtrace - new startup (2009) in X-ray sim

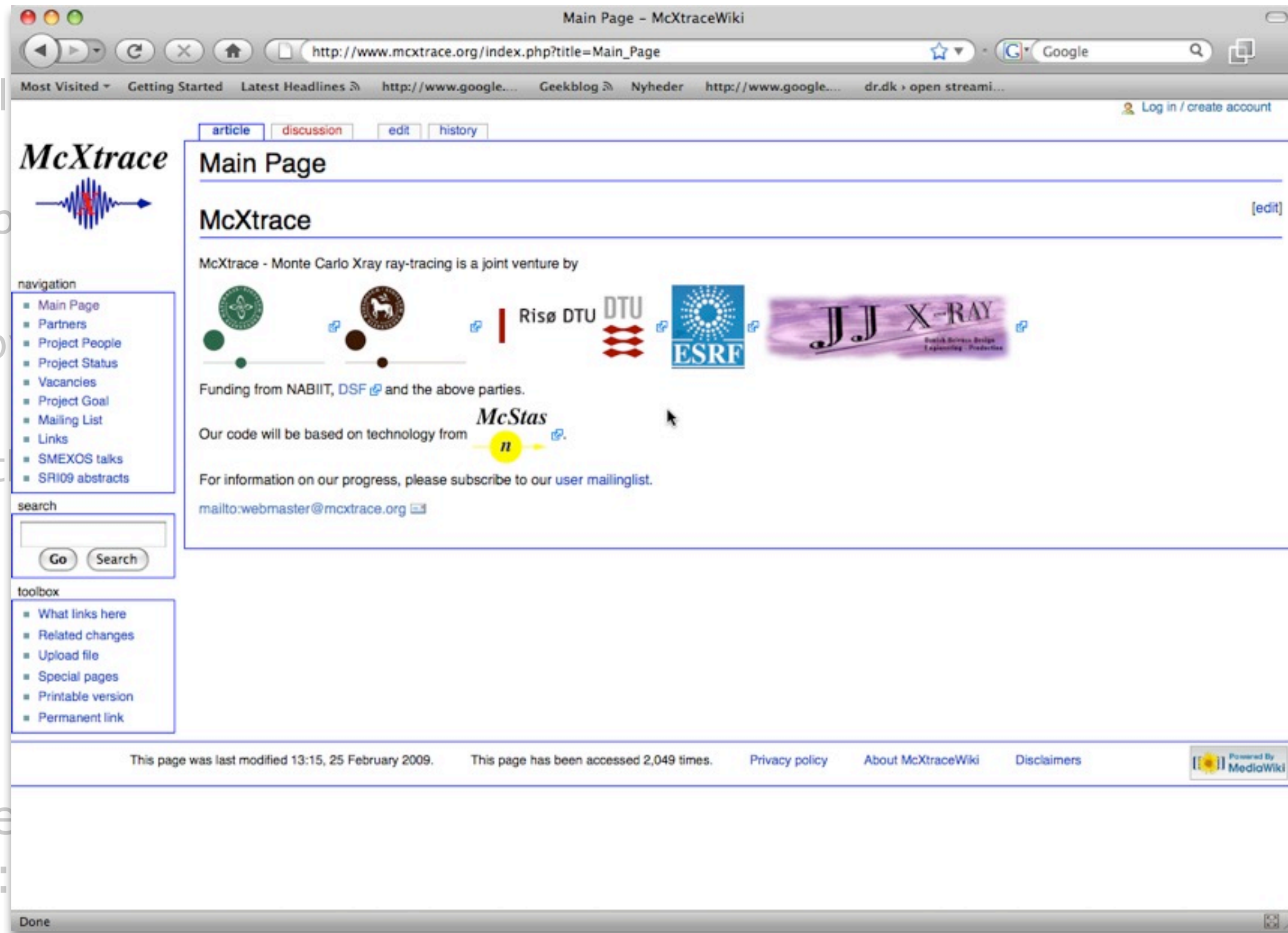
- Flexible, general simulation utility for neutron scattering experiments.

• Original

• Develop

• V. 1.0 b

• Current



Project  
http:

list

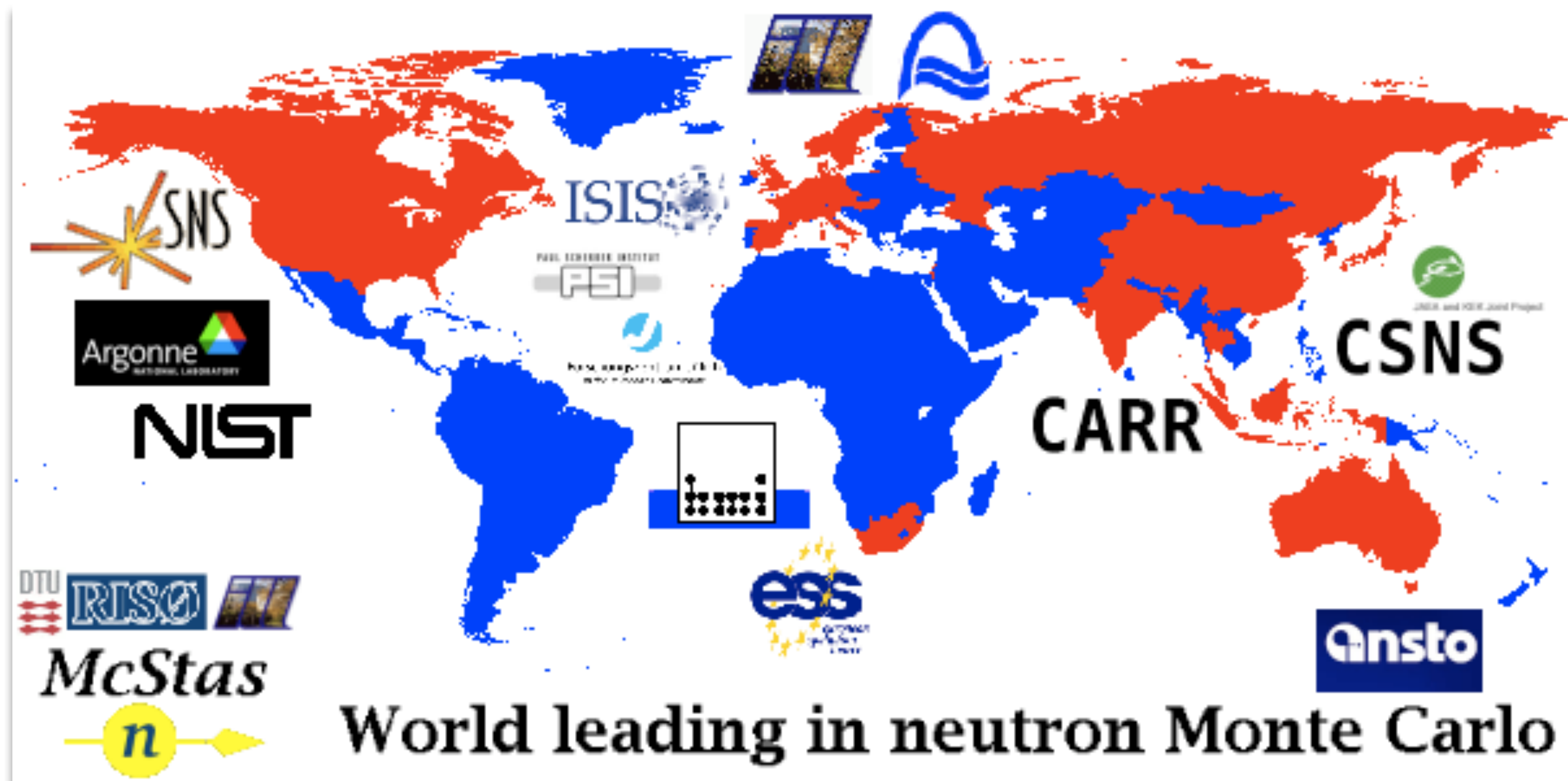
Risø DTU, Niels Bohr Institute, Institut Laue-Langevin





# McStas Introduction

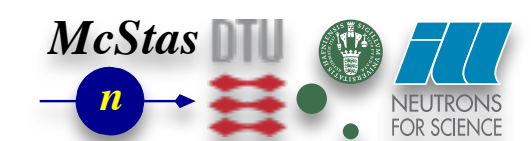
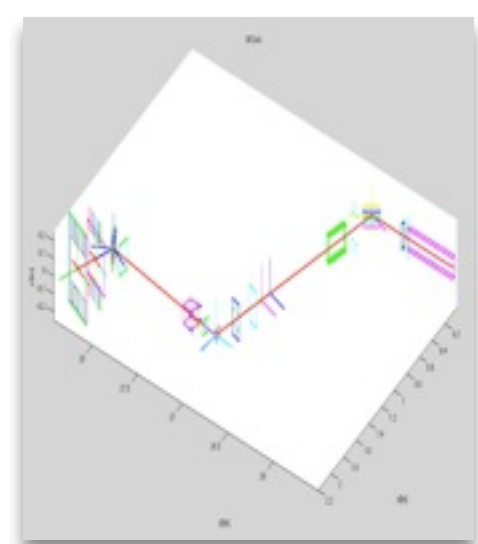
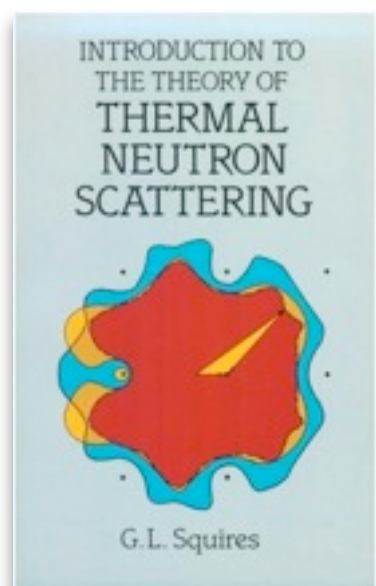
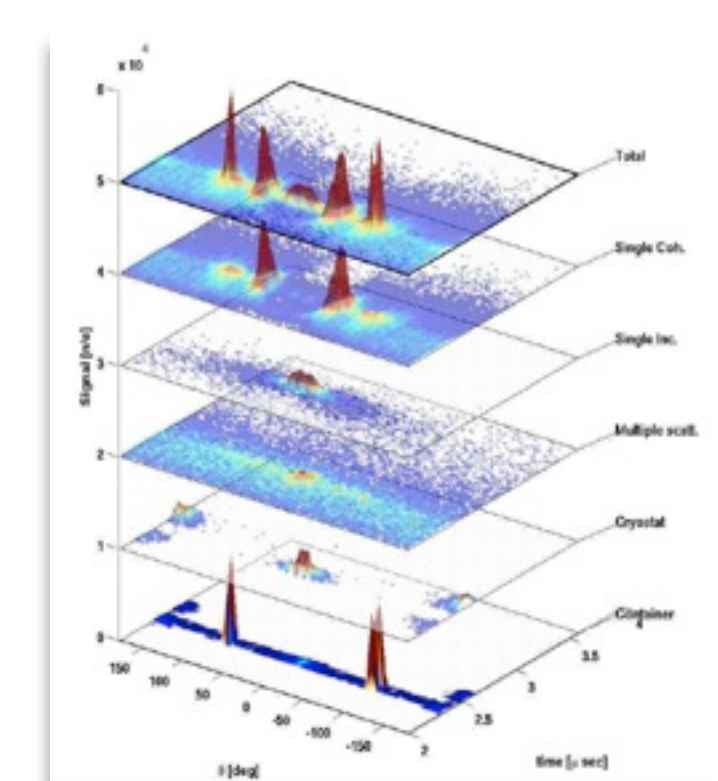
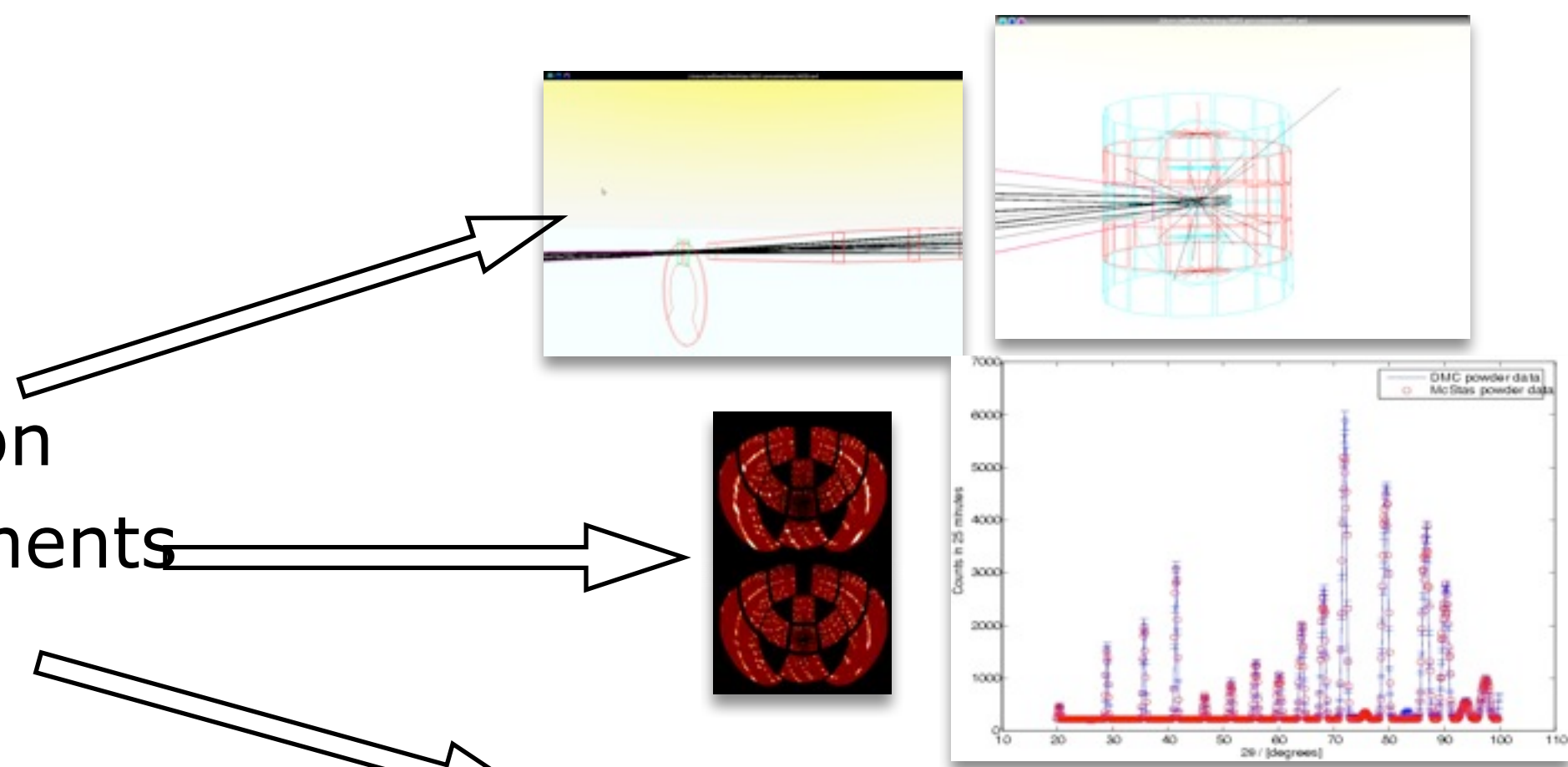
- Used at all major neutron sources



# What is McStas used for?

- Instrumentation
- Virtual experiments
- Data analysis
- Teaching

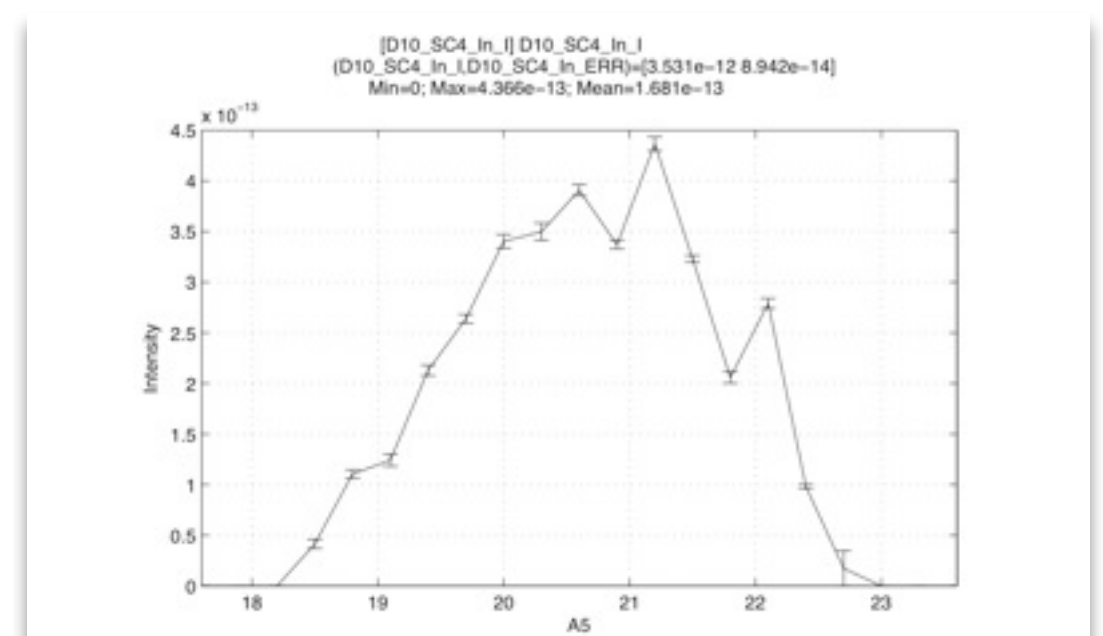
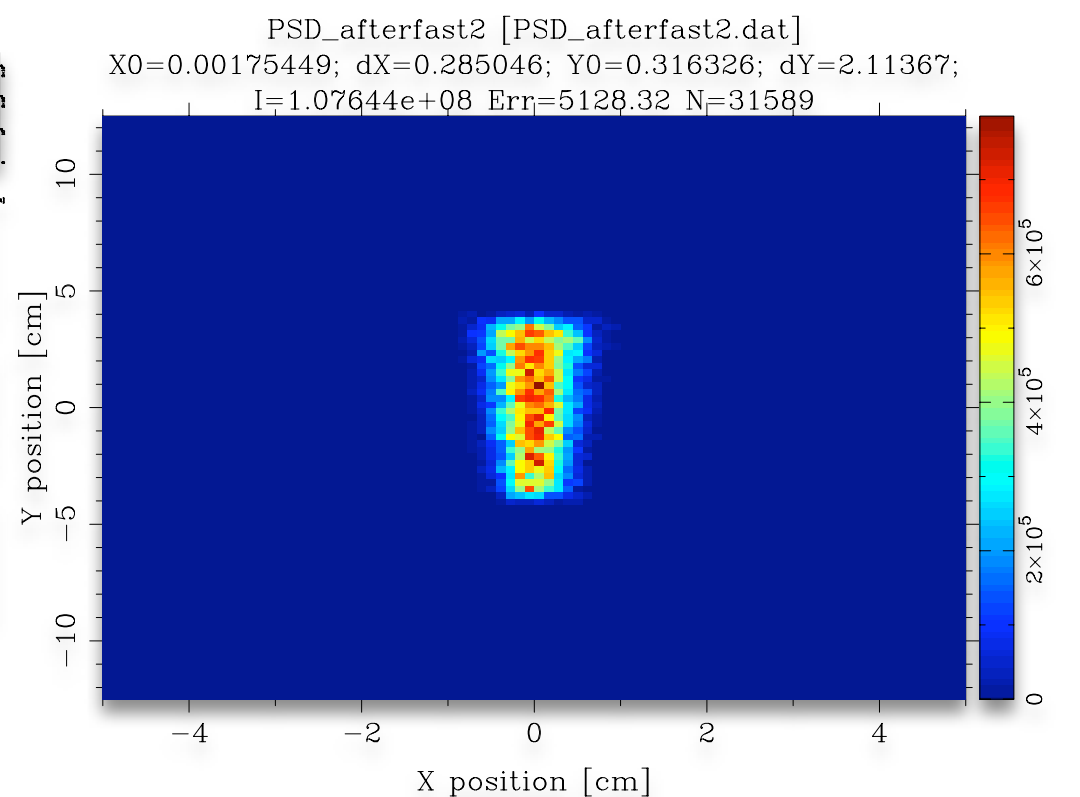
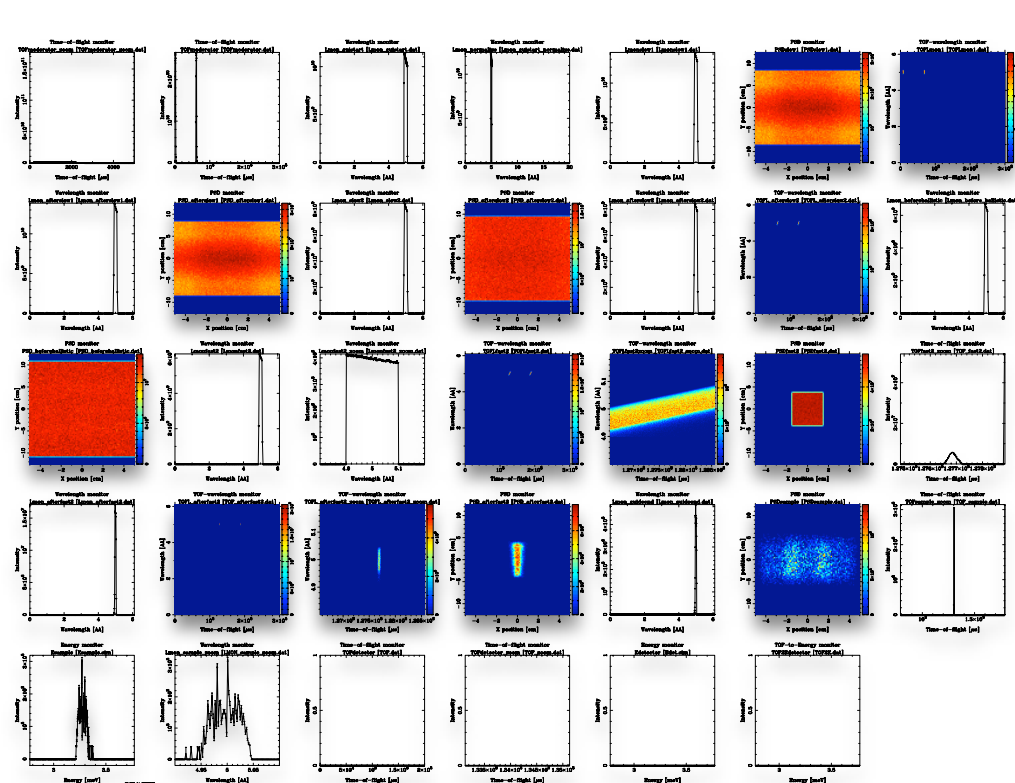
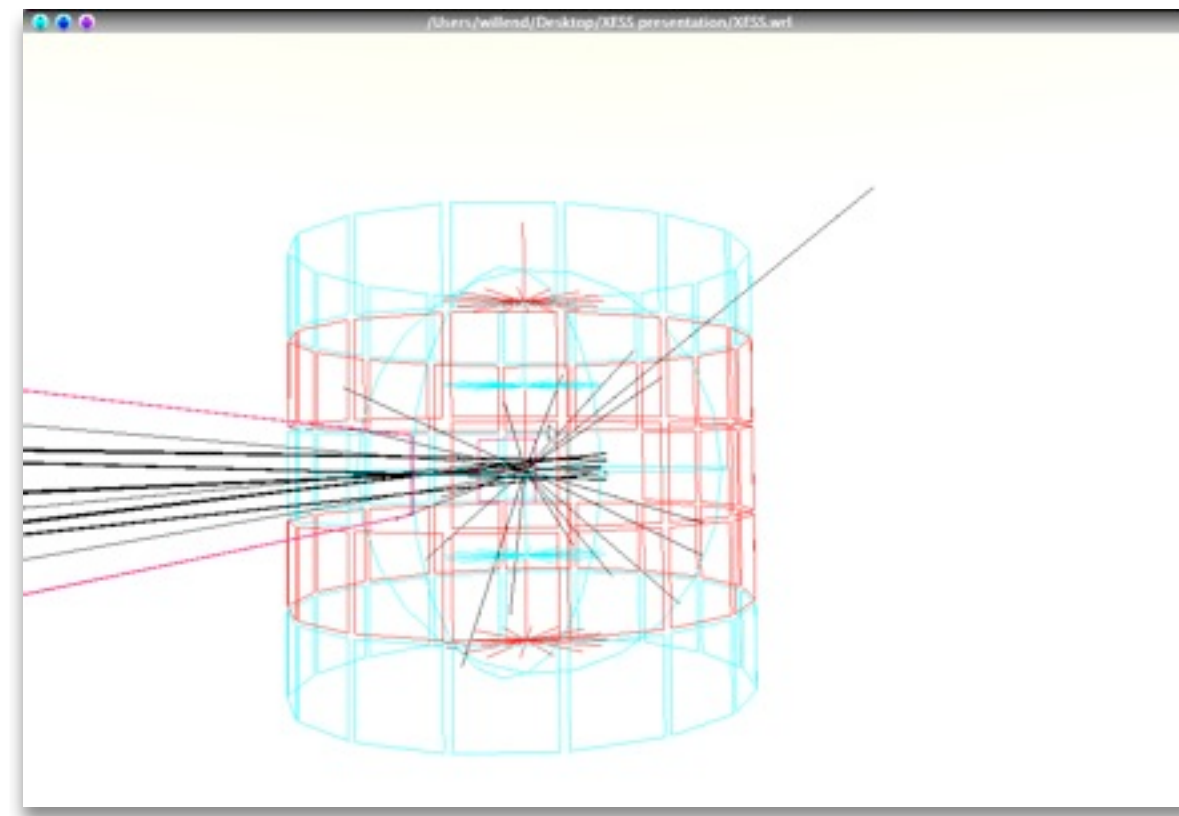
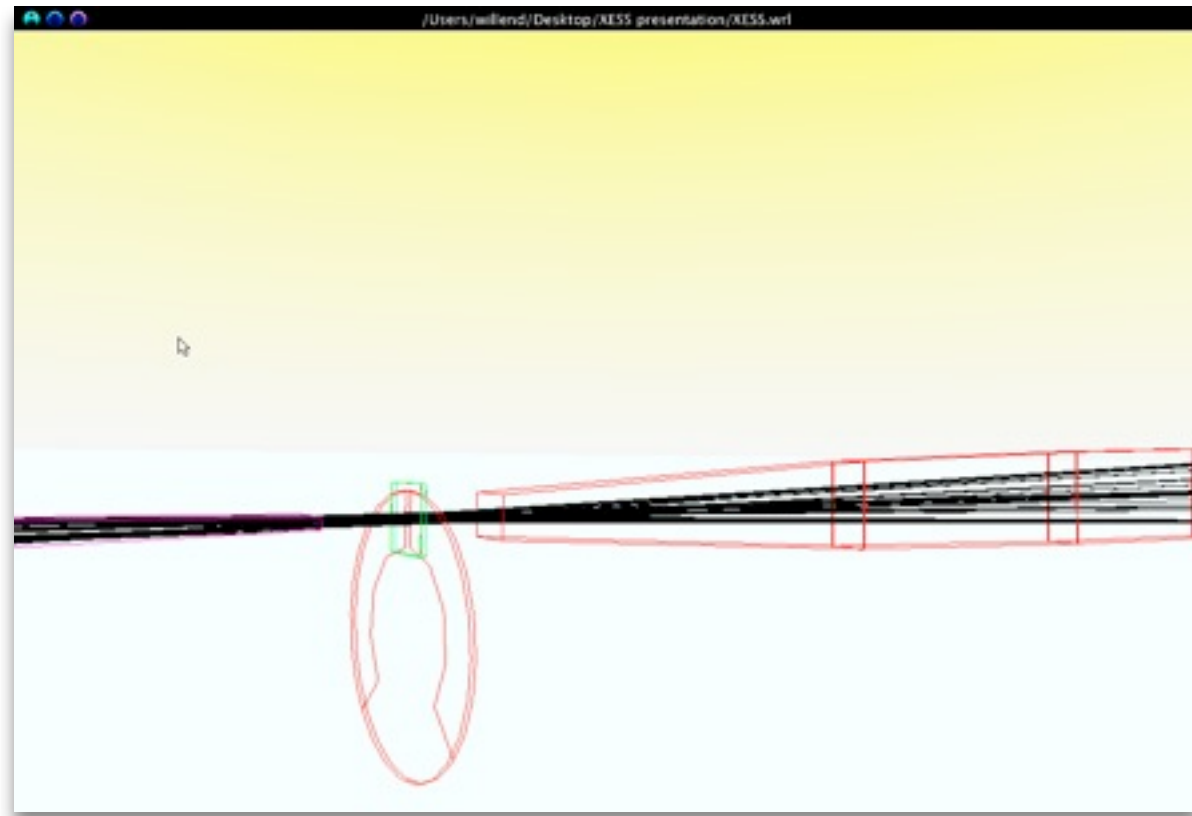
(KU 2005-2009)



Risø DTU, Niels Bohr Institute, Institut Laue-Langevin

# Instrumentation

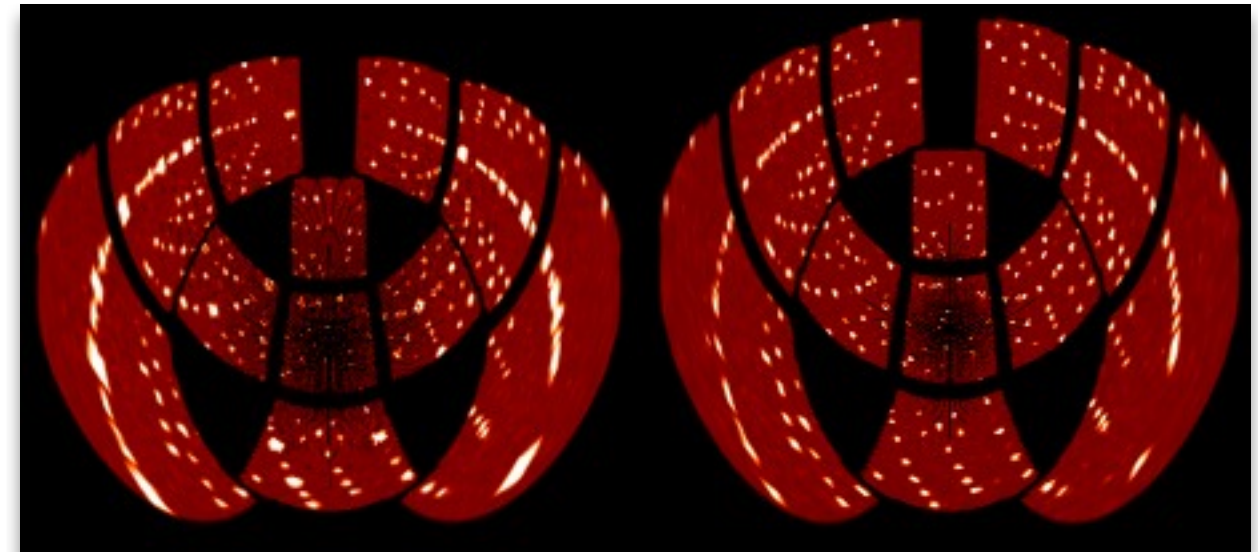
- Design and optimization of instruments



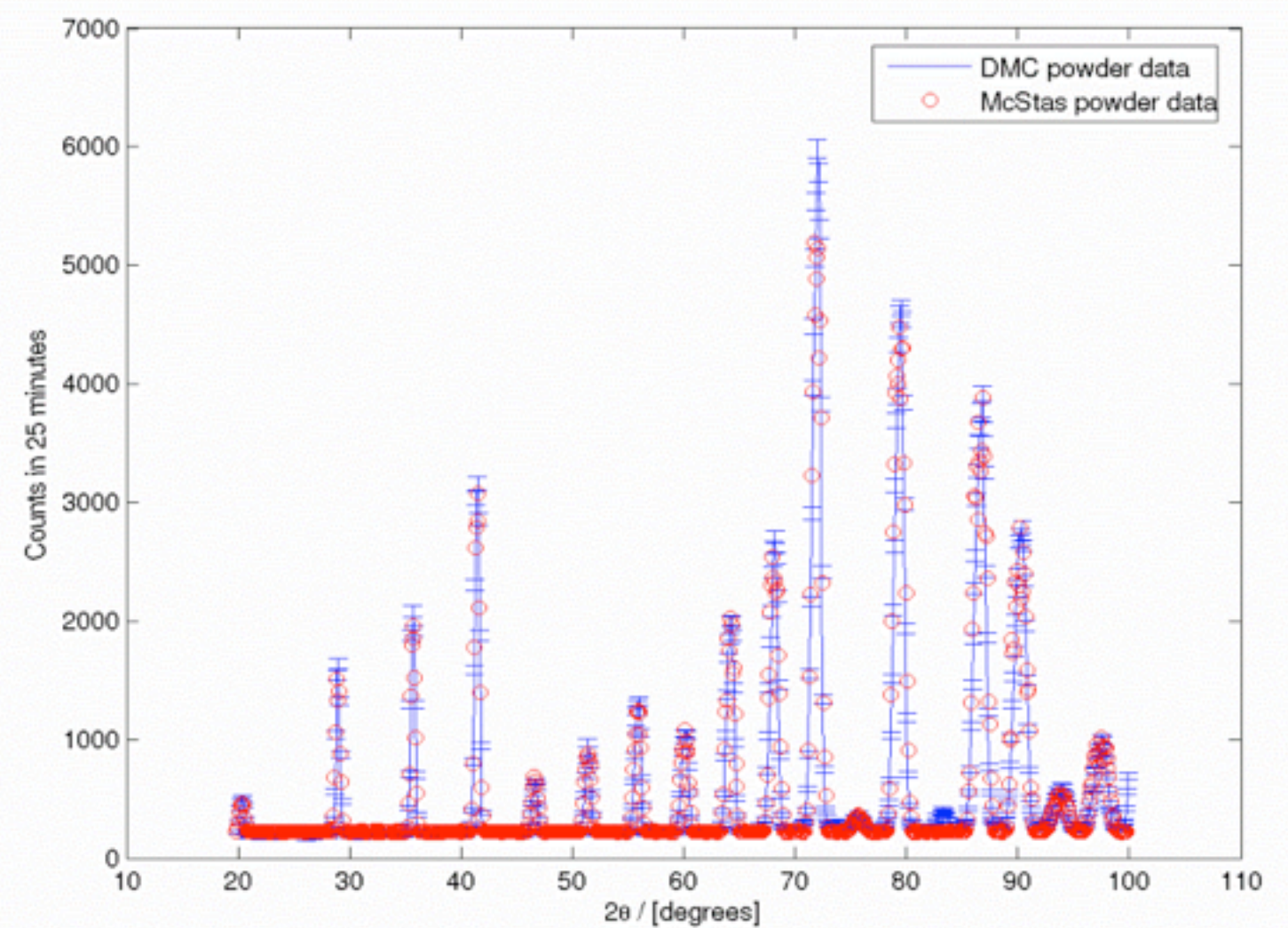
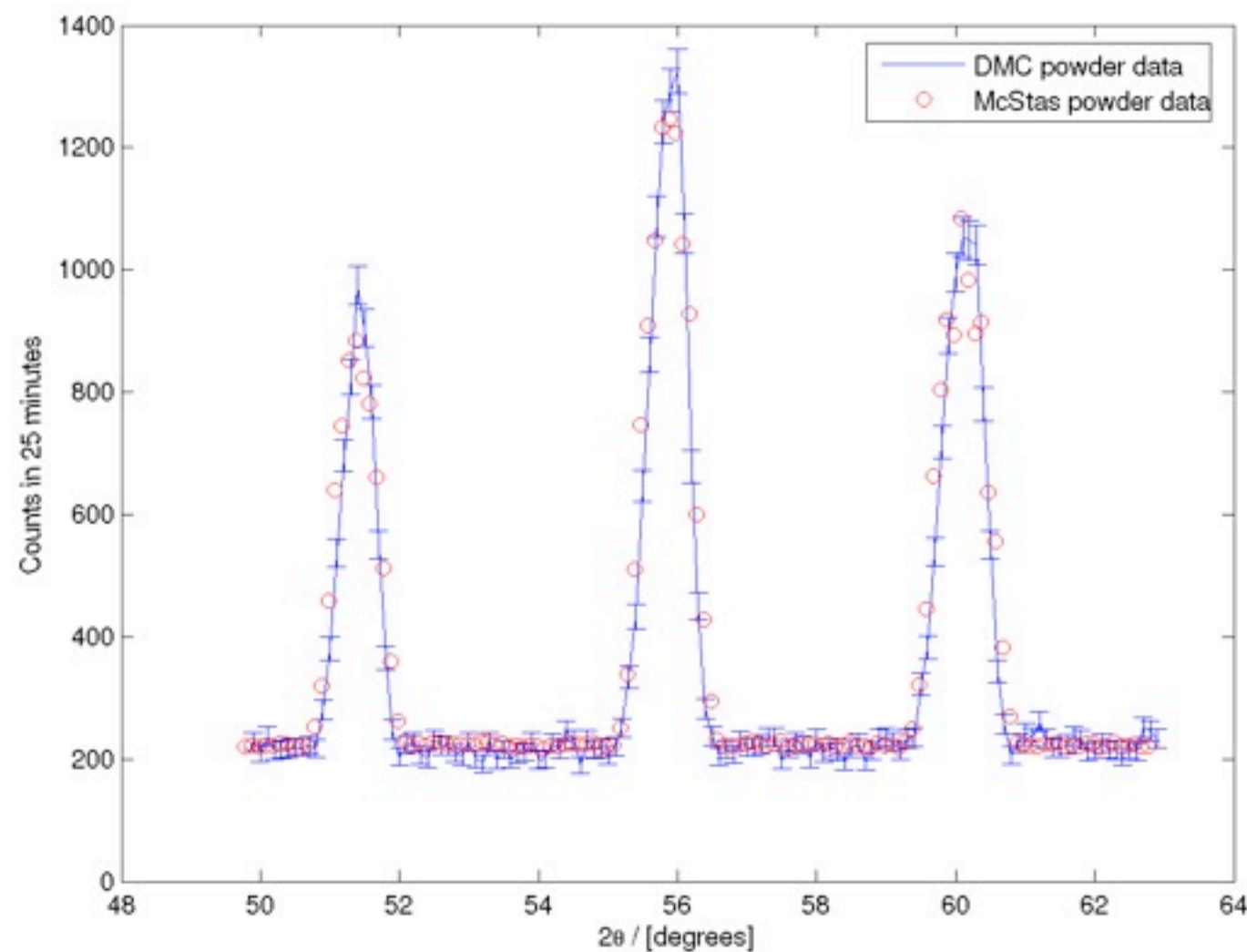


# Virtual experiments (VE) (definition:)

- Simulation of a complete experiment
- ... from source to detector
- Ideally controlled like real experiment.
- Data analysed by "real" analysis programs



A. Daud-Aladine, ISIS



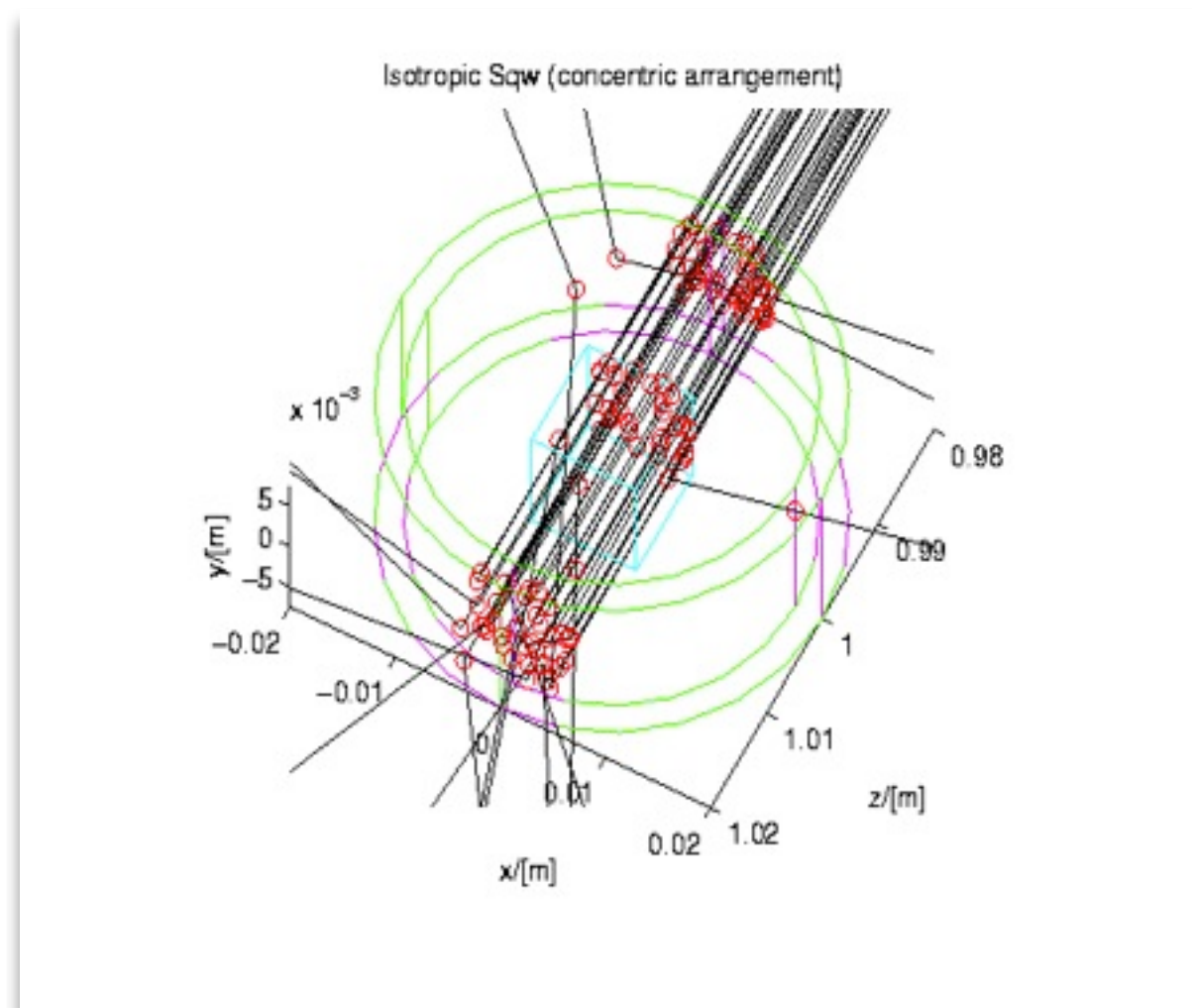
P. Willendrup, Risø DTU; Uwe Filges, L. Keller, PSI



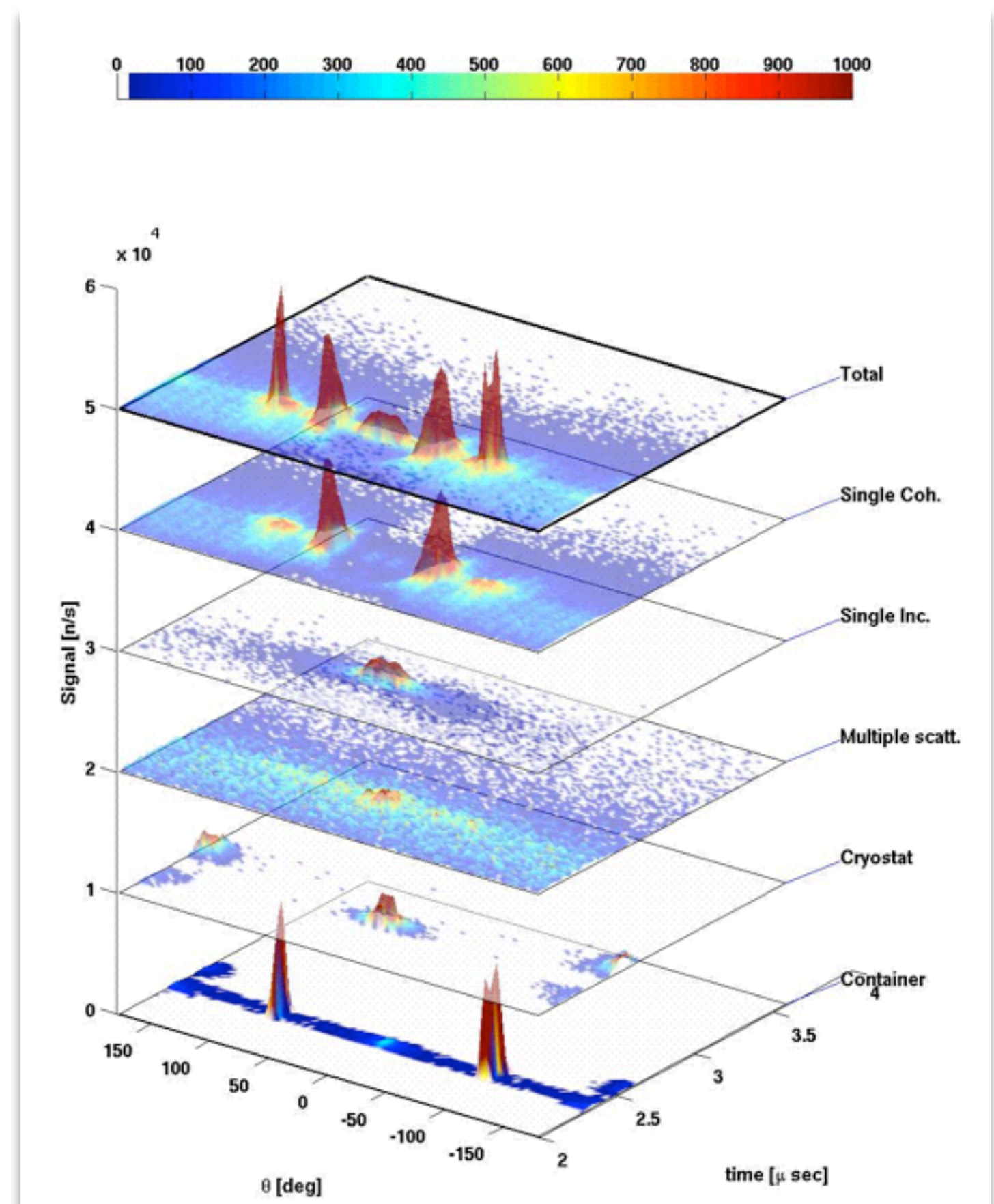
# Data analysis (1)

(using VE techniques)

- Virtual TOF exp. at IN6, ILL
- Liquid Ge sample
- Coherent / incoherent
- Multiple scattering
- And sample environment
- All contributions can be separated by VE !



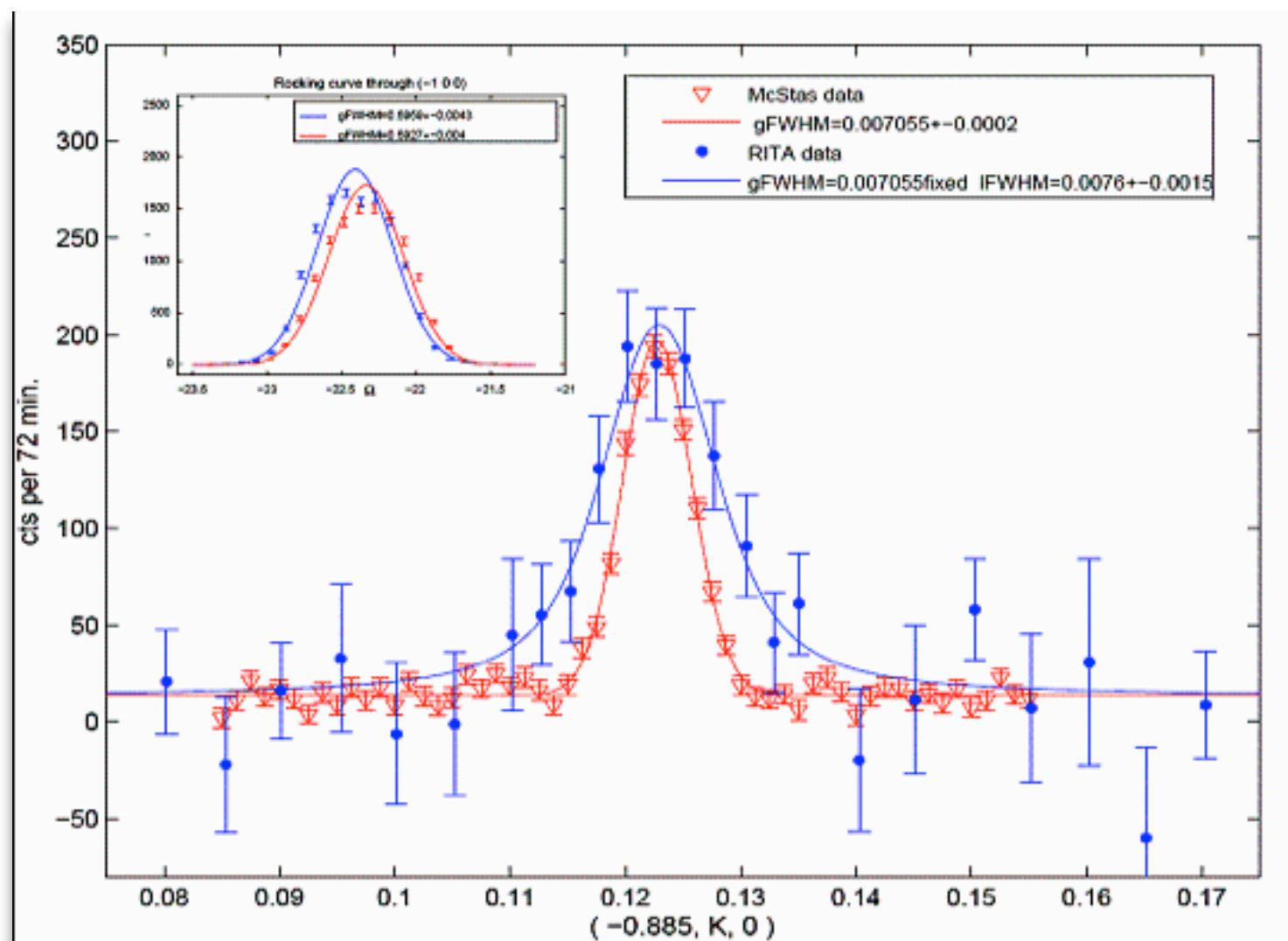
E. Farhi, ILL



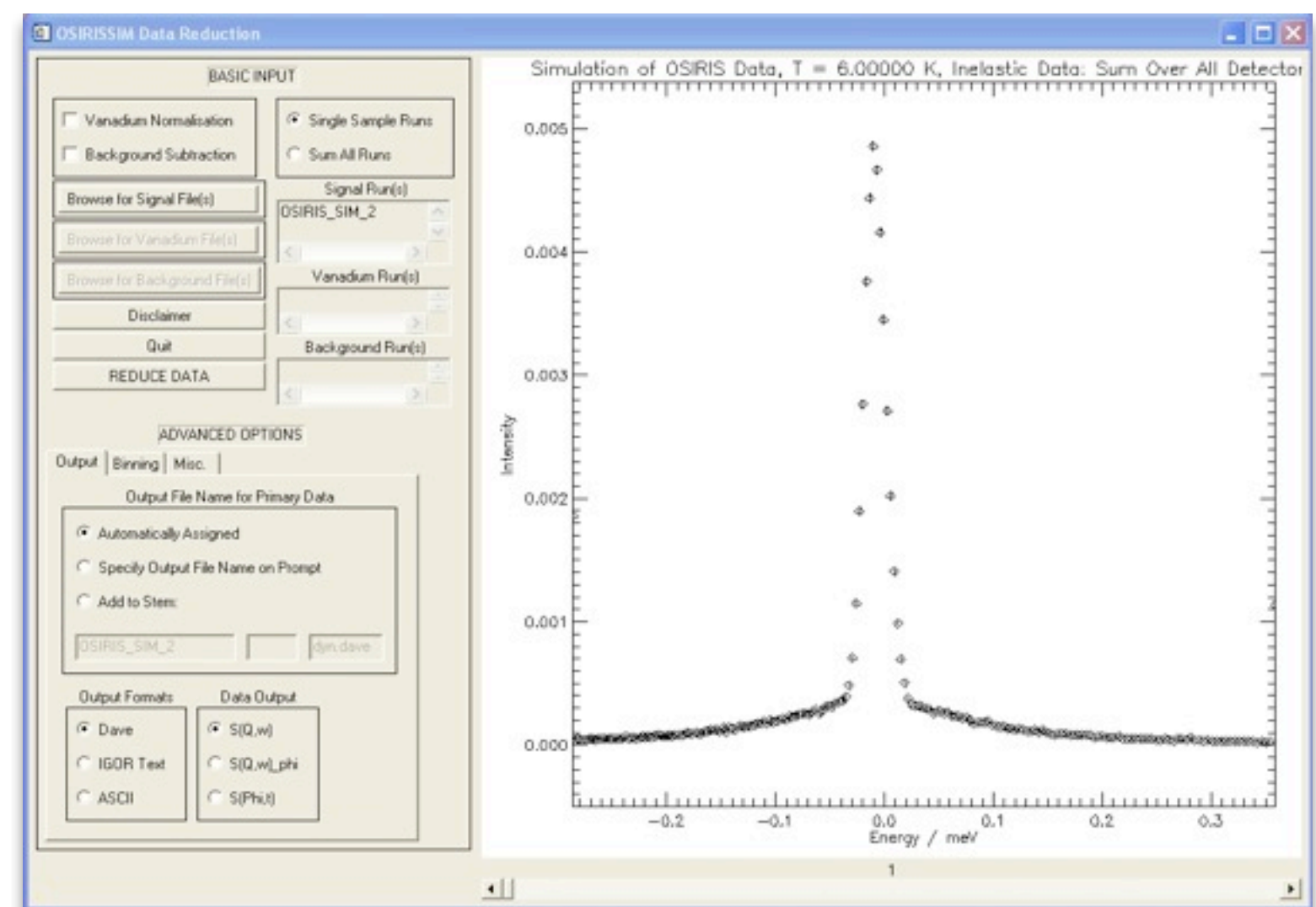
# Data Analysis (2)

## (using VE techniques)

- VE data has been used to test data analysis programs
- ... and to check resolution effects



L. Udby, Risø-DTU

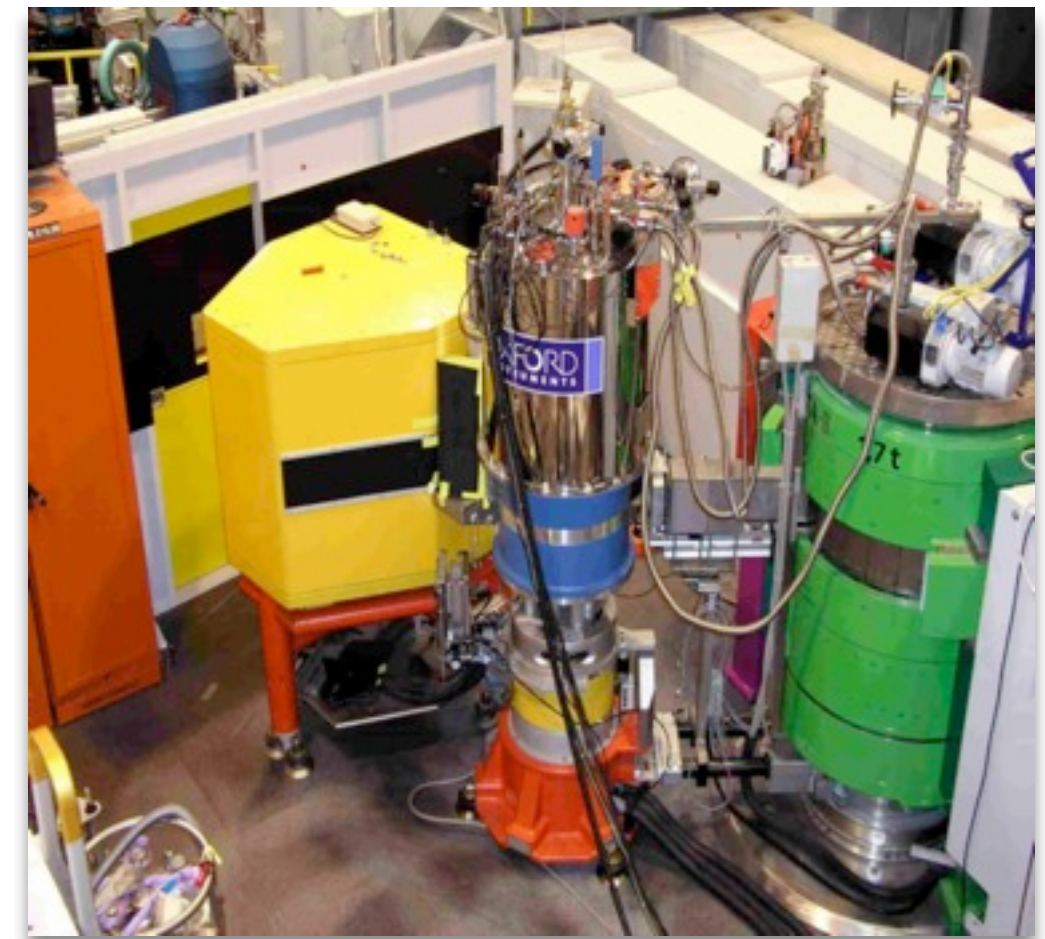
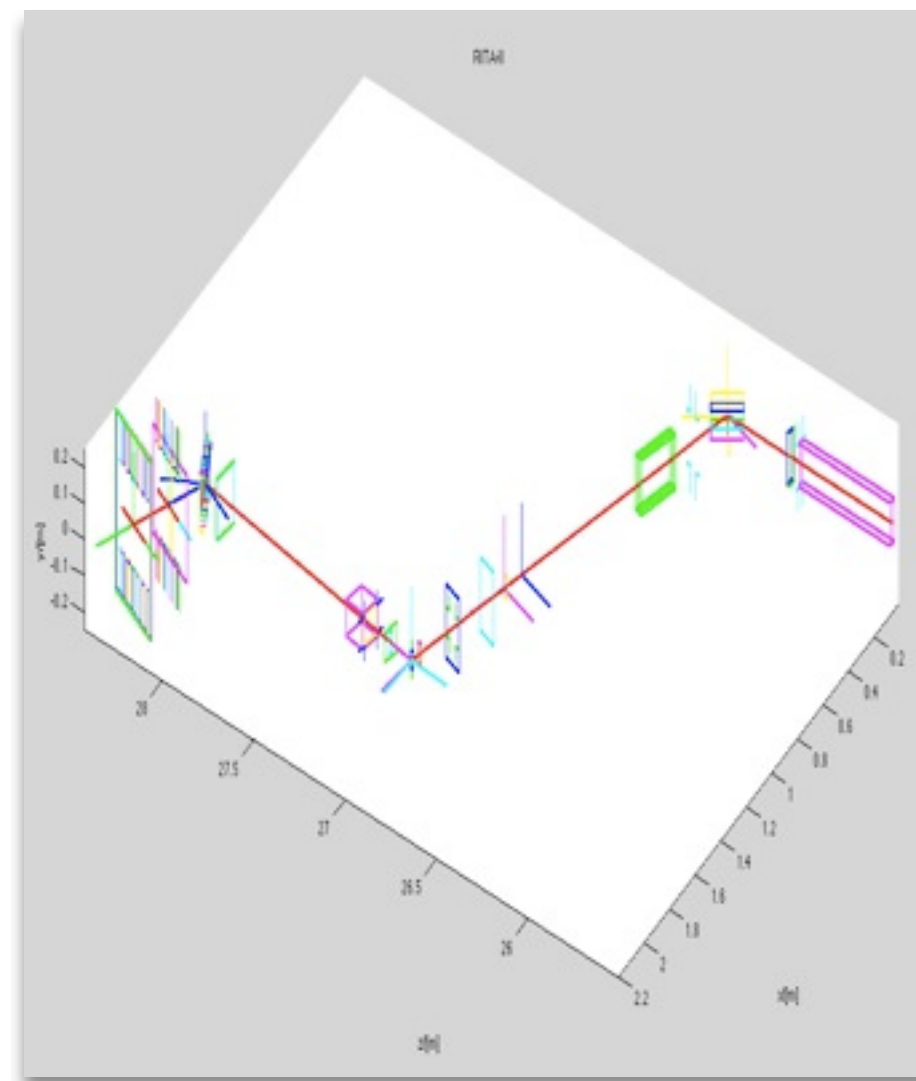
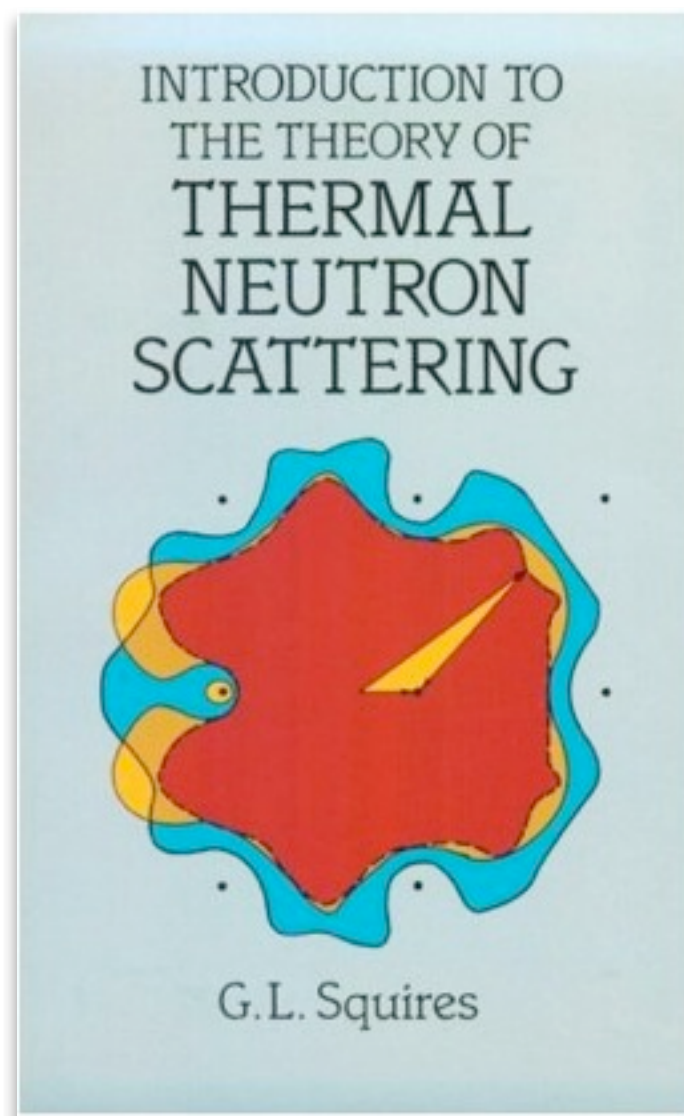


P. Tregenna-Piggott, PSI



# Teaching / training purposes

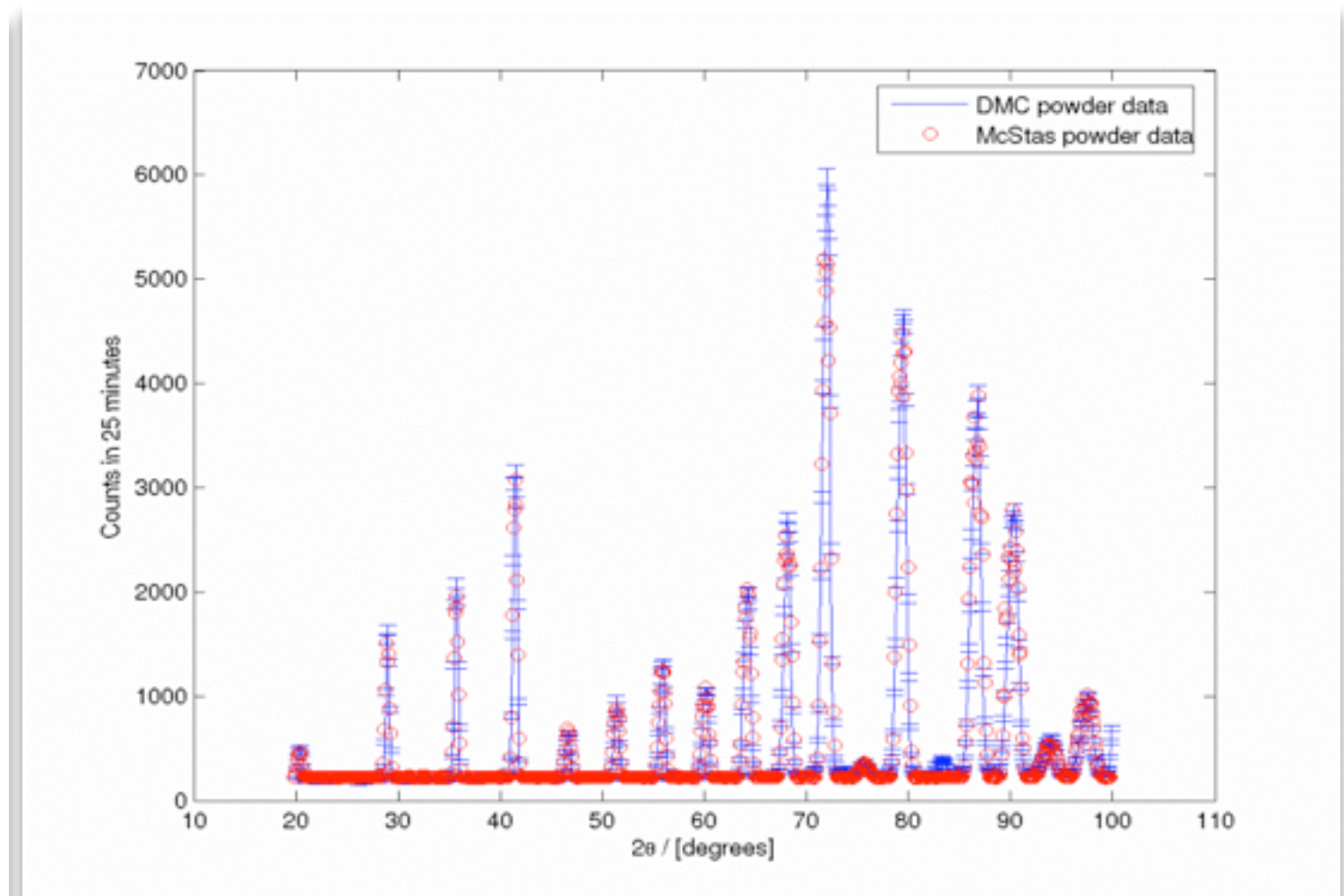
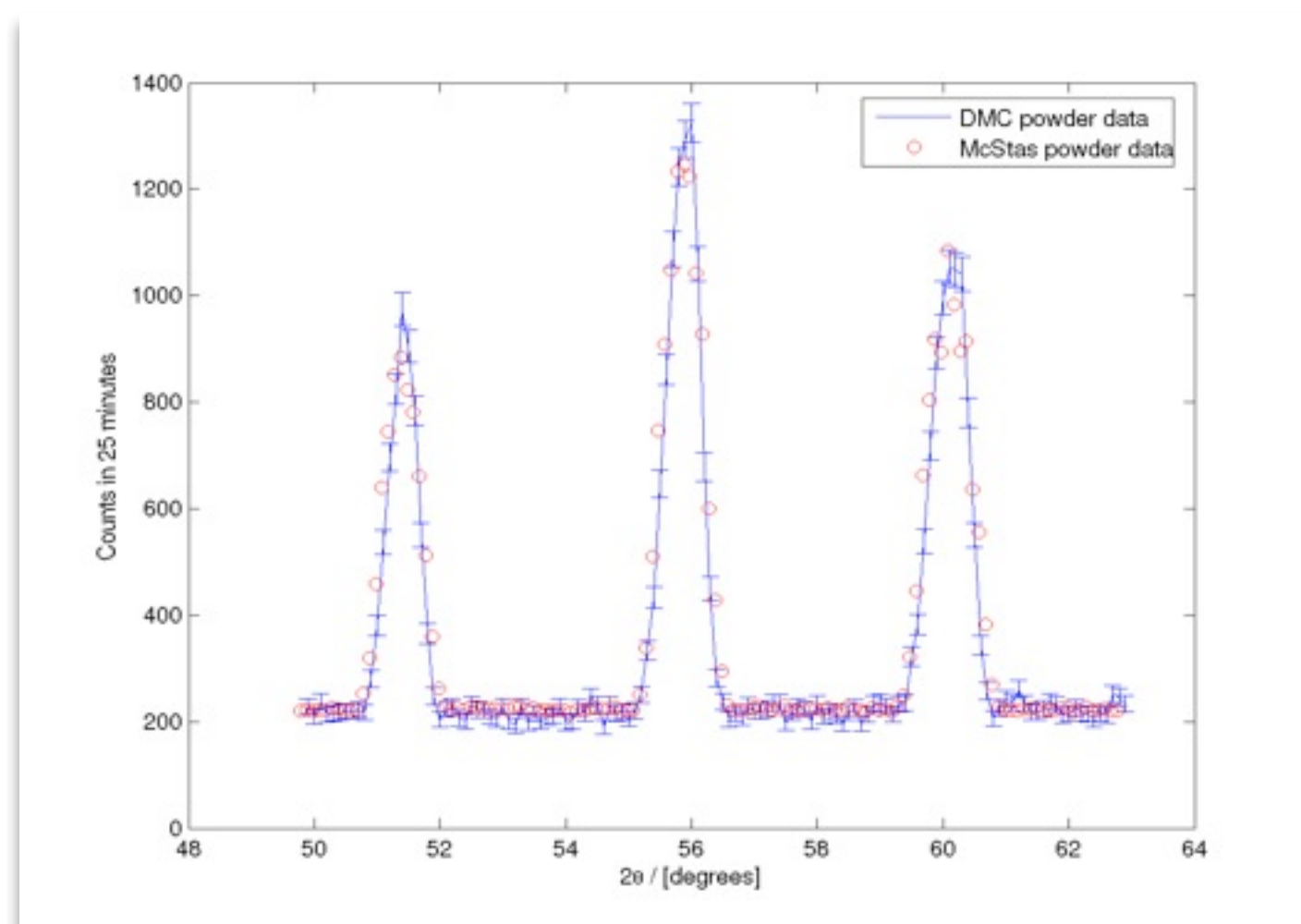
- Workshops (like this one!)
- Teaching
  - University of Copenhagen course on Neutron Scattering





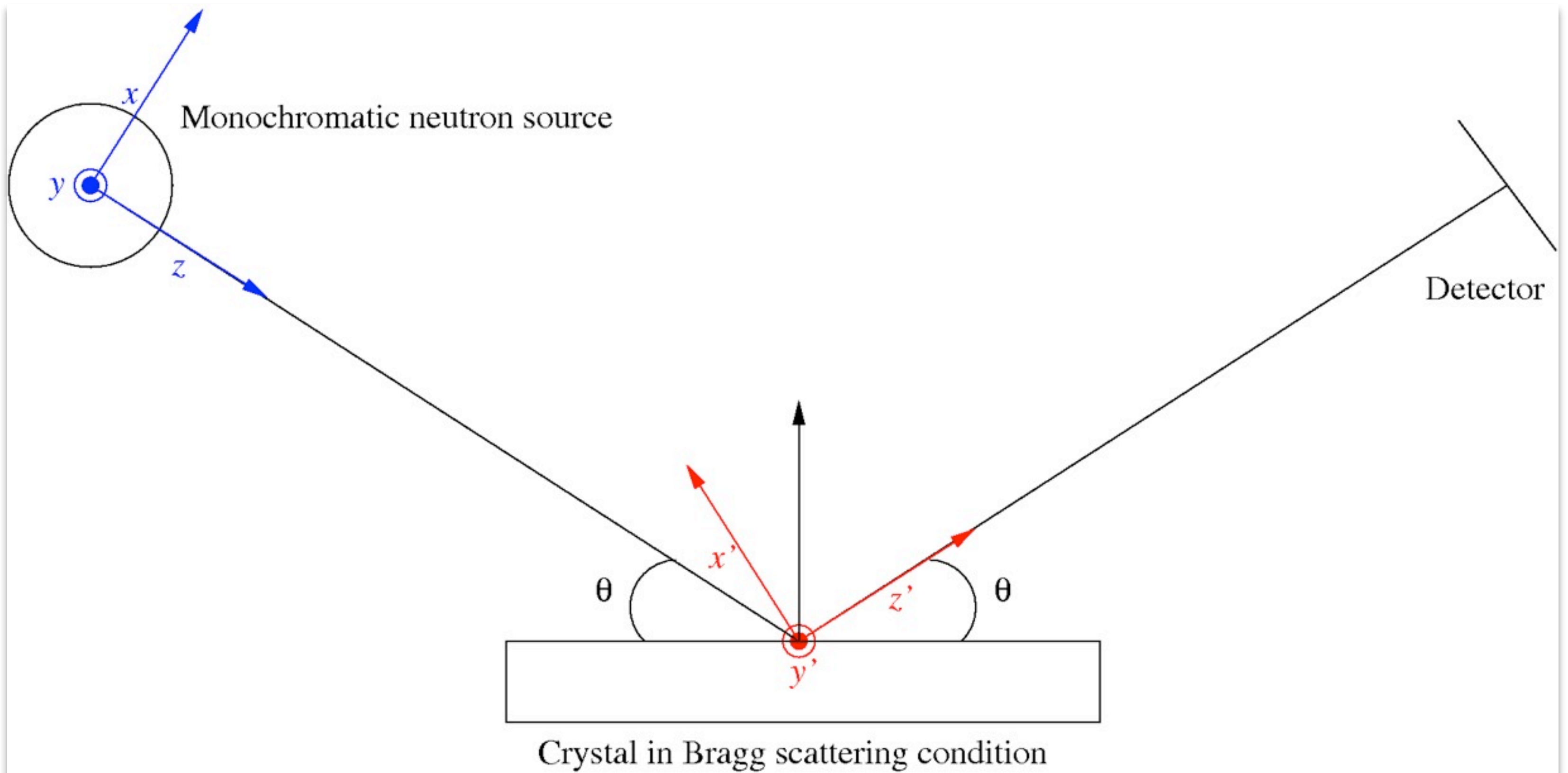
# Reliability - cross comparisons

- Much effort has gone into this
- Here: simulations vs. exp. at powder diffract. DMC, PSI
- The bottom line is
- McStas agree very well with other packages (NISP, VitESS, IDEAS, RESTRAX, ...)
- Experimental line shapes are within 5%
- Absolute intensities are within 10-30%
- Common understanding: McStas is reliable

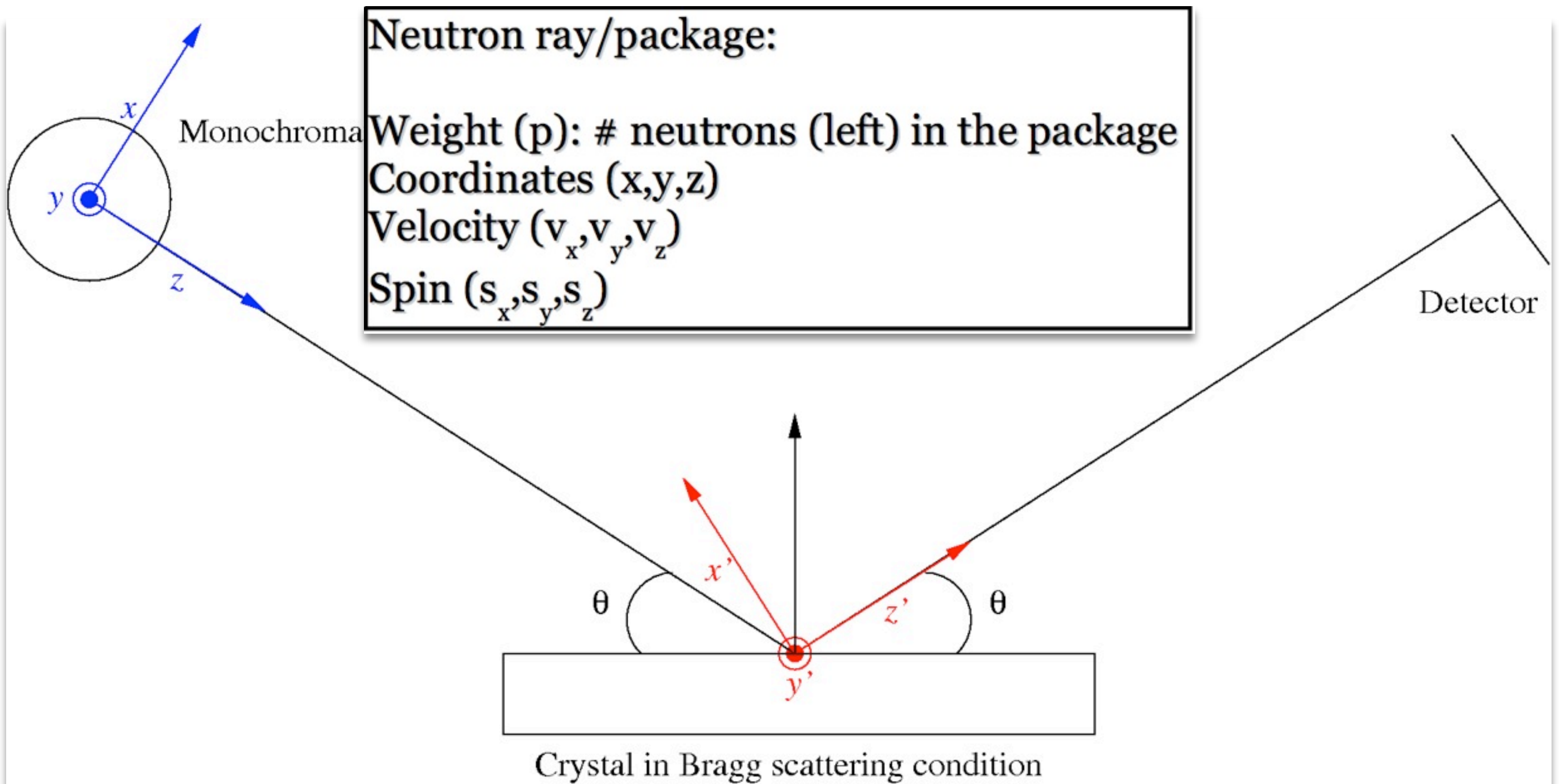


P. Willendrup, Risø DTU; Uwe Filges, L. Keller, PSI

# McStas: key concepts

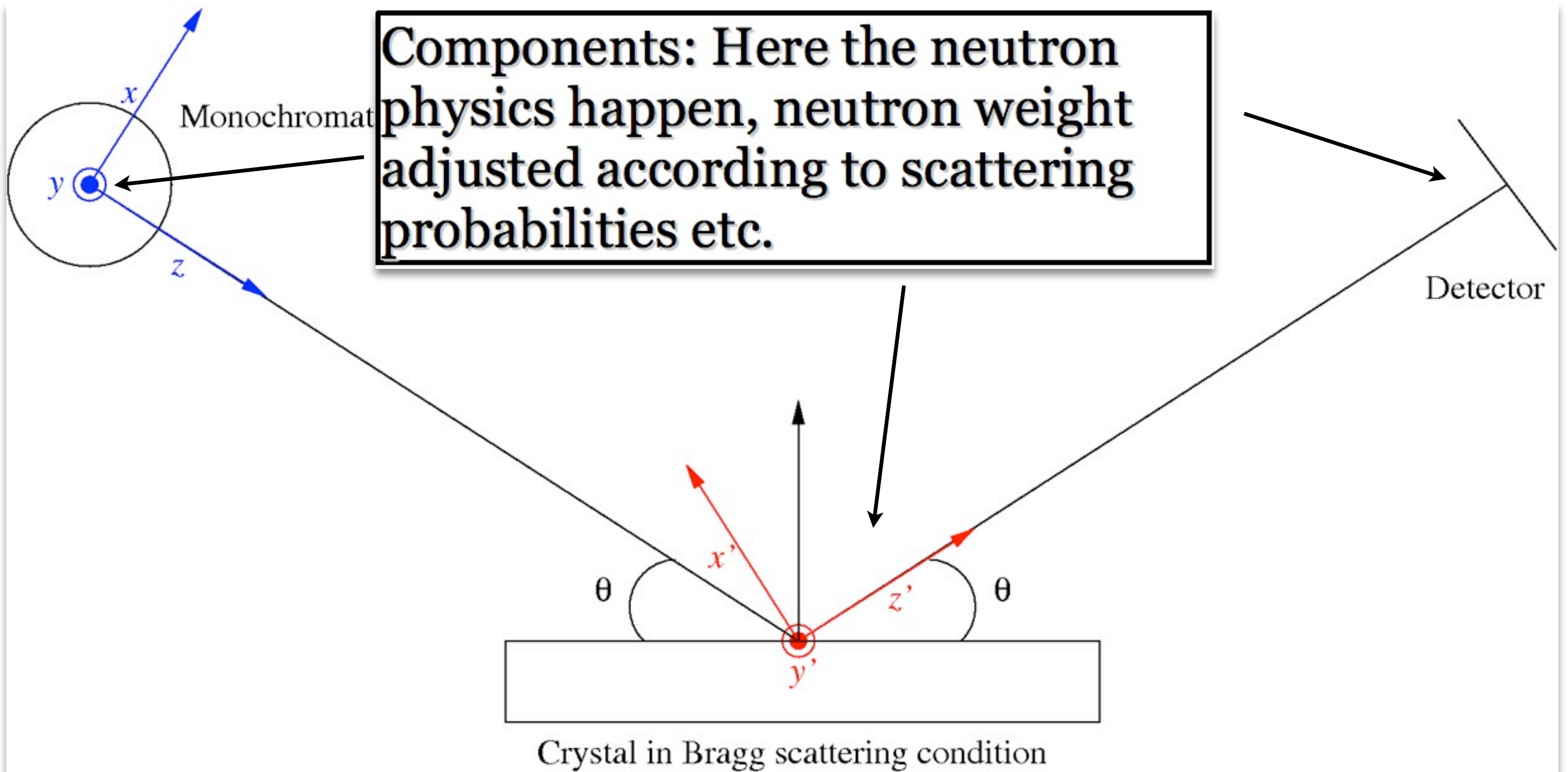


# McStas: key concepts

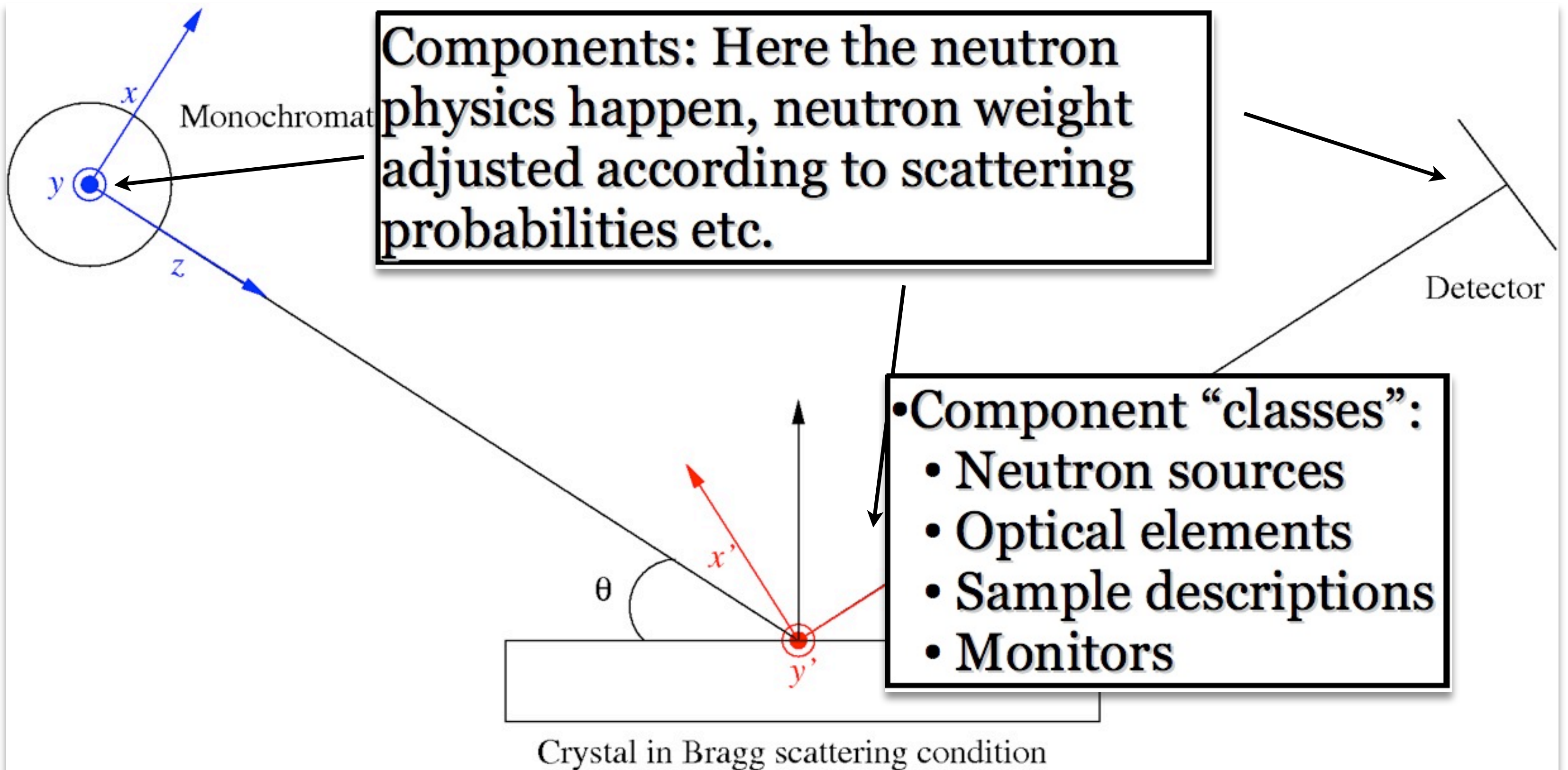




# McStas: key concepts

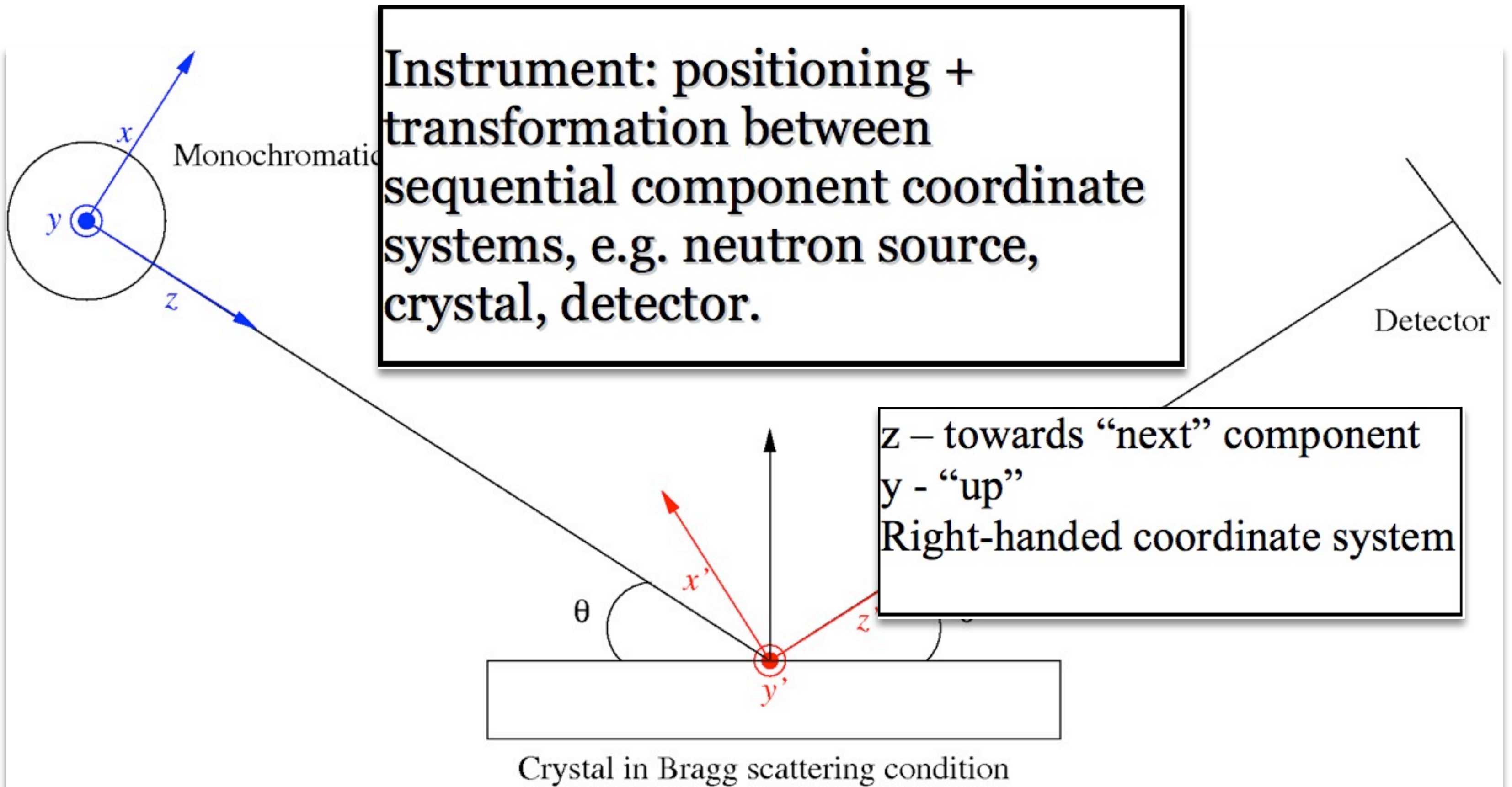


# McStas: key concepts





# McStas: key concepts





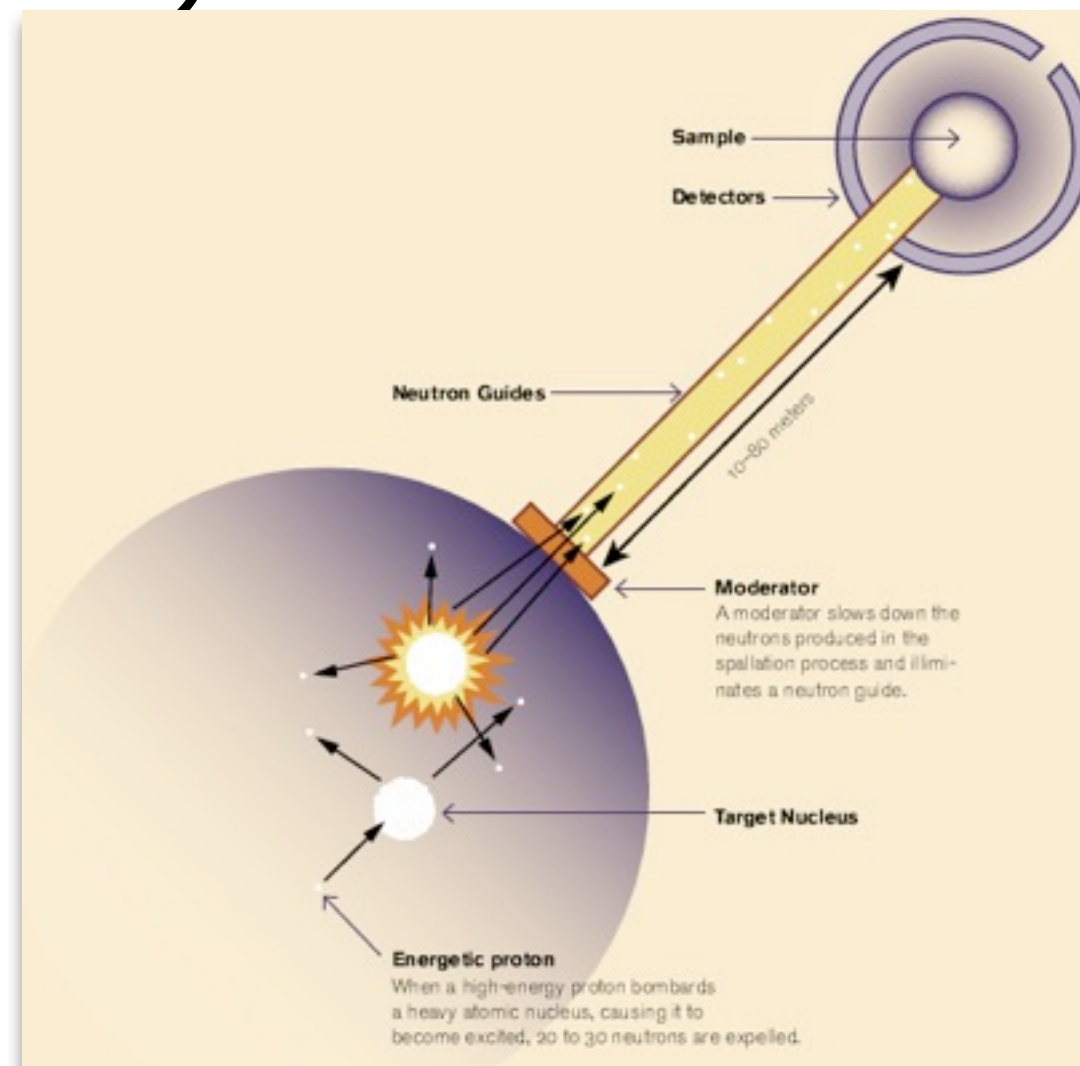
# McStas overview

- Portable code (Unix/Linux/Mac/Win32)



- 'Component' files ( $\sim 100$ ) inserted from library

- Sources
- Optics
- Samples
- Monitors
- If needed, write your own comps



# Implementation

- Three levels of source code:
  - Instrument file (All users)
  - Component files (Some users)
  - ANSI c code (no users)



# Instrument file

```
DEFINE INSTRUMENT My_Instrument(DIST=10)

/* Here comes the TRACE section, where the actual      */
/* instrument is defined as a sequence of components.  */
TRACE

/* The Arm() class component defines reference points and orientations */
/* in 3D space.                                         */
COMPONENT Origin = Arm()
  AT (0, 0, 0) ABSOLUTE

COMPONENT Source = Source_simple(
  radius = 0.1, dist = 10, xw = 0.1, yh = 0.1, E0 = 5, dE = 1)
  AT (0, 0, 0) RELATIVE Origin

COMPONENT Emon = E_monitor(
  filename = "Emon.dat", xmin = -0.1, xmax = 0.1, ymin = -0.1,
  ymax = 0.1, Emin = 0, Emax = 10)
  AT (0, 0, DIST) RELATIVE Origin

COMPONENT PSD = PSD_monitor(
  nx = 128, ny = 128, filename = "PSD.dat", xmin = -0.1,
  xmax = 0.1, ymin = -0.1, ymax = 0.1)
  AT (0, 0, 1e-10) RELATIVE Emon

/* The END token marks the instrument definition end */
END
```

Written by you!

# Component file

```
*****
*
* Mcstas, neutron ray-tracing package
* Copyright 1997-2002, All rights reserved
* Risoe National Laboratory, Roskilde, Denmark
* Institut Laue Langevin, Grenoble, France
*
* Component: Source_flat
*
* %I
* Written by: Kim Lefmann
* Date: October 30, 1997
* Modified by: KL, October 4, 2001
* Modified by: Emmanuel Farhi, October 30, 2001. Serious bug corrected.
* Version: $Revision: 1.22 $
* Origin: Risoe
* Release: McStas 1.6
*
* A circular neutron source with flat energy spectrum and arbitrary flux
*
* %D
* The routine is a circular neutron source, which aims at a square target
* centered at the beam (in order to improve MC-acceptance rate). The angular
* divergence is then given by the dimensions of the target.
* The neutron energy is uniformly distributed between E0-dE and E0+dE.
*
* Example: Source_flat(radius=0.1, dist=2, xw=.1, yh=.1, E0=14, dE=2)
*
* %P
* radius: (m) Radius of circle in (x,y,0) plane where neutrons
* are generated.
* dist: (m) Distance to target along z axis.
* xw: (m) Width(x) of target
* yh: (m) Height(y) of target
* E0: (meV) Mean energy of neutrons.
* dE: (meV) Energy spread of neutrons.
* Lambda0 (AA) Mean wavelength of neutrons.
* dLambda (AA) Wavelength spread of neutrons.
* flux (1/(s*cm**2*st)) Energy integrated flux
*
* %E
*****/

DEFINE COMPONENT Source_simple
DEFINITION PARAMETERS ()
SETTING PARAMETERS (radius, dist, xw, yh, E0=0, dE=0, Lambda0=0, dLambda=0, flux=1)
OUTPUT PARAMETERS ()
STATE PARAMETERS (x, y, z, vx, vy, vz, t, s1, s2, p)
DECLARE
%{
double pmul, pdir;
%}
INITIALIZE
%{
pmul=flux*PI*1e4*radius*radius/mcget_ncount();
%}
Z-1
```

```
TRACE
%{
double chi,E,Lambda,v,r, xf, yf, rf, dx, dy;

t=0;
z=0;

chi=2*PI*rand01(); /* Choose point on source */
r=sqrt(rand01()*radius); /* with uniform distribution. */
x=r*cos(chi);
y=r*sin(chi);
randvec_target_rect(&xf, &yf, &rf, &pdir,
0, 0, dist, xw, yh, ROT_A_CURRENT_COMP);

dx = xf-x;
dy = yf-y;
rf = sqrt(dx*dx+dy*dy+dist*dist);

p = pdir*pmul;

if(Lambda0==0) {
E=E0+dE*randpml(); /* Choose from uniform distribution */
v=sqrt(E)*SE2V;
} else {
Lambda=Lambda0+dLambda*randpml();
v = K2V*(2*PI/Lambda);
}

vz=v*dist/rf;
vy=v*dy/rf;
vx=v*dx/rf;
%}

MCDISPLAY
%{
magnify("xy");
circle("xy", 0, 0, 0, radius);
%}

END
```

Written by developers  
and possibly you!





# Generated c-code

```
/* Automatically generated file. Do not edit.
 * Format:      ANSI C source code
 * Creator:     McStas <http://neutron.risoe.dk>
 * Instrument:  My_Instrument.instr (My Instrument)
 * Date:       Sat Apr  9 15:27:56 2005
 */

/* THOUSANDS of lines removed here.... */

/* TRACE Component Source. */
SIG_MESSAGE("Source (Trace)");
mcDEBUG_COMP("Source")
mccoordschange(mccposrSource, mcrotrSource,
  &mcnlx, &mcnly, &mcnlz,
  &mcnlvx, &mcnlvy, &mcnlvz,
  &mcnlt, &mcnlxs, &mcnlisy);
mcDEBUG_STATE(mcnlx, mcnly, mcnlz, mcnlvx, mcnlvy, mcnlvz, mcnlt, mcnlxs, mcnlisy, mcnlp)
#define x mcnlx
#define y mcnly
#define z mcnlz
#define vx mcnlvx
#define vy mcnlvy
#define vz mcnlvz
#define t mcnlt
#define s1 mcnlxs
#define s2 mcnlisy
#define p mcnlp
STORE_NEUTRON(2, mcnlx, mcnly, mcnlz, mcnlvx, mcnlvy, mcnlvz, mcnlt, mcnlxs, mcnlisy, mcnlsz, mcnlp);
mcScattered=0;
mcNCounter[2]++;
#define mcompcurname Source
#define mcompcurindex 2
{ /* Declarations of SETTING parameters. */
MCNUM radius = mccSource_radius;
MCNUM dist = mccSource_dist;
MCNUM xw = mccSource_xw;
MCNUM yh = mccSource_yh;
MCNUM E0 = mccSource_E0;
MCNUM dE = mccSource_dE;
MCNUM Lambda0 = mccSource_Lambda0;
MCNUM dLambda = mccSource_dLambda;
MCNUM flux = mccSource_flux;
#line 58 "Source_simple.comp"
{
double chi, E, Lambda, v, r, xf, yf, rf, dx, dy;

t=0;
z=0;

chi=2*PI*rand01(); /* Choose point on source */
r=sqrt(rand01()*radius); /* with uniform distribution. */
x=r*cos(chi);
y=r*sin(chi);

randvec_target_rect(&xf, &yf, &rf, &pdire,
  0, 0, dist, xw, yh, ROT_A_CURRENT_COMP);
```

Written by mcstas!

McStas is a (pre)compiler!

Input is .comp and .instr files +  
runtime functions for e.g. random  
numbers

Output is a single c-file, which can  
be compiled using e.g. gcc.

Can take input arguments if  
needed.





# McStas overview

The screenshot displays the McStas software interface with several windows open:

- McStas: h8\_test.instr**: The main application window showing the instrument file, simulation results, and status. The status is "Done". It lists various components like Monochromator, Detectors (D0 to D10), and a He3H detector.
- Run simulation h8\_test.instr**: A dialog box for running the simulation. It includes fields for "Instrument source" (h8\_test.instr), "Instrument parameters" (Lambda (D): 2.36), "Output to (dir):", "Neutron count" (1000000), "gravity (BEWARE)", "Random seed", "Clustering" (None (single CPU)), and "Number of nodes" (2). There are buttons for "Simulate", "Plot results, Format: PGPLOT", and "Start".
- mcdisplay controls**: A window for controlling the 3D display of the instrument. It shows a 3D model of the instrument components in a coordinate system with axes labeled z/[m].
- PGPLOT Window 1**: A window displaying a grid of 12 heatmaps, which are 2D intensity distributions at different detector positions.
- TOF diagram: Xdiag.out**: A plot showing the time-of-flight (TOF) distribution. The y-axis is labeled "Z Axis [m]" and the x-axis is labeled "TOF [ms]". The plot shows several distinct peaks of intensity.



# McStas overview

The screenshot displays the McStas software interface. The main window, titled "McStas: h8\_test.instr", shows the instrument file and simulation results. A "Run simulation h8\_test.instr" dialog box is open, allowing users to set parameters such as "Instrument source: h8\_test.instr", "Instrument parameters (D-floating point, I-integer, S-string)", "Lambda (D): 2.36", "Output to (dir):", "Neutron count: 1000000", "gravity (BEWARE)", "Random seed:", "# steps: 0", and "Plot results, Format: PGPLOT". A "Let's see it run!" text box is overlaid on the dialog. The background shows a 3D visualization of the instrument components and a "PGPLOT Window 1" displaying a grid of heatmaps. A "TOF diagram: Xtalog.out" plot is also visible, showing neutron counts versus time-of-flight (TOF) and distance (z Axis [m]).