Welcome

Monday program:

13:00 Lunch
15:00 Introduction to McStas and VITESS, including demo
16:00 Afternoon coffee break
16:30 Introduction, continued
18:00 / 18:30 Dinner
19:00 / 19:30 Installation help

Practical information:

Wifi: Backafall / user: guest / password: password

Printer at ven2010\HP (Windows/Mac/CUPS)

Cabins available from after lunch / afternoon

McStas introduction

ICNX 2009 pre-workshop on McStas

Peter Willendrup[1]

Emmanuel Farhi[2]

Erik Knudsen[1]

Kim Lefmann[3]

[1]Materials Research Division, RISØ DTU, Roskilde, Denmark

[2]Scientific Computing, Institut Laue-Langevein (ILL), Grenoble, France

[3]Niels Bohr Institute, University of Copenhagen, Copenhagen, Denmark.

McStas project http://www.mcstas.org mcstas-users@mcstas.org

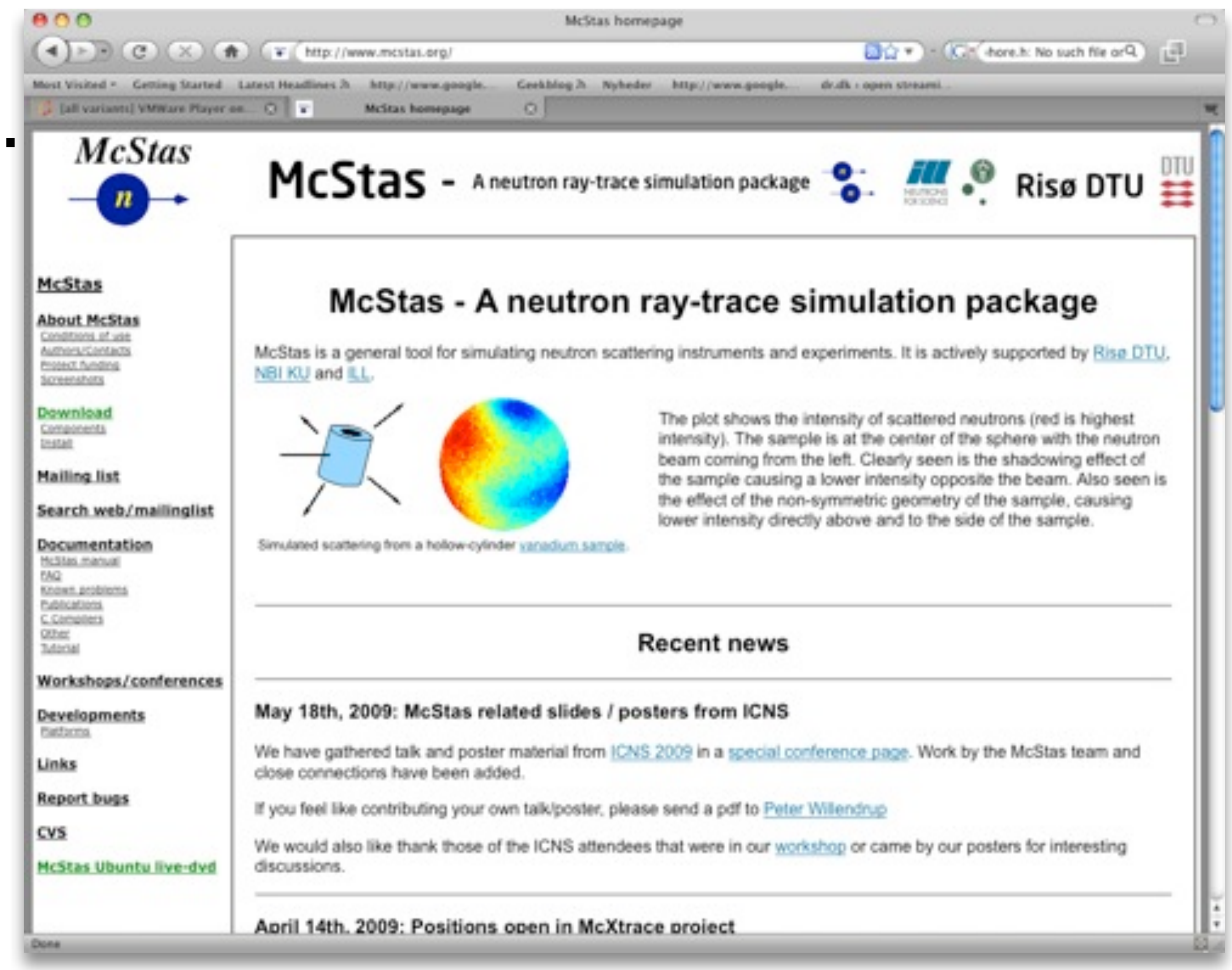
Risø DTU, Niels Bohr Institute, Institut Laue-Langevin

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 mailinglist

McStas Introduction

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

•Flexible, general simulation utility for neutron scattering experiments.

•Original

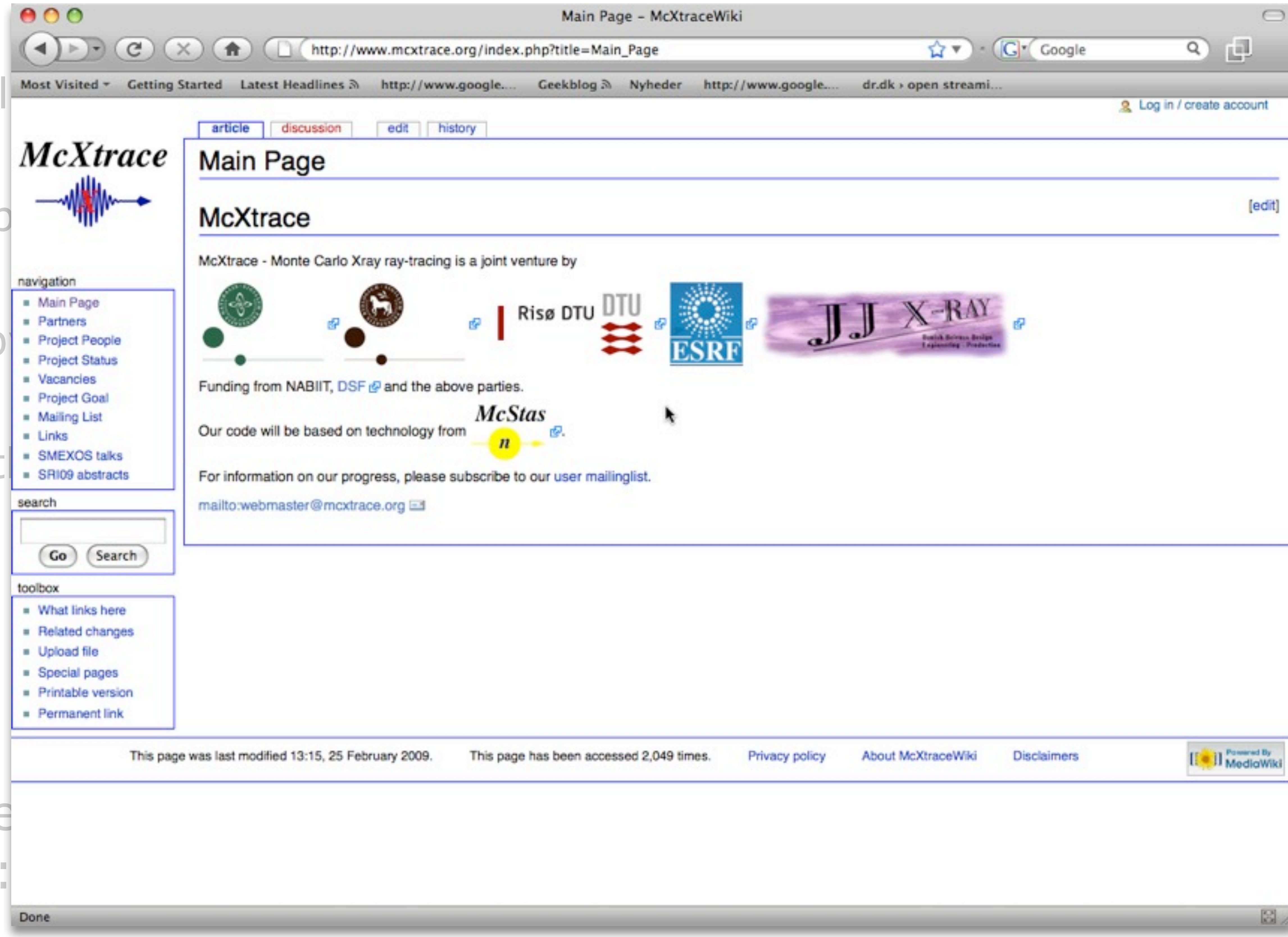
•Develop

•V. 1.0 b

•Current

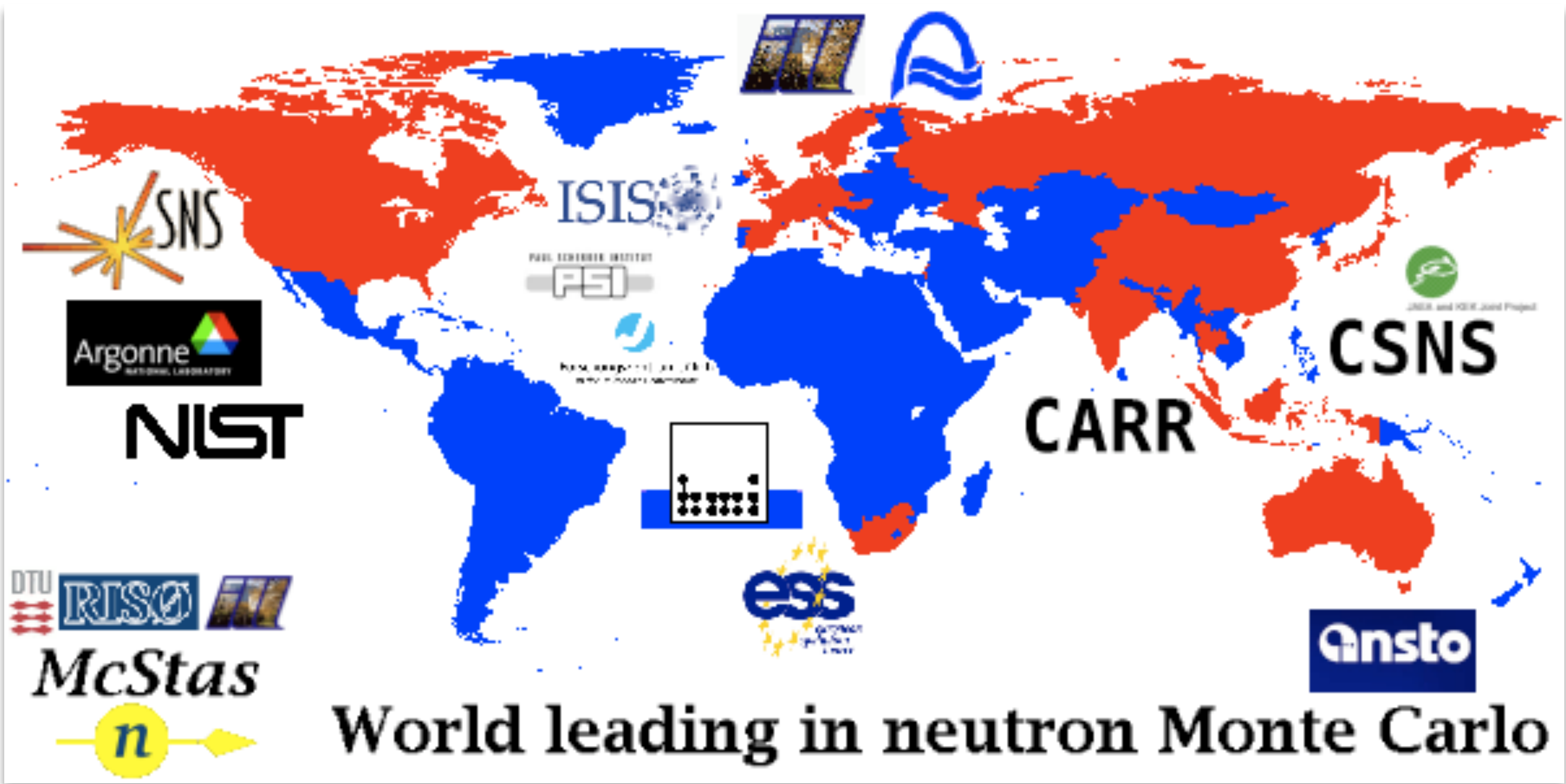
Proje
http:

list

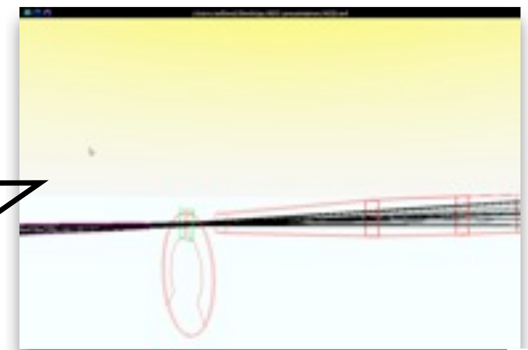
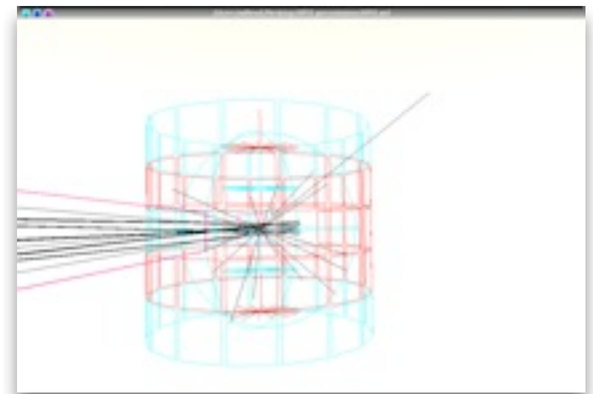


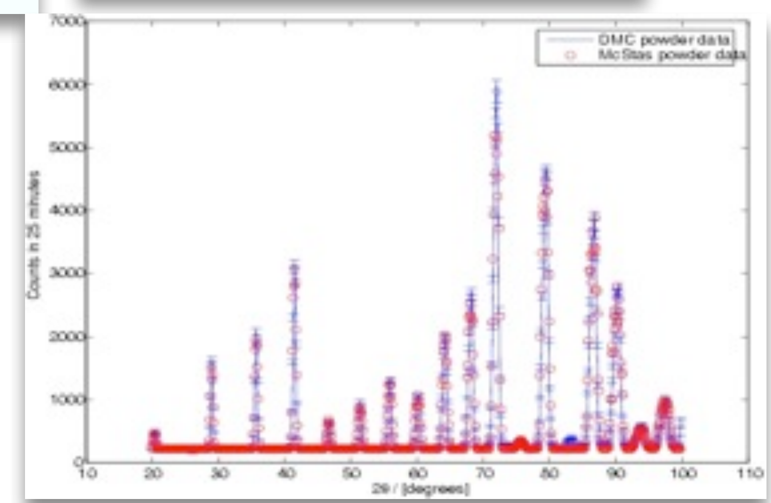
McStas Introduction

- Used at all major neutron sources



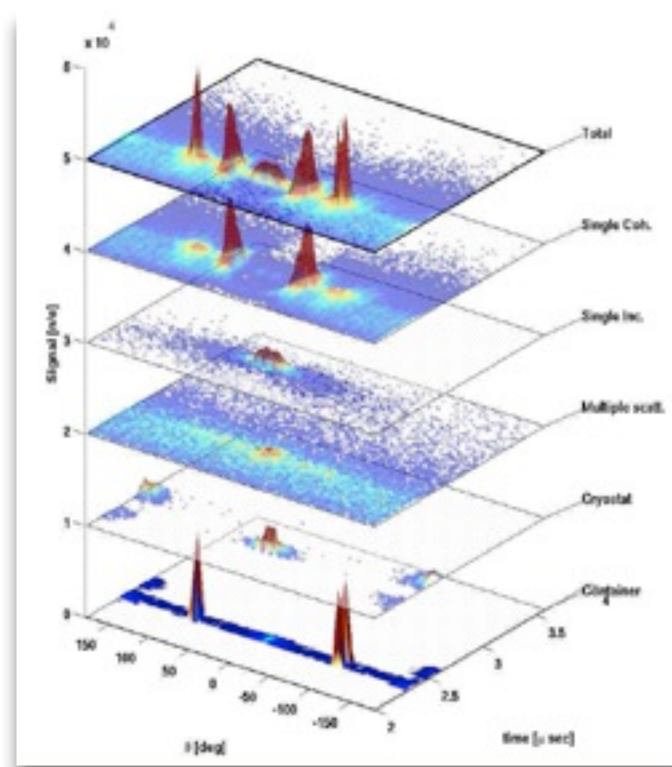
What is McStas used for?

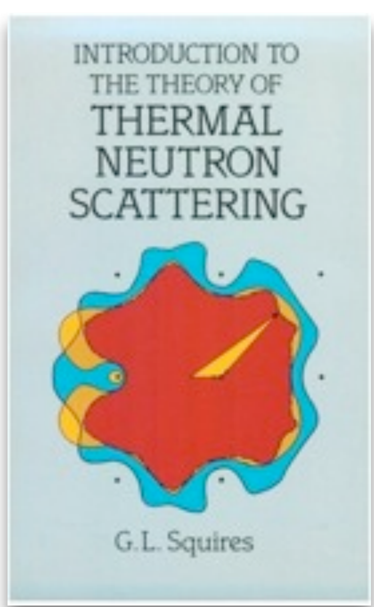



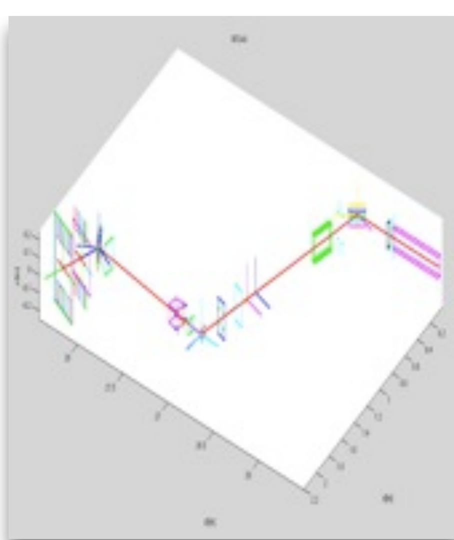


- Instrumentation
- Virtual experiments
- Data analysis
- Teaching

(KU 2005-2009)

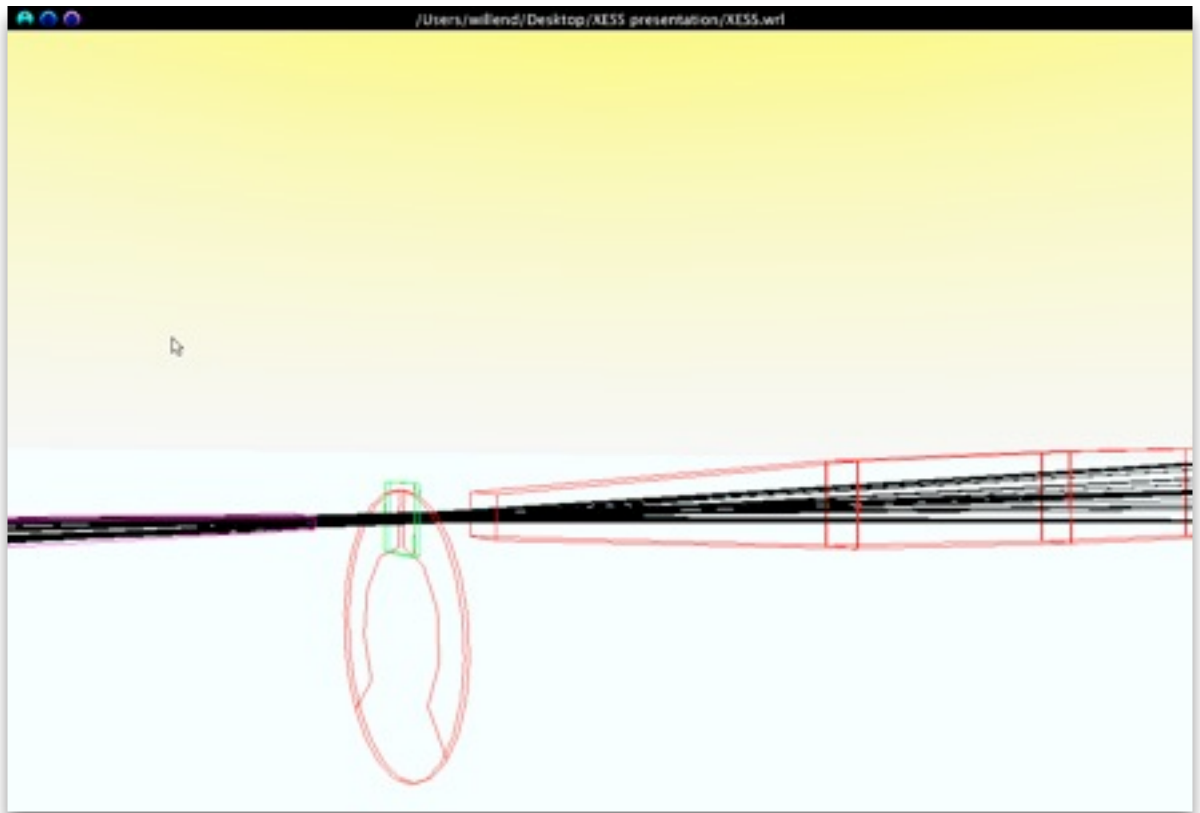


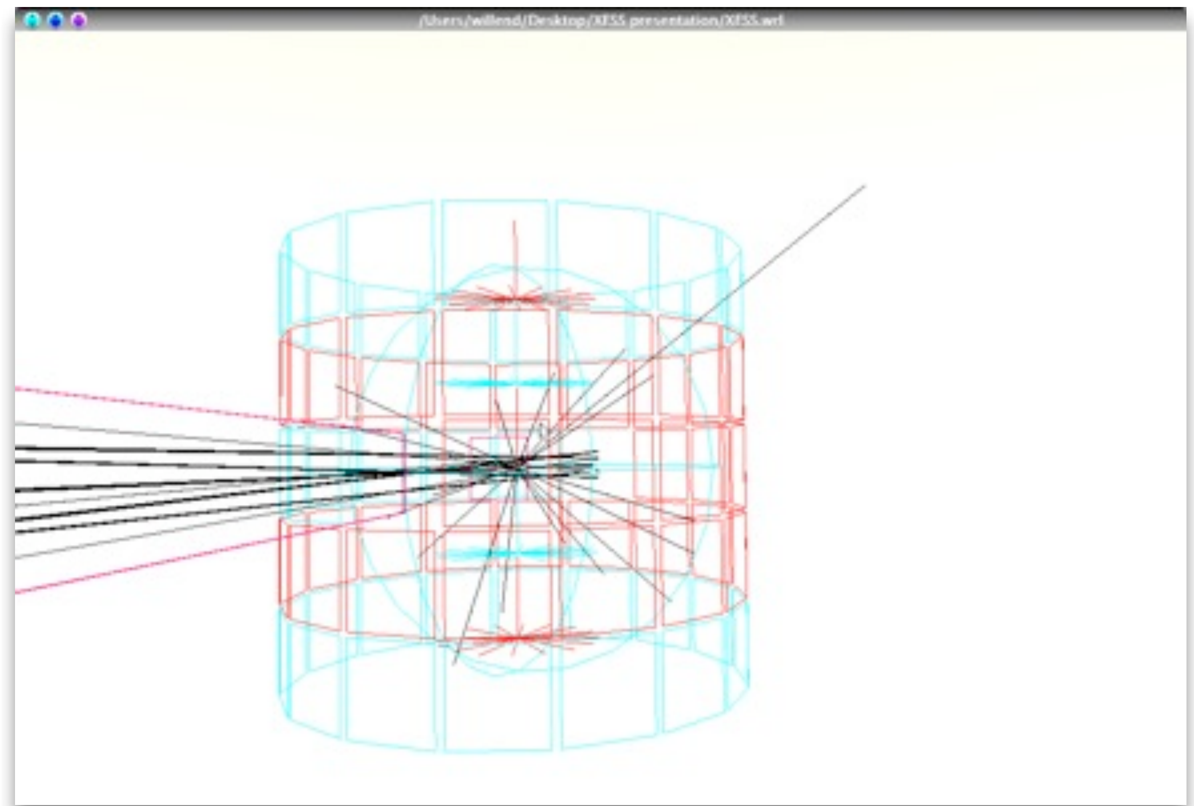


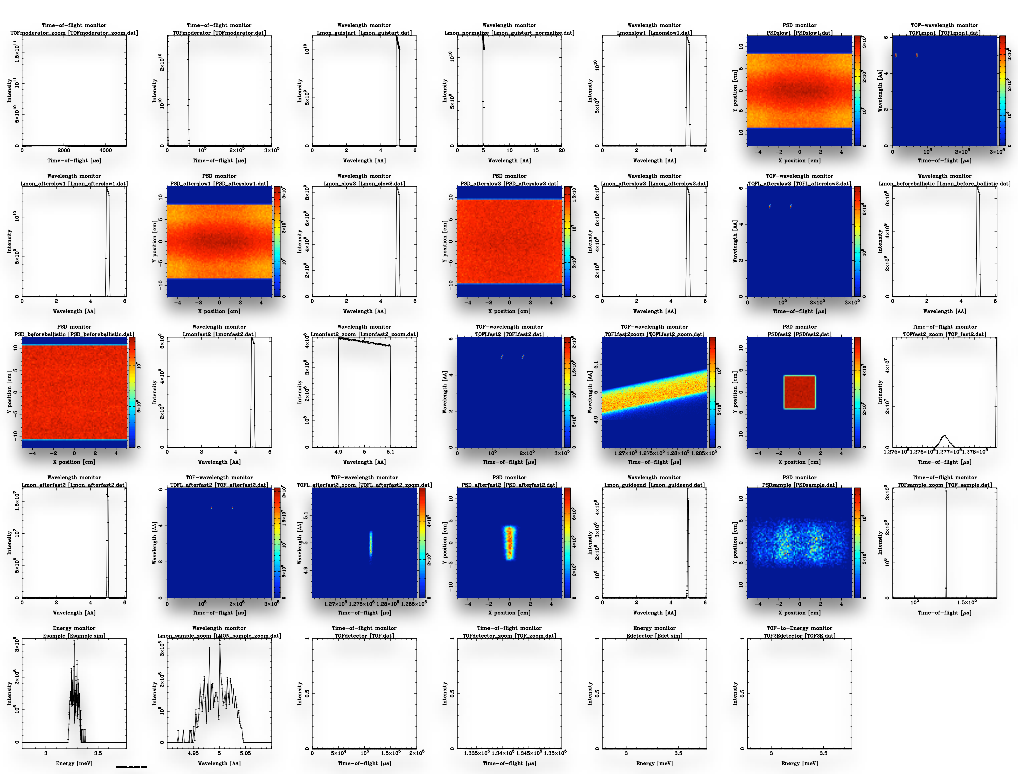


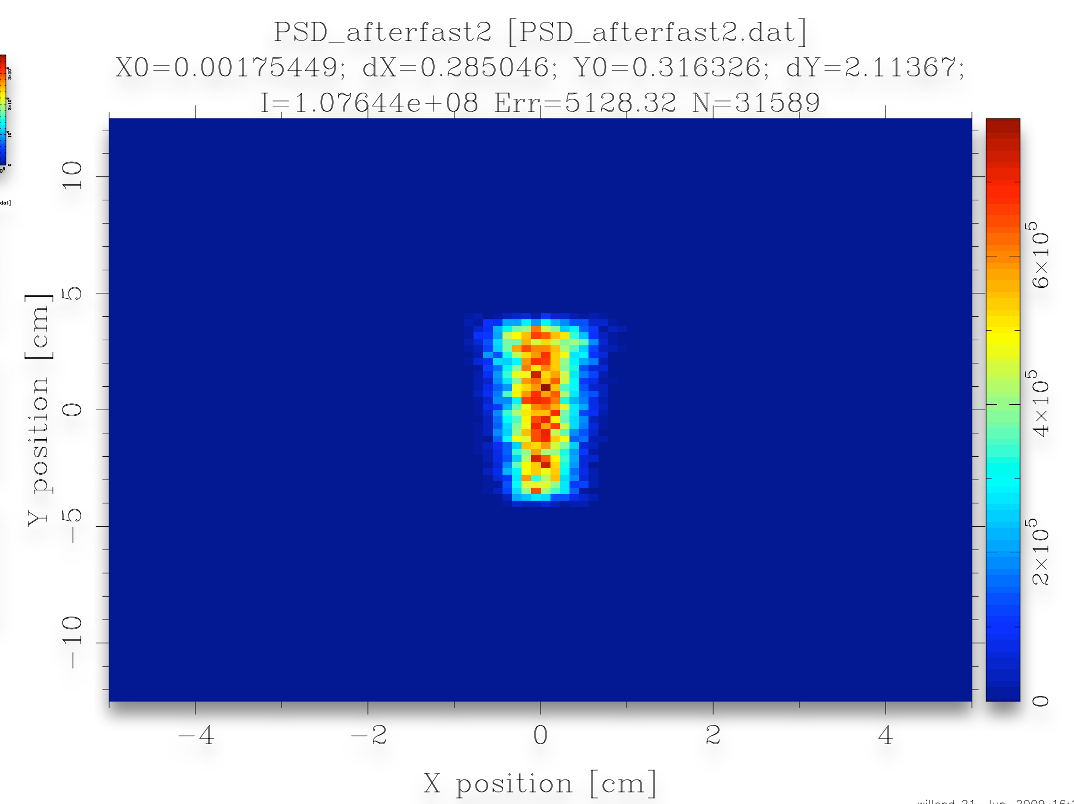

Instrumentation

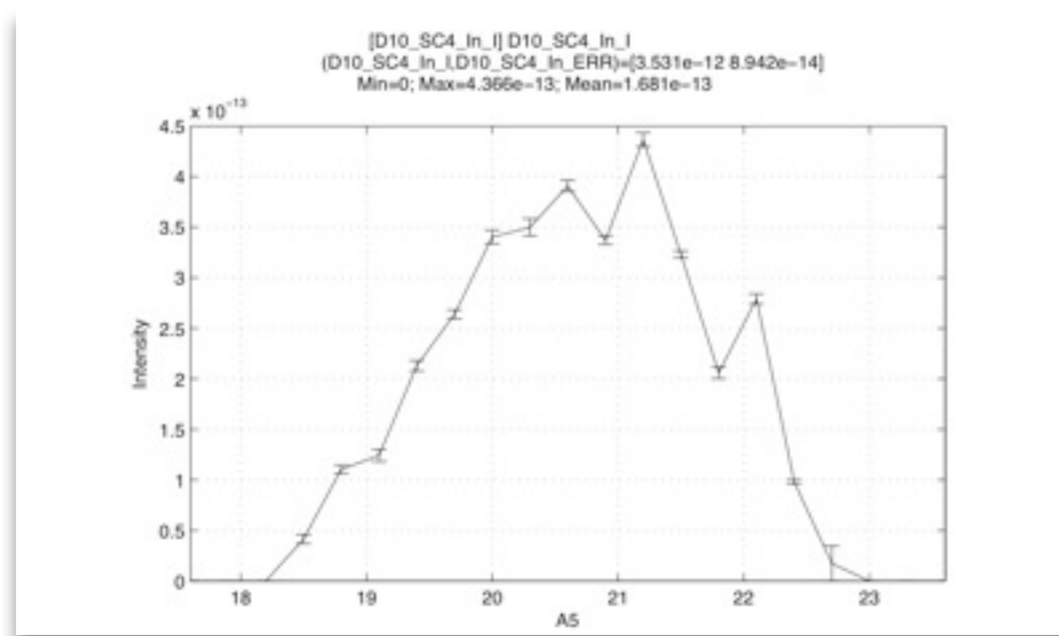
- Design and optimization of instruments











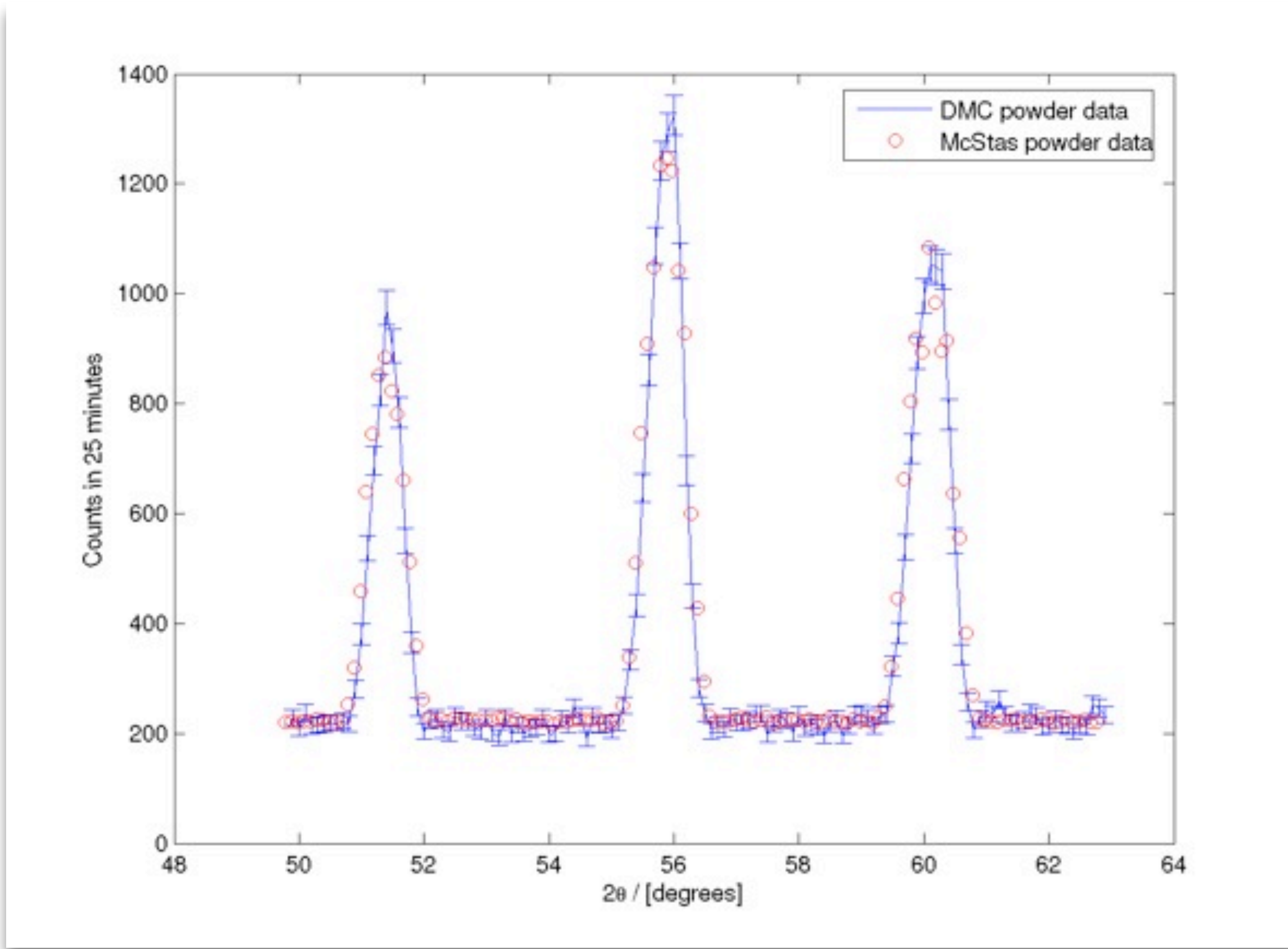
Risø DTU, Niels Bohr Institute, Institut Laue-Langevin

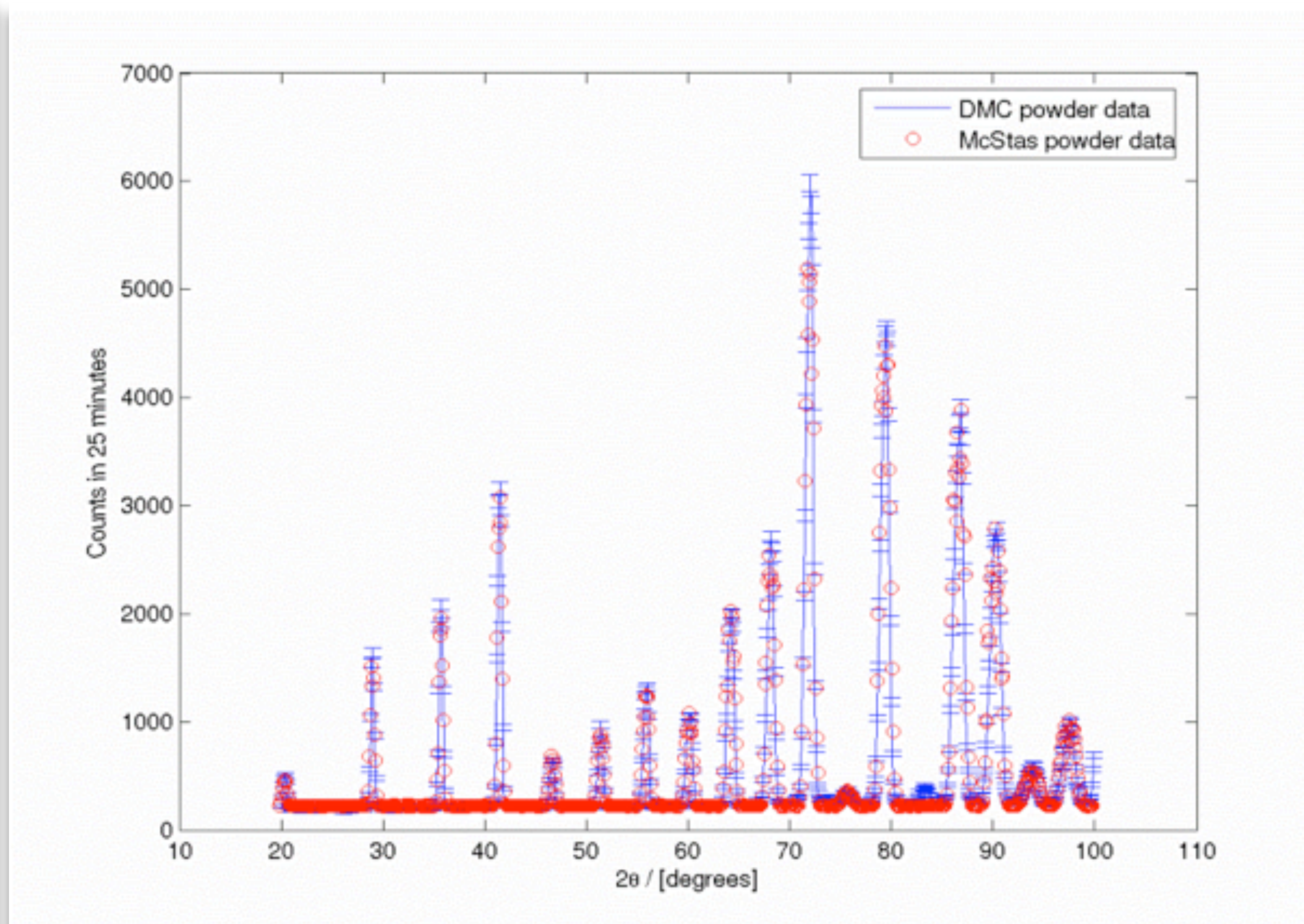
Virtual experiments (VE)

(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

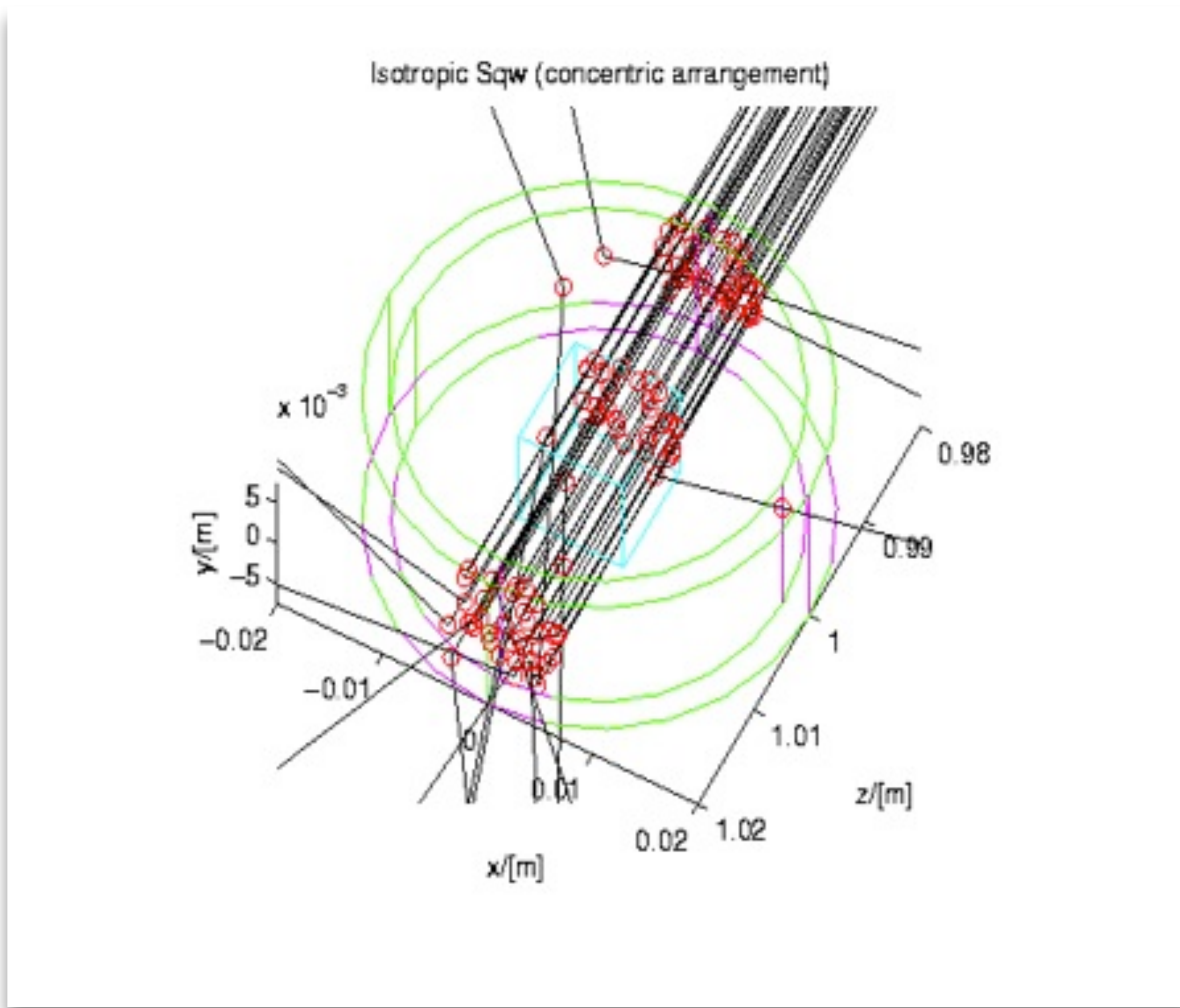


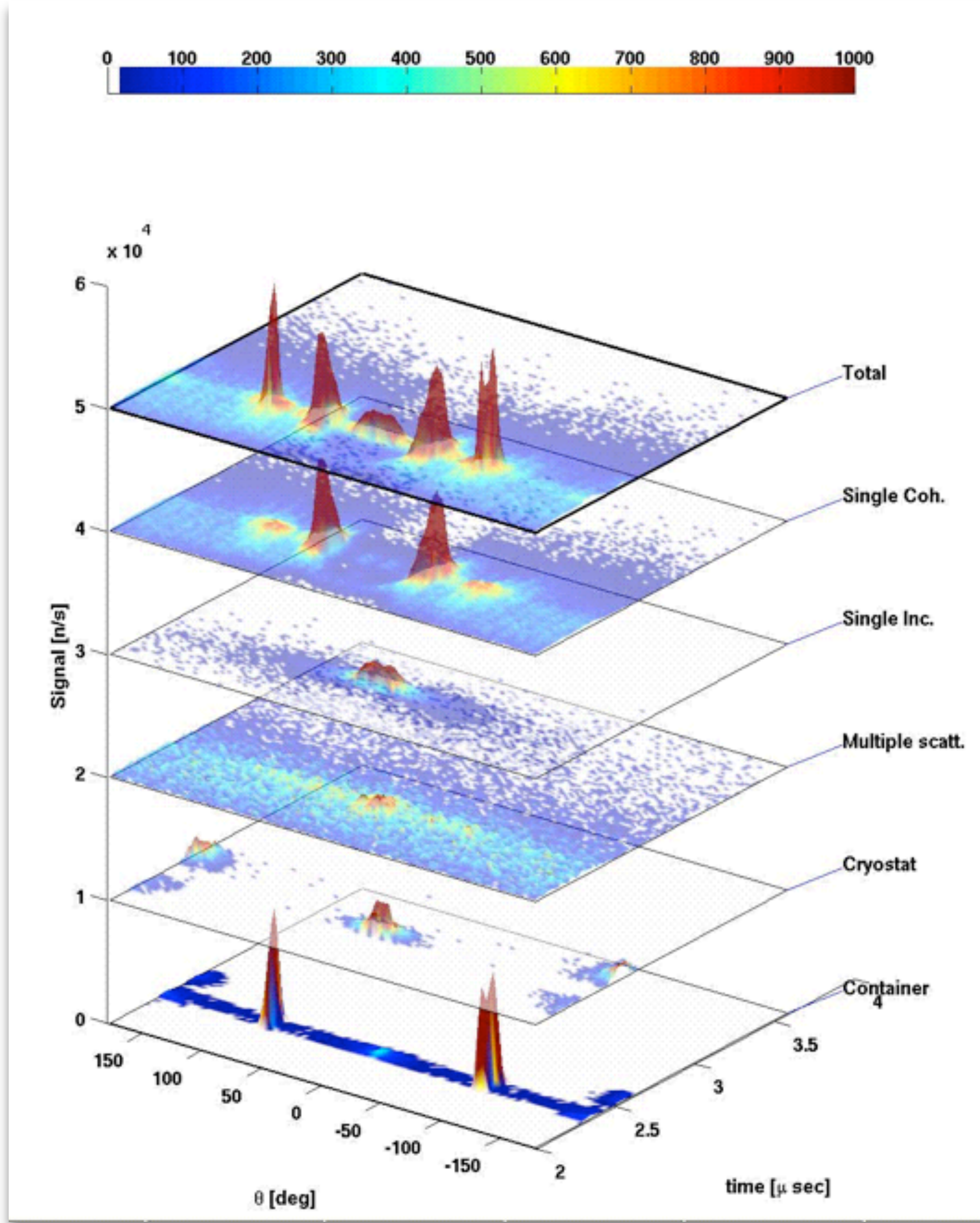


P. Willendrup, Risø DTU; Uwe Filges, L. Keller, PSI

Data analysis (1)
(using VE techiques)

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

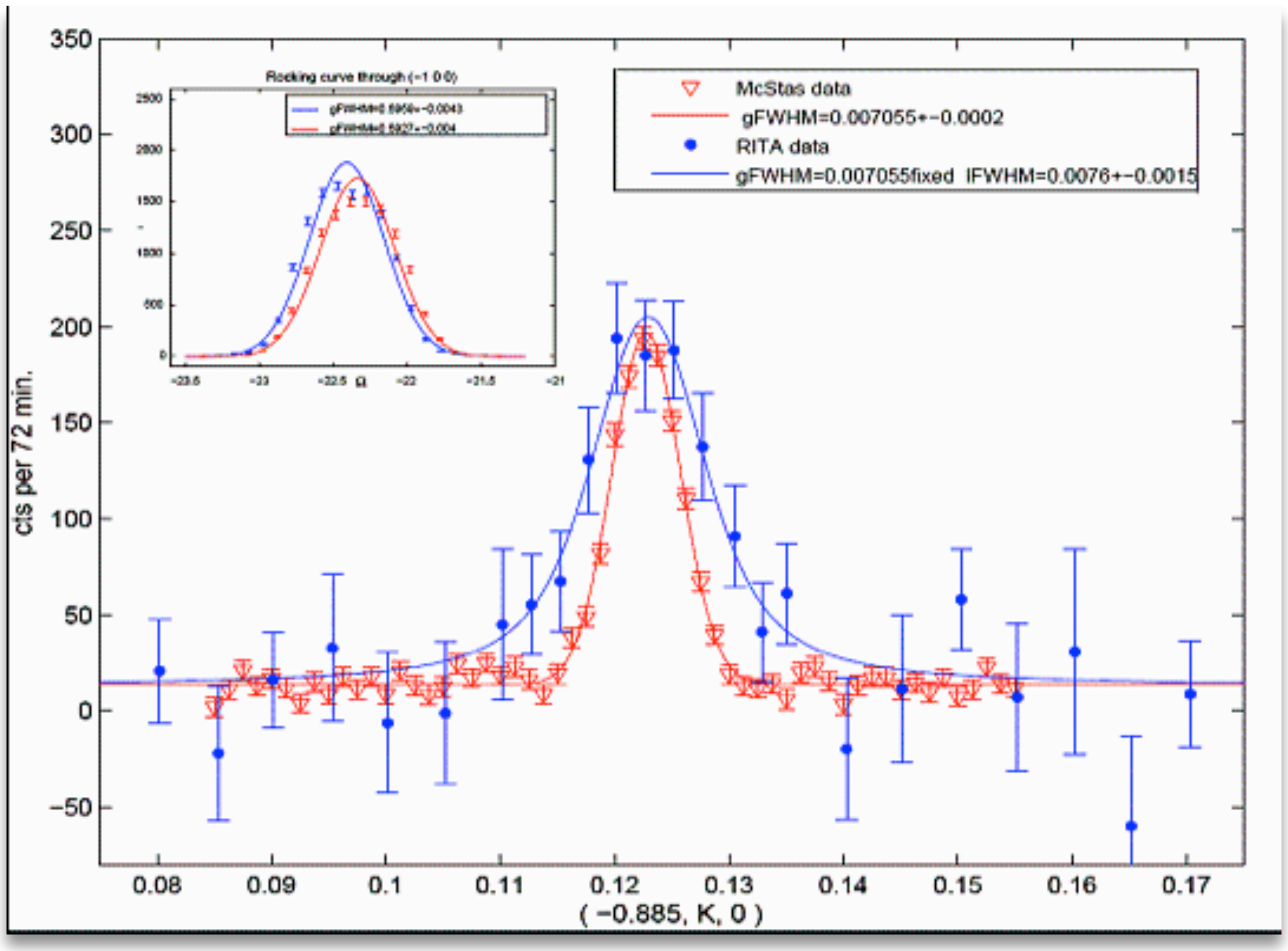




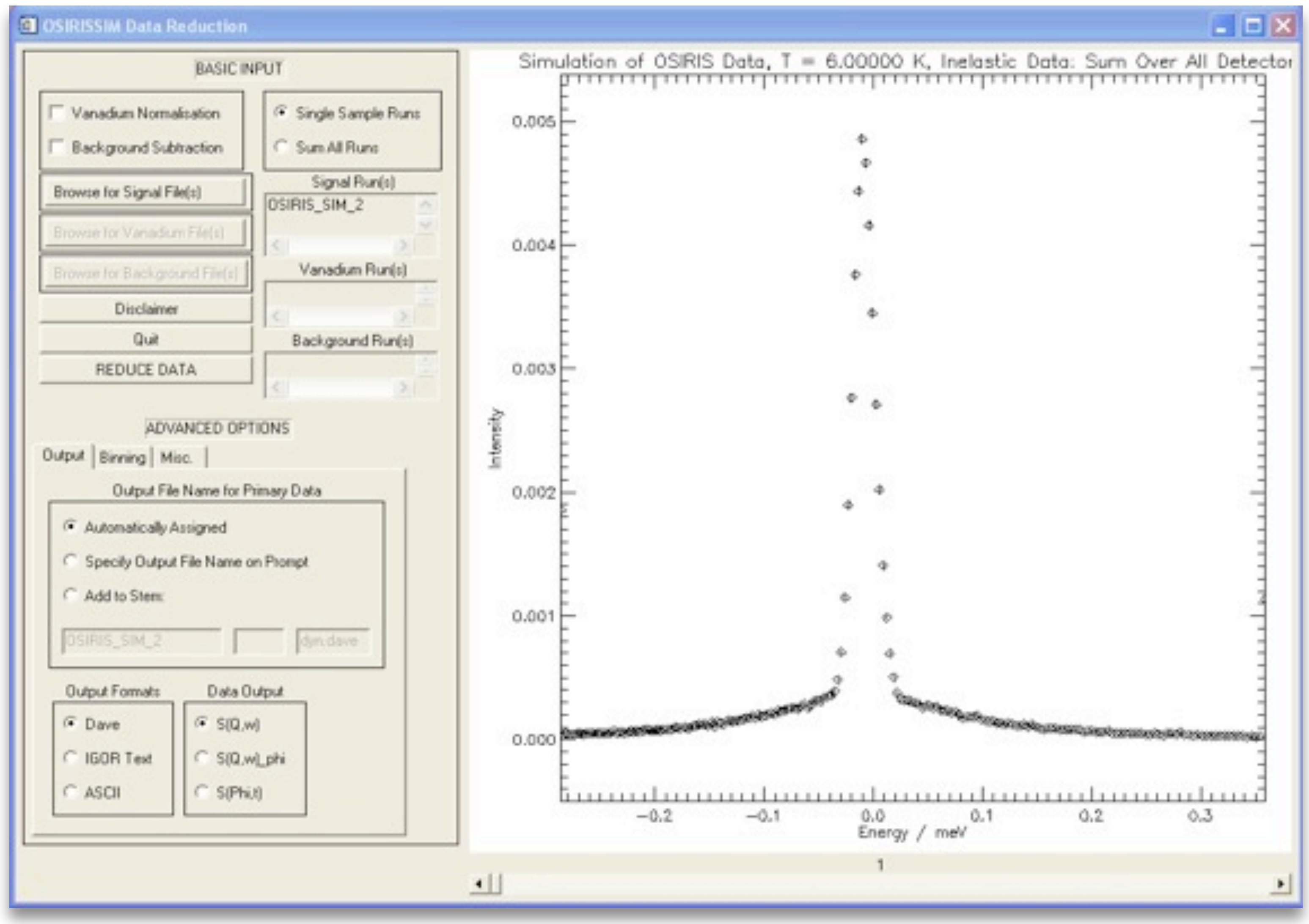
E. Farhi, ILL

Data Analysis (2)
(using VE techniques)

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



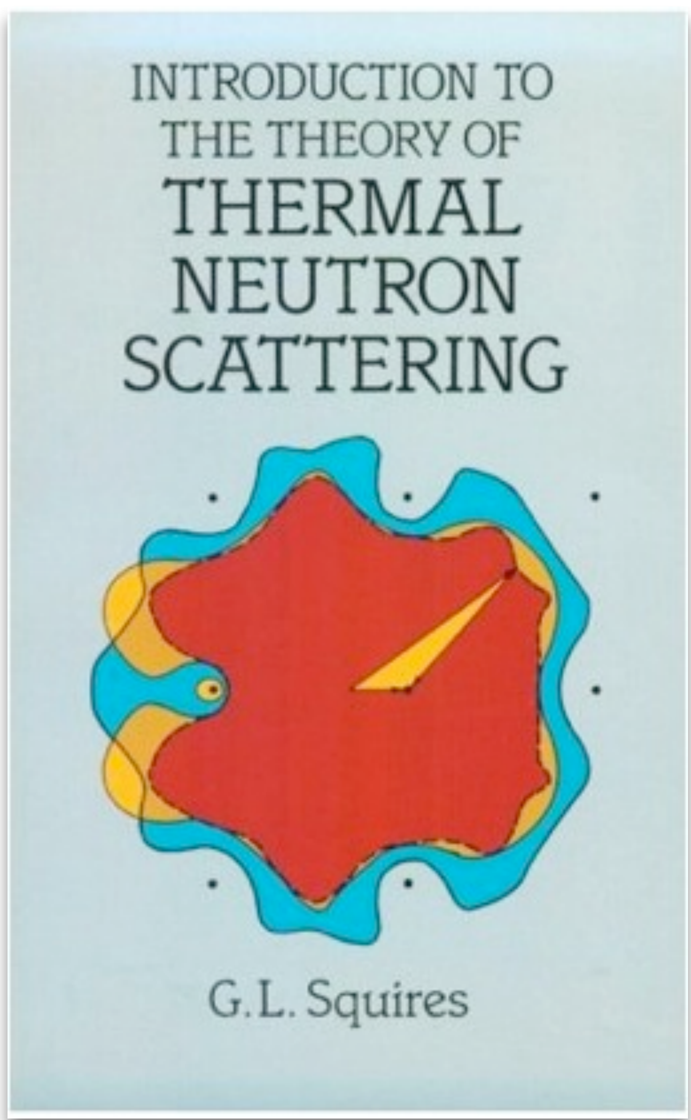
L. Udby, Risø-DTU

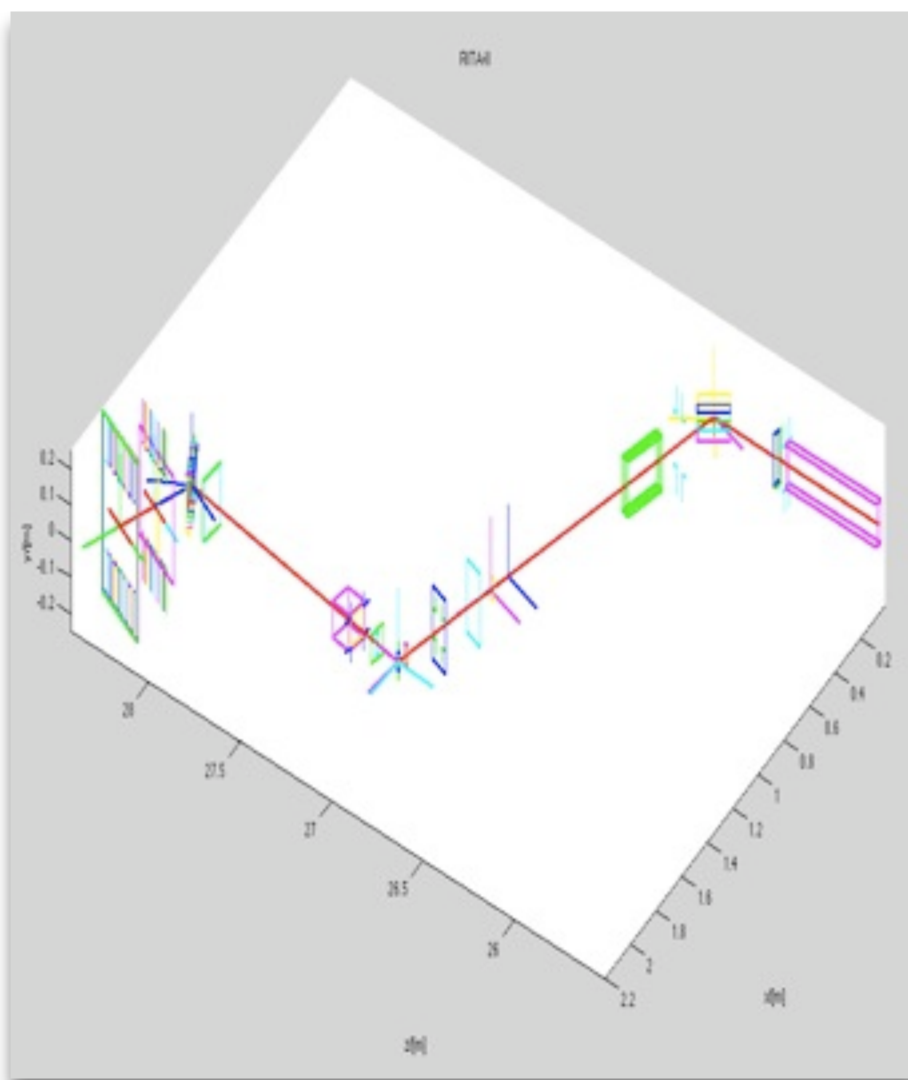


P. Tregenna-Piggott, PSI

Teaching / training purposes

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

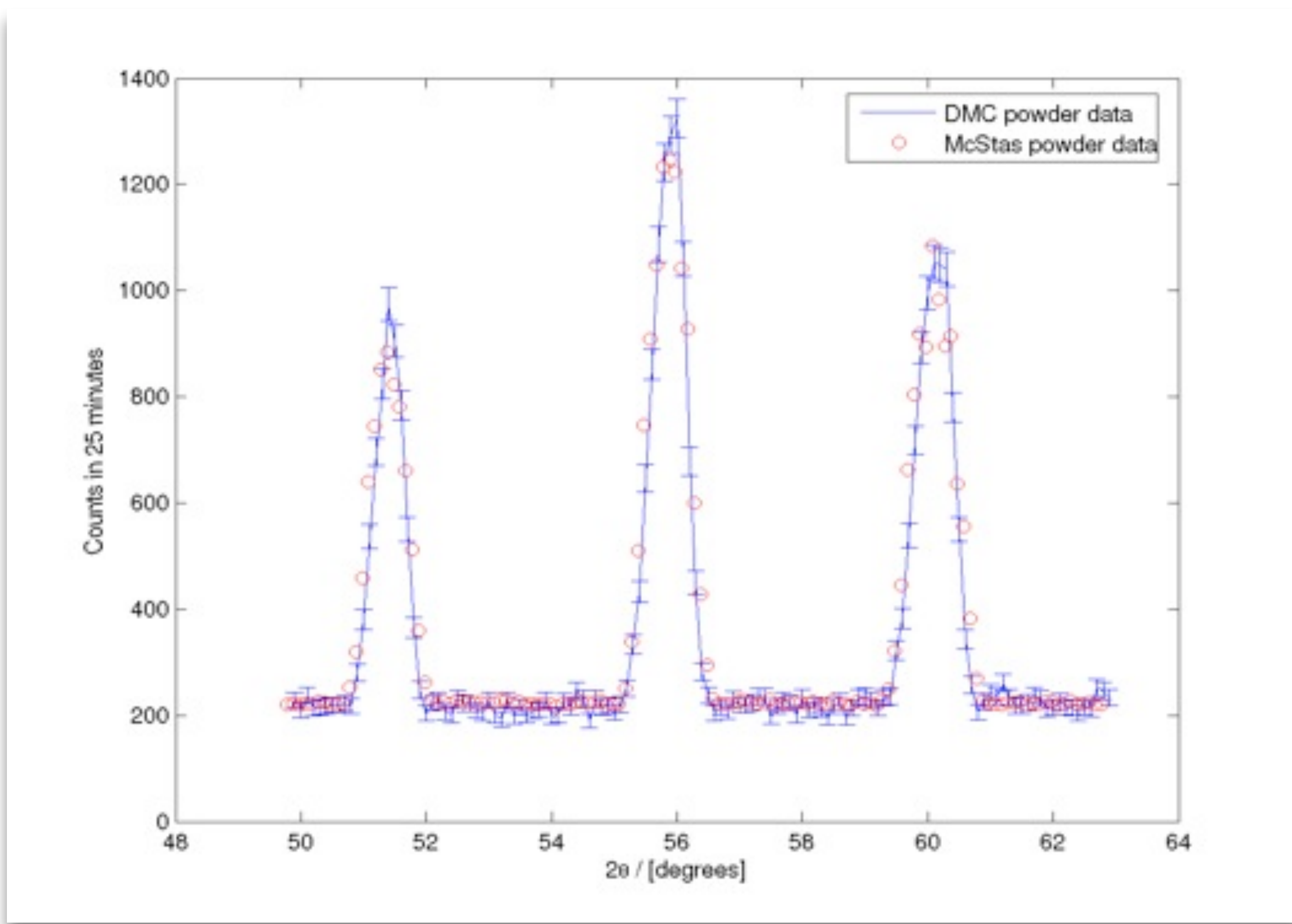


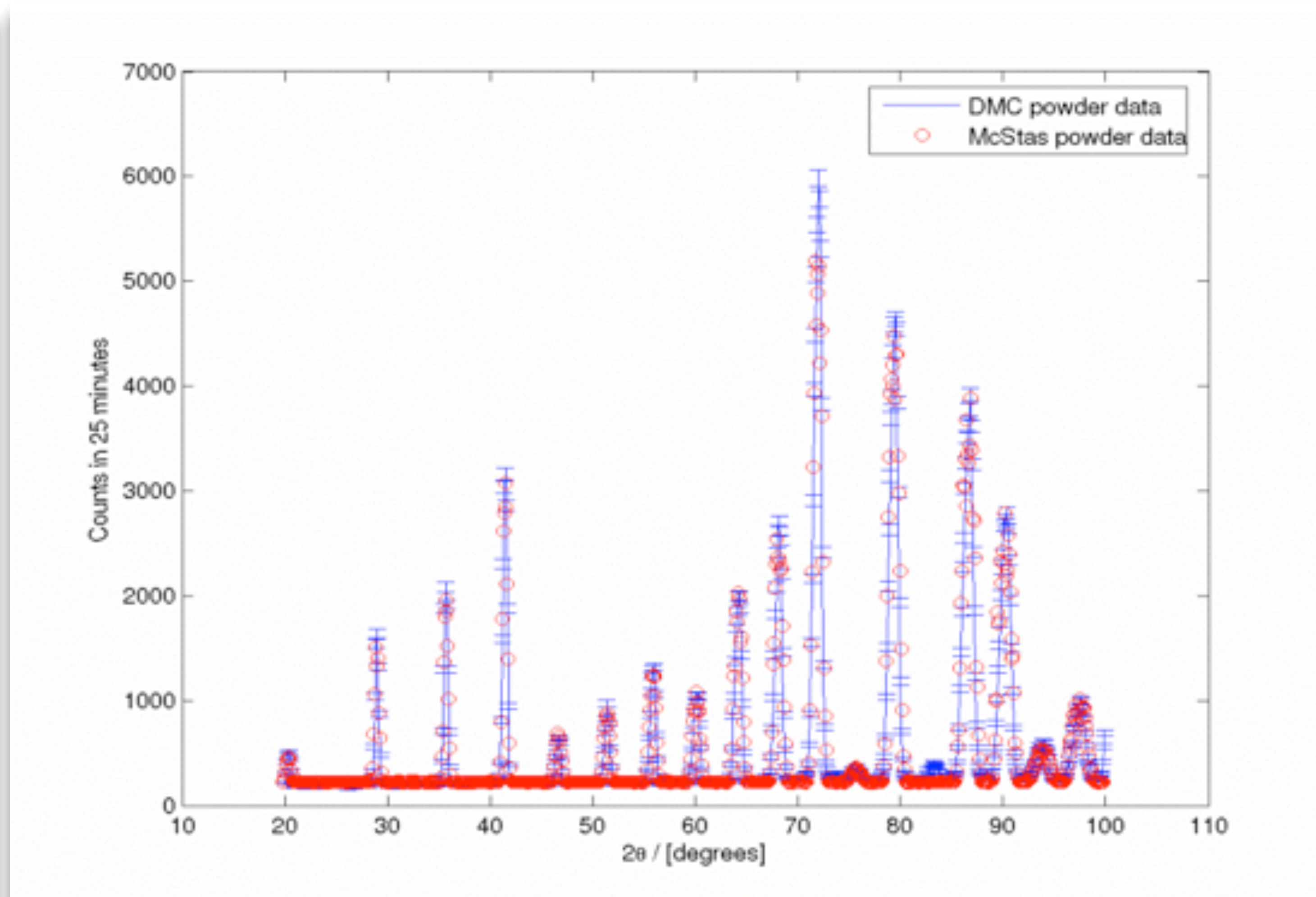


Risø DTU, Niels Bohr Institute, Institut Laue-Langevin

Reliability - cross comparisons

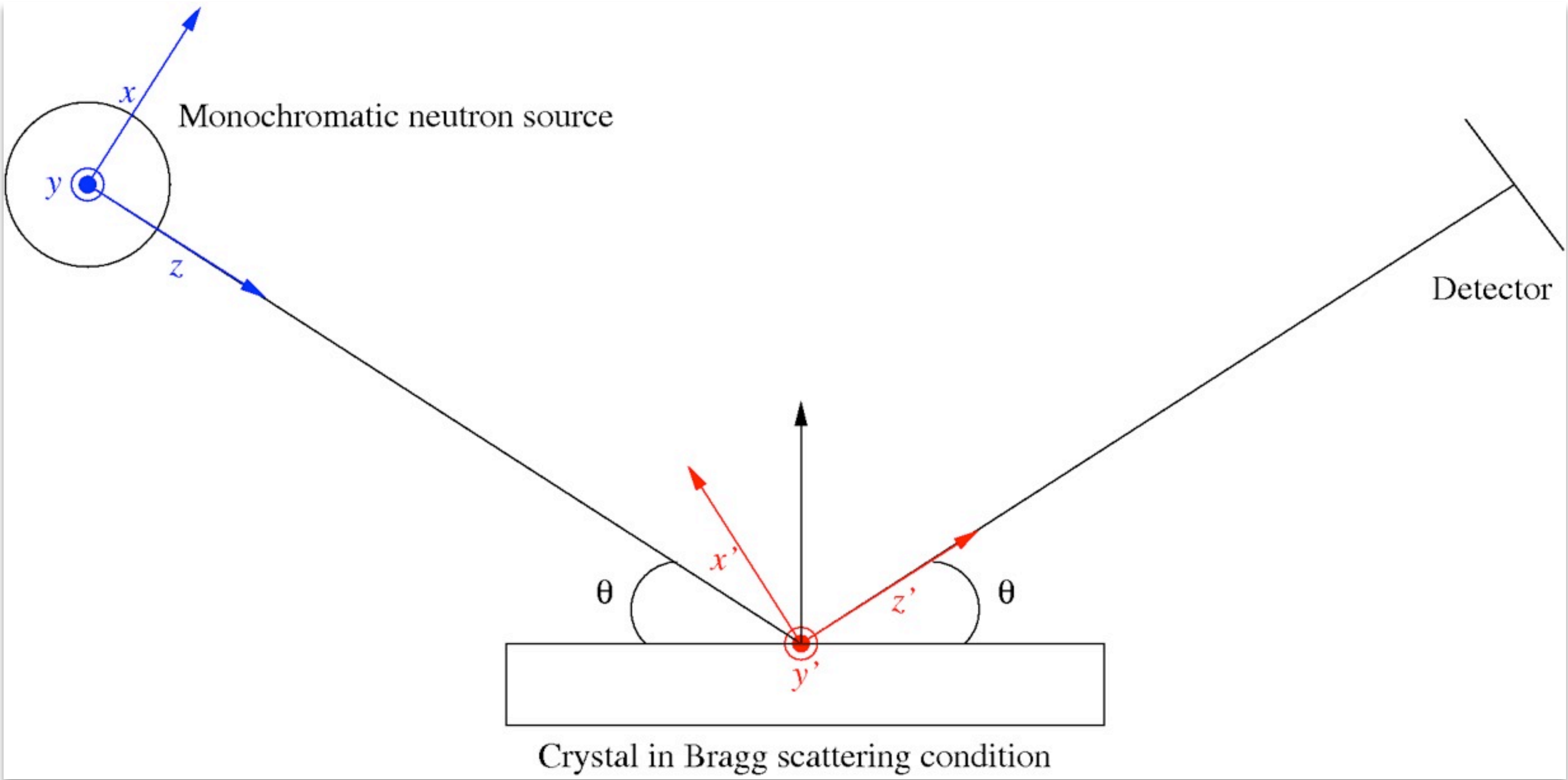
- Much effort has gone into this
- Here: simulations vs. exp. at powder diffract. DMC, PSI
- The bottom line is
- McStas agree very well with other packages (NISP, VitESS, IDEAS, RESTRAX, ...)
- Experimental line shapes are within 5%
- Absolute intensities are within 10-30%
- Common understanding: McStas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



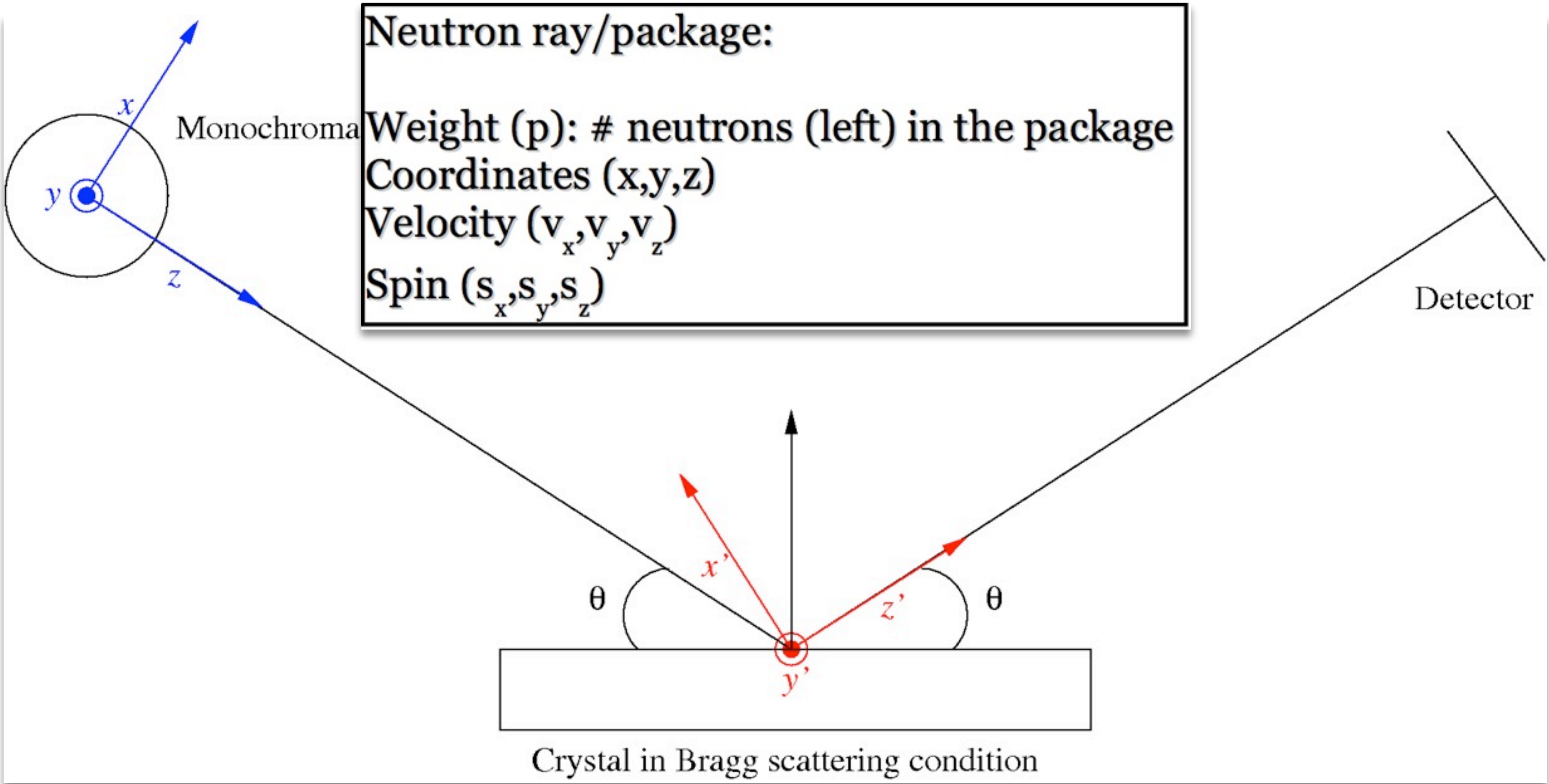


P. Willendrup, Risø DTU; Uwe Filges, L. Keller, PSI

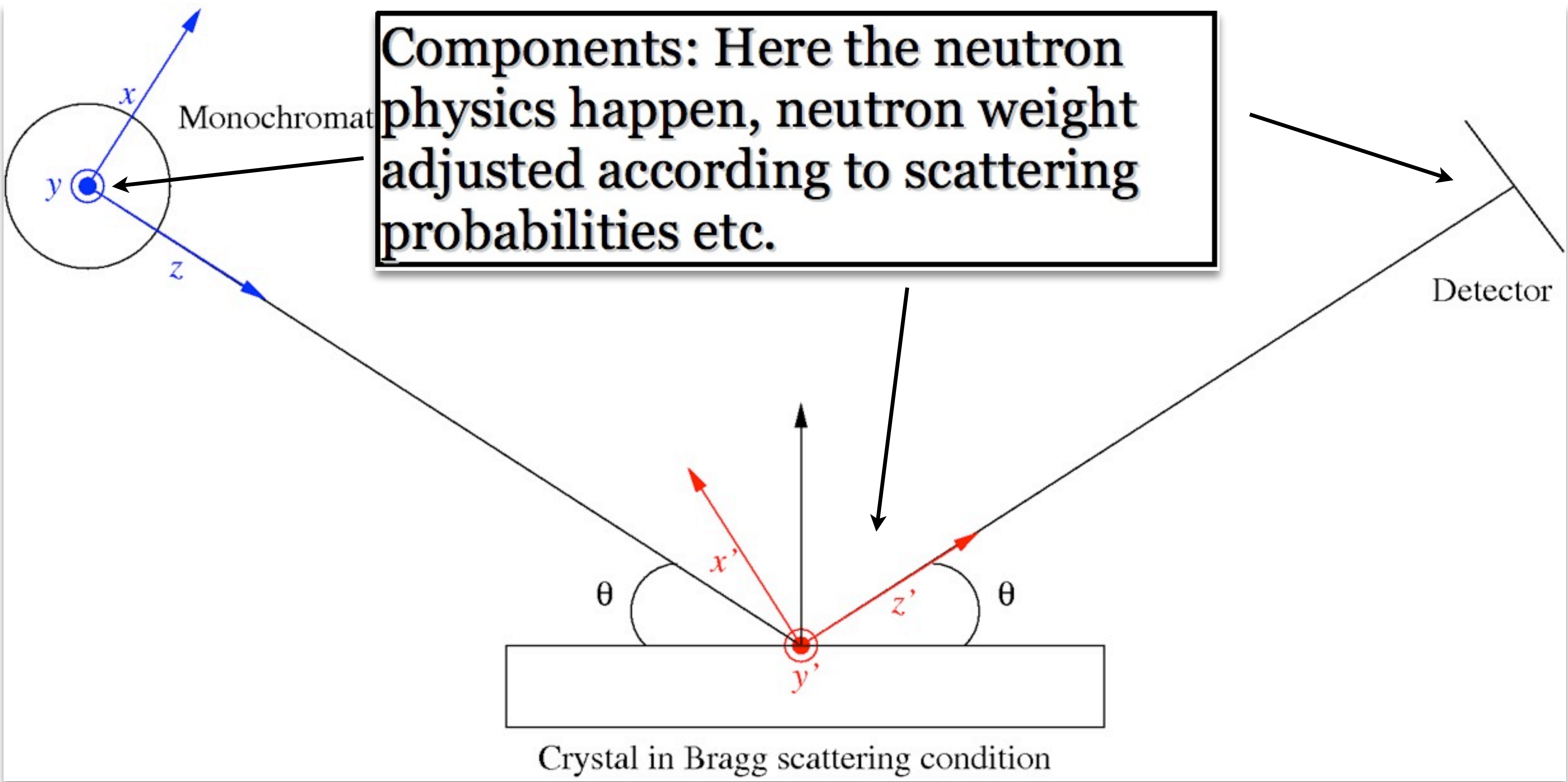
McStas: key concepts



McStas: key concepts

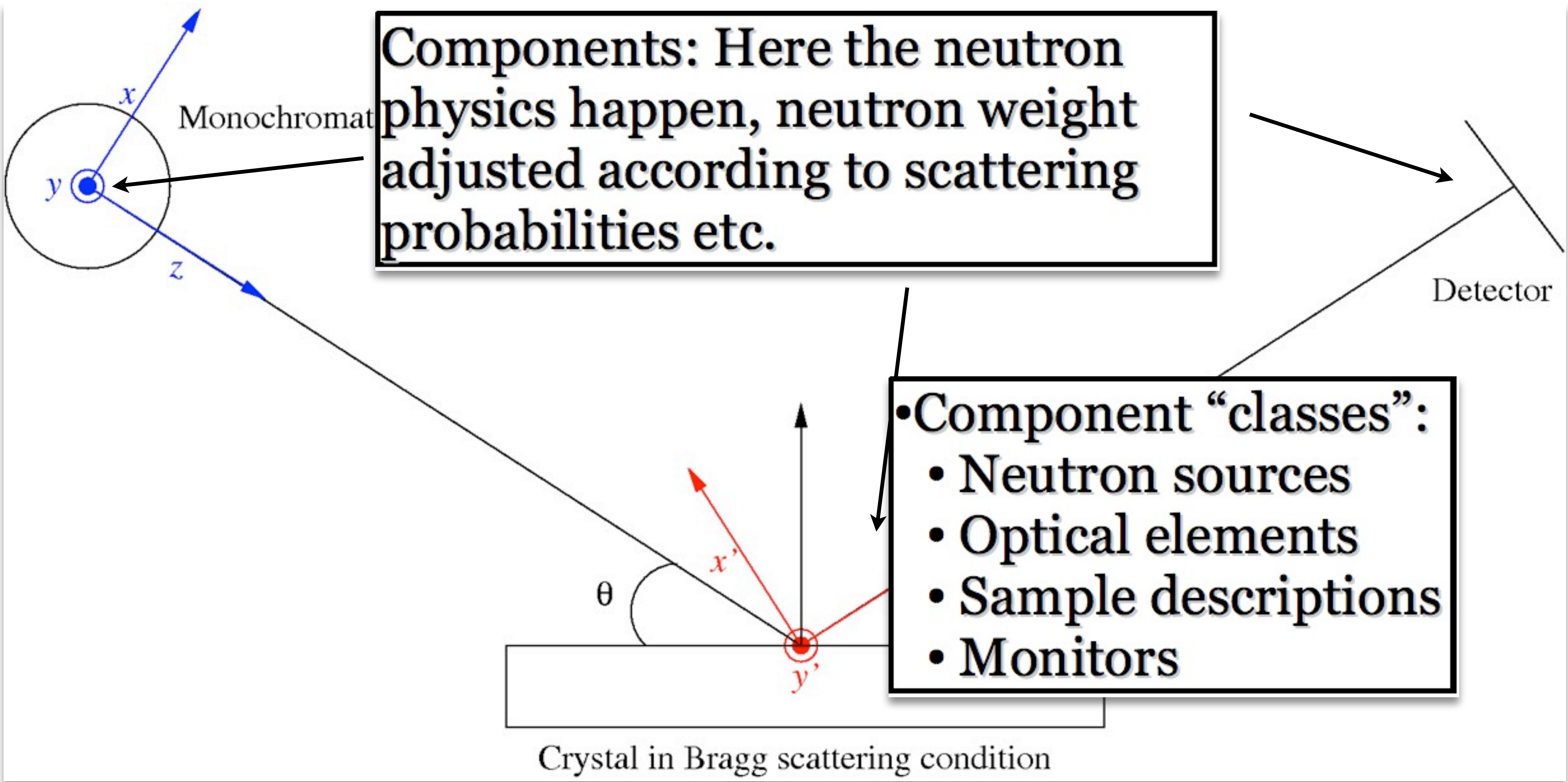


McStas: key concepts



Risø DTU, Niels Bohr Institute, Institut Laue-Langevin

McStas: key concepts

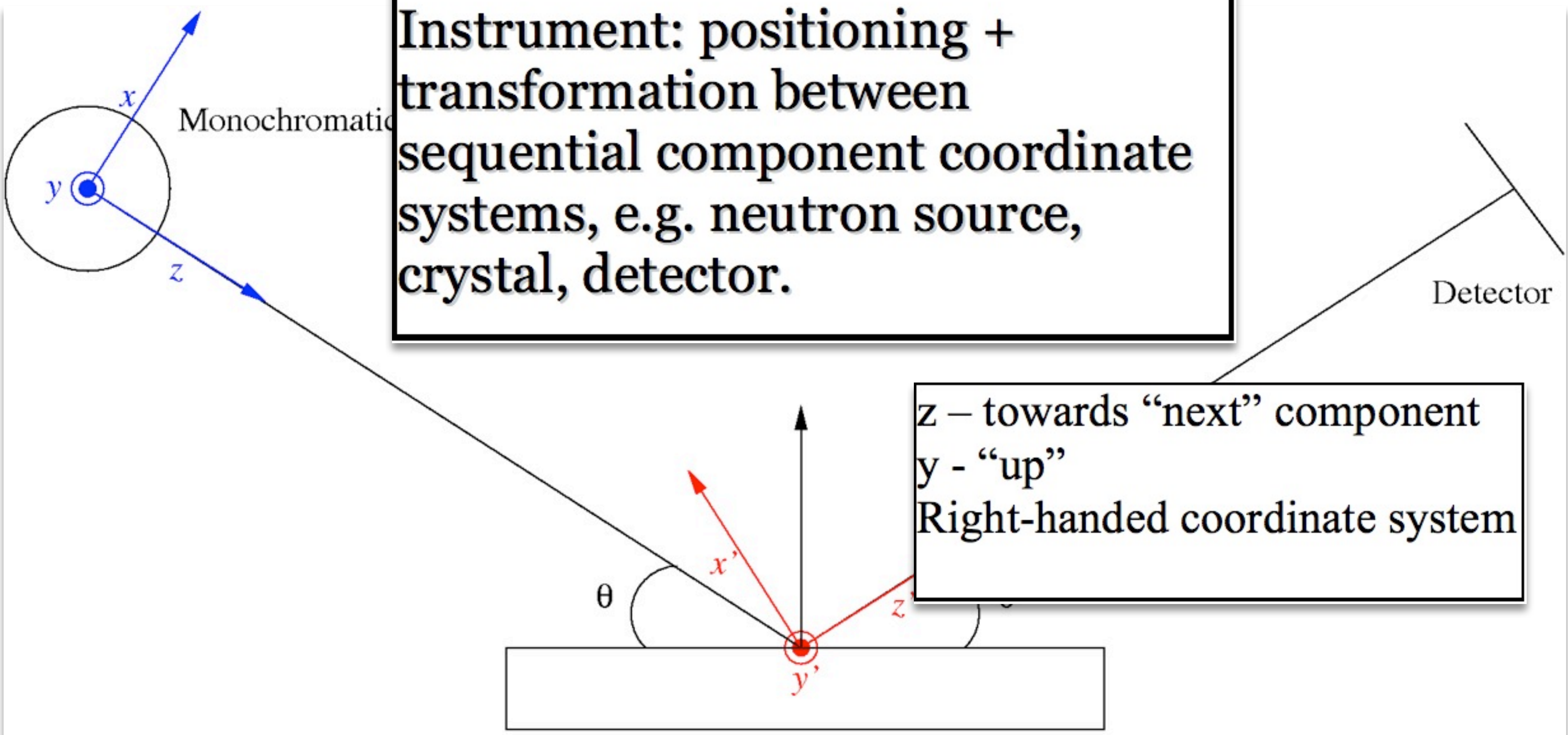


Components: Here the neutron physics happen, neutron weight adjusted according to scattering probabilities etc.

- Component “classes”:
  - Neutron sources
  - Optical elements
  - Sample descriptions
  - Monitors

McStas: key concepts

Instrument: positioning + transformation between sequential component coordinate systems, e.g. neutron source, crystal, detector.



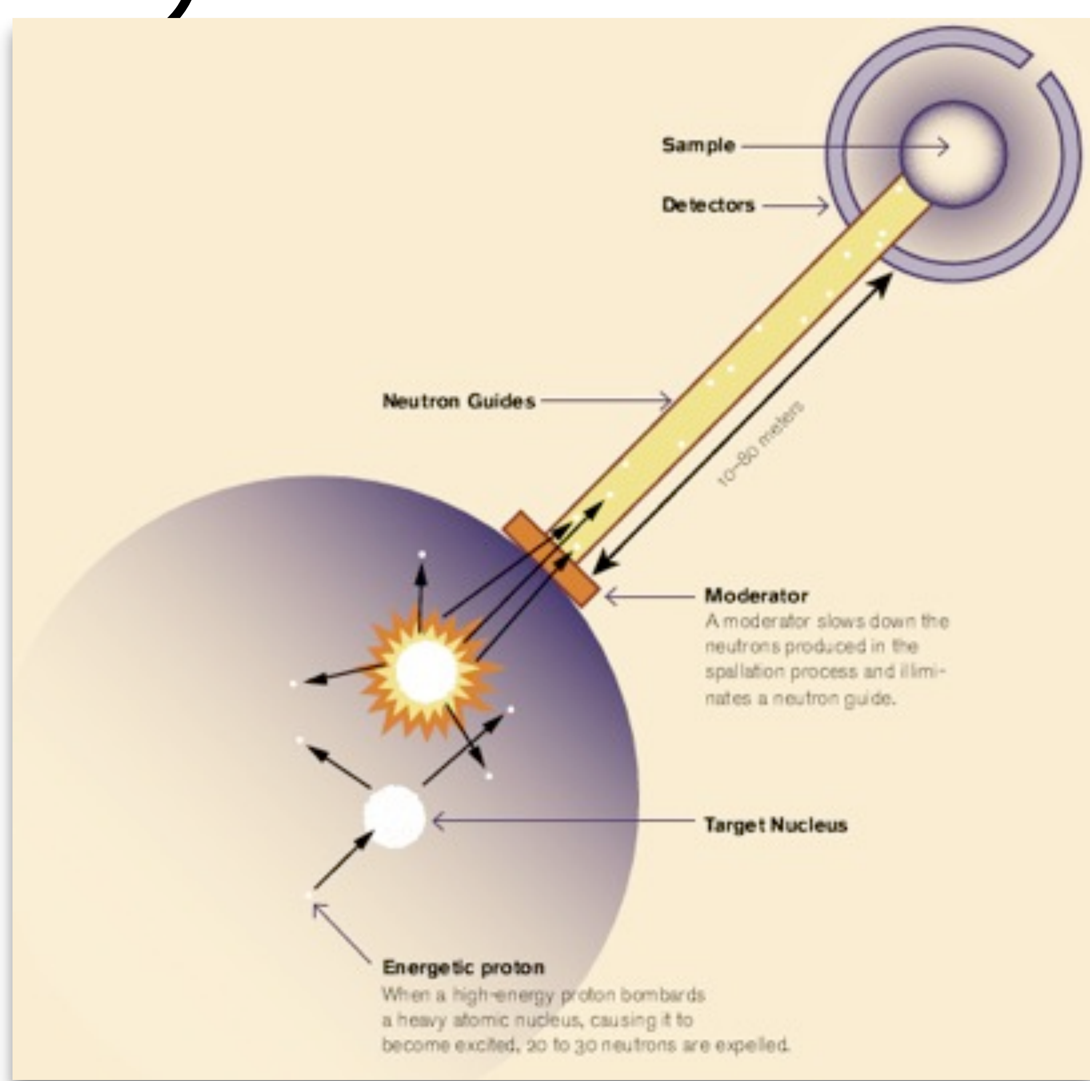
Crystal in Bragg scattering condition

z – towards "next" component
y - "up"
Right-handed coordinate system

McStas overview

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

- 'Component' files (~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();
%}
```

```
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*randpm1();              /*  Choose from uniform distribution */
   v=sqrt(E)*SE2V;
 } else {
   Lambda=Lambda0+dLambda*randpm1();
   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(mcposrSource, mcrotrSource,
    &mcnlx, &mcnly, &mcnlz,
    &mcnlvx, &mcnlvy, &mcnlvz,
    &mcnlt, &mcnlsx, &mcnlsy);
  mcDEBUG_STATE(mcnlx, mcnly, mcnlz, mcnlvx, mcnlvy, mcnlvz,mcnlt,mcnlsx,mcnlsy, mcnlp)
#define x mcnlx
#define y mcnly
#define z mcnlz
#define vx mcnlvx
#define vy mcnlvy
#define vz mcnlvz
#define t mcnlt
#define s1 mcnlsx
#define s2 mcnlsy
#define p mcnlp
  STORE_NEUTRON(2,mcnlx, mcnly, mcnlz, mcnlvx,mcnlvy,mcnlvz,mcnlt,mcnlsx,mcnlsy, mcnlsz, mcnlp);
  mcScattered=0;
  mcNCounter[2]++;
#define mccompcurname Source
#define mccompcurindex 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, &pdir,
        0, 0, dist, xw, yh, ROT_A_CURRENT_COMP);
```

Written by mcstas!

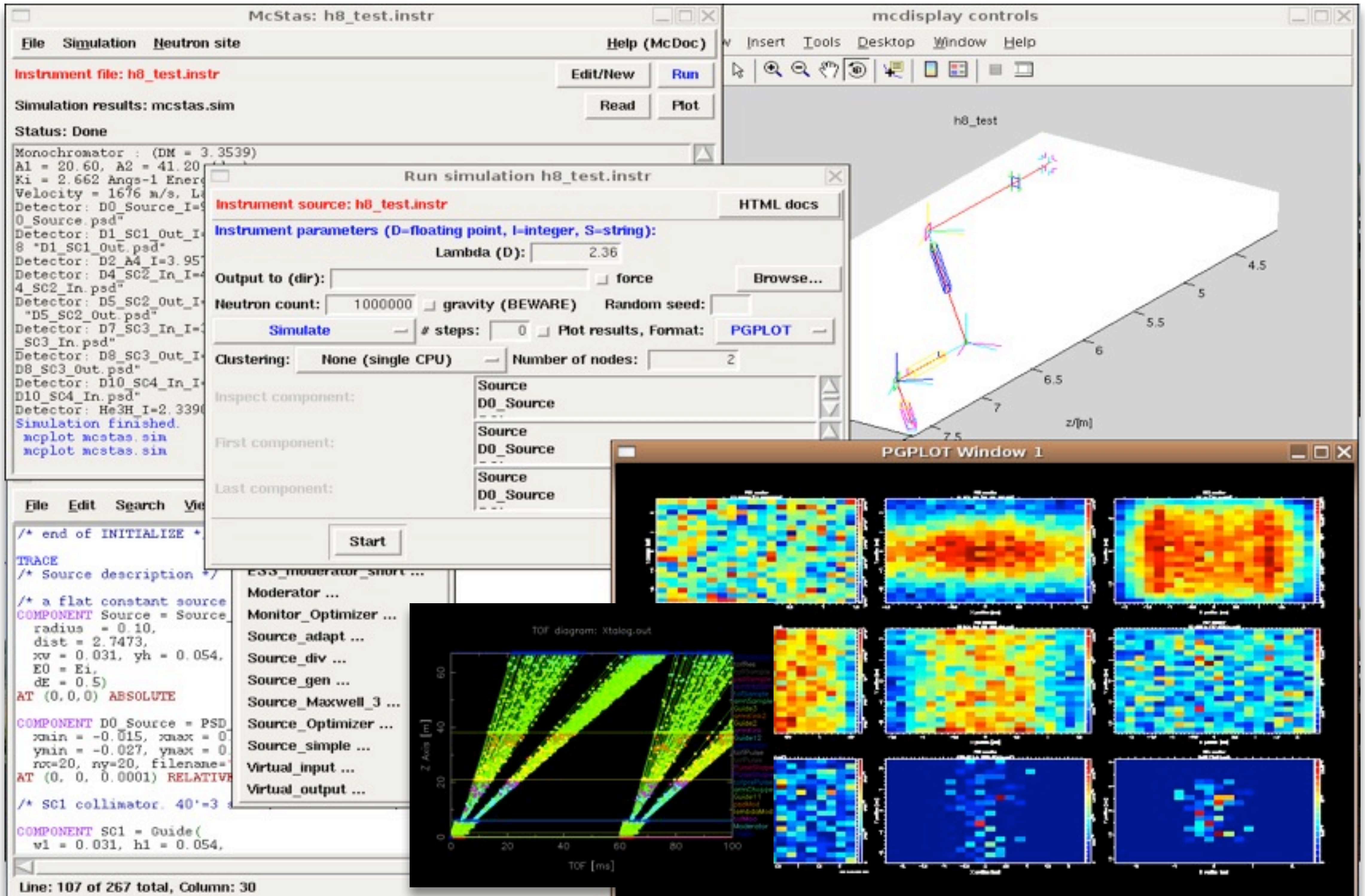
McStas is a (pre)compiler!

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.

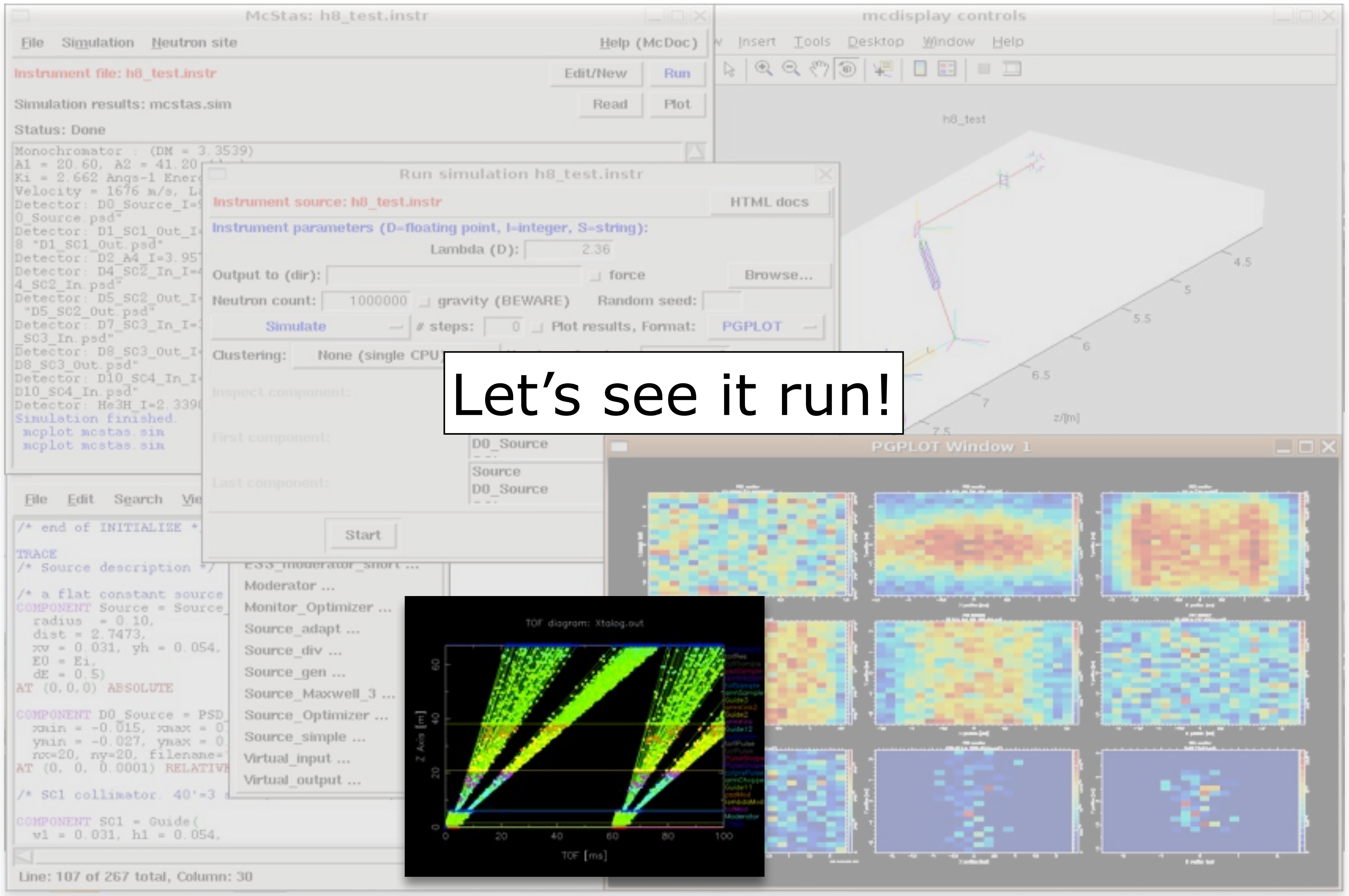
Can take input arguments if needed.

McStas overview



Risø DTU, Niels Bohr Institute, Institut Laue-Langevin

McStas overview



Introduction to VITESS

V irtual
 I nstrumentation
T ool for the
E uropean
S pallation
S ource

Klaus Lieutenant

History of VITESS

Idea of Ferenc Mezei to realize a package to simulate all kinds of neutron scattering instruments especially on neutron spallation sources because of the European spallation source (ESS)

Important dates

1998: Some existing programs put together, GUI added

1999: Release of VITESS 1.0
First complete instruments simulated

2000: SCANS collaboration started (followed by MCNSI in FP6) (McStas, VITESS, …)

2001: Release of VITESS 2.0 containing polarisation, absolute flux values, improved GUI
Several ESS instruments simulated

2003: Decision: ESS will not be built in the near future
VITESS will be used for instruments on other sources

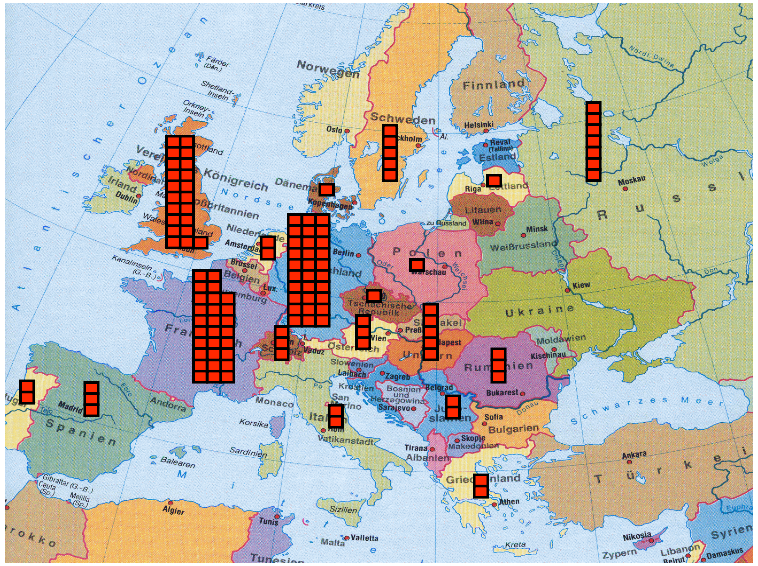
2005: VITESS group at HMI closed

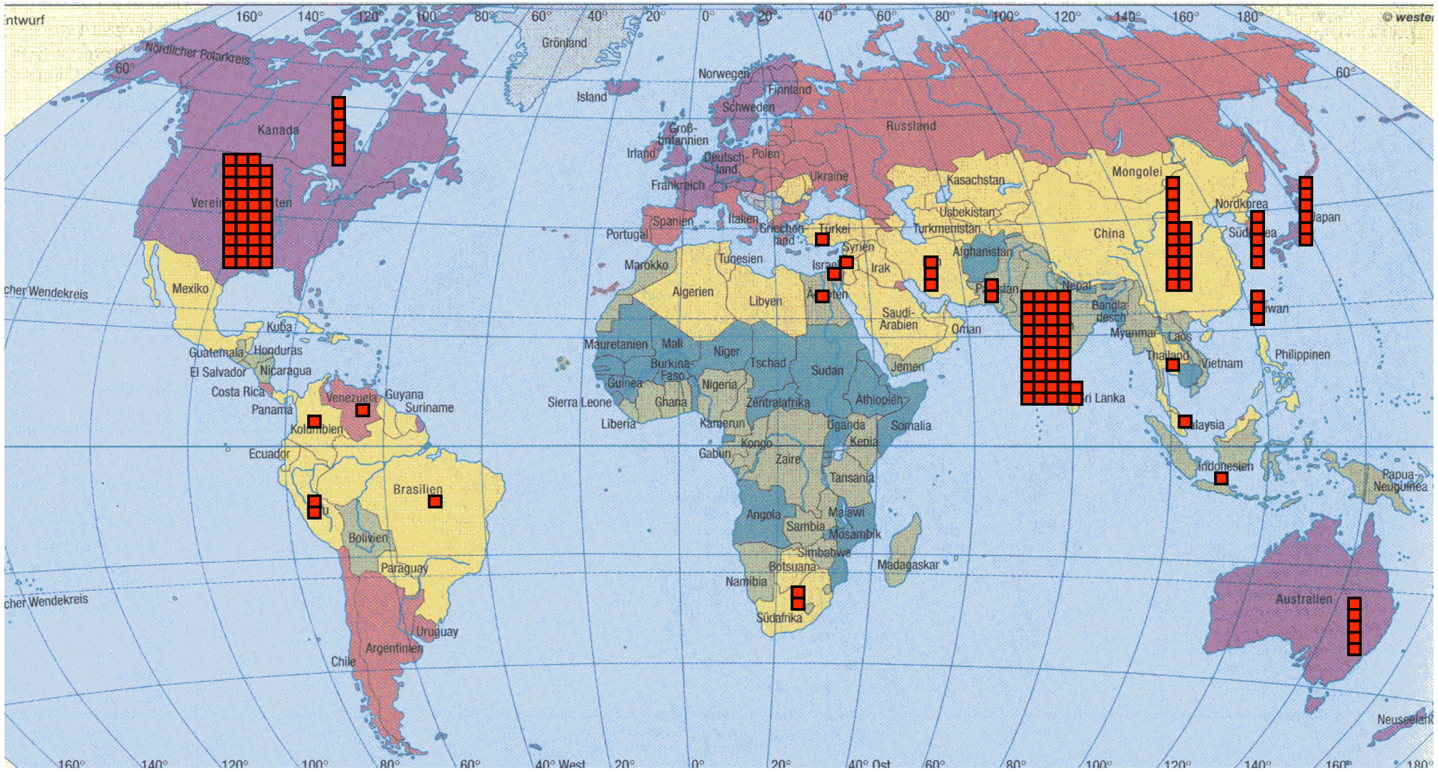
2006: VITESS released under GNU license
larger developer team, all working only partly on VITESS

2008: Version 2.8 released

2009: Latest version 2.9 released

Distribution of Downloads in 2003





Present Status

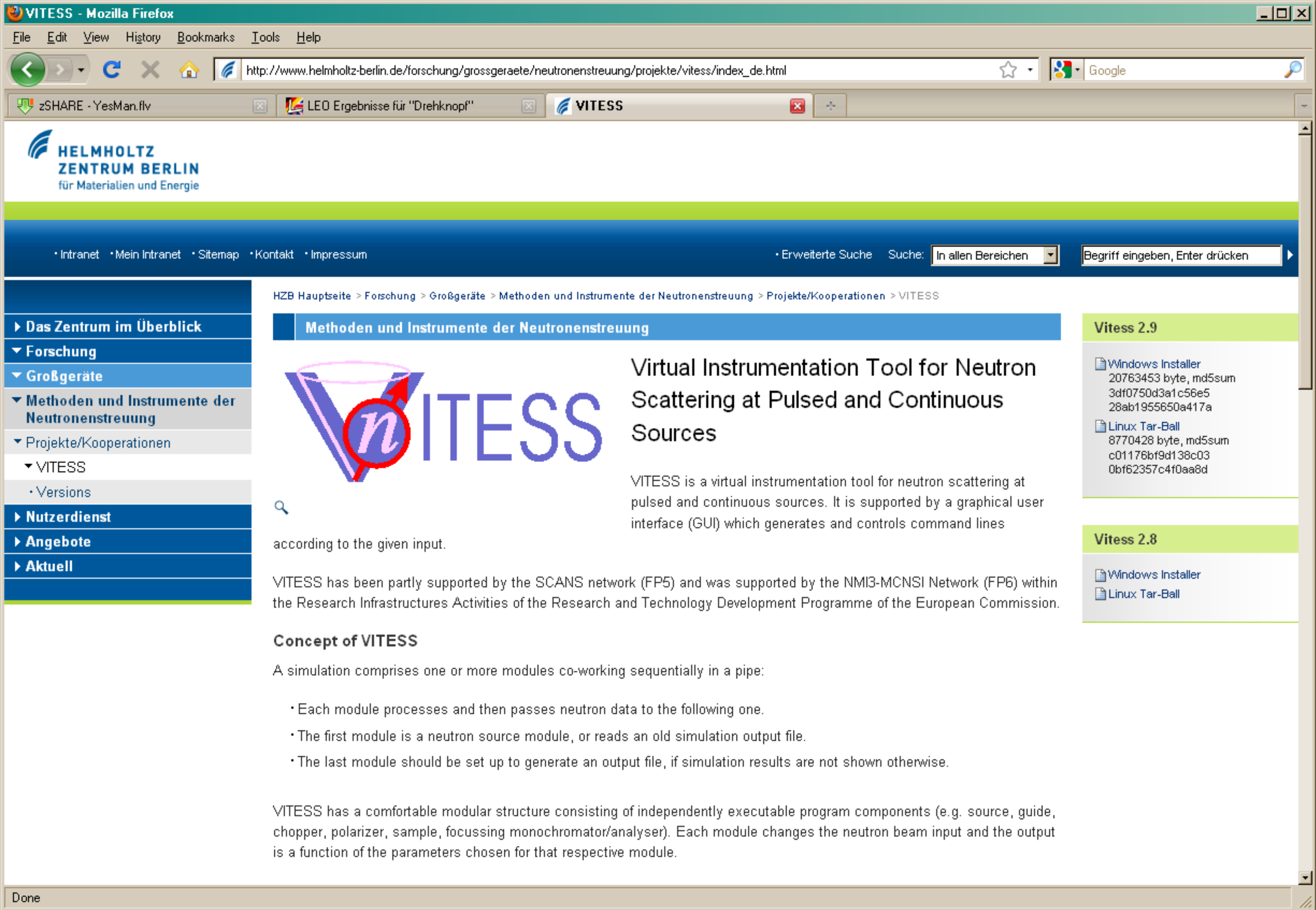
Staff

- Michael Fromme (HMI; GUI and release of new versions)
- Sergey Manoshin (JINR; development of new modules)
- Klaus Lieutenant (IFE; module development)
- Andreas Houben (RWTH Aachen, module development)
- Phillip Bentley (ANSTO; optimisation routine)
- Geza Zsigmond (PSI; maintenance of his modules)

Program

- Executables for Windows/DOS, Unix (SunOS: versions from 5.6, OSF1 V4.0) and Linux (versions from 2.0.35), Macintosh on demand
- free of charge
- Can be downloaded from internet address http://www.helmholtz-berlin.de/vitess/

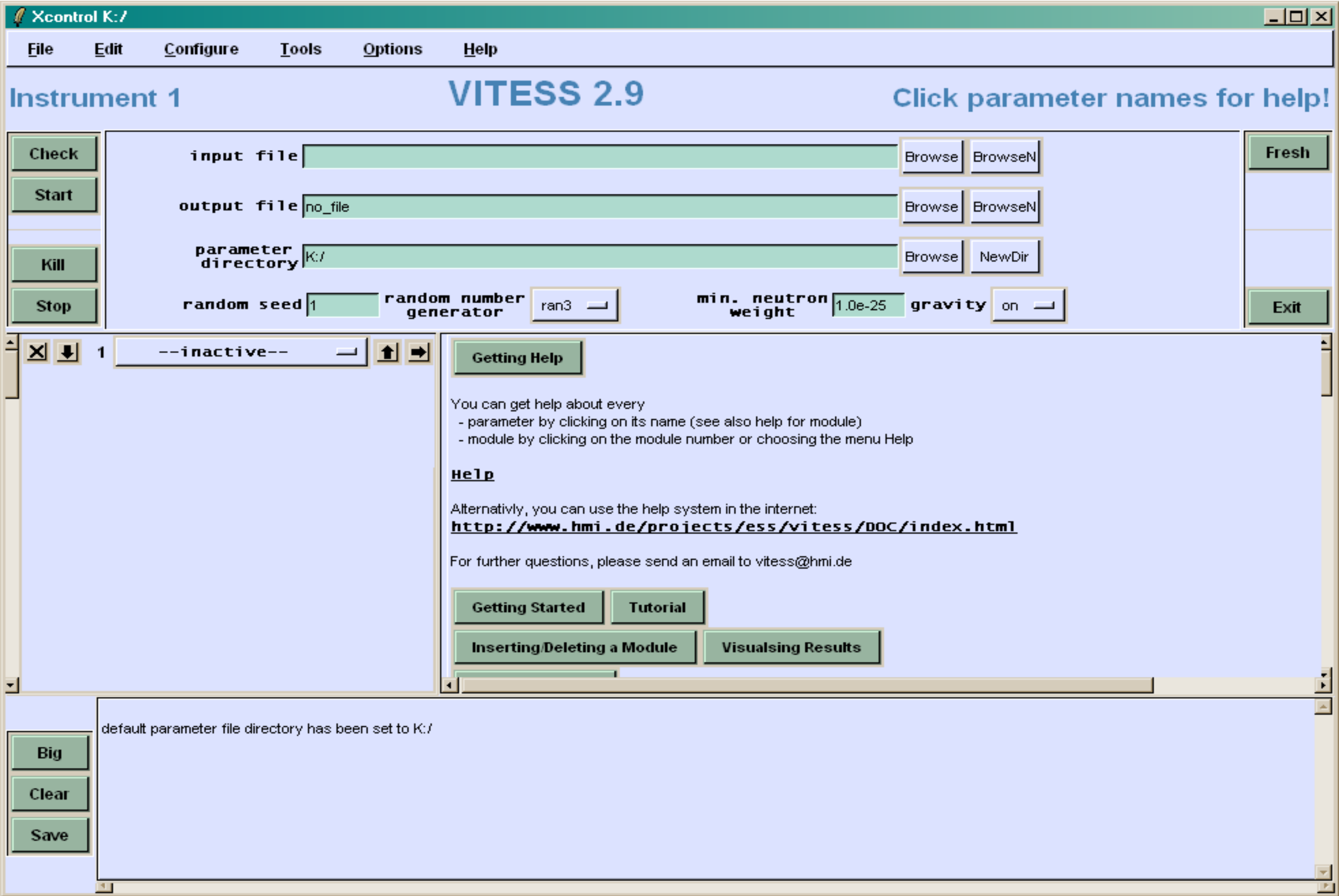
Home Page ‘www.helmholtz-berlin.de/vitess’



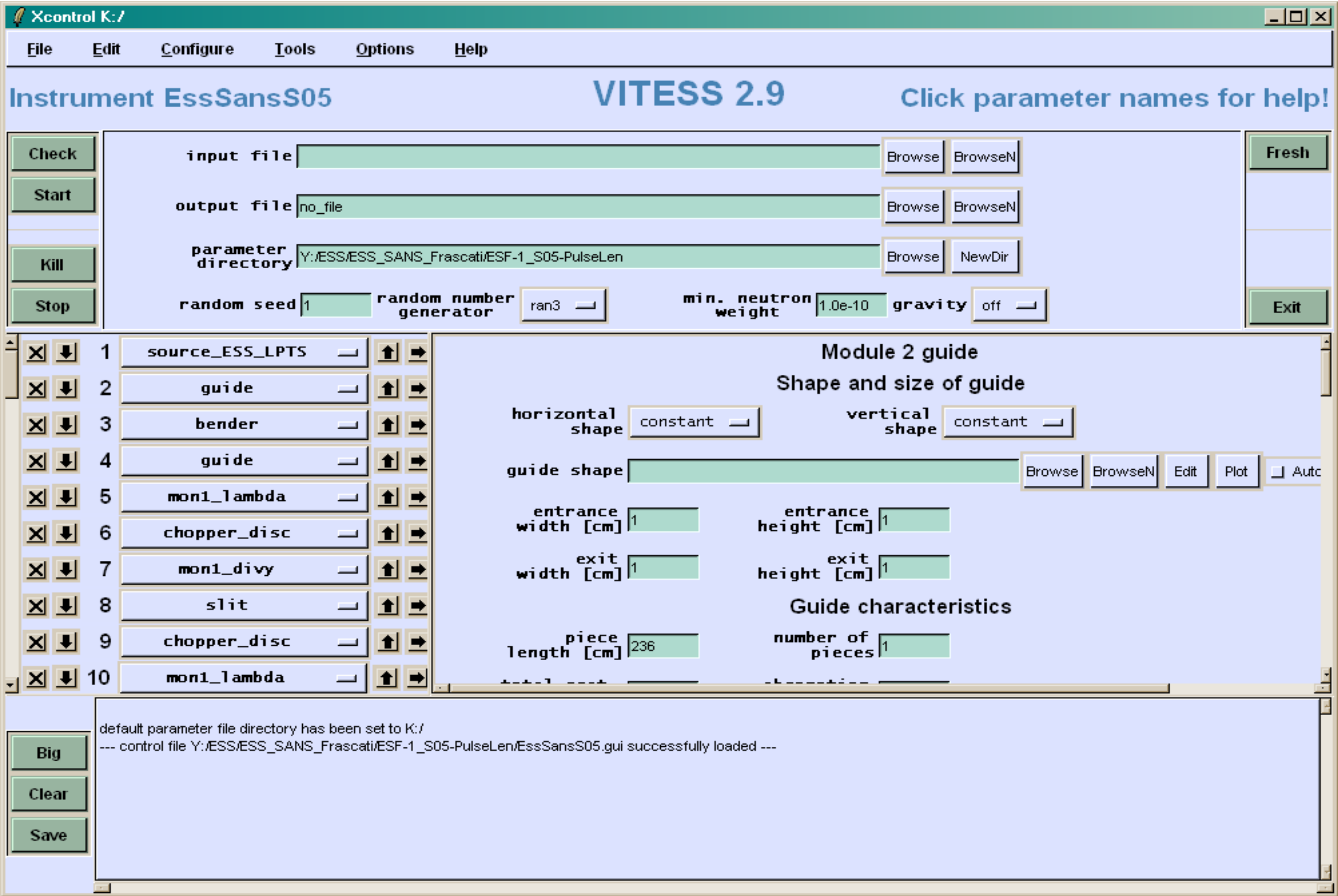
Ven 2010

essworkshop.org

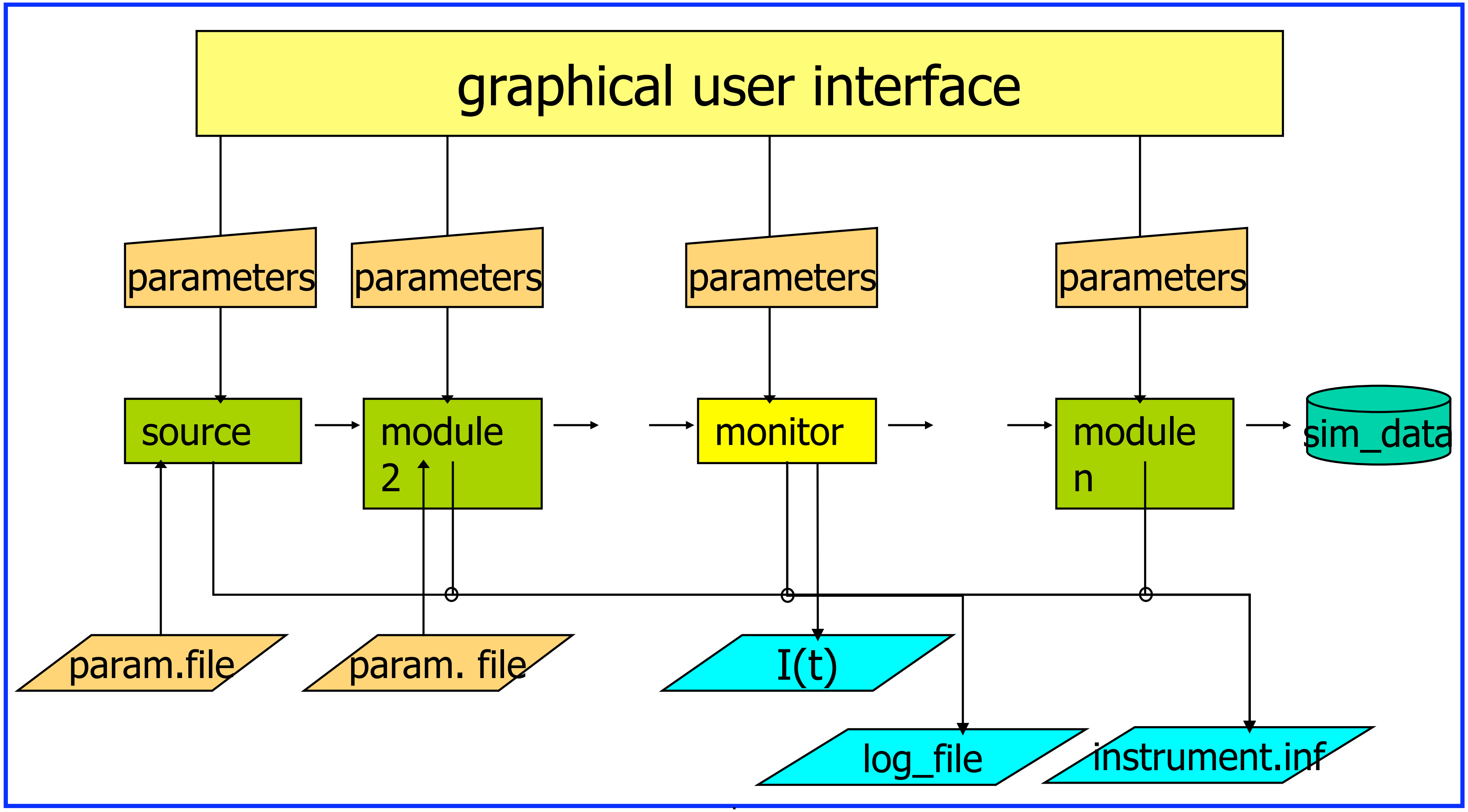
GUI after program start



GUI after loading an instrument



Concept of VITESS



Parameter set transferred

Right-handed system

x: along the beamline
y: to the left
z: (vertically) up

ID

criterion 'ray tracing'

'colour'

Time of flight t [ms]

wavelength λ [Å]

count rate p [n/s]

location of neutron x [cm]

location of neutron y [cm]

location of neutron y [cm]

flight direction $v_x/|\underline{v}| = \cos \alpha$

flight direction $v_y/|\underline{v}| = \cos \beta$

flight direction $v_z/|\underline{v}| = \cos \gamma$

Spin P_x

Spin P_y

Spin P_z

Modules representing Hardware

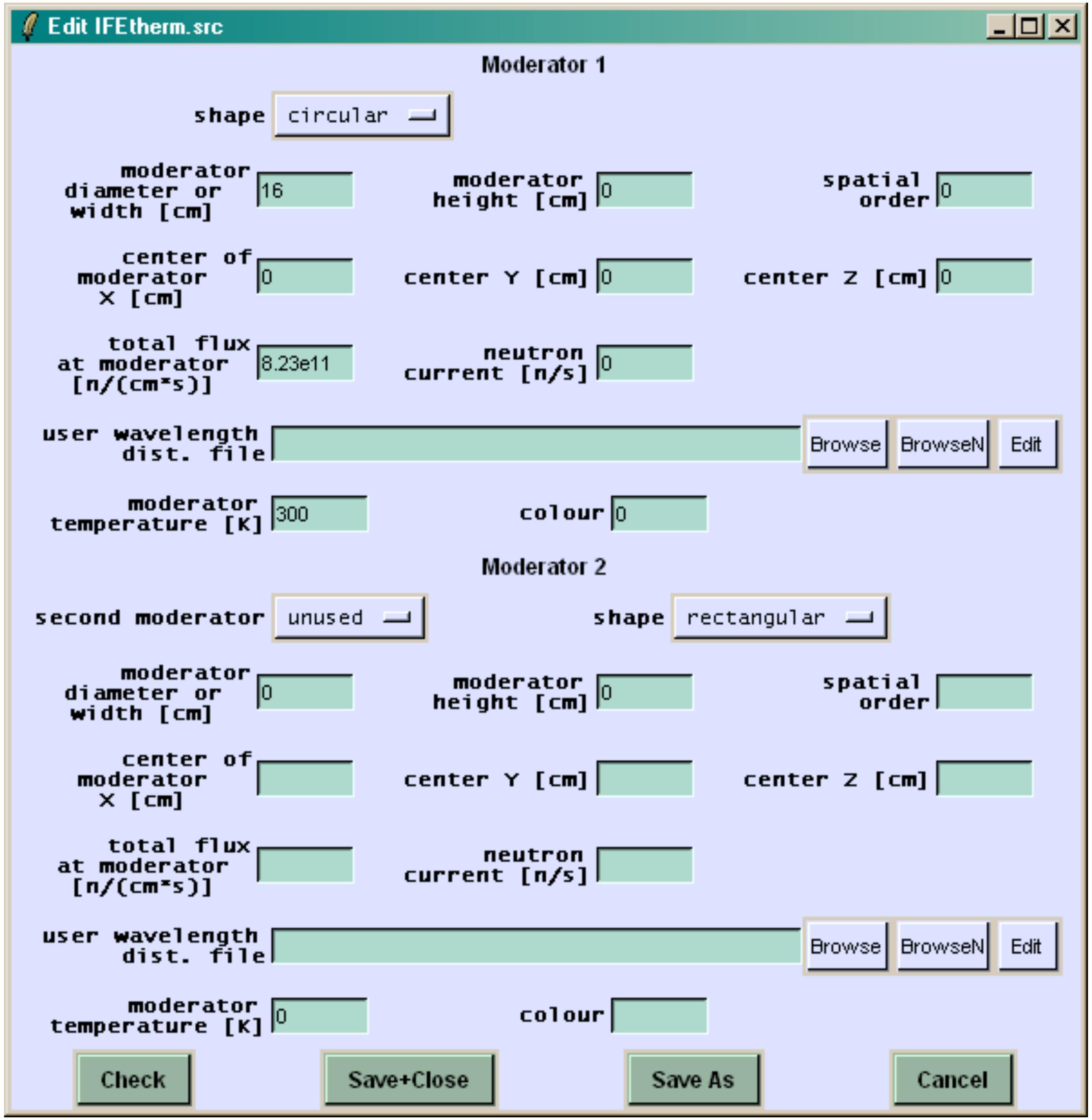
Sources	reactor	SPSS	LPSS		
Space + Windows	space	slit	window/ beamstop	multiple windows	grid
Choppers	disk	Fermi str.	Fermi cur		
Mono/Ana	flat	focus	foc.user		
Modules f. Polaris.	^{3}He-pol.	coil flip.	prec.field	sesansfield	pol. mirror
	SM-polar.	grad.flip.	rot. field	res.Drabkin	
Samples	elast.isot.	powder	SANS	reflecto.	
	inelastic	sngl.crys.	S(Q)	environm	
Collimat.	coll.simpl	coll.soller	coll.radial		
	guide	bender	vel.select	SM-ensmbl	detector

Risø DTU DTU

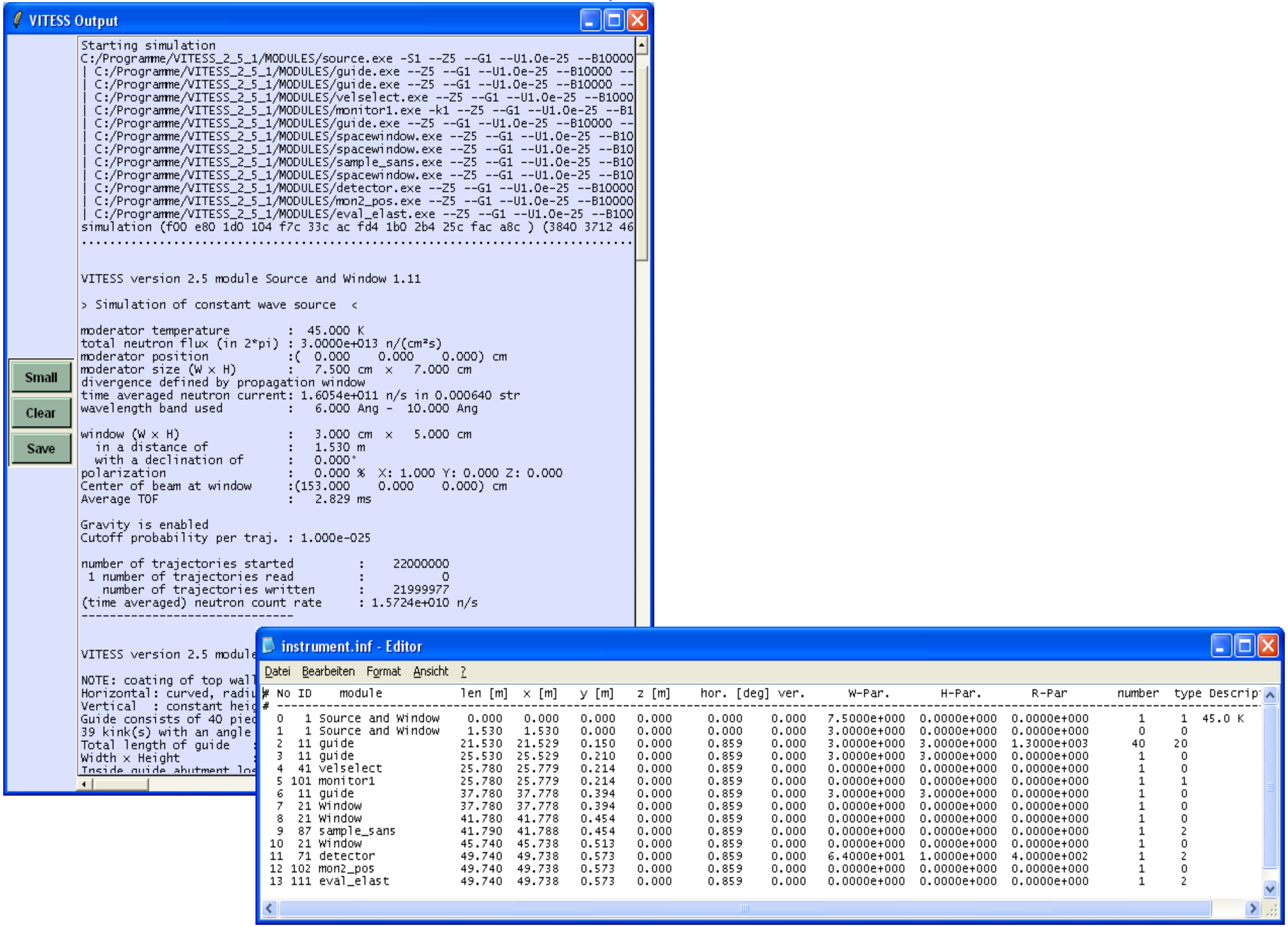
Modules for Data Evaluation

Monitor 1	Time	Wavelen	y-/z-pos	div-y/-z		
Monitor 2	y-pos z-pos	div-y div-z	TOF λ	λ div_y/-z	y div-y	z div-z
MonPol 1	Time	Wavelen	y-/z-pos	div_y/-z		
MonPol 2	y-pos z-pos					
Evaluate	Elastic	Inelast				
	Frame					
	Writeout					
	Visualize					

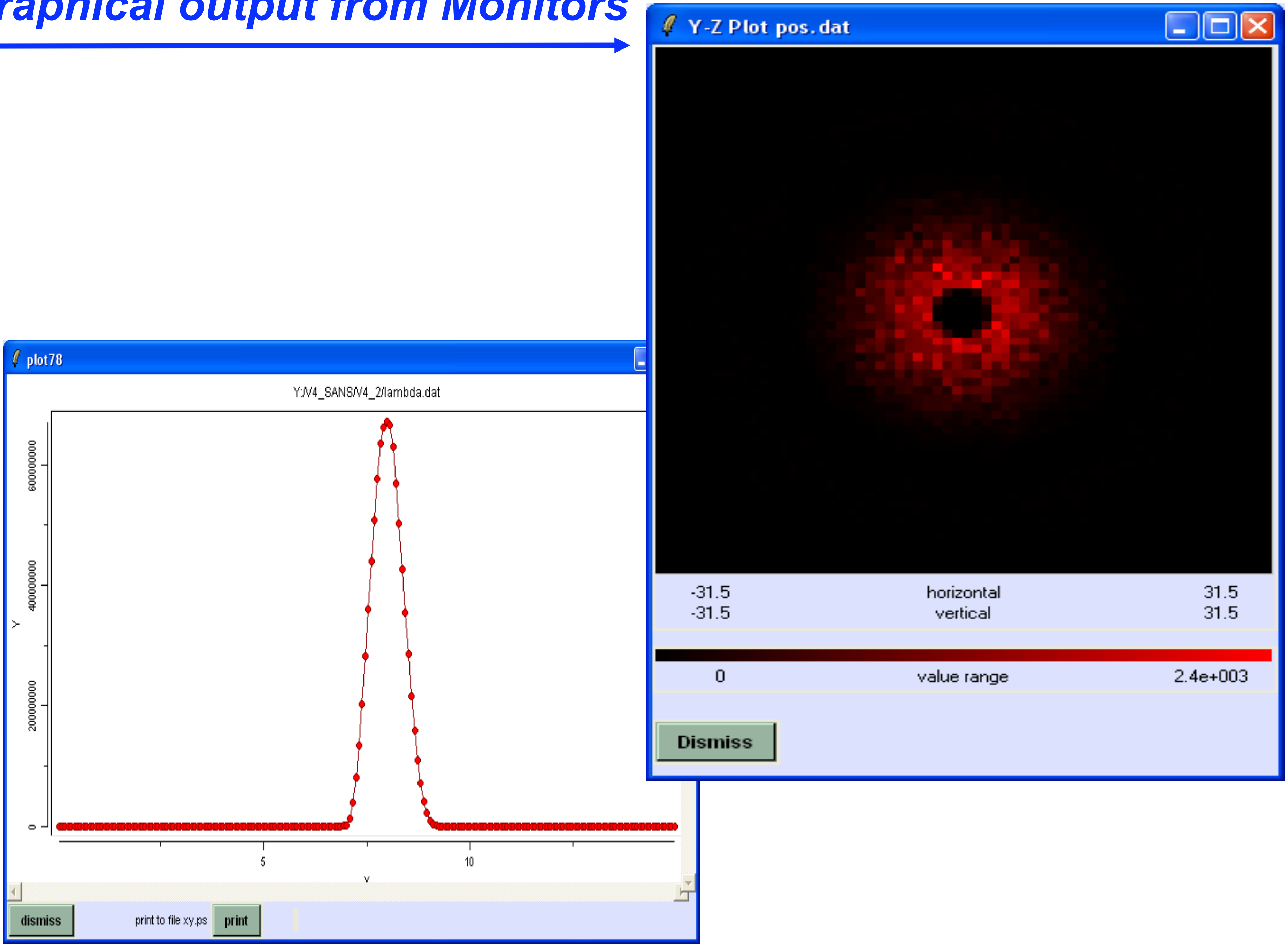
Input Data: Parameter File



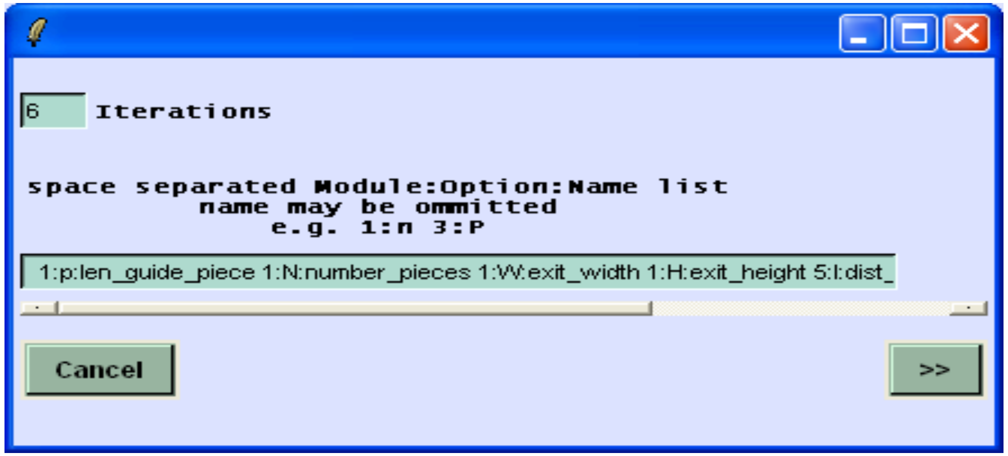
Output Data: Log File and Instrument Data

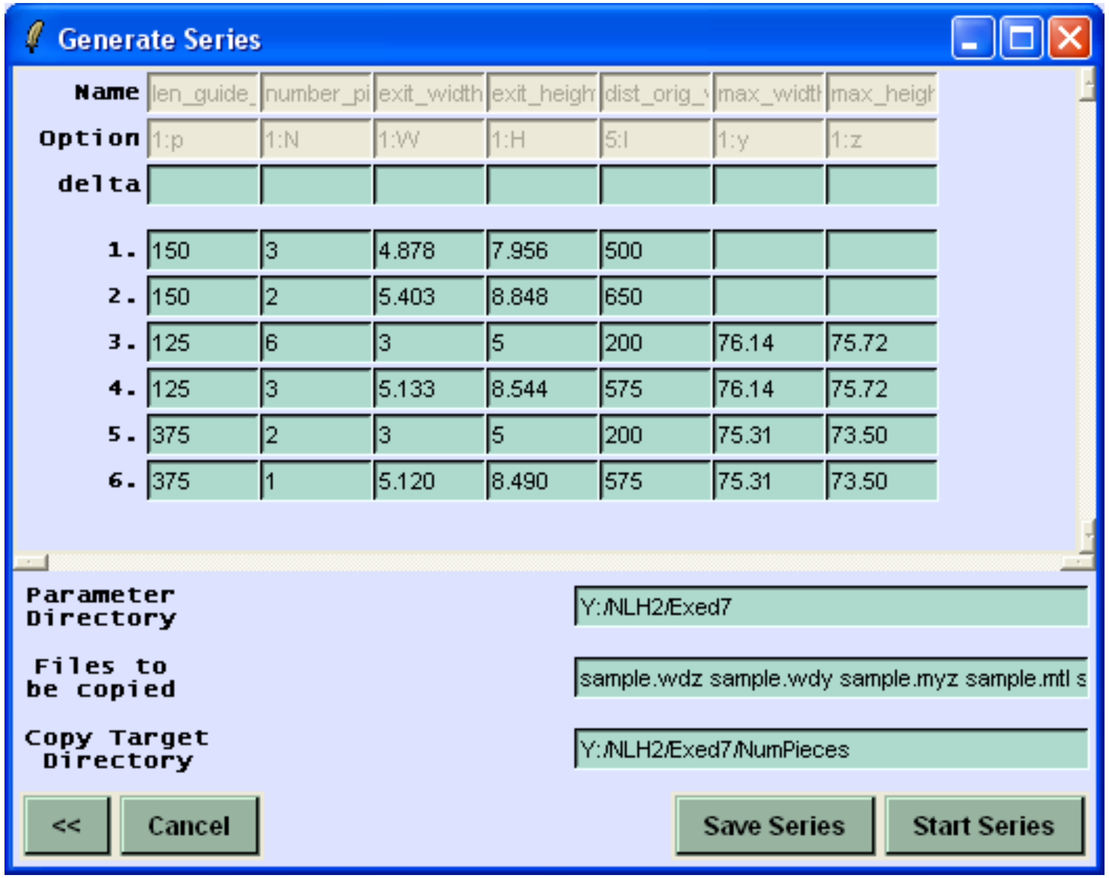


Graphical output from Monitors



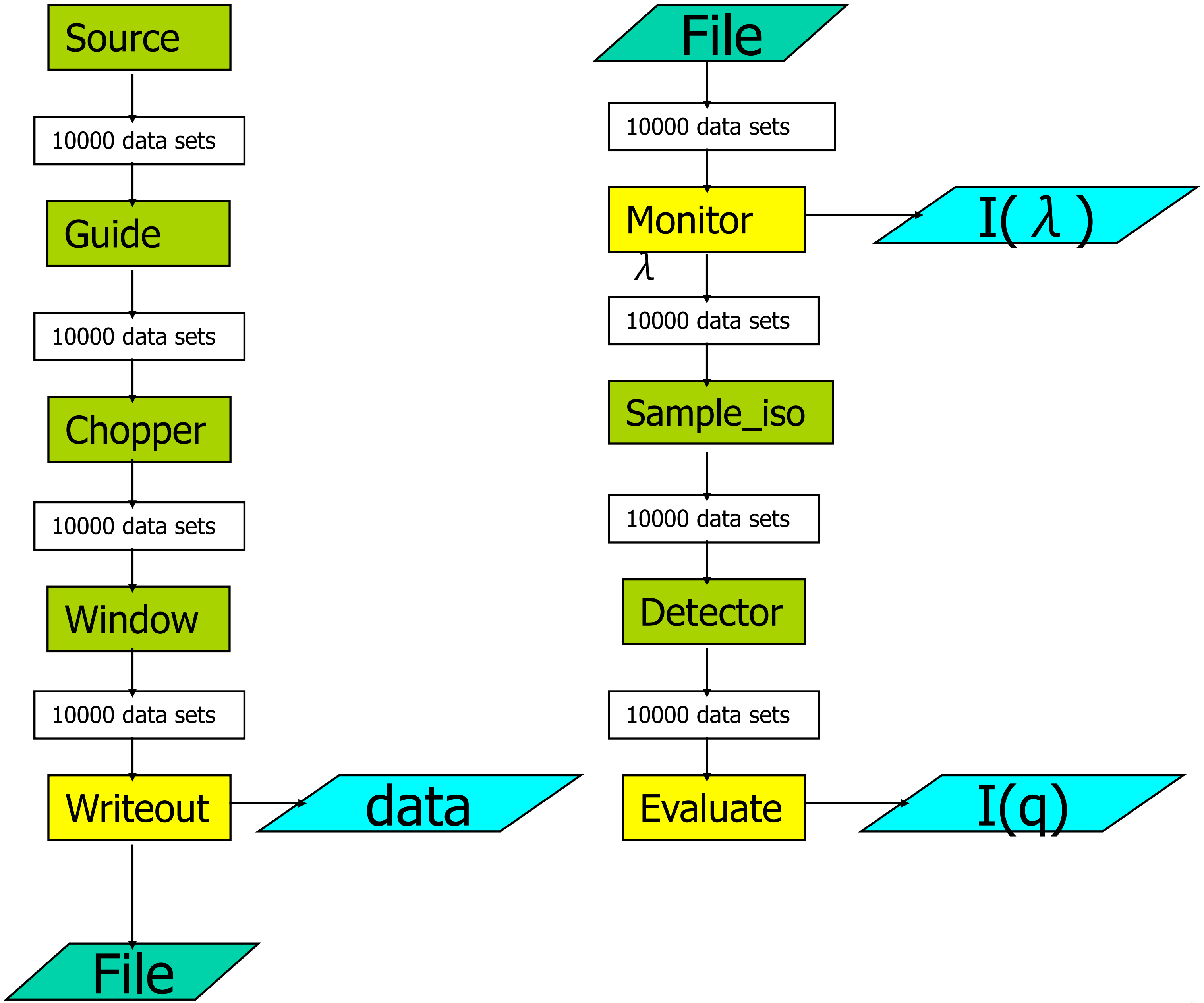
Generate Series





Name	len_guide_	number_pi	exit_width	exit_heigh	dist_orig_	max_widtl	max_heigh
Option	1:p	1:N	1:W	1:H	5:l	1:y	1:z
delta							
1.	150	3	4.878	7.956	500		
2.	150	2	5.403	8.848	650		
3.	125	6	3	5	200	76.14	75.72
4.	125	3	5.133	8.544	575	76.14	75.72
5.	375	2	3	5	200	75.31	73.50
6.	375	1	5.120	8.490	575	75.31	73.50

Splitting of the Simulation



Tools

- Most are used to calculate input data

AsciiToBinary	GenerMirrorFile	ChopPhases
DefineDirection	GenerSurfFile	ChopSystem
Direct View	StdDeviation	CrysAnalySpec
GenerateBatch	LatticeDist.	

- Others to visualize output

DistTimePlot	VisualOutput

Parallel Computing

Computer Grids

File|'Save as Grid Command'

saves the command line in a form that it can be used on computer clusters

Multi-core Processors

Ongoing development

Tests made for

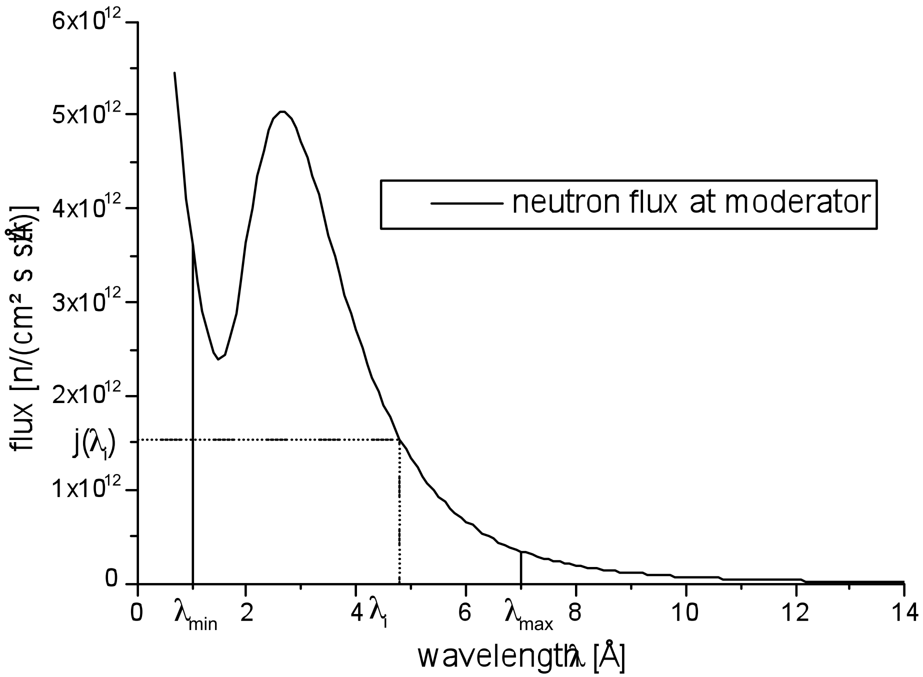
- Guide
- Fermi chopper
- supermirror ensemble
- polariser supermirror

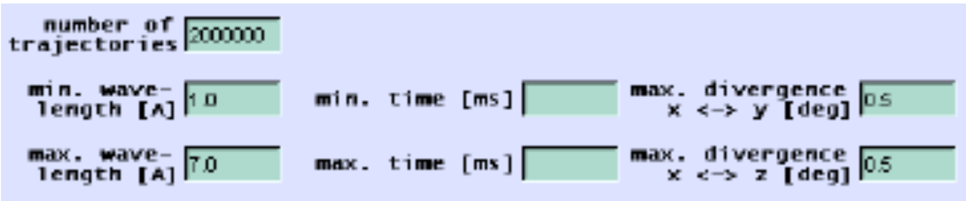
Support

Address to Michael Fromme (fromme@helmholtz-berlin.de)

Thank you for your attention !

Absolute Flux Values





- $I_{CW} = \int j(\lambda)\, d\lambda$

  $\approx (\lambda_{max}-\lambda_{min})/N \sum_i j(\lambda_i)$

- $I_{SS} \approx (\lambda_{max}-\lambda_{min})(t_{max}-t_{min})/N \sum j(\lambda_i,t_i)$

- Each trajectory represents a package of a certain number of neutrons with the same starting conditions
- By statistical processes like reflection, the number of neutrons in the package decreases, while the number trajectories remains unchanged
- If the neutron package does not pass a module regularly, the trajectory is taken out of consideration
- A neutron count rate can be calculated from the number of neutrons in a package
- Summing of the count rates of all packages gives the neutron count rate at any point of the instrument

Welcome, day 2

Tuesday morning program:

9:00-9:15 Today's strategy / forming groups
- VITESS / VITESS + McStas groups in conference room
- McStas-only groups in meeting room

9:15-9:45 McStas guided hands-on session

9:45-10:15 VITESS guided hands-on session

10:15-10:30 Coffee

10:30-10:45 Presentation of the next exercises

10:45-12:30 Guide exercises

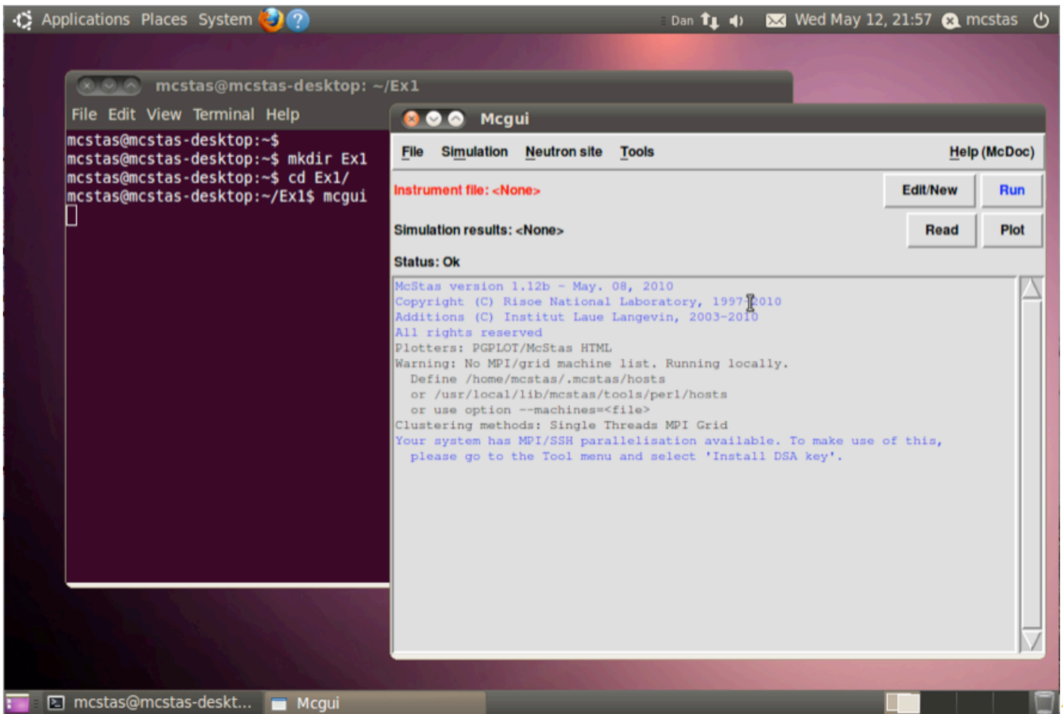
13:00 Lunch

Ex. 1: Basics, Source, Monitors, Guides

1.1: mcgui, editor, Source_simple, Guide, Monitors

Starting mcgui

- Start a terminal (OS dependent)
- cd to a dir of choice
- Issue the mcgui command



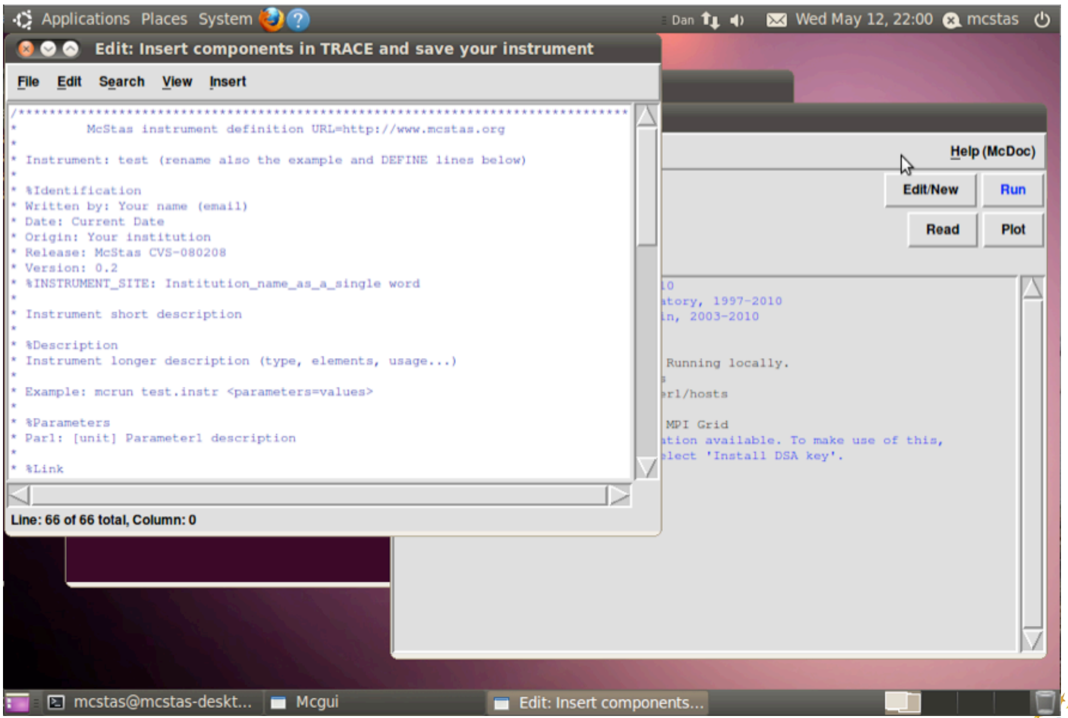
Ven 2010
essworkshop.org

Ex. 1 / Peter Willendrup

Press Edit/New to create a new file

  On emerging window, choose Insert - Instrument template

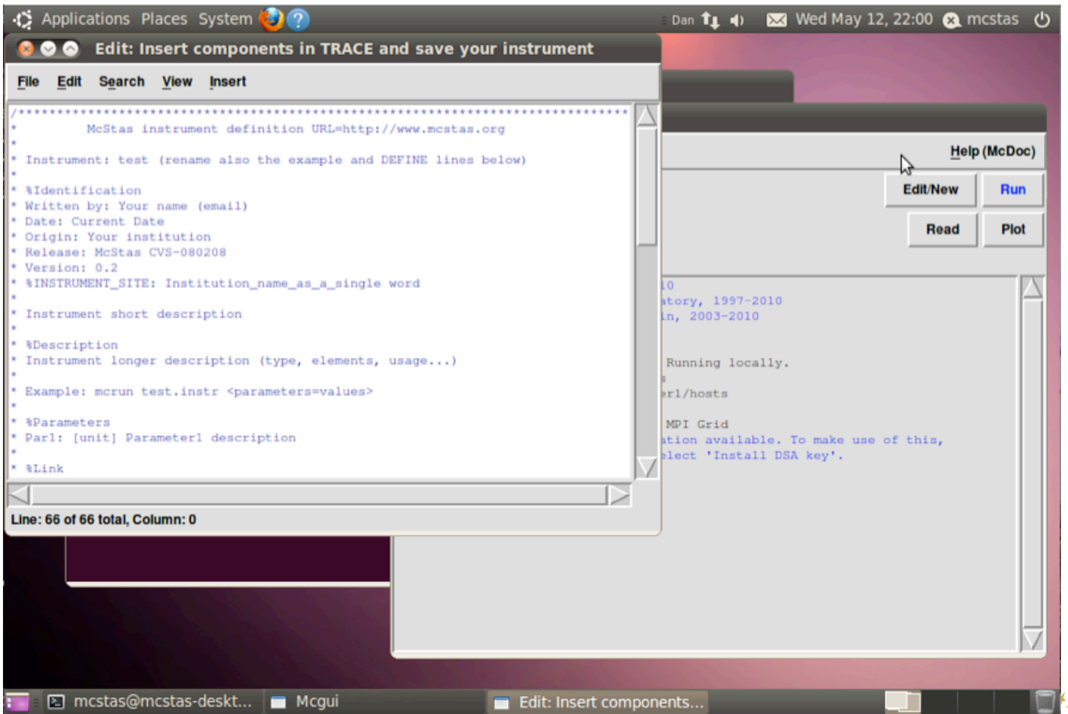
  Choose File - Save As - Ex01.instr



Ven 2010
essworkshop.org

Scroll to TRACE section and insert cursor after the Origin comp

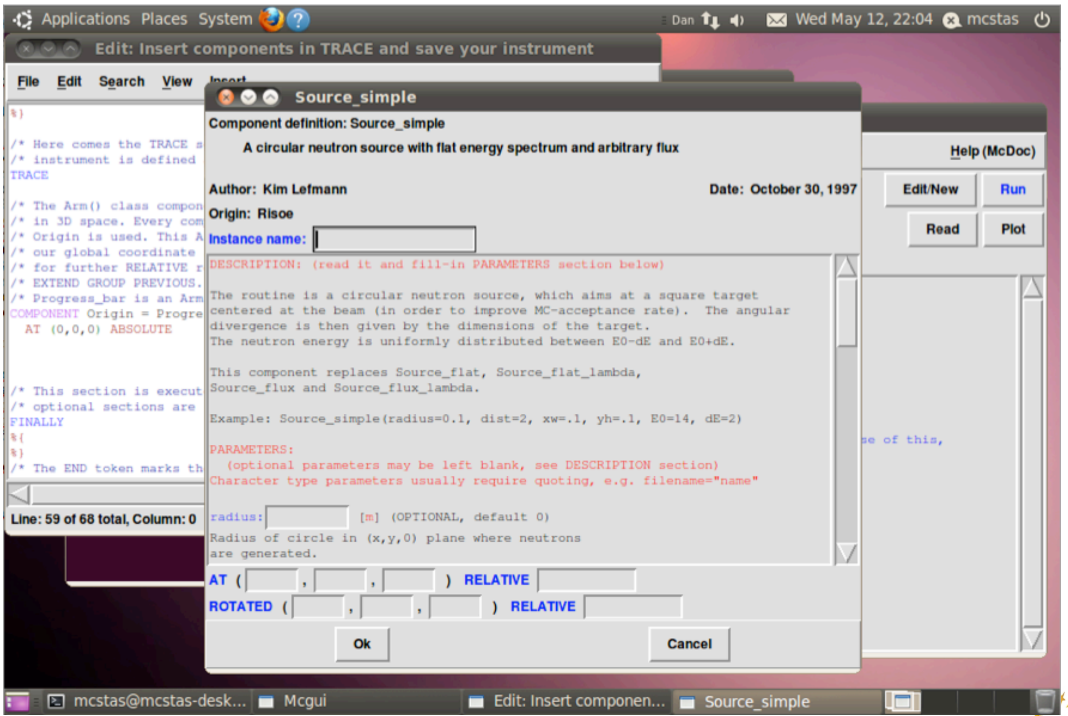
  Chose Insert - Source - Source_simple



Ven 2010
essworkshop.org

Scroll to TRACE section and insert cursor after the Origin comp

Chose Insert - Source - Source_simple

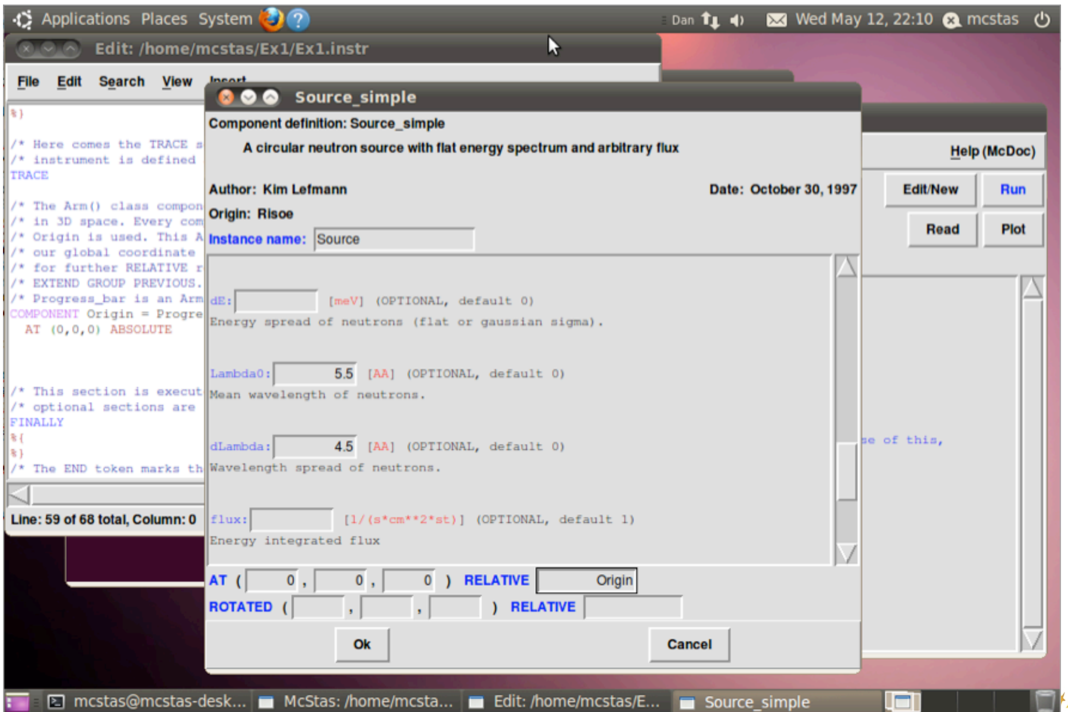


Ex. 1 / Peter Willendrup

Name the component Source

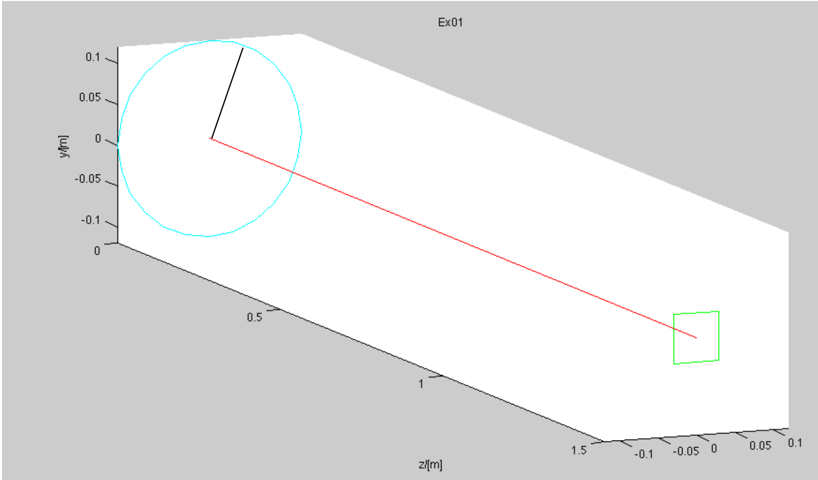
Choose parameters: radius=0.12, Lambda0=5.5, dLambda=4.5, dist=1.5, xw=0.06, yh=0.06

Place the comp at (0,0,0) RELATIVE Origin



Ex. 1 / Peter Willendrup

Significance of Source_simple parameters



Input parameters

Parameters in **boldface** are required; the others are optional.

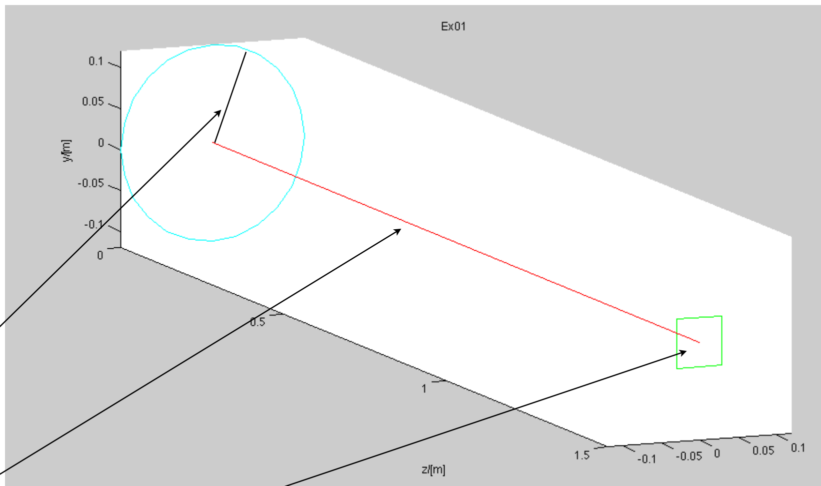
Name	Unit	Description	Default
radius	m	Radius of circle in (x,y,0) plane where neutrons are generated.	0
height	m	Height of rectangle in (x,y,0) plane where neutrons are generated.	0
width	m	Width of rectangle in (x,y,0) plane where neutrons are generated.	0
dist	m	Distance to target along z axis.	
xw	m	Width(x) of target	
yh	m	Height(y) of target	
E0	meV	Mean energy of neutrons.	0
dE	meV	Energy spread of neutrons (flat or gaussian sigma).	0
Lambda0	AA	Mean wavelength of neutrons.	0
dLambda	AA	Wavelength spread of neutrons.	0
flux	1/(s*cm**2*st)	Energy integrated flux	1
gauss	1	Gaussian (1) or Flat (0) energy/wavelength distribution	0
compat	1	Apply weighting/sampling as now obsolete Source_flat component	0

Ex. 1 / Peter Willendrup

Ven 2010
essworkshop.org

nmi3 | Risø DTU

Significance of Source_simple parameters



Input parameters

Parameters in **boldface** are required; the others are optional.

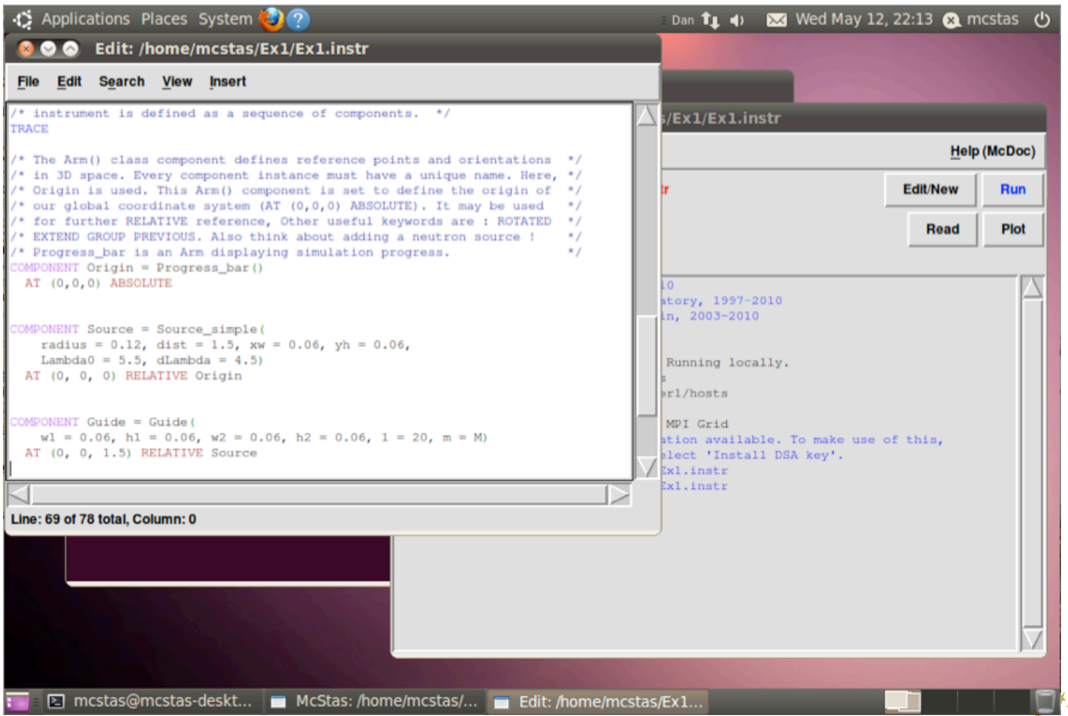
Name	Unit	Description	Default
radius	m	Radius of circle in (x,y,0) plane where neutrons are generated.	0
height	m	Height of rectangle in (x,y,0) plane where neutrons are generated.	0
width	m	Width of rectangle in (x,y,0) plane where neutrons are generated.	0
dist	m	Distance to target along z axis.	
xw	m	Width(x) of target	
yh	m	Height(y) of target	
E0	meV	Mean energy of neutrons.	0
dE	meV	Energy spread of neutrons (flat or gaussian sigma).	0
Lambda0	AA	Mean wavelength of neutrons.	0
dLambda	AA	Wavelength spread of neutrons.	0
flux	1/(s*cm**2*st)	Energy integrated flux	1
gauss	1	Gaussian (1) or Flat (0) energy/wavelength distribution	0
compat	1	Apply weighting/sampling as now obsolete Source_flat component	0

Ex. 1 / Peter Willendrup

Ven 2010
essworkshop.org

nmi3 | Risø DTU

Insert Optics - Guide - of dimension 0.06 x 0.06 m, length 20 m, 1.5 meters along z after Source. Use an m value of ‘M’. Name the component.



Significance of Guide input parms

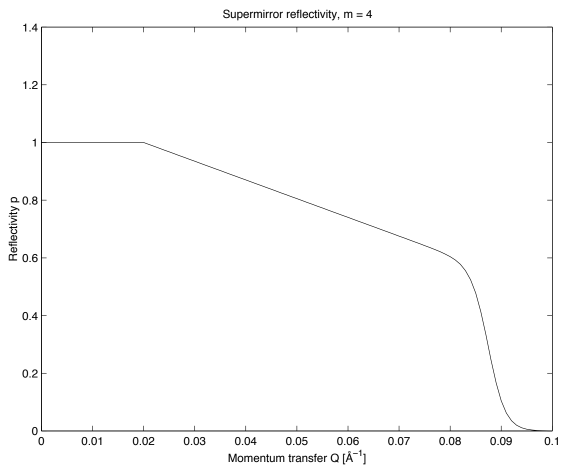


Figure 5.1: A typical reflectivity curve for a supermirror, Eq. (5.2). The used values are $m = 4$, $R_0 = 1$, $Q_c = 0.02$ Å^{-1}, $\alpha = 6.49$ Å, $W = 1/300$ Å^{-1}.

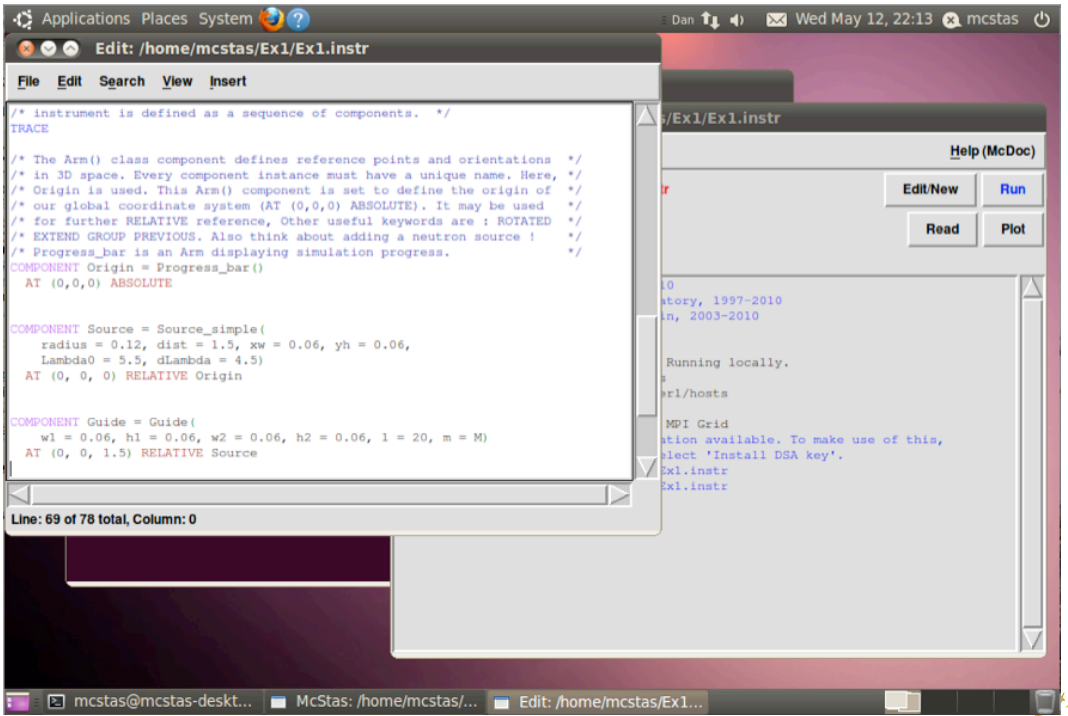
Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
reflect	str	Reflectivity file name. Format [q(Angs-1) R(0-1)]	0
w1	m	Width at the guide entry	
h1	m	Height at the guide entry	
w2	m	Width at the guide exit	
h2	m	Height at the guide exit	
l	m	length of guide	
R0	1	Low-angle reflectivity	0.99
Qc	AA-1	Critical scattering vector	0.0219
alpha	AA	Slope of reflectivity	6.07
m	1	m-value of material. Zero means completely absorbing.	2
W	AA-1	Width of supermirror cut-off	0.003

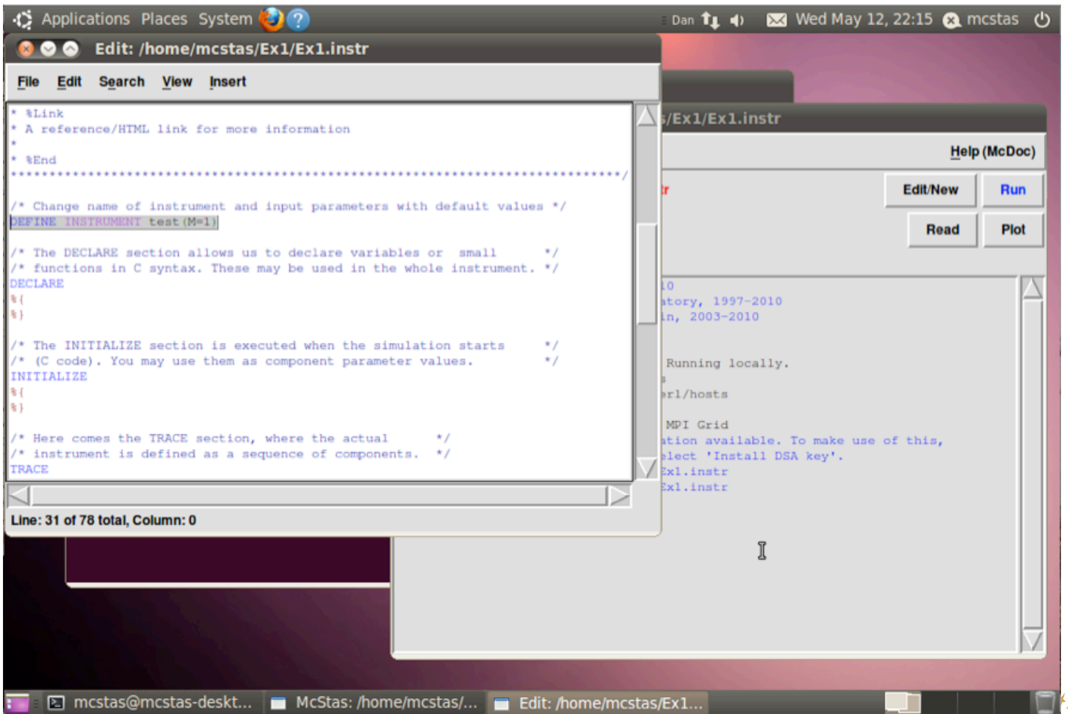
Ven 2010
essworkshop.org

Scroll to the top of the window and locate the
DEFINE INSTRUMENT Test(Par1=1)
line. Define an input parameter called M, with a default value of 1



Ex. 1 / Peter Willendrup

Scroll to the top of the window and locate the
DEFINE INSTRUMENT Test(Par1=1)
line. Define an input parameter called M, with a default value of 1



Ex. 1 / Peter Willendrup

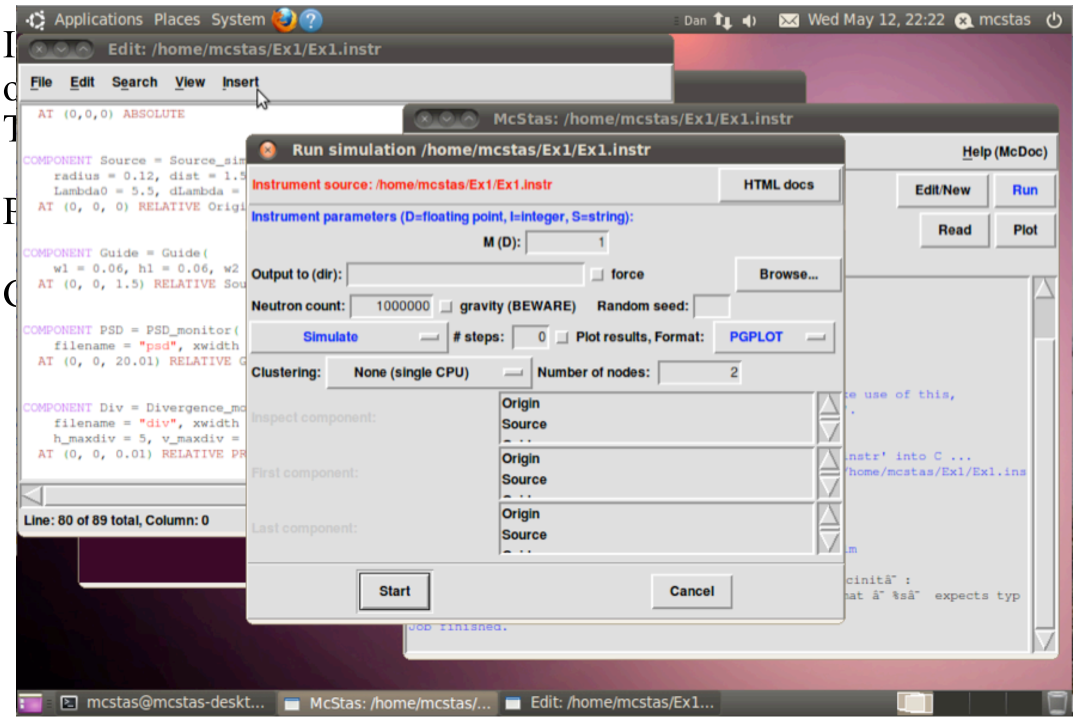
Insert a PSD monitor of dimension 0.07 x 0.07 m, define an output filename, AT (0,0,20.01) RELATIVE Guide

Insert a Divergence monitor of dimension 0.07 x 0.07 m, define an output filename, maximum divergence 5 degrees in both directions. To be placed AT (0,0,0.01) RELATIVE PREVIOUS

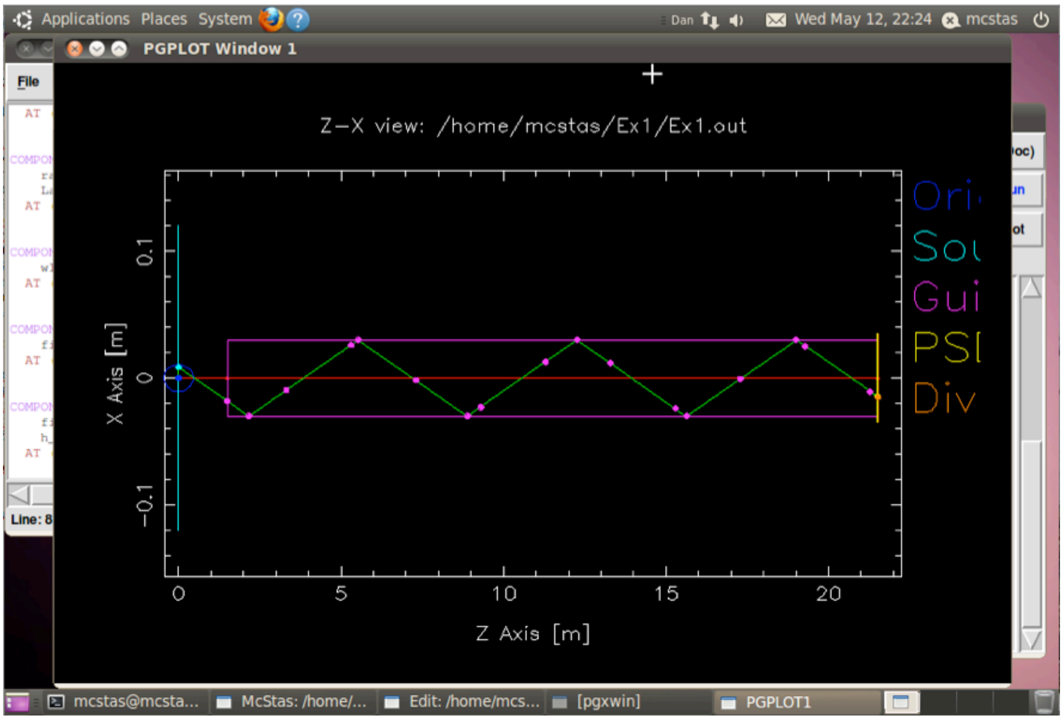
Press save

Go on the main window, press run, you should now get....

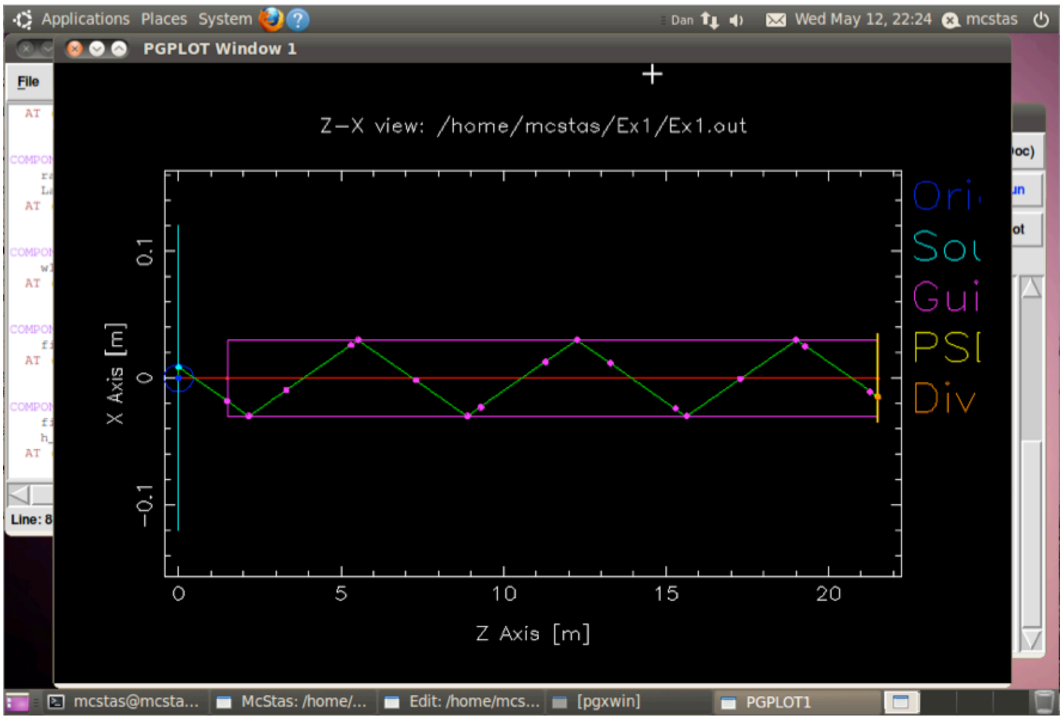
Insert a PSD monitor of dimension 0.07 x 0.07 m, define an output filename, AT (0,0,20.01) RELATIVE Guide



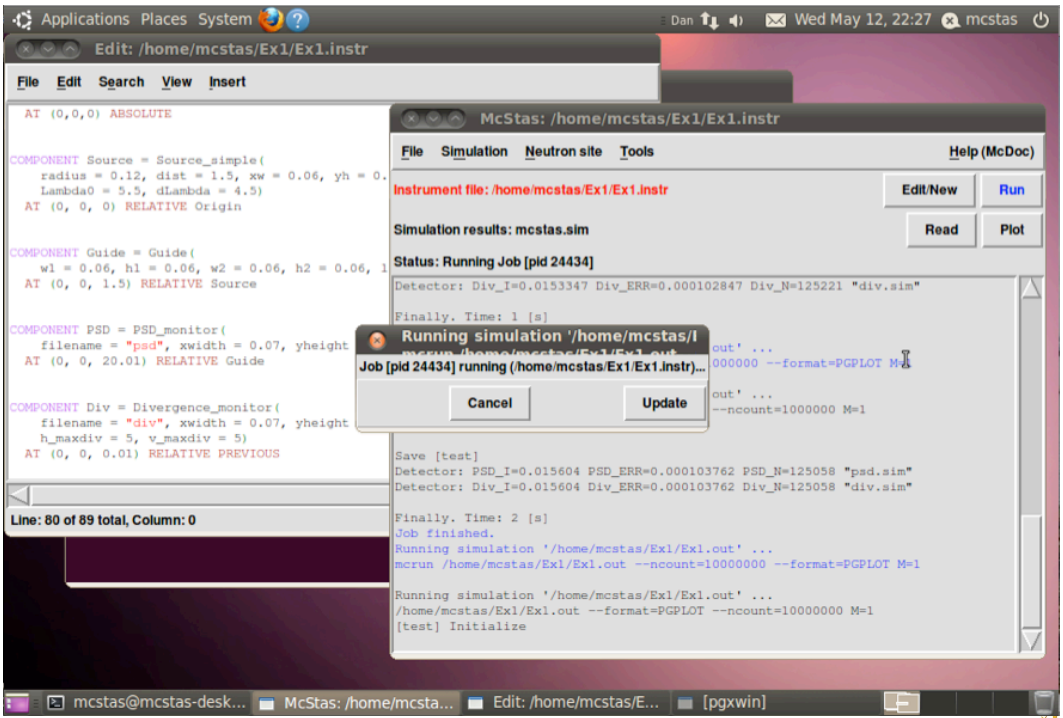
Select the 'TRACE' mode and press Start - you will get a view of the instrument. Try zooming (place cursor, press z, drag, click)



Right-click to unzooom.
Click a few times and see the visualization of neutron rays



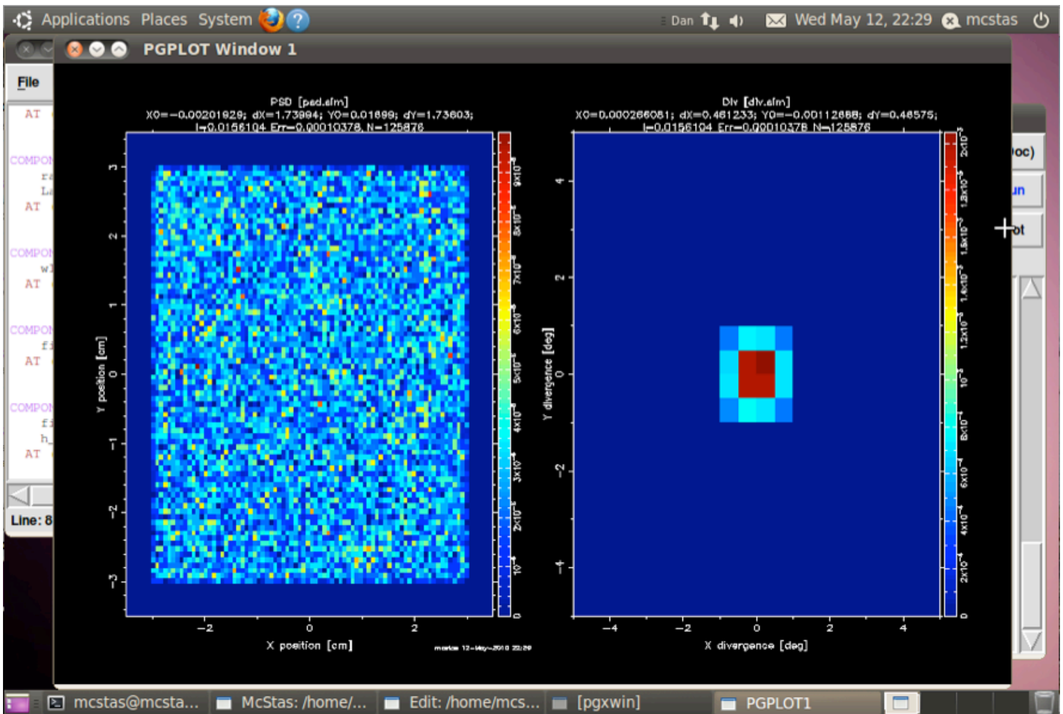
Press ‘q’ to exit the visualisation, close the window.
Press run again and choose simulate mode, start
Once the simulation terminates, press Plot and you will get...



Ex. 1 / Peter Willendrup

Ven 2010
essworkshop.org

Press ‘q’ to exit the visualisation, close the window.
Press run again and choose simulate mode, start
Once the simulation terminates, press Plot and you will get...



Ex. 1 / Peter Willendrup

Ven 2010
essworkshop.org

Clicking one of the panels will zoom that monitor, clicking again zoom out
Shortcut keys:

Click on a plot for full-window view.
Press key for hardcopy (in graphics window), 'Q' to quit
'P' BW postscript
'C' color postscript
'N' PNG file
'M' PPM file
'G' GIF file
'L' Toggle log10 plotting mode
'T' Toggle contour plotting mode
'Q' quit

On the run dialogue, we will now:

1) Define an output directory (otherwise subsequent sims will overwrite results)

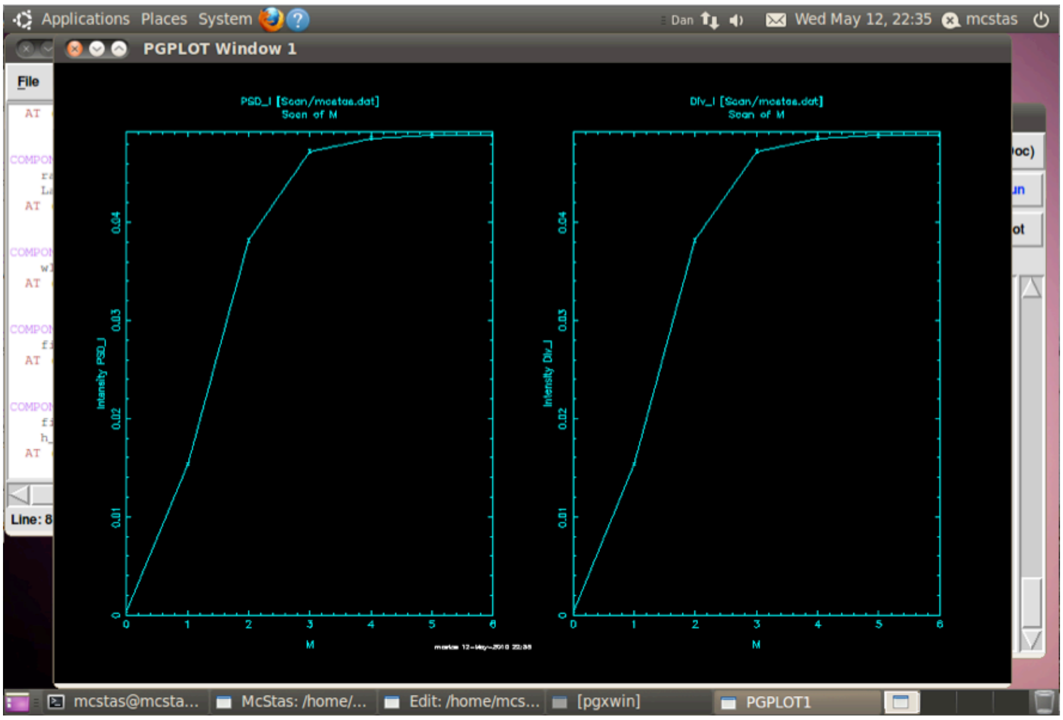
2) Perform a scan by
a) Setting 0,6 for the value of M
b) Fill the ‘steps’ field by the number 7

A series (7) of simulations will now run corresponding to:

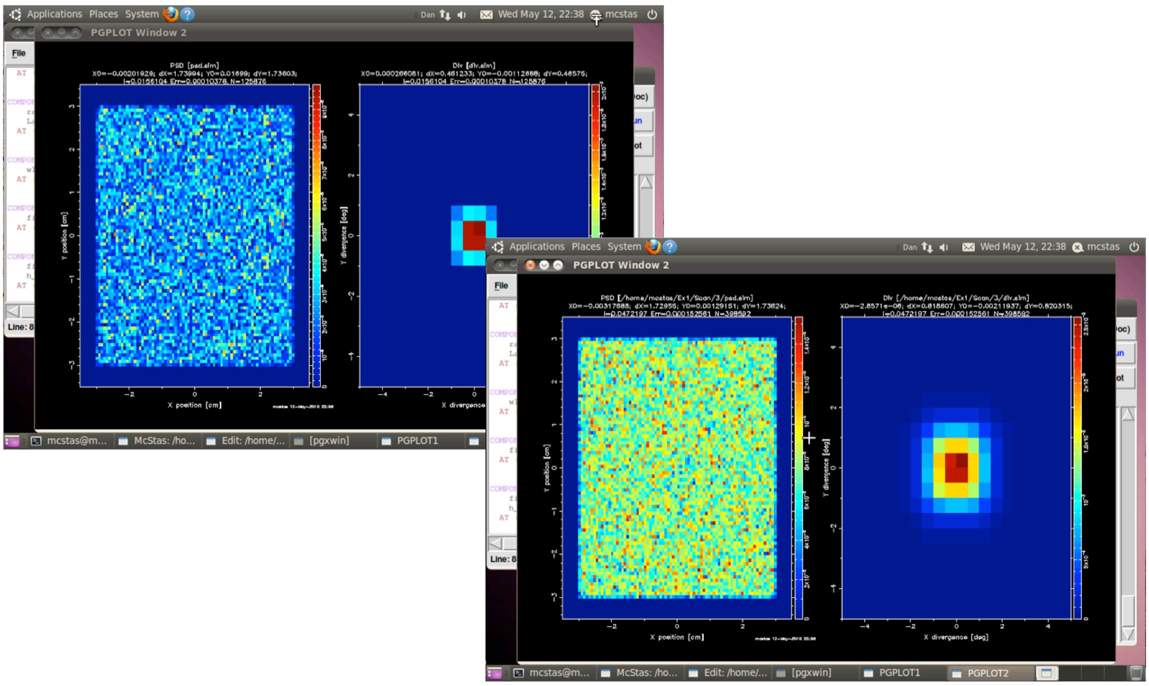
M=0 - simple, non-reflecting beamtube (absorbing walls)
M=1,6 - Guide mirrors of increasing quality

After performing the scan, press Plot and you should get...

A report of integrated intensity of the monitors, as fct. of the scanned variable



Try using the Tools - Plot other results to compare the individual scan steps (browse to Scan/0 , Scan/3, ...)



Try using the Tools - Plot other results to compare the individual scan steps (browse to Scan/0 , Scan/3, ...)

*Task 1.1: **Source** + Linear Guide + Monitors*

1. Define new directory for simulation
   ‘parameter directory’ | “NewDir”   -> Browse + Give Name

2. Define Source

   - Module 1 – chose ‘inactive’|’source’|’source constant wave’
   - Show parameters by clicking on “->”
   - Give name of ‘moderator description file’, e.g. “constant.mod”
   - Choose “Edit” this file
   - Chose ‘shape’ “circular” and set ‘moderator diameter’ 12 cm as well as center of moderator X’, ‘...Y’ and ‘...Z’ = (0,0,0) cm
   - Give name of ‘user wavelength distribution file’, e.g. “constant.dat”
   - Give intensities: 0.0 Å   1.0e12   (first row)
     and   20.0 Å   1.0e12   (second row)
   - Finish with “Save+Close”
   - Give ‘min. wavelength’ and ‘max. wavelength’ 1 – 10 Å
   - Chose ‘direction defined’ “by virtual window”
   - Fill propagation with ‘Distance to window’ 150 cm
     ‘window width’ and ‘window height’ 6 cm
   - SAVE as ‘GuideLinear.gui’

Task 1.1: *Source +* ***Linear Guide*** *+ Monitors*

3. Define Guide
   - Module 2 – chose ‘inactive’|’guide’|’guide’
   - Set ‘entrance width’, ‘... height’, ‘exit width’ and ‘... height’ = 10 cm
   - Switch “AutoPlot” off
   - Give ‘piece length’ (2000 cm)
   - Browse *InstallationDirectory*|FILES|reflectivity files|mirr1a.dat to fill ‘left plane’ to have a m=1 coating
   - SAVE instrument

4. Include Space
   - Make space for a new module by clicking on ‘arrow_down’ of module 2
   - Module 2 – chose ‘inactive’|’space and window’|’space’
   - Give ‘distance’ 150 cm

Task 1.1: Source + Linear Guide + *Monitors*

4. Define Position Monitor
    - Module 4 – chose ‘inactive’|’visualize data’|’mon2_pos’
    - Set ‘minimal y-value’ and ‘minimal z-value’ to -3.5
    - Set ‘maximal y-value’ and ‘maximal z-value’ to 3.5
    - Set ‘number y-bins’ and ‘number z-bins’ to 70
5. Define Divergence Monitor
    - Module 5 – chose ‘inactive’|’visualize data’|’mon2_div’
    - Set all ‘minimal ...’ and ‘maximal ...’ values to 5
    - Switch “AutoPlot” off
    - SAVE instrument

6. Finish
    - “Check”, “Start”
    - Check log file
    - Check by looking at ‘File’|’Edit *.inf file’|instrument.inf

Task 1.1: *Source + Linear Guide + Monitors –* ***vary m-value***

5. Vary m-value of guide

- Copy mirr30opt, mirr40opt from *InstallationDirectory*|FILES|reflectivity files to *parameter diectory*
- Start ‘Tools|GenerateMirrorFiles’ and give
  - reflectivity(Q=0): 1
  - m =... : 2
  - Q_c = ... : 0.0217
  - reflectivity(m*Q_c): 0.95
  - Width : 0.0033
  - Name : “mirr20opt.dat” and terminate
- Click on the texts ’left plane’, ’right plane’and ‘top plane’ of the guide module
- Chose ‘File|GenerateSeries’
- Set 4 Iterations
- Go “>>” and fill table with ‘mirr1a.dat’, ‘mirr20opt.dat’, ‘mirr30opt.dat’ and ‘mirr40opt.dat’
- Fill ‘files to be copied’ with ‘pos.dat’, ‘div.dat’ and ‘instrument.inf’
- SAVE instrument and START

Ex. 1: Basics, Source, Monitors, Guides, continued

1.2-4, curved, ballistic, elliptic and parabolic guides

1.2: Curved guide:

Open the instrumentfile Ex_1_2.instr given to you

Study the instrumentfile, notice use of the PREVIOUS keyword

Notice input parameters of guide m-value, angular rotation of guide segments

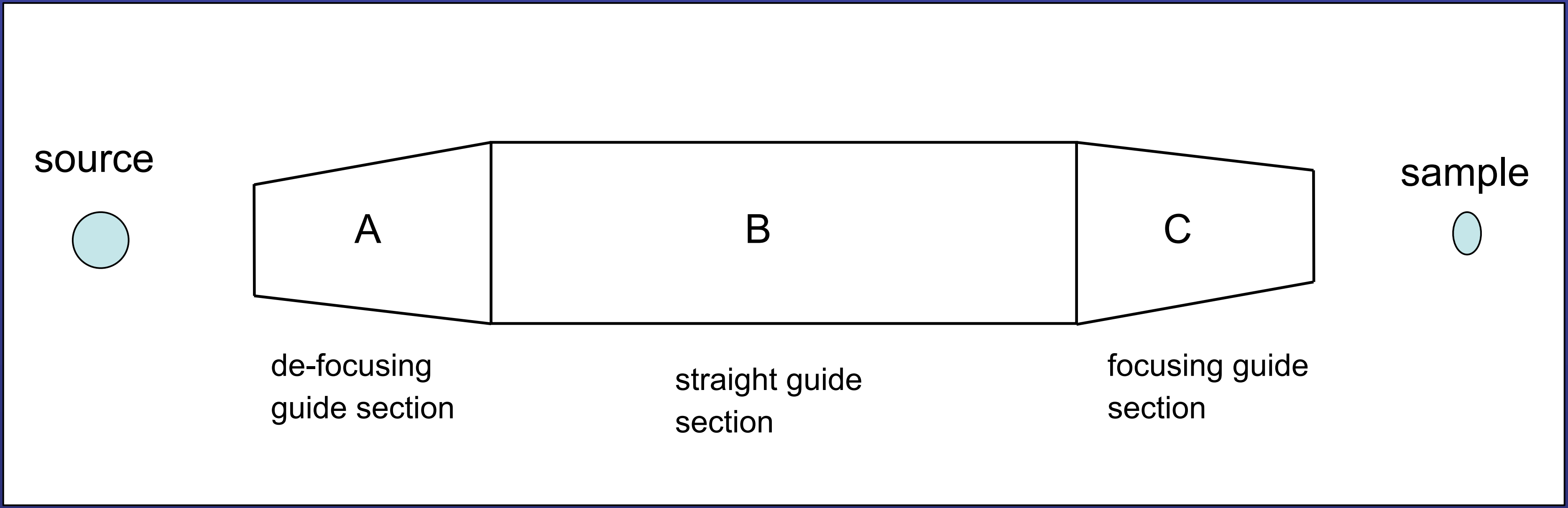
Question: What is the relevant rotation angle to achieve a guide curvature of 1 km?

Try performing a TRACE

Try varying the guide curvature, notice effect on divergence and beam profile

Other curved guides:
 Use McDoc -> Component Library Index to look at Guide_curved plus Bender from the McStas lib

1.3 Ballistic Guides



Goal : transport/focus more neutrons at the sample position

Disadvantage: increasing neutron divergence

Simulation: using standard guide component

1.3: Ballistic guide:

Open the instrumentfile Ex_1_3.instr given to you

Study the instrumentfile, notice use of the DECLARE and INITIALIZE sections

Notice the use of Source_gen to describe the PSI cold source

Notice the input parameter sa_pos, to vary the guide - sample position distance.

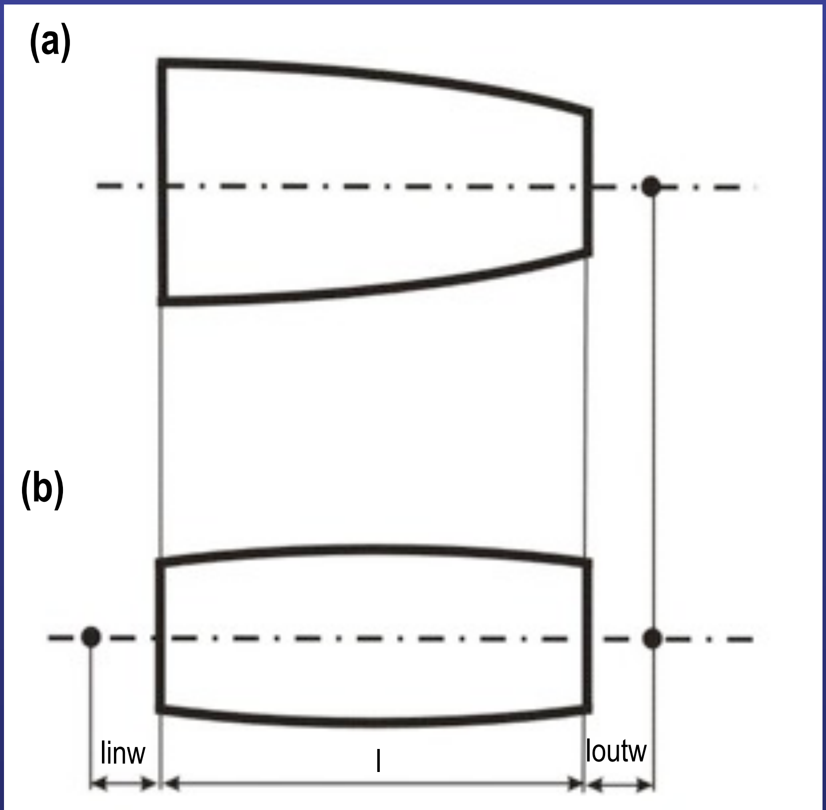
Compile and TRACE to have an overview of the instrument.

Run a simulation and notice the wavelength distr. before and after guide.

Task: Scan sa_pos between 0 and 1 m in 11 steps. Notice the effect on beam profiles and divergence.

1.4 Elliptic / parabolic Guides

Guide_tapering Component



Parameters for the parabolic (a) and elliptic (b) focusing guide in x-plane

```
COMPONENT cguide = Guide_tapering (
   w1 = 0.035, h1 = 0.012, linw = 0, loutw = 0.3,
   l=1.0, linh=0, louth = 0.3, option="parabolical",
   R0 = 0.995, Qcx = 0.0217, Qcy = 0.0217,
   alphax = 4.954, alphay = 4.954, W = 0.003,
   mx = 3, my = 3, segno = 20)
AT (0,0,1.5) RELATIVE arm1 ROTATED (0,0,0) RELATIVE arm1
```

```
COMPONENT cguide = Guide_tapering (
   w1 = 0.035, h1 = 0.012, linw = 0.3, loutw = 0.3,
   l=10.0, linh=0.3, louth = 0.3, option="elliptical",
   R0 = 0.995, Qcx = 0.0217, Qcy = 0.0217,
   alphax = 4.954, alphay = 4.954, W = 0.003,
   mx = 3, my = 3, segno = 100)
AT (0,0,1.5) RELATIVE arm1 ROTATED (0,0,0) RELATIVE arm1
```

```
COMPONENT cguide = Guide_tapering (
   w1 = 0.035, h1 = 0.012, linw = 0.3, loutw = 0.3,
   l=10.0, linh=0.3, louth = 0.3, option="file=input.dat",
   R0 = 0.995, Qcx = 0.0217, Qcy = 0.0217,
   alphax = 4.954, alphay = 4.954, W = 0.003,
   mx = 3, my = 3, segno = 100)
AT (0,0,1.5) RELATIVE arm1 ROTATED (0,0,0) RELATIVE arm1
```

1.4: Elliptic guide:

Open the instrumentfile Ex_1_4.instr given to you

Notice the smaller moderator surface, for optimal use of the elliptic guide

Notice the extra input parameter fp, for definition of the guide exit focal point.

Compile and TRACE to have an overview of the instrument.

Run a simulation and notice the wavelength distr. before and after guide. Compare with ballistic guide.

Task: At sa_pos fixed at 0.5 m, vary fp between 0 and 1 m in 11 steps. Notice the effect on beam profiles and divergence. Compare with parabolic guide (Ex_1_4a.instr).

Task 1.2: Curved Guide

1. Save as a new instrument
   - ‘Configure’|‘Set Instrument Name’ to “GuideCurved”
2. Change horizontal shape to ‘curved’, set a radius of curvature of 1000 m and the guide to consist of 20 pieces of 100 cm each

Task 1.3: Ballistic Guide

1. Start from the 'Simple Guide'. Save as a new instrument
2. Use 3 times the guide module to build a ballistic guide (in both dimensions) that
   - opens to twice the cross-section over the first 5 m (shape: linear)
   - has constant cross-section over the followin 10 m (shape: constant)
   - and converges to the original cross-section over the last 5 m (shape: linear)

Task 1.4: Elliptic Guide

1. Start from the ‘Curved Guide’. Save as a new instrument
2. Change both shapes to ‘elliptic’ and give 100 cm as distance from guide exit to the focal points

1.5 Capture flux estimates

- *Definition of capture flux*
- *Measurement and accuracy*
- *McStas simulation*

1.6 Guide losses

- *Mechanisms involved*
- *Simulation*

E. Farhi, ILL Computing for Science

Disclaimer: in case of errors and uncertainties, please correct me...

Capture flux: definition

The *'capture flux'* is the standard way to measure an integrated flux in facilities. A white beam is absorbed into a gold foil, in an energy range up to 500 meV neutrons. Then this is normalized to the thermal neutron absorption cross section for λ=1.8 Å (2200 m/s), and finally:

$$\Phi_c = \int_0^{0.5eV} \frac{d\Phi}{d\lambda} \frac{\lambda}{\lambda_{2200m/s}} d\lambda$$

Even though the formula is valid for thermal neutrons, it has been extended to cold and hot neutrons.

So, in a few words, the real integrated flux $\Phi = \int \frac{d\varphi}{d\lambda} d\lambda$

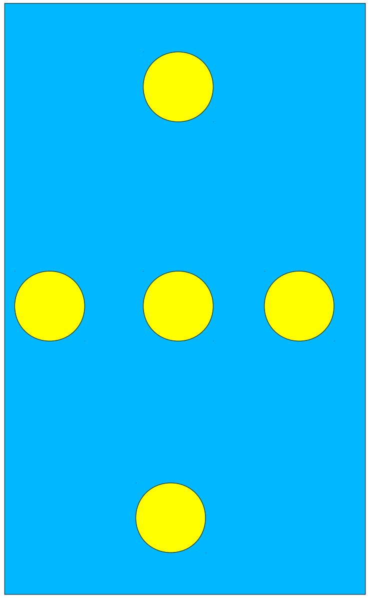
is roughly $\Phi_c \sim \Phi$ for a peaked flux around 1.8 Å

$\Phi_c \sim 2.2\ \Phi$ for a peaked flux around 4 Å

Capture flux = integrated, wavelength weighted flux

Capture flux: measurement and accuracy

The health physics/guide staff put 1 cm^2 gold foils in the beam, and they measure their activity after irradiation. The procedure is very standard, and unchanged for a long time.



σ_{abs}=98.65 σ_{coh}=7.32 σ_{inc}=0.43 [barns]

The intrinsic measurement accuracy of the method is of the order of 10 %.

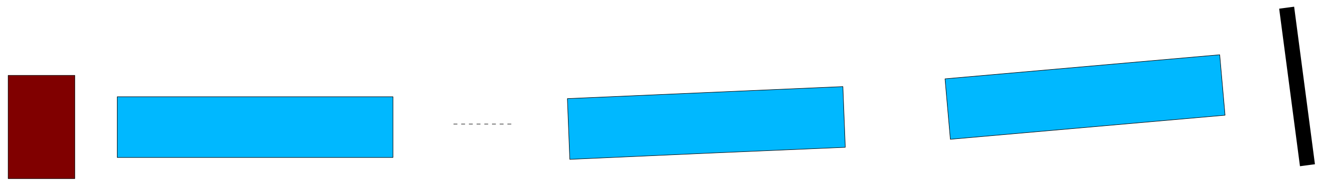
Simulating capture flux measurement with McStas:

```
Monitor_nD(options= "capture per cm2", …)
```

Capture flux: let's simulate !

Aim: Build a guide fed by a continuous thermal-cold source.

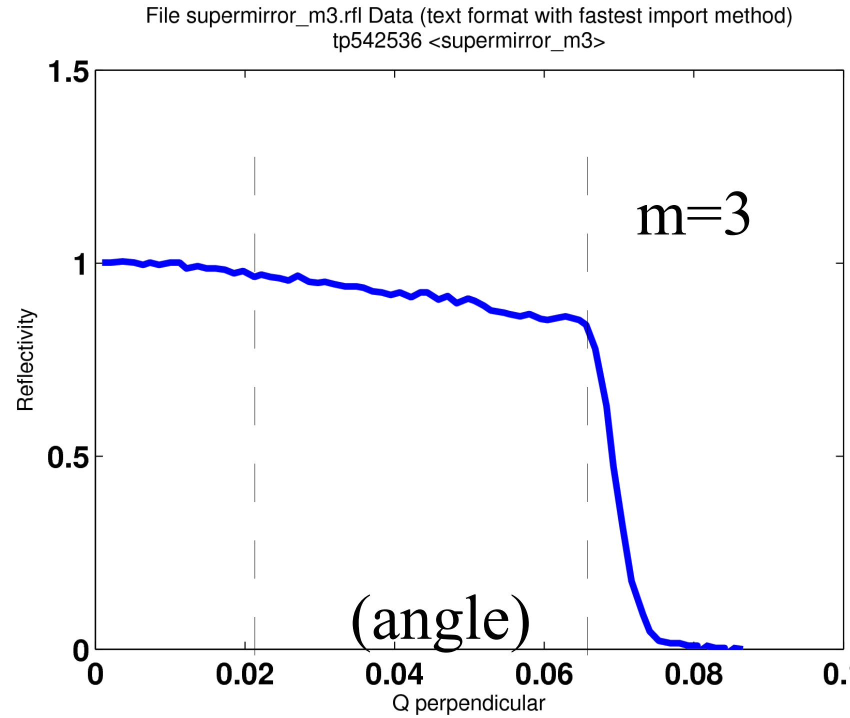
1) Re-use Ex 1.2 (curved guide)
2) At 50 cm from the end of the guide, add a *capture flux monitor*.
3) Simulate
4) Re-simulate with a reduced wavelength range, and then again with shifted range towards hot and cold neutrons. *What do you notice ?*



Guide losses: total reflection

Guides transporting neutrons are not 100 % efficient.

Their reflectivity depends on the material, number and quality of the multi-layers deposited on top of the glass or metal substrate surface.



Non reflected neutrons are either absorbed or scattered. In both cases, this creates background and radiation to protect from with proper shielding.

A rule of thumb says that maximum divergence transmitted by a guide is:

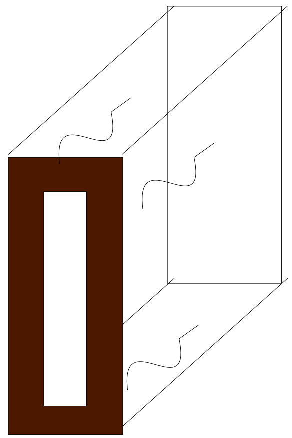
$$\alpha\ [deg] = m * lambda\ [Angs] * 0.1$$

Guide losses: causes for losses

Some causes of non reflection:

- too high divergence, above total reflection angle (depends on material cross section and *m*-value)
- low angle incoherent scattering
- poor waviness of surfaces (poor polishing)
- dirty surfaces (dust, grease, …)

In addition to radiations, the losses damage materials by creating He bubbles which propagate cracks. Glass turns dark and brittle.



Guide losses: estimating neutron losses

From the curved guide assembled previously:

1. Re-use Ex 1.5
2. Insert capture flux monitors in between guide elements
3. Run simulation with m=1 and m=3
4. *Estimate the losses per meter* (in absolute and percentage)
5. *Does super mirror coating increases background at the end of the guide ?*

2.1: Monochromators

Components

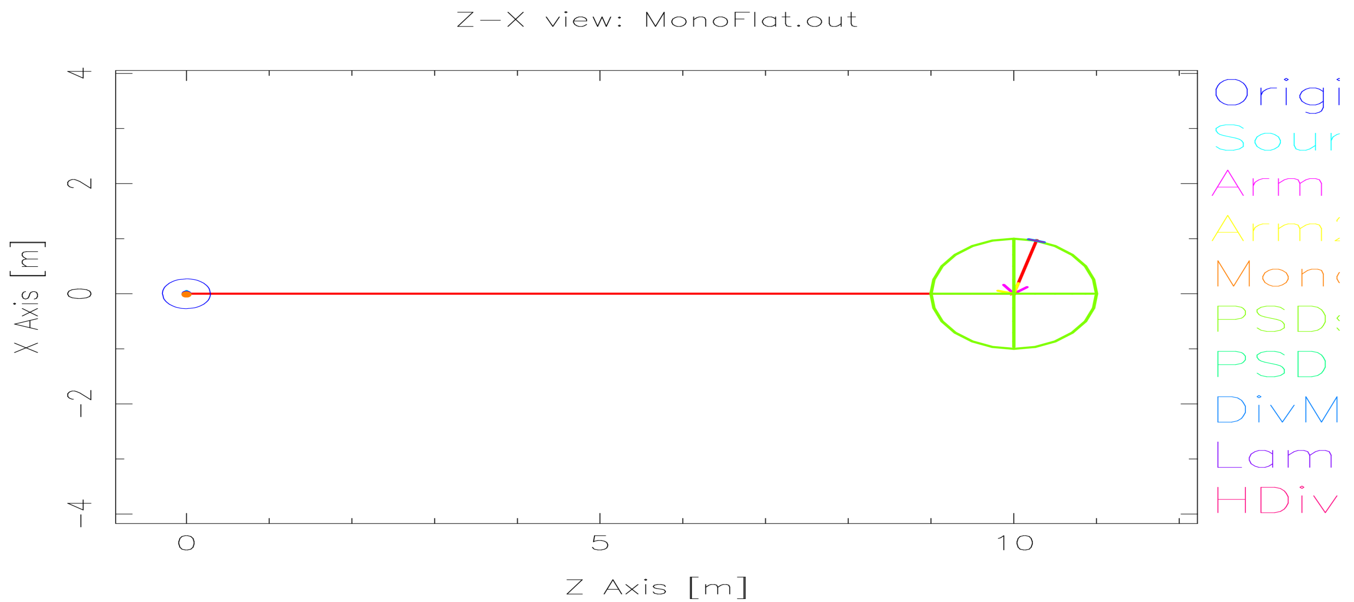
- Monochromator_flat (Ex 2.1.1)
- Monochromator_curved (Ex 2.1.2)
- Single_crystal (Ex 2.1.3)

Use in instrument

- Monochromator
- Analyser
- Sample

Build an instrument using

- Source_simple (0.1m*0.1m, dist=10, L0,dL, flat L distribution)
- Two Arm :one for rotation of mono and one for scattering
- Monochromator_flat (0.1m*0.1m @ z=10m, mosaic=40,r0=0.8,EXTEND if not scattered then absorb)
- PSD_monitor, Divergence_monitor, L_monitor



Properties:

- Infinitely thin, one scattering vector perpendicular to surface

  - no multiple scattering/secondary extinction

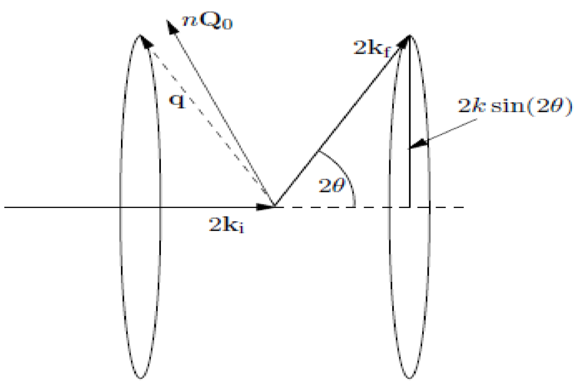
  - total reflectivity r0, not scattering cross sections

- Mosaic, vertical and horizontal η
- No variance of lattice parameter $\Delta d/d$=0

Algorithm:

- If intersect determine order n, $n\mathrm{Q}_0 = 2\mathrm{k}_i \sin\theta$
- From mosaicity η and angle α from Q_0 find prob $p_{\mathrm{reflect}} = R_0 e^{-\alpha^2/2\eta^2}$
- If reflected, determine direction on D-S cone
- Calculate weight for $\varphi \in [-\pi;\pi]$

$$f_{\mathrm{MC}}(\varphi) = \frac{1}{\sqrt{2\pi}(\alpha/\cos\theta)} e^{-\varphi^2/2(\alpha/\cos\theta)^2}$$



Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
zmin	m	Lower z-bound of crystal	0
zmax	m	Upper z-bound of crystal	0
ymin	m	Lower y-bound of crystal	0
ymax	m	Upper y-bound of crystal	0
width			0
height			0
mosaich	arc minutes	Horisontal mosaic (in Z direction) (FWHM)	30.0
mosaicv	arc minutes	Vertical mosaic (in Y direction) (FWHM)	30.0
r0	1	Maximum reflectivity	0.7
Q	AA-1	Magnitude of scattering vector	1.8734
DM	Angstrom	monochromator d-spacing instead of Q = 2*pi/DM	0

- `width = 0.1, height = 0.1,`
- `mosaich = MOSH, mosaicv = MOSV,`
- `r0 = 0.8, Q = 1.8734` (PG 002)

Basic setup

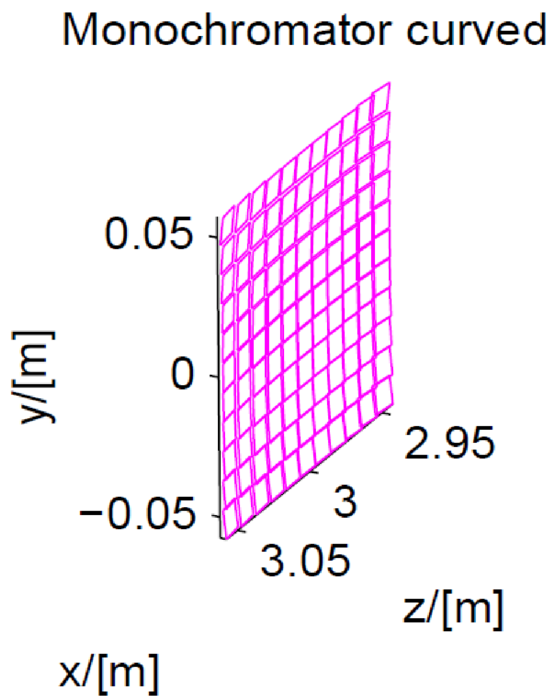
- Set source wavelength 4.0-4.1Å (`LMIN=4.0`, `LMAX=4.1`)
- Put mosaicity to 40 min (`MOSH=40`, `MOSV=40`)
- Set the monitors at the Bragg angle for the monochromator scattering for λ=4.045Å (rotate `a2`)
- Set monochromator rotation angle in scattering condition (`a1=a2/2`)
- Observe the wavelength distribution (n=1e6 rays is enough...)

Play!

- Try to put a broader wavelength interval from the source (2.0-4.1Å)
- Observe wavelength distribution
- Change to (vertical) mosaicity and observe the PSD
- Change the (horizontal) mosaicity and observe the energy monitor
- If you put a PSD_monitor_4PI (`radius=1-nm`) at the sample position you can confirm that only one scattering vector is present

Properties

- Array of single mosaic crystals (blades) with one scattering vector
- Infinitely thin, one scattering vector perp. to each surface of blade
  - no multiple scattering/secondary extinction
  - total reflectivity *r(k)*, not scattering cross sections
  - total transmission *t(k)*
- Mosaic, vertical and horizontal η
- No variance of lattice parameter $\Delta d/d$=0



Algorithm

For each individual blade the same as `Monochromator_flat`

2.1.2: Monochromator_curved

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
reflect	str	reflectivity file name of text file as 2 columns [k, R]	0
transmit	str	transmission file name of text file as 2 columns [k, T]	0
zwidth	m	horizontal width of an individual slab	0.01
yheight	m	vertical height of an individual slab	0.01
gap	m	typical gap between adjacent slabs	0.0005
NH	columns	number of slabs horizontal	11
NV	rows	number of slabs vertical	11
mosaich	arc minutes	Horisontal mosaic FWHM	30.0
mosaicv	arc minutes	Vertical mosaic FWHM	30.0
r0	1	Maximum reflectivity. O unactivates component	0.7
t0	1	transmission efficiency	1.0
Q	AA-1	Scattering vector	1.8734
RV	m	radius of vertical focussing. flat for 0	0
RH	m	radius of horizontal focussing. flat for 0	0
DM	Angstrom	monochromator d-spacing instead of Q=2*pi/DM	0
mosaic	arc minutes	sets mosaich=mosaicv	0
width	m	total width of monochromator	0
height	m	total height of monochromator	0
verbose	0/1	verbosity level	0

- 5 vertical slabs :`NV=5`, `yheight=0.02`, `zwidth=0.1`, `RV=1`
- Use reflecivity list '`HOPG.rfl`' provided in McStas datafiles
- Use transmission list '`HOPG.trm`' provided in McStas datafiles
- `r0 = 1,Q = 1.8734` (PG 002)

Basic setup

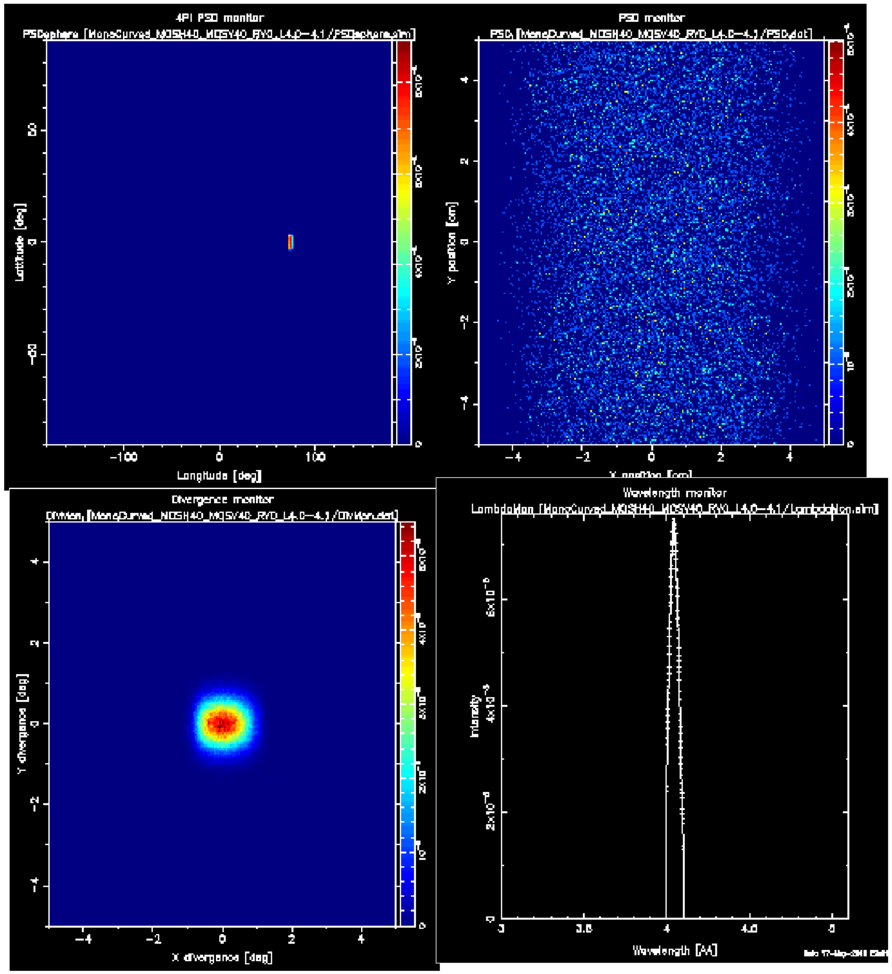
- Set source wavelength 4.0-4.1Å (`LMIN=4.0`, `LMAX=4.1`)
- Put mosaicity to 40 min (`MOSH=40`, `MOSV=40`)
- Set monochromator rotation angle `a1` in scattering condition
- Set the monitors `a2` at the Bragg angle for the monochromator scattering
- Observe the wavelength distribution (n=1e6 is enough...)

Play!

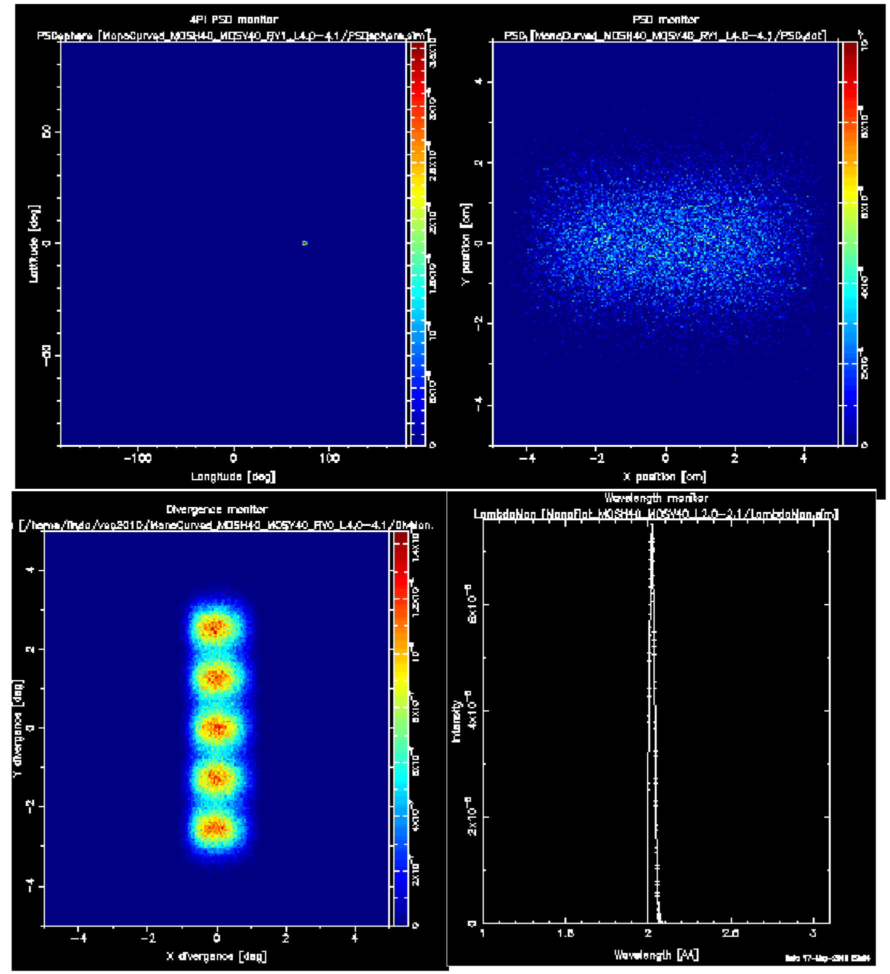
- Observe the influence of the focusing monochromator on the PSD (you can put it flat by setting `RV=0`)
- Observe the influence of the focusing monochromator on the divergence
- You can change the incoming wavelength (2.0-2.1 Å, second order scattering) and observe the intensity is smaller due to smaller reflectivity in comparison to constant r0

2.1.2: Monochromator_curved

No focus

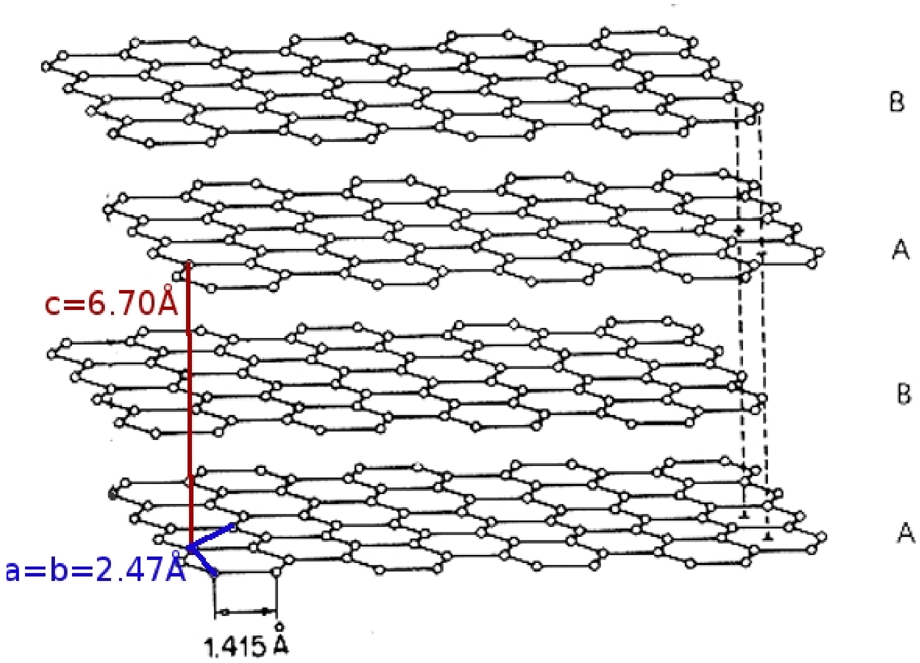


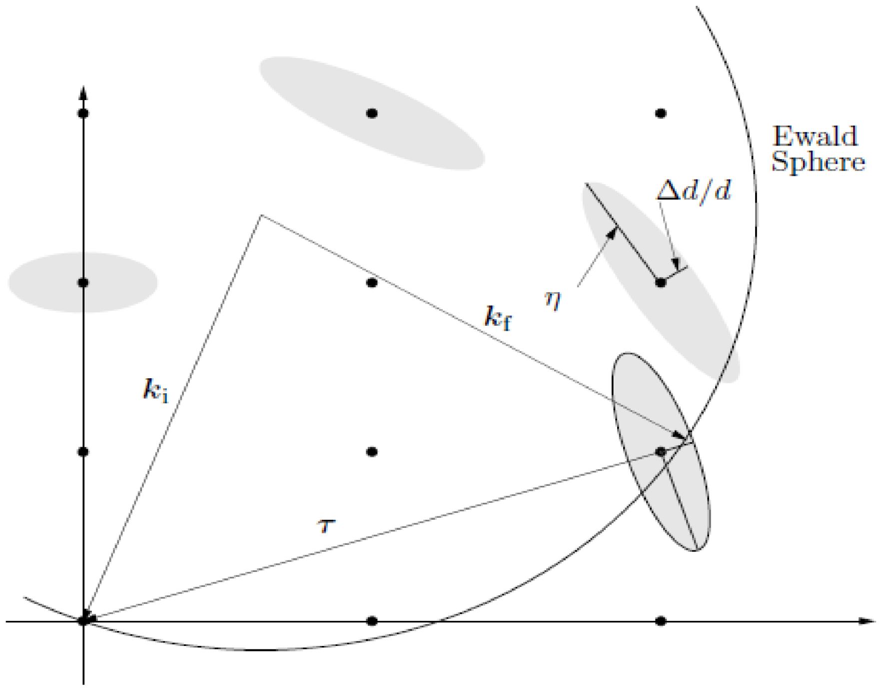
With focus



Properties

- Thick, flat single crystal
  - multiple scattering
  - absorption
- - incoherent scattering
- Mosaic, isotropic (anisotropic around sample lattice axes)
- Variance of lattice parameter $\Delta d/d$=0





Algorithm

The overview of the algorithm used in the Single_crystal component is as follows:

1. Check if the neutron intersects the crystal. If not, no action is taken.
2. Search through a list of reciprocal lattice points of interest, selecting those that are close enough to the Ewald sphere to have a non-vanishing scattering probability. From these, compute the total coherent cross-section $\sigma_{\rm coh}$ (see below), the absorption cross-section $\sigma_{\rm abs} = \sigma_{2200}\frac{2200 \text{ m/s}}{v}$, and the total cross-section $\sigma_{\rm tot} = \sigma_{\rm coh} + \sigma_{\rm inc} + \sigma_{\rm abs}$.
3. The transmission probability is $\exp(-\frac{\sigma_{\rm tot}}{V_0}\ell)$ where ℓ is the length of the flight path through the crystal. A Monte Carlo choice is performed to determine whether the neutron is transmitted. Optionally, the user may set a fixed Monte Carlo probability for the first scattering event, for example to boost the statistics for a weak reflection.
4. For non-transmission, the position at which the neutron will interact is selected from an exponential distribution. A Monte Carlo choice is made of whether to scatter coherently or incoherently. Absorption is treated by weight adjustment (see below).
5. For incoherent scattering, the outgoing wave vector $\boldsymbol{k}_{\rm f}$ is selected with a random direction.
6. For coherent scattering, a reciprocal lattice vector is selected by a Monte Carlo choice, and $\boldsymbol{k}_{\rm f}$ is found (see below).
7. Adjust the neutron weight as dictated by the Monte Carlo choices made.
8. Repeat from (2) until the neutron is transmitted (to simulate multiple scattering).

For point 2, the distance *dist* between a reciprocal lattice point and the Ewald sphere is considered small enough to allow scattering if it is less than five times the maximum axis of the Gaussian, $dist \leq 5\max(\sigma_1, \sigma_2, \sigma_3)$.

nmi3 Risø DTU DTU ess european spallation source

2.1.3: Single_crystal

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
reflections	string	File name containing structure factors of reflections. Use empty ("") or NULL for incoherent scattering only	
xwidth	m	Width of crystal	
yheight	m	Height of crystal	
zthick	m	Thichness of crystal (no extinction simulated)	
delta_d_d	1	Lattice spacing variance, gaussian RMS	1e-4
mosaic	arc minutes	Crystal mosaic (isotropic), gaussian RMS	-1
mosaic_h	arc minutes	Horizontal (rotation around Y) mosaic (anisotropic), gaussian RMS	-1
mosaic_v	arc minutes	Vertical (rotation around Z) mosaic (anisotropic), gaussian RMS	-1
mosaic_n	arc minutes	Out-of-plane (Rotation around X) mosaic (anisotropic), gaussian RMS	-1
recip_cell	1	Choice of direct/reciprocal (0/1) unit cell definition	0
ax	-		0
ay	AA or AA^-1	Coordinates of first (direct/recip) unit cell vector	0
az	-		0
bx	-		0
by	AA or AA^-1	Coordinates of second (direct/recip) unit cell vector	0
bz	-		0
cx	-		0
cy	AA or AA^-1	Coordinates of third (direct/recip) unit cell vector	0
cz	-		0
p_transmit	1	Monte Carlo probability for neutrons to be transmitted without any scattering. Used to improve statistics from weak reflections	-1
absorption	barns	Absorption cross-section per unit cell at 2200 m/s	0
incoherent	barns	Incoherent scattering cross-section per unit cell	0
aa	deg	.	0
bb	deg	unit cell angles alpha, beta and gamma. Then uses norms of vectors a,b and c as lattice parameters	0
cc	deg	.	0
order	1	limit multiple scattering up to given order (0: all, 1: first, 2: second, ...)	0
powder			0

2.1.3: Single_crystal

Basic setup

- A 2mm slab 0.1m*0.1m, small variance of lattice par, var. mos.:

```
xwidth = 0.002, yheight = 0.1, zthick = 0.1,
delta_d_d = 1e-4, mosaic = MOS
```

- Put the crystal with c axis along x, and b axis along z:

```
ax = 0, ay = 2.14, az = -1.24,
bx = 0, by = 0, bz = 2.47,
cx = 6.71, cy = 0, cz = 0,
```

($\alpha = 120$ deg)

- Set the right reflection list (h k l F^2 [barns]) for graphite

```
reflections = "Graphite_long.dat",
```

- σ_{abs}, σ_{inc} [barns] for graphite

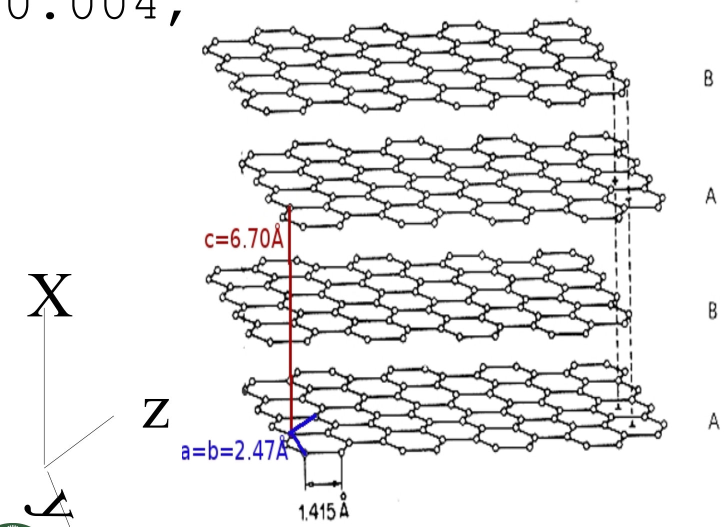
```
absorption = 0.014, incoherent = 0.004,
```

- Multiple scattering `order = 0` (all)

•Set monochromator rotation angle `a1` in scattering condition

• Set the monitors `a2` at the Bragg angle for the monochromator scattering

• Using wavelength 4.0-4.1Å you should get the same Bragg spot as before



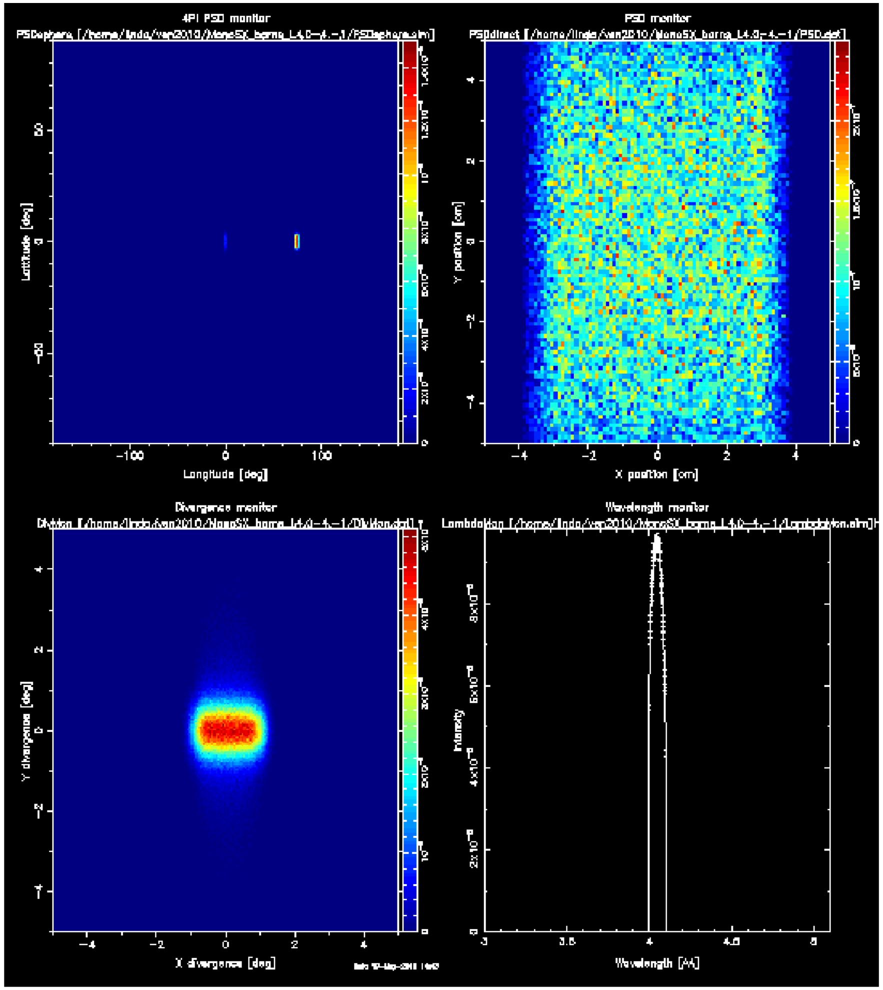
2.1.3: Single_crystal

Play!

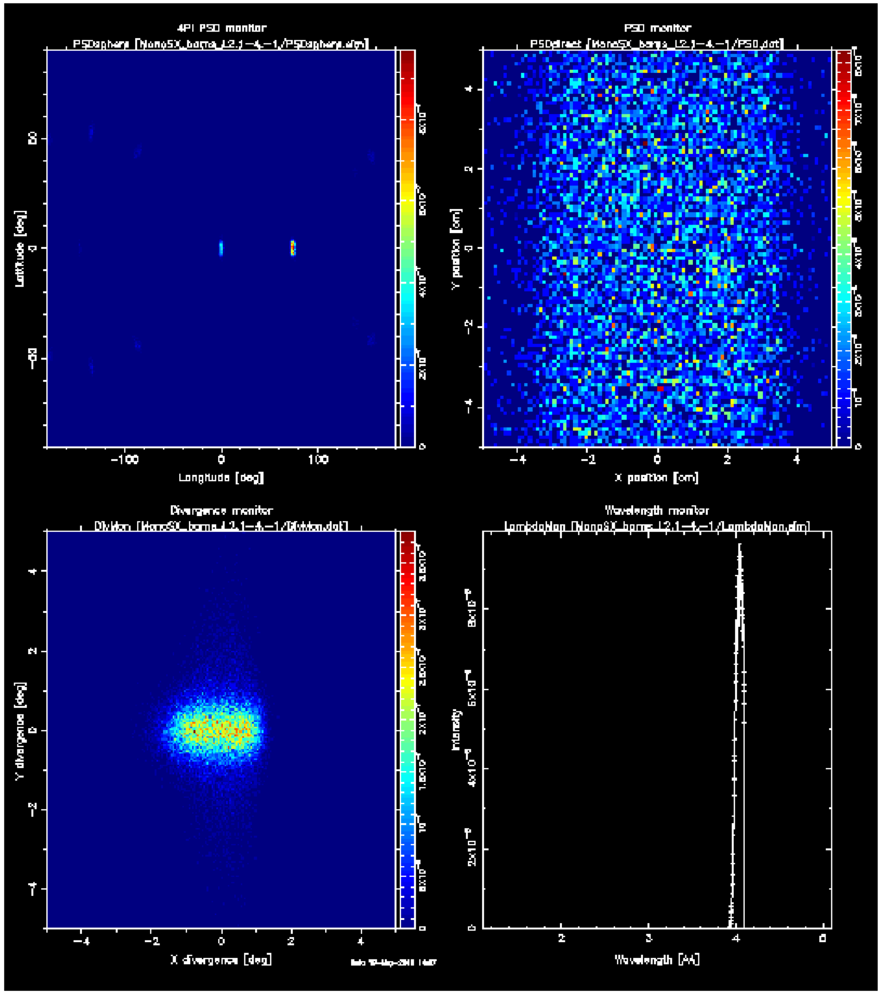
- Set a broader wavelength band from the source (2.1-4.1 Å)
- Observe the many reflections on the 4π PSD! (use log-scale)
  -this is why we need monochromator shielding :)
- You can also increase the incoherent scattering or absorption crosssection to observe the effect
- Or with the mosaicity or variance of latticespacing

2.1.3: Single_crystal

λ=4.0-4.1Å



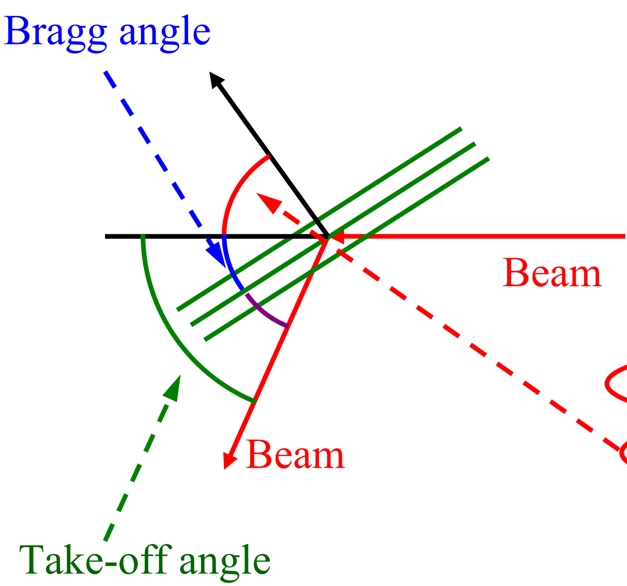
λ=2.1-4.1Å

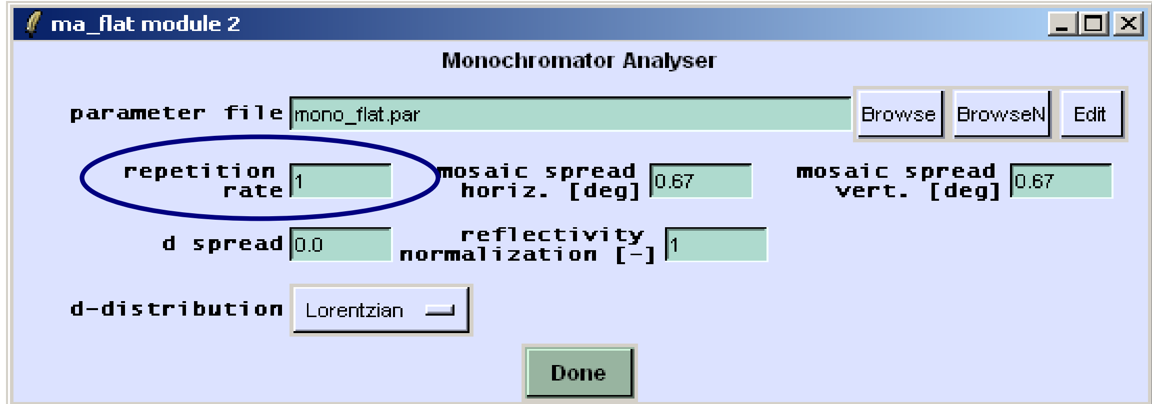


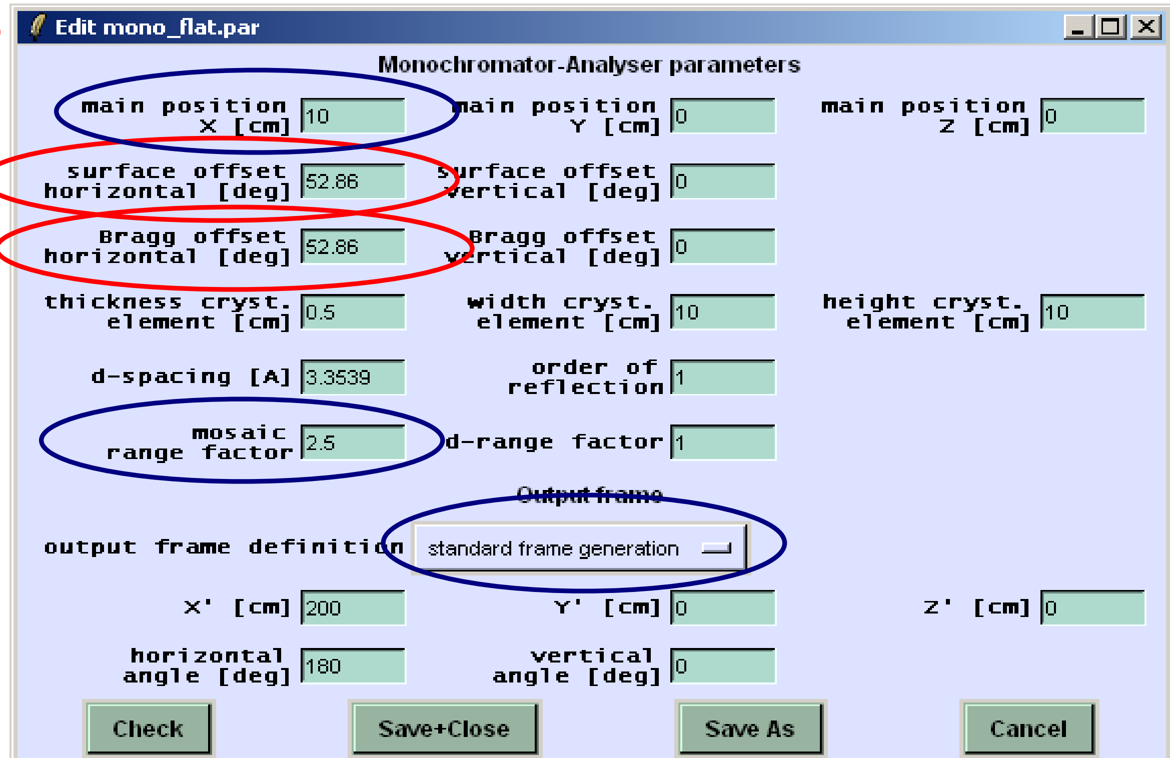
Task 2.1.1: Flat Monochromator (ma_flat)

1. Use the 'user wavelength distribution file' from Task 1.1 (linear Guide) to build a source of 10 x 10 cm^2
2. Send a beam in the range 1.0 – 7.0 Å to a flat monochromator (ma_flat) of 10 x 10 cm^2 size and 0.5 cm thickness in a distance of 10 m
3. Choose
   - a mosaicity of 40' (horizontal and vertical)
   - a d-spacing of 3.3539 Å
   - a take-off angle of 74.28º to get a wavelength of 4.05 Å
   - Δd/d = 0, mosaic range factor 2.5
   - reflectivity normalization and repetition rate 1
   - standard frame generation
4. Add mon1_lambda, mon2_pos (20x20 cm2) and mon2_div (-5° to 5° in both directions) in 0.828 m distance
5. Check resulting wavelength, adapt generated wavelength range

Task 2.1.1: Parameters of a Monochromator Element





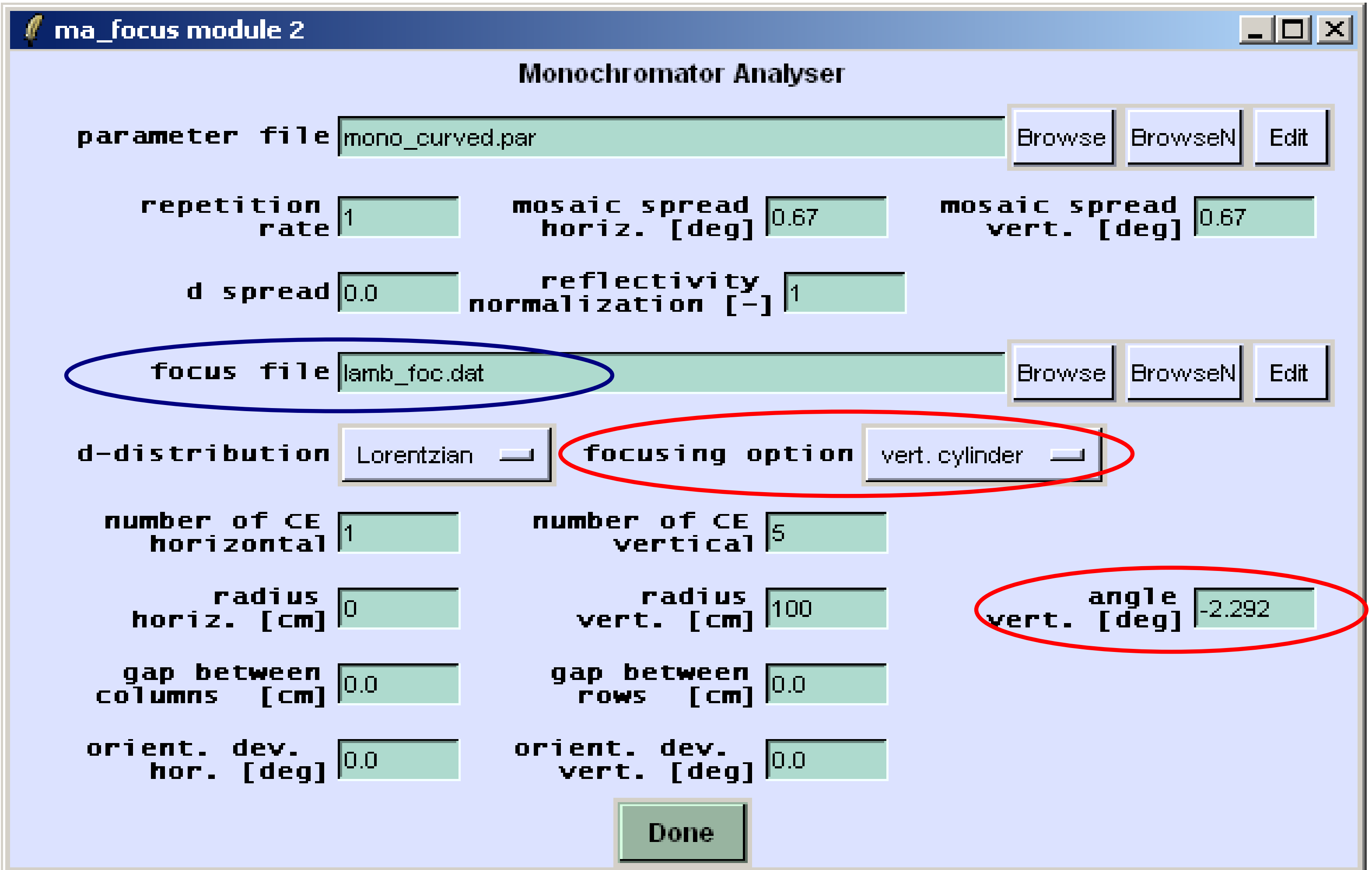


nmi3 Risø DTU DTU ess european spallation source

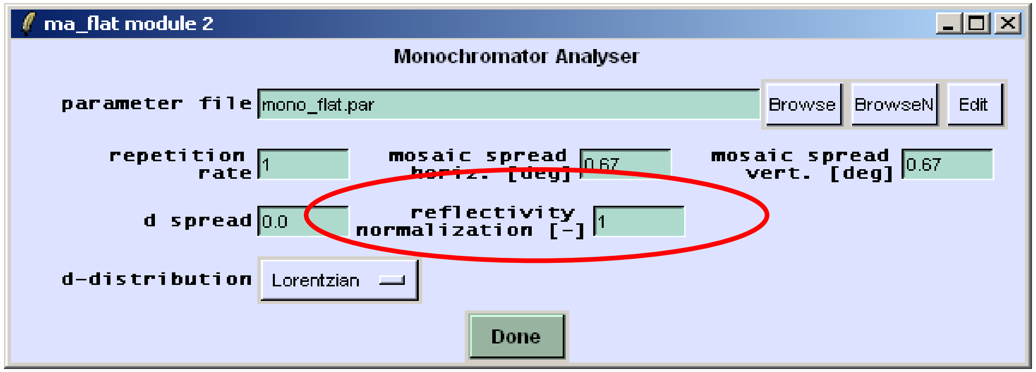
Task 2.1.2: Focusing Monochromator (ma_focus)

1. Exchange the flat by a focusing monochromator that
   - consists of 5 pieces of 2 cm height, 10 cm width and 0.5 cm thickness
   - has a shape of a vertical cylinder, radius 1 m
   - has a minimal vertical angle of -2.292°
   - has the same mosaicity, d-spacing and take-off angle as the flat one
2. Compare wavelength and divergence distribution as well as spot size

Task 2.1.2: Parameters of a Monochromator Ensemble



2.1.3: Normalization



Basic idea: determine ratio of outgoing to incoming intensity for

- beam of divergence 0
- sufficiently large monochromator element

Necessary for each take-off angle

E. Intensity normalization

The sophisticated computation procedures of this module (e.g. considering a 2-dim. distribution function for the mosaicity) leads to a very good description of the factors which influence the resolution behavior of the whole instrument.

For reliable intensity comparisons (to other types of instruments) it might be necessary to renormalize the calculation. The following procedure is recommended:

1. Use the module *source_CWS* and adjust the ideal wavelength which is defined by fulfilling the Bragg condition with a mosaic of the crystal element under consideration which corresponds to the ideal direction $\mathbf{n}_{CE}$. Generate neutrons with nearly no or a very small divergence in each direction.
2. Simulate the reflection of this neutron beam at the crystal element described by this module (using the data known about the material, mosaicity a.s.o. and adjusting appropriate mosaic- and d-ranges).
3. Divide the output count-rate of this module by the input count-rate to obtain the rate of reflection R_sim.
4. Compare R_sim with the experimental peak reflectivity R_exp, known from mosaicity measurements (referring to the same ideal wavelength as used above) for the crystal under consideration, to obtain the rescaling parameter P= (R_exp/R_sim).
5. Now the intended VITESS simulation (e.g. an instrument which uses crystal elements of this type) can be performed. For this do not change the values and ranges (which have been used under 2.) for the mosaicity, d-spacing and neutron repetition rate, but renormalize the chosen reflectivity *R* by multiplication with P. Although the new value for *R* may now exceed 100% and differ from R_exp, the module will simulate correctly both, the intensity and resolution behavior of the crystal element system.

Background estimate: what is background ?

Background is … everything you do not want to see.

Origin of background:

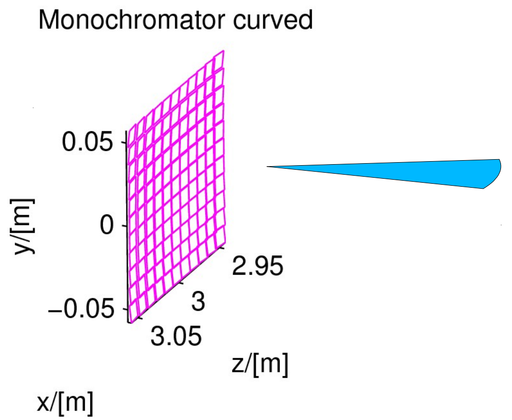
- Fast neutrons, gamma rays …
- Scattering from any unwanted part in the beam
- Mechanics, dust, hydrogenated molecules, …
- Sample environment

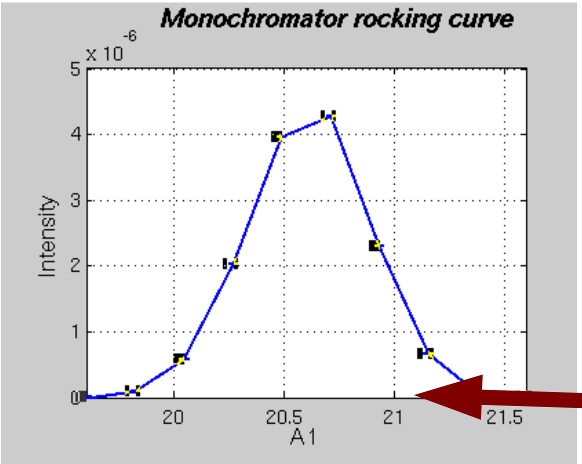
Appears usually as a low level signal, below measurement.
It has no reason to be constant...

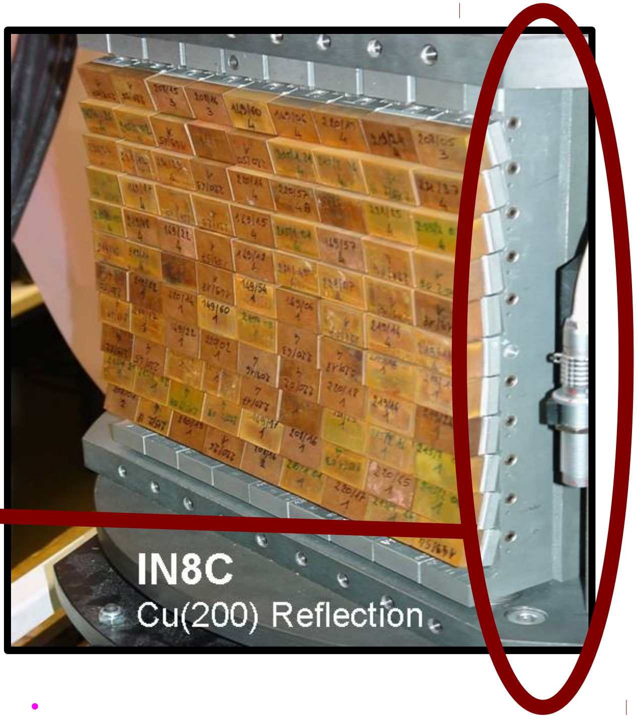
Background estimate: mechanics contribution

Monochromators are used to extract a
sharp neutron energy distribution from a white beam.
Rely on Bragg's law.

Use single crystal assembly, with focusing geometry.
Size: Typically 20x20 cm



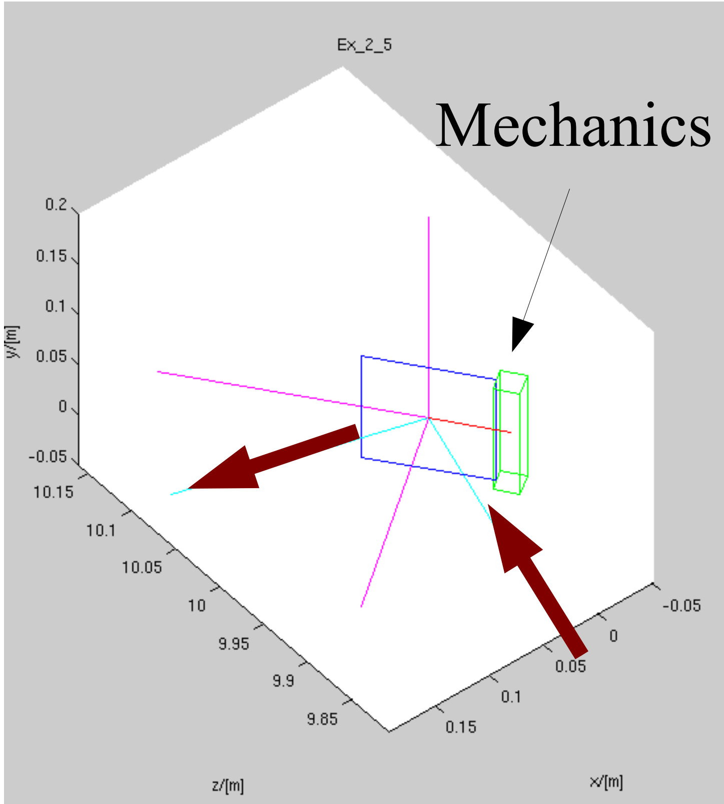




Optics mechanics DO scatter, and may be massive

Background estimate: monochromator simulation

We shall insert a piece of metal next to the monochromator, and a dedicated monitor to record only neutrons scattered from this piece.



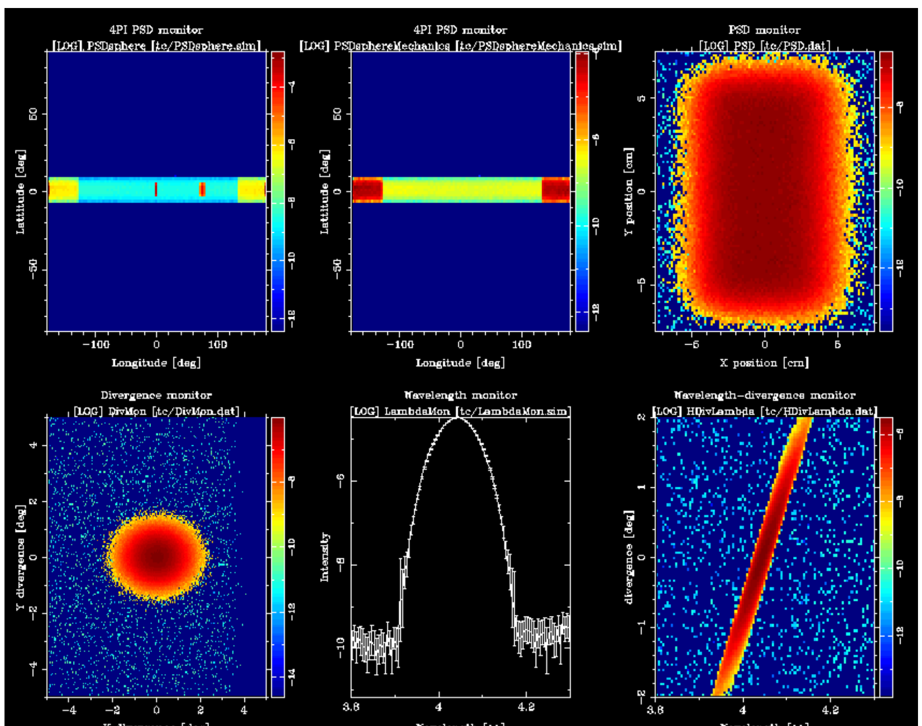
1) Get the *Ex_2_1_4* example
2) Define a new instrument input 'string' parameter 'mount' that will specify the material, set as « Al.laz » as default
3)Define a '*flag_mechanics*' variable in the *DECLARE* block
4) Add a *PowderN* instance at 6 cm from the Monochromator, as a 2x2x10 cm bar

5) *Make it so* that it sets the '*flag_mechanics*' to 1 when neutron has scattered
6) Add a sphere detector that records only *flag_mechanics* neutrons

Background estimate: exercise

We shall now use that instrument

1) Run the simulation with 1e7 neutrons and Aluminium mount in directory '*Al*'
2) Repeat with mount=Cu.laz (copper) in directory '*Cu*'
3) Compare the parasitic Bragg peaks and the background level. *Which is best ?* You may press the '***L***' key to toggle log-scale
4) Wavelength is around λ=4 Angs. What will happen for faster neutrons ?



Ex. 2.2: Rotating, moving parts

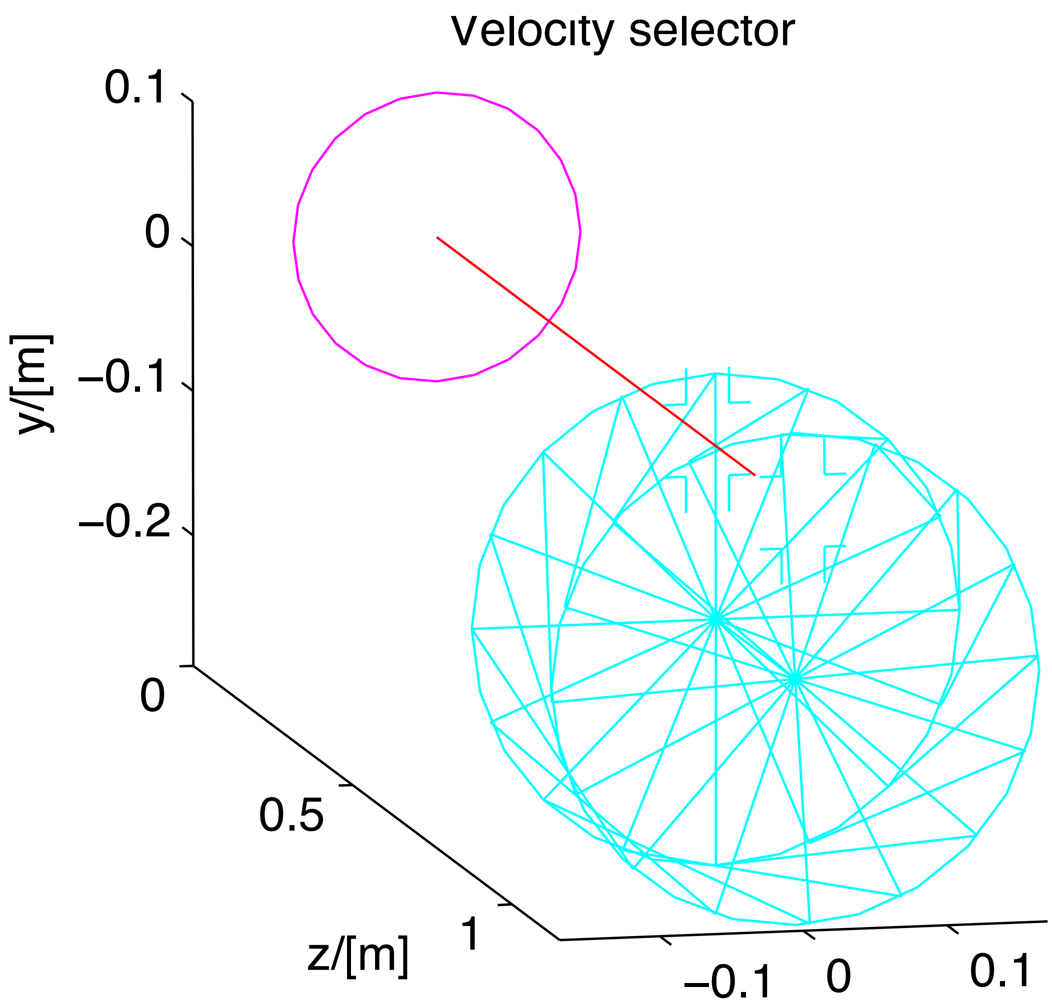
2.2.1 Velocity selector

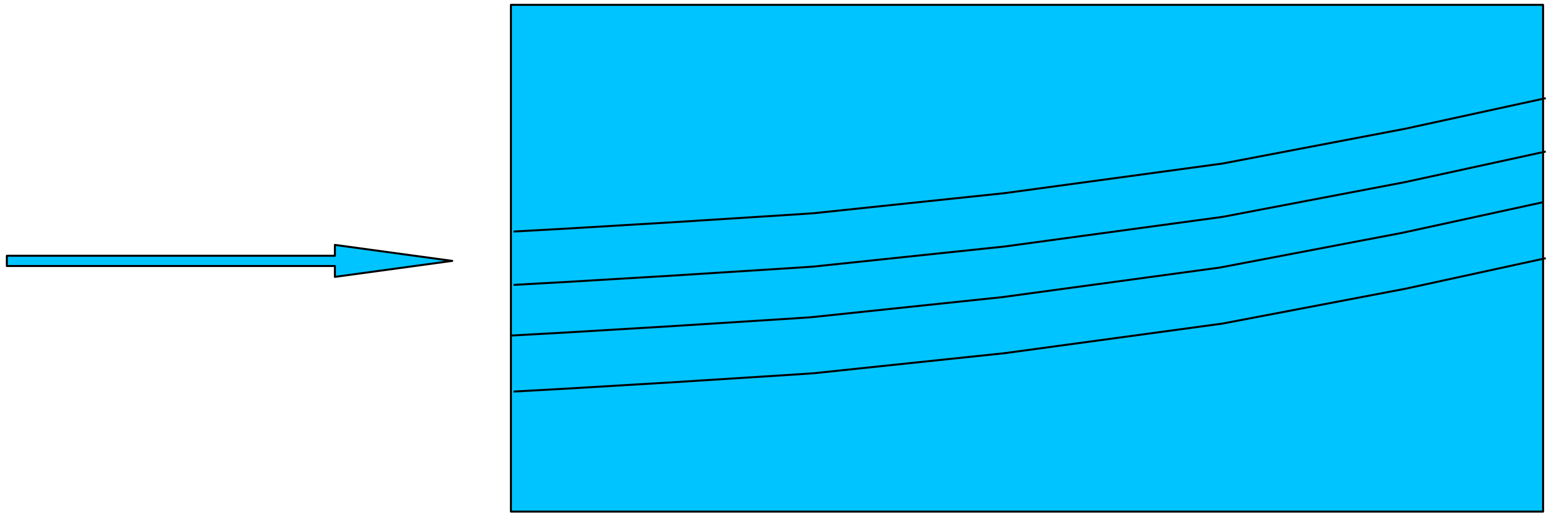
2.2.2 Disk Chopper

2.2.3 Fermi Chopper

2.2.1: Velocity selector:

As you saw, monochromators define a very monochromatic beam. A greater bandwidth monochromatization device is a velocity selector





Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
width	m	Width of entry aperture	0.03
height	m	Height of entry aperture	0.05
l0	m	Distance between apertures	0.30
r0	m	Height from aperture centre to rotation axis	0.12
phi	deg	Twist angle along the cylinder	48.298
l1	m	Length of cylinder (less than l0)	0.25
tb	m	Thickness of blades	0.0004
rot	rpm	Cylinder rotation speed, counter-clockwise	20000
nb	1	Number of Soller blades	72

Exercise 2.2.1

Open the Ex_2_2_1.instr instrument

Notice use of wavelength monitors L_mon

Notice use of the V_select component

Input parameter ROT defines selector rotational velocity (RPM)

Perform a TRACE at the default ROT=20000 RPM

Perform a SIMULATE of 1e7 neutrons at default ROT

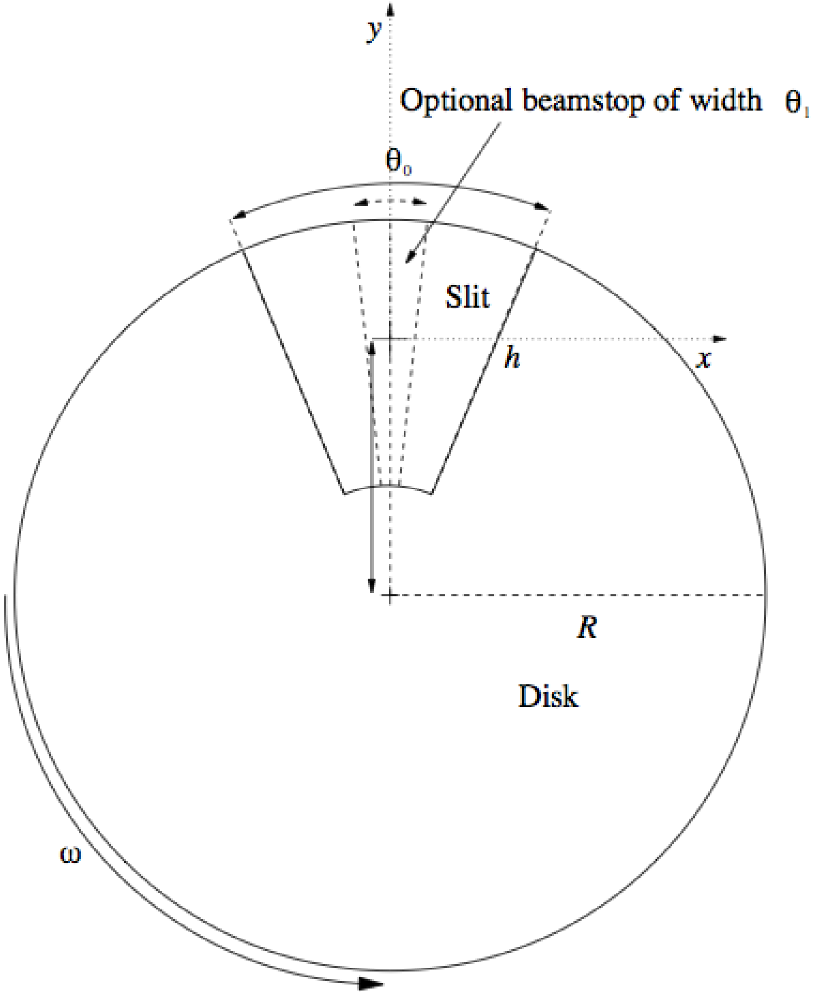
Estimate the relative bandwidth $\delta\lambda/\lambda$ of the transmitted beam

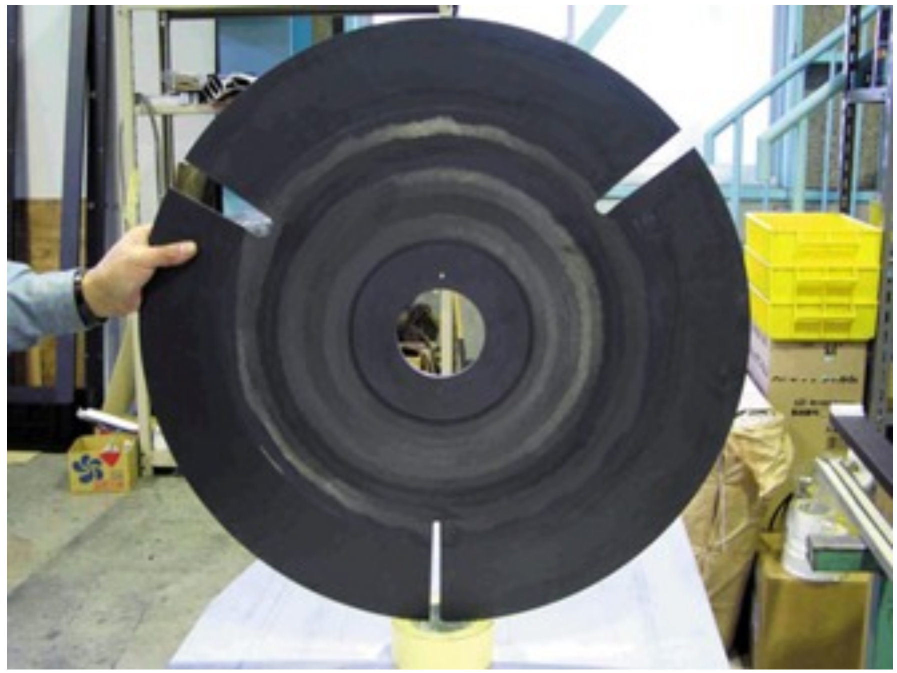
Perform a series of simulations in the range
10000 < ROT < 50000 (5 steps)

Compare the transmitted beam in the different cases

2.2.2: Disk chopper:

A Disk Chopper is also a rotating device, selecting neutrons. The travelled distance in the device is much smaller (disk), for defining time structure in the neutron beam.



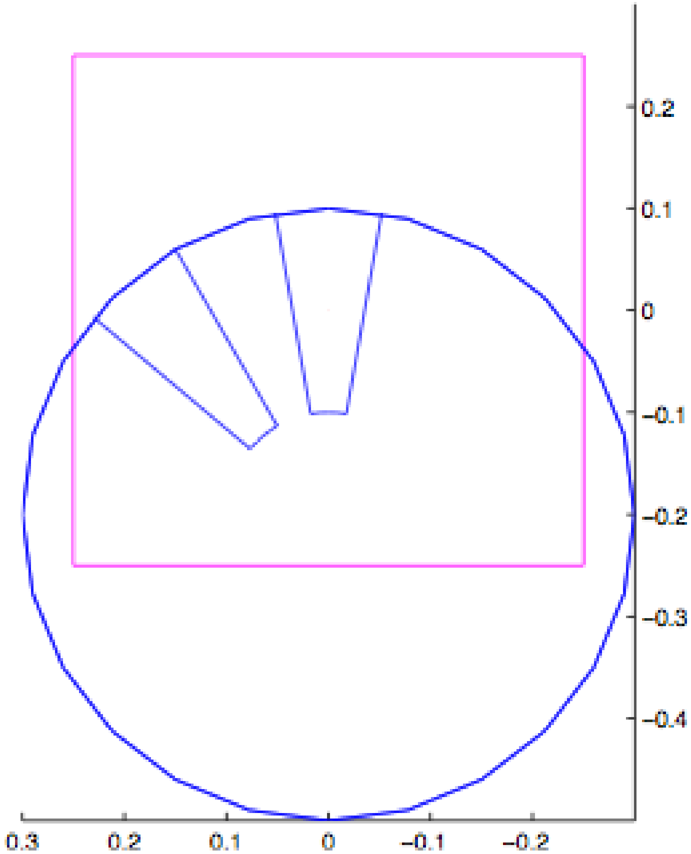


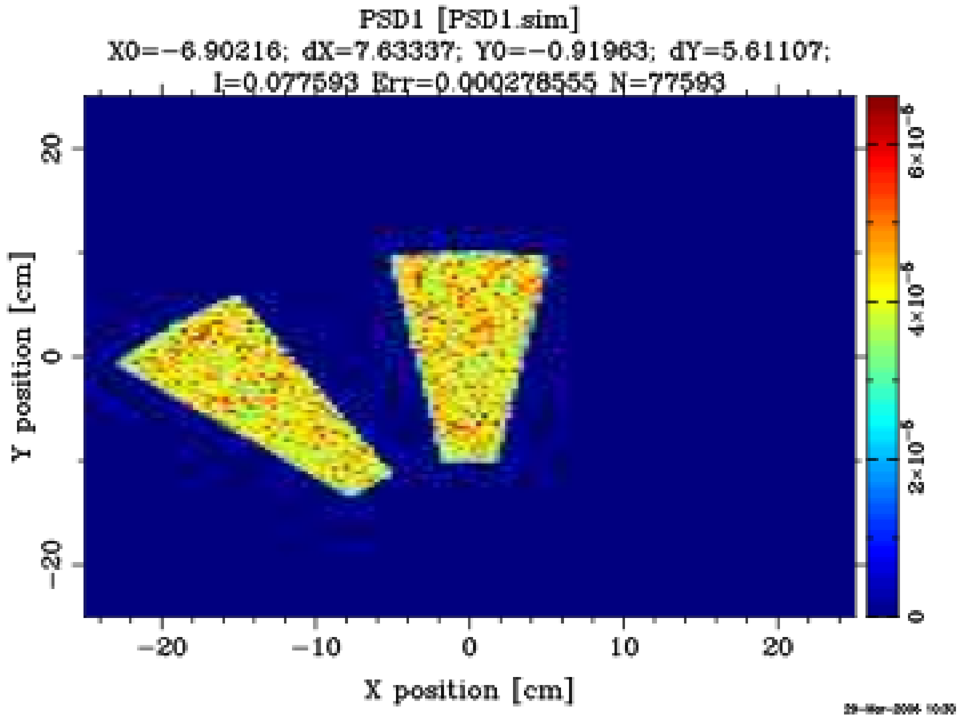
Ven 2010
essworkshop.org

Ex. 1 / Peter Willendrup

2.2.2: Disk chopper:

A Disk Chopper is also a rotating device, selecting neutrons. The travelled distance in the device is much smaller (disk), for defining time structure in the neutron beam.





2.2.2: Disk chopper:

Parameter significance

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
theta_0	deg	Angular width of the slits.	
R	m	Radius of the disc	
h	m	Slit height (if = 0, equal to R). Auto centering of beam at h/2.	0
omega	rad/s	Angular frequency of the Chopper (algebraic sign defines the direction of rotation)	
n	1	Number of slits	3
j	s	Jitter in the phase	0
theta_1	deg	Angular width of optional beamstop in chopper windows	0
t_0	s	Time 'delay'.	0
IsFirst	0/1	Set it to 1 for the first chopper position in a cw source (it then spreads the neutron time distribution)	0
n_pulse	1	Number of pulses (Only if IsFirst)	1
abs_out	0/1	Absorb neutrons hitting outside of chopper radius?	1
phi_0	deg	Angular 'delay' (suppresses t_0)	0
w	m	'width' of slits for compatibility with Chopper.comp	0
wc	m	'width' of beamstops for compatibility with Chopper.comp	0
compat	1	Chopper placement compatible with original Chopper.comp	0

2.2.2: Disk chopper:

Used parameters

- R, radius of disk-chopper (we use 0.5 m)
- n, number of openings (we use 2)
- phi_0 (angular phase at t=0, in degrees, we use 90 deg)
- omega (angular frequency of chopper)
- theta_0 (angular width of each chopper opening)

Exercise 2.2.2

Open the Ex_2_2_2.instr instrument

Notice use of the EXTEND %{ %} section, defining a time structure (1 second, flat distribution)

Notice use of Monitor_nD, our “Swiss army knife” monitor

options=”t auto bins=200”

options="t auto bins=200 x auto bins=200"

- Automatic binning if wished
- Monitors any state (or user) variable vs. any other
- Assumes various shapes/geometries
- ...

Instrument input parameters:

f (Hz) - chopper frequency $\omega=2\pi*f$ in component parm list)

Theta0 (degrees) - opening width of slit(s)

Exercise 2.2.2

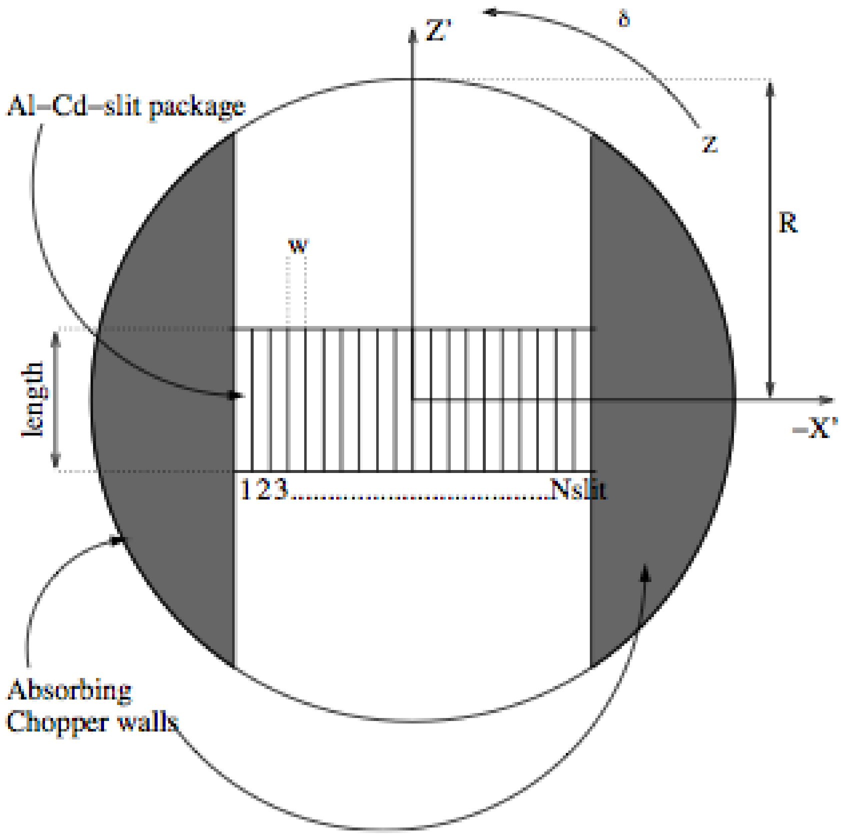
Make a TRACE to get an overview of the instrument

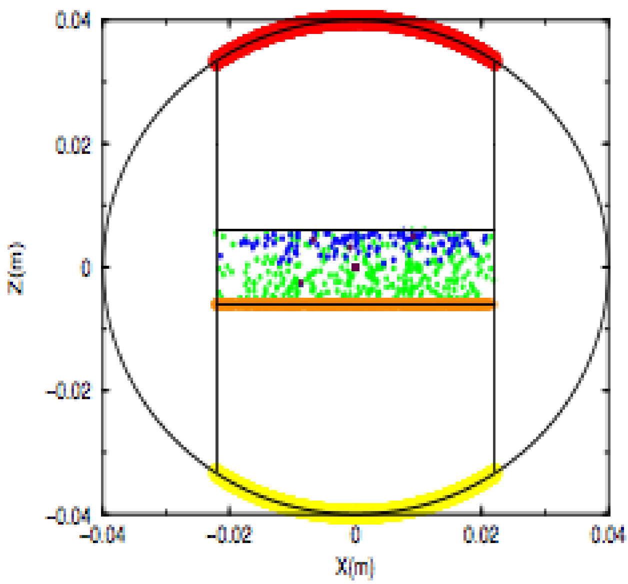
SIMULATE 1e7 neutrons at the default of f=5Hz and Theta0=10 degrees. While simulation is ongoing, estimate the number of pulses per second?

Try another 1e7 at f=1 hz. Notice space-time correlation in the third TOF panel

At a given frequency, try changing the Theta0 chopper opening to higher and lower value. Comment on the results.

2.2.3 Fermi chopper - summary



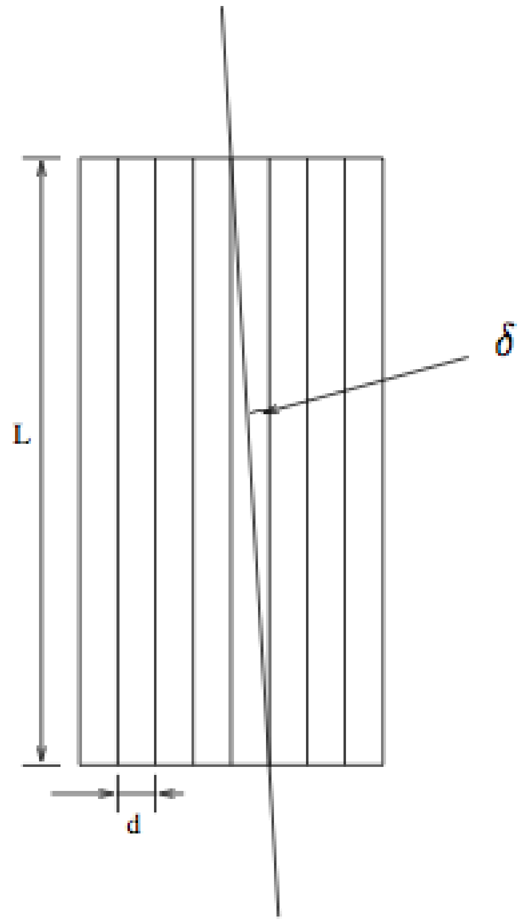


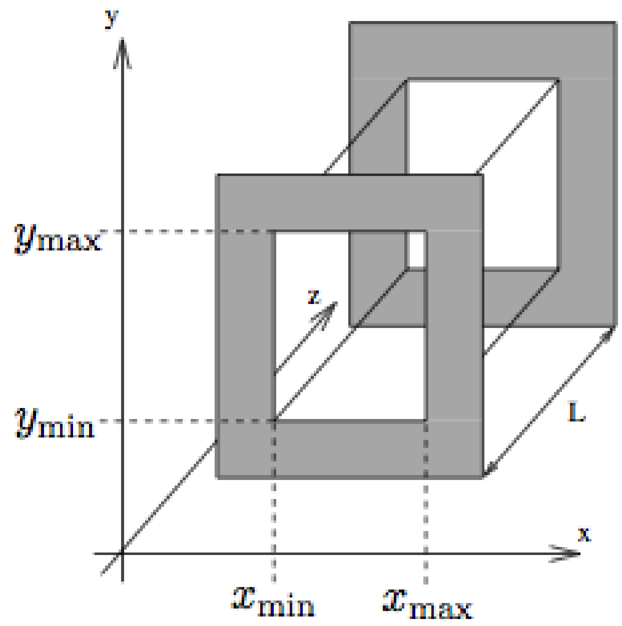
2.3 Slits - short summary

Name:	Slit
Author:	System
Input parameters	x_{min}, x_{max}, y_{min}, y_{max}
Optional parameters	r, p_{cut}
Notes	

2.4 Collimators - linear collimator - short summary

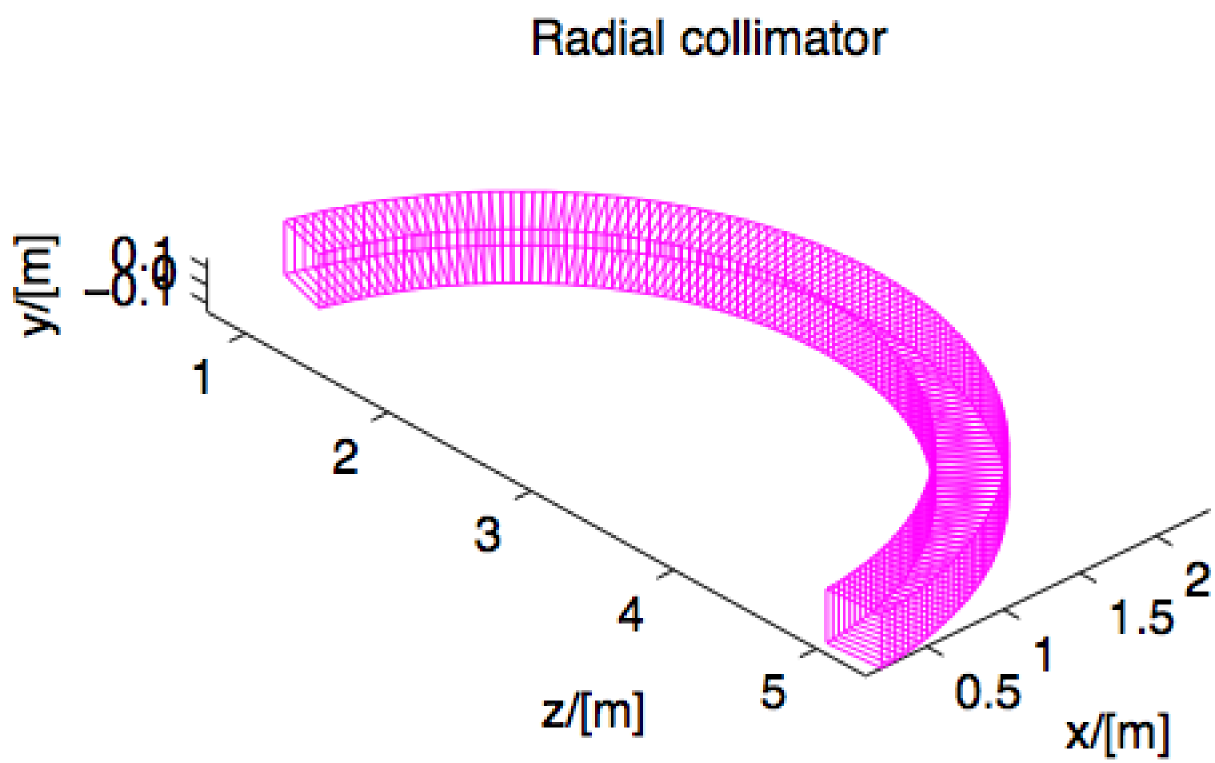
Name:	Collimator_linear
Author:	System
Input parameters	x_{min}, x_{max}, y_{min}, y_{max}, L, δ
Optional parameters	
Notes	





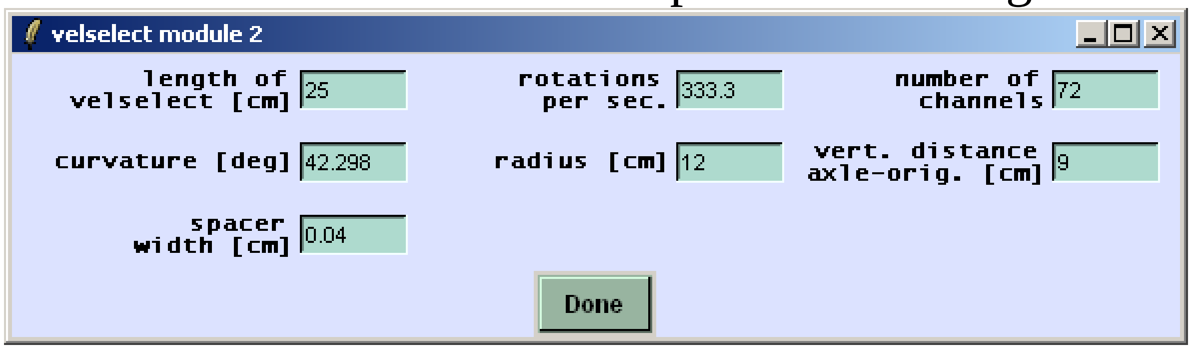
2.4 Collimators - radial collimator - short summary

Name:	Collimator_radial
Author:	(System) E.Farhi, ILL
Input parameters	w_1, h_1, w_2, h_2, len, θ_{min}, θ_{max}, $nchan$, $radius$
Optional parameters	$divergence$, $nblades$, roc and others
Notes	Validated



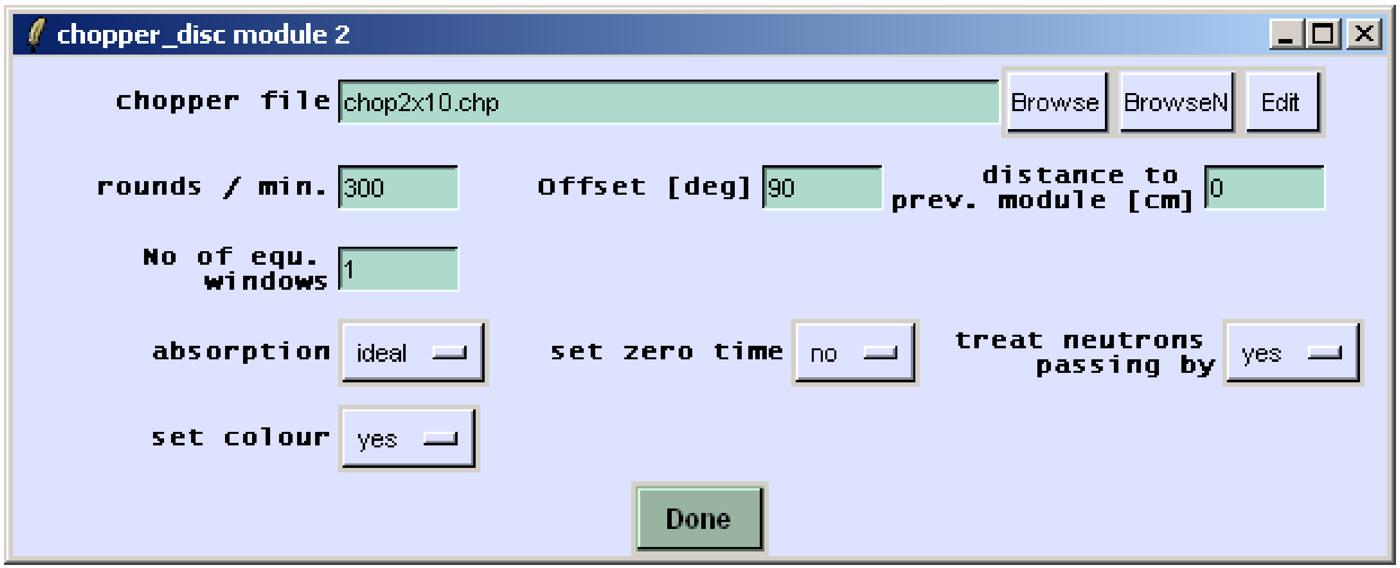
Task 2.2.1: Velocity Selector

1. Exchange the monochromator by a velocity selector and set selector parameters to
   - 72 channels of 25 cm length and 42.298° curvature
   - 333.3 rotation per second
   - radius of selector 12 cm, blade width 0.04 cm
   - chose proper distance beam and axle
2. Change source
   - Sent neutrons between 1 and 12 Å to a spot of 3 x 5 cm^2 (WxH) in a distance of 3 m
3. Run instrument and compare wavelength distribution with that of a monochromator
4. Run a series of 5 different rotational speeds in the range 10000 – 50000 rpm

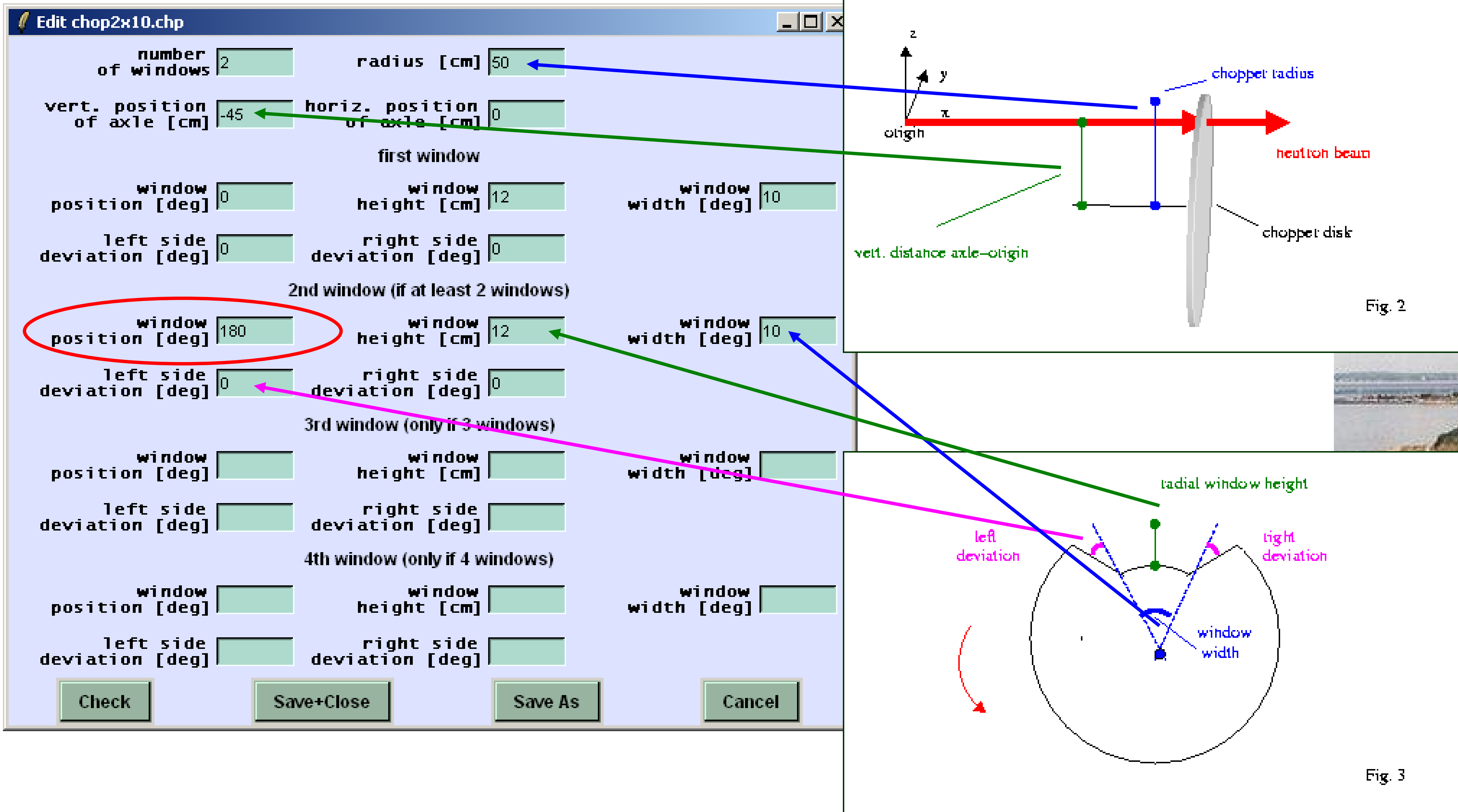


Task 2.2.2: Disc Chopper

1. Exchange velocity selector by a disc chopper and set the following parameters
   - radius 50 cm
   - 2 openings of 10° (at positions 0° and 180°)
   - initial phase 90°
   - frequency 300 rpm



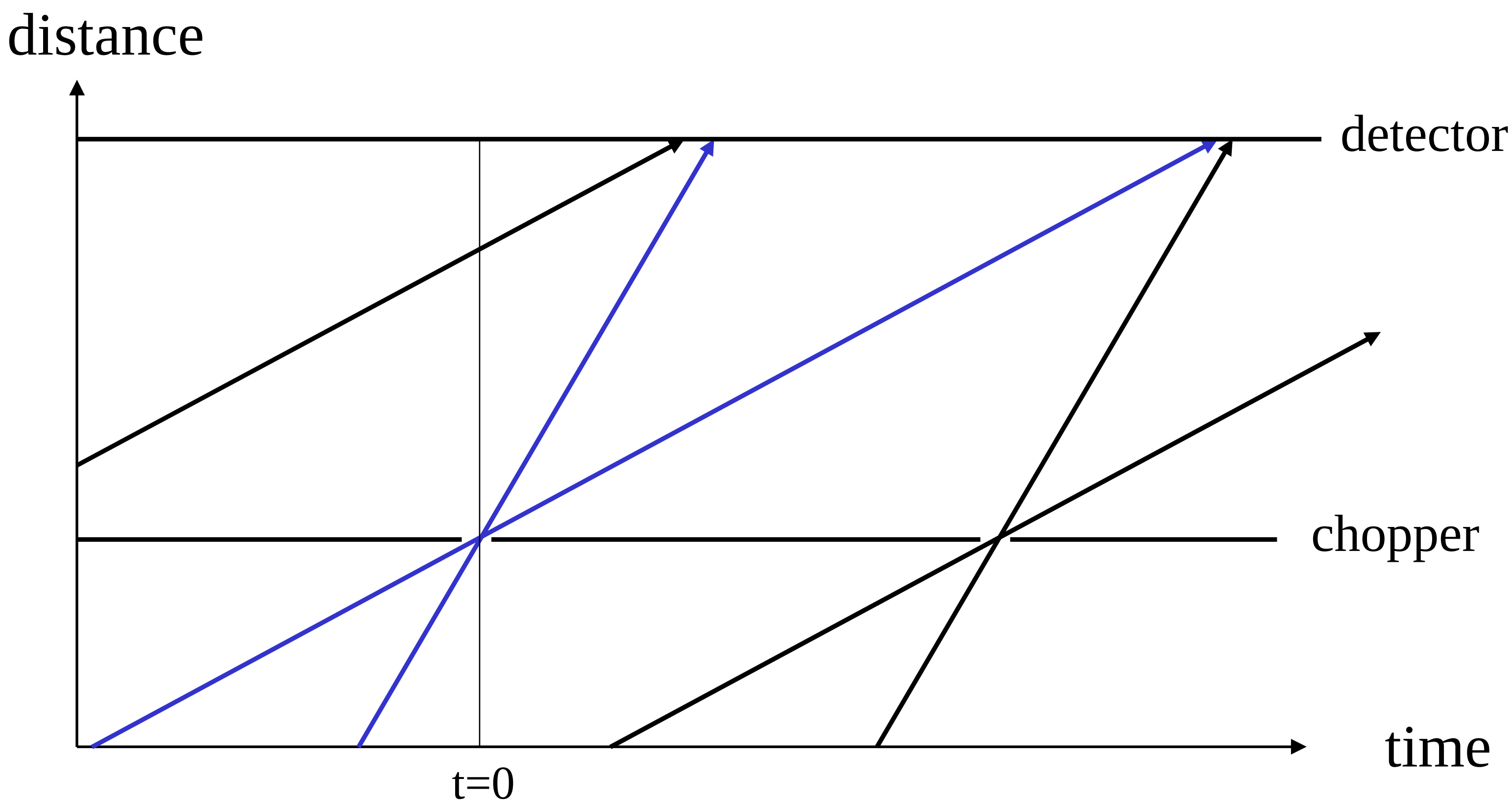
Task 2.2.2: Disc Chopper – File



Task 2.2.2: Disc Chopper

2. Change source

- Add proper time interval for the neutrons to start (in the source module) to select neutrons between 2 Å and 4 Å



Slits and Collimators

1. Apertures
   - slit (rectangular, ideal)
   - spacewindow (circular + rectangular, material in window and outside)
   - spacewindow_multiple (several windows)

2. Collimators
   - collimator_soller (simple, analytic)
   - collimator (better)
   - collimator_radial

3- Sample environments

3.1 concentric geometries
3.2 background estimates

E. Farhi, ILL

Sample environment: definition

A sample environment is all that surrounds sample.

Any material in the beam acts as a sample: it may absorb and scatter.

Usual environments are concentric:

- Furnaces
- Cryostats

Some may be non-concentric/symmetric

- Magnets
- Pressure cells

Some materials used in sample environments:

Al, Cu, Nb, ...

Exercise 3.1: source+sample+detector

Goal: build a simulation of a scattering sample.

1)Start McGUI, and click on Edit
2)Select menu in Editor: Insert/Instrument Template
3)Change instrument name as Ex_3_1 and save
4)Add input parameters (lambda=2, string sample=”SiO2_quartza.lau”)
5)In the TRACE after *Origin*, insert a Source_simple(radius=0.005, dist=4,xw=0.02,yh=0.02,Lambda0=lambda,dLambda=0.1)
6)At 4 m, add a sample PowderN(reflections=sample, radius=0.005,yheight=0.02,d_phi=50)
7)Around the sample, add a banana detector Monitor_nD(xwidth=2,yheight=1, options="banana theta y, auto", bins=180)
8)Run simulation. Observe onion rings (scattering from a powder gives rings which angle give the atomic spacing). Press '***L***' key for log-scale

Simulating a concentric arrangement

With McStas, any concentric geometry should be described symmetrically *w.r.t.* the sample position, e.g. :

```
COMPONENT entry_side= Comp(blah, concentric=1)

COMPONENT sample= …

COMPONENT exit_side= COPY(entry_side)(concentric=0)
```

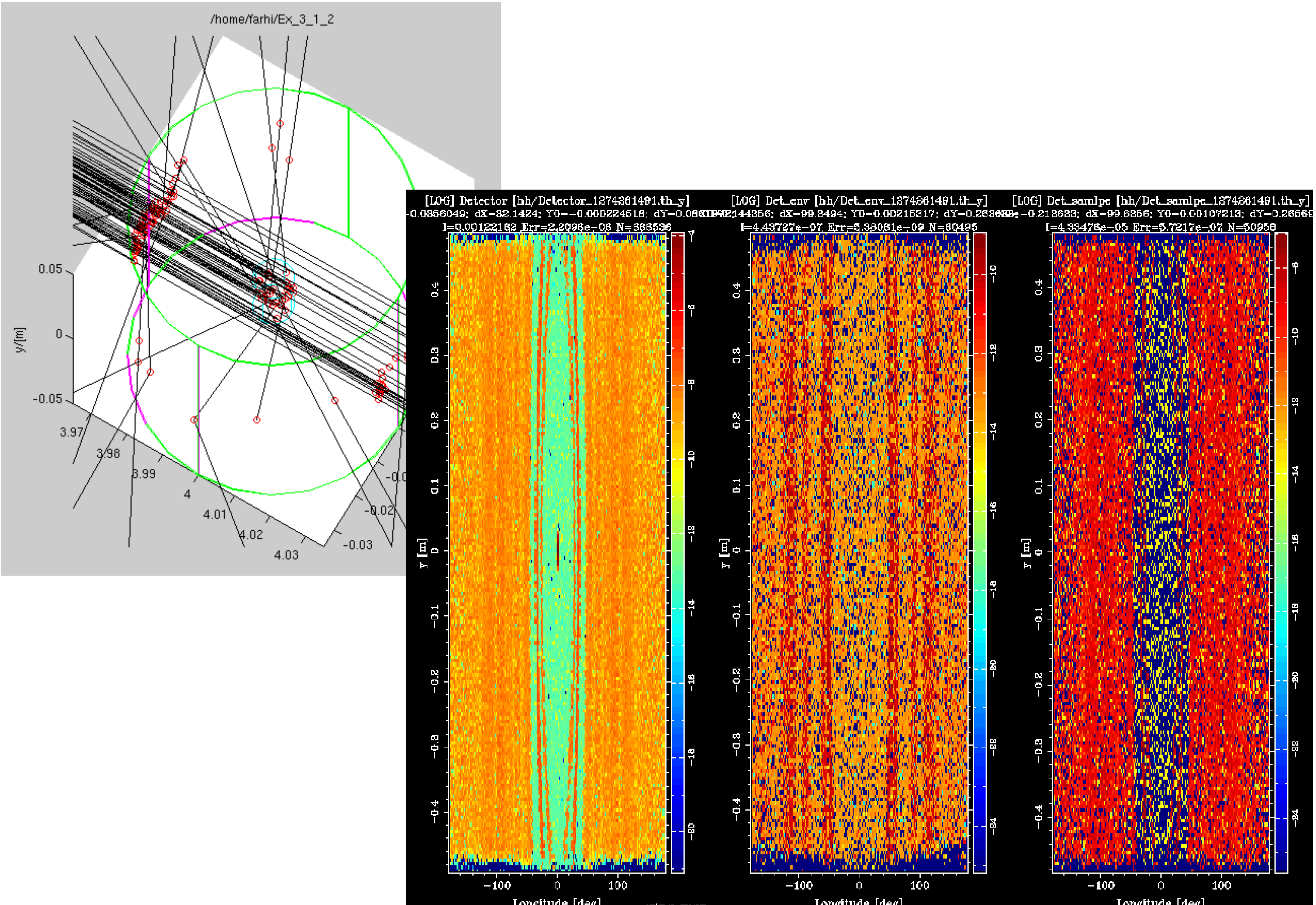
This works for the powder and liquid/amorphous/glassy materials.

Simulating a concentric arrangement

Goal: surround the previous sample with a cylinder of Aluminium

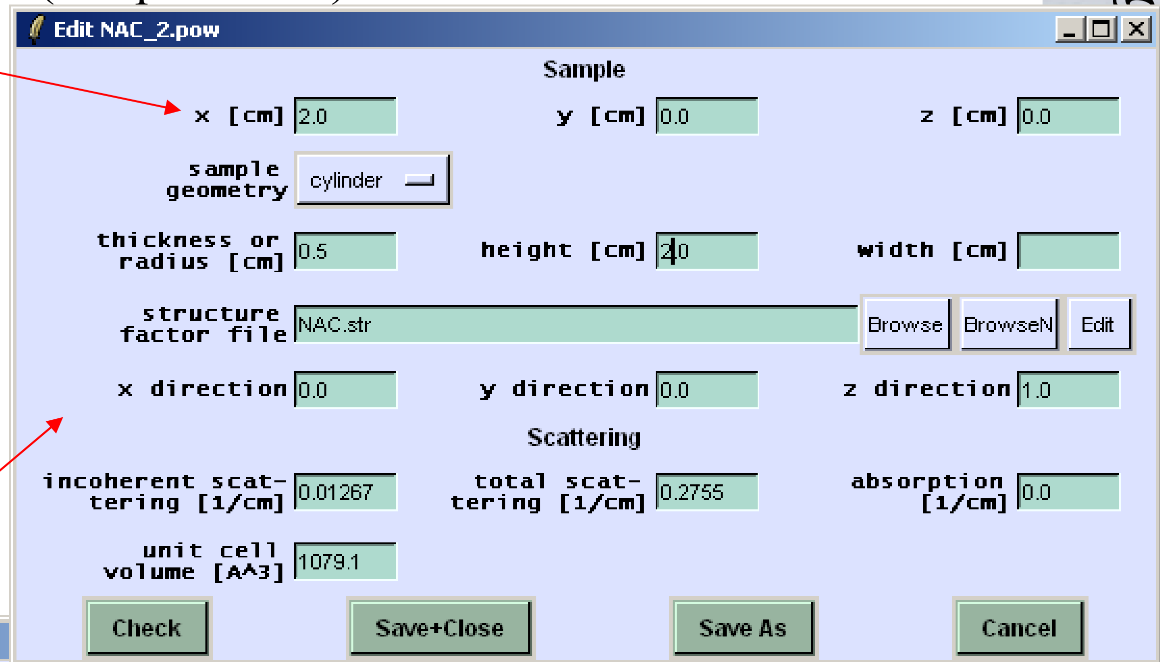
1)Before the Sample, add a cylinder
entry_side=PowderN(reflections="Al.laz", radius=0.035, radius_i=0.035-0.0002,d_phi=50,tfrac=0.8,concentric=1) centred on the sample
2)After the sample, add a exit_side=COPY(entry_side)(concentric=0)
3)Re-run simulation. *Are there additional rings ?*
4)In the DECLARE %{ … %} add int flag_env,flag_sample;
5)EXTEND the Origin with flag_env=flag_sample=0;
6)EXTEND PowderN components with e.g. if (SCATTERED) flag_*blah*=1;
7)Duplicate the Detector with copies that only activates WHEN (flag_*blah*)
8)Re-run. Compare the intensity from the sample and the environment.

What you should get

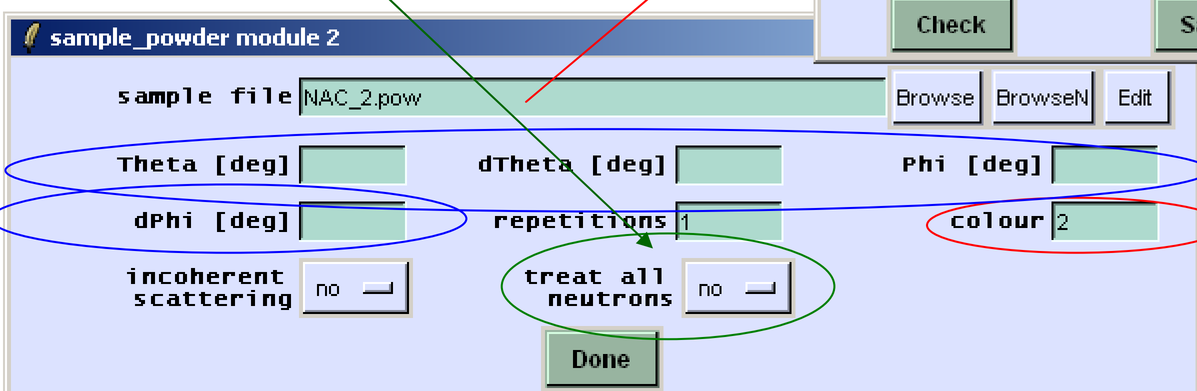


Task 3.1: Powder Sample

1. Use a ‘user wavelength distribution file’ from a previous task to create a source of 1 cm diameter and bring neutrons of 1.99 – 2.01 Å to a spot of 2 x 2 cm^2 in a distance of 4 m
2. Add a cylindrical sample_powder (sample: NAC) of 1 cm diameter and 2 cm height in 2 cm distance.
3. Add ‘mon2_div’ to visualize Bragg rings



Remove neutrons that are not scattered



Restriction of the solid angle into which neutrons are scattered

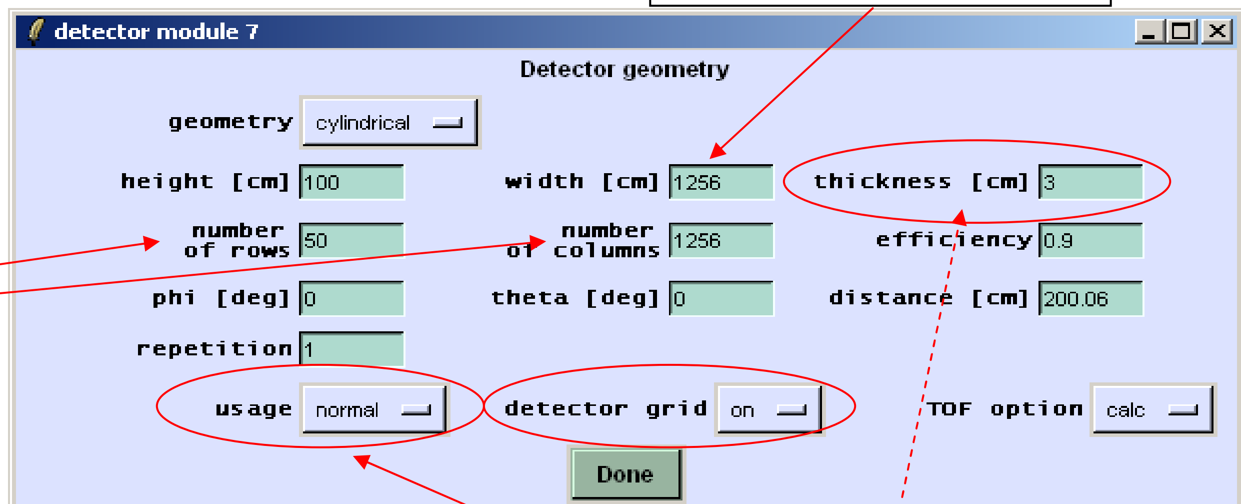
Mark scattered neutrons

Task 3.1: Detector

1. Add a cylindrical detector of 1 m height and 2 m radius all around the sample
2. Add ‘eval_elast’ to see intensity as a function of scattering angle and to determine d-spacings

total length of the detector

determines size of the detector cells



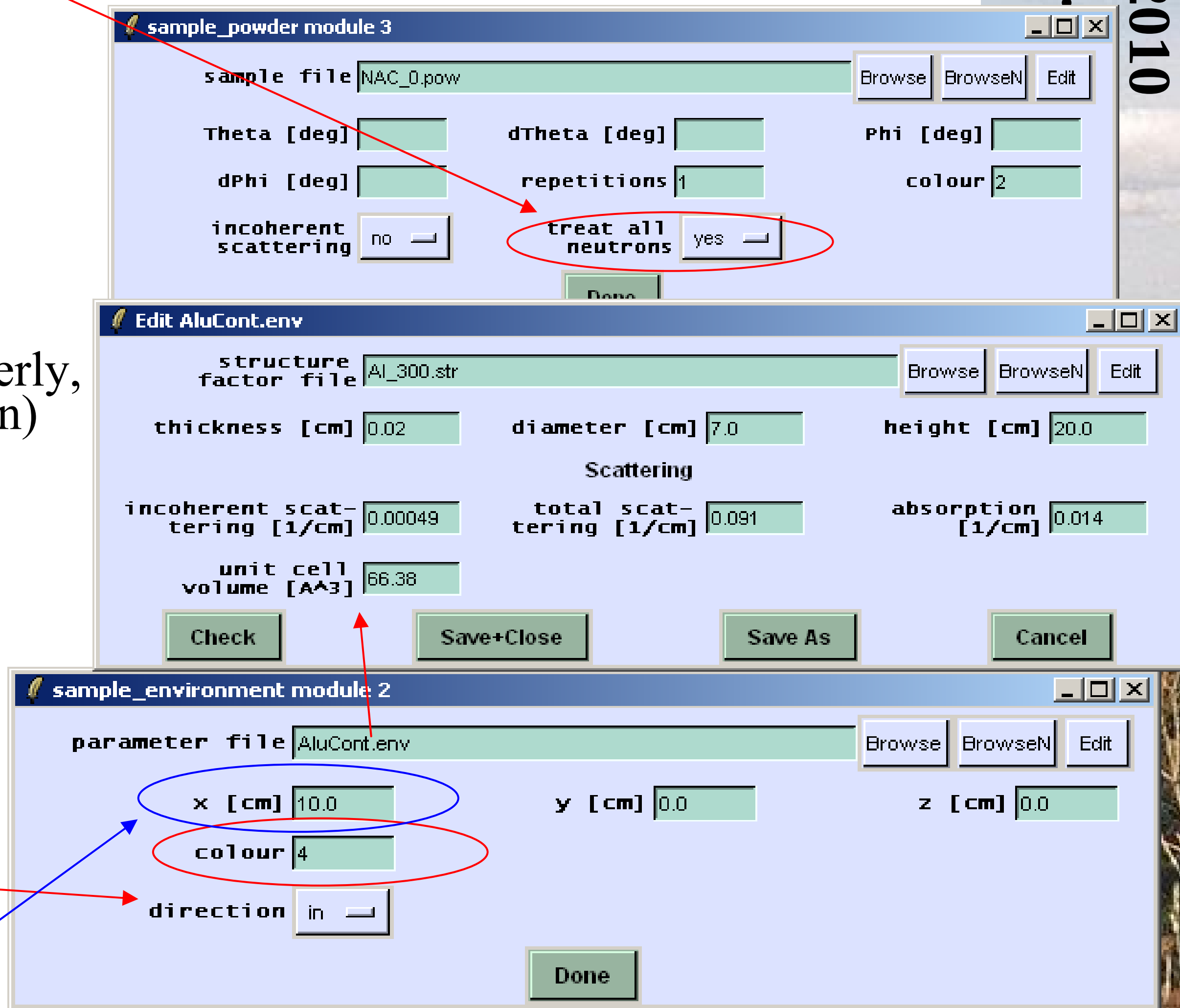
normal: wavelength dependent detection probability + diffusion into neighboring cells

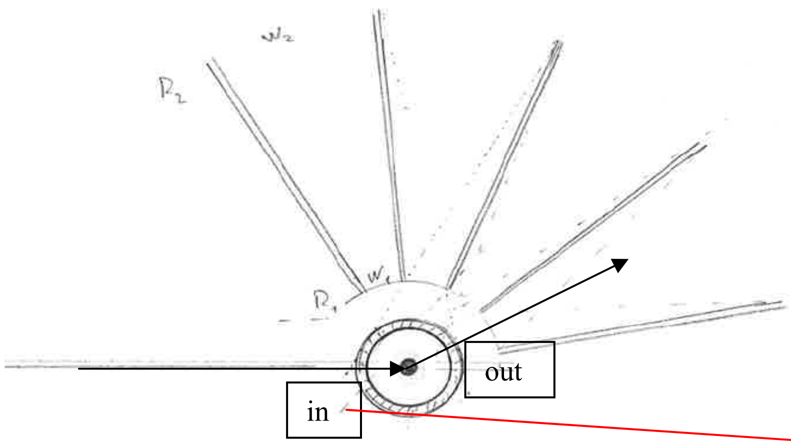
Task 3.2: Background by Sample Environment

1. Add a concentric aluminum cylinder of 7 cm diameter and 0.2 mm (Alu_cont.env) thickness around the sample using twice the module 'sample_environment', now transmitted neutrons must be treated as well

2. Estimate the background by
   - coloring neutrons and
   - checking the log file and/or separating the contributions using eval_elast

(some trajectories are not treated properly, bug will be fixed i the next version)



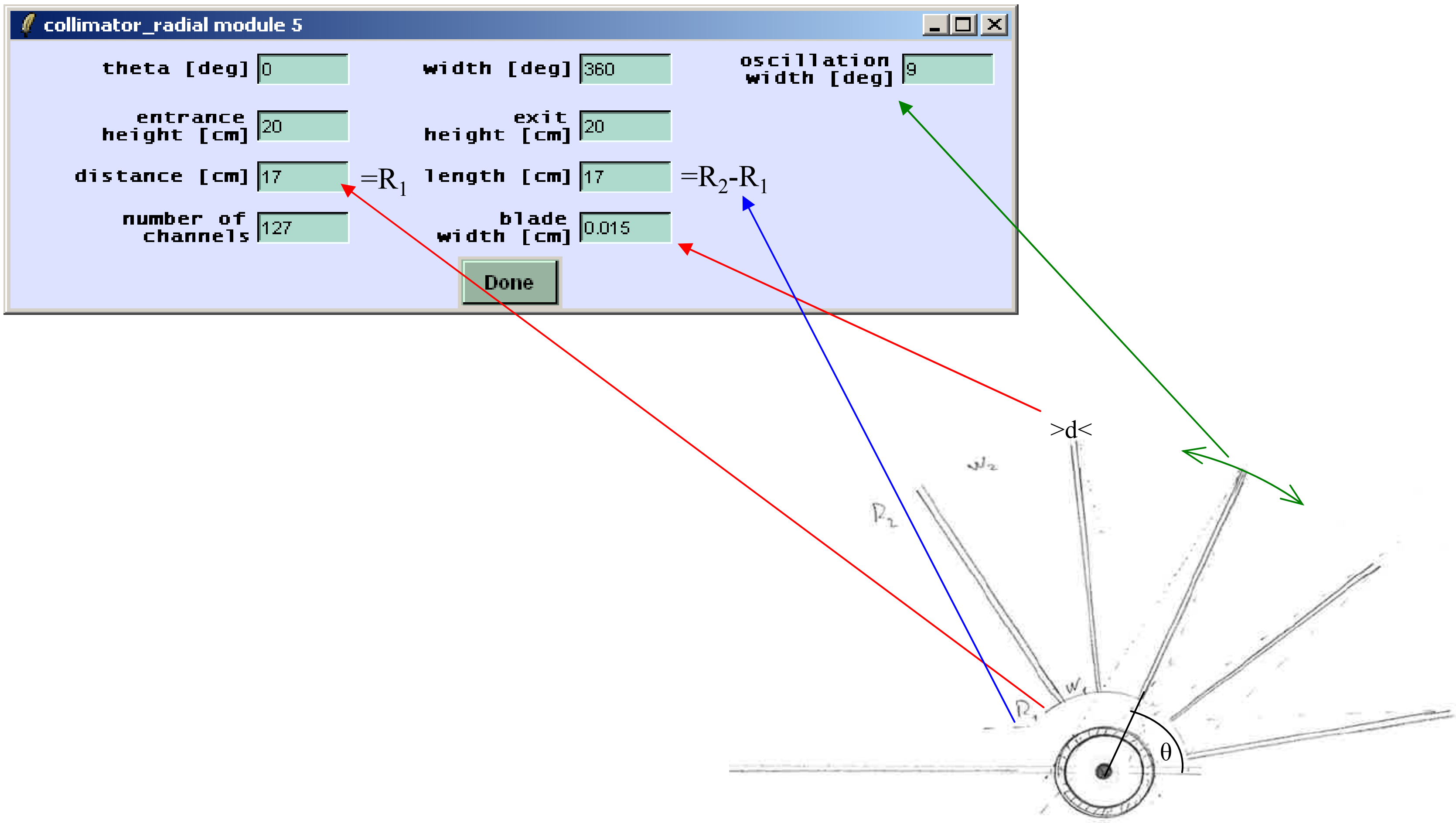


Stop beam in front of the sample environment

essworkshop.
Ven 2010

Task 3.2: Radial Collimator

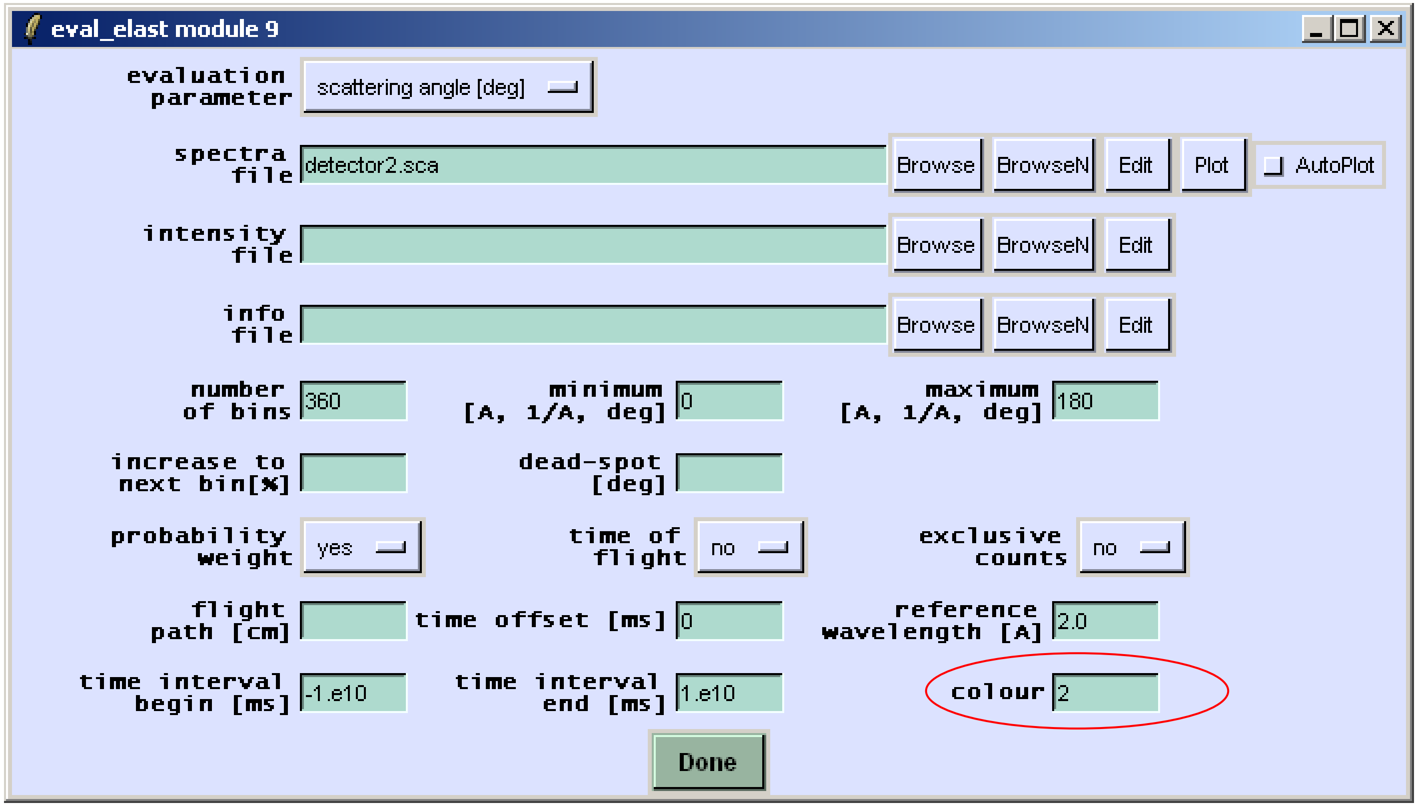
1. If you like: add a radial collimator around the aluminum cylinder and check the reduction of background



Task 3.2: Eval_elast

parameter:

scattering angle
d-spacing
Q



Welcome, days 4+5

Thursday program:

9:00-13:00 Morning session, including
- Samples / sample environment
- Detectors
- (Coffee break)

13:00-15:00 Lunch + break (walk to Kyrkbacken harbour?)

15:00-1900 Afternoon session,
- Entire instrument, assembled from previous exercises
- (Coffee break)

After dinner: Online presentation “how to write a component”

Welcome, days 4+5

Friday program:

9:00-11:30 Morning session:
- Discussion
- Training feedback
- Individual user projects
- Closure

apx. 12:20 Departure from Backafallsbyn

12:40 Ferry to Landskrona

4- Detectors

4.1 A small Gas detector (BIDIM26, do it yourself)

4.2 Effect of the housing

Detectors: introduction

Simple ideal detectors are usually part of any simulation.
Efficiency is 100%. I personally use Monitor_nD.

I will present how we simulate more realistically gas detectors.

When neutron enter a gas cell, it creates at some point a (p,t) pair.
These charges drift, under electrical field, to a wire where the position is detected e.g. by charge division and coincidence criteria.

A cloud of charges is thus created around an incoming trajectory.

Let's see that...

Detectors: a multi-wire gas chamber

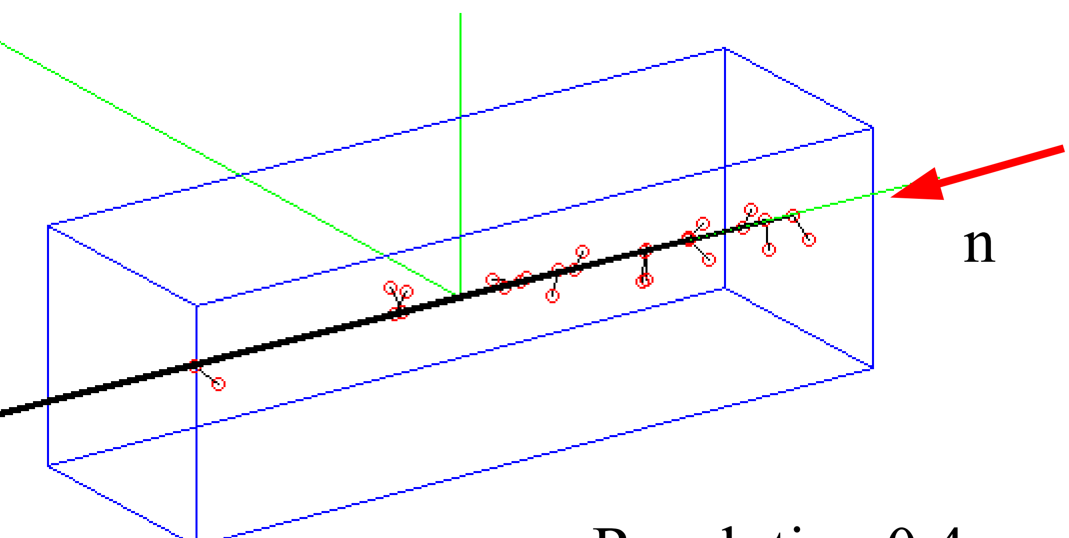
Detector model for MWPC:

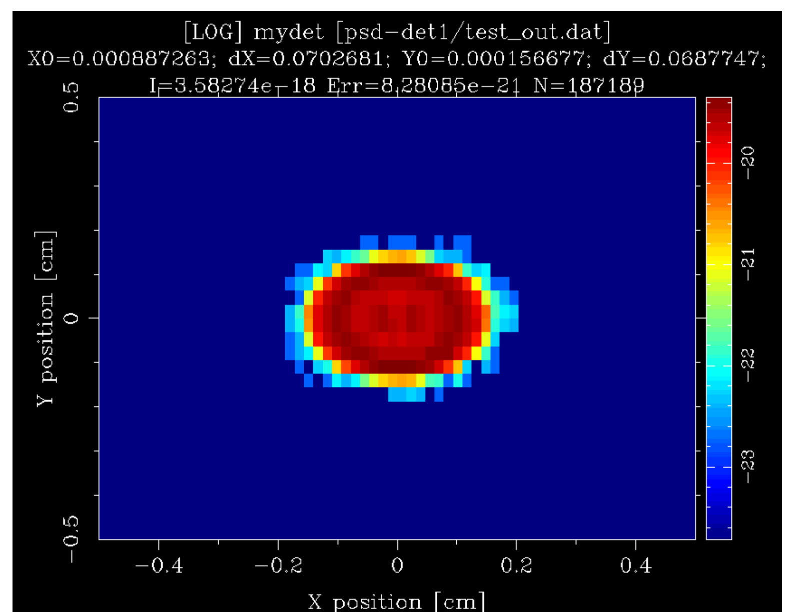
Gas chamber with wires, (p,t) charge drift

Can study:

- detector spatial resolution
- background generated from detector housing.
- estimate detector saturation (cur. not implemented)

Detection area 1x1 cm, He 5 bars, CF4 1 bar.





Resolution 0.4 cm

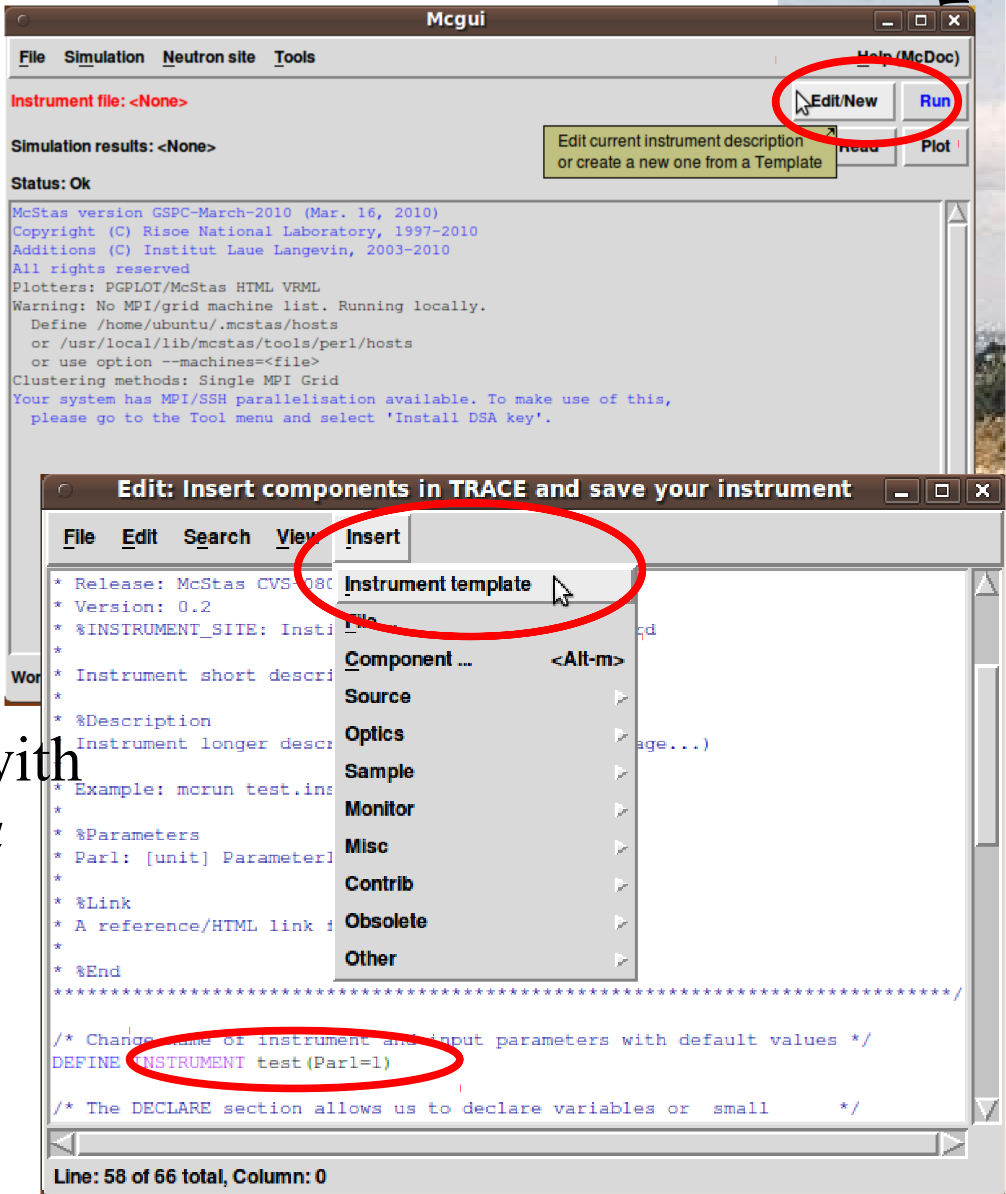
Detectors: simulating a simple gas cell

Select button **Edit/New.** *The Editor opens*
Select menu **Insert/Template** in the Editor

Change the instrument **name** as **PSD_test** and parameter ***lambda=2***

Position the **cursor** after the **TRACE** keyword and ***Progress_bar***

Insert a **Source/Source_simple** and call this instance '*Source*'
Make it a disk of ***radius=0.1*** [mm], ***focusing*** to a 0.1 x 0.1 [mm^2] at 2 m with neutron wavelength ***lambda0=lambda dlambda=0.1***



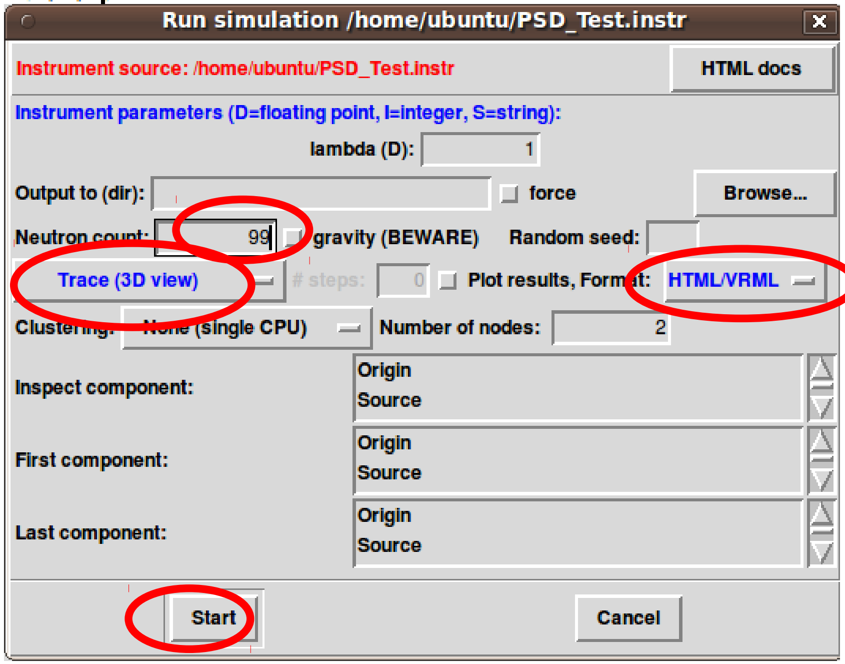
Detectors: simulating a simple gas cell

Insert a **Contrib/PSD_Detector**
at 2 [m] from the *PREVIOUS* component. Make it a BIDIM26 Detector, but with 2.6x2.6 cm2 FN_Conv="He3inHe.table", FN_Stop="He3inCF4.table"

```
COMPONENT Source = Source_simple(
    radius = 0.1e-3, focus_xw = .1e-3, focus_yh = .1e-3,
    lambda0 = lambda)
  AT (0, 0, 0) RELATIVE PREVIOUS

COMPONENT MILAND19 = PSD_Detector(xwidth=0.192, yheight=0.192, nx=64, ny=64,
                     zdepth=0.03, threshold=100, borderx=-1, bordery=-1,
                     PressureConv=5, PressureStop=1,
                     FN_Conv="Gas_tables/He3inHe.table", FN_Stop="Gas_tables/He
                     xChDivRelSigma=0, yChDivRelSigma=0,
                     filename="BIDIM19.psd")
  AT (0,0,2) RELATIVE PREVIOUS
```

Save instrument as '*PSD_test.instr*', and click the **Run** button.

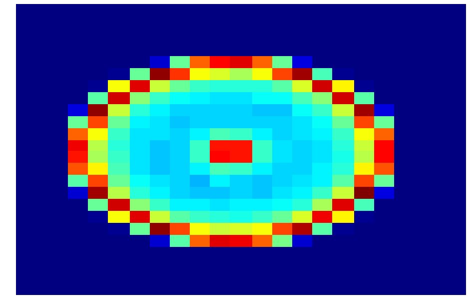


Select **Trace (3D)** instead of *Simulate*.

Click on the **Start** button.
Look at neutrons inside the detector
zoom with Z key, pass neutrons with SPACE key
Change to Format=PGPLOT and zoom.

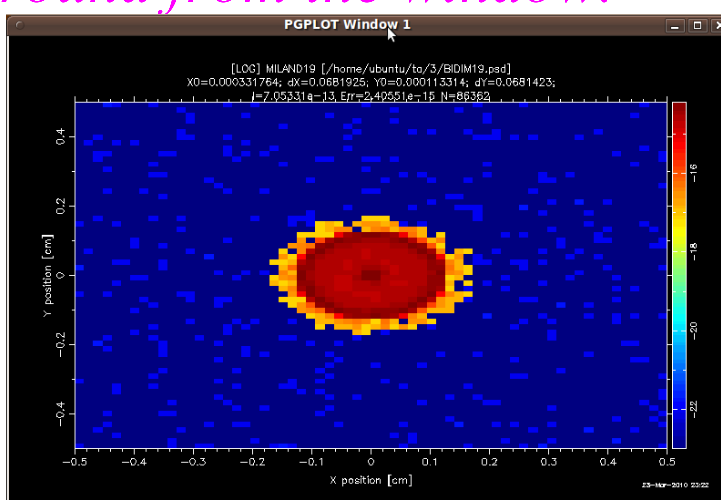
Detectors: adding housing

Close the Trace view and click again on the **Run** button.
Now select **Simulate** mode with ***Neutron count=1e6.***
Start simulation and **Plot** results.
What is the detector resolution ?



Add a 1 [mm] **Al** layer with *reflections="Al.laz"* in front of the detector, using the *PowderN* component. This is to model the detector entry window.

Launch a single simulation with *lambda=1* and Plot results. Show Log scale with 'L' key.
Estimate the background from the window.



Detectors: wavelength behaviour

Get the **Ex_4.instr** file from essworkshop.org/storage.

Improvements:

•'*window*' parameter to specify the housing material

•Ideal detectors for total signal and scattered from housing

Launch simulation with default *window*=Al.laz

Repeat with *window=Fe.laz* (to stand the pressure)

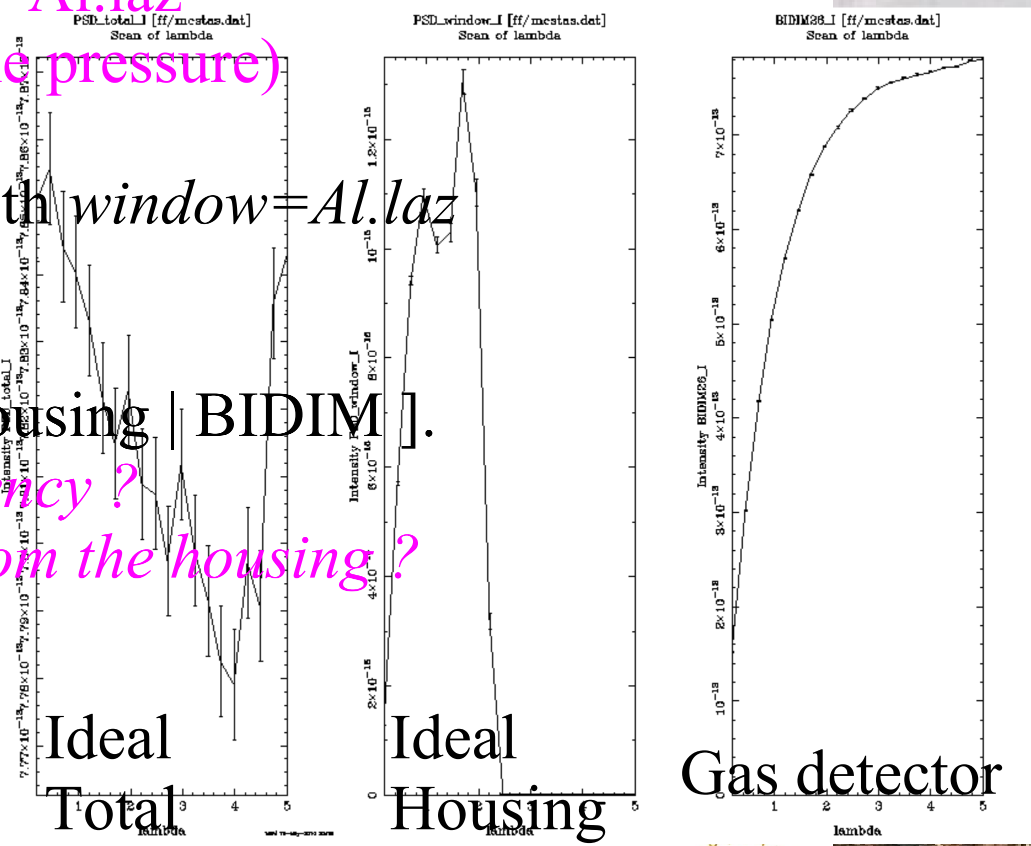
Launch a set of simulations scanning with *window=Al.laz*

lambda=.2,5 [Angs] in 20 steps.

Plot results [Ideal total | scattered on housing | BIDIM].

Comments about the gas detector efficiency ?

Comments about the scattered signal from the housing ?



Ven 2010
essworkshop.org

Ex. 5, a complete instrument

Retrieve the instrumentfile Ex_5.instr from the website

Edit the instrumentfile, removing monitors.

Insert an Arm instead:

```
COMPONENT Arm3 = Arm()
AT (0,0,1) RELATIVE Arm2
EXTEND %{
    flag_env = 0;
    flag_sample = 0;
%}
```

Paste the relevant sections of the instrumentfile from Exercise 3:

- Insert sample and sample environment
- The “string sample=.... “ input parameter
- DECLARE section with flags and
- Add relevant EXTEND section of the Origin component

Put the sample and sample environment AT (0, 0, 0) RELATIVE Arm3

1) Perform a TRACE and a simulation, to retrieve results similiar to Exercise 3 (Will require a higher ncount to achieve similar quality data)

(If a long simulation time - 15 minutes + - is required, this could be a good time for coffee...)

Info: To reduce simulation time in the following task, change your Arm3 to:

```
SPLIT 10 COMPONENT Arm3 = Arm()
AT (0,0,1) RELATIVE Arm2
EXTEND %{
    flag_env = 0;
    flag_sample = 0;
%}
```

(explanation will be given by organisers)

2) Using the component manual (Help (McDoc) - Component manual, page 113), adjust the options=”” string to record a (2θ,I) dataset (a powder pattern) of good quality (n=1e7 or more).

Also append the word “parallel” to your options string.

3) Vary the monochromator vertical radius of curvature (e.g. 0.5-3.5m) and observe the effect on the powder patterns .

Based on the integrated intensity (and line shapes), pick an optimal curvature

4) Optional exercises:

Insert a radial collimator between sample and detector (using pen/paper, choose the relevant divergence/geometry)

Insert a beamstop to remove the direct beam. (Where should it be put in the instrumentfile?)

Task 5: Instrument

1. Combine the Monochromator from Task 2.1 with sample, sample environment and detector from Task 3 to build a whole instrument