Introduction to McStas

Peter Willendrup[1,5], Emmanuel Farhi[2], Erik Knudsen[1,5], Emmanouela Rantsiou[3,6], Kim Lefmann[4,5]

[1]Physics Department, Technical University of Denmark, Denmark
[2]Calcul Scientifique, Institut Laue-Langevin, France
[3]Laboratory for Developments and Methods, Paul Scherrer Institute, Switzerland
[4]Niels Bohr Institute, University of Copenhagen, Denmark
[5]ESS design update programme, Denmark
[6]ESS design update programme, Switzerland

McStas

Introduction to McStas, Berlin MC school 2013

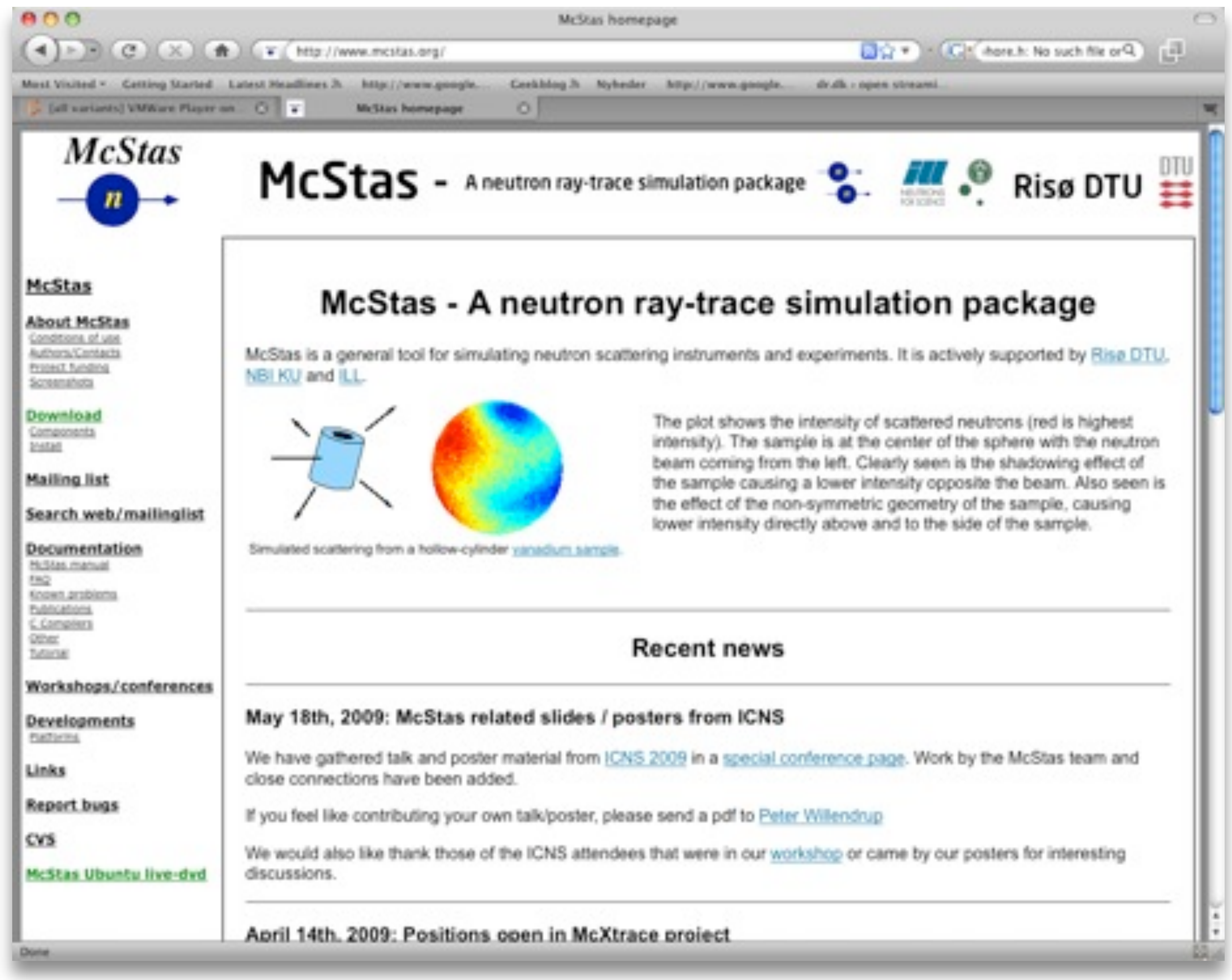
Agenda

- An introduction to McStas
- Instruments, components, the neutron ray
- Quick overview of example instruments
- A quick demo

McStas Introduction

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

GNU GPL license
Open Source



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

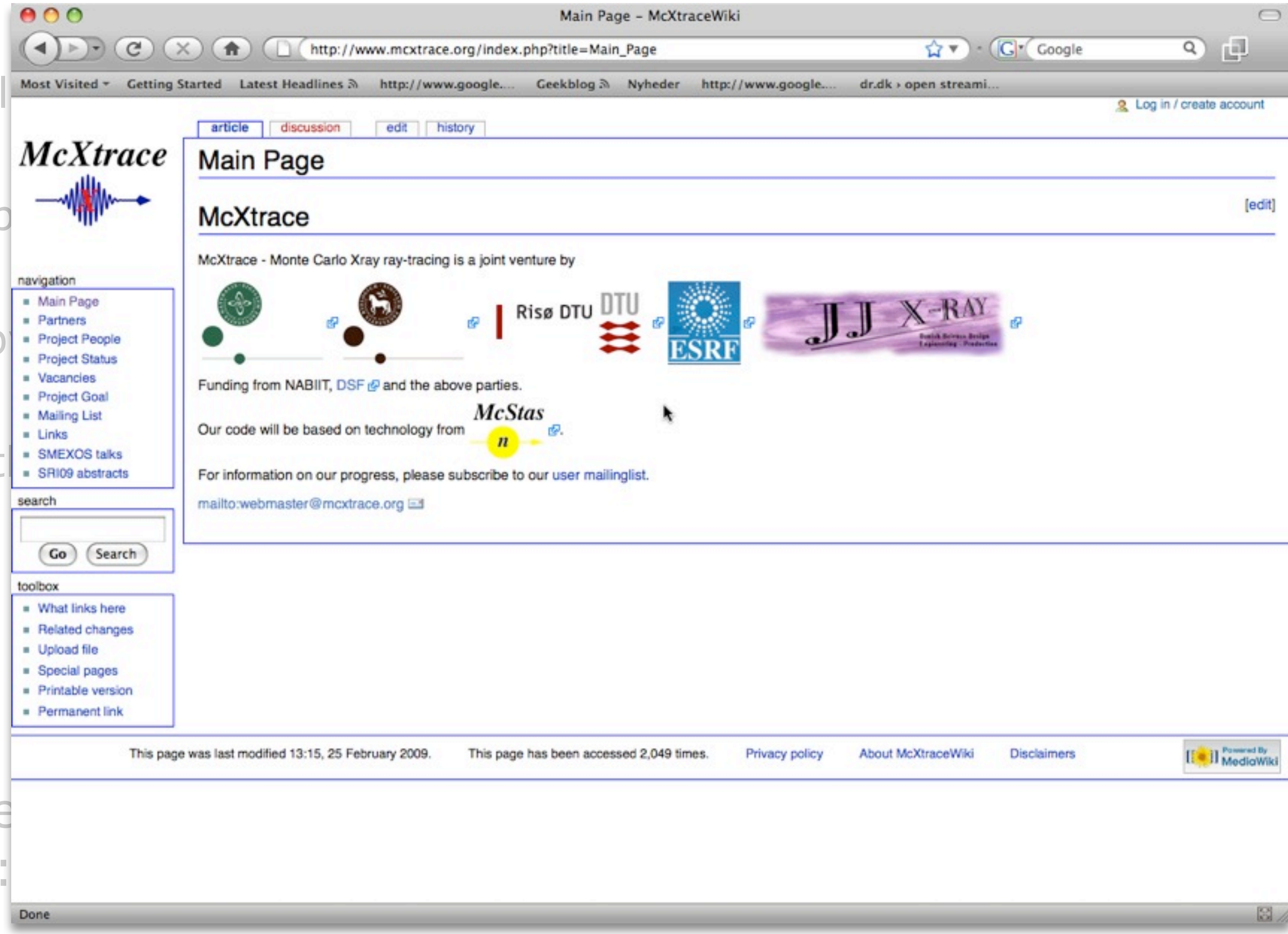
mcstas-users@mcstas.org mailinglist

McStas Introduction

McXtrace - since jan 2009 similar in X-rays

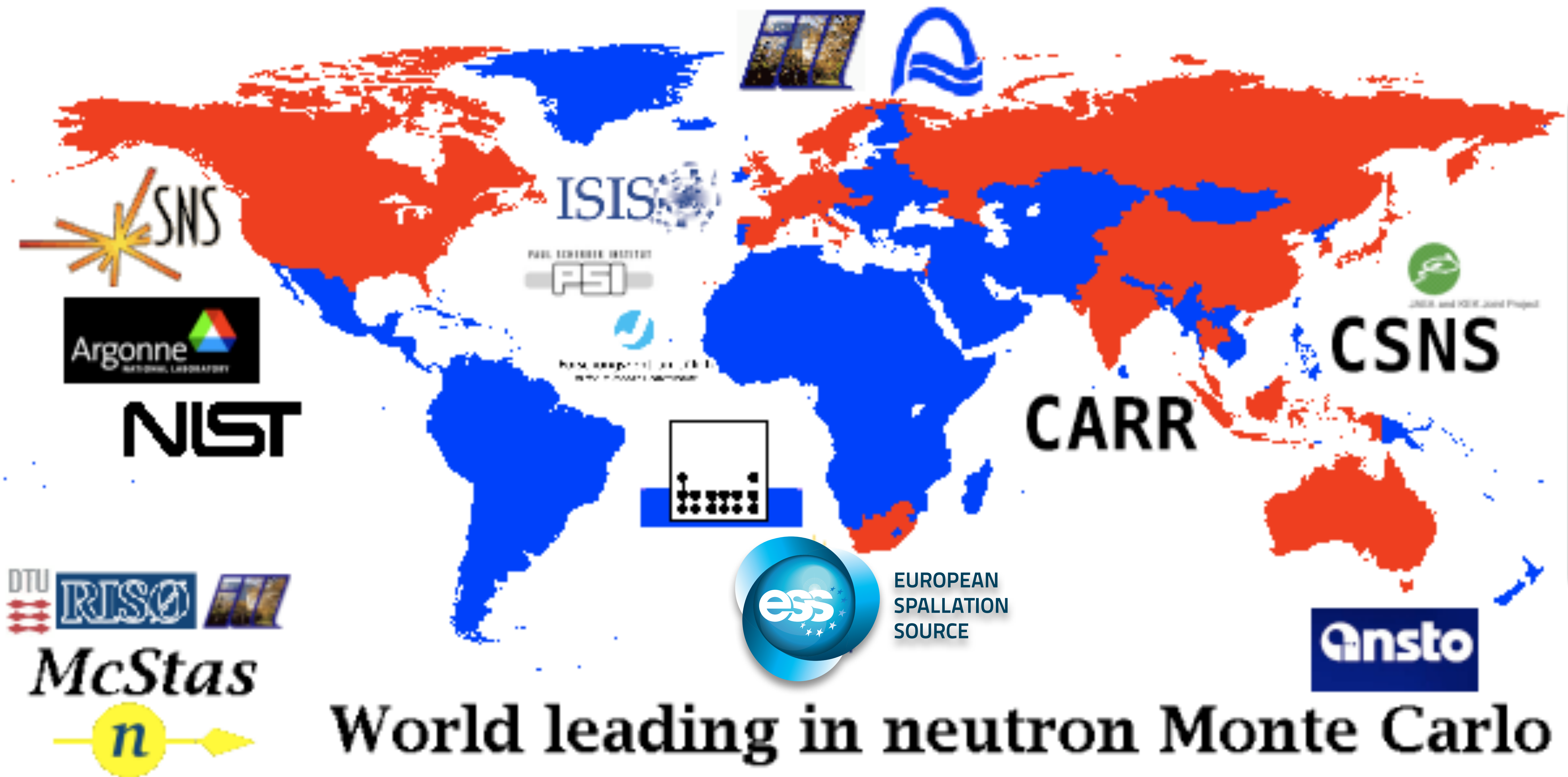
- Flexible, general simulation utility for neutron scattering experiments.
- Original ... s
- Develop ...
- V. 1.0 b ...
- Current ...

Proje ... http: ... list

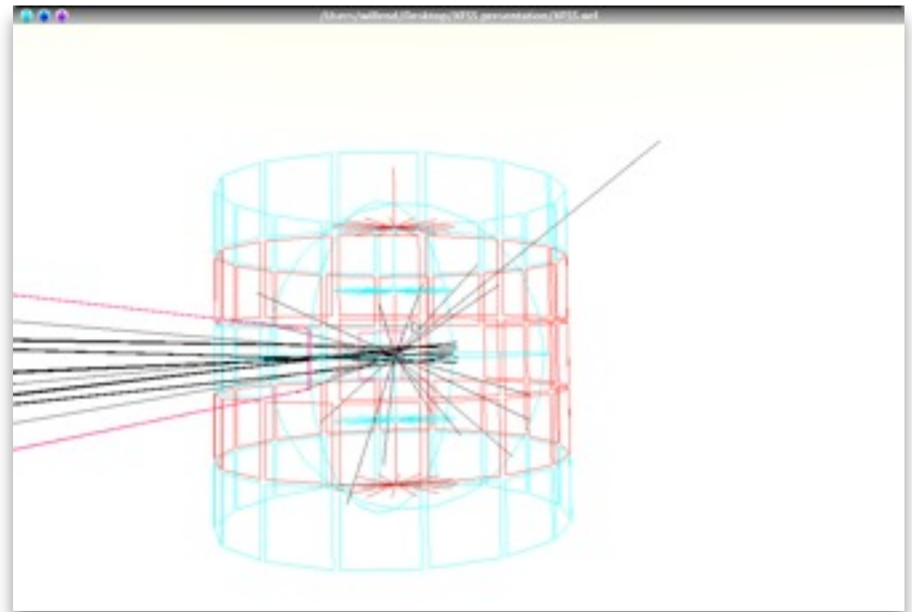


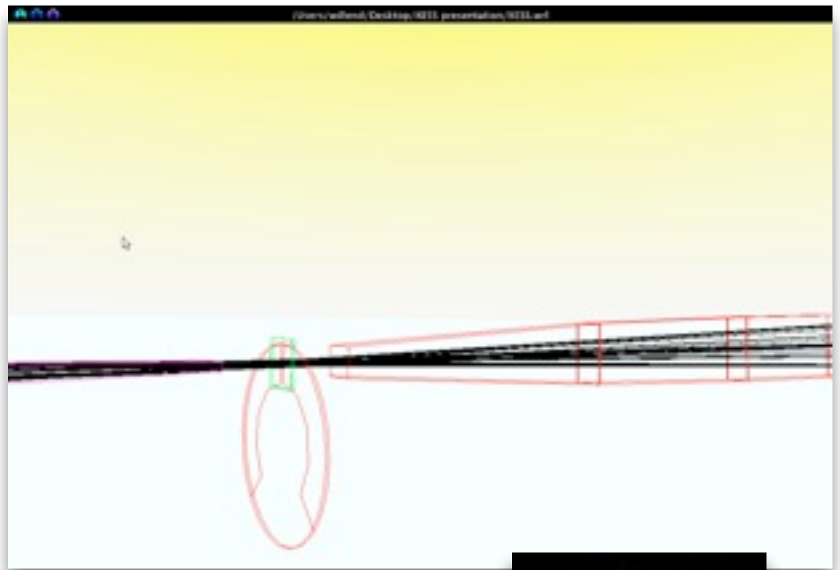
- Synergy, knowledge transfer, shared infrastructure

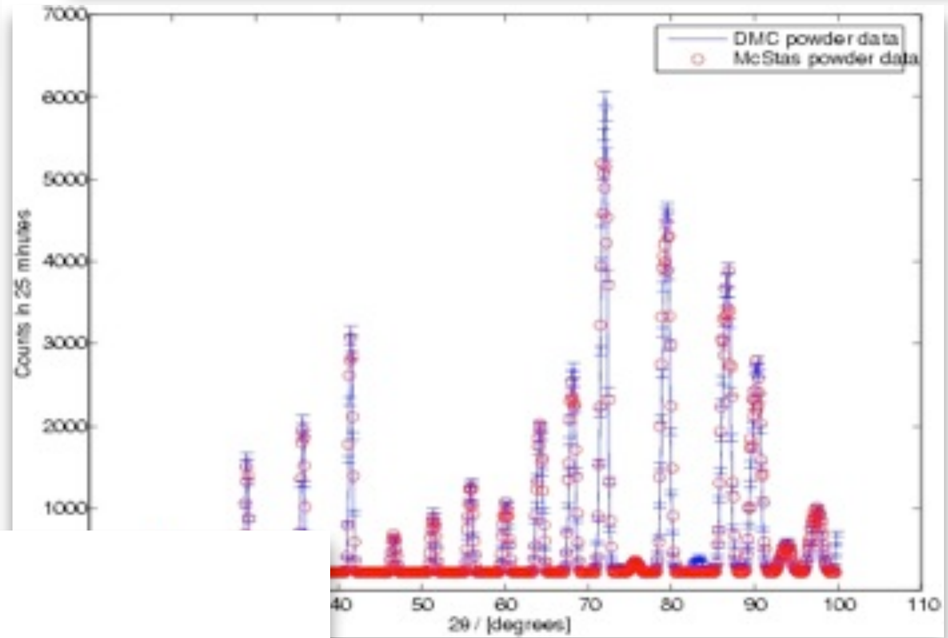
McStas Introduction



What is McStas used for?

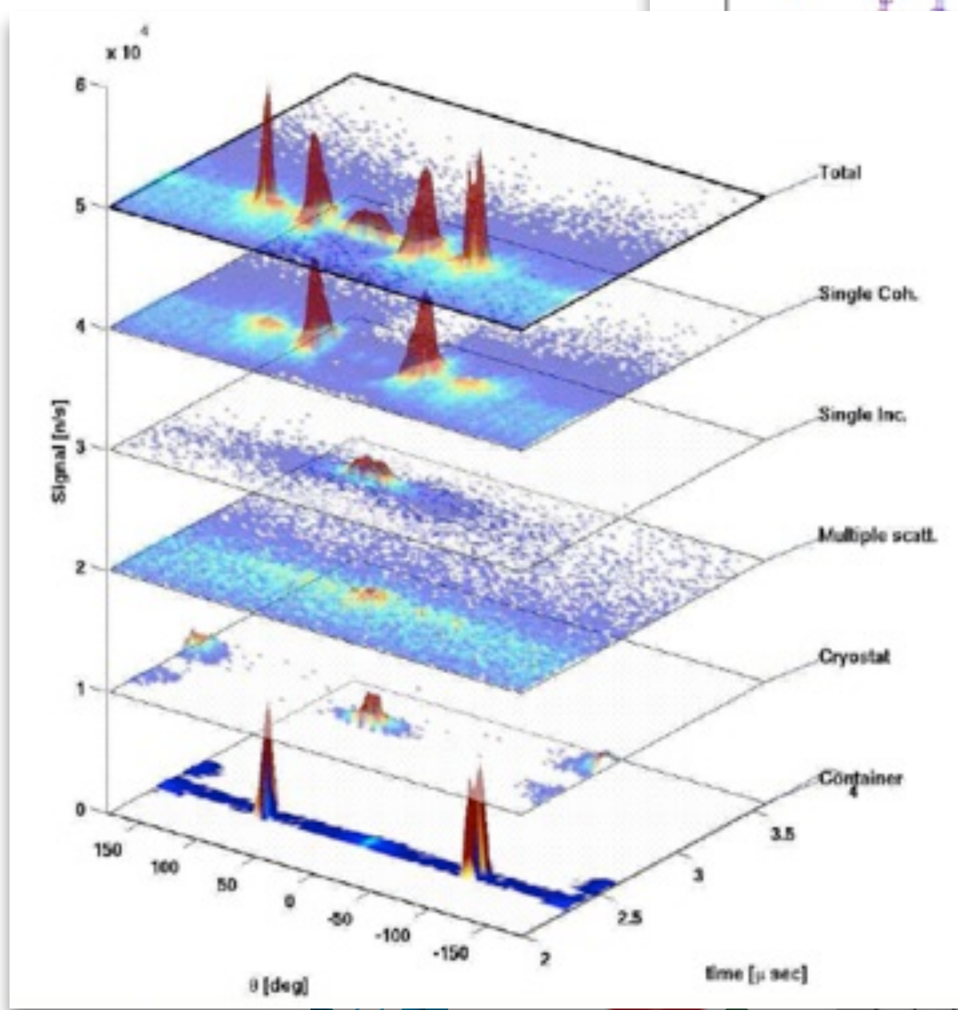


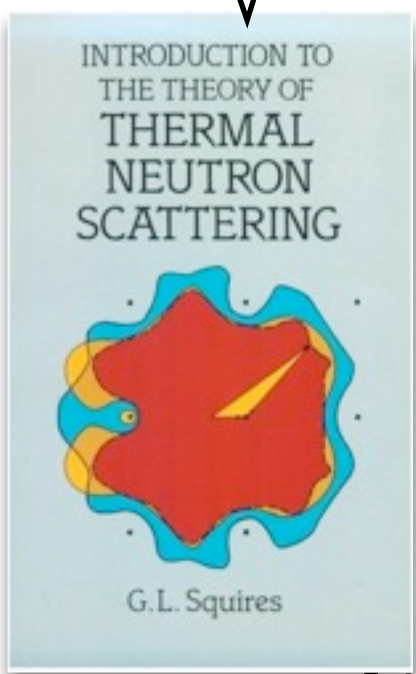


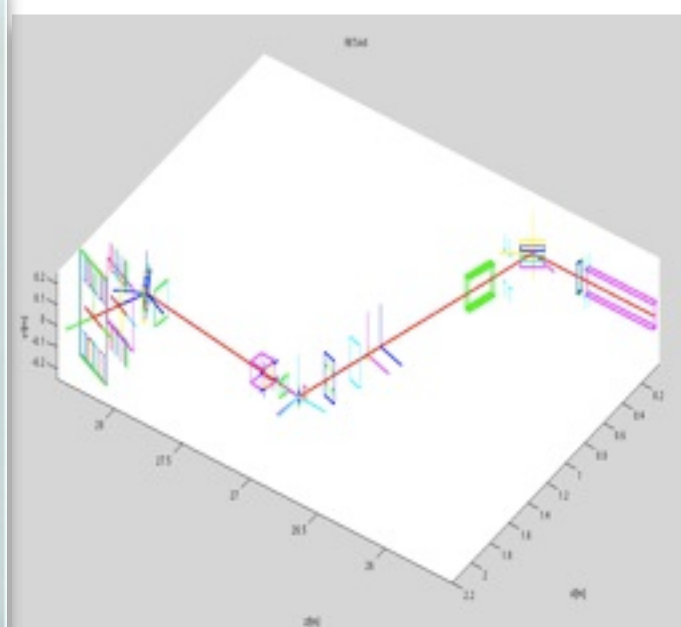


- Instrumentation
- Virtual experiments
- Data analysis
- Teaching

(KU, DTU, schools & workshops)

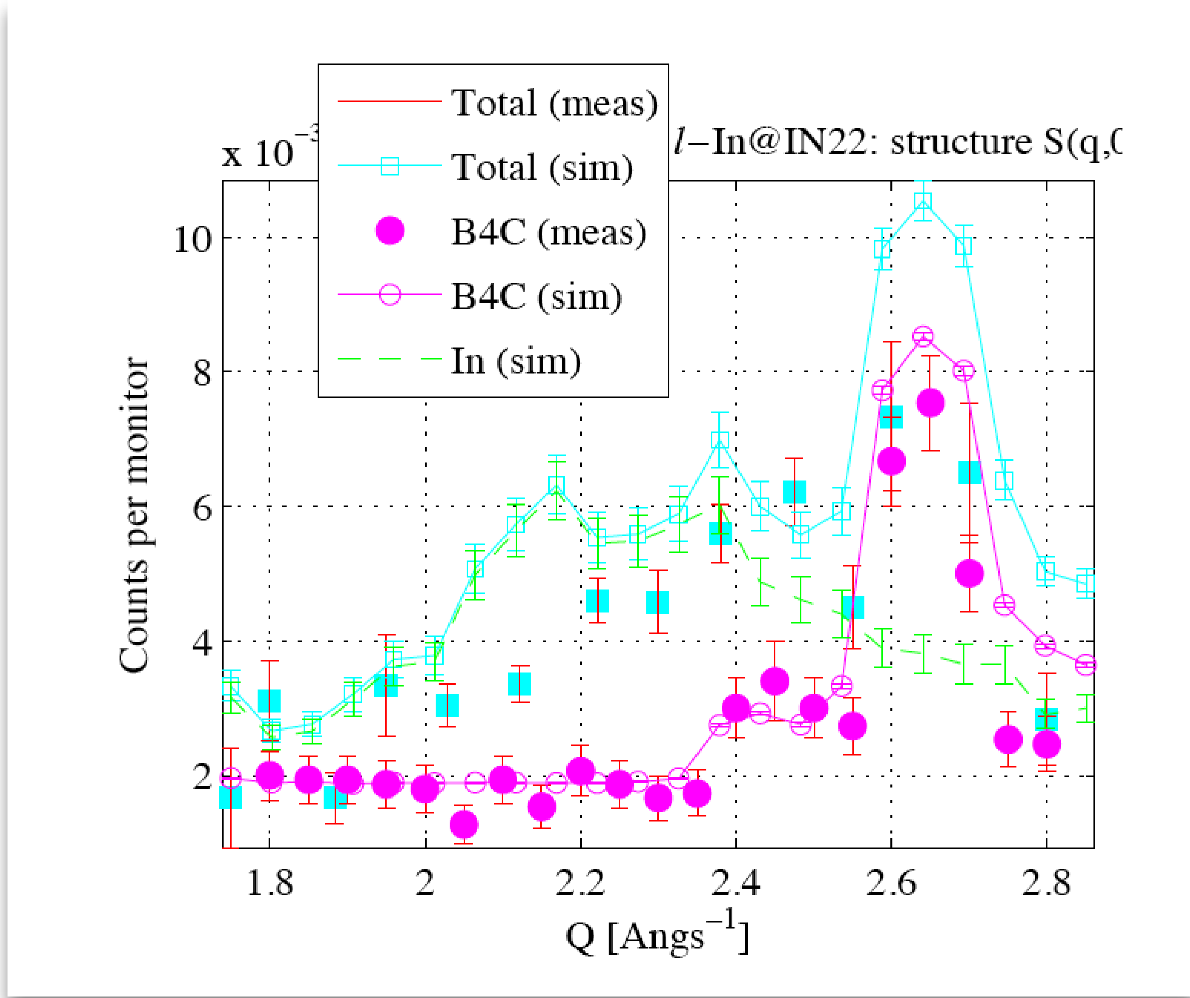




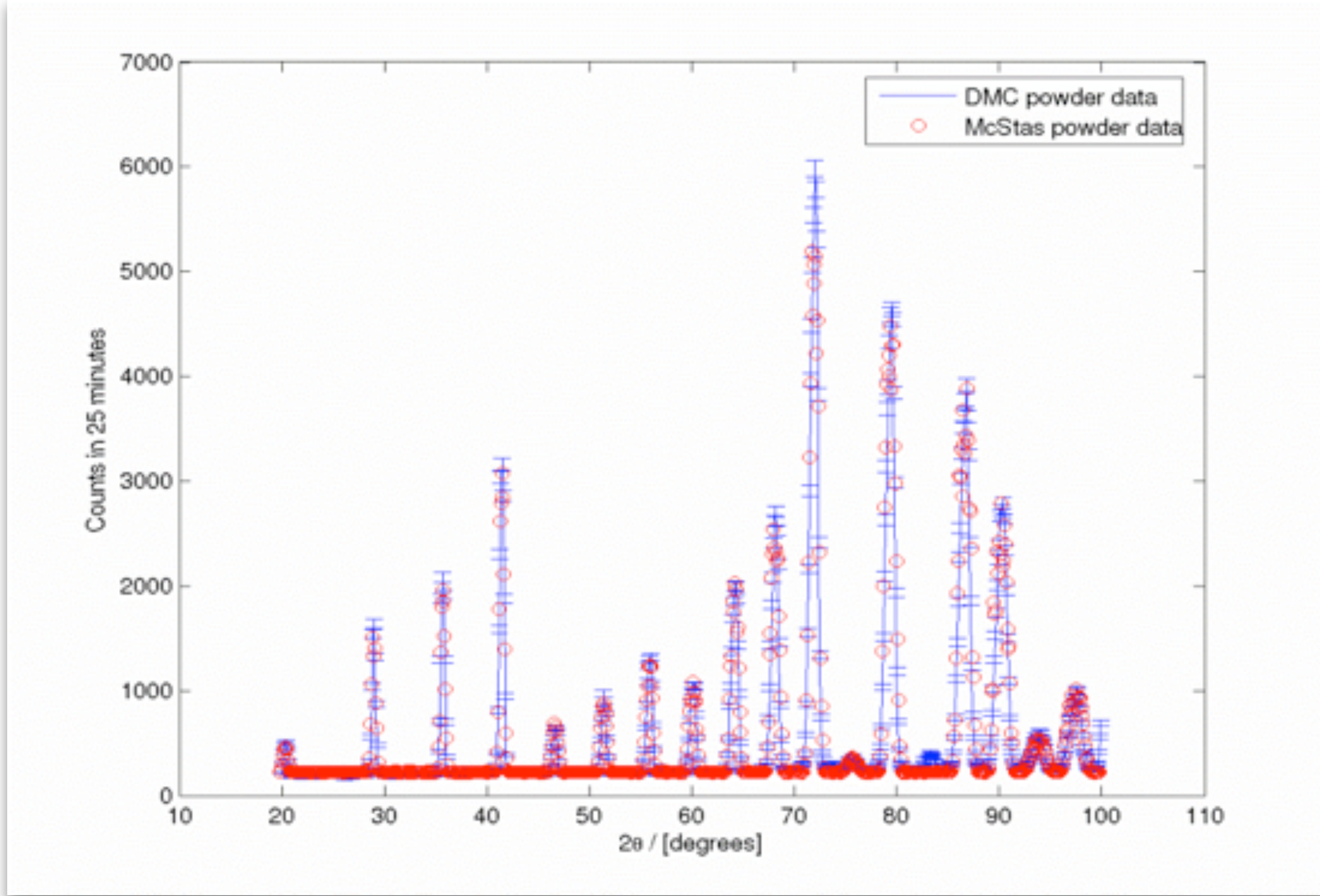


Reliability - cross comparisons

- 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%
- Common understanding: McStas and similar codes are reliable

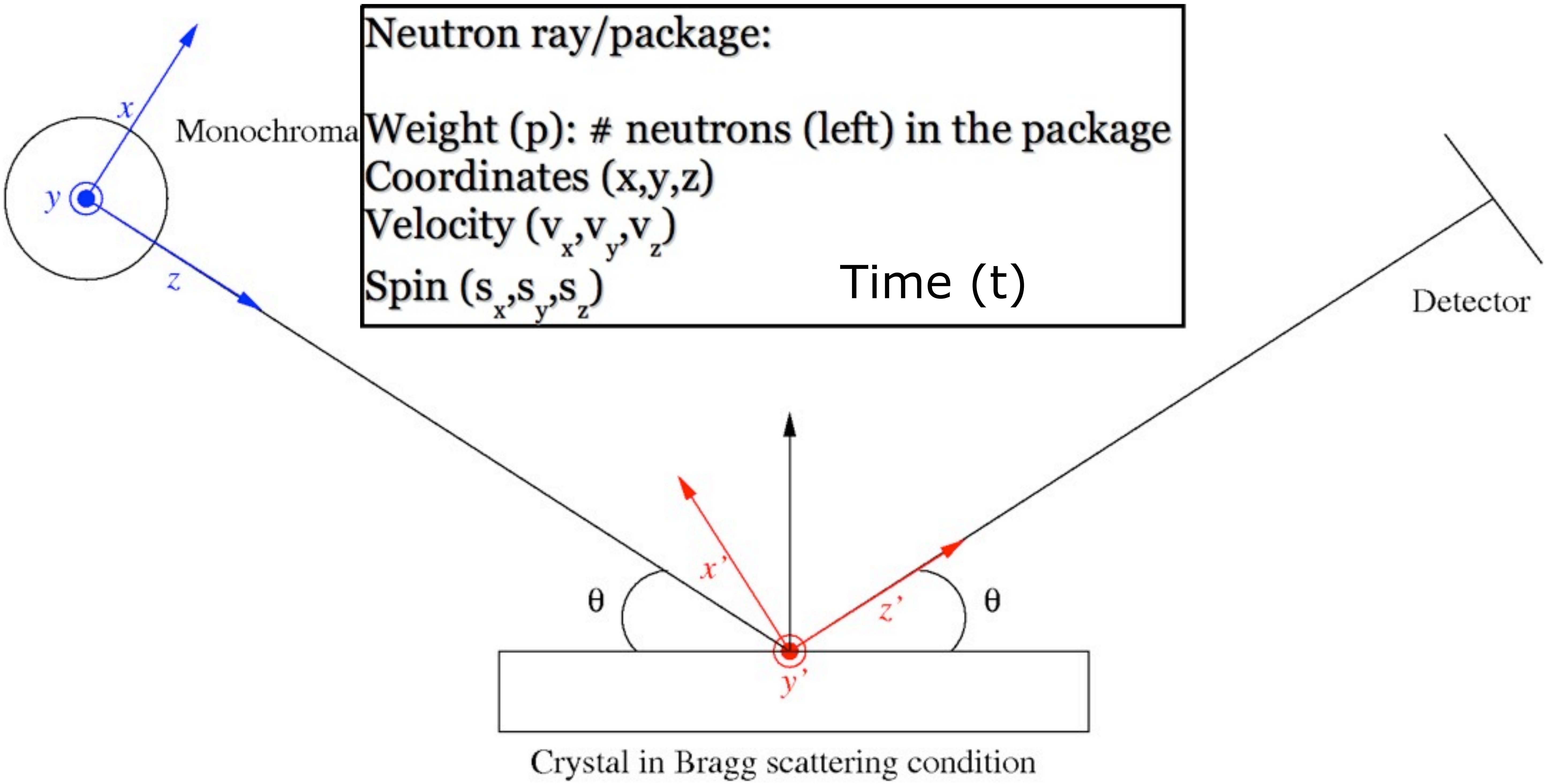


E. Farhi, P. Willendrup et al., in preparation



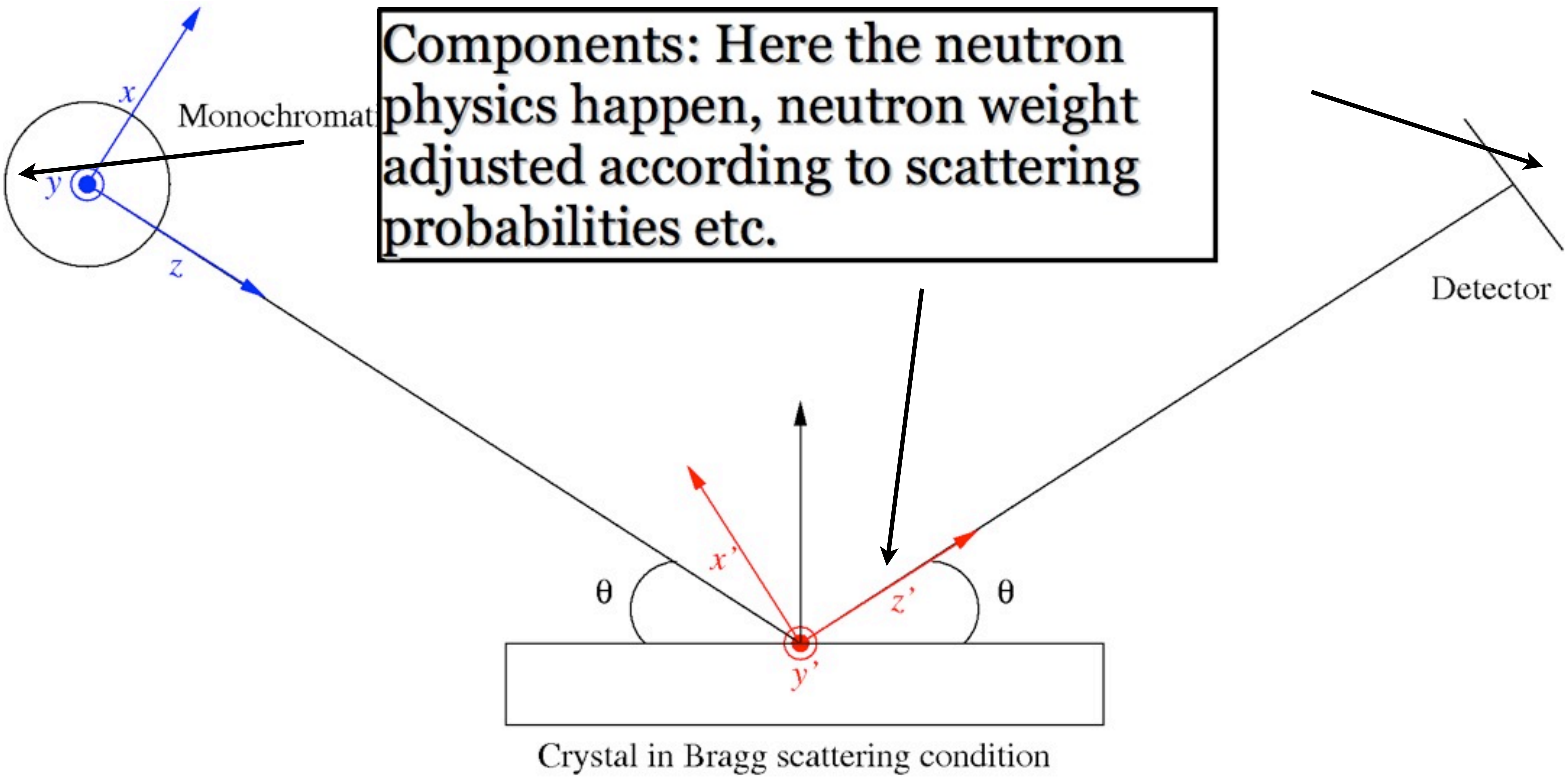
P. Willendrup et al., Physica B, 386, (2006), 1032.

McStas: key concepts

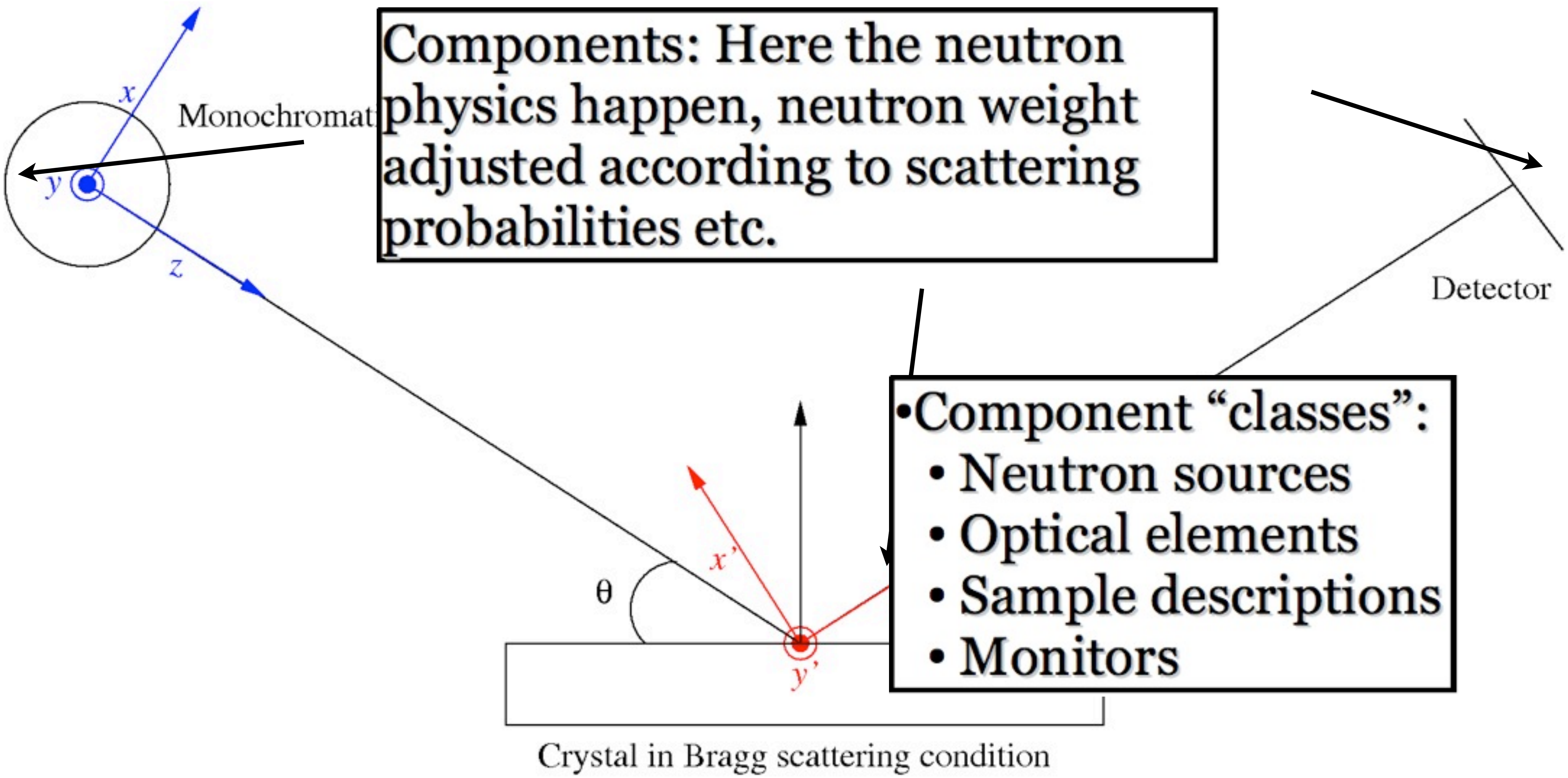


McStas: key concepts

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



McStas: key concepts



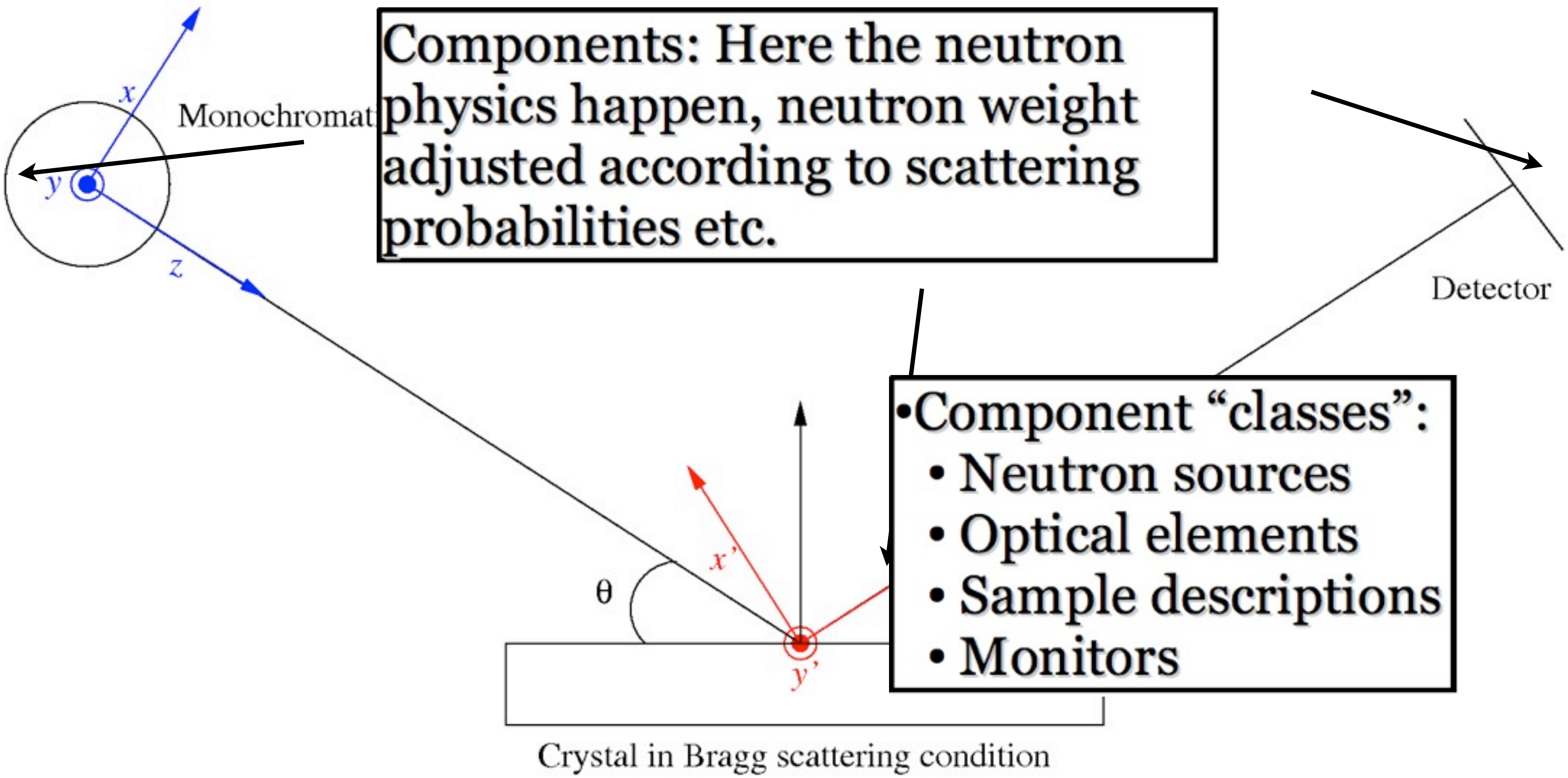
Components: Here the neutron physics happen, neutron weight adjusted according to scattering probabilities etc.

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

McStas: key concepts

Local, internal coordinate system!

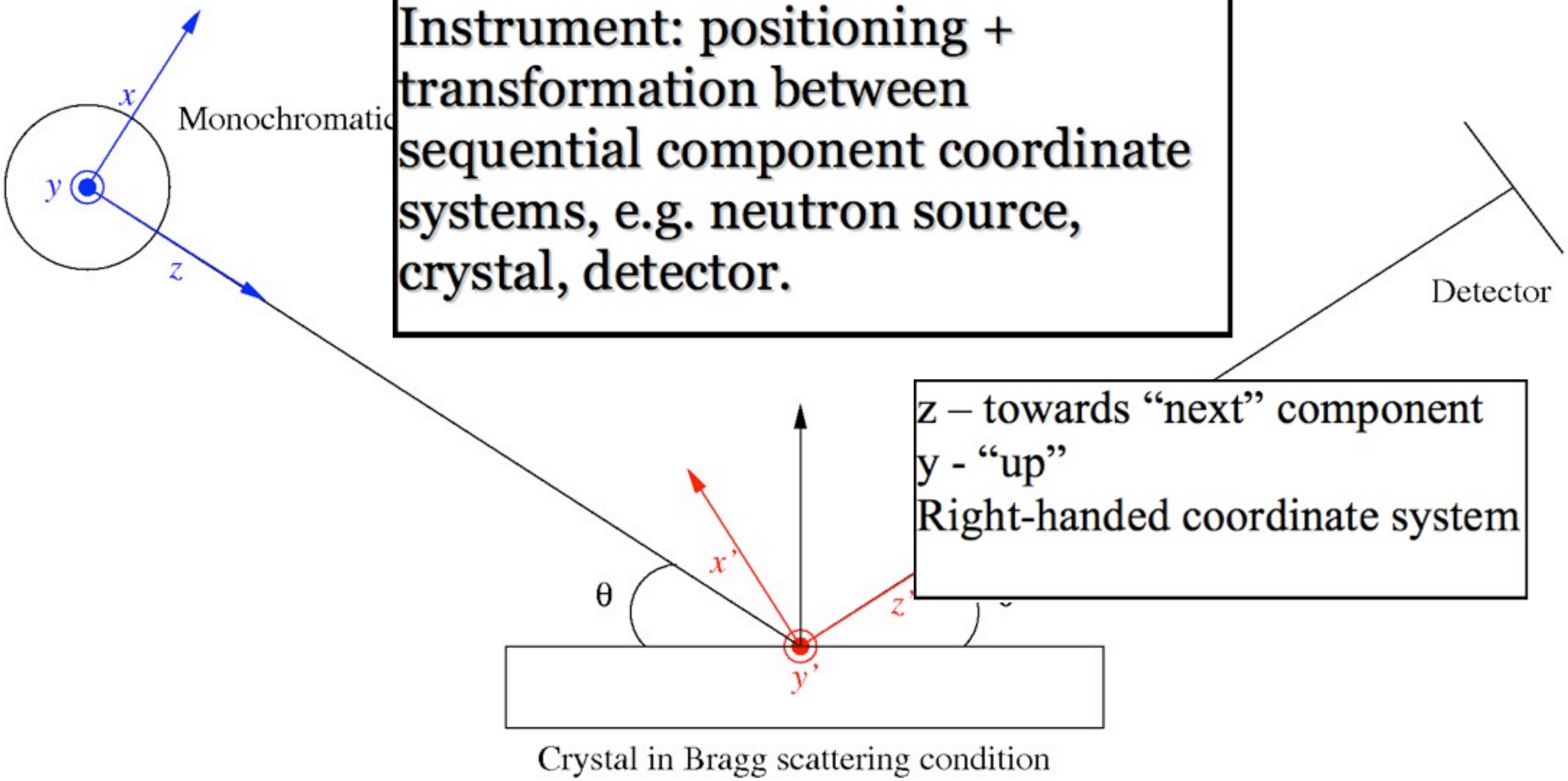
Components: Here the neutron physics happen, neutron weight adjusted according to scattering probabilities etc.



- 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.



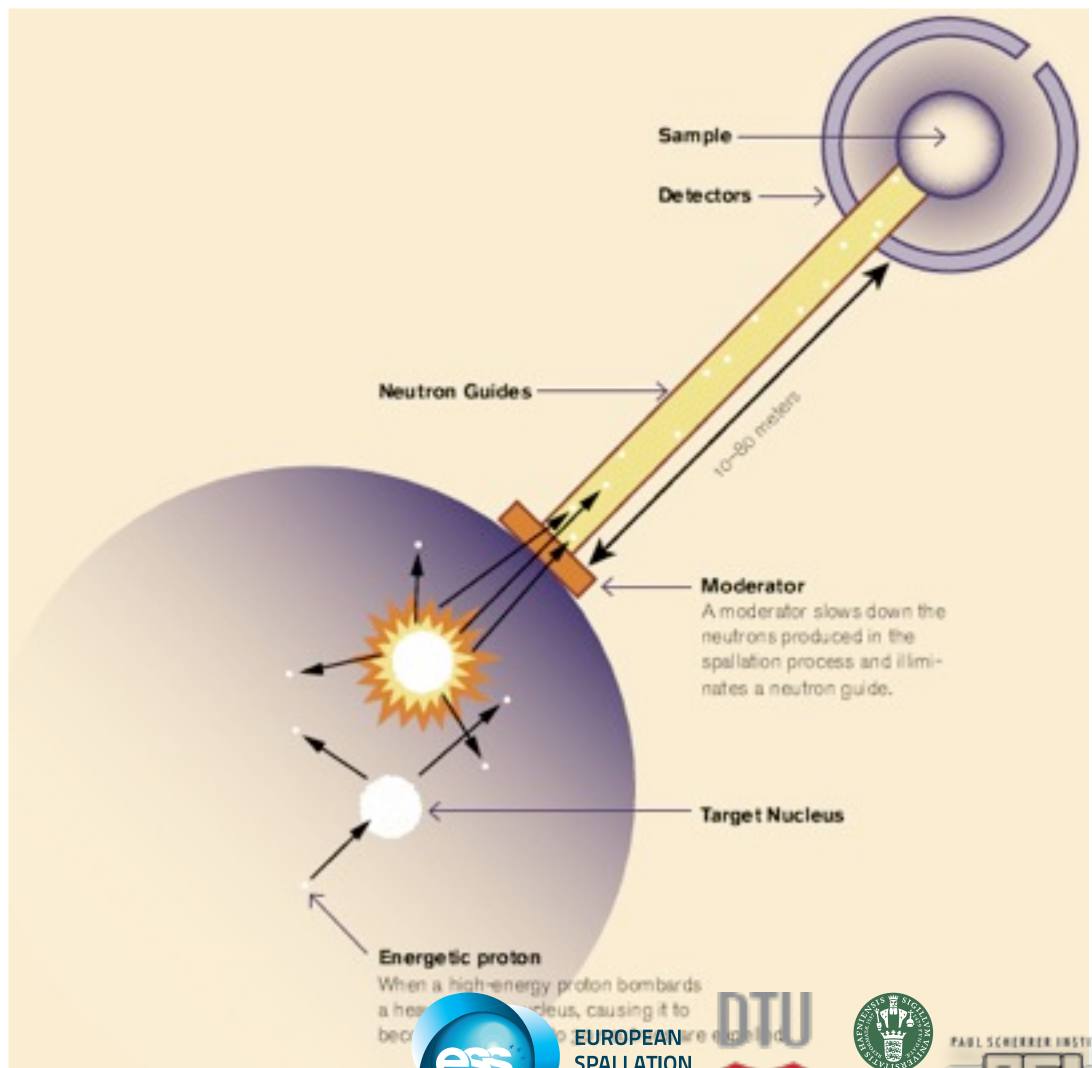
z – towards “next” component
y - “up”
Right-handed coordinate system

McStas overview

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

- Ran on everything from iPhone to 1000+ node cluster!

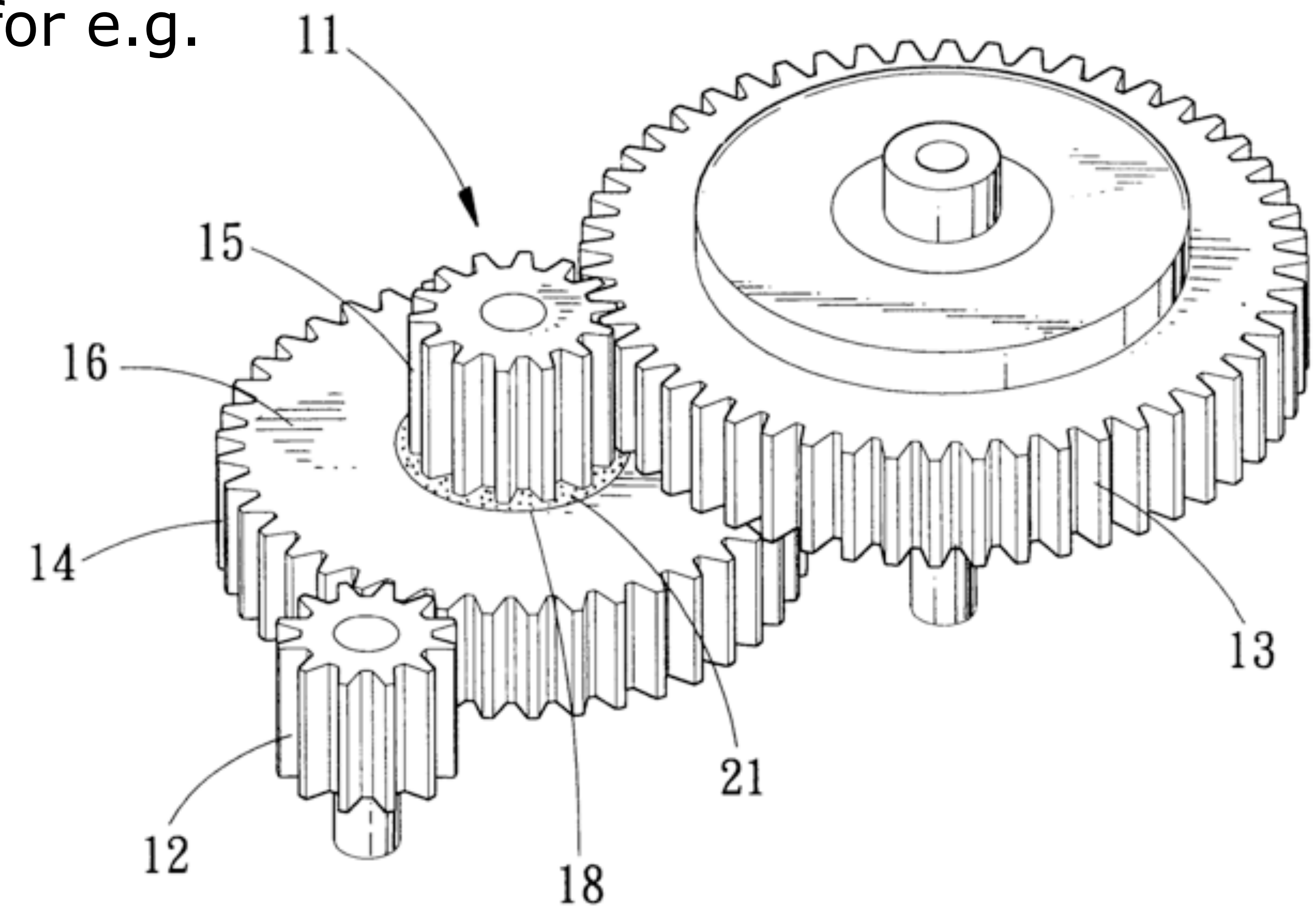
- 'Component' files (~100) inserted from library
  - Sources
  - Optics
  - Samples
  - Monitors
  - If needed, write your own comps



- DSL + ISO-C code gen.

Under-the-hood / inner workings

- Domain-specific-language (DSL) based on compiler technology (LeX+Yacc)
  - Simple Instrument language $\xrightarrow{\text{Code generation}}$ ISO C
- Component codes realizing beamline parts (including user contribs)
- Library of common functions for e.g.
  - I/O
  - Random numbers
  - Physical constants
  - Propagation
  - Precession in fields
  - ...



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

```c
/* 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!

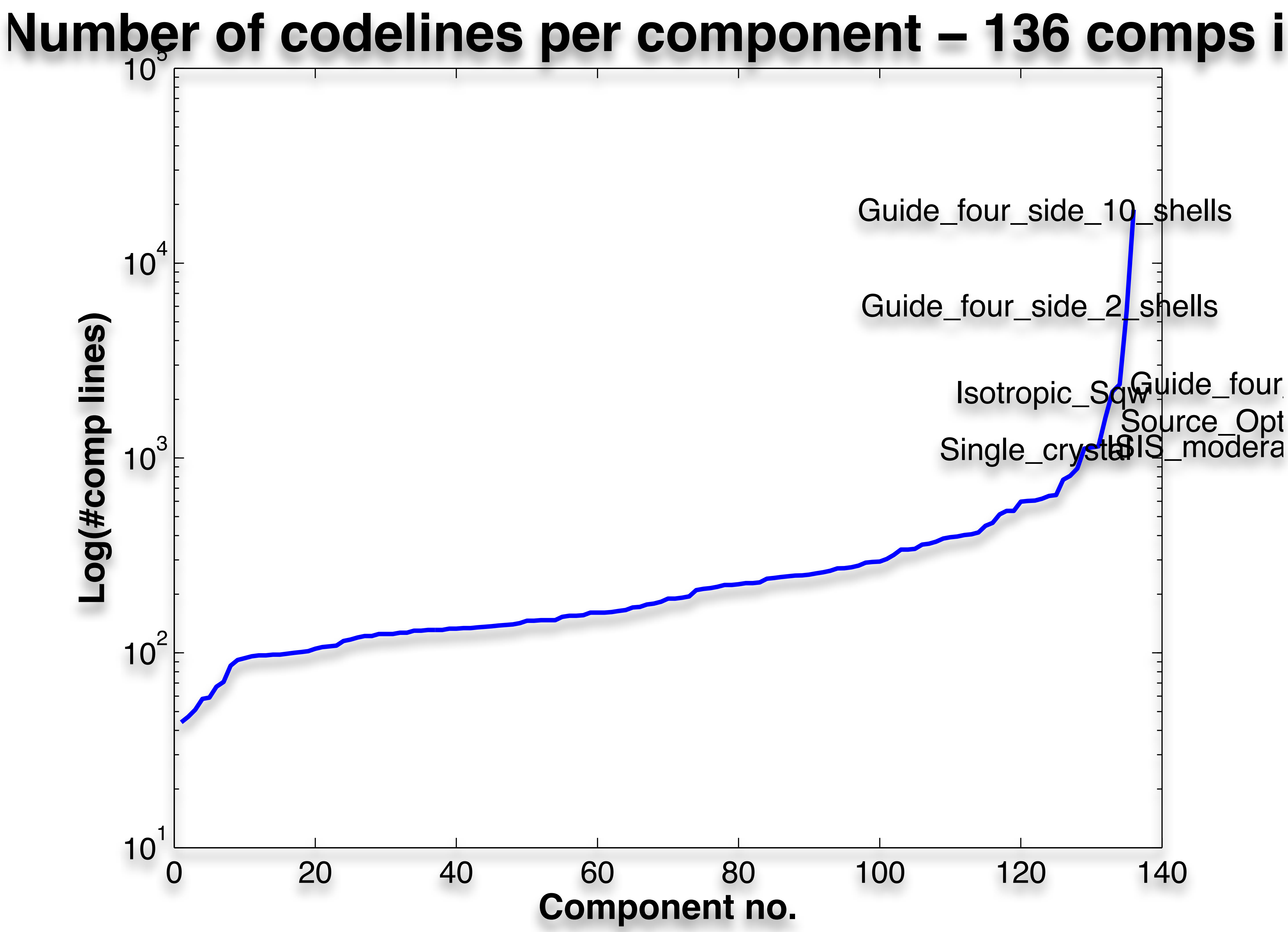
Input is .comp and .instr files + runtime functions for e.g. random numbers

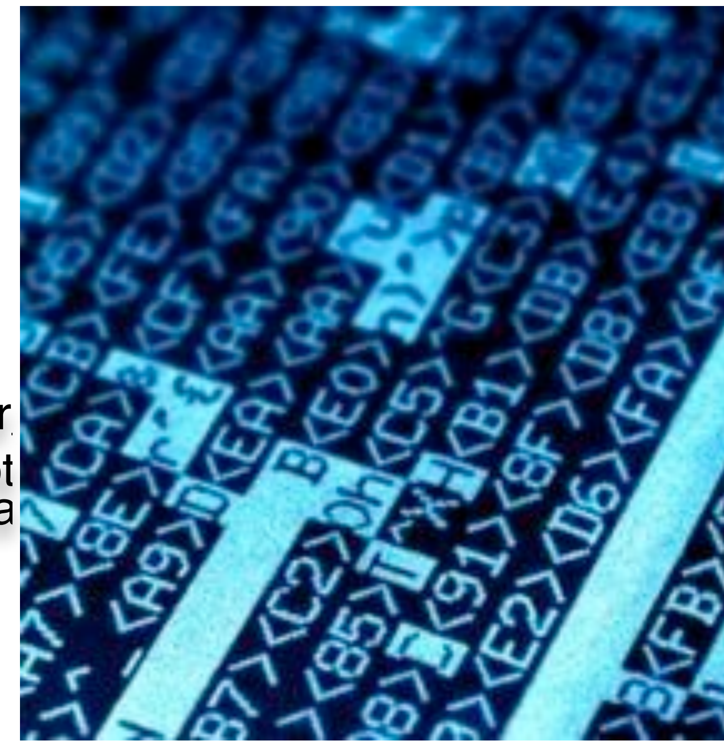
Output is a single c-file, which can be compiled using e.g. gcc.

Can take input arguments if needed.

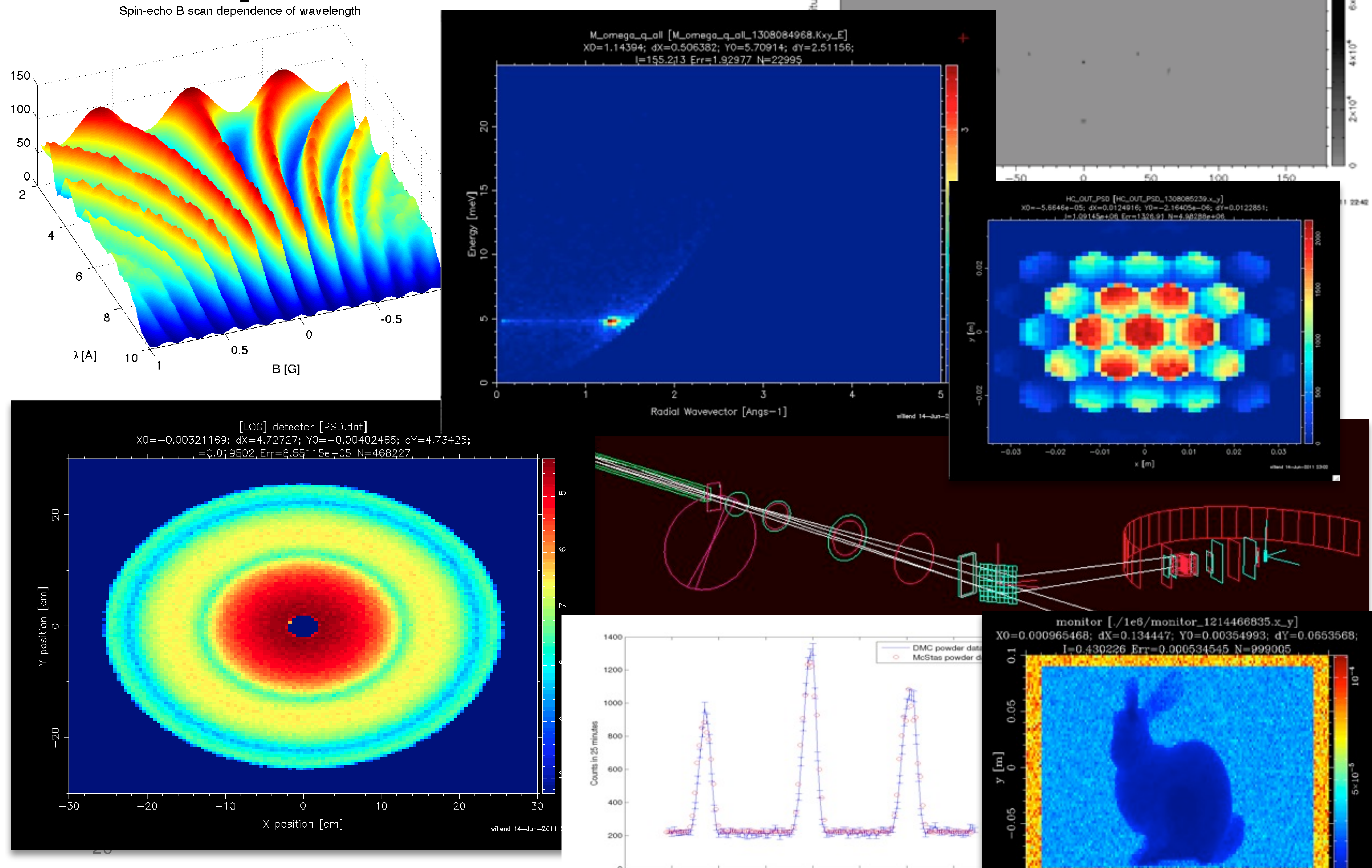
Writing new comps or understanding existing is not that complex...

- Check our long list of components and look inside... Most of them are quite simple and short... Statistics:





Example suite: 86 instruments



Including user contribs

- Well-developed community support
  - 30-40% of existing and new additions are from users
  - No direct refereeing of the code, but these requirements:
    - At least one test-instrument
    - Meaningful documentation headers (in-code docs)
  - Contributions go in dedicated contrib/ section of library

- Natural life-cycle of contrib's
  - Bug-fixes are applied both by contributor and developers
  - If contributor becomes unavailable either:
    - Many users of comp: Promote to official components, e.g. in optics/
    - Few/no users of comp: Move to obsolete/ until next major release

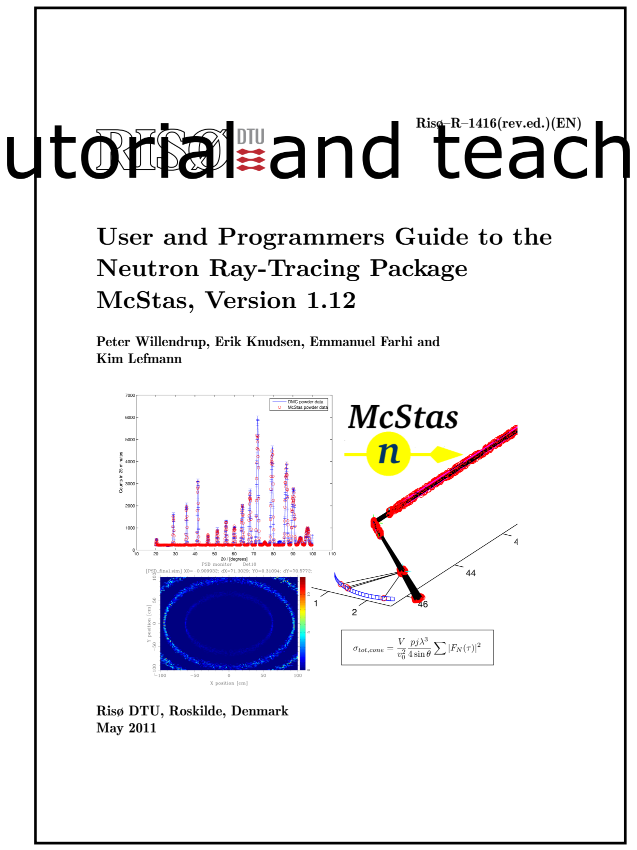
Documentation

- Basic use info is available inside comp & instr codes, extracte
  html

- 100+ page manuals documenting
  - Metalanguage
  - What is "under the hood"
  - Examples of practical use plus advanced features
  - Assumptions and algorithms applied in the components

- More than 70 example instruments

- Various tutorial and teach yourself solutions

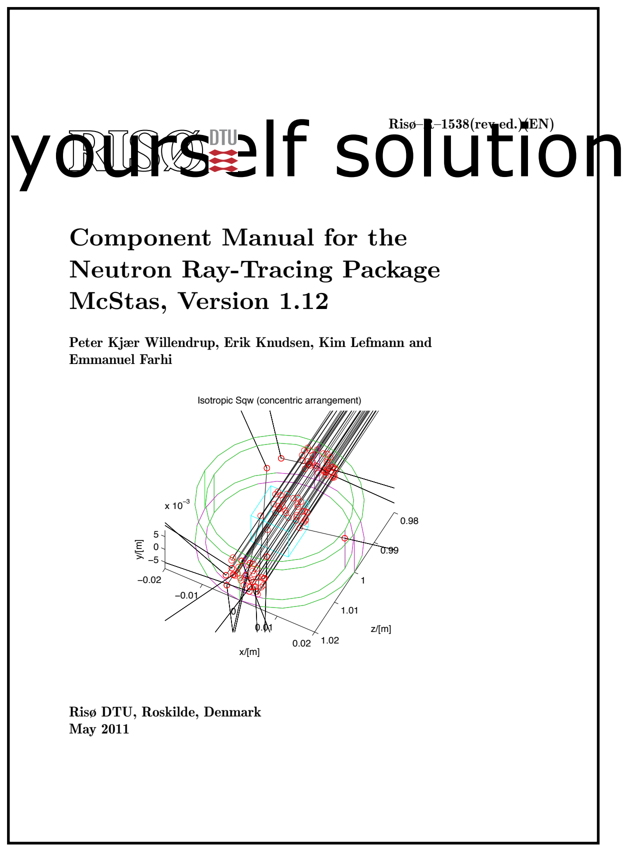


Risø–R–1416(rev.ed.)(EN)

User and Programmers Guide to the Neutron Ray-Tracing Package McStas, Version 1.12

Peter Willendrup, Erik Knudsen, Emmanuel Farhi and Kim Lefmann

Risø DTU, Roskilde, Denmark
May 2011

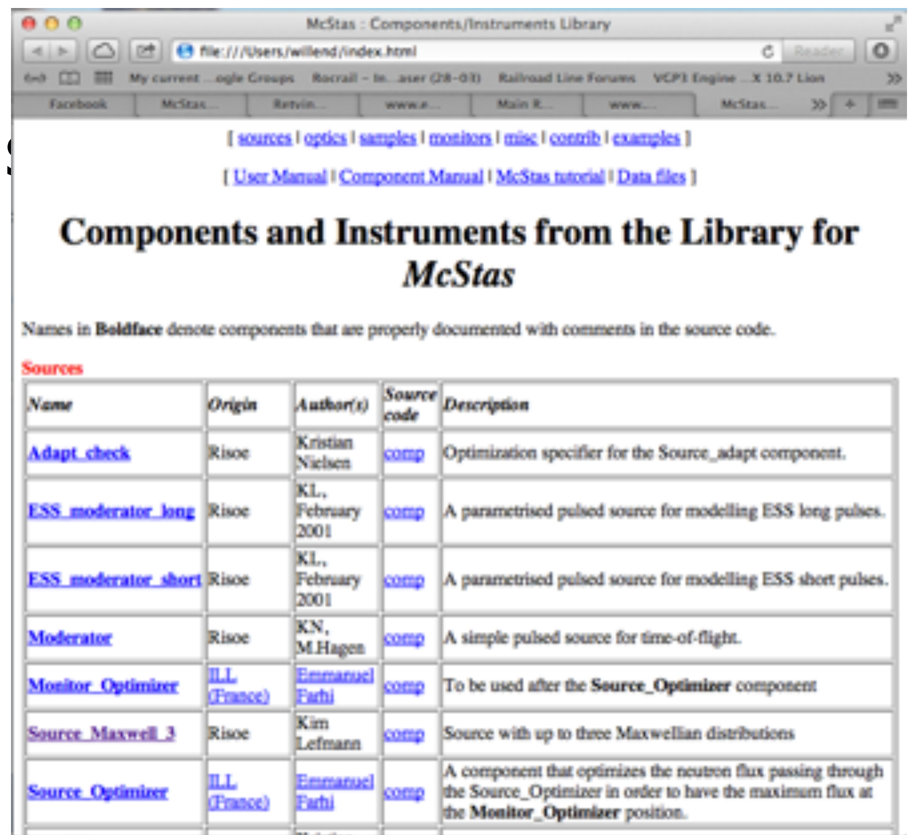


Risø–R–1538(rev.ed.)(EN)

Component Manual for the Neutron Ray-Tracing Package McStas, Version 1.12

Peter Kjær Willendrup, Erik Knudsen, Kim Lefmann and Emmanuel Farhi

Risø DTU, Roskilde, Denmark
May 2011



McStas : Components/Instruments Library

[sources | optics | samples | monitors | misc | contrib | examples]

[User Manual | Component Manual | McStas tutorial | Data files]

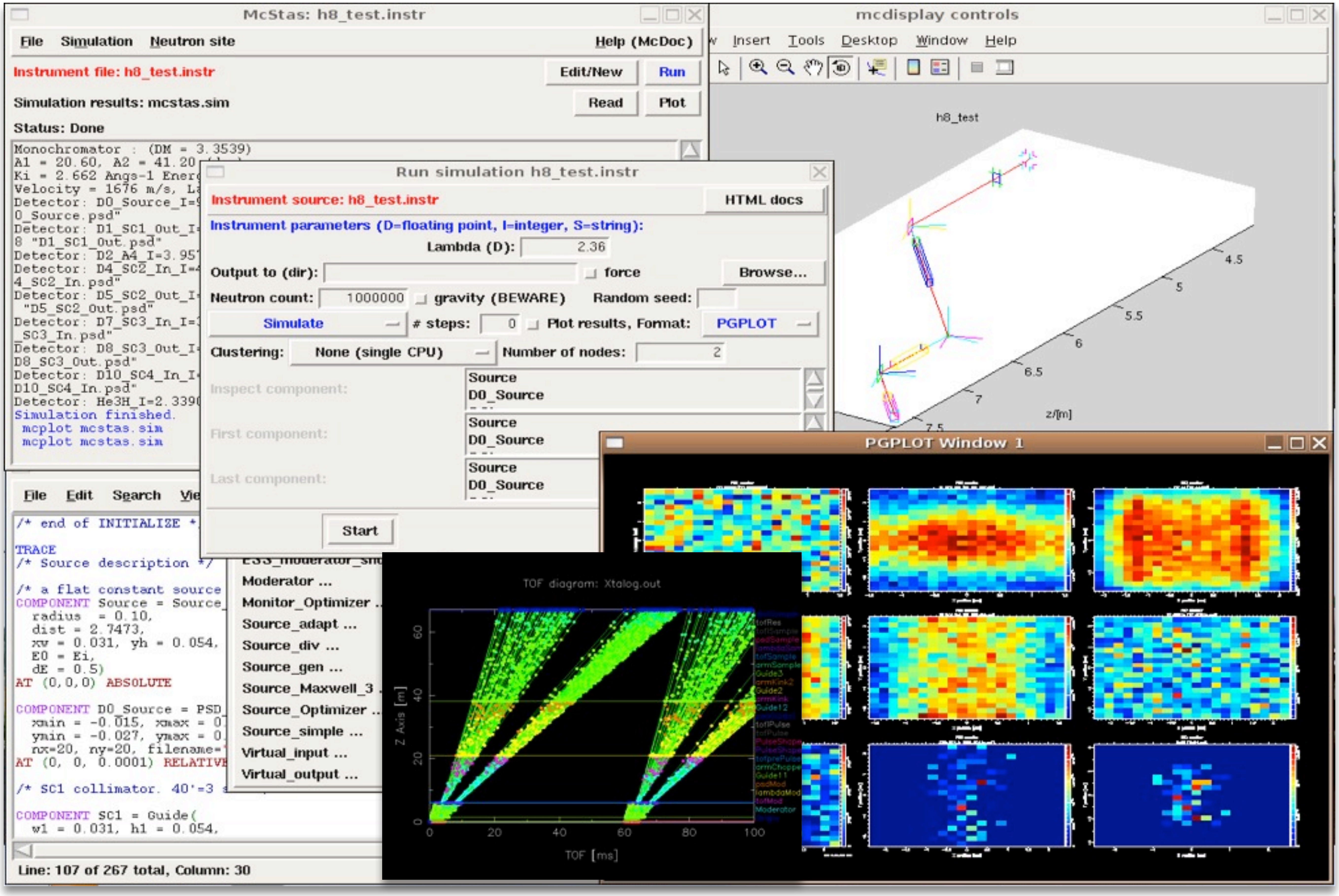
Components and Instruments from the Library for *McStas*

Names in **Boldface** denote components that are properly documented with comments in the source code.

Sources

Name	*Origin*	*Author(s)*	*Source code*	*Description*
Adapt_check	Risoe	Kristian Nielsen	comp	Optimization specifier for the Source_adapt component.
ESS_moderator_long	Risoe	KL, February 2001	comp	A parametrised pulsed source for modelling ESS long pulses.
ESS_moderator_short	Risoe	KL, February 2001	comp	A parametrised pulsed source for modelling ESS short pulses.
Moderator	Risoe	KN, M.Hagen	comp	A simple pulsed source for time-of-flight.
Monitor_Optimizer	ILL (France)	Emmanuel Farhi	comp	To be used after the **Source_Optimizer** component
Source_Maxwell_3	Risoe	Kim Lefmann	comp	Source with up to three Maxwellian distributions
Source_Optimizer	ILL (France)	Emmanuel Farhi	comp	A component that optimizes the neutron flux passing through the Source_Optimizer in order to have the maximum flux at the **Monitor_Optimizer** position.

Enough Talk!
Let’s see McStas run?



EUROPEAN SPALLATION SOURCE DTU PAUL SCHERRER INSTITUT PSI ILL NEUTRONS FOR SCIENCE McStas

People

- The success of the project is also about the people:
  - Present McStas team members





- K Lefmann E Farhi P Willendrup E Knudsen U Filges

  - Past McStas team members




- K Nielsen PO Åstrand K Lieutenant P Christiansen

Demo time?