

McStas Introduction

ESS DMSC McStas training June 20th-22nd 2018

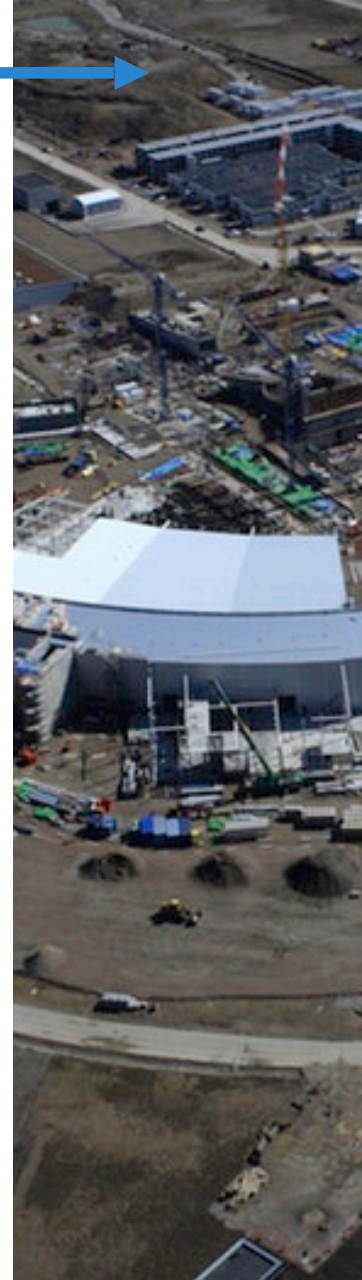


EUROPEAN
SPALLATION
SOURCE

McStas

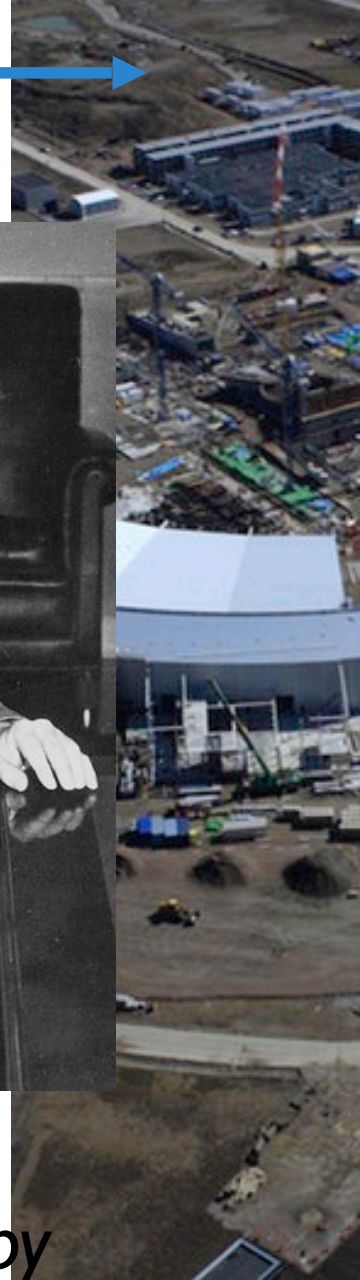


Agenda

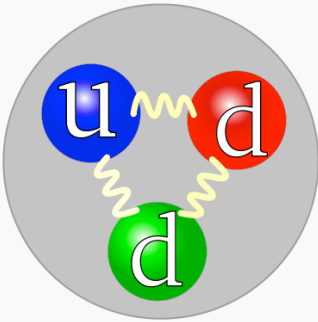


- | *A (very) brief introduction to neutrons, Monte Carlo & raytracing*
- | *Components of neutron instruments*
- | *How McStas works under the hood*
- | *Components and instruments*
- | *A demo*

McStas: Neutrons, Monte Carlo & ray-tracing



Neutron



The quark structure of the neutron. (The color assignment of individual quarks is not important; what is important is that all three colors are present.)

Classification	Baryon
Composition	1 up quark, 2 down quarks
Statistics	Fermionic
Interactions	Gravity, Weak, Strong, Electromagnetic
Symbol	n^0 , N^0
Antiparticle	Antineutron
Theorized	Ernest Rutherford ^{[1][2]} (1920)
Discovered	James Chadwick ^[1] (1932)
Mass	1.674 927 351(74) $\times 10^{-27}$ kg ^[3] 939.565 378(21) MeV/c ² ^[3] 1.008 664 916 00(43) u ^[3]
Mean lifetime	881.5(15) s (free)
Electric charge	0 e 0 C
Electric dipole moment	$< 2.9 \times 10^{-26}$ e·cm
Electric polarizability	1.16(15) $\times 10^{-3}$ fm ³



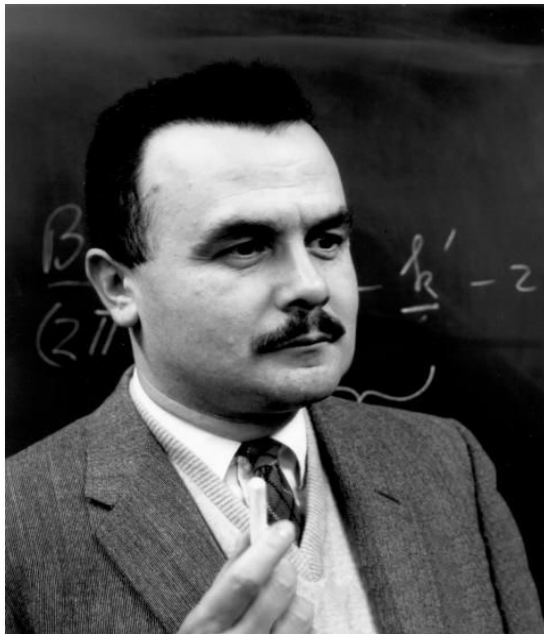
Subatomic particle discovered by
Sir James Chadwick, 1932

Nobel Prize in Physics, 1994



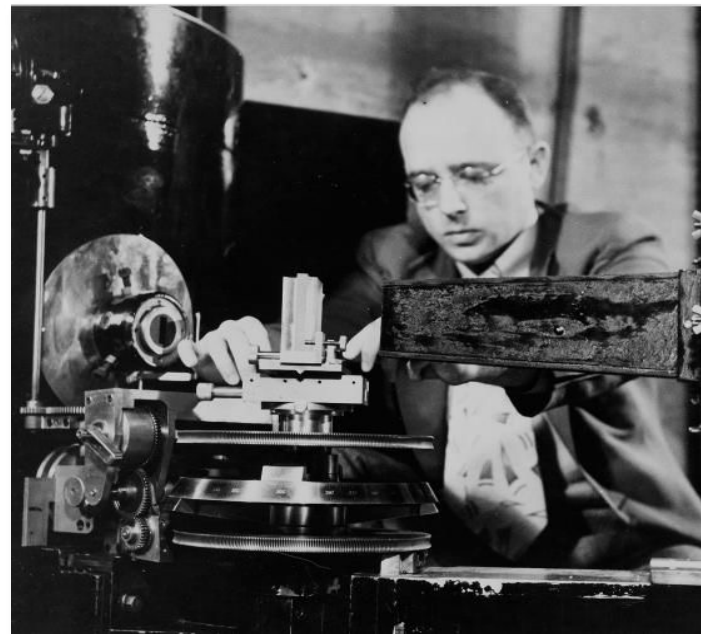
Awarded for “pioneering contributions to the development of neutron scattering techniques for studies of condensed matter”

Bertram N. Brockhouse

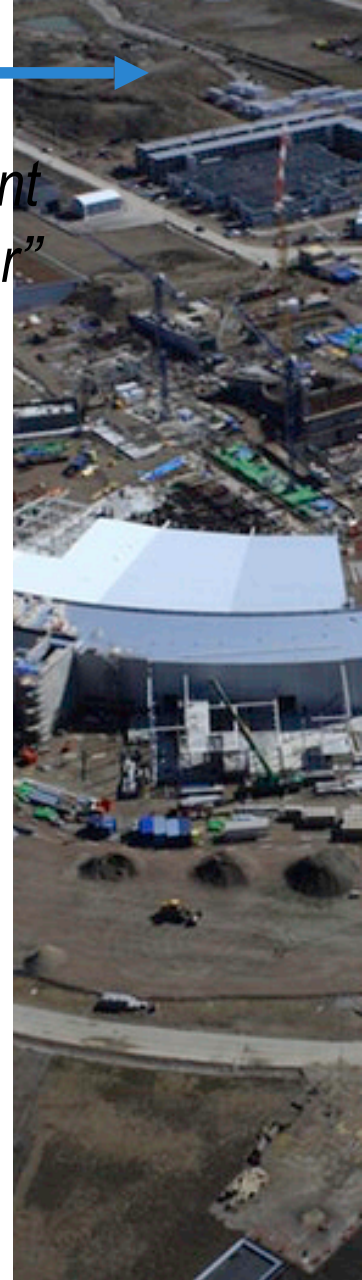


Neutron spectroscopy

Clifford G. Shull



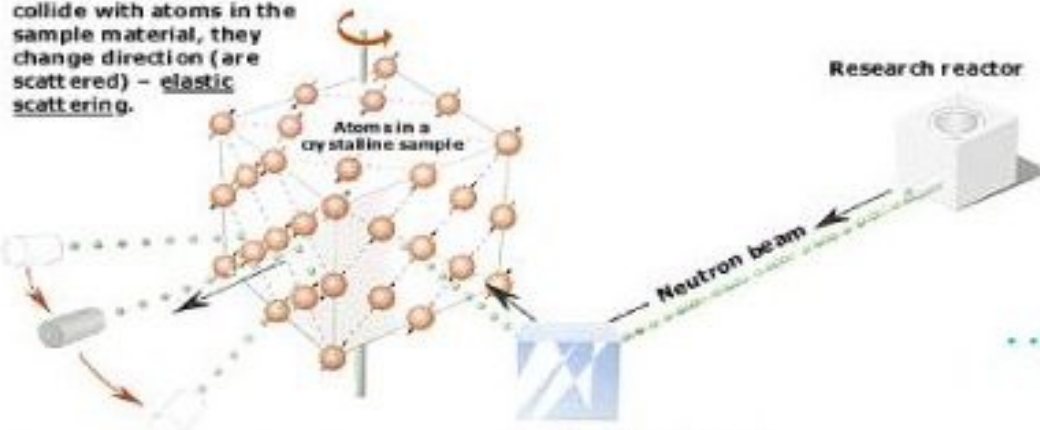
Neutron diffraction technique



1994 Nobel prize to Shull & Brockhouse

Neutrons show where the atoms are....

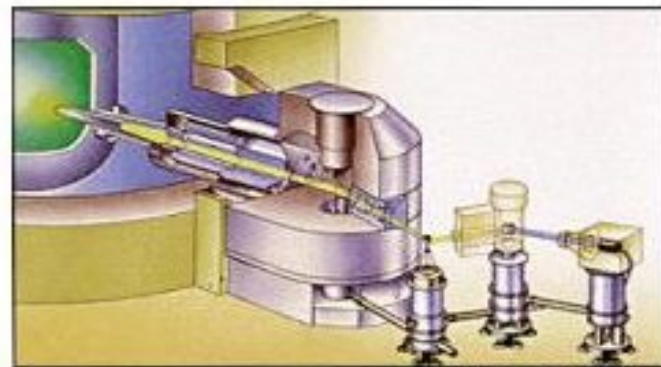
When the neutrons collide with atoms in the sample material, they change direction (are scattered) – elastic scattering.



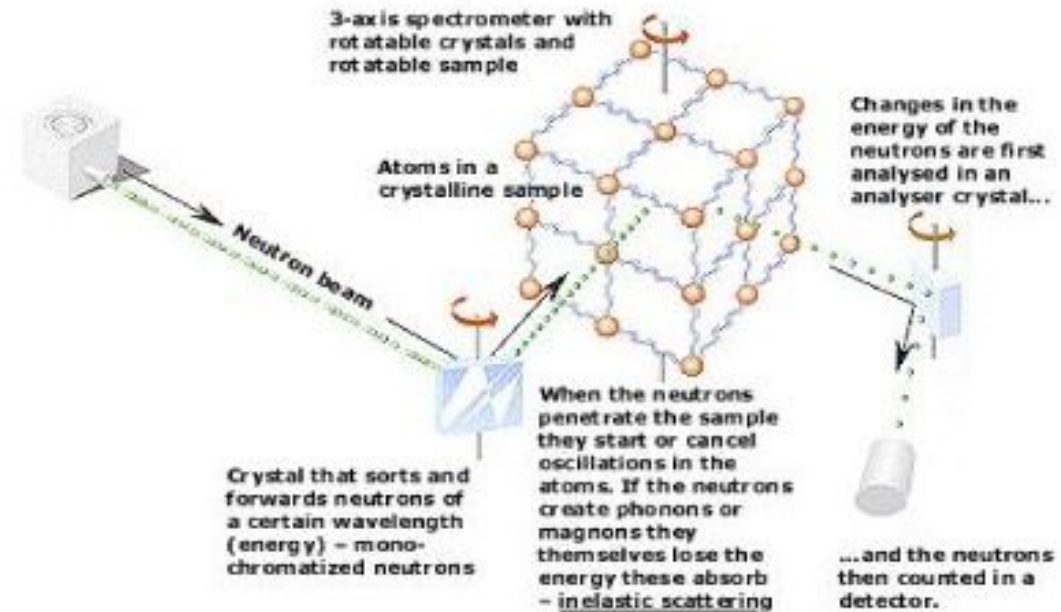
Detectors record the directions of the neutrons and a diffraction pattern is obtained.

The pattern shows the positions of the atoms relative to one another.

...and what the atoms do.

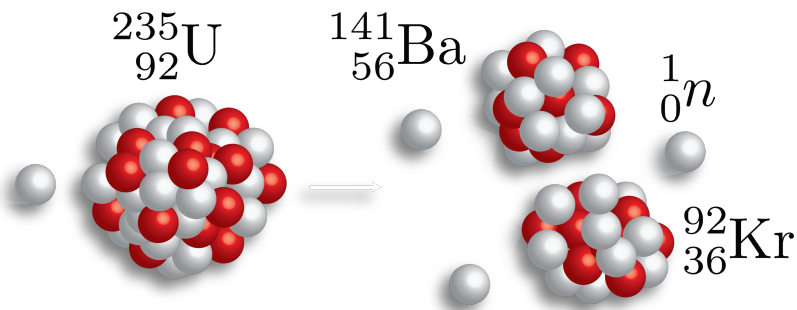


3-axis spectrometer

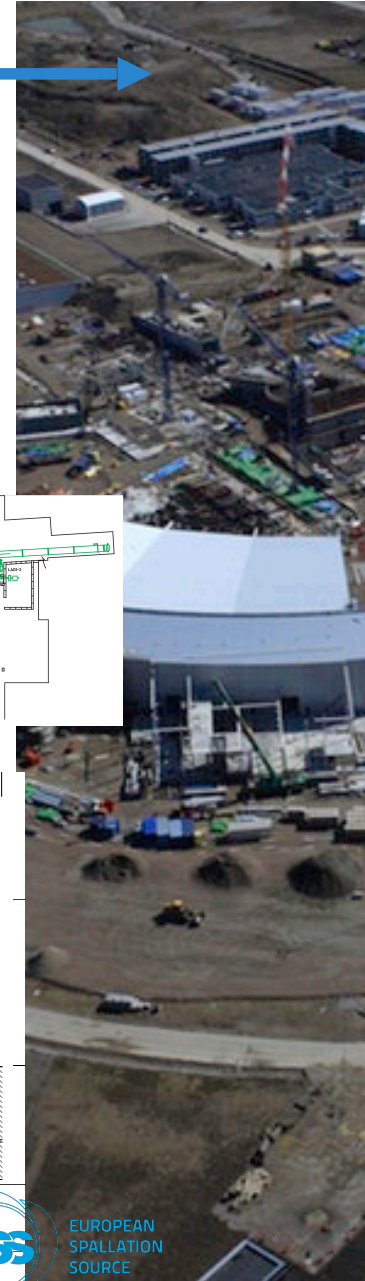
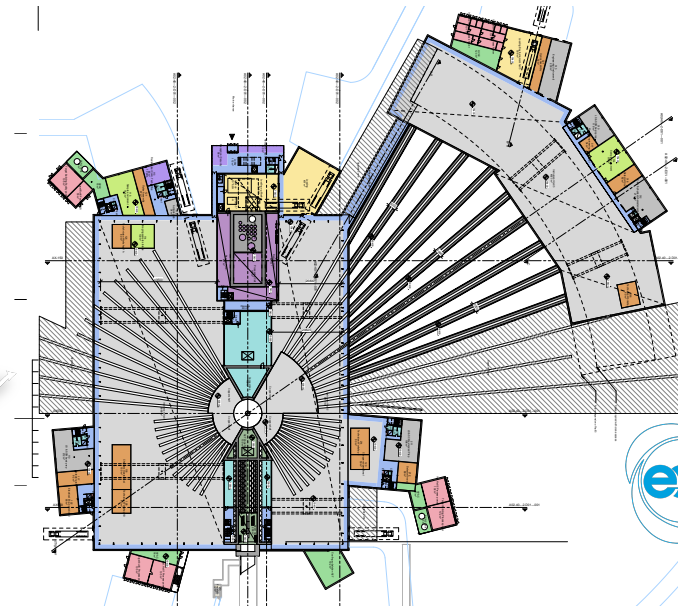
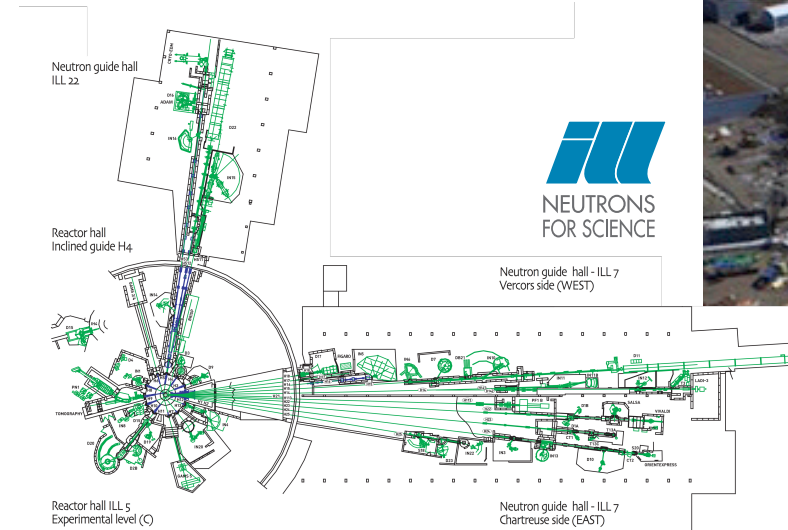
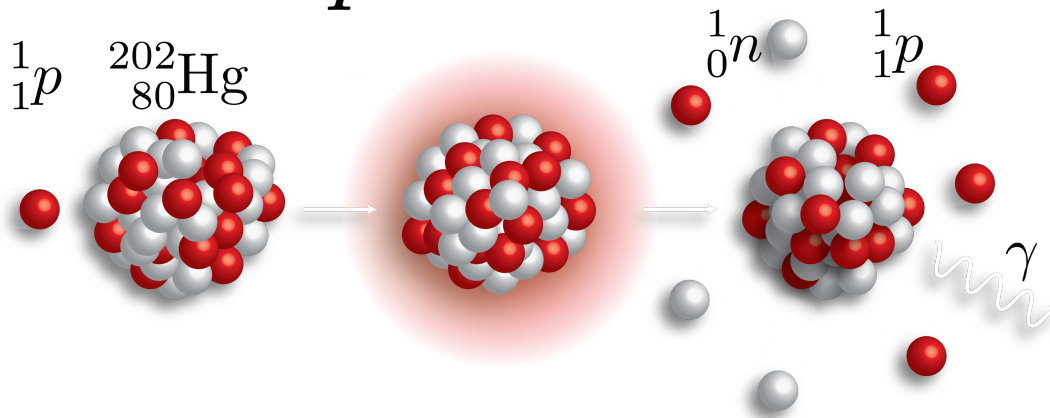


Neutron facilities

Fission



Spallation



Origin of Monte Carlo methods

Used by Nature since ... (a long time) : diversity of Life



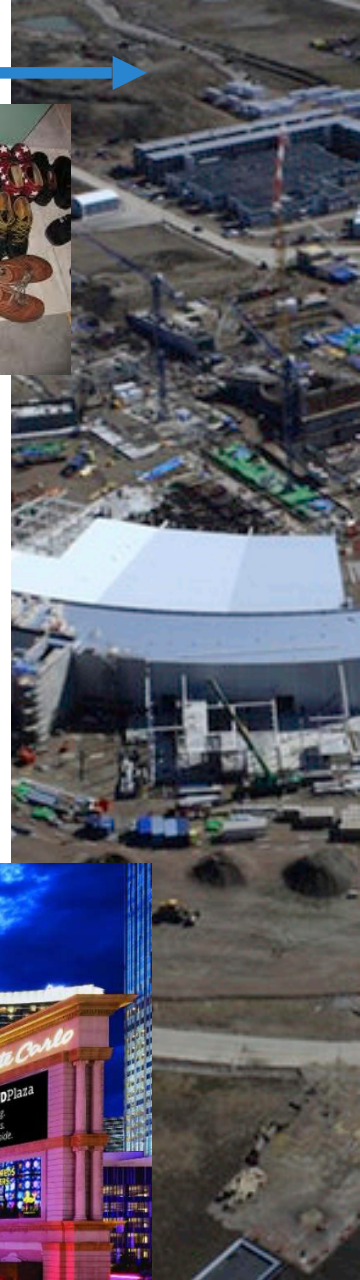
First application using computers:

Metropolis, Ulam and Von Neumann at Los Alamos, 1943

Neutron Scattering and Absorption in U and Pu, Origin of MCNP

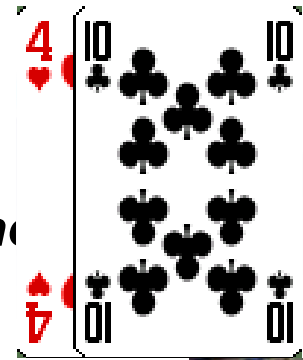
Name:

Monte Carlo casino, a random generator (Ulam's father played poker)



What are Monte Carlo methods ?

- | Use random generators (play poker)
- | Explore a complex and large phase space (many parameters)
- | Integrates microscopic random events into measurable quantities not usual regular sampling integration



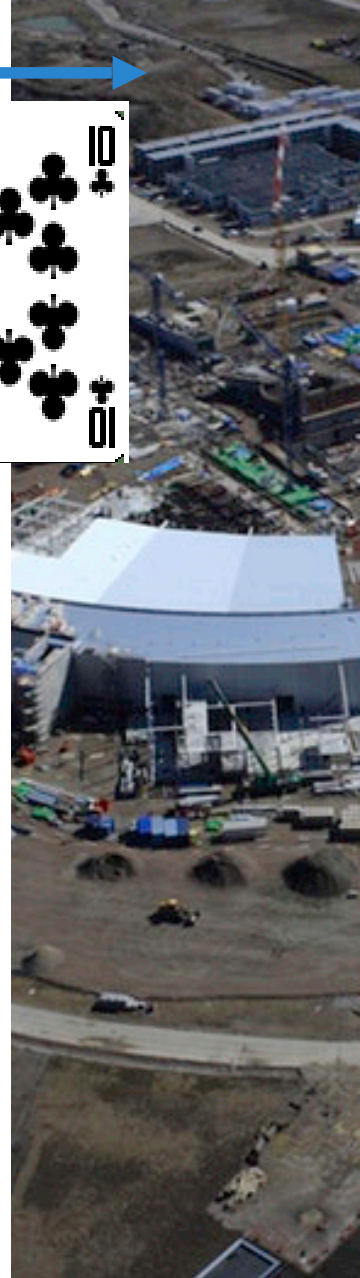
- | Metropolis algorithm: model energy gap E as a probability

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1, a < u_i < b}^n f(u_i) = \frac{1}{b-a} \int_a^b f(u) du$$

- | Integrals converge faster than any other method (for $d > 3$) when using enough independent events (central limit theorem)

$$p \propto e^{-E/kT}$$

- | F. James, Rep. Prog. Phys., Vol. **43** (1980) 1145.



How to implement Monte Carlo methods ?

Good random generator:

from thermal electronic noise (hardware)

or quasi-random generators \Rightarrow quasi-Monte-Carlo

We encounter a probability $0 < p < 1$.

Crude Monte-Carlo (yes/no choice):

We shoot n events $\xi \in [0, 1]$

We keep events that satisfy $\xi < p$

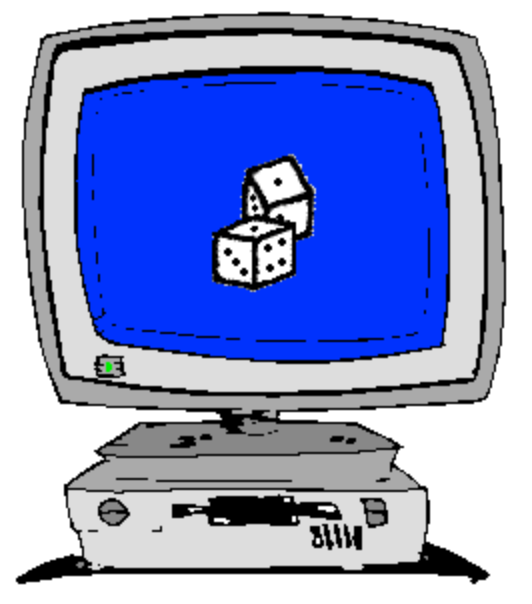
np events \rightarrow low statistics

Importance sampling (fuzzy choice – event weighting):

Keep n events, no more random number...

But associate a **weight** p to each of them (we set $\xi = p$)

Retain statistical accuracy ($1/\div n$)



When to use Monte Carlo methods

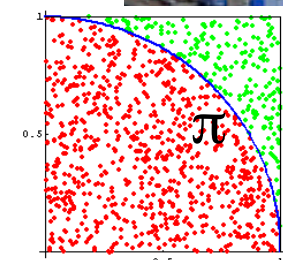
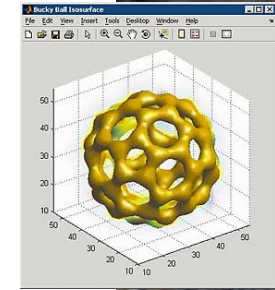
Dimensionality of phase space must be large ($d > 5$)

Overall complexity is beyond reasonable analytical methods

Each event can be computed easily and independently MC is the 'lazy guy' method – think microscopic

Examples:

- Estimate π from a circle/square (“Buffon needle”)
- Area under/inside a curve/volume (integration)
- Molecular Dynamics
- spin-system phase transitions (Ising model)
- nuclear reactions
- ray-tracing (light, particles)



Number of points for which
 $\{ x^2+y^2 \leq 1, (x,y) \in [0,1] \}$
 Ratio circle/square $\rightarrow \pi/4$

Examples of Monte Carlo programs

**Each time physics takes place (scattering, absorption, ...)
random choices are made.**

Light ray-tracing: PoV-RAY and others ...

Nuclear reactor simulations (neutron transport):

MCNP, Tripoli, GEANT4, FLUKA

Neutron Ray-Tracing propagation:

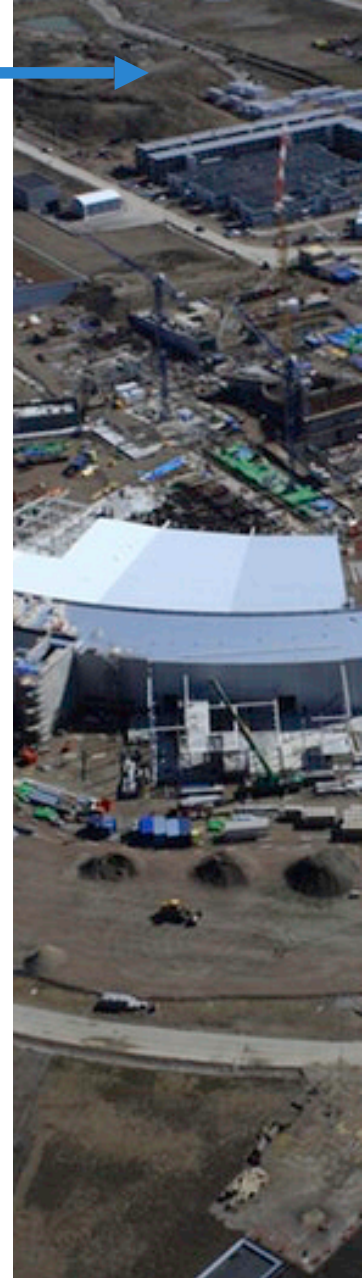
McStas <www.mcstas.org>, Vitess, Restrax, NISP, IDEAS

Neutrons are described as $(\mathbf{r}, \mathbf{v}, \mathbf{s}, t)$, and are transported along instrument models.

Propagation simply uses Newton rules, incl. gravitation.

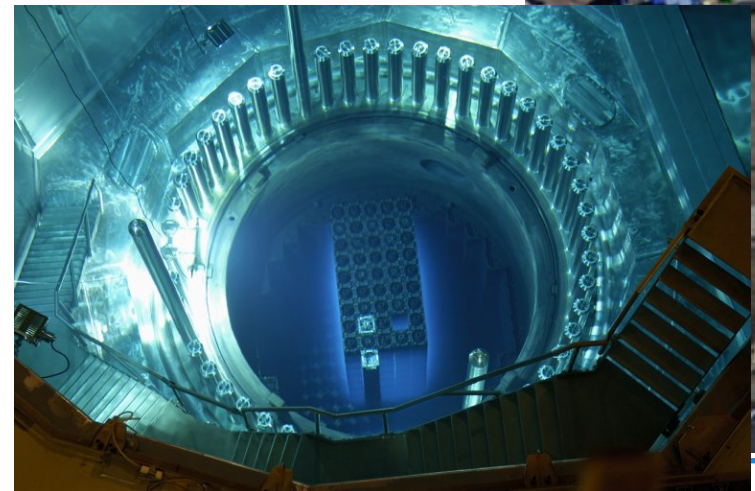
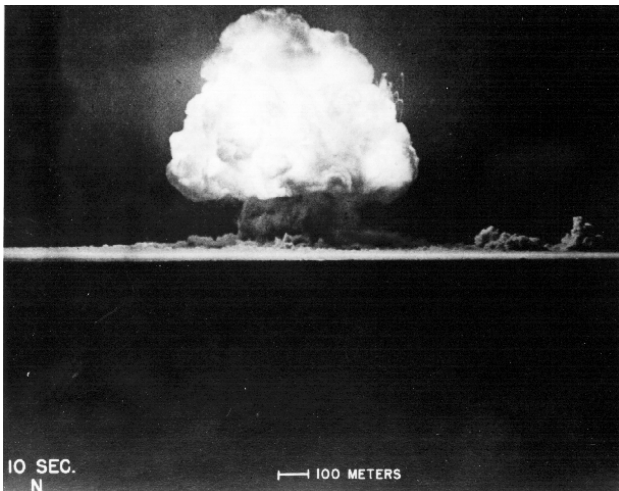
X-ray tracing

Shadow, McXtrace, RAY, ...

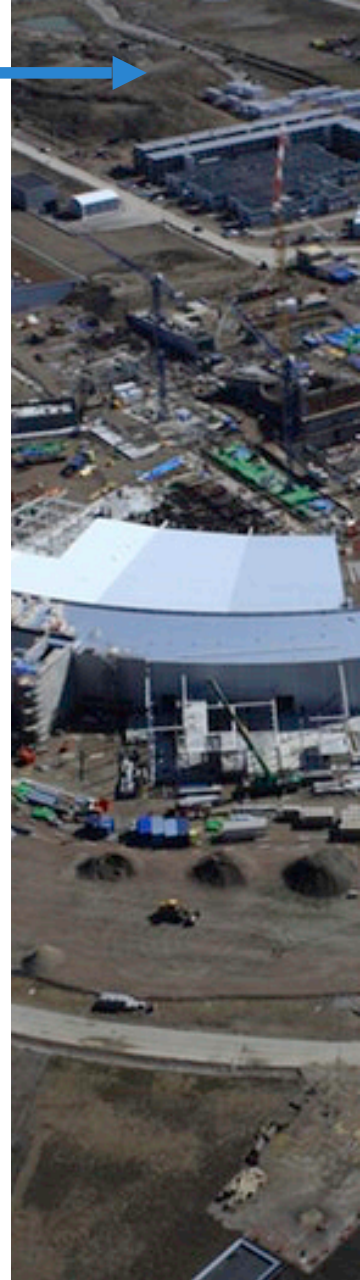
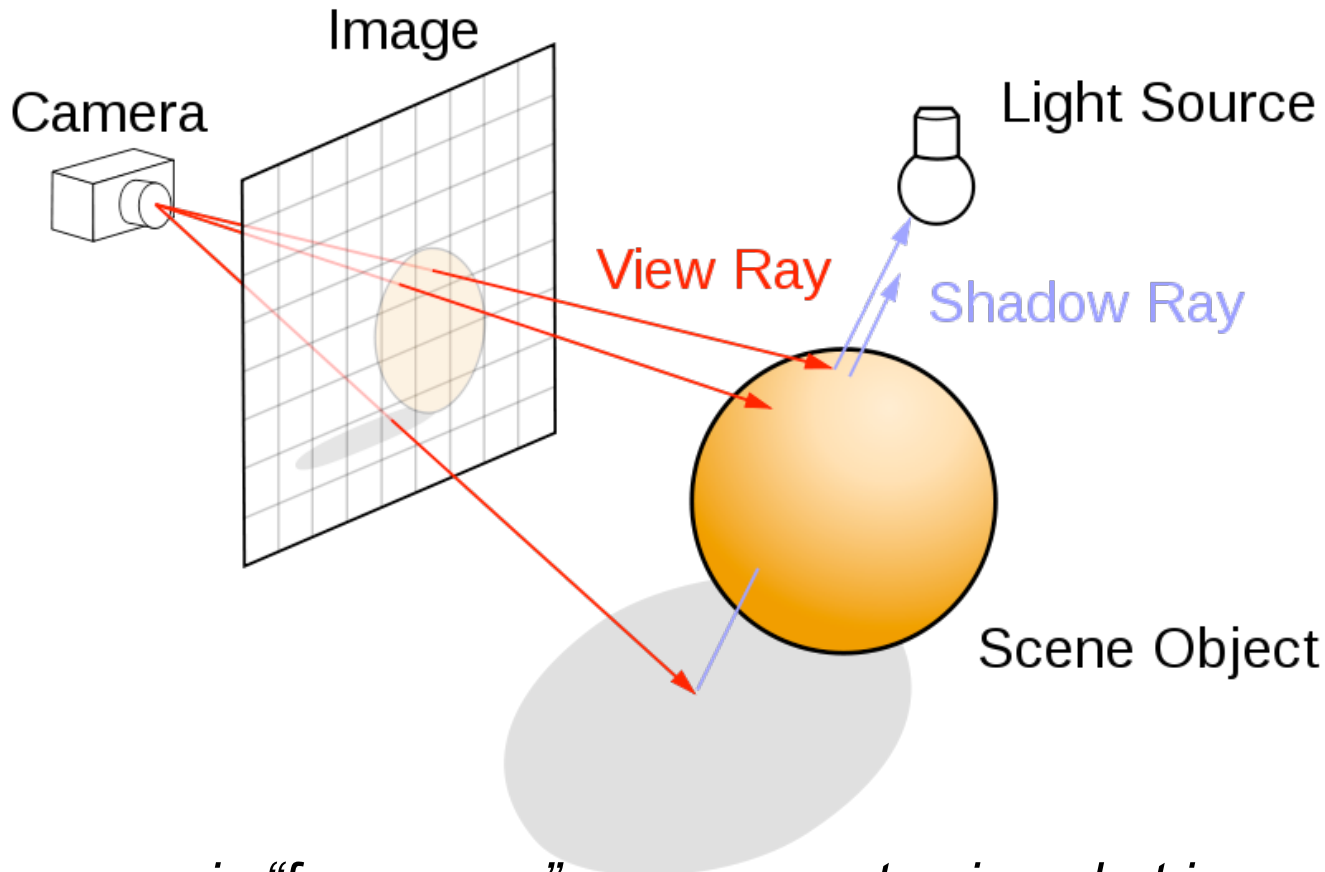


Monte Carlo techniques

- Los Alamos has since then developed and perfected many different monte carlo codes leading to what is today known as the codes MCNP5 and MCNPX
- State of the art is MCNPX (or soon the merged MCNP6 code) that features numerous (even exotic) particles
- MCNP was originally Monte Carlo Neutron Photon, later N-Particle
- Mainly used for high-energy particle descriptions in weapons, power reactors and routinely used for estimating dose rates and needed shielding
- Does not to date handle coherent scattering of neutrons due to the focus on high energies



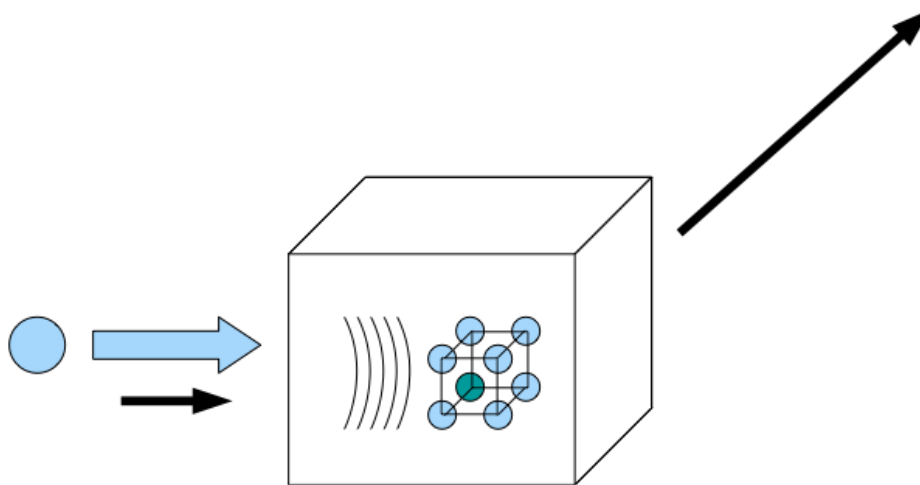
Ray-tracing methods



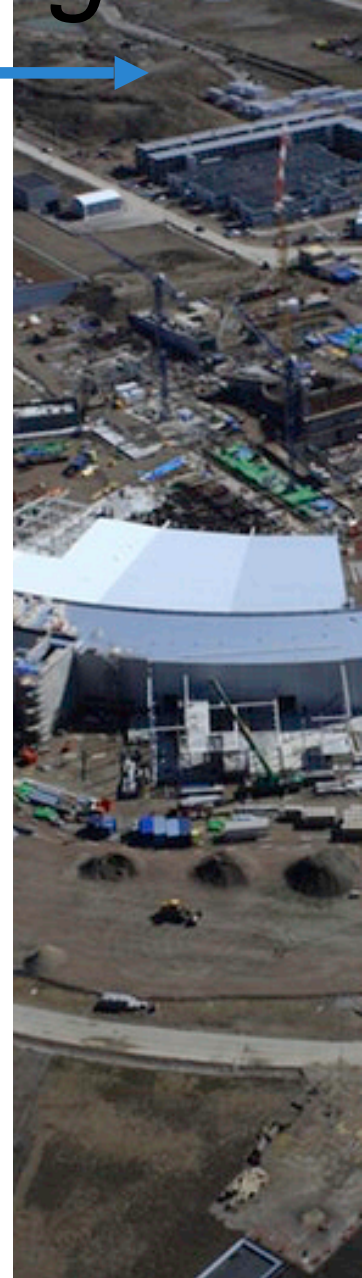
- When neutrons move in “free space”, we use ray-tracing - but in most cases in direction source -> detector
- Of course parabolas rather than straight lines are used to implement gravity

Elements of Monte-Carlo raytracing

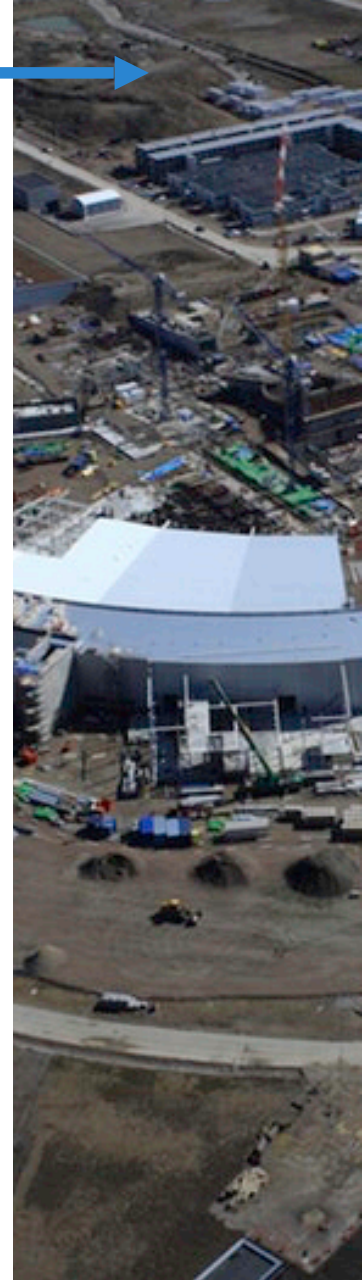
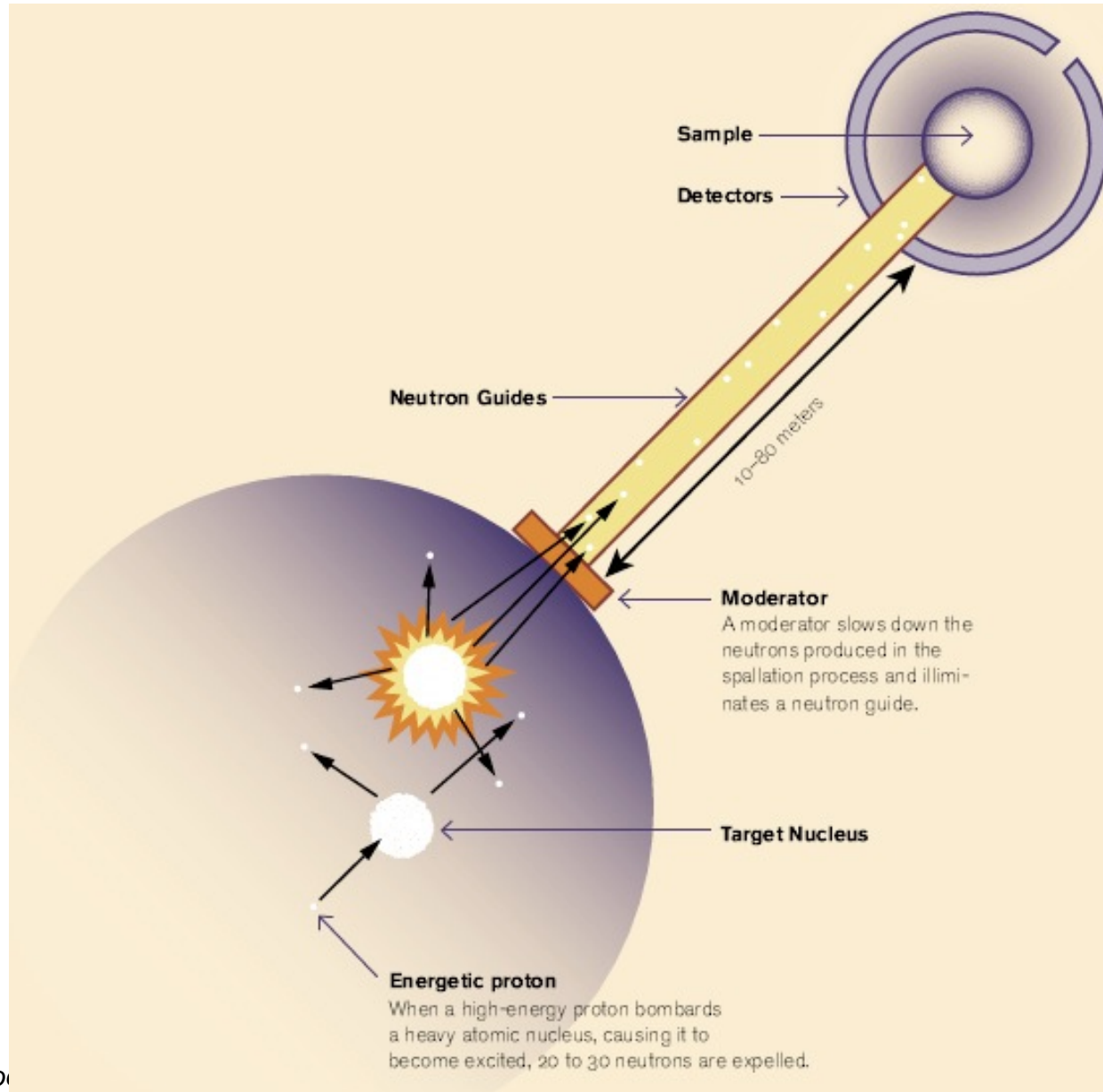
- | Instrument Monte Carlo methods implement coherent scattering effects
- | Uses deterministic propagation where this can be done
- | Uses Monte Carlo sampling of “complicated” distributions and stochastic processes and multiple outcomes with known probabilities are involved
- | - I.e. inside scattering matter
- | Uses the particle-wave duality of the neutron to switch back and forward between deterministic ray tracing and Monte Carlo approach



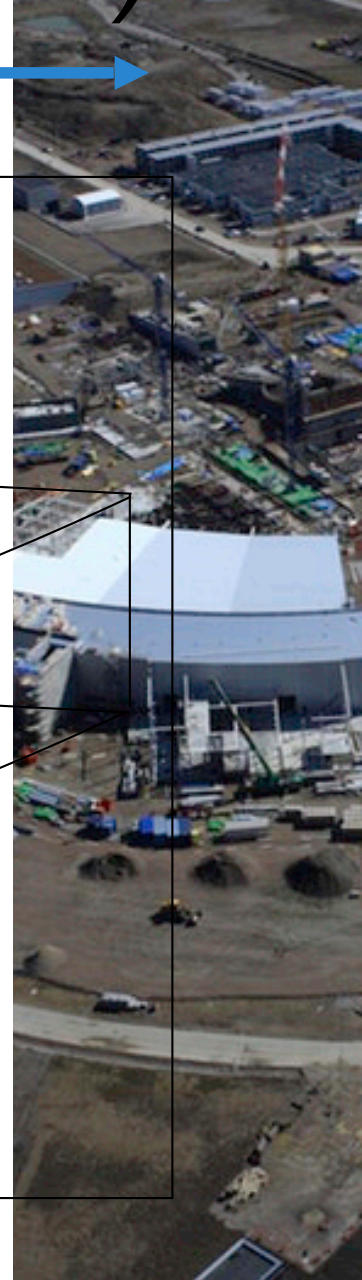
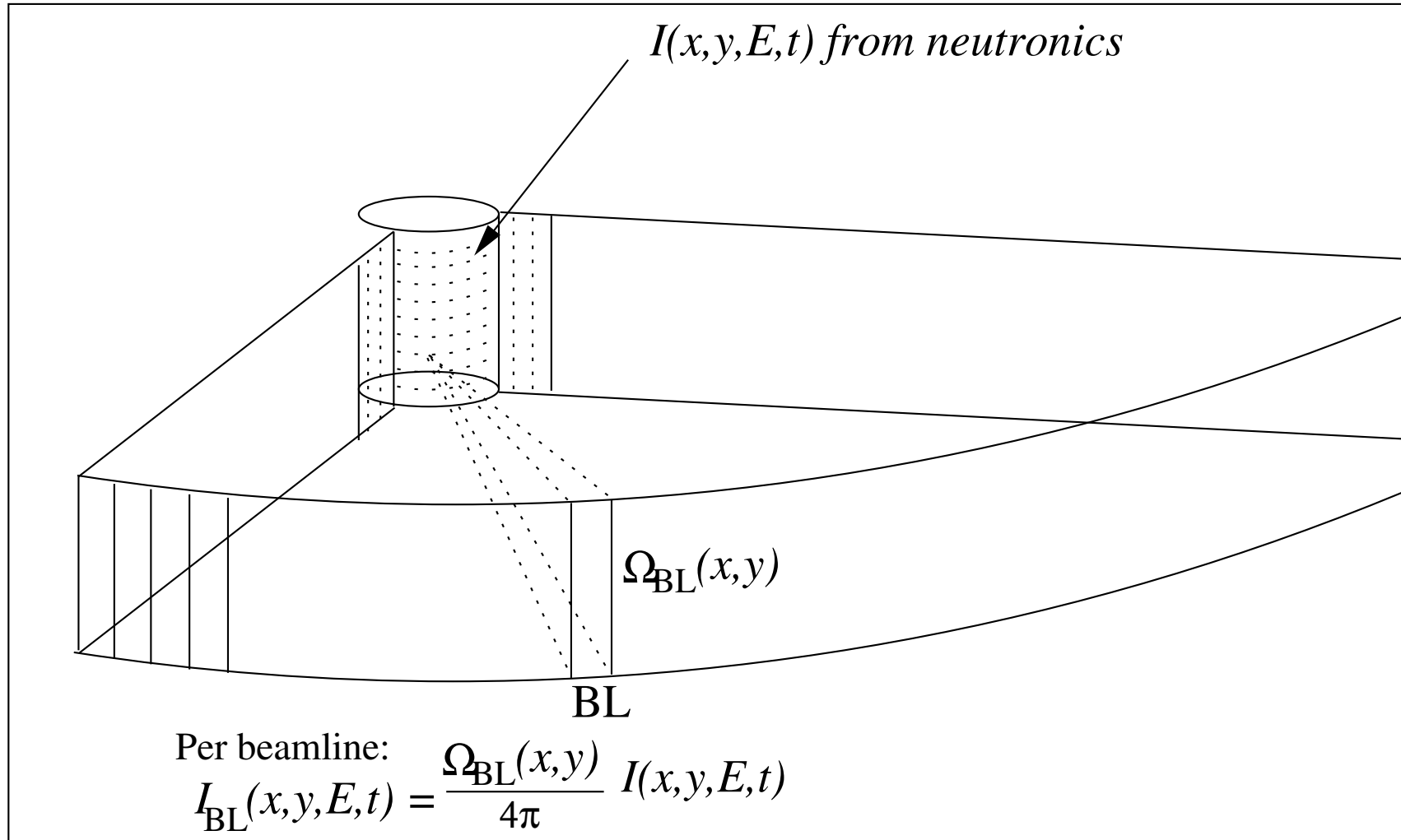
- | Result: A realistic and efficient transport of neutrons in the thermal and cold range



Components of neutron instruments



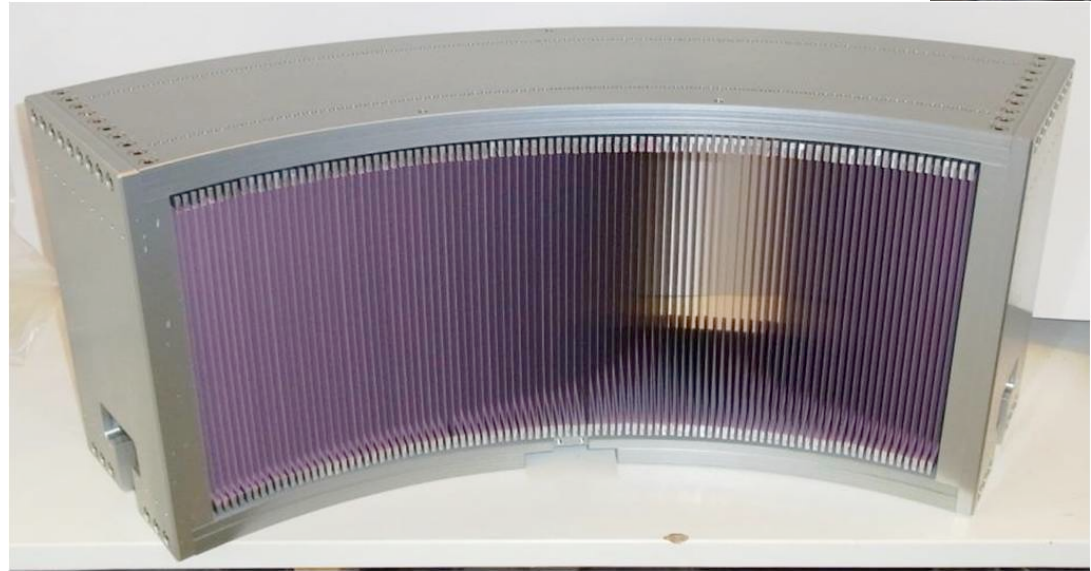
Moderators... (Where McStas starts)



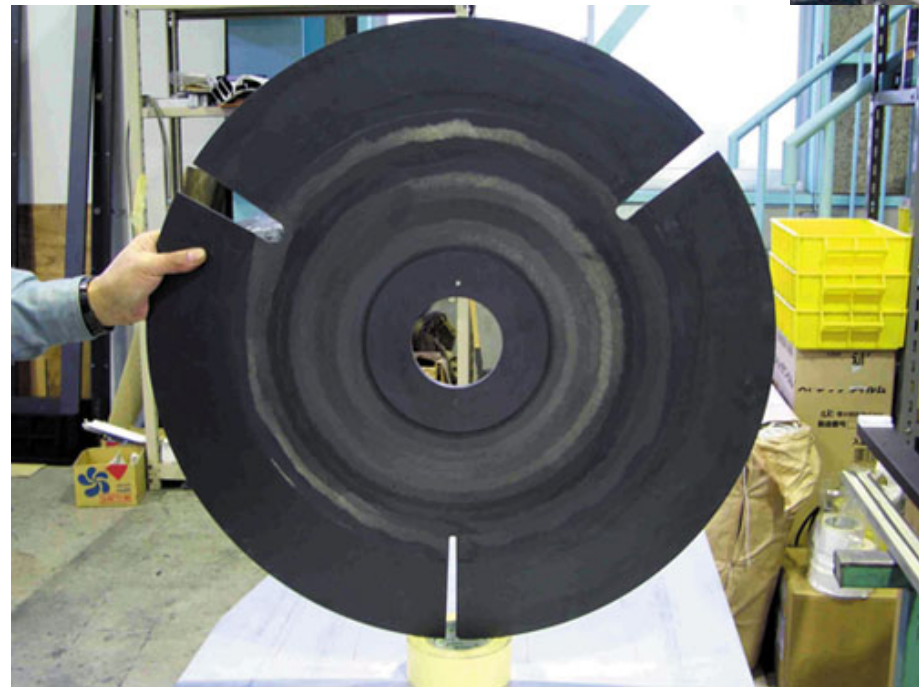
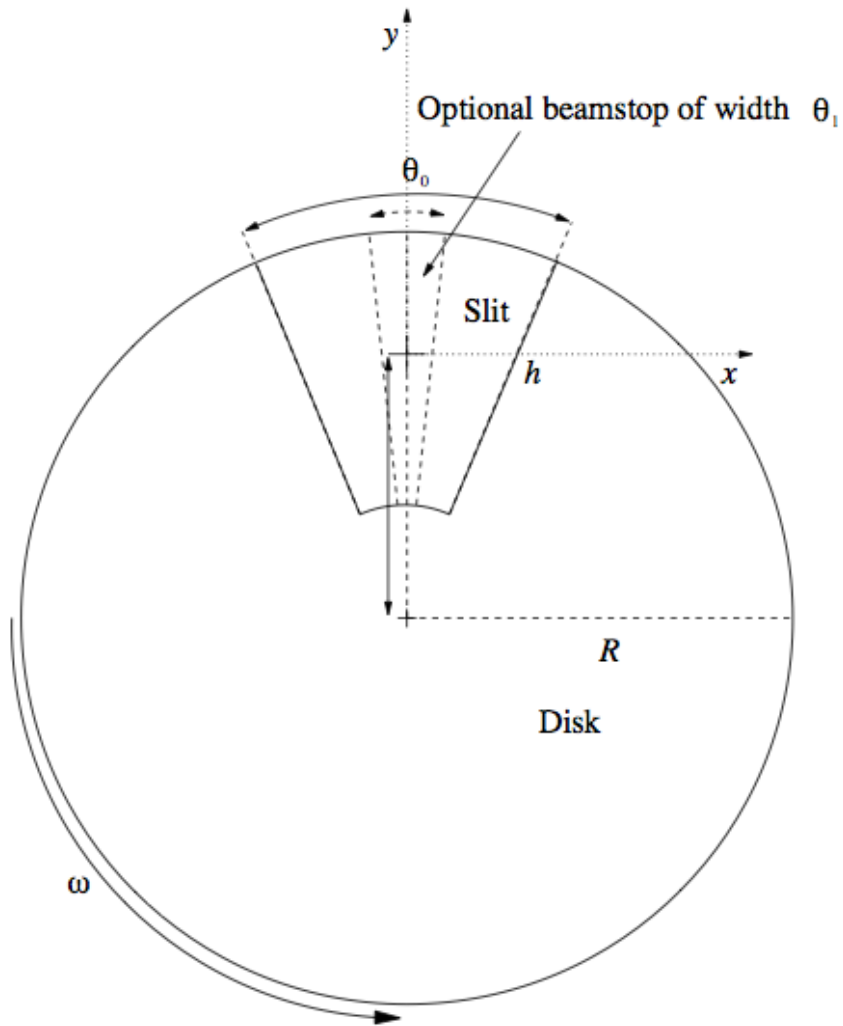
Neutron guides



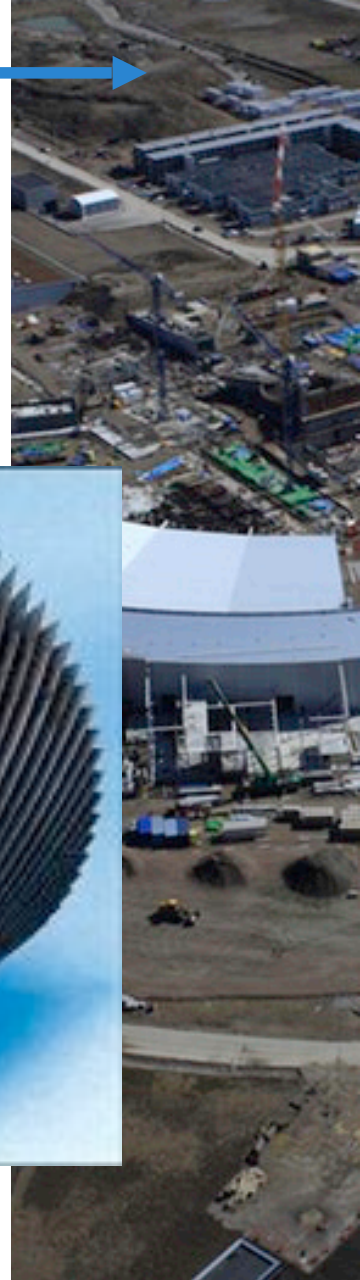
Collimators & slits



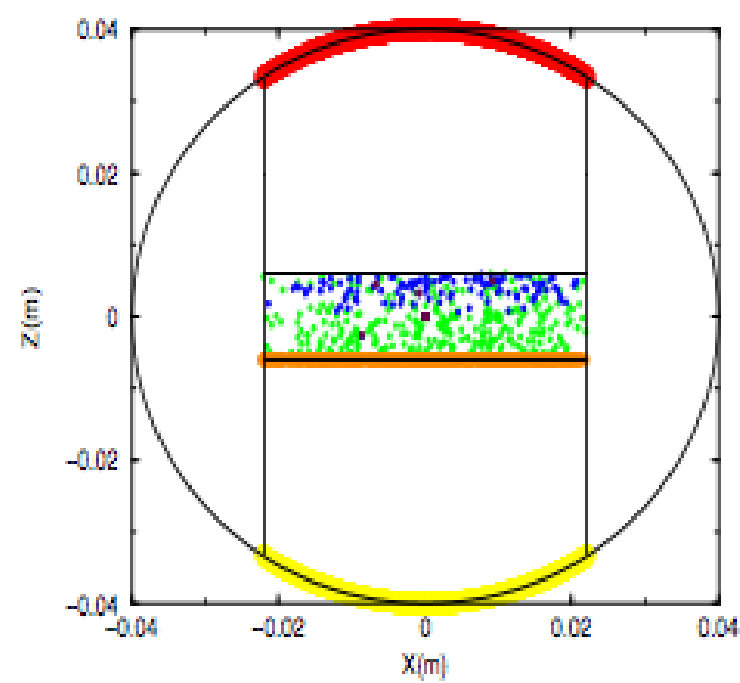
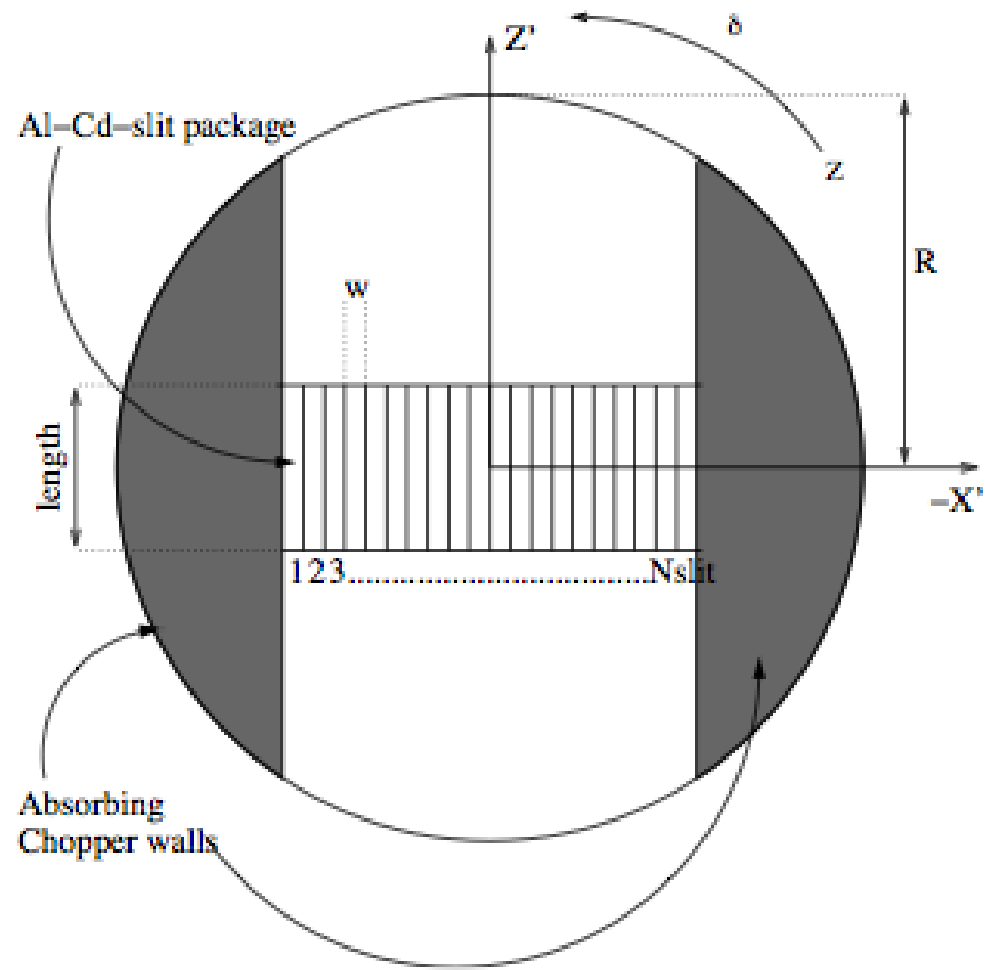
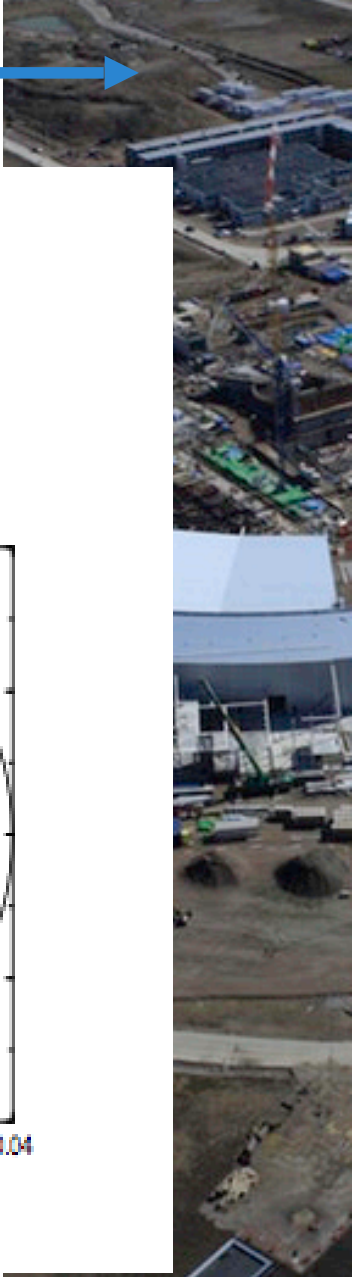
Disk Choppers



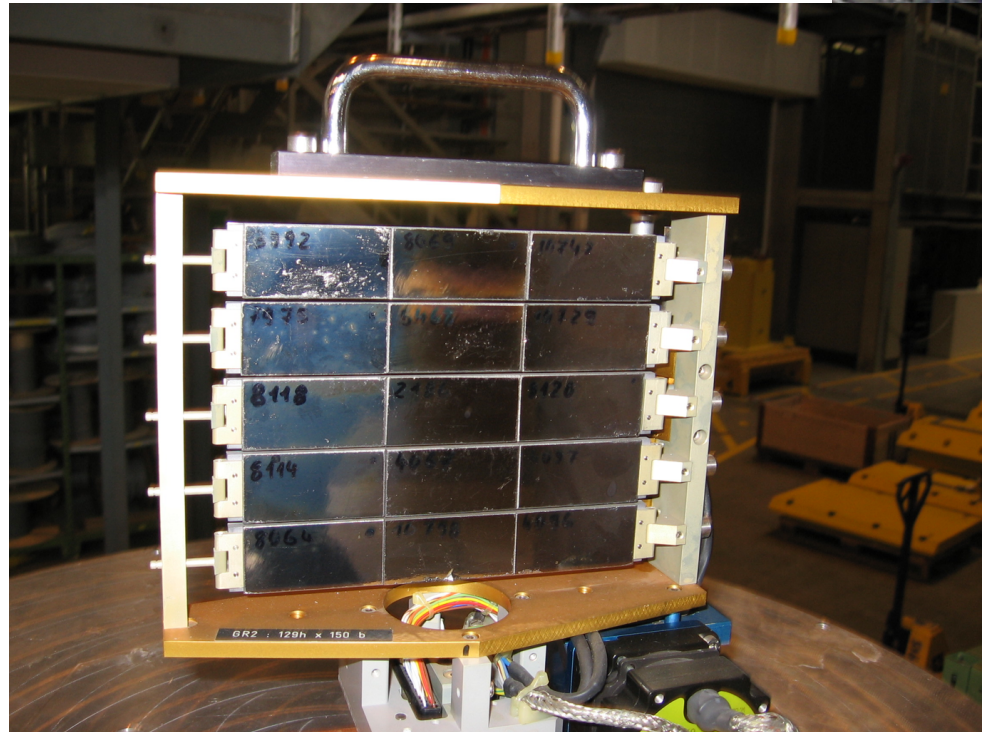
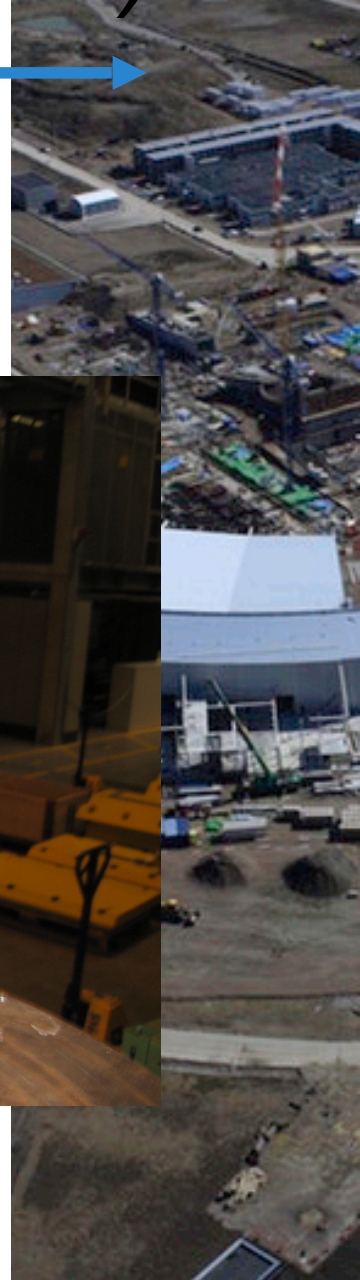
Velocity selector



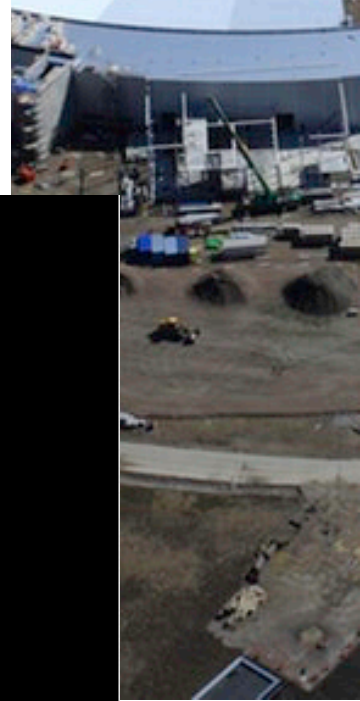
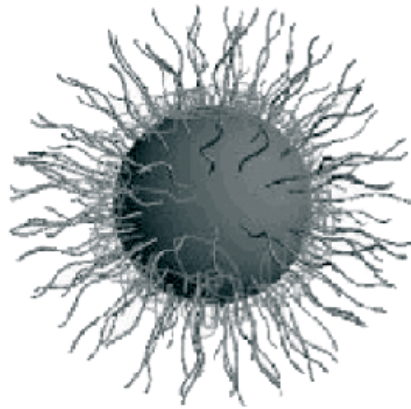
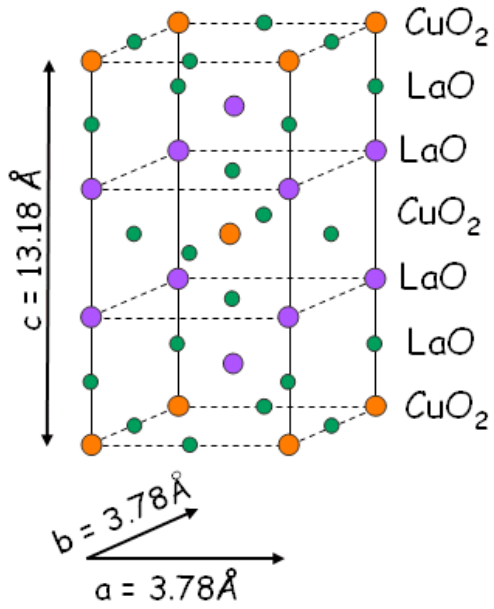
Fermi Choppers



Crystal monochromators (and analyzers)

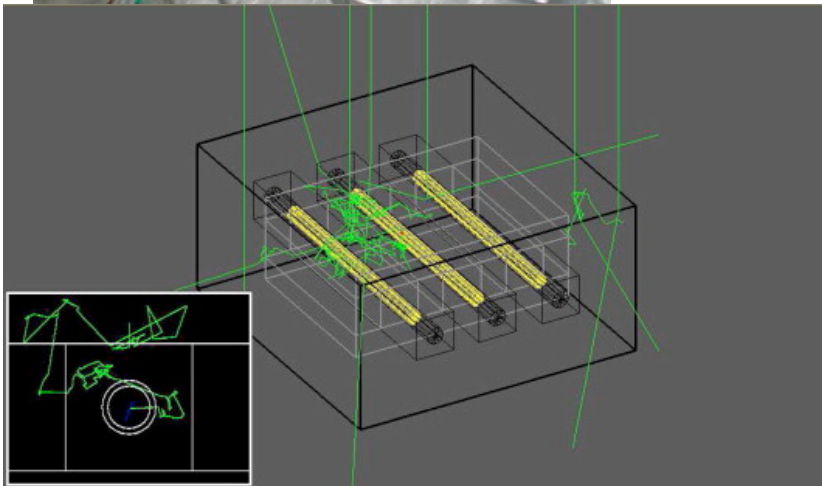
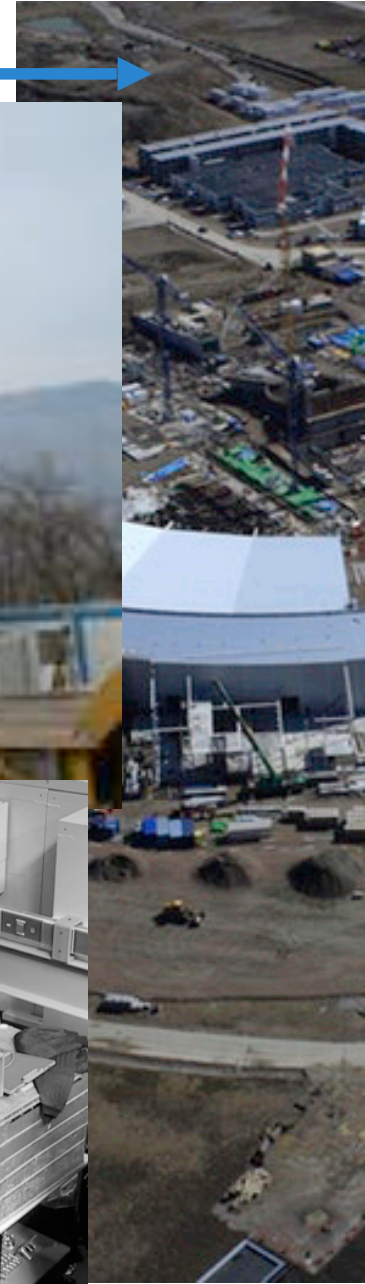


Samples studied...



Detectors

X

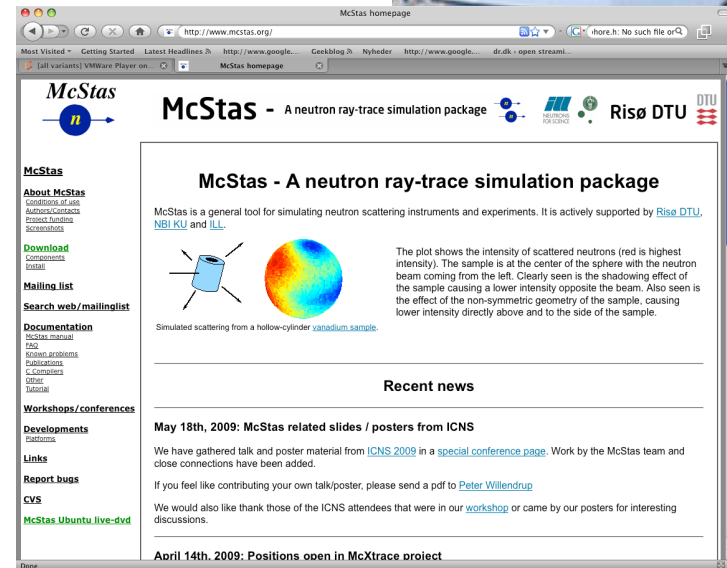


McStas Introduction

- 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, ESS DMSC
- V. 1.0 by K Nielsen & K Lefmann (1998) RISØ
- Currently 2.5+1 people full time plus students



GNU GPL
license
Open Source

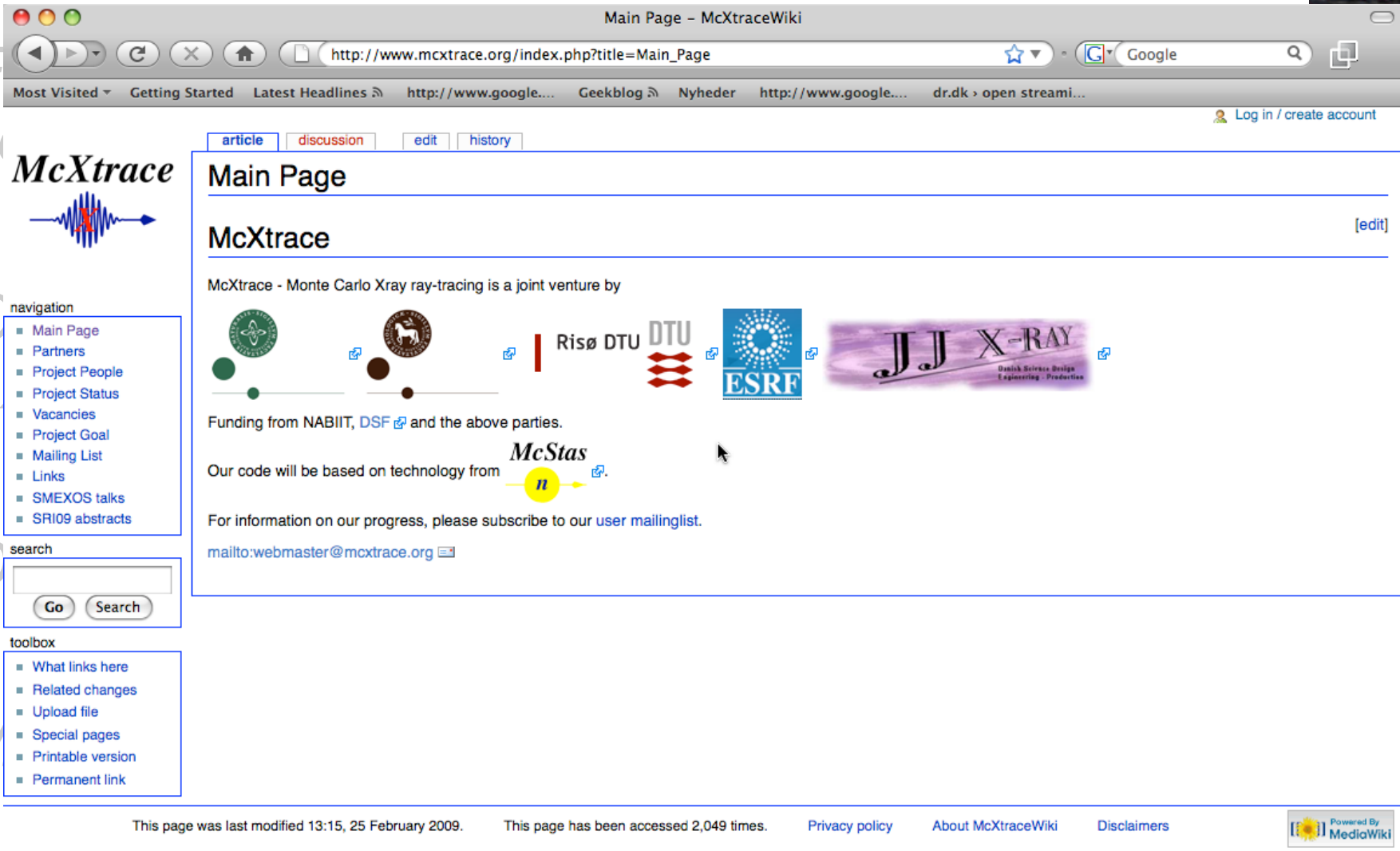


Project website at
<http://www.mcstas.org>

mcstas-users@mcstas.org mailinglist

McXtrace - since jan 2009 similar for X-rays

McStas Introduction



The screenshot shows a web browser window titled "Main Page - McXtraceWiki". The address bar contains the URL "http://www.mcxtrace.org/index.php?title=Main_Page". The page content includes a navigation menu on the left with items like "Main Page", "Partners", "Project People", "Project Status", "Vacancies", "Project Goal", "Mailing List", "Links", "SMEXOS talks", and "SRI09 abstracts". The main content area features the "McXtrace" logo, a list of partner logos (including NABIIT, DSF, Risø DTU, DTU, and ESRF), and text stating "McXtrace - Monte Carlo Xray ray-tracing is a joint venture by" and "Funding from NABIIT, DSF and the above parties." It also mentions "Our code will be based on technology from McStas" and provides a mailing list subscription link: "mailto:webmaster@mcxtrace.org". The footer contains the text "This page was last modified 13:15, 25 February 2009. This page has been accessed 2,049 times." and links for "Privacy policy", "About McXtraceWiki", and "Disclaimers". A "Powered By MediaWiki" logo is also present.

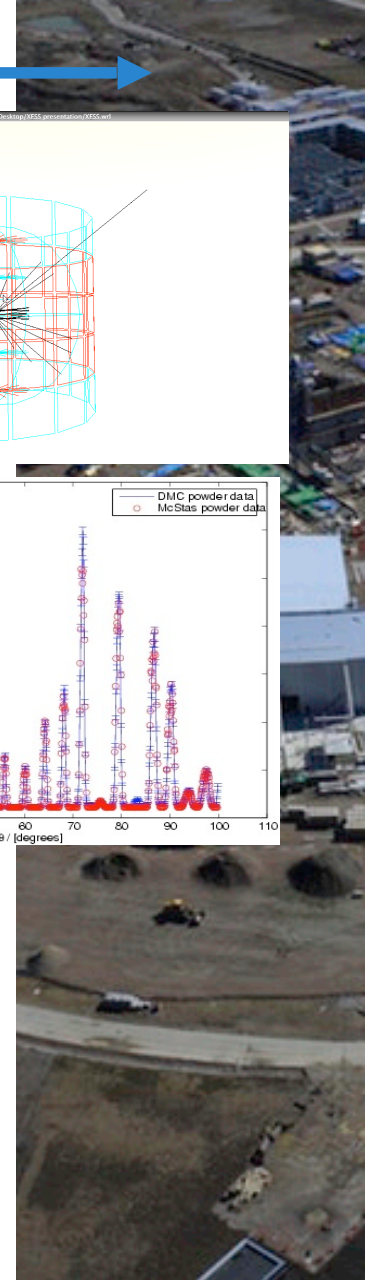
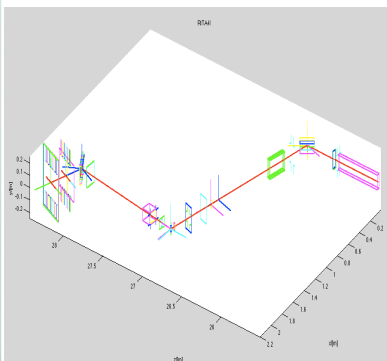
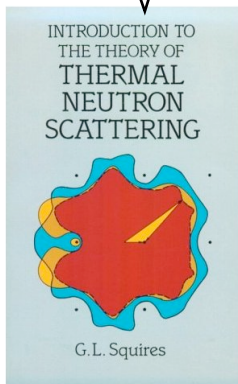
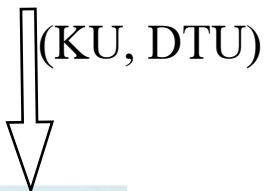
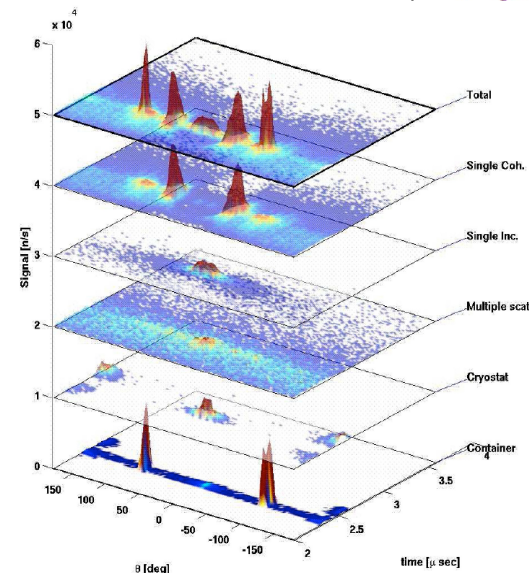
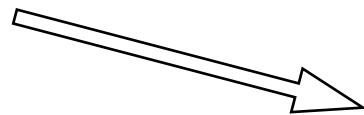
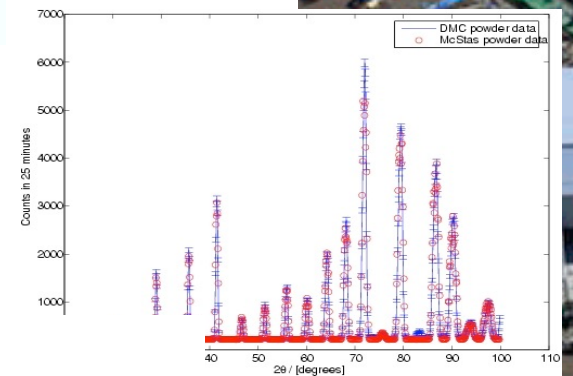
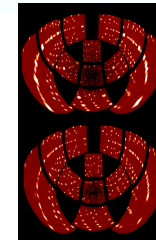
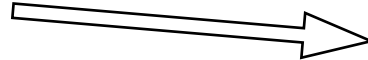
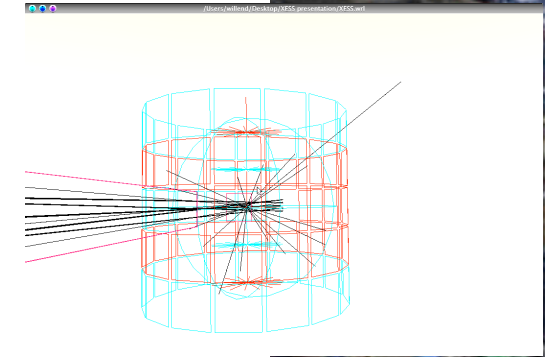
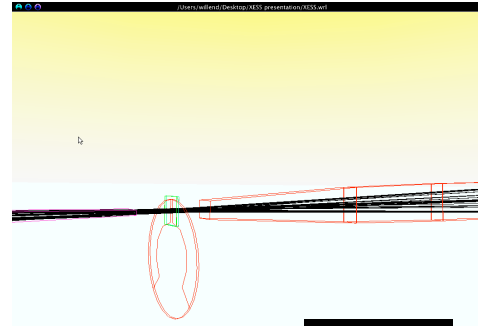
- Synergy, knowledge transfer, shared infrastructure

Used in many places

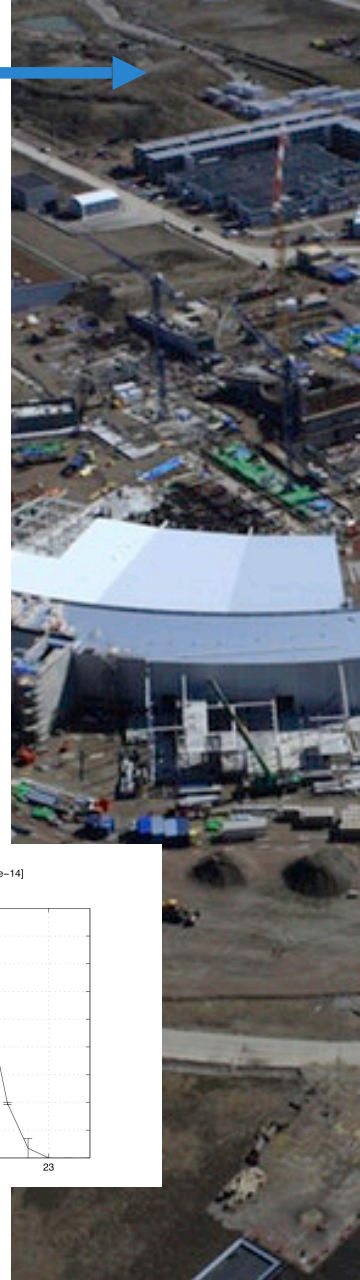


What is McStas used for?

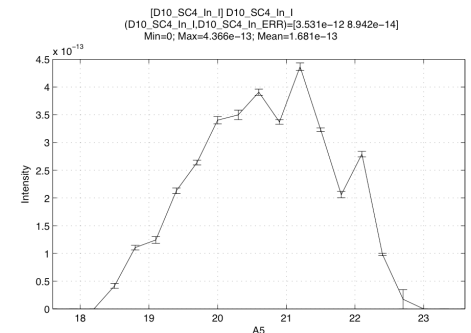
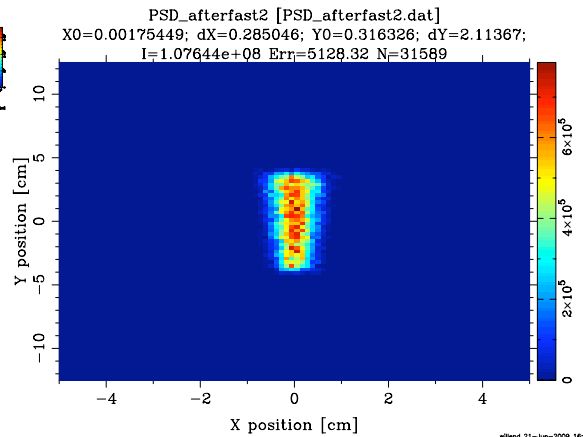
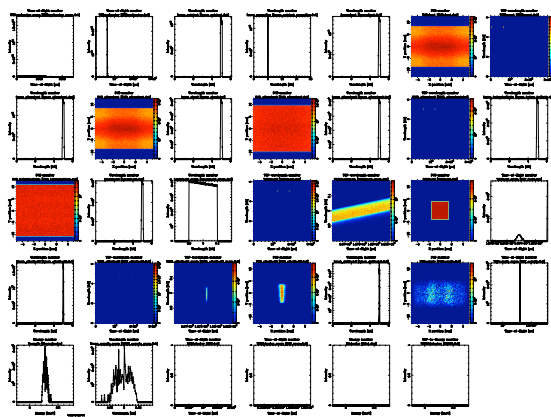
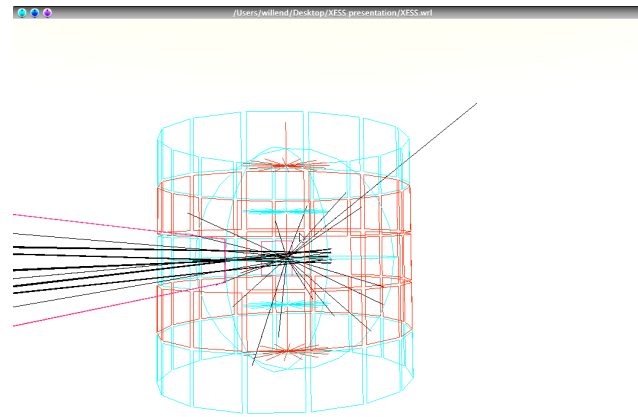
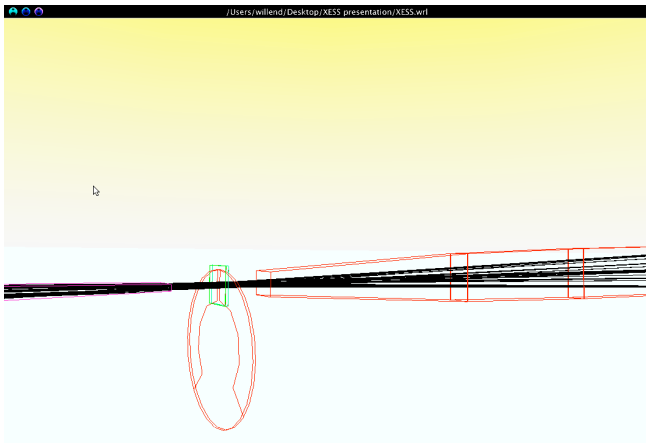
- | Instrumentation
- | Planning
- | Construction
- | Virtual experiments
- | Data analysis
- | Teaching



Instrumentation



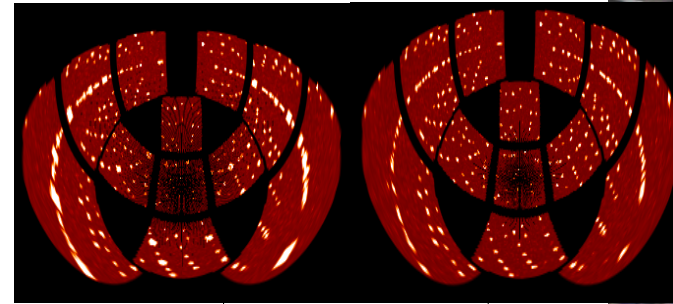
- Design and optimization of instruments



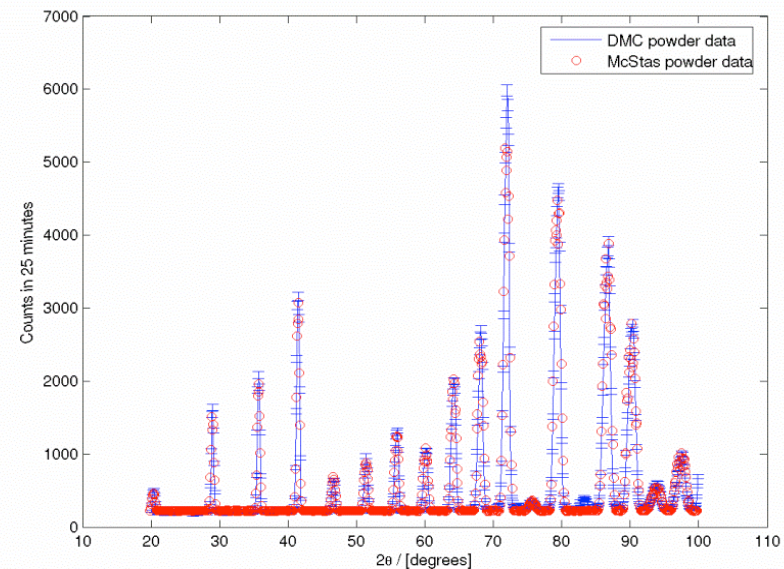
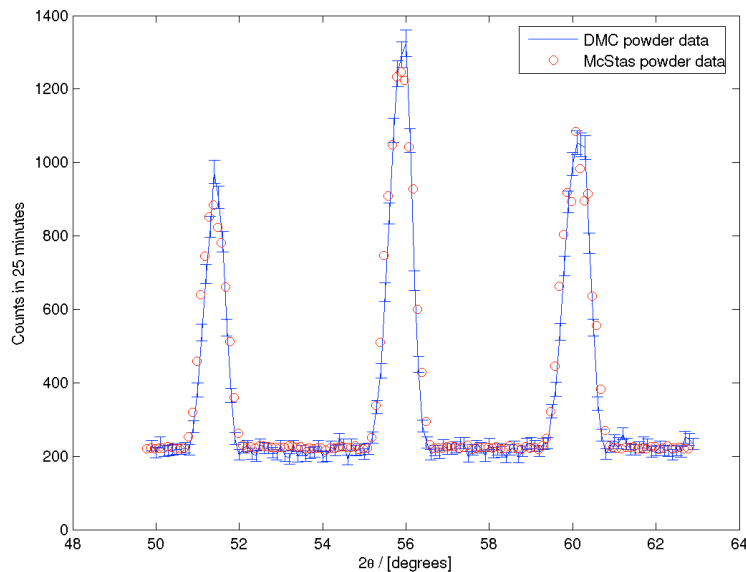
Virtual experiments (VE)

(definition:)

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



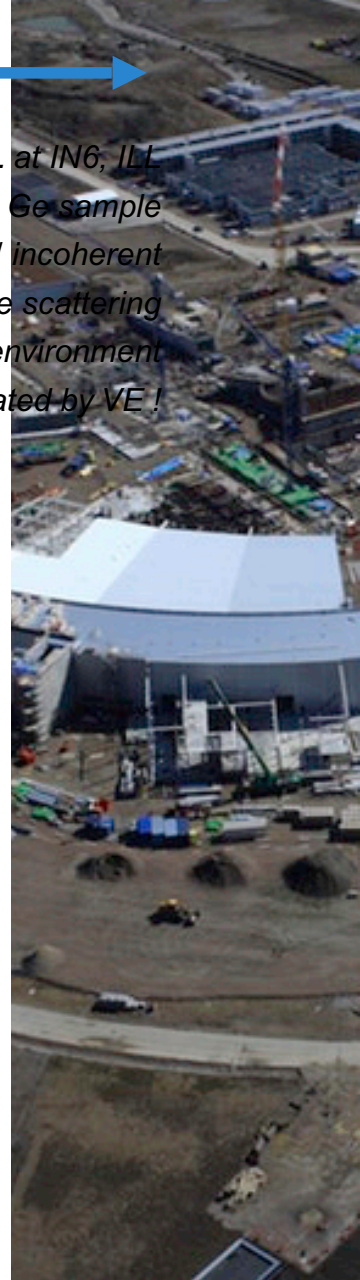
A. Daud-Aladine, ISIS



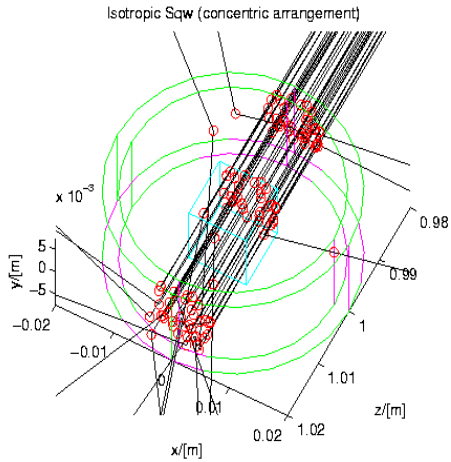
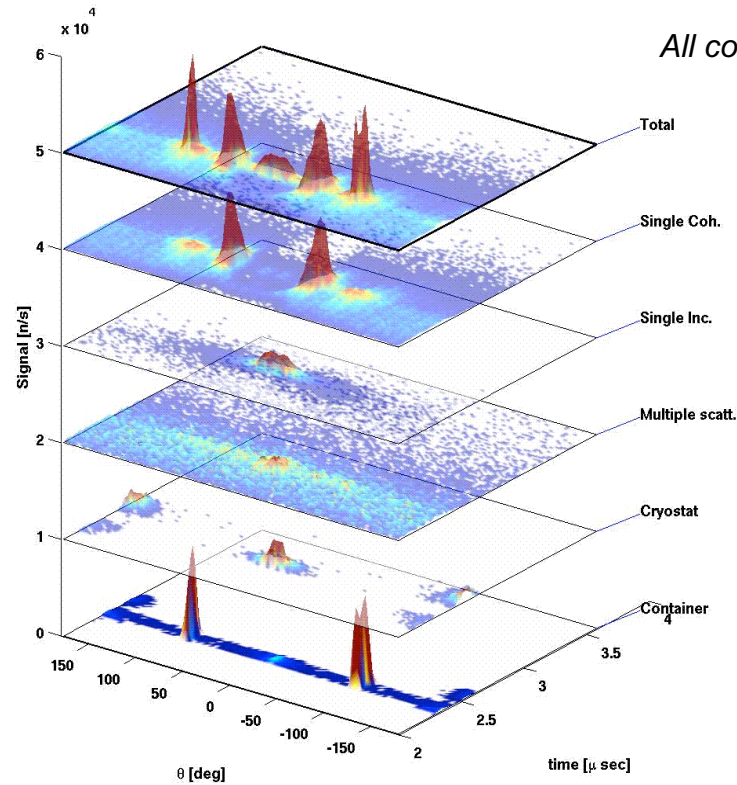
P. Willendrup, Risø DTU; Uwe Filges, L. Keller, PSI

Data analysis (1)

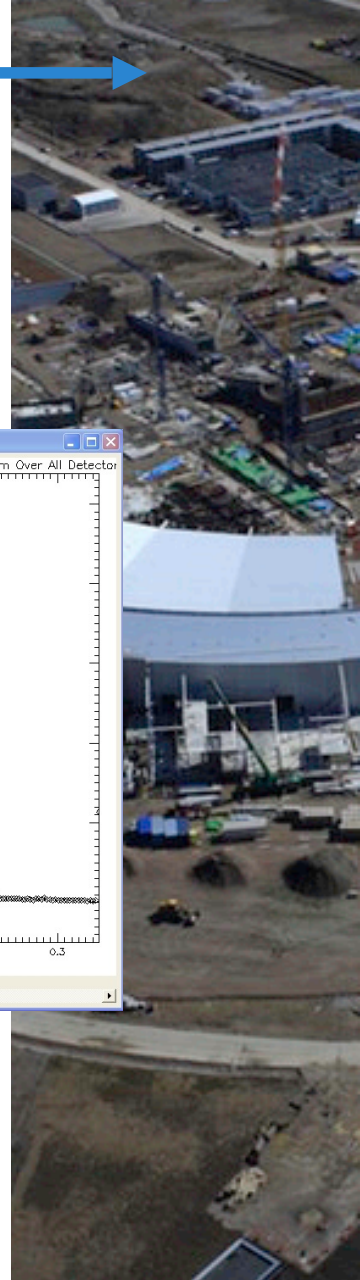
(using VE techniques)



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



E. Farhi, ILL

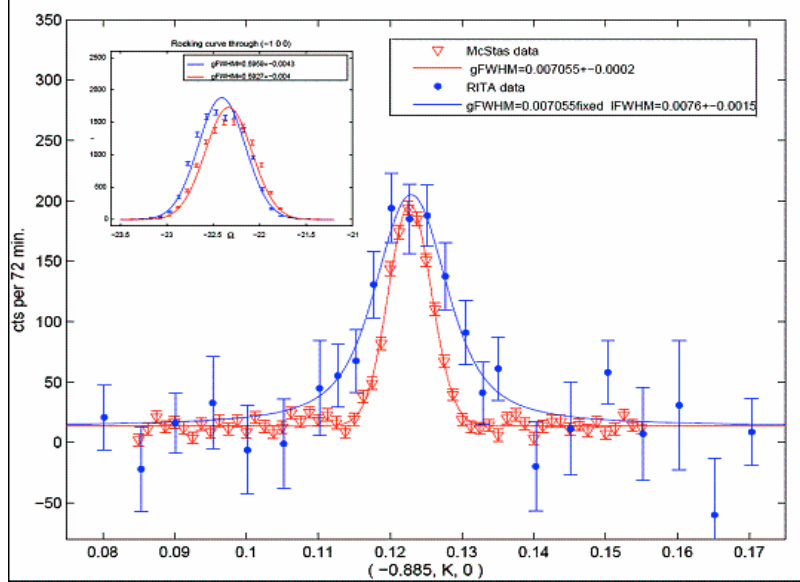


Data Analysis (2)

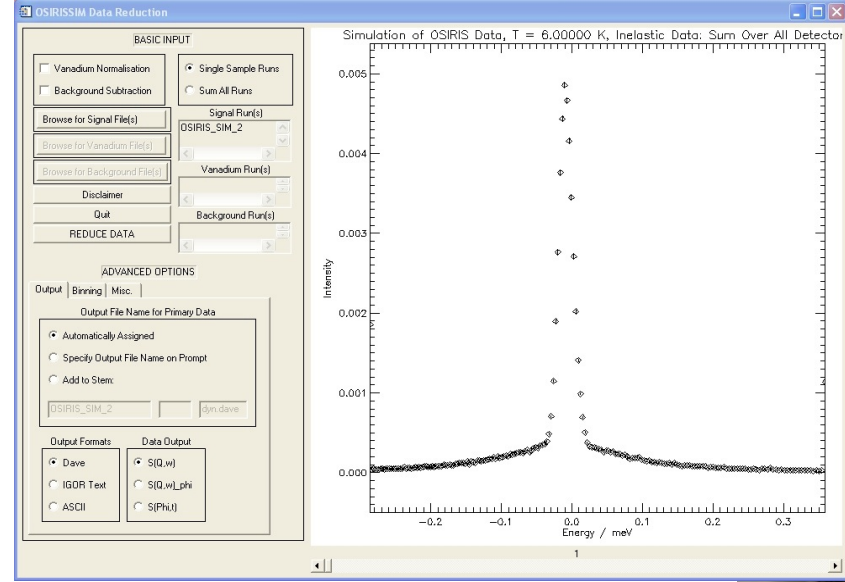
(using VE techniques)

VE data has been used to test data analysis programs

... and to check resolution effects



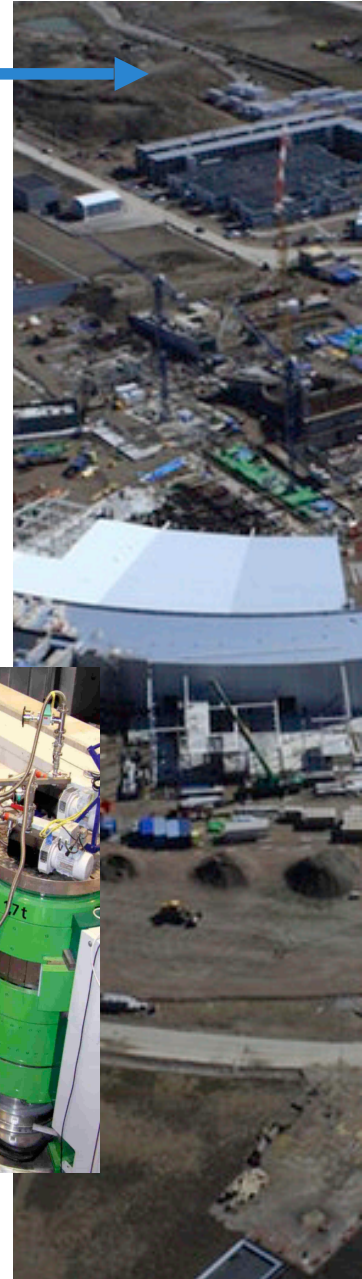
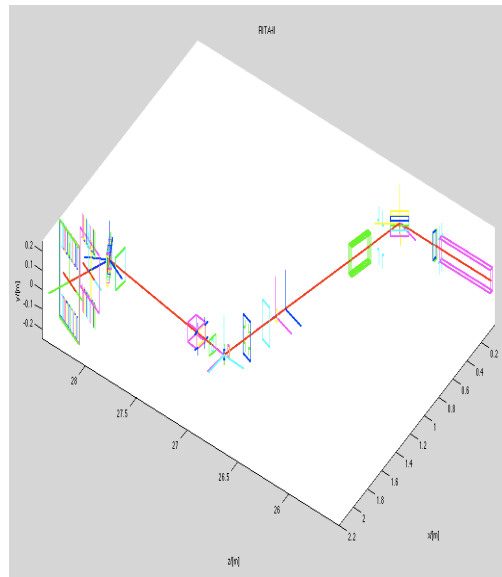
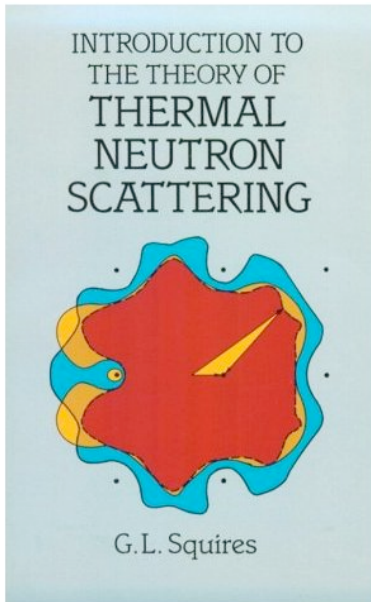
L. Udby, Risø-DTU



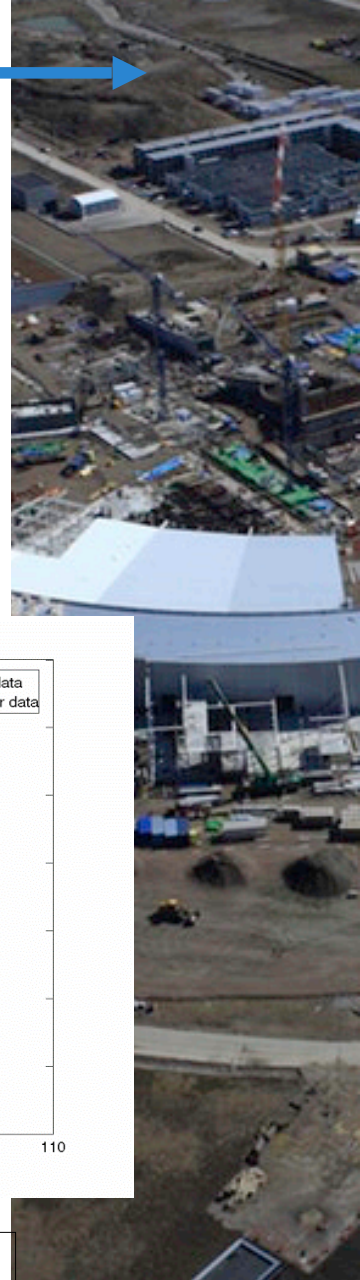
P. Tregenna-Piggott, PSI

Teaching / training purposes

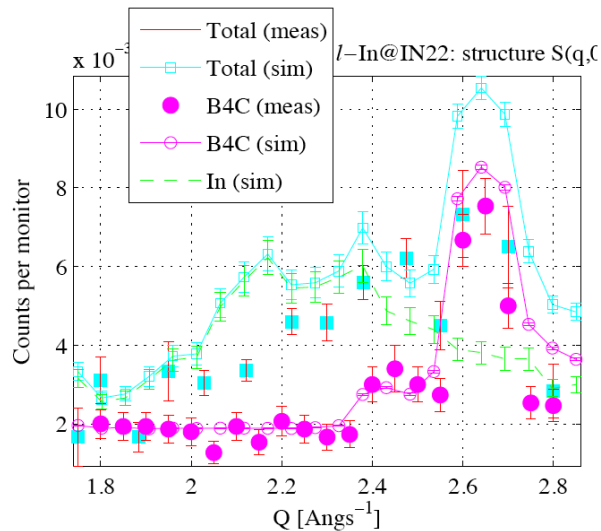
- Workshops
- Teaching
 - University of Copenhagen course on Neutron Scattering 2005-, DTU 2011-



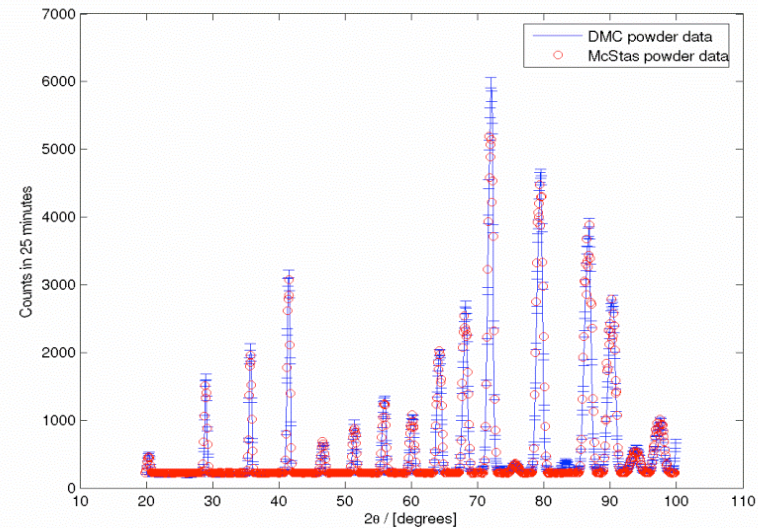
Reliability - cross comparisons



- ┆ 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

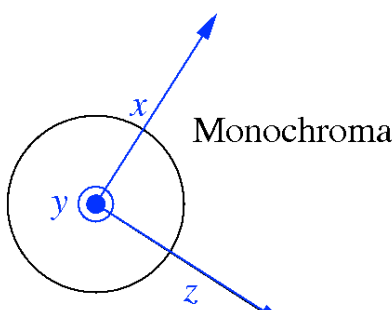


E. Farhi, P. Willendrup et al., in preparation

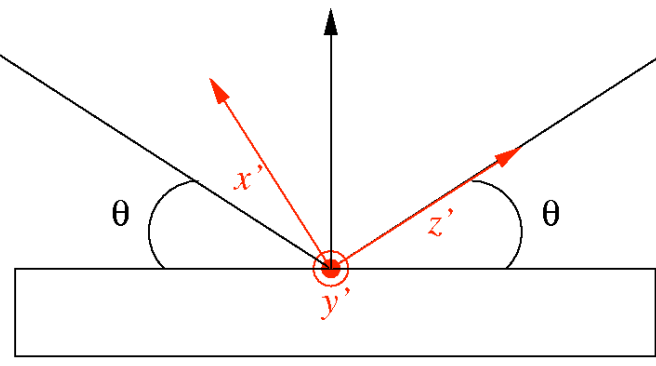


P. Willendrup et al., Physica B, 386, (2006), 1032.

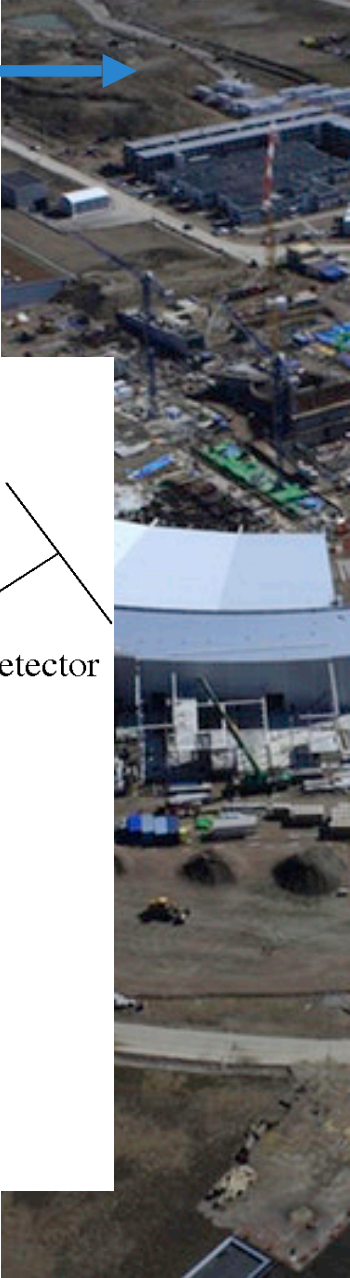
McStas: key concepts



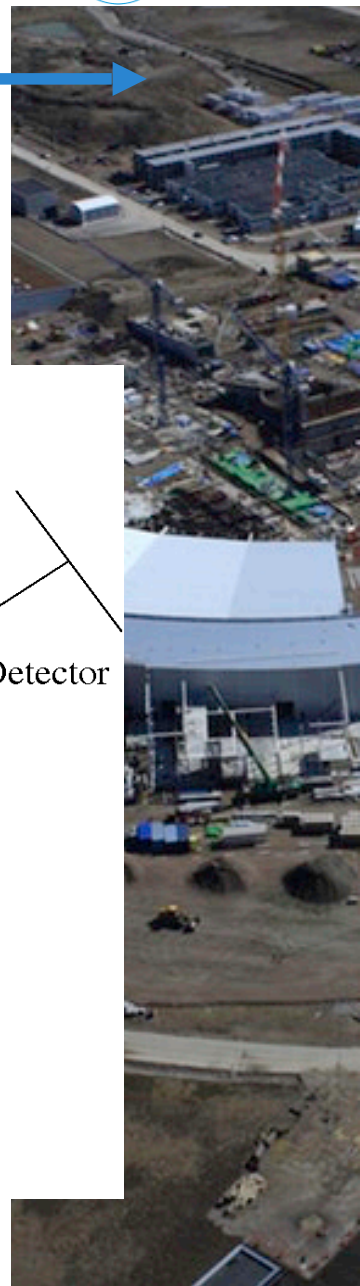
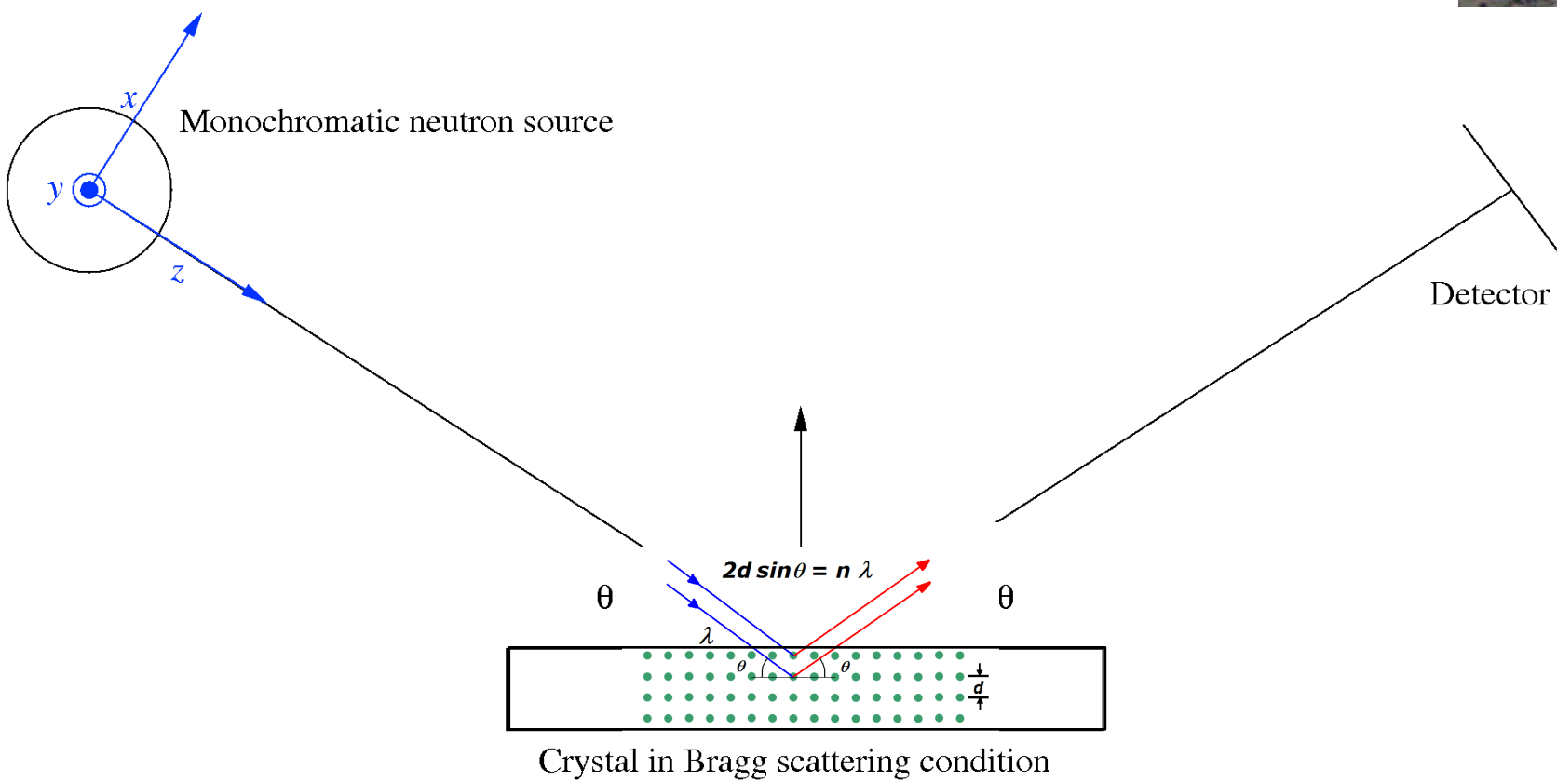
Neutron ray/package:
 Weight (p): # neutrons (left) in the package
 Coordinates (x,y,z)
 Velocity (v_x, v_y, v_z)
 Spin (s_x, s_y, s_z)
 Time (t)



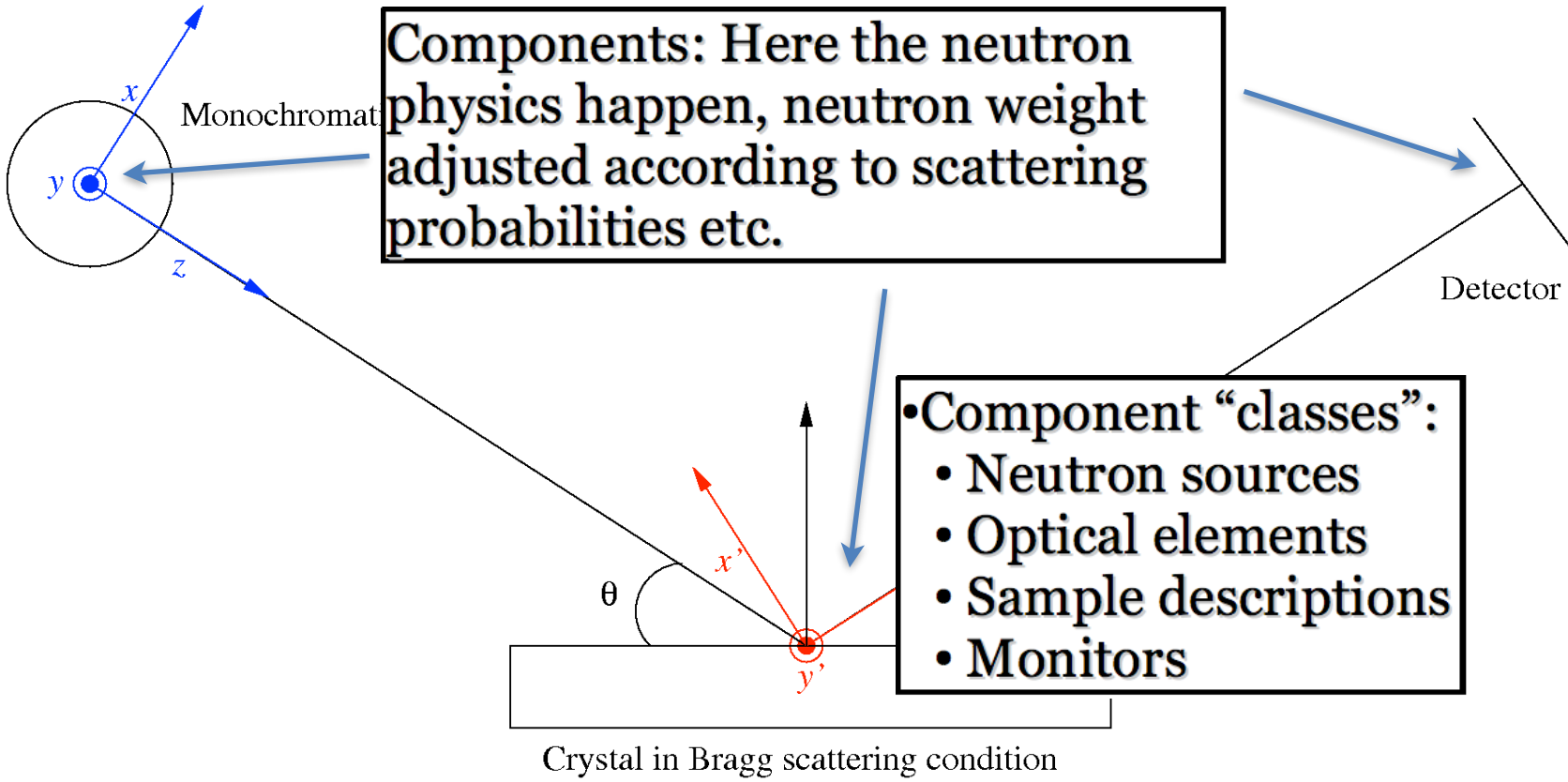
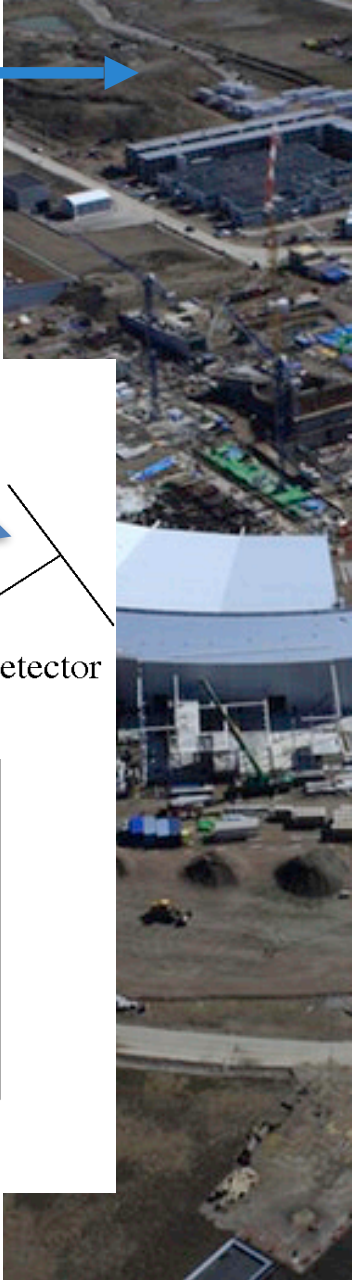
Detector



McStas: key concepts

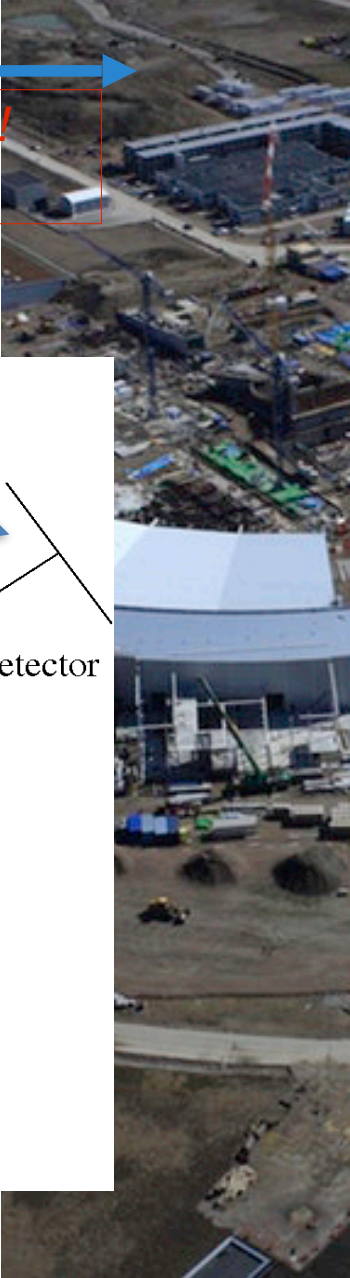
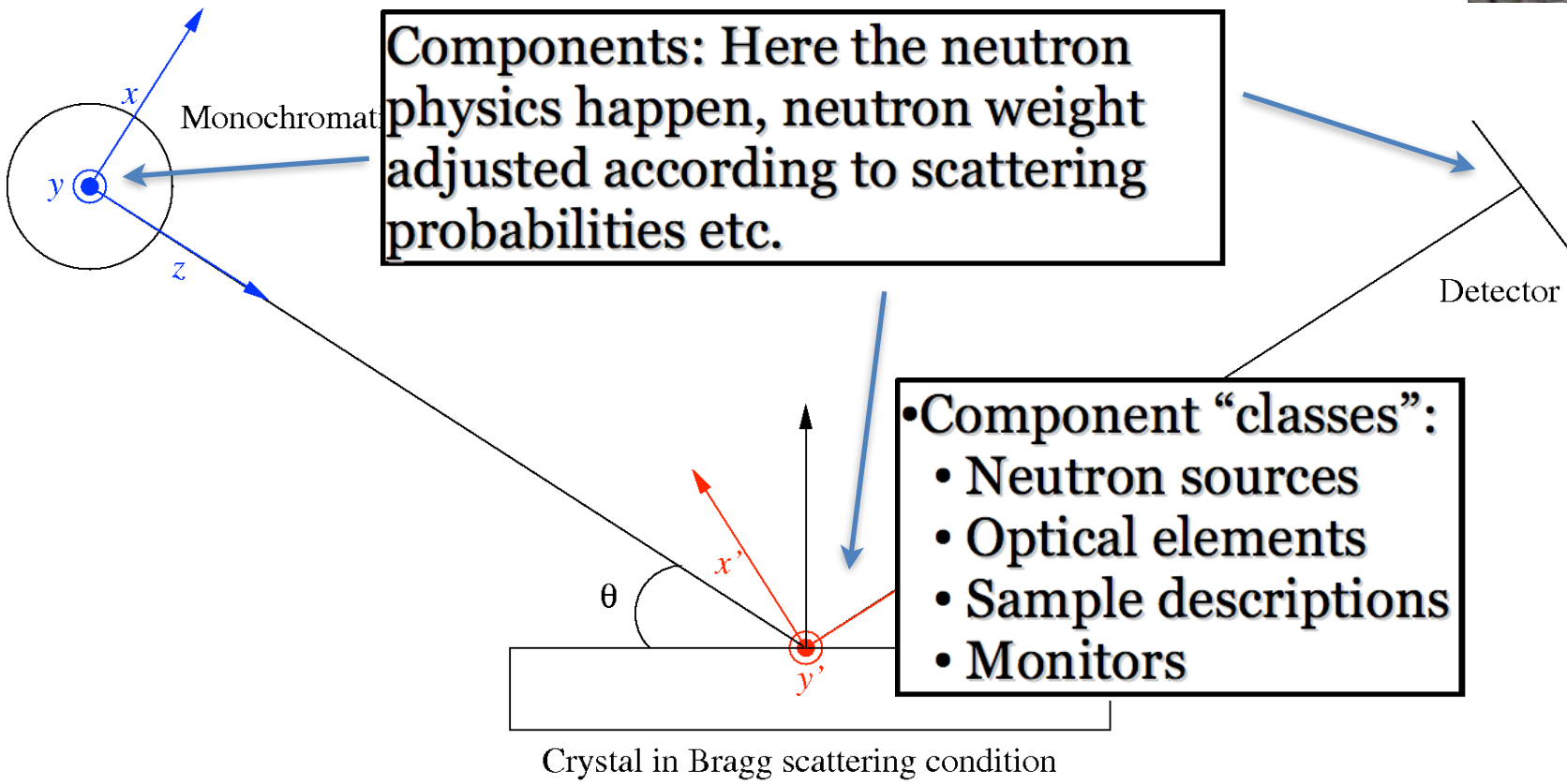


McStas: key concepts

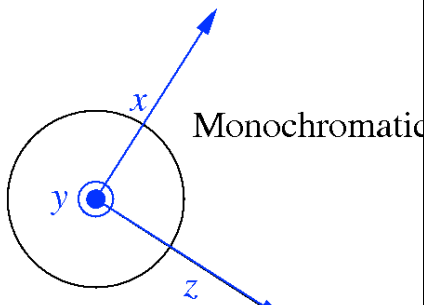
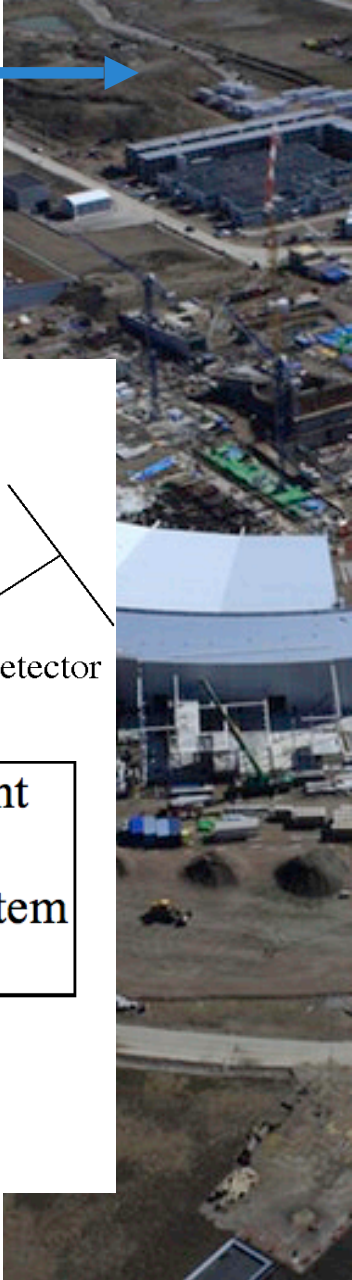


McStas: key concepts

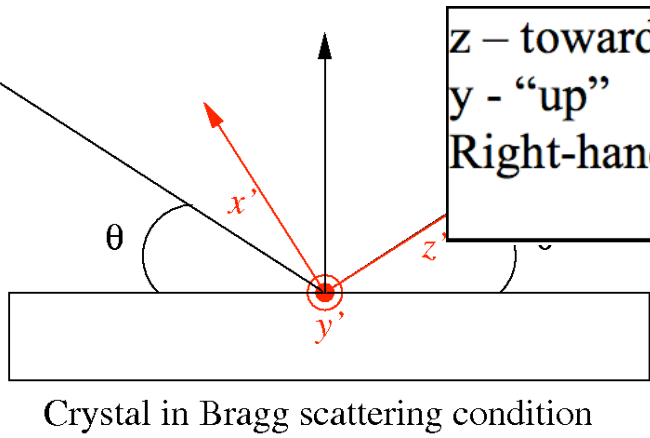
Local, internal coordinate system!



McStas: key concepts



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



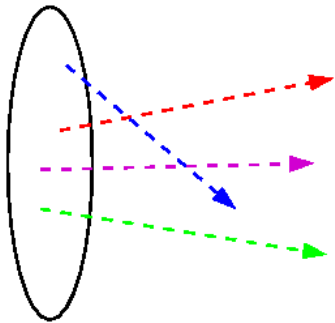
z - towards "next" component
y - "up"
Right-handed coordinate system

Detector

In the big picture...

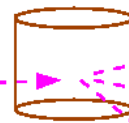


1. Particles emitted with random starting conditions via MC



2. Particles are "ray-traced" through space

3. Will eventually meet other objects e.g. a studied experimental sample and get scattered via MC again



4. At various points in the instrument the particle states are measured in so-called monitors or detectors



McStas overview

Portable code (Unix/Linux/Mac/Windows)

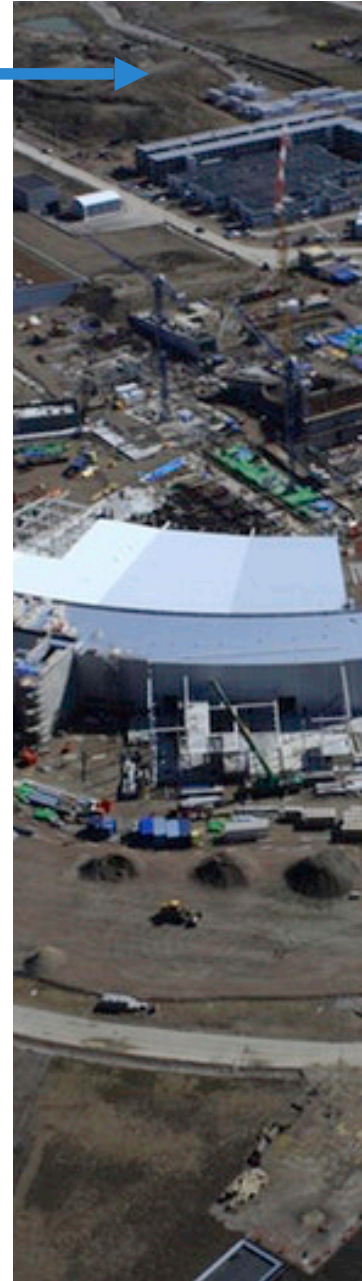
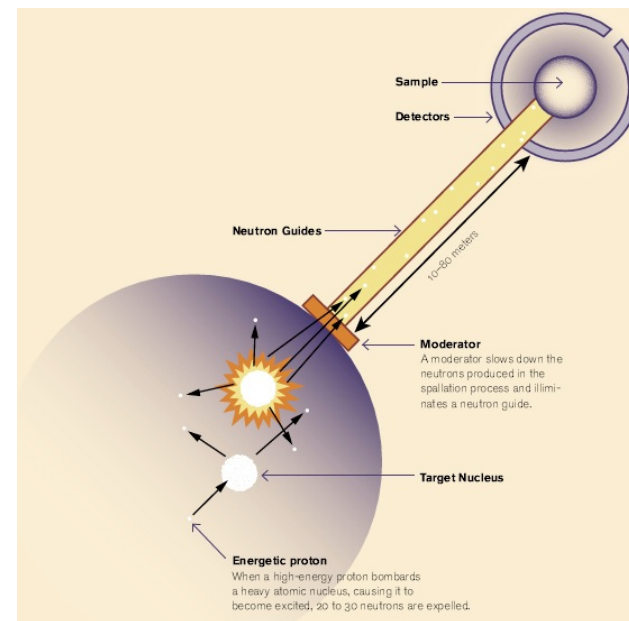


Ran on everything from iPhone to 1000+ node cluster!

'Component' files (~100) inserted from library

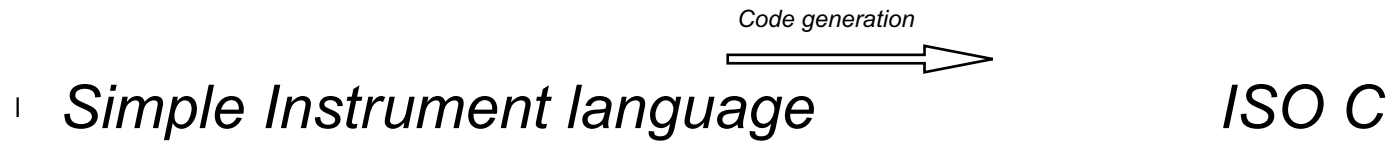
- ┆ Sources
- ┆ Optics
- ┆ Samples
- ┆ Monitors
- ┆ If needed, write your own comps

DSL + ISO-C code gen.



Under-the-hood / inner workings

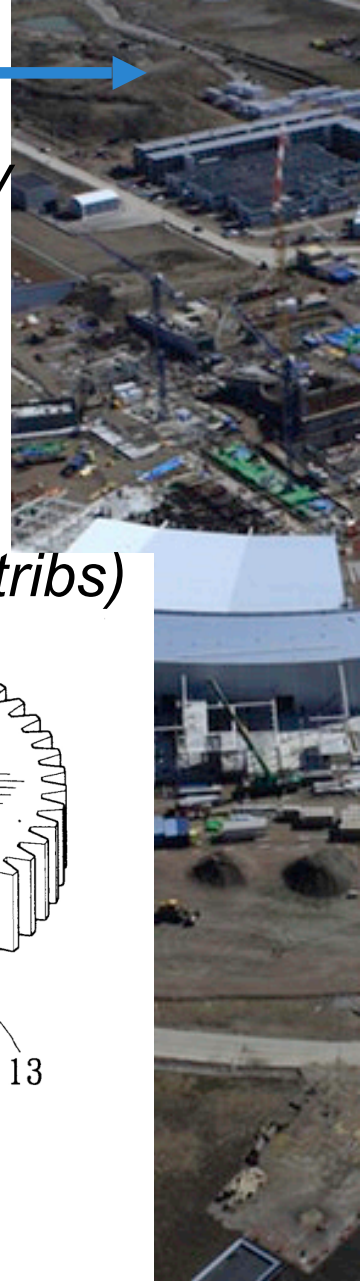
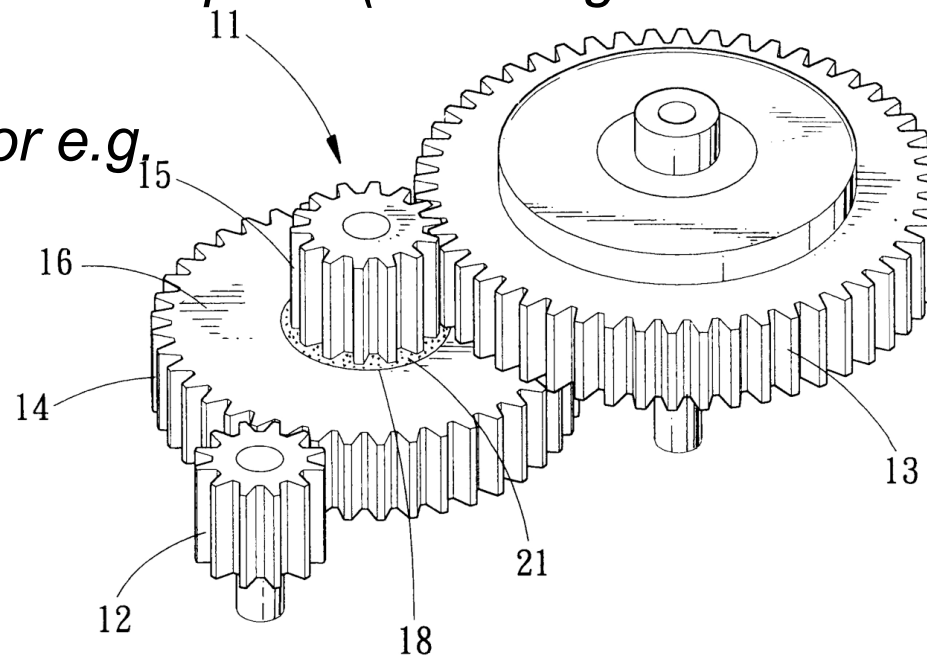
- Domain-specific-language (DSL) based on compiler technology (LeX+Yacc)



- Component codes realizing beamline parts (including user contribs)

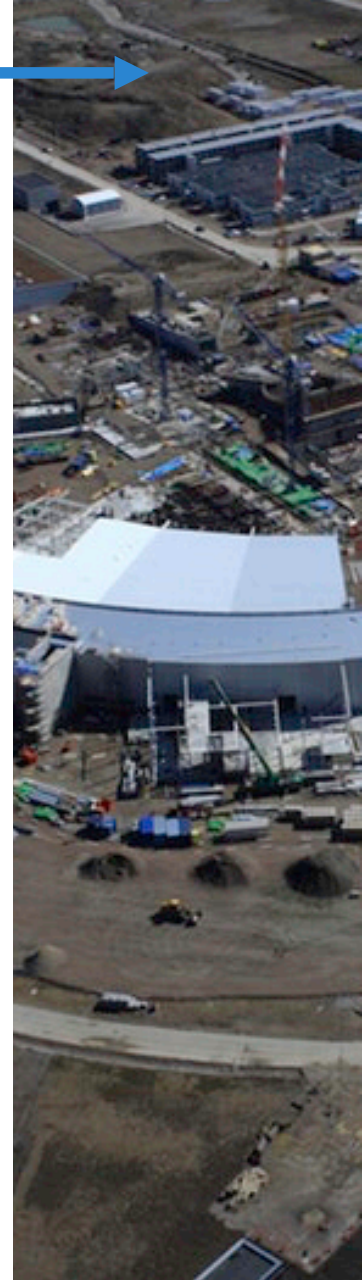
- Library of common functions for e.g.

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

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

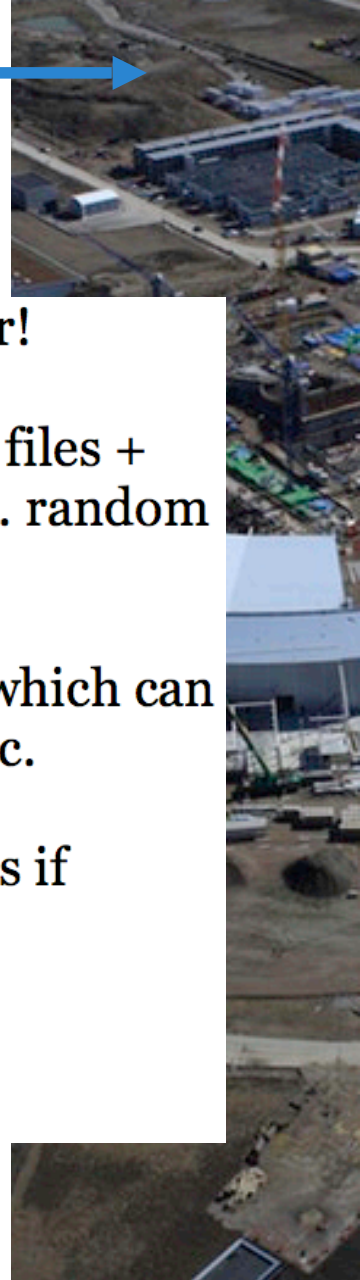
MCDISPLAY
%{
magnify("xy");
circle("xy", 0, 0, 0, radius);
%}

END

```

Written by developers
and possibly you!

Generated c-code



```

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

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

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

t=0;
z=0;

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

randvec_target_rect(&xf, &yf, &rf, &dir,
  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

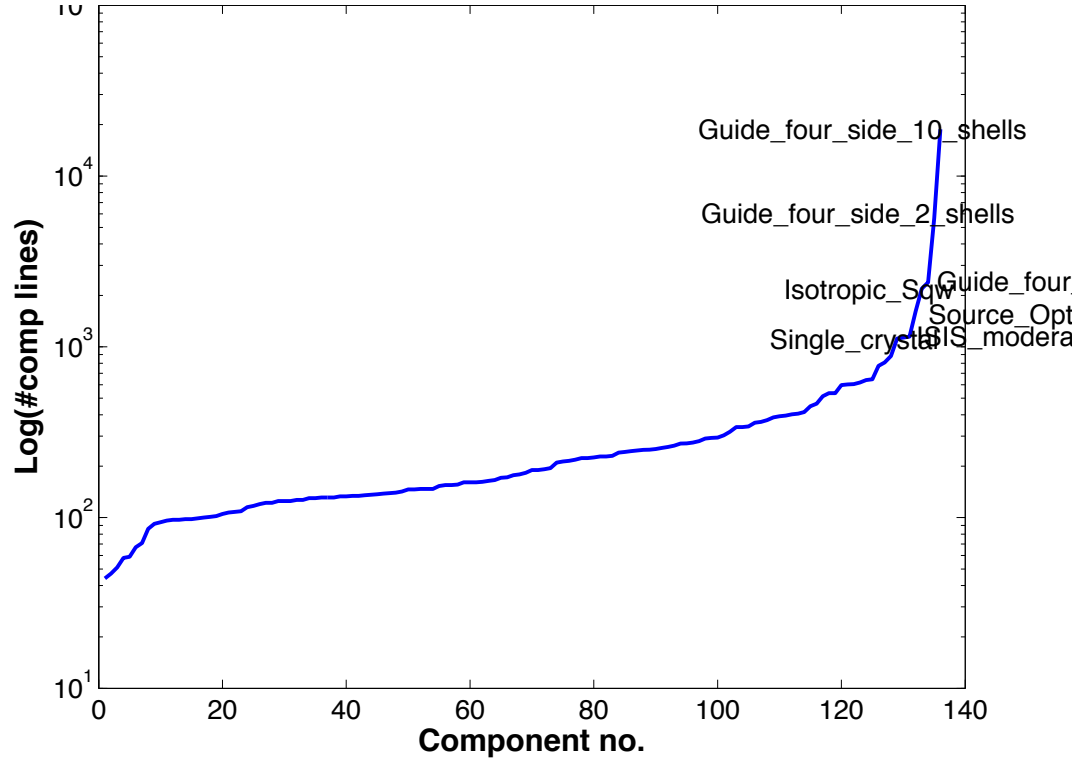
Output is a single c-file, which can
be compiled using e.g. gcc.

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:

Number of lines of code per component - 199 comps in total i)



Including user contribs

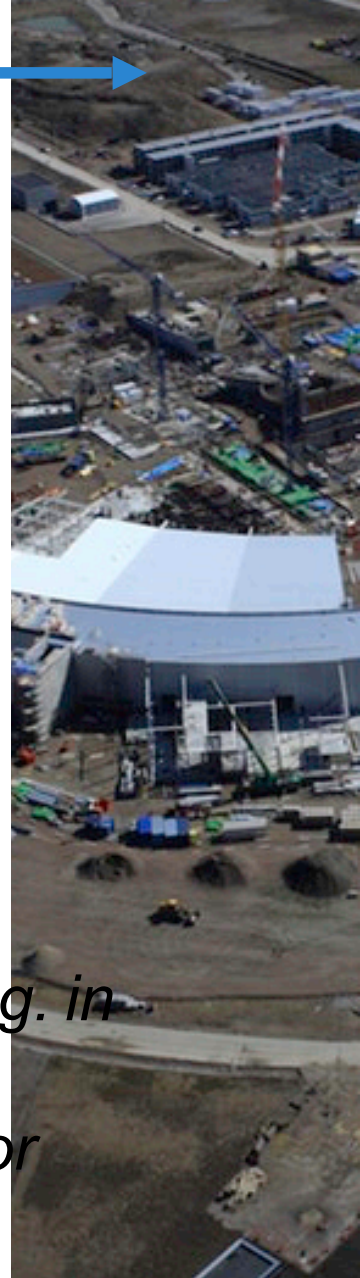


| Well-developed community support

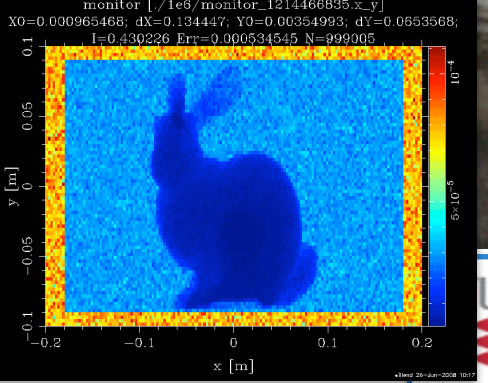
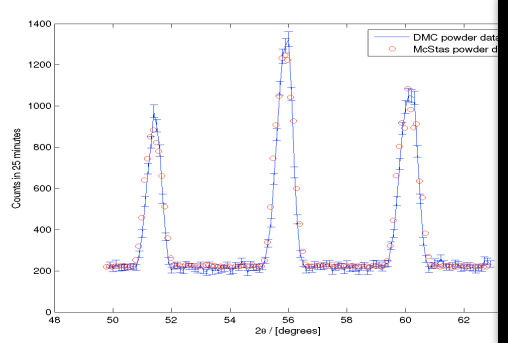
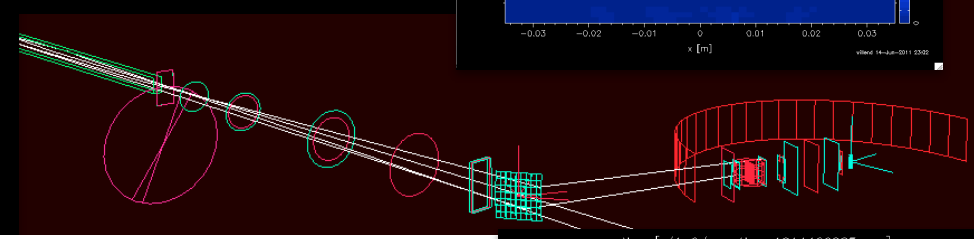
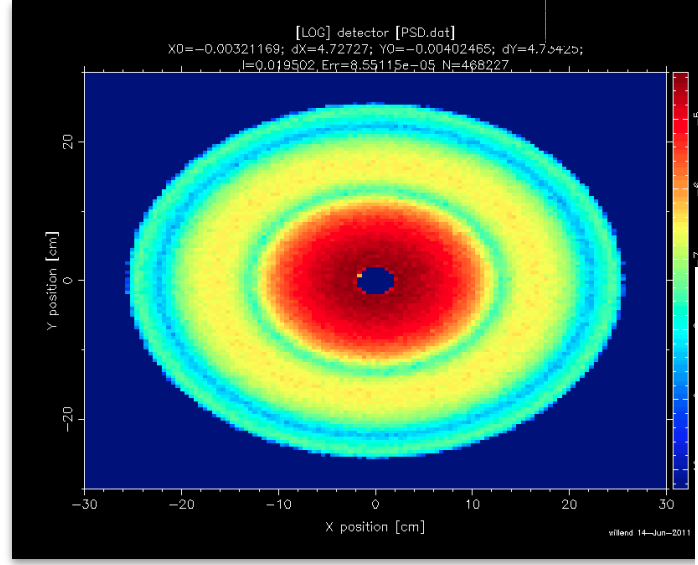
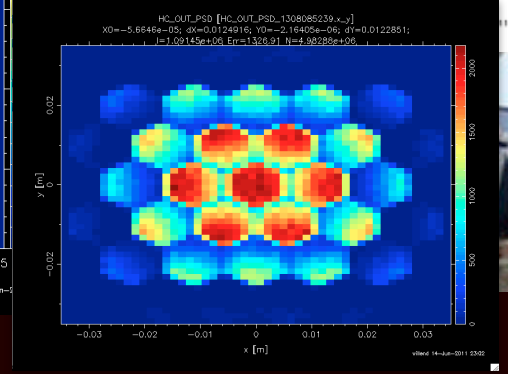
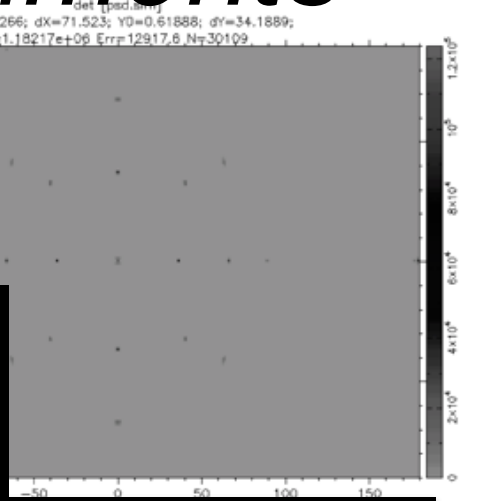
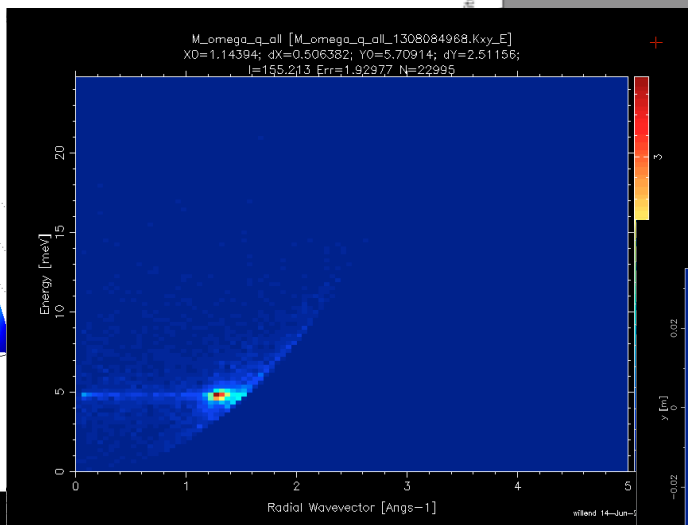
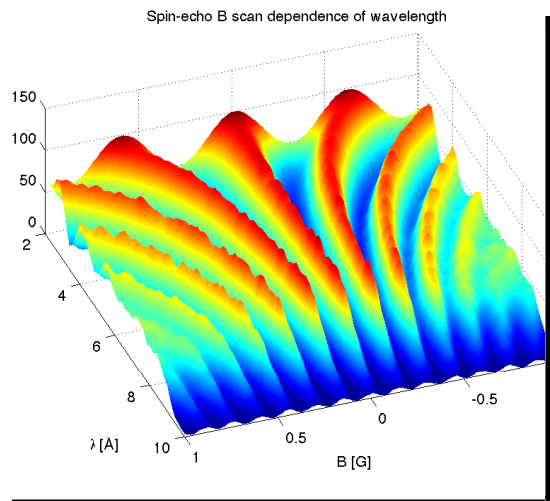
- 30-40% of existing and new additions are from users
- No direct refereeing of the code, but these requirements:
 - At least one test-instrument
 - Meaningful documentation headers (in-code docs)
- Contributions go in dedicated contrib/ section of library

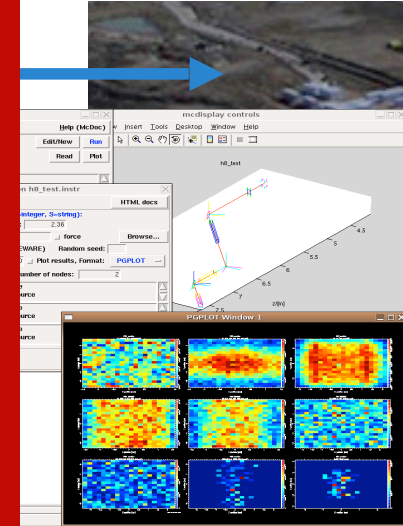
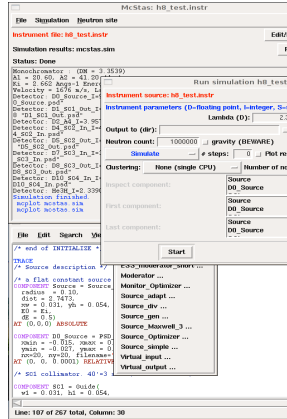
| Natural life-cycle of contrib's

- 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



Example suite: ~140 instruments





**THIS IS NOT
THE END
IT'S JUST
THE BEGINNING**

