McStas Introduction

McStas

Agenda

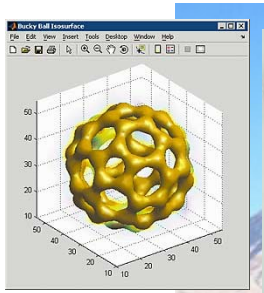
- *A (very) brief introduction to Monte Carlo & raytracing*
- *Components of neutron instruments*
- *How McStas works under the hood*
- *Components and instruments*
- *A demo*

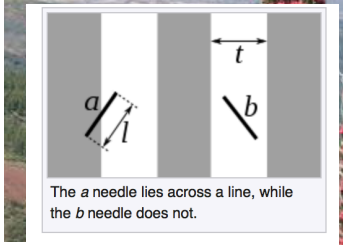
Monte Carlo methods

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





The *a* needle lies across a line, while the *b* needle does not.

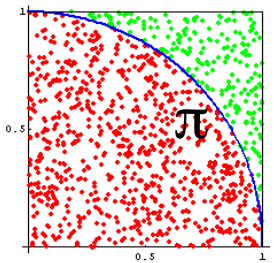
In mathematics, **Buffon's needle problem** is a question first posed in the 18th century by Georges-Louis Leclerc, Comte de Buffon:[1]

> Suppose we have a floor made of parallel strips of wood, each the same width, and we drop a needle onto the floor. What is the probability that the needle will lie across a line between two strips?

Buffon's needle was the earliest problem in geometric probability to be solved; it can be solved using integral geometry. The solution, in the case where the needle length is not greater than the width of the strips, can be used to design a Monte Carlo method for approximating the number π, although that was not the original motivation for de Buffon's question.[2]

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 → π/4

DTU ess EUROPEAN SPALLATION SOURCE

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

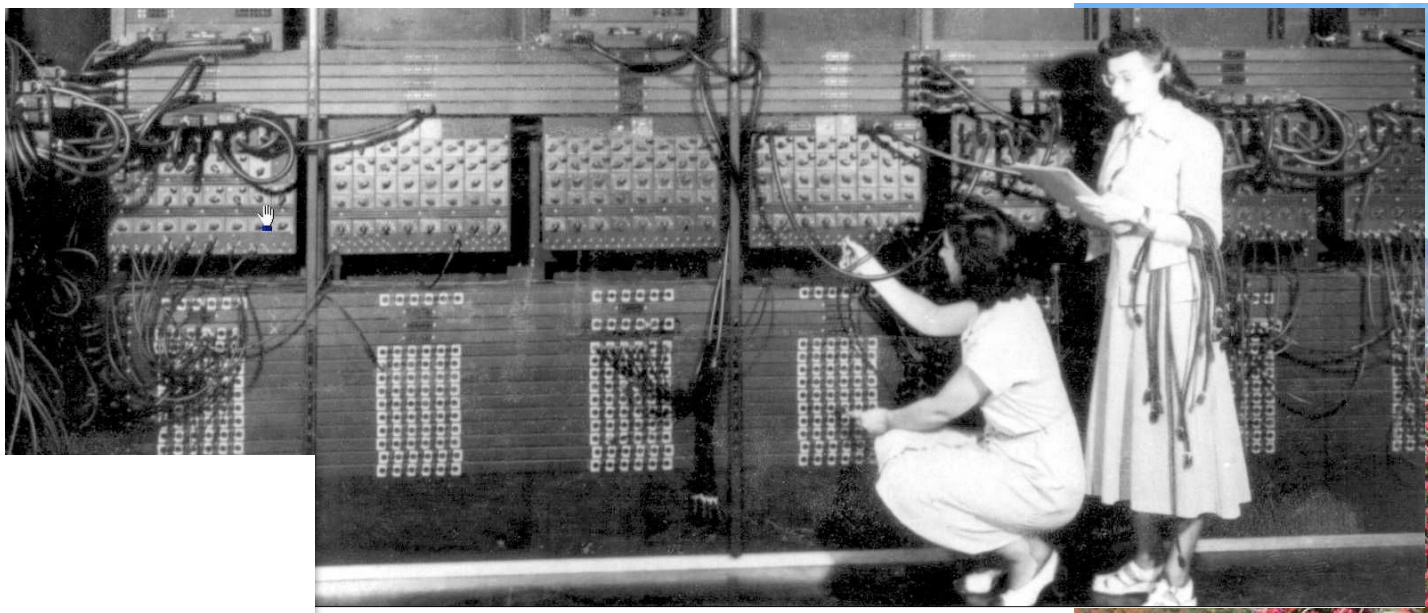
Neutron Scattering and Absorption in U and Pu, Origin of MCNP



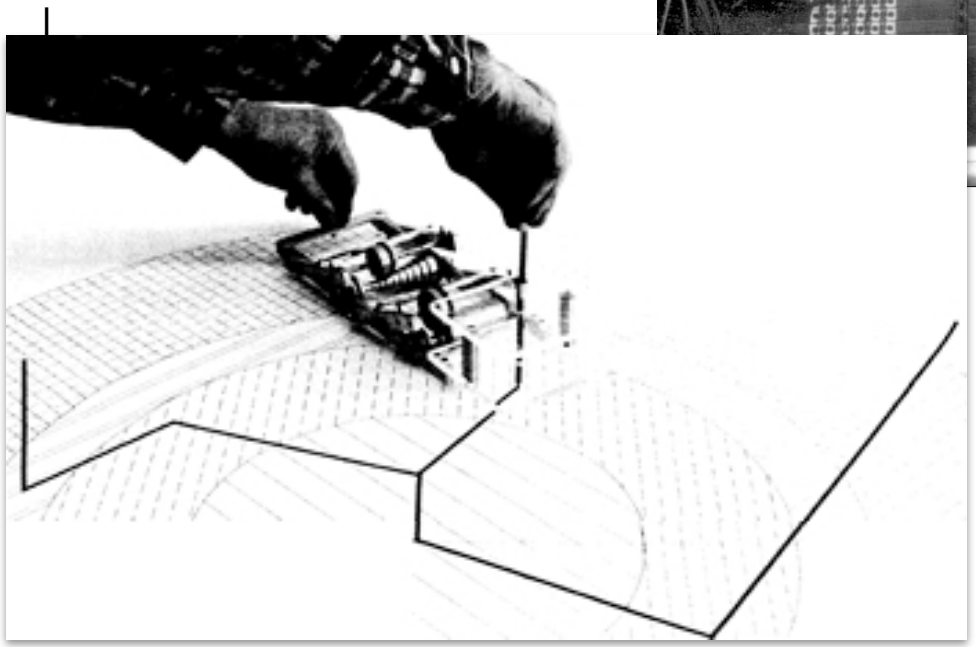
Name:

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

Monte Carlo techniques



ENIAC



FERMIAC

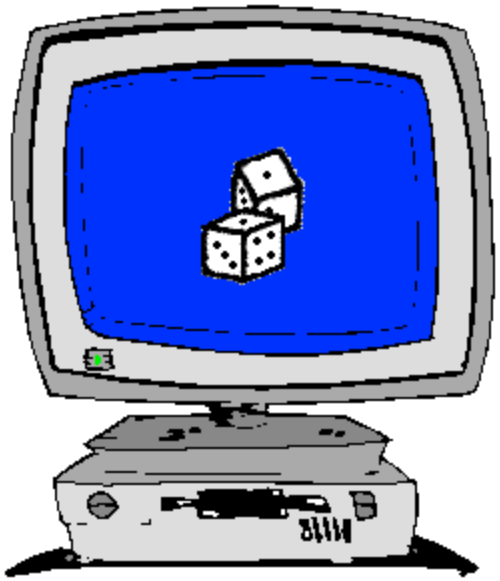
How to implement Monte Carlo methods ?

Good random generator:

from thermal electronic noise (hardware)

or quasi-random generators => 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 $\longrightarrow$ 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 ($1/\div n$)

DTU ess EUROPEAN SPALLATION SOURCE

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 **(r, v, s, t)**, and are transported along instrument models.*

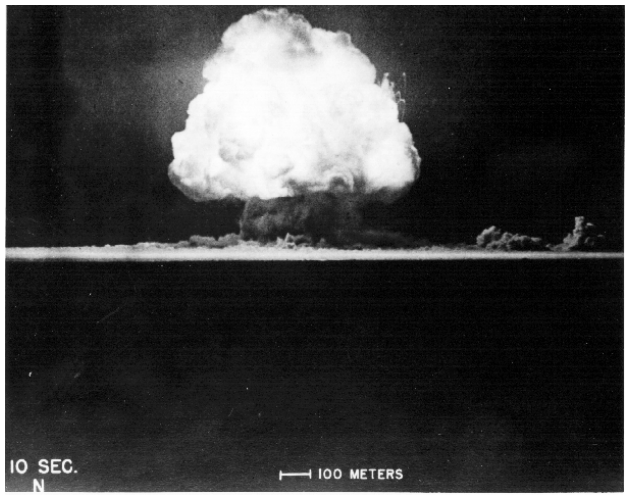
Propagation simply uses Newton rules, incl. gravitation.

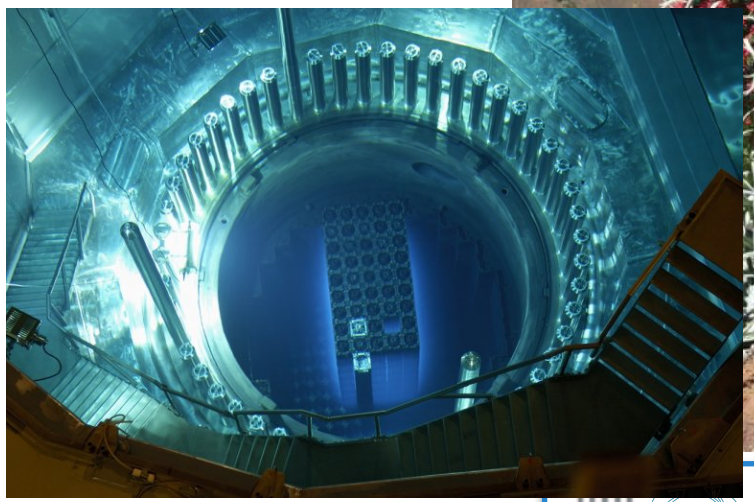
X-ray tracing

Shadow, McXtrace, RAY, …

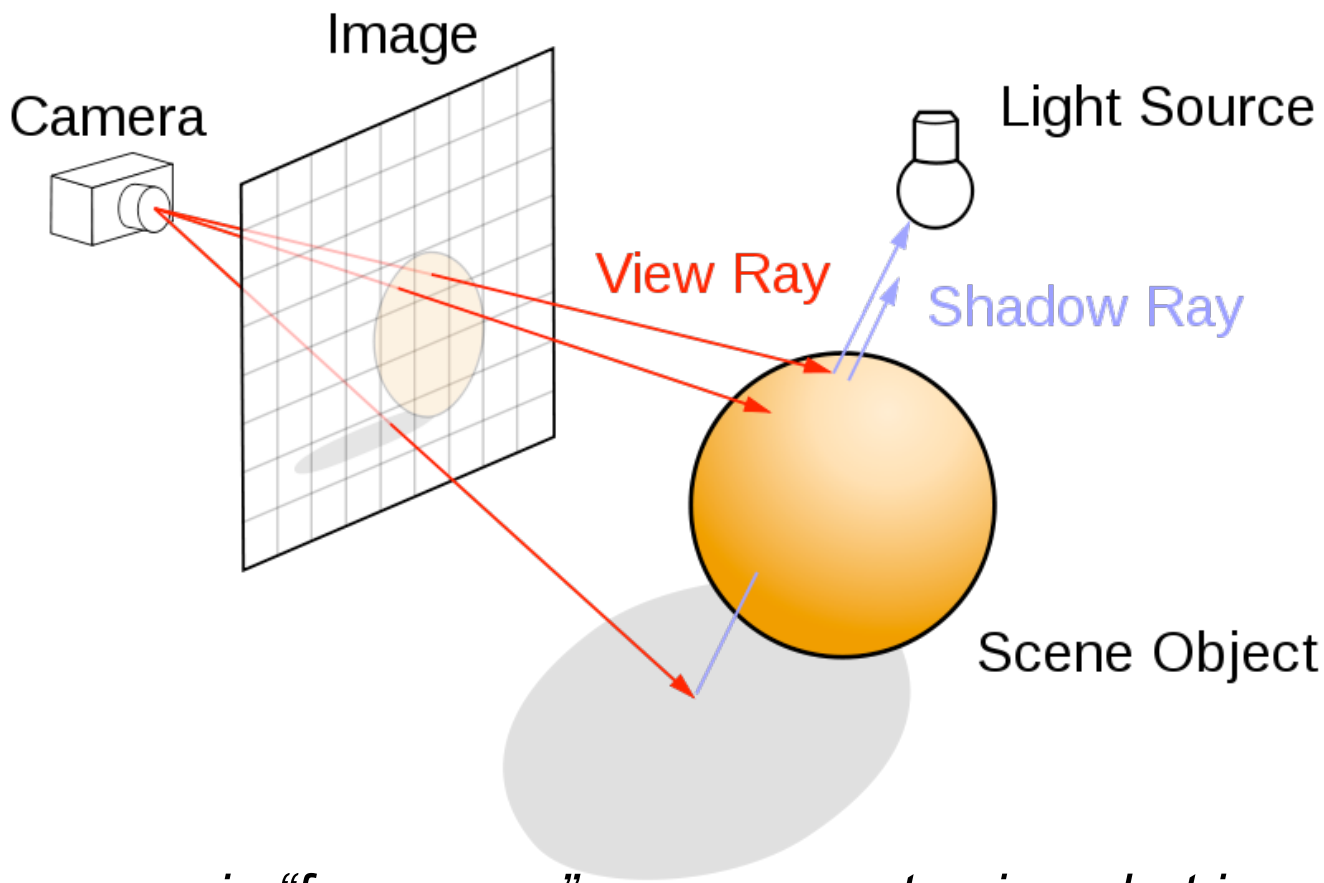
Monte Carlo techniques

- *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*





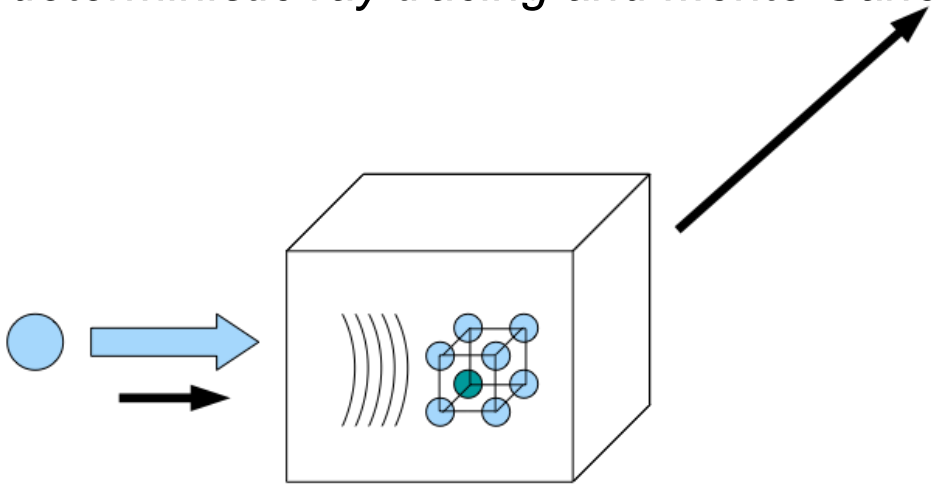
Ray-tracing methods



- *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 uses to implement gravity*

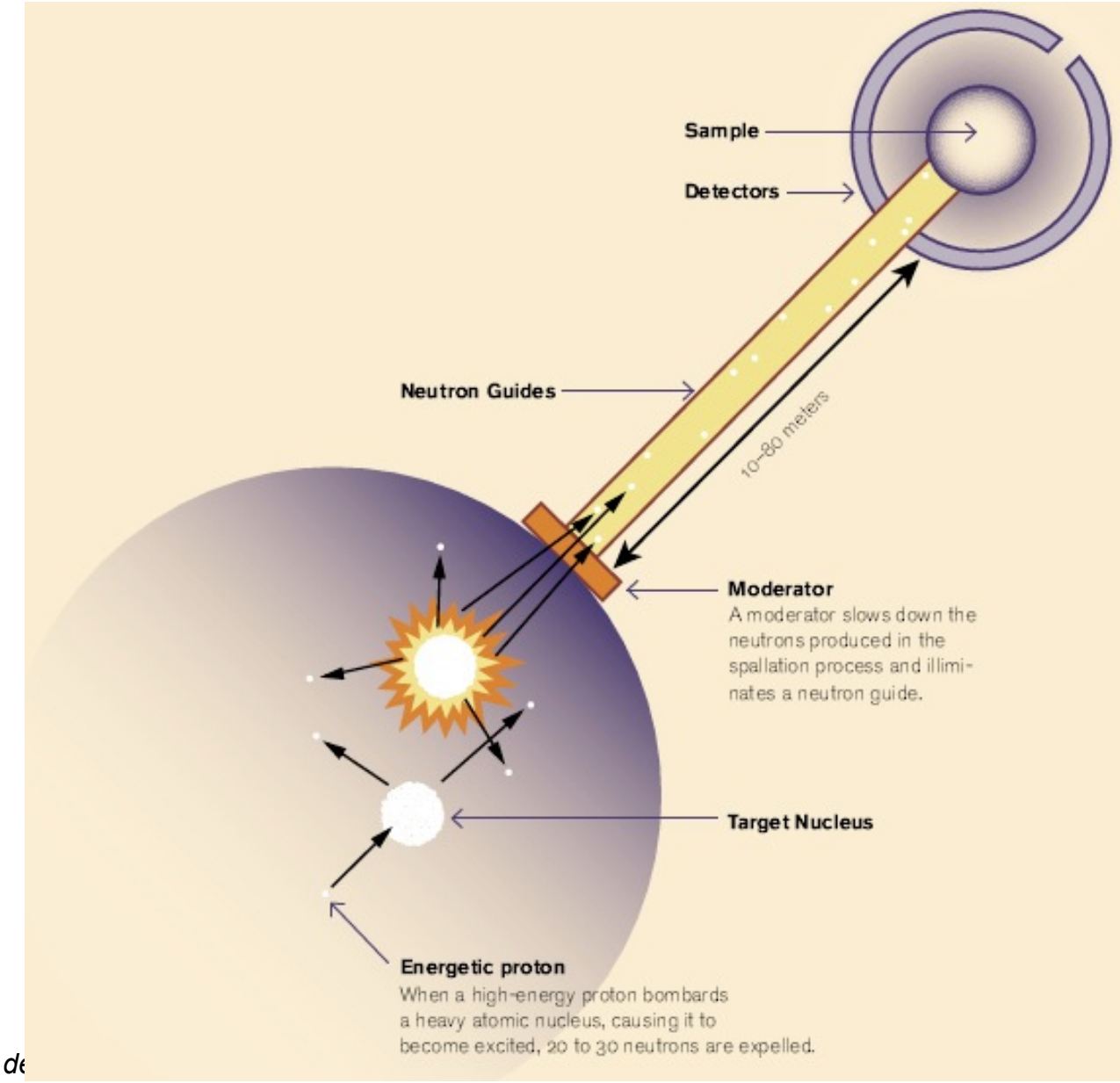
Elements of Monte-Carlo raytracing

- *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*



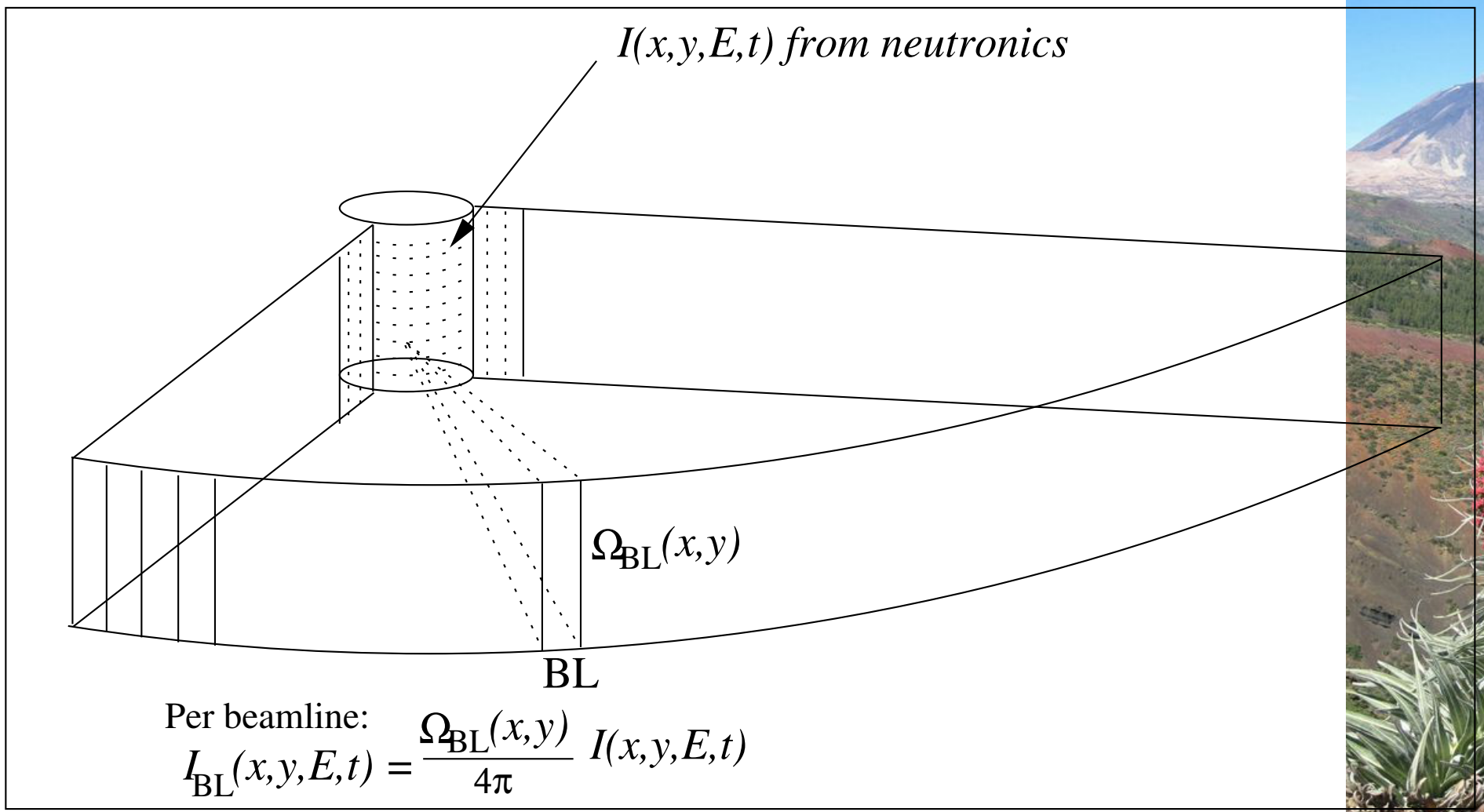
- *Result: A realistic and efficient transport of neutrons in the thermal and cold range*

Components of neutron instruments



DTU ess EUROPEAN SPALLATION SOURCE

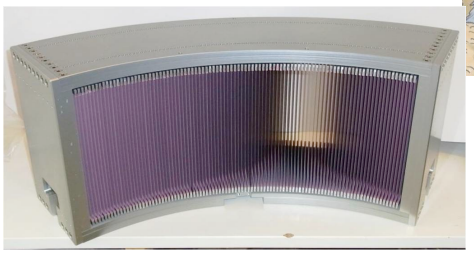
Moderators... (Where McStas starts)

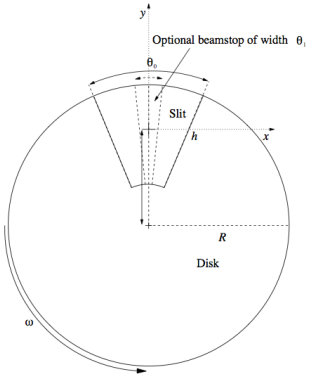


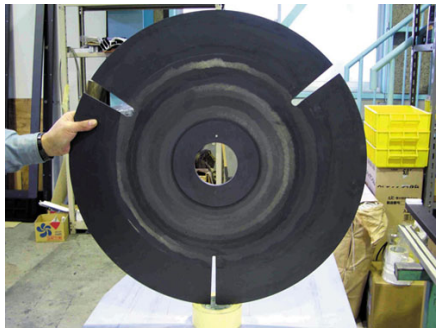
Per beamline:

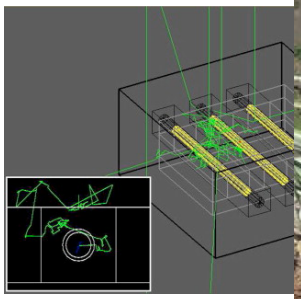
$$I_{BL}(x,y,E,t) = \frac{\Omega_{BL}(x,y)}{4\pi} I(x,y,E,t)$$

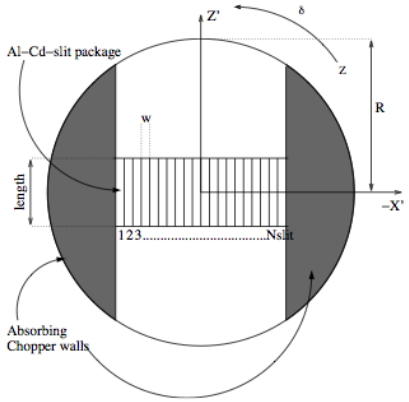
Neutron optics and other instrument components

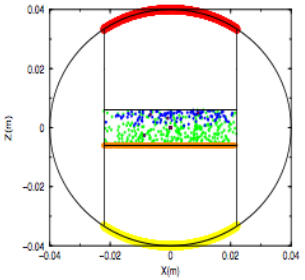




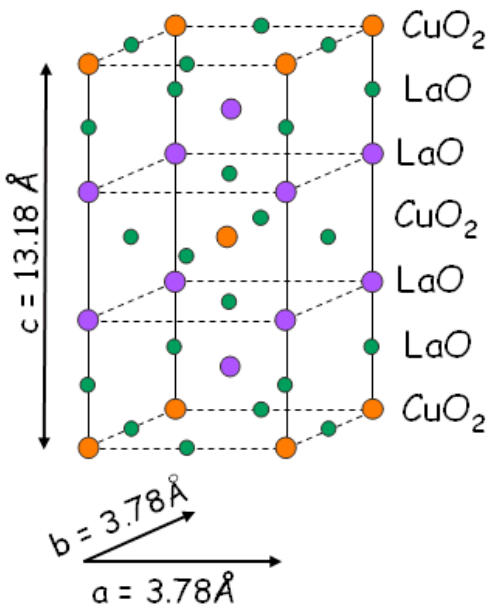


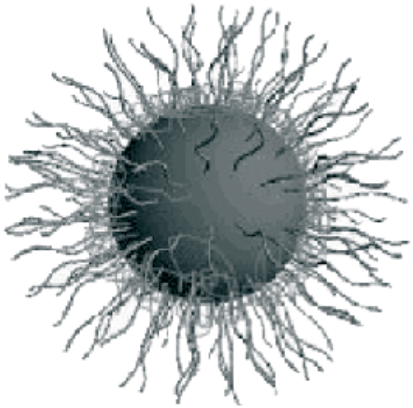




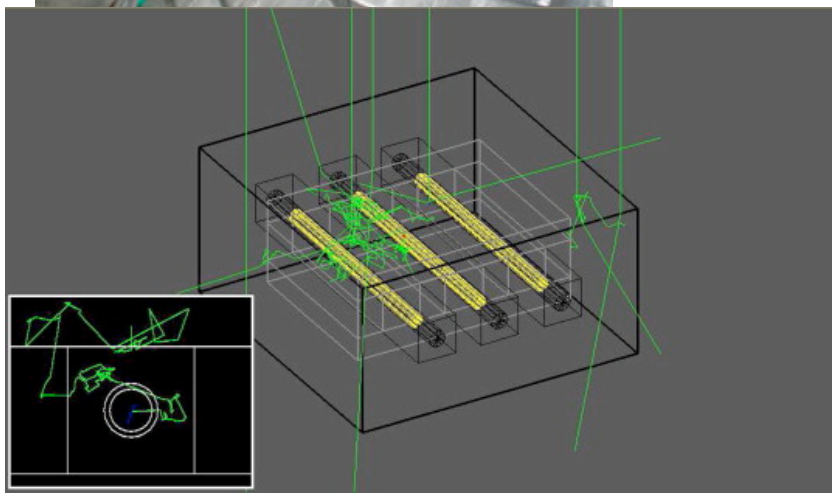


Samples studied...





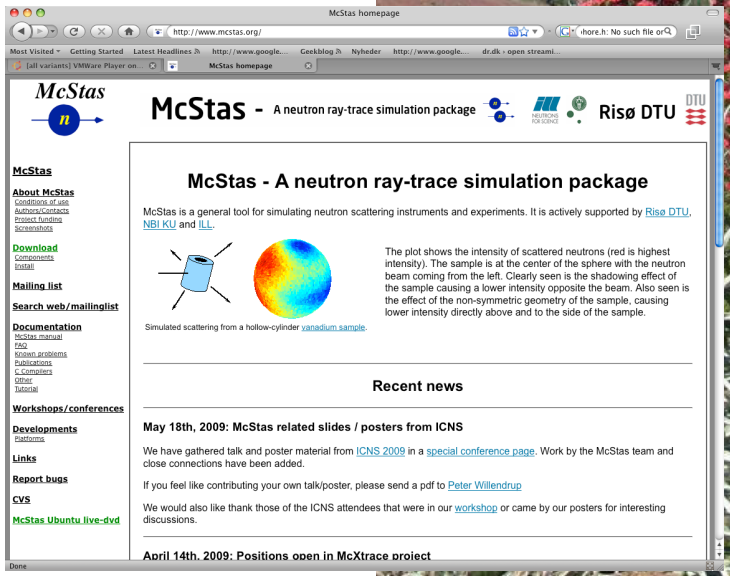
Detectors



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

DTU ess EUROPEAN SPALLATION SOURCE

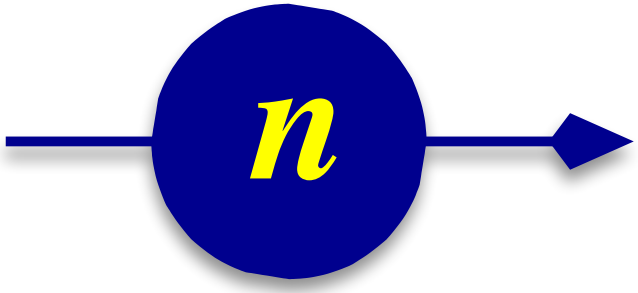
McStas

- Fle
- Cu

RISØ

McStas

v. 1.0 1998
v. 2.5 2018



DTU

PAUL SCHERRER INSTITUT
PSI

NEUTRONS FOR SCIENCE

EUROPEAN SPALLATION SOURCE

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

mcstas-users@mcstas.org mailinglist

DTU EUROPEAN SPALLATION SOURCE

McXtrace - since jan 2009 similar for X-rays

Main Page – McXtraceWiki

http://www.mcxtrace.org/index.php?title=Main_Page

Most Visited | Getting Started | Latest Headlines | http://www.google.... | Geekblog | Nyheder | http://www.google.... | dr.dk › open streami...

Log in / create account

article | discussion | edit | history

McXtrace

Main Page

McXtrace [edit]

McXtrace - Monte Carlo Xray ray-tracing is a joint venture by

Risø DTU DTU ESRF JJ X-RAY

Funding from NABIIT, DSF and the above parties.

Our code will be based on technology from McStas.

For information on our progress, please subscribe to our user mailinglist.

mailto:webmaster@mcxtrace.org

navigation
- Main Page
- Partners
- Project People
- Project Status
- Vacancies
- Project Goal
- Mailing List
- Links
- SMEXOS talks
- SRI09 abstracts

search

Go | Search

toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link

This page was last modified 13:15, 25 February 2009. This page has been accessed 2,049 times. Privacy policy About McXtraceWiki Disclaimers Powered By MediaWiki

Done

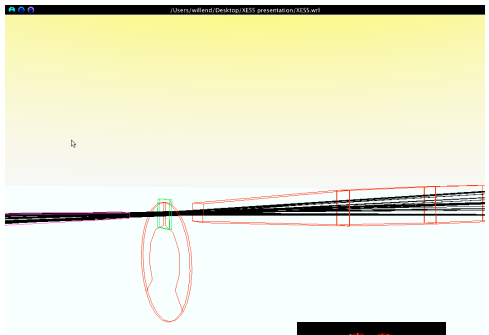
- *Synergy, knowledge transfer, shared infrastructure*

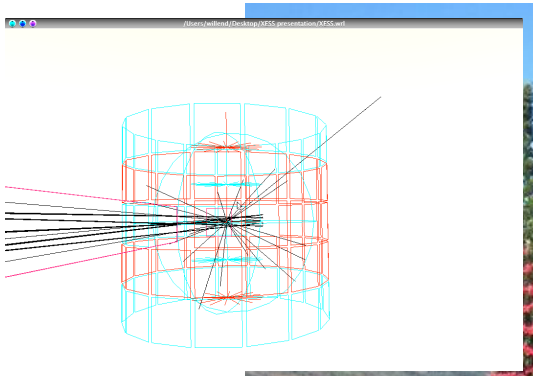
EUROPEAN SPALLATION SOURCE

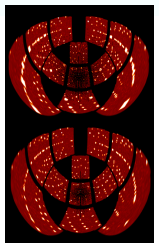
Used in many places

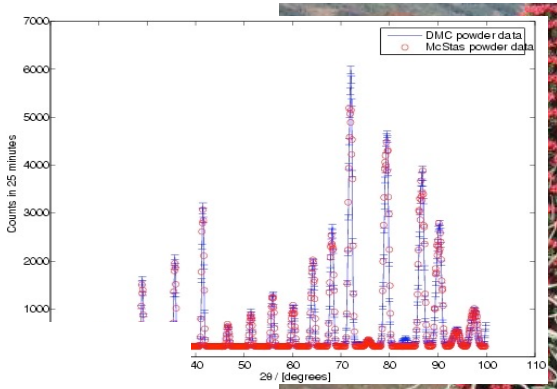
What is McStas used for?

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

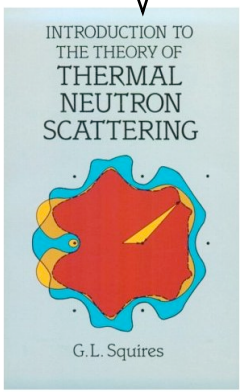


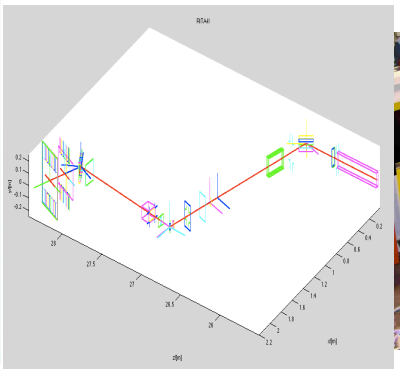


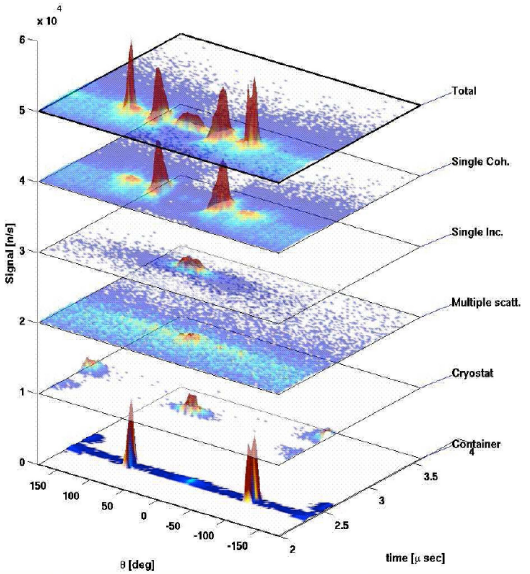




(KU, DTU)



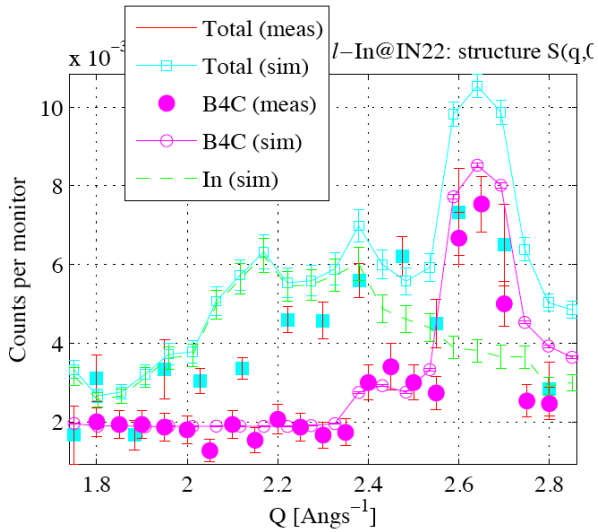




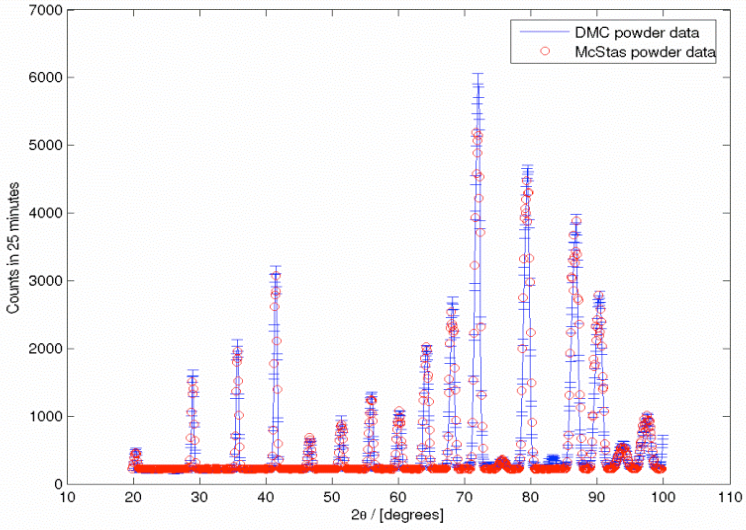
DTU ess EUROPEAN SPALLATION SOURCE

Reliability - cross comparisons

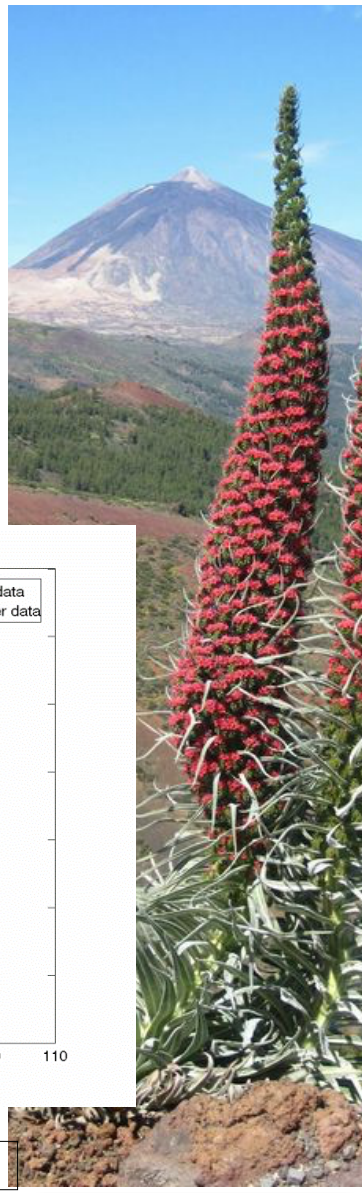
- *Much effort has gone into this*
- *Here: simulations vs. exp. at powder diffract. DMC, PSI*
- *The bottom line is*
- *McStas agree very well with other packages (NISP, Vitess, IDEAS, RESTRAX, ...)*
- *Experimental line shapes are within 5%*
- *Absolute intensities are within 10%*
- *Common understanding: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



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



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

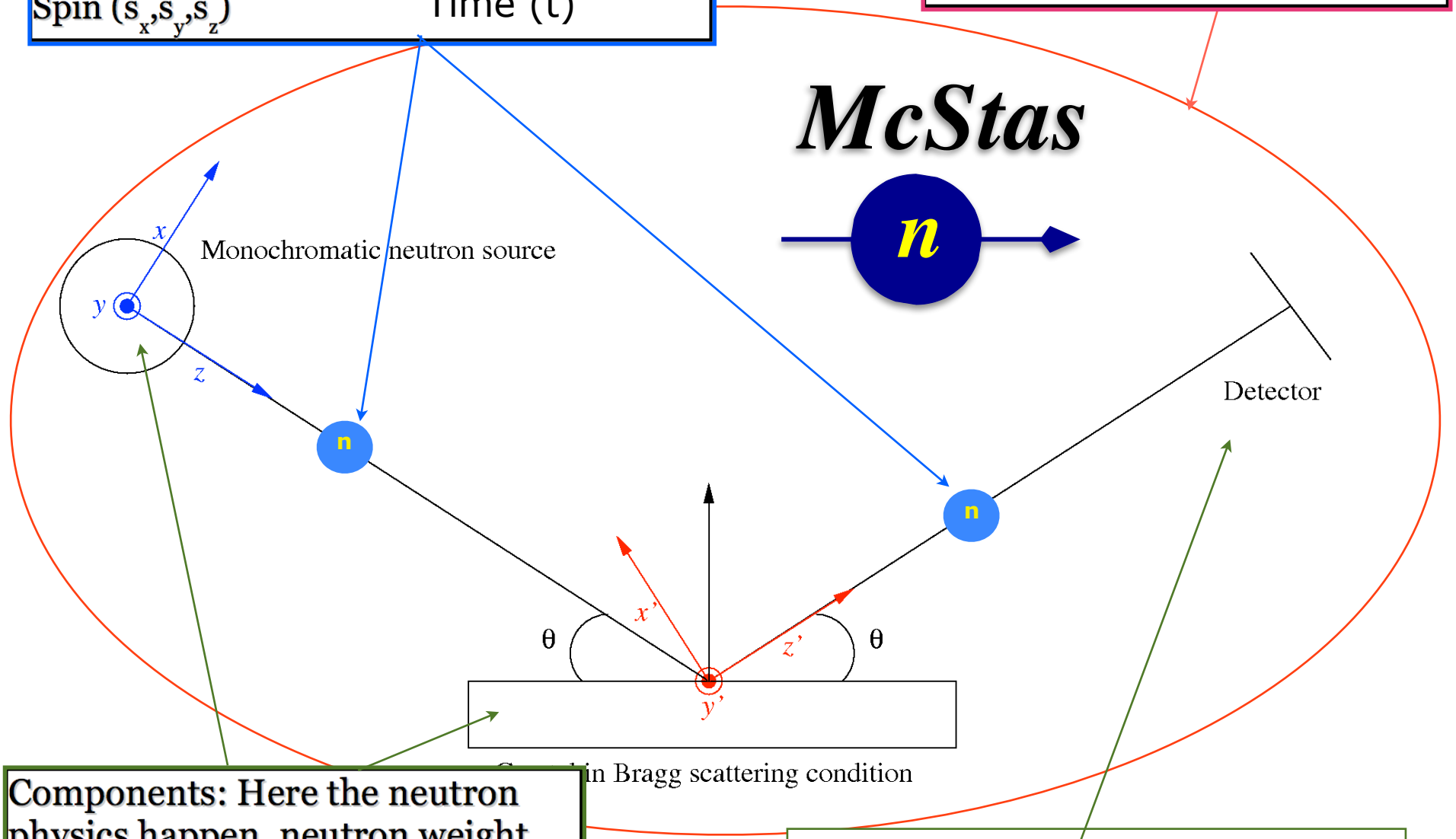


Neutron ray/package:

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)

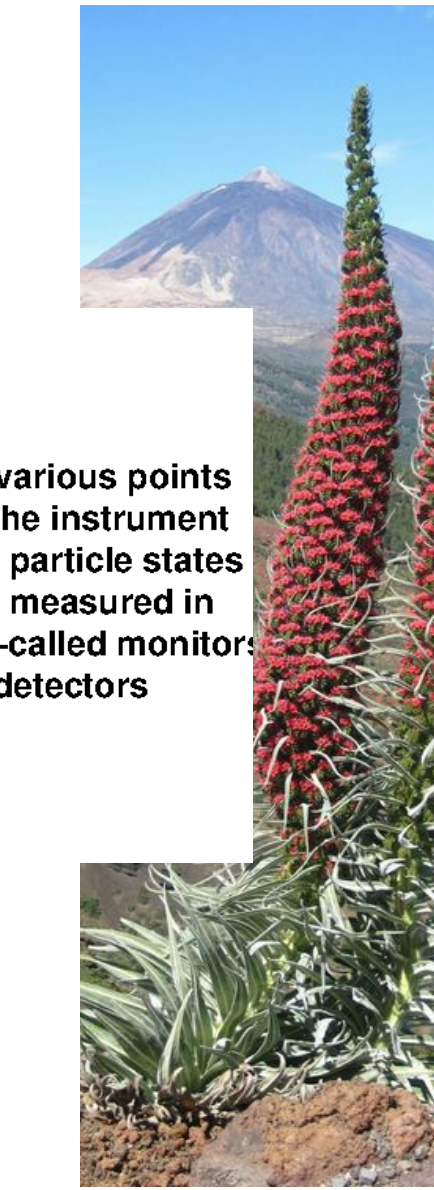
Instrument: positioning + transformation between sequential component coordinate systems, e.g. neutron source, crystal, detector.



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

Local, internal coordinate system!

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

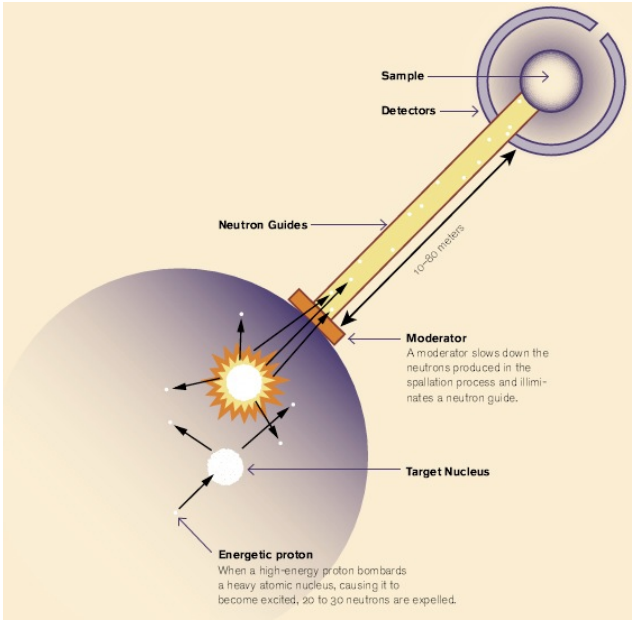
McStas overview

- Portable code (Unix/Linux/Mac/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

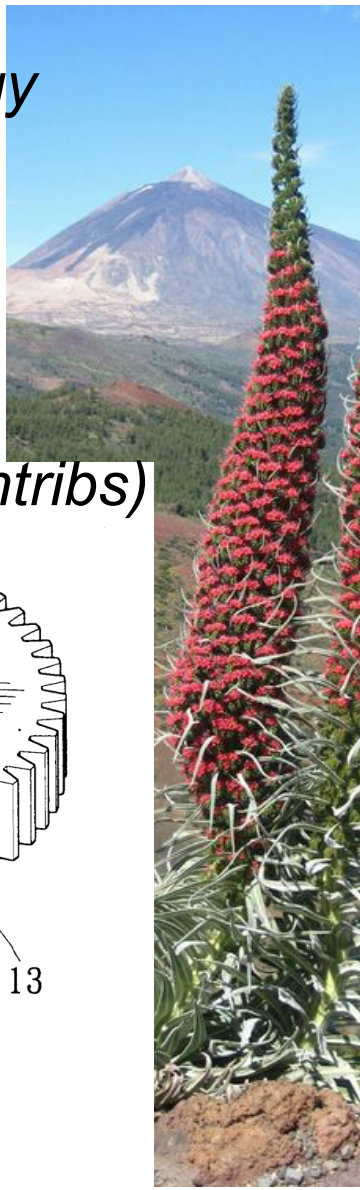
- DSL + ISO-C code gen.

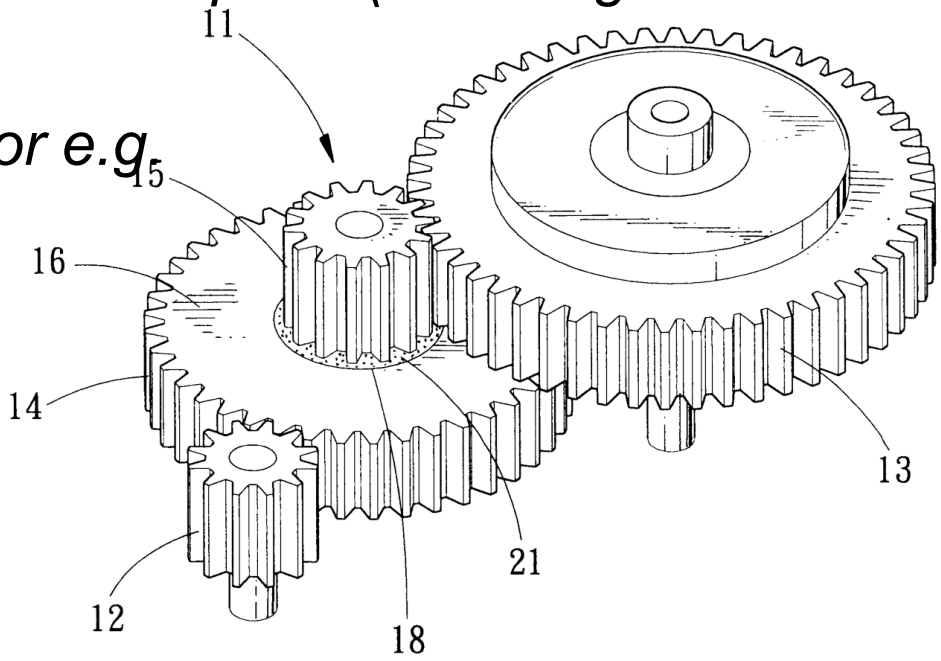


DTU ess EUROPEAN SPALLATION SOURCE

Under-the-hood / inner workings

- *Domain-specific-language (DSL) based on compiler technology (LeX+Yacc)*
  - *Simple Instrument language* —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

```
/* 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);
```

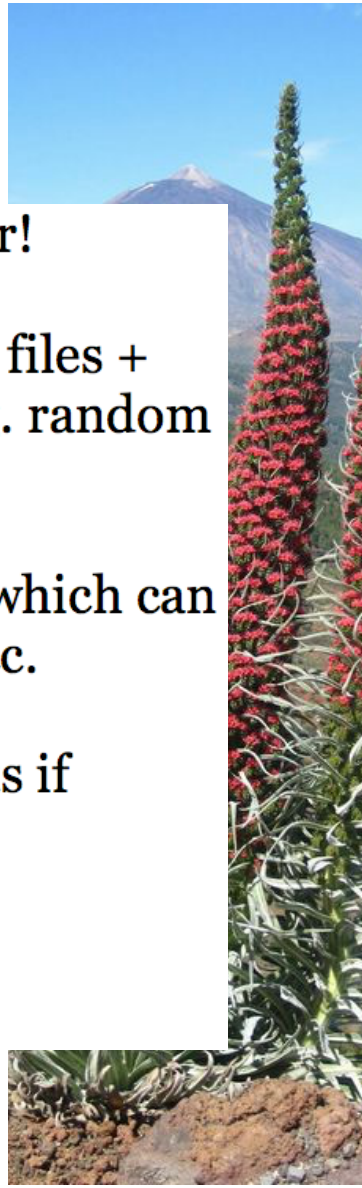
Written by mcstas!

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.

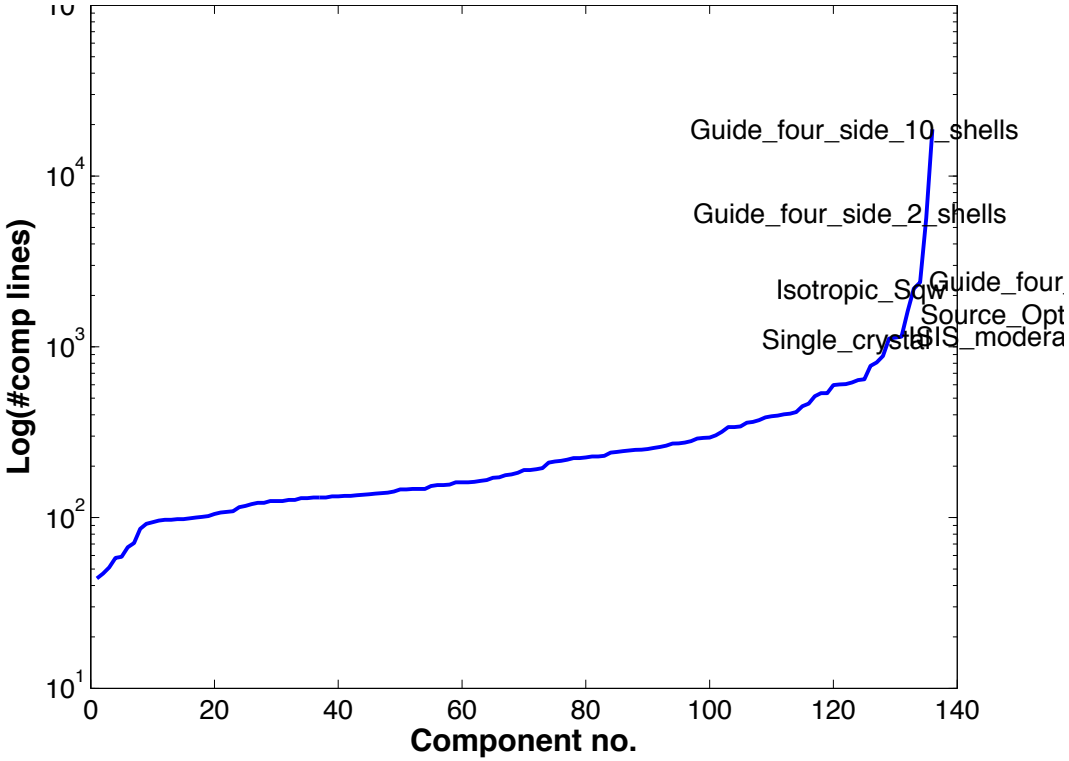
Can take input arguments if needed.

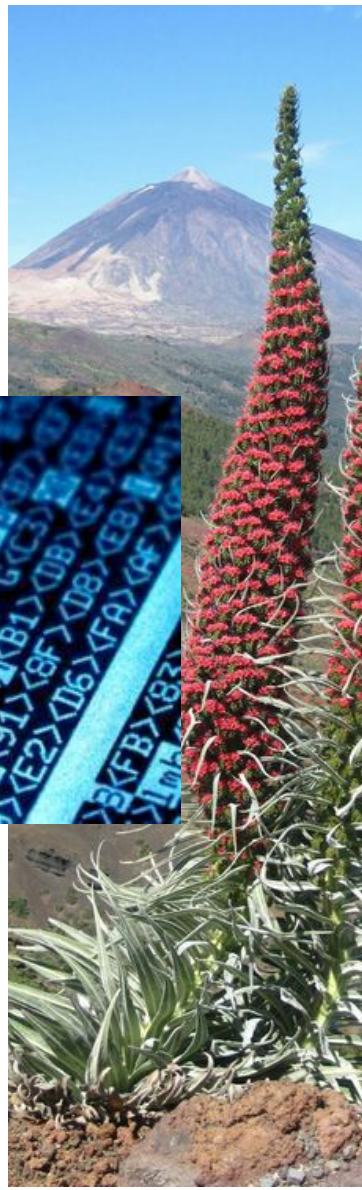


Writing new comps or understanding existing is not that complex...

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

Number of lines of code per component - 203 comps in total ;)



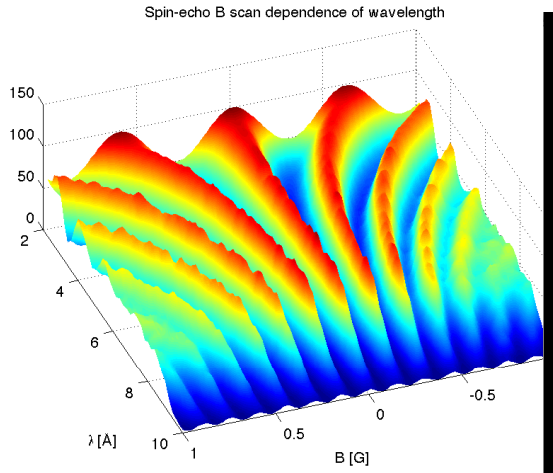


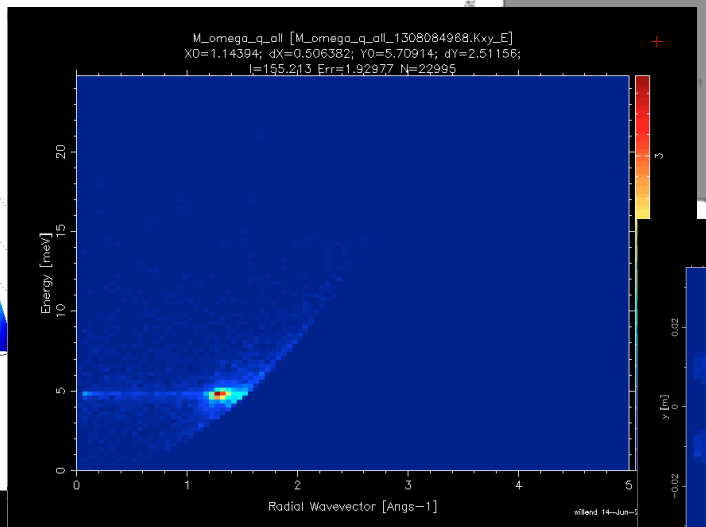
Including user contribs

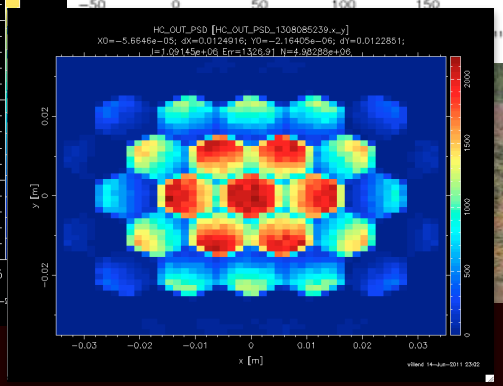
- *Well-developed community support*
  - *30-40% of existing and new additions are from users*
  - *No direct refereeing of the code, but these requirements:*
    - *At least one test-instrument*
    - *Meaningful documentation headers (in-code docs)*
  - *Contributions go in dedicated contrib/ section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

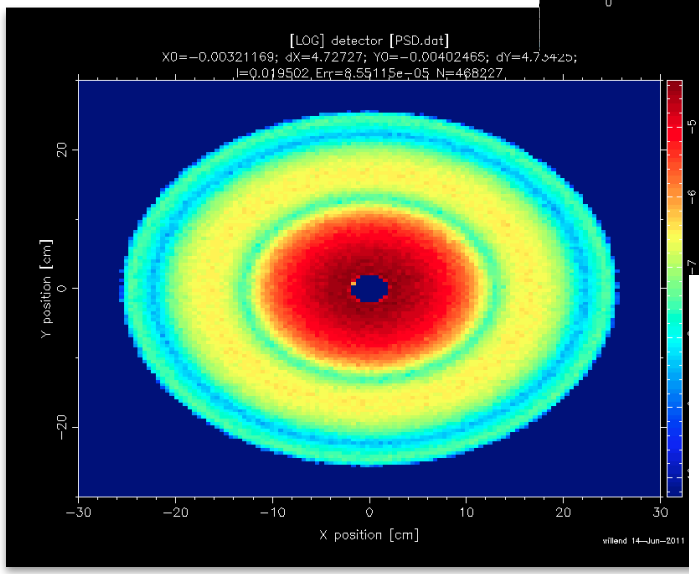
- *Natural life-cycle of contrib's*
  - *Bug-fixes are applied both by contributor and developers*
  - *If contributor becomes unavailable either:*
    - *Many users of comp: Promote to official components, e.g. in optics/*
    - *Few/no users of comp: Move to obsolete/ until next major release*

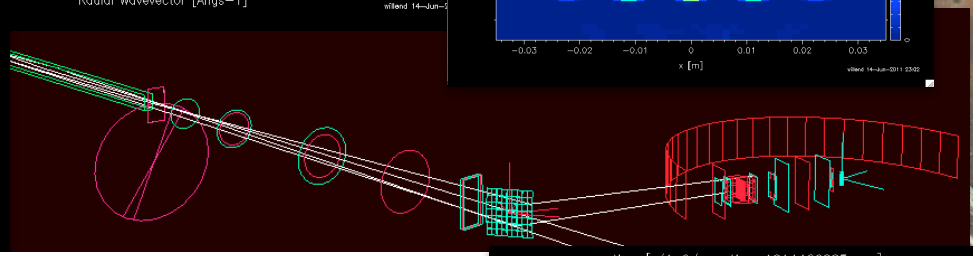
Example suite: 169 instruments

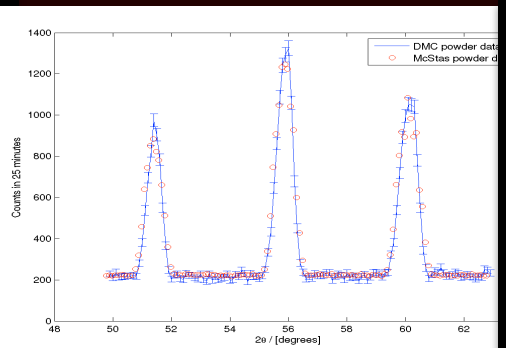


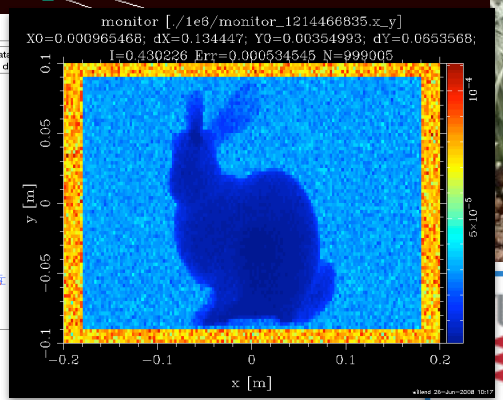






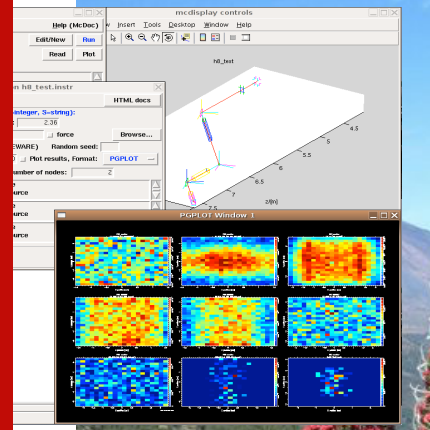






Feedback and help

- *Please:*
- *- Enroll to mcstas-users@mcstas.org*
- *- Post your question there or to mcstas-support@mcstas.org*
- *- There is no such thing as a stupid question!*
- *- Like us on Facebook? :)*



THIS IS NOT THE END

IT'S JUST THE BEGINNING

!!! DEMO TIME !!!

DTU ess EUROPEAN SPALLATION SOURCE