McStas for…

School of Neutron Scattering
Francesco Paolo Ricci

SoNS

Instrument Simulation
July 8th 8:30-10:30

Peter Willendrup,
DTU Physics & ESS DMSC

McStas

This project is funded by the European Union (H2020 GA no. 654000)

SINE 2020

This project is funded by the European Union (H2020 GA no. 654000)

brightness

Agenda

1)Who is your lecturer?

2)A (very) brief introduction to neutrons, MC & raytracing

3)Neutron instruments & components

4)How McStas “works” + is under the hood

5)A demo

6)Modelling sample environments in McStas

7)Comparing with experiments & what to keep in mind when simulating…

8)How to get to learn more / Pointers to self-education

Agenda

1) Who is your lecturer?

2) A (very) brief introduction to neutrons, MC & raytracing

3) Neutron instruments & components

4) How McStas "works" + is under the hood

5) A demo

6) Modelling sample environments in McStas

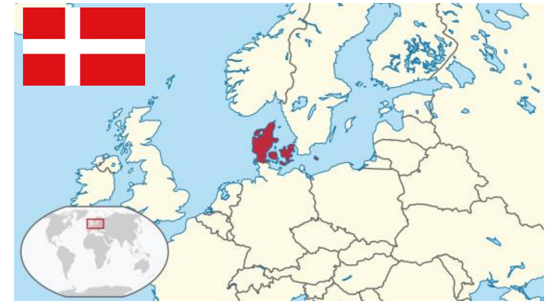
Break somewhere around here…

7) Comparing with experiments & what to keep in mind when simulating…

8) How to get to learn more / Pointers to self-education

About the lecturer

- Born 1973 in Copenhagen, DK



- BSc. in Physics - RISØ / Univ. CPH 1997 *"Neutron diffraction and magnetic structures" (Ho-Er alloys)*

- Master-courses in X-ray scattering, atomic physics, solid state physics, "computer physics", numerical analysis

- MSc. in Physics, BSc. in Mathematics from Univ. CPH year 2000
  "Point-spread Functions in Tomography using Filtered Back-projection Reconstruction" (CT/PET/SPECT scanners)

- Software solutions for Neurobiology Research Unit, Copenhagen University Hospital 2000-2002
  3D brain-scan visualisation, alignment MR-PET etc…

Neurobiology Research Unit
Rigshospitalet
Copenhagen University Hospital

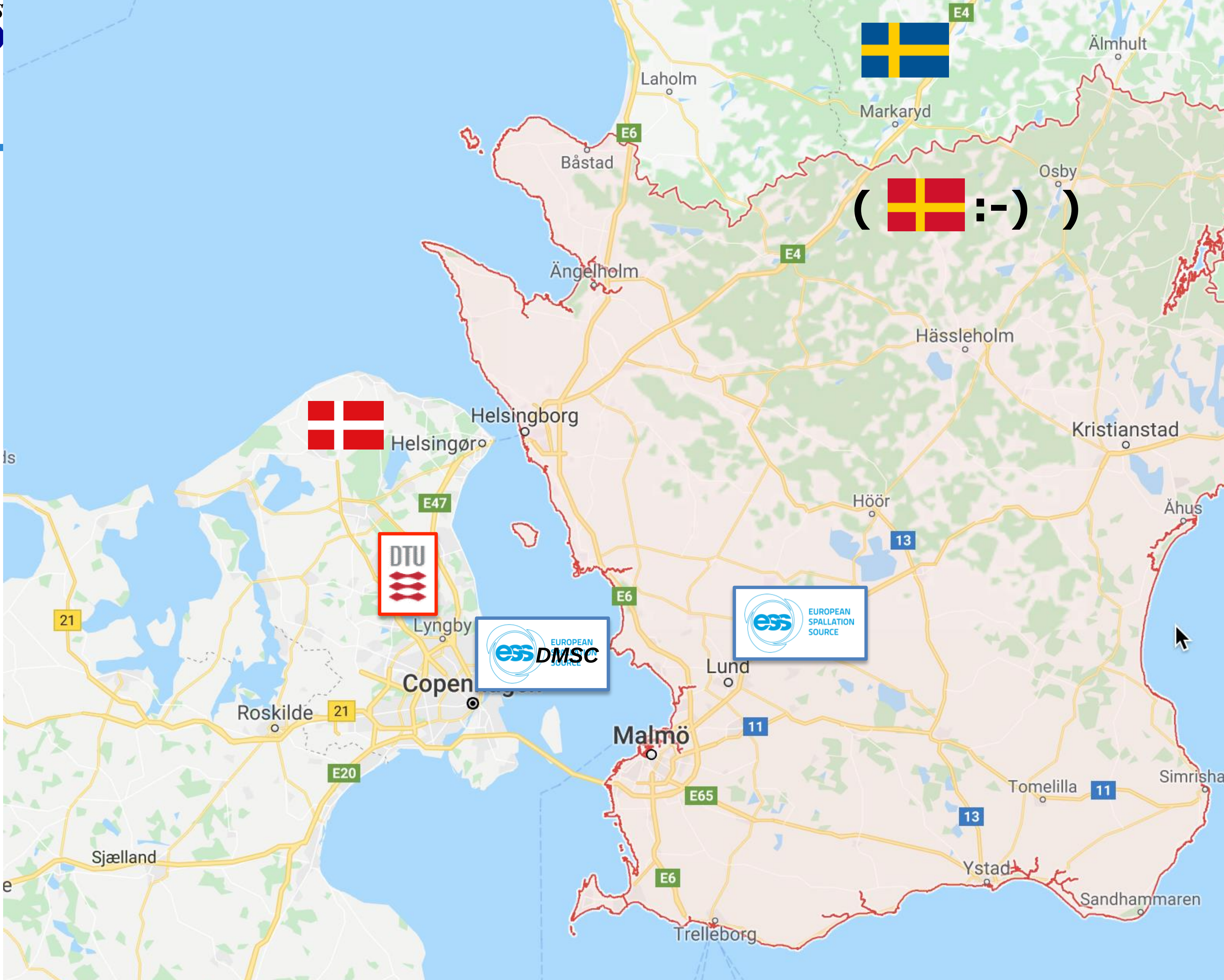
DTU
EUROPEAN SPALLATION SOURCE

About the lecturer

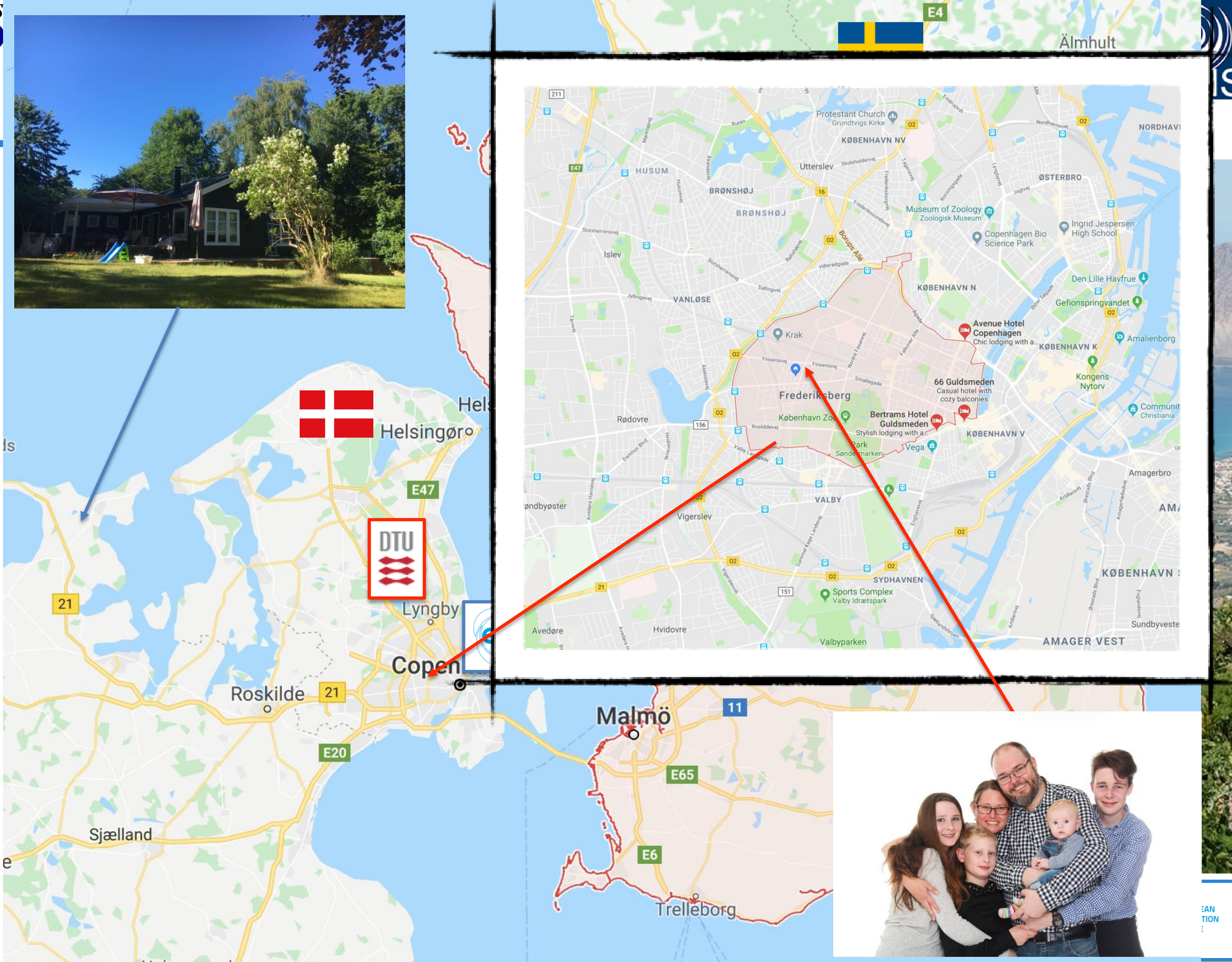
- *2002- “development engineer” on the McStas project*
  - *2002-2007 Risø National Lab*
  - *2007-2011 Risø DTU*
  - *2012- DTU Physics*
  - *2015- 1/3 seconded to ESS DMSC*

- *External funding from EU projects, ISIS TS2 project, ESS project etc. Currently ESS project + SINE2020*

- *Daily tasks wrt. McStas:*
  - *Develop new functionality - be it physics or infrastructure-wise*
  - *Work with users to solve their problems (in any area of neutron scattering…)*
  - *Software expert for “anything McStas”*
  - *Teach users about the code and how to use it efficiently*


Laholm
Markaryd
Älmhult
Båstad
Osby
(:-))
Ängelholm
Hässleholm
Helsingborg
Helsingør
Kristianstad
Höör
Åhus
DTU
Lyngby
EUROPEAN SPALLATION SOURCE
DMSC
Lund
Roskilde
Malmö
Tomelilla
Simrishamn
Sjælland
Ystad
Sandhammaren
Trelleborg
E4
E6
E47
E20
E65
21
11
13

McS



Älmhult
E4
Helsingør
E47
DTU
Lyngby
Copen
Roskilde
21
E20
Sjælland
Malmö
11
E65
E6
Trelleborg
HUSUM
BRØNSHØJ
Utterslev
KØBENHAVN NV
Protestant Church
Grundtvigs Kirke
ØSTERBRO
NORDHAVN
Museum of Zoology
Zoologisk Museum
Ingrid Jespersen
High School
Copenhagen Bio
Science Park
Islev
KØBENHAVN N
Den Lille Havfrue
Gefionspringvandet
VANLØSE
Avenue Hotel
Copenhagen
Chic lodging with a...
Amalienborg
KØBENHAVN K
Krak
Frederiksberg
66 Guldsmeden
Casual hotel with
cozy balconies
Kongens
Nytorv
Rødovre
København Zoo
Bertrams Hotel
Guldsmeden
Stylish lodging with a...
KØBENHAVN V
Vega
Amagerbro
VALBY
Vigerslev
KØBENHAVN S
SYDHAVNEN
Sports Complex
Valby Idrætspark
Avedøre
Hvidovre
Valbyparken
AMAGER VEST
Sundbyvester

The Neutron

Life time:	$\tau_{1/2} = 890s$
Mass:	$m = 1.675 \times 10^{-27} kg$
Charge:	$Q = 0$
Spin:	$s = \hbar/2$
Magnetic moment:	$\mu/\mu_n = -1.913$

$$E = \frac{1}{2} m v^2 = \frac{\hbar^2 k^2}{2m}, \qquad \lambda = 2\pi/k$$

$$E = 81.81 \cdot \lambda^{-2} = 2.07 \cdot k^2 = 5.23 \cdot v^2$$

Mr. Neutron

	Energy	Wavelength	n-Wavevector	Velocity	Frequency
cold neutrons:	E = 1 meV	l = 9.0446 Å	k = 0.6947 1/Å	v = 437 m/s	v = 0.2418 THz
	E = 5 meV	l = 4.0449 Å	k = 1.5534 1/Å	v = 978 m/s	v = 1.2090 THz
thermal	E = 25 meV	l = 1.8089 Å	k = 3.4734 1/Å	v = 2187 m/s	v = 6.045 THz
neutrons:	E = 50 meV	l = 1.2791 Å	k = 4.9122 1/Å	v = 3093 m/s	v = 12.090 THz

Cross section: coherent + incoherent + absorption

The Neutron

Life time:	$\tau_{1/2} = 890s$
Mass:	$m = 1.675 \times 10^{-27} kg$
Charge:	$Q = 0$
Spin:	$s = \hbar/2$
Magnetic moment:	$\mu/\mu_n = -1.913$

$$E = \frac{1}{2}mv^2 = \frac{\hbar^2 k^2}{2m}, \qquad \lambda = 2\pi/k$$

$$E = 81.81 \cdot \lambda^{-2} = 2.07 \cdot k^2 = 5.23 \cdot v^2$$

Mr. Neutron

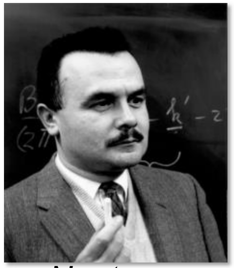
Slow enough to be manipulated via mechanical devices

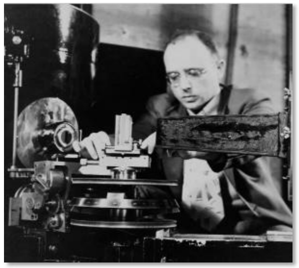
	Energy	Wavelength	n-Wavevector	Velocity	Frequency
cold neutrons:	E = 1 meV	l = 9.0446 Å	k = 0.6947 1/Å	v = 437 m/s	v = 0.2418 THz
	E = 5 meV	l = 4.0449 Å	k = 1.5534 1/Å	v = 978 m/s	v = 1.2090 THz
thermal	E = 25 meV	l = 1.8089 Å	k = 3.4734 1/Å	v = 2187 m/s	v = 6.045 THz
neutrons:	E = 50 meV	l = 1.2791 Å	k = 4.9122 1/Å	v = 3093 m/s	v = 12.090 THz

Cross section: coherent + incoherent + absorption

1994 Nobel prize to Shull & Brockhouse

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

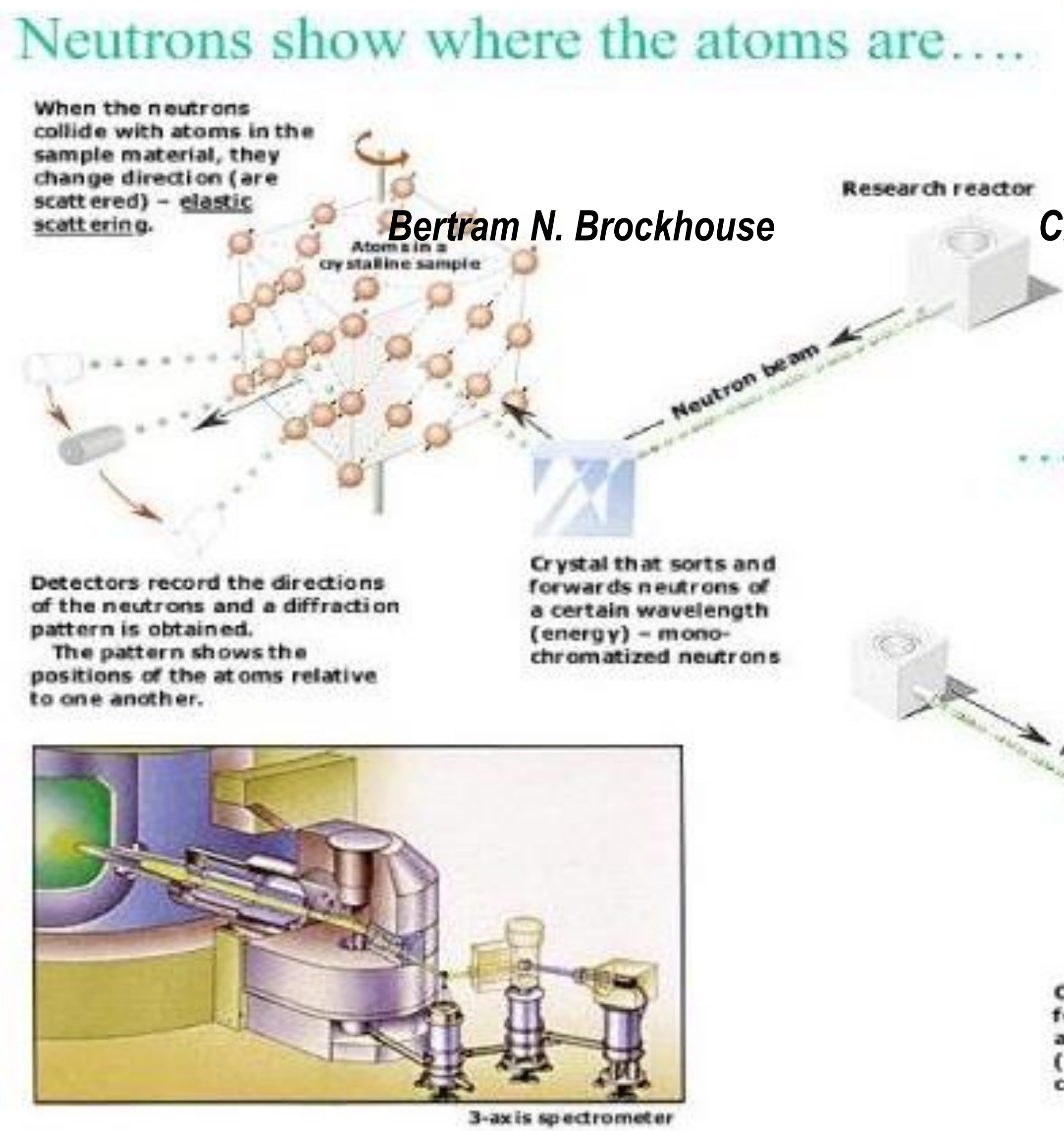


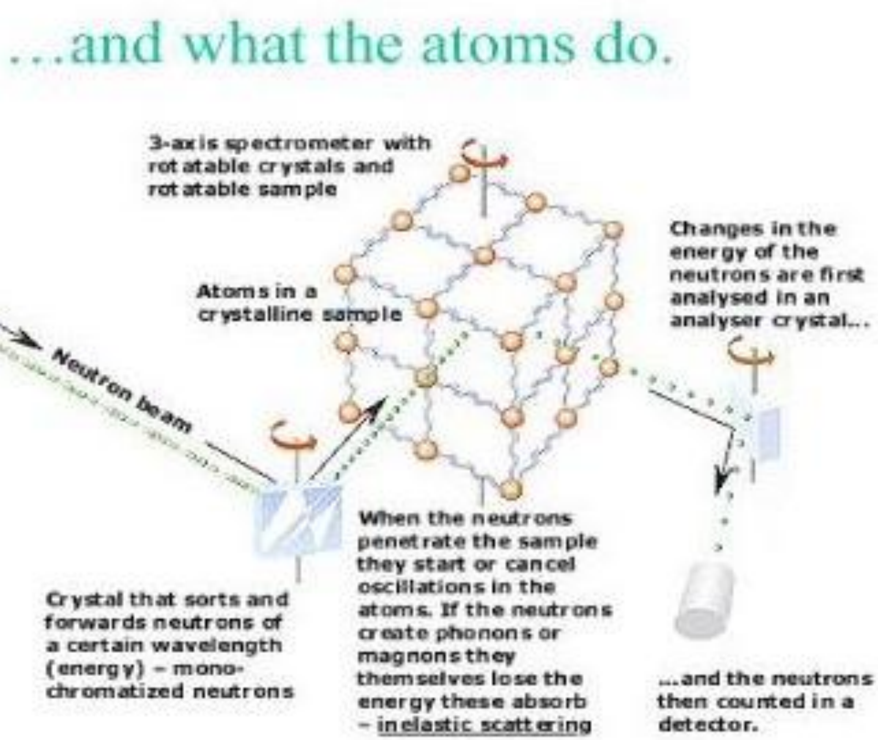


Neutron spectroscopy

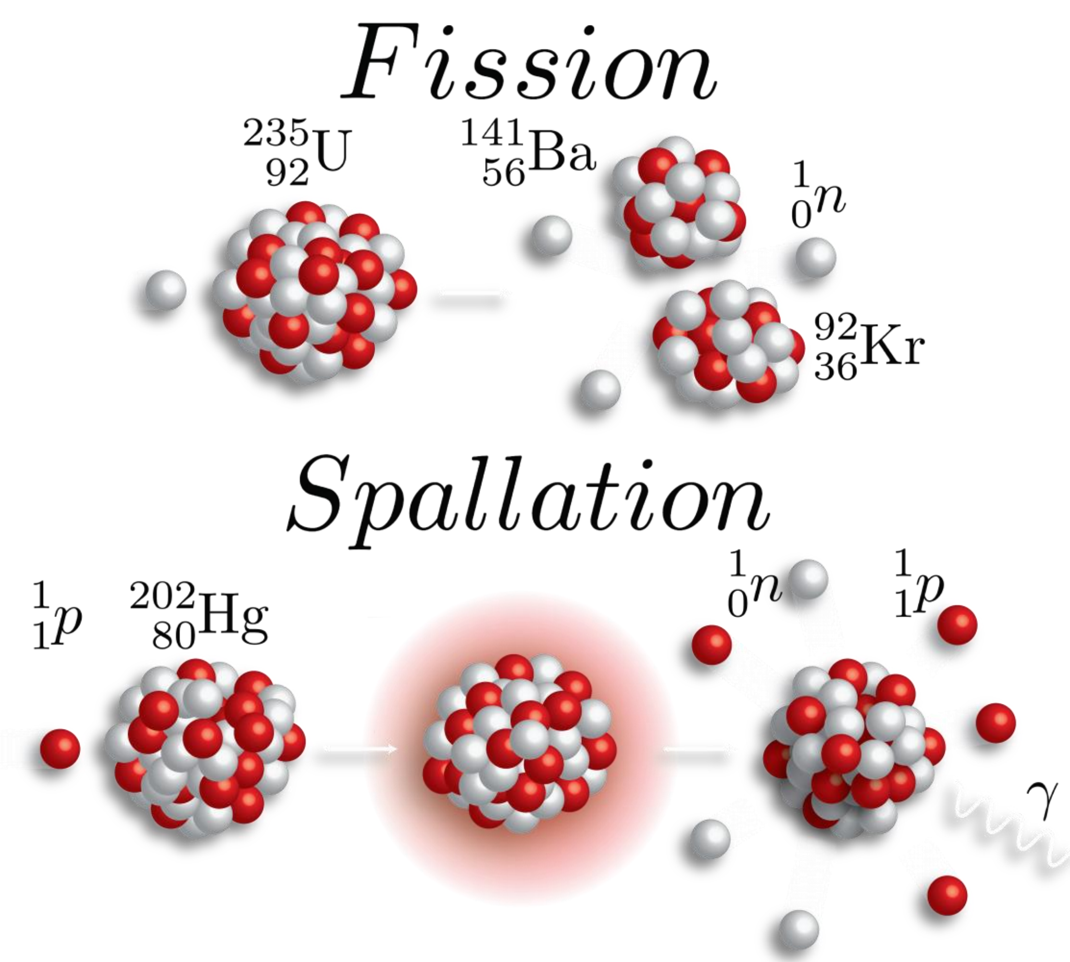
Neutron diffraction

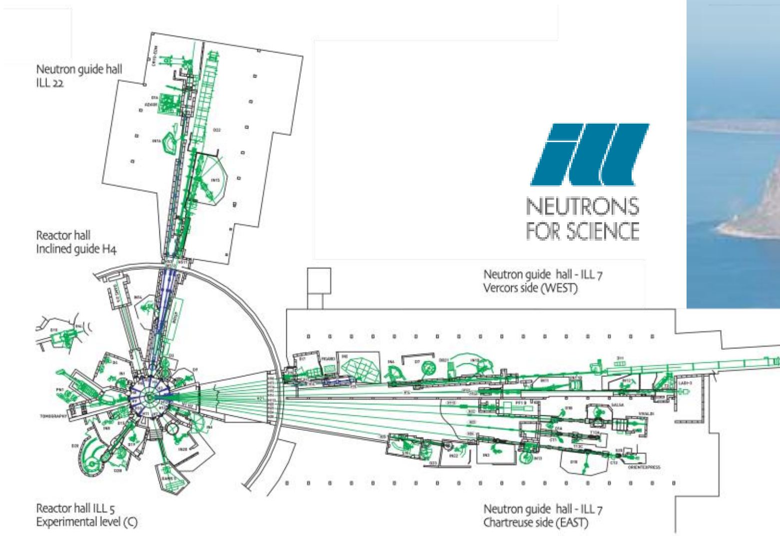
Bertram N. Brockhouse **Clifford G. Shull**

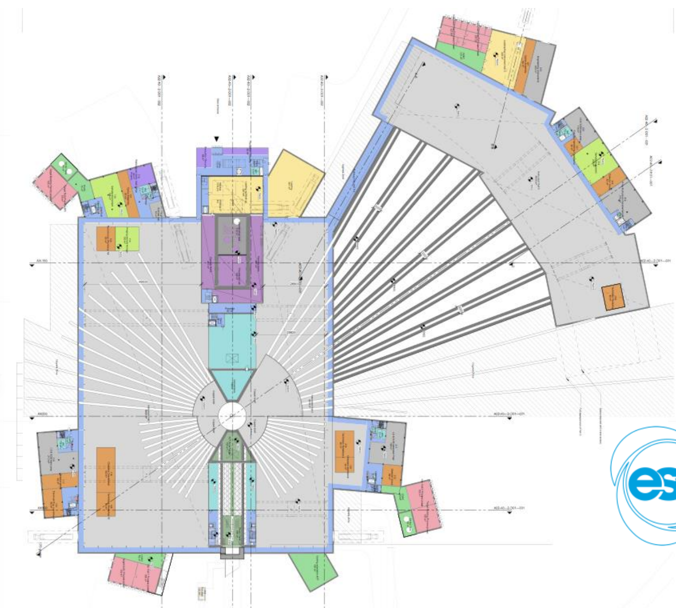


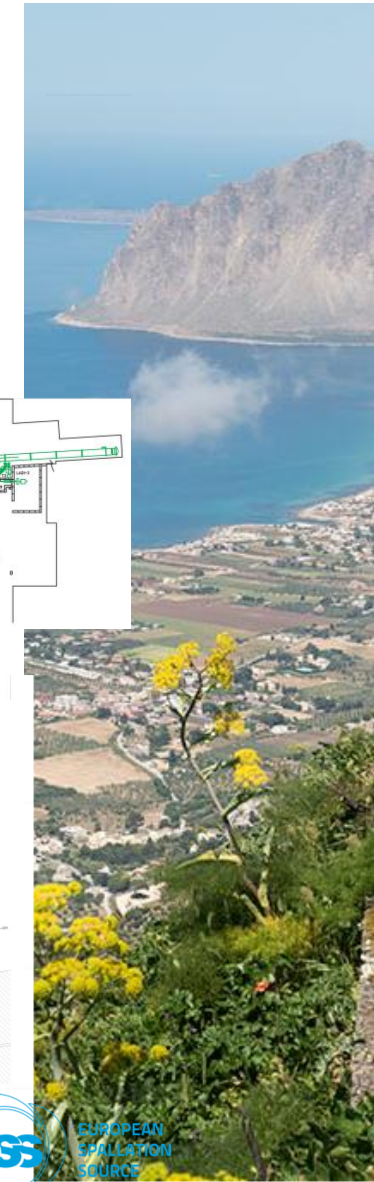


Neutron facilities





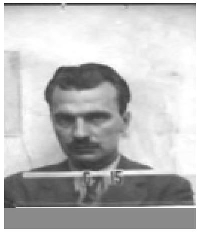




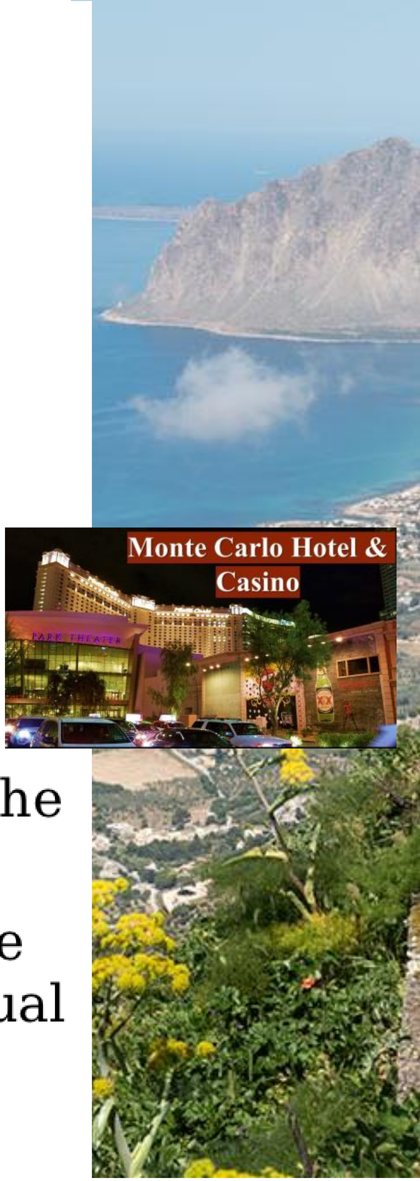
Monte Carlo Technique

- During WW2, “numerical experiments” were applied at Los Alamos for solving mathematical complications of computing fission, criticality, neutronics, hydrodynamics, thermonuclear detonation etc.

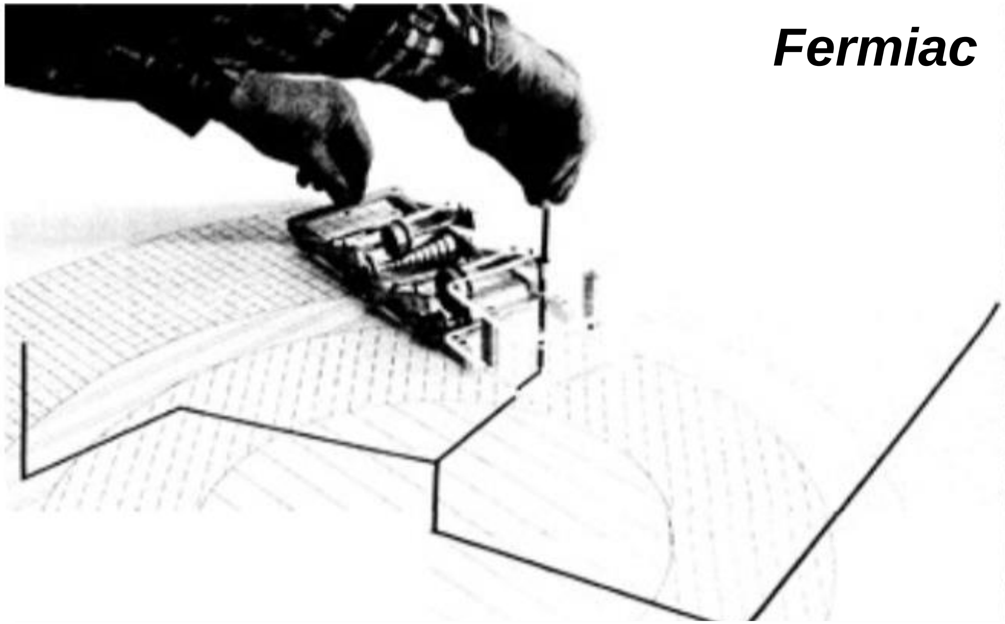


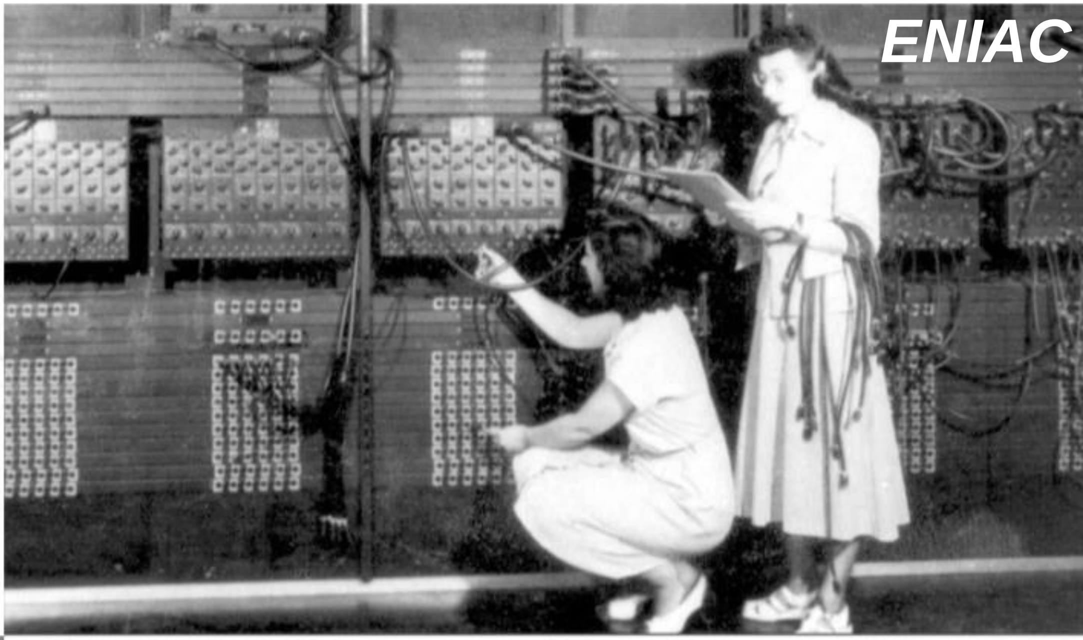


- Notable fathers: John v. Neumann, Stanislav Ulam, Nicholas Metropolis
- Named “Monte Carlo” after Ulam’s fathers frequent visits to the Monte Carlo casino in Las Vegas
- Initially “implemented” by letting large numbers of women use tabularized random numbers and hand calculators for individual particle calculations
- Later, analogue and digital computing devices were used



Early Monte Carlo





4. – How the Fermiac works

The *Fermiac*, as shown in fig. 3, mainly consists of three parts:

1. The *lucite platform*, that serves as a neutron direction selector.
2. The *rear drum*, that measures the elapsed time based on the velocity of the particular neutron in question.
3. The *front drum*, that measures the distance traveled by the neutron between subsequent collisions based on the neutron velocity and the properties of the material being traversed.

Before operating the *Fermiac*, you need to make a scale drawing of the nuclear device under exam; this is obtained by projecting on a plane the concentric sections of the different materials, as shown in fig. 4. Then you need to decide an initial collection of source neutrons (at the T-Division usually 100 neutrons were taken), and for each one you need to determine the location of the first collision or possible escape. This is achieved by statistical considerations on the characteristics of the type of material being traversed. You also need to establish the nature of the collision of each neutron: elastic, inelastic scattering or fission (if the material allows), and the distance to the next collision. Then you can operate the Fermiac to follow the fate of each neutron.

ENIAC (/ˈiːniæk, ˈɛ-/; **Electronic Numerical Integrator and Computer**)[1][2] was amongst the earliest electronic general-purpose computers made. It was Turing-complete, digital and able to solve "a large class of numerical problems" through reprogramming.[3][4]

Although ENIAC was designed and primarily used to calculate artillery firing tables for the United States Army's Ballistic Research Laboratory,[5][6] its first programs included a study of the feasibility of the thermonuclear weapon.[7] In 1992, the Ballistic Research Laboratory became a part of the US Army Research Laboratory.

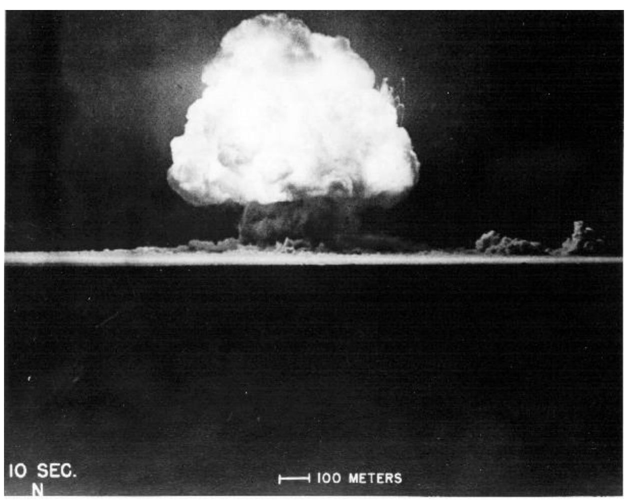
ENIAC was completed in 1945 and first put to work for practical purposes at the end of that year.[8]

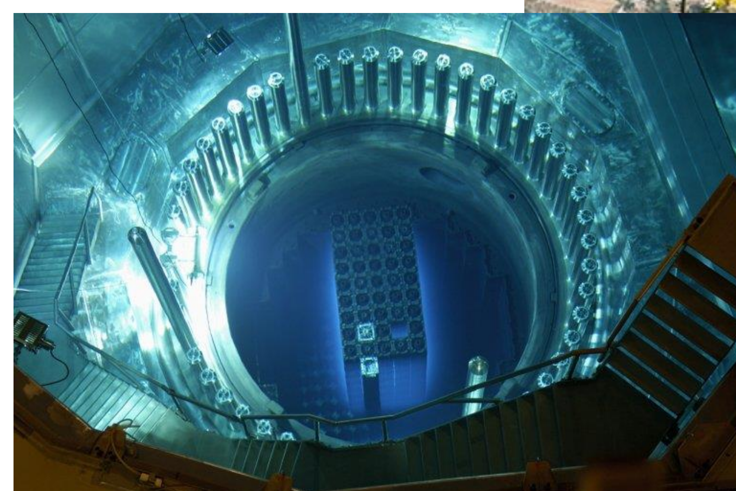
ENIAC was formally dedicated at the University of Pennsylvania on February 15, 1946 and was heralded as a "Giant Brain" by the press.[9] It had a speed on the order of one thousand times faster than that of electro-mechanical machines; this computational power, coupled with general-purpose programmability, excited scientists and industrialists alike. The combination of speed and programmability allowed for thousands more calculations for problems, as ENIAC calculated a trajectory in 30 seconds that took a human 20 hours (allowing one ENIAC hour to displace 2400 human hours).[10]

https://issuu.com/coccetti/docs/fermiac_coccetti_2016

Monte Carlo Technique

- *Los Alamos has since then developed and perfected many different monte carlo codes leading to what is today known as the codes MCNP5, MCNPX, MCNP6*
- *State of the art is MCNPX (or the merged MCNP6.x 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 - materials are to first order all “gasses”*

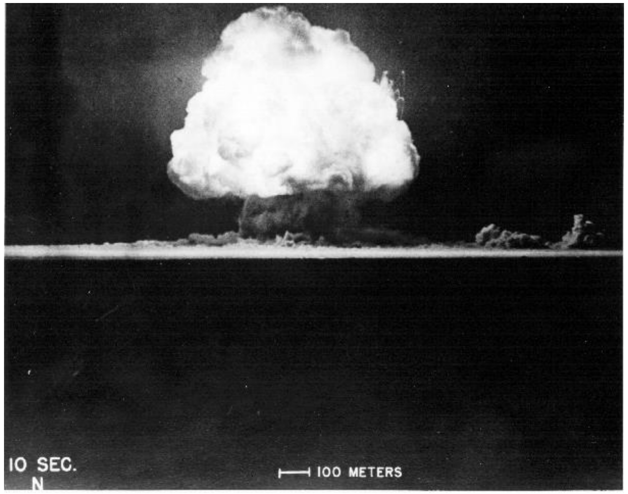


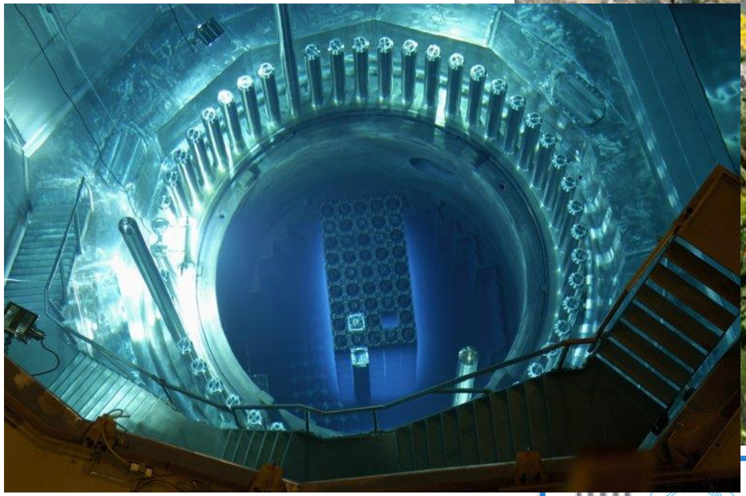


Monte Carlo Technique

- *Los Alamos has since then developed and perfected many different monte carlo codes leading to what is today known as the codes MCNP5, MCNPX, MCNP6*
- *State of the art is MCNPX (or the merged MCNP6.x 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 - materials are to first order all "gasses"*

MCNP6





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:

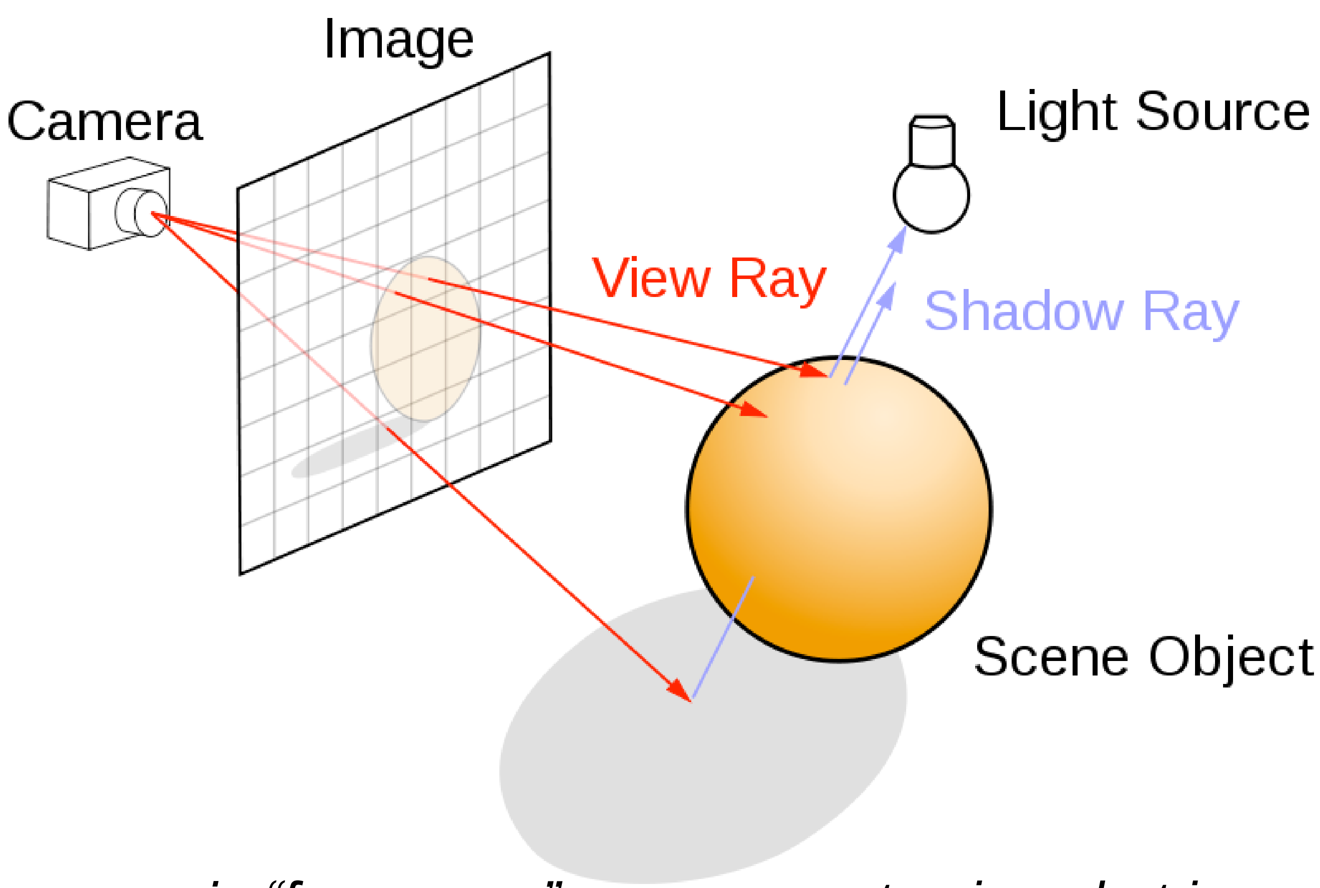
McStas <www.mcstas.org>, Vitess, Restrax, NISP, IDEAS
Neutrons are described as ***(r, v, s, t)***, and are transported along instrument models.

Propagation simply uses Newton rules, incl. gravitation.

X-ray tracing

Shadow, McXtrace, RAY, …

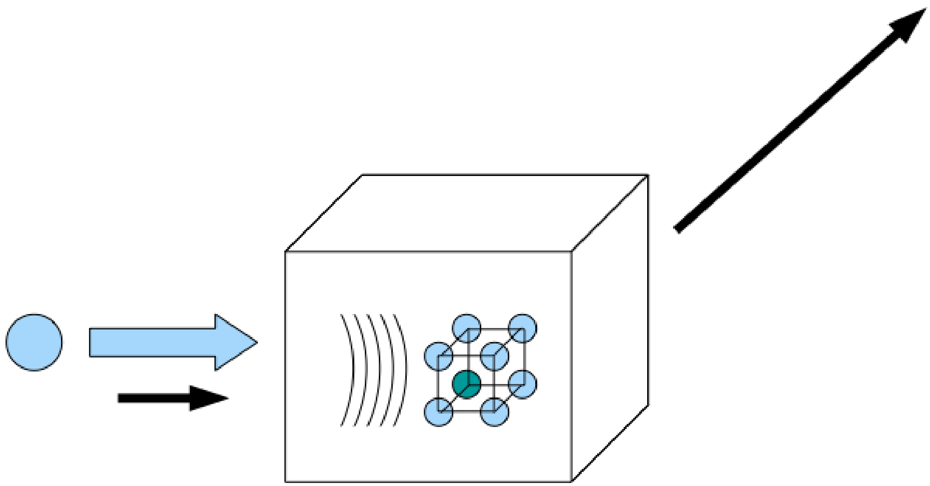
Ray-tracing methods



- *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*

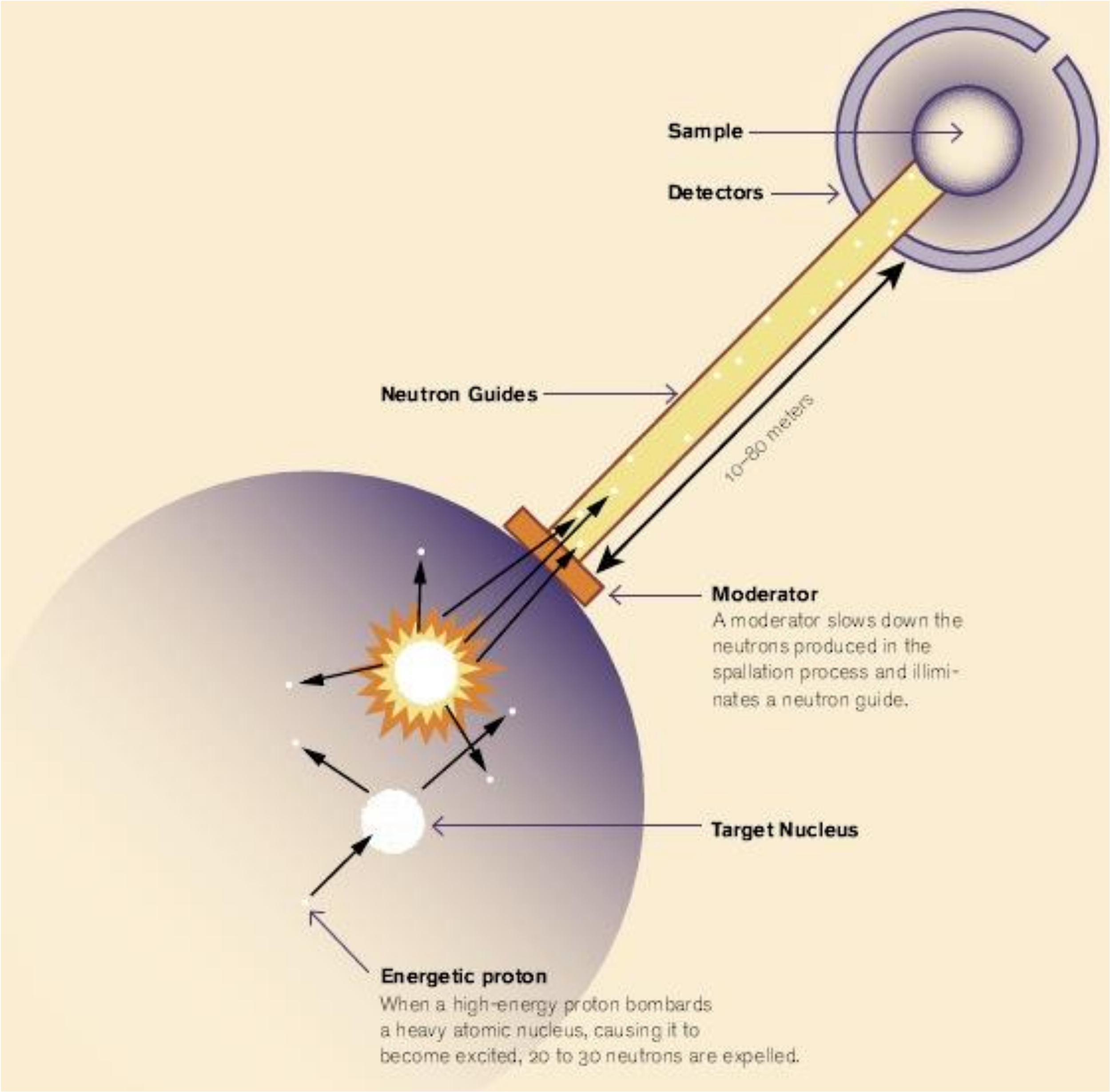
Elements of Monte-Carlo raytracing

- *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*

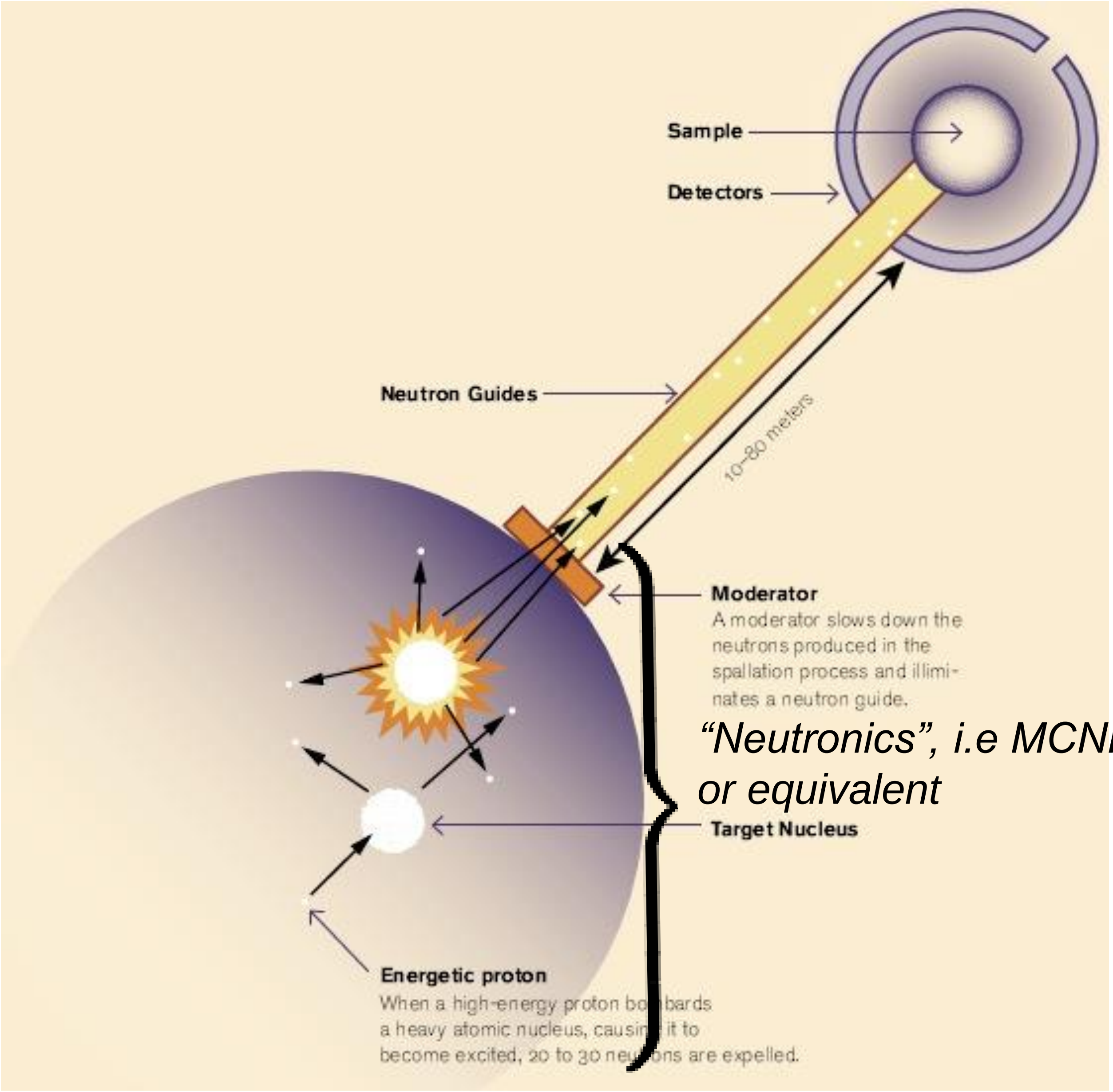


- *Result: A realistic and efficient transport of neutrons in the thermal and cold range*

Neutron instruments & components

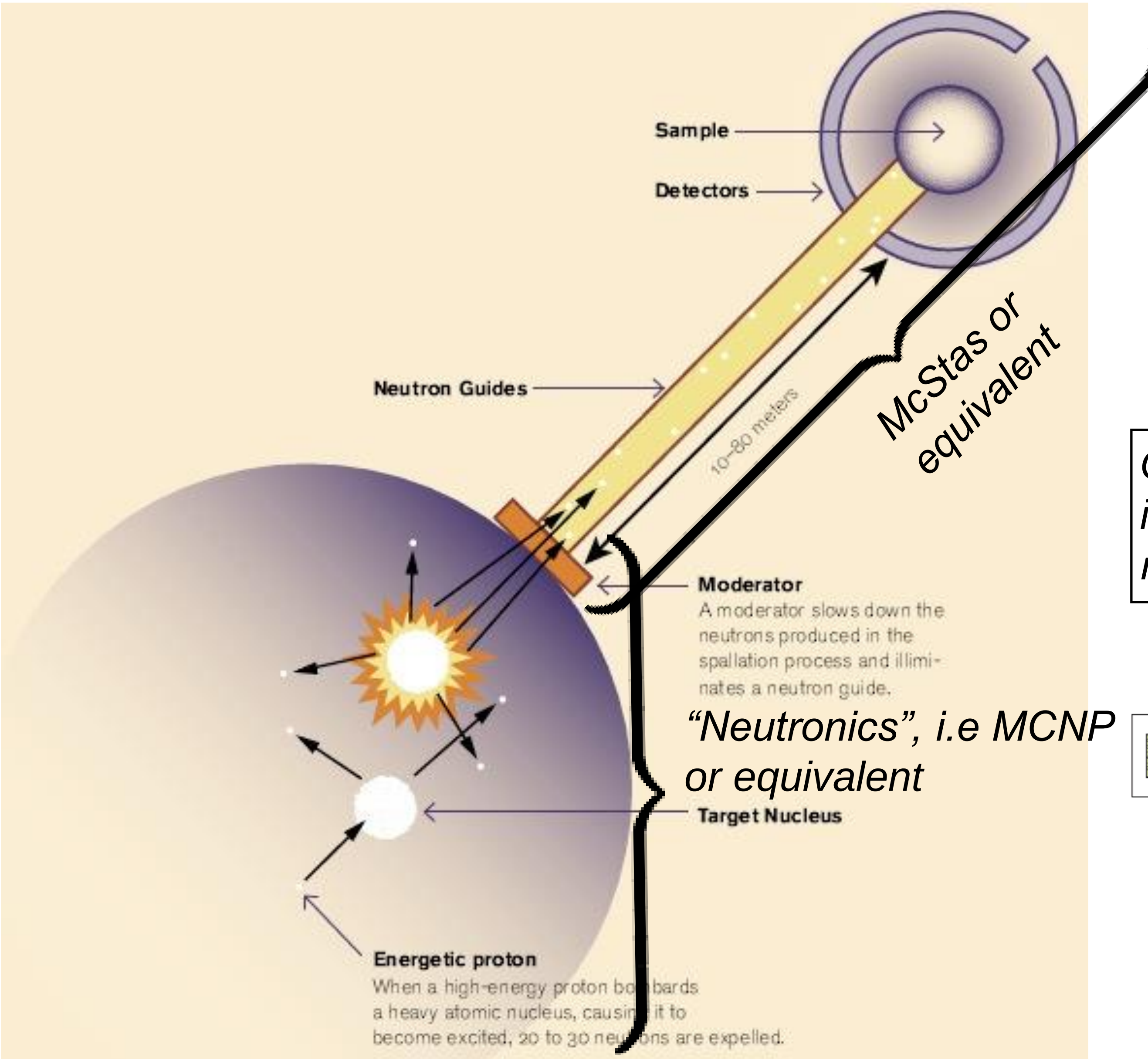


Neutron instruments & components



"Neutronics", i.e MCNP or equivalent

Neutron instruments & components



McStas

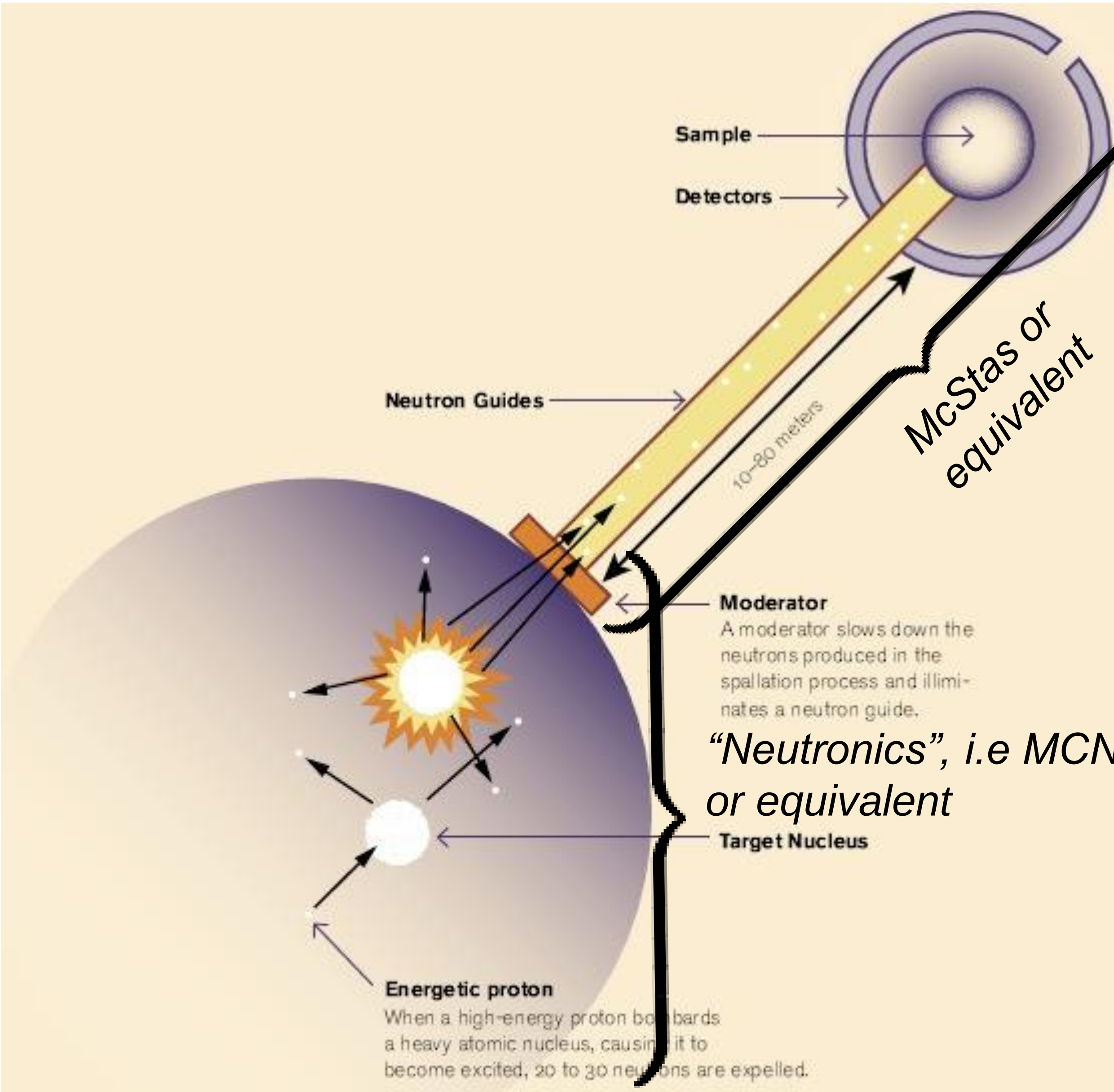
n

Cross-over point is the neutron moderator…

MCNP6

DTU
EUROPEAN SPALLATION SOURCE

Neutron instruments & components

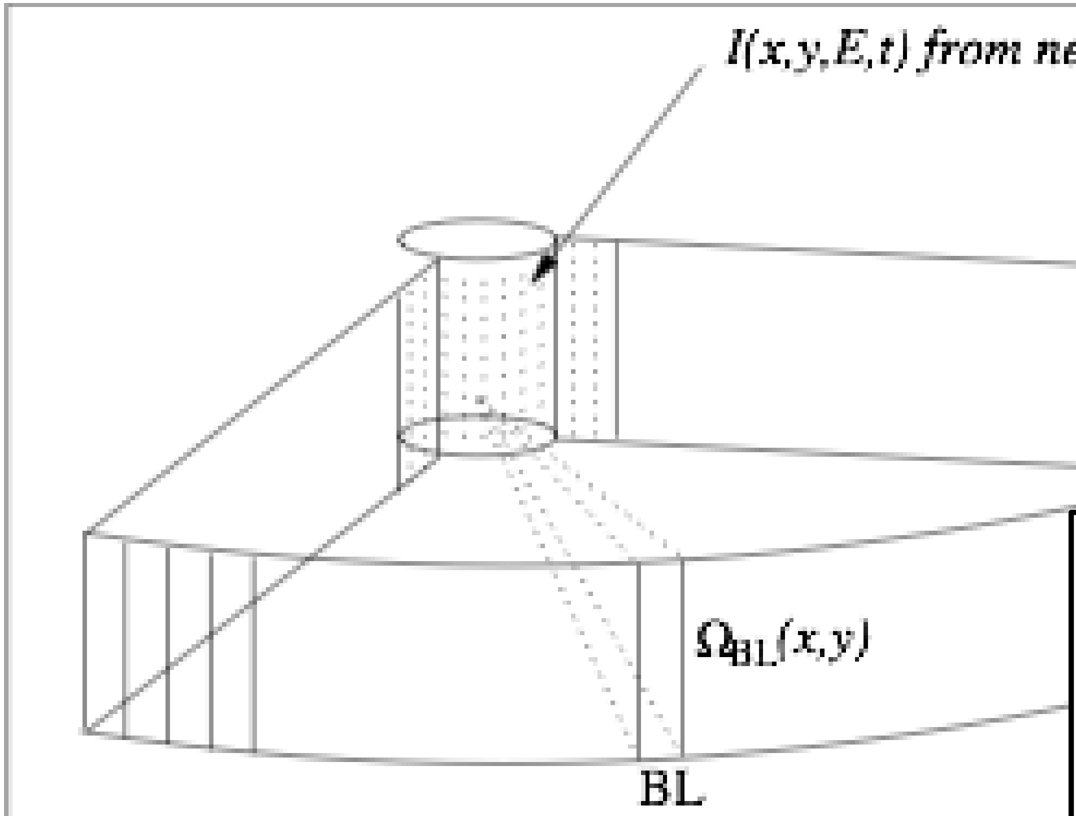


McStas

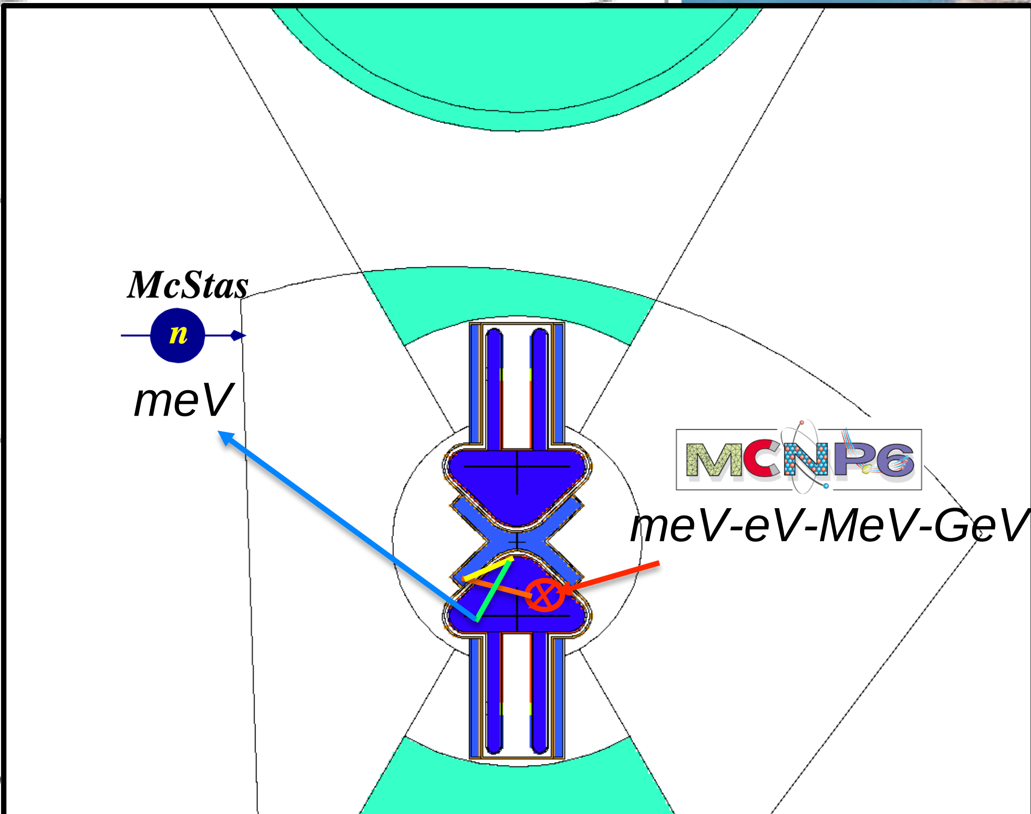
Cross-over point is the neutron moderator…

MCNP6

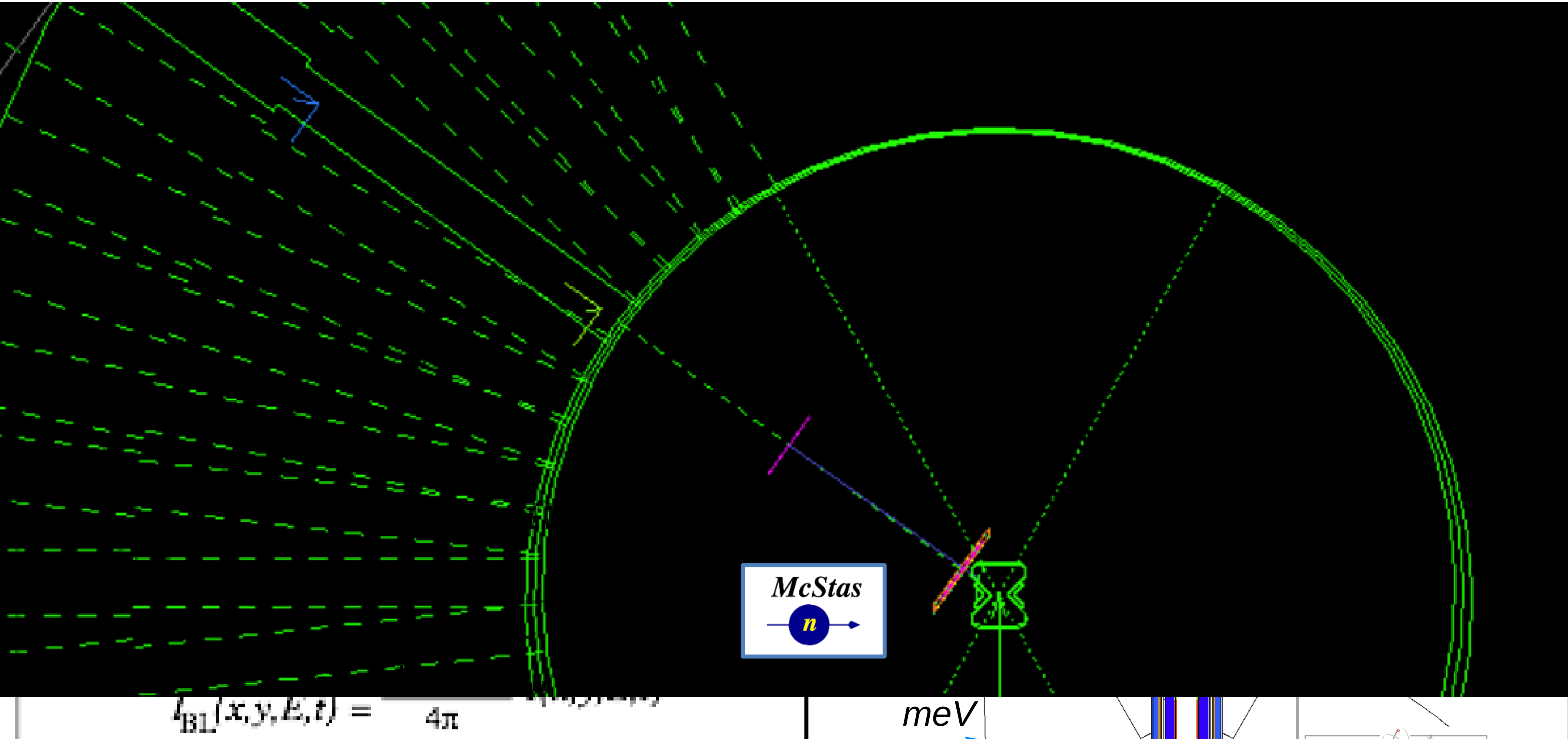
Moderators... (Where McStas starts)

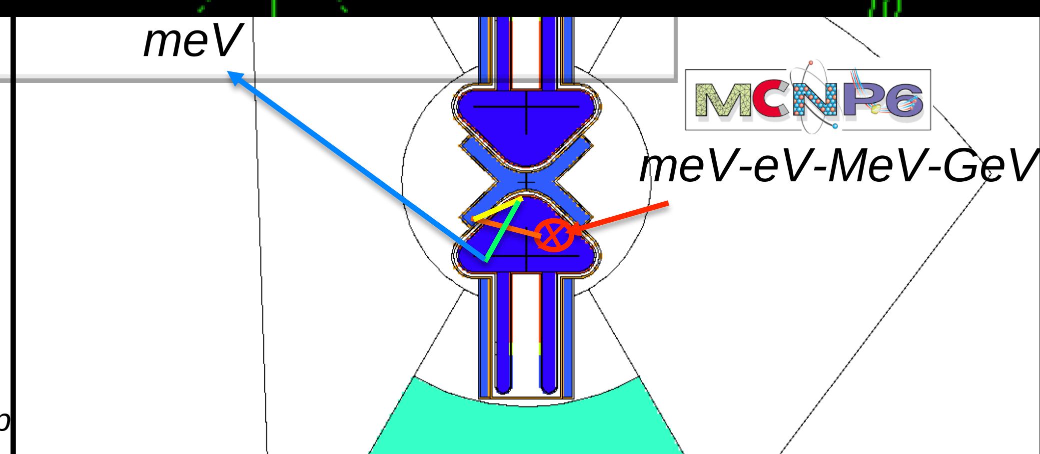


$$I_{BL}(x,y,E,t) = \frac{\Omega_{BL}(x,y)}{4\pi} I(x,y,E,t)$$



Moderators... (Where McStas starts)

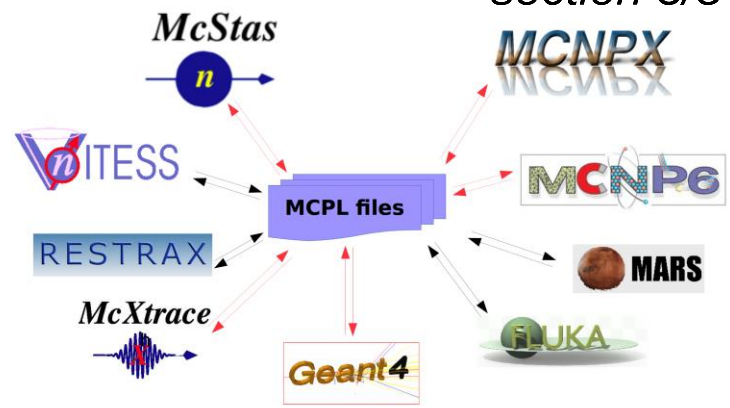




“Normal” sources in McStas

- Describe the moderator(s) ONLY
- Describe only thermalised neutrons
- Uses analytical curves to express spectra, emission profiles etc., typically modelled using e.g. MCNP
- Advantages:
  - **Fast** runtime (ESS_butterfly instrument w. guide transports 1e7 neutron rays in 11 seconds on 4 cores)
  - No stat/bias issues generating further events
  - Excellent “match” for what the neutron optics can actually transport!
- Disadvantages:
  - The above issues in lacking description of high-energy particles and non-moderator particles

What is MCPL?



- *MCPL is short for Monte Carlo Particle List*
- *Interchange-file-format for Geant4, MCNP(x), McStas etc…*
- *Created by T Kittelmann, ESS Detector Group, input from DTU, ESS etc.*
- *GitHub site: https://mctools.github.io/mcpl/*
- *Paper: [T. Kittelmann et. al. Computer Physics Communications (218) pp. 17-42]()*
- *Talk on the subject here:*
- *http://coimbra2016.essworkshop.org/slides/MCPL_Kittelmann.pdf*

This project is funded by the European Union (H2020 GA no. 654000)

This project is funded by the European Union (H2020 GA no. 654000)

MCPL Monte Carlo Particle Lists

MCPL file as a McStas source

- *Describe any particle reaching area near the beam port, arriving from anywhere*
- *Describe all neutrons, any energy - but also gammas etc…*
- *Implemented using the SSW card in MCPN in combination with a dxtran sphere “around” beam port*
- *Advantages:*
  - *Describes anything MCNP generates, including thermalised neutrons off the reflector…*
  - *Could be used in attempt to model signal-to-noise @ sample, with combined McStas+MCNP transport…*
- *Disadvantages:*
  - *Stat/bias issues generating further events (simple repetition brings nothing new)*
  - *Files are Gb size as a starting point*
  - *In comparison with “normal” McStas source* ***slow***

MCNP6

SINE2020 brightness

MCPL Monte Carlo Particle Lists

DTU EUROPEAN SPALLATION SOURCE

Example use of MCPL for ESS

- In the file, a Dirac-delta (time) proton beam on target, 1e5 protons “NPS”.
  - I.e. McStas component/instrument includes normalisation factors
  - 1/1e5 protons/simulation -> neutrons / incoming proton
  - 1.56e16 protons/s -> neutrons / second (default McStas intensity units)
  - MC choice on time within pulse length

- ESS butterfly 1 design “current model”
- Simulations run for “the expected day-1 instruments”
- “dxtran sphere” used to illuminate beam port

- Size ~ 3-5Gb / beam line
- Includes ~ 1e8 particles in total
- - out of these ~50% are neutrons
- Once reduced to “transportable neutrons” ~ 1e7 events
- Takes 5-10 minutes to process through instr w/guide if not filtered in any way…

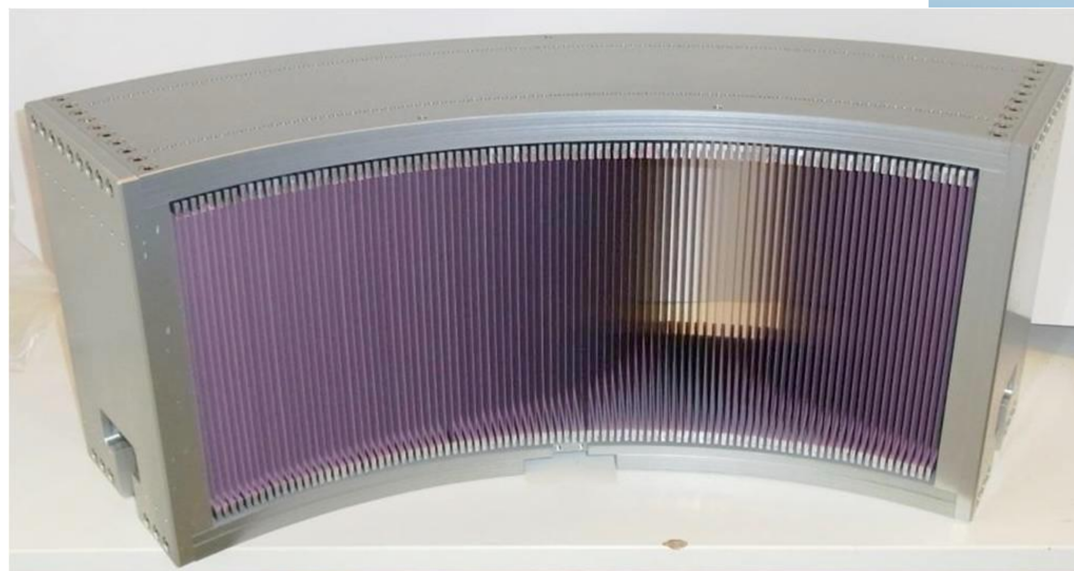
MCPL Monte Carlo Particle Lists

Neutron guides

Providing transport to the experiment area with few losses

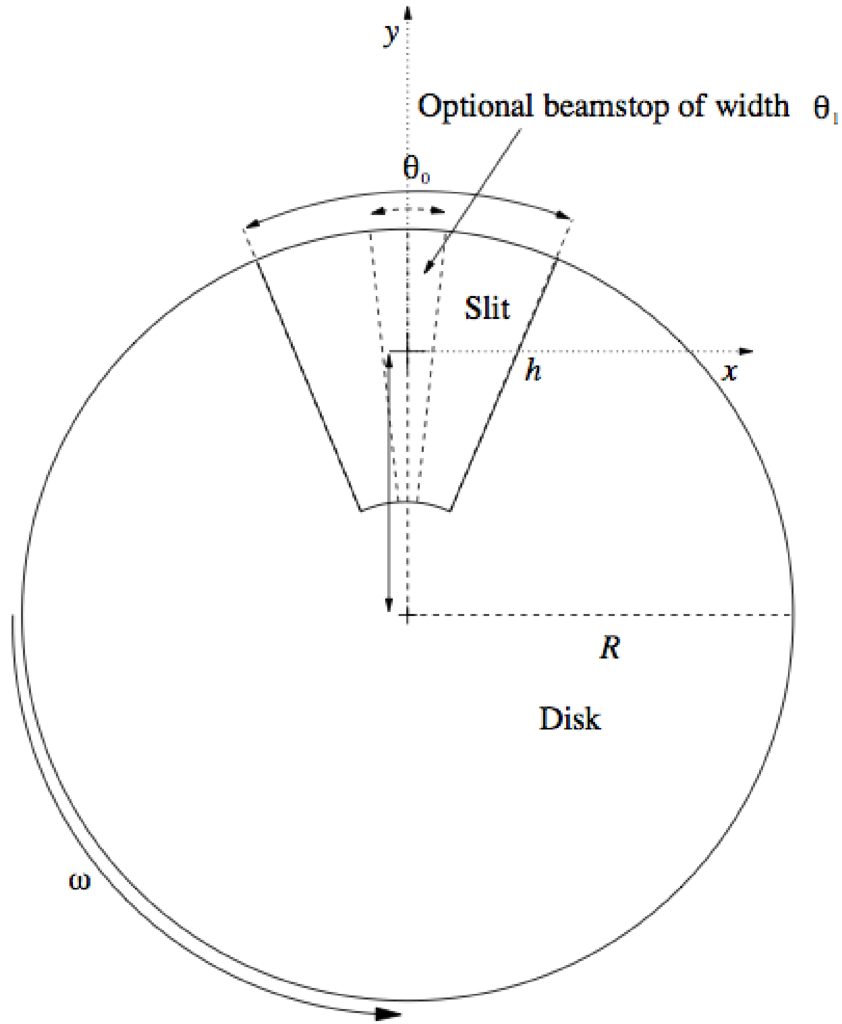
(external total reflection…)

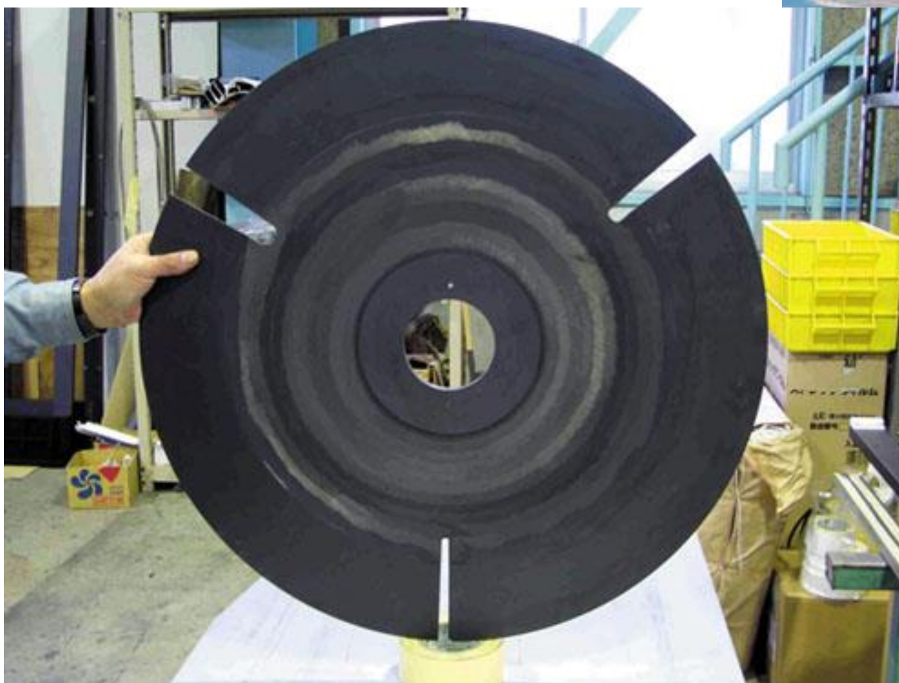
Collimators & slits



Allows to manipulate the beam cross-section and divergence…

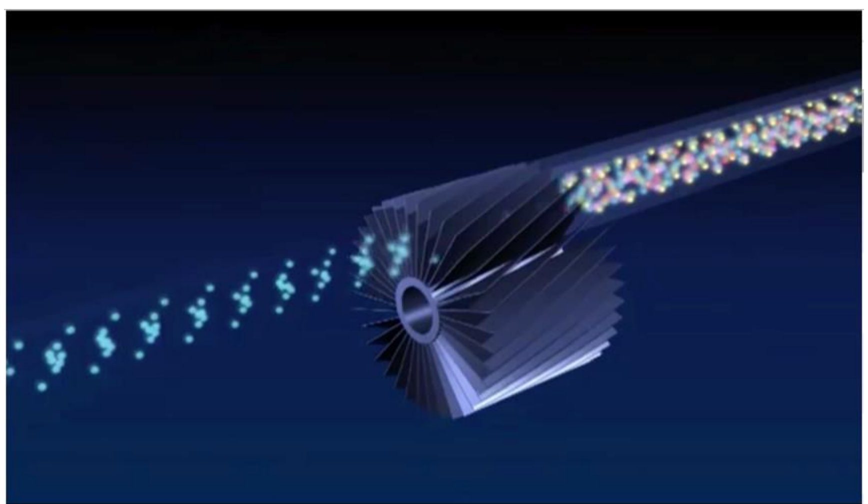
Disk Choppers





Temporal beam manipulation…

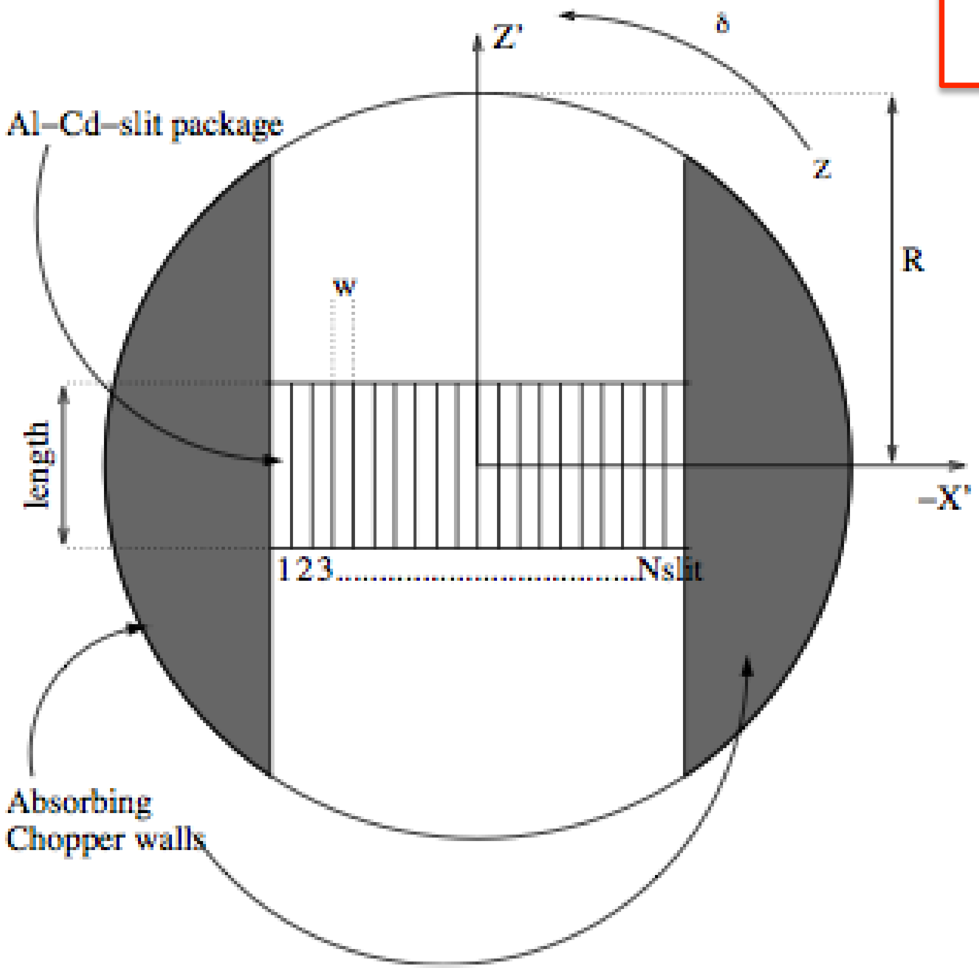
Velocity selector

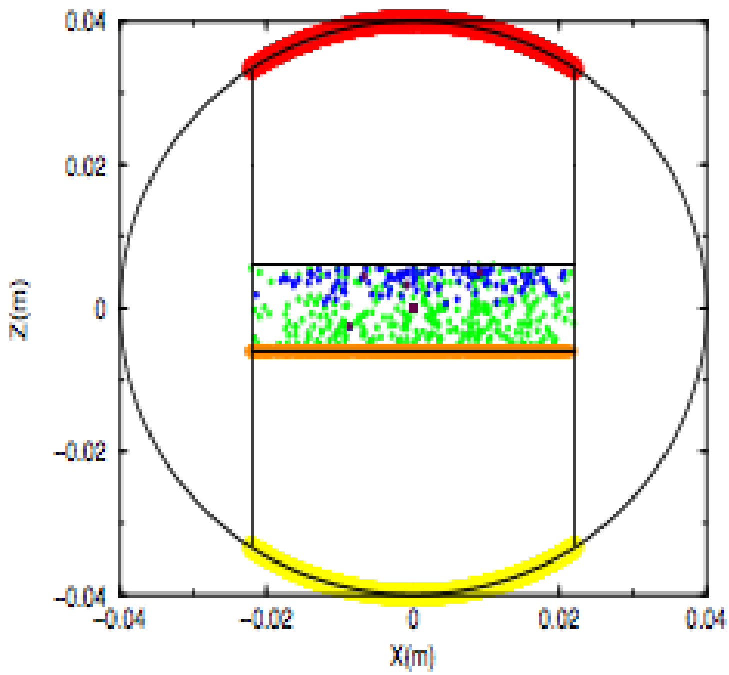


λ *"bandwith" selection,* $\frac{\delta\lambda}{\lambda} \approx 10\%$

Fermi Choppers

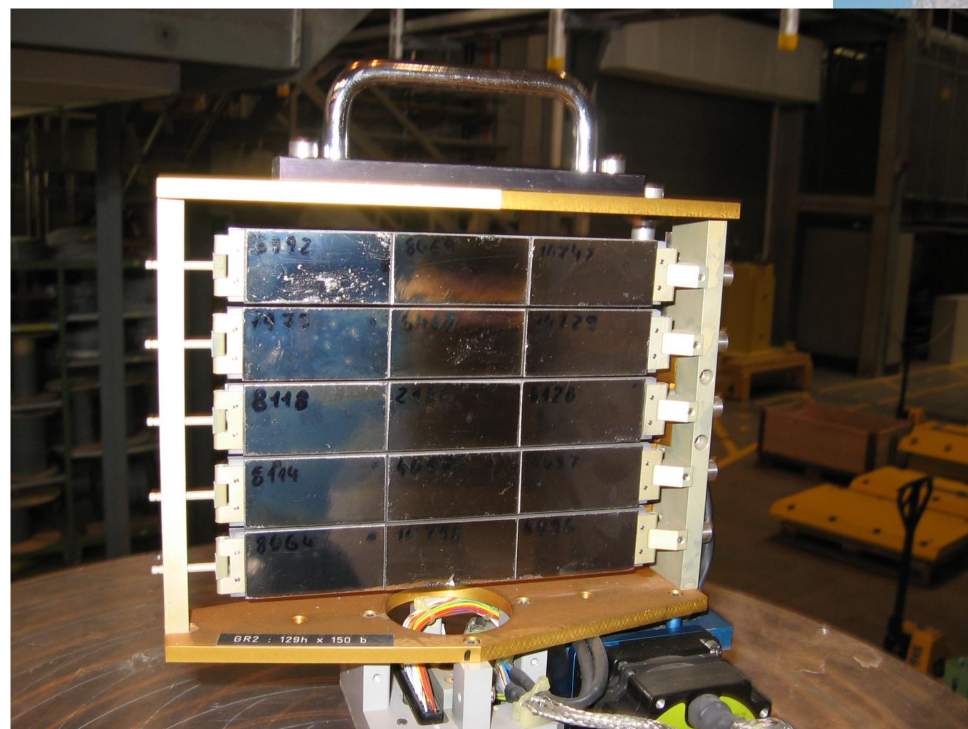
Rotating set of slits with optional insert "slit-package". Temporal + spectral manipulation.



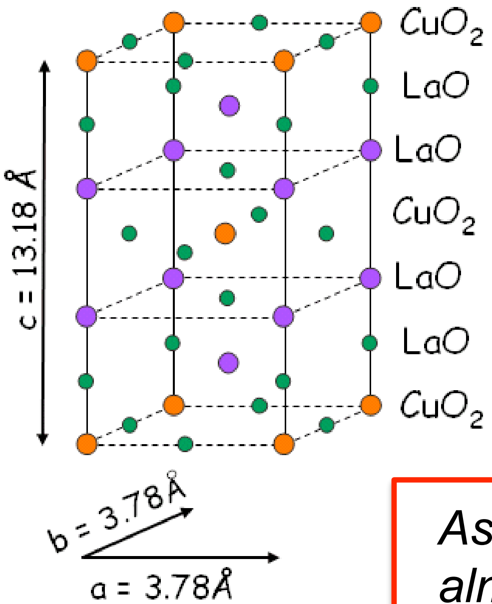


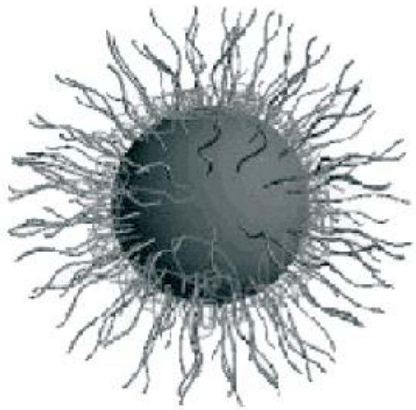
Crystal monochromators (and analyzers)

Bragg's law in action for monochromatisation, $\frac{\delta\lambda}{\lambda} \approx 1 - 3\%$



Samples studied...

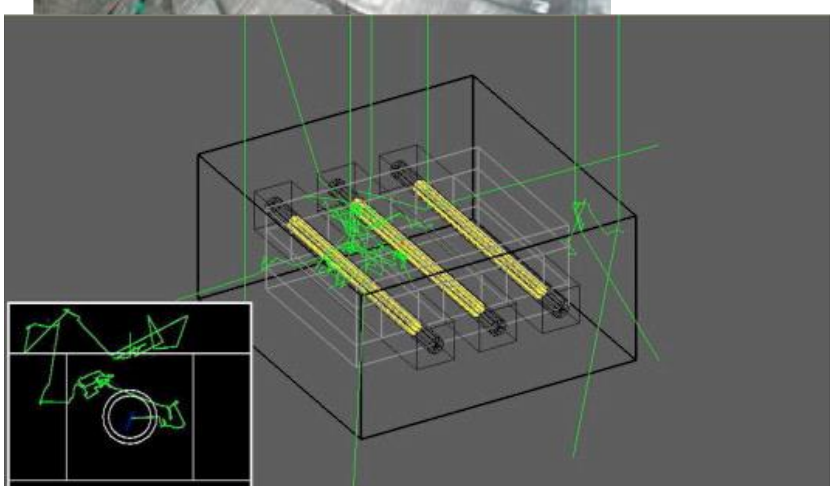




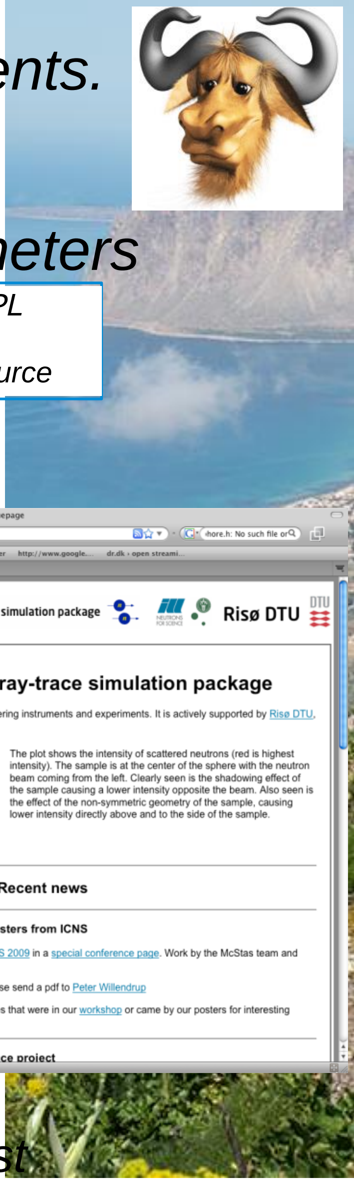
As you have already seen, "matter" in almost any form imaginable…

Detectors

Detectors of any form or size, in McStas we call them "monitors" - they are in most cases perfect probes w/o interaction…



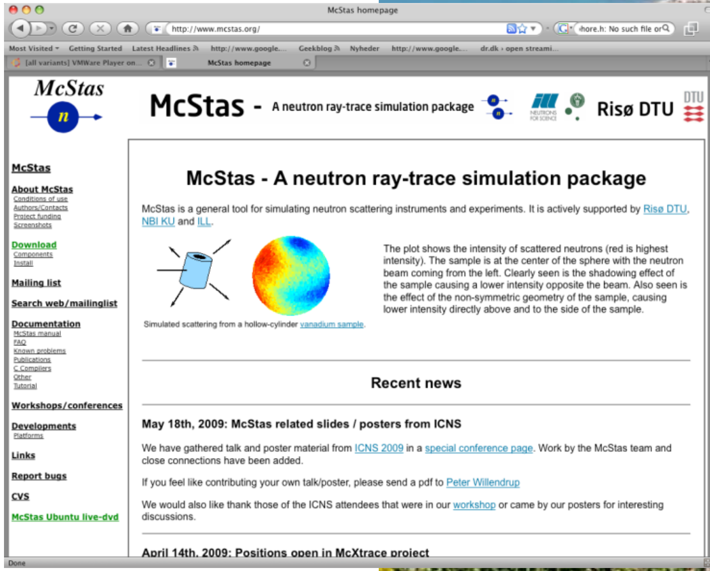
McStas Introduction



- *Flexible, general simulation utility for neutron scattering experiments.*
- *Original design for* ***M****onte* ***c****arlo* ***S****imulation of* ***t****riple* ***a****xis* ***s****pectrometers*

GNU GPL license
Open Source

- *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*

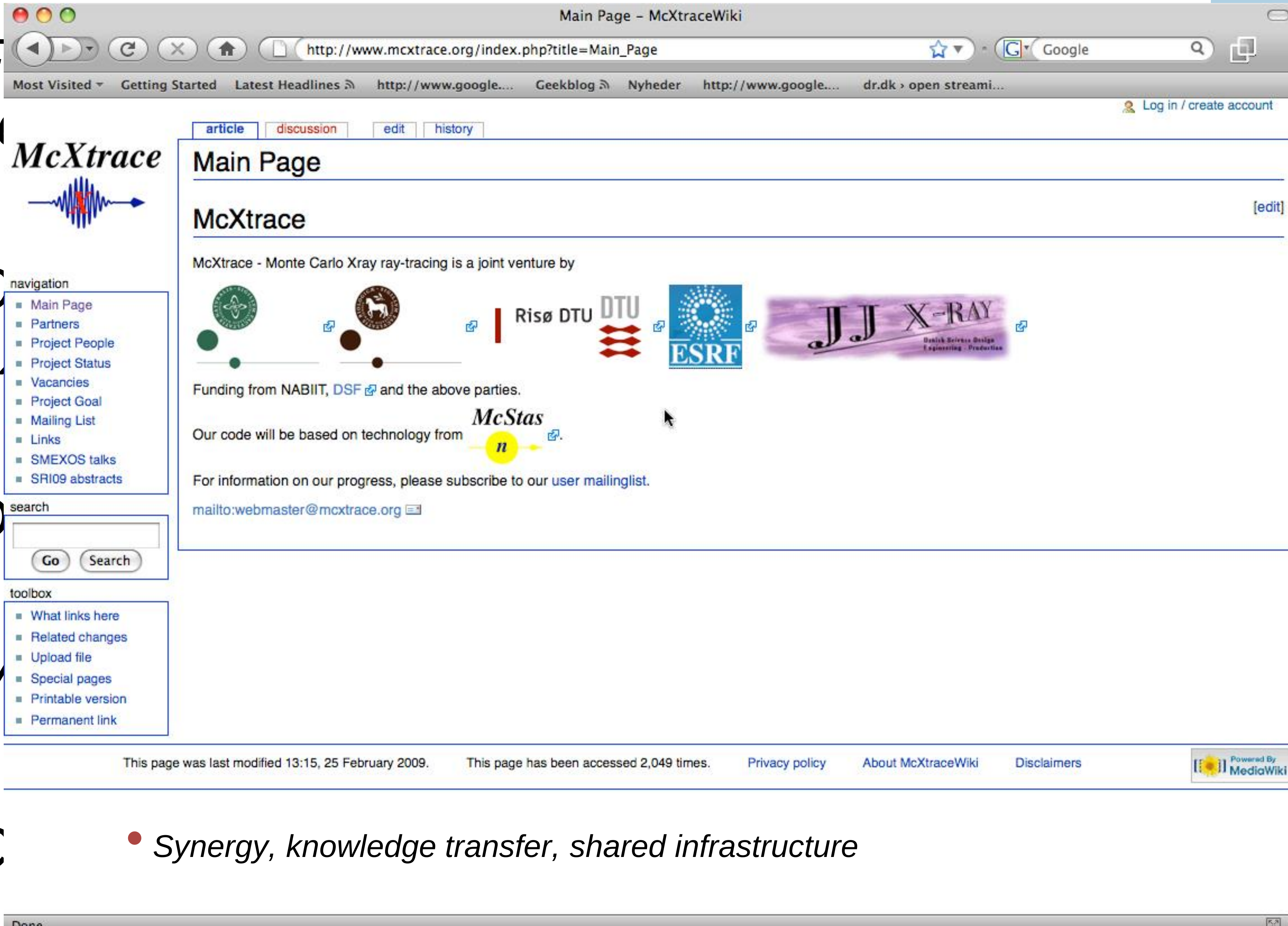


Project website at *http://www.mcstas.org*

mcstas-users@mcstas.org mailinglist

McXtrace - since jan 2009 similar for X-rays

McStas Introduction



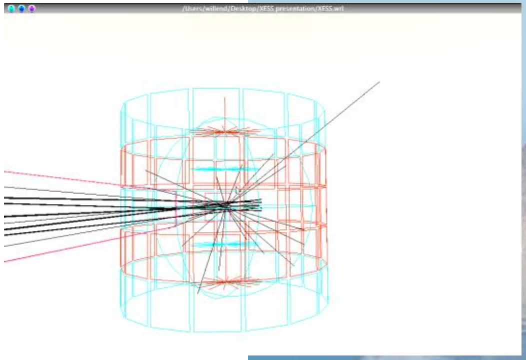
- Synergy, knowledge transfer, shared infrastructure

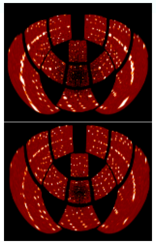
Used in many places

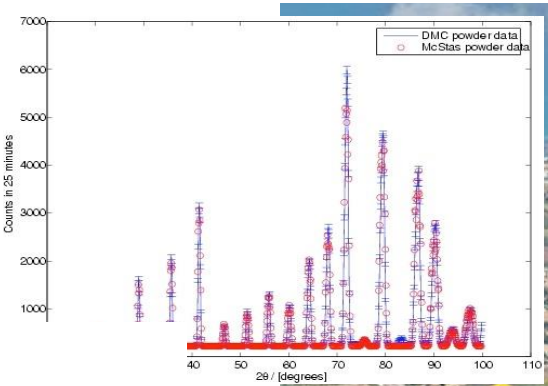
What is McStas used for?

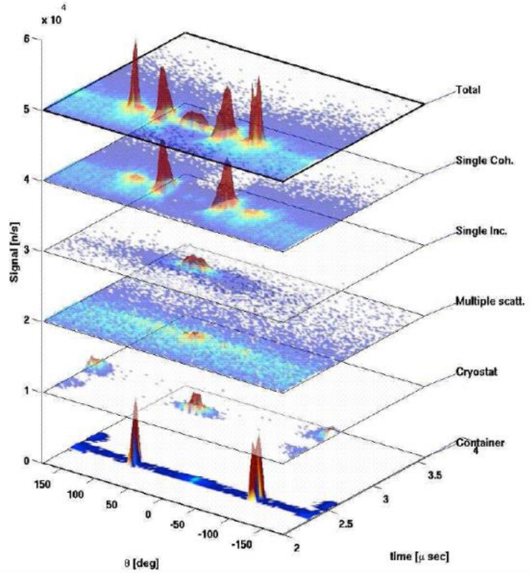
- *Instrumentation*
- *Planning*
- *Construction*
- *Virtual experiments*
- *Data analysis*
- *Teaching*

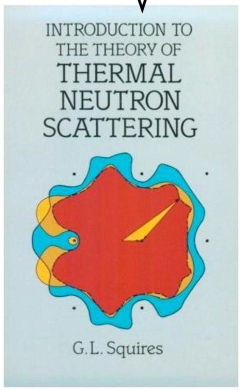
(KU, DTU)

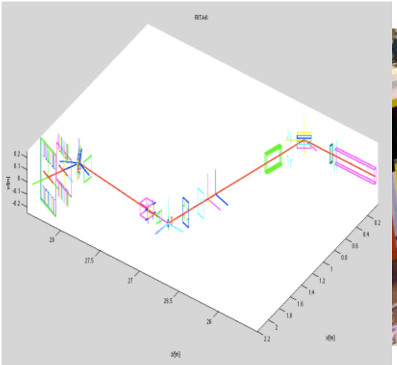






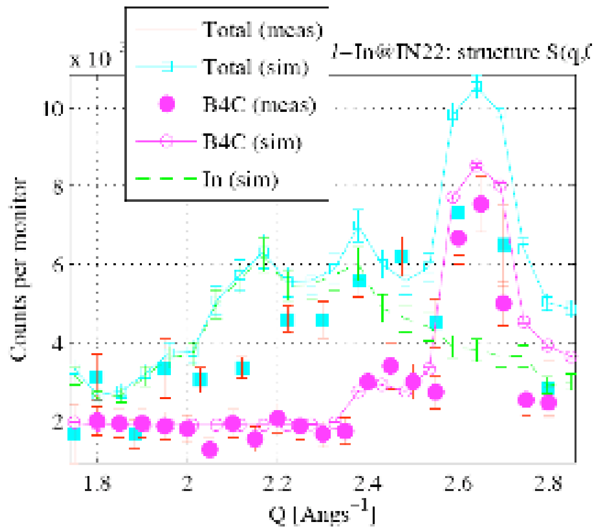




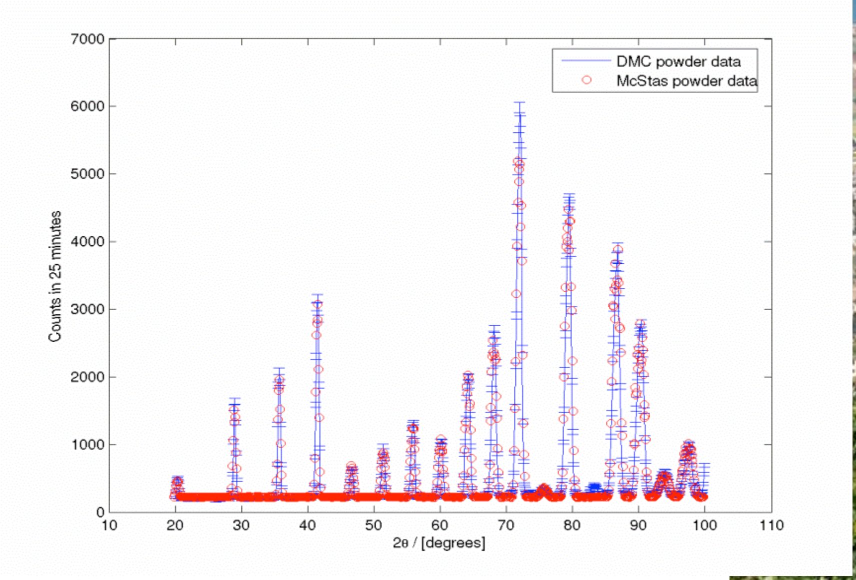


Reliability - cross comparisons

- 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



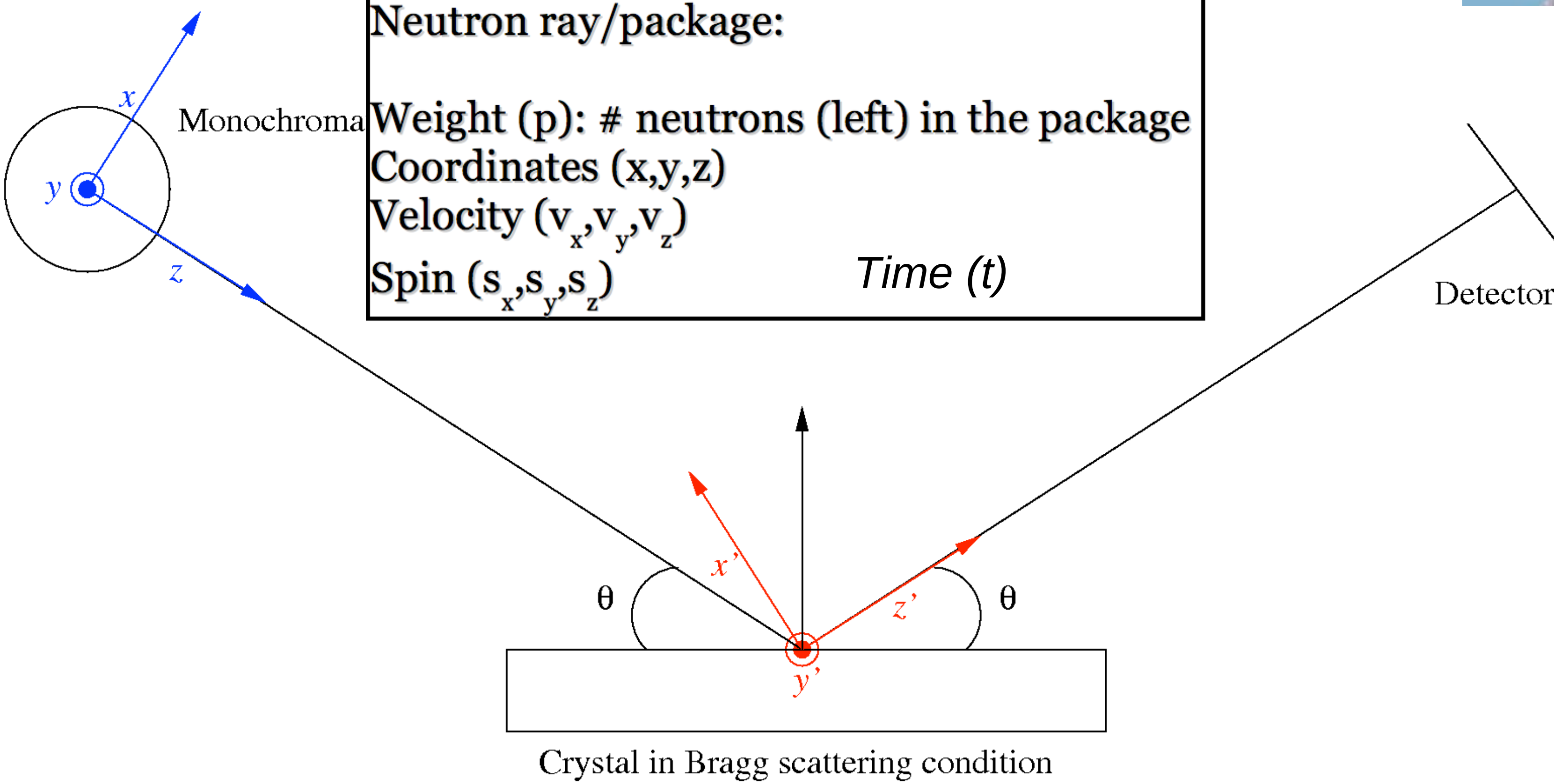
P. Willendrup et al., Physica B, 386, (2006), 1032.

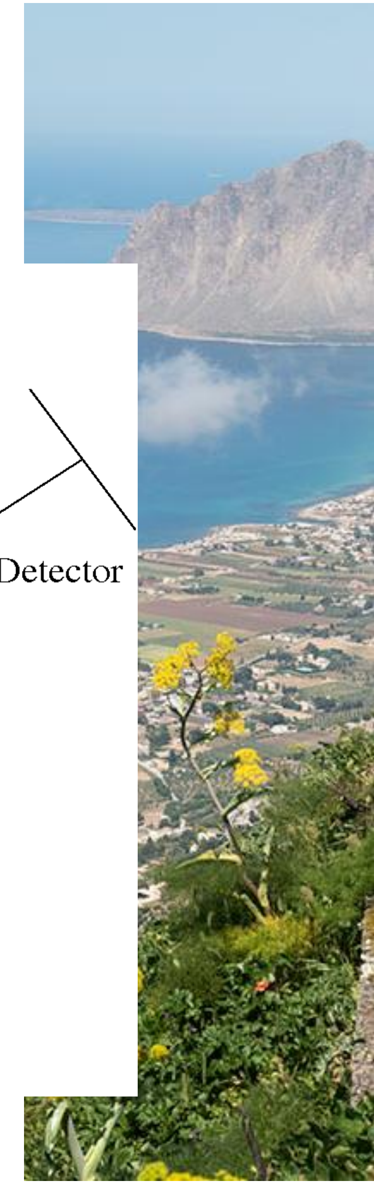
DTU
ess EUROPEAN SPALLATION SOURCE

McStas: key concepts

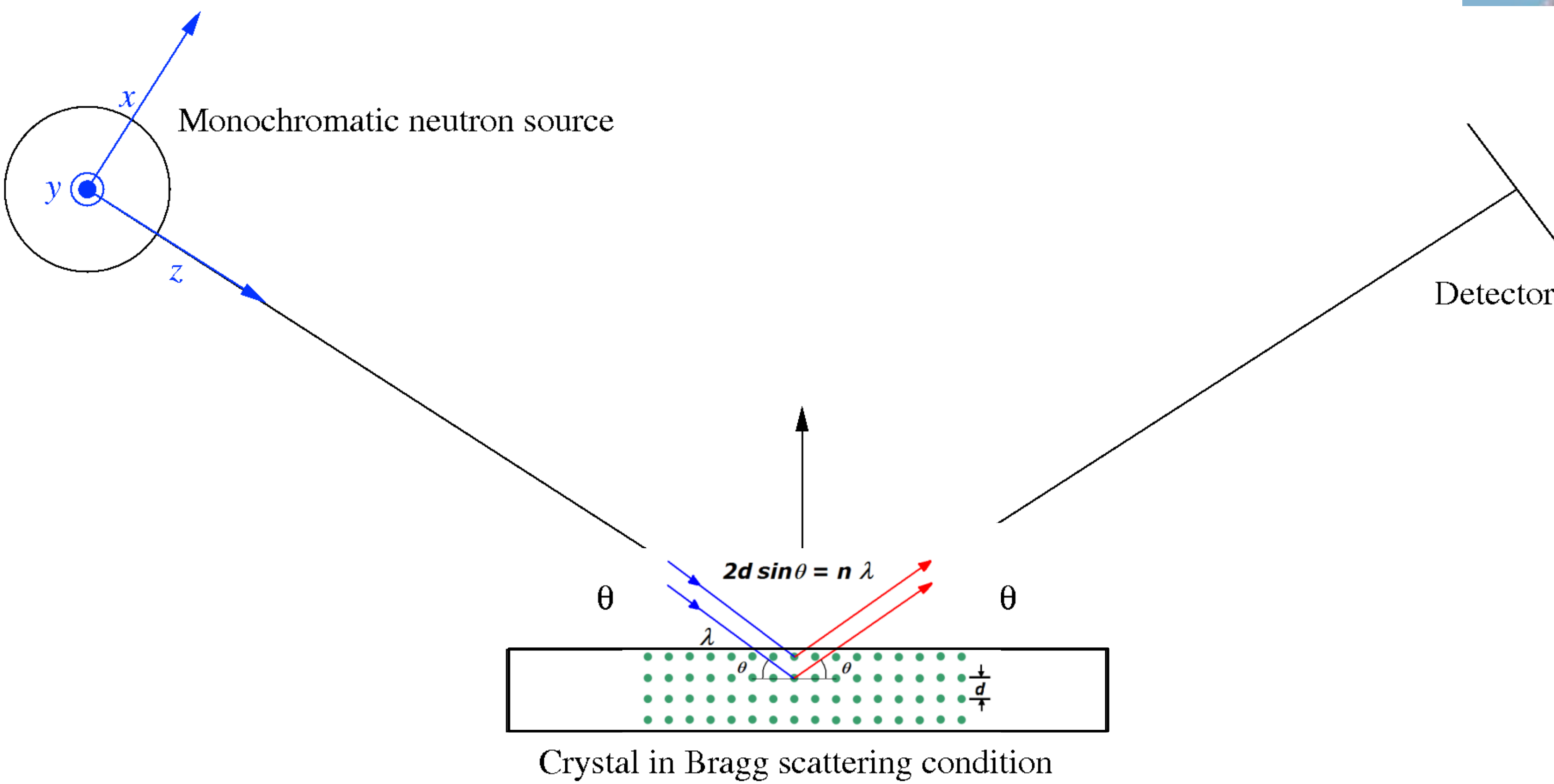
Neutron ray/package:

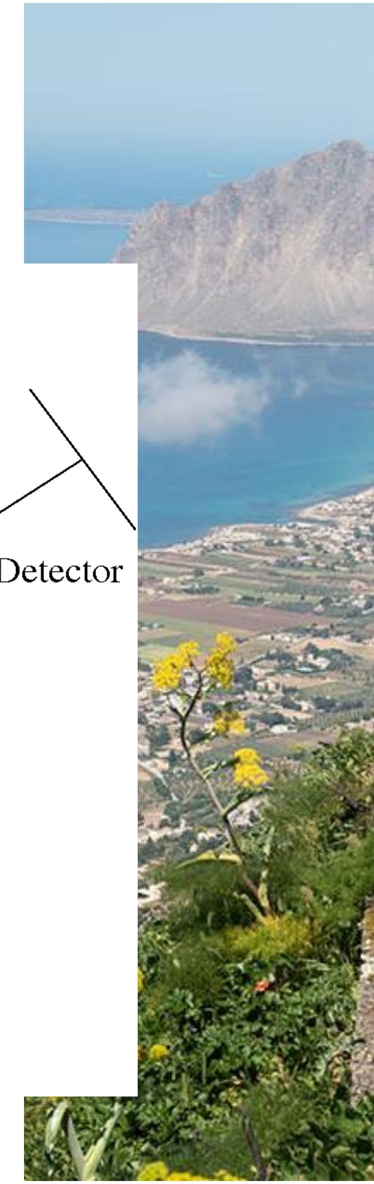
Weight (p): # neutrons (left) in the package
Coordinates (x,y,z)
Velocity (v_x,v_y,v_z)
Spin (s_x,s_y,s_z)
Time (t)





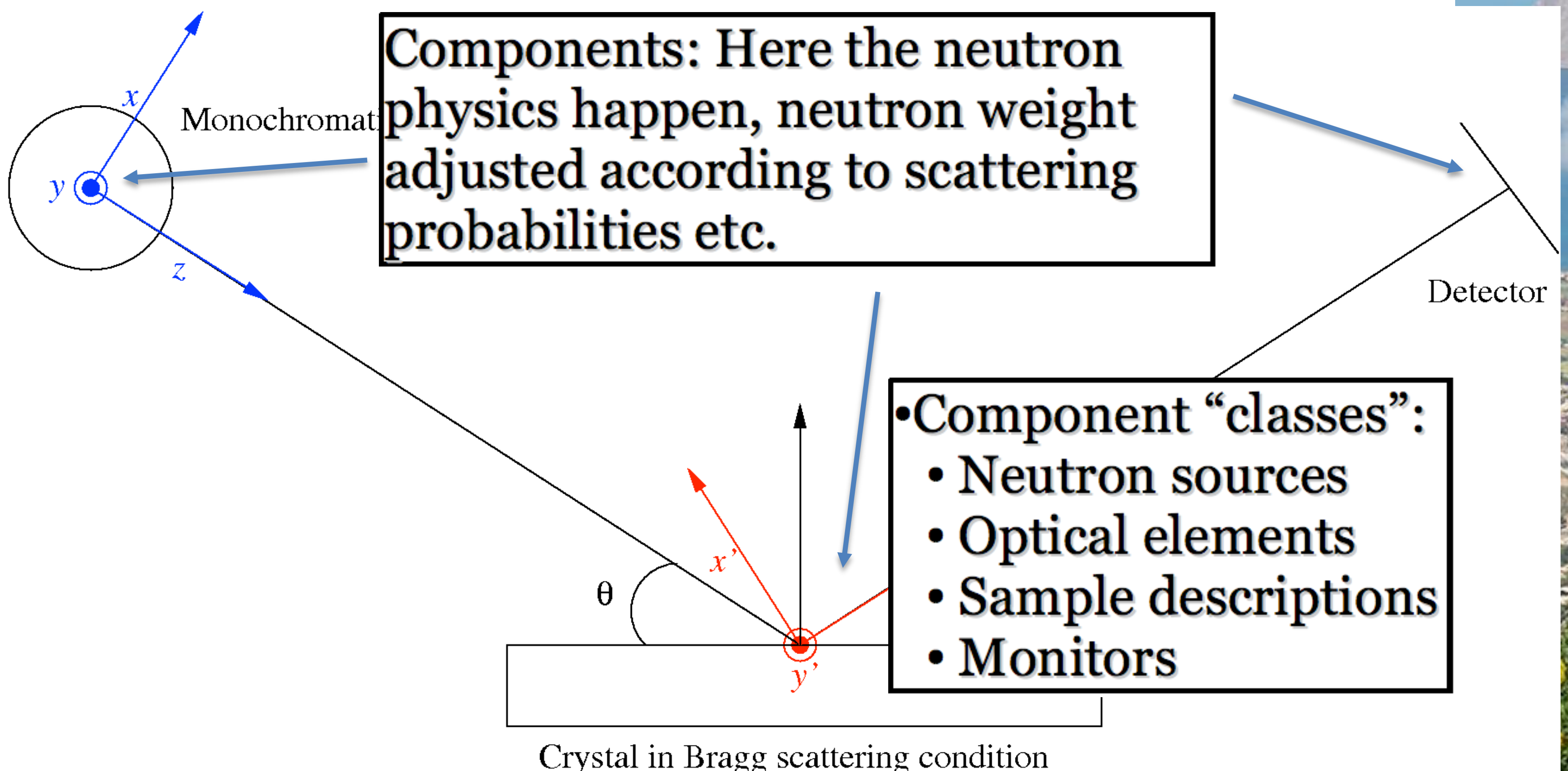
McStas: key concepts





McStas: key concepts

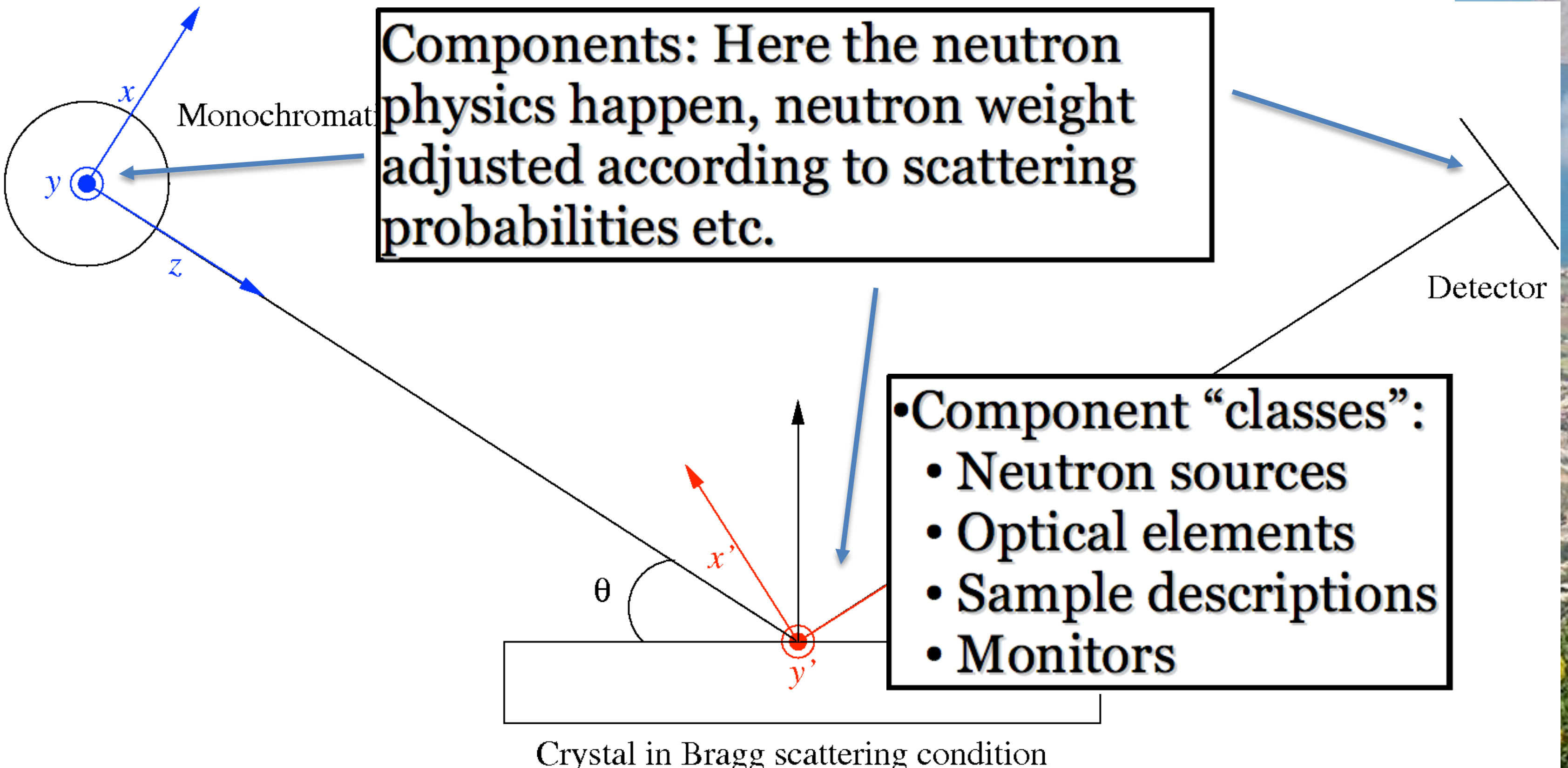
Components: Here the neutron physics happen, neutron weight adjusted according to scattering probabilities etc.



- Component "classes":
  - Neutron sources
  - Optical elements
  - Sample descriptions
  - Monitors

McStas: key concepts

Local, internal coordinate system!



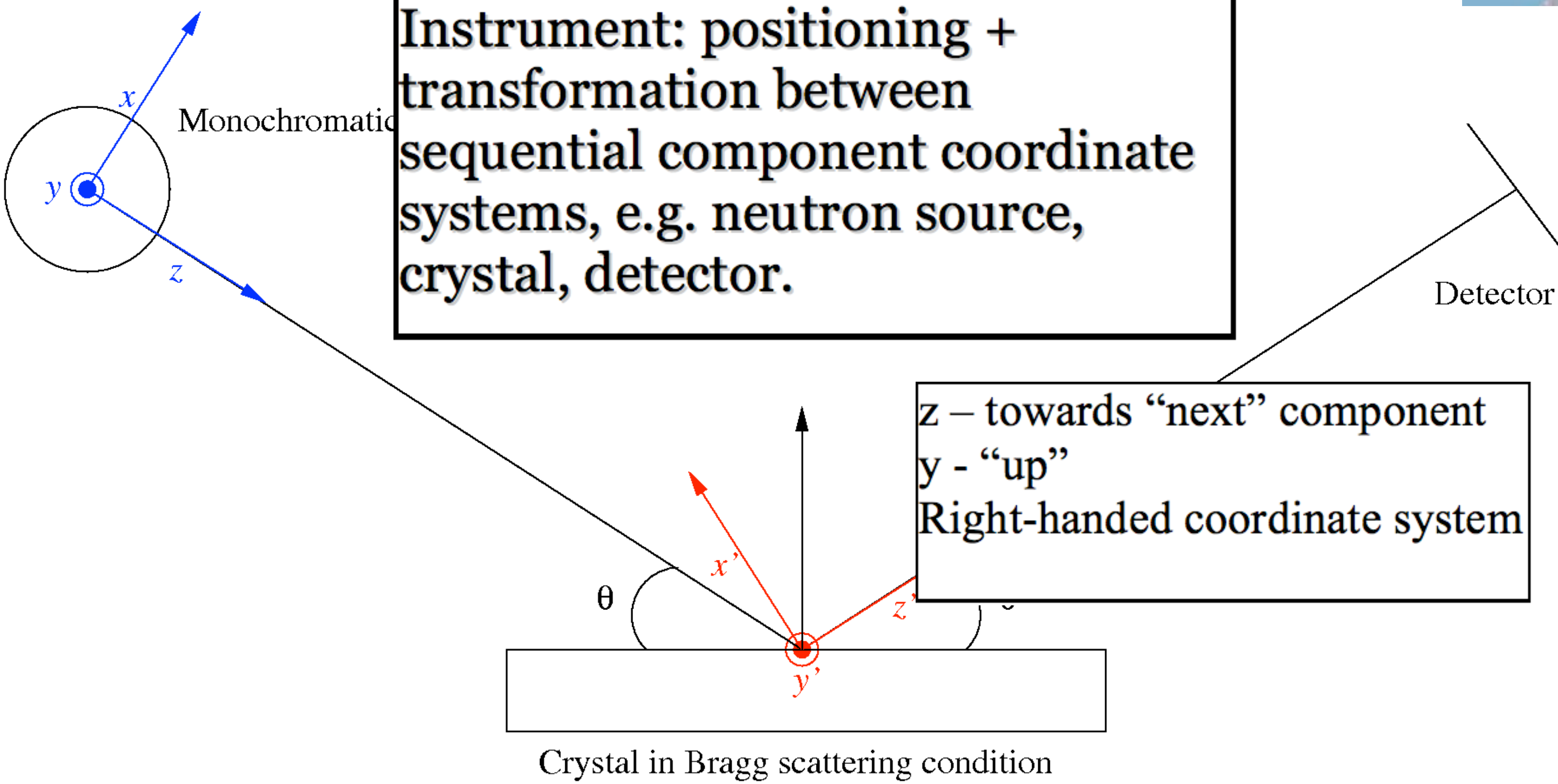
Components: Here the neutron physics happen, neutron weight adjusted according to scattering probabilities etc.

- Component "classes":
  - Neutron sources
  - Optical elements
  - Sample descriptions
  - Monitors

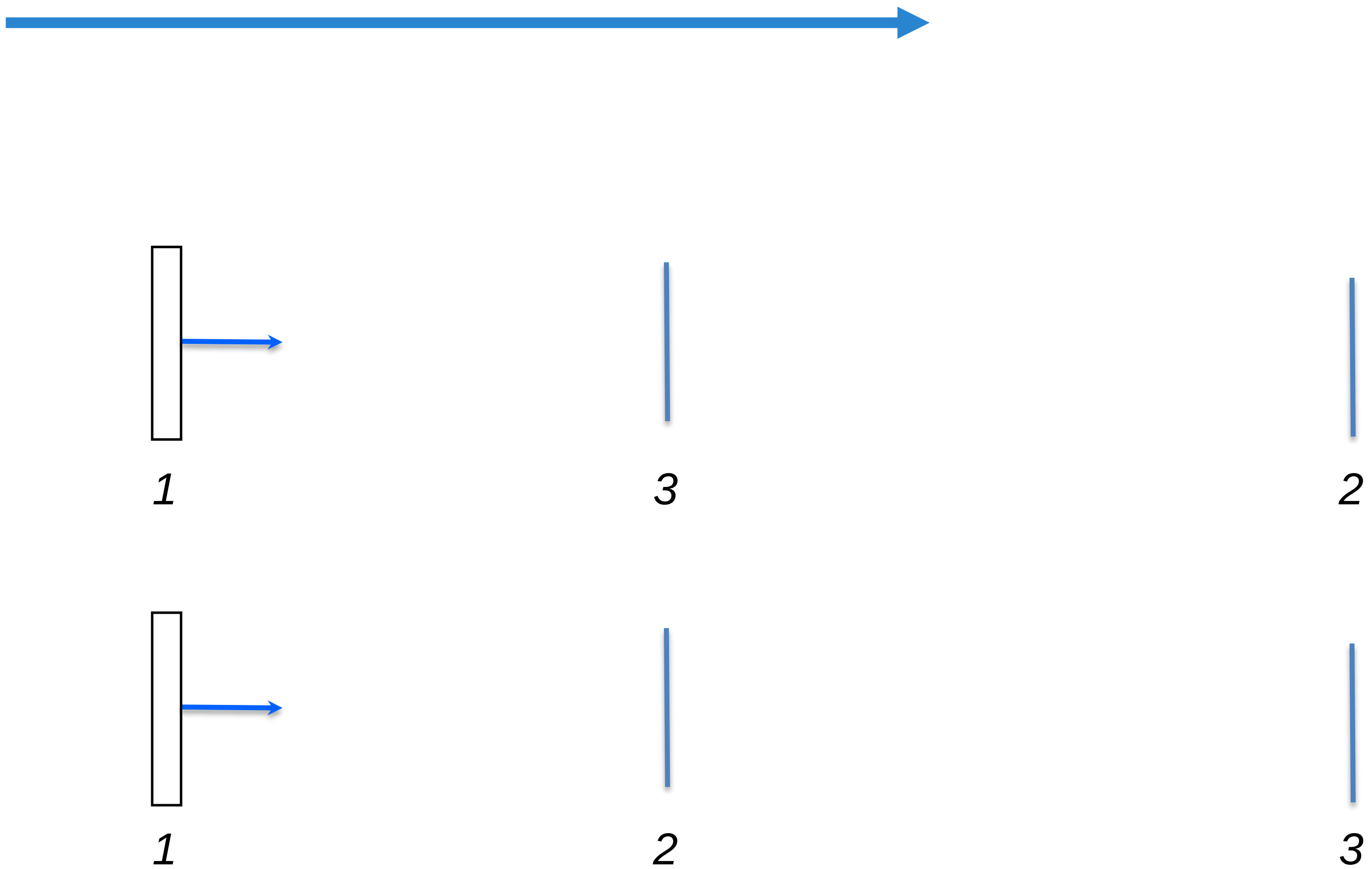
McStas: key concepts

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

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

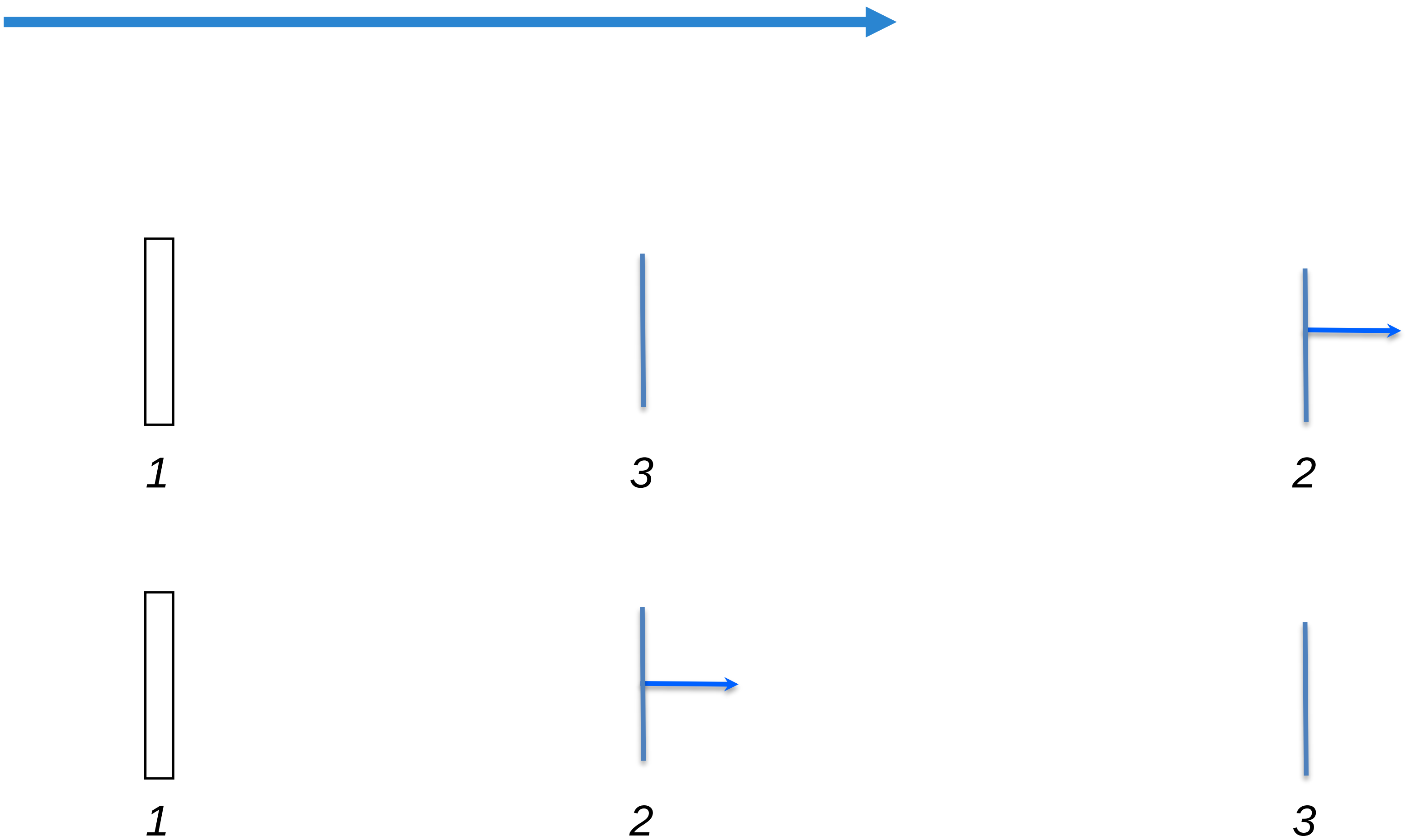


Order of components is important



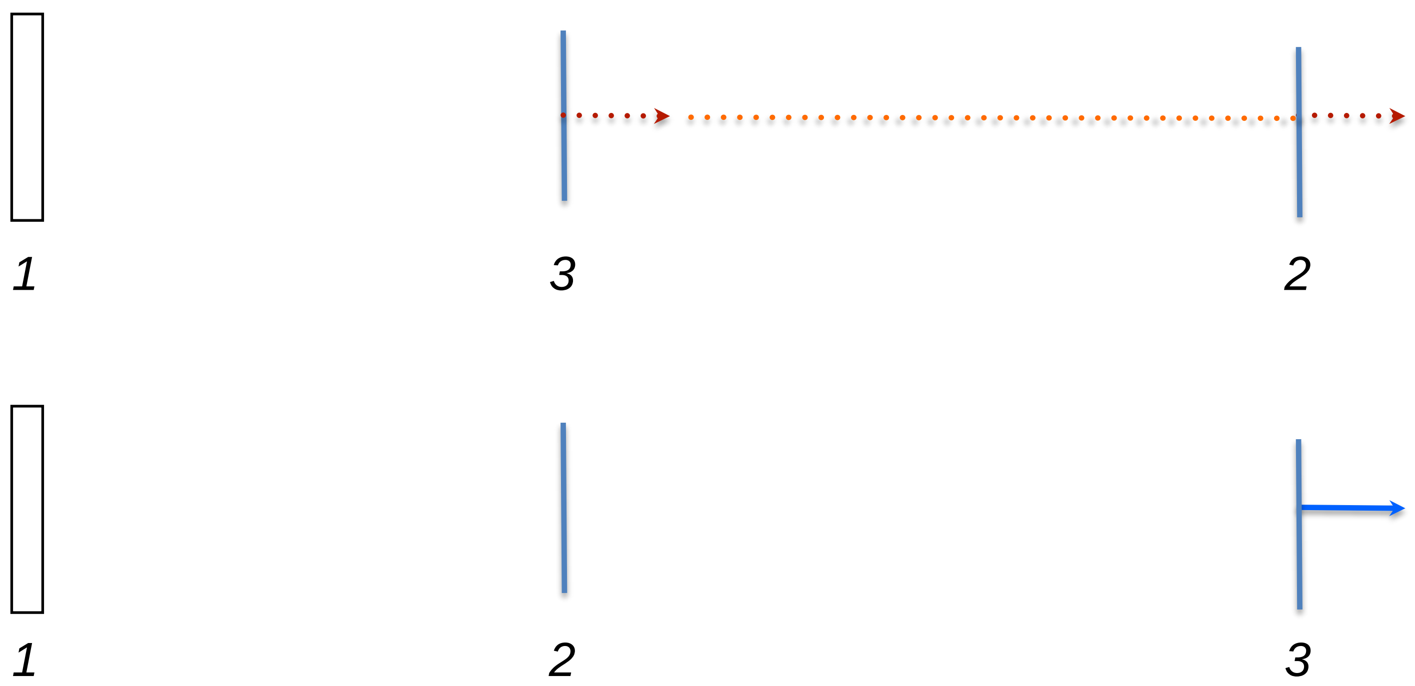
Starting at the source

Order of components is important



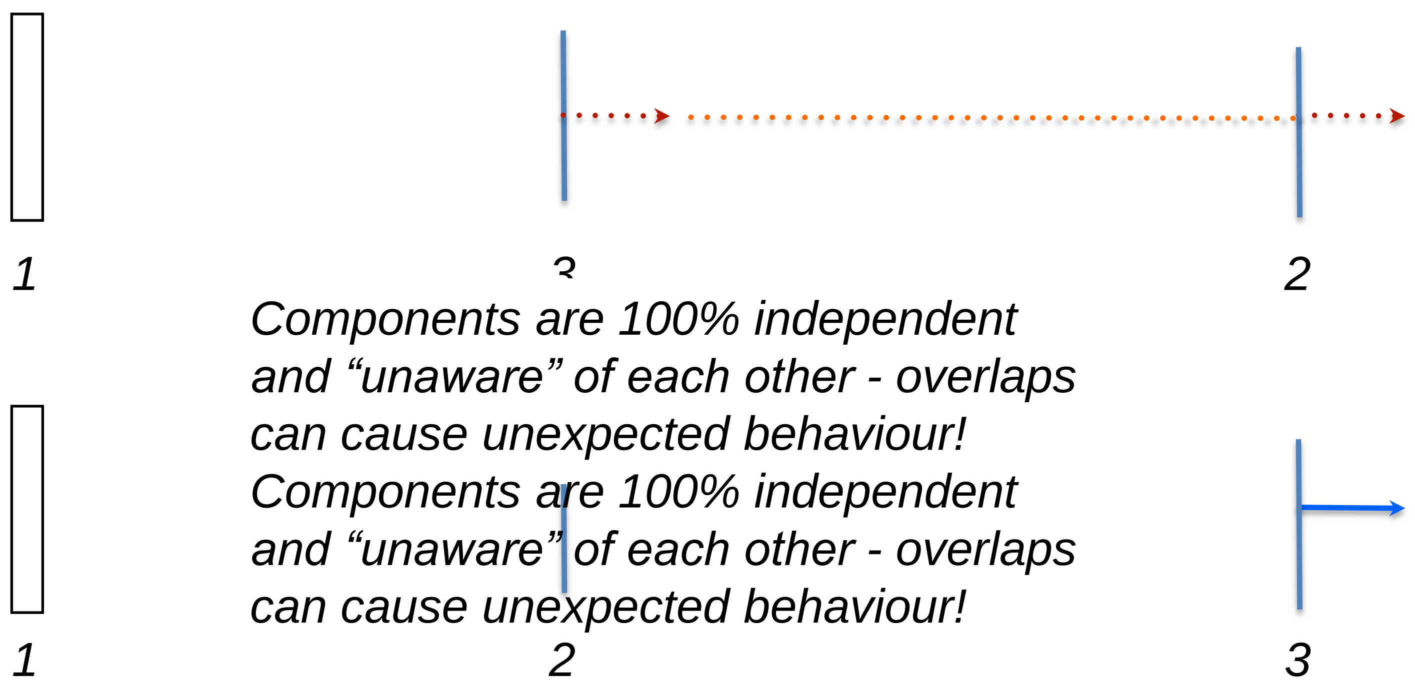
Moving to first comp in the list

Order of components is important



Moving to 3rd comp in list requires "moving back in time".
Default behavior is to ABSORB this type of neutron.
For monitors use restore_neutron=1 in this case.
For homegrown comps use ALLOW_BACKPROP macro.

Order of components is important



Components are 100% independent and "unaware" of each other - overlaps can cause unexpected behaviour!

Components are 100% independent and "unaware" of each other - overlaps can cause unexpected behaviour!

Moving to 3rd comp in list requires "moving back in time".
Default behavior is to ABSORB this type of neutron.
For monitors use restore_neutron=1 in this case.
For homegrown comps use ALLOW_BACKPROP macro.

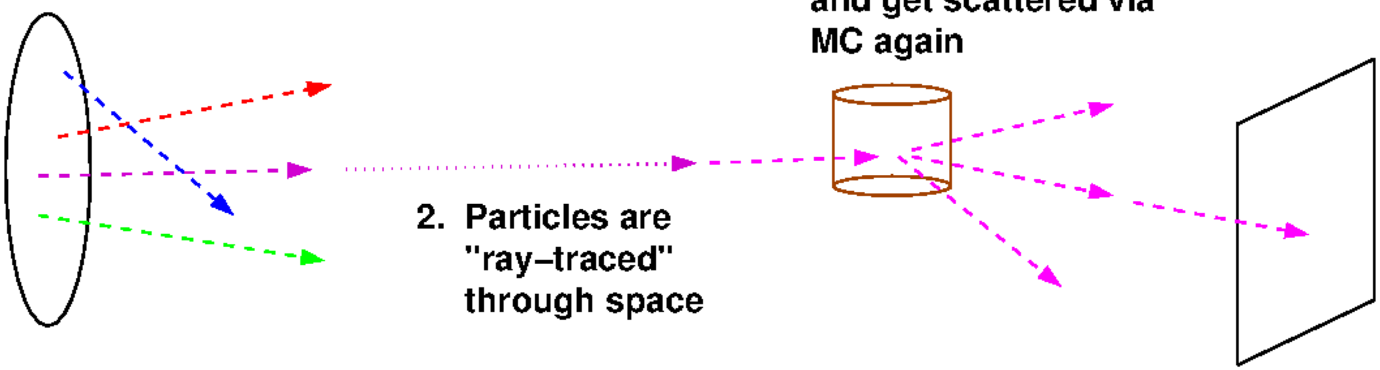
In the big picture…

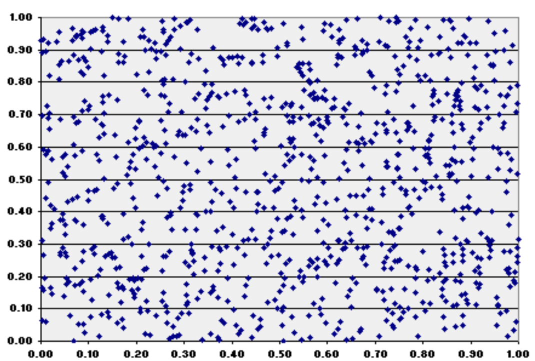
1. Particles emitted with random starting conditions via MC

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 = 

\+ 

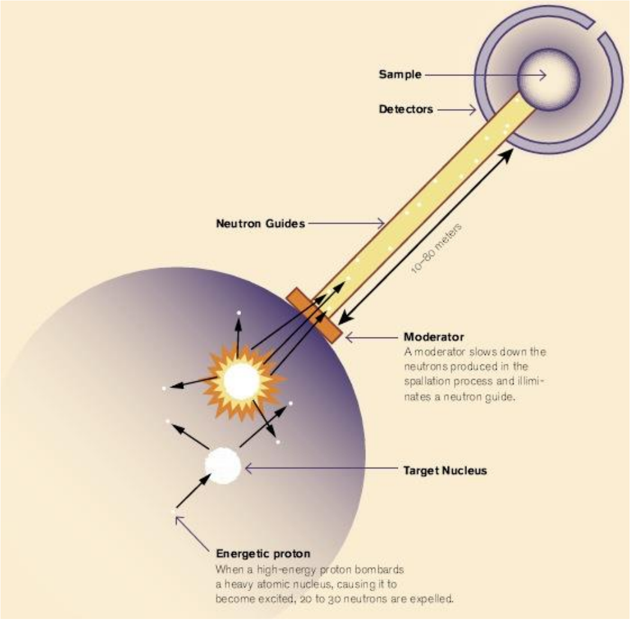
McStas overview

- *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.*



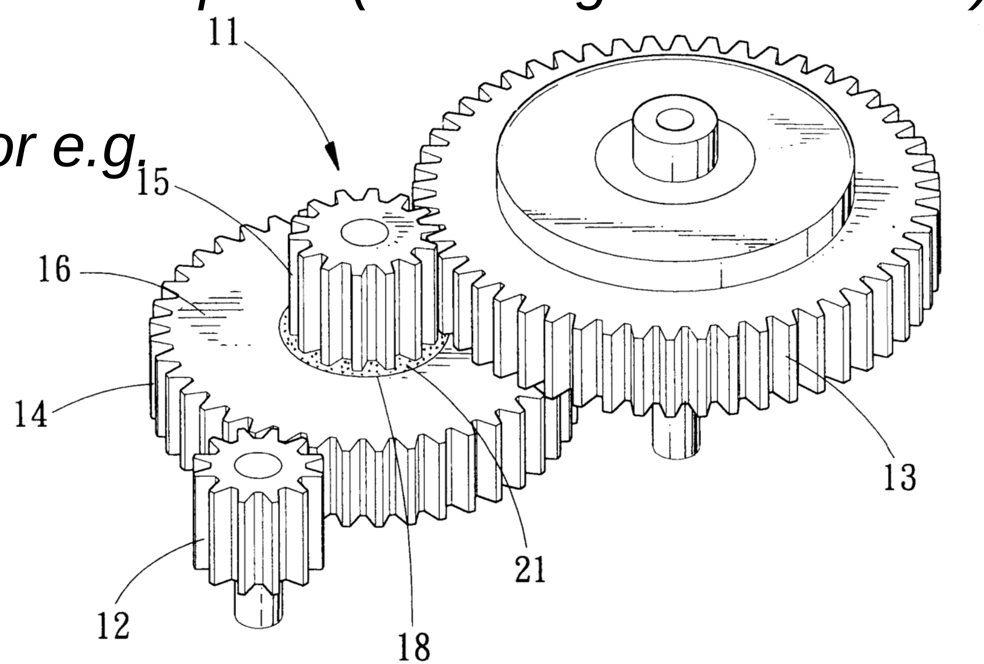
Under-the-hood / inner workings

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

  - *Simple Instrument language* —*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

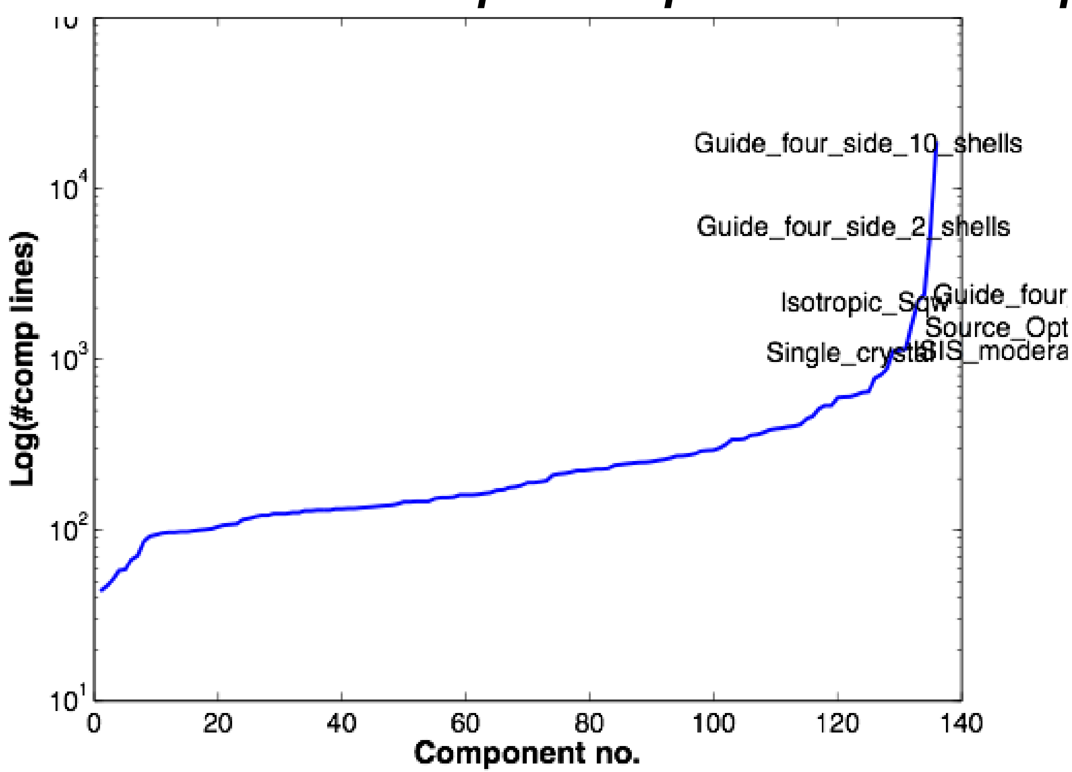
Output is a single c-file, which can be compiled using e.g. gcc.

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 - 199 comps in total 5)

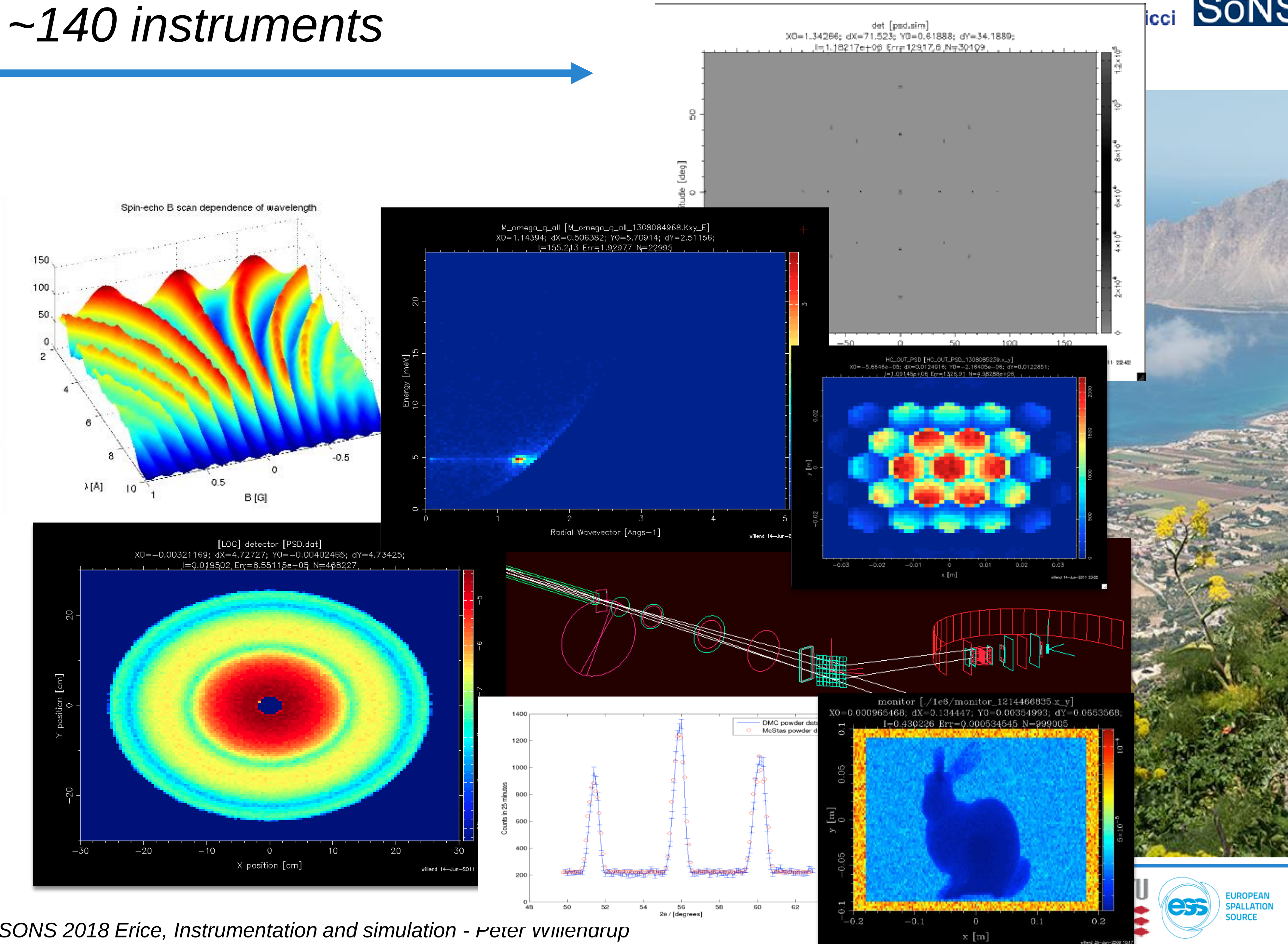


Including user contribs

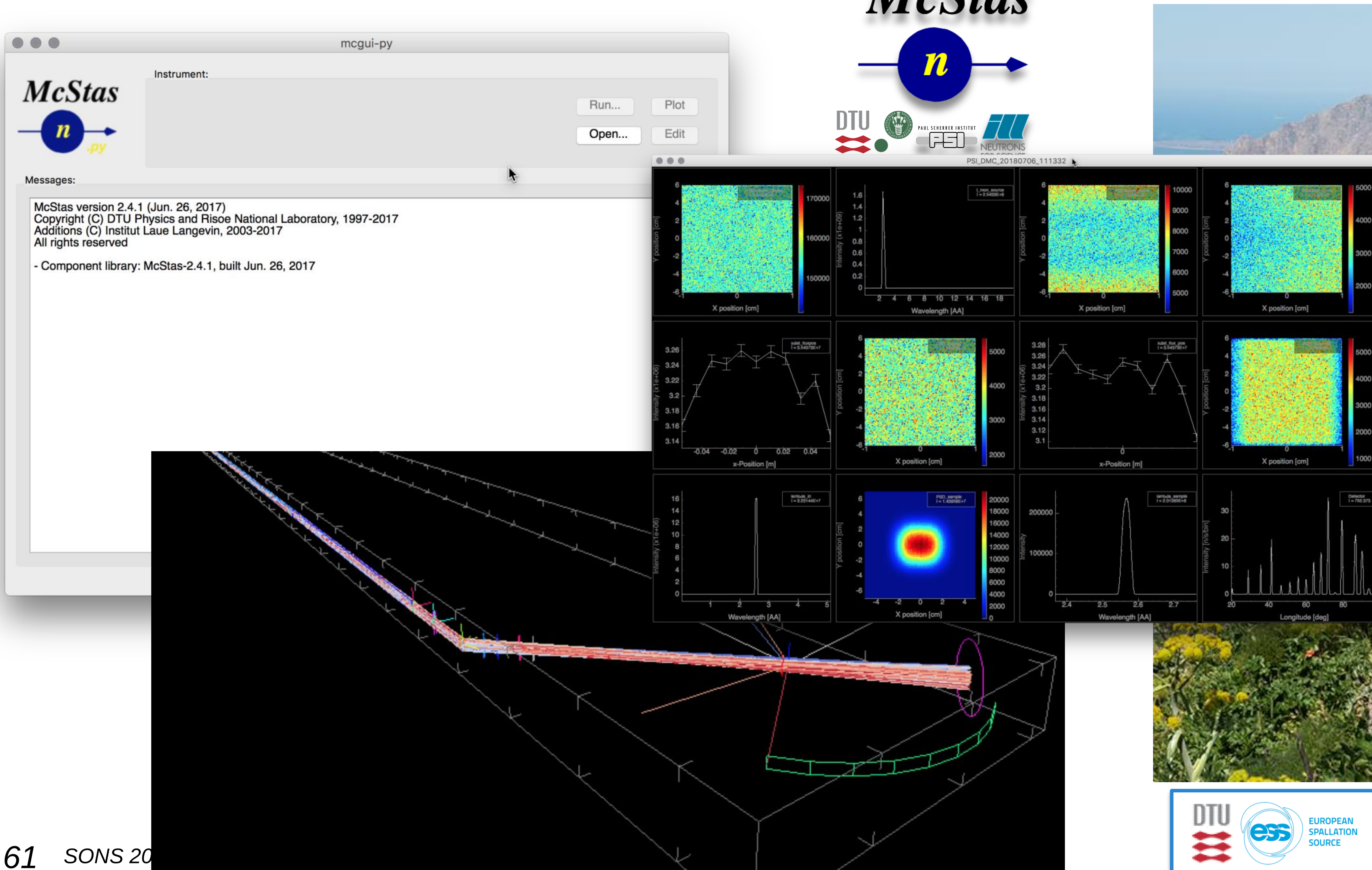
- *Well-developed community support*
  - ***30-40%*** *of existing and new additions are* ***from users***
  - *No direct refereeing of the code, but these requirements:*
    - ***At least one test-instrument***
    - ***Meaningful documentation*** *headers (in-code docs)*
  - *Contributions go in dedicated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

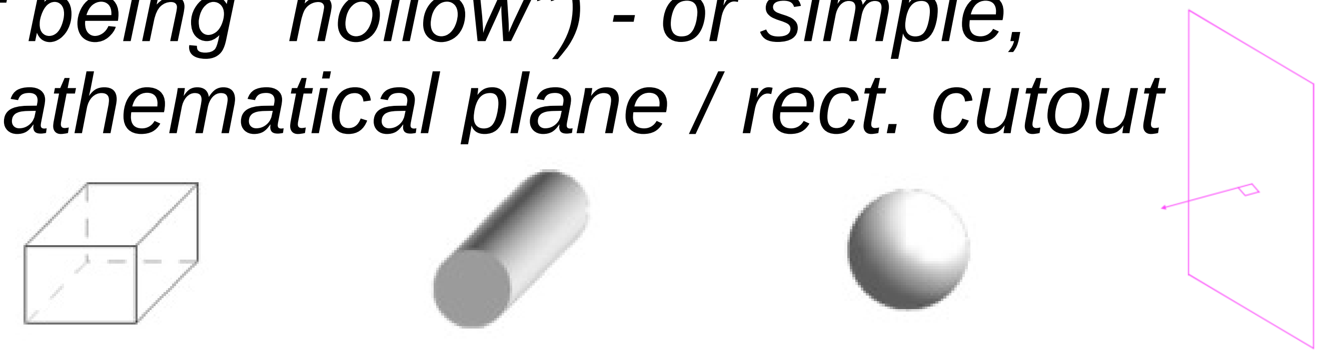


Time for a demo??

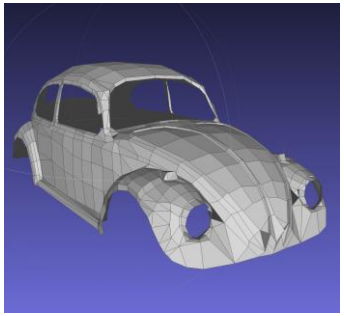
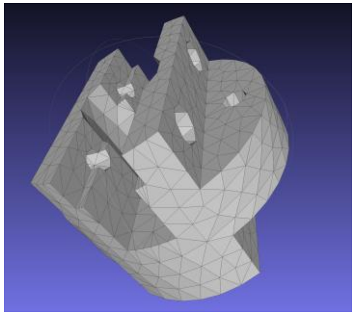
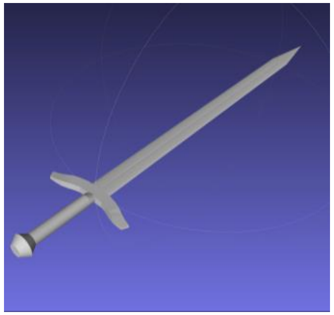


Sample + environment, geometrical constraints

- Geometries of objects in McStas, either
  - Box, cylinder, sphere (all with possibility of being "hollow") - or simple, mathematical plane / rect. cutout



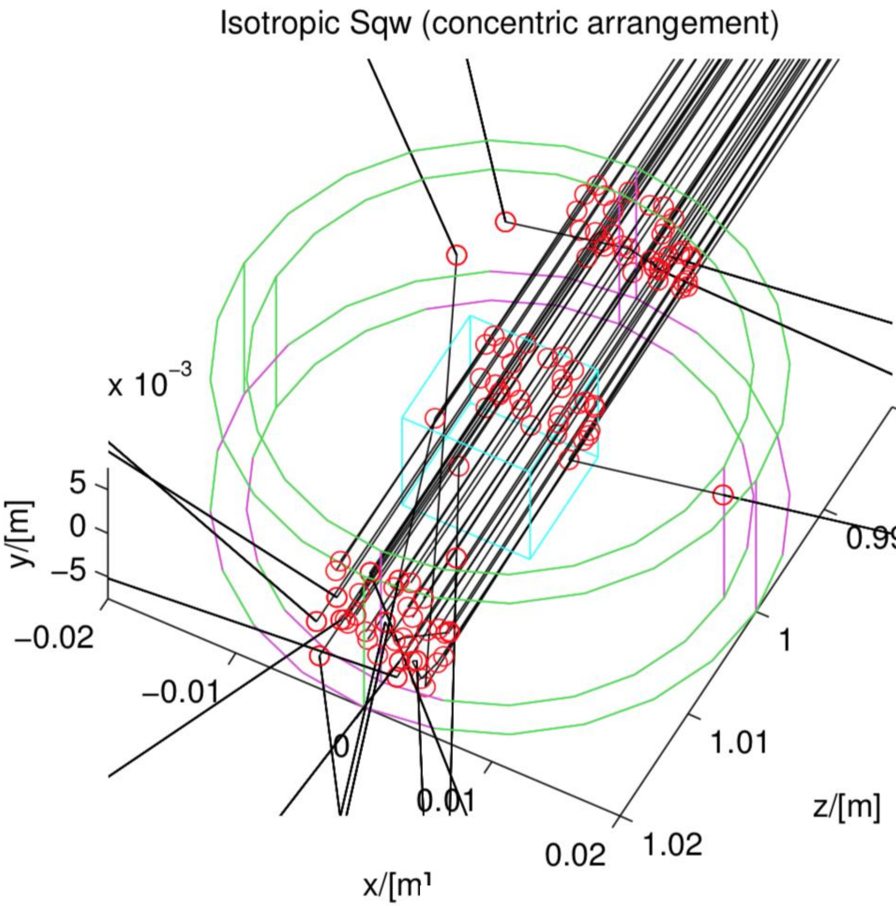
  - Polygonal surface (GeomView OFF format)

  - Almost always homogenous material

Sample + environment - concentric geometries

```
/* external shield */
COMPONENT Environment_in=Isotropic_Sqw(
  radius = environment_radius, yheight = 0.1, thickness=environment_thickness,
  Sqw_coh=environment, concentric=1, verbose=0, order=1, d_phi=2*RAD2DEG*atan2(1, LSD)
) WHEN (environment_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND %{
  flag_env += SCATTERED;
%}

/* sample container */
COMPONENT Container_in=Isotropic_Sqw(
  xwidth  = sample_width+1e-4+container_thickness,
  zdepth  = sample_thickness+1e-4+container_thickness,
  yheight = sample_height, thickness=container_thickness,
  Sqw_coh=container, concentric=1, verbose=0, order=1, d_phi=2*RAD2DEG*atan2(1, LSD)
  ) WHEN(container_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND
%{
  flag_env += SCATTERED;
%}

COMPONENT SampleS=Isotropic_Sqw(
  xwidth = sample_width, zdepth=sample_thickness, yheight = sample_height,
  Sqw_coh= sample_coh, Sqw_inc= sample_inc, p_interact=0.9,
  d_phi=2*RAD2DEG*atan2(1, LSD), order=1)
WHEN (flag_sample_choice == 1)
AT (0, 0, 0)   RELATIVE Sample_rot
EXTEND
%{
  flag_sample += SCATTERED;
%}

COMPONENT SampleV=Incoherent(xwidth = sample_width, zdepth=sample_thickness, yheight = sample_hei
  focus_ah = 2*RAD2DEG*atan2(1, LSD), focus_aw=150.0)
WHEN (flag_sample_choice == 2)
AT (0, 0, 0)   RELATIVE Sample_rot
EXTEND
%{
  flag_sample += SCATTERED;
%}

COMPONENT Container_out=COPY(Container_in)(concentric=0)
WHEN(container_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND
%{
  flag_env += SCATTERED;
%}

/* external shield */
COMPONENT Environment_out=COPY(Environment_in)(concentric=0)
WHEN (environment_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND %{
  flag_env += SCATTERED;
%}
```



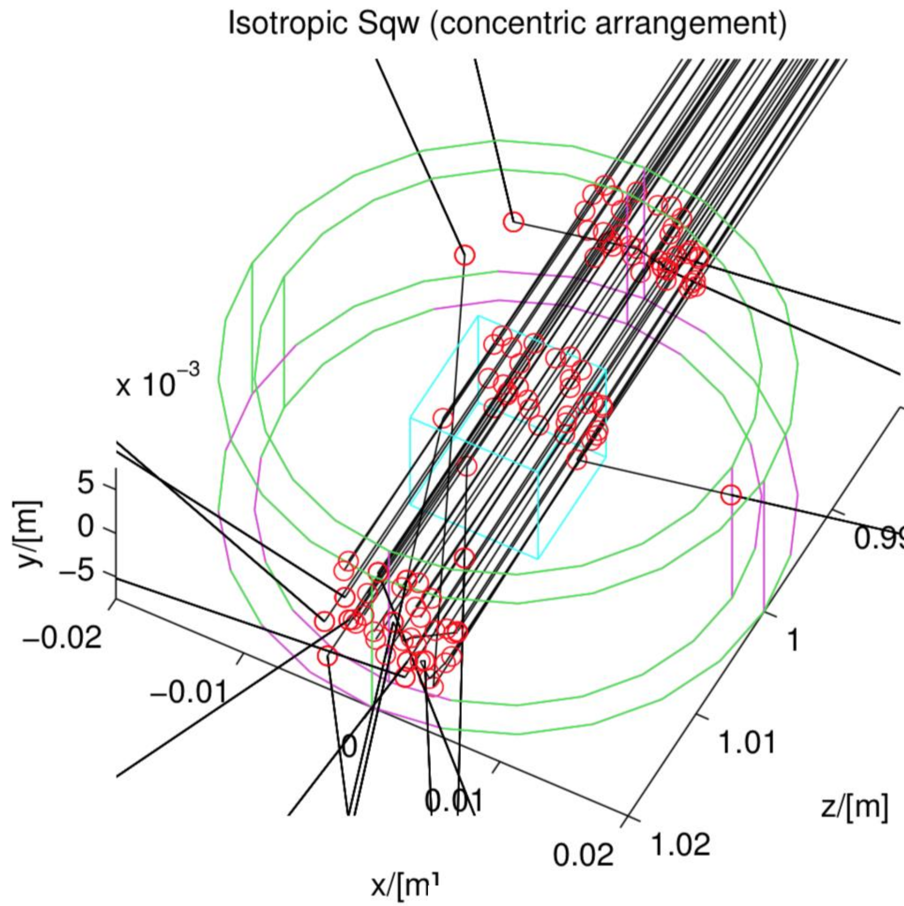
"concentric" geometries

Sample + environment - concentric geometries

```
/* external shield */
COMPONENT Environment_in=Isotropic_Sqw(
  radius = environment_radius, yheight = 0.1, thickness=environment_thickness,
  Sqw_coh=environment, concentric=1, verbose=0, order=1, d_phi=2*RAD2DEG*atan2(1, LSD)
) WHEN (environment_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND %{
  flag_env += SCATTERED;
%}

/* sample container */
COMPONENT Container_in=Isotropic_Sqw(
  xwidth  = sample_width+1e-4+container_thickness,
  zdepth  = sample_thickness+1e-4+container_thickness,
  yheight = sample_height, thickness=container_thickness,
  Sqw_coh=container, concentric=1, verbose=0, order=1, d_phi=2*RAD2DEG*atan2(1, LSD)
  ) WHEN(container_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND
%{
  flag_env += SCATTERED;
%}

COMPONENT SampleS=Isotropic_Sqw(
  xwidth = sample_width, zdepth=sample_thickness, yheight = sample_height,
  Sqw_coh= sample_coh, Sqw_inc= sample_inc, p_interact=0.9,
  d_phi=2*RAD2DEG*atan2(1, LSD), order=1)
WHEN (flag_sample_choice == 1)
AT (0, 0, 0)   RELATIVE Sample_rot
EXTEND
%{
  flag_sample += SCATTERED;
%}

COMPONENT SampleV=Incoherent(xwidth = sample_width, zdepth=sample_thickness, yheight = sample_hei
  focus_ah = 2*RAD2DEG*atan2(1, LSD), focus_aw=150.0)
WHEN (flag_sample_choice == 2)
AT (0, 0, 0)   RELATIVE Sample_rot
EXTEND
%{
  flag_sample += SCATTERED;
%}

COMPONENT Container_out=COPY(Container_in)(concentric=0)
WHEN(container_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND
%{
  flag_env += SCATTERED;
%}

/* external shield */
COMPONENT Environment_out=COPY(Environment_in)(concentric=0)
WHEN (environment_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND %{
  flag_env += SCATTERED;
%}
```

A

A'



"concentric" geometries

Sample + environment - concentric geometries

```
/* external shield */
COMPONENT Environment_in=Isotropic_Sqw(
  radius = environment_radius, yheight = 0.1, thickness=environment_thickness,
  Sqw_coh=environment, concentric=1, verbose=0, order=1, d_phi=2*RAD2DEG*atan2(1, LSD)
) WHEN (environment_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND %{
  flag_env += SCATTERED;
%}
```

A

```
/* sample container */
COMPONENT Container_in=Isotropic_Sqw(
  xwidth  = sample_width+1e-4+container_thickness,
  zdepth  = sample_thickness+1e-4+container_thickness,
  yheight = sample_height, thickness=container_thickness,
  Sqw_coh=container, concentric=1, verbose=0, order=1, d_phi=2*RAD2DEG*atan2(1, LSD))
  ) WHEN(container_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND
%{
  flag_env += SCATTERED;
%}
```

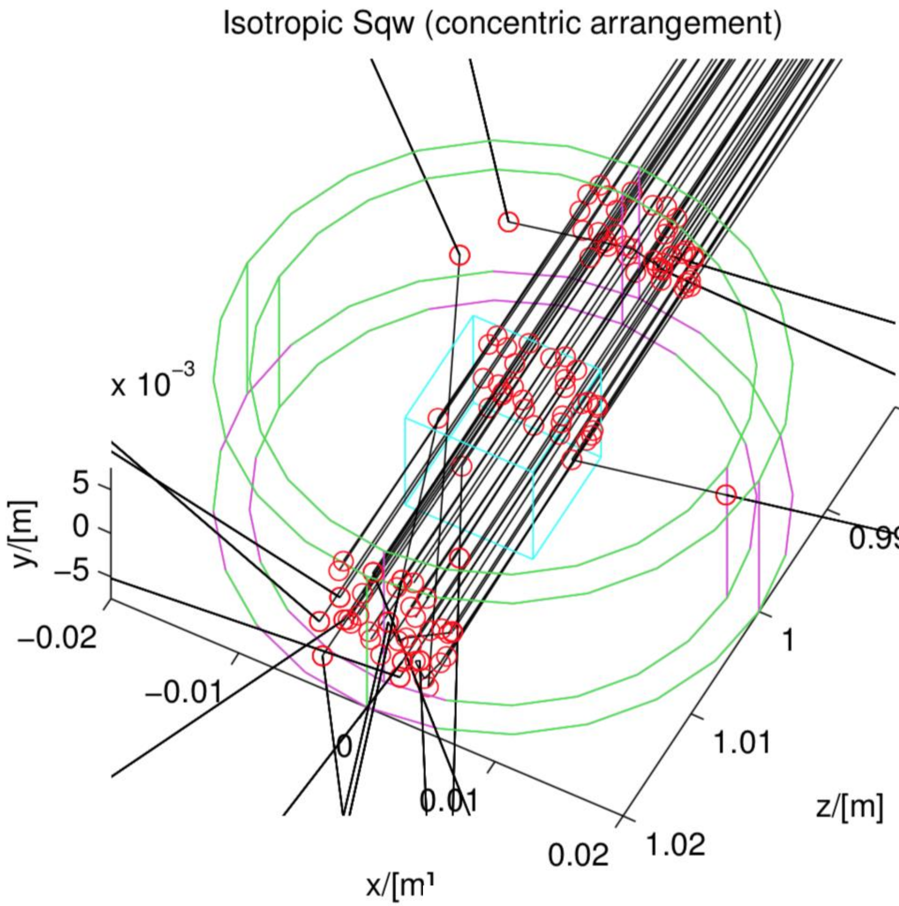
B

```
COMPONENT SampleS=Isotropic_Sqw(
  xwidth = sample_width, zdepth=sample_thickness, yheight = sample_height,
  Sqw_coh= sample_coh, Sqw_inc= sample_inc, p_interact=0.9,
  d_phi=2*RAD2DEG*atan2(1, LSD), order=1)
WHEN (flag_sample_choice == 1)
AT (0, 0, 0)   RELATIVE Sample_rot
EXTEND
%{
  flag_sample += SCATTERED;
%}

COMPONENT SampleV=Incoherent(xwidth = sample_width, zdepth=sample_thickness, yheight = sample_hei
  focus_ah = 2*RAD2DEG*atan2(1, LSD), focus_aw=150.0)
WHEN (flag_sample_choice == 2)
AT (0, 0, 0)   RELATIVE Sample_rot
EXTEND
%{
  flag_sample += SCATTERED;
%}

COMPONENT Container_out=COPY(Container_in)(concentric=0)
WHEN(container_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND
%{
  flag_env += SCATTERED;
%}
```

B'

```
/* external shield */
COMPONENT Environment_out=COPY(Environment_in)(concentric=0)
WHEN (environment_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND %{
  flag_env += SCATTERED;
%}
```

A'



"concentric" geometries

Sample + environment - concentric geometries

A

```
/* external shield */
COMPONENT Environment_in=Isotropic_Sqw(
  radius = environment_radius, yheight = 0.1, thickness=environment_thickness,
  Sqw_coh=environment, concentric=1, verbose=0, order=1, d_phi=2*RAD2DEG*atan2(1, LSD)
) WHEN (environment_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND %{
  flag_env += SCATTERED;
%}
```

B

```
/* sample container */
COMPONENT Container_in=Isotropic_Sqw(
  xwidth  = sample_width+1e-4+container_thickness,
  zdepth  = sample_thickness+1e-4+container_thickness,
  yheight = sample_height, thickness=container_thickness,
  Sqw_coh=container, concentric=1, verbose=0, order=1, d_phi=2*RAD2DEG*atan2(1, LSD)
  ) WHEN(container_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND
%{
  flag_env += SCATTERED;
%}
```

C

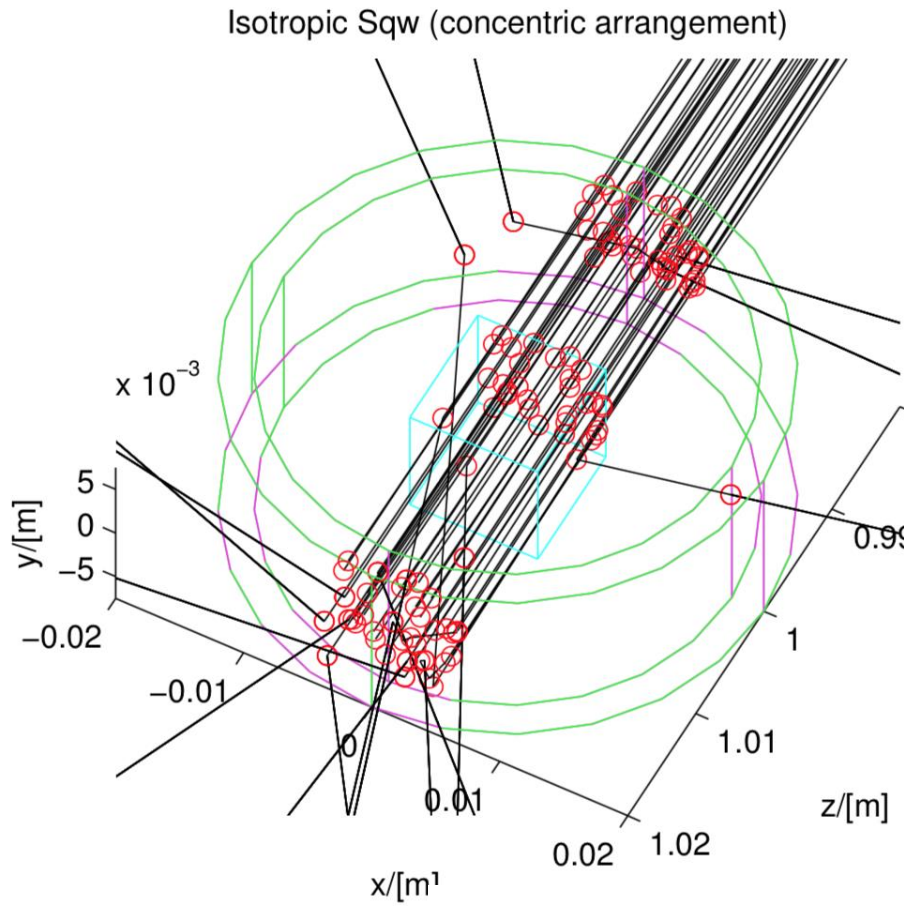
```
COMPONENT SampleS=Isotropic_Sqw(
  xwidth = sample_width, zdepth=sample_thickness, yheight = sample_height,
  Sqw_coh= sample_coh, Sqw_inc= sample_inc, p_interact=0.9,
  d_phi=2*RAD2DEG*atan2(1, LSD), order=1)
WHEN (flag_sample_choice == 1)
AT (0, 0, 0)   RELATIVE Sample_rot
EXTEND
%{
  flag_sample += SCATTERED;
%}

COMPONENT SampleV=Incoherent(xwidth = sample_width, zdepth=sample_thickness, yheight = sample_hei
  focus_ah = 2*RAD2DEG*atan2(1, LSD), focus_aw=150.0)
WHEN (flag_sample_choice == 2)
AT (0, 0, 0)   RELATIVE Sample_rot
EXTEND
%{
  flag_sample += SCATTERED;
%}
```

B'

```
COMPONENT Container_out=COPY(Container_in)(concentric=0)
WHEN(container_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND
%{
  flag_env += SCATTERED;
%}
```

A'

```
/* external shield */
COMPONENT Environment_out=COPY(Environment_in)(concentric=0)
WHEN (environment_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND %{
  flag_env += SCATTERED;
%}
```



"concentric" geometries

Sample + environment - concentric geometries

```
/* external shield */
COMPONENT Environment_in=Isotropic_Sqw(
  radius = environment_radius, yheight = 0.1, thickness=environment_thickness,
  Sqw_coh=environment, concentric=1, verbose=0, order=1, d_phi=2*RAD2DEG*atan2(1, LSD)
) WHEN (environment_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND %{
  flag_env += SCATTERED;
%}
```

A

```
/* sample container */
COMPONENT Container_in=Isotropic_Sqw(
  xwidth  = sample_width+1e-4+container_thickness,
  zdepth  = sample_thickness+1e-4+container_thickness,
  yheight = sample_height, thickness=container_thickness,
  Sqw_coh=container, concentric=1, verbose=0, order=1, d_phi=2*RAD2DEG*atan2(1, LSD)
  ) WHEN(container_thickness > 0)
AT (0, 0, 0) RELATIVE Sample_rot
EXTEND
%{
  flag_env += SCATTERED;
%}
```

B

```
COMPONENT SampleS=Isotropic_Sqw(
  xwidth = sample_width, zdepth=sample_thickness, yheight = sample_height,
  Sqw_coh= sample_coh, Sqw_inc= sample_inc, p_interact=0.9,
  d_phi=2*RAD2DEG*atan2(1, LSD), order=1)
WHEN (flag_sample_choice == 1)
AT (0, 0, 0) RELATIVE Sample_rot
EXTE
%{
  fl
%}

COMP
  fc
WHEN
AT (
EXTE
%{
  fl
%}

COMP
WHEN
AT (
EXTE
%{
  fl
%}

/* external
COMPONENT En
WHEN (envir
AT (0, 0, 0)
EXTEND %{
  flag_env += SCATTERED;
%}
```

C

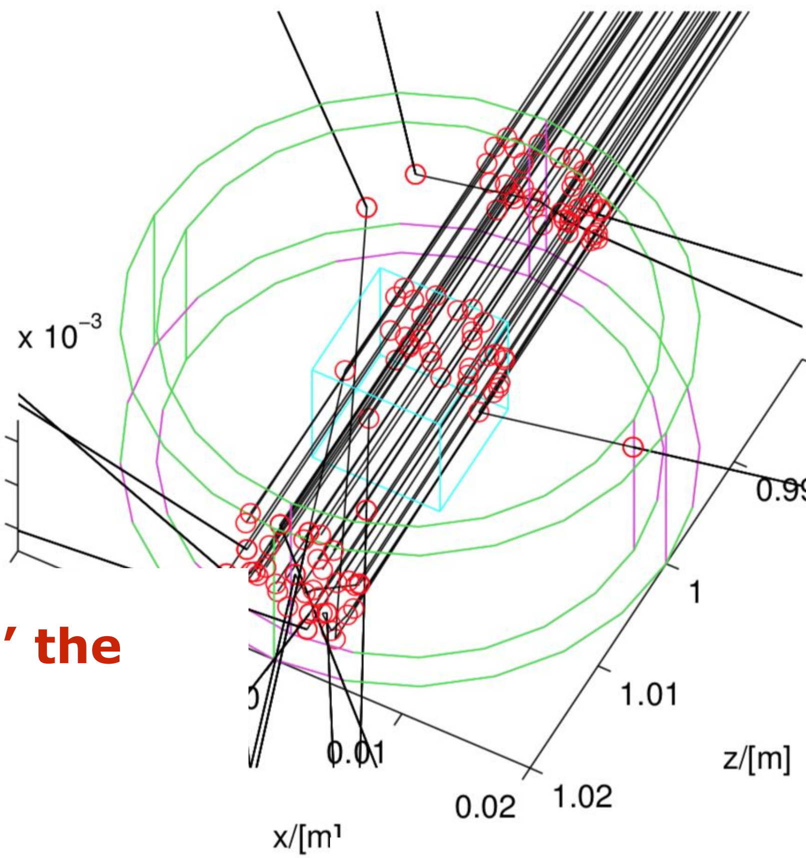
A'

Main limitations:

* Intrinsic "linear" component sequence
* No 'overlapping' volumes
* Multiple scattering not fully described:
  * multiple scattering 'between' the
  * 'inside' a single component
* No "one-stop" solution

Main limitations:

* Intrinsic "linear" component sequence
* No 'overlapping' volumes
* Multiple scattering not fully described:
  * multiple scattering 'between' the
  * 'inside' a single component



component

geometries

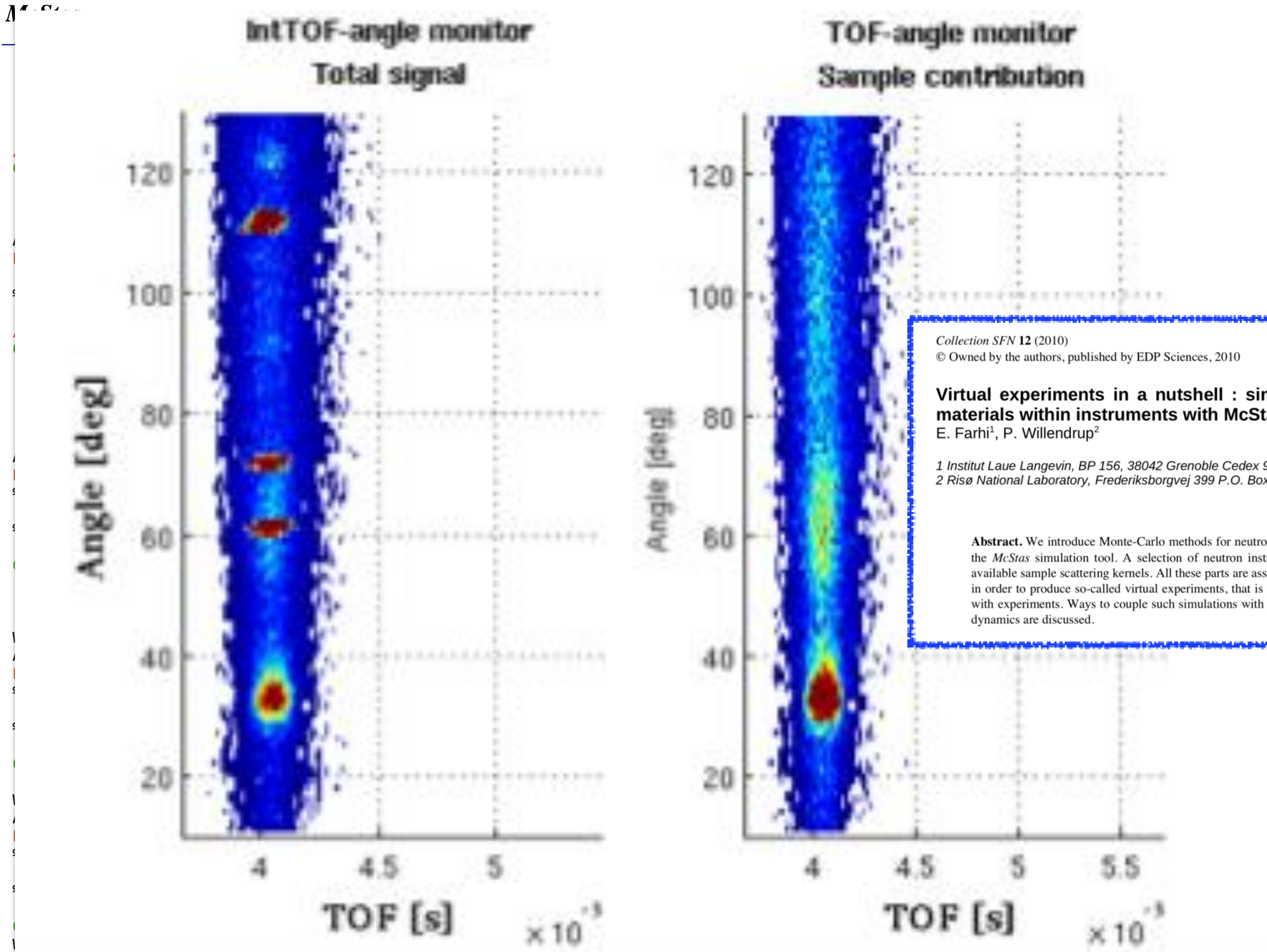


Figure 22. The time-of-flight/angle signal acquired from the liquid spectrometer with sample environment model. The total signal from liquid rubidium in a Nb container and Al external shield is shown on the left, whereas the sample-only contribution is shown on the right. Intensity is shown in log scale, with colors ranging from blue (low) to red (high).

Collection SFN **12** (2010)

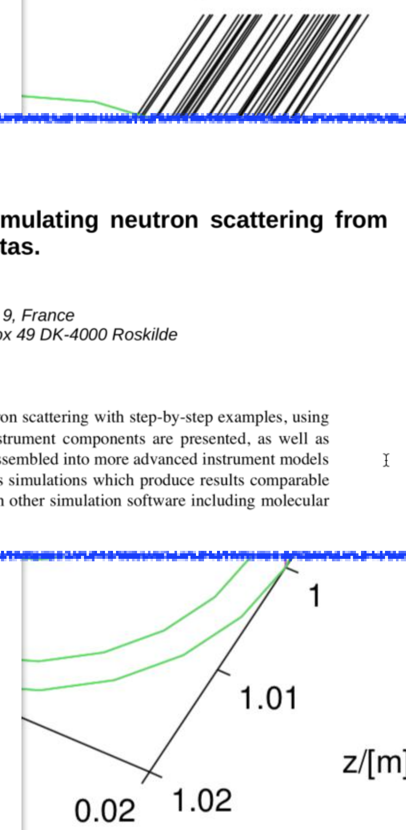
Virtual experiments in a nutshell : simulating neutron scattering from materials within instruments with McStas.

E. Farhi[1], P. Willendrup[2]

1 Institut Laue Langevin, BP 156, 38042 Grenoble Cedex 9, France
2 Risø National Laboratory, Frederiksborgvej 399 P.O. Box 49 DK-4000 Roskilde

Abstract. We introduce Monte-Carlo methods for neutron scattering with step-by-step examples, using the *McStas* simulation tool. A selection of neutron instrument components are presented, as well as available sample scattering kernels. All these parts are assembled into more advanced instrument models in order to produce so-called virtual experiments, that is simulations which produce results comparable with experiments. Ways to couple such simulations with other simulation software including molecular dynamics are discussed.

centric arrangement)



tries

Union concept from Mads Bertelsen (KU)

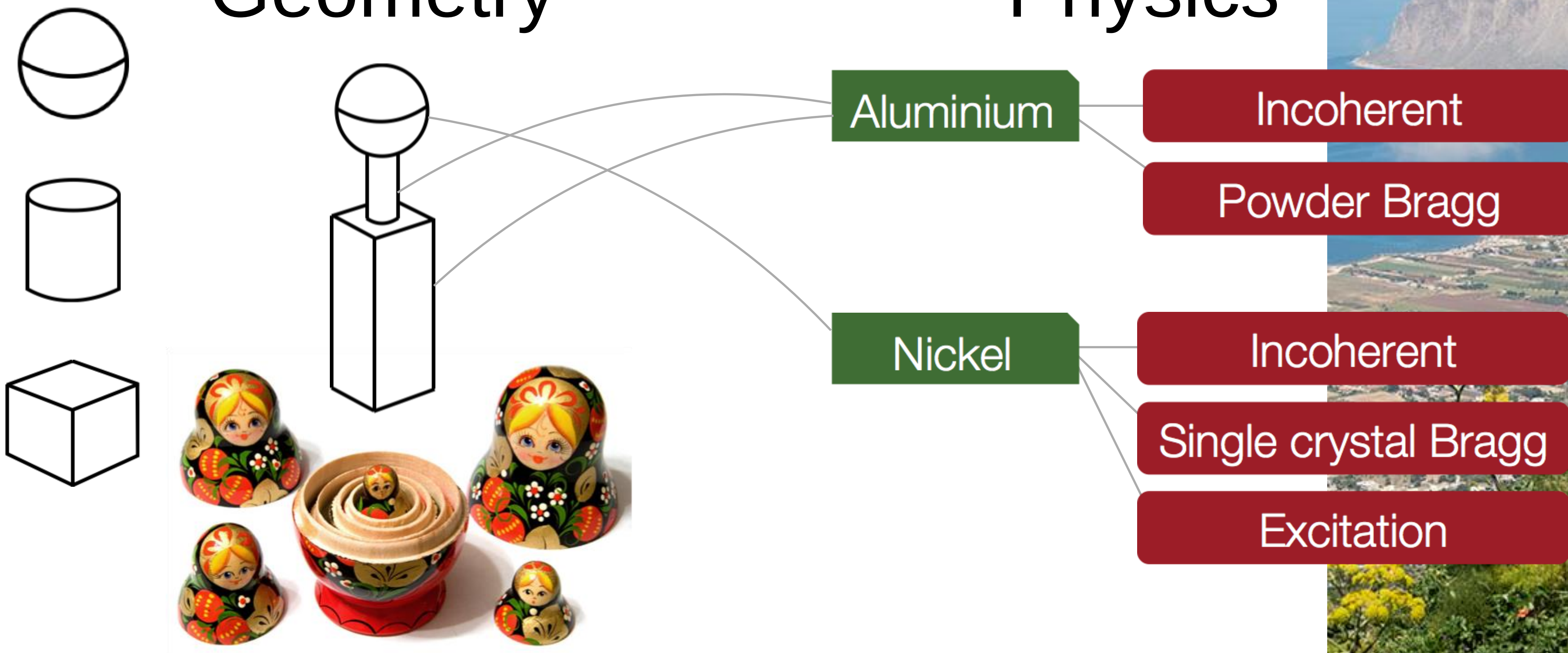
- Sample holders with complicated geometry
- Many different materials
- Inside sample environment
- Co aligned crystals
- Twinned crystals

 Slide courtesy of Mads Bertelsen, KU

McStas Union components - Idea

Geometry

Physics



Slide courtesy of Mads Bertelsen, KU

McStas Union components - Priority

- Each geometry is assigned a material definition and a priority
- Priority decides which material is simulated in regions where several overlap
- This can be used to construct complex geometries with a range of materials

 Slide courtesy of Mads Bertelsen, KU

McStas Union components - Use

sigma in [barns]
unit_cell_volume in [Å^3]

```
COMPONENT Al_incoherent = Incoherent_process(
  sigma=4*0.0082,packing_factor=1,
  unit_cell_volume=66.4)
AT (0,0,0) ABSOLUTE

COMPONENT Al_powder = Powder_process(
  reflections="Al.laz")
AT (0,0,0) ABSOLUTE

COMPONENT Al = Union_make_material(
  my_absorption=100*4*0.231/66.4,
  process_string="Al_incoherent,Al_powder")
AT (0,0,0) ABSOLUTE
```

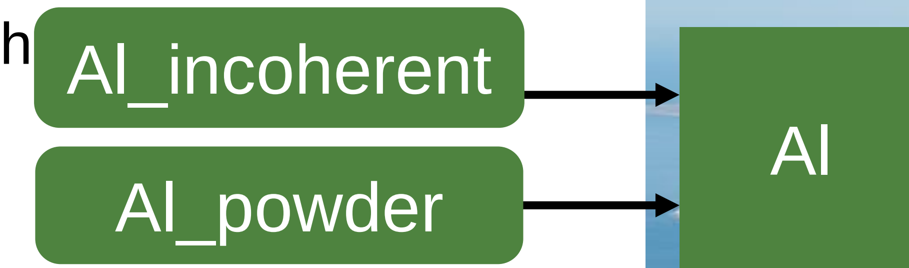
my [1/m] = cross section per unit cell / unit cell volume

 Slide courtesy of Mads Bertelsen, KU

McStas Union components - Use

sigma in [barns]
unit_cell_volume in [Å^3]

```
COMPONENT Al_incoherent = Incoh
  sigma=4*0.0082,packing_factor=1,
  unit_cell_volume=66.4)
AT (0,0,0) ABSOLUTE

COMPONENT Al_powder = Powder_process(
  reflections="Al.laz")
AT (0,0,0) ABSOLUTE

COMPONENT Al = Union_make_material(
  my_absorption=100*4*0.231/66.4,
  process_string=“Al_incoherent,Al_powder”)
AT (0,0,0) ABSOLUTE
```



my [1/m] = cross section per unit cell / unit cell volume

McStas Union components - Use

Uses our Al definition!

```
COMPONENT cryostat_shell = Union_cylinder(
  radius_input=0.15,height_input=0.4,
  priority_input=10,material_string="Al")
AT (0,0.0,0) RELATIVE target
ROTATED (0,0,0) RELATIVE target
```

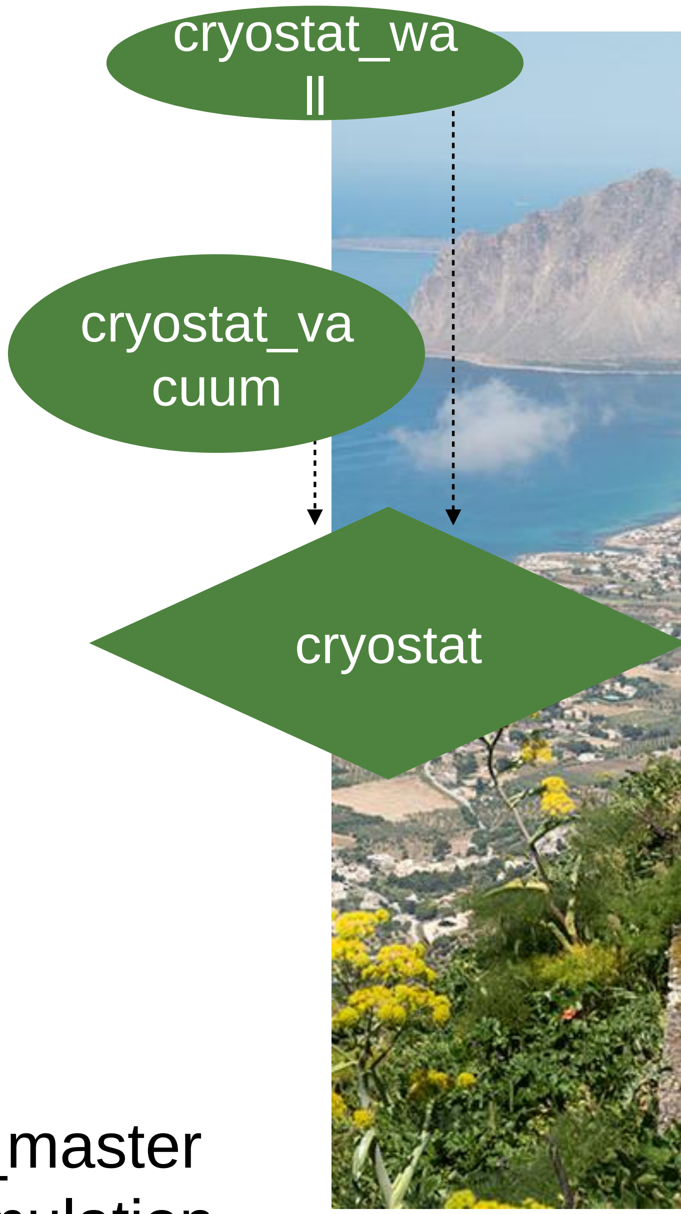
Uses default material definition

```
COMPONENT cryostat_vacuum = Union_cylinder(
  radius_input=0.147,height_input=0.4,
  priority_input=11,material_string="Vacuum")
AT (0,0.0,0) RELATIVE target
ROTATED (0,0,0) RELATIVE target
```

Does not do any simulation what so ever

Slide courtesy of Mads Bertelsen, KU

McStas Union components - Use

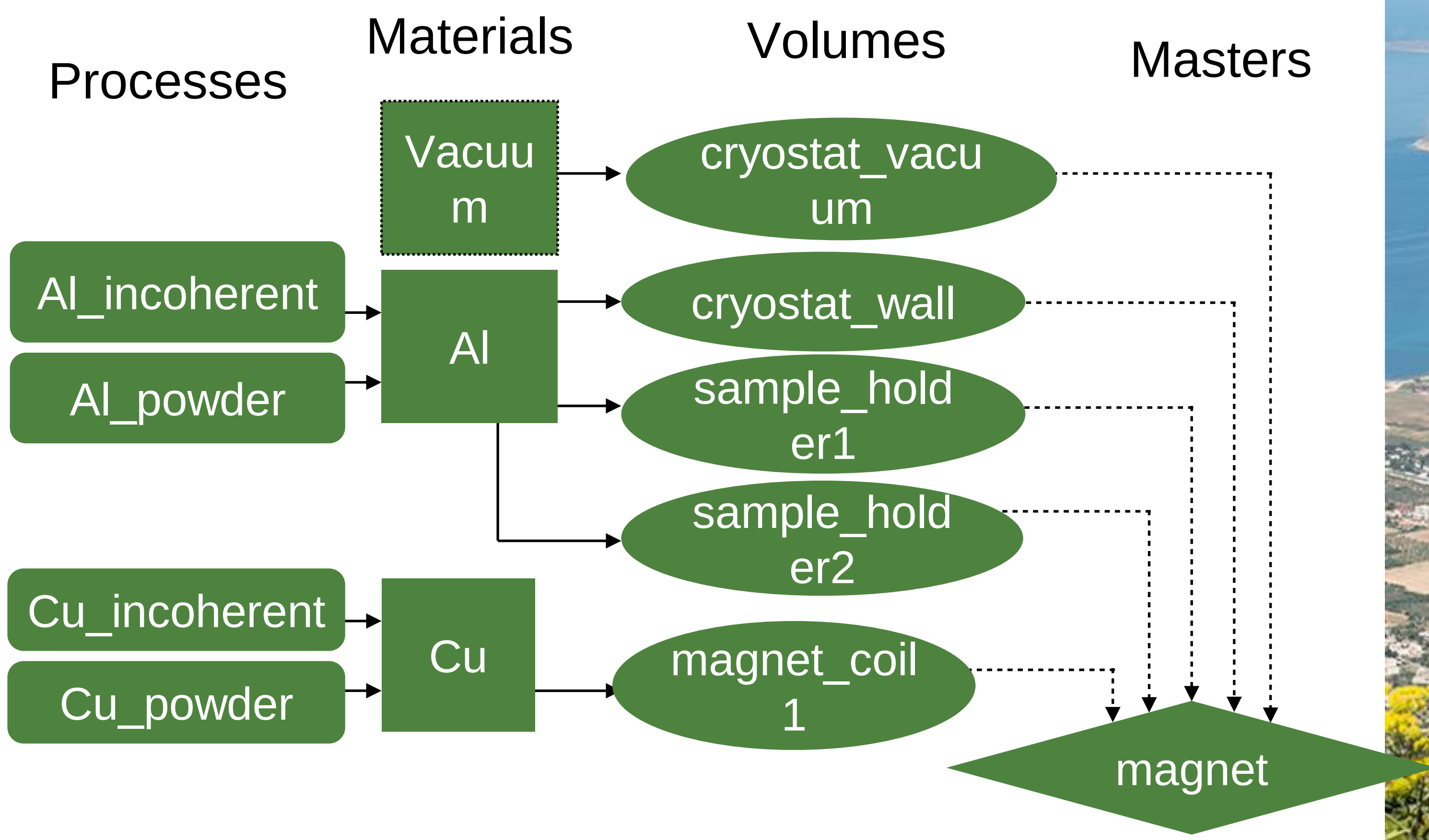



```
COMPONENT cryostat_shell = Union_cylinder(
  radius_input=0.15,height_input=0.4,
  priority_input=10,material_string="Al")
AT (0,0.0,0) RELATIVE target
ROTATED (0,0,0) RELATIVE target

COMPONENT cryostat_vacuum = Union_cylinder(
  radius_input=0.147,height_input=0.4,
  priority_input=11,material_string="Vacuum")
AT (0,0.0,0) RELATIVE target
ROTATED (0,0,0) RELATIVE target

COMPONENT cryostat = Union_master()
AT (0,0,0) RELATIVE target
ROTATED (0,0,0) RELATIVE target
```

The Union_master does the simulation

Slide courtesy of Mads Bertelsen, KU

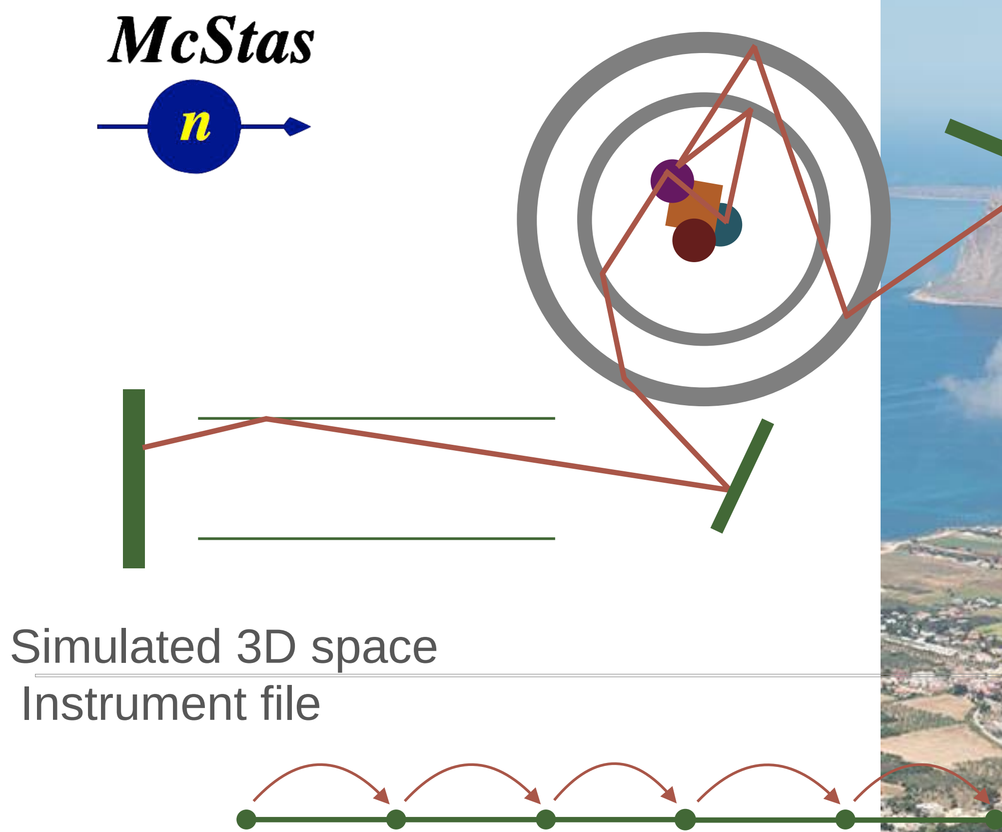
McStas Union components - Use



Slide courtesy of Mads Bertelsen, KU

Union in instrument file

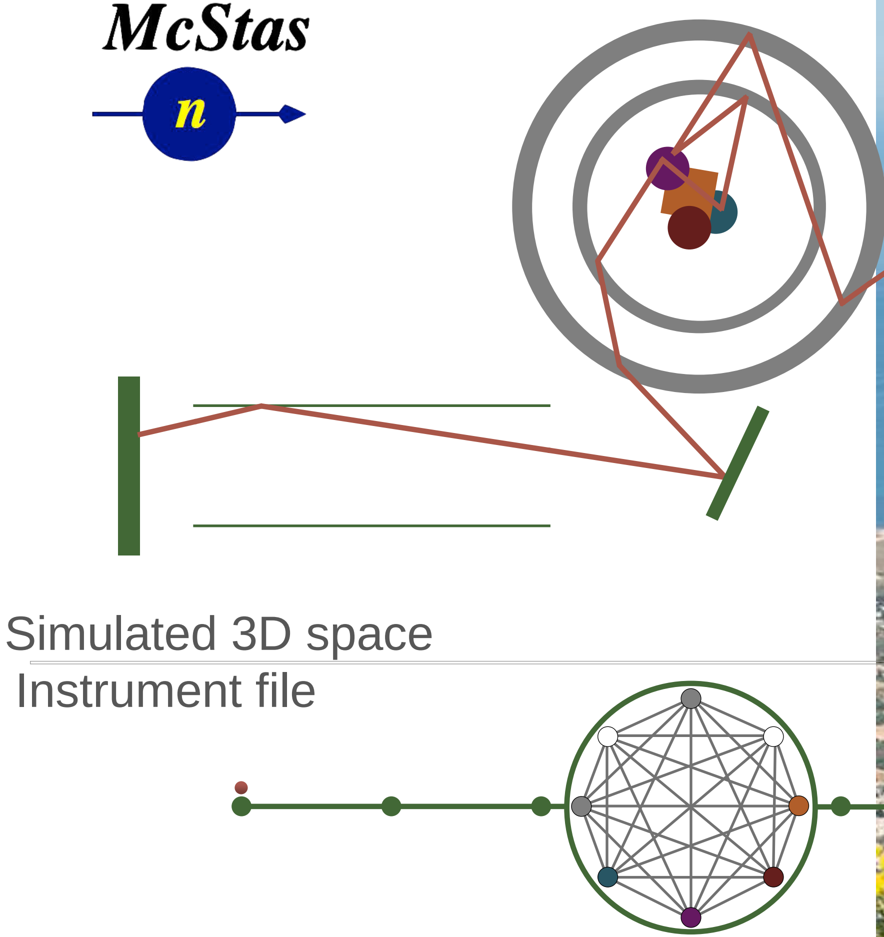
- Only the Union_master component affects the McStas simulation



Slide courtesy of Mads Bertelsen, KU

Union in instrument file

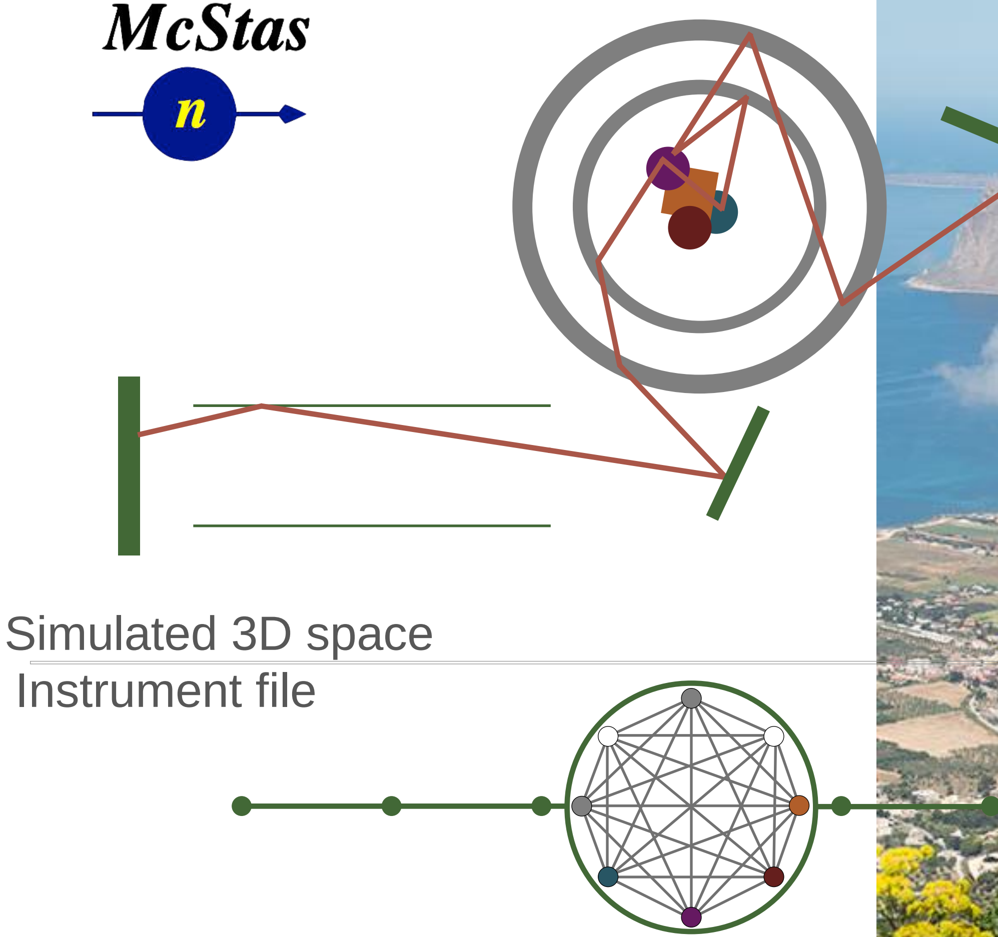
- Only the Union_master component affects the McStas simulation
- The Union_master component uses a network for propagation



Slide courtesy of Mads Bertelsen, KU

Union in instrument file

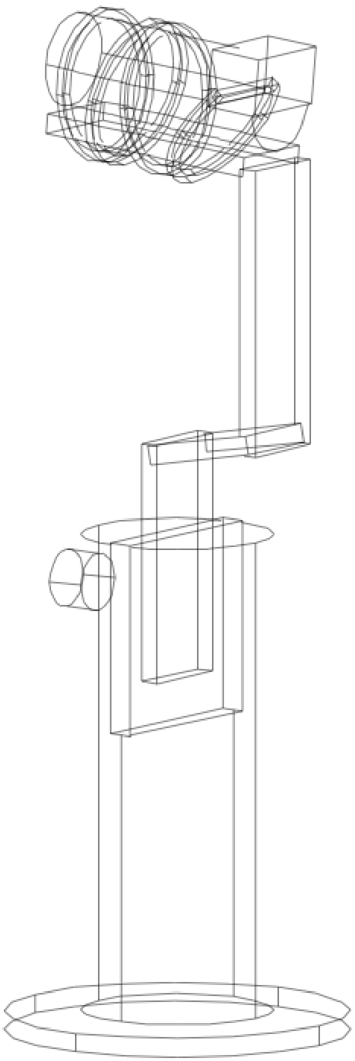
- Only the Union_master component affects the McStas simulation
- The Union_master component uses a network for propagation
- Analysis prior to simulation reduces the network complexity



Slide courtesy of Mads Bertelsen, KU

McStas Union components

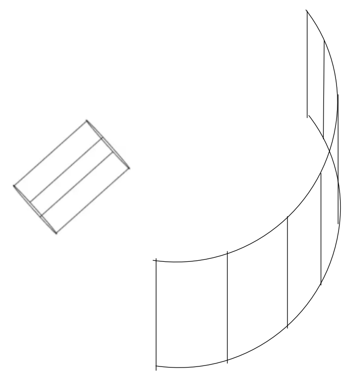
- Replicated from picture
- Easily assembled using Union components in McStas
- Material definitions made for sample / Aluminium
- Al absorption exaggerated

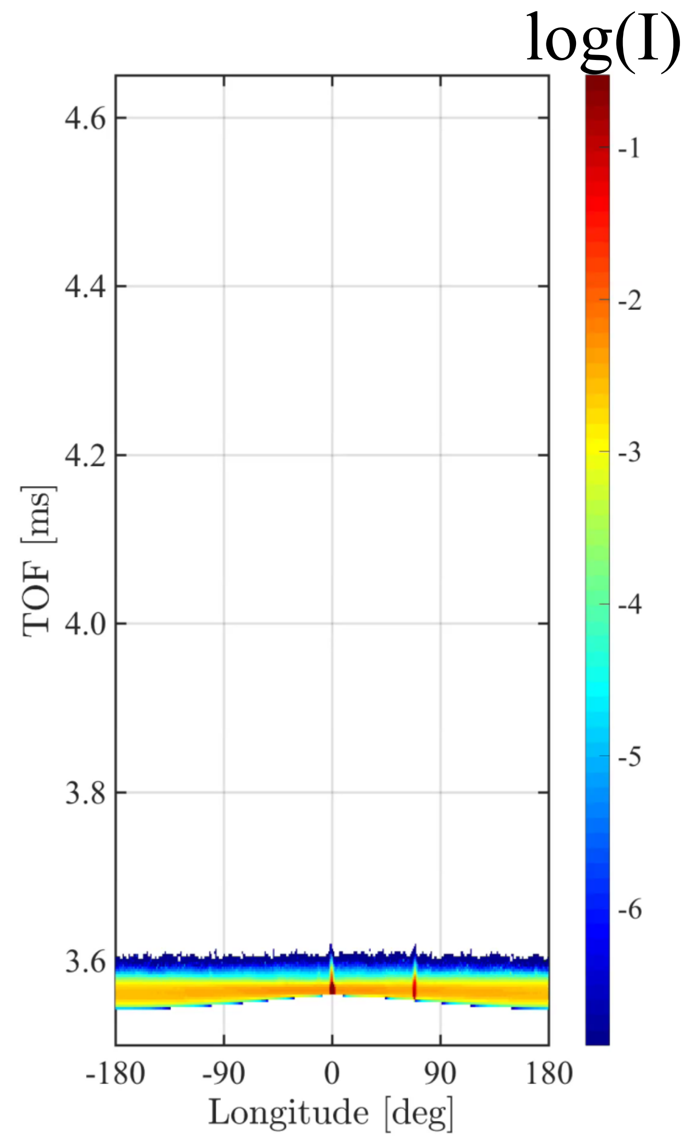


Slide courtesy of Mads Bertelsen, KU

Building a sample

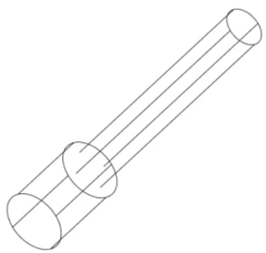
5 meV
beam

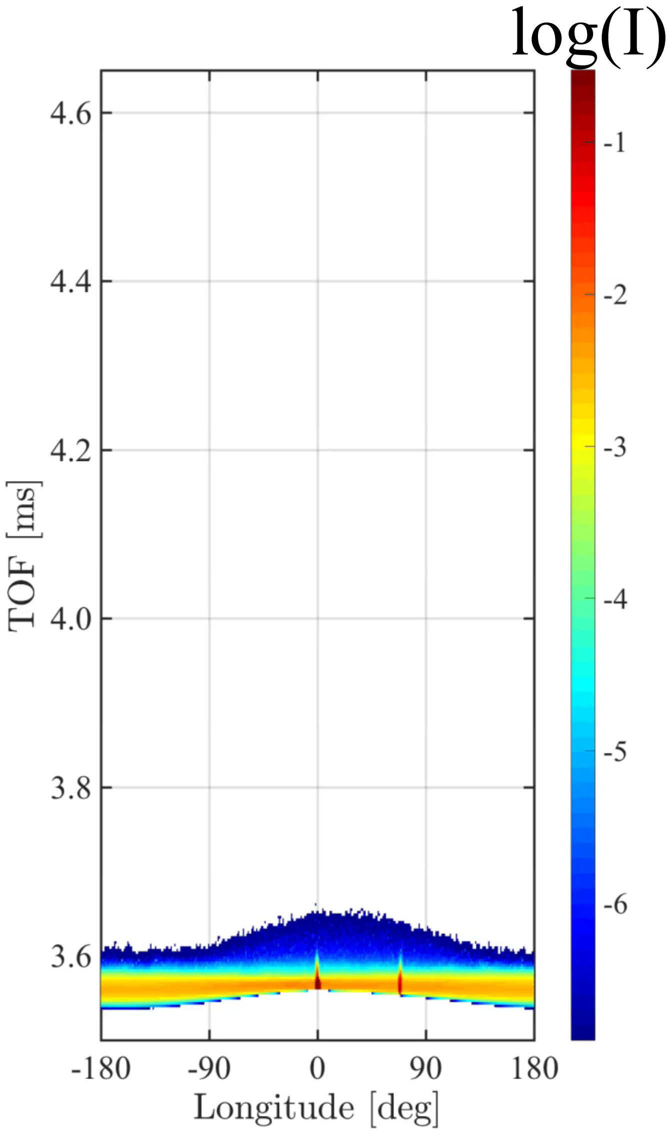




Slide courtesy of Mads Bertelsen, KU

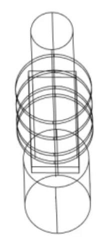
Building a sample

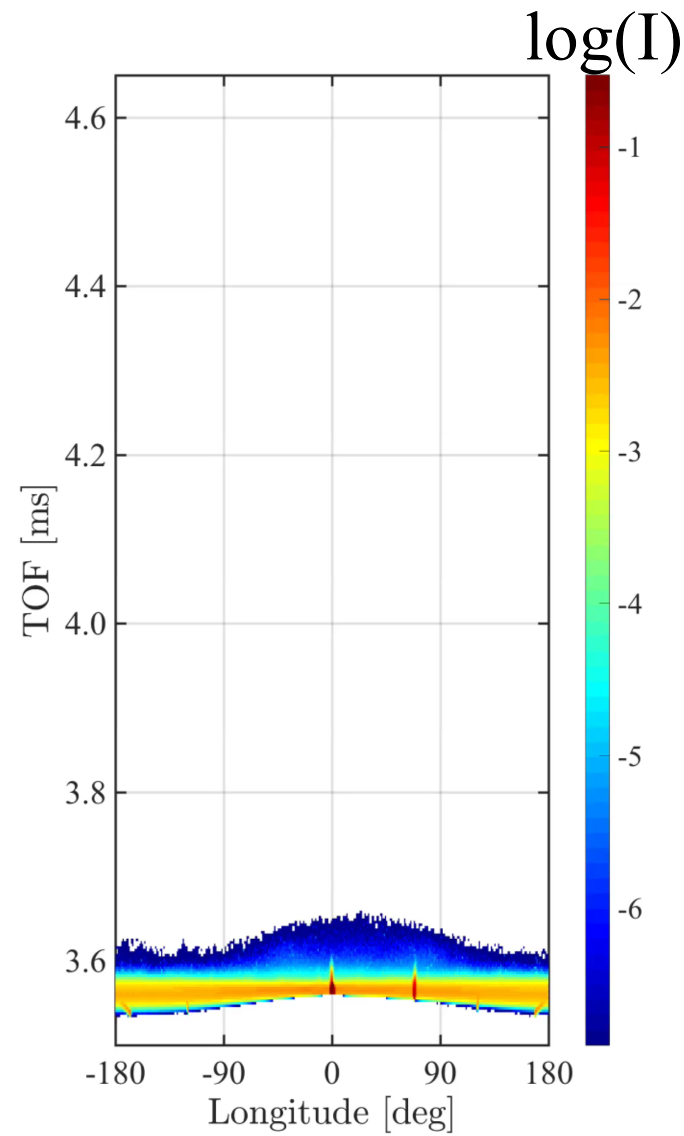




Slide courtesy of Mads Bertelsen, KU

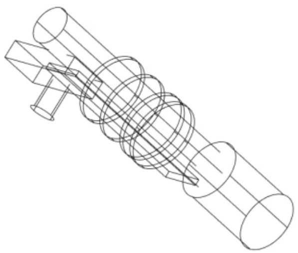
Building a sample

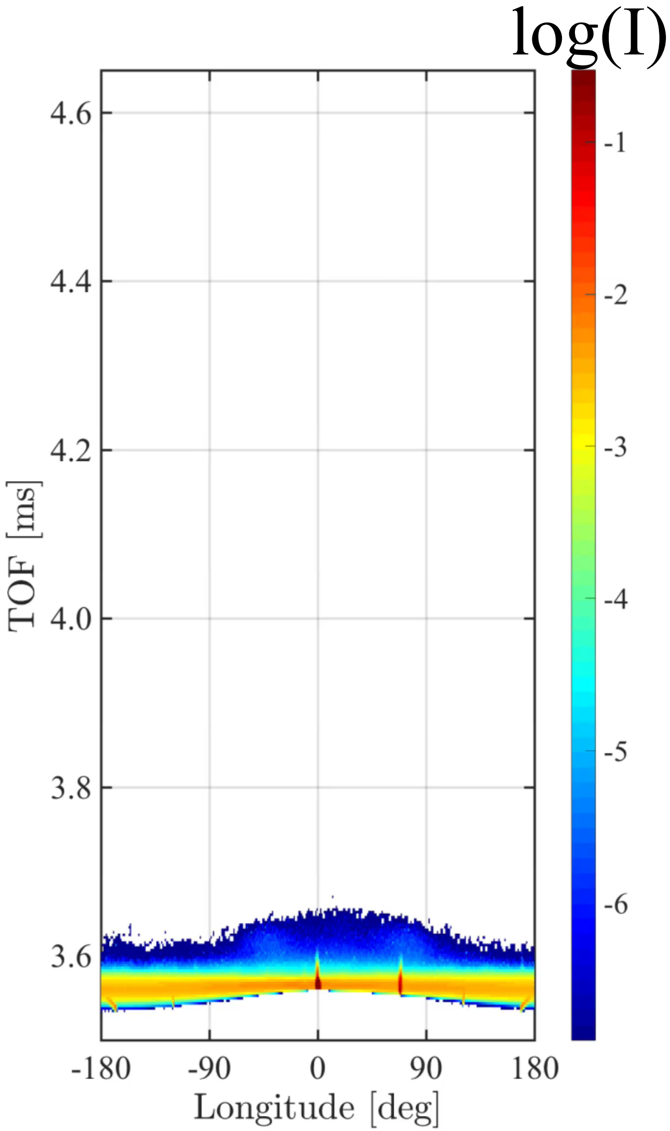




Slide courtesy of Mads Bertelsen, KU

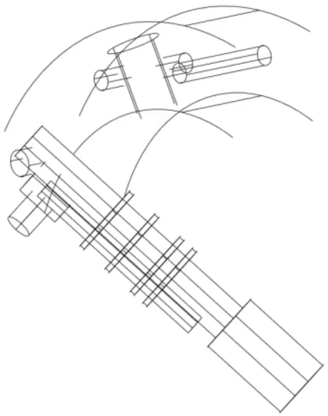
Building a sample

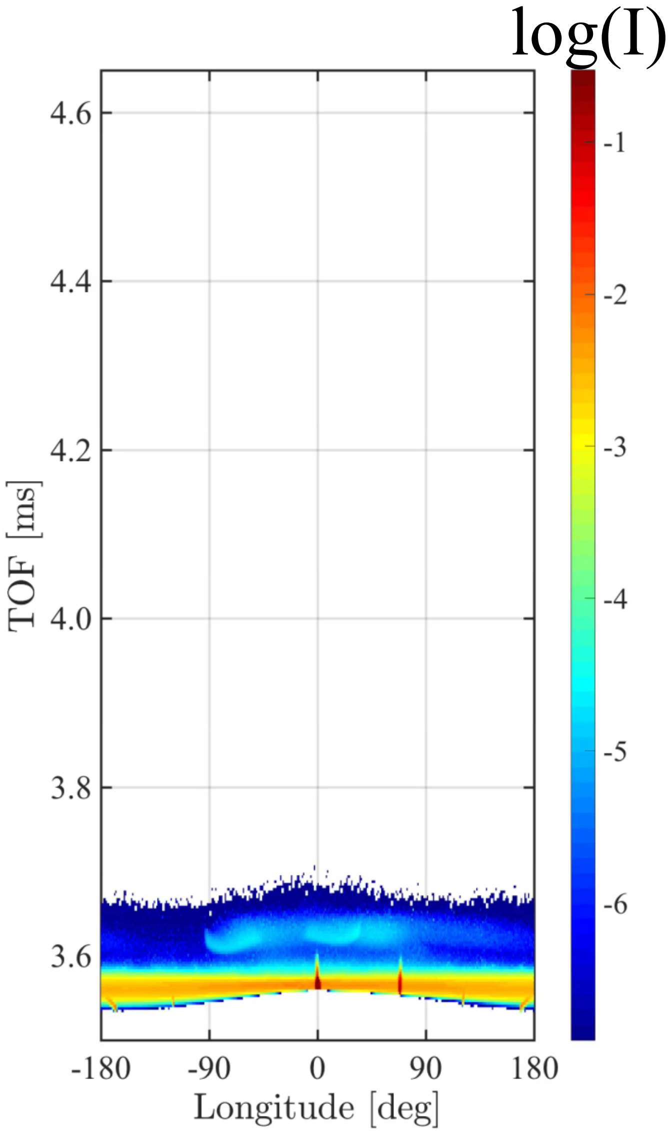




Slide courtesy of Mads Bertelsen, KU

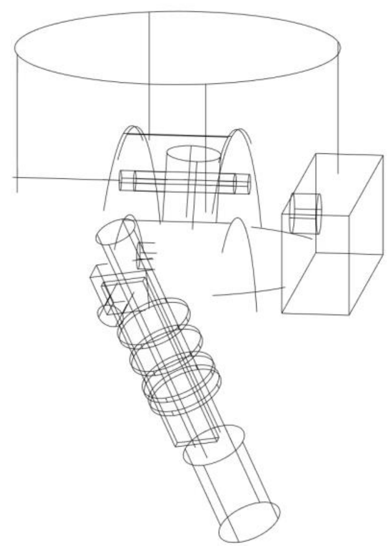
Building a sample

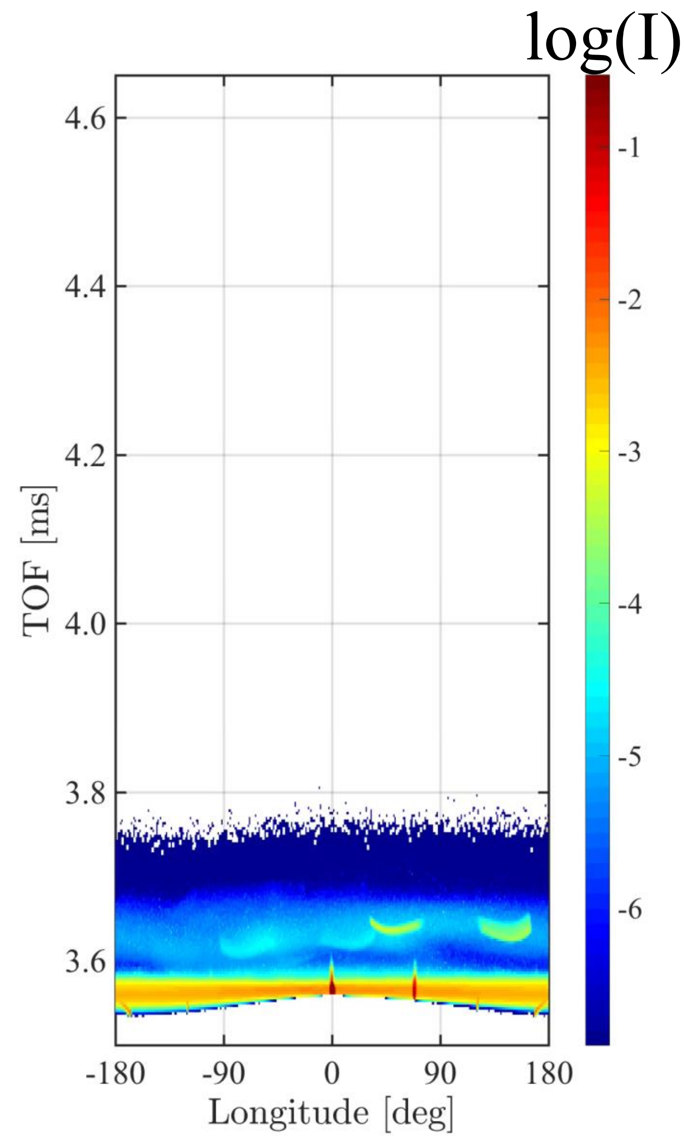




Slide courtesy of Mads Bertelsen, KU

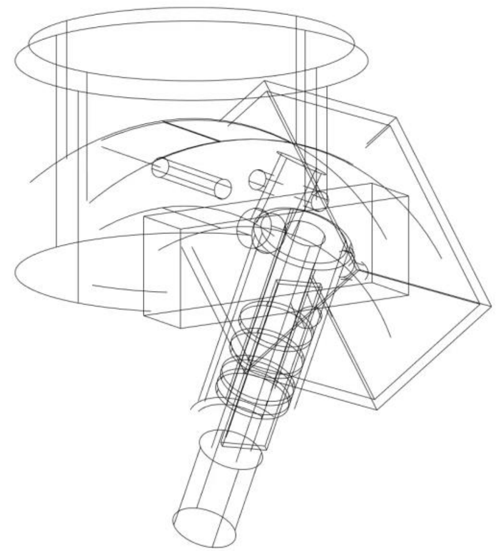
Building a sample

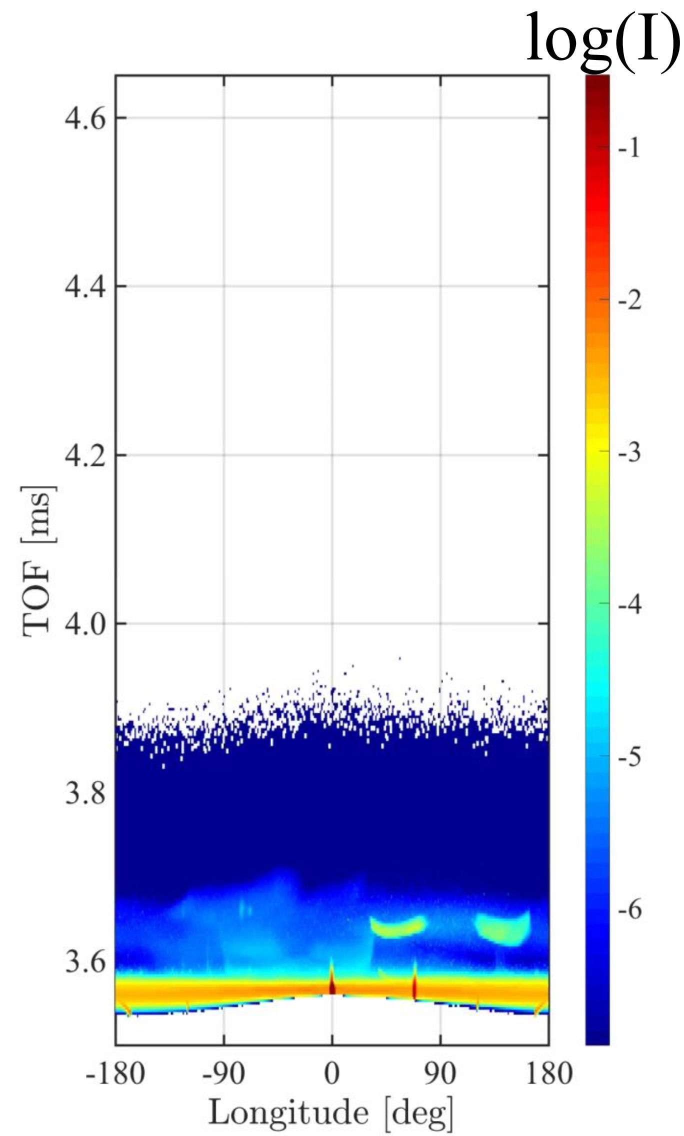




 Slide courtesy of Mads Bertelsen, KU

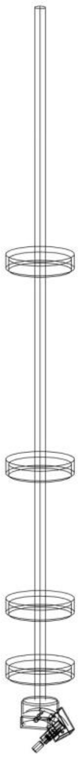
Building a sample

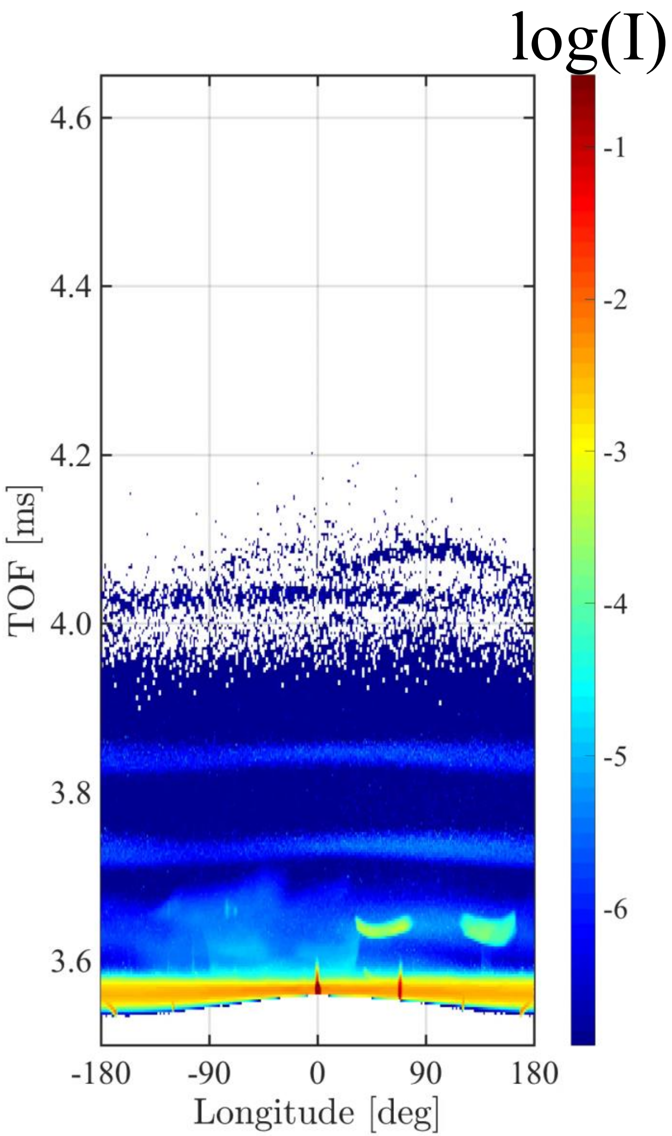




Slide courtesy of Mads Bertelsen, KU

Building a sample

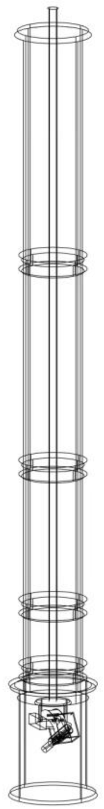


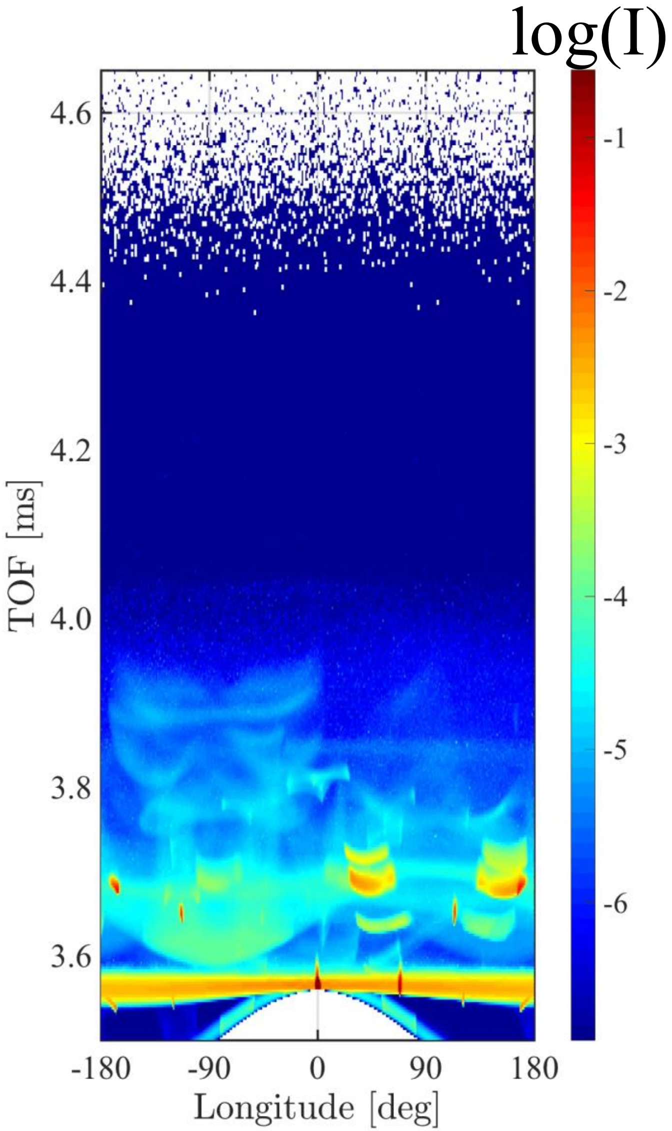


Slide courtesy of Mads Bertelsen, KU

Building a sample

Image from NIST webpage

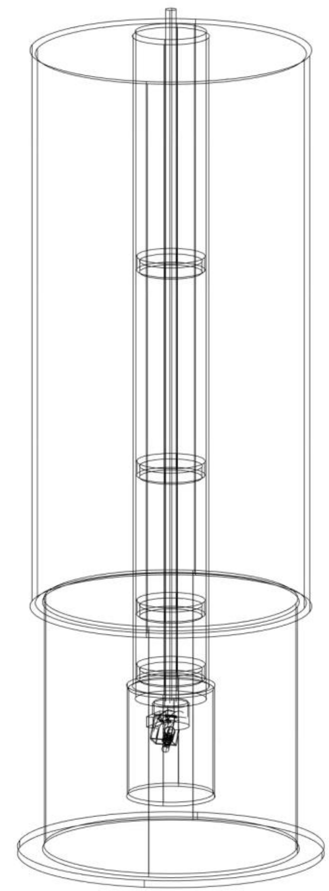


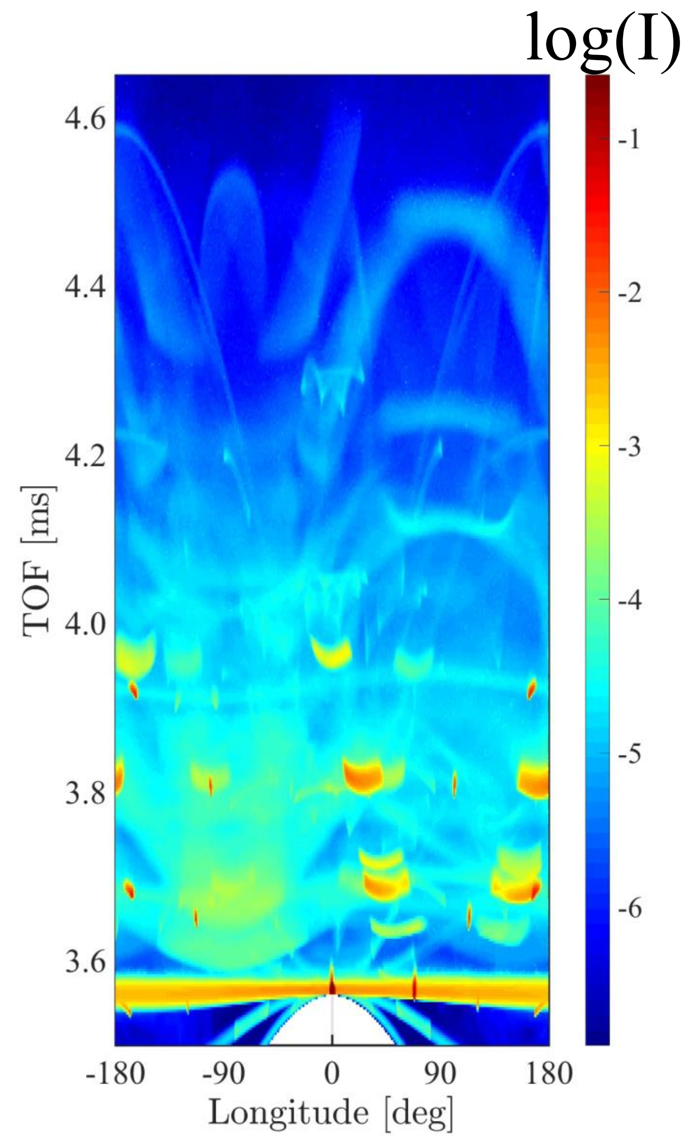


Slide courtesy of Mads Bertelsen, KU

Building a sample

Image from NIST webpage

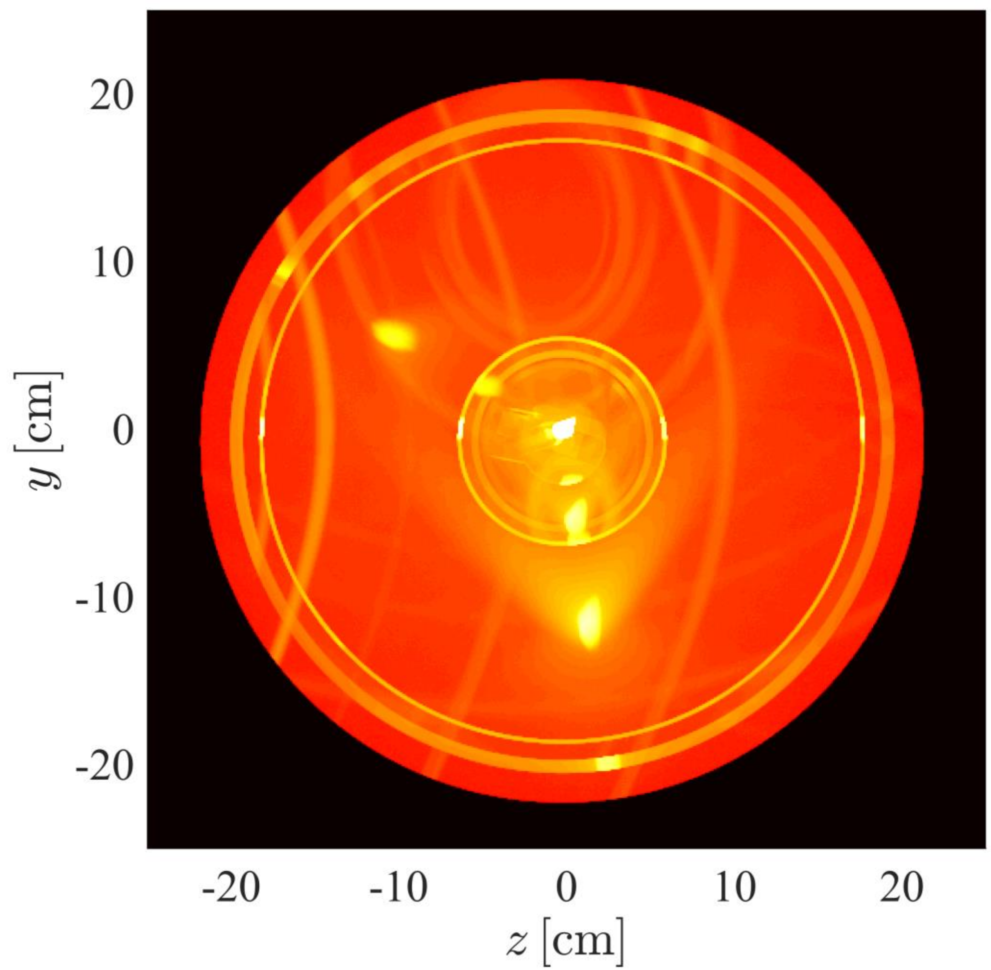




 Slide courtesy of Mads Bertelsen, KU

Union loggers

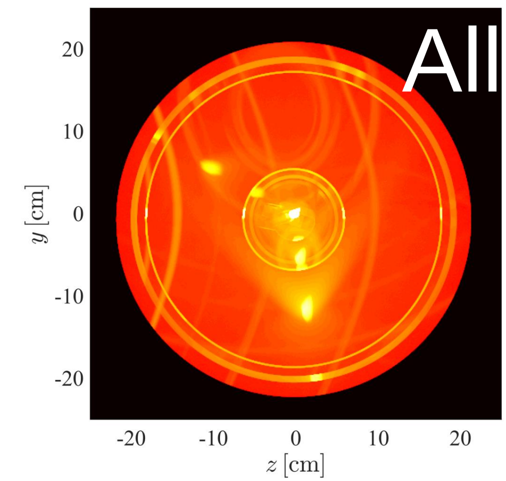
- Loggers can provide insight to what occurred during a simulation
- Here scattered intensity viewed from above the cryostat

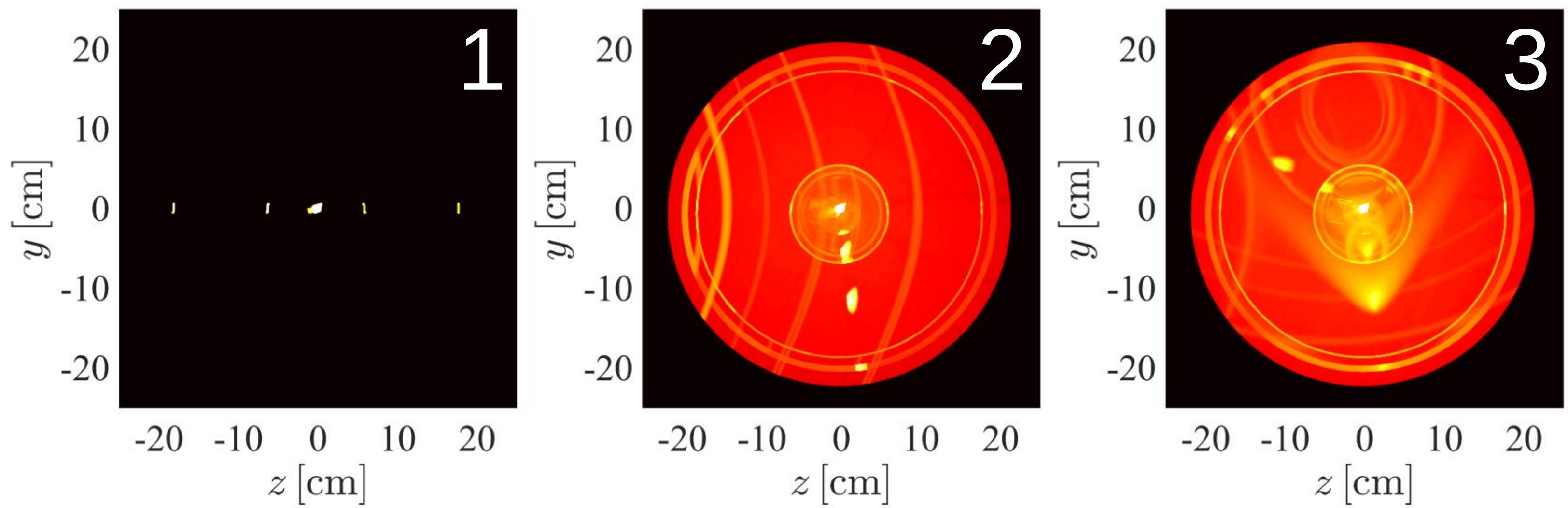


Slide courtesy of Mads Bertelsen, KU

Union loggers

- Loggers can provide insight to what occurred during a simulation
- Here scattered intensity viewed from above the cryostat

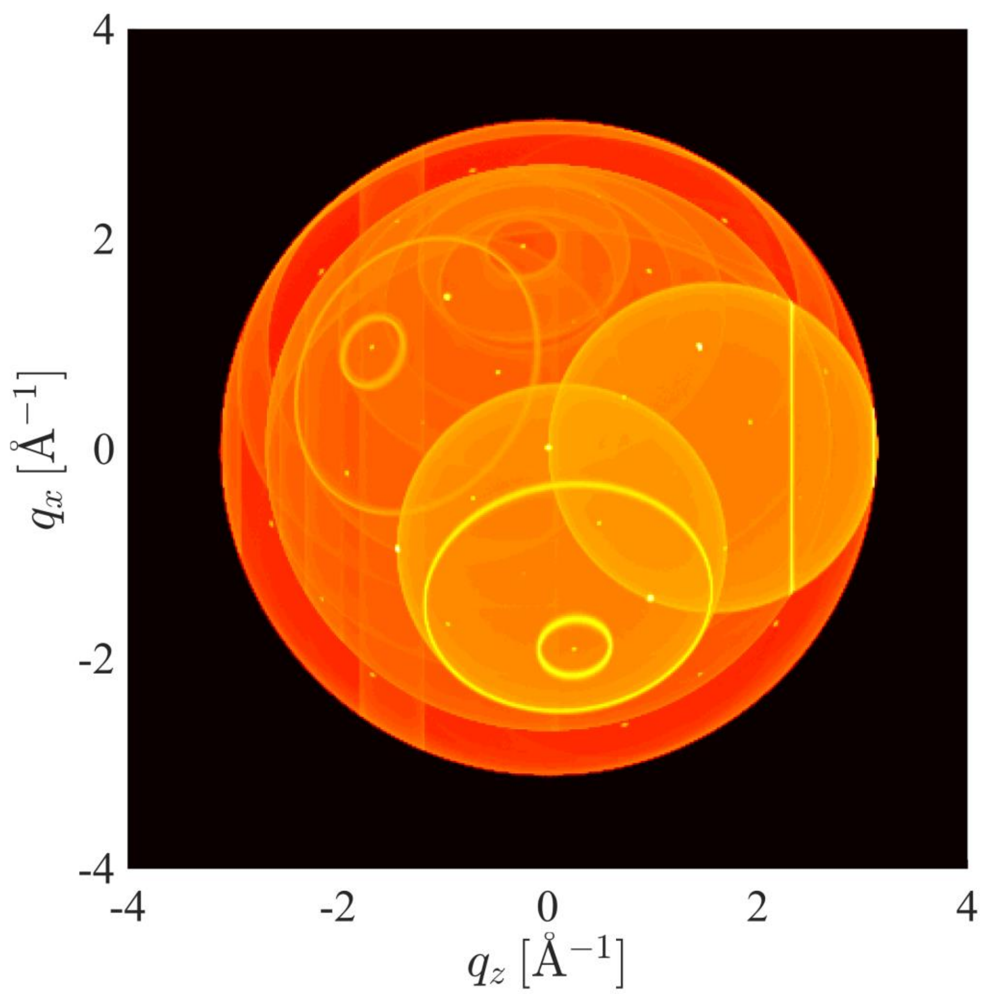




Slide courtesy of Mads Bertelsen, KU

Union loggers

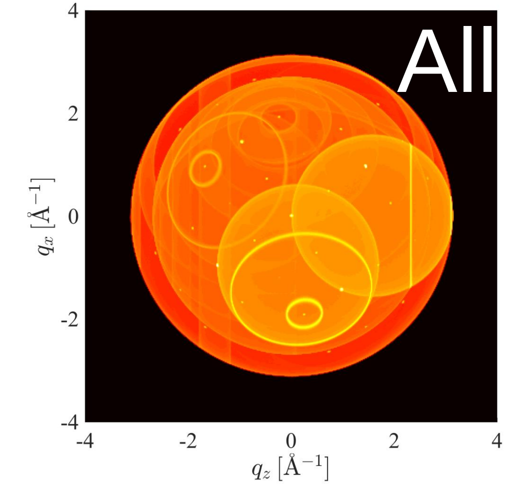
- Loggers can provide insight to what occurred during a simulation
- Here the scattering vector projected onto the scattering plane

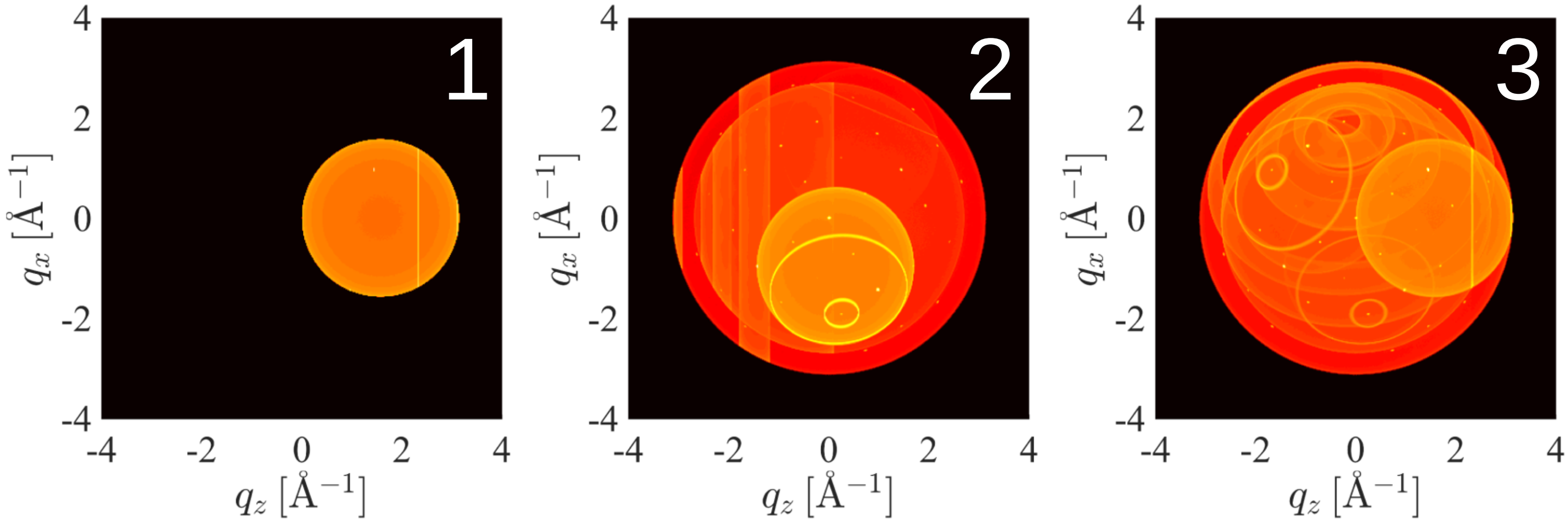


Slide courtesy of Mads Bertelsen, KU

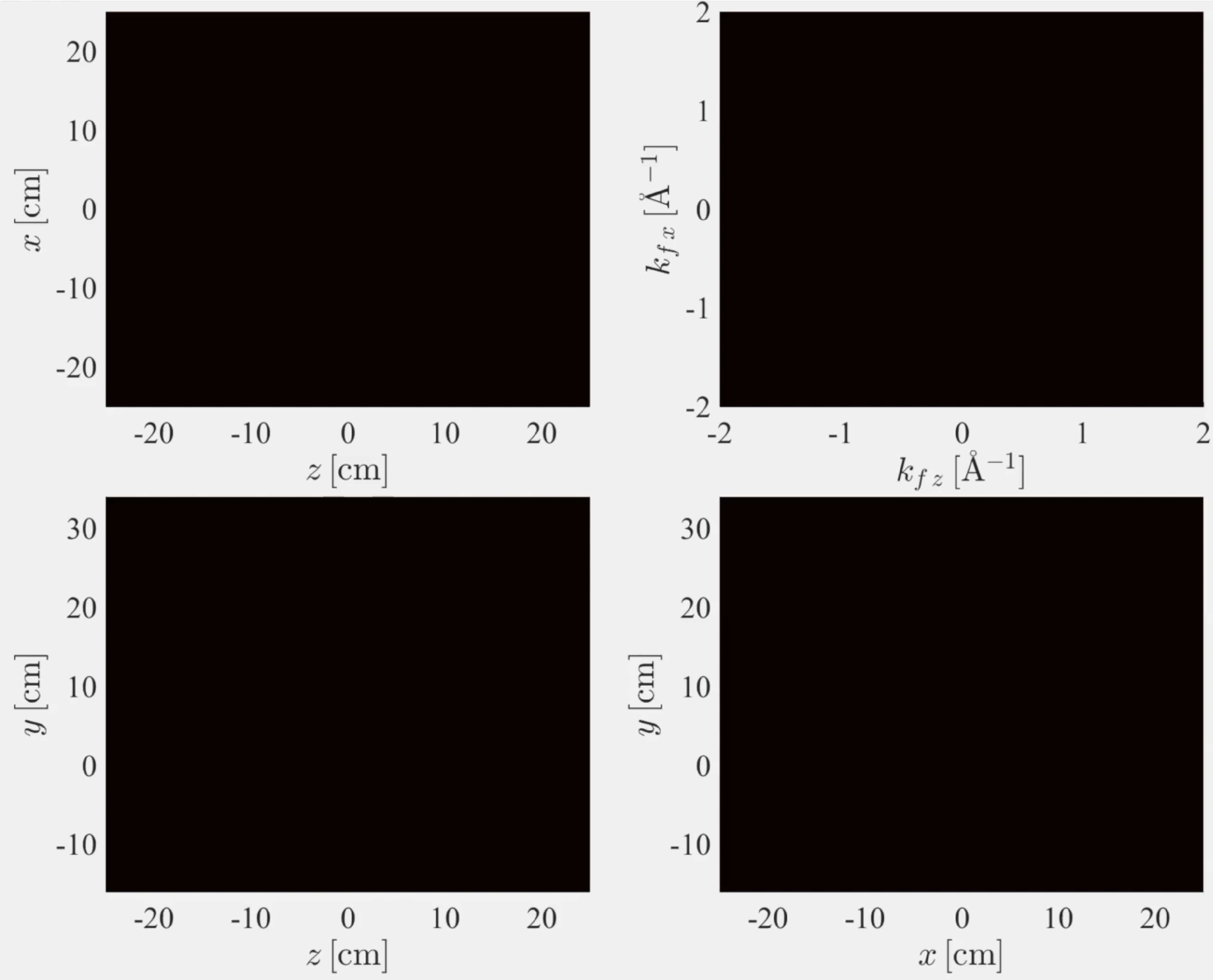
Union loggers

- Loggers can provide insight to what occurred during a simulation
- Here the scattering vector projected onto the scattering plane





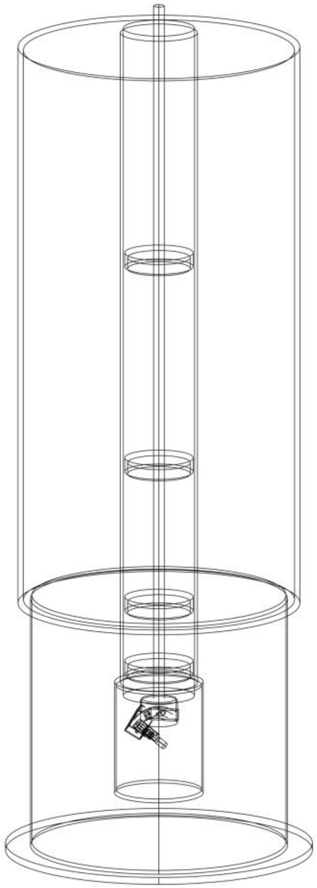
Slide courtesy of Mads Bertelsen, KU

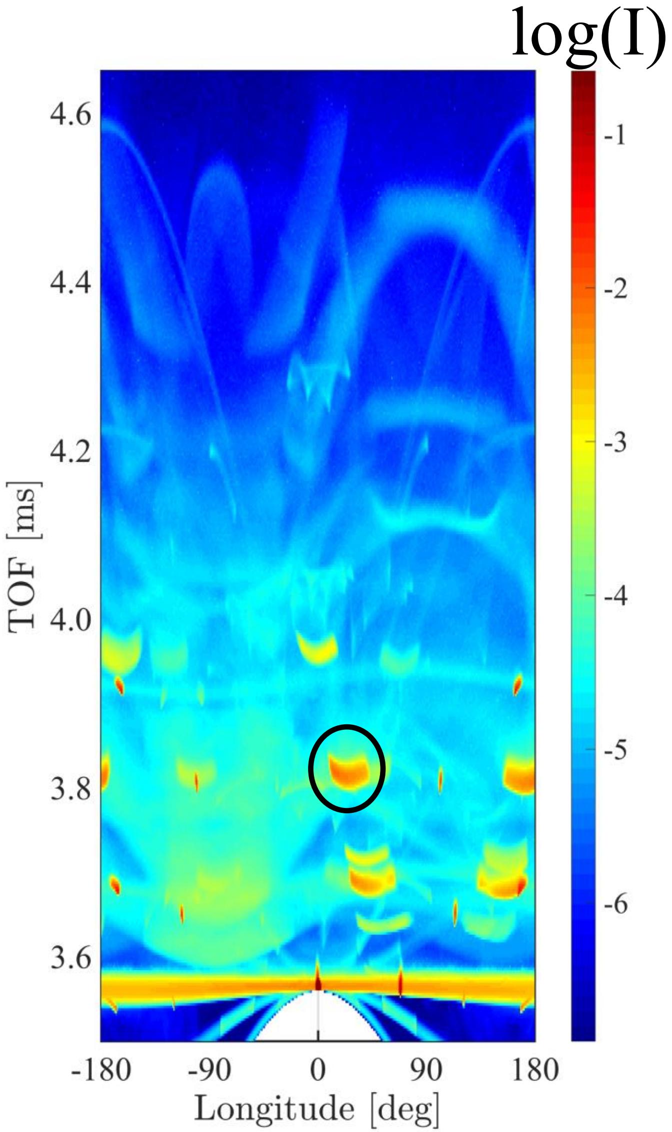


x [cm]
z [cm]
$k_{f\,x}$ [Å^{-1}]
$k_{f\,z}$ [Å^{-1}]
y [cm]
z [cm]
y [cm]
x [cm]

McStas Union conditionals

- Necessary to understand origin of specific parts of background
- Union components contains conditional tools

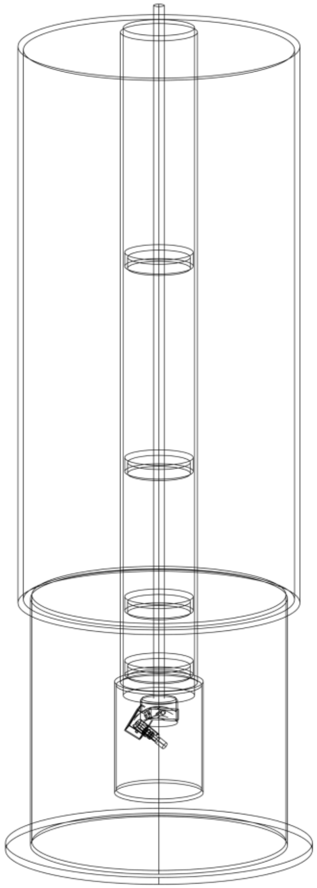


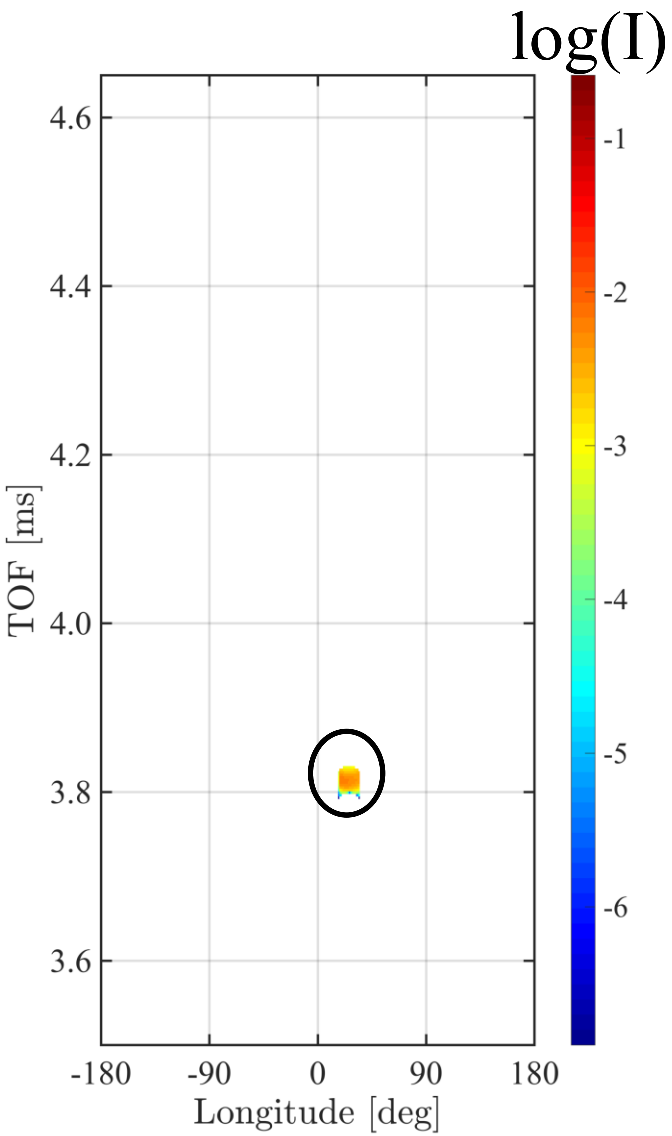


Slide courtesy of Mads Bertelsen, KU

McStas Union conditionals

- Necessary to understand origin of specific parts of background
- Union components contains conditional tools

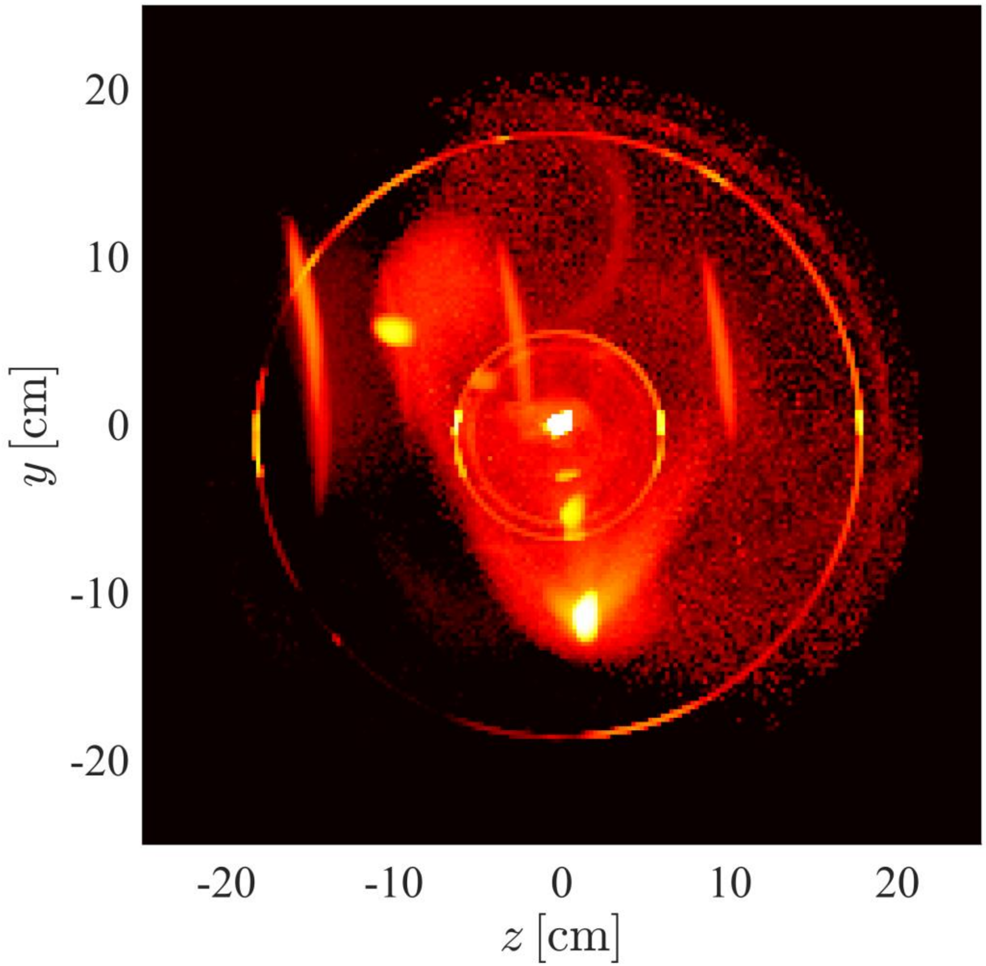




Slide courtesy of Mads Bertelsen, KU

McStas Union conditionals

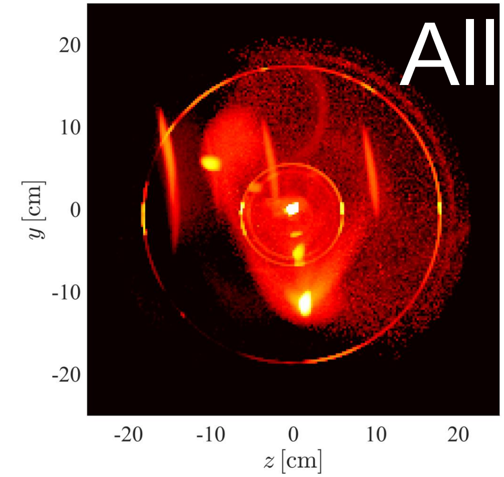
- Conditionals modify loggers so that only rays with correct final state is recorded
- Here scattering contributing to a certain background event is shown

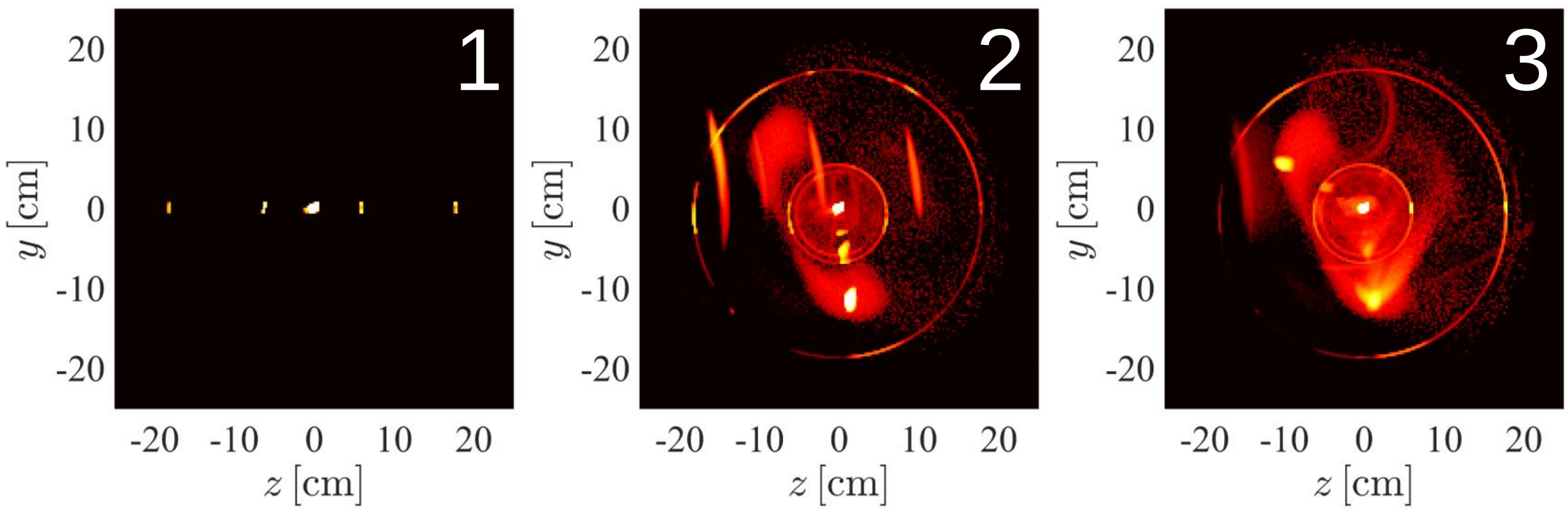


Slide courtesy of Mads Bertelsen, KU

McStas Union conditionals

- Conditionals modify loggers so that only rays with correct final state is recorded
- Here scattering contributing to a certain background event is shown

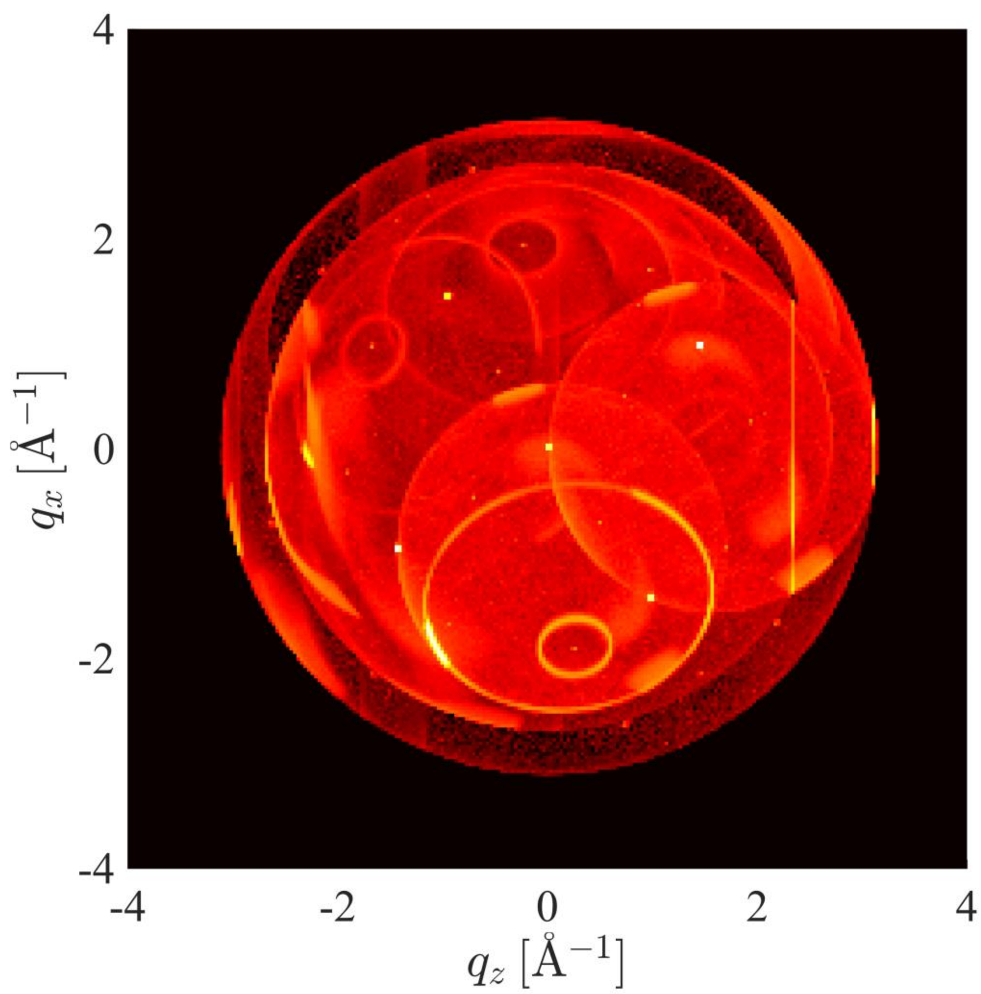




Slide courtesy of Mads Bertelsen, KU

McStas Union conditionals

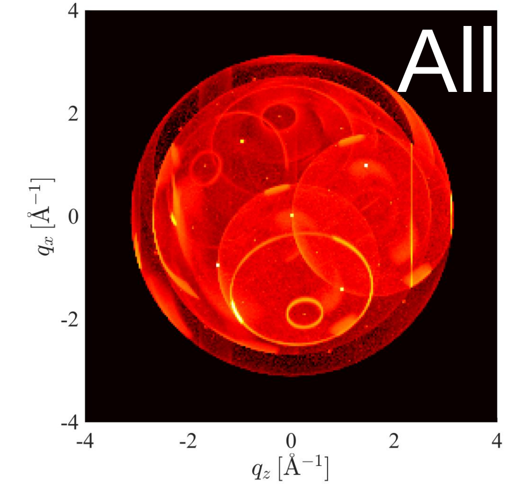
- Conditionals modify loggers so that only rays with correct final state is recorded
- Here scattering contributing to a certain background event is shown

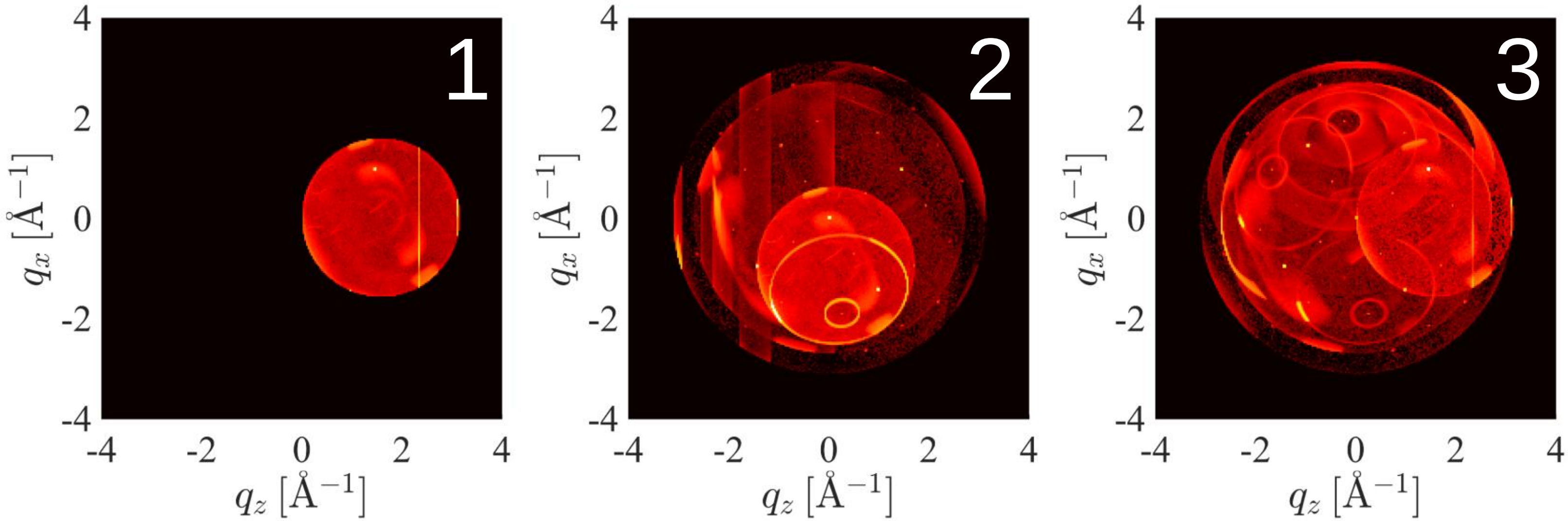


Slide courtesy of Mads Bertelsen, KU

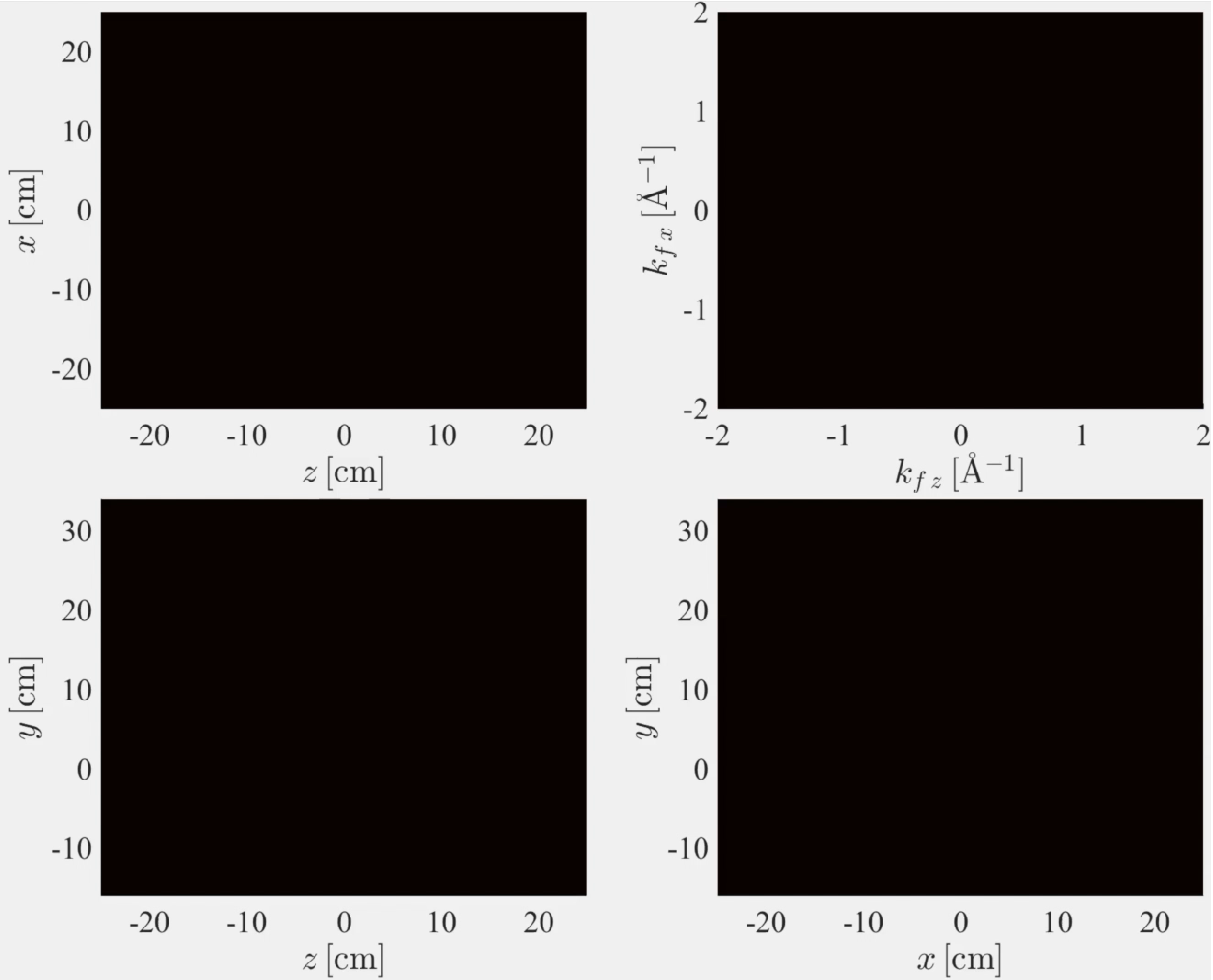
McStas Union conditionals

- Conditionals modify loggers so that only rays with correct final state is recorded
- Here scattering contributing to a certain background event is shown





Slide courtesy of Mads Bertelsen, KU



20
10
0
-10
-20
x [cm]
-20
-10
0
10
20
z [cm]
2
1
0
-1
-2
k_{f x} [Å^{-1}]
-2
-1
0
1
2
k_{f z} [Å^{-1}]
30
20
10
0
-10
y [cm]
-20
-10
0
10
20
z [cm]
30
20
10
0
-10
y [cm]
-20
-10
0
10
20
x [cm]

MACS Instrument simulation

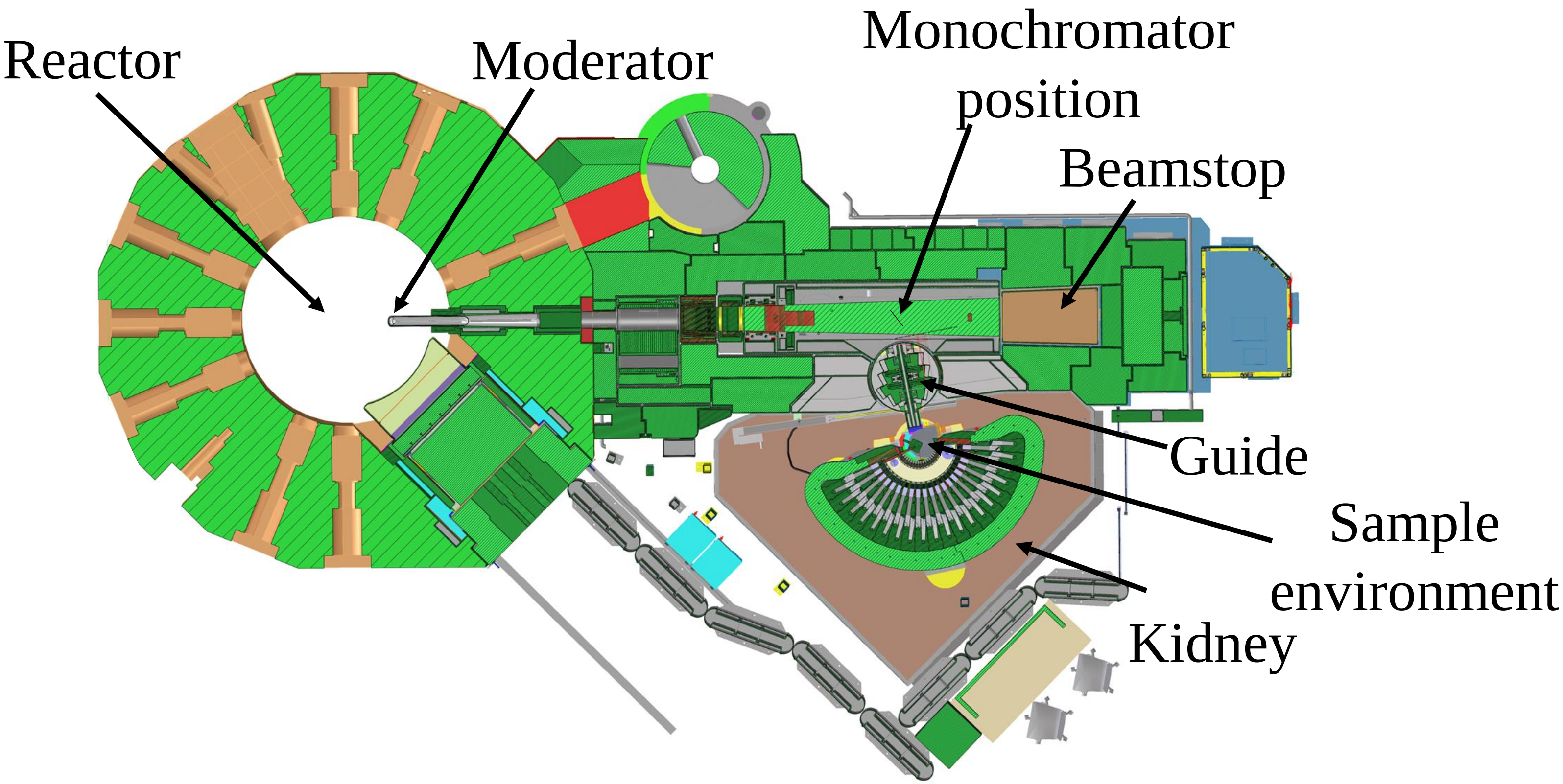
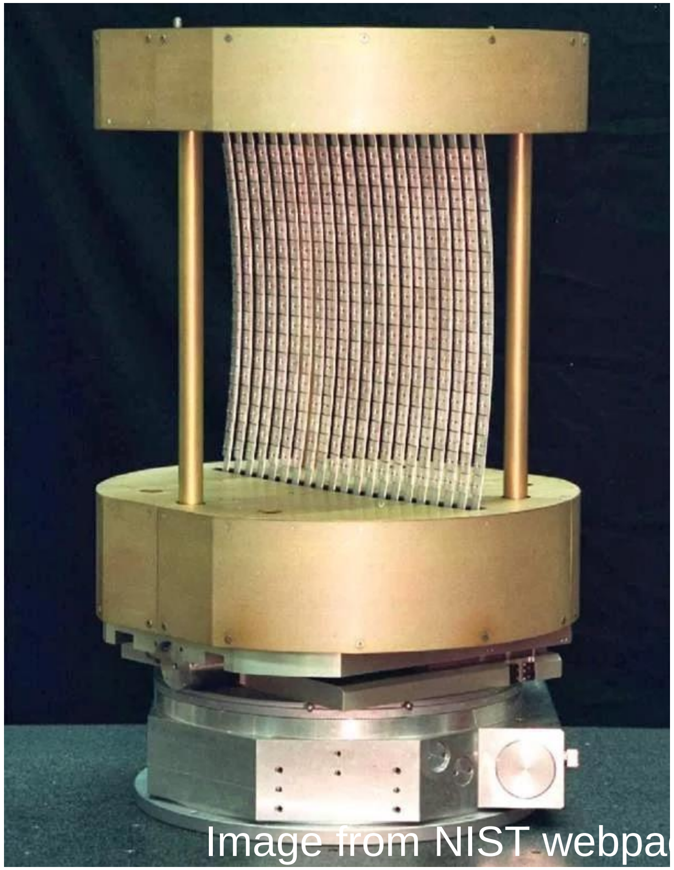
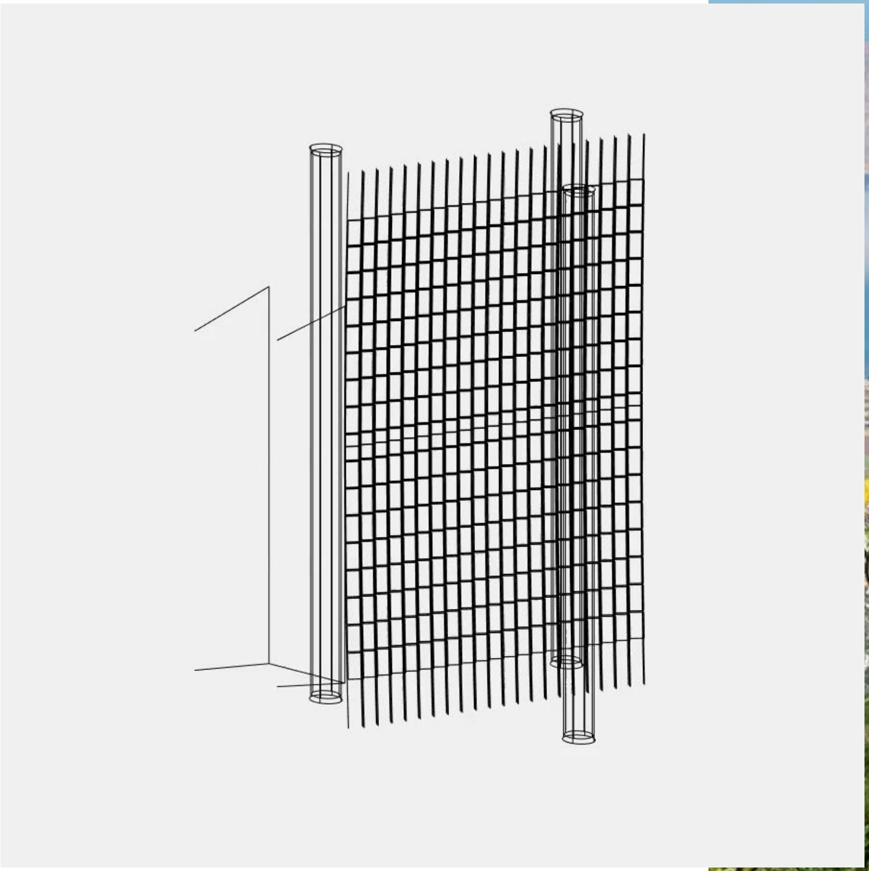


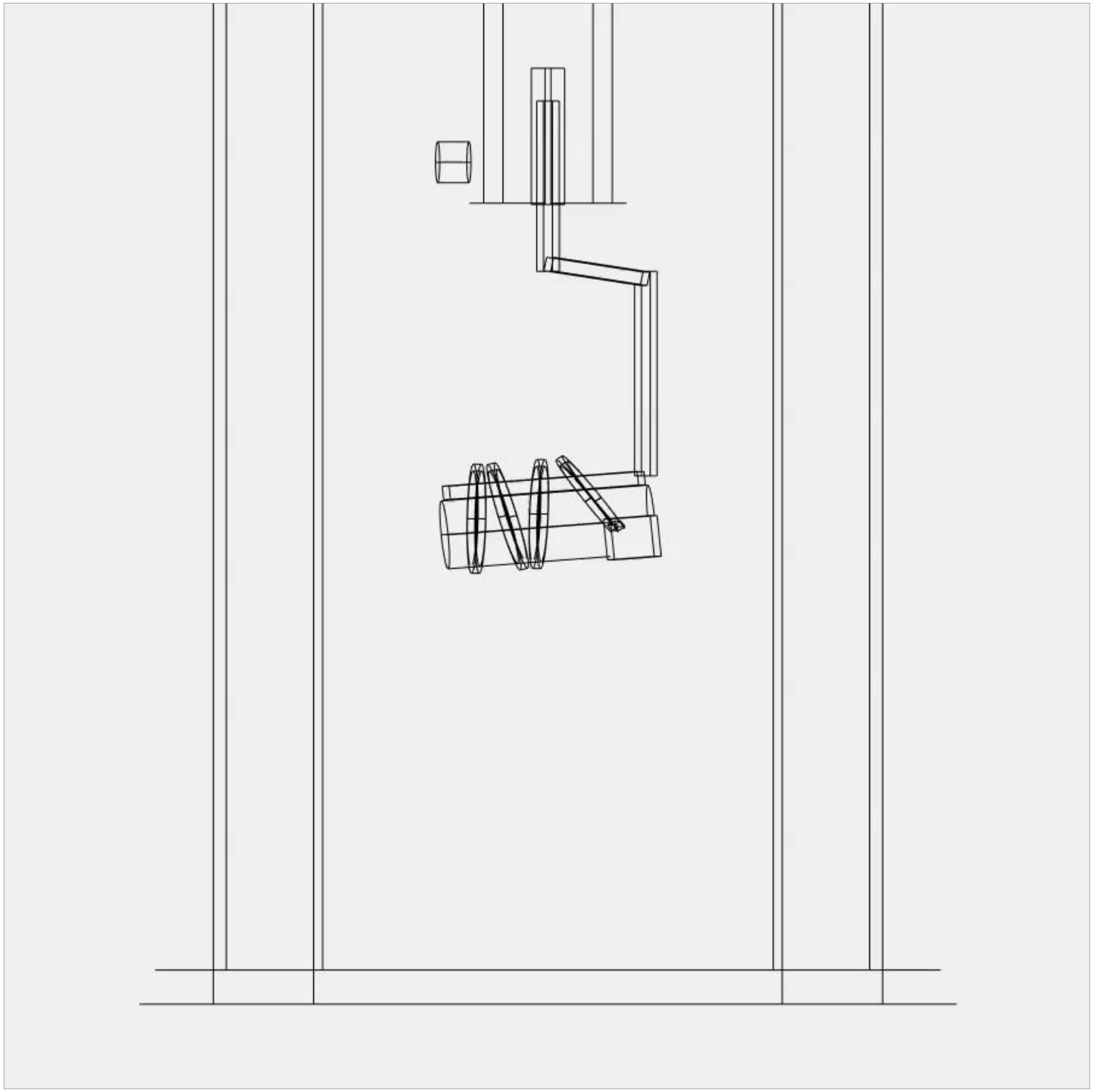
Image from NIST webpage

MACS Instrument simulation





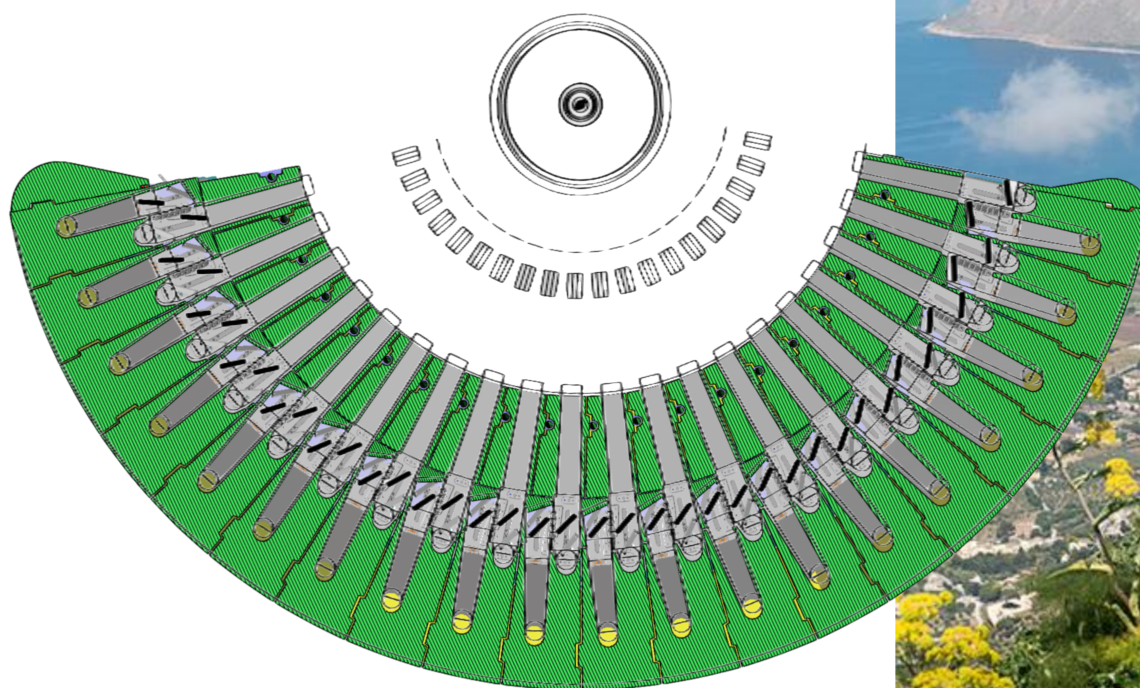
Slide courtesy of Mads Bertelsen, KU

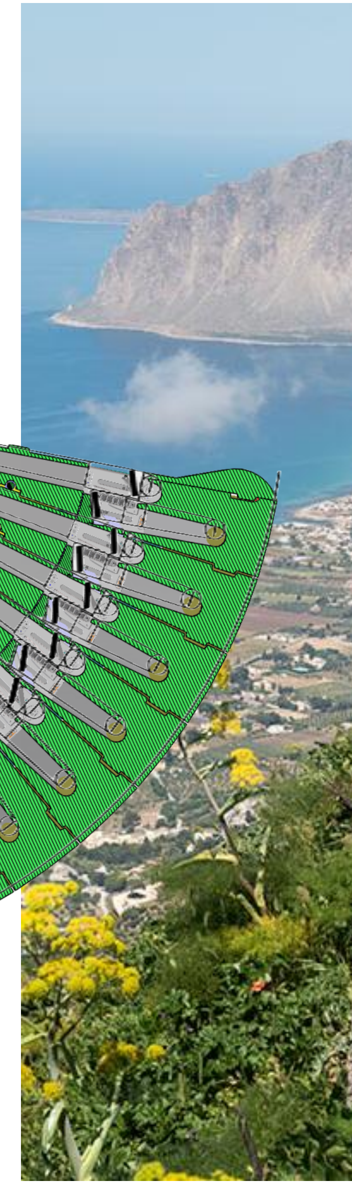


Slide courtesy of Mads Bertelsen, KU

MACS Instrument simulation

- CAD model of instrument backend
- More than 600 geometries
- 2 Union_master components

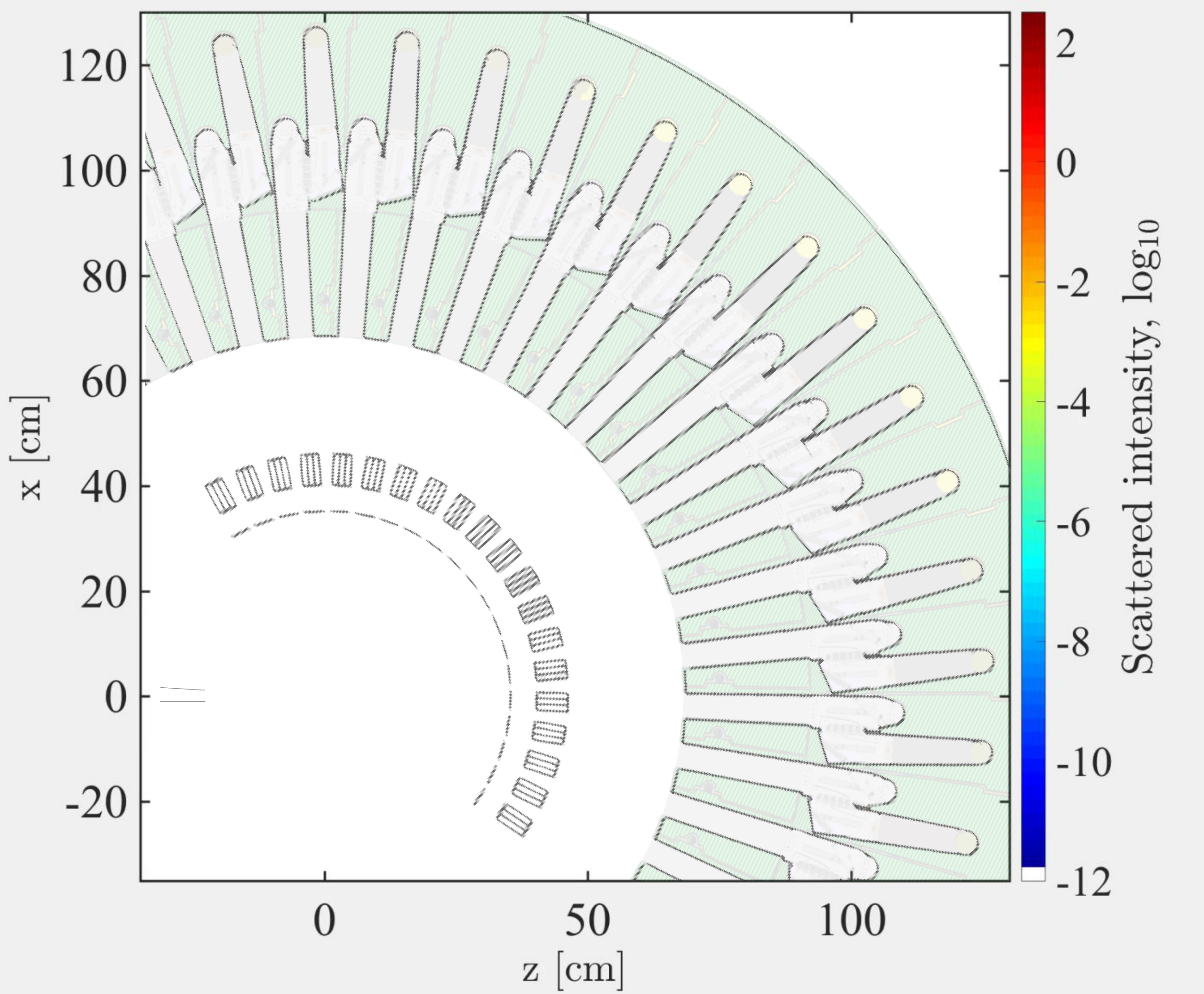




Slide courtesy of Mads Bertelsen, KU

MACS

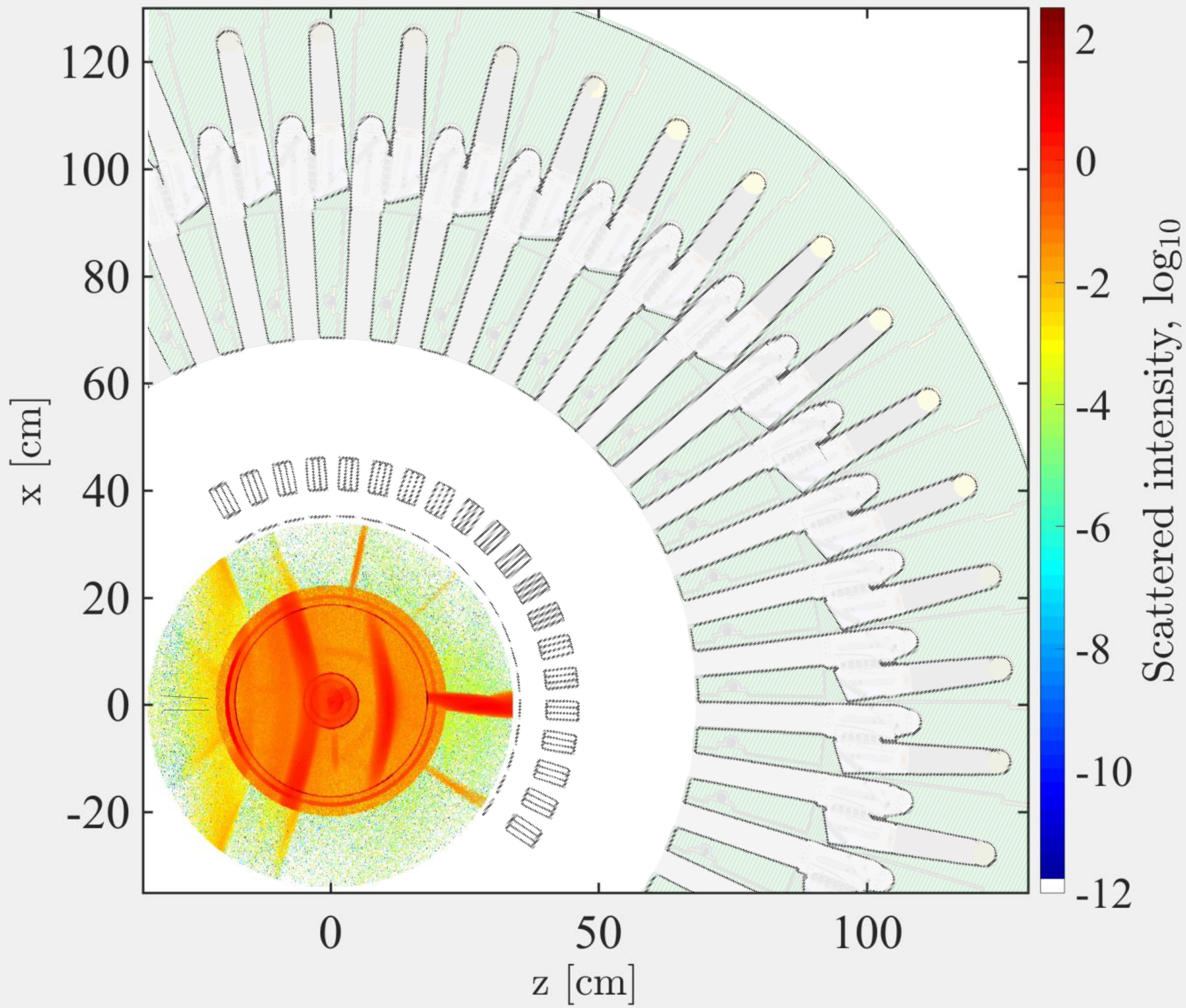
- Air scattering around cryostat
- Initial and final energy: 5 meV



Slide courtesy of Mads Bertelsen, KU

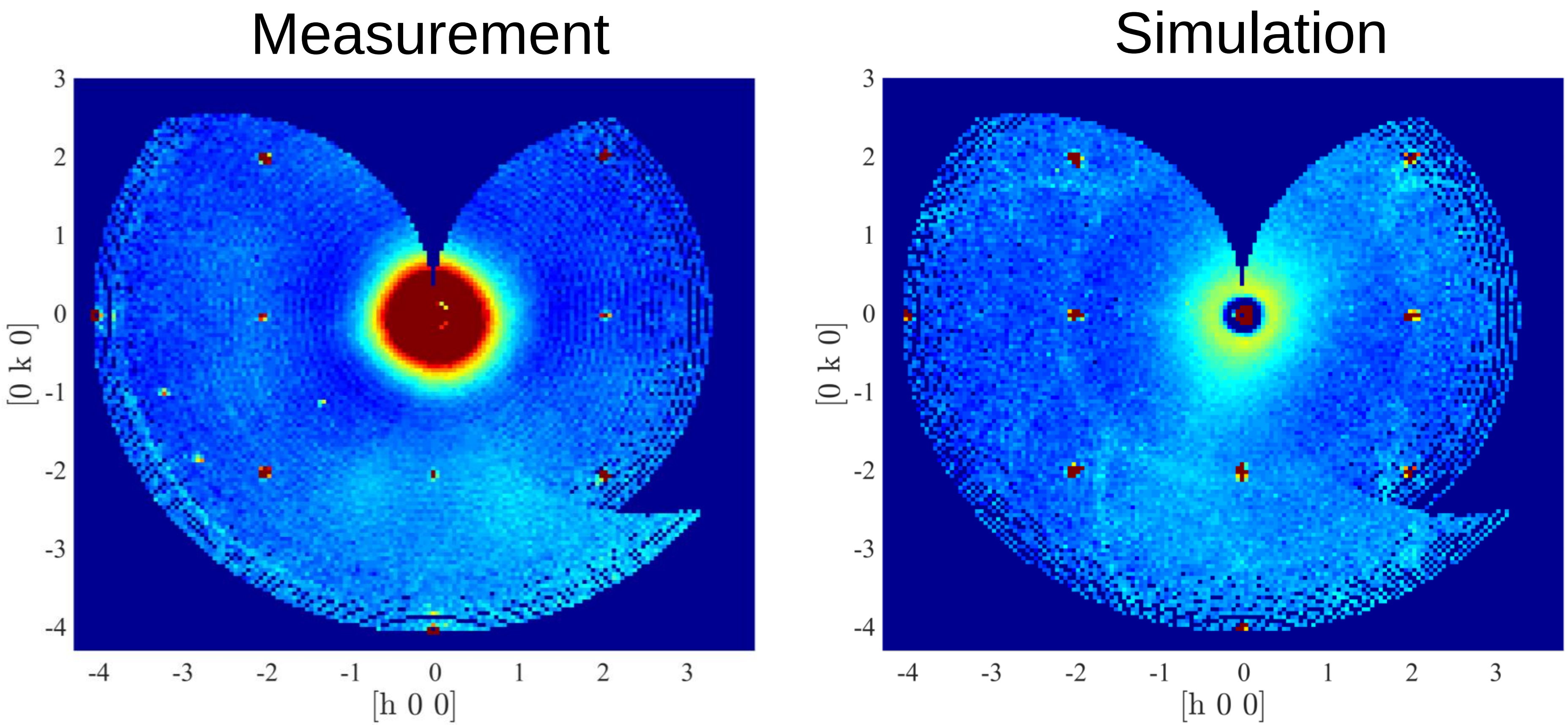
MACS

- Air scattering around cryostat
- Initial and final energy: 5 meV



Slide courtesy of Mads Bertelsen, KU

MACS Results



- *Many similarities*
- *Features missing in simulation (e.g. SANS)*
- *Some structures "exaggerated" in simulation*
- *Phonon-scattering not yet included, may account "smeared" background*

 Slide courtesy of Mads Bertelsen, KU

Important points to remember

1. Your simulation will only contain elements you provided / defined

2. … to the precision you defined

3. Answers the questions you posed

4. Background essentially only from “sample”, or sample-near objects, except if explicitly added (i.e. experimental bg., from MCNP, …)

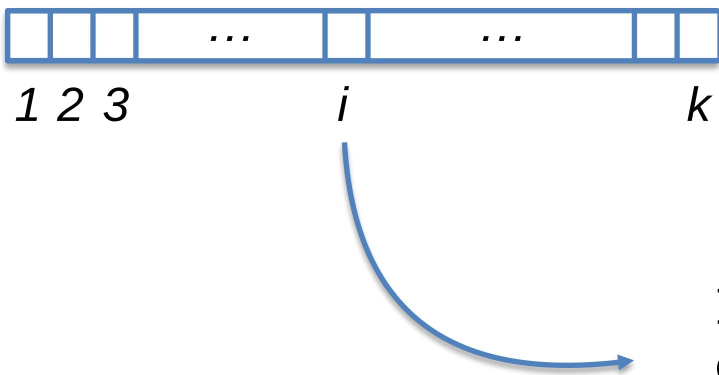
DTU
ess EUROPEAN SPALLATION SOURCE

Simulation to experiment comparison

- *What is really the information content…?*
- *McStas sources generally provide “intensity” in units of neutrons/s (into a chosen solid angle)*
- *That intensity is carried through the instrument on a discrete set of “neutron rays”*

In a histogram sense

- *Imagine a histogram, e.g. I(λ)*



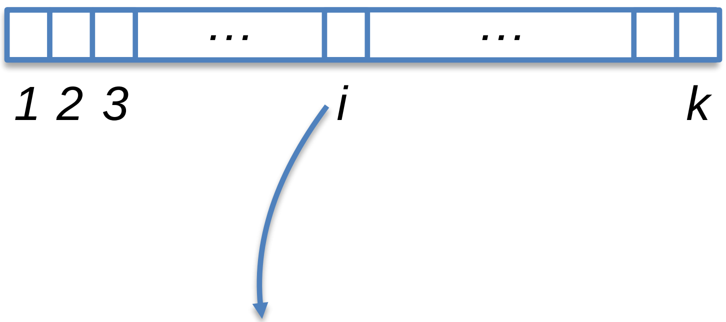
In bin *i*, *N* events each carrying a fractional intensity p_j so that

$$I = \sum_N p_j$$

- *The RMS variance over that set becomes our statistical error bar E*

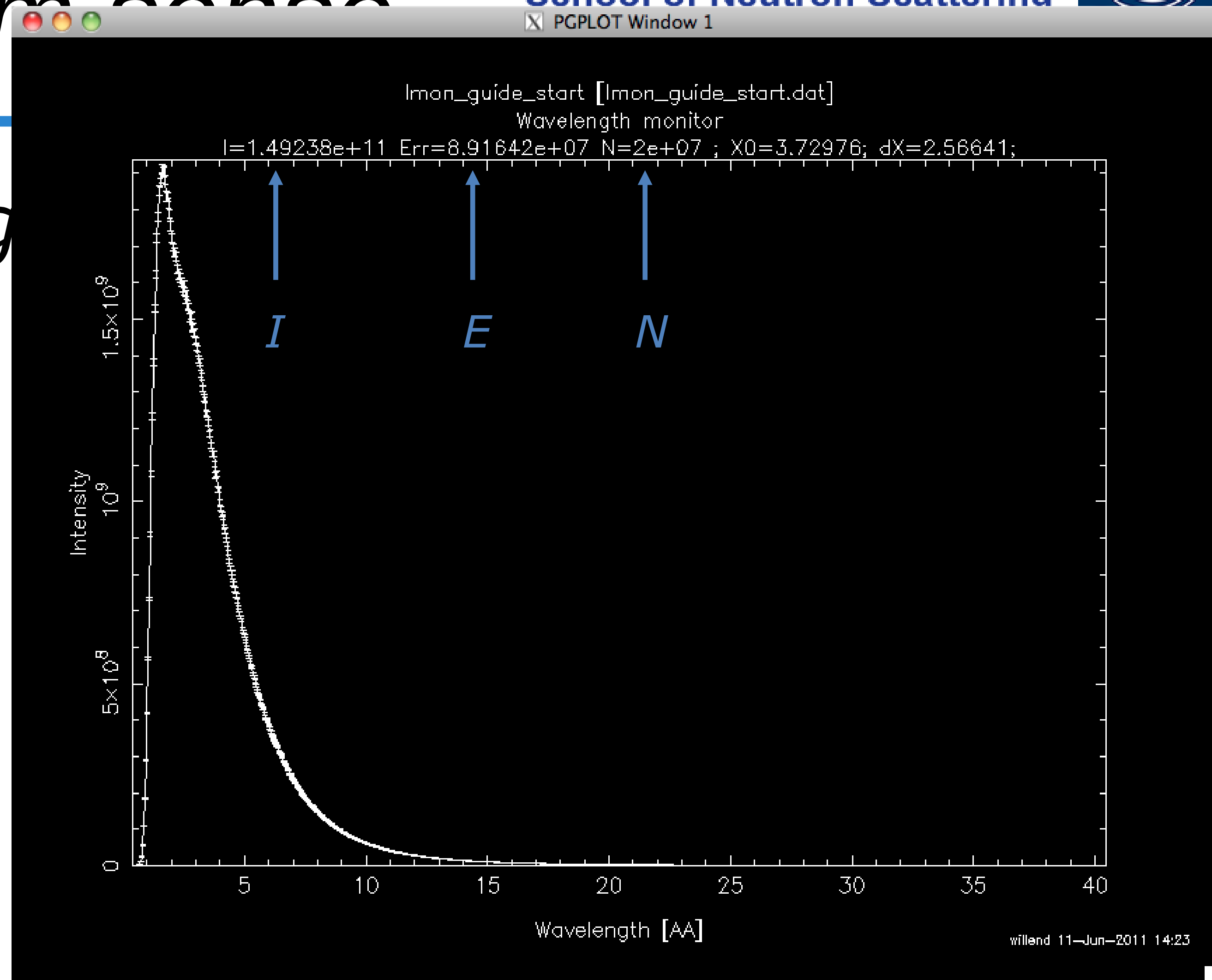
In a histogram sense

- *Imagine a histog*



In bin *i*, *N* events each carrying a fractional intensity p_j so that

$$I = \sum_N p_j$$



- *The RMS variance over that set becomes our statistical error bar E*

From "Virtual experiments - the ultimate aim of neutron ray-tracing simulations", K. Lefmann et al., Journal of Neutron Research 16, 97-111 (2008)

Let n be the number of neutron rays reaching the detector, and let the rays have (different) weights, w_i. The simulated intensity is then given by

$$I = \sum_{i=1}^{n} w_i. \qquad (1)$$

The estimate of the error on this number is calculated in the McStas manual [1], and the standard deviation is approximated by

$$\sigma^2(I) = \sum_{i=1}^{n} w_i^2. \qquad (2)$$

In real experiments, $w_i = 1$, whence we reach $I = n$ and $\sigma(I) = \sqrt{I}$ as expected (for counts exceeding 10). Let the virtual time be denoted by t. The simulated counts during this time becomes

$$C = tI, \qquad (3)$$

- *The RMS variance over that set becomes our statistical error bar E*

From "Virtual experiments - the ultimate aim of neutron ray-tracing simulations", K. Lefmann et al., Journal of Neutron Research 16, 97-111 (2008)

and its error bar estimate is

$$\sigma^2(C) = t^2\sigma^2(I). \tag{4}$$

However, to simulate a realistic counting statistics, we must fulfill

$$\sigma_{\mathrm{VE}}(C_{\mathrm{VE}}) = \sqrt{C_{\mathrm{VE}}}. \tag{5}$$

This is obtained by adding to (3) a Gaussian noise $E(\Sigma)$ of mean value zero and standard deviation Σ:

$$C_{\mathrm{VE}} = tI + E(\Sigma). \tag{6}$$

The standard deviation for the VE becomes

$$\sigma^2_{\mathrm{VE}}(C) = t^2\sigma^2(I) + \Sigma^2. \tag{7}$$

Now, the requirement (5) allows us to determine Σ:

$$\Sigma^2 = tI - t^2\sigma^2(I). \tag{8}$$

Since Σ^2 must remain positive, we reach an upper limit on t

$$t_{\max} = \frac{I}{\sigma^2(I)}. \tag{9}$$

Sketch of an algorithm…

1. *On a given McStas histogram*

2. *For the non-zero bins, calculate*

$$t_{\max} = \frac{I}{\sigma^2(I)}\,.$$

3. *The smallest* $t_{\mathtt{max}}$ *defines the "maximal counting time" allowed by your statistics*

4. *Preferably a "background" should be added - use a "known experimental value" or an estimate, be it back of envelope or from e.g. MCNP…*

Still interested?

Overview of web resources for McStas

Get the code, report bugs etc.

- McStas website
- McStas mailinglist subscription (Please enrol!)
- McStas Facebook page (Please follow us!)
- McStas downloads
- McStas+McXtrace GitHub
- McStas+McXtrace issues + bug reporting

Neutron scattering + McStas e-learning

- e-neutrons website (free enrolment)

Tutorials, howto's, docs

- How McStas works - in 2 minutes
- Tutorial: Build a SANS
- Tutorial: Build a diffractometer (outdated in certain parts)
- "Virtual experiments in a nutshell" (JDN 2010)
- McStas user manual - **Better use mcdoc -m in the terminal!**
- McStas component manual - **Better use mcdoc -c in the terminal!**
- McStas component docs - **Better use mcdoc in the terminal!**
- McStas sample model functionality matrix (not fully up to date)
- McStas and McXtrace GitHub wiki - tutorials, guides and more

ESS specific McStas docs

- McStas space on ESS confluence
- Running McStas on the ESS DMSC cluster - plus general DMSC cluster info
- MCPL input to describe the ESS source
- 'Benchmarking' website for the ESS_butterfly component

Material from schools

- Material from ESS DMSC McStas workshop June 2018
- Website and materials from a 3-day STFC event (including public Dropbox link to presentations etc.)
- Website and materials from a 3-day ESS event (including public Dropbox link to presentations etc.)
- Website and materials from a 2-day NOBUGS event (including public Dropbox link to presentations etc.)
- Dropbox with materials from a 5-day event in Bariloche, Argentina

Still interested?

Overview of web resources for McS

Get the code, report bugs etc.

- McStas website
- McStas mailinglist subscription (Please enrol!)
- McStas Facebook page (Please follow us!)
- McStas downloads
- McStas+McXtrace GitHub
- McStas+McXtrace issues + bug reporting

Neutron scattering + McStas e-learning

- e-neutrons website (free enrolment)

Tutorials, howto's, docs

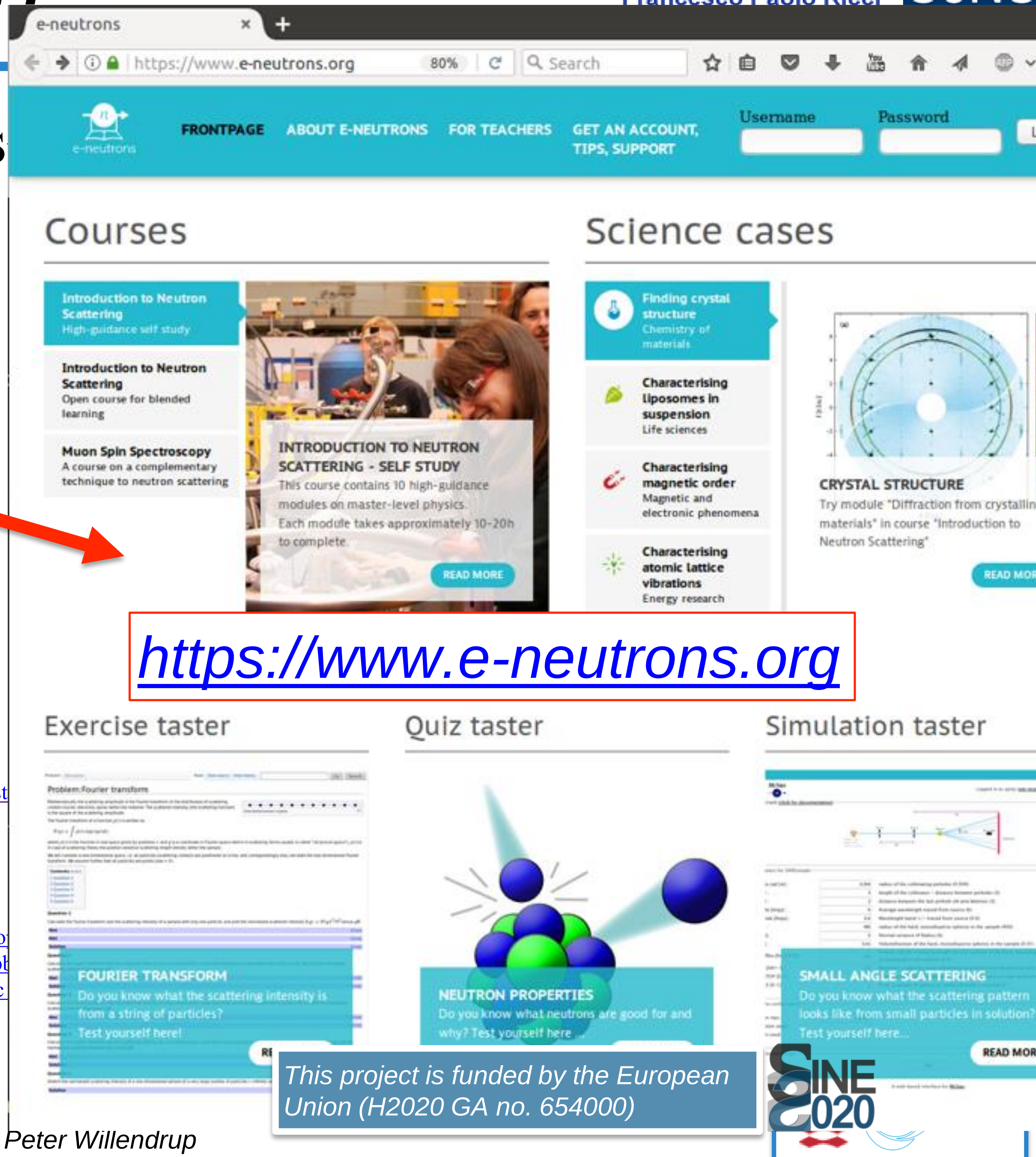
- How McStas works - in 2 minutes
- Tutorial: Build a SANS
- Tutorial: Build a diffractometer (outdated in certain parts)
- "Virtual experiments in a nutshell" (JDN 2010)
- McStas user manual - **Better use mcdoc -m in the terminal!**
- McStas component manual - **Better use mcdoc -c in the terminal!**
- McStas component docs - **Better use mcdoc in the terminal!**
- McStas sample model functionality matrix (not fully up to date)
- McStas and McXtrace GitHub wiki - tutorials, guides and more

ESS specific McStas docs

- McStas space on ESS confluence
- Running McStas on the ESS DMSC cluster - plus general DMSC clust
- MCPL input to describe the ESS source
- 'Benchmarking' website for the ESS_butterfly component

Material from schools

- Material from ESS DMSC McStas workshop June 2018
- Website and materials from a 3-day STFC event (including public Dro
- Website and materials from a 3-day ESS event (including public Dropb
- Website and materials from a 2-day NOBUGS event (including public
- Dropbox with materials from a 5-day event in Bariloche, Argentina



Thanks!!

- *Questions?*

McStas

This project is funded by the European Union (H2020 GA no. 654000)

This project is funded by the European Union (H2020 GA no. 654000)

MCPL Monte Carlo Particle Lists