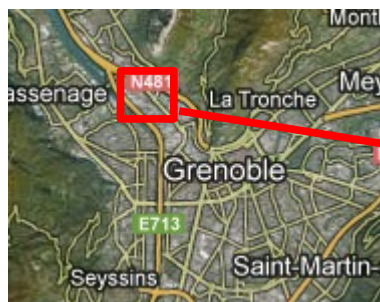


An introduction to neutron scattering with McStas

E. Farhi, ILL/Computing for Science



Staff: 400
Exp./y: 1100+
~30 instruments

~~~ *The Menu* ~~~

- What is **Monte-Carlo** ? (not only a casino)
- Instrument simulation **codes**
- Instrument **components**
- Examples of **neutron instruments**
- **Virtual experiments**
- What we can do with TRIGA@PUSPATI
- Exercises

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

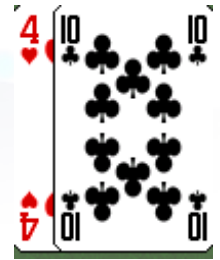
Neutron Scattering and Absorption in *U* and *Pu*, Origin of *MCNP*

Name:

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



What are Monte Carlo methods ?



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

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

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

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

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

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

How to implement Monte Carlo methods ?

Good random generator:

- from thermal electronic noise (hardware)
- or quasi-random generators => *quasi-Monte-Carlo*

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

Crude Monte-Carlo (yes/no choice):

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

We keep events that satisfy $\xi < p$

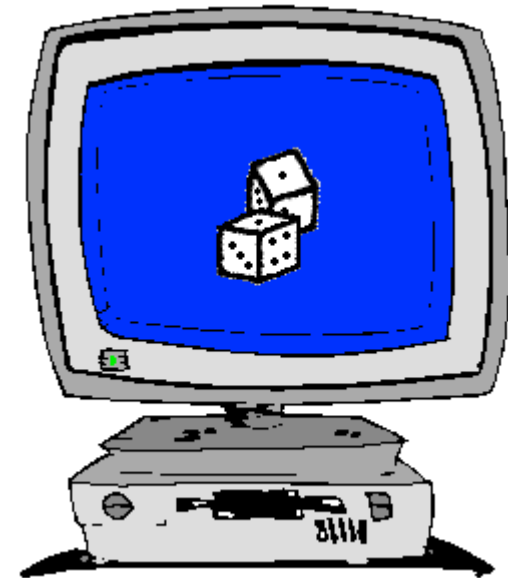
np events \rightarrow low statistics

Importance sampling (fuzzy choice – event weighting):

Keep n events, no more random number...

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

Retain statistical accuracy ($1/\sqrt{n}$)



When to use Monte Carlo methods

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

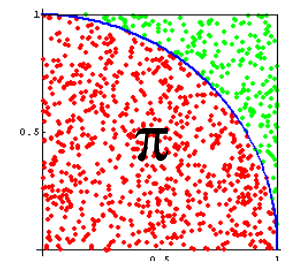
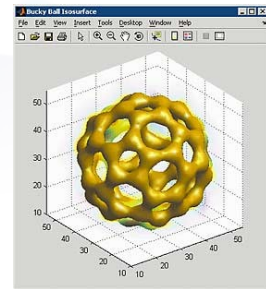
Overall complexity is beyond reasonable analytical methods

Each event can be computed easily and independently

MC is the '*lazy guy*' method – think microscopic

Examples:

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



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

Some Monte Carlo programs

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

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

Nuclear reactor simulations (neutron transport):

MCNP, Tripoli, GEANT4, FLUKA

Neutron Ray-Tracing propagation:

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

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

Propagation simply uses Newton rules, incl. gravitation.

X-ray tracing


Shadow, McXtrace

Presentation of McStas

McStas: History, funding, team



Flexible, general simulation utility for neutron scattering experiments.

Original design for Monte Carlo Simulation of triple axis spectrometers. 
Developed at RISØ and ILL (started 1998) by *K. Lefmann* and *K. Nielsen*.
Mainly funded by EU (FP 4-7): *P. Willendrup*, *E. Farhi*, *E. Knudsen/2* + others
Open source, GPL2. Works on all systems (*Linux*, *Win\$*, *MacOSX*, *iPhone...*)

Most widely used code in its ecological niche (compared to other similar codes)..

A few hundred users worldwide (230 registered)
some contributors (community based).
*Has been used to model most existing and future
neutron scattering instruments.*

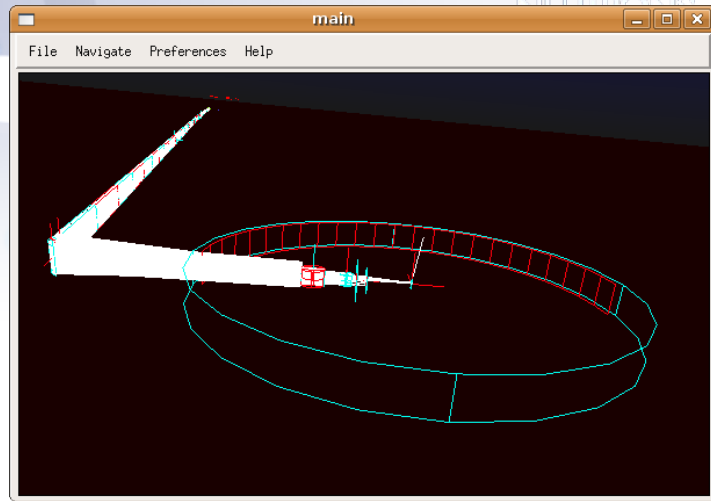
<www.mcstas.org>

mcstas-users@mcstas.org mailing list

mcstas-support@mcstas.org developer contact

Main capabilities of McStas

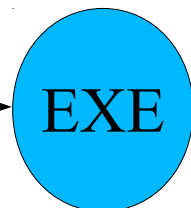
McStas requires an **instrument description** (text)
 Many instrument examples are given with the software, covering all instrument classes (TOF, TAS, SANS, DIF, ...).



```

Edit: PSI_DMC.instr
File Edit Search View Insert
DEFINE INSTRUMENT PSI_DMC(lambda = 2.5666, R = 0.87, R_curve = 0.87,
TRACE
COMPONENT source_arm = Progress_bar()
AT (0, 0, 0) ABSOLUTE
COMPONENT source = Source_Maxwell_3(
  yheight=0.156, xwidth=0.126,
  Lmin=lambda-ldiff/2, Lmax=lambda+ldiff/2,
  dist=1.5, focus_xw = 0.02, focus_yh = 0.12,
  T1=296.16, I1=8.5E11,
  T2=40.68, I2=5.2E11)
AT (0,0,0) RELATIVE source_arm ROTATED (0,0,0) RELATIVE source_arm
COMPONENT PSDbefore_guides = PSD_monitor(
  nx = 128, ny = 128, filename = "PSDbefore_guides",
  xwidth = 0.02, yheight = 0.12)
AT (0, 0, 1.49999) RELATIVE source_arm
  
```

Parameters

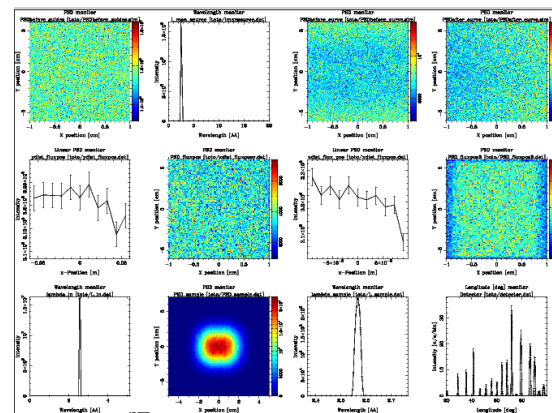


Results

3D view

'Monitors'

McStas creates an instrument executable model for which configuration parameters can be changed at will.



The McStas guts (it works but you don't feel it)

$n=(r,v,t,s)$

Components

n'

Instrument

Kernel

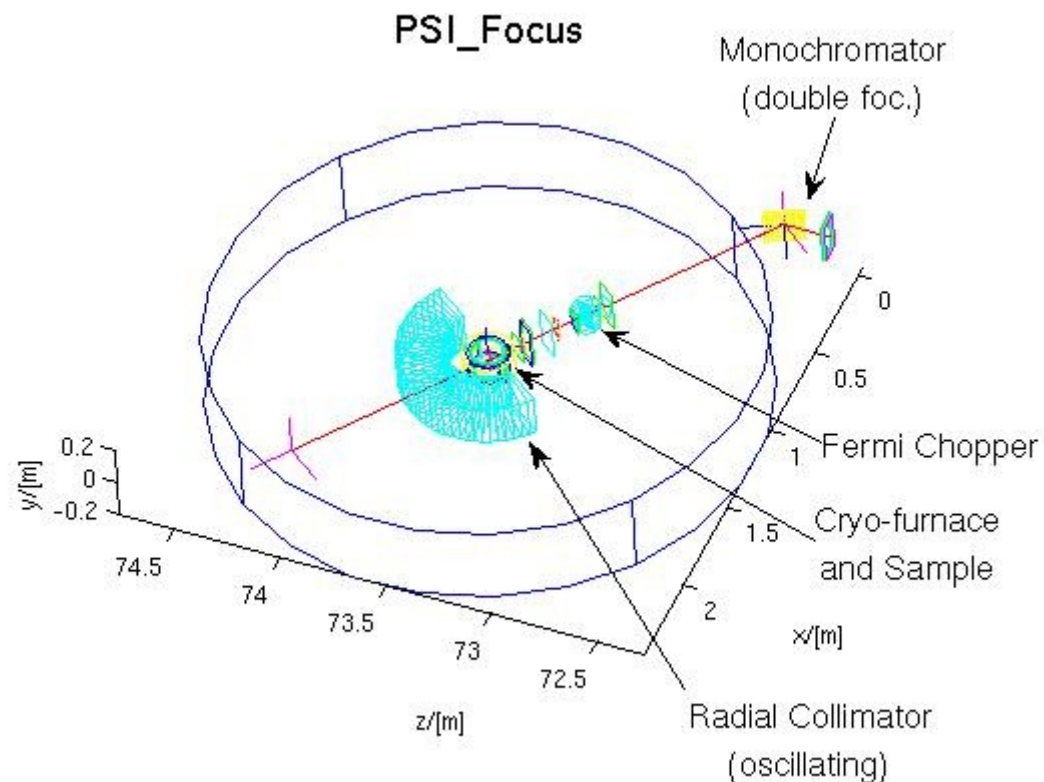
- Propagation between components
- Positioning, geometry, random numbers...
- Code generation (in C) from meta language

Component level

- Physics
- Interactions neutron \leftrightarrow Matter

Instrument

- Relative positions of components
- Interaction with you
- Can be executed as single shot, scans, geometry view and optimize mode



What you get when using McStas

Main instrument components

- *Sources* (area, Maxwellian, events from other MC codes, pulsed)
- *Optics* (slits, monochromators, moving optics, guides...)
- *Samples* (powders, colloids, Xtals, liquids and amorphous). Most of the physics is here (structure, dynamics)
- *Detectors/Monitors* (write data files in many formats).

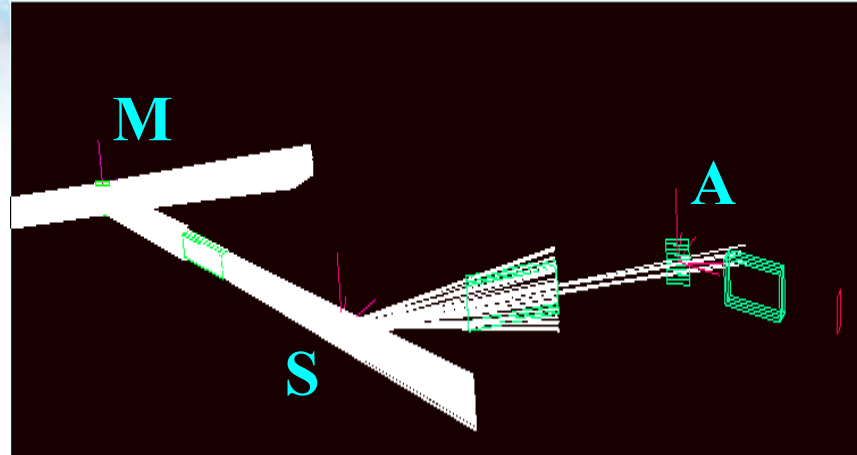
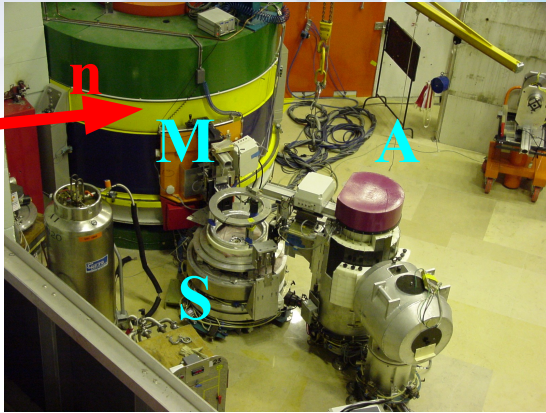
Main instrument examples:

- We have example instruments for all type of experiments
- Virtual experiments

Tools:

- 3D view of instruments and particle trajectories
- Instrument editor (text file)
- Data plotter
- Cluster support (MPI, grid) up to 1000 cpu
- Optimizers (embedded and with <Fit.mccode.org>)

Simulating instrument: what for ?



A Triple-Axis Spectrometer (IN20)

- **Design** new instruments
- **Optimize existing** instruments (flux/resolution)
- **Optimize usage** of existing instruments (better experiments)
- **Get accustomed** with the instruments before coming at the ILL
- Measure the **instrument effects** on the simulated 'ideal' sample signal
- Estimate complex effects like :
 - absorption, multiple scattering, geometry, resolution function...
- **Compare** virtual experiments with real ones, possibly during exp.

What Mcstas does

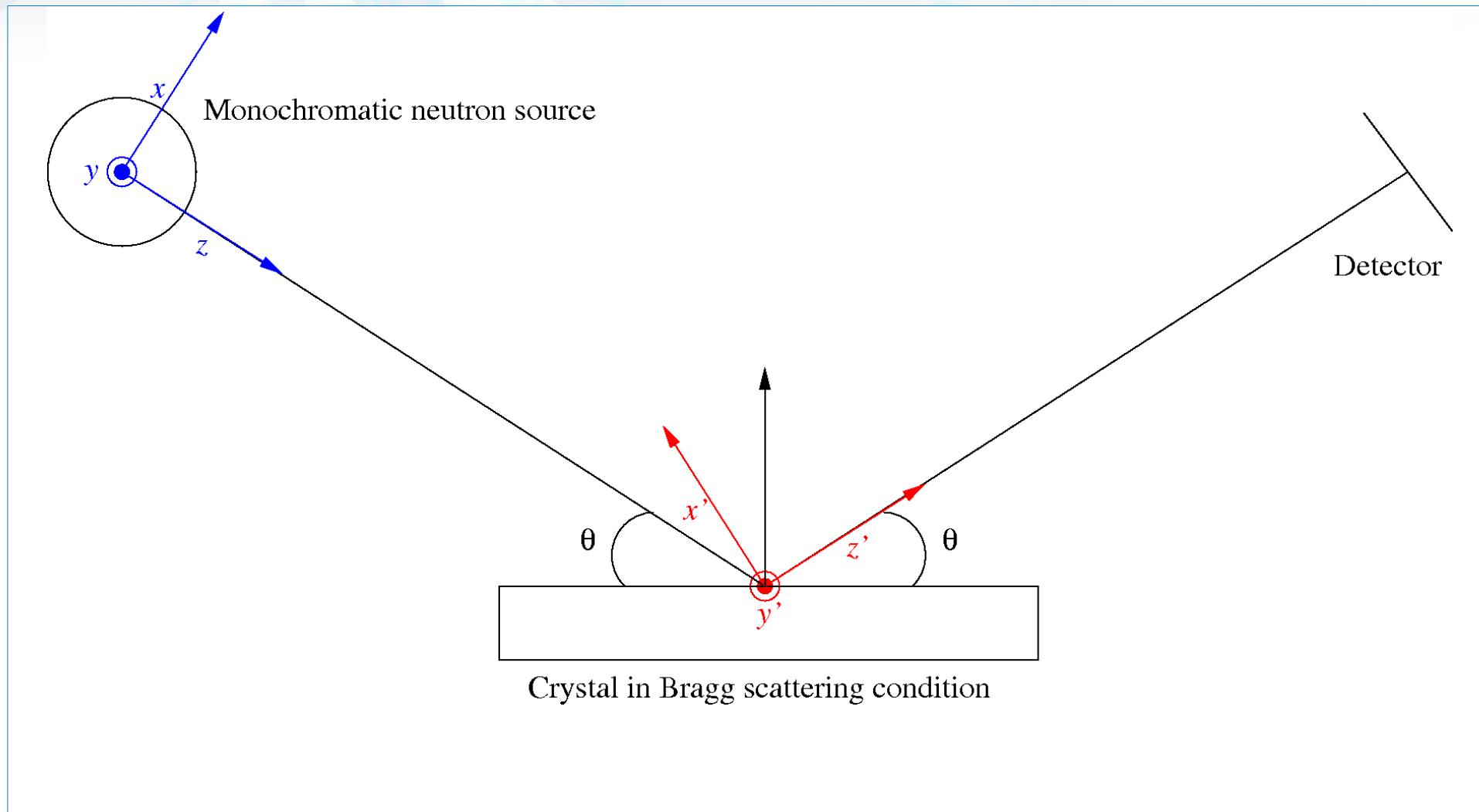
- Propagates neutrons with Newton laws, and handle low energy scattering ($E < 100$ meV)
- Suitable for neutron scattering studies, e.g. SANS, diffraction, imaging
- Can model background from scattering on surrounding materials
- Can use neutron from MCNP with PTRAC card

What McStas does NOT

- Can not handle high energy moderation
- Not appropriate to estimate fast neutron background and γ
- Not appropriate for in-pile design (shielding, beam tubes)

Schematic neutron propagation

n



Schematic neutron propagation **n**

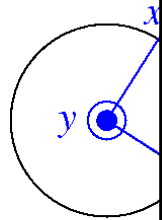
Neutron ray/package:

Weight (p): # neutrons (left) in the package

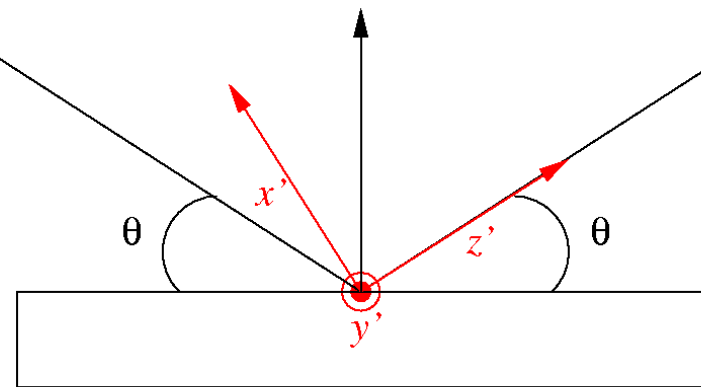
Coordinates (x, y, z)

Velocity (v_x, v_y, v_z)

Spin (s_x, s_y, s_z) , Time (t)



ector

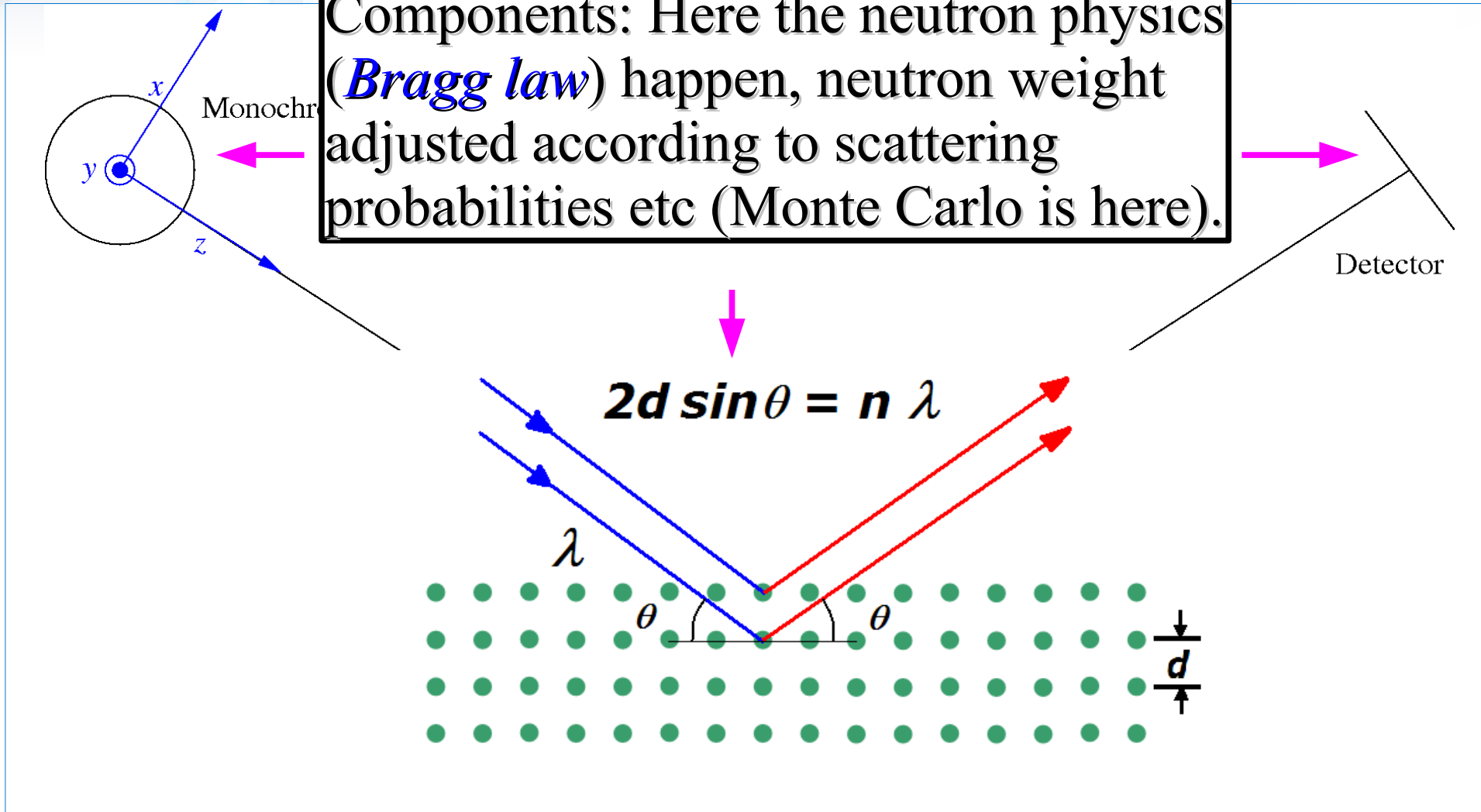


Crystal in Bragg scattering condition

Schematic neutron propagation

n

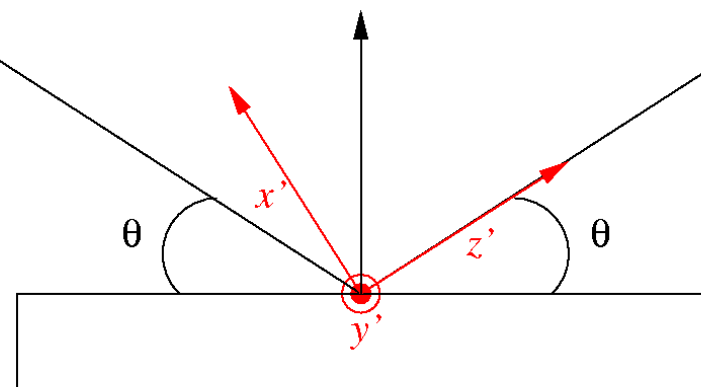
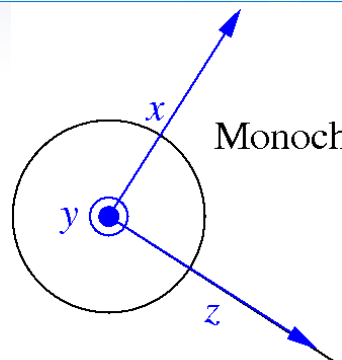
Components: Here the neutron physics (*Bragg law*) happen, neutron weight adjusted according to scattering probabilities etc (Monte Carlo is here).



Schematic neutron propagation

n

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



Crystal in Bragg scattering condition

Detector

McStas grammar

Instruments are defined from a text file. General syntax is based upon C.
Basic syntax:

```
DEFINE INSTRUMENT name(parameter1=value1, ...)
```

Can be changed
from cli and GUI
(e.g. scans)

```
DECLARE % { /* global variables */  
% }
```

```
INITIALIZE % { /* code executed when starting */  
% }
```

```
TRACE /* sequence executed for each neutron event */  
COMPONENT InstanceName=component(parameters...)
```

```
AT (position) RELATIVE reference
```

```
ROTATED (orientation) RELATIVE reference
```

```
...
```

```
END
```

McStas components

Sources (neutrons)

Continuous (reactor), Pulsed

Moving optics

Choppers (disk and Fermi), velocity selectors, phase space transformers

Static optics

Guides, mirrors, benders, lenses, collimators, slits, filters, monochromators

Samples/materials

Single crystals, liquids, gas, polymers, hard spheres

Detectors

Histogram and event monitors, gas detectors

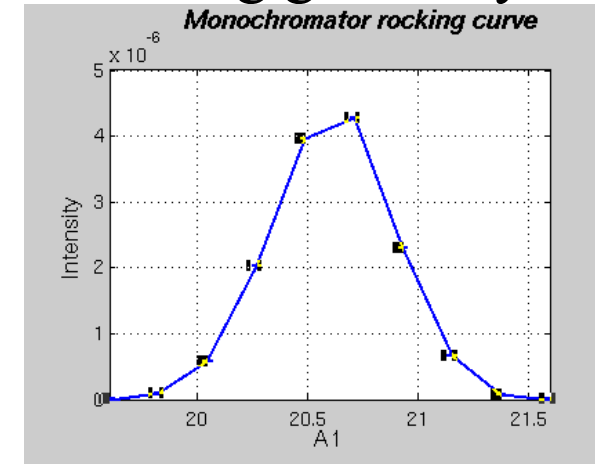
Total 190+ components, 90+ examples, manuals, tutorials...

Neutron optics: monochromators (primary spect.)



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

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



Cu, Ge, Si, C (graphite)
CuMnAl, CoFe (pol.)

Design and characterisation are eased using simulation,

- intensity vs. curvature radius,
- thickness of blades
- rocking curve (reflectivity)
- scattering from the mechanics (generated background)
- choice of crystals

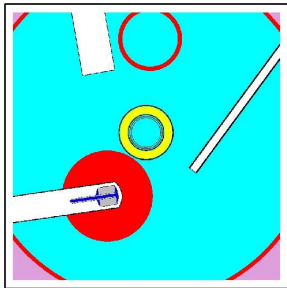
Source: Generating neutron events

Neutron events may be currently generated from:

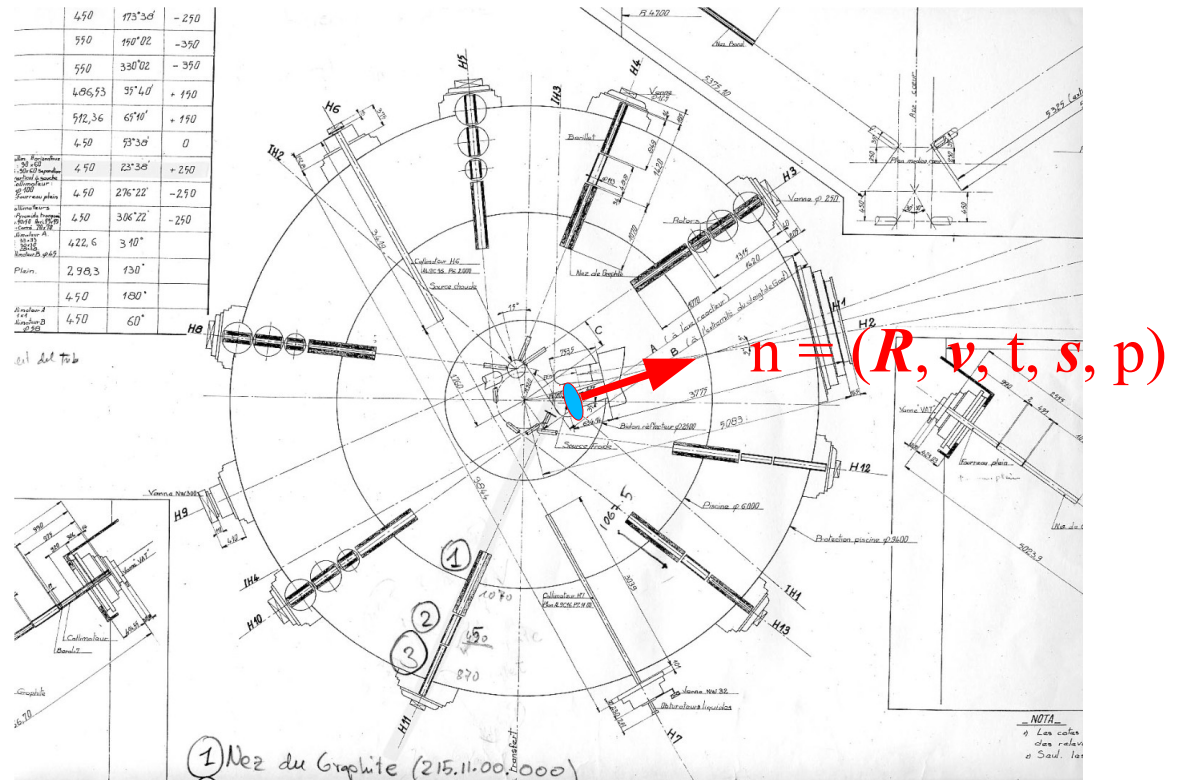
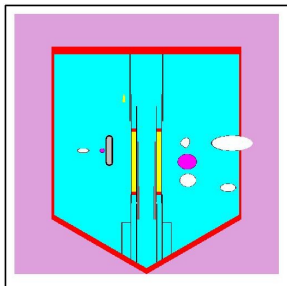
- A black body (Maxwellian distributions)
- Measured flux/divergence distributions
- MCNP and Tripoli (nuclear reactor simulation codes) event files
- Pulsed sources (SNS, ESS, ISIS, J-PARC...)

Other neutron generators are in principle easy to write (e.g. from GEANT, FLUKA, ...)

top



side



Virtual experiments: definition

Virtual experiments

=

Instrument simulation

⊗

Realistic Sample model

=

Neutron *flight simulator*

and then we play with it without breaking the instrument and sample.

- E. Farhi *et al*, *J. of Comp. Phys.* **228** (2009) 5251
- E. Farhi and P. Willendrup, *Collection SFN* **12** (2011) 303
- E. Farhi *et al.*, *J. Neut. Res.*, **18** (2013)

Benefits from a virtual experiment

Benefits:

- The resolution from the instrument is **implicitly convolved** with the sample, without analytical integration
- The 'spurious' effects from the instrument can be included
- Treat absorption in the sample (self attenuation= 2^{nd} extinction)
- Treat coherent/incoherent scattering
- May include multiple scattering (depends on the sample component)
- May model the measurement background

Requirements:

- Need as accurate instrument description as possible
- Use proper sample component/kernel
- Give accurate data to the sample component (experiment, MD, ...)
- Be critical regarding the results (*it's so easy to get rubbish*)

McStas: available sample components

Most instrument components are 'easy' to implement.
 The samples are often more complex.

<i>Scattering</i>	<i>Input</i>	<i>Coherent</i>		<i>Incoherent</i>		<i>Multiple</i>
		<i>Elastic</i>	<i>Inelastic</i>	<i>Elastic</i>	<i>Inelastic</i>	
PowderN	$ F^2(q) , abc, \sigma$	Powder		X		
Single_crystal	$ F^2(hkl) , abc, \sigma$	Crystal		X		X
Isotropic_Sqw	$S(q,w), \sigma$	Isotropic	X	X	X	X
Incoherent	σ			X	QuasiE	
Sans_sphere	ρ, σ, R	colloid				
Sample_nxs	CIF data	Powder		X		X

Sample geometry: box, cylinder, sphere, PLY/OFF file

What can be done at TRIGA PUSPATI

Neutron diffraction

- Bragg law: $2d \sin \theta = n\lambda$ coherent scattering
- We send a single λ and get diffraction rings at angles θ for each d spacing.
- In a cubic crystal $d = a / \sqrt{h^2 + k^2 + l^2}$
- $\theta = a \sin(n\lambda \sqrt{h^2 + k^2 + l^2} / 2a)$
- If $\lambda > 2a$, no Bragg diffraction can occur
- Cold beam (e.g. [SANS@PUSPATI](#)) no diffraction
→ Bragg edge

Diffraction example

templateSANS

$\lambda=6 \pm 0.5 \text{ \AA}$

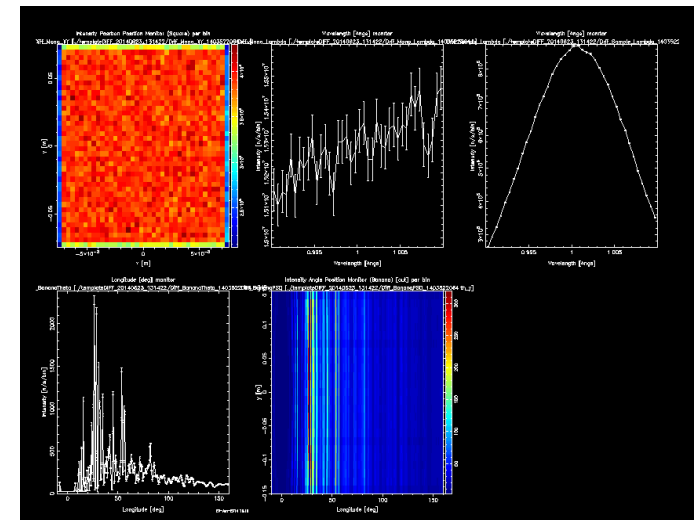
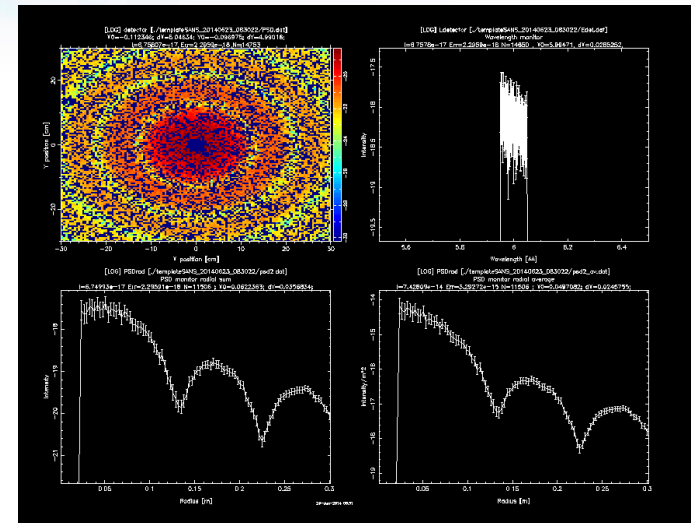
Latex colloid solution $\rho=100 \text{ \AA}$

Detector 128×128 (30 cm)

Similar with $\lambda=2$ (thermal), $\rho=50$

TemplateDIFF

$\lambda=6 \text{ \AA}$, PG(002) monochromator

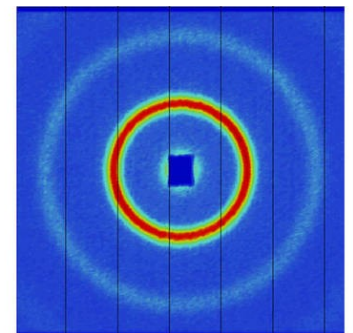


What to use neutrons for ?

- I will describe a set of simple instruments that can make use of neutrons for studying of materials
- Only use basic 'optics' components
- In principle, we can install such instruments on a TRIGA reactor

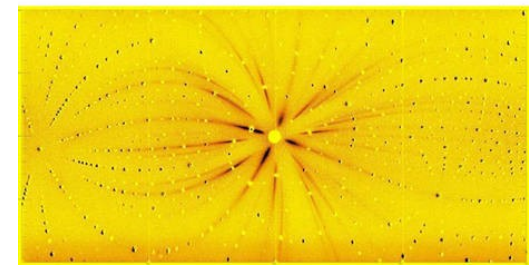
SANS

- Should better use 'cold' neutrons with $E < 5 \text{ meV}$
- Scattering obtained from structure on length scales in the material (low angle diffraction).
- $Q \sim 4\pi/\lambda \sin(\theta) \sim 2\pi r/(\lambda L)$ with L =sample-detector and r =radius of scattering ring on detector
- λ should be 'rather' monochromatic and cold/thermal
- Q range for **SANS@RTP** $0.01\text{-}0.1 \text{ Angs}^{-1}$
- Plot $I(Q^2)$, get R_G



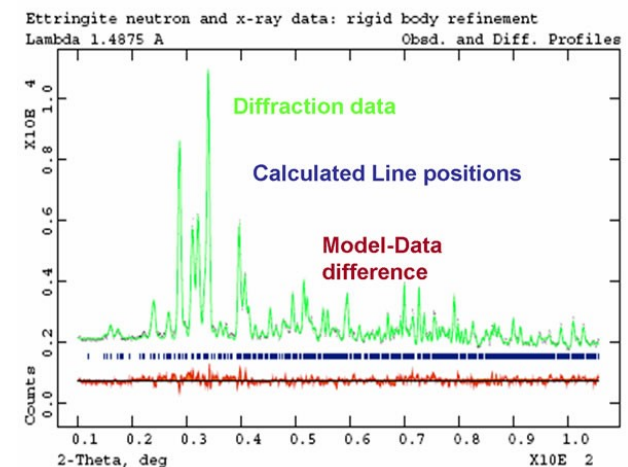
Diffraction: Laue

- Uses a white incoming beam (or band) → full flux
- Diffracts on a crystal sample → allow to get its structure
- Needs a 'small' divergence/beam section
- Needs an image plate/PSD to acquire image
- Adapted to thermal wavelength
- Easy to get qualitative but accurate data analysis is hard



Diffraction : powder

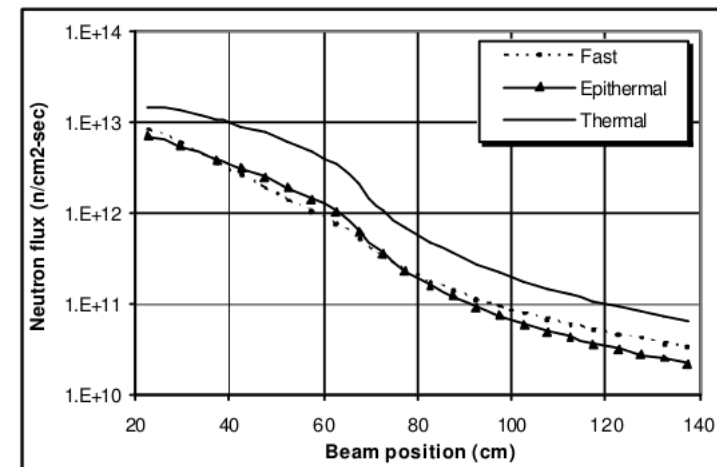
- Requires a monochromatic beam
- Requires a movable detector around sample or PSD
- Diffracts neutron on powder → give its structure
- Adapted to thermal wavelength
- Applications for industry: metallurgy, strain/stress



Neutrons from TRIGA

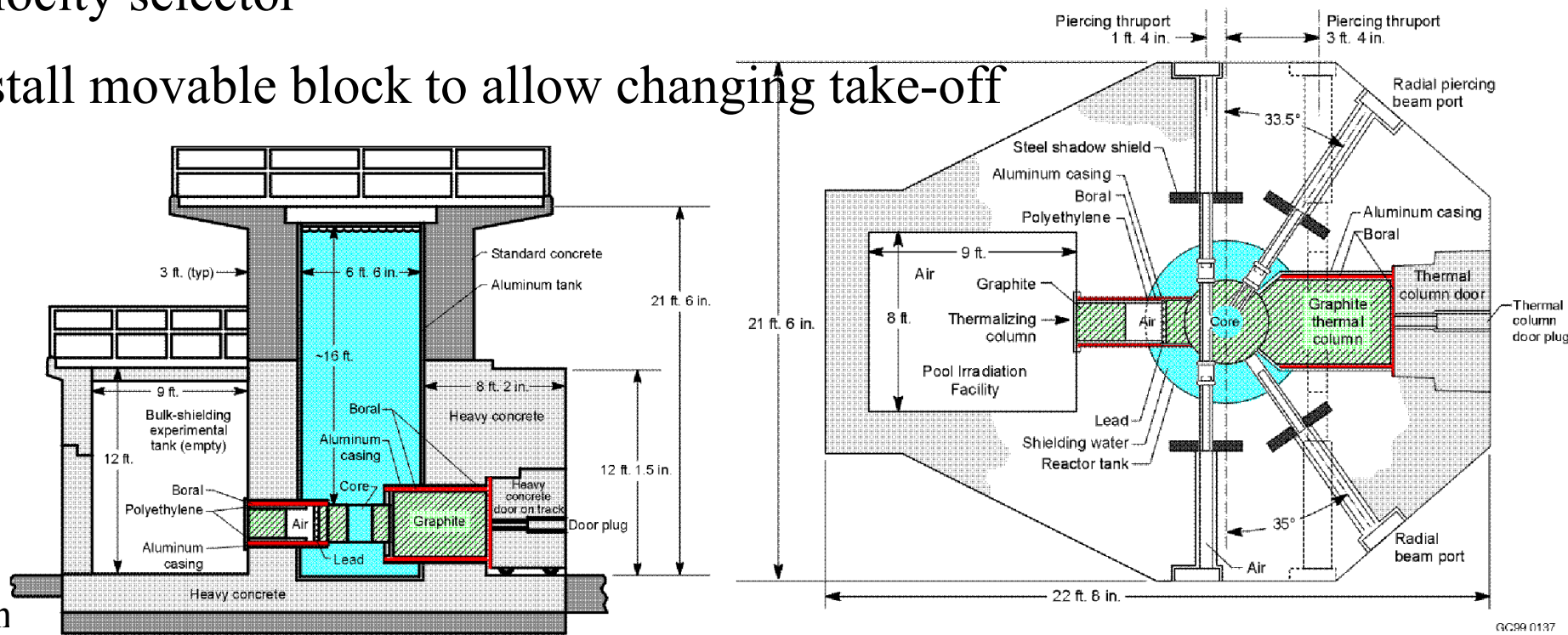
- 1MW nominal power $\sim 1/100e$ of ILL
- Flux is mostly thermal, with $1/100e$ fast/epithermal.
- Should allow to perform diffraction experiments within a day
- Tangential tubes as good as the radial ones, less γ

Radial tube, from Teruel,
Rizwan-uddin PHYSOR2004



Possibilities at TRIGA PUSPATI

- Use central thimble door plug to get maximum flux
- Use tangential tubes to reduce γ -level for NR/Laue
- Upgrade SANS monochromator (re-align)
- Install a powder diffractometer using one monochromator blade or velocity selector
- Install movable block to allow changing take-off



Tutorials

Source-Tube
Monochromator
Sample-Detector

McStas Installation Party

[<mcstas.org/download/>](http://mcstas.org/download/)

Windows:

- Use Installer (32bits – OK on 64), press Next until the End. Logout/Login.
- Will make use of Matlab if present at install (plotting).

Linux:

- Use Debian package (32 and 64 bits) on Ubuntu *et al.*

```
cd /etc/apt/sources.list.d
```

```
sudo wget http://packages.mccode.org/debian/mccode.list
```

```
sudo apt-get update ; sudo apt-get install mcstas-suite
```

- Recommended 3rd party:

OpenMPI, *freewrl* from [<www.freewrl.org>](http://www.freewrl.org) or *view3dscene*

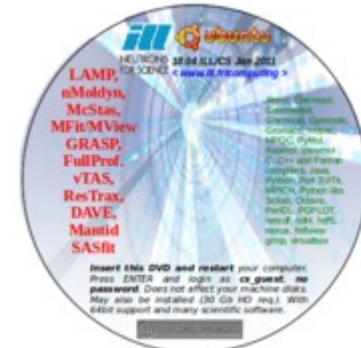


MacOSX: we supply a *dmg*, which requires Xcode (gcc)+Xquartz.

Any machine, without installation:

Get our **LiveDVD** at [<http://mcstas.org/ubuntu/>](http://mcstas.org/ubuntu/)

and [<http://www.ill.eu/instruments-support/useful-tools/>](http://www.ill.eu/instruments-support/useful-tools/)



McStas: Help me !

<www.mcstas.org> RTFM

<mcstas-users@mcstas.org> quick answer guaranteed !



McGUI Help menu (right top side)

→ Manuals, Component and example instrument list/doc.

McGUI Neutron Site menu: Example instruments listed

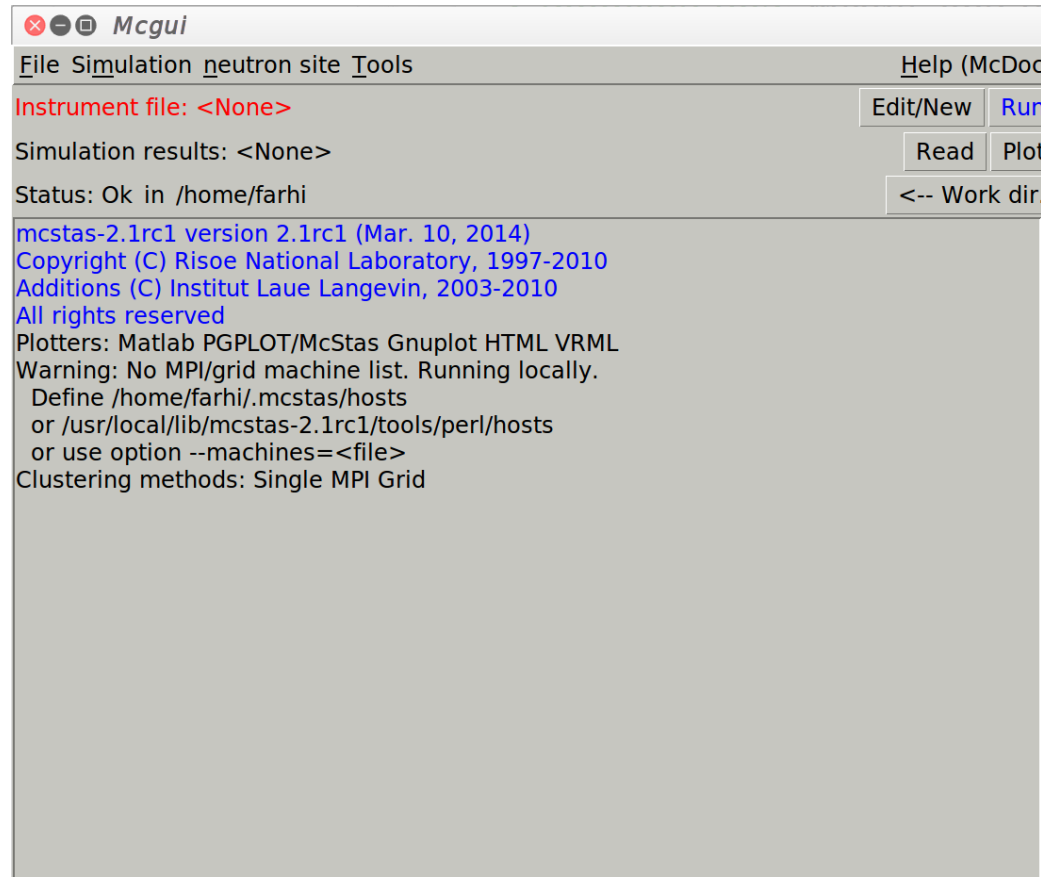
→ You will probably find an instrument which matches your needs, as a starting point.

McStas: Getting started

Windows and MacOSX: click on the McStas/McGUI Desktop Icon

LiveDVD: select menu Applications/Education/McStas

Command line: *mcgui*, and for geeks *mcrun*, *mcplot*, *mcdisplay*



The screenshot shows a window titled "Mcgui" with a menu bar containing "File", "Simulation", "neutron site", and "Tools". The window displays the following text:

```
Instrument file: <None>
Simulation results: <None>
Status: Ok in /home/farhi

mcstas-2.1rc1 version 2.1rc1 (Mar. 10, 2014)
Copyright (C) Risoe National Laboratory, 1997-2010
Additions (C) Institut Laue Langevin, 2003-2010
All rights reserved
Plotters: Matlab PGPLOT/McStas Gnuplot HTML VRML
Warning: No MPI/grid machine list. Running locally.
  Define /home/farhi/.mcstas/hosts
  or /usr/local/lib/mcstas-2.1rc1/tools/perl/hosts
  or use option --machines=<file>
Clustering methods: Single MPI Grid
```

Exercise 1: Source and Monitor

1. Start the McStas GUI: McGUI
2. Select *Neutron Site/Templates/template (test)*
3. Click on the **Edit** button. Remove the $Par1=0$ parameter.
4. Position the cursor in the TRACE after the *Origin* component.
5. From the Insert menu, select *Source/Source_gen* component
6. **Insert** a 'source' instance name, a 15x15 cm² geometry, focusing on a $\phi=12$ cm at 1.35 m, shooting $\lambda=4\pm 3.5$ Å neutrons, and position it at the (0,0,0) relative to the *Origin*. Use $I1=1e11$, $T1=300$ (thermal, TRIGA).
7. **Insert** a 'monitor' instance : 12x12 cm² Monitor_nD with *options="auto lambda bins=50"* at 1.3 m (along Z) from the *source*.
8. Save your instrument as *Source.instr*,
NOT in Documents and Settings
(*Windows does not handle spaces in file names correctly*).
9. Click on the **Run** button, indicate the directory to save to, e.g. *Source* and press **Start**.



Warning : in the following nasty neutrons will be shot
Virtual radiation hazard
Trespassers will be shot!
Survivors will be shot again!

Exercise 2: Monochromator

- In the DEFINE INSTRUMENT line, define $\lambda=4$ and $DM=3.355$
- At 50 cm from the previous λ -monitor, add an *Arm*
- On this *Arm*, add a *Monochromator* component $h*w=9*11$ cm, $DM=DM$, and rotate it around Y-axis with angle $asin(\lambda / (2*DM)) * RAD2DEG$
- Still on the *Arm*, define a second *Arm* rotated by twice the previous angle
- At 50 cm from the last *Arm*, place a *Monitor_nD* sensitive to wavelength
- Save simulation as *Monochromator.instr* and run it

Exercise 3: Sample and Detector

- At 2 m flight path, place a *Sans_spheres* 1 cm³
- At 3 m away place a PSD 30 cm² 128 bins
- Save simulation as *RTP_SANS_0.instr* and run it

Exercise 4: Be-filter

- In between the source and the 1st λ -monitor, install a *PowderN* with reflection file as “Be.laz” 15cm side.
- Save instrument *RTP_SANS_1.instr* and run it

Exercise 5: background from sample environment

- Around the sample, install a SiO₂ or V container, thickness 50 μm, with *PowderN(concentric=1)*. Place a COPY of this component after the sample with *concentric=0*.
- Copy the last PSD and set a condition to only detect the scattering on the container. This requires to define in DECLARE a 'flag' variable, and set it flag=SCATTERED in an EXTEND block in the container.
- Save instrument *RTP_SANS_2.instr* and run it

Exercise 6: Radiography

- Get the initial *Source.instr* (Exercise 1) and save it as *Radiography.instr*
- Just before the λ -monitor, place a PowderN with geometry="socket.off", made of copper 'Cu.laz'
- COPY the λ -monitor as an XY monitor, 512x512 pixels on 20x20 cm.
- Save and run instrument.

