ISTSI2019

St. Petersburg, June 29th 2019

News from the McStas project

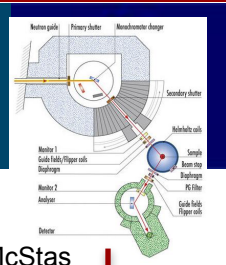
- including interoperability solutions for SIMRES, Vitess and MCNP

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

Agenda

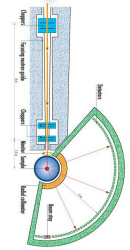
- Quick introduction to McStas
- Interoperability with other MC packages, such as SIMRES, Vitess and MCNP
- Recent and planned developments

McStas Introduction



McStas

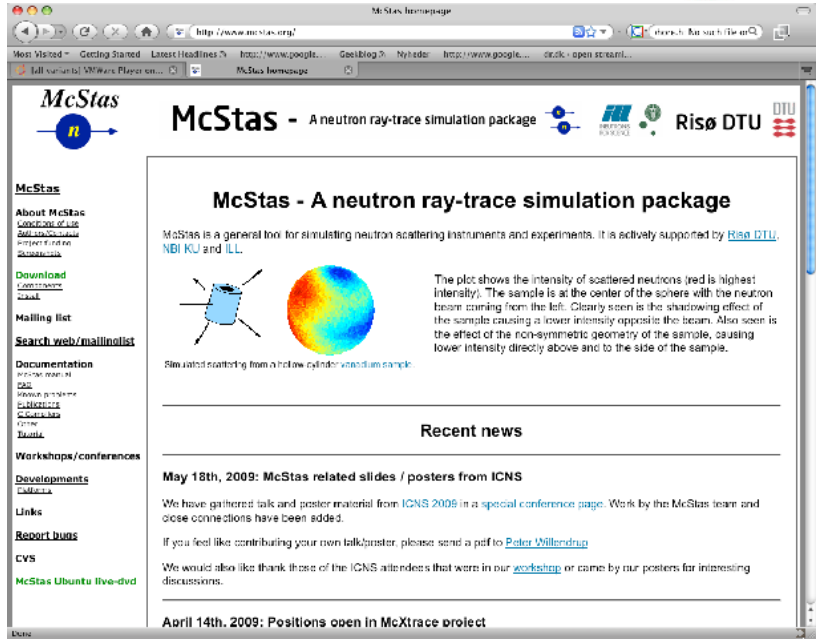
McStof?



GNU GPL license

Open Source

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



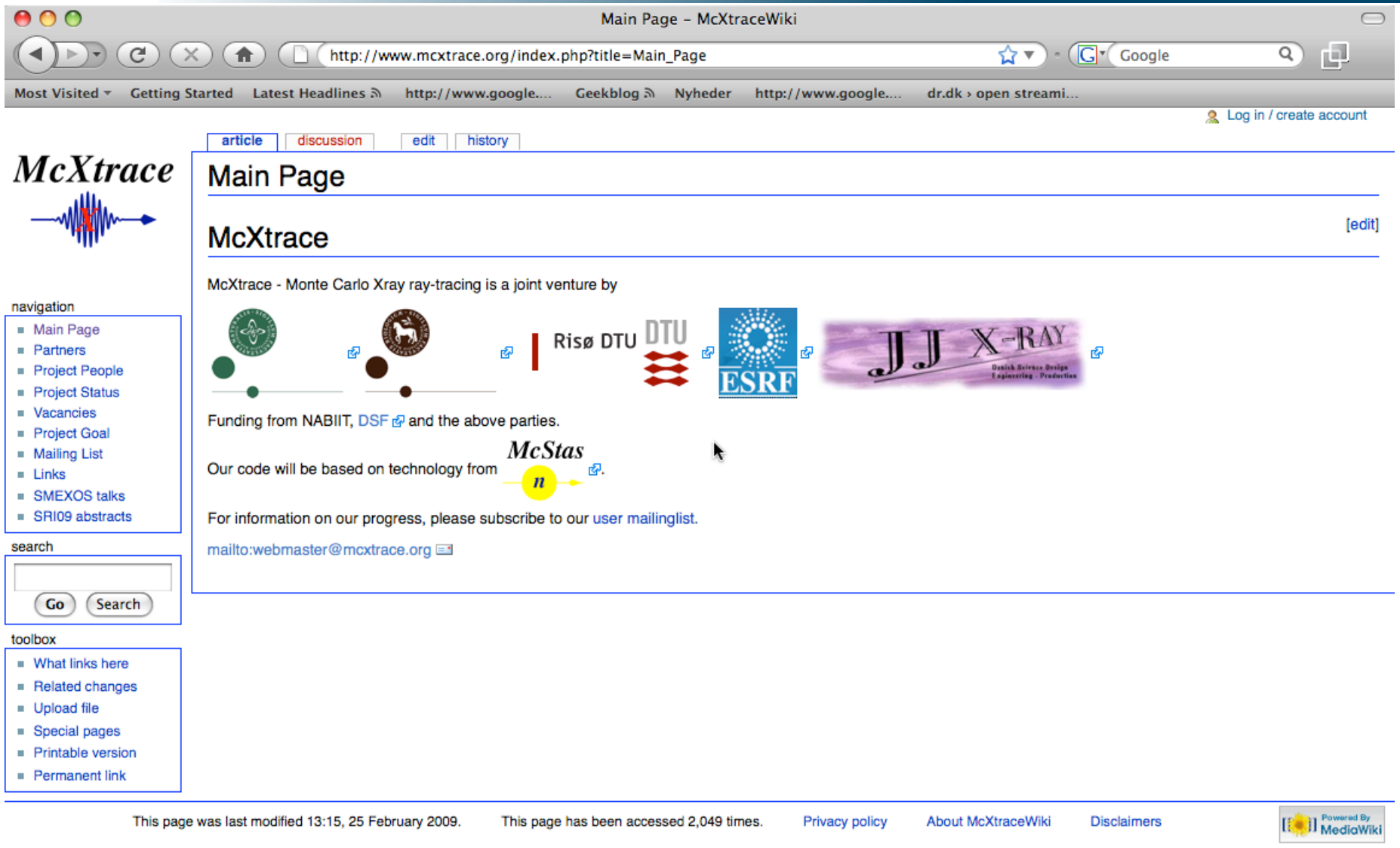
Project website at

http://www.mcstas.org

mcstas-users@mcstas.org mailinglist

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

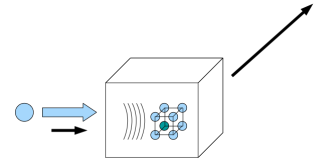
McXtrace - since jan 2009 similar for X-rays

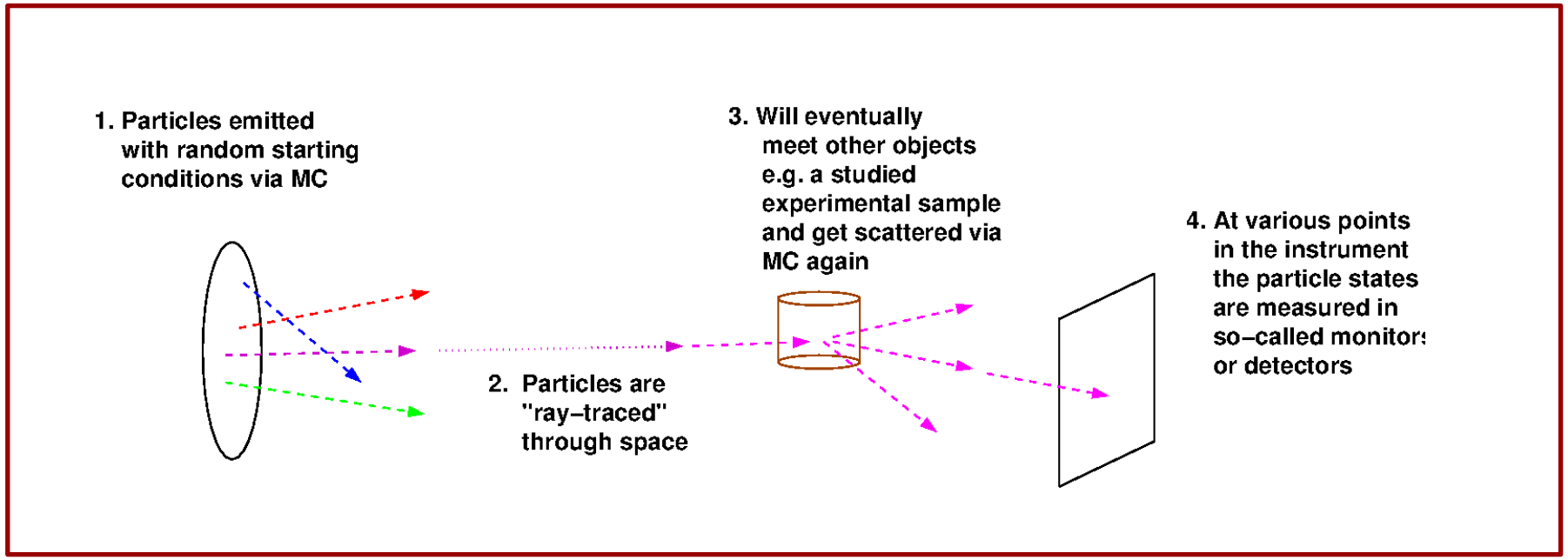


- Synergy, knowledge transfer, shared infrastructure

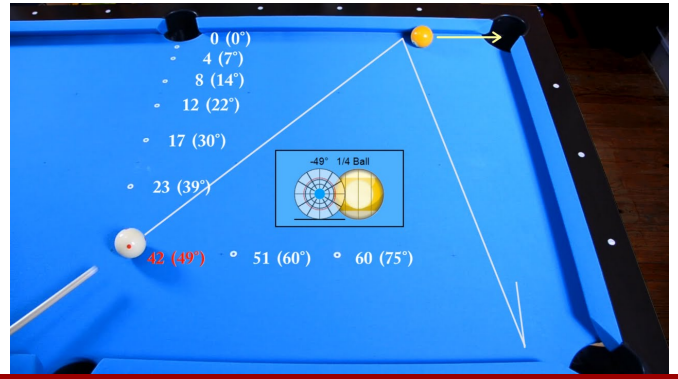
Used in many places

In the big picture, McStas is this…

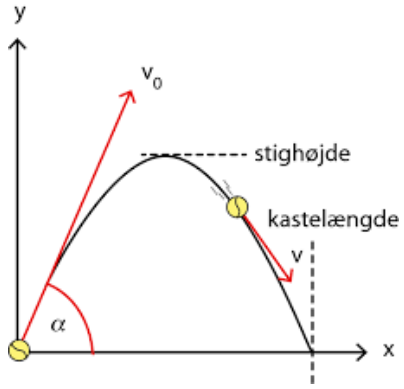




- Classical Newtonian mechanics, i.e.

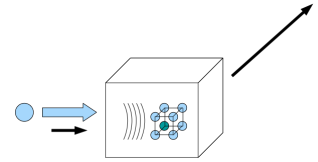


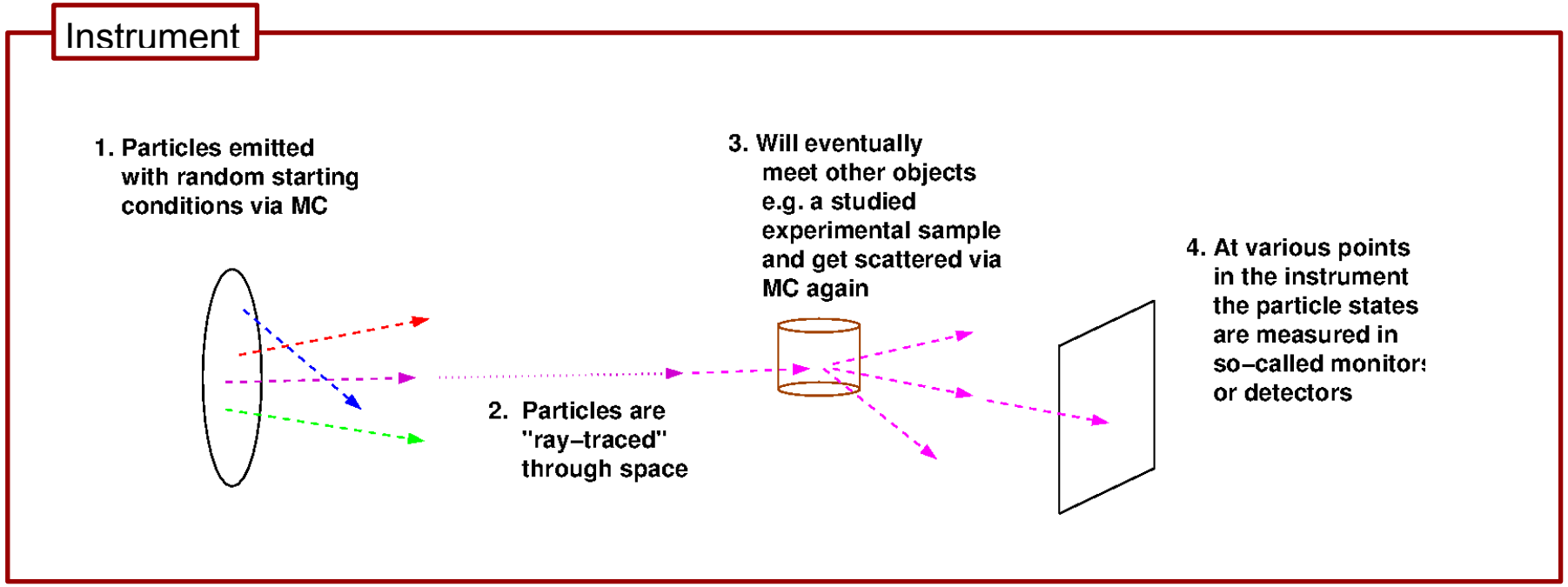
and



- (independent, particles though…)

In the big picture, McStas is this…

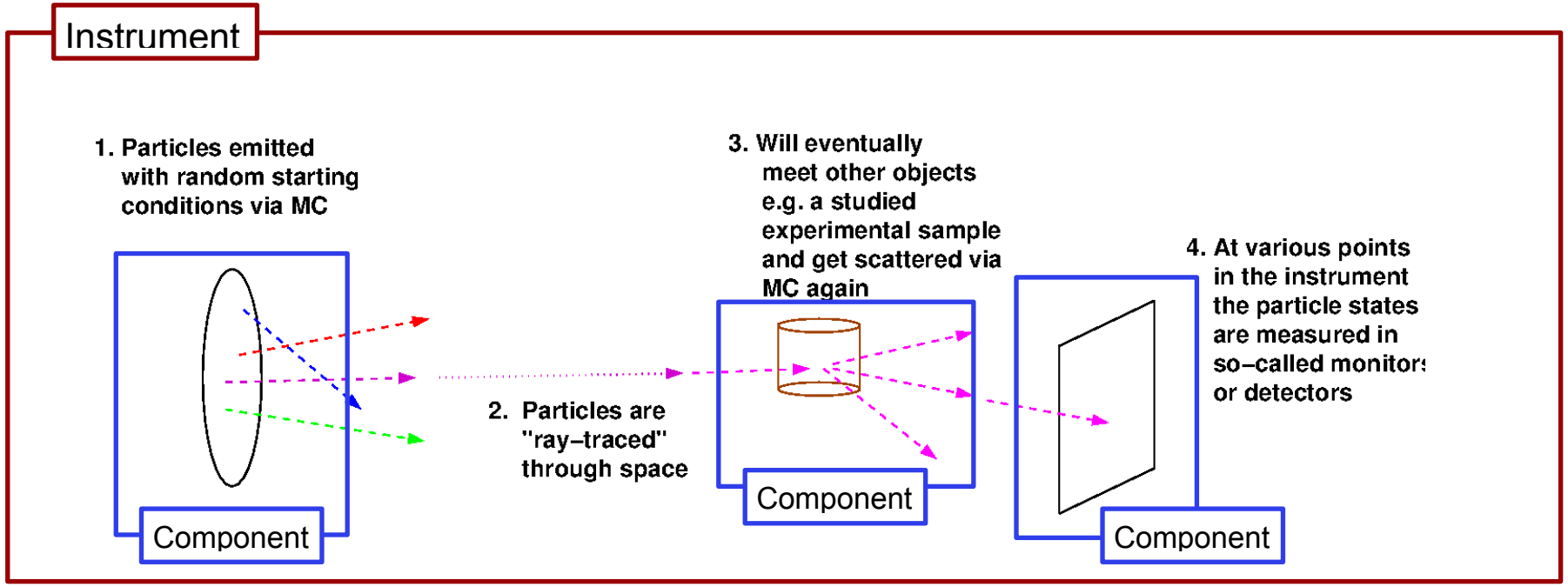


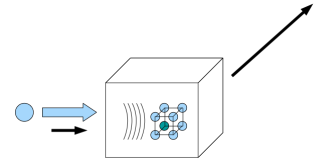


The instrument defines our “lab coordinate system”

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

In the big picture, McStas is this…



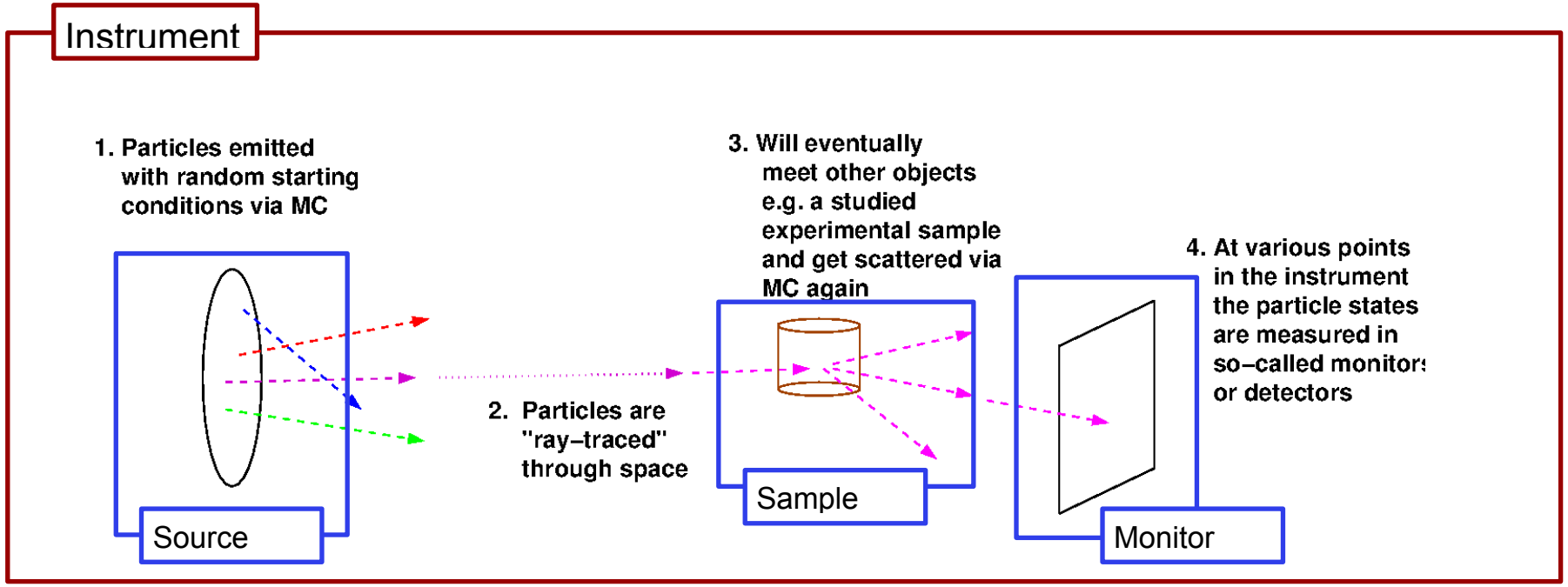


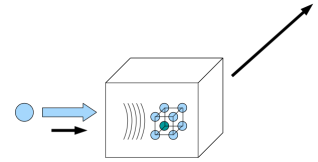
The instrument defines our “lab coordinate system”

The components define devices or features available in our instrument

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

In the big picture, McStas is this…



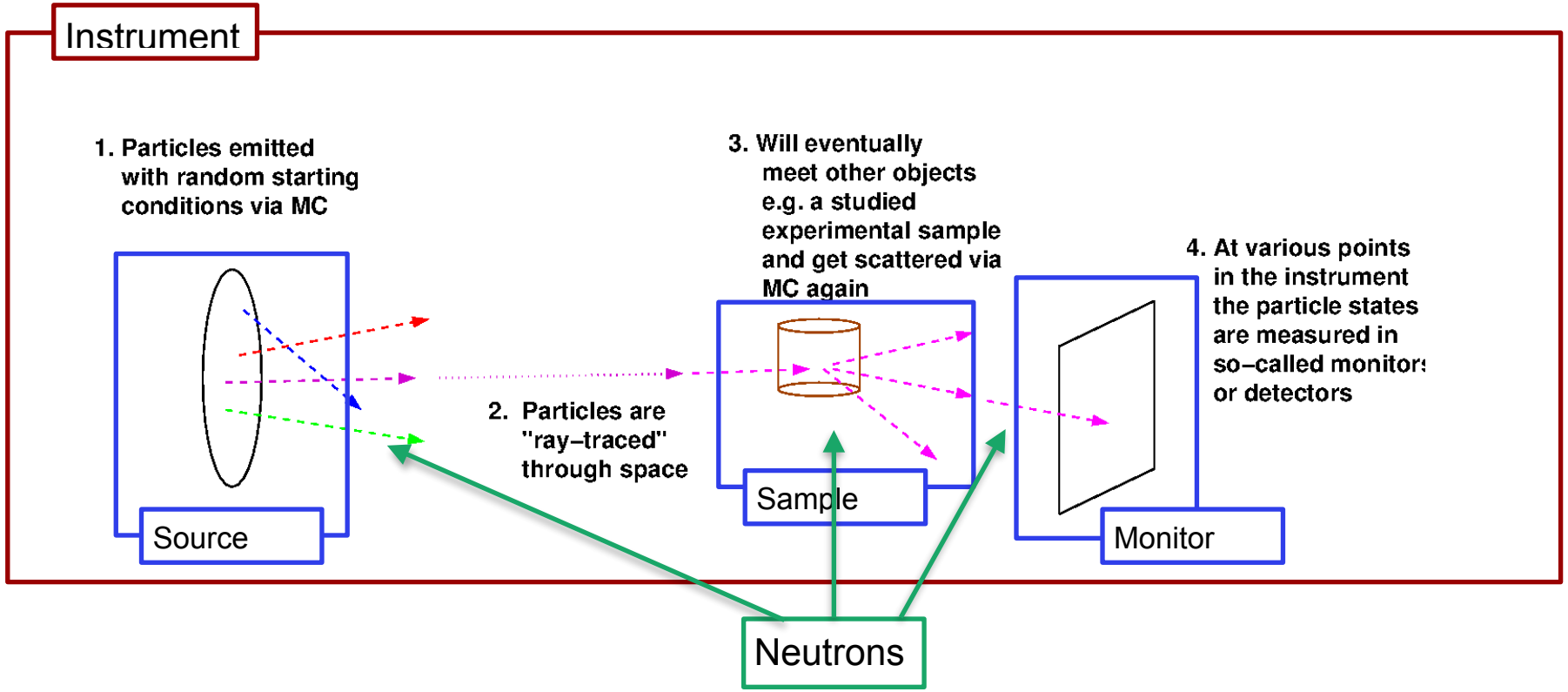


The instrument defines our “lab coordinate system”

The components define devices or features available in our instrument - they have different function

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

In the big picture, McStas is this…



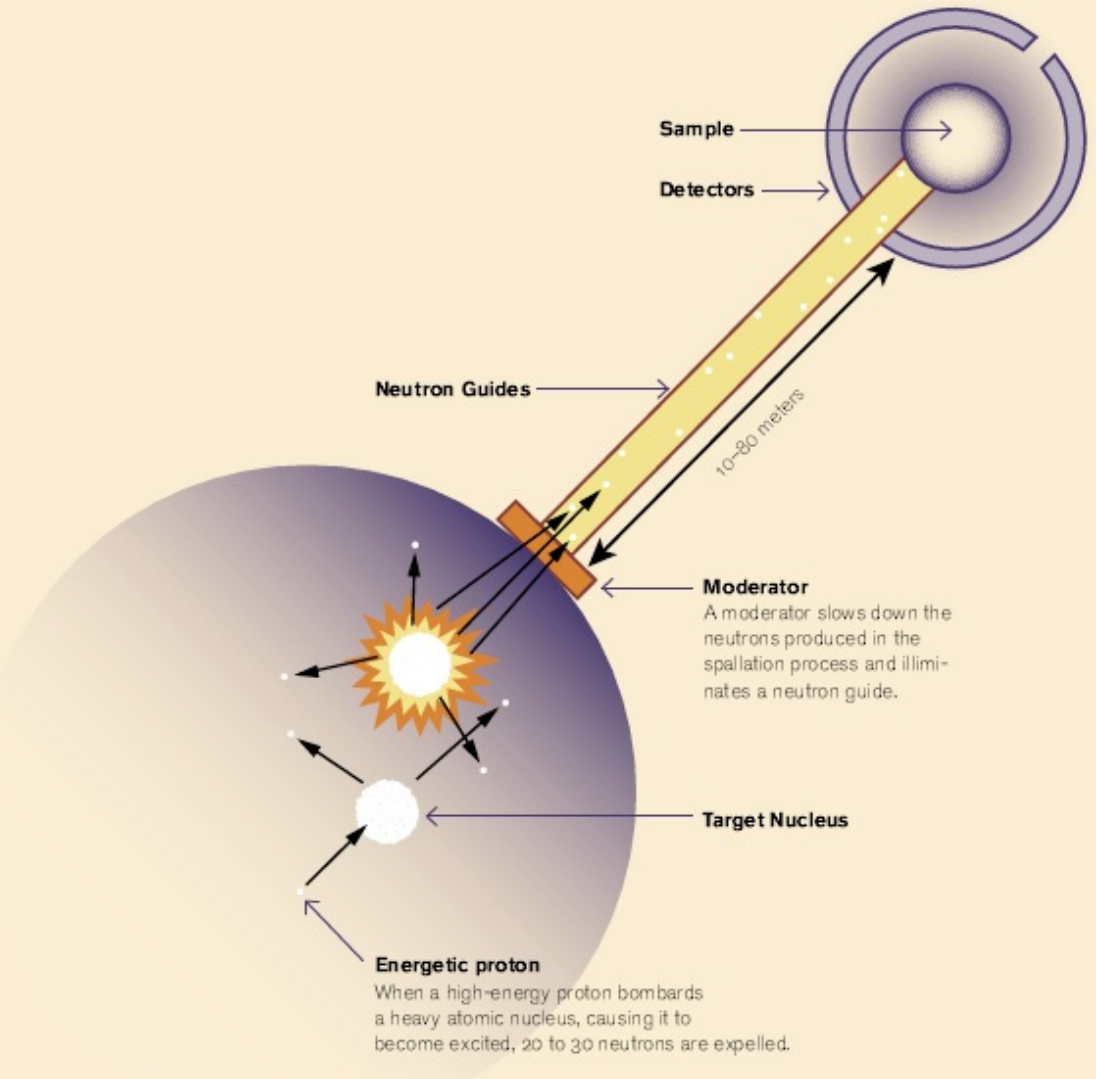
The instrument defines our “lab coordinate system”

The components define devices or features available in our instrument - they have different function

Neutron particles are passed on from one component to the next, changing state under way

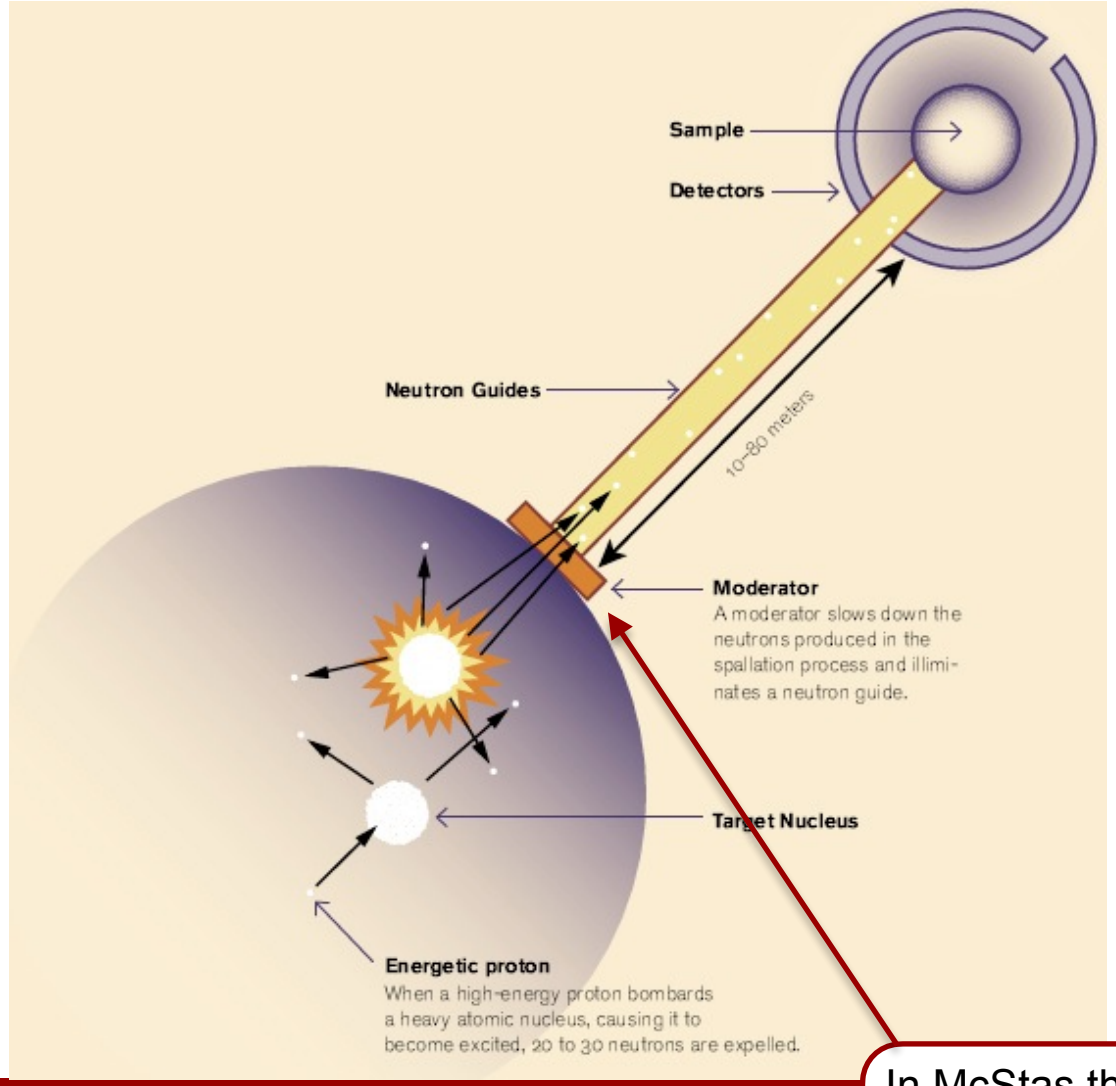
This project is funded by the European Union (GA no. 654000)

Components of neutron instruments



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

Components of neutron instruments

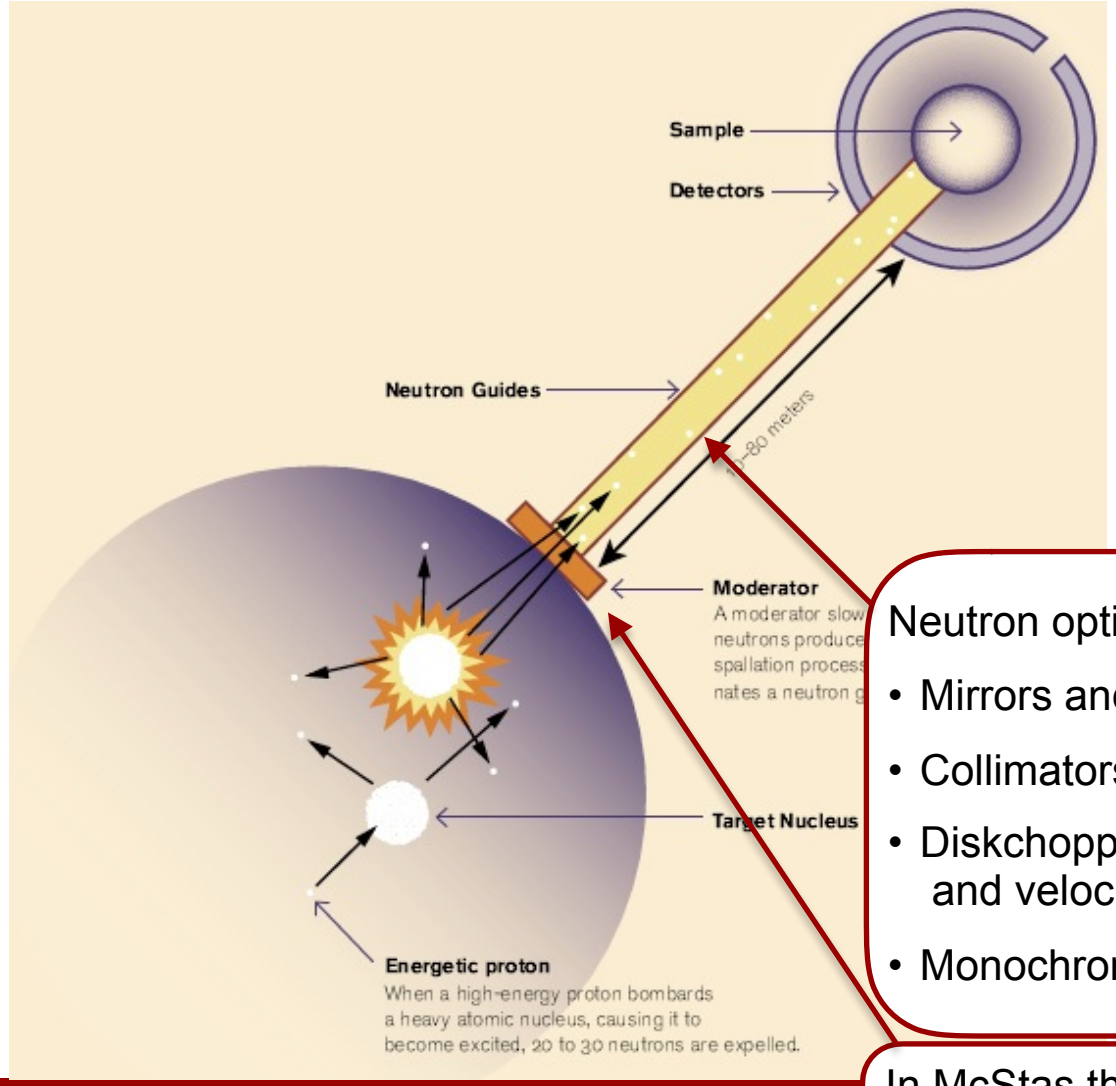


In McStas the moderator is the “source”

“Input” from e.g. MCNP

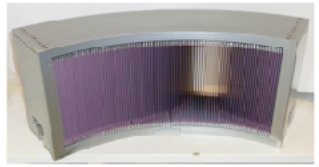
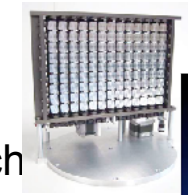
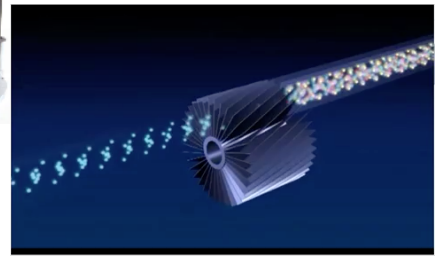
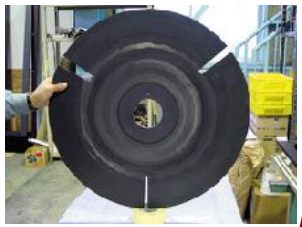
This project is funded by the European Union (GA no. 654000)

Components of neutron instruments



Neutron optics include things like:

- Mirrors and guides
- Collimators and slits
- Diskchoppers, Fermi ch
  and velocity selectors
- Monochromators/Analysers

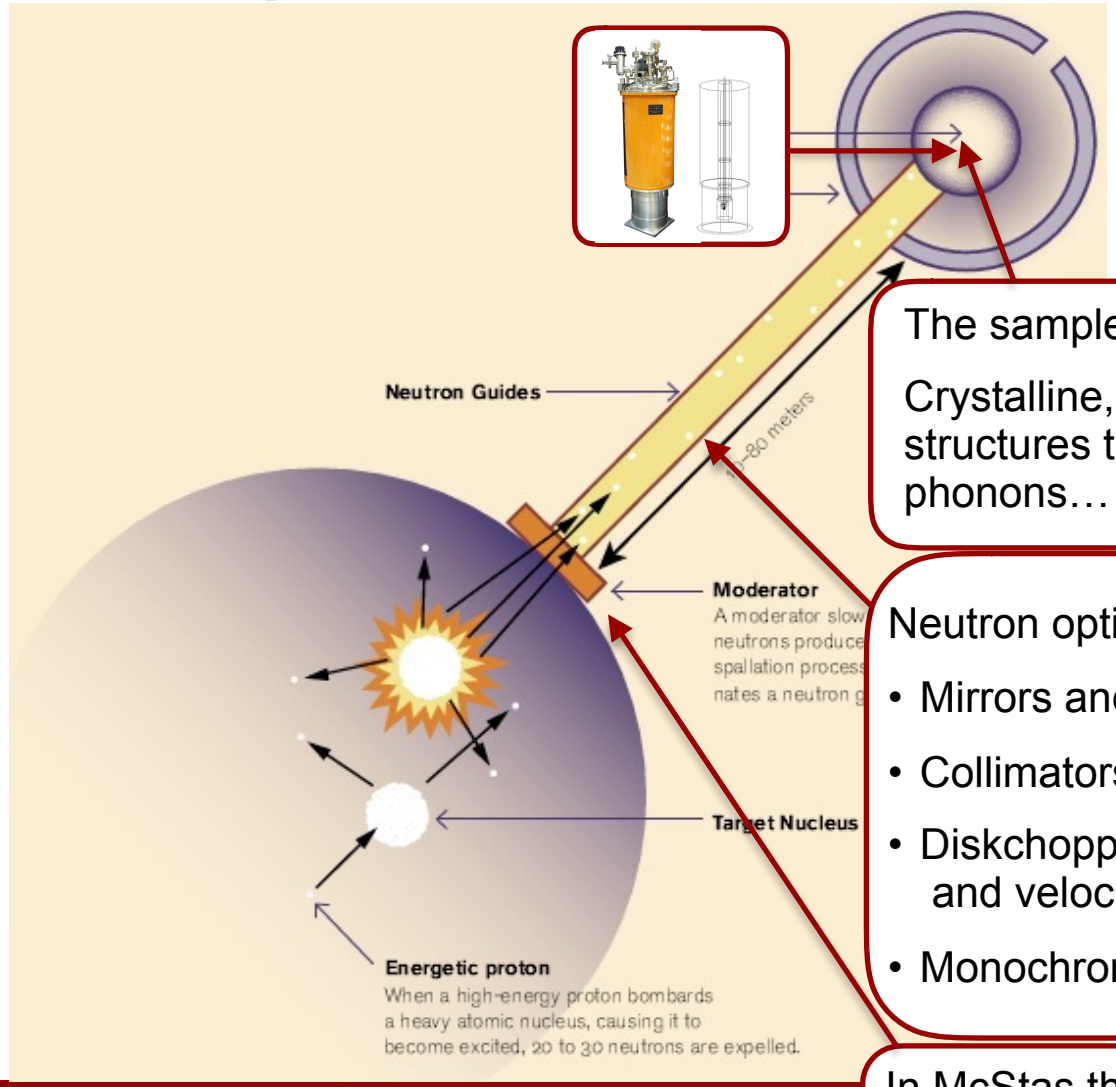






In McStas the moderator is the “source”

“Input” from e.g. MCNP

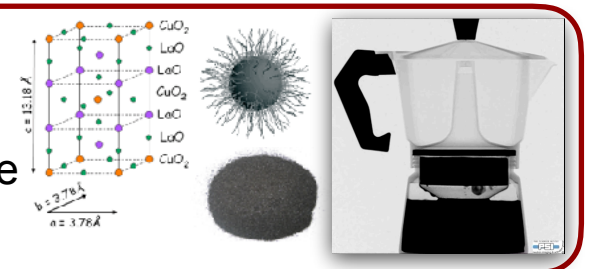
This project is funded by the European Union (GA no. 654000)

Components of neutron instruments



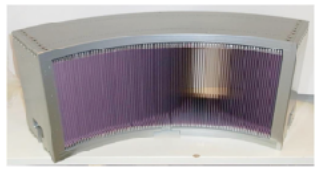
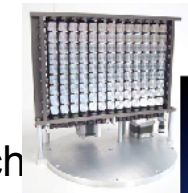
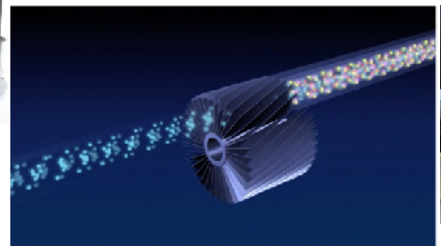
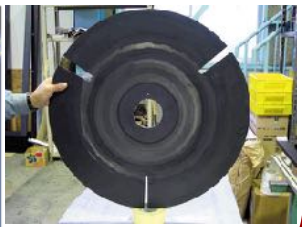
The sample (optionally with env.):

Crystalline, powders, liquids, micelles, structures to image, inelastic features like phonons…



Neutron optics include things like:

- Mirrors and guides
- Collimators and slits
- Diskchoppers, Fermi ch and velocity selectors
- Monochromators/Analysers

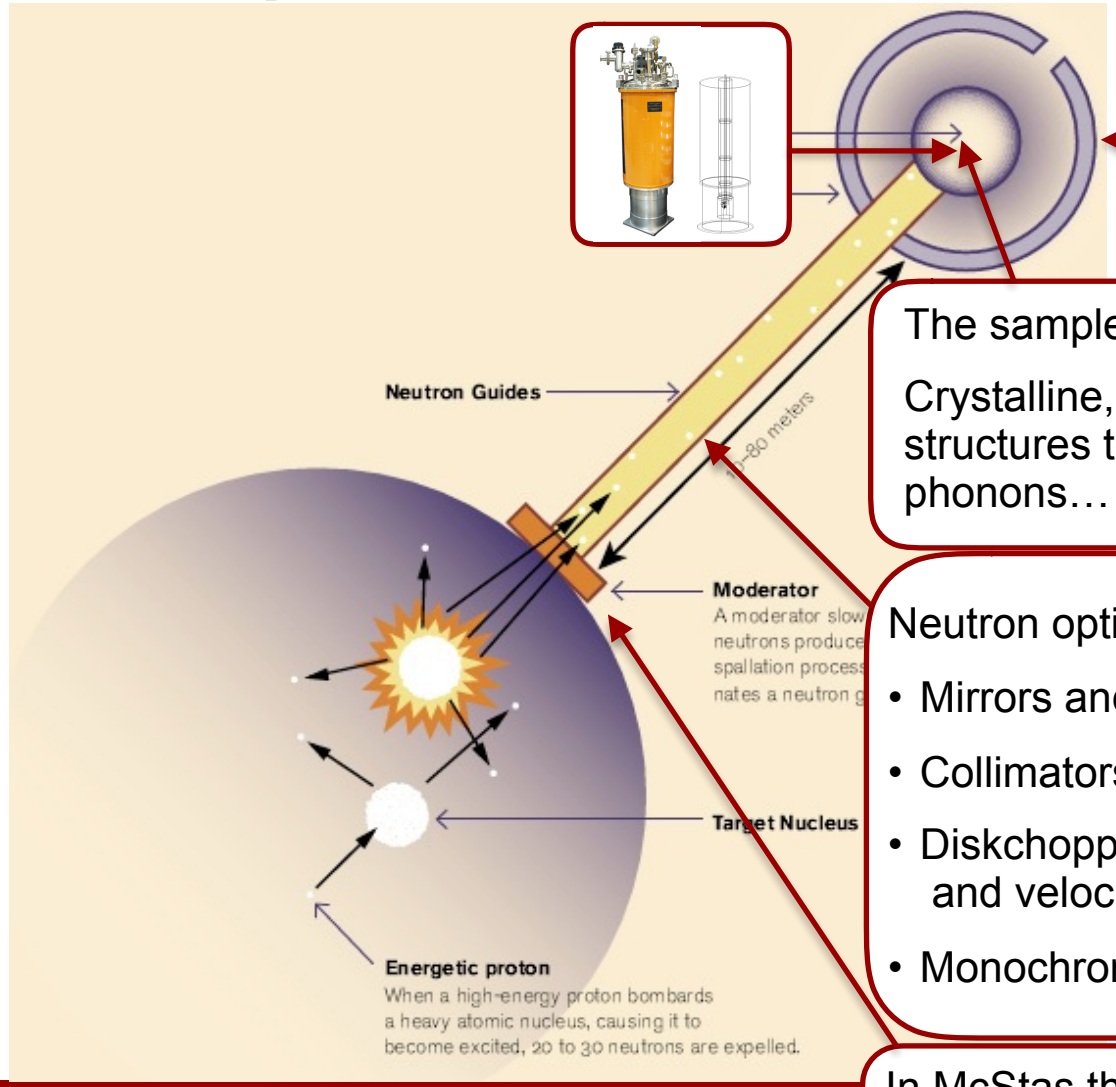
In McStas the moderator is the “source”

“Input” from e.g. MCNP

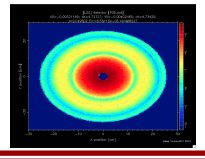
This project is funded by the European Union (GA no. 654000)

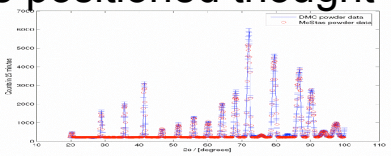
Components of neutron instruments

(One exception: He^3 gas detector)



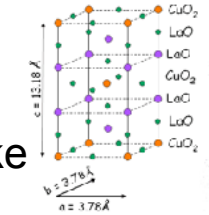
Detectors are “monitors” in McStas. Mostly they act as “perfect probes” and can be positioned thought your instrument gathering 1D/2D/ event lists…

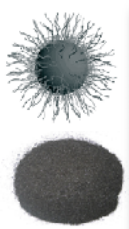




The sample (optionally with env.):

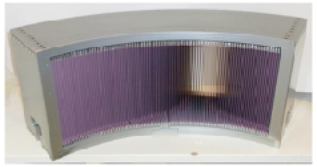
Crystalline, powders, liquids, micelles, structures to image, inelastic features like phonons…

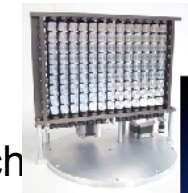


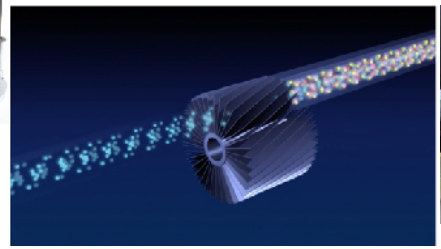


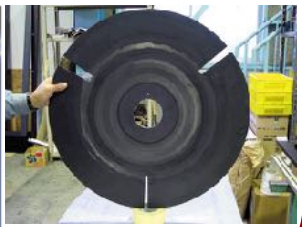
Neutron optics include things like:

- Mirrors and guides
- Collimators and slits
- Diskchoppers, Fermi ch and velocity selectors
- Monochromators/Analysers









In McStas the moderator is the “source”

“Input” from e.g. MCNP

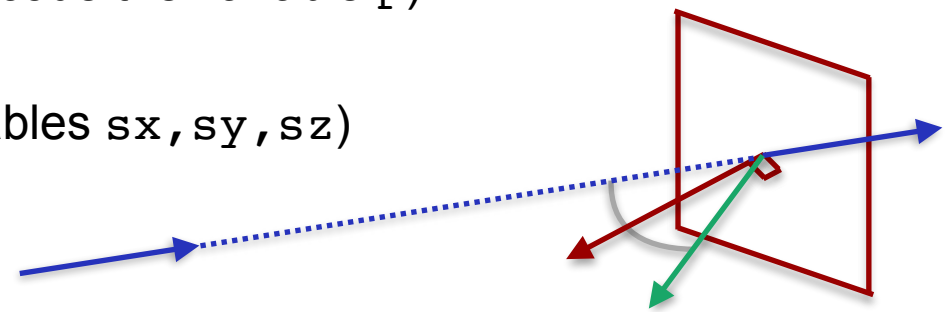
This project is funded by the European Union (GA no. 654000)

Neutron rays in McStas - what are they?

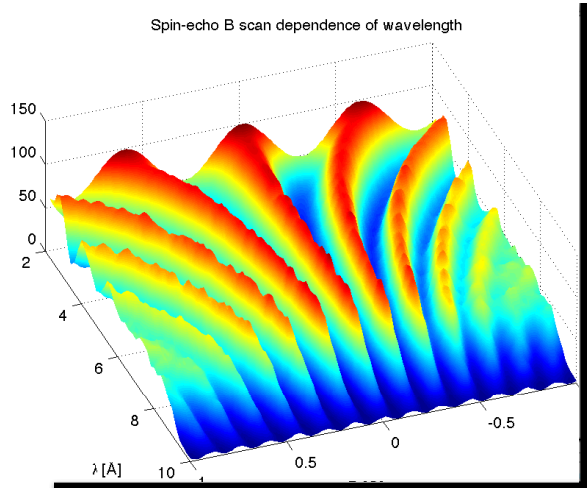
Neutron **ray** in McStas:	
Location	`x,y,z`
Velocity	`vx,vy,vz`
Time	`t`
Polarisation.	`sx,sy,sz`
Intensity	`p`

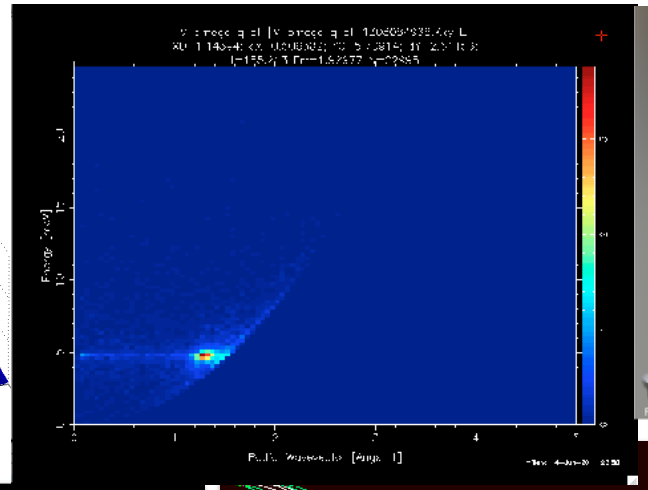
- Defining the neutron starting conditions imply setting:

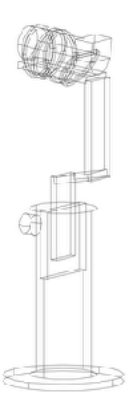
  - The **location** in space, i.e. $\vec{r}$ (in the code variables `x,y,z`)
  - The **direction** and λ / E_{kin} (in the code variables `vx,vy,vz`)
  - The **time** (in the code the variable `t`)
  - The **intensity** / weight of the neutron ray (in the code the variable `p`)
  - If needed the **polarisation** (in the code the variables `sx,sy,sz`)

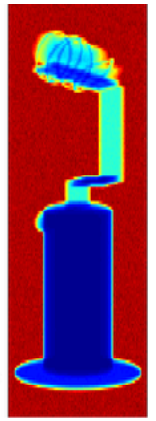


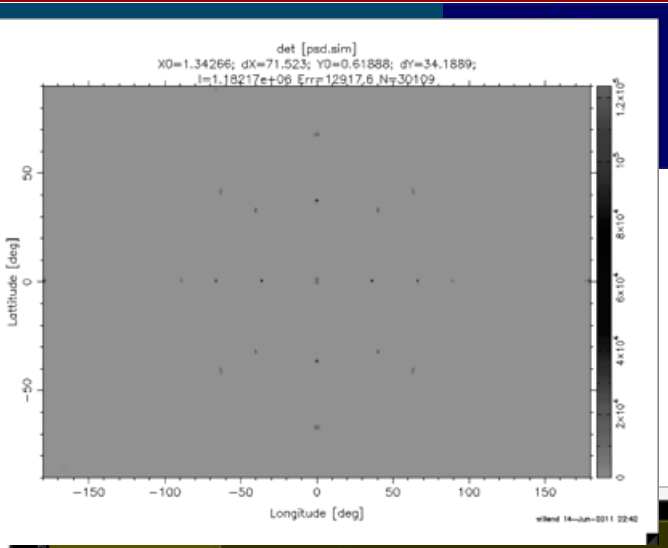
Example suite: 201 instruments

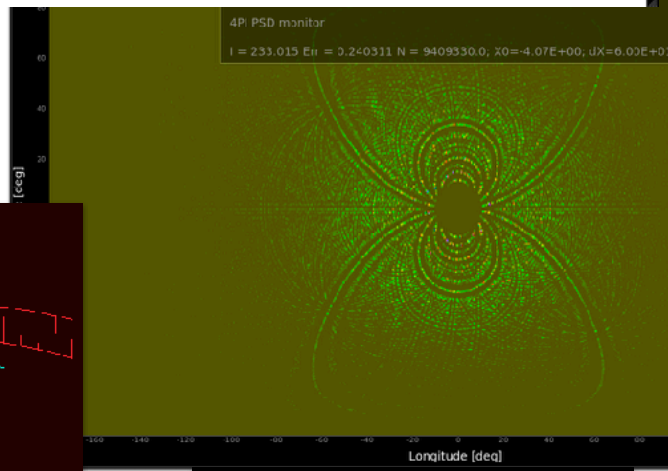


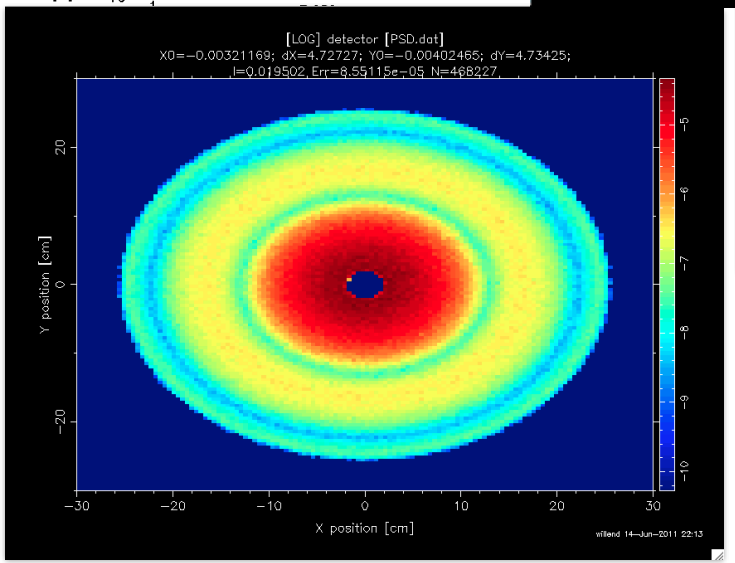


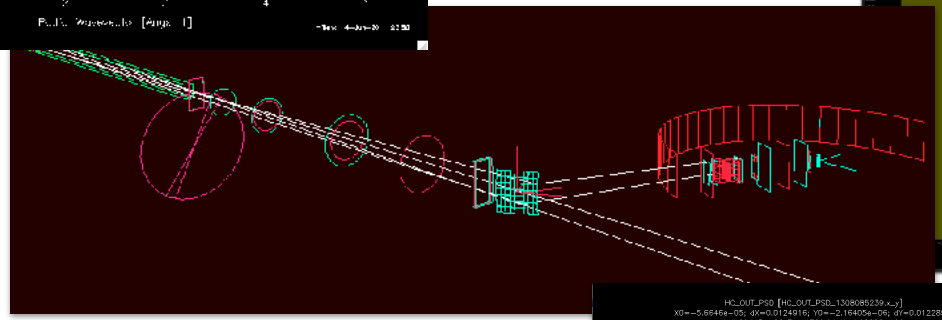


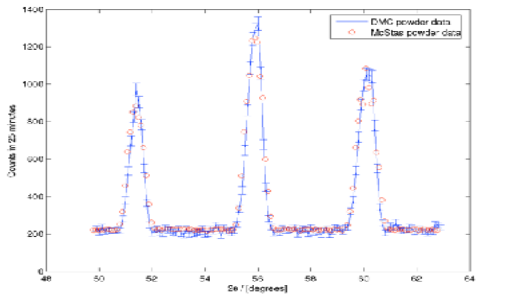


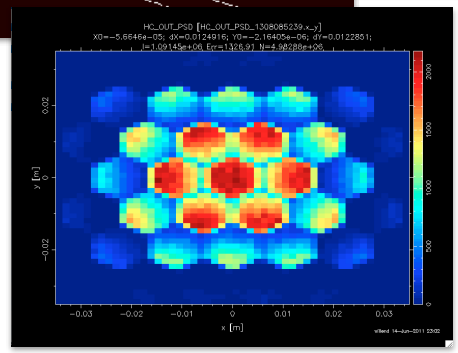


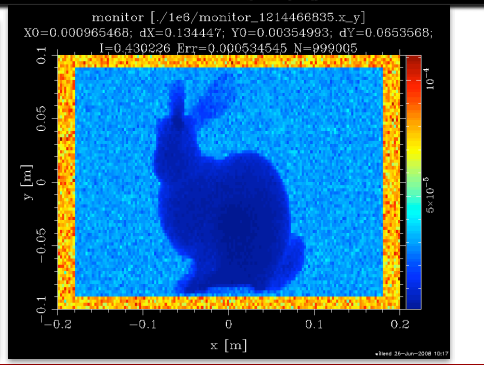








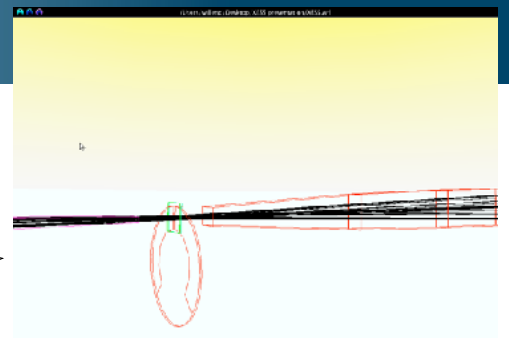


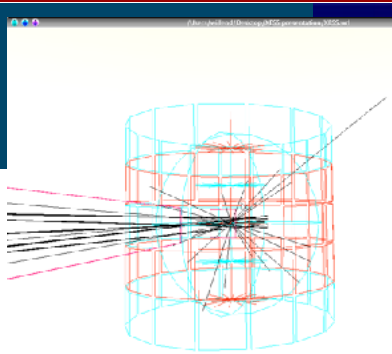


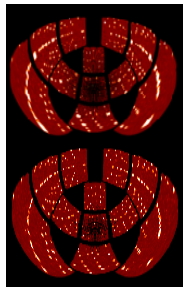
This project is funded by the European Union (GA no. 654000)

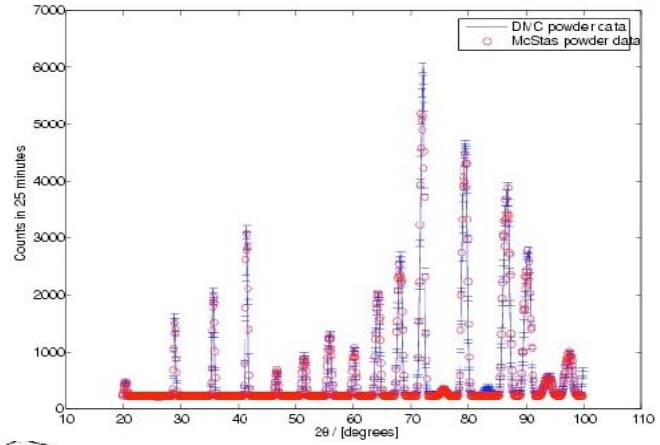
What is McStas used for?

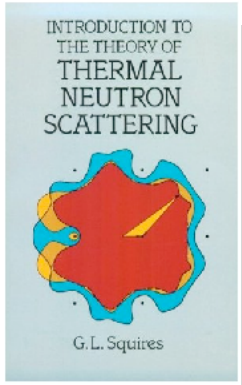
- Instrumentation
- Planning
- Construction
- Virtual experiments
- Data analysis
- Teaching
  (KU, DTU)

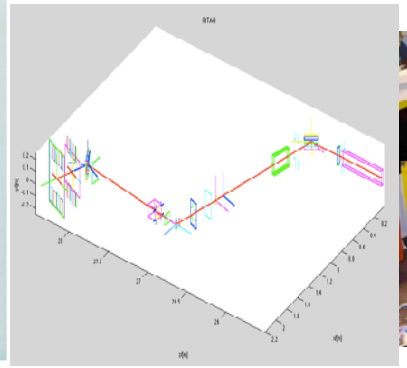


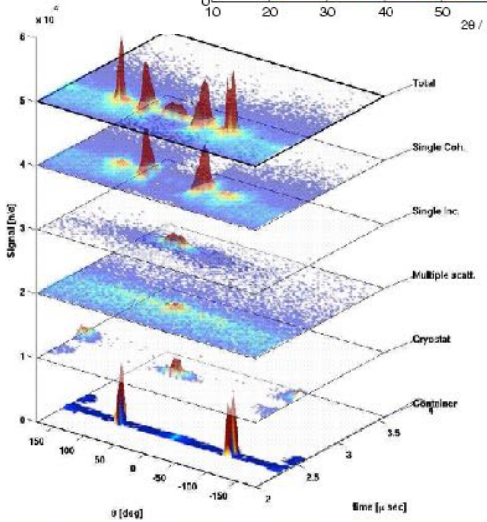








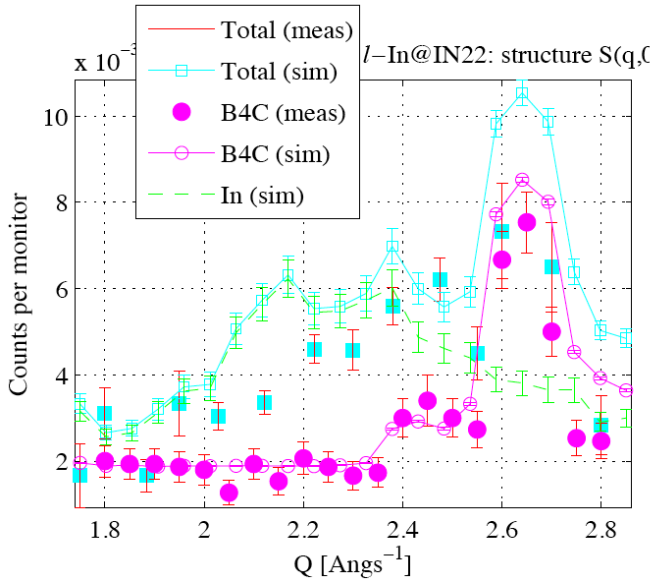




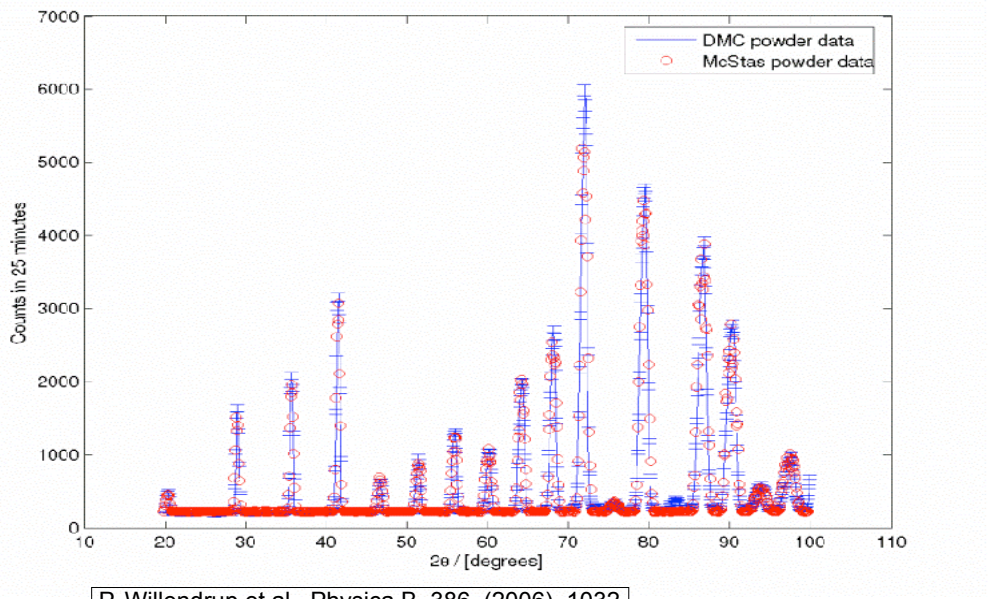
This project is funded by the European Union (GA no. 654000)

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



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

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

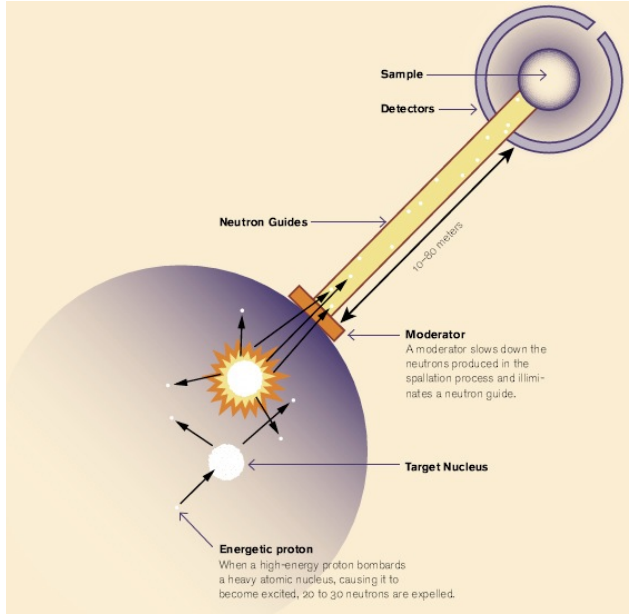
McStas overview

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

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

- 'Component' files (~200) inserted from library
  - Sources
  - Optics
  - Samples
  - Monitors
  - If needed, write your own comps

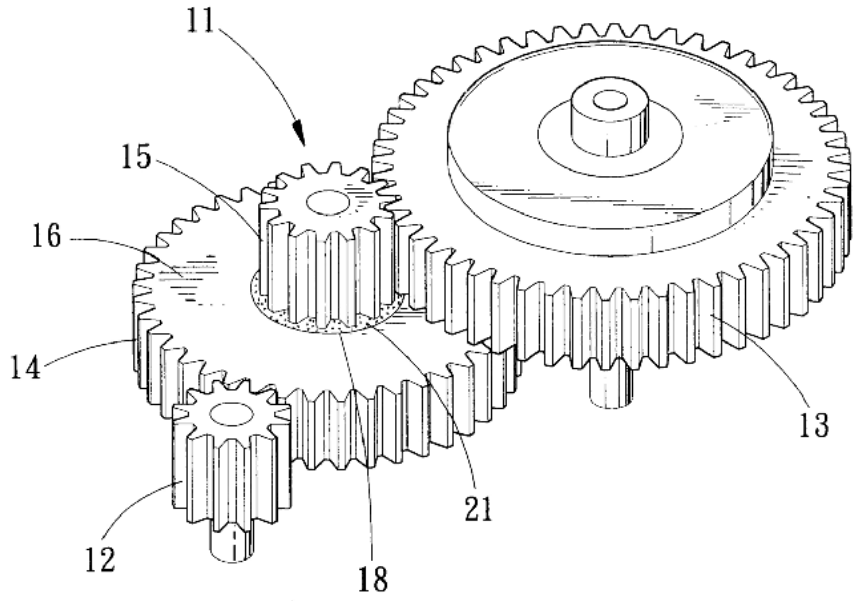
- DSL + ISO-C code gen.



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

Under-the-hood / inner workings

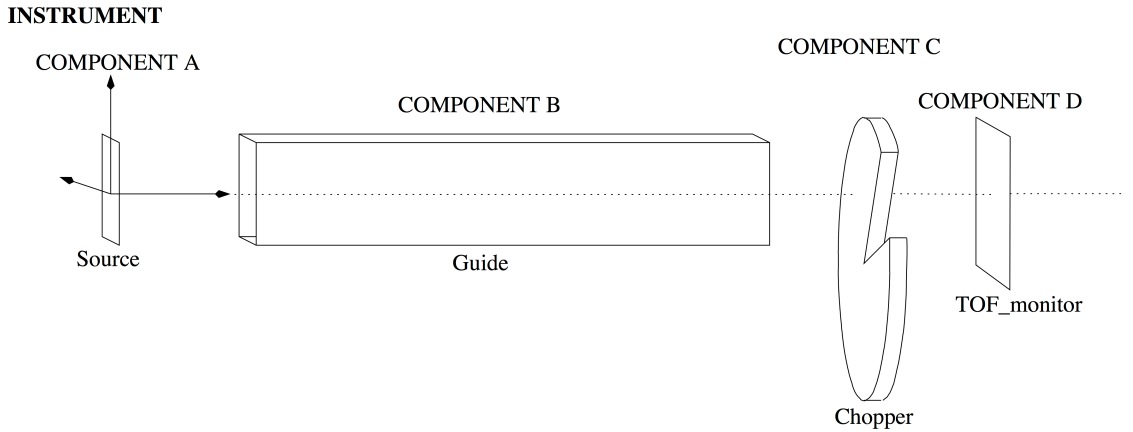
- Domain-specific-language (DSL) based on compiler technology (LeX+Yacc)
  - Simple Instrument language —Code generation→ ISO C
- Component codes realizing beamline parts (including user contribs)
- Library of common functions for e.g.
  - I/O
  - Random numbers
  - Physical constants
  - Propagation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



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

Implementation

- Three levels of source code:
  - Instrument file (All users)
  - Component files (Some users)
  - ANSI c code (no users)



The "tool layer" consists of programs manipulated by the McStas user:

mcgui, graphical user interface	**mcplot, visualize histogram outp.**	**mcdisplay, visualize instrument**

mcgui is used to assemble an instrument file, which is taken over by the McStas system

```
DEFINE INSTRUMENT Example(Param1=1, string Param2="two", ...)

COMPONENT A = Source(Parameters...)
AT (0, 0, 0) ABSOLUTE

COMPONENT B = Guide(Parameters...)
AT (0, 0, 1) RELATIVE A

COMPONENT C = DiskChopper(Parameters...)
AT (0, 0, 1) RELATIVE B

COMPONENT D = TOF_monitor(Parameters, filename="Tof.dat")
AT (0, 0, Param1) RELATIVE PREVIOUS
```

"Instrument file"

Source.comp – c–code

Guide.comp – c–code

DiskChopper.comp – c–code

TOF_monitor.comp – c–code

Component library

Random numbers | I/O | Physical consts.

"Kernel and runtime c–code"

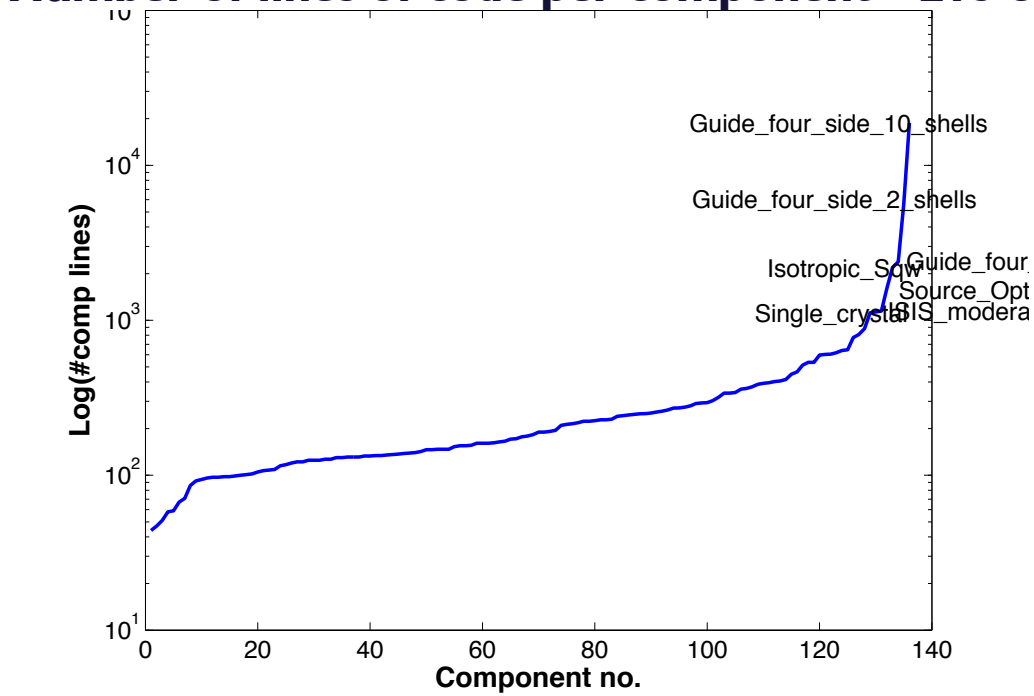
The McStas system generates an "ISO C file" and an executable from instrument file and c–codes

The simulation executable produces data output which can be visualized using the mcplot and mcdisplay tools

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

Writing new comps or understanding existing is not complex…

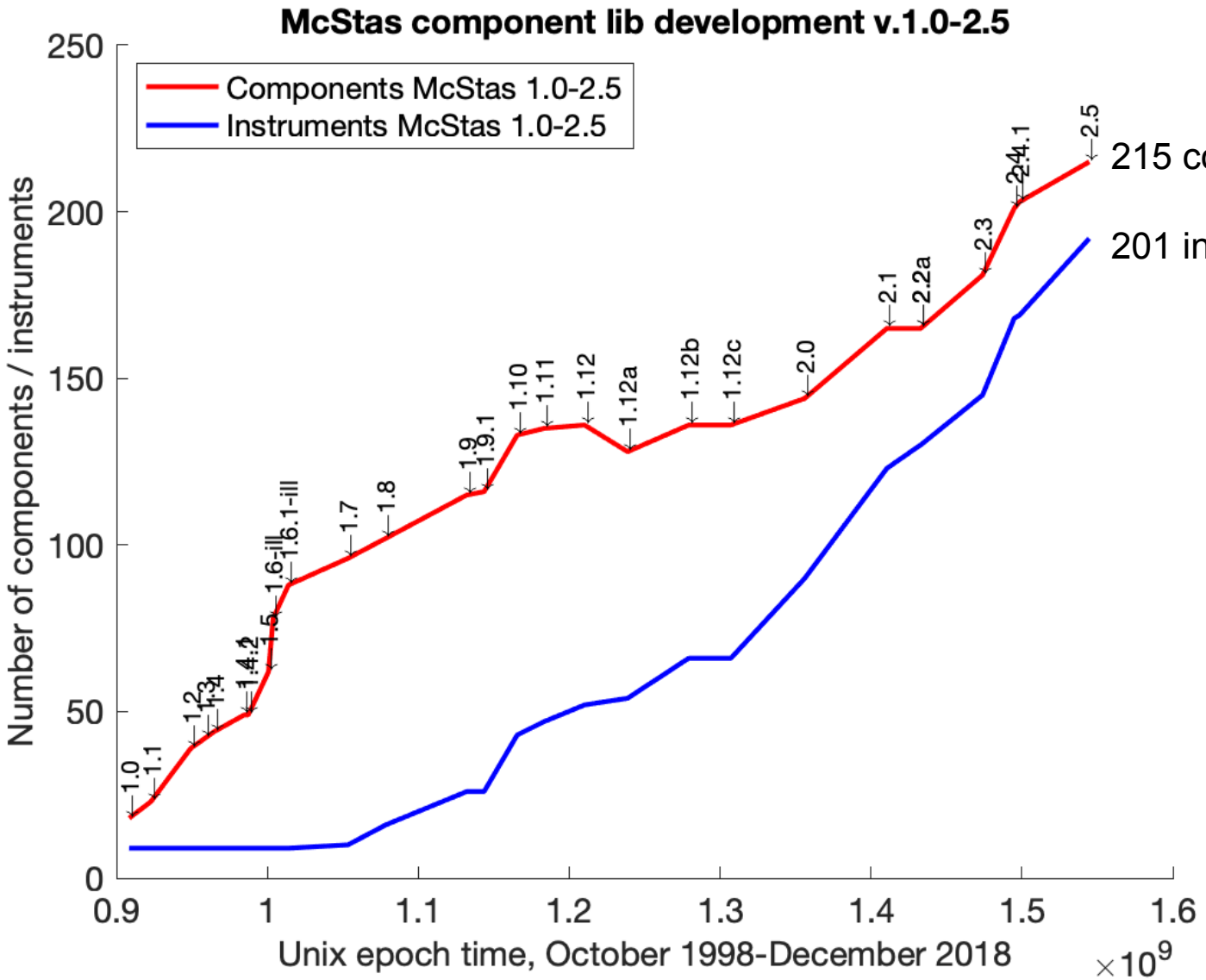
- Check our long list of components and look inside... Most of them are quite simple and short... Statistics:



- Well-developed community support
  - 45% of existing components 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

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

Component and example lib over time



215 components

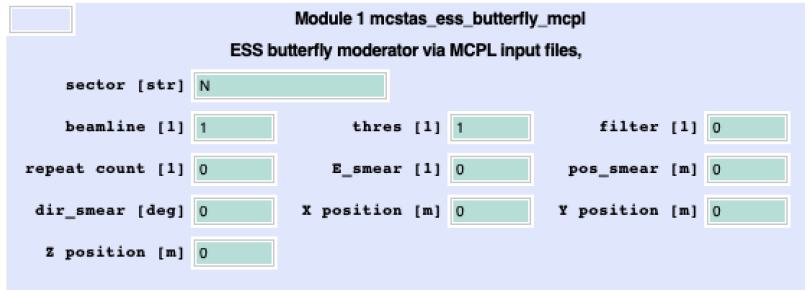
201 instruments

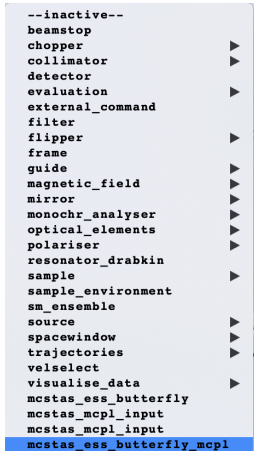
Plan to increase “test coverage” further by addition of more test-oriented instruments

Interoperability with Vitess

- Legacy solution:
  - Event files in “Vitess format”, i.e. what Vitess communicates on the pipe
  - Event files in “McStas formet” i.e. what McStas used to write using Virtual_output

  - mcstas2vitess script (perl, K Nielsen ~ Y2K) that produces (example here)
    - a McStas instrument (and binary) containing the component with normal Vitess I/O
    - a tcl snippet for inclusion in the Vitess gui





- Todays solution:
  - Use either mcstas2vitess or MCPL files which are supported in both codes

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

Interoperability with SIMRES

- Facilitated using MCPL files
  - For details, see MCPL talk

- Enables encapsulation of McStas components directly in SIMRES
  - Standard naming for the i/o files, allows embedding directly in any given SIMRES release, already prepared are these comps / instrs:

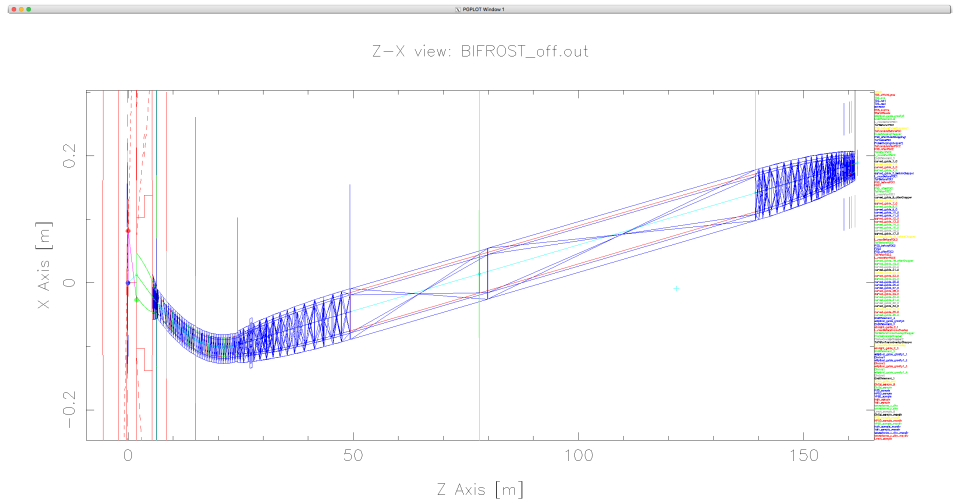
```
McStas_Isotropic_Sqw.instr
McStas_PowderN.instr
McStas_Single_crystal.instr
```

- The same method could be used on the Vitess side
- For further details see SIMRES talk

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

Interoperability with MCNP/X

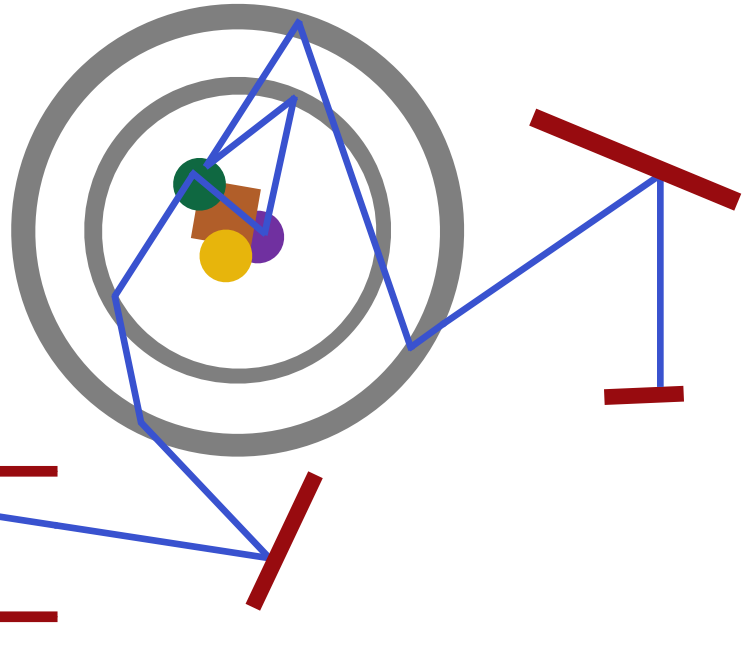
- Legacy solution:
  - Event files in “MCNP ptrac format” (ascii, error-prone one-way?)
- Today’s recommended solution:
  - Use MCPL files which are supported in both codes
- Thinking in direction of auto-transferring parts of MCNP geometry to the McStas side (CombLayer)

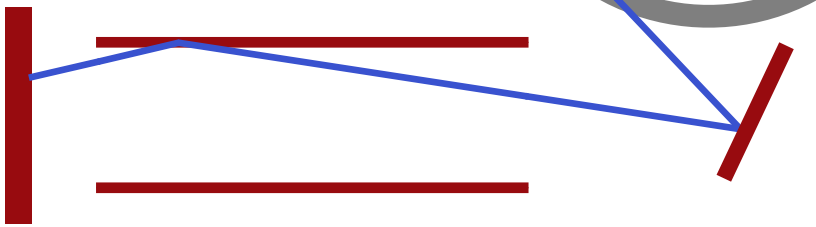


This project is funded by the European Union (GA no. 654000)

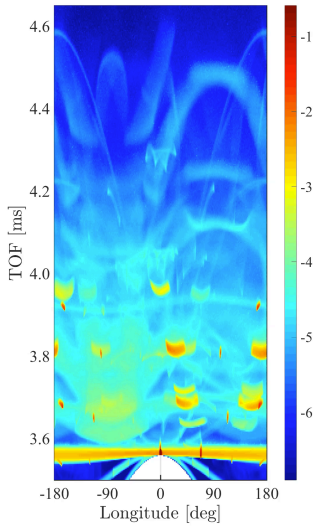
Recent development:
Union component set by Mads Bertelsen

- Allows to break McStas linearity around e.g. sample
- Decouple geometry and physics
- Model complex arrangements like e.g. sample environments
- **Student project summer 2019** to establish several of the most used sample envs directly in McStas (instr files)

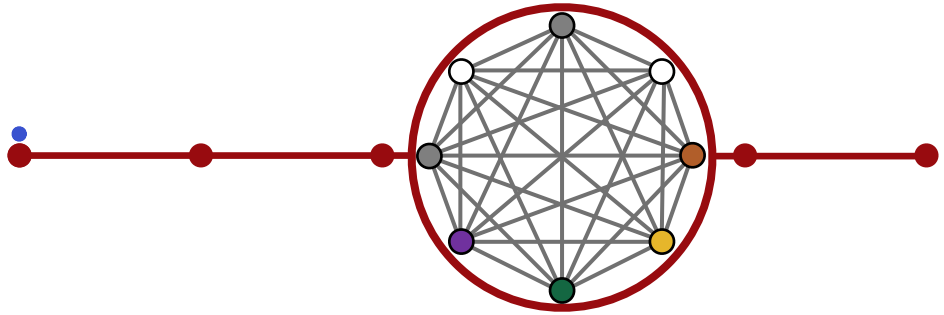




3D space of simulation



Instrument component sequence



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

Planned developments: Next releases, GPU support

- **2.6 is coming late summer / fall**
- Will try to have **3.0 beta at same time / shortly after**
- Main 3.0 wishlist item:
  - Modernised code-generator ⇨ **GPU**. Work started 2018 GPU hackathon in Dresden, reapplied to be part of 2019 GPU hackathon at CSCS.

McCode on GPU?

More to come in 2019!

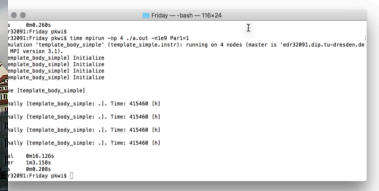
Port heavy part of component to GPU (Shows some potential)

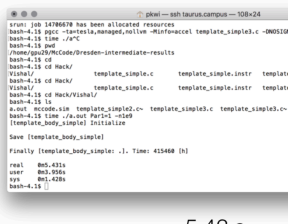
Port neutron loop to GPU
- Lots of code
- Code generation
- Compiles! Speed up today

McStas / McXtrace instrument

CPU MPI 4 cores (95 % usage)

GPU 5% usage





16.12 s (Single core 56.0 s)

5.43 s

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

Thanks

- Questions?

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