New developments in the McStas project

The long awaited 2.0 release...

Peter Willendrup[1]
Emmanuel Farhi[2]
Erik Knudsen[1]
Johan Sejr Brinch Nielsen[1]
Uwe Filges[3]
Kim Lefmann[4]

[1]Materials Research Division, Risø DTU, Roskilde, Denmark
[2]Calcul Scientifique, Institut Laue-Langevein (ILL), Grenoble, France
[3]Paul Scherrer Institut, Villigen, Switzerland
[4]Niels Bohr Institute, University of Copenhagen, Copenhagen, Denmark

McStas project http://www.mcstas.org

Agenda

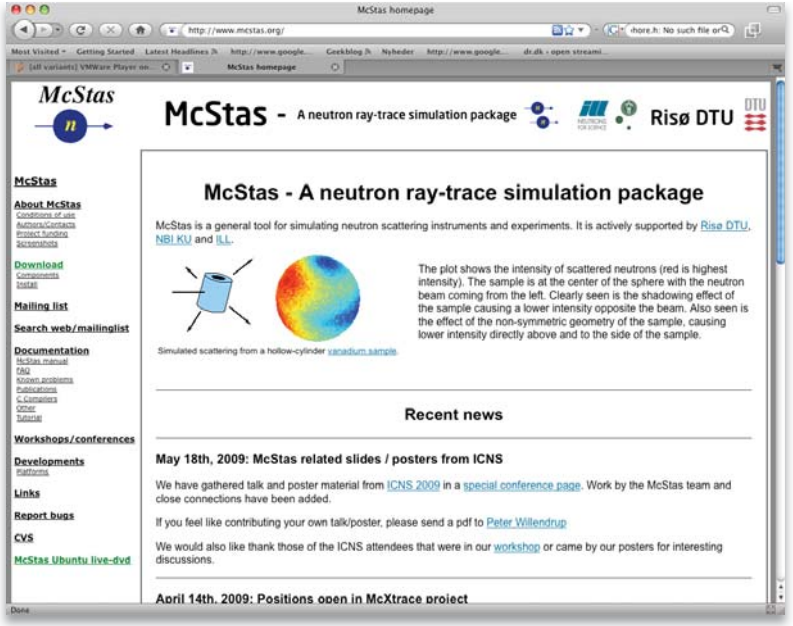
- Project, history, background (brief)
- What is new in the 2.0 release

Part I

- A brief introduction to McStas

McStas Introduction

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



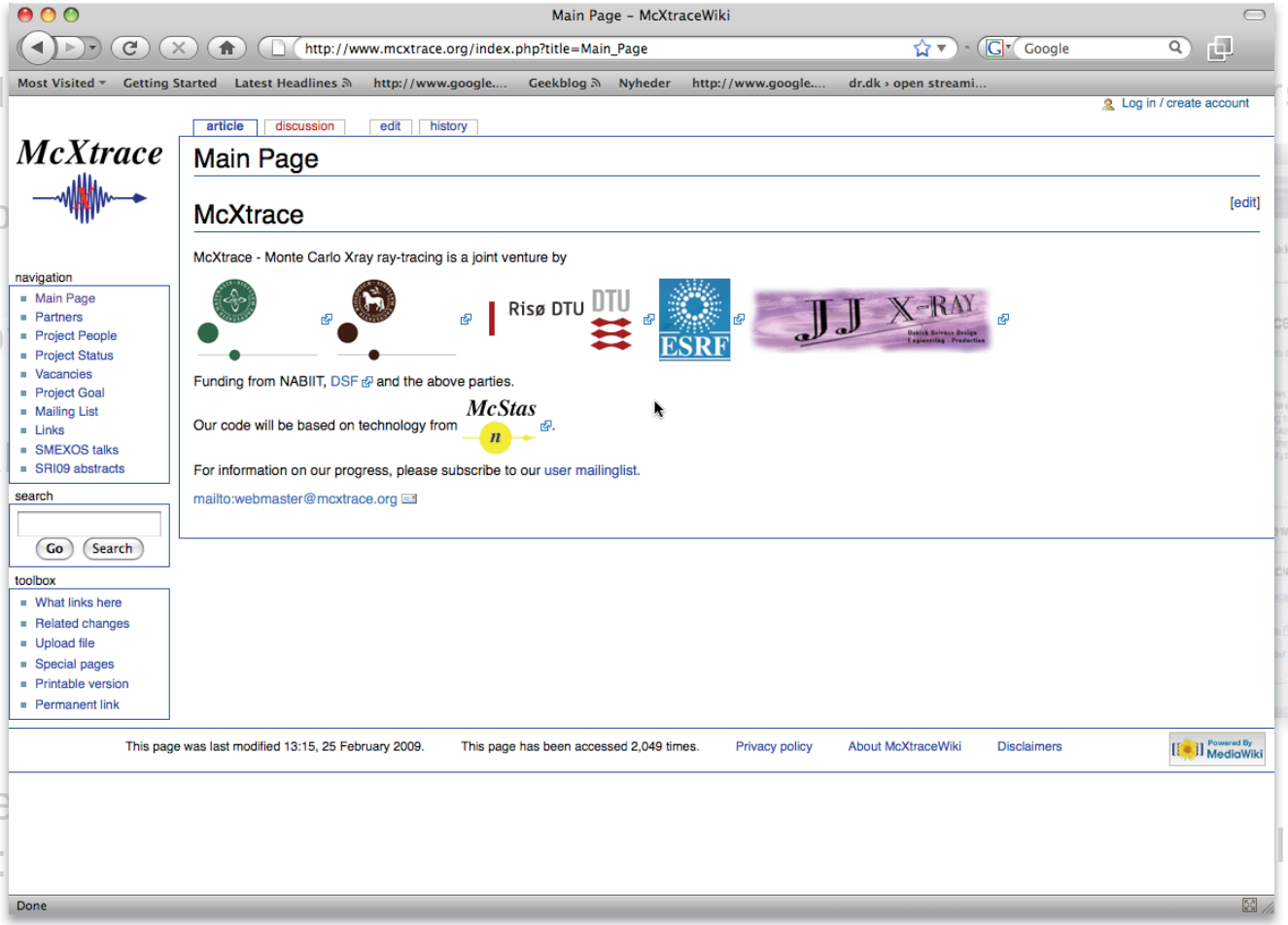
GNU GPL license
Open Source

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

mcstas-users@mcstas.org mailinglist

McXtrace - since jan 2009 similar in X-rays

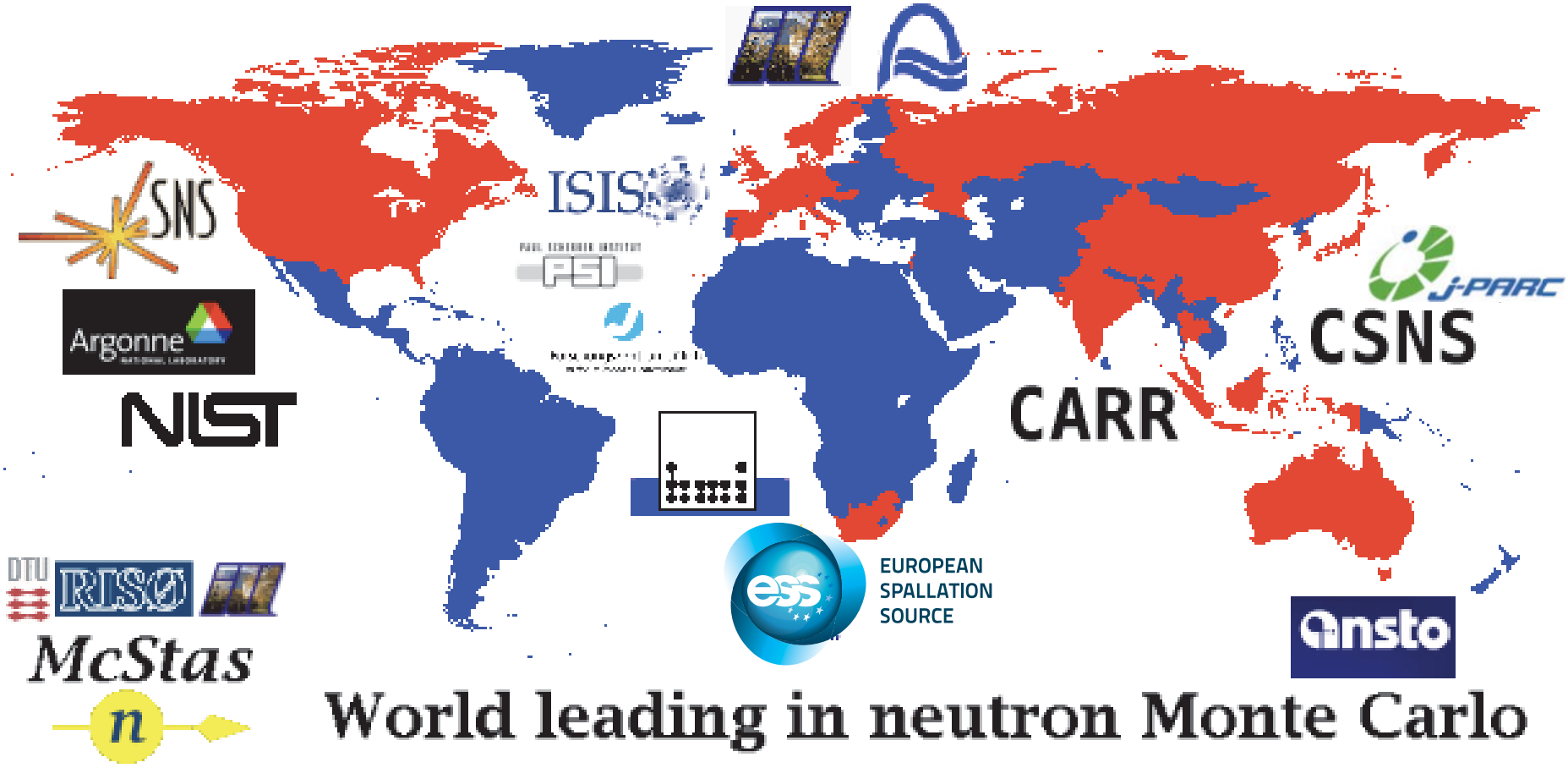
- Flexible, general simulation utility for neutron scattering experiments.
- Original
- Develop
- V. 1.0 b
- Current



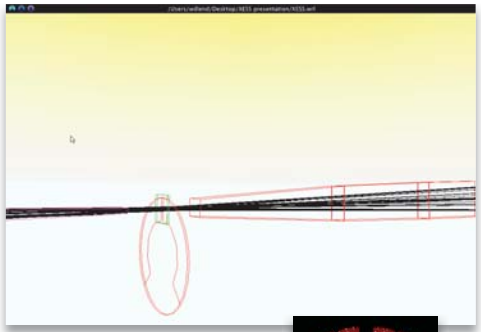
- Synergy, knowledge transfer, shared infrastructure

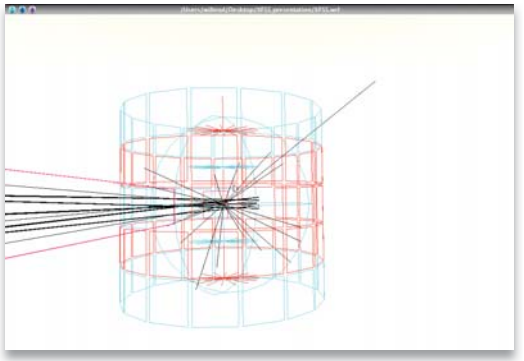
McStas Introduction

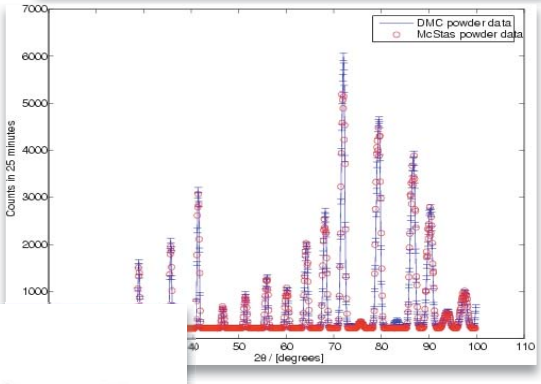
- Used at all major neutron sources (or instrumentation efforts)



What is McStas used for?

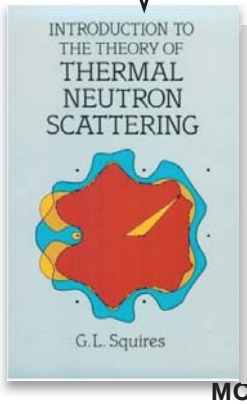


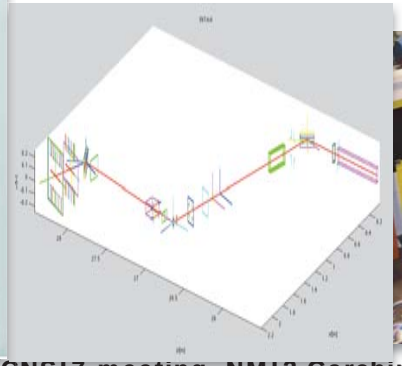


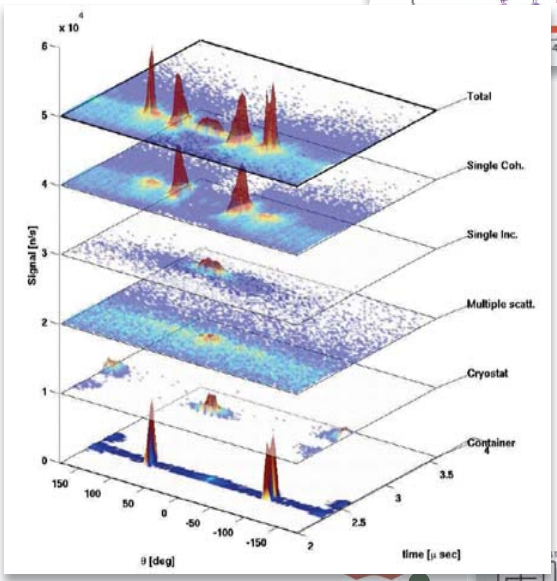


- Instrumentation
- Virtual experiments
- Data analysis
- Teaching

(KU 2005-2011)





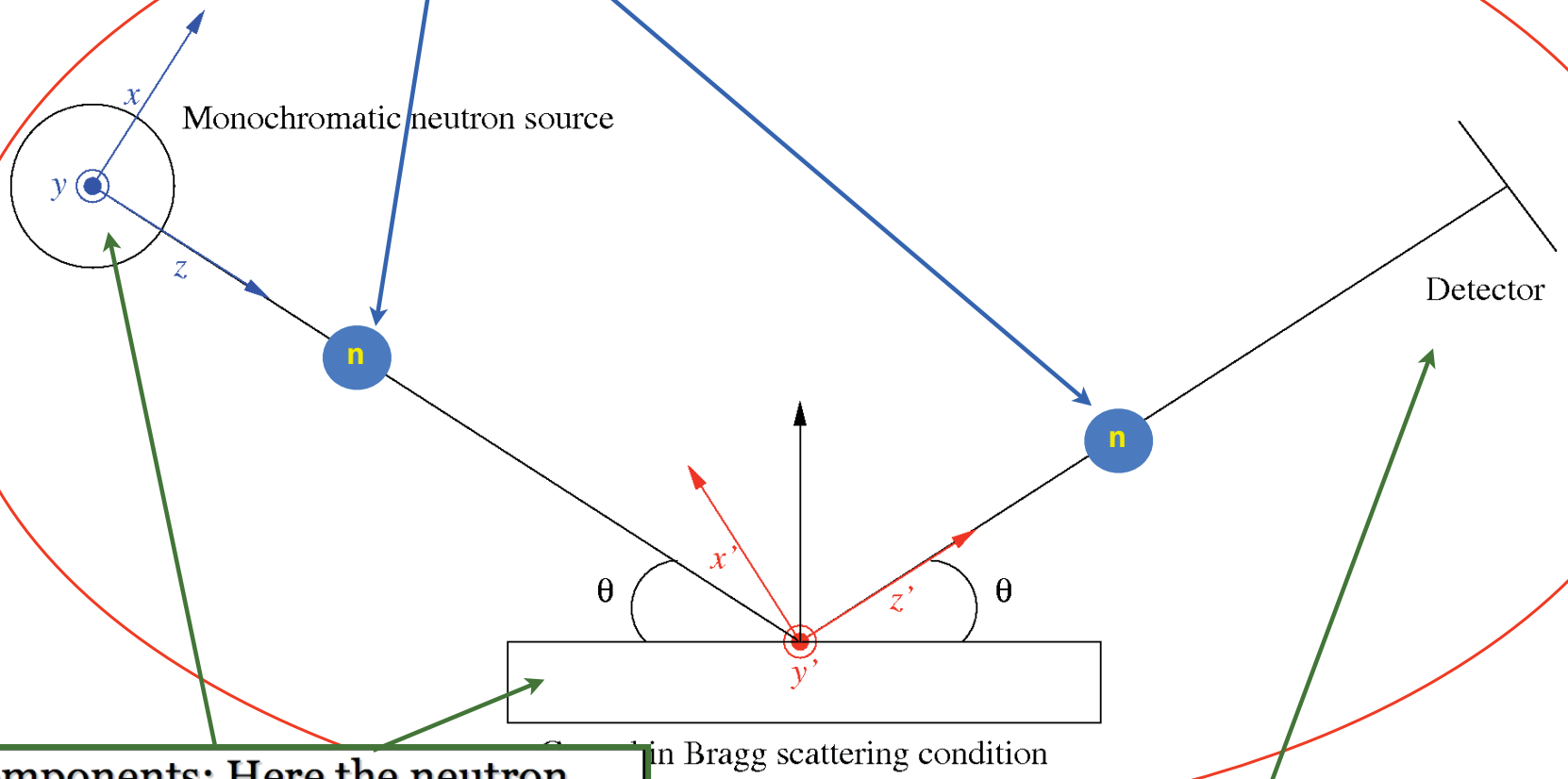


Neutron ray/package:

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

Time (t)

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

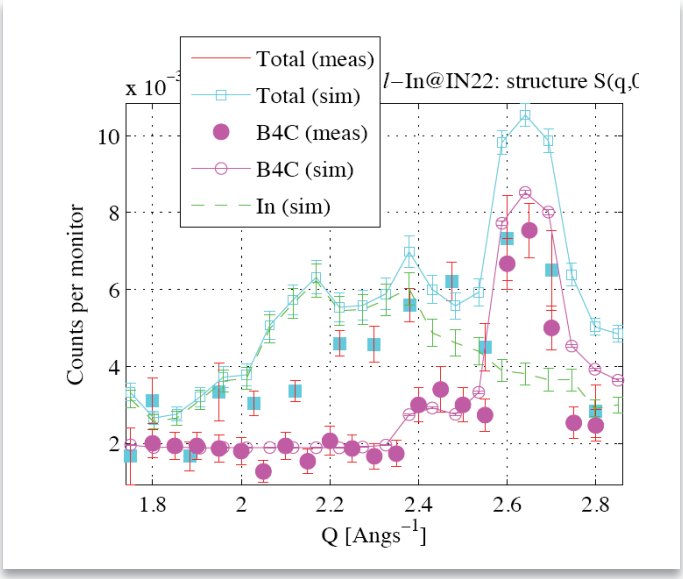


Components: Here the neutron physics happen, neutron weight adjusted according to scattering probabilities etc.

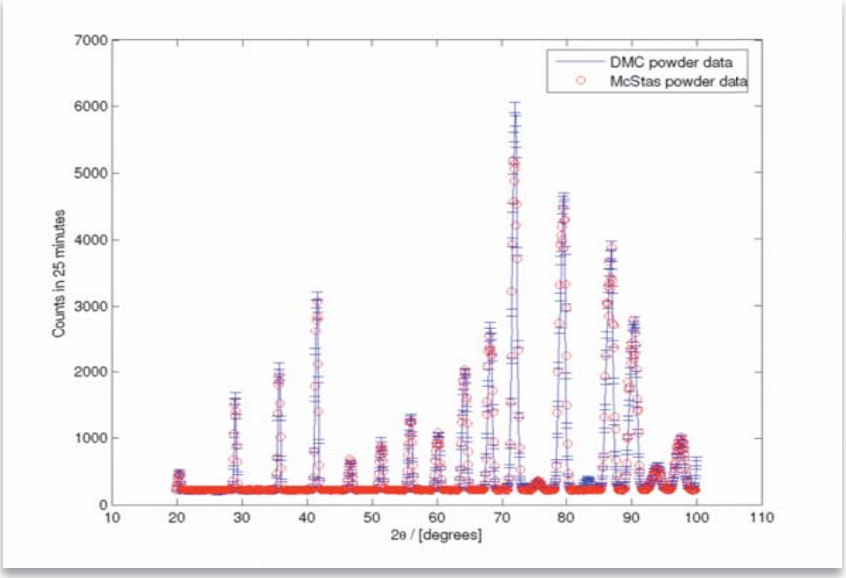
Local, internal coordinate system!

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 10-30%
- Common understanding: McStas is reliable



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



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

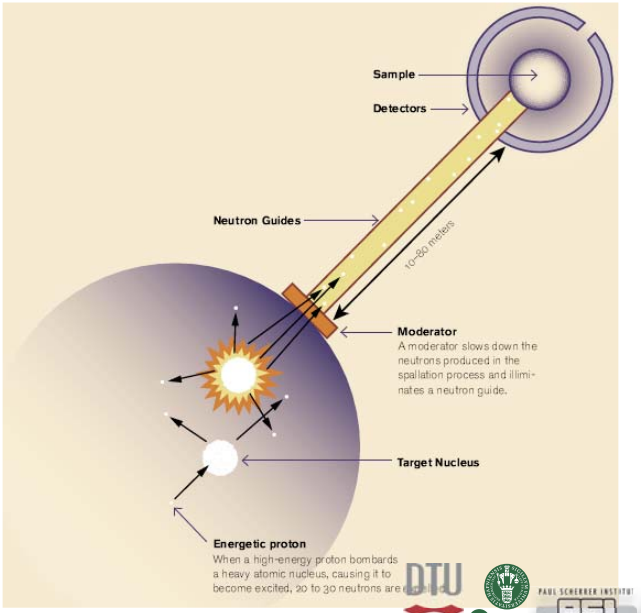
McStas overview

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

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

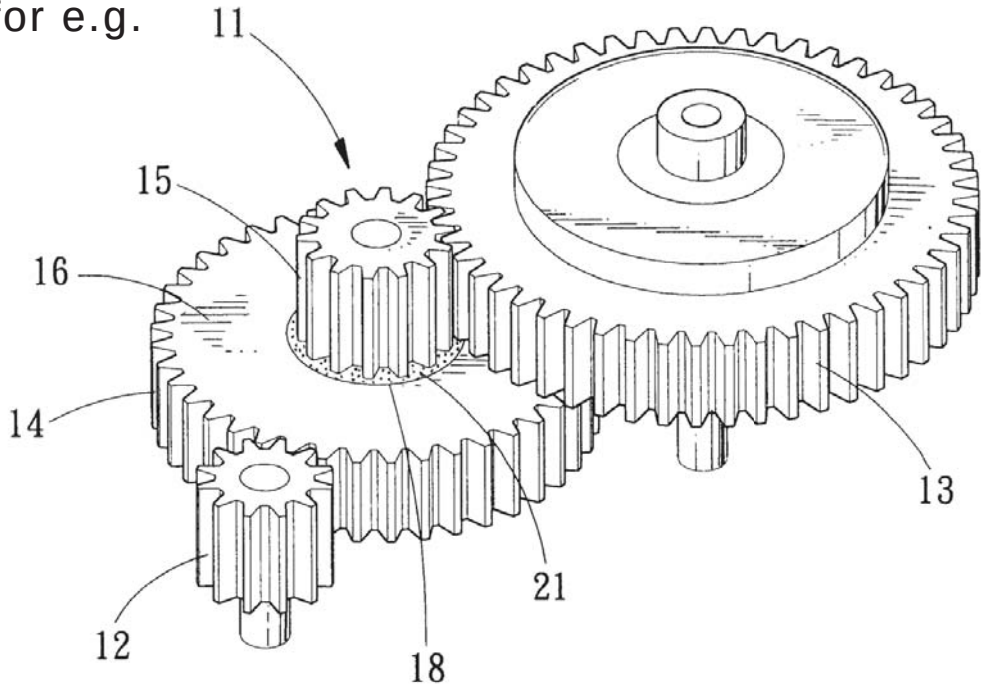
- 'Component' files (~100) inserted from library
  - Sources
  - Optics
  - Samples
  - Monitors
  - If needed, write your own comps

- 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
  - ...



Principles

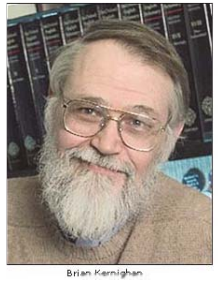
- Prefer minimum intrusion solutions
  - (has only proven semi-faulty approach in perl-layer)

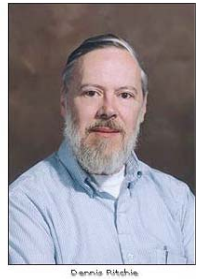
- If someone does it well, rather wrap/interface than re-implement
  - Material structures from CGRAPH / Lazy
  - Interface with e.g. MCNP

mcnp

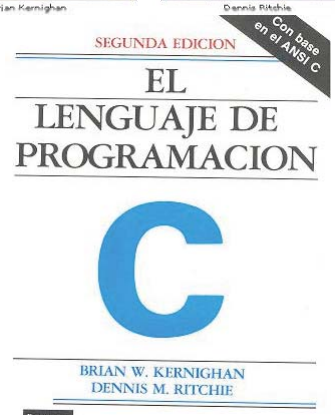
- Try to minimize external dependencies (libs etc.)
  - Basic number-crunching dependency for McStas is C!
  - Automated parameter variation requires perl
  - Gui layer requires perl-Tk
  - Graphics layer requires either
    - PGPLOT, perl-PGPLOT and PDL
    - GNUPLOT
    - Matlab
    - ...

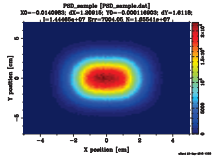
K & R

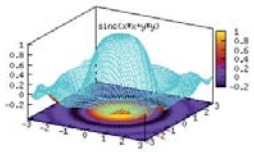


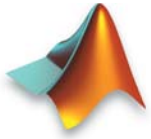


Brian Kernighan Dennis Ritchie









Including user contribs

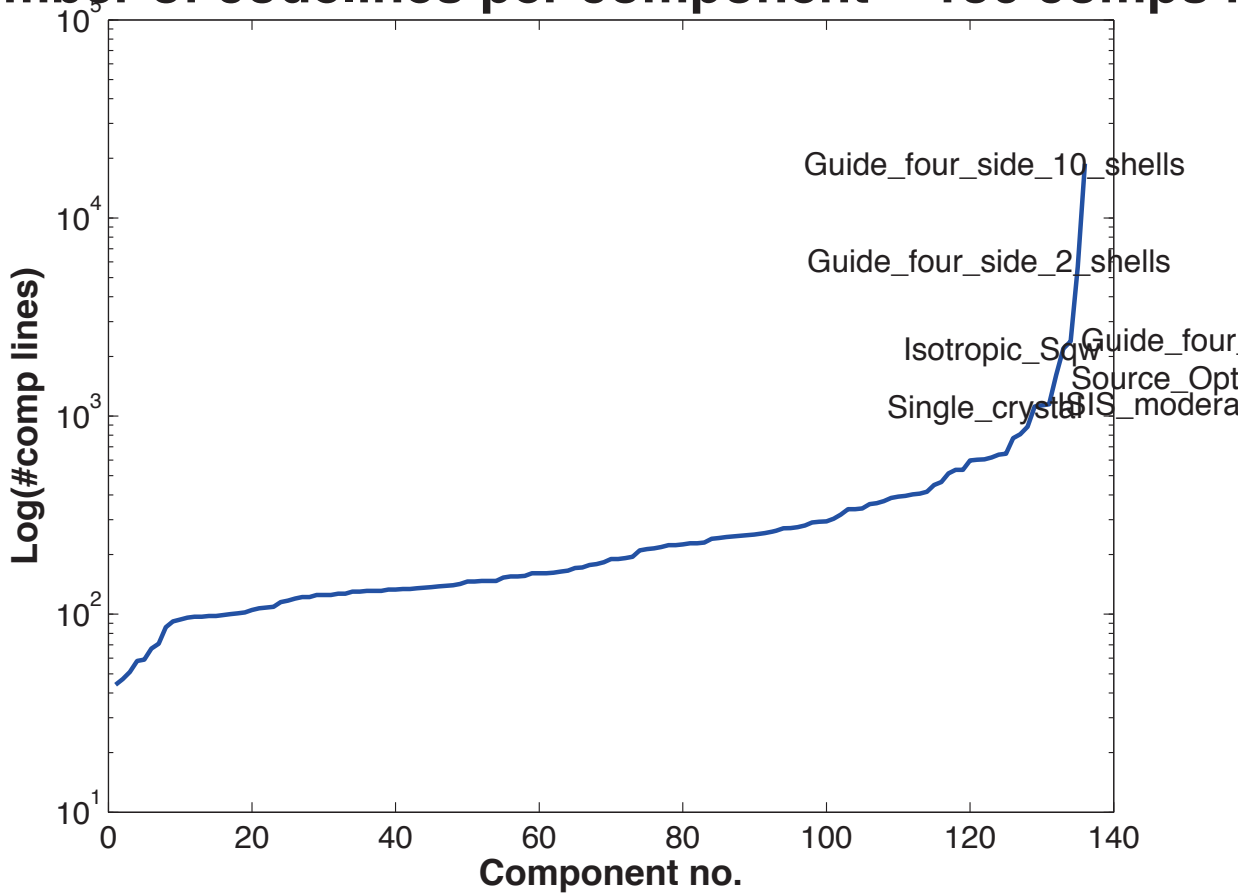
- Well-developed community support
  - **30-40% of existing and new additions are from users**
  - No direct refereeing of the code, but these requirements:
    - **At least one test-instrument**
    - **Meaningful documentation headers (in-code docs)**
  - Contributions go in dedicated contrib/ section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

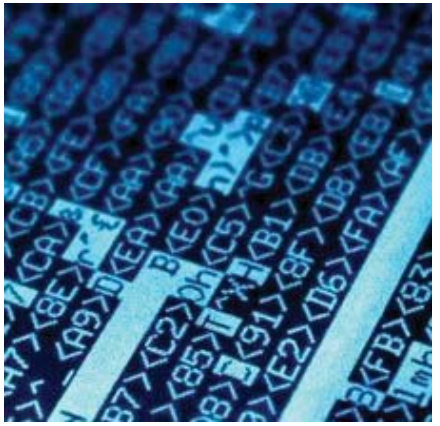
- Natural life-cycle of contrib's
  - Bug-fixes are applied both by contributor and developers
  - If contributor becomes unavailable either:
    - Many users of comp: Promote to official components, e.g. in optics/
    - Few/no users of comp: Move to obsolete/ until next major release

Writing / modifying comps is not complex...

- Most comps are quite simple and short ~ 100 lines
- 30-40% of existing and new additions are from users
- Requirement: Test-scenario for documentation & unit test

Number of codelines per component – 136 comps i





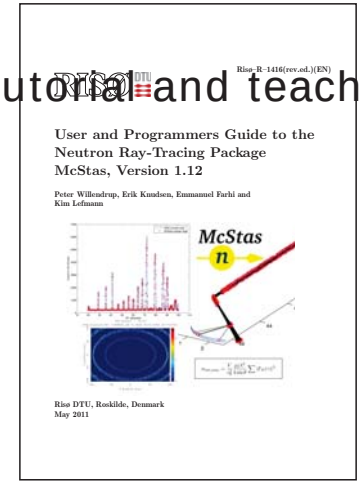
Documentation

- Basic use info is available inside comp & instr codes, extracte
  html

- 100+ page manuals documenting
  - Metalanguage
  - What is “under the hood”
  - Examples of practical use plus advanced features
  - Assumptions and algorithms applied in the components

- More than 70 example instruments

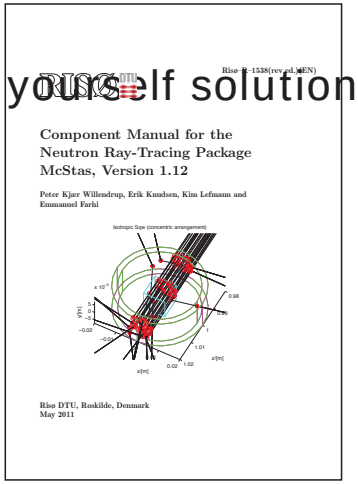
- Various tutorial and teach yourself solutions


Risø-R-1416(rev.ed.)(EN)

User and Programmers Guide to the Neutron Ray-Tracing Package McStas, Version 1.12

Peter Willendrup, Erik Knudsen, Emmanuel Farhi and Kim Lefmann

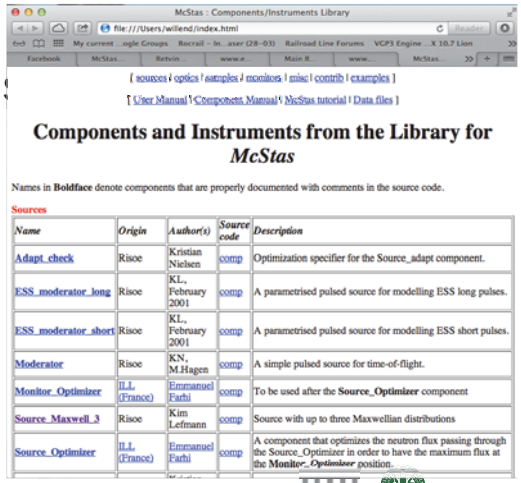
Risø DTU, Roskilde, Denmark
May 2011


Risø-R-1538(rev.ed.)(EN)

Component Manual for the Neutron Ray-Tracing Package McStas, Version 1.12

Peter Kjær Willendrup, Erik Knudsen, Kim Lefmann and Emmanuel Farhi

Risø DTU, Roskilde, Denmark
May 2011


McStas : Components/Instruments Library

[sources | optics | samples | monitors | misc | contrib | examples]

[User Manual | Components Manual | McStas tutorial | Data files]

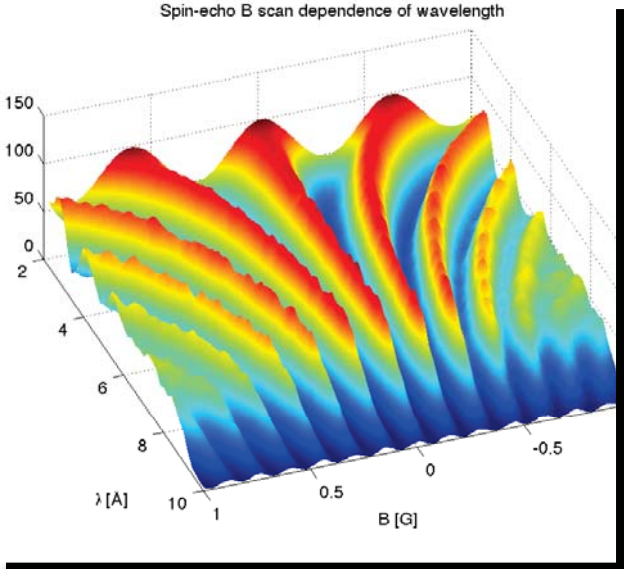
Components and Instruments from the Library for *McStas*

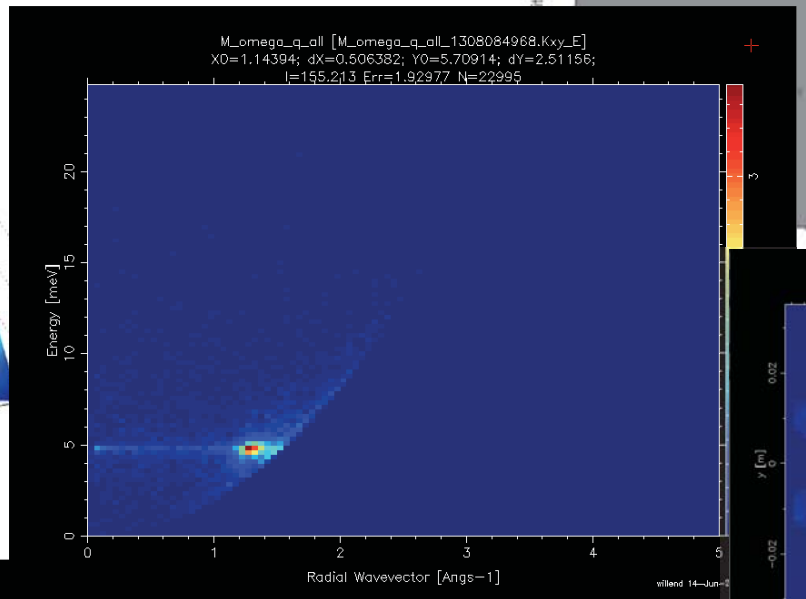
Names in **Boldface** denote components that are properly documented with comments in the source code.

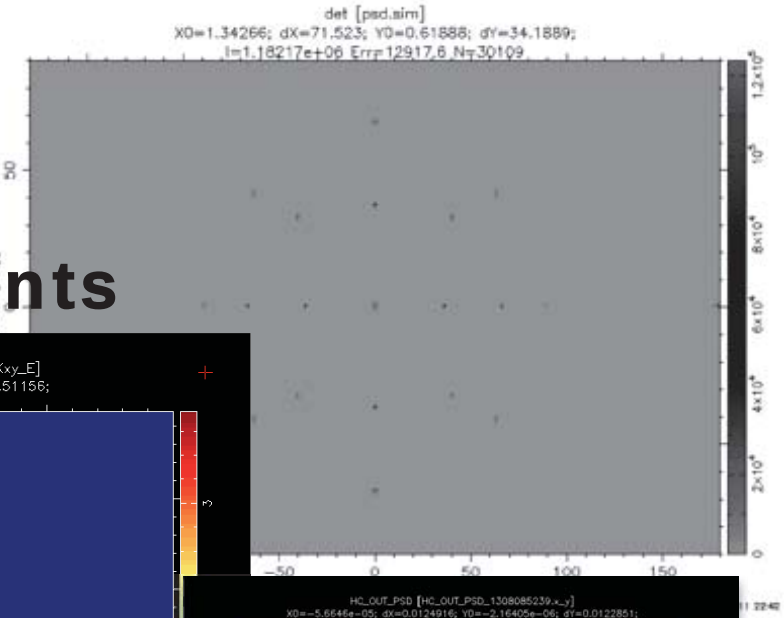
Sources

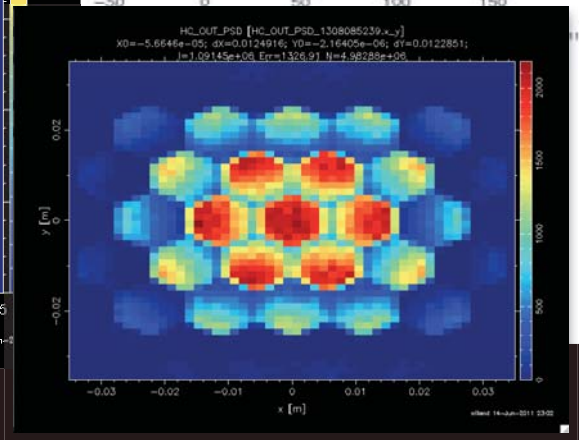
Name	*Origin*	*Author(s)*	*Source code*	*Description*
Adapt_check	Risoe	Kristian Nielsen	comp	Optimization specifier for the Source_adapt component.
ESS_moderator_long	Risoe	KL, February 2001	comp	A parametrised pulsed source for modelling ESS long pulses.
ESS_moderator_short	Risoe	KL, February 2001	comp	A parametrised pulsed source for modelling ESS short pulses.
Moderator	Risoe	KN, M.Hagen	comp	A simple pulsed source for time-of-flight.
Monitor_Optimizer	ILL (France)	Emmanuel Farhi	comp	To be used after the **Source_Optimizer** component
Source_Maxwell_3	Risoe	Kim Lefmann	comp	Source with up to three Maxwellian distributions
Source_Optimizer	ILL (France)	Emmanuel Farhi	comp	A component that optimizes the neutron flux passing through the Source_Optimizer in order to have the maximum flux at the **Monitor_Optimizer** position.

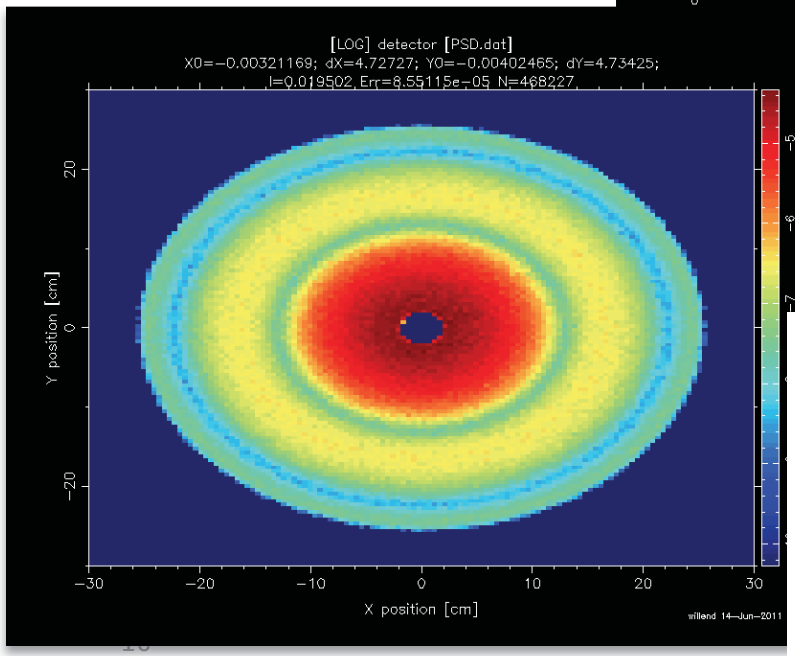
Example suite: 86 instruments

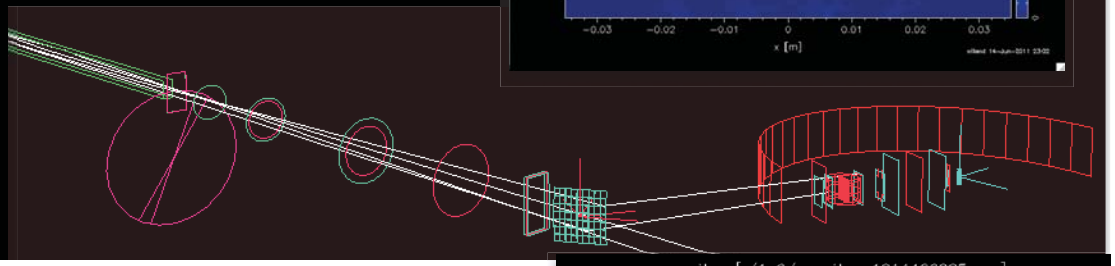


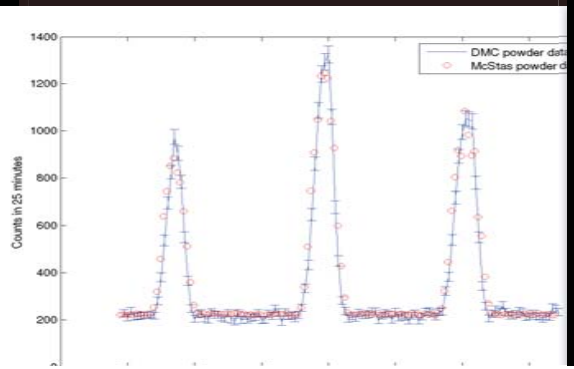


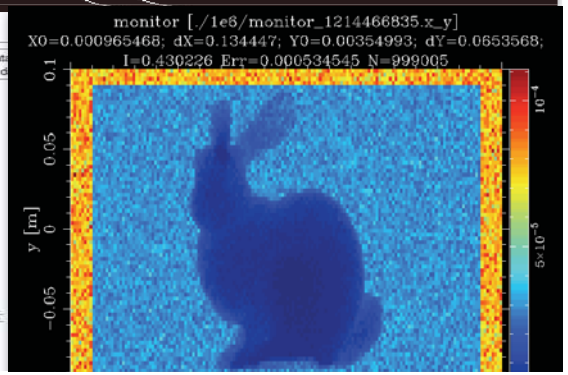










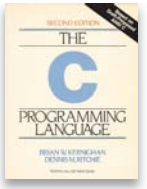


Part II

- A brief overview of the McStas 2.0 release (October 2012)

Future support tool situation

- Basic calc/sim functionality support still provided by C

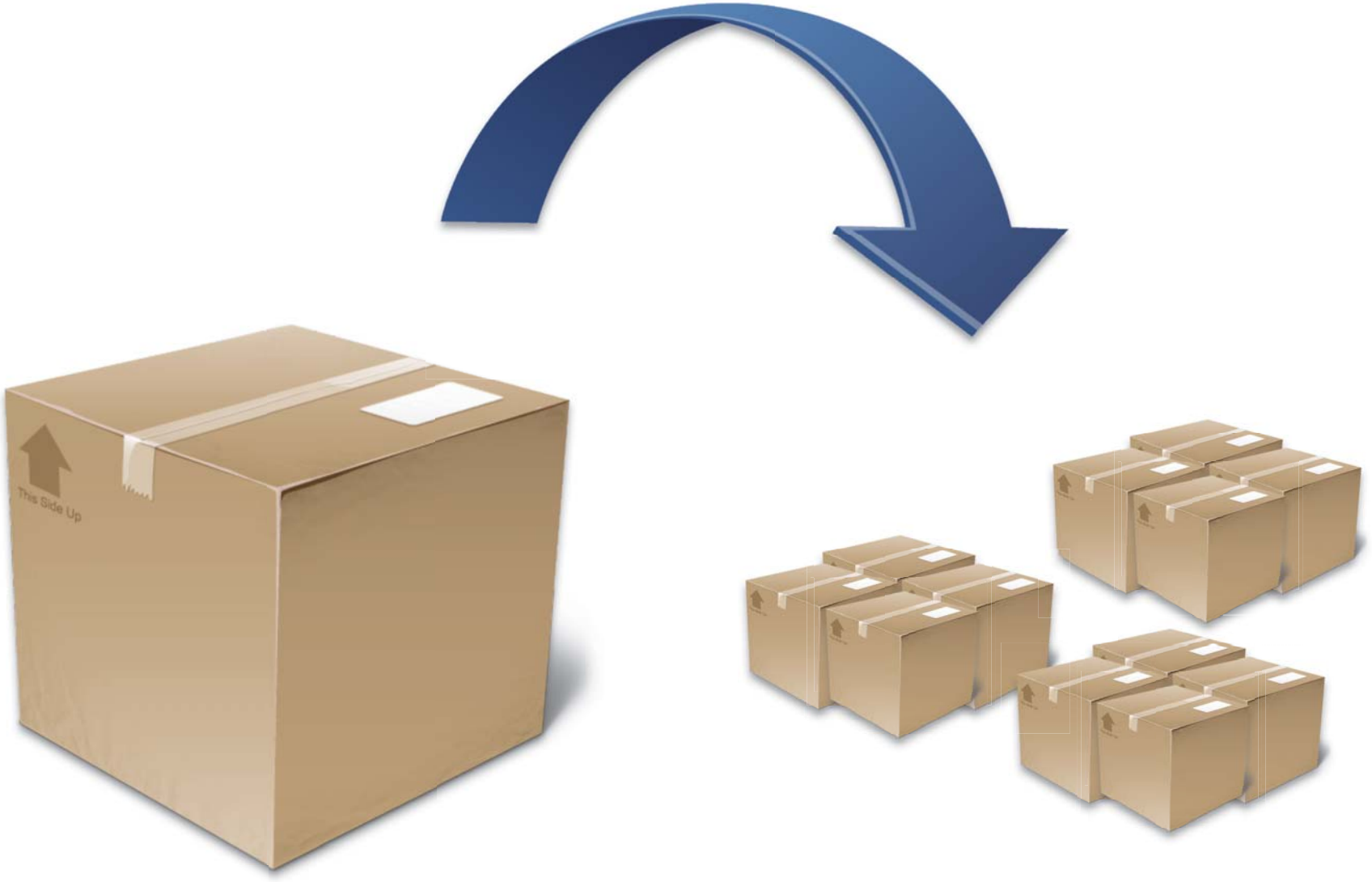


- Perl tools are there - but should be abandoned

- Python will provide future scripting and glue
- GUI widgets likely wxwidgets or Qt or standard editor??
- Plotting using matplotlib (also considering q'n'd GNUPLOT hooks)
- High-level support tools provided using e.g.

- **Johan Brinch done the first developments in this direction for McStas 2.0**

From one monolith to many smaller packages

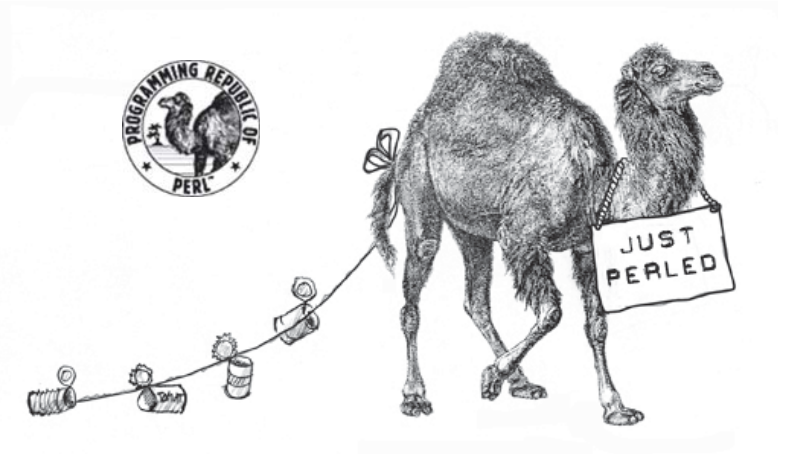


Central, physics implementation

- Core package:
  - mcstas-2.0
    - code generation
    - central low-level libraries
    - One dependency: a C-compiler

- Component/library package:
  - mcstas-comps-2.0
    - component libraries
    - example instruments
    - material definitions
    - One dependency: the core package

High level tools



- Perl package:
  - mcstas-tools-Perl-2.0
    - Slightly tuned mcgui
    - mcrun, mcplot, mcdisplay, mcplot
    - Interoparability with Python packages
    - Dependencies: mcstas-2.0, Perl, Perl::Tk, Perl::Tk::CodeText, Perl::PDL, pgplot & Perl::PGPLOT or Gnuplot

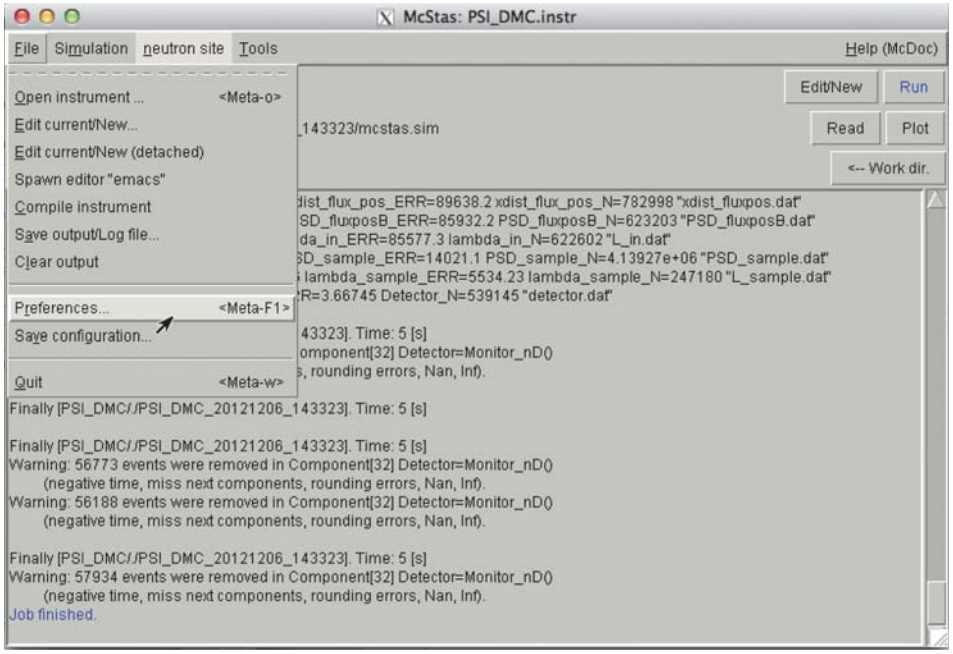
- Python packages:
  - mcstas-tools-Python-mcrun
    - New Python implementation, compatible with mcrun.pl syntax
    - Dependencies: mcstas-2.0, Python, Python-yaml
  - mcstas-tools-Python-mcplot-matplotlib
    - New Python implementation, similar in functionality to mcplot.pl
    - Dependencies: mcstas-2.0, Python, matplotlib
  - mcstas-tools-Python-mcplot-chaco
    - New Python implementation, similar in functionality to mcplot.pl
    - Dependencies: mcstas-2.0, Python, chaco

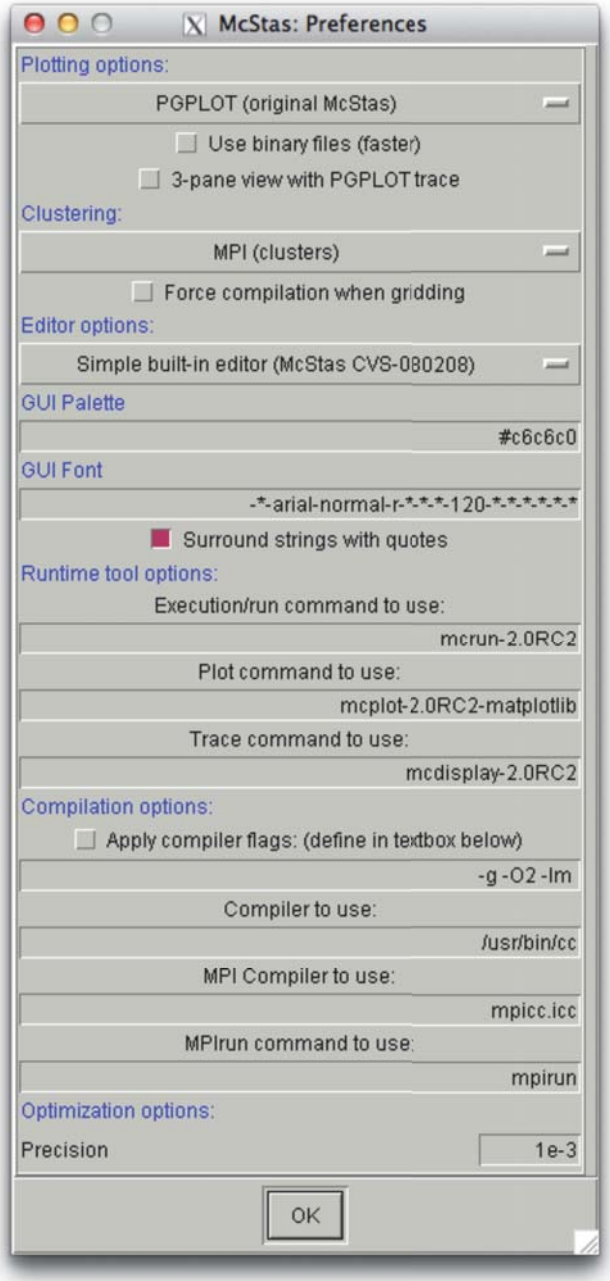
Modernized build/package system

- CMake+CPack replace ./configure && make && home grown scripts
- Supported platforms:

Tool highlights

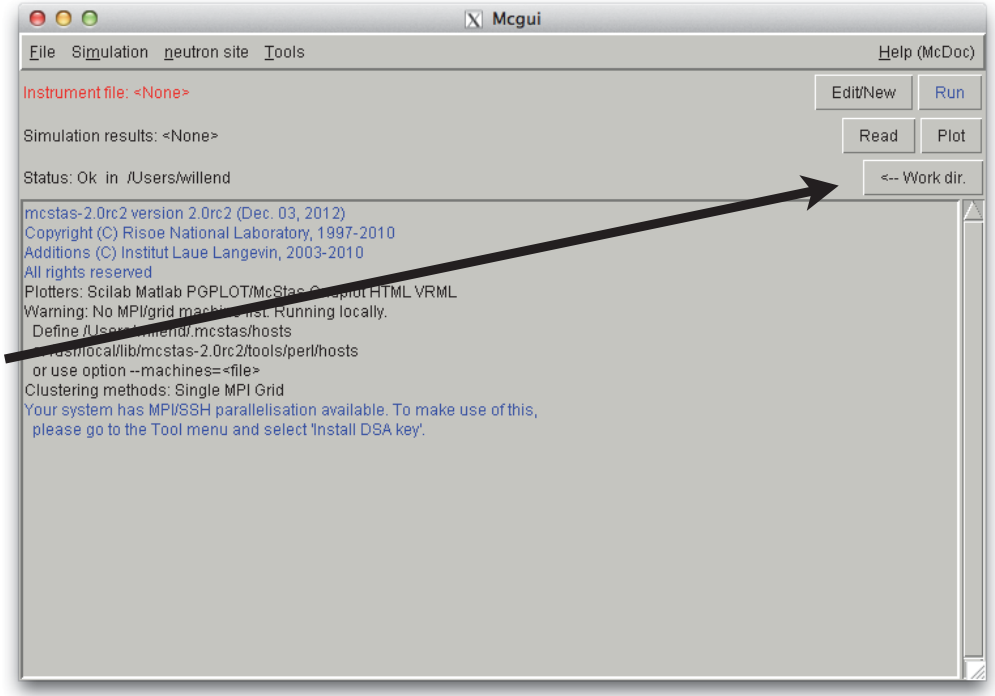
- Configuration of used tools etc. via ‘preferences’





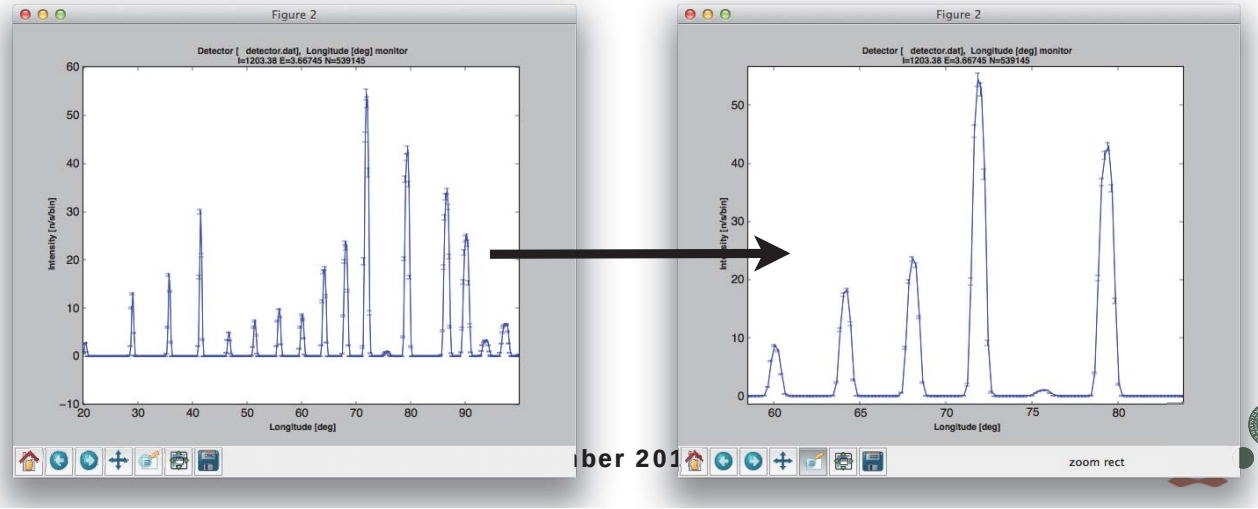
Tool highlights

- Workdir change in mcgui via button



- Output always go to directory, i.e. no accidental overwriting of results (if no dirname given e.g. PSI_DMC_20121204_133444)

- Interactive zooming in the Python mcplot tools



Core / runtime highlights

- **Backward compatibility 1:** Some backward-compatibility is lost in the new version of the code-generation, as we no longer use the information defined in the component STATE_PARAMETERS and POLARISATION_PARAMETERS macros. A warning is issued in case of components using these macros

- **Backward compatibility 2**: We have standardized naming of component parameters - see the file NOMENCLATURE installed in the McStas library. The code generation will give error messages of e.g. using xw and yh where one should now be using xwidth and yheight.

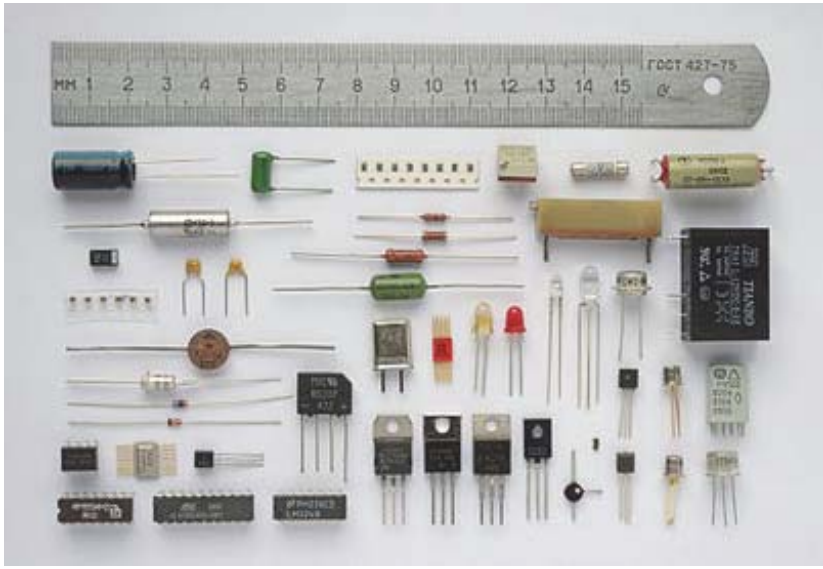
- **Polarization 1:** Support for tabulated magnetic fields. New code for handling fast interpolation in sparse data is included in the share/interpolation/ area. A new version of the SE_example.instr takes use of this feature.

- **Polarization 2:** We allow overlapping magnetic fields. All components that use the PROP_ routines for propagation implicitly allows larmor precession.

- **Future keyword:** ASSEMBLY allowing ‘metacomponents’

Component highlights

- - **Sample_nxs.comp** from Mirko Boin, HZB. Diffraction/imaging oriented sample where one defines the structure
- by means of the NXS crystallography library which is now also included with McStas. Comes with the test-
- instruments Test_Sample_nxs_diffraction.instr and Test_Sample_nxs_imaging.instr. NOTE: Special compilation
- required, see instrument source codes!
- - **Elliptic_guide_gravity.comp** from Henrik Carlsen, NBI. Analytical approach to describing an elliptical guide
- geometry where gravity is taken into account. Included in the Reflectometer.instr from Anette Vickery, NBI.
- - **Suite of SANS-samples** from Martin Cramer Pedersen, NBI. Various approaches to describing SANS diffraction
- e.g. using the PDB data bank. Component names are SANSShells.comp SANSPDBFast.comp SANSPDB.comp
- SANSNanodiscsWithTagsFast.comp SANSNanodiscsWithTags.comp SANSNanodiscsFast.comp SANSNanodiscs.comp
- SANSLiposomes.comp SANSEllipticCylinders.comp SANSCylinders.comp SANSCurve.comp SANSSpheres.comp. Included
- (some as comments) in the TestSANS.instr.
- - **SANSQMonitor.comp** also from Martin Cramer Pedersen, NBI. Q-monitor for SANS also included in TestSANS.instr.
- - **TOF2Q_cylPSD_monitor** from Anette Vickery, NBI. Time-of-flight vs. q monitor of cylindrical shape. Included
- in the Reflectometer.instr test instrument.
- - **SNS_source_analytic.comp** from Franz X. Gallmeier, SNS. Smooth-fit description of the SNS-moderators. Fits are
- derived from the same underlying raw-data as the ordinary SNS_source.comp is using directly with linear
- interpolation.
- - **Brilliance_monitor.comp** from Peter Willendrup, DTU Fysik. Monitor for easy evaluation of mean and instantaneous
- source brilliance for source comparison. Used in the ESS_brilliance.instr test instrument.
- - **TOF_PSD_monitor_rad.comp** from Kim Lefmann, KU. Derived from PSD_monitor_rad by Henrich Frielinghaus, FZJ.
- Position-sensitive TOF monitor with radially averaging.
- - **PSD_TOF_monitor.comp** from Peter Willendrup, DTU Fysik. PSD-monitor with a number of independent time-slices.
- Derived from PSD_monitor by Kim Lefmann, KU.
- - **Source_gen4.comp** from Jonas O Birk, NBI and Uwe Filges PSI. Version of source_gen with PSI-specific changes,
- e.g. a high-energy tail contribution. Included in the RITA-II.instr from Linda Udby, NBI.
- - **Absorber.comp** from Peter Willendrup, DTU Fysik. Slab of perfectly absorbing material. Included in the RITA-II.instr
- from Linda Udby, NBI.
- - **PSD_monitor_psf.comp** and PSD_monitor_psf_eff.comp from Kim Lefmann and Linda Udby, KU. Two derivatives of
- PSD_monitor.comp both with gaussian point-spread-function and _eff with a 1/k efficiency parameter. Included
- in the RITA-II.instr from Linda Udby, NBI.
- - **Virtual_mcnp_ss_input.comp** and Virtual_mcnp_ss_output.comp from Esben Klinkby, DTU Nutech. Allows to read and write
- MCNP/MCNPX 'source surfaces'. For use in simulations where neutrons need transport in both codes.
- - **Virtual_mcnp_ss_Guide.comp** from Esben Klinkby, DTU Nutech. Single guide piece sitting in a 'sandwich' between
- an input and an output MCNP/X source surface.
- - **ESS_moderator_long.comp** patches from Kim Lefmann KU (multiple-pulses, TOF-focusing) and Esben Klinkby DTU Nutech
- (geometry and spectrum from ESS MCNPX models). Thanks to Klaus Lieutenant from Vitess/HZB for providing adjusted
- parameters for the 'Mezei moderator' and a wavelength-dependent corretion term.
- - Many other components received updates...
-



Instrument highlights

- - HZB_NEAT.instr from Emmanuel Farhi, ILL - written in collaboration with Ruep E Lechner. The NEAT spectrometer from the HZB.
- - TestSANS.instr from Martin Cramer Pedersen, NBI. Test SANS setup for the new SANS components described above.
- - Reflectometer.instr from Anette Vickery, NBI. Reflectometer on pulsed source. Test instrument for a couple of components from NBI contributors decribed above.
- - Test_Sample_nxs_imaging.instr and Test_Sample_nxs_diffraction.instr from Mirko Boin, HZB. Test instruments for the use of Sample_nxs.comp. NOTE: Special compilation required, see instrument source codes!
- - RITA-II.instr from Linda Udby, NBI. Very detailed and complete, experimentally benchmarked description of the RITA-II spectrometer at PSI. Use the RITA-II_test.sh script for test-scans.
- - Test_SSR_SSW.instr and Test_SSR_SSW_Guide.instr from Esben Klinkby DTU Nutech. Test instruments for the above mentioned components.
- - ESS_bispectral.instr from Henrik Jacobsen, NBI. Test setup of ESS source and bispectral extraction

Other developments 1 - work on McStas-MCNPX interfaces

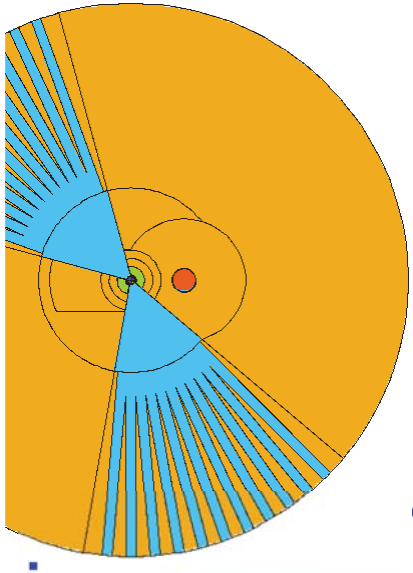
EUROPEAN SPALLATION SOURCE

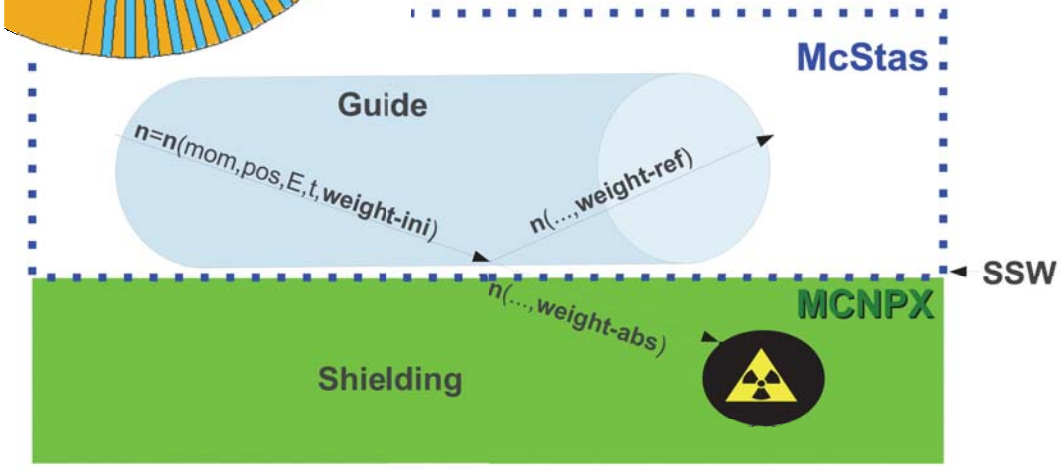
DTU

PAUL SCHERRER INSTITUT PSI

The task:

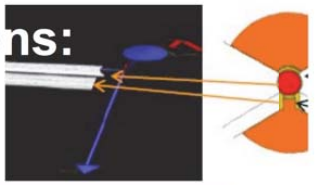
"Interfacing the MCNP and McStas Monte Carlo codes for improved optimization of the ESS moderator-beam extraction systems"





The solutions:

- Tally



- Ptrac

```
 3000      2      10     179
100        2       0
  0.00000E+00  0.28640E+00
0.43531E+00 -0.10000E+01
0.00000E+00  0.00000E+00
0.10000E+00  0.10000E+01
0.33356E-02
```

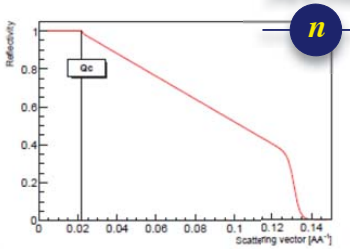
MCNPX

SSR SSW

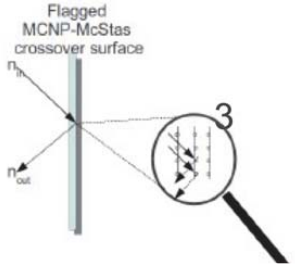
- SSW

McStas

- Supermirror



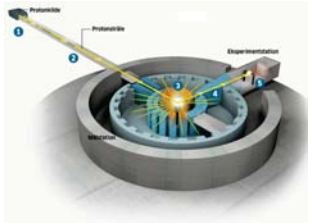
- Compile



DTU PAUL SCHERRER INSTITUT PSI NEUTRONS FOR SCIENCE McStas

Other developments 2 - New Python-based tools and web frontend

See J. Brinch poster



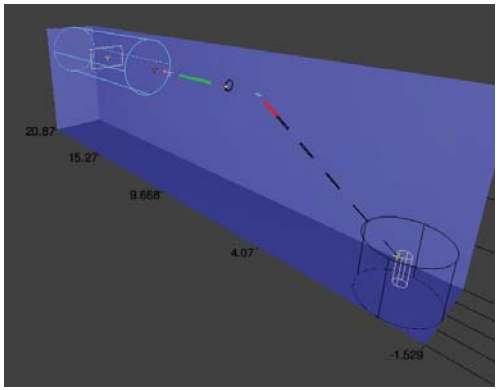
Neutron simulations from the web

Johan Brinch[1], Peter Willendrup[1], Emmanuel Farhi[2]
1. Physics Department, Technical University of Denmark; {JSBN,PKWI}@FYSIK.DTU.DK
2. Institut Laue-Langevin, Grenoble, France

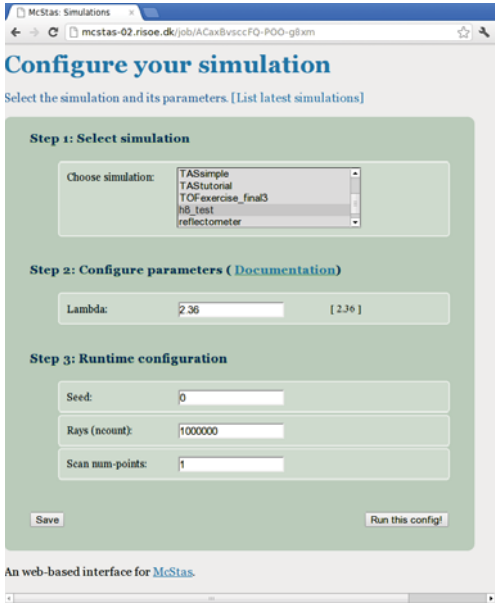
Overview

McStas is developing a web-interface for the simulation software, based on HTML and Javascript. The web-interface allows a user to run the McStas simulation software from a server; without installing it herself.

3D Visualisation



Example: Configure



Features

Permanent URLs: Both the URLs for configuration and results are unique and permanent; they can easily be shared with fellow students or colleagues.

Parameter verification: A sanity check eliminates basic mistakes before running the simulation. Most helpful for students.

Authentication: A simple authentication measure prevents anonymous users from using the system.

Limits: Upper limits can be set for the neutron count to prevent users from running very long simulations.

Parallelism: Simulations can be run in parallel; both with many workers and on a per simulation basis using MPI.

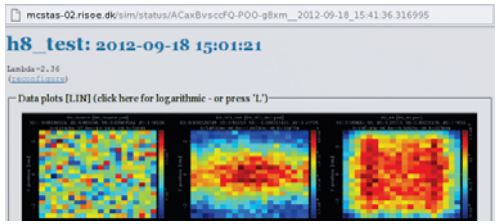
1. Configure

First, the instrument to simulate is chosen and configured. Parameters are adjusted and saved before the simulation is started.

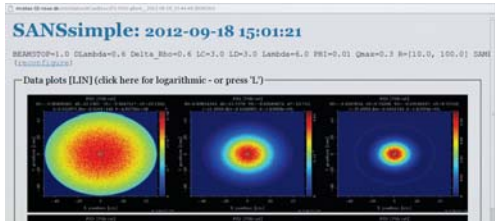
2. Simulate

Once the simulation has been configured, it

Example: Result overview



Scanning a parameter



Other developments 3. Guide losses...

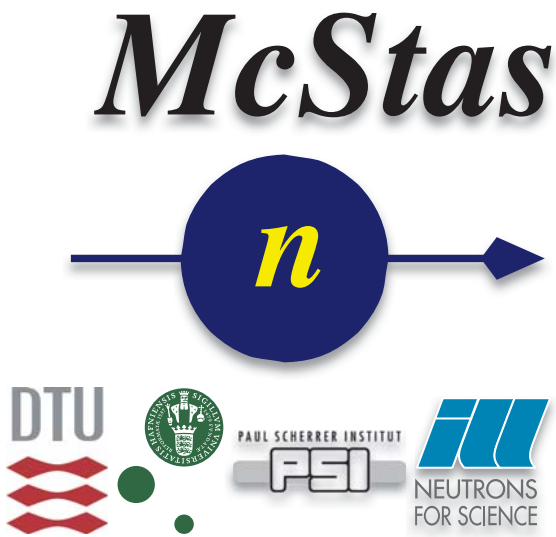
- Easy to use method using a set of components...
- re-“Looping” part of the guide

Other stuff

- Interface-code for the iFit data analysis package http://ifit.mccode.org
- Interface-code for the Mantid analysis package in the making (Post Doc hired at ESS DMSC) Mantid
- Database of source brilliances plus component parameters (Common with Vitess) currently: ILL, ISIS, ESS, FRM-II, HZB, SNS in prep VITESS
- **2.0RC2 version fully functional! Please go ahead and try it**
- **http://mcstas-20rc2.mcstas.org**

Conclusions

- McStas is a stable and mature code
- 2.0RC1 and RC2 have been released, final 2.0 is in preparation
- McStas has a very complete suite of components and instrument examples, many nice contributions from our users
- If you feel like test-driving 2.0, please go and get it (beta-status) from http://mcstas-20rc2.mcstas.org - all functionality is there



People

- The success of the project is also about the people:
  - Present McStas team members

- K Lefmann

E Farhi

P Willendrup

E Knudsen

U Filges

J Brinch

  - Past McStas team members

- K Nielsen

PO Åstrand

K Lieutenant

P Christiansen