Technical University of Denmark

DTU

New developments in the McStas neutron instrument simulation package

Willendrup, Peter Kjær; Bergbäck Knudsen, Erik; Klinkby, Esben Bryndt; Nielsen, T.; Farhi, E.; Filges, U.; Lefmann, K.

DTU Library
Technical Information Center of Denmark

New developments in the McStas neutron instrument simulation package

New developments in the McStas neutron instrument simulation package

PK Willendrup[1,7]**, EB Knudsen**[1]**, E Klinkby**[2,7]**, T Nielsen**[3,6,7]**, E Farhi**[4]**, U Filges**[5,8]**, K Lefmann**[6,7]

[1]NEXMAP, Physics Department, Technical University of Denmark, Denmark
[2]Center for Nuclear Technologies, Technical University of Denmark, Denmark
[3]ESS-DMSC, Copenhagen, Denmark
[4]Institut Laue-Langevin (ILL), France
[5]Paul Scherrer Institute (PSI), Switzerland
[6]Niels Bohr Institute (NBI), University of Copenhagen, Denmark
[7]Danish in-kind contribution toward the ESS
[8]Swiss in-kind contribution toward the ESS

E-mail: `pkwi@fysik.dtu.dk`

Abstract.

The McStas neutron ray-tracing software package is a versatile tool for building accurate simulators of neutron scattering instruments at reactors, short- and long-pulsed spallation sources such as the European Spallation Source. McStas is extensively used for design and optimization of instruments, virtual experiments, data analysis and user training. McStas was founded as a scientific, open-source collaborative code in 1997.

This contribution presents the project at its current state and gives an overview of the main new developments in McStas 2.0 (December 2012) and McStas 2.1 (expected fall 2013), including many new components, component parameter uniformisation, partial loss of backward compatibility, updated source brilliance descriptions, developments toward new tools and user interfaces, web interfaces and a new method for estimating beam losses and background from neutron optics.

1. Introduction

Since wide-spread development of neutron Monte Carlo ray-tracing techniques began in the late 1990s, the technique has supplemented analytic approaches when designing neutron scattering instruments. Before the arrival of todays easy-to-use codes, the instrument scientist would often develop the simulation software from scratch, thereby effectively duplicating existing work. In Monte Carlo ray-tracing, neutrons are transported through space and matter via an efficient and powerful combination of techniques: When complicated physical processes with a stochastic nature occur, Monte Carlo sampling from e.g. source spectra is used, but when deterministic propagation through space is sufficient, ray-tracing on parabolic paths in the gravitational field is applied.

At the RISØ national lab in Denmark, the RITA[1] instrument project was developing during the mid-to-late 1990s. In 1995, the NISP[2] package from Los Alamos National Laboratory was the first and only generally available Monte Carlo tool for neutron instrument design. Initially, NISP was considered for the instrument design of RITA, but was not found optimal

for this project, hence an other solution was found. The RISØ department head at that time, Kurt Clausen, decided to seek funding for what would eventually become the McStas[3]-[5] project. From the beginning the idea was to create an open-source collaboration platform where instrument scientists could build on simulation work previously done by others. A collaboration was started between computer scientist Kristian Nielsen and Kim Lefmann, Henrik Rønnow and several other physicists at the RISØ Physics Department during 1997, and the public McStas 1.0 version was released in 1998. In 1997 and 2000 two other simulation codes, RESTRAX[6] and VITESS[7] became available, later joined by the IDEAS[8] code from Oak Ridge National Laboratory. In the years that followed, several collaborative comparison projects proved the value of having a suite of simulation packages available, many bugs were found and corrected, and a situation of friendly competition provided inspiration for new features for all of the packages. Robert McGreevy who was at the time leading the SCANS work package on simulation was very actively supporting the instrument simulation packages in general, but specifically McStas due to its collaborative nature, as described below.

Due to needs for its modernisation projects, the Institut Laue-Langevin (ILL) became a very important McStas partner site, supporting core development of the packages. Later the efforts were also joined by the University of Copenhagen and the Paul Scherrer Institute (PSI).

One significant difference between McStas and the other projects always was its open-source, collaborative nature designed to use component codes written by the individual instrument scientist, later to be shared via the McStas component library. In McStas these small and medium sized component-codes implement the individual parts of a beamline. McStas users need not write components themselves, but always have the option to do so when a component is missing or needs improvement.

For the user operating McStas only knowledge of a simple instrument file containing a logical and easy to read meta-language is needed. The McStas software suite then translates the instrument file into a computation-effective and portable c-code using powerful code-generation and compiler techniques.

One other area where McStas historically had a strong point in regard to the other codes, was the availability of *sample* components, allowing the user to perform a Virtual Experiment[9], where one combines sample and instrument response to give a close-to-physical virtual dataset as output.

Today McStas has an impressive suite of components in all categories, counting 148 pieces of code, out of which around 40% are user contributions. Further, 100 example instruments gives the new user rich inspiration for her or his own projects.

2. Structure and use of the code
McStas was designed with flexibility and modularity in mind, and can be thought of as a framework for neutron instrument simulation.

Structurally, McStas consists of several *layers* and types of code, as illustrated in Figure 1, where the flow of defining and running a simulation should be read from top to bottom. In the figure, solid boxes contain elements of the code that are typically manipulated by the user, and dashed boxes contain elements of the system parts of McStas.

- The user defines an instrument using the mcgui program on the "tool layer", thereby assembling a so-called instrument file containing a logical and easy to read meta-language, positioning components of the beamline in a coordinate frame. The lab- or instrument coordinate system is right-handed, with y defining the vertical axis and z the direction of the neutron beam.

- Each of the components correspond to a McStas component file in the component library, written in a type of structured C-code. A user does not need to write component files, but
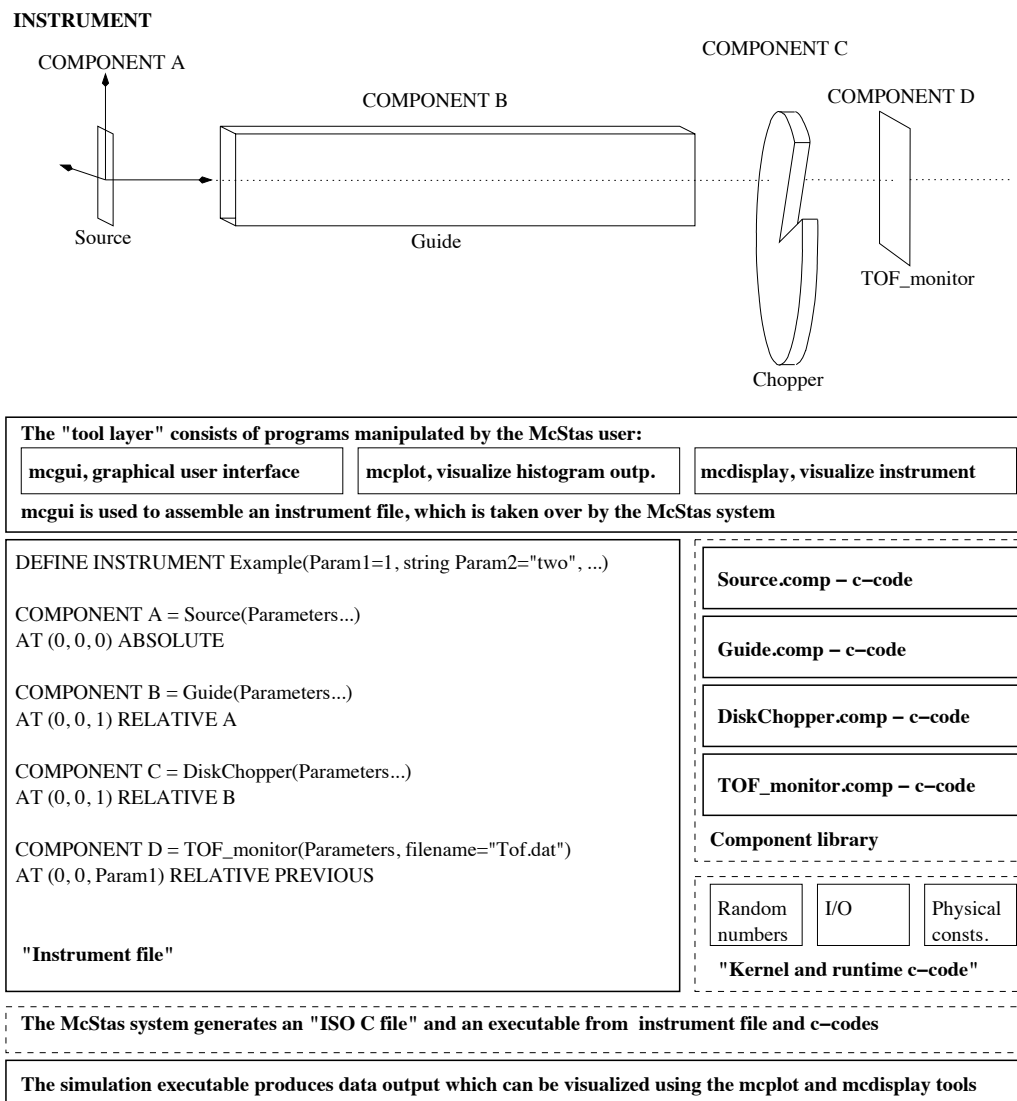
Figure 1. Illustration of a simulation workflow in McStas, to be read from top to bottom. Solid boxes contain elements of the code that are typically manipulated by the user, and dashed boxes contain elements of the system parts of McStas. See the text for details.

has the option to do so. As explained in the introduction, components are often contributed to the development team from the world-wide McStas user community. The components each have their local coordinate frame in which the neutron should be propagated between an *incoming* and *outging* state, e.g. before and after a scattering process. The McStas system takes care of transformation of the neutron state from one component coordinate system to the next.

- From the instrument code, the component codes and special kernel/runtime code containing e.g. a random number generator, propagation routines and physical constants, a computation-effective and portable c-code is generated by the McStas system using a powerful code-generation and technique. Further, a c-compiler is called, generating a binary executable, typically referred to as the *simulation*. These two steps happen transparently and without user interaction.

- Following the compilation process, the simulation can be run to produce output data, in the form of event datafiles, histograms or an interactive visualization of the instrument geometry. The tools layer contains the tools mcplot and mcdisplay, for histograms and instrument geometry data respectively.

3. User contributions - and how we handle them

As mentioned before, many of the components and instruments provided with McStas are contributed from outside the core development team, which we generally feel is an important strength and difference of McStas with respect to the competing codes. To underline the Open Source nature of McStas, we have chosen to distribute the code with contributions under the Gnu Public License [10], version 2.0.

We always welcome contributions and have a relatively informal refereeing procedure: A contributed component requires a contributed test-instrument that demonstrates its use. The in-instrument and in-component documentation must be meaningful, and the code must be relatively readable. If the receiving McStas team member finds that the quality of the code does not live up to the general standard of the McStas code, there is typically an iterative process of mutual communication and improvement of the code.

A contributed component will always start its McStas life within the *contrib* directory of the component library, to signify that it was not written by a McStas core team member.

The McStas team takes care of supporting user requests for both the system- and contributed components, but a contributed component will always contain the name and contact information of the author, who is considered the final responsible of the code.

When a contributor is no longer available to support his code, the McStas team will decide to either retire the component or to promote the component to a fully supported system component, based on estimates of the usability, performance and quality of the component.

4. A quick tour of the example suite

The McStas example instruments are a valuable resource for new users. We currently have 100 example instruments in different categories, out of which a few have been listed

- Large scale structure instruments
  - KWS-II from Jülich/FRM-II, available in several versions with different sample types included.
  - SANS-II from PSI - see figure 2
  - Crisp reflectometer from ISIS TS1, including a very nice multilayer-sample (requires GNU Scientific Library)
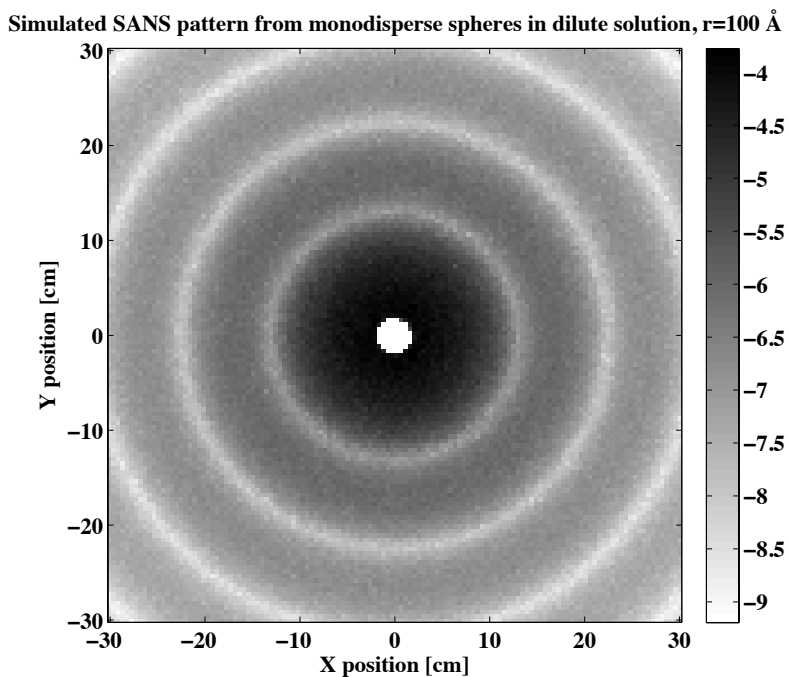- Diffractometers

Simulated SANS pattern from monodisperse spheres in dilute solution, r=100 Å

Figure 2. Virtual SANS data, showing the a pattern from monodisperse spheres in dilute solution on a position sensitive detector, logarithmic colorscale. Unpublished simulation data. Instrument file templateSANS.instr by K. Lefmann.
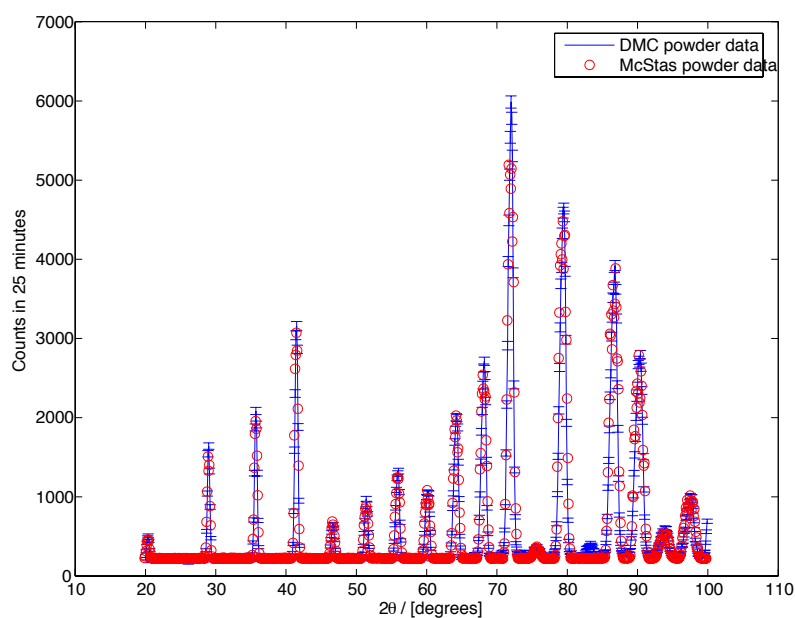

Figure 3. The figure shows a comparison of virtual and measuered diffraction $Na_2Ca_3Al_2F_{14}$ data[11], taken using the DMC powder diffractomer instrument at PSI. The McStas instrument file PSI_DMC.instr by P. Willendrup, U. Filges, L. Keller is included in the McStas distribution.
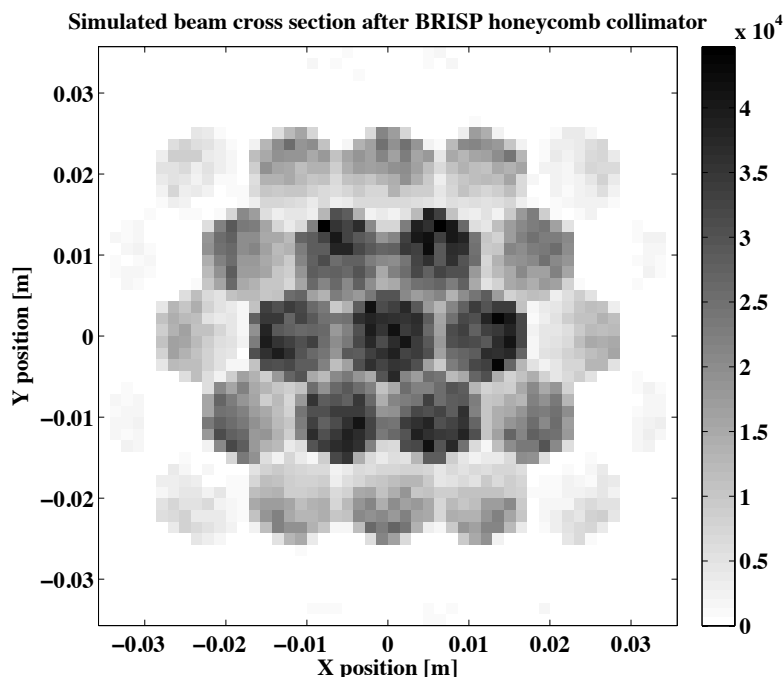
Figure 4. The figure shows a simulation of the beam cross section after the honeycomb-shaped guide at the ILL BRISP spectrometer. Unpublished simulation data. The used instrument file ILL_BRISP.instr by E. Farhi and N. Formissano is included in the McStas distribution.
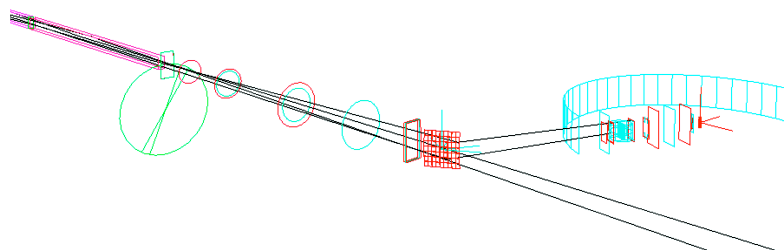

Figure 5. 3D-visualization of the PSI Focus instrument, using the mcdisplay tool. In the left part of the image the final guide-tube a disk-chopper and slits are shown, in the middle the double-focusing monochromator is visible, in the right-most part Fermi-chopper, sample position and detectors are visible. Unpublished simulation data. The used instrument file PSI_FOCUS.instr by U. Filges. See also [14] is included in the McStas distribution.

- D1A from ILL, a high-resolution powder diffractometer
- D20 from ILL, a high-intensity powder diffractometer
- DMC from PSI - see Figure 3
- Generic Laue diffractometer
- Triple axis instruments
    - IN12 from ILL, a cold triple axis
    - IN22 from ILL, a thermal triple axis on a guide (CRG)
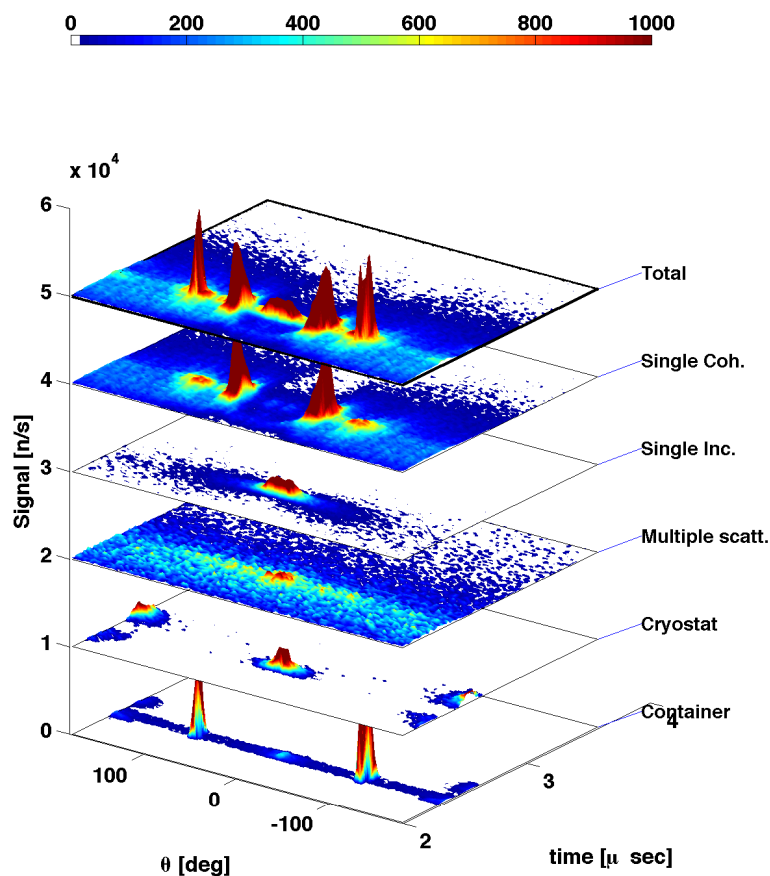   - IN20 from ILL, a thermal triple axis on a beamtube

Figure 6. Virtual experiment on liquid Ge, see [15]. Using labeling of individual neutron scattering processes, the full detector signal is decomposed into parts: Coherent, incoherent and multiple scattering contributions from the sample, contaminations from sample container and surrounding sample environment (cryostat). The used instrument file template_TOF.instr by E. Farhi is included in the McStas distribution.

  – h8 from Brookhaven, Brockhouse's original TAS
  – TAS1 from Risø, decomissioned
- TOF inelastic
  – IN5 from ILL, a cold TOF spectrometer
  – IN6 from ILL, a cold TOF spectrometer
  – BRISP from ILL - see Figure 4, small-angle spectrometer (CRG)
  – HET from ISIS TS1, decomissioned
  – Focus from PSI - see Figure 5
  – IN5-type from ESS
  – Generic direct geometry TOF - see Figure 6
- Hybrid inelastic
  – Prisma2 from ISIS TS1, decomissioned
- Spin echo, a simple spin-echo instrument based on transmission pl
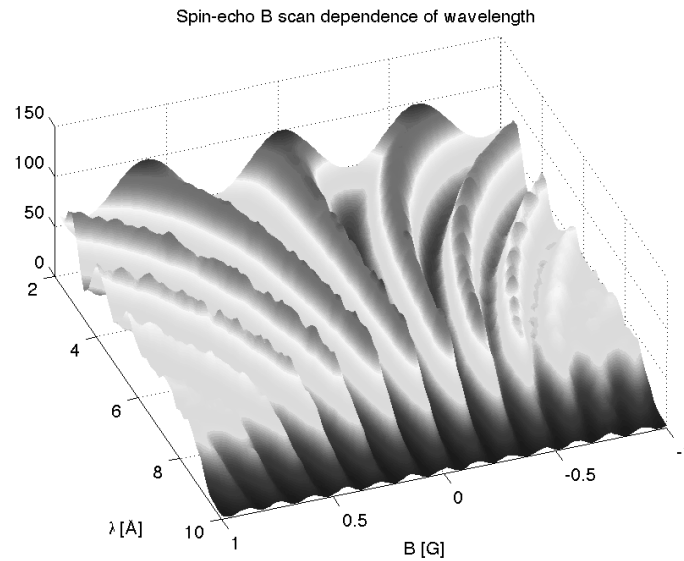 – Generic Spin-Echo instrument - see Figure 7
- Imaging

Figure 7. A series of concatenated, virtual spin-echo scans. As a function of varied wavelength, spin-echo groups of varying width are revealed when scanning a simulated correction coil. The instrument file used is SE_example.instr by E. Knudsen and P. Willendrup, is included in the McStas distrubtion. See also [16] and [17].
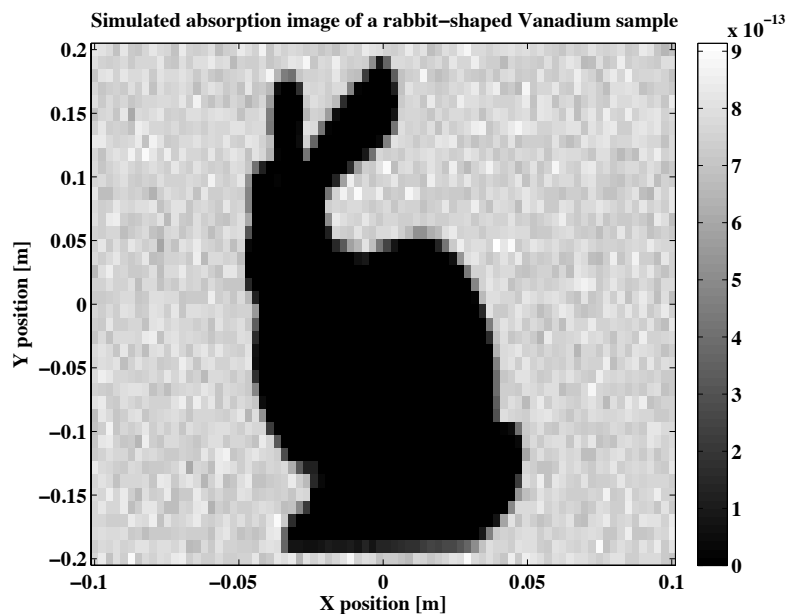

Figure 8. Absorption image / single projection in the tomography example. A rabbit-shaped Vanadium sample is revealed. Unpublished simulation data. The used instrument file Tomography.instr by P. Willendrup and R. Arnerin is included in the McStas distribution.

– Generic tomography station - see Figure 8

Supplementing these, a number of test instruments showing usage of individual components are included, as well as the very special "conversion tool" histogrammer.instr. Built on the general ability to read and use event-datafiles from McStas, VITESS, MCNP/MCNPX[12]-[13] and Tripoli, the histogrammer serves as an illustration of the flexibility of the McStas framework: One, simple, general instrument file provides a generic tool to generate *any* histogram-plot from any type of supported event data. And compiled once and for all, defines a stand-alone, alternative plotting tool e.g. for use with Vitess.

5. McStas releases 2.0 and 2.1

The latest version of McStas was release 2.0, dating from december 21st 2012. Our reason to increment the major version number related to certain loss of backward compatibility. We did however sacrifice the compatibility for at good cause: A nomenclature for parameter naming was defined, allowing much easier exchange of components between e.g. guides of different type, essentially only the component type need to be changed.

The most important general changes in the 2.0 release were a newly defined, uniformized parameter naming across components (which does break some backward compatibility for user-instruments) and a more robust and complete implementation of polarised neutron transport, including transport in user-defined tabularised magnetic fields e.g. from Radia[18] or freefem[19]).

From the user perspective, the addition of a new tool layer based on Python (which will eventually replace Perl) plus interfaces to Mantid[20], MCNP[21] and iFit[22] are important additions. With release 2.0, we have further started an effort together with the Vitess team of documenting how to define source brilliances for seleceted neutron sources with the two packages respectively. Currently, benchmarked parameters have been defined for the ILL, HZB, FRM-II, ISIS and ESS sources.

An important new feature of the sample components and some optics components is the support for abstract geometrical outer shapes, facilitated via the Geomview OFF file format[23].

A short-list of the most important new components added are

- Sample_nxs.comp from Mirko Boin, Helmholtz Zentrum Berlin. Diffraction/imaging oriented sample where one defines the structure by means of the NXS crystallography library[24] which is now also included with McStas. The component comes with the test-instruments Test_Sample_nxs_diffraction.instr and Test_Sample_nxs_imaging.instr.

- Mirror_Curved_Bispectral.comp and Mirror_Elliptic_Bispectral.comp from Henrik Jacobsen, NBI. Mirrors for the bispectral ESS instrument ESS_2001_bispectral.instr. See [25].

- Virtual_mcnp_ss_input.comp, Virtual_mcnp_ss_output.comp and Virtual_mcnp_ss_Guide.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, see [26].

- A large suite of SANS-samples from Martin Cramer Pedersen, NBI Various approaches to describing SANS diffraction e.g. using the Protein Data Bank (PDB)[27]. The TestSANS.instr instrument is provided for use cases.

For a full set of release notes for the McStas 2.0 release, please consult the McStas website [5].

The next release McStas 2.1 is expected early 2014, depending on development progress. The main remaining parts are a documentation overhaul and improvements to the new Python-based tool layer, however for many types of usage, the development version available via our Subversion source code repository http://www.mcstas.org/svn is in practice functional. We hereby recommend any experienced user to upgrade to the development version at will.

Users experiencing problems with either the development version or the actual McStas 2.1 release are encouraged to inform the McStas developers via the user mailinglist mcstas-users@mcstas.org.

6. Summary

McStas has during its 16 years long history of development proven to be a healthy and active project. What seperates McStas form its competitors is mainly the applied code-generation technique, the quality of documentation and the strongly collaborative nature of the package, allowing users to contribute.

The package includes an extensive example library consisting of instruments of all types, readily available as inspiration for the user.

The latest release is McStas 2.0 released in december 2012 and McStas 2.1 is planned for early 2014.

Roberts T., Roennow H., Roth S., Rycroft S., Sales M., Schedler R., Schober H., Stewart R., Udby L., Venturi G., Willendrup P., Zsigmond G., Vickery A., van Vuure T.

Since the list is likely not complete, we would also like to thank our large user base for all of their input, ideas and problems to fix.

References

[1] TE Mason, KN Clausen, G Aeppli et al. Canadian Journal of Physics **73**/11-12 pp 697-702 (1995).
[2] PA Seeger et al. Argonne National Laboratory Report ANL **98**/33, pp 202-218, (1998).
[3] K. Lefmann and K. Nielsen , Neutron News **10**, 20, (1999).
[4] P Willendrup, E Farhi, EB Knudsen, U Filges and K Lefmann, Journal of Neutron Research **17**, 35-43 (2014)
[5] McStas project website http://www.mcstas.org/
[6] J Saroun and J Kulda, Physica B **234**, pp. 1102, (1997).
[7] D Wechsler, G Zsigmond, F Streffer and F Mezei, Neutron News **11**, p 25, (2000).
[8] WT Lee and XL Wang, Neutron News **13** pp. 30-34, (2002)
[9] K Lefmann et al., Journal of Neutron Research **16**, pp. 97-111 (2008)
[10] GNU website http://www.gnu.org/licenses/gpl.html
[11] P Willendrup, U Filges, L Keller, E Farhi, K Lefmann , Physica B-Cond Matt **385** p 1032 (2006)
[12] X-5 Monte Carlo Team, Los Alamos report 5LA-UR-03-1987
[13] LS Waters, GW McKinney, JW Durkee, ML Fensin, J. S. Hendricks, et al., AIP Conf.Proc. **896** (2007)
[14] See http://lns00.psi.ch/mcworkshop/papers/focus_intercomparison.pdf
[15] E Farhi, V Hugouvieux, M Johnson and W Kob, Journal of Computational Physics **228**, pp. 5251-5261 (2009)
[16] EB Knudsen, L Udby, P Willendrup, K Lefmann and WG Bouwman, Physica B-Cond Matt **406** p 2361 (2011)
[17] EB Knudsen, A Tranum-Rømer, P Christiansen, P Willendrup, Lefmann K, Journal of Neutron Research **17**, 27-34 (2014)
[18] Radia website http://www.esrf.eu/Accelerators/Groups/InsertionDevices/Software/Radia
[19] Freefem website http://www.freefem.org/
[20] Mantid website http://www.mantidproject.org
[21] MCNPX website at LANL http://mcnpx.lanl.gov/
[22] iFit website http://ifit.mccode.org
[23] Geomview website http://www.geomview.org/docs/html/OFF.html
[24] M Boin Journal of Applied Crystallography **45** 603-607 (2012)
[25] H Jacobsen, K Lieutenant, C Zendler, K Lefmann NIMA **717** 69-76 (2013)
[26] E Klinkby et al. NIMA **700** 106-110 (2013)
[27] PDB website http://www.rcsb.org/pdb/home/home.do