Deliverable Number:	D8.8
Deliverable Title:	Port of selected scattering kernels from McStas to SIMRES

Delivery date:	Month 38
Leading beneficiary:	DTU
Dissemination level:	Public
Status:	final
Authors:	Erik Bergbäck Knudsen, DTU Physics
	Peter Willendrup, DTU Physics

Project number:	654000
Project acronym:	SINE2020
Project title:	Worldclass Science and Innovation with Neutrons in Europe 2020
Starting date:	1st of October 2015
Duration:	48 months
Call identifier:	H2020-INFRADEV-2014-2015
Funding scheme:	Combination of CP & CSA – Integrating Activities

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 654000

Abstract

This report serves as a the documentation for deliverable D8.8 of the SINE2020 project: Port of selected scattering kernels from McStas to SIMRES.

The McStas[1] distribution contains a large number of components that are used as sample models ranging from very simple benchmarking tools (e.g. Incoherent) to full scale virtual sample-models such as Isotropic Sqw and Single crystal. A full overview list of all included sample models and capabilities is maintained at[2] A snapshot of this web page is shown as Figure 2. This project was intended toward porting the most important/commonly used ones to SIMRES[3] to serve both communities. Such a capability could serve two purposes in that it would match the capabilities of SIMRES allowing in some cases to access the backward ray tracing mechanism for speed and still perform accurate simulation with sample models.

In general porting a large code-base to another framework requires careful thought in terms of reliability and maintainability. Simply translating a large piece of code is not considered best practice: as the main development branch evolves and gets updates, the ported code tends to lag behind. It is far better to have an automatic mechanism to transfer code through, or to maintain an solid standard interface between codes. In the case of McStas and SIMRES, the use of MCPL-files[4] may serve that purpose. This effectively renders *all* sample-components of McStas available to the SIMRES community all at once

Interface using McStas instruments and MCPL-files

Figure 1 below shows the minimal McStas-code needed to connect a single crystal, in this case a corundum crystal, to SIMRES. The code simply exposes some parameters of a McStas-crystal to the outside world.

```
1  COMPONENT in = MCPL_input(repeat=1, filename="vin.mcpl")
2  AT(0,0,0) RELATIVE Origin
3
4  COMPONENT single_crystal = Single_crystal(
5       reflections="Al2O3_sapphire.lau",
6       yheight=0.05, radius=0.01, mosaic=1, delta_d_d=1e-4,
7       az=4.757, ay=0, az=0, bx=2.3785, by=0, bz=-3.364,
8       cx=0, cy=12.9877, cz=0,
9       p_transmit=0.1)
10 AT (0, 0, 0) RELATIVE PREVIOUS
11
12 COMPONENT out = MCPL_output(filename="vout.mcpl")
13 AT(0,0,0) RELATIVE PREVIOUS
```

Figure 1: Listing of minimal McStas sample instrument, with a Sapphire single crystal as scattering element. In principle the sample description can be any McStas instrument - that can thus be integrated into SIMRES.

An example of a bare-bones surrounding SIMRES-defined diffractometer is shown in Figure 3 and Figure 4. In Figure 3 we define a primary spectrometer that generates neutrons from a source-model and transports them to the sample area. Once there, they Monte Carlo-neutrons are saved to disk in the efficient MCPL file-format. From this file events are read by the minimal McStas instrument-listing Figure 1, processed by the sample kernel and fed into a new MCPL-file. This latter file is read by the SIMRES secondary spectrometer, and the events contained in it are processed further until detected or otherwise terminated.

Method summary

In summary, the developed simulation pipeline, from a user perspective, is:

1. Define the primary spectrometer in SIMRES
2. Define the secondary spectrometer in SIMRES
3. Set parameters for the sample scattering kernel in a minimal McStas-file, and compile it.
4. In sequence:
    a. run primary spectrometer,
    b. run minimal McStas sample
    c. run secondary spectrometer.

This procedure effectively "ports" *all* available McStas sample kernels to SIMRES in one go, and keeps the operating modalities of both simulation programs with *no* additional maintenance effort. The primary upshot of the developed strategy is that the risk of breakdowns is kept at an absolute minimum. The division of responsibility is clear - additionally the procedure is very accessible to users. Users can use their established work flows as they please for anything that is not the sample, and only have to write sample parameters in McStas. Users who require more complex sample models on the other hand have access to the full McStas sample range and combinations thereof, e.g. the Union-concept[5].

Ready-made ports of selected samples

To further illustrate the possibilities and ease of use of McStas samples from within SIMRES (or other codes supporting the MCPL format), a subset of the McStas sample components have been wrapped in the same type of instrument file as illustrated in Figure 1**.** For standardization purposes all of the instrument files take input from an MCPL file named similarly to the instrument file and give output with a similar naming scheme, i.e.

`McStas_[sample]_in.mcpl → McStas_[sample].instr →McStas_[sample]_out.mcpl`

The instrument-interfaces are simplified with respect to the full functionality of the corresponding componets[6] e.g. those parameters that can be used to overwrite the input given via a *reflection list* in a normal McStas simulation have been left out. Also, focusing-oriented parameters are not sensitive to the position of other elements in the instrument file, i.e. no target_index parameter is available.

The finalised set of interface-instruments developed for SIMRES use are

1. McStas_PowderN.instr[7] with input parameters and example values:
   - **`reflections=Al.lau`**
     Input file for reflections, laz and lau formats from McStas accepted
   - **`geometry=cube.off`**
     Name of an Object File Format (OFF) or PLY file for complex geometry.
   - **`radius, xwidth, yheight, zdepth`**
     Geometry parameters [m].
   - **`thickness`**
     Thickness of hollow sample [m].
   - **`pack`**
     Packing factor.
   - **`d_omega, d_phi`**
     Horizontal (incoherent only) and vertical focusing limits [deg]
   - **`focus_flip`**
     Controls the sense of d_phi. If 0 d_phi is measured against the xz-plane. If nonzero, d_phi is measured against zy-plane.
   - **`tth_sign`**
     Sign of the scattering angle. If 0, the sign is chosen randomly
   - **`barns`**
     Flag to indicate if $|F^2|$ from 'reflections' is in barns or fm^2 (*barns* = 1 for laz, *barns* = 0 for lau type files).
   - **`rot_x, rot_y, rot_z`**
     Define rotations of the sample wrt. the incoming beam direction [deg]

2. McStas_Single_crystal.instr[8] with input parameters and example values:
   - **`reflections=Al.lau`**
     File name containing structure factors of reflections. Use empty ("") or NULL for incoherent scattering only
   - **`geometry=cube.off`**
     Name of an Object File Format (OFF) or PLY file for complex geometry.

- **radius, xwidth, yheight, zdepth**
  Geometry parameters [m].
- **delta_d_d, mosaic, mosaic_a, mosaic_b, mosaic_c**
  Definition of lattice spacing spread and mosacities.
- **barns**
  Flag to indicate if $|F^2|$ from 'reflections' is in barns or fm^2 (*barns* = 1 for laz, *barns* = 0 for lau type files).
- **order**
  Limit multiple scattering up to given order (0: all, 1: first, 2: second, ...)
- **rot_x, rot_y, rot_z**
  [deg] Define rotations of the sample wrt. the incoming beam direction

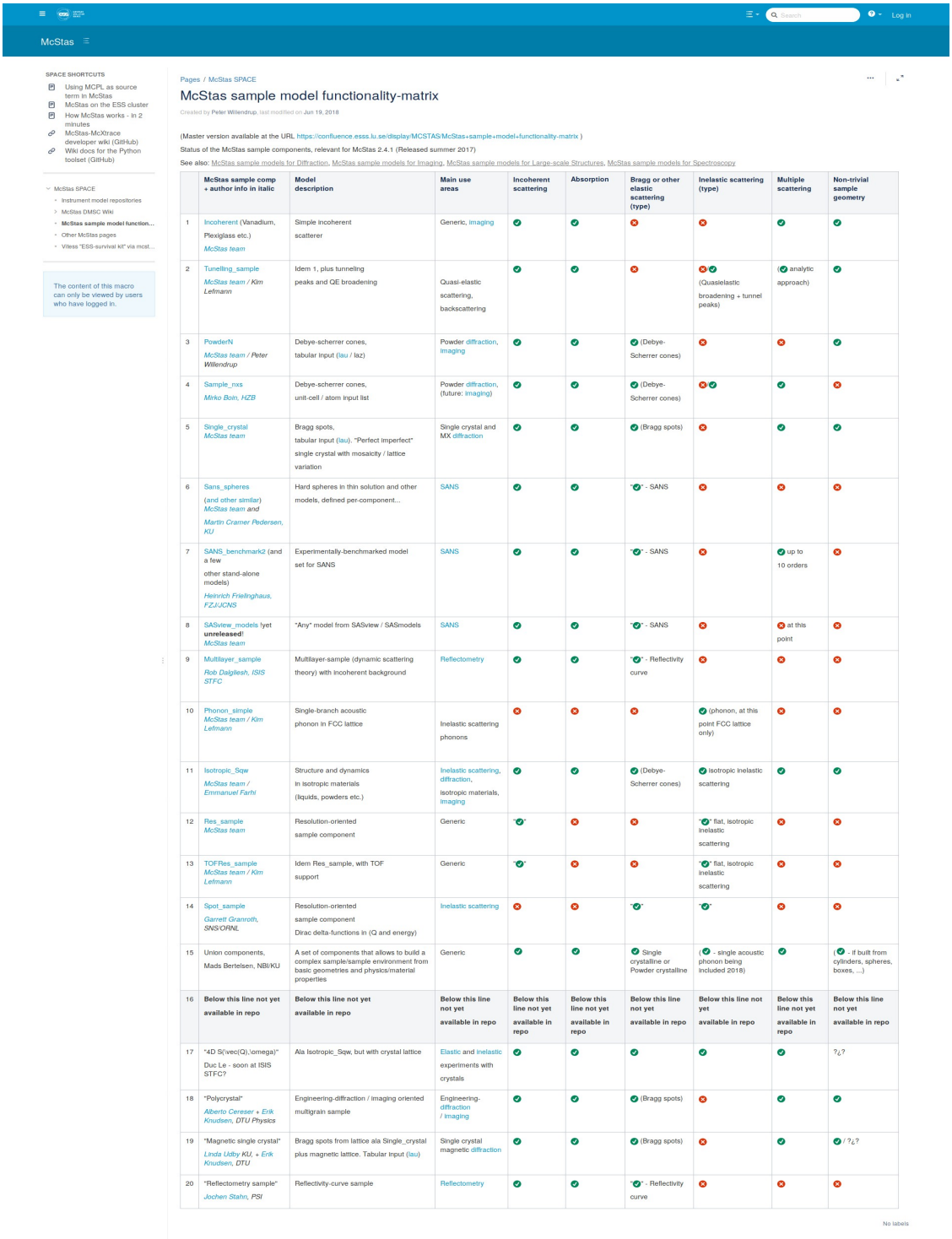
3. McStas_Isotropic_Sqw.instr[9] with input parameters and example values:

- **Sqw_coh**
  Name of the file containing the values of Q, w and S(Q,w) Coherent part; Q in Angs-1, E in meV, S(q,w) in meV-1. Use 0, NULL or "" to disable.
- **Sqw_inc**
  Name of the file containing the values of Q, w and S(Q,w). Incoher- ent (self) part. Use 0, NULL or "" to scatter isotropically (V-like).
- **geometry=cube.off**
  Name of an Object File Format (OFF) or PLY file for complex geometry.
- **radius, xwidth, yheight, zdepth**
  Geometry parameters [m].
- **thickness**
  Thickness of hollow sample [m].
- **threshold**
  Value under which S(Q,w) is not accounted for. to set according to the S(Q,w) values, i.e. not too low.
- **order**
  Limit multiple scattering up to given order (0: all, 1: first, 2: second, ...)
- **T**
  Temperature of sample, detailed balance. Use T=0 to disable it. and T=-1 to automatically set it from non-classical S(q,w).
- **verbose**
  Verbosity level (0:silent, 1:normal, 2:verbose, 3:debug). A ver- bosity¿1 also computes dispersions and S(q,w) analysis.
- **d_phi**
  Scattering vertical angular spreading (usually the height of the next component/detector). Use 0 for full space. This is only relevant for single scattering (order=1).

- **classical**
  Assumes the S(q,w) data from the files is a classical S(q,w), and multiply that data by exp(hw/2kT) on up/down energy sides. Use 0 when obtained from raw experiments, 1 from molecular dynamics. Use -1 to guess from a data set including both energy sides.
- **norm**
  Normalize S(q,w) when -1 (default). Use raw data when 0, multiply S(q,w) when norm>0.
- **powder_barns**
  0 when F^2 data in powder file are fm^2, 1 when in barns (barns=1 for laz, barns=0 for lau type files).
- **quantum_correction**
  Specify the type of quantum correction to use ”Boltzmann”/”Schofield”, ”harmonic”/”Bader” or ”standard”/”Frommhold” (default)
- **rot_x, rot_y, rot_z**
  Define rotations of the sample wrt. the incoming beam direction [deg]

Conclusion

As outlined in the deliverable report, a general mechanism for the use of McStas code in RESTRAX has been devised via the MCPL event interchange format. The developed McStas instrument files will become publicly available with mcstas-2.5.1 expected during the spring of 2019. As a further example of the possibility to exchange simulation data between SIMRES and McStas, the `ESS_BEER_MCPL` example instrument was already released with mcstas-2.5 in December 2018. The general availability of MCPL support in SIMRES will come with the next release expected during the spring of 2019.



McStas

SPACE SHORTCUTS

- Using MCPL as source term in McStas
- McStas on the ESS cluster
- How McStas works - in 2 minutes
- McStas-McXtrace developer wiki (GitHub)
- Wiki docs for the Python toolset (GitHub)

McStas SPACE

- Instrument model repositories
- McStas DMSC Wiki
- **McStas sample model function...**
- Other McStas pages
- Vitess "ESS-survival kit" via mcst...

The content of this macro can only be viewed by users who have logged in.

Pages / McStas SPACE

McStas sample model functionality-matrix

Created by Peter Willendrup, last modified on Jun 19, 2018

(Master version available at the URL https://confluence.esss.lu.se/display/MCSTAS/McStas+sample+model+functionality-matrix)

Status of the McStas sample components, relevant for McStas 2.4.1 (Released summer 2017)

See also: McStas sample models for Diffraction, McStas sample models for Imaging, McStas sample models for Large-scale Structures, McStas sample models for Spectroscopy

	McStas sample comp + author info in italic	Model description	Main use areas	Incoherent scattering	Absorption	Bragg or other elastic scattering (type)	Inelastic scattering (type)	Multiple scattering	Non-trivial sample geometry
1	Incoherent (Vanadium, Plexiglass etc.) *McStas team*	Simple incoherent scatterer	Generic, imaging	✓	✓	✗	✗	✓	✓
2	Tunelling_sample *McStas team / Klm Lefmann*	Idem 1, plus tunnelling peaks and QE broadening	Quasi-elastic scattering, backscattering	✓	✓	✗	✗✓ (Quasielastic broadening + tunnel peaks)	(✓ analytic approach)	✓
3	PowderN *McStas team / Peter Willendrup*	Debye-scherrer cones, tabular input (lau / laz)	Powder diffraction, imaging	✓	✓	✓ (Debye-Scherrer cones)	✗	✗	✓
4	Sample_nxs *Mirko Boin, HZB*	Debye-scherrer cones, unit-cell / atom input list	Powder diffraction, (future: imaging)	✓	✓	✓ (Debye-Scherrer cones)	✗✓	✓	✗
5	Single_crystal *McStas team*	Bragg spots, tabular input (lau). "Perfect imperfect" single crystal with mosaicity / lattice variation	Single crystal and MX diffraction	✓	✓	✓ (Bragg spots)	✗	✓	✓
6	Sans_spheres (and other similar) *McStas team and Martin Cramer Pedersen, KU*	Hard spheres in thin solution and other models, defined per-component...	SANS	✓	✓	"✓" - SANS	✗	✗	✗
7	SANS_benchmark2 (and a few other stand-alone models) *Heinrich Frielinghaus, FZJ/JCNS*	Experimentally-benchmarked model set for SANS	SANS	✓	✓	"✓" - SANS	✗	✓ up to 10 orders	✗
8	SASview_models !yet **unreleased**! *McStas team*	"Any" model from SASview / SASmodels	SANS	✓	✓	"✓" - SANS	✗	✗ at this point	✗
9	Multilayer_sample *Rob Dalgliesh, ISIS STFC*	Multilayer-sample (dynamic scattering theory) with incoherent background	Reflectometry	✓	✓	"✓" - Reflectivity curve	✗	✗	✗
10	Phonon_simple *McStas team / Klm Lefmann*	Single-branch acoustic phonon in FCC lattice	Inelastic scattering phonons	✗	✗	✗	✓ (phonon, at this point FCC lattice only)	✗	✗
11	Isotropic_Sqw *McStas team / Emmanuel Farhi*	Structure and dynamics in isotropic materials (liquids, powders etc.)	Inelastic scattering, diffraction, isotropic materials, imaging	✓	✓	✓ (Debye-Scherrer cones)	✓ Isotropic inelastic scattering	✓	✓
12	Res_sample *McStas team*	Resolution-oriented sample component	Generic	"✓"	✗	✗	"✓" flat, isotropic inelastic scattering	✗	✗
13	TOFRes_sample *McStas team / Klm Lefmann*	Idem Res_sample, with TOF support	Generic	"✓"	✗	✗	"✓" flat, isotropic inelastic scattering	✗	✗
14	Spot_sample *Garrett Granroth, SNS/ORNL*	Resolution-oriented sample component Dirac delta-functions in (Q and energy)	Inelastic scattering	✗	✗	"✓"	"✓"	✗	✗
15	Union components, Mads Bertelsen, NBI/KU	A set of components that allows to build a complex sample/sample environment from basic geometries and physics/material properties	Generic	✓	✓	✓ Single crystalline or Powder crystalline	(✓ - single acoustic phonon being included 2018)	✓	(✓ - if built from cylinders, spheres, boxes, ...)
16	**Below this line not yet available in repo**	**Below this line not yet available in repo**	**Below this line not yet available in repo**	**Below this line not yet available in repo**	**Below this line not yet available in repo**	**Below this line not yet available in repo**	**Below this line not yet available in repo**	**Below this line not yet available in repo**	**Below this line not yet available in repo**
17	"4D S(\vec(Q),\omega)" Duc Le - soon at ISIS STFC?	Ala Isotropic_Sqw, but with crystal lattice	Elastic and inelastic experiments with crystals	✓	✓	✓	✓	✓	?¿?
18	"Polycrystal" *Alberto Cereser + Erik Knudsen, DTU Physics*	Engineering-diffraction / imaging oriented multigrain sample	Engineering-diffraction / imaging	✓	✓	✓ (Bragg spots)	✗	✓	✓
19	"Magnetic single crystal" *Linda Udby KU, + Erik Knudsen, DTU*	Bragg spots from lattice ala Single_crystal plus magnetic lattice. Tabular input (lau)	Single crystal magnetic diffraction	✓	✓	✓ (Bragg spots)	✗	✓	✓ / ?¿?
20	"Reflectometry sample" *Jochen Stahn, PSI*	Reflectivity-curve sample	Reflectometry	✓	✓	"✓" - Reflectivity curve	✗	✗	✗

No labels

Figure 2: A snapshot of the McStas sample table maintained at [2]. For each sample model its main limitations and features are indicated, e.g. if it includes multiple scattering.

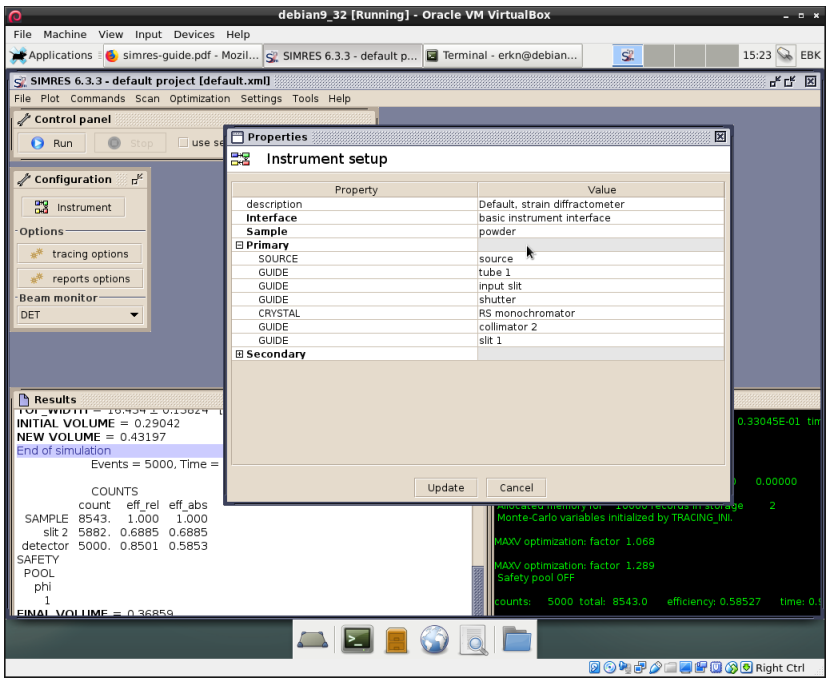


Figure 3: Screenshot of SIMRES showing an example front-end of a simplistic diffractometer, up to the sample position.

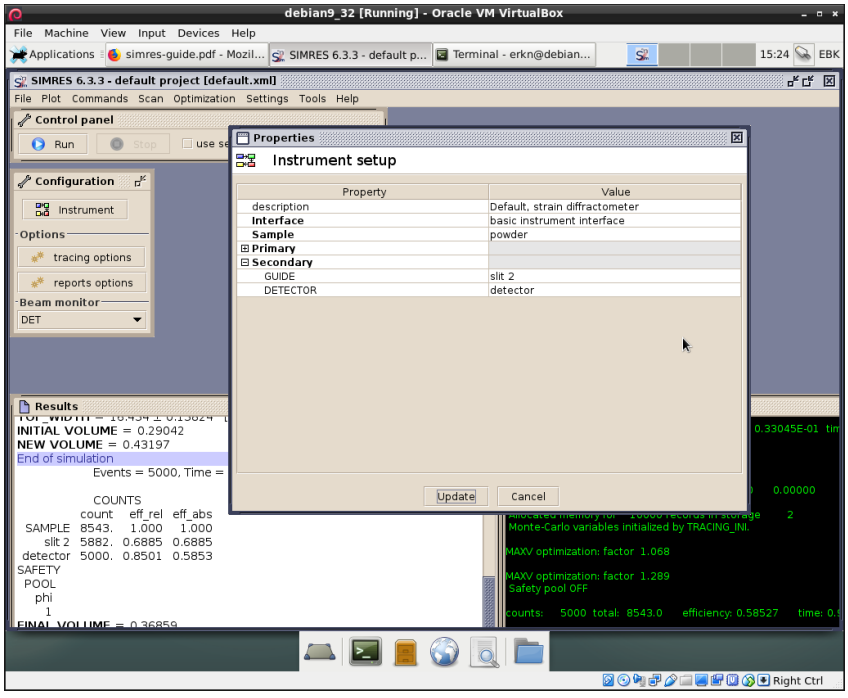


Figure 4: Screenshot of SIMRES showing an example back-end of a simplis- tic diffractometer, from the sample, to the detector(s). In this case it merely consists of a slit and a detector.

Acknowledgements

The authors would like to thank Jan Šaroun (NPI) for the inclusion of MCPL into the SIMRES package, thereby facilitating the inclusion of McStas sample functionality to that package. We would further like to express our gratitude toward Thomas Kittelmann (ESS) who was a driving factor in the development of the MCPL code .

---

[1] See the McStas website at http://www.mcstas.org and the publication https://doi.org/10.3233/JNR-130004

[2] McStas-sample table at the ESS Confluence page https://confluence.esss.lu.se/display/MCSTAS/McStas+sample+model+ functionality-matrix

[3] See the RESTRAX/SIMRES website at http://neutron.ujf.cas.cz/ restrax/ and the publication https://doi.org/10.1016/j.nima.2010. 06.219

[4] Monte Carlo Particle List (MCPL) format, see https://doi.org/10.1016/ j.cpc.2017.04.012, https://mctools.github.io/mcpl/, and SINE2020 deliverable report D8.2

[5] McStas-Union Concept by Mads Bertelsen, see e.g. https://europeanspallationsource.se/article/2018/02/08/instrument-simulation-software-advances-help-eu-interview-mads-bertelsen

[6] Component documentation at the McStas website. http://www.mcstas. org/download/components/

[7] Source code for McStas_PowderN.instr at https://github.com/McStasMcXtrace/McCode/blob/master/mcstas-comps/examples/McStas_PowderN.instr

[8] Source code for McStas_Single_crystal.instr at https://github.com/McStasMcXtrace/McCode/blob/master/mcstas-comps/examples/McStas_Single_crystal.instr

[9] Source code for McStas_Isotropic_Sqw.instr at https://github.com/McStasMcXtrace/McCode/blob/master/mcstas-comps/examples/McStas_Isotropic_Sqw.instr