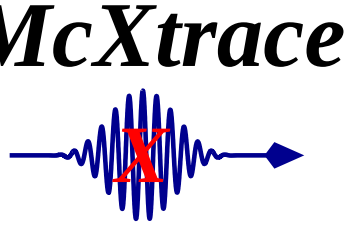
Physics Department,
Technical University of Denmark
2800 Kongens Lyngby, Denmark



Component Manual for the Xray-Tracing Package McXtrace, version 3.10.4

E. B. Knudsen, A. Prodi, J. Baltser, P. Willendrup,
A. Vickery, E. Farhi, K. Lefmann, S. Schmidt

November, 2026

The software package McXtrace is a tool for carrying out highly complex Monte Carlo ray-tracing simulations of X-ray beamlines to high precision. The simulations can compute all aspects of the performance of instruments and can thus be used to optimize the use of existing equipment, design new instrumentation, and carry out virtual experiments for e.g. training, experimental planning or data analysis. McXtrace is based is based on a unique design, inhereted from its sister McStas, where an automatic compilation process translates high-level textual instrument descriptions into efficient ANSI-C code. This design makes it simple to set up typical simulations and also gives essentially unlimited freedom to handle more unusual cases. This report constitutes the component manual for McXtrace, and, together with the manual for the McXtrace system, it contains full documentation of all aspects of the program. It covers a description of all official components of the McXtrace package with some theoretical background. Selected test instruments and representative McXtrace simulations performed with these instruments are described in the User Manual.

This report documents the components for McXtrace version 3.10.4, released November, 2026.

The authors are:

Erik Bergbäck Knudsen
Physics Department, Techical Univerisity of Denmark, Kgs. Lyngby, Denmark
email: erkn@fysik.dtu.dk

Andrea Prodi
Niels Bohr Institute, University of Copenhagen, Denmark
European Synchrotron Radiation Facility, Grenoble, France
email: aprodi@nbi.ku.dk

Peter Kjær Willendrup
Physics Department, Techical Univerisity of Denmark, Kgs. Lyngby, Denmark
email: pkwi@fysik.dtu.dk

Kim Lefmann
Niels Bohr Institute, University of Copenhagen, Denmark
email: lefmann@fys.ku.dk

Emmanuel Farhi
Synchrotron SOLEIL, Saint-Aubin, France
email: emmanuel.farhi@synchrotron-soleil.fr

Contents

Preface and acknowledgements **12**

1. About the component library **14**
1.1. Authorship . . . 14
1.2. Symbols for x-ray scattering and simulation . . . 14
1.3. Component coordinate system . . . 15
1.4. About data files . . . 15
1.5. Component source code . . . 16
1.6. Documentation . . . 16
1.7. Disclaimer, bugs . . . 16

2. Monte Carlo Techniques and simulation strategy **17**
2.1. X-ray simulations . . . 17
2.1.1. Monte Carlo ray tracing simulations . . . 18
2.2. The x-ray weight . . . 18
2.2.1. Statistical errors of non-integer counts . . . 19
2.3. Weight factor transformations during a Monte Carlo choice . . . 20
2.3.1. Direction focusing . . . 20
2.4. Stratified sampling . . . 21
2.5. Accuracy of Monte Carlo simulations . . . 22

3. Source components **23**
3.0.1. Photon flux and Brilliance . . . 23
3.1. The `Source_pt` McXtrace Component . . . 25
3.2. The `Source_flat` McXtrace Component . . . 27
3.3. The `Source_div` McXtrace Component . . . 28
3.4. The `Source_gaussian` McXtrace Component . . . 30
3.5. The `Source_lab` McXtrace Component . . . 32
3.6. Other sources components: virtual sources (event files) . . . 37

4. Beam optical components: Arms, slits, filters etc. **38**
4.1. The `Arm` McXtrace Component . . . 38
4.2. The `Slit` McXtrace Component . . . 39
4.3. The `Slit_N` McXtrace Component . . . 40
4.4. The `Beamstop` McXtrace Component . . . 41
4.5. The `Mask` McXtrace Component . . . 43
4.6. The `Filter` McXtrace Component . . . 44
4.6.1. Example . . . 46

4.7. The Chopper_simple McXtrace Component 47

5. Refractive optical components: lenses **50**
5.1. The Lens_simple McXtrace Component 50
5.2. The Lens_parab McXtrace Component 51
5.3. The Lens_parab_Cyl McXtrace Component 52
5.4. The Lens_elliptical McXtrace Component 54

6. Reflective optical components: mirrors **56**
6.1. The Mirror_curved McXtrace Component 56
6.2. The Mirror_parabolic McXtrace Component 58
6.3. The Mirror_elliptic McXtrace Component 59
6.4. The Multilayer_elliptic McXtrace Component 61
6.4.1. Definition of the reference frames 63
6.4.2. Algorithm 63
6.4.3. Reflection of the ray in the mirror 64
6.4.4. Mirror reflectivity 65
6.5. The TwinKB_ML McXtrace Component 65

7. Samples **67**
7.0.1. Scattering notation 67
7.0.2. Weight transformation in samples; focusing 68
7.0.3. Future development of sample components 69
7.1. The Absorption_sample McXtrace Component 69
7.2. The Saxs_spheres McXtrace Component 72
7.3. The PowderN McXtrace Component 74
7.3.1. Files formats: powder structures 78
7.3.2. Geometry, physical properties, concentricity 79
7.3.3. Powder scattering 80
7.3.4. Algorithm 82
7.4. The Single_crystal McXtrace Component 83
7.5. The Molecule_2state McXtrace Component 89

8. Monitors and detectors **91**
8.1. The Monitor McXtrace Component 92
8.2. The E_monitor McXtrace Component 92
8.3. The L_monitor McXtrace Component 93
8.4. The PSD_monitor McXtrace Component 94
8.5. The PSD_monitor_coh McXtrace Component 96
8.6. The PSD_monitor_4PI McXtrace Component 97
8.7. The EPSD_monitor McXtrace Component 98
8.8. The W_psd_monitor McXtrace Component 99
8.9. The Monitor_nD McXtrace Component 101
8.9.1. The Monitor_nD geometry 106

8.9.2. The photon parameters that can be monitored . . . 107
8.9.3. Important options . . . 108
8.9.4. The output files . . . 108
8.9.5. Monitor equivalences . . . 109
8.9.6. Usage examples . . . 109
8.9.7. Monitoring user variables . . . 110

9. Special-purpose components 112
9.1. The `Progress_bar` McXtrace Component . . . 113
9.2. The `Text` McXtrace Component . . . 114
9.3. The `Shadow_input` McXtrace Component . . . 115
9.4. The `Shadow_output` McXtrace Component . . . 115

10. The Union framework: sample environments and multiple scattering 117
10.1. Union framework core: initialization, master, and termination . . . 117
10.2. The `Union_init` McXtrace Component . . . 117
10.3. The `Union_master` McXtrace Component . . . 118
10.4. The `Union_stop` McXtrace Component . . . 119
10.5. Union geometry components . . . 120
10.6. The `Union_box` McXtrace Component . . . 120
10.7. The `Union_cylinder` McXtrace Component . . . 122
10.8. The `Union_sphere` McXtrace Component . . . 124
10.9. The `Union_cone` McXtrace Component . . . 126
10.10. Union material definition . . . 127
10.11. The `Union_make_material` McXtrace Component . . . 127
10.12. Union scattering/interaction process components . . . 129
10.13. The `Incoherent_process` McXtrace Component . . . 129
10.14. The `Compton_xrl_process` McXtrace Component . . . 130
10.15. The `Rayleigh_xrl_process` McXtrace Component . . . 131
10.16. The `KN_xrl_process` McXtrace Component . . . 132
10.17. The `Powder_process` McXtrace Component . . . 133
10.18. The `Template_process` McXtrace Component . . . 135

11. SasView/sasmodels form factors: small-angle scattering 137
11.1. SasView form-factor components . . . 137
11.2. The `SasView_adsorbed_layer` McXtrace Component . . . 137
11.3. The `SasView_barbell` McXtrace Component . . . 139
11.4. The `SasView_barbell_aniso` McXtrace Component . . . 141
11.5. The `SasView_bcc_paracrystal` McXtrace Component . . . 142
11.6. The `SasView_bcc_paracrystal_aniso` McXtrace Component . . . 144
11.7. The `SasView_binary_hard_sphere` McXtrace Component . . . 146
11.8. The `SasView_broad_peak` McXtrace Component . . . 148
11.9. The `SasView_capped_cylinder` McXtrace Component . . . 149
11.10. The `SasView_capped_cylinder_aniso` McXtrace Component . . . 151

11.11The SasView_core_multi_shell McXtrace Component 153
11.12The SasView_core_shell_bicelle McXtrace Component 154
11.13The SasView_core_shell_bicelle_aniso McXtrace Component 155
11.14The SasView_core_shell_bicelle_elliptical McXtrace Component . . 158
11.15The SasView_core_shell_bicelle_elliptical_aniso McXtrace Component 160
11.16The SasView_core_shell_bicelle_elliptical_belt_rough McXtrace Component 162
11.17The SasView_core_shell_bicelle_elliptical_belt_rough_aniso McXtrace Component 164
11.18The SasView_core_shell_cylinder McXtrace Component 166
11.19The SasView_core_shell_cylinder_aniso McXtrace Component 168
11.20The SasView_core_shell_ellipsoid McXtrace Component 170
11.21The SasView_core_shell_ellipsoid_aniso McXtrace Component 172
11.22The SasView_core_shell_parallelepiped McXtrace Component 174
11.23The SasView_core_shell_parallelepiped_aniso McXtrace Component . 176
11.24The SasView_core_shell_sphere McXtrace Component 179
11.25The SasView_correlation_length McXtrace Component 180
11.26The SasView_cylinder McXtrace Component 182
11.27The SasView_cylinder_aniso McXtrace Component 184
11.28The SasView_dab McXtrace Component 185
11.29The SasView_ellipsoid McXtrace Component 187
11.30The SasView_ellipsoid_aniso McXtrace Component 188
11.31The SasView_elliptical_cylinder McXtrace Component 190
11.32The SasView_elliptical_cylinder_aniso McXtrace Component 192
11.33The SasView_fcc_paracrystal McXtrace Component 194
11.34The SasView_fcc_paracrystal_aniso McXtrace Component 196
11.35The SasView_flexible_cylinder McXtrace Component 197
11.36The SasView_flexible_cylinder_elliptical McXtrace Component . . 199
11.37The SasView_fractal McXtrace Component 201
11.38The SasView_fractal_core_shell McXtrace Component 202
11.39The SasView_fuzzy_sphere McXtrace Component 204
11.40The SasView_gauss_lorentz_gel McXtrace Component 206
11.41The SasView_gaussian_peak McXtrace Component 207
11.42The SasView_gel_fit McXtrace Component 209
11.43The SasView_guinier McXtrace Component 210
11.44The SasView_guinier_porod McXtrace Component 212
11.45The SasView_hardsphere McXtrace Component 213
11.46The SasView_hayter_msa McXtrace Component 215
11.47The SasView_hollow_cylinder McXtrace Component 217
11.48The SasView_hollow_cylinder_aniso McXtrace Component 218
11.49The SasView_hollow_rectangular_prism McXtrace Component 220
11.50The SasView_hollow_rectangular_prism_aniso McXtrace Component . 222

11.51 The SasView_hollow_rectangular_prism_thin_walls McXtrace Component . . . 224
11.52 The SasView_lamellar_hg McXtrace Component . . . 226
11.53 The SasView_lamellar_hg_stack_caille McXtrace Component . . . 227
11.54 The SasView_lamellar_stack_caille McXtrace Component . . . 229
11.55 The SasView_lamellar_stack_paracrystal McXtrace Component . . . 231
11.56 The SasView_line McXtrace Component . . . 232
11.57 The SasView_linear_pearls McXtrace Component . . . 234
11.58 The SasView_lorentz McXtrace Component . . . 235
11.59 The SasView_mass_fractal McXtrace Component . . . 237
11.60 The SasView_mass_surface_fractal McXtrace Component . . . 238
11.61 The SasView_mono_gauss_coil McXtrace Component . . . 240
11.62 The SasView_multilayer_vesicle McXtrace Component . . . 241
11.63 The SasView_onion McXtrace Component . . . 243
11.64 The SasView_parallelepiped McXtrace Component . . . 244
11.65 The SasView_parallelepiped_aniso McXtrace Component . . . 245
11.66 The SasView_peak_lorentz McXtrace Component . . . 247
11.67 The SasView_pearl_necklace McXtrace Component . . . 249
11.68 The SasView_poly_gauss_coil McXtrace Component . . . 250
11.69 The SasView_polymer_excl_volume McXtrace Component . . . 252
11.70 The SasView_polymer_micelle McXtrace Component . . . 253
11.71 The SasView_porod McXtrace Component . . . 255
11.72 The SasView_power_law McXtrace Component . . . 257
11.73 The SasView_pringle McXtrace Component . . . 258
11.74 The SasView_raspberry McXtrace Component . . . 259
11.75 The SasView_rectangular_prism McXtrace Component . . . 261
11.76 The SasView_rectangular_prism_aniso McXtrace Component . . . 262
11.77 The SasView_rpa McXtrace Component . . . 264
11.78 The SasView_sc_paracrystal McXtrace Component . . . 265
11.79 The SasView_sc_paracrystal_aniso McXtrace Component . . . 267
11.80 The SasView_sphere McXtrace Component . . . 268
11.81 The SasView_spinodal McXtrace Component . . . 270
11.82 The SasView_squarewell McXtrace Component . . . 271
11.83 The SasView_stacked_disks McXtrace Component . . . 273
11.84 The SasView_stacked_disks_aniso McXtrace Component . . . 275
11.85 The SasView_star_polymer McXtrace Component . . . 277
11.86 The SasView_stickyhardsphere McXtrace Component . . . 278
11.87 The SasView_superball McXtrace Component . . . 280
11.88 The SasView_superball_aniso McXtrace Component . . . 281
11.89 The SasView_surface_fractal McXtrace Component . . . 283
11.90 The SasView_teubner_strey McXtrace Component . . . 285
11.91 The SasView_triaxial_ellipsoid McXtrace Component . . . 286
11.92 The SasView_triaxial_ellipsoid_aniso McXtrace Component . . . 288
11.93 The SasView_two_lorentzian McXtrace Component . . . 290

11.94 The SasView_two_power_law McXtrace Component . . . 292
11.95 The SasView_vesicle McXtrace Component . . . 293

12. Contributed components **296**
12.1. Contributed optics: crystals, masks, and mirrors . . . 296
12.2. The Bragg_crystal_simple McXtrace Component . . . 296
12.3. The Bragg_crystal_BC McXtrace Component . . . 297
12.4. The Bragg_crystal_bent_BC McXtrace Component . . . 299
12.5. The Laue_crystal_BC McXtrace Component . . . 302
12.6. The Attenuating_mask McXtrace Component . . . 304
12.7. The Mirror_toroid_pothole McXtrace Component . . . 305
12.8. Contributed monitors and detectors . . . 306
12.9. The Detector_pn McXtrace Component . . . 306
12.10 The PSD_monitor_rad McXtrace Component . . . 307
12.11 The SAXSQMonitor McXtrace Component . . . 308
12.12 Contributed SAXS (small-angle X-ray scattering) samples . . . 309
12.13 The SAXSCurve McXtrace Component . . . 309
12.14 The SAXSSpheres McXtrace Component . . . 310
12.15 The SAXSCylinders McXtrace Component . . . 311
12.16 The SAXSEllipticCylinders McXtrace Component . . . 312
12.17 The SAXSShells McXtrace Component . . . 313
12.18 The SAXSLiposomes McXtrace Component . . . 314
12.19 The SAXSNanodiscs McXtrace Component . . . 316
12.20 The SAXSNanodiscsFast McXtrace Component . . . 317
12.21 The SAXSNanodiscsWithTags McXtrace Component . . . 319
12.22 The SAXSNanodiscsWithTagsFast McXtrace Component . . . 321
12.23 The SAXSPDB McXtrace Component . . . 322
12.24 The SAXSPDBFast McXtrace Component . . . 323

13. Obsolete components **326**

14. AstroX components: X-ray optics for astronomical instrumentation **327**
14.1. Sources . . . 327
14.2. The Source_extended McXtrace Component . . . 327
14.3. Mirror shells . . . 328
14.4. The Shell_c McXtrace Component . . . 328
14.5. The Shell_p McXtrace Component . . . 330
14.6. The Shell_h McXtrace Component . . . 332
14.7. Ring optics . . . 334
14.8. The Ring_c McXtrace Component . . . 334
14.9. The Ring_p McXtrace Component . . . 336
14.10 The Ring_h McXtrace Component . . . 337
14.11 Pore optics . . . 339
14.12 The Pore_c McXtrace Component . . . 339

14.13 The `Pore_p` McXtrace Component . . . 340
14.14 The `Pore_h` McXtrace Component . . . 342
14.15 Micropore/microchannel plate (MM) optics . . . 344
14.16 The `MM_c` McXtrace Component . . . 344
14.17 The `MM_p` McXtrace Component . . . 345
14.18 The `MM_h` McXtrace Component . . . 347

A. The Union components **349**
A.1. Motivation . . . 349
A.1.1. Volumes, priority and overlap . . . 350
A.2. Component classes . . . 350
A.2.1. Loading the Union library . . . 351
A.2.2. Scattering processes . . . 351
A.2.3. Union_make_material . . . 352
A.2.4. Geometry components . . . 352
A.2.5. Union_master . . . 354
A.3. The propagation algorithm . . . 354
A.3.1. Known limitations . . . 355
A.4. A complete example . . . 355
A.5. Further examples . . . 357

B. Libraries and conversion constants **358**
B.1. Run-time calls and functions (`mcxtrace-r`) . . . 358
B.1.1. Photon propagation . . . 358
B.1.2. Coordinate and component variable retrieval . . . 359
B.1.3. Coordinate transformations . . . 360
B.1.4. Mathematical routines . . . 361
B.1.5. Output from detectors . . . 361
B.1.6. Ray-geometry intersections . . . 362
B.1.7. Random numbers . . . 362
B.2. Reading a data file into a vector/matrix (Table input, `read_table-lib`) . 363
B.3. Constants for unit conversion etc. . . . 366

C. The McXtrace terminology **367**

Bibliography **368**

Index and keywords **369**

Preface and acknowledgements

This document contains information on the x-ray scattering components which are the building blocks for defining instruments in the Monte Carlo Xray-tracing program McXtrace version 3.10.4. The initial release in June 2011 of version 1.0 was presented in Ref. [LN99]. The reader of this document is not supposed to have specific knowledge of xray scattering, but some basic understanding of physics is helpful in understanding the theoretical background for the component functionality. For details about setting up and running simulations, we refer to the McXtrace system manual [Knu+14]. We assume familiarity with the use of the C programming language.

We would like to explicitly thank all the partners in this project:

- The European Synchrotron Radiation Facility (ESRF), Grenoble, France
- SAXSLAB Aps., Lundtofte, Denmark
- Risø DTU, Roskilde, Denmark
- Niels Bohr Institute, Univeristy of Copenhagen, Copenhagen, Denmark.
- Faculty for Life Sciences, University of Copenhagen, Copenhagen, Denmark.

As McXtrace has inherited much of its functionality from its sister McStas we take the oppurtunity to thank Dir. Kurt N. Clausen, PSI, for his continuous support to McStas and for having initiated the project. Continuous support to McStas has also come from Prof. Robert McGreevy, ISIS.

We have further benefited from discussions with many other people in the scattering community, too numerous to mention here.

The users who contributed components to this manual are acknowledged as authors of the individual components. We encourage other users to contribute components with manual entries for inclusion in future versions of McXtrace.

In case of errors, questions, or suggestions, do not hesitate to contact the user/developer community by writing to the user mailiing list `mcxtrace-users@mcxtrace.org` or consult the McXtrace home page [Mcx]. A special developement website (shared with the sister project McStas) complete with bug/request reporting service is available [Tra].

We would like to kindly thank all McXtrace component contributors. This is the way we improve the software alltogether.

If you **appreciate** this software, please subscribe to the `mcxtrace-users@mcxtrace.org` email list, send us a smiley message, and contribute to the package.

We also encourage you to refer to this software when publishing results, with the following citation:

- E. B. Knudsen, et.al, J. Applied Cryst. **46**, pp 679, 2013.

1. About the component library

This McXtrace Component Manual consists of the following major parts:

- An introduction to the use of Monte Carlo methods in McXtrace.
- A thorough description of system components, with one chapter per major category: Sources, optics, monochromators, samples, monitors, and other components.
- The McXtrace library functions and definitions that aid in the writing of simulations and components in Appendix B.
- An explanation of the McXtrace terminology in Appendix C.

Additionally, you may refer to the list of example instruments from the library in the McXtrace User Manual.

1.1. Authorship

The component library is maintained by the McXtrace system group. A number of basic components "belongs" the McXtrace system, and are supported and tested by the McXtrace team.

Other components are contributed by specific authors, who are listed in the code for each component they contribute as well as in this manual. McXtrace users are encouraged to send their contributions to us for inclusion in future releases.

1.2. Symbols for x-ray scattering and simulation

In the description of the theory behind the component functionality we will use the usual symbols $\mathbf{r}$ for the position (x, y, z) of the particle (unit m), and $\mathbf{k}$ for the particle wave vector (k_x, k_y, k_z) (unit Å^{-1}). The wavelength is the reciprocal wave vector, $\lambda = 2\pi/k$. By convention energy is usually given i keV and may be calculated from the wavelength by: $\lambda = \frac{12.398}{E}$

In general, vectors are denoted by boldface symbols.

Subscripts "i" and "f" denotes "initial" and "final", respectively, and are used in connection with the photon state before and after an interaction with the component in question.

MCXTRACE/data	Description
*.lau	Laue pattern file, as issued from Crystallographica. For use with Single_crystal and PowderN. Data: [h k l Mult. d-space 2Theta F-squared]
*.laz	Powder pattern file, as obtained from Lazy/ICSD. For use with PowderN.
*.txt	General text file, containing ascii data. Currently used for elemental data extracted from NIST[Nis].

Table 1.1.: Data files of the McXtrace library.

1.3. Component coordinate system

All mentioning of component geometry refer to the local coordinate system of the individual component. The axis convention is so that the z axis is along the photon propagation axis, the y axis is vertical up, and the x axis points left when looking along the z-axis, completing a right-handed coordinate system. Most components 'position' (as specified in the instrument description with the `AT` keyword) corresponds to their volume centre. The mirrors are an important counterexmaple. In this case the 'position' is the centre of the mirror surface

Components are not necessarily designed to overlap. This may lead to loss of rays. Warnings will be issued during simulation if sections of the instrument are not reached by any xrays, or if a significant number of xrays are removed. This is usually the sign of either overlapping components or low intensity.

1.4. About data files

Some components require external data files, e.g. lattice crystallographic definitions for Laue and powder pattern diffraction, absorption and reflectivity files, etc.

Such files distributed with McXtrace are located in the `data` sub-directory of the `MCXTRACE` library. Components that make use of the McXtrace file system, including the `read-table` library (see section B.2) may access all McXtrace data files without making local copies. Of course, you are welcome to define your own data files, and eventually contribute to McXtrace if you find them useful.

File extensions are not interpreted by McXtrace but help in identifying relevant files per application. Table 1.1 lists the current default file extensions.

McXtrace itself generates both simulation and monitor data files, the structure of which is explained in the User Manual (see end of chapter 'Running McXtrace').

1.5. Component source code

Source code for all components may be found in the `MCXTRACE` library subdirectory of the McXtrace installation; the default is `/usr/local/lib/mcxtrace/` on Unix-like systems and `C:\mcxtrace\lib` on Windows systems, but it may be changed using the `MCXTRACE` environment variable.

In case users only require to add new features, preserving the existing features of a component, using the `EXTEND` keyword in the instrument description file is recommended. For larger modification of a component, it is advised to make a copy of the component file into the working directory. A component file in the local directory will in McXtrace takes precedence over a library component of the same name.

1.6. Documentation

As a complement to this Component Manual, we encourage users to use the `mxdoc` front-end which enables to display both the catalog of the McXtrace library, e.g using:

```
mxdoc
```

as well as the documentation of specific components, e.g with:

```
mxdoc --text name
mxdoc file.comp
```

The first line will search for all components matching the *name*, and display their help section as text. For instance, `mxdoc .laz` will list all available Lazy data files, whereas `mxdoc --text Monitor` will list most Monitors. The second example will display the help corresponding to the *file.comp* component, using your BROWSER setting, or as text if unset. The `--help` option will display the command help, as usual.

An overview of the component library is also given at the McXtrace home page [Mcx] and in the User Manual [Knu+14].

1.7. Disclaimer, bugs

We would like to emphasize that the usage of both the McXtrace software, as well as its components are the responsability of the users. Indeed, obtaining accurate and reliable results requires a substantial work when writing instrument descriptions. This also means that users should read carefully both the documentation from the manuals [Knu+14] and from the component itself (using `mcdoc` *comp*) before reporting errors. Most anomalous results often originate from a wrong usage of some part of the package.

Anyway, if you find that either the documentation is not clear, or the behavior of the simulation is undoubtedly anomalous, you should report this to us at `mcxtrace-users@mcxtrace.org` and/or refer to our special bug/request reporting service [Mcz].

2. Monte Carlo Techniques and simulation strategy

This chapter explains the simulation strategy and the Monte Carlo techniques used in McXtrace. We first explain the concept of the x-ray weight factor, and discuss the statistical errors in dealing with sums of x-ray weights. Secondly, we give an expression for how the weight factor transforms under a Monte Carlo choice and specialize this to the concept of direction focusing. Finally, we present a way of generating random numbers with arbitrary distributions. More details are available in the Appendix concerning random numbers in the User manual.

2.1. X-ray simulations

X-ray scattering beamlines are built as a series of optical elements. Each of these elements modifies the beam characteristics (e.g. divergence, wavelength spread, spatial and temporal distributions) in a way which, for simple x-ray beam configurations, may be modelled with analytical methods.

However, real x-ray beamlines consist of a large number of optical elements, and this brings additional complexity by introducing strong correlations between x-ray beam parameters like divergence and position - which is the basis of the acceptance diagram method - but also wavelength and time. The usual analytical methods, such as phase-space theory, then reach their limit of validity in the description of the resulting effects.

In order to cope with this difficulty, Monte Carlo (MC) methods (for a general review, see Ref. [Jam80]) may be applied to the simulation of x-ray beamlines. The use of probability is commonplace in the description of microscopic physical processes. Integrating these events (absorption, scattering, reflection, ...) over the x-ray trajectories results in an estimation of measurable quantities characterizing the beamline. Moreover, using variance reduction (importance sampling) where possible, reduces the computation time and gives better accuracy.

Implementations of the MC method for X-ray beamlines already exist, most notable is probably *SHADOW*[WCC94], originally developed by the late Franco Cerrina and coworkers, now developed further by M. Sanchez Del Rio at the ESRF[Rio+11][Sha]. Other implementations of the same concept are *RAY*[Sch08] from BESSY and *Xtrace*[Bau+07]. hosted at ANKA

2.1.1. Monte Carlo ray tracing simulations

Mathematically, the Monte-Carlo method is an application of the law of large numbers [Jam80; GRR92]. Let $f(u)$ be a finite continuous integrable function of parameter u for which an integral estimate is desirable. The discrete statistical mean value of f (computed as a series) in the uniformly sampled interval $a < u < b$ converges to the mathematical mean value of f over the same interval.

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1, a\leq u_i\leq b}^{n} f(u_i) = \frac{1}{b-a} \int_a^b f(u)du \tag{2.1}$$

In the case were the u_i values are regularly sampled, we come to the well known midpoint integration rule. In the case were the u_i values are randomly (but uniformly) sampled, this is the Monte-Carlo integration technique. As random generators are not perfect, we rather talk about *quasi*-Monte-Carlo technique. We encourage the reader to refer to James [Jam80] for a detailed review on the Monte-Carlo method.

2.2. The x-ray weight

A totally realistic semi-classical simulation will require that each x-ray is at any time either present or lost. On many beamlines the sheer abundance of x-ray photons makes it impractical to trace each and every photon from the source. This is particularly the case at XFELs. Additionally, only a very small fraction of the initial x-rays will ever be detected, and simulations of this kind will therefore waste much time in dealing with x-rays that never hit the detector.

A way of dealing with these issues and speed up calculations is to introduce a x-ray "weight factor" for each simulated ray and to adjust this weight according to the path of the ray. If *e.g.* the reflectivity of a certain optical component is 10%, and only reflected x-rays ray are considered later in the simulations, the x-ray weight will be multiplied by 0.10 when passing this component, but every x-ray is allowed to reflect in the component. In contrast, the totally realistic simulation of the component would require on average ten incoming x-rays for each reflected one.

Let the initial x-ray weight be p_0 and let us denote the weight multiplication factor in the j'th component by π_j. The resulting weight factor for the x-ray ray after passage of the whole beamline becomes the product of all contributions

$$p = p_n = p_0 \prod_{j=1}^{n} \pi_j. \tag{2.2}$$

Each adjustement factor should be $0 < \pi_j < 1$, except in special circumstances, so that total flux can only decrease through the simulation. For convenience, the value of p is updated (within each component) during the simulation.

Simulation by weight adjustment is performed whenever possible. This includes

- Transmission through filters and windows.

- Reflection from monochromator (and analyser) crystals with finite reflectivity and mosaicity.
- Reflections from mirrors.
- Passage of a continuous beam through a chopper.
- Scattering from all types of samples.

2.2.1. Statistical errors of non-integer counts

In a typical simulation, the result will consist of a count of x-ray histories ("rays") with different weights. The sum of these weights is an estimate of the mean number of x-rays hitting the monitor (or detector) per second in a "real" experiment. One may write the counting result as

$$I = \sum_i p_i = N\overline{p}, \tag{2.3}$$

where N is the number of rays hitting the detector and the horizontal bar denotes averaging. By performing the weight transformations, the (statistical) mean value of I is unchanged. However, N will in general be enhanced, and this will improve the accuracy of the simulation.

To give an estimate of the statistical error, we proceed as follows: Let us first for simplicity assume that all the counted x-ray weights are almost equal, $p_i \approx \overline{p}$, and that we observe a large number of x-rays, $N \geq 10$. Then N almost follows a normal distribution with the uncertainty $\sigma(N) = \sqrt{N}$ [1]. Hence, the statistical uncertainty of the observed intensity becomes

$$\sigma(I) = \sqrt{N}\overline{p} = I/\sqrt{N}, \tag{2.4}$$

as is used in real x-ray experiments (where $\overline{p} \equiv 1$). For a better approximation we return to Eq. (2.3). Allowing variations in both N and $\overline{p}$, we calculate the variance of the resulting intensity, assuming that the two variables are independent:

$$\sigma^2(I) = \sigma^2(N)\overline{p}^2 + N^2\sigma^2(\overline{p}). \tag{2.5}$$

Assuming as before that N follows a normal distribution, we reach $\sigma^2(N)\overline{p}^2 = N\overline{p}^2$. Further, assuming that the individual weights, p_i, follow a Gaussian distribution (which in some cases is far from the truth) we have $N^2\sigma^2(\overline{p}) = \sigma^2(\sum_i p_i) = N\sigma^2(p_i)$ and reach

$$\sigma^2(I) = N\left(\overline{p}^2 + \sigma^2(p_i)\right). \tag{2.6}$$

The statistical variance of the p_i's is estimated by $\sigma^2(p_i) \approx (\sum_i p_i^2 - N\overline{p}^2)/(N-1)$. The resulting variance then reads

$$\sigma^2(I) = \frac{N}{N-1}\left(\sum_i p_i^2 - \overline{p}^2\right). \tag{2.7}$$

[1] This is not correct in a situation where the detector counts a large fraction of the x-rays in the simulation, but we will neglect that for now.

For almost any positive value of N, this is very well approximated by the simple expression

$$\sigma^2(I) \approx \sum_i p_i^2. \tag{2.8}$$

As a consistency check, we note that for all p_i equal, this reduces to eq. (2.4)

In order to compute the intensities and uncertainties, the detector components in McXtrace will keep track of $N = \sum_i p_i^0, I = \sum_i p_i^1$, and $M_2 = \sum_i p_i^2$.

2.3. Weight factor transformations during a Monte Carlo choice

When a Monte Carlo choice must be performed, *e.g.* when the initial energy and direction of the x-ray ray is decided at the source, it is important to adjust the x-ray weight so that the combined effect of x-ray weight change and Monte Carlo probability of making this particular choice equals the actual physical properties we like to model.

Let us follow up on the simple example of transmission. The probability of transmitting the real x-ray is P, but we make the Monte Carlo choice of transmitting the x-ray every time: $f_{\mathrm{MC}} = 1$. This must be reflected on the choice of weight multiplier π_j given by the master equation In the "real" semi-classical world, there is a distribution (probability density) for the x-rays in the six dimensional (energy, direction, position) space of $\Pi(E, \mathbf{\Omega}, \mathbf{r}) = dP/(dEd\mathbf{\Omega}d^3\mathbf{r})$ depending upon the source type and its parameters (such as gap, period, field strength etc. for an undulator). In the Monte Carlo simulations, the six coordinates are for efficiency reasons in general picked from another distribution: $f_{\mathrm{MC}}(E, \mathbf{\Omega}, \mathbf{r}) \neq \Pi(E, \mathbf{\Omega}, \mathbf{r})$, since one would *e.g.* often generate only x-rays within a certain parameter interval. However, we must then require that the weights are adjusted by a factor π_j (in this case: $j = 1$) so that

$$f_{\mathrm{MC}}\pi_j = P. \tag{2.9}$$

This probability rule is general, and holds also if, e.g., it is decided to transmit only half of the rays ($f_{\mathrm{MC}} = 0.5$). An important different example is elastic scattering from a powder sample, where the Monte-Carlo choices are the particular powder line to scatter from, the scattering position within the sample and the final x-ray direction within the Debye-Scherrer cone.

2.3.1. Direction focusing

An important application of weight transformation is direction focusing. Assume that the sample scatters the x-rays in many directions. In general, only x-rays in some of these directions will stand any chance of being detected. These directions we call the *interesting directions*. The idea in focusing is to avoid wasting computation time on x-rays scattered in the other directions. This trick is an instance of what in Monte Carlo terminology is known as *importance sampling*.

If *e.g.* a sample scatters isotropically over the whole 4π solid angle, and all interesting directions are known to be contained within a certain solid angle interval $\Delta\mathbf{\Omega}$, only these solid angles are used for the Monte Carlo choice of scattering direction. According to Eq. (2.9), the weight factor will then have to be changed by the amount $\pi_j = |\Delta\mathbf{\Omega}|/(4\pi)$. One thus ensures that the mean simulated intensity is unchanged during a "correct" direction focusing, while a too narrow focusing will result in a lower (*i.e.* wrong) intensity, since we cut x-rays rays that should have reached the final detector.

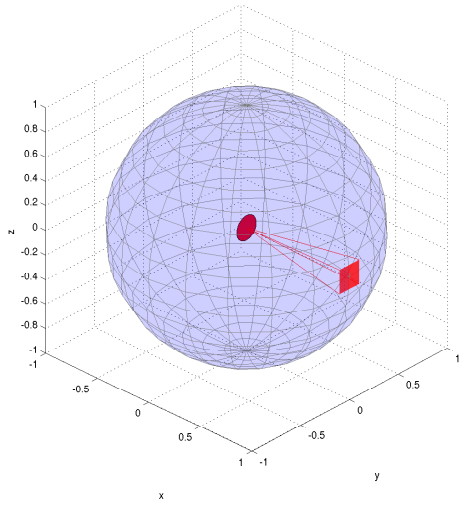


Figure 2.1.: Illustration of the effect of direction focusing in McXtrace. Weights of x-rays emitted into a certain solid angle are scaled down by the full unit sphere area.

2.4. Stratified sampling

One particular efficiency improvement technique is the so-called *stratified sampling*. It consists in partitioning the event distributions in representative sub-spaces, which are then all sampled individualy. The advantage is that we are then sure that each sub-space is well represented in the final integrals. This means that instead of shooting N events, we define D partitions and shoot $r = N/D$ events in each partition. We may define partitions so that they represent 'interesting' distributions, e.g. from events scattered on a monochromator or a sample. The sum of partitions should equal the total space integrated by the Monte Carlo method, and each partition must be sampled randomly.

In the case of McXtrace, the stratified sampling is used when repeating events, i.e. when using the SPLIT keyword in the TRACE section on beamline descriptions. We emphasize here that the number of repetitions r should not exceed the dimensionality of the Monte Carlo integration space (which is $d = 10$ for x-ray events) and the dimen-

Records	Accurarcy
10^3	10 %
10^4	2.5 %
10^5	1 %
10^6	0.25 %
10^7	0.05 %

Table 2.1.: Accuracy estimate as a function of the number of statistical events used to estimate an integral with McXtrace.

sionality of the partition spaces, i.e. the number of random generators following the stratified sampling location in the beamline.

2.5. Accuracy of Monte Carlo simulations

When running a Monte Carlo, the meaningful quantities are obtained by integrating random events into a single value (e.g. flux), or onto an histogram grid. The theory [Jam80] shows that the accuracy of these estimates is a function of the space dimension d and the number of events N. For large numbers N, the central limit theorem provides an estimate of the relative error as $1/\sqrt{N}$. However, the exact expression depends on the random distributions.

McXtrace uses a space with $d = 12$ parameters to describe x-rays (position, wavevector, weight, polarisation, phase, time). We show in Table 2.1 a rough estimate of the accuracy on integrals as a function of the number of records reaching the integration point. This stands both for integrated flux, as well as for histogram bins - for which the number of events per bin should be used for N.

3. Source components

McXtrace contains a number of different source components, and any simulation will usually contain exactly one of these sources. The main function of a source is to determine a set of initial parameters $(\mathbf{r}, \mathbf{k}, t)$ for each photon ray. This is done by Monte Carlo choices from suitable distributions. For example, in most present sources the initial position is found from a uniform distribution over the source surface. The initial photon wavenumber is selected within an interval of either the corresponding energy or the corresponding wavelength.

For pulsed sources, the choice of the emission time, t, is being made on basis of detailed analytical expressions. For other sources, t is set to zero. In the case one would like to use a steady state source with time-of-flight settings, the emission time of each photon ray should be determined using a Monte Carlo choice. This may be achieved by the `EXTEND` keyword in the instrument description source as in the example below:

```
TRACE

COMPONENT MySource=Source_pt(...) AT (...)
EXTEND
%{
  t = 1e-3*randpm1(); /* set time to +/- 1 ms */
%}
```

Also take a look at the **Chopper_simple** component.

3.0.1. Photon flux and Brilliance

The flux of the sources deserves special attention. The total intensity is defined as the sum of weights of all emitted x-rays during one simulation (the unit of total photon weight is thus photons per second). The flux, ψ, at an instrument is defined as intensity per area perpendicular to the beam direction.

The source Brilliance, Φ, is defined in different units (See e.g. [ANM11]): the number of photon rays emitted per second from a 1 m^2m area on the source surface, with direction within a 1 m^2rad angle window, and with wavelength within a 1% interval. The total intensity of real photons emitted towards a given diaphragm (units: ph./s) is therefore (for constant Φ):

$$I_{\text{total}} = \Phi A \Delta\Omega \Delta\lambda, \tag{3.1}$$

where A is the source area, $\Delta\Omega$ is the solid angle of the diaphragm as seen from the source surface, and $\Delta\lambda$ is the width of the wavelength interval in which photons are emitted (assuming a uniform wavelength spectrum).

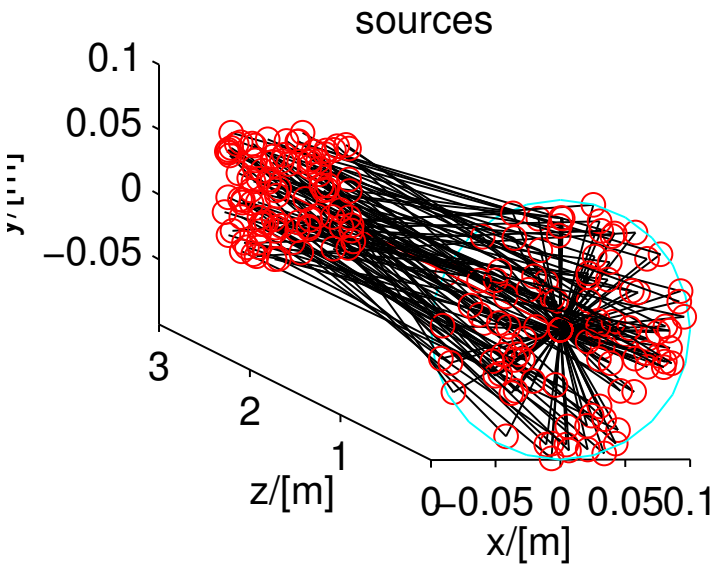


Figure 3.1.: A circular source component (at z=0) emitting photon rays randomly, either from a model, or from a data file.

The simulations are performed so that detector intensities are independent of the number of photon histories simulated (although more photon histories will give better statistics). If N_{sim} denotes the number of x-ray histories to simulate, the initial photon weight p_0 must be set to

$$p_0 = \frac{N_{\text{total}}}{N_{\text{sim}}} = \frac{\Phi(\lambda)}{N_{\text{sim}}} A\Omega\Delta\lambda, \tag{3.2}$$

where the source brilliance is now given a λ-dependence.

As a start, we recommend new McXtrace users to use the **Source_flat** component. For a slightly more realistic sources are supply **Source_flat** with a spectrum file (for instance generated by SPECTRA [TK01]) or **Source_gaussian**.

Optimizers can dramatically improve the statistics, but may occasionally give wrong results, due to misleaded optimization. You should always check such simulations with (shorter) non-optimized ones.

Other ways to speed-up simulations are to read events from a file. See section 3.6 for details.

3.1. The `Source_pt` McXtrace Component

Release: McXtrace 0.1
An x-ray point source

Identification

- **Author:** Erik Knudsen
- **Origin:** Risoe
- **Date:** June 29th, 2009

Description

A simple source model emitting photons from a point source uniformly into 4pi. A square target centered on the Z-axis restricts the beam to that aperture. If an input spectrum datafile (spectrum_file) is not specified, the beam is restricted to emit photons between E0+-dE keV, or lambda0+-dlambda Å, whichever is given. The input spectrum file should be formatted such that x-ray energy/wavelength is in the first column and the intensity in the second. Any preceding lines starting with # are considered part of the file header. If a datafile is given, a nonzero E¤0 value indicates that is is parametrized by energy (in keV) as opposed to wavelength (in Å). Wavelength is the default. Flux is given in the unit photons/s

Example: Source_pt(dist=1,focus_xw=0.1,focus_yh=0.1, lamda=0.231, dlambda=0.002)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
focus_xw	m	Width of target	0
focus_yh	m	Height of target	0
focus_x0	m	x-cocordinate of target centre.	0
focus_y0	m	y-coordinate of target centre.	0
flux	ph/s	Total flux radiated from the source.	0
dist	m	Distance from source plane to sampling window.	1
E0	keV	Mean energy of xrays.	0
dE	keV	Energy half spread of x-rays.	0
lambda0	Å	Mean wavelength of x-rays.	0
dlambda	Å	Wavelength half spread of x-rays (flat or gaussian sigma).	0
phase	rad	Set phase to something given.	0

Name	Unit	Description	Default
randomphase	0/1	If nonzero, the phase of the emotted photon is random, i.e. source is fully incoherent. otherwise the value of phase is used.	1
gauss	1	Gaussian (1) or Flat (0) energy/wavelength distribution	0
spectrum_file	string	File from which to read an input spectrum.	""
verbose	1	Output more information runtime.	0

Links

- Component source code found in file `Source_pt.comp`.

A mathematical point emitting photons with a spectrum either uniform, gaussian or generated from a datafile

The simplest source model, where a mathematical point source at $(0, 0, 0)$ emits photons. The wavevector of the emitted photons is picked randomly in a defining aperture *focus_xw* by *focus_yh* m at $(0, 0, dist)$. Please note that this aperture is merely a virtual aperture used to reduce the sampling space. This has a few implications: Other components may be placed without reference to the aperture, but if the aperture does not fill the full acceptance window of subsequent components your simulations will be biased. The aperture is simply there to provide efficient sampling.

If a *spectrum_file* is not supplied, the xray is given a weight which is the total wavelength-integrated intensity downscaled by the solid angle subtended by the defining aperture. The Energy/wavelength spectrum of the emitted photons is centered around *E0* or $\lambda 0$ with a width of *dE* or $d\lambda$ respectively. *E0* takes precedence over $\lambda 0$. Thus if $E0 \neq 0$ the the combination *E0, dE* is used, $\lambda 0, d\lambda$ otherwise. If *gauss* (the default) is set to be non-zero the spectrum is Gaussian with mean *E0* ($\lambda 0$) and standard deviation *dE* ($d\lambda$), otherwise the spectrum is uniform in $E0 \pm dE$ ($\lambda 0 \pm d\lambda$).

If a *spectrum_file is* supplied, a slightly different strategy is adopted. In this case the wavelength/energy range implied by the datafile is sampled unformly and each ray is assigned a weight corresponding to the intensity indicated by linear interpolation between datapoints at that wavelength. This implies an oversampling of weak parts of the intensity spectrum.

Currently only completely coherent or fully incoherent beams are supported. If *randomphase* is specified emitted photons will be assigned a random phase, otherwise it is set to the value of *phase*.

3.2. The `Source_flat` McXtrace Component

Release: McXtrace 0.1_alpha

A flat rectangular or circular surface emitting x-rays

Identification

- **Author:** Erik Knudsen
- **Origin:** Risoe
- **Date:** September 25, 2009

Description

A circular or rectangular xray source. Spectrum may be either gaussian or uniform around a central wavelength/energy or read from a datafile. Xrays are considered emitted uniformly into 4pi, but a square target retricts the beam to that window and scales the beam intensity accordingly. If an input spectrum datafile (spectrum_file) is not specified, the beam is restricted to emit photons between E0+-dE keV, or lambda0+-dlambda Å, whichever is given. The input spectrum file should be formatted such that x-ray energy/wavelength is in the first column and the intensity in the second. Any preceding lines starting with # are considered part of the file header. If a datafile is given, a nonzero E0 value indicates that is is parametrized by energy (in keV) as opposed to wavelength (in Å). Wavelength is the default. Flux is set in the unit photons/s

Example: Source_flat(xwidth=1e-3,yheight=1e-3, focus_xw=0.5e-2, focus_yh=0.45e-2,dist=1, E0=E0, dE=DE)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	m	Radius of circle in (x,y,0) plane where x-rays are generated.	0
yheight	m	Height of rectangle in (x,y,0) plane where x-rays are generated.	0
xwidth	m	Width of rectangle in (x,y,0) plane where x-rays are generated. Overrides xmin and xmax.	0
xmin	m	Lower bound of x-interval where photons are generated.	0
xmax	m	Upper bound of x-interval where photons are generated.	0
dist	m	Distance to target along z axis.	0

Name	Unit	Description	Default
focus_xw	m	Width of target	.045
focus_yh	m	Height of target	.12
E0	keV	Mean energy of xrays.	0
dE	keV	Energy half spread of x-rays (flat or gaussian sigma).	0
lambda0	Å	Mean wavelength of x-rays.	0
dlambda	Å	Wavelength half spread of x-rays.	0
flux	pht/s	Total flux radiated from the source	0
gauss	1	Gaussian (1) or Flat (0) energy/wavelength distribution	0
randomphase		If nonzero, the phase of the emotted photon is random, i.e. source is fully incoherent. otherwise the value of phase is used.	1
phase	rad	Set phase to something given.	0
spectrum_file	string	Filename for optional spectrum-file	""

Links

- Component source code found in file `Source_flat.comp`.

A flat surface emitting photons with a spectrum either uniform, gaussian or generated from a datafile

A simple source model, with a flat surface emitting photons. The surface in the xy-plane is specified as a rectangle with dimensions $xwidth \times yheight$ m^2, or as a circle with *radius* m. The initial x-ray position is chosen randomly in the source surface — its wavevector is chosen randomly (exactly as in the case of **Source_pt**) (section 3.1) in the defining aperture with height *focus_yh* and width *focus_xw* placed at $(0, 0, dist)$.

Just as for **Source_pt** the aperture is for efficiency purposes and, if misused, may cause biasing.

With the exception of source size related parameters, all other parameters are identical to **Source_pt** Note that this also applies to coherence and photon phase. If *randomphase*= 0 then a photon emitted from $(x_0, y_0, 0)$ will in phase with a photon emitted from $(x_1, y_1, 0)$. I.e. full transversal coherence.

3.3. The `Source_div` McXtrace Component

Release: McXtrace 0.1

X-ray source with Gaussian or uniform divergence

Identification

- **Author:** Erik Knudsen
- **Origin:** Risoe
- **Date:** November 11, 2009

Description

A flat rectangular surface source with uniform or Gaussian divergence profile and focussing. If the parametere gauss is not set (the default) the divergence profile is flat in the range [-focus_ax,focus_ay]. If gauss is set, the focux_ax,focus_ay is considered the standard deviation of the gaussian profile. Currently focussing is only active for flat profile. The "focus window" is defined by focus_xw,focus_yh and dist. The spectral intensity profile is uniformly distributed in the energy interval defined by e0+-dE/2 or by wavelength lambda0+-dlambda/2

Example: Source_div(xwidth=0.1, yheight=0.1, focus_aw=2, focus_ah=2, E0=14, dE=2, gauss=0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
spectrum_file	string	File from which to read the spectral intensity profile	"NULL"
xwidth	m	Width of source.	0
yheight	m	Height of source.	0
dist	m	Downstream distance to place sampling target window	0
focus_xw	m	Width of sampling window	0
focus_yh	m	Height of sampling window	0
focus_aw	rad	Standard deviation (Gaussian) or maximal (uniform) horz. width divergence.	0
focus_ah	rad	Standard deviation (Gaussian) or maximal (uniform) vert. height divergence.	0
focus_ar	rad	Standard deviation (Gaussian) or maximal (uniform) radial divergence.	0
radius	m	Radius of circular source	0
E0	keV	Mean energy of X-rays.	0
dE	keV	Energy half spread of X-rays. If gauss==0 dE is the half-spread, i.e. E\in[E0-dE,E0+dE], if gauss!=0 it's interpreted as the standard dev.	0

Name	Unit	Description	Default
lambda0	Å	Mean wavelength of X-rays (only relevant for E0=0).	0
dlambda	Å	Wavelength half spread of X-rays.	0
flux	1/(s * mm**2 *mrad**2 * energy unit)	flux per energy unit, Å or keV.	0
gauss	1	Criterion: 0: uniform, 1: Gaussian distribution of energy/wavelength.	0
gauss_a	1	Criterion: 0: uniform, 1: Gaussian divergence distribution.	0
randomphase	1	If !=0 the photon phase is chosen randomly.	1
phase	1	Value of the photon phase (if randomphase==0).	0
verbose	0/1	Show more information	0

Links

- Component source code found in file `Source_div.comp`.

A continuous source with specified divergence

Source_div is a rectangular source, $w \times h$, which emits a beam of a specified divergence around the direction of the z-axis.

Just as for **Source_pt**, if a *spectrum_file* is not supplied, the xray is given a weight which is the total wavelength-integrated intensity downscaled by the solid angle subtended by the defining aperture and the energy/wavlength spectrum is centered around *E0* or $\lambda 0$ with width dE or $d\lambda$ respectively. The profile is Gaussian if $gauss \neq 0$, uniform if $gauss = 0$, If a *spectrum_file is* supplied, a slightly different strategy is adopted. In this case the wavelength/energy range implied by the datafile is sampled unformly and each ray is assigned a weight corresponding to the intensity indicated by linear interpolation between datapoints at that wavelength. This implies an oversampling of weak parts of the intensity spectrum.

The beam intensity is uniform over the whole of the source and the source divergences are *focus_ah* and *focus_aw* in degrees. Unless $gauss_a \neq= 0$ the individual photons are sampled from a uniform divergence distribution, otherwise a Gaussian is used.

3.4. The `Source_gaussian` McXtrace Component

Gaussian cross-section source

Identification

- **Author:** Jana Baltser & Erik Knudsen
- **Origin:** NBI
- **Date:** April, 2011.

Description

A simple source model emitting photons from a gaussian distribution in the X-Y plane with the specified standard deviations and divergence. A square target centered on the beam (Z-axis) may be used to restrict the beam to that aperture. If no target aperture is given the full gaussian cross-section is used. Further, the beam is restricted to emit photons between E0+-dE keV, or lambda0+-dlambda, whichever is given, if a spectrum_file is not specified, in which case the contents of the file dictates the emitted spectrum.

Example: Source_gaussian(sig_x=10e-6,sig_y=10e-6,dist=15,sigPr_x=9e-6, sigPr_y=9e-6,E0=12.5, dE=0.1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
spectrum_file		File from which to read the spectral intensity profile	"NULL"
sig_x	m	Horizontal standard deviation of source (rms source size).	1
sig_y	m	Vertical standard deviation of source (rms source size).	0
sigPr_x	rad	Angular horizontal divergence	0
sigPr_y	rad	Angular vertical divergence Optional Parameters:	0
flux		Scaling factor to set the total emitted unrestricted flux.	1
brilliance		Unit in spectrum_file is Brilliance - apply corrections to get to raw flux.	0
dist	m	Distance from source plane to sampling window.	1
gauss	0/1	Gaussian (1) or uniform (0) spectrum profile.	0
focus_xw	m	Width of sampling window dist m downstream from source to allow focused sampling.	0

Name	Unit	Description	Default
focus_yh	m	Height of sampling window dist m downstream from source to allow focused sampling.	0
E0	keV	Centre of emitted energy spectrum (overrides spectrum_file)	0
dE	kev	Half-width (or std. dev.) of emitted energy spectrum.	0
lambda0	Å	Centre of emitted wavelength spectrum.	0
dlambda	Å	Half-width (or std. dev.) of emitted wavelength spectrum.	-1
phase	rad	The initial phase of the photons.	0
randomphase	rad	If nonzero phase is random (incoherent radiation), otherwise it is set to the value of phase	1

Links

- Component source code found in file `Source_gaussian.comp`.

the model has a gaussian distribution of intensity

A simplified version of a completely incoherent source of horizontal and vertical sizes *sig_x* and *sig_y* respectively with angular divergence *sigPr_x* and *sigPr_y*. Can be used to model an undulator source emitting a photon beam that has gaussian distribution. This naturally generates a larger Gaussian profile at a distance. A sampling window at $(0, 0, dist)$ may be specified by the parameters *focus_xw* and *focus_yh*, which restricts the sampling phases space of the emitted photons (See section 3.1 for details).

The energy spectrum emitted by **Source_gaussian** source is completely analogous to **Source_pt**, as is its photon phase functionality.

3.5. The `Source_lab` McXtrace Component

Laboratory x-ray source, using xraylib for all material data.

Identification

- **Author:** Erik B. Knudsen (original Source_lab), xraylib port derived from it
- **Origin:** Kgs. Lyngby
- **Date:** May 2012 (original), 2026 (xraylib version)

Description

Model of a laboratory x-ray tube, generating x-rays by bombarding a target by electrons. Given a input energy E0 of the electron beam, x-rays are emitted from the accessible emission lines The geometry of the tube is assumed to be: # The electron beam hits a slab of surface material surface at a right angle illuminating an area of width by height, # where width is measured along the component X-axis. # The centre of the electron beam at the anode surface is the origin of the component. # The Z-axis of the component points at the centre of the exit window (focus_xw by focus yh) placed at a distance dist from the origin. # The angle between the Z-axis and the anode surface is the take_off angle. For a detailed sketch of the geometry see the component manual.

The Bremsstrahlung emitted is modelled using the model of Kramer (1923) as restated in International Tables of Crystallography C 4.1

Characteristic radiation (K-lines Ka1, Ka2, Kb1, Kb3) is modelled by Lorentzian (default) or Gaussian energy profiles. Line energies, level widths (FWHM = K level width + final level width) and relative intensities (radiative rates, strongest line = 1) are obtained from xraylib for ANY mono-atomic anode; no datafile is needed. Lines that fall outside [Emin,Emax] are dropped.

Absorption of emitted x-rays while travelling through the target anode is included, using the xraylib total mass attenuation coefficient (tabulated once in INITIALIZE on a log grid, with extra points on both sides of every absorption edge, and interpolated log-log in TRACE).

Example: Source_lab(material="Cu",Emin=1, E0=80) Legacy: Source_lab(material_datafile="Cu.txt",Emin= E0=80)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
material	string	Anode element, symbol ("Cu", "W", ...) or atomic number as a string ("29").	"Cu"
width	m	Width of electron beam impinging on the anode.	1e-3
height	m	Height of electron beam impinging on the anode.	1e-3
thickness	m	Thickness of the anode material slab.	100e-6
E0	kV	Acceleration voltage of xray tube.	20
Emax	keV	Maximum energy to sample. Default (Emax=0) is to set it to E0.	0
Emin	keV	Minimum energy to sample.	1
focus_xw	m	Width of exit window.	5e-3
focus_yh	m	Height of exit window.	5e-3
take_off	deg	Take off angle of beam centre.	6

Name	Unit	Description	Default
dist	m	Distance between centre of illuminated target and exit window.	1
tube_current	A	Electron beam current.	1e-3
frac	0-1	Fraction of statistic to use for Bremsstrahlung.	0.1
lorentzian	0/1	If nonzero Lorentzian (more correct) line profiles are used.	1
xwidth	m	Width of the anode material slab (0: practically infinite).	0
yheight	m	Height of the anode material slab (0: practically infinite).	0
exit_window_refpt	m	If set, the AT position and exit window will coincide (legacy behaviour).	0
material_datafile	string	Legacy parameter, kept for backward compatibility. Only the file NAME is used: the element is taken from it ("Cu.txt", "../data/Mo.txt" -> "Cu", "Mo"); the file is not read. If set, it overrides 'material'.	""

Links

- Component source code found in file `Source_lab.comp`.

X-ray tube laboratory source

Source_lab is a model of a laboratory X-ray tube. An electron ray hits a target of specified material. Currently, only single material targets are allowed. To model multiple material targets one could construct a model with two or more sources simultaneously. This has consequences for intensity of the source which should be downscaled accordingly.

An electron beam of rectangular transverse crossection (*width,height*) and energy *E0* impinges on the target of material. Wrt. the electron beam, the target is considereded infinitely thick. The beam is considered to have uniform intenisty. Thus, the spatial distribution of x-ray generation will be exponential in the depth of the material.

Further, an exit aperture is defined with dimensions (*focus_xw,focus_yh*). The centre of the aperture is situated at a distance *dist* m from where the electron beam hits the target slab at an elevation of *take_off* (see Figure 3.5).

The **Source_lab** coordinate system has its origin in the center of the elctron beam at the surface of the anode material and is oriented such that the z-axis points at the center of the exit window, and the x-axis is parallel to the *width* of the electron beam.

Note that the exit aperture is merely an opening. If the material absorption of a window, e.g. Be, is to be taken into account a **Filter** (section 4.6) should be inserted

after the exit aperture.

For each photon ray to be generated, a Monte Carlo choice is made to generate either a Bremsstrahlung photon or one from one of the x-ray emission lines of the material. $(1 - frac)$ of the rays are generated from characteristic emission, and $(frac)$ from Bremsstrahlung. In most cases Bremsstrahlung is unwanted background, which is why the default is 0.1. Note that this *only* governs how much of the available statistics is diverted into simulating Bremsstrahlung. It does not have an impact on what intensity is detected in subsequent monitors — only on the errorbars of the detected numbers.

The spectral characteristics of the generated Bremsstrahlung is goverened by the model suggested by Kramer [Kra23]. Although disputed in several subsequent papers, the model is simple, and sufficiently accurate for many background estimation purposes.

Characteristic emission on the other hand is sampled from a set of Lorentzian functions with central wavelengths found in the work by [Bea67] with spectral widths taken from [KO79].

An example of beam spectral characteristics emitted from a Cu-anode target detected 1 mm from an exit aperture of 1×1 cm^2 10 cm downstream from the target at a *take_off* angle of 6° is seen in figure 3.5.

Source_lab includes a set of common anode materials: {Cr, Co, Cu, Ga, Mo, Ag, W}. More materials can be added by the following procedure:

1. Find the atomic number, Z, for the material you want to add. So far only single materials are supported.

2. Look up (and note down) the central energies of (up to) 6 characteristic lines for the material in e.g. [Bea67]. Also note down the number lines you have recorded.

3. Look up the natural spectral widths of those lines in [KO79].

4. Find the relative intensity of the the set of lines. For instance from the x-ray data booklet.

5. Note down the ionization energy $E_I(Z)$, and flourescence yield, $Y(Z)$, of the anode material.

With this information assemble a code line as:
$\{Z, E_I(Z), Y(Z), n, \{[E_C]\}, \{[W]\}, \{[I]\}\}$,
and put it in the source file `Source_lab.comp`, just above the line that reads
`{0,0.0,0.0,0,{0,0,0,0,0,0},{0,0,0,0,0,0},{0,0,0,0,0,0}}`
in the `SHARE` section of the component source code. Here $[E_C]$ refers to a comma separated list of characteristic energies in keV, $[W]$ a list of characteristic widths in keV, and $[I]$ a list of relative line intensities. Lastly, n denotes the number of x-ray lines.

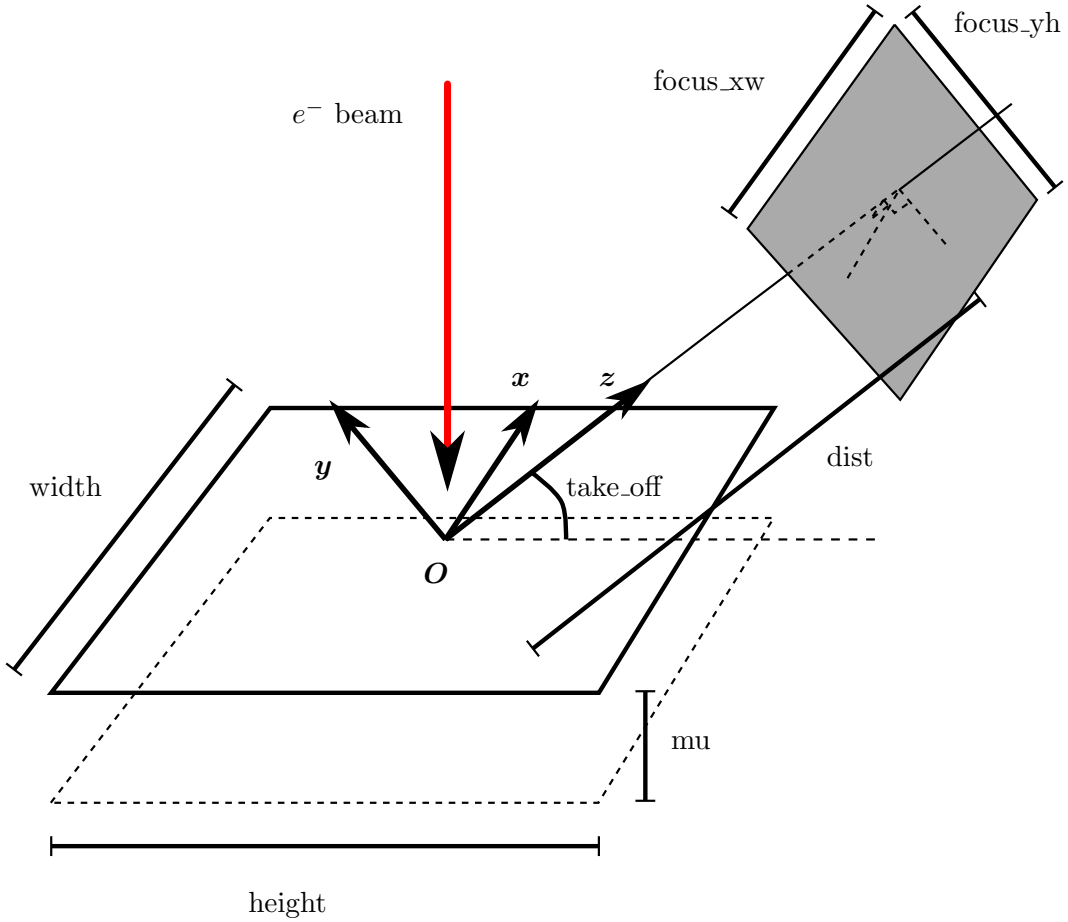


Figure 3.2.: Geometry of the **Source_lab** component. An electron beam impinges at a right angle on an anode material, where X-rays are generated. The Origin is defined to be a the centre of the electron beam on the anode surface, and the coordinate system is oriented such that the z-axis point towards the exit aperture.

Figure 3.3.: Intensity vs. wavelenghth for a Cu-anode laboratory source.

3.6. Other sources components: virtual sources (event files)

4. Beam optical components: Arms, slits, filters etc.

This chapter contains a number of optical components used to modify the x-ray beam in various ways, as well as the "generic" component **Arm**.

4.1. The `Arm` McXtrace Component

Arm/optical bench

Identification

- **Author:** Kim Lefmann and Kristian Nielsen
- **Origin:** Risoe
- **Date:** September 2009

Description

An arm does not actually do anything, it is just there to set up a new coordinate system.
Example: Arm()

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default

Links

- Component source code found in file `Arm.comp`.

The generic component

The component **Arm** is empty; is resembles an optical bench and has no effect on the x-ray. The purpose of this component is only to provide a standard means of defining a local coordinate system within the instrument definition. Other components may then be positioned relative to the **Arm** component using the McXtrace meta-language. The

use of **Arm** components in beamline definitions is not required but is recommended for clarity. **Arm** has no input parameters.

The first Arm instance in an instrument definition may be changed into a **Progress_bar** (sec. 9.1) component in order to display simulation progress on the fly, and possibly save intermediate results.

4.2. The `Slit` McXtrace Component

Rectangular/circular slit

Identification

- **Author:** Erik Knudsen
- **Origin:** DTU Physics
- **Date:** June 16, 2009

Description

Based on Slit-comp by Kim Lefmann and Henrik Roennow A simple rectangular or circular slit. You may either specify the radius (circular shape), which takes precedence, or rectangular bounds. No transmission around the slit is allowed.

Example: Slit(xmin=-0.01, xmax=0.01, ymin=-0.01, ymax=0.01) Slit(radius=0.01)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
xmin	m	Lower x bound.	0
xmax	m	Upper x bound.	0
ymin	m	Lower y bound.	0
ymax	m	Upper y bound.	0
radius	m	Radius of slit in the z=0 plane, centered at Origin.	0
xwidth	m	Width of slit. Overrides xmin,xmax.	0.001
yheight	m	Height of slit. Overrides ymin,ymax.	0.001
dist	m	Distance from slit plane to plane containing resampling target.	0
focus_xw	m	Width of resampling window.	0
focus_yh	m	Height of resampling window.	0
focus_x0	m	Centre (x) of resampling window.	0
focus_y0	m	Centre (y) of resampling window.	0

Links

- Component source code found in file `Slit.comp`.

A beam defining diaphragm

The component **Slit** is a very simple construction. It sets up an opening at $z = 0$, and propagates the photon rays onto this plane (by the kernel call PROP_Z0). Photons within the slit opening are unaffected, while all other photons are discarded by the kernel call ABSORB.

By using **Slit**, some photons contributing to the background in a real experiment will be neglected. These are, for instance, the ones that scatter off the inner side of the slit, penetrate the slit material, or clear the outer edges of the slit.

The opening of the slit is determined by specifying either a *radius*, a width *(xwidth)* and a height *(yheight)*, or absolute limits in x and y *(xmin, xmax, ymin, ymax)*, in order of precedence. If *radius* is set, the opening is considered circular.

Slit may be used to model slit diffraction, thourgh judicious use of the resmapling parameters: *focus_xw,focus_yh, focus_x0, focus_y0*. The similarity in parameter naming for sources (chapter 3) is intentional as computationally the processes are the same. If present the focus parameters cause the ray to be regarded as a Huygens wave and a new direction going towards the resampling window is computed. The resulting difference in phase gives rise to maxima and minima as expected. This feature should be used with care: the further away from the optical axis, the more statistics and finer binning is needed to get meaningful results. Please refer to [BK+13] for more deatils.

4.3. The `Slit_N` McXtrace Component

Release: McXtrace 0.1

Rectangular/circular slit, duplicated

Identification

- **Author:** Erik Knudsen
- **Origin:** Risoe
- **Date:** June 16, 2009

Description

Based on Slit-comp by Kim Lefmann and Henrik Roennow A simple rectangular or circular slit. You may either specify the radius (circular shape), which takes precedence, or the rectangular bounds. The slits are separated with 'd', and arranged along X. No transmission around the slit is allowed. If cutting option is used, low-weight x-rays are ABSORBED

Example: Slit_N(xwidth=0.01, yheight=0.01, d=0.02) Slit_N(radius=0.01, cut=1e-10, d=0.03)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	m	Radius of slit in the z=0 plane, centered at Origo	0
cut		Lower limit for allowed weight (1)	0
xwidth	m	Width of slit. Overrides xmin,xmax.	0
yheight	m	Height of slit. Overrides ymin,ymax.	0
N	1	Number of slit openings along X	2
d	m	Separation of slits	0

Links

- Component source code found in file `Slit_N.comp`.

multiple slits

Documentation pending.

4.4. The `Beamstop` McXtrace Component

Rectangular/circular beam stop.

Identification

- **Author:** Kristian Nielsen
- **Origin:** Risoe
- **Date:** March 2011

Description

A simple rectangular or circular beam stop. Infinitely thin and infinitely absorbing. The beam stop is by default rectangular. You may either specify the radius (circular shape), or the rectangular bounds.

Example: Beamstop(xmin=-0.05, xmax=0.05, ymin=-0.05, ymax=0.05) Beamstop(radius=0.1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
xmin	m	Lower x bound	-0.05
xmax	m	Upper x bound	0.05
ymin	m	Lower y bound	-0.05
ymax	m	Upper y bound	0.05
xwidth	m	Width of beamstop (x). Overrides xmin,xmax.	0
yheight	m	Height of beamstop (y). Overrides ymin,ymax.	0
radius	m	Radius of the beam stop in the z=0 plane, centered at Origo	0

Links

- Component source code found in file `Beamstop.comp`.

A photon absorbing area

The component **Beamstop** can be seen as the reverse of the **Slit** component. It sets up an area at the $z = 0$ plane. Photons that hit the plane within this area are ABSORB'ed, while all others are unaffected.

By using this component, some photons contributing to the background in a real experiment will be neglected. These are the ones that scatter off the side of the (real) beamstop, or penetrate the absorbing material. Further, the holder of the beamstop is not simulated.

Beamstop can be either circular or rectangular. The input parameters of **Beamstop** are either height and width *(xwidth, yheight)* or the four coordinates,*(xmin, xmax, ymin, ymax)* defining the opening of a rectangle, or the *radius* of a circle, depending on which parameters are specified.

If the "direct beam" (e.g. after a monochromator or sample) should not be simulated, it is possible to emulate an ideal beamstop so that only the scattered beam is left; without the use of **Beamstop**: This method is useful for instance in the case where only photons scattered from a sample are of interest. The example below removes the direct beam and any background signal from other parts of the beamline

```
COMPONENT MySample=PowderN(...) AT (...)
EXTEND
%{
  if (!SCATTERED) ABSORB;
%}
```

4.5. The `Mask` McXtrace Component

A masking image object

Identification

- **Author:** Erik Knudsen
- **Origin:** DTU Physics
- **Date:** March 2014

Description

The Mask component takes an image as input either as an standard image file in png or pnm format, or as an ascii file (see below for format), and uses the image as a mask. For instance as a manner of measuring resolution for imaging applications. If the image is supplied as a png or pnm file, they interpretation of the pixels varies depending on the file. If the image is grayscale the pixel values are directly mapped to opacity (or transparency if invert is set) values in the range [0..1]. If the image has RGB channels the R channel is considered most significant, the B channel least significant. The resulting number, e.g. R*255^2 + G*255 + B, is then mapped to a real valued opacity. Additionally png images may have an alpha channel - which is then considered the least significant channel. Palette mapped pngs are as of yet _not_ supported. A regular ascii file may be supplied - in which case the file is like the one below #any initial line starting with a hash is silently ignored

```
0.0 1.0 0.0 1.0 0.0
0.5 0.0 0.5 0.0 0.5
0.0 0.25 0.0 0.25 0.0
0.75 0.0 0.75 0.0 0.75
1.0 0.0 1.0 0.0 1.0
```

...which defines a 5x5 mask with a kind of checkerboard pattern.

By default the values from the masking image are interepreted as opacity (1 is fully blocking). If invert is nonzero this is inverted and the values are considered as transparency (1 is fully transmissive)

N.b. If you want to use the png-option of the component you must have libpng installed _and_ link your compiled instrument to it. Assuming libpng is installed you may do this by adding "-DUSE_PNG=1 -lpng" to 1) the MCXTRACE_CFLAGS environment variable or 2) to the compiler flags textbox in the GUI. Open File->Configuration and edit the textbox.

The virtual option of the Mask, is intended as a help to use a png-image as a grayscale distribution. If the virtual flag is set, rays are propagated to the mask plane and the pixel value at the intersection point is read, but the rays remain unaffected. The pixel

value is stored in the variable named in the string maskvar. This should be a USERVAR set from the instrument file.

Example: Mask(xwidth=0.1, yheight=0.1, mask=Test_Mask_input_file.mask)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
xwidth	m	Width of the masking object	0.1
yheight	m	Height of the masking object	0.1
mask	str	Name of file containing the masking image	"Test_Mask_input_file
invert	0/1	When 0 => masked values are opaque, when 1 => masked values are transparent.	0
virtual	0/1	Mask does not affect the x-ray, but does still read the pixel value of the pixel hit and stores it in masking.	0

Links

- Component source code found in file `Mask.comp`.

A masking object for imaging purposes

This component may take an image file as an inout parameter and uses it as a mask. That way commercial resolution test imaging may be uses in simulations. N.B. no resampling s implemented. That could be targeted using the contributed `Focus`-component.

4.6. The `Filter` McXtrace Component

Release: McXtrace 1.1

Block of an attenuating material

Identification

- **Author:** Erik Knudsen
- **Origin:** DTU Physics
- **Date:** Jan 24, 2011

Description

A chunk of attenuating material. Attenuation is computed through the effective length travelled within the material. No scattering is modelled at present.

Filter shape may be a cylinder, a sphere, a box or any other shape.

```
box/plate:       xwidth x yheight x zdepth
cylinder:        radius x yheight (along Y axis)
sphere:          radius
any shape:       geometry=OFF/PLY_file
```

Example: Filter(material_datafile="Ge.txt", geometry="wire.ply",xwidth=0.02,yheight=0,zdepth=0) Example: Filter(material_datafile="Ge.txt",xwidth=0.02,yheight=0.02, zdepth=1e-4) Example: Filter(material_datafile="Ge.txt",radius=1e-4,yheight=0.02) Example: Filter(material_datafile="Ge.txt",radius=1e-3, refraction=1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
refraction	0/1	If nonzero, refraction is enabled. Rays hitting the surface below the critical angle are reflected.	0
fixed_delta	0/1	Use a fixed delta to compute refraction - useful for debugging.	0
material_datafile	str	File where the material parameters for the filter may be found. Format is similar to what may be found off the NIST website. [Be.txt]	"Be.txt"
geometry	str	File containing the polygon definition of a general shape object. When xwidth is also given, the object is rescaled accordingly (OFF/PLY)	0
xwidth	m	Width of block.	0
yheight	m	Height of block.	0
zdepth	m	Thickness of block.	0
radius	m	Radius of cylinder or sphere.	0
mu_col	idx	Column index to pick up absorption length mu, counted from 0. Use with non-standard input file, e.g. 2-3 column files. -1 means attempt autodetect.	-1

Links

- Component source code found in file `Filter.comp`.
- Meshlab - viewer for OFF files
- Geomview and Object File Format (OFF) - OFF file definition and examples
- jroff.jar - Java version of Geomview (display only)
- Qhull - for calculating a convex hull from points.
- Powercrust - for reconstructing a solid geometry from a point cloud.
- Most material datafile inpus may obtained from NIST FFast

A general absorption filter model

This component is a filter in the shape of a rectangular block or a general shape defined by a set of polygons. Given an input file containing material parameters. Neccessary parameters are nominal density and a parameterization of the linear attenuation coefficient, μ as a function of wavelength (or energy).

The model is very simple: Firstly the X-ray is traced to find intersection points between ray and filter (0 or 2). If no intersection is found the x-ray is left untouched and nothing further happens. Assuming the ray intersects the filter: Secondly, the path length dl within the filter is computed. Thirdly a $\mu = f(\lambda, \text{material})$ is computed by interpolating in a datafile, and the x-ray weight is adjusted according to $p = p \exp(-\mathrm{d}l * \mu)$. The x-ray is left at the point where it exits the filter block (the 2nd intersection).

Example data files corresponding to all elements up to $Z = 92$ are distributed with McXtrace in the `MCXTRACE/data` directory as `*.txt` files. These tables have been extracted from the NIST FFAST [Nis] x-ray database. To generate other datafiles from the same source a simple shell script: `MCXTRACE/data/get_xray_db_data` is also distributed with McXtrace Running this script will connect to the NIST webiste and download a `.html` file. This output must now be modified such that `html`-tags are removed and all header lines begin with #

4.6.1. Example

This is an example of how to download and generate datafiles for the `Filter.comp` and others.

The distributed tables have been extracted from the NIST x-ray database. To ease generation of more dtafiles from the same source a simple shell script:
`MCXTRACE/data/get_xray_db_data`
is also distributed with McXtrace

Running this script will connect to the NIST webiste and download a `.html` file. This output must now be modified wuch that `html`-tags are removed and all header lines begin with #.

```
/usr/local/lib/mcxtrace/data/get_xray_db_data 3 output.dat
```

where the second parameter (3) is the atom number of the material, for which we want to generate a datafile. Now open the generated datafile (`output.dat`) with your favourite text editor and make sure the file ends up looking like this

```
#Li (Z 3)
#Atomic weight: A[r]  6.941000
#Nominal density: rho 5.3300E-01
#    rho[a](barns/atom) = [mu/rho](cm^2 g^-1)  x  1.15258E+01
#    E(eV) [mu/rho](cm^2 g^-1) = f[2](e atom^-1)  x  6.06257E+06
#    2 edges. Edge energies (keV):
#
#
#    K      5.47500E-02  L I    5.34000E-03
#
#Relativistic correction estimate f[rel] (H82,3/5CL) = -9.8613E-04,
#    -6.0000E-04 e atom^-1
#    Nuclear Thomson correction f[NT] = -7.1131E-04 e atom^-1
#
#-------------------------------------------------------------------------------
#Form Factors, Attenuation and Scattering Cross-sections
#Z=3, E = 0.001 - 433 keV
#
#    E        f[1]         f[2]        [mu/rho]    [sigma/rho]  [mu/rho]   [mu/rho][K] lambda
#                                   Photoelectric Coh+inc      Total
#   keV        e atom^-1    e atom^-1   cm^2 g^-1   cm^2 g^-1   cm^2 g^-1   cm^2 g^-1  nm
5.233200E-03  9.08733E-01  0.0000E+00  0.0000E+00  2.3914E-07  2.3914E-07  0.000E+00  2.369E+02
5.313300E-03  8.59283E-01  0.0000E+00  0.0000E+00  2.5404E-07  2.5404E-07  0.000E+00  2.333E+02
5.334660E-03  8.03599E-01  0.0000E+00  0.0000E+00  2.5813E-07  2.5813E-07  0.000E+00  2.324E+02
5.366700E-03  8.56971E-01  1.0769E-01  1.2165E+05  2.6435E-07  1.2165E+05  0.000E+00  2.310E+02
.
.
.
3.788588E+02  3.00000E+00  3.9121E-08  6.2602E-07  8.4389E-02  8.4390E-02  6.123E-07  3.273E-03
4.050001E+02  3.00000E+00  3.3438E-08  5.0054E-07  8.2127E-02  8.2128E-02  4.895E-07  3.061E-03
4.329451E+02  3.00000E+00  2.8581E-08  4.0022E-07  7.9892E-02  7.9892E-02  3.913E-07  2.864E-03
```

Please make sure you don't forget to remove the html-tags in the bottom of the file as well. In the future we will set up a more streamlined way of doing this.

4.7. The `Chopper_simple` McXtrace Component

Ideal chopper

Identification

- **Author:**
- **Origin:** Risoe
- **Date:** August 2011

Description

Ideal model of a chopper situated at Z=0. If a photon arrives at the chopper plane at a time of t = [t0 +-n*T: t0 +-n*T +tau], where T is the period of the chopper, t0 the initial delay and tau the opening time of the chopper, it is left untouched - otherwise it is ABSORBed. If on a continous source the isfirst parameter may be used. In this case the photon time is _defined_ by the chopper. In other words no photons are absorbed, the photon time is merely sampled within the chopper window. t_rise is the rise-time of

the chopper opening giving a trapezoidal shape. Limitations: this component does not take chopper geometry into account. If isfirst only samples in the first chopper opening window.

Example: Chopper_simple(

```
t0 = -0.5/M_C, T = 20e-6, tau = 100e-12, xwidth = 1e-4,
```

yheight = 1e-4, isfirst = 1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
t0	s	Initial delay of the opening time	0
T	s	Period of the chopper	1
tau	s	Opening time of the chopper	0.1
xwidth	m	Height of the chopper opening	0.1
yheight	m	Width of the chopeper opening	0.1
isfirst	0/1	Is the chopper the first chopper on a continous source.	0
t_rise	s	Rise time of the chopper pulse.	0
tjit	1	Timing jitter in terms of the opening time tau. For each ray the opening window will be shifted by a random amount within t=[-tjit*.5*tau,tjit*.5*tau]	0

Links

- Component source code found in file `Chopper_simple.comp`.

An ideal chopper

`Chopper_simple` is an idelized version of a chopper with a rectangular chopper opening which may open instantly and has no side-scattering etc.

It models the chopper with a blocking infinitely thin aperture which becomes transparent in the time interval $t \in [t0, t0 + \tau]$. This is an idelized version of a chopper where the chopper opening is rectangular which may open instantly (if $t_rise = 0$, the default). For nonzero rise time the aperture simply becomes gradually less opaque for $t \in [t0 - t_rise, t0]$.

For correct normalization of intesity a chopper period, T, must also be set.

isfirst is useful when using **Chopper_simple** with continous sources who inherently have no time-dependence. Thus the emission time of the photon ray is arbitrary, and

the chopper defines the temporal signature of the beam, i.e. it simply sets the time-parameter of the photon ray randomly in the opening window of the chopper. Naturally this should only be used for the *first* chopper element in a simulation.

5. Refractive optical components: lenses

An X-ray refractive lens, often referred to as a Compund Refractive Lens (CRL), is a fairly new type of device, which has gained popularity in the last few years. An early study showing the feasiblity of such devices may be found in [SKS96]. As the refractive index of X-rays is $n \approx 1$ a number of lenses, stacked together is usually necessary to bring the focal length to practical values. McXtrace includes a few lens components, which all have slightly different charactertics.

5.1. The `Lens_simple` McXtrace Component

Simple refractive x-ray lens

Identification

- **Author:** Erik Knudsen
- **Origin:** Risoe
- **Date:** June 16, 2009

Description

Models a stack of N refractive lenses, with a radius of curvature, r, at the apex. The model is a thin-lens approximation where photons are refracted in a the YZ plane at Z=0 (focusing in y only). Absorption is generally disregarded may be handled through the use of the optional transmission parameter T, where 0<=T<=1. Thus, the lens has the focal length of f=R/(2*N*&delta) where the x-ray refractive index is written: n = 1 - &delta + i &beta. Photons outside the aperture are absorbed.

Example: Lens_simple(xwidth=1e-5, yheight=1e-5, material_datafile="Be.txt",N=100,r=0.3e-3)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
xwidth	m	Width of lens aperture.	0
yheight	m	Height of lens aperture.	0

Name	Unit	Description	Default
radius	m	Radius of lens aperture (if non-zero, overrides xwidth & yheight).	0
T	0-1	Transmission efficiency of the lens.	1
r	m	The radius of curvature of the lens. Optional parameters:	3e-4
N	1	The number of successive lenses in the stack.	1
verbose	0/1	Extra information for debugging.	0
f	m	Focal length - overrides the material_datafile - and diregards chromatic aberration.	0
material_datafile		File where the material parameters for the lens may be found. Format is similar to what may be found off the NIST website.	"Be.txt"

Links

- Component source code found in file `Lens_simple.comp`.
- material datafile obtained from http://physics.nist.gov/cgi-bin/ffast/ffast.pl

Thin lens approximation

This models a thin-lens approximation of a stack of parabolic refractive x-ray lenses

5.2. The `Lens_parab` McXtrace Component

X-ray compound refractive lens (CRL) with a profile of the parabola

Identification

- **Author:** Jana Baltser and Erik Knudsen
- **Origin:** NBI
- **Date:** August 2010, modified July 2011

Description

A simple X-ray compound refractive lens (CRL) with a profile of the parabola in rotation simulates the photons' movement on passing through it. The CRL focuses in 2D Photons that miss the circular aperture of any surface (or undergo total external reflection) are absorbed.

Example: Lens_parab(material_datafile = "Be.txt", r=200e-6, r_ap=0.5e-3, d=50e-6, N=16)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
material_datafile	str	Datafile containing f1 constants	"Be.txt"
r	m	Radius of curvature (circular approximation at the tip of the profile).	0.5e-3
r_ap	m	Radius of circular aperture, which also defines the depth of the lens profile.	1.4e-3
d	m	Distance between two surfaces of the lens along the propagation axis.	.1e-3
N	1	Number of single lenses in a stack.	1

Links

- Component source code found in file `Lens_parab.comp`.

Thick parabolic CRL

Component model of a stack of compund refractive lenses. Each lens in the stack is modelled by two $2D$ parabolic surfaces, and rays are traced through all the complete stack taking the displacement of the surfaces into account. This is naturally less efficient than a thin lens approximation.

The logic behind the model is the following: a photon interacts with the surface of the lens at a certain angle Θ, which alters in accordance with Snell's law upon photon's entering the lens's material. The combination of the refractive process inside material (that is characterized by a material datafile) and the interaction with a geometrical surface results in a photon's new trajectory, i.e. in focusing.

The functionality of `Lens_parab_Cyl`, `Lens_parab_rough`, and `Lens_parab_Cyl_rough` will be merged into this component.

5.3. The `Lens_parab_Cyl` McXtrace Component

X-ray compound refractive lens (CRL) with a parabolic cylinder shape

Identification

- **Author:** Jana Baltser and Erik Knudsen
- **Origin:** NBI

- **Date:** April 2011

Description

An X-ray compound refractive lens (CRL) with a parabolic cylinder profile focusing in 1D, i.e. onto a line

The lens is invariant along the x-axis and has a parabolic profile along the y-axis defined as z = y^2/(2r). Thus, i.e. it focuses onto a line along the x-axis. N>1 means that a stack of lenses is to be simulated. Each lens consists of a pair of two opposing parabolic surfaces with a distance d between them. The reference point of the component is at the bottom of the first parabolic surface. Photons that miss the aperture of any surface (or undergo total external reflection) are absorbed.

Example: Lens_parab_Cyl(r=.5e-3,yheight=1.3e-3,xwidth=1.3e-3,d=.1e-3,N=21, material_datafile="Be.txt")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
material_datafile	str	Datafile containing f1 constants	"Be.txt"
r	m	Radius of curvature (circular approximation at the tip of the profile).	.5e-3
yheight	m	The CRL's aperture along Y.	1.2e-3
xwidth	m	The width of the CRL in the invariant direction x.	1.2e-3
d	m	Distance between two surfaces of the lens along the propagation axis.	.1e-3
N	1	Number of single lenses in a stack.	1

Links

- Component source code found in file `Lens_parab_Cyl.comp`.

Thick 1D-parabolic CRL

This component is based on the same logical approach as the **Lens_parab** with one significant difference - geometrical surface. Parabolic cylinder (parabolic curvature along the vertical axes only, invariant along the horizontal) and thus ofocuses the beam in only in the vertical direction.

This component and its functionality is scheduled to be merged into **Lens_parab**

5.4. The `Lens_elliptical` McXtrace Component

X-ray compound refractive lens (CRL) with an elliptic profile

Identification

- **Author:** Jana Baltser and Erik Knudsen
- **Origin:** NBI
- **Date:** August 2010

Description

A simple X-ray compound refractive lens (CRL) with an elliptic profile simulates the photons' movement on passing through it. Each surface is an elliptic paraboloid x^2/r1^2 + y^2/r2^2 = (z-z0)/w limited to the aperture x^2/r1^2 + y^2/r2^2 <= 1. Attenuation coefficient mu and f1 are taken from the material datafile (e.g. Be.txt), in the same format as used by Lens_parab and Lens_simple. Photons that miss the aperture of any surface (or undergo total external reflection) are absorbed.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
material_datafile	Be.txt	File where the material parameters for the filter may be found. Format is similar to what may be found off the NIST website.	"Be.txt"
r1	m	Semi-axis of the elliptic aperture along the X axis	0.42e-3
r2	m	Semi-axis of the elliptic aperture along the Y axis	0.8e-3
w	m	Depth of one parabolic surface along the propagation axis (at the aperture edge)	0.46e-3
d	m	Distance between two surfaces of the lens along the propagation axis	0.2e-4
Transmission	1	Currently not used by the component	1
N	1	Number of single lenses in a stack	1

Links

- Component source code found in file `Lens_elliptical.comp`.
- material datafile obtained from http://physics.nist.gov/cgi-bin/ffast/ffast.pl

Refractive lens

doc. pend.

6. Reflective optical components: mirrors

This section describes advanced reflective X-ray optics components such as mirrors.

6.1. The `Mirror_curved` McXtrace Component

Rectangular curved mirror (paraboloid/quasi-spherical/cylindrical) or flat

Identification

- **Author:** Emmanuel Farhi, adapted from Mirror by Erik B Knudsen
- **Origin:** SOLEIL
- **Date:** November 2026

Description

Like the Mirror component, the mirror is by default tangent to the local XZ plane (mirror surface at y=0, nominal beam along +z), and placed in the local YZ plane instead by setting 'yheight' or 'width'. The surface has two principal curvature radii: Rx: radius bending the surface across the mirror width, i.e. in the local XZ plane for an XZ mirror, or in the YZ plane for a YZ mirror (perpendicular, sagittal radius). Rz: radius bending the surface along the mirror length (in the local YZ plane), i.e. focusing along the beam (meridional, tangential radius). A radius equal to 0 means the surface is NOT curved in that direction.

```
- Rx  =   Rz      -> quasi-spherical mirror (paraxial toroid, large Rx=Rz).
- Rx !=   Rz      -> toroidal/astigmatic mirror (paraxial sag intersection).
```

- Rx !=0 Rz =0 -> cylindrical - Rx =0 Rz !=0 -> bent mirror (along beam, e.g. for KB)

```
- Rx  =   Rz  =0  -> perfectly flat mirror, equivalent to Mirror.
```

The radii usually follow f_t=Rz*sin(theta)/2 and f_s=Rx/sin(theta)/2 where theta is the tilt angle, and f_t and f_s are tangential and sagittal focus distances.

Sign convention: R > 0 means the surface sags towards the normal side (towards +y for an XZ mirror): the centre of curvature lies upstream and a beam along +z is focused. R > 0 converges (concave).

Surface: N = (T^2/(2 Rx) + z^2/(2 Rz)), where N is the surface normal (y for an XZ mirror, x for a YZ mirror) and T the transverse coordinate (x for an XZ mirror, y for a YZ mirror). The mirror is truncated to the aperture —T— <= (xwidth—yheight)/2 and —z— <= zdepth/2. With one non-zero radius the ray/cylinder intersection is exact (the surface stays flat along the other direction); with two non-zero radii the sag equation (paraxial toroid) is used, exact in the limit of aperture << R. Use Rx=Rz for quasi-spherical optics. The mirror is located at its center. Default geometry is a mirror in the XZ plane, as the Mirror component. When using the 'width' or 'yheight' parameter, the mirror flips in the YZ plane. To model a toroid, better use the Mirror_toroid component.

Reflectivity is handled exactly as in Mirror: either constant (R0) or from a datafile/coating recipe ('coating').

Example: Mirror_curved(xwidth=1e-3, zdepth=1e-2, Rx=1.0, Rz=0, R0=1, coating="B4C.dat") Example: Mirror_curved(xwidth=2e-3, zdepth=5e-2, Rx=0.5, Rz=0.7) Example: Mirror_curved(yheight=1e-3, zdepth=1e-2, Rx=0.3) (YZ-plane mirror)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
zdepth	m	Mirror aperture length along local z (beam direction)	0.1
xwidth	m	Mirror aperture width along local x (XZ mirror, default)	0.01
yheight	m	Setting yheight > 0 overrides xwidth and places the mirror in the YZ plane (aperture along local y), as for Mirror.	0
Rx	m	Sagittal curvature radius (across the mirror width). 0 = flat.	0
Rz	m	Tangential curvature radius (along the beam). 0 = flat.	0
coating	str	Filename containing reflectivities (or coating).	""
R0	0-1	Constant reflectivity	1
radius	m	Radius of tangential curvature along Z = Rz (compatibility).	0
length	m	Length of the unbent mirror along Z = zdepth (compatibility).	0
width	m	Width of the mirror along Y = yheight (compatibility).	0

Links

- Component source code found in file `Mirror_curved.comp`.

Cylindrically curved mirror

Models a cylindrical mirror, positioned in the XZ-plane curving towards positive X. The input parameter *radius* defines the radius of curvature and the mirror size is given by *length* and *width* where length and width is along Z and Y respectively. *coating* and *R0* are mutually exclusive. If *R0* is nonzero, it is taken as the reflectivity value, irrespective of wavelength, whereas coating nominates a file from which to read values for f_1 and f_2. See [Nis] for definitions. For elements $Z \in [1, 92]$ datafiles are distributed with the McXtrace system that may be used as: `coating="Rh.txt"`. or `coating="Al.txt"`.

This component is scheduled to be merged with `Mirror_parabolic` and `Mirror_elliptic`

6.2. The `Mirror_parabolic` McXtrace Component

Idealized parabolic mirror (in XZ)

Identification

- **Author:** Erik Knudsen
- **Origin:** Risoe
- **Date:** Feb 11, 2010

Description

Takes a reflectivity as input and reflects rays in a ideal geometry parabolic mirror. The mirror is positioned in the zx-plane curving towards positive y. The axis of the paraboloid is y, so the focal point lies at (0, f_x, 0) with f_x = a^2/4 for the x-curve, and f_z = b^2/4 for the z-curve. For a != b the mirror is astigmatic: there are two different focal lengths, not a single focus.

Example: Mirror_parabolic(R0=1, a=1, b=0, xwidth=0.02, yheight=0, zdepth=0.05)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
R0	1	Reflectivity of mirror.	1
a	sqrt(m)	Transverse curvature scale, if zero - the mirror is flat along x.	1
b	sqrt(m)	Longitudinal curvature scale, if zero, flat along z.	1

Name	Unit	Description	Default
xwidth	m	Width of the mirror, measured as arc length s along the parabola from the vertex (not the projected extent).	0.1
zdepth	m	Length of the mirror, measured as arc length s along the parabola from the vertex (not the projected extent).	0.1
yheight	m	Thickness of mirror, one-sided: tolerance on the substrate side (below the surface). If 0 (the default) the mirror is mathematically thin. Only has an effect for rays grazing the mirror edge-on.	0
focusx	m	Transverse focal length along X. Sets a.	0
focusz	m	Longitudinal focal length along Z. Sets b.	0
radius	m	Focal length (historical name); sets focusx and focusz together.	0

Links

- Component source code found in file `Mirror_parabolic.comp`.

Mirror with a parabolic curvature profile.

This component is scheduled to be merged with `Mirror_elliptic`

6.3. The `Mirror_elliptic` McXtrace Component

Idealized elliptic mirror.

Identification

- **Author:** Erik Knudsen
- **Origin:** Risoe
- **Date:** Feb 11, 2010

Description

Takes a reflectivity as input and reflects rays in a ideal geometry elliptic mirror. The mirror is positioned such that the a-axis of the mirror ellipsoid is on the x-axis, the b-axis is along the y-axis and the c is along the z-axis. The reference point of the mirror is the ellipsoid centre, offset by one half-axis along the y-axis (See the component manual

for a drawing). This means that to position the mirror correctly, the user positions the ellipsoid governing the mirror shape, not the mirror itself.

Only the lower half of the ellipsoid (y < y_b) is the mirror surface. Rays are reflected from both sides of the surface: from the concave (inner) side, and from the convex (back) side, which de-focuses the beam. The reflectivity only depends on the grazing angle, and not on the side. The complex reflectivity is applied: its modulus scales the ray weight and its argument is added to the ray phase.

The foci of the ellipsoid are situated at y = y_b and $z = \pm\sqrt{z_c^2 - y_b^2}$. For a grazing incidence (z_c > y_b) angle theta and collimated incoming beam, the focal lengths are situated downstream at (from ellipse centre): - f_t = z_cˆ2 sin(theta) / (2 y_b) - f_s = x_aˆ2 / (2 y_b sin(theta)) Typical incidence angle is sin(theta) = y_b/z_c

Limitation: the polarisation (Ex, Ey, Ez) is not transformed by the reflection.

Example: Mirror_elliptic(zdepth=150e-3, xwidth=150e-3, x_a=1.025, y_b=1.025, z_c=1.025)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
x_a	m	1st short half axis (along x). Commonly set to zero, which really implies infinite value, so the mirror is an elliptic cylinder.	0
y_b	m	2nd short half axis (along y), which is also the presumed near-normal direction, reflection near the y-z plane. Must not be 0.	1.0
z_c	m	long half axis (along z). Commonly a=0. b=c, which creates a circular cylindrical surface.	1.0
zdepth	m	Depth (length) of the mirror along Z.	0.2
xwidth	m	Width of the mirror along X.	0.2
R0	1	Constant reflectivity of mirror, used when coating is "NULL" or "".	1
coating	str	Datafile containing either mirror material constants or reflectivity numbers. Use "NULL" (or "") for a constant reflectivity R0.	"NULL"
length	m	alternate name for zdepth (obsolete). When > 0, it overrides zdepth.	0
width	m	alternate name for xwidth (obsolete). When > 0, it overrides xwidth.	0

Name	Unit	Description	Default
radius	m	Spherical radius, Sets x_a=y_b=z_c=radius	0

Links

- Component source code found in file `Mirror_elliptic.comp`.

Mirror with a elliptic curvature profile.

This component is scheduled to be merged with `Mirror_parabolic`

6.4. The `Multilayer_elliptic` McXtrace Component

Elliptic multilayer mirror (in XZ)

Identification

- **Author:** Jana Baltser, Peter Willendrup, Anette Vickery, Andrea Prodi, Erik Knudsen
- **Origin:** NBI
- **Date:** February 2011

Description

Reads reflectivity values from a data input file (Ref.dat) for a Si/W multilayer. The multilayer code reflects ray in an ideal geometry, does not include surface imperfections

The mirror is positioned such that the long axis of the mirror elliptical surface coincides with z-axis

The algorithm: Incoming photon's coordinates and direction (k-vector) are transformed into an elliptical reference frame (elliptical parameters are calculated according to the mirror's position and its focusing distances and the incident angle), the intersection point is then defined. A new, reflected photon is then starting at the point of intersection.

Example: Multilayer_elliptic(coating = "Ref_W_B4C.txt", theta = 1.2,

```
s1 = 1, s2 = 2, length = 0.1, width = 0.1, R0 = 1,
```

Emin=7, Emax=10, Estep=0.05)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
coating	str	Datafile containing reflectivity values as a function of q and E.	"Ref_W_B4C.txt"
theta	deg	Design angle of incidence.	1.2
s1	m	Design distance from the source to the multilayer.	0
s2	m	Design focusing distance of the multilayer.	0
length	m	alternate name for zdepth (obsolete)	0.5
width	m	alternate name for xwidth (obsolete)	0.2
R0	1	Maximal reflectivity	1
Emin	keV	Lower limit of energy interval in datafile. Overrides what's written in the datafile header.	-1
Emax	keV	Upper limit of energy interval in datafile. Overrides what's written in the datafile header.	-1
Estep	keV	Step between energy sample points in datafile. Overrides what's written in the datafile header.	-1
Gamma		High electron density fraction of bilayer (in kinematical appr.).	0
Lambda	Å	Thickness of bilayer (in kinematical appr.).	0
rho_AB	e/Å^3	Number electron density constrast in bilayer (in kinematical appr.).	0
N	1	Number of bilayers (in kinematical appr.).	0
xwidth	m	Width of the mirror along X-axis.	0
zdepth	m	Length of the mirror along Z.	0

Links

- Component source code found in file `Multilayer_elliptic.comp`.

Elliptically curved mirror coated with a multilayer

The component **Multilayer_elliptic** models a single rectangular reflecting multilayer mirror plate with elliptical curvature. It can be used as a sample component, to *e.g.* as-

semble a Kirkpatrick-Baez focusing system or in combination with a double-crystal monochromator.

Figure 6.1*Left* shows a side view of a mirror (the blue section of the ellipse) in the McXtrace coordinate system. At the mirror center, the mirror tangent is parallel to the z axis and the mirror normal is parallel to the y axis. The width of the mirror is w and in $y-z$ plane the mirror has the curvature of an ellipse with major axis a and minor axis b,

$$\frac{z^2}{a^2}+\frac{y^2}{b^2}=1\,,\ |x|<\frac{w}{2}\,. \tag{6.1}$$

The length of the mirror is L. The coordinates of the mirror center $(0, Y_0, Z_0)$ and the ellipse parameters a, b are determined uniquely by the central glancing angle, the source-mirror distance and the mirror-image distance. The position of the mirror is chosen to be at the positive side of the y axis.

The input parameters of this component are: *theta* [°], the incident angle; *s1* [m], the distance from the source to the multilayer; *s2* [m], the focusing distance of the multilayer; *length* [m], the length of the mirrors; *width* [m], the width of the mirror along the x-axis; R, the reflectivity.

6.4.1. Definition of the reference frames

The direction and position of the incoming photon is defined relative to the coordinate system illustrated in Fig. 6.1*Left* (in the code referred to as *McXtrace coordinate system*):

- the y-axis is parallel to the central mirror normal
- the z-axis is parallel to the central mirror tangent
- the origin is at the mirror center

However, all the calculations are conducted in another reference frame which is illustrated in Fig. 6.1 *Right*(in the following referred to as the *Ellipse coordinate system*):

- the z-axis is parallel to major axis of ellipse
- the y-axis is parallel to minor axis of ellipse
- the origin is at the center of the ellipse
- the mirror center at $(0, Y_0, Z_0)$, uniquely determined by the glancing angle at the mirror center, the source-mirror distance and mirror-image distance.

6.4.2. Algorithm

1. The photon is generated with a starting point $\mathbf{S}$ and a direction $\mathbf{V}_{\text{in}}$ defined in the *McXtrace* coordinate system.

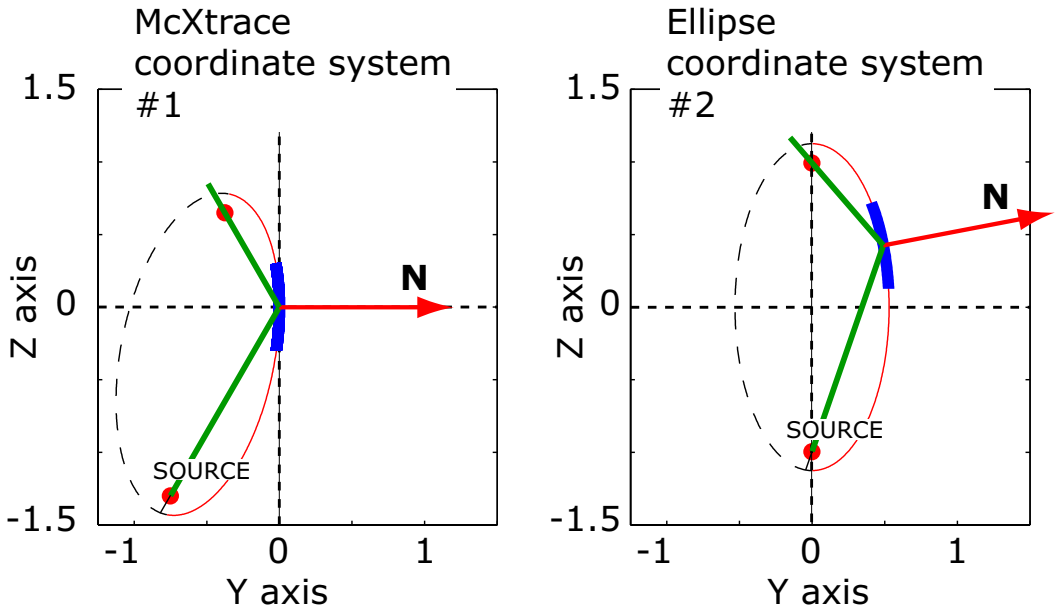


Figure 6.1.: The same image in different coordinate systems.
Left: *McXtrace System* with the y-axis is parallel to the central mirror normal, the z-axis is parallel to the central mirror tangent and the origin is at the mirror center.
Right: *Ellipse System* with the z-axis parallel to major axis of ellipse, the y-axis is parallel to minor axis of ellipse and the origin is at the center of the ellipse.

2. All calculations are performed in the *Ellipse* coordinate system, so to proceed the basis is changed to that reference frame.

3. The 2 intersections of the ray with the ellipse are determined.

4. It is checked if any of the intersections are within the area defined by the mirror.

5. If one of the solutions is valid, the reflection of that ray is determined.

6. The coordinates of the starting point and direction of the reflected ray are calculated using the basis of the *McXtrace* coordinate system.

6.4.3. Reflection of the ray in the mirror

The tangent and normal to the ellipse $z^2/a^2 + y^2/b^2 = 1$ at the point (Y, Z) are found by implicit differentiation:

$$\frac{2z}{a^2} + \frac{2y}{b^2}\frac{dy}{dz} = 0\,, \tag{6.2}$$

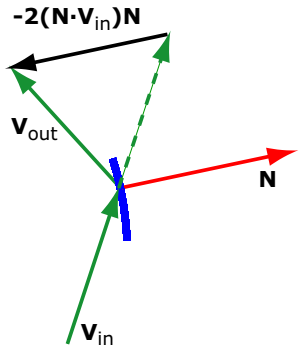


Figure 6.2.: The reflection of the unit vector $\mathbf{V}_{\text{in}}$ in the mirror with the normal unit vector $\boldsymbol{N}$ is $\boldsymbol{V}_{\text{out}} = \boldsymbol{V}_{\text{in}} - 2(\boldsymbol{N} \cdot \boldsymbol{V}_{\text{in}})\boldsymbol{N}$

so at the point (Y, Z) the slope of the tangent is $\frac{dy}{dz} = -\frac{Z\,b^2}{Y\,a^2}$. The slope of the normal is minus the inverse of the tangent slope, so the coordinates of the mirror normal are

$$N_x = 0 \quad N_y = \frac{a^2\,Y}{b^2\,Z} \quad N_z = 1\,. \tag{6.3}$$

With $\mathbf{V}_{\text{in}}$ and $\mathbf{N}$ denoting unit vectors (direction and normal respectively), the direction of the reflected ray is calculated as

$$\boldsymbol{V}_{\text{out}} = \boldsymbol{V}_{\text{in}} - 2(\boldsymbol{N} \cdot \boldsymbol{V}_{\text{in}})\boldsymbol{N} = \begin{pmatrix} V_{\text{in}x} - 2(\boldsymbol{N} \cdot \boldsymbol{V}_{\text{in}})N_x \\ V_{\text{in}y} - 2(\boldsymbol{N} \cdot \boldsymbol{V}_{\text{in}})N_y \\ V_{\text{in}z} - 2(\boldsymbol{N} \cdot \boldsymbol{V}_{\text{in}})N_z \end{pmatrix} \tag{6.4}$$

6.4.4. Mirror reflectivity

At present, the Multilayer_elliptic Mirror component uses a reflectivity table *reflect*, which 1st column is q [$\mathring{A}^{-1}$] and from the 2nd column on as the reflectivity R in [0-1] as function of tabulated energy [keV]. An example file, calculated for a particular Si/W multilayer, is provided (`reflectivity.txt`). User provided reflectivity data files can be parsed by the component.

6.5. The `TwinKB_ML` McXtrace Component

Montel optic model (aka side-by-side Kirkpatrick Baez)

Identification

- **Author:** Jana Baltser, Peter Willendrup, Anette Vickery, Andrea Prodi, Erik Knudsen, Jesper Buch Jensen
- **Origin:** NBI
- **Date:** May 2012

Description

Models a Montel optic, or Twin Kirkpatrick Baez mirror optic (hence the component name). The mirror are fully abutting, i.e. there's is no gap between them, and perfectly elliptic.

Reads reflectivity values from a data input file for a W/B4C multilayer. The multilayer code reflects ray in an ideal geometry, the reflectivity datafile accounts for surface roughness, sigma.

The mirror is positioned such that the long axis of the mirror elliptical surface coincides with the z-axis.

The algorithm: Incoming photon's coordinates and direction (k-vector) are transformed into an elliptical reference frame (elliptical parameters are calculated according to the mirror's position and its focusing distances and the * incident angle), the intersection point is then defined. A new, reflected photon is then starting at the point of intersection.

Example: TwinKB_ML(theta=1.2, s1=.045 , s2=.9 , length=0.06 , width=0.2 , R0=0 , reflectivity_datafile="Ref_W_B4C.txt")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
reflectivity_datafile	str	File which contains reflectivities as a function of q.	"Ref.txt"
theta	deg	Incident angle	1.2
s1	m	Distance from the source to the multilayer	
s2	m	Focusing distance of the multilayer	
length	m	Length of the mirrors	0.6
width	m	Width of the mirror along x-axis	0.2
R0	0-1	Constant reflectivity, R0=1 for an ideal situation. If R0=0, the code reads the reflectivity from the datafile	0

Links

- Component source code found in file `TwinKB_ML.comp`.

Side-by-side Kirkpatrick-Baez mirror pair

Models a pair of perpendicular, elliptically curved mirrors, known as a Montel-mirror or Side-by-side Kirkpatrick-Baez mirror.

7. Samples

This class of components models the sample of the experiment. This is by far the most challenging part of an xray scattering instrument to model. However, for purpose of simulating instrument performance, details of the samples are rather unimportant, allowing for simple approximations. On the contrary, for full virtual experiments it is of importance to have realistic and detailed sample descriptions. McXtrace contains both simple and detailed samples.

An important component class is elastic Bragg scattering from an ideal powder. The component **PowderN** models a powder scatterer with reflections given in an input file. The component includes absorption, incoherent scattering, direct beam transmission and can assume *concentric* shape, i.e. can be used for modelling sample enviroments.

Next type is Bragg scattering from single crystals. Two types of single crystal components exist in McXtrace at present: **Bragg_crystal** (in the optics category) and **Single_crystal**. **Bragg_crystal** is in fact most often used as a monchromator crystal. It models a crystal where peak broadening is dominated by the Darwin width. Currently it only handles a single defined reflection. If more than one is wanted this could be accomplished by using two instances in a `GROUP` and dynamically choose between them with a `WHEN`-statement. For details on the `GROUP` and `WHEN` constructs see the main McXtrace user manual [Knu+14]

Much more general, the component **Single_crystal** is a single crystal sample (with multiple scattering) that allows the input of an arbitrary unit cell and a list of structure factors, read from a LAZY / Crystallographica file. This component also allows anisotropic mosaicity and $\Delta d/d$ lattice space variation.

Isotropic small-angle scattering is simulated in **Saxs_Spheres**, which models scattering from a collection of hard spheres (dilute colloids). Furthermore, a whole series of sample components modelling various SAXS standard sample types are available in the `contrib` component library section.

7.0.1. Scattering notation

In sample components, we use a notation common for scattering experiments, where the wave vector transfer is denoted the *scattering vector*

$$\boldsymbol{q} \equiv \boldsymbol{k}_\mathrm{i} - \boldsymbol{k}_\mathrm{f}. \tag{7.1}$$

In analygo, the *energy transfer* is given by

$$\hbar\omega \equiv E_\mathrm{i} - E_\mathrm{f} = \frac{\hbar^2}{2m_\mathrm{n}}\left(k_\mathrm{i}^2 - k_\mathrm{f}^2\right). \tag{7.2}$$

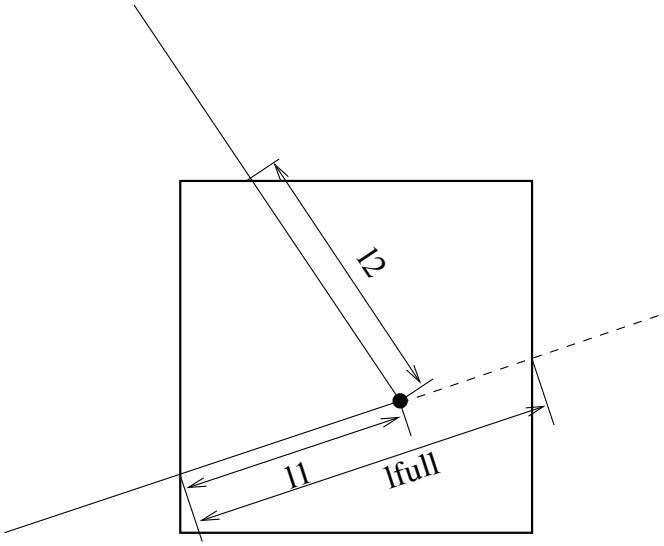


Figure 7.1.: The geometry of a scattering event within a powder sample.

7.0.2. Weight transformation in samples; focusing

Within many samples, the incident beam is attenuated by scattering and absorption, so that the illumination varies considerably throughout the sample. For single crystals, this phenomenon is known as *secondary extinction* [Bac75], but the effect is important for all samples. In analytical treatments, attenuation is difficult to deal with, and is thus often ignored, making a *thin sample approximation*. In Monte Carlo simulations, the beam attenuation is easily taken care of, as will be shown below. In the description, we ignore multiple scattering, which is however implemented in some sample components.

The sample has an absorption cross section per unit cell of σ_c^a and a scattering cross section per unit cell of σ_c^s. The x-ray path length in the sample before the scattering event is denoted by l_1, and the path length within the sample after the scattering is denoted by l_2, see figure 7.1. We then define the inverse penetration lengths as $\mu^s = \sigma_c^s/V_c$ and $\mu^a = \sigma_c^a/V_c$, where V_c is the volume of a unit cell. Physically, the attenuation along this path follows

$$f_{\mathrm{att}}(l) = \exp(-l(\mu^s + \mu^a)), \tag{7.3}$$

where the normalization $f_{\mathrm{att}}(0) = 1$.

The probability for a given x-ray to be scattered from within the interval $[l_1; l_1 + dl]$ will be

$$P(l_1)dl = \mu^s f_{\mathrm{att}}(l_1)dl, \tag{7.4}$$

while the probability for a x-ray to be scattered from within this interval into the solid angle Ω *and* not being scattered further or absorbed on the way out of the sample is

$$P(l_1, \Omega)dld\Omega = \mu^s f_{\mathrm{att}}(l_1) f_{\mathrm{att}}(l_2)\gamma(\Omega)d\Omega dl, \tag{7.5}$$

where $\gamma(\Omega)$ is the directional distribution of the scattered x-rays, and l_2 is determined by Monte Carlo chocies of l_1, Ω, and from the sample geometry, see e.g. figure 7.1.

In our Monte-Carlo simulations, we may choose the scattering parameters by making a Monte-Carlo choice of l_1 and Ω from a distribution different from $P(l_1, \Omega)$. By doing this, we must adjust π_i according to the probability transformation rule (2.9). If we *e.g.* choose the scattering depth, l_1, from a flat distribution in $[0; l_{\text{full}}]$, and choose the directional dependence from $g(\Omega)$, we have a Monte Carlo probability

$$f(l_1, \Omega) = g(\Omega)/l_{\text{full}}, \tag{7.6}$$

l_{full} is here the path length through the sample as taken by a non-scattered x-ray (although we here assume that all simulated x-rays are being scattered). According to (2.9), the x-ray weight factor is now adjusted by the amount

$$\pi_i(l_1, \Omega) = \mu^s l_{\text{full}} \exp\left[-(l_1 + l_2)(\mu^a + \mu^s)\right] \frac{\gamma(\Omega)}{g(\Omega)}. \tag{7.7}$$

In analogy with the source components, it is possible to define "interesting" directions for the scattering. One will then try to focus the scattered x-rays, choosing a $g(\Omega)$, which peaks around these directions. To do this, one uses (7.7), where the fraction $\gamma(\Omega)/g(\Omega)$ corrects for the focusing. One must choose a proper distribution so that $g(\Omega) > 0$ in every interesting direction. If this is not the case, the Monte Carlo simulation gives incorrect results. All samples have been constructed with a focusing and a non-focusing option.

7.0.3. Future development of sample components

There is still room for much more development of functionality in McXtrace samples.

7.1. The `Absorption_sample` McXtrace Component

Sample component with absorbing materials.

Identification

- **Author:** Erik B Knudsen
- **Origin:** Risoe
- **Date:** March 2011

Description

A sample component consisting of a volume of one material and volume of another material inside. This is useful as a phantom for simulating tomography experiments. The inner material can be left unset (all 0), to only have one volume.

Sample shape may be a cylinder, a sphere, a box or any other shape

```
box/plate:       xwidth x yheight x zdepth
cylinder:        radius x yheight
sphere:          radius (yheight=0)
any shape:       geometry=OFF/PLY file
```

Example: Absorption_sample(material_datafile_o="Mn.txt", xwidth_o = 0.5, yheight_o = 0.5, zdepth_o = 0.0001, rho_o=7.15)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
material_datafile_i	str	Name of file containing material data for inclusion.	""
material_datafile_o	str	Name of file containing material data for outer volume.	""
radius_o	m	Radius of "outer" enclosing material cylinder (0)	0
xwidth_o	m	Width of "outer" enclosing material box (yheight)	0
yheight_o	m	Height of "outer enclosing material box (1)	1
zdepth_o	m	Thickness of outer enclosing material box (yheight)	0
radius_i	m	Radius of "inner" enclosed material cylinder (0)	0
xwidth_i	m	Width of "inner" enclosed material box (yheight)	0
yheight_i	m	Height of "inner enclosed material (1)	0.0
zdepth_i	m	Thickness of inner enclosed material box (yheight)	0
x_i	m	Center x-coordinate of "inner" object	0
y_i	m	Center y-coordinate of "inner" object	0
z_i	m	Center z-coordinate of "inner" object	0
rho_i	g/cm^3	density of the enclosed material	0
rho_o	g/cm^3	density of the enclosing material	0
geometry_i	str	Name of an inner Object File Format (OFF) or PLY file for complex geometry. The OFF/PLY file may be generated from XYZ coordinates using qhull/powercrust [str]	""

Name	Unit	Description	Default
geometry_o	str	Name of the outer Object File Format (OFF) or PLY file for complex geometry. The OFF/PLY file may be generated from XYZ coordinates using qhull/powercrust	""

Links

- Component source code found in file `Absorption_sample.comp`.
- Meshlab https://www.meshlab.net/
- Geomview and Object File Format (OFF) <http://www.geomview.org>
- Java version of Geomview (display only) jroff.jar <http://www.holmes3d.net/graphics/roffview/>
- qhull <http://qhull.org>
- Powercrust https://www.cs.ucdavis.edu/~amenta/powercrust.html

An absorption phantom

This component models an absorption phantom with a single inclusion in surrounding material. It is intended use is for tomographic imaging simulations.

7.2. The `Saxs_spheres` McXtrace Component

Release: McXtrace 1.1

Sample for Small Angle X-ray Scattering - hard spheres in thin solution, mono disperse.

Identification

- **Author:** E. B. Knudsen, P. Willendrup, K. Lefmann, L. Arleth
- **Origin:** DTU Fysik
- **Date:** 28.10.2010

Description

Sample for use in a SAXS instrument, models hard, monodisperse spheres in thin solution. The shape of the sample may be a filled box with dimensions xwidth, yheight, zdepth, a cylinder with dimensions radius and yheight, a filled sphere with radius R.

Example: Saxs_spheres(R = 20, Phi = 1e-3, Delta_rho = 0.6, sigma_abs = 50, xwidth=0.01, yheight=0.01, zdepth=0.005)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sphere_mtrl	str	Material datafile from which to find absorption. If none is given absorption is neglected.	""
R	Å	Radius of scattering hard spheres	100
Phi	1	Particle volume fraction	1e-3
Delta_rho	fm/Å^3	Excess scattering length density	0.6
xwidth	m	Horiz. dimension of sample, as a width	0
yheight	m	Vert . dimension of sample, as a height for cylinder/box	0
zdepth	m	Depth of sample	0
radius	m	Outer radius of sample in (x,z) plane for cylinder/sphere	0
target_x	m	Position of target to focus at, along X	0
target_y	m	Position of target to focus at, along Y	0
target_z	m	Position of target to focus at, along Z	6
target_index	1	Relative index of component to focus at, e.g. next is +1	0
focus_xw	m	Horiz. dimension of a rectangular area	0
focus_yh	m	Vert. dimension of a rectangular area	0

Name	Unit	Description	Default
focus_aw	deg	Horiz. angular dimension of a rectangular area	0
focus_ah	deg	Vert. angular dimension of a rectangular area	0
focus_r	m	Detector (disk-shaped) radius	0
mu_c	5	Column of the datafile which contains absorption coefficients.	0

Links

- Component source code found in file `Saxs_spheres.comp`.

A model of dilute hard spheres in solution for SAXS-use

7.3. The `PowderN` McXtrace Component

General powder sample (N lines, single scattering, incoherent scattering)

Identification

- **Author:** P. Willendrup, L. Chapon, K. Lefmann, A.B.Abrahamsen, N.B.Christensen, E.M.Lauridsen.
- **Origin:** McXtrace release 1.2
- **Date:** 4.2.98

Description

General powder sample with many scattering vectors possibility for intrinsic line broadening incoherent background ratio is computed from material datafile. No multiple scattering. No secondary extinction.

Based on Powder1/Powder2/Single_crystal. Geometry is a powder filled cylinder, sphere, box or any shape from an OFF file. Incoherent scattering is only provided here to account for a background. The efficient is highly improved when restricting the vertical scattering range on the Debye-Scherrer cone (with 'd_phi' and 'focus_flip'). The unit cell volume Vc may also be computed when giving the density, the atomic/molecular weight and the number of atoms per unit cell.

Sample shape: Sample shape may be a cylinder, a sphere, a box or any other shape.

```
box/plate:       xwidth x yheight x zdepth (thickness=0)
```

hollow box/plate:xwidth x yheight x zdepth and thickness>0

```
cylinder:        radius x yheight (thickness=0)
```

hollow cylinder: radius x yheight and thickness>0 (no lids: the cavity spans the full yheight, as a sample can without lids)

```
sphere:          radius (yheight=0 thickness=0)
hollow sphere:   radius and thickness>0 (yheight=0)
any shape:       geometry=OFF_file
```

The complex geometry option handles any closed non-convex polyhedra. It computes the intersection points of the xray with the object transparently, so that it can be used like a regular sample object. It supports the PLY, OFF and NOFF file format but not COFF (colored faces). Such files may be generated from XYZ data using: qhull < coordinates.xyz Qx Qv Tv o > geomview.off or powercrust coordinates.xyz and viewed with geomview or java -jar jroff.jar (see below). The default size of the object depends of the OFF file data, but its bounding box may be resized using xwidth,yheight and zdepth.

If you use this component and produce valuable scientific results, please cite authors with references bellow (in Links).

Example: PowderN(reflections = "c60.lau", d_phi = 15 , radius = 0.01,

```
yheight = 0.05, Vc = 1076.89, delta_d_d=0, DW=1)
```

Powder definition file format Powder structure is specified with an ascii data file 'reflections'. The powder data are free-text column based files. The reflection list should be ordered by decreasing d-spacing values.

```
... d ... F2
```

Lines begining by '#' are read as comments (ignored) but they may contain the following keywords (in the header):

```
#Vc           <value of unit cell volume Vc [Angs^3]>
```

#Debye_Waller <value of Debye-Waller factor DW>

```
#delta_d_d/d    <value of delta_d_d/d width for all lines>
```

These values are not read if entered as component parameters (Vc=...)

The signification of the columns in the numerical block may be set using the 'format' parameter, by defining signification of the columns as a vector of indexes in the order format={j,d,F2,DW,delta_d_d/d,1/2d,q,F} Signification of the symbols is given below. Indices start at 1. Indices with zero means that the column are not present, so that: Crystallographica={ 4,5,7,0,0,0,0,0 }

```
Fullprof         ={ 4,0,8,0,0,5,0,0 }
Lazy             ={17,6,0,0,0,0,0,13}
```

At last, the format may be overridden by direct definition of the column indexes in the file itself by using the following keywords in the header (e.g. '#column_j 4'):

```
#column_j     <index of the multiplicity 'j' column>
#column_d     <index of the d-spacing 'd' column [Angs]>
#column_F2    <index of the squared str. factor '|F|^2' column [b]>
#column_F     <index of the structure factor norm '|F|' column>
#column_DW    <index of the Debye-Waller factor 'DW' column>
#column_Dd    <index of the relative line width delta_d_d/d broadening 'Dd' column>
```

#column_inv2d <index of the 1/2d=sin(theta)/lambda 'inv2d' column>

```
#column_q     <index of the scattering wavevector 'q' column [Angs-1]>
```

Last, CIF, FullProf and ShelX files can be read, and converted to F2(hkl) lists if 'cif2hkl' is installed. The CIF2HKL env variable can be used to point to a proper executable, else the McCode, then the system installed versions are used.

Concentricity

PowderN assumes 'concentric' shape, i.e. can contain other components inside its optional inner hollow. Example, Sample in Al cryostat:

COMPONENT Cryo = PowderN(reflections="Al.laz", radius = 0.01, thickness = 0.001, concentric = 1, p_interact=0.1) AT (0,0,0) RELATIVE Somewhere

COMPONENT Sample = some_other_component(with geometry FULLY enclosed in the hollow) AT (0,0,0) RELATIVE Somewhere

COMPONENT Cryo2 = COPY(Cryo)(concentric = 0) AT (0,0,0) RELATIVE Somewhere

(The second instance of the cryostat component can also be written out completely using PowderN(...). In both cases, this second instance needs concentric = 0.) The concentric arrangment can not be used with OFF geometry specification.

This sample component can advantageously benefit from the SPLIT feature, e.g. SPLIT COMPONENT pow = PowderN(...)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
reflections	str	Input file for reflections (LAZ LAU CIF, FullProf, ShelX). Use only incoherent scattering if NULL or "".	"NULL"
material	Be.txt	File where the material parameters for the absorption may be found. Format is similar to what may be found off the NIST website.	"NULL"
geometry	str	Name of an Object File Format (OFF) or PLY file for complex geometry. The OFF/PLY file may be generated from XYZ coordinates using qhull/powercrust.	"NULL"
format	{}	Name of the format, or list of column indexes (see Description). N.b. no quotes!	{0,0,0,0,0,0,0,0}
mat_format	{}	Format of the asorption parameter file.	{0,0,0,0,0}
radius	m	Outer radius of sample in (x,z) plane.	0
yheight	m	Height of sample y direction.	0
xwidth	m	Horiz. dimension of sample, as a width.	0
zdepth	m	Depth of box sample.	0

Name	Unit	Description	Default
thickness	m	Thickness of hollow sample. Negative value extends the hollow volume outside of the box/cylinder.	0
pack	1	Packing factor	1
Vc	Å^3	Volume of unit cell=nb atoms per cell/density of atoms.	0
delta_d_d	1	Global relative Delta_d/d spreading when the 'w' column is not available. Use 0 if ideal.	0
p_inc	1	Fraction of incoherently scattered rays.	0.1
p_transmit	1	Fraction of transmitted (only attenuated) rays.	0.1
DW	1	Global Debye-Waller factor when the 'DW' column is not available. Use 1 if included in F2.	0
nb_atoms	1	Number of sub-unit per unit cell, that is ratio of sigma for chemical formula to sigma per unit cell.	1
d_omega	deg	Horizontal focus range (only for incoherent scattering), 0 for no focusing.	0
d_phi	deg	Angle corresponding to the vertical angular range to focus to, e.g. detector height. 0 for no focusing.	0
tth_sign	1	Sign of the scattering angle. If 0, the sign is chosen randomly (left and right). ONLY functional in combination with d_phi and ONLY applies to bragg lines.	0
p_interact	1	Fraction of events interacting with sample, e.g. 1-p_transmit-p_inc.	0
concentric	1	Indicate that this component has a hollow geometry and may contain other components. It should then be duplicated after the inside part (only for box, cylinder, sphere).	0
density	g/cm^3	Density of material. rho=density/weight/1e24*N_A.	0
weight	g/mol	Atomic/molecular weight of material.	0
barns	1	Flag to indicate if —F—^2 from 'reflections' is in barns or fm^2, (barns=1 for laz/cif, barns=0 for lau type files).	1

Name	Unit	Description	Default
focus_flip	1	Controls the sense of d_phi. If 0 d_phi is measured against the xz-plane. If !=0 d_phi is measured against zy-plane.	0

Links

- Component source code found in file `PowderN.comp`.
- See also: Single_crystal
- See ICSD Inorganic Crystal Structure Database
- Web Elements
- Fullprof powder refinement
- Crystallographica software (free license)
- Geomview and Object File Format (OFF)
- Java version of Geomview (display only) jroff.jar
- qhull
- powercrust
- cif2hkl https://gitlab.com/soleil-data-treatment/soleil-software-projects/cif2hkl
- material datafile obtained from http://physics.nist.gov/cgi-bin/ffast/ffast.pl

A general powder sample

The powder diffraction component **PowderN** models a powder sample with background coming only from incoherent scattering and no multiple scattering. At the users choice, a given percentage of the incoming events may be transmitted (attenuated) to model the direct beam. The component can also assume *concentric* shape, i.e. be used for describing sample environment (cryostat, sample container etc.).

The description of the powder comes from a file in one of the standard output formats LAZY, FULLPROF, or CRYSTALLOGRAPHICA.

7.3.1. Files formats: powder structures

Data files of type `lau` and `laz` in the McXtrace distribution data directory are self-documented in their header.

```
PowderN(<geometry parameters>, filename="Al.laz")
```

Other column-based file formats may also be imported e.g. with parameters such as:

```
format=Crystallographica
format=Fullprof
format={1,2,3,4,0,0,0,0}
```

In the latter case, the indices define order of columns parameters multiplicity, lattice spacing, F^2, Debye-Waller factor and intrinsic line width.

The column signification may as well explicitely be set in the data file header using any of the lines:

```
#column_j     <index of the multiplicity 'j' column>
#column_d     <index of the d-spacing 'd' column>
#column_F2    <index of the squared str. factor '|F|^2' column [b]>
#column_F     <index of the structure factor norm '|F|' column>
#column_DW    <index of the Debye-Waller factor 'DW' column>
#column_Dd    <index of the relative line width Delta_d/d 'Dd' column>
#column_inv2d <index of the 1/2d=sin(theta)/lambda 'inv2d' column>
#column_q     <index of the scattering wavevector 'q' column>
```

Other component parameters may as well be specified in the data file header with lines e.g.:

```
#V_rho        <value of atom number density [at/Angs^3]>
#Vc           <value of unit cell volume Vc [Angs^3]>
#sigma_abs    <value of Absorption cross section [barns]>
#sigma_inc    <value of Incoherent cross section [barns]>
#Debye_Waller <value of Debye-Waller factor DW>
#Delta_d/d    <value of Delta_d/d width for all lines>
#density      <value of material density [g/cm^3]>
#weight       <value of material molar weight [g/mol]>
#nb_atoms     <value of number of atoms per unit cell>
```

Further details on file formats are available in the `mxdoc` page of the component.

7.3.2. Geometry, physical properties, concentricity

The sample has the shape of a solid cylinder, radius r and height h or a box-shaped sample of size *xwidth* x *yheight* x *zdepth*. At the users choice, an inner 'hollow' can be specified using the parameter *thickness*.

PowderN can assume a *concentric* shape, i.e. can contain other components inside the inner hollow. To allow this, two almost identical copies of the PowderN components must be set up *around* the internal component(s), for example:

```
COMPONENT Cryo = PowderN(reflections="Al.laz", radius = 0.01, thickness = 0.001,
                         concentric = 1)
```

```
AT (0,0,0) RELATIVE Somewhere

COMPONENT Sample = some_other_component(with geometry FULLY enclosed in the hollow)
AT (0,0,0) RELATIVE Somewhere

COMPONENT Cryo2 = COPY(Cryo)(concentric = 0)
AT (0,0,0) RELATIVE Somewhere
```

As outlined, the first instance of PowderN *must* have `concentric = 1` and the second instance *must* have `concentric = 0`. Furthermore, the component(s) inside the hollow *must* have a geometry which can be fully contained inside the hollow.

In addition to the coherent scattering specified in the `reflections` file, absorption- and incoherent cross sections can be given using the input parameters σ_c^a and σ_i^s.

The Bragg scattering from the powder, σ_c^s is calculated from the input file, with the parameters Q, $|F(Q)|^2$, and j for the scattering vector, structure factor, and multiplicity, respectively. The volume of the unit cell is denoted Vc, while the sample packing factor is f_{pack}.

Focusing is performed by only scattering into one angular interval, $d\phi$ of the Debye-Scherrer circle. The center of this interval is located at the point where the Debye-Scherrer circle intersects the half-plane defined by the initial velocity, $\boldsymbol{v}_{\text{i}}$, and a user-specified vector, $\boldsymbol{f}$.

7.3.3. Powder scattering

An ideal powder sample consists of many small crystallites, although each crystallite is sufficiently large not to cause measurable size broadening. The orientation of the crystallites is evenly distributed, and there is thus always a large number of crystallites oriented to fulfill the Bragg condition

$$n\lambda = 2d\sin\theta, \tag{7.8}$$

where n is the order of the scattering (an integer), λ is the x-ray wavelength, d is the lattice spacing of the sample, and 2θ is the scattering angle, see figure 7.2. As all crystal orientations are realised in a powder sample, the x-rays are scattered within a *Debye-Scherrer cone* of opening angle 4θ [Bac75].

Equation (7.8) may be cast into the form

$$|\boldsymbol{Q}| = 2|\boldsymbol{k}|\sin\theta, \tag{7.9}$$

where $\mathbf{Q}$ is a vector of the reciprocal lattice, and $\mathbf{k}$ is the wave vector of the x-ray. It is seen that only reciprocal vectors fulfilling $|\boldsymbol{Q}| < 2|\boldsymbol{k}|$ contribute to the scattering. For a complete treatment of the powder sample, one needs to take into account all these $\mathbf{Q}$-values, since each of them contribute to the attenuation.

The strength of the Bragg reflections is given by their structure factors

$$\left|\sum_j b_j \exp(\boldsymbol{R}_j \cdot \boldsymbol{Q})\right|^2, \tag{7.10}$$

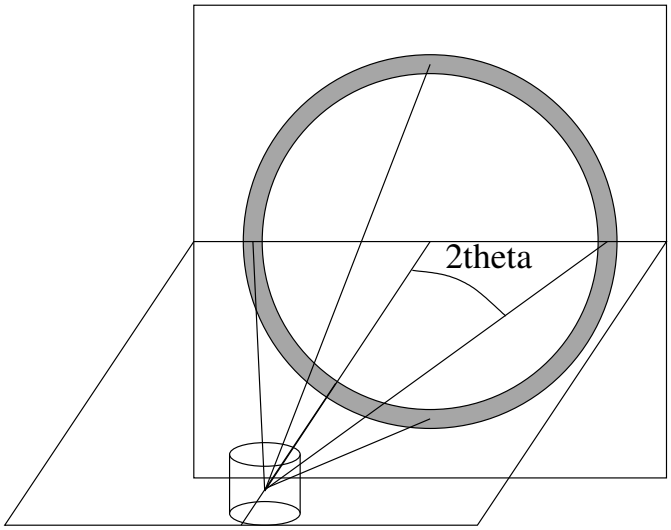


Figure 7.2.: The scattering geometry of a powder sample showing part of the Debye-Scherrer cone (solid lines) and the Debye-Scherrer circle (grey).

where the sum runs over all atoms in one unit cell. This structure factor is non-zero only when Q equals a reciprocal lattice vector.

The textbook expression for the scattering cross section corresponding to one Debye-Scherrer cone reads [Squ78, ch.3.6], with $V = NV_0$ being the total sample volume:

$$\sigma_{\text{cone}} = \frac{V}{V_0^2} \frac{\lambda^3}{4 \sin\theta} \sum_Q |F(Q)|^2. \tag{7.11}$$

For our purpose, this expression should be changed slightly. Firstly, the sum over structure factors for a particular Q is replaced by the sum over essentially different reflections multiplied by their multiplicity, j. Then, a finite packing factor, f, is defined for the powder, and finally, the Debye-Waller factor is multiplied on the elastic cross section to take lattice vibrations into account (no inelastic background is simulated, however). We then reach

$$\sigma_{\text{cone,Q}} = j_Q f \exp(-2W) \frac{V}{V_0^2} \frac{\lambda^3}{4 \sin\theta} |F(Q)|^2 \tag{7.12}$$

$$= f \exp(-2W) \frac{N}{V_0} \frac{4\pi^3}{k^2} \frac{j_Q |F(Q)|^2}{Q} \tag{7.13}$$

in the thin sample approximation. For samples of finite thickness, the beam is being attenuated by the attenuation coefficient

$$\mu_{\text{Q}} = \sigma_{\text{cone,Q}} / V. \tag{7.14}$$

For calibration it may be useful to consider the total intensity scattered into a detector of effective height h, covering only one reflection [Squ78, ch.3.6]. A cut though the

Debye-Scherrer cone perpendicular to its axis is a circle. At the distance r from the sample, the radius of this circle is $r\sin(2\theta)$. Thus, the detector (in a small angle approximation) counts a fraction $h/(2\pi r\sin(2\theta))$ of the scattered x-rays, giving a resulting count intensity:

$$I = \Psi\sigma_{\text{cone,Q}}\frac{h}{2\pi r\sin(2\theta)}, \tag{7.15}$$

where Ψ is the flux at the sample position.

For clarity we repeat the meaning and unit of the symbols:

Ψ	$\text{s}^{-1}\text{m}^{-2}$	Incoming intensity of x-rays
I	s^{-1}	Detected intensity of x-rays
h	m	Height of detector
r	m	Distance from sample to detector
f	1	Packing factor of the powder
j	1	Multiplicity of the reflection
V_0	m^3	Volume of unit cell
$\lvert F(\boldsymbol{Q})\rvert^2$	m^2	Structure factor
$\exp(-2W)$	1	Debye-Waller factor
μ_{Q}	m^{-1}	Linear attenuation factor due to scattering from one powder line.

A powder sample will in general have several allowed reflections $\boldsymbol{Q}_j$, which will all contribute to the attenuation. These reflections will have different values of $|F(\boldsymbol{Q}_j)|^2$ (and hence of Q_j), j_j, $\exp(-2W_j)$, and θ_j. The total attenuation through the sample due to scattering is given by $\mu^s = \mu^s_{\text{inc}} + \sum_j \mu^s_j$, where μ^s_{inc} represents the incoherent scattering.

7.3.4. Algorithm

The algorithm of **PowderN** can be summarized as

- Check if the x-ray intersects the sample (otherwise ignore the following).
- Calculate the attenuation coefficients for scattering and absorption.
- Perform Monte Carlo choices to determine the scattering position, scattering type (coherent/incoherent), and the outgoing direction.
- Perform the necessary weight factor transformation.

7.4. The `Single_crystal` McXtrace Component

Mosaic single crystal with multiple scattering vectors, optimised for speed with large crystals and many reflections.

Identification

- **Author:** Kristian Nielsen
- **Origin:** Risoe
- **Date:** December 1999

Description

Single crystal with mosaic. Delta-D/D option for finite-size effects. Rectangular geometry. Multiple scattering and secondary extinction included. The mosaic may EITHER be specified isotropic by setting the mosaic input parameter, OR anisotropic by setting the mosaic_a, mosaic_b, and mosaic_c parameters. If you get strange results, check the mosaicity and delta(d)/d parameters, as this component is not suited for ideal/perfect mosaic crystals. The crystal lattice can be bent locally, keeping the external geometry unchanged. Curvature is spherical along vertical and horizontal axes.

Speed/stat optimisation using SPLIT In order to dramatically improve the simulation efficiency, we recommend to use a SPLIT keyword on this component (or prior to it), as well as to disable the multiple scattering handling by setting order=1. This is especially powerful for large reflection lists such as with macromolecular proteins. When an incoming particle is identical to the preceeding, reciprocal space initialisation is skipped, and a Monte Carlo choice is done on available reflections from the last repciprocal space calculation! To assist the user in choosing a "relevant" value of the SPLIT, a rolling average of the number of available reflections is calculated and presented in the component output.

Mosacitiy modes The component features three independent ways of parametrising mosaicity: a) The original algorithm where mosaicity is implemented by extending each reflection by a Gaussian "cigar" in reciprocal space, characterised by the parameters mosaic and delta_d_d. (Also known as "isotropic mosaicity"). b) A similar mode where mosaicities can be non-isotropic and given as the parameters mosaic_a, mosaic_b and mosaic_c, around the unit cell axes. (Also known as "anisotropic mosaicity"). c) Given two "macroscopically"/experimentally measured width/mosaicities of two independent reflections, parametrised by the list mosaic_AB = {mos_a, mos_b, a_h, a_k, a_l, b_h, b_k, b_l}, a set of microscopic mosaicities as in b) are estimated (internally) and applied. (Also known as "phenomenological mosaicity").

Powder-mode When the powder mode is used (powder=0-1), a randomised transformation of the particle direction is made before and after scattering, thereby letting the single crystal behave as a crystallite of either a powder (crystallite orientation fully randomised).

Curved crystal mode The component features a method to curve the lattice planes slightly with respect to the outer geometry of the crystal. The method is implemented as a transformation on the particle direction vector, and should be used only in cases where: a) The reflection lattice vector is ˜ orthogonal to the crystal surface. b) The modelled curvarture is "small" with respect to the crystal surface.

Sample shape Sample shape may be a cylinder, a sphere, a box or any other shape:

```
box/plate:       xwidth x yheight x zdepth
cylinder:        radius x yheight
sphere:          radius (yheight=0)
any shape:       geometry=OFF/PLY file
```

The complex geometry option handles any closed non-convex polyhedra. It computes the intersection points of the photon ray with the object transparently, so that it can be used like a regular sample object. It supports the PLY, OFF and NOFF file format but not COFF (colored faces). Such files may be generated from XYZ data using: qhull < coordinates.xyz Qx Qv Tv o > geomview.off or powercrust coordinates.xyz and viewed with geomview or java -jar jroff.jar (see below). The default size of the object depends on the OFF/PLY file data, but its bounding box may be resized using xwidth,yheight and zdepth.

Crystal definition file format Crystal structure is specified with an ascii data file. Each line contains 4 or more numbers, separated by white spaces:

```
h k l ... F2
```

The first three numbers are the (h,k,l) indices of the reciprocal lattice point, and the 7-th number is the value of the structure factor —F—**2, in barns. The rest of the numbers are not used; the file is in the format output by the Crystallographica program. The reflection list should be ordered by decreasing d-spacing values. Lines begining by '#' are read as comments (ignored). Most sample parameters may be defined from the data file header, following the same mechanism as PowderN.

Current data file header keywords include, for data format specification: #column_h <index of the Bragg Qh column> #column_k <index of the Bragg Qk column> #column_l <index of the Bragg Ql column> #column_F2 <index of the squared str. factor '—F—^2' column [b]> #column_F <index of the structure factor norm '—F—' column> and for material specification: #sigma_inc <value of incoherent cross section [barns]> #Delta_d/d <value of Delta_d/d width for all lines> #lattice_a <value of the a lattice parameter [Å]> #lattice_b <value of the b lattice parameter [Å]> #lattice_c <value of the c lattice parameter [Å]> #lattice_aa <value of the alpha lattice angle [deg]> #lattice_bb <value of the beta lattice angle [deg]> #lattice_cc <value of the gamma lattice angle [deg]>

Last, CIF, FullProf and ShelX files can be read, and converted to F2(hkl) lists when 'cif2hkl' is installed. The CIF2HKL env variable can be used to point to a proper executable, else the McCode or the system installed versions are used.

Satellite Bragg peaks - surface crystal truncation rods (CTR) It is known that scattering from a finite crystal introduces a broadening of Bragg peaks, seen in surface diffraction at grazing angle [Robinson and Tweet, Rep. Prog. Phys. 55 (1992) 599)]. The CTR is specified as two vectors, which hold the squared Fourier transform of the crystal geometry, for instance:

```
bulk:       Dirac peak      (FT of infinity)
half-bulk:  Dirac+1/k^2     (FT of half plane, k in rlu)
layer:      sinc(PI*k*d)^2  (FT of a top-hat, thickness 'd')
```

These vectors are given as 'surf_k' [in 1/Å] and 'surf_FT2', both of length 'surf_size'. The CTR is to be applied along vector 'surf_dir' which indicates the surface normal {nx,ny,nz} in real space. When surf_size=-1, the component sets the proper truncation function (only for thin box and disk shapes). This feature is an approximation of the real surface scattering, and does not handle complex geometries (clusters, wetting, multilayers, ...). It may be used as well to describe over-structure satellite peaks.

Example: Single_crystal(xwidth=0.01, yheight=0.01, zdepth=0.01, mosaic = 5, reflections="Si.lau")

A diamond crystal plate, cut for (002) reflections Single_crystal(xwidth = 0.002, yheight = 0.1, zdepth = 0.1, mosaic = 5, delta_d_d=3e-4, reflections = "C-diamond.lau",

```
ax=0,      ay=2.14,   az=-1.24,
bx = 0,    by = 0,    bz =  2.47,
cx = 6.71, cy = 0,    cz =  0)
```

A adrenaline protein Single_crystal(xwidth=0.005, yheight=0.005, zdepth=0.005, mosaic = 5, reflections="adrenaline.lau")

Also, always use a non-zero value of delta_d_d.

This sample component can advantageously benefit from the SPLIT feature, e.g. SPLIT COMPONENT sx = Single_crystal(...)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
reflections	string	File name containing structure factors of reflections (LAZ LAU CIF, FullProf, ShelX). Use empty ("") or NULL for incoherent scattering only	0
geometry	str	Name of an Object File Format (OFF) or PLY file for complex geometry. The OFF/PLY file may be generated from XYZ coordinates using qhull/powercrust	0

Name	Unit	Description	Default
mosaic_AB	arc_minutes, arc_minutes, 1, 1, 1, 1, 1, 1	In-plane mosaic rotation and plane vectors (anisotropic), mosaic_A, mosaic_B, A_h,A_k,A_l, B_h,B_k,B_l. Puts the crystal in the in-plane mosaic state. Vectors A and B define plane in which the crystal roation is defined, and mosaic_A, mosaic_B, denotes the resp. mosaicities (gaussian RMS) with respect to the two reflections chosen by A and B (Miller indices).	{0,0, 0,0,0, 0,0,0}
xwidth	m	Width of crystal	0
yheight	m	Height of crystal	0
zdepth	m	Depth of crystal (no extinction simulated)	0
radius	m	Outer radius of sample in (x,z) plane	0
delta_d_d	1	Lattice spacing variance, gaussian RMS (longitudinal mosaic) e.g. 1e-4 to 1e-3.	1e-3
mosaic	arc minutes	Crystal mosaic (isotropic), gaussian RMS. Puts the crystal in the isotropic mosaic model state, thus disregarding other mosaicity parameters, e.g. 1-10.	-1
mosaic_a	arc minutes	Horizontal (rotation around lattice vector a) mosaic (anisotropic), gaussian RMS. Put the crystal in the anisotropic crystal vector state. i.e. model mosaicity through rotation around the crystal lattice vectors. Has precedence over in-plane mosaic model.	-1
mosaic_b	arc minutes	Vertical (rotation around lattice vector b) mosaic (anisotropic), gaussian RMS.	-1
mosaic_c	arc minutes	Out-of-plane (Rotation around lattice vector c) mosaic (anisotropic), gaussian RMS	-1
recip_cell	1	Choice of direct/reciprocal (0/1) unit cell definition	0
barns	1	Flag to indicate if —F—ˆ2 from 'reflections' is in barns or fmˆ2. barns=1 for laz/cif and isotropic constant elastic scattering (reflections=NULL), barns=0 for lau type files	0
ax	Å or $Å^{-1}$	Coordinates of first (direct/recip) unit cell vector	0

Name	Unit	Description	Default
ay	Å or $Å^{-1}$	a on y axis	0
az	Å or $Å^{-1}$	a on z axis	0
bx	Å or $Å^{-1}$	Coordinates of second (direct/recip) unit cell vector	0
by	Å or $Å^{-1}$	b on y axis	0
bz	Å or $Å^{-1}$	b on z axis	0
cx	Å or $Å^{-1}$	Coordinates of third (direct/recip) unit cell vector	0
cy	Å or $Å^{-1}$	c on y axis	0
cz	Å or $Å^{-1}$	c on z axis	0
p_transmit	1	Monte Carlo probability for photons to be transmitted without any scattering. Used to improve statistics from weak reflections	0.001
sigma_inc	barns	Incoherent scattering cross-section per unit cell (uniform). Fully isotropic and constant. Use -1 to inactivate	0
aa	deg	Unit cell angles alpha, beta and gamma. Then uses norms of vectors a,b and c as lattice parameters	0
bb	deg	Beta angle	0
cc	deg	Gamma angle	0
order	1	Limit multiple scattering up to given order (0: all, 1: first, 2: second, ...)	1
extra_order	1	When using order, allow additional multiple scattering without coherent scattering, sensible with very large unit cells (0: disable, 1: one extra, 2: two extra, ...)	0
RX	m	Radius of horizontal along X lattice curvature. flat for 0	0
RY	m	Radius of vertical along Y lattice curvature. flat for 0	0
powder	1	Flag to indicate powder mode, for simulation of Debye-Scherrer cones via random crystallite orientation. A powder texture can be approximated with powder within 0-1	0

Name	Unit	Description	Default
deltak	Å-1	Equality-threshold for use in SPLIT settings. If difference between all ki_{x,y,z} are less than deltak from previous particle, the two are considered alike enough to jump directly to the MC choice between 'active' reflections	1e-6
material_datafile	Be.txt	File where the material parameters for the absorption may be found. Format is similar to what may be found off the NIST website.	"Si.txt"
surf_size	1	Length of the surf_k and surf_FT vectors. When set as -1, CTR is automatically set for the box/thin disk geometry.	0
surf_k	1/Å	Momentum 'k' distribution around 0 for the CTR, length 'surf_size'.	NULL
surf_FT2	1	Intensity —FT(real space)—^2 distribution as CTR of a single Bragg peak, length 'surf_size'.	NULL

Links

- Component source code found in file `Single_crystal.comp`.
- See ICSD Inorganic Crystal Structure Database
- Web Elements
- Fullprof powder refinement
- Crystallographica software
- Geomview and Object File Format (OFF)
- Java version of Geomview (display only) jroff.jar
- qhull
- powercrust
- material datafile obtained from http://physics.nist.gov/cgi-bin/ffast/ffast.pl
- cif2hkl https://gitlab.com/soleil-data-treatment/soleil-software-projects/cif2hkl

The single crystal component

Documentation Pending

7.5. The `Molecule_2state` McXtrace Component

Disordered optical-excitable molecule sample.

Identification

- **Author:** Erik B Knudsen
- **Origin:** DTU Physics
- **Date:** October 2012

Description

A sample model for pump probe experiments which models disordered molecules in a volume (rectangular, cylindrical, or spherical). Molecules can be in one of two states (0 and 1). Scattering is either specified through F vs. q scattering curves or as a set of atom positions from which F vs. q is computed. At t=-delta_t, a fraction of the molecules are put in state 1, from which they decay exponentially, with time constant t_relax, into state 0. For t<-delta_t all of the molecules are in the state specified by *initial_state*. To improve statistics, scattering may be limited to a "forward" cone with opening angle in [psimin, psimax]. Furthermore, scattering may be restricted to the azimuthal segment between [etamin,etamax].

Example: Molecule_2state(nq=512,state_0_file="Fe_bpy_GS_DFT.txt",state_1_file="Fe_bpy_ES_DFT.txt",radi
psimin=0, psimax=15*DEG2RAD, etamin=-1*DEG2RAD,etamax=1*DEG2RAD, t_relax=600e-
12, delta_t=100e-9, excitation_yield=0.2)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
delta_t	s	Delay between the exciting event t=0. delay is negative, i.e. delta_t>0 means the exciting event happens before t=0.	100e-9
excitation_yield	1	Mean fraction of molecules that get excited.	0.2
t_relax	s	Mean relaxation time (into state 0) of excited molecules.	100e-9
initial_state	0/1	Which state is Molecule_2state in for t<delta_t? Useful for modelling something that changes state slowly.	0
psimin	rad	Minimum scattering angle off the optical axis.	0

Name	Unit	Description	Default
psimax	rad	Maximum scattering angle off the optical axis.	M_PI_2
etamin	rad	Minimum scattering angle around the optical axis.	-M_PI
etamax	rad	Maximum scattering angle around the optical axis.	M_PI
radius	m	Radius of cylindrical of spherical sample.	0
yheight	m	Height of rectangular or cylindrical sample.	0
xwidth	m	Width of rectangular sample.	0
zdepth	m	Depth (thickness) of rectangular sample.	0
concentration	m	Concentration or packing factor of sample.	1
p_transmit	m	Fraction of statistics devoted to sample direct (unscattered) beam.	0.1
form_factors	str	File from which to read atomic form factors. Defualt amounts to use the one shipped with McXtrace.	"FormFactors.txt"
state_0_file	str	Isotropic scattering factors (parameterized by q), or atom positions are specified for state 0.	NULL
state_1_file	str	Isotropic scattering factors (parameterized by q), or atom positions are specified for state 1.	NULL
nq	1	Number of q-bins if F is to be computed from atom positions (Debye formalism).	512
material_datafile	str	Where to read f1 and f2 factors from in order to handle absorption.	"Be.txt"
q_parametric	0/1	When 0: Assume that datafiles contains atom positions. 1: datafiles contains F vs. q data.	0
Emax	keV	Maximal energy for which scattering factors are computed. Must be larger than the maximal impinging energy.	80

Links

- Component source code found in file `Molecule_2state.comp`.

Excitable time-dependent sample model

Further Documentaion pending

8. Monitors and detectors

In real scattering experiments, detectors and monitors play quite different roles. One wants the detectors to be as efficient as possible, counting all photons (absorbing them in the process), while the monitors measure the intensity of the incoming beam, and must as such be almost transparent, interacting only with (roughly) 0.1-1% of the photons passing by. In computer simulations, it is of course possible to detect every xray without absorbing it or disturbing any of its parameters. Hence, the two components have very similar functions in the simulations, and we do not distinguish between them. For simplicity, they are from here on just called **monitors**.

Another important difference between computer simulations and real experiments is that one may allow the monitor to be sensitive to any xray property, as *e.g.* direction, energy, and divergence, in addition to what is found in real-world detectors (space and time). One may, in fact, let the monitor record correlations between these properties.

When a monitor detects a xray, a number counting variable is incremented: $n_i = n_{i-1} + 1$. In addition, the photon weight p_i is added to the weight counting variable: $I_i = I_{i-1} + p_i$, and the second moment of the weight is updated: $M_{2,i} = M_{2,i-1} + p_i^2$. As also discussed chapter 2, after a simulation of N rays the detected intensity (in units of photonts/sec.) is I_N, while the estimated errorbar is $\sqrt{M_{2,N}^2}$.

Several different monitor components have been developed for McXtrace, but we have decided to support only the most important ones. One example of the monitors we have omitted is the single monitor, **Monitor**, that measures just one number (with errorbars) per simulation. This effect is mirrored by any of the 1- or 2-dimensional components we support, e.g. the `PSD_monitor`. In case additional functionality of monitors is required, a few lines of code in existing monitors can easily be modified.

Another solution is the "Swiss army knife" of monitors, **Monitor_nD**, that can handle almost any simulation requirement, but may prove challenging for inexperienced users or users who like to make their own modifications.

8.1. The `Monitor` McXtrace Component

Release: McXtrace 0.1
Simple monitor.

Identification

- **Author:** Erik Knudsen
- **Origin:** Risoe
- **Date:** June 22, 2009

Description

A square single monitor that measures the intergated intensity of the incoming x-rays.
Example: Monitor(xwidth=0.1, yheight=0.1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
xwidth	m	Width of detector.	0.1
yheight	m	Height of detector.	0.1
restore_xray	m	If set, the monitor does not influence the xray state	0

Links

- Component source code found in file `Monitor.comp`.

Simple intensity monitor

The **Monitor** component is a simple photon counter that merely detects the integrated intensity in an aperture *xwidth* by *yheight* m^2. It does *not* write a separate datafile, but reports the I, I_err, N-signals to the console. The signals represent intensity, error estimate, and number of statistical events (photon rays). If a scan is performed, the intensity recorded by **Monitor** will be written to the scan datafile.

8.2. The `E_monitor` McXtrace Component

Release: McXtrace 0.1
Energy-sensitive monitor.

Identification

- **Author:** Erik Knudsen
- **Origin:** Risoe
- **Date:** June 22, 2009

Description

A square single monitor that measures the energy of the incoming x-rays.
Example: E_monitor(xwidth=0.1, yheight=0.1, Emin=1, Emax=50, nE=20, filename="Output.nrj")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
nE	m	Number of energy channels.	20
filename	str	Name of file in which to store the detector image.	0
xwidth	m	Width of detector.	0.1
yheight	m	Height of detector.	0.1
Emin	keV	Minimum energy to detect.	
Emax	keV	Maximum energy to detect.	
restore_xray	0/1	If set, the monitor does not influence the x-ray state.	0

Links

- Component source code found in file `E_monitor.comp`.

The energy-sensitive monitor

The component **E_monitor** is sensitive to the xray energy, which in binned in nE bins between $E_{\min}$ and $E_{\max}$ (in keV).

The output parameters from `E_monitor` are the total counts, and a file with 1-dimensional data vs. E, similar to `TOF_monitor`.

8.3. The `L_monitor` McXtrace Component

Release: McXtrace 0.1
Wavelength-sensitive monitor.

Identification

- **Author:** Kristian Nielsen and Kim Lefmann
- **Origin:** Risoe
- **Date:** June 22, 2009

Description

A square single monitor that measures the wavelength of the incoming xray.

Example: L_monitor(xmin=-0.1, xwidth=0.1, yheight=0.1, nL=20, filename="Output.L", Lmin=0.1, Lmax=1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
nL	m	Number of wavelength channels.	20
filename	str	Name of file in which to store the detector image.	0
xwidth	m	Width of detector.	0.1
yheight	m	Height of detector.	0.1
Lmin	Å	Minimum wavelength to detect.	
Lmax	Å	Maximum wavelength to detect.	
restore_xray		If set, the monitor does not influence the x-ray state.	0

Links

- Component source code found in file `L_monitor.comp`.

The wavelength sensitive monitor

The component **L_monitor** is very similar to **E_monitor**. This component is just sensitive to the xray wavelength. The wavelength spectrum is output in a one-dimensional histogram. between $\lambda_{\min}$ and $\lambda_{\max}$ (measured in Å).

As for the two other 1-dimensional monitors, this component outputs the total counts and a file with the histogram.

8.4. The `PSD_monitor` McXtrace Component

Position-sensitive monitor.

Identification

- **Author:** Erik B Knudsen
- **Origin:** Risoe
- **Date:** June 22, 2009

Description

Based on neutron component written by Kim Lefmann An (nx times ny) pixel PSD monitor. This component may also be used as a beam detector. If instead of xwidth, yheight a radius is given, the component has a circular footprint and integrates circularly (caking).

Example: PSD_monitor(xwidth=0.1, yheight=0.1, nx=90, ny=90, filename="Output.psd")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
filename	str	Name of file in which to store the detector image.	0
xwidth	m	Width of detector.	0.05
yheight	m	Height of detector.	0.05
radius	m	Radius of circular detetor.	0
restore_xray	1	If set, the monitor does not influence the xray state.	1
nowritefile	1	If set, monitor will skip writing to disk	0
nx	1	Number of pixel columns.	90
ny	1	Number of pixel rows.	90
nr	1	Number of radial pixels.	0

Links

- Component source code found in file `PSD_monitor.comp`.

The PSD monitor

The component `PSD_monitor` resembles other monitors, e.g. `Monitor`, and also propagates the photon ray to the detector surface in the (x, y)-plane, where the detector window is set by the x and y input coordinates. The PSD monitor is sensitive to the arrival position of the of the photon ray. The rectangular monitor window, given by the x and y limits is divided into $n_x \times n_y$ pixels.

The output from `PSD_monitor` is the integrated counts, n, I, M_2, as well as three two-dimensional arrays of counts: $n(x,y), I(x,y), M_2(x,y)$. The arrays are written to a file, `filename`, and can be read e.g. by the tool **mxplot**, see the system manual.

8.5. The `PSD_monitor_coh` McXtrace Component

Position-sensitive monitor with phase integration.

Identification

- **Author:** Erik Knudsen
- **Origin:** Risoe
- **Date:** March 13, 2010

Description

An (n times m) pixel PSD monitor taking phase into account. As the i:th ray hits a pixel (j,k) in the monitor the intensity in that pixel will be updated as a complex sum, i.e. $P_i = P_{i-1} + p_i exp{-\phi_i}$.

By setting ratio<1 the effective pixel area becomes a fraction of the ideal (which is to divide the xwidth and yheight intervals into nx and ny abutting subintervals). This reduces the monitor effective area by ratioˆ2. If the centering flag is set - the monitor will treat all rays as if they hit a pixel center. This behaves as if ratio -> 0, but at no cost in statistics.

Example: PSD_monitor_coh(xwidth=0.1, yheight=0.1, nx=90, ny=90, filename="Output.psd")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
nx		Number of pixel columns.	90
ny		Number of pixel rows.	90
filename	m	Name of file in which to store the detector images - the suffixes .abs and .arg will be added.	0
restore_xray		If set, the monitor does not influence the xray state.	0
xwidth	m	Width of detector.	0.05
yheight	m	Height of detector.	0.05
ratio		ratio between pixel area and effective pixel area.	1

Name	Unit	Description	Default
centering		Treat all rays as if they hit the center of the pixel.	1
nowritefile	1	If set, monitor will skip writing to disk	0

Links

- Component source code found in file `PSD_monitor_coh.comp`.

The coherent PSD monitor

The component `PSD_monitor_coh` closely resembles its parent the **PSD_monitor**, and also propagates the photon ray to the detector surface in the (x, y)-plane, where the detector window is set by the x and y input coordinates. The coherent PSD monitor, though, is also sensitive considers the phase of the photon ray and emits not one but two datafiles, where one (*filename*.abs) represents the coherent intensity and the other (*filename*.arg) the phase of the collected beam.

The first output file (.abs) from `PSD_monitor_coh` is the integrated counts, n, I, M_2, as well as three two-dimensional arrays of counts: $n(x, y), I(x, y), M_2(x, y)$. These arrays are written to a file, `filename.abs`, and can be read e.g. by the tool **mxplot**, see the system manual. Likewise, the second output file (.arg) contains the integrated phases, and are written to the file `filename.arg`

8.6. The `PSD_monitor_4PI` McXtrace Component

Release: McXtrace 0.1

Spherical position-sensitive detector.

Identification

- **Author:** Erik Knudsen
- **Origin:** Risoe
- **Date:** June 23rd, 2009

Description

Based on neutron component by Kim Lefmann and Kristian Nielsen An (n times m) pixel spherical PSD monitor using a cylindrical projection. Mostly for test and debugging purposes.

Example: PSD_monitor_4PI(radius=0.1, nx=90, ny=90, filename="Output.psd")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	m	Radius of detector	1
restore_xray	1	If set, the monitor does not influence the xray state	0
nx	1	Number of pixel columns	90
ny	1	Number of pixel rows	90
filename	str	Name of file in which to store the detector image	0

Links

- Component source code found in file `PSD_monitor_4PI.comp`.

A 4 PI steradian spherical monitor.

PSD_monitor_4PI is an unphysical idealized component. It takes on the shape of a complete sphere with a given *radius*, records the longitude and latitude of photon rays as the pass through the sphere. The sphere is binned in constant spaced bins in longitude and latitude. This has the consequence that the image will be distorted when plotted onto a plane, much like projections of a world map.

The output from `PSD_monitor` is the integrated counts, n, I, M_2, as well as three two-dimensional arrays of counts: $n(x,y), I(x,y), M_2(x,y)$. The arrays are written to a file, `filename`, and can be read e.g. by the tool **mxplot**, see the system manual.

8.7. The `EPSD_monitor` McXtrace Component

Position-energy-sensitive monitor.

Identification

- **Author:** Erik B Knudsen
- **Origin:** DTU
- **Date:** June 22, 2009

Description

An nx times ny pixel energy resolved PSD monitor, which only counts photons with energy in an interval given by Emin and Emax in nE energy bins. The default energy

interval is (almost) infinite, with a single bin. If nE>1 the component will output nE detector files + one which is integrated over the full energy interval.

Example: EPSD_monitor(xwidth=0.1, yheight=0.1, nx=90, ny=90, filename="Output.psd")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
nx	1	Number of pixel columns.	90
ny	1	Number of pixel rows.	90
filename	str	Name of file in which to store the detector image.	0
restore_xray	1	If set, the monitor does not influence the xray state.	0
xwidth	m	Width of detector.	0.1
yheight	m	Height of detector.	0.1
Emax	keV	Upper bound of energy interval.	0
Emin	keV	Lower bound of energy interval.	0
nE	1	Number of energy bins.	1
nowritefile	1	If set, monitor will skip writing to disk	0

Links

- Component source code found in file `EPSD_monitor.comp`.

Energy-selective PSD monitor

The **EPSD_monitor** closely resembles **PSD_monitor** with the difference that this component may be given an energy interval: $[E_{min}, E_{max}]$, in which it is sensitive. Photon rays falling within this interval are detected, those outside are ignored.

The output from **EPSD_monitor** is the integrated counts, n, I, M_2, as well as three two-dimensional arrays of counts: $n(x,y), I(x,y), M_2(x,y)$. The arrays are written to a file, `filename`, and can be read e.g. by the tool **mxplot**, see the system manual.

8.8. The `W_psd_monitor` McXtrace Component

Release: McXtrace 0.1

Position-sensitive wattage monitor.

Identification

- **Author:** Erik Knudsen

- **Origin:** Risoe
- **Date:** June 22, 2009

Description

Based on neutron PSD component written by Kim Lefmann An n times m pixel PSD wattage monitor. This component may also be used as a beam detector.
Example: W_psd_monitor(xwidth=0.1, yheight=0.1, nx=90, ny=90, filename="Output.psd")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
nx	1	Number of pixel columns	90
ny	1	Number of pixel rows	90
filename	str	Name of file in which to store the detector image	0
restore_xray	1	If set, the monitor does not influence the xray state	0
xwidth	m	Width of detector.	0.1
yheight	m	Height of detector.	0.1

Links

- Component source code found in file `W_psd_monitor.comp`.

A power vs. position monitor

doc. pend.

8.9. The Monitor_nD McXtrace Component

Release: McXtrace 1.2

This component is a general Monitor that can output 0/1/2D signals (Intensity or signal vs. [something] and vs. [something] ...)

Identification

- **Author:** Emmanuel Farhi
- **Origin:** ILL
- **Date:** 14th Feb 2000.

Description

This component is a general Monitor that can output 0/1/2D signals It can produce many 1D signals (one for any variable specified in option list), or a single 2D output (two variables correlation). Also, an additional 'list' of photon events can be produced. By default, monitor is square (in x/y plane). A disk shape is also possible The 'cylinder' and 'banana' option will change that for a banana shape The 'sphere' option simulates spherical detector. The 'box' is a box. The cylinder, sphere and banana should be centered on the scattering point. In normal configuration, the Monitor_nD measures the current parameters of the photon that is being detected. USERVARS may be used in order to study correlations between a neutron being detected in a Monitor_nD place, and given parameters that are monitored elsewhere (at the point of initialisation of the USERVARS). The monitor can also act as a 3He gas detector, taking into account the detection efficiency.

The 'bins' and 'limits' modifiers are to be used after each variable, and 'auto','log' and 'abs' come before it. (eg: auto abs log hdiv bins=10 limits=[-5 5]) When placed after all variables, these two latter modifiers apply to the signal (e.g. intensity). Unknown keywords are ignored. If no limits are specified for a given observable, reasonable defaults will be applied. Note that these implicit limits are **even** applied in list mode.

Implicit limits for typical variables: (consult monitor_nd-lib.c if you don't find your variable here) x, y, z: Derived from detection-object geometry

```
k: [0 10] Angs-1
v: [0 1e6] m/s
t: [0 1] s
```

p: [0 FLT_MAX] in intensity-units

```
vx, vy: [-1000 1000] m/s
vz: [0 10000] m/s
kx, ky: [-1 1] Angs-1
kz: [-10 10] Angs-1
```

energy, omega: [0 100] meV lambda,wavelength: [0 100] Å sx, sy, sz: [-1 1] in polarisation-units angle: [-50 50] deg divergence, vdiv, hdiv, xdiv, ydiv: [-5 5] deg longitude, lattitude: [-180 180] deg photon: [0 simulaton_ncount] id, pixel id: [0 FLT_MAX]

```
uservars u0,u1,u2,u3,u4,u5,u6,u7,u8,u9: [-1e10 1e10]
```

In the case of multiple components at the same position, the 'parallel' keyword must be used in each instance instead of defining a GROUP.

Possible options are Variables to record: kx ky kz k wavevector [Å-1] Wavevector on x,y,z and norm

```
vx vy vz v              [m/s]    Velocity on x,y,z and norm
x y z radius            [m]      Distance, Position and norm
xy, yz, xz              [m]      Radial position in xy, yz and xz plane
kxy kyz kxz             [Angs-1] Radial wavevector in xy, yz and xz plane
vxy vyz vxz             [m/s]    Radial velocity in xy, yz and xz plane
t time                  [s]      Time of Flight
energy omega            [keV]    energy of photon
lambda wavelength       [Angs]   wavelength of photon
sx sy sz                [1]      Spin
vdiv ydiv dy            [deg]    vertical divergence (y)
hdiv divergence xdiv    [deg]    horizontal divergence (x)
angle                   [deg]    divergence from <z> direction
theta longitude         [deg]    longitude (x/z) for sphere and cylinder
phi   lattitude         [deg]    lattitude (y/z) for sphere and cylinder

user0 user1             will monitor the [Mon_Name]_Vars.UserVariable{0|1|2|3|4|5}
user2 user3             to be assigned in an other component (see below)
```

user4 user5 user6 user7 user8 user9

```
Premonitoring:              Please use uservars in place of the former PreMonitor_nD.

p i                     [phts/s  or  phts/cm^2/s]  ('intensity', 'flux': name the def
ncounts n photon        [1]      photon ID, i.e current event index
pixel id                [1]      pixelID in histogram made of preceeding vars, e.g. '
```

Other options keywords are:

```
abs                         Will monitor the abs of the following variable or of the
auto                        Automatically set detector limits for one/all
all  {limits|bins|auto}     To set all limits or bins values or auto mode
binary {float|double}       with 'source' option, saves in compact files
bins=[bins=20]              Number of bins in the detector along dimension
borders                     To also count off-limits photons (X < min or X > max)
```

```
capture                    weight by capture flux (not validated)
exclusive                  absorb photon out of monitor limits
file=string                Detector image file name. default is component name, plus date and v
incoming                   Monitor incoming beam in non flat det
limits=[min max]           Lower/Upper limits for axes (see up for the variable unit)
```

list=[counts=1000] or all For a long file of photon characteristics with [counts] or all events

```
log                        Will monitor the log of the following variable or of the signal (if
min=[min_value]            Same as limits, but only sets the min or max
```

max=[max_value]

```
multiple                   Create multiple independant 1D monitors files
no or not                  Revert next option
outgoing                   Monitor outgoing beam (default)
parallel                   Use this option when the next component is at the same position (par
per cm2                    Intensity will be per cm^2 (detector area). Displays beam section.
per steradian              Intensity will be per steradian (requires auto)
signal=[var]               Will monitor [var] instead of usual intensity
slit or absorb             Absorb photons that are out detector
source                     The monitor will save photon states
inactivate                 To inactivate detector (0D detector)
verbose                    To display additional informations
```

Detector shape options (specified as xwidth,yheight,zdepth or x/y/z/min/max)

```
box                        Box of size xwidth, yheight, zdepth.
cylinder                   To get a cylindrical monitor (diameter is xwidth or set radius, heig
banana                     Same as cylinder, without top/bottom, on restricted angular area; us
disk                       Disk flat xy monitor. diameter is xwidth.
sphere                     To get a spherical monitor (e.g. a 4PI) (diameter is xwidth or set r
square                     Square flat xy monitor (xwidth, yheight).
previous                   The monitor uses PREVIOUS component as detector surface.
```

EXAMPLES: MyMon = Monitor_nD(xwidth = 0.1, yheight = 0.1, zdepth = 0, options = "intensity per cm2 angle,limits=[-5 5] bins=10,with borders, file = mon1"); will monitor photon angle from [z] axis, between -5 and 5 degrees, in 10 bins, into "mon1.A" output 1D file options = "sphere theta phi outgoing" for a sphere PSD detector (out beam) and saves into file "MyMon_[Date_ID].th_ph" options = "banana, theta limits=[10,130], bins=120, y" a theta/height banana detector

```
options = "angle radius all auto"   is a 2D monitor with automatic limits
```

options = "list=1000 kx ky kz energy" records 1000 photon event in a file options = "multiple kx ky kz, auto abs log t, and list all photons" makes 4 output 1D files and produces a complete list for all photons and monitor log(abs(tof)) within automatic limits (for t) options = "theta y, sphere, pixel min=100" a 4pi detector which outputs an event list with pixelID from the actual detector surface, starting from index 100.

To dynamically define a number of bins, or limits:

```
Use in DECLARE:    char op[256];
```

Use in INITIALIZE: sprintf(op, "lambda limits=[%g %g], bins=%i", lmin, lmax, lbin);

```
Use in TRACE:      Monitor_nD(... options=op ...)
```

How to monitor any instrument/component variable into a Monitor_nD Suppose you want to monitor a variable 'age' which you assign somwhere in the instrument: COMPONENT MyMonitor = Monitor_nD(xwidth = 0.1, yheight = 0.1, user1="age", username1="Age of the Captain [years]", options="user1, auto")

```
AT ...
```

%BUGS The 'auto' option for guessing optimal variable bounds should NOT be used with MPI as each process may use different limits.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
user0	str	Variable name of USERVAR to be monitored by user0.	""
user1	str	Variable name of USERVAR to be monitored by user1.	""
user2	str	Variable name of USERVAR to be monitored by user2.	""
user3	str	Variable name of USERVAR to be monitored by user3.	""
user4	str	Variable name of USERVAR to be monitored by user4.	""
user5	str	Variable name of USERVAR to be monitored by user5.	""
user6	str	Variable name of USERVAR to be monitored by user6.	""
user7	str	Variable name of USERVAR to be monitored by user7.	""

Name	Unit	Description	Default
user8	str	Variable name of USERVAR to be monitored by user8.	""
user9	str	Variable name of USERVAR to be monitored by user9.	""
xwidth	m	Width of detector.	0
yheight	m	Height of detector.	0
zdepth	m	Thickness of detector (z).	0
bins	1	Number of bins to force for all variables. Use 'bins' keyword in 'options' for heterogeneous bins	0
min	u	Minimum range value to force for all variables. Use 'min' or 'limits' keyword in 'options' for other limits	-1e40
max	u	Maximum range value to force for all variables. Use 'max' or 'limits' keyword in 'options' for other limits	1e40
restore_xray	0—1	If set, the monitor does not influence the photon state. Equivalent to setting the 'parallel' option.	0
radius	m	Radius of sphere/banana shape monitor	0
options	str	String that specifies the configuration of the monitor. The general syntax is "[x] options..." (see Descr.).	"NULL"
filename	str	Output file name (overrides file=XX option).	"NULL"
geometry	str	Name of an OFF file to specify a complex geometry detector	"NULL"
nowritefile	1	If set, monitor will skip writing to disk	0
username0	str	Name assigned to User0	"NULL"
username1	str	Name assigned to User1	"NULL"
username2	str	Name assigned to User2	"NULL"
username3	str	Name assigned to User3	"NULL"
username4	str	Name assigned to User4	"NULL"
username5	str	Name assigned to User5	"NULL"
username6	str	Name assigned to User6	"NULL"
username7	str	Name assigned to User7	"NULL"
username8	str	Name assigned to User8	"NULL"
username9	str	Name assigned to User9	"NULL"

Links

- Component source code found in file `Monitor_nD.comp`.

A general Monitor for 0D/1D/2D records

The component **Monitor_nD** is a general Monitor that may output any set of physical parameters regarding the passing photons. The generated files are either a set of 1D signals ([Intensity] *vs.* [Variable]), or a single 2D signal ([Intensity] *vs.* [Variable 1] *vs.* [Variable 2]), and possibly a simple long list of selected physical parameters for each photon ray.

The input parameters for **Monitor_nD** are its dimensions *xmin, xmax, ymin, ymax* (in metres) and an *options* string describing what to detect, and what to do with the signals, in clear language. The *xwidth, yheight, zdepth* may also be used to enter dimensions.

Eventhough the possibilities of Monitor_nD are numerous, its usage remains as simple as possible, specially in the *options* parameter, which 'understands' normal language. The formatting of the *options* parameter is free, as long as it contains some specific keywords, that can be sometimes followed by values. The *no* or *not* option modifier will revert next option. The *all* option can also affect a set of monitor configuration parameters (see below).

As the usage of this component enables to monitor virtually anything, and thus the combinations of options and parameters is infinite, we shall only present the most basic configuration. The reader should refer to the on-line component help, using e.g. `mcdoc Monitor_nD.comp`.

8.9.1. The Monitor_nD geometry

The monitor shape can be selected among seven geometries:

1. (*square*) The default geometry is flat rectangular in (xy) plane with dimensions $x_{\min}, x_{\max}, y_{\min}$, $y_{\max}$, or x_{width}, y_{height}.

2. (*box*) A rectangular box with dimensions $x_{width}, y_{height}, z_{depth}$.

3. (*disk*) When choosing this geometry, the detector is a flat disk in (xy) plane. The radius is then

$$\text{radius} = \max(\text{abs } [x_{\min}, x_{\max}, y_{\min}, y_{\max}, x_{width}/2, y_{height}/2]). \tag{8.1}$$

4. (*sphere*) The detector is a sphere with the same radius as for the *disk* geometry.

5. (*cylinder*) The detector is a cylinder with revolution axis along y (vertical). The radius in (xz) plane is

$$\text{radius} = \max(\text{abs } [x_{\min}, x_{\max}, x_{width}/2]), \tag{8.2}$$

   and the height along y is

$$\text{height} = |y_{\max} - y_{\max}| \text{or} y_{height}. \tag{8.3}$$

6. (*banana*) The same as the cylinder, but without the top/bottom caps, and on a restricted angular range. The angular range is specified using a `theta` variable limit specification in the `options`.

7. (*previous*) The detector has the shape of the previous component. This may be a surface or a volume. In this case, the photon is detected on the previous component, and there is no photon propagation.

By default, the monitor is flat, rectangular. Of course, you can choose the orientation of the **Monitor_nD** in the instrument description file with the usual `ROTATED` modifier.

For the *box*, *sphere* and *cylinder*, the outgoing photons are monitored by default, but you can choose to monitor incoming photons with the *incoming* option.

At last, the *slit* or *absorb* option will ask the component to absorb the photons that do not intersect the monitor. The *exclusive* option word removes photons which are similarly outside the monitor limits (that may be other than geometrical).

The *parallel* option keyword is of common use in the case where the **Monitor_nD** is superposed with other components. It ensures that photons are detected independently of other geometrical constrains. This is generally the case when you need e.g. to place more than one monitor at the same place.

8.9.2. The photon parameters that can be monitored

There are many different variables that can be monitored at the same time and position. Some can have more than one name (e.g. `energy` or `omega`).

```
1   kx ky kz k wavevector [Angs-1] (    usually axis are
2   vx vy vz v            [m/s]        x=horz., y=vert., z=on axis)
3   x y z                 [m]      Distance, Position
4   kxy vxy xy radius     [m]      Radial wavevector, velocity and position
5   t time                [s]      Time of Flight
6   energy omega          [meV]
7   lambda wavelength     [Angs]
8   p intensity flux      [n/s] or [n/cm^2/s]
9   ncounts               [1]
10  sx sy sz              [1]      Spin
11  vdiv ydiv dy          [deg]    vertical divergence (y)
12  hdiv divergence xdiv  [deg]    horizontal divergence (x)
13  angle                 [deg]    divergence from  direction
14  theta longitude       [deg]    longitude (x/z) [for sphere and cylinder
        ]
15  phi   lattitude       [deg]    lattitude (y/z) [for sphere and cylinder
        ]
```

as well as four other special variables

```
1   user user1            will monitor the [Mon_Name]_Vars.UserVariable
        {1|2}
2   user2 user3           to be assigned in an other component (see below)
```

To tell the component what you want to monitor, just add the variable names in the *options* parameter. The data will be sorted into *bins* cells (default is 20), between some default *limits*, that can also be set by user. The *auto* option will automatically determine what limits should be used to have a good sampling of signals.

8.9.3. Important options

Each monitoring records the flux (sum of weights p) versus the given variables, except if the `signal=<variable>` word is used in the `options`.

The `auto` option is probably the most useful one: it asks the monitor to determine automatically the best limits for each variable, in order to obtain the most significant monitored histogram. This option should preceed each variable, or be located after all variables in which case they are all affected. On the other hand, one may manually set the limits with the `limits=[min max]` option. If no limits are set monitor_nd uses predefined limits that usually make sense for most x-ray scattering simulations. Example: the default upper energy limit is 100 meV, but may be changed with an options string like `options="energy limits 0 200"`. Note that the limits also apply in list mode (see below).

The `log` and `abs` options should be positioned before each variable to specify logarithmic binning and absolute value respectively.

The *borders* option will monitor variables that are outside the limits. These values are then accumulated on the 'borders' of the signal.

8.9.4. The output files

By default, the file names will be the component name, followed by a time stamp and automatic extensions showing what was monitored (such as `MyMonitor.x`). You can also set the filename in *options* with the *file* keyword followed by the file name that you want. The extension will then be added if the name does not contain a dot (.). Finally, the $filename$ parameter may also be used.

The output files format are standard 1D or 2D McXtrace detector files. The *no file* option will *inactivate* monitor, and make it a single 0D monitor detecting integrated flux and counts. The *verbose* option will display the nature of the monitor, and the names of the generated files.

The 2D output

When you ask the **Monitor_nD** to monitor only two variables (e.g. *options* = "x y"), a single 2D file of intensity versus these two correlated variables will be created.

The 1D output

The **Monitor_nD** can produce a set of 1D files, one for each monitored variable, when using 1 or more than 2 variables, or when specifying the *multiple* keyword option.

McXtrace monitor	Monitor_nD equivalent
E_monitor	*options*="energy bins=*nchan* limits=[$E_{min}E_{max}$]"
EPSD_monitor	*options*="energy bins=n_E limits=[$E_{min}E_{max}$], x bins=nx" $xmin$=x_{min} $xmax$=x_{max}
L_monitor	*options*="lambda bins=nh limits=[$-\lambda_{max}/2\lambda_{max}/2$]" *filename*=$file$
Monitor	*options*="inactivate"
PSD_monitor_4PI	*options*="theta y, sphere"
PSD_monitor	*options*="x bins=nx, y bins=ny" $xmin$=x_{min} $xmax$=x_{max} $ymin$=y_{min} $ymax$=y_{max} *filename*=$file$

Table 8.10.: Using Monitor_nD in place of other components. All limits specifications may be advantageously replaced by an *auto* word preceeding each monitored variable. Not all file and dimension specifications are indicated (e.g. filename, xmin, xmax, ymin, ymax).

The List output

The **Monitor_nD** can additionally produce a *list* of variable values for photons that pass into the monitor. This feature is additive to the 1D or 2D output. By default only 1000 events will be recorded in the file, but you can specify for instance "*list* 3000 photons" or "*list all* photons". This last option may require a lot of memory and generate huge files. Note that the limits to the measured parameters also apply in this mode. To exemplify, a monitor_nd instance with the option string `"list all k"` will *only* record those photons which have a below 2000 AA^{-1}, whereas an instance with the option string `"list all kx ky kz 0 2000"` will record all photons with $|k_x, k_y| < 2000$ AA^{-1} and $0 < v_z < 2000$ AA^{-1}. Thus, in this latter case, any neutron travelling in the negative z-direction will be disregarded.

8.9.5. Monitor equivalences

In the following table 8.10, we show how the Monitor_nD may substitute any other McXtrace monitor.

8.9.6. Usage examples

- ```
  COMPONENT MyMonitor = Monitor_nD(
      xmin = -0.1, xmax = 0.1,
      ymin = -0.1, ymax = 0.1,
      options = "energy auto limits")
  ```
  will monitor the photon energy in a single 1D file (a kind of E_monitor)
- `options = "banana, theta limits=[10,130], bins=120, y bins=30"`
  is a theta/height banana detector.
  ```

- `options = "banana, theta limits=[10,130], auto time"`
  is a theta/time-of-flight banana detector.

- `options="x bins=30 limits=[-0.05 0.05] ; y"`
  will set the monitor to look at x and y. For y, default bins (20) and limits values (monitor dimensions) are used.

- `options="x y, auto, all bins=30"`
  will determine itself the required limits for x and y.

- `options="multiple x bins=30, y limits=[-0.05 0.05], all auto"`
  will monitor the photon x and y in two 1D files.

- `options="x y z kx ky kz, all auto"`
  will monitor each of these variables in six 1D files.

- `options="x y z kx ky kz, list all, all auto"`
  will monitor all these photons' variables in one long list, one row per photon event.

- `options="multiple x y z kx ky kz, and list 2000, all auto"`
  will monitor all the photons' variables in one list of 2000 events and in six 1D files.

- `options="signal=energy, x y"`
  is a PSD monitor recording the mean energy of the beam as a function of x and y.

8.9.7. Monitoring user variables

There are two ways to monitor any quantity with Monitor_nD. This may be e.g. the number of reflections in a mirror system, or the wavevector and energy transfer at a sample. The only requirement is to define the `user1` (and optionally `user2,user3`) variables of a given Monitor_nD instance.

Directly setting the user variables (simple)

The first method uses the `user1` and `username1` component parameters to directly transfer the value and label, such as in the following example:

```
TRACE
(...)
COMPONENT UserMonitor = Monitor\_nD(
  user1    = log(t), username1="Log(time)",
  options  ="auto user1")
```

The values to assign to `user2` and `user3` must be global instrument variables, or a component output variable as in `user1=MC_GETPAR(some_comp, outpar)`. Similarly, the `user2,user3` and `username2,username3` parameters may be used to control the second and third user variable, to produce eventually 2D/3D user variable correlation data and custom event lists.

Setting indirectly the user variables (only for professionals)

It is possible to control the user variables of a given Monitor_nD instance anywhere in the instrument description. This method requires more coding, but has the advantage that a variable may be defined to store the result of a computation locally, and then transfer it into the UserMonitor, all fitting in an EXTEND block.

This is performed in a 4-step process:

1. Declare that you intend to monitor user variables in a Monitor_nD instance (defined in TRACE):

```
DECLARE
%{ (...)
  %include "monitor_nd-lib"
  MONND\_DECLARE(UserMonitor); // will monitor custom things in
      UserMonitor
%}
```

2. Initialize the label of the user variable (optional):

```
INITIALIZE
%{
   (...)
  MONND\_USER\_TITLE(UserMonitor, 1, "Log(time)");
%}
```

   The value '1' could be '2' or '3' for the `user2,user3` variable.

3. Set the user variable value in a TRACE component EXTEND block:

```
TRACE
(...)
COMPONENT blah = blah\_comp(...)
EXTEND
%{  // attach a value to user1 in UserMonitor, could be much more
    complex here.
  MONND\_USER\_VALUE(UserMonitor, 1, log(t));
%}
(...)
```

4. Tell the Monitor_nD instance to record user variables:

```
TRACE
(...)
COMPONENT UserMonitor = Monitor\_nD(options="auto user1")
(...)
```

Setting the user variable values may either make use of the photon parameters (x,y,z, vx,vy,vz, phi, t, Ex,Ey,Ez, p), access the internal variables of the component that sets the user variables (in this example, those from the `blah` instance), access any component OUTPUT parameter using the `MC_GETPAR` C macro(see chapter B), or simply use a global instrument variable. Instrument parameters can not be used directly.

9. Special-purpose components

The chapter deals with components that are not easily included in any of the other chapters because of their special nature, but which are still part of the McXtrace system.

One part of these components deals with splitting simulations into two (or more) stages. For example, the front end of a beamline is often not changed much, and a long simulation of photon rays "surviving" through initial optics could be reused for several simulations of the beamline back-end, speeding up the simulations by (typically) one or two orders of magnitude. The components for doing this trick is **Virtual_input** and **Virtual_output**, which stores and reads photon rays, respectively.

For integration with SHADOW [Rio+11] the components **Shadow_inpt** and **Shadow_output** have been written.

Progress_bar is a simulation utility that displays the simulation status, but assumes the form of a component.

9.1. The `Progress_bar` McXtrace Component

Release: McXtrace 1.0
A simulation progress bar

Identification

- **Author:** Emmanuel Farhi
- **Origin:** ILL
- **Date:** 2009

Description

An indicator of the progress of the simulation, monitoring the Init, Trace with the achieved percentage, and the Finally section. Intermediate savings (e.g. triggered by USR2 signal) are also shown. This component should be positioned at the very begining of the instrument The profile option will save the intensity and number of events for each component It may be used to evaluate the simulation efficiency.
Example: Progress_bar(percent=10,flag_save=1) AT (0,0,0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
profile	str	file name to save the simulation profile in. If set to "", it is set to the name of the instrument.	"NULL"
percent	0-100	percentage interval between updates. Default is 10%.	10
flag_save	0—1	flag to enable intermediate saving for all monitors	0
minutes	min	time in minutes between updates (Overrides percent flag).	0

Links

- Component source code found in file `Progress_bar.comp`.

Dynamic information output

The component **Progress_bar** is a special version of an Arm and has no effect on the x-ray. Only one instance of progress bar is allowed - usually it is the very first component of a beamline simulation.

If *percent*> 0 (default is 10%) the porgress bar prints an update when approximately *percent %* of the total number of rays have been simulated. If *minutes*> 0 (overrides *percen*) this update happens approx. every *minutes* min. If *flag_save*≠ 0, the update process also includes saving the status of all monitors to disk. This is useful if something is likely to interrupt the progress of long simulations ot avoid loss of data. *profile* may be set to save the simulation profile to a filename other than the name of the instrument (the default).

9.2. The `Text` McXtrace Component

Release: McXtrace 3.9
Draw a text

Identification

- **Author:** Emmanuel Farhi
- **Origin:** SOLEIL
- **Date:** Oct 10, 2026

Description

A text with given size and orientation. The text is drawn in upper letters.
Example: Text(text="SOLEIL", size=0.5, plane="XY")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
text	str	Text to draw.	
size	m	Size of text.	0.5
plane	str	"XY","YZ",or "XZ"	"XY"
pen_thickness	1	Stroke diameter as fraction of text size. 0 draws thin lines, >0 draws each stroke as a cylinder with spherical joints.	0.1

Links

- Component source code found in file `Text.comp`.

9.3. The `Shadow_input` McXtrace Component

Release: McXtrace 0.1

Read x-ray state parameters from SHADOW x-ray event file.

Identification

- **Author:** Andrea Prodi
- **Origin:** Risoe/ILL
- **Date:** November 21, 2011

Description

Source-like component reading x-ray state parameters from a SHADOW x-ray event file. Used to interface McXtrace components or simulations into SHADOW.

Example: Shadow_input(file="MySource.00", bufsize = 10000, repeat_count = 2)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
file	string	Filename of x-ray file to read. Default (NULL) is standard input. Empty string "" unactivates component	""
bufsize	records	Size of x-ray input buffer	10000
repeat_count	1	Number of times to repeat each x-ray read	1

Links

- Component source code found in file `Shadow_input.comp`.

Reading input from Shadow

Documentation pending

9.4. The `Shadow_output` McXtrace Component

Release: McXtrace 0.1

Write x-ray state parameters to SHADOW x-ray event file.

Identification

- **Author:** Andrea Prodi
- **Origin:** Risoe/ILL
- **Date:** November 21, 2011

Description

Detector-like component writing x-ray state parameters to a SHADOW x-ray file. Used to interface McXtrace components or simulations into SHADOW. Each photon is 104 bytes.

Note that when standard output is used, as is the default, no monitors or other components that produce terminal output must be used, or the x-ray output from this component will become corrupted.

Example: Shadow_output(file="MySource.vit", bufsize = 10000, progress = 1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
file	string	Filename of x-ray file to write. Default is standard output	""
bufsize	records	Size of x-ray output buffer	1000
progress	flag	If not zero, output dots as progress indicator	0

Links

- Component source code found in file `Shadow_output.comp`.

Saving the photon rays for use with SHADOW

Documentation pending.

10. The Union framework: sample environments and multiple scattering

The Union framework, originally contributed by Mads Bertelsen for McStas and ported to McXtrace, is a set of components that together allow the description of complex, possibly nested or overlapping, sample environments with multiple scattering, by separating the description of *geometry* (where matter is) from *physical process* (what happens to a photon there, e.g. Compton or Rayleigh scattering, photoelectric absorption, or powder diffraction). For a conceptual introduction to the framework – volumes, priority, masks, exit volumes, and the underlying propagation algorithm – see chapter A. This chapter documents the input parameters of every individual Union component.

10.1. Union framework core: initialization, master, and termination

10.2. The `Union_init` McXtrace Component

Deprecated no-op placeholder, kept for backward compatibility.

Identification

- **Author:** Mads Bertelsen
- **Origin:** ESS DMSC
- **Date:** 20.08.15

Description

Part of the Union components, a set of components that work together and thus sperates geometry and physics within McXtrace.

This component used to be required as the FIRST Union component. It defined the shared library (union-lib.c) and owned the shared global lists, which the other Union components reached via COMP_GETPAR3.

That is no longer needed: every Union component loads the Union library (union-lib) itself, and the library owns the shared lists. The Union_init and Union_stop components are therefore optional.

This definition is kept only so that existing instruments that still place a Union_init component continue to compile. It emits a deprecation warning and has no effect.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default

Links

- Component source code found in file `Union_init.comp`.
- **/

10.3. The `Union_master` McXtrace Component

The Master Union assembles specifications (e.g. processes, materials, geometries).

Identification

- **Author:** Mads Bertelsen and Erik B Knudsen
- **Origin:** ESS DMSC & DTU Physics
- **Date:** 20.08.15

Description

Part of the Union components, a set of components that work together and thus sperates geometry and physics within McXtrace. The use of this component requires other components to be used.

1) One specifies a number of processes using process components 2) These are gathered into material definitions using Union_make_material 3) Geometries are placed using Union_box/cylinder/sphere, assigned a material 4) This master component placed after all of the above

Only in step 4 will any simulation happen, and per default all geometries defined before this master, but after the previous will be simulated here.

There is a dedicated manual available for the Union_components

Algorithm: Described elsewhere

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
verbal	0/1	Toogles terminal output describing the defined simulation	1
list_verbal	0/1	Toogles information of all internal lists in intersection network	0
finally_verbal	0/1	Toogles information about cleanup performed in finally section	0
allow_inside_start	0/1	Set to 1 if rays are expected to start inside a volume in this master	0
enable_tagging	0/1	Enable tagging of ray history (geometry, scattering process)	0
history_limit	1	Limit the number of unique histories that are saved	300000
enable_conditionals	0/1	Use conditionals with this master	1
inherit_number_of_scattering_events	0/1	Inherit the number of scattering events from last master	0
display_mode	0/1/2	Volume display in mcdisplay: 0 follows –trace, 1 forces lines, 2 polyhedra (only with –trace=2, –trace=1 always gives lines)	0
init	string	Deprecated and unused. Accepted so that instruments written for McStas/McXtrace 3.8.7 and earlier, which name the Union_init component here, still compile.	""

Links

- Component source code found in file `Union_master.comp`.

10.4. The `Union_stop` McXtrace Component

Deprecated no-op placeholder, kept for backward compatibility.

Identification

- **Author:** Mads Bertelsen and Erik B Knudsen
- **Origin:** ESS DMSC & DTU Physics
- **Date:** 20.08.15

Description

Part of the Union components, a set of components that work together and thus separates geometry and physics within McXtrace.

This component used to be required as the LAST component. It pulled in the dispatch switch (union-suffix.c) so it would be compiled after all the process components had defined their PROCESS DETECTOR macros.

That is no longer needed: each process registers its own case with the dispatch in Union_master, whose TRACE code is compiled after every component SHARE section. The Union_init and Union_stop components are therefore optional.

This definition is kept only so that existing instruments that still place a Union_stop component continue to compile. It emits a deprecation warning and has no effect.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default

Links

- Component source code found in file `Union_stop.comp`.
- **********/

10.5. Union geometry components

10.6. The `Union_box` McXtrace Component

Implementation of a box geometry - to be filled with a material

Identification

- **Author:** Mads Bertelsen and Erik B Knudsen
- **Origin:** ESS DMSC & DTU Physics
- **Date:** 20.08.15

Description

Part of the Union components, a set of components that work together and thus separates geometry and physics within McXtrace. The use of this component requires other components to be used.

1) One specifies a number of processes using process components 2) These are gathered into material definitions using Union_make_material 3) Geometries are placed using Union_box/cylinder/sphere, assigned a material 4) A Union_master component placed after all of the above

Only in step 4 will any simulation happen, and per default all geometries defined before this master, but after the previous will be simulated here.

There is a dedicated manual available for the Union components

The position of this component is the center of the box, extending xwidth/2, yheight/2, and zdepth/2 in each direction respectively.

It is allowed to overlap components, but it is not allowed to have two parallel planes that coincide. This will crash the code on run time.

Algorithm: Described elsewhere

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
material_string	str	material name of this volume, defined using Union_make_material	0
priority	1	priotiry of the volume (can not be the same as another volume) A high priority is on top of low.	
xwidth	m	width of the box volume	
yheight	m	height of the box volume	
zdepth	m	depth of the box volume	
xwidth2	m	optional different width at the +z box face	-1
yheight2	m	optional different height at the +z box face	-1
visualize	1	set to 0 if you wish to hide this geometry in mcdisplay	1
target_index	1	Focuses on component a component this many steps further in the component sequence	0
target_x	m	\	0
target_y	m	- Position of target to focus at	0
target_z	m	/	0
focus_aw	deg	horiz. angular dimension of a rectangular area	0
focus_ah	deg	vert. angular dimension of a rectangular area	0
focus_xw	m	horiz. dimension of a rectangular area	0

Name	Unit	Description	Default
focus_xh	m	vert. dimension of a rectangular area	0
focus_r	m	focusing on circle with this radius	0
p_interact	1	probability to interact with this geometry [0-1]	0
mask_string	str	Comma separated list of geometry names which this geometry should mask	0
mask_setting	str	"All" or "Any", should the masked volume be simulated when the ray is in just one mask, or all.	0
number_of_activations	1	Number of subsequent Union_master components that will simulate this geometry	1
init	string	Deprecated and unused. Accepted so that instruments written for McStas/McXtrace 3.8.7 and earlier, which name the Union_init component here, still compile.	""

Links

- Component source code found in file `Union_box.comp`.

10.7. The `Union_cylinder` McXtrace Component

Implementation of a cylinder geometry - to be filled with a material

Identification

- **Author:** Mads Bertelsen and Erik B Knudsen
- **Origin:** ESS DMSC & DTU Physics
- **Date:** 20.08.15

Description

Part of the Union components, a set of components that work together and thus separates geometry and physics within McXtrace. The use of this component requires other components to be used.

1) One specifies a number of processes using process components 2) These are gathered into material definitions using Union_make_material 3) Geometries are placed using Union_box/cylinder/sphere, assigned a material 4) A Union_master component placed after all of the above

Only in step 4 will any simulation happen, and per default all geometries defined before this master, but after the previous will be simulated here.

There is a dedicated manual available for the Union components

The position of this component is the center of the cylinder, and it thus extends yheight/2 up and down along y axis.

It is allowed to overlap components, but it is not allowed to have two parallel planes that coincide. This will crash the code on run time.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
material_string	str	material name of this volume, defined using Union_make_material	0
priority	1	priority of the volume (can not be the same as another volume) A high priority is on top of low.	
radius	m	Outer radius volume in (x,z) plane	
yheight	m	Cylinder height in (y) direction	
visualize	1	set to 0 if you wish to hide this geometry in mcdisplay	1
target_index	1	Focus on a component this many steps further in the component sequence.	0
target_x	m	\	0
target_y	m	- Position of target to focus at	0
target_z	m	/	0
focus_aw	deg	horiz. angular dimension of a rectangular area	0
focus_ah	deg	vert. angular dimension of a rectangular area	0
focus_xw	m	horiz. dimension of a rectangular area	0
focus_xh	m	vert. dimension of a rectangular area	0
focus_r	m	focusing on circle with this radius	0
p_interact	1	probability to interact with this geometry [0-1]	0
mask_string	str	Comma seperated list of geometry names which this geometry should mask	0
mask_setting	str	"All" or "Any", should the masked volume be simulated when the ray is in just one mask, or all.	0

Name	Unit	Description	Default
number_of_activations	1	Number of subsequent Union_master components that will simulate this geometry	1
init	string	Deprecated and unused. Accepted so that instruments written for McStas/McXtrace 3.8.7 and earlier, which name the Union_init component here, still compile.	""

Links

- Component source code found in file `Union_cylinder.comp`.

10.8. The `Union_sphere` McXtrace Component

Implementation of a sphere geometry - to be filled with a material

Identification

- **Author:** Mads Bertelsen and Erik B Knudsen
- **Origin:** ESS DMSC & DTU Physics
- **Date:** 20.08.15

Description

Part of the Union components, a set of components that work together and thus separates geometry and physics within McXtrace. The use of this component requires other components to be used.

1) One specifies a number of processes using process components 2) These are gathered into material definitions using Union_make_material 3) Geometries are placed using Union_box/cylinder/sphere, assigned a material 4) A Union_master component placed after all of the above

Only in step 4 will any simulation happen, and per default all geometries defined before this master, but after the previous will be simulated here.

There is a dedicated manual available for the Union components

The position of this component is the center of the sphere.

It is allowed to overlap components, but it is not allowed to have two parallel planes that coincide. This will crash the code on run time.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
material_string	str	material name of this volume, defined using Union_make_material	0
priority	1	priority of the volume (can not be the same as another volume) A high priority is on top of low.	
radius	m	Radius of sphere	
visualize	1	set to 0 if you wish to hide this geometry in mcdisplay	1
target_index	1	Focus on a component this many steps further in the component sequence.	0
target_x	m	\	0
target_y	m	- Position of target to focus at	0
target_z	m	/	0
focus_aw	deg	horiz. angular dimension of a rectangular area	0
focus_ah	deg	vert. angular dimension of a rectangular area	0
focus_xw	m	horiz. dimension of a rectangular area	0
focus_xh	m	vert. dimension of a rectangular area	0
focus_r	m	focusing on circle with this radius	0
p_interact	1	probability to interact with this geometry [0-1]	0
mask_string		Comma separated list of geometry names which this geometry should mask	0
mask_setting		"All" or "Any", should the masked volume be simulated when the ray is in just one mask, or all.	0
number_of_activations	1	Number of subsequent Union_master components that will simulate this geometry	1
init	string	Deprecated and unused. Accepted so that instruments written for McStas/McXtrace 3.8.7 and earlier, which name the Union_init component here, still compile.	""

Links

- Component source code found in file `Union_sphere.comp`.

10.9. The `Union_cone` McXtrace Component

Cone geometry component for Union components

Identification

- **Author:** Mads Bertelsen and Erik B Knudsen
- **Origin:** ESS DMSC & DTU Physics
- **Date:** 20.08.15

Description

Part of the Union components, a set of components that work together and thus separates geometry and physics within McXtrace. The use of this component requires other components to be used.

1) One specifies a number of processes using process components 2) These are gathered into material definitions using Union_make_material 3) Geometries are placed using Union_box/cylinder/sphere, assigned a material 4) A Union_master component placed after all of the above

Only in step 4 will any simulation happen, and per default all geometries defined before this master, but after the previous will be simulated here.

There is a dedicated manual available for the Union components

The position of this component is the center of the cone, and it thus extends yheight/2 up and down along y axis.

It is allowed to overlap components, but it is not allowed to have two parallel planes that coincide. This will crash the code on run time.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
material_string	string	Material name of this volume, defined using Union_make_material	0
priority	1	Priotiry of the volume (can not be the same as another volume) A high priority is on top of low.	
radius	m	Radius volume in (x,z) plane	0
radius_top	m	Top radius volume in (x,z) plane	0
radius_bottom	m	Bottom radius volume in (x,z) plane	0
yheight	m	Cone height in (y) direction	
visualize	1	Set to 0 if you wish to hide this geometry in mcdisplay	1

Name	Unit	Description	Default
target_index	1	Focuses on component a component this many steps further in the component sequence	0
target_x	m	\	0
target_y	m	- Position of target to focus at	0
target_z	m	/	0
focus_aw	deg	Horiz. angular dimension of a rectangular area	0
focus_ah	deg	Vert. angular dimension of a rectangular area	0
focus_xw	m	Horiz. dimension of a rectangular area	0
focus_xh	m	Vert. dimension of a rectangular area	0
focus_r	m	Focusing on circle with this radius	0
p_interact	1	Probability to interact with this geometry [0-1]	0
mask_string	string	Comma seperated list of geometry names which this geometry should mask	0
mask_setting	string	"All" or "Any", should the masked volume be simulated when the ray is in just one mask, or all.	0
number_of_activations	1	Number of subsequent Union_master components that will simulate this geometry	1
init	string	Deprecated and unused. Accepted so that instruments written for McStas/McXtrace 3.8.7 and earlier, which name the Union_init component here, still compile.	""

Links

- Component source code found in file `Union_cone.comp`.

10.10. Union material definition

10.11. The `Union_make_material` McXtrace Component

Component that takes a number of Union processes and constructs a Union material

Identification

- **Author:** Mads Bertelsen and Erik B Knudsen

- **Origin:** ESS DMSC & DTU Physics
- **Date:** 20.08.15

Description

Part of the Union components, a set of components that work together and thus separates geometry and physics within McXtrace. The use of this component requires other components to be used.

1) One specifies a number of processes using process components 2) These are gathered into material definitions using this component 3) Geometries are placed using Union_box/cylinder/sphere, assigned a material 4) A Union_master component placed after all of the above

Only in step 4 will any simulation happen, and per default all geometries defined before the master, but after the previous will be simulated here.

There is a dedicated manual available for the Union_components

Algorithm: Described elsewhere

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
process_string	string	Comma seperated names of physical processes	"NULL"
my_absorption	1/m	Inverse penetration depth from absorption at standard energy	
absorber	0/1	Control parameter, if set to 1 the material will have no scattering processes	0
material_string	string	List of elements present in the material. Triggers a search for materials constants files.	"NULL"
init	string	Deprecated and unused. Accepted so that instruments written for McStas/McXtrace 3.8.7 and earlier, which name the Union_init component here, still compile.	""

Links

- Component source code found in file `Union_make_material.comp`.

10.12. Union scattering/interaction process components

10.13. The `Incoherent_process` McXtrace Component

Component implementing a true incoherent scattering process

Identification

- **Author:** Mads Bertelsen and Erik B Knudsen
- **Origin:** ESS DMSC & DTU Physics
- **Date:** 20.08.15

Description

This Union_process is based on the Incoherent.comp component originally written by Kim Lefmann and Kristian Nielsen.

Part of the Union components, a set of components that work together and thus separates geometry and physics within McXtrace. The use of this component requires other components to be used.

1) One specifies a number of processes using process components like this one 2) These are gathered into material definitions using Union_make_material 3) Geometries are placed using Union_box / Union_cylinder, assigned a material 4) A Union_master component placed after all of the above

Only in step 4 will any simulation happen, and per default all geometries defined before the master, but after the previous will be simulated here.

There is a dedicated manual available for the Union_components

Algorithm: This component is an approximation. It scatters completely isotropically.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sigma	barns	Incoherent scattering cross section	5.08
f_QE	1	Fraction of quasielastic scattering (rest is elastic)	0
gamma	1	Lorentzian width of quasielastic broadening (HWHM)	0
packing_factor	1	How dense is the material compared to optimal 0-1	1
unit_cell_volume	$Å^3$	Unit_cell_volume	13.8

Name	Unit	Description	Default
interact_fraction	1	How large a part of the scattering events should use this process 0-1 (sum of all processes in material = 1)	-1
init	string	Deprecated and unused. Accepted so that instruments written for McStas/McXtrace 3.8.7 and earlier, which name the Union_init component here, still compile.	""

Links

- Component source code found in file `Incoherent_process.comp`.
- The test/example instrument Test_Phonon.instr.

10.14. The `Compton_xrl_process` McXtrace Component

Component that implements a Compton scattering process

Identification

- **Author:** Mads Bertelsen and Erik B Knudsen
- **Origin:** ESS DMSC & DTU Physics & United Neux
- **Date:** 20.08.15

Description

This Union_process is based on the Incoherent.comp component originally written by Kim Lefmann and Kristian Nielsen

Part of the Union components, a set of components that work together and thus separates geometry and physics within McXtrace. The use of this component requires other components to be used.

1) One specifies a number of processes using process components like this one 2) These are gathered into material definitions using Union_make_material 3) Geometries are placed using Union_box / Union_cylinder, assigned a material 4) A Union_master component placed after all of the above

Only in step 4 will any simulation happen, and per default all geometries defined before the master, but after the previous will be simulated here.

There is a dedicated manual available for the Union_components

Algorithm: Described elsewhere

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
density	g/cm^3	Nominal density of material.	0
atomno	1	Atomic number.	14
element	str	The element (symbol) of the material. Overrides atomno.	""
init	string	Deprecated and unused. Accepted so that instruments written for McStas/McXtrace 3.8.7 and earlier, which name the Union_init component here, still compile.	""

Links

- Component source code found in file `Compton_xrl_process.comp`.
- The test/example instrument Test_Phonon.instr.

10.15. The `Rayleigh_xrl_process` McXtrace Component

Component that implements a Rayleigh scattering process

Identification

- **Author:** Mads Bertelsen and Erik B Knudsen
- **Origin:** ESS DMSC & DTU Physics & United Neux
- **Date:** 20.08.15

Description

This Union_process is based on the Incoherent.comp component originally written by Kim Lefmann and Kristian Nielsen

Part of the Union components, a set of components that work together and thus separates geometry and physics within McXtrace. The use of this component requires other components to be used.

1) One specifies a number of processes using process components like this one 2) These are gathered into material definitions using Union_make_material 3) Geometries are placed using Union_box / Union_cylinder, assigned a material 4) A Union_master component placed after all of the above

Only in step 4 will any simulation happen, and per default all geometries defined before the master, but after the previous will be simulated here.

There is a dedicated manual available for the Union_components

Algorithm: Described elsewhere

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
density	g/cm^3	Nominal density of material.	0
atomno	1	Atomic number.	14
element	str	The element (symbol) of the material. Overrides atomno.	""
init	string	Deprecated and unused. Accepted so that instruments written for McStas/McXtrace 3.8.7 and earlier, which name the Union_init component here, still compile. OUTPUT PARAMETERS:	""

Links

- Component source code found in file `Rayleigh_xrl_process.comp`.
- The test/example instrument Test_Phonon.instr.

10.16. The `KN_xrl_process` McXtrace Component

A component implementing the Klein-Nishina cross section as a Union physics process

Identification

- **Author:** Mads Bertelsen and Erik B Knudsen
- **Origin:** ESS DMSC & DTU Physics & United Neux
- **Date:** 20.08.15

Description

This Union_process is based on the Incoherent.comp component originally written by Kim Lefmann and Kristian Nielsen

Part of the Union components, a set of components that work together and thus separates geometry and physics within McXtrace. The use of this component requires other components to be used.

1) One specifies a number of processes using process components like this one 2) These are gathered into material definitions using Union_make_material 3) Geometries are placed using Union_box / Union_cylinder, assigned a material 4) A Union_master component placed after all of the above

Only in step 4 will any simulation happen, and per default all geometries defined before the master, but after the previous will be simulated here.

There is a dedicated manual available for the Union_components

Algorithm: Described elsewhere

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
density	g/cm^3	Nominal density of material.	0
init	string	Deprecated and unused. Accepted so that instruments written for McStas/McXtrace 3.8.7 and earlier, which name the Union_init component here, still compile. OUTPUT PARAMETERS:	""

Links

- Component source code found in file `KN_xrl_process.comp`.
- The test/example instrument Test_Phonon.instr.

10.17. The `Powder_process` McXtrace Component

A sample component implementing a powder scattering process

Identification

- **Author:** Mads Bertelsen and Erik B Knudsen
- **Origin:** ESS DMSC & DTU Physics & United Neux
- **Date:** 20.08.15

Description

This Union_process is based on the PowerN.comp component.

Part of the Union components, a set of components that work together and thus separates geometry and physics within McXtrace. The use of this component requires other components to be used.

1) One specifies a number of processes using process components like this one 2) These are gathered into material definitions using Union_make_material 3) Geometries are placed using Union_box / Union_cylinder, assigned a material 4) A Union_master component placed after all of the above

Only in step 4 will any simulation happen, and per default all geometries defined before the master, but after the previous will be simulated here.

There is a dedicated manual available for the Union_components

Algorithm: Described elsewhere

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
reflections	string	Input file for reflections. No scattering if NULL or "" [string]	"NULL"
material	str	Reflection list for power	"NULL"
packing_factor	1	How dense is the material compared to optimal 0-1	1
Vc	Å^3	Volume of unit cell=nb atoms per cell/density of atoms.	0
delta_d_d	0/1	Global relative delta_d_d/d broadening when the 'w' column is not available. Use 0 if ideal.	0
DW	1	Global Debye-Waller factor when the 'DW' column is not available. Use 1 if included in F2	0
nb_atoms	1	Number of sub-unit per unit cell, that is ratio of sigma for chemical formula to sigma per unit cell	1
density	g/cm^3	Density of material. rho=density/weight/1e24*N_A.	0
weight	g/mol	Atomic/molecular weight of material.	0
barns	1	Flag to indicate if —F—^2 from 'reflections' is in barns or fm^2 (barns=1 for laz, barns=0 for lau type files).	1

Name	Unit	Description	Default
Strain	ppm	Global relative delta_d_d/d shift when the 'Strain' column is not available. Use 0 if ideal.	0
interact_fraction	1	How large a part of the scattering events should use this process 0-1 (sum of all processes in material = 1)	-1
format	no quotes	Name of the format, or list of column indexes (see Description).	{0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
mat_format	vector	List order in reflection list file	{0,0,0,0,0}

Links

- Component source code found in file `Powder_process.comp`.

10.18. The `Template_process` McXtrace Component

Template for a new contributor to create their own physical process.

Identification

- **Author:** Mads Bertelsen and Erik B Knudsen
- **Origin:** ESS DMSC & DTU Physics
- **Date:** 20.08.15

Description

This is a template for a new contributor to create their own physical process. The comments in this file are meant to teach the user about creating their own process file, rather than explaining this one. For comments on how this code works, look in the Incoherent_process.comp.

Part of the Union components, a set of components that work together and thus sperates geometry and physics within McXtrace. The use of this component requires other components to be used.

1) One specifies a number of processes using process components like this one 2) These are gathered into material definitions using Union_make_material 3) Geometries are placed using Union_box / Union_cylinder, assigned a material 4) A Union_master component placed after all of the above

Only in step 4 will any simulation happen, and per default all geometries defined before the master, but after the previous will be simulated here.

There is a dedicated manual available for the Union_components

Algorithm: Described elsewhere

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sigma	barns	Scattering cross section	5.08
packing_factor	1	Material packing factor	1
unit_cell_volume	Å^3	Unit cell volume	13.8
interact_fraction	1	How large a part of the scattering events should use this process 0-1 (sum of all processes in material = 1)	-1
init	string	Deprecated and unused. Accepted so that instruments written for McStas/McXtrace 3.8.7 and earlier, which name the Union_init component here, still compile. OUTPUT PARAMETERS: Template_storage // Important to update this output paramter effective_my_scattering // Variable used in initialize	""

Links

- Component source code found in file `Template_process.comp`.

11. SasView/sasmodels form factors: small-angle scattering

This chapter documents the `sasmodels` category: a large set of small-angle scattering (SAXS/USAXS) form-factor models, generated automatically from the open-source SasView/sasmodels project (`https://github.com/SasView/sasmodels`) and exposed to McXtrace as individual `SasView_*` components (each wrapping one sasmodels kernel). This gives access to the same, actively maintained library of scattering-law implementations used for SAXS/SANS data fitting in SasView itself, directly as McXtrace sample components, without needing to hand-implement each form factor.

Most models exist in a plain (orientation-averaged, i.e. powder/isotropic) form and, where the shape has a well-defined orientation, an `_aniso` variant that additionally accepts orientation parameters. For the underlying physics of any individual model (defining equation, parameter meaning, validity range), consult the corresponding page in the SasView documentation (`https://www.sasview.org/docs/`); the parameter tables below are generated directly from each component's McDoc header and give the McXtrace-specific parameter names, units and defaults.

See also the `contrib` chapter's native (non-SasView-derived) SAXS sample models (`SAXSSpheres`, `SAXSCylinders`, ...) for an alternative, McXtrace-specific implementation of many of the same shapes.

11.1. SasView form-factor components

11.2. The `SasView_adsorbed_layer` McXtrace Component

SasView adsorbed_layer model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_adsorbed_layer component, generated from adsorbed_layer.c in sasmodels.

Example: SasView_adsorbed_layer(second_moment, adsorbed_amount, density_shell, radius, volfraction, sld_shell, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
second_moment	Å	([0.0, inf]) Second moment of polymer distribution.	23.0
adsorbed_amount	mg/m^2	([0.0, inf]) Adsorbed amount of polymer.	1.9
density_shell	g/cm^3	([0.0, inf]) Bulk density of polymer in the shell.	0.7
radius	Å	([0.0, inf]) Core particle radius.	500.0
volfraction	None	([0.0, inf]) Core particle volume fraction.	0.14
sld_shell	1e-6/Å^2	([-inf, inf]) Polymer shell SLD.	1.5
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent SLD. Optional parameters:	6.3
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0

Name	Unit	Description	Default
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_adsorbed_layer.comp`.

11.3. The `SasView_barbell` McXtrace Component

SasView barbell model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_barbell component, generated from barbell.c in sasmodels.

Example: SasView_barbell(sld, sld_solvent, radius_bell, radius, length, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_bell=0.0, pd_radius=0.0, pd_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Barbell scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
radius_bell	Å	([0, inf]) Spherical bell radius.	40
radius	Å	([0, inf]) Cylindrical bar radius.	20

Name	Unit	Description	Default
length	Å	([0, inf]) Cylinder bar length. Optional parameters:	400
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_bell		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_barbell.comp`.

11.4. The `SasView_barbell_aniso` McXtrace Component

SasView barbell model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_barbell component, generated from barbell.c in sasmodels.

Example: SasView_barbell_aniso(sld, sld_solvent, radius_bell, radius, length, theta, Phi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_bell=0.0, pd_radius=0.0, pd_length=0.0, pd_theta=0.0, pd_Phi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Barbell scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
radius_bell	Å	([0, inf]) Spherical bell radius.	40
radius	Å	([0, inf]) Cylindrical bar radius.	20
length	Å	([0, inf]) Cylinder bar length. Optional parameters:	400
theta			60
Phi			60
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01

Name	Unit	Description	Default
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_bell		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_barbell_aniso.comp`.

11.5. The `SasView_bcc_paracrystal` McXtrace Component

SasView bcc_paracrystal model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_bcc_paracrystal component, generated from bcc_paracrystal.c in sasmodels.

Example: SasView_bcc_paracrystal(dnn, d_factor, radius, sld, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
dnn	Å	([-inf, inf]) Nearest neighbour distance.	220
d_factor		([-inf, inf]) Paracrystal distortion factor.	0.06
radius	Å	([0, inf]) Particle radius.	40
sld	1e-6/Å^2	([-inf, inf]) Particle scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	1
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1

Name	Unit	Description	Default
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_bcc_paracrystal.comp`.

11.6. The `SasView_bcc_paracrystal_aniso` McXtrace Component

SasView bcc_paracrystal model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_bcc_paracrystal component, generated from bcc_paracrystal.c in sasmodels.

Example: SasView_bcc_paracrystal_aniso(dnn, d_factor, radius, sld, sld_solvent, theta, Phi, Psi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_theta=0.0, pd_Phi=0.0, pd_Psi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
dnn	Å	([-inf, inf]) Nearest neighbour distance.	220
d_factor		([-inf, inf]) Paracrystal distortion factor.	0.06
radius	Å	([0, inf]) Particle radius.	40
sld	1e-6/Å^2	([-inf, inf]) Particle scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	1
theta			60
Phi			60
Psi			60
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0

Name	Unit	Description	Default
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Psi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_bcc_paracrystal_aniso.comp`.

11.7. The `SasView_binary_hard_sphere` McXtrace Component

SasView binary_hard_sphere model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_binary_hard_sphere component, generated from binary_hard_sphere.c in sasmodels.

Example: SasView_binary_hard_sphere(radius_lg, radius_sm, volfraction_lg, volfraction_sm, sld_lg, sld_sm, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_lg=0.0, pd_radius_sm=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_lg	Å	([0, inf]) radius of large particle.	100
radius_sm	Å	([0, inf]) radius of small particle.	25
volfraction_lg		([0, 1]) volume fraction of large particle.	0.1
volfraction_sm		([0, 1]) volume fraction of small particle.	0.2
sld_lg	1e-6/Å^2	([-inf, inf]) scattering length density of large particle.	3.5

Name	Unit	Description	Default
sld_sm	1e-6/Å^2	([-inf, inf]) scattering length density of small particle.	0.5
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	6.36
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_lg		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius_sm		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_binary_hard_sphere.comp`.

11.8. The `SasView_broad_peak` McXtrace Component

SasView broad_peak model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_broad_peak component, generated from broad_peak.c in sasmodels.
Example: SasView_broad_peak(porod_scale, porod_exp, peak_scale, correlation_length, peak_pos, width_exp, shape_exp, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_correlation_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
porod_scale		([-inf, inf]) Power law scale factor.	1e-05
porod_exp		([-inf, inf]) Exponent of power law.	3.0
peak_scale		([-inf, inf]) Scale factor for broad peak.	10.0
correlation_length	Å	([-inf, inf]) screening length.	50.0
peak_pos	1/Å	([-inf, inf]) Peak position in q.	0.1
width_exp		([-inf, inf]) Exponent of peak width.	2.0
shape_exp		([-inf, inf]) Exponent of peak shape. Optional parameters:	1.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005

Name	Unit	Description	Default
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_correlation_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_broad_peak.comp`.

11.9. The `SasView_capped_cylinder` McXtrace Component

SasView capped_cylinder model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_capped_cylinder component, generated from capped_cylinder.c in sasmodels.

Example: SasView_capped_cylinder(sld, sld_solvent, radius, radius_cap, length, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_radius_cap=0.0, pd_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Cylinder scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
radius	Å	([0, inf]) Cylinder radius.	20
radius_cap	Å	([0, inf]) Cap radius.	20
length	Å	([0, inf]) Cylinder length. Optional parameters:	400
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0

Name	Unit	Description	Default
pd_radius_cap		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_capped_cylinder.comp`.

11.10. The `SasView_capped_cylinder_aniso` McXtrace Component

SasView capped_cylinder model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_capped_cylinder component, generated from capped_cylinder.c in sasmodels.

Example: SasView_capped_cylinder_aniso(sld, sld_solvent, radius, radius_cap, length, theta, Phi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_radius_cap=0.0, pd_length=0.0, pd_theta=0.0, pd_Phi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Cylinder scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1

Name	Unit	Description	Default
radius	Å	([0, inf]) Cylinder radius.	20
radius_cap	Å	([0, inf]) Cap radius.	20
length	Å	([0, inf]) Cylinder length. Optional parameters:	400
theta			60
Phi			60
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius_cap		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0

Name	Unit	Description	Default
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_capped_cylinder_aniso.comp`.

11.11. The `SasView_core_multi_shell` McXtrace Component

SasView core_multi_shell model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_core_multi_shell component, generated from core_multi_shell.c in sasmodels.

Example: SasView_core_multi_shell(sld_core, radius, sld_solvent, n, sld[n], thickness[n], model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thickness[n]=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default

Links

- Component source code found in file `SasView_core_multi_shell.comp`.

11.12. The `SasView_core_shell_bicelle` McXtrace Component

SasView core_shell_bicelle model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_core_shell_bicelle component, generated from core_shell_bicelle.c in sasmodels.
Example: SasView_core_shell_bicelle(radius, thick_rim, thick_face, length, sld_core, sld_face, sld_rim, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thick_rim=0.0, pd_thick_face=0.0, pd_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0, inf]) Cylinder core radius.	80
thick_rim	Å	([0, inf]) Rim shell thickness.	10
thick_face	Å	([0, inf]) Cylinder face thickness.	10
length	Å	([0, inf]) Cylinder length.	50
sld_core	1e-6/Å^2	([-inf, inf]) Cylinder core scattering length density.	1
sld_face	1e-6/Å^2	([-inf, inf]) Cylinder face scattering length density.	4
sld_rim	1e-6/Å^2	([-inf, inf]) Cylinder rim scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	1
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0

Name	Unit	Description	Default
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_rim		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_face		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_core_shell_bicelle.comp`.

11.13. The `SasView_core_shell_bicelle_aniso` McXtrace Component

SasView core_shell_bicelle model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_core_shell_bicelle component, generated from core_shell_bicelle.c in sasmodels.

Example: SasView_core_shell_bicelle_aniso(radius, thick_rim, thick_face, length, sld_core, sld_face, sld_rim, sld_solvent, theta, Phi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thick_rim=0.0, pd_thick_face=0.0, pd_length=0.0, pd_theta=0.0, pd_Phi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0, inf]) Cylinder core radius.	80
thick_rim	Å	([0, inf]) Rim shell thickness.	10
thick_face	Å	([0, inf]) Cylinder face thickness.	10
length	Å	([0, inf]) Cylinder length.	50
sld_core	1e-6/Å^2	([-inf, inf]) Cylinder core scattering length density.	1
sld_face	1e-6/Å^2	([-inf, inf]) Cylinder face scattering length density.	4
sld_rim	1e-6/Å^2	([-inf, inf]) Cylinder rim scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	1
theta			90
Phi			0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01

Name	Unit	Description	Default
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_rim		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_face		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_core_shell_bicelle_aniso.comp`.

11.14. The `SasView_core_shell_bicelle_elliptical` McXtrace Component

SasView core_shell_bicelle_elliptical model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_core_shell_bicelle_elliptical component, generated from core_shell_bicelle_elliptical.c in sasmodels.

Example: SasView_core_shell_bicelle_elliptical(radius, x_core, thick_rim, thick_face, length, sld_core, sld_face, sld_rim, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thick_rim=0.0, pd_thick_face=0.0, pd_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0, inf]) Cylinder core radius r_minor.	30
x_core	None	([0, inf]) Axial ratio of core, X = r_major/r_minor.	3
thick_rim	Å	([0, inf]) Rim shell thickness.	8
thick_face	Å	([0, inf]) Cylinder face thickness.	14
length	Å	([0, inf]) Cylinder length.	50
sld_core	1e-6/Å^2	([-inf, inf]) Cylinder core scattering length density.	4
sld_face	1e-6/Å^2	([-inf, inf]) Cylinder face scattering length density.	7
sld_rim	1e-6/Å^2	([-inf, inf]) Cylinder rim scattering length density.	1
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	6

Name	Unit	Description	Default
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_rim		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_face		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_core_shell_bicelle_elliptical.comp`.

11.15. The `SasView_core_shell_bicelle_elliptical_aniso` McXtrace Component

SasView core_shell_bicelle_elliptical model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_core_shell_bicelle_elliptical component, generated from core_shell_bicelle_elliptical.c in sasmodels.

Example: SasView_core_shell_bicelle_elliptical_aniso(radius, x_core, thick_rim, thick_face, length, sld_core, sld_face, sld_rim, sld_solvent, theta, Phi, Psi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thick_rim=0.0, pd_thick_face=0.0, pd_length=0.0, pd_theta=0.0, pd_Phi=0.0, pd_Psi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0, inf]) Cylinder core radius r_minor.	30
x_core	None	([0, inf]) Axial ratio of core, X = r_major/r_minor.	3
thick_rim	Å	([0, inf]) Rim shell thickness.	8
thick_face	Å	([0, inf]) Cylinder face thickness.	14
length	Å	([0, inf]) Cylinder length.	50
sld_core	1e-6/Å^2	([-inf, inf]) Cylinder core scattering length density.	4
sld_face	1e-6/Å^2	([-inf, inf]) Cylinder face scattering length density.	7
sld_rim	1e-6/Å^2	([-inf, inf]) Cylinder rim scattering length density.	1
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	6
theta			90.0
Phi			0

Name	Unit	Description	Default
Psi			0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_rim		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_face		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0

Name	Unit	Description	Default
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Psi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_core_shell_bicelle_elliptical_aniso.comp`.

11.16. The `SasView_core_shell_bicelle_elliptical_belt_rough` McXtrace Component

SasView core_shell_bicelle_elliptical_belt_rough model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_core_shell_bicelle_elliptical_belt_rough component, generated from core_shell_bicelle_elliptical in sasmodels.

Example: SasView_core_shell_bicelle_elliptical_belt_rough(radius, x_core, thick_rim, thick_face, length, sld_core, sld_face, sld_rim, sld_solvent, sigma, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thick_rim=0.0, pd_thick_face=0.0, pd_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0, inf]) Cylinder core radius r_minor.	30

Name	Unit	Description	Default
x_core	None	([0, inf]) Axial ratio of core, X = r_major/r_minor.	3
thick_rim	Å	([0, inf]) Rim or belt shell thickness.	8
thick_face	Å	([0, inf]) Cylinder face thickness.	14
length	Å	([0, inf]) Cylinder length.	50
sld_core	1e-6/Å^2	([-inf, inf]) Cylinder core scattering length density.	4
sld_face	1e-6/Å^2	([-inf, inf]) Cylinder face scattering length density.	7
sld_rim	1e-6/Å^2	([-inf, inf]) Cylinder rim scattering length density.	1
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	6
sigma	Å	([0, inf]) Interfacial roughness. Optional parameters:	0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0

Name	Unit	Description	Default
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_rim		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_face		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_core_shell_bicelle_elliptical_belt_rough.c`

11.17. The `SasView_core_shell_bicelle_elliptical_belt_rough_aniso` McXtrace Component

SasView core_shell_bicelle_elliptical_belt_rough model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_core_shell_bicelle_elliptical_belt_rough component, generated from core_shell_bicelle_elliptical in sasmodels.

Example: SasView_core_shell_bicelle_elliptical_belt_rough_aniso(radius, x_core, thick_rim, thick_face, length, sld_core, sld_face, sld_rim, sld_solvent, sigma, theta, Phi, Psi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thick_rim=0.0, pd_thick_face=0.0, pd_length=0.0, pd_theta=0.0, pd_Phi=0.0, pd_Psi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0, inf]) Cylinder core radius r_minor.	30
x_core	None	([0, inf]) Axial ratio of core, X = r_major/r_minor.	3
thick_rim	Å	([0, inf]) Rim or belt shell thickness.	8
thick_face	Å	([0, inf]) Cylinder face thickness.	14
length	Å	([0, inf]) Cylinder length.	50
sld_core	1e-6/Å^2	([-inf, inf]) Cylinder core scattering length density.	4
sld_face	1e-6/Å^2	([-inf, inf]) Cylinder face scattering length density.	7
sld_rim	1e-6/Å^2	([-inf, inf]) Cylinder rim scattering length density.	1
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	6
sigma	Å	([0, inf]) Interfacial roughness. Optional parameters:	0
theta			90.0
Phi			0
Psi			0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5

Name	Unit	Description	Default
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_rim		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_face		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Psi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_core_shell_bicelle_elliptical_belt_rough_ar`

11.18. The `SasView_core_shell_cylinder` McXtrace Component

SasView core_shell_cylinder model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC

- **Date:**

Description

SasView_core_shell_cylinder component, generated from core_shell_cylinder.c in sasmodels.

Example: SasView_core_shell_cylinder(sld_core, sld_shell, sld_solvent, radius, thickness, length, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thickness=0.0, pd_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld_core	1e-6/Å^2	([-inf, inf]) Cylinder core scattering length density.	4
sld_shell	1e-6/Å^2	([-inf, inf]) Cylinder shell scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
radius	Å	([0, inf]) Cylinder core radius.	20
thickness	Å	([0, inf]) Cylinder shell thickness.	20
length	Å	([0, inf]) Cylinder length. Optional parameters:	400
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1

Name	Unit	Description	Default
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thickness		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_core_shell_cylinder.comp`.

11.19. The `SasView_core_shell_cylinder_aniso` McXtrace Component

SasView core_shell_cylinder model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_core_shell_cylinder component, generated from core_shell_cylinder.c in sasmodels.

Example: SasView_core_shell_cylinder_aniso(sld_core, sld_shell, sld_solvent, radius, thickness, length, theta, Phi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5,

focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thickness=0.0, pd_length=0.0, pd_theta=0.0, pd_Phi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld_core	1e-6/Å^2	([-inf, inf]) Cylinder core scattering length density.	4
sld_shell	1e-6/Å^2	([-inf, inf]) Cylinder shell scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
radius	Å	([0, inf]) Cylinder core radius.	20
thickness	Å	([0, inf]) Cylinder shell thickness.	20
length	Å	([0, inf]) Cylinder length. Optional parameters:	400
theta			60
Phi			60
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0

Name	Unit	Description	Default
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thickness		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_core_shell_cylinder_aniso.comp`.

11.20. The `SasView_core_shell_ellipsoid` McXtrace Component

SasView core_shell_ellipsoid model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_core_shell_ellipsoid component, generated from core_shell_ellipsoid.c in sasmodels.

Example: SasView_core_shell_ellipsoid(radius_equat_core, x_core, thick_shell, x_polar_shell, sld_core, sld_shell, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01,

zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_equat_core=0.0, pd_thick_shell=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_equat_core	Å	([0, inf]) Equatorial radius of core.	20
x_core	None	([0, inf]) axial ratio of core, X = r_polar/r_equatorial.	3
thick_shell	Å	([0, inf]) thickness of shell at equator.	30
x_polar_shell		([0, inf]) ratio of thickness of shell at pole to that at equator.	1
sld_core	1e-6/Å^2	([-inf, inf]) Core scattering length density.	2
sld_shell	1e-6/Å^2	([-inf, inf]) Shell scattering length density.	1
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	6.3
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0

Name	Unit	Description	Default
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_equat_core		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_shell		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_core_shell_ellipsoid.comp`.

11.21. The `SasView_core_shell_ellipsoid_aniso` McXtrace Component

SasView core_shell_ellipsoid model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_core_shell_ellipsoid component, generated from core_shell_ellipsoid.c in sasmodels.

Example: SasView_core_shell_ellipsoid_aniso(radius_equat_core, x_core, thick_shell, x_polar_shell, sld_core, sld_shell, sld_solvent, theta, Phi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_equat_core=0.0 pd_thick_shell=0.0, pd_theta=0.0, pd_Phi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_equat_core	Å	([0, inf]) Equatorial radius of core.	20
x_core	None	([0, inf]) axial ratio of core, X = r_polar/r_equatorial.	3
thick_shell	Å	([0, inf]) thickness of shell at equator.	30
x_polar_shell		([0, inf]) ratio of thickness of shell at pole to that at equator.	1
sld_core	1e-6/Å^2	([-inf, inf]) Core scattering length density.	2
sld_shell	1e-6/Å^2	([-inf, inf]) Shell scattering length density.	1
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	6.3
theta			0
Phi			0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0

Name	Unit	Description	Default
pd_radius_equat_core		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_shell		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_core_shell_ellipsoid_aniso.comp`.

11.22. The `SasView_core_shell_parallelepiped` McXtrace Component

SasView core_shell_parallelepiped model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_core_shell_parallelepiped component, generated from core_shell_parallelepiped.c in sasmodels.

Example: SasView_core_shell_parallelepiped(sld_core, sld_a, sld_b, sld_c, sld_solvent, length_a, length_b, length_c, thick_rim_a, thick_rim_b, thick_rim_c, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length_a=0.0, pd_length_b=0.0, pd_length_c=0.0, pd_thick_rim_a=0.0, pd_thick_rim_b=0.0, pd_thick_rim_c=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld_core	1e-6/Å^2	([-inf, inf]) Parallelepiped core scattering length density.	1
sld_a	1e-6/Å^2	([-inf, inf]) Parallelepiped A rim scattering length density.	2
sld_b	1e-6/Å^2	([-inf, inf]) Parallelepiped B rim scattering length density.	4
sld_c	1e-6/Å^2	([-inf, inf]) Parallelepiped C rim scattering length density.	2
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	6
length_a	Å	([0, inf]) Shorter side of the parallelepiped.	35
length_b	Å	([0, inf]) Second side of the parallelepiped.	75
length_c	Å	([0, inf]) Larger side of the parallelepiped.	400
thick_rim_a	Å	([0, inf]) Thickness of A rim.	10
thick_rim_b	Å	([0, inf]) Thickness of B rim.	10
thick_rim_c	Å	([0, inf]) Thickness of C rim. Optional parameters:	10
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1

Name	Unit	Description	Default
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_length_a		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length_b		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length_c		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_rim_a		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_rim_b		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_rim_c		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_core_shell_parallelepiped.comp`.

11.23. The `SasView_core_shell_parallelepiped_aniso` McXtrace Component

SasView core_shell_parallelepiped model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_core_shell_parallelepiped component, generated from core_shell_parallelepiped.c in sasmodels.

Example: SasView_core_shell_parallelepiped_aniso(sld_core, sld_a, sld_b, sld_c, sld_solvent, length_a, length_b, length_c, thick_rim_a, thick_rim_b, thick_rim_c, theta, Phi, Psi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length_a=0.0, pd_length_b=0.0, pd_length_c=0.0, pd_thick_rim_a=0.0, pd_thick_rim_b=0.0, pd_thick_rim_c=0.0, pd_theta=0.0, pd_Phi=0.0, pd_Psi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld_core	1e-6/Å^2	([-inf, inf]) Parallelepiped core scattering length density.	1
sld_a	1e-6/Å^2	([-inf, inf]) Parallelepiped A rim scattering length density.	2
sld_b	1e-6/Å^2	([-inf, inf]) Parallelepiped B rim scattering length density.	4
sld_c	1e-6/Å^2	([-inf, inf]) Parallelepiped C rim scattering length density.	2
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	6
length_a	Å	([0, inf]) Shorter side of the parallelepiped.	35
length_b	Å	([0, inf]) Second side of the parallelepiped.	75
length_c	Å	([0, inf]) Larger side of the parallelepiped.	400
thick_rim_a	Å	([0, inf]) Thickness of A rim.	10
thick_rim_b	Å	([0, inf]) Thickness of B rim.	10
thick_rim_c	Å	([0, inf]) Thickness of C rim. Optional parameters:	10
theta			0
Phi			0
Psi			0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0

Name	Unit	Description	Default
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_length_a		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length_b		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length_c		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_rim_a		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_rim_b		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_rim_c		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0

Name	Unit	Description	Default
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Psi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_core_shell_parallelepiped_aniso.comp`.

11.24. The `SasView_core_shell_sphere` McXtrace Component

SasView core_shell_sphere model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_core_shell_sphere component, generated from core_shell_sphere.c in sasmodels.

Example: SasView_core_shell_sphere(radius, thickness, sld_core, sld_shell, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thickness=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0, inf]) Sphere core radius.	60.0
thickness	Å	([0, inf]) Sphere shell thickness.	10.0
sld_core	1e-6/Å^2	([-inf, inf]) core scattering length density.	1.0
sld_shell	1e-6/Å^2	([-inf, inf]) shell scattering length density.	2.0
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	3.0

Name	Unit	Description	Default
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thickness		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_core_shell_sphere.comp`.

11.25. The `SasView_correlation_length` McXtrace Component

SasView correlation_length model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_correlation_length component, generated from correlation_length.c in sasmodels.

Example: SasView_correlation_length(lorentz_scale, porod_scale, cor_length, porod_exp, lorentz_exp, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_cor_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
lorentz_scale		([0, inf]) Lorentzian Scaling Factor.	10.0
porod_scale		([0, inf]) Porod Scaling Factor.	1e-06
cor_length	Å	([0, inf]) Correlation length, xi, in Lorentzian.	50.0
porod_exp		([0, inf]) Porod Exponent, n, in qˆ-n.	3.0
lorentz_exp		([0, inf]) Lorentzian Exponent, m, in 1/(1 + (q.xi)ˆm). Optional parameters:	2.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0

Name	Unit	Description	Default
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_cor_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_correlation_length.comp`.

11.26. The `SasView_cylinder` McXtrace Component

SasView cylinder model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_cylinder component, generated from cylinder.c in sasmodels.

Example: SasView_cylinder(sld, sld_solvent, radius, length, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Cylinder scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
radius	Å	([0, inf]) Cylinder radius.	20
length	Å	([0, inf]) Cylinder length. Optional parameters:	400
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_cylinder.comp`.

11.27. The `SasView_cylinder_aniso` McXtrace Component

SasView cylinder model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_cylinder component, generated from cylinder.c in sasmodels.

Example: SasView_cylinder_aniso(sld, sld_solvent, radius, length, theta, Phi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_length=0.0, pd_theta=0.0, pd_Phi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Cylinder scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
radius	Å	([0, inf]) Cylinder radius.	20
length	Å	([0, inf]) Cylinder length. Optional parameters:	400
theta			60
Phi			60
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0

Name	Unit	Description	Default
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_cylinder_aniso.comp`.

11.28. The `SasView_dab` McXtrace Component

SasView dab model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_dab component, generated from dab.c in sasmodels.

Example: SasView_dab(cor_length, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_cor_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
cor_length	Å	([0, inf]) correlation length. Optional parameters:	50.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0

Name	Unit	Description	Default
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_cor_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_dab.comp`.

11.29. The `SasView_ellipsoid` McXtrace Component

SasView ellipsoid model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_ellipsoid component, generated from ellipsoid.c in sasmodels.

Example: SasView_ellipsoid(sld, sld_solvent, radius_polar, radius_equatorial, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_polar=0.0, pd_radius_equatorial=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Ellipsoid scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
radius_polar	Å	([0, inf]) Polar radius.	20
radius_equatorial	Å	([0, inf]) Equatorial radius. Optional parameters:	400

Name	Unit	Description	Default
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_polar		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius_equatorial		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_ellipsoid.comp`.

11.30. The `SasView_ellipsoid_aniso` McXtrace Component

SasView ellipsoid model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_ellipsoid component, generated from ellipsoid.c in sasmodels.

Example: SasView_ellipsoid_aniso(sld, sld_solvent, radius_polar, radius_equatorial, theta, Phi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_polar=0.0, pd_radius_equatorial=0.0, pd_theta=0.0, pd_Phi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Ellipsoid scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
radius_polar	Å	([0, inf]) Polar radius.	20
radius_equatorial	Å	([0, inf]) Equatorial radius. Optional parameters:	400
theta			60
Phi			60
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0

Name	Unit	Description	Default
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_polar		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius_equatorial		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_ellipsoid_aniso.comp`.

11.31. The `SasView_elliptical_cylinder` McXtrace Component

SasView elliptical_cylinder model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_elliptical_cylinder component, generated from elliptical_cylinder.c in sasmodels.

Example: SasView_elliptical_cylinder(radius_minor, r_ratio, length, sld, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_minor=0.0, pd_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_minor	Å	([0, inf]) Ellipse minor radius.	20.0
r_ratio		([1, inf]) Ratio of major radius over minor radius.	1.5
length	Å	([1, inf]) Length of the cylinder.	400.0
sld	1e-6/Å^2	([-inf, inf]) Cylinder scattering length density.	4.0
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	1.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0

Name	Unit	Description	Default
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_minor		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_elliptical_cylinder.comp`.

11.32. The `SasView_elliptical_cylinder_aniso` McXtrace Component

SasView elliptical_cylinder model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_elliptical_cylinder component, generated from elliptical_cylinder.c in sasmodels.

Example: SasView_elliptical_cylinder_aniso(radius_minor, r_ratio, length, sld, sld_solvent, theta, Phi, Psi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_minor=0.0, pd_length=0.0, pd_theta=0.0, pd_Phi=0.0, pd_Psi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_minor	Å	([0, inf]) Ellipse minor radius.	20.0

Name	Unit	Description	Default
r_ratio		([1, inf]) Ratio of major radius over minor radius.	1.5
length	Å	([1, inf]) Length of the cylinder.	400.0
sld	1e-6/Å^2	([-inf, inf]) Cylinder scattering length density.	4.0
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	1.0
theta			90.0
Phi			0
Psi			0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_minor		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0

Name	Unit	Description	Default
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Psi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_elliptical_cylinder_aniso.comp`.

11.33. The `SasView_fcc_paracrystal` McXtrace Component

SasView fcc_paracrystal model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_fcc_paracrystal component, generated from fcc_paracrystal.c in sasmodels.

Example: SasView_fcc_paracrystal(dnn, d_factor, radius, sld, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
dnn	Å	([-inf, inf]) Nearest neighbour distance.	220
d_factor		([-inf, inf]) Paracrystal distortion factor.	0.06
radius	Å	([0, inf]) Particle radius.	40

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Particle scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	1
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_fcc_paracrystal.comp`.

11.34. The SasView_fcc_paracrystal_aniso McXtrace Component

SasView fcc_paracrystal model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_fcc_paracrystal component, generated from fcc_paracrystal.c in sasmodels.

Example: SasView_fcc_paracrystal_aniso(dnn, d_factor, radius, sld, sld_solvent, theta, Phi, Psi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_theta=0.0, pd_Phi=0.0, pd_Psi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
dnn	Å	([-inf, inf]) Nearest neighbour distance.	220
d_factor		([-inf, inf]) Paracrystal distortion factor.	0.06
radius	Å	([0, inf]) Particle radius.	40
sld	1e-6/Å^2	([-inf, inf]) Particle scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	1
theta			60
Phi			60
Psi			60
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01

Name	Unit	Description	Default
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Psi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_fcc_paracrystal_aniso.comp`.

11.35. The `SasView_flexible_cylinder` McXtrace Component

SasView flexible_cylinder model component as sample description.

Identification

- **Author:** Jose Robledo

- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_flexible_cylinder component, generated from flexible_cylinder.c in sasmodels.

Example: SasView_flexible_cylinder(length, kuhn_length, radius, sld, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length=0.0, pd_kuhn_length=0.0, pd_radius=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
length	Å	([0, inf]) Length of the flexible cylinder.	1000.0
kuhn_length	Å	([0, inf]) Kuhn length of the flexible cylinder.	100.0
radius	Å	([0, inf]) Radius of the flexible cylinder.	20.0
sld	1e-6/Å^2	([-inf, inf]) Cylinder scattering length density.	1.0
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	6.3
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1

Name	Unit	Description	Default
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_kuhn_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_flexible_cylinder.comp`.

11.36. The `SasView_flexible_cylinder_elliptical` McXtrace Component

SasView flexible_cylinder_elliptical model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_flexible_cylinder_elliptical component, generated from flexible_cylinder_elliptical.c in sasmodels.

Example: SasView_flexible_cylinder_elliptical(length, kuhn_length, radius, axis_ratio, sld, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length=0.0, pd_kuhn_length=0.0, pd_radius=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
length	Å	([0, inf]) Length of the flexible cylinder.	1000.0
kuhn_length	Å	([0, inf]) Kuhn length of the flexible cylinder.	100.0
radius	Å	([1, inf]) Radius of the flexible cylinder.	20.0
axis_ratio		([0, inf]) Axis_ratio (major_radius/minor_radius.	1.5
sld	1e-6/Å^2	([-inf, inf]) Cylinder scattering length density.	1.0
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	6.3
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0

Name	Unit	Description	Default
pd_kuhn_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_flexible_cylinder_elliptical.comp`.

11.37. The `SasView_fractal` McXtrace Component

SasView fractal model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_fractal component, generated from fractal.c in sasmodels.

Example: SasView_fractal(volfraction, radius, fractal_dim, cor_length, sld_block, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_cor_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
volfraction		([0.0, 1]) volume fraction of blocks.	0.05
radius	Å	([0.0, inf]) radius of particles.	5.0
fractal_dim		([0.0, 6.0]) fractal dimension.	2.0
cor_length	Å	([0.0, inf]) cluster correlation length.	100.0
sld_block	1e-6/Å^2	([-inf, inf]) scattering length density of particles.	2.0

Name	Unit	Description	Default
sld_solvent	1e-6/Å^2	([-inf, inf]) scattering length density of solvent. Optional parameters:	6.4
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_cor_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_fractal.comp`.

11.38. The `SasView_fractal_core_shell` McXtrace Component

SasView fractal_core_shell model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_fractal_core_shell component, generated from fractal_core_shell.c in sasmodels.

Example: SasView_fractal_core_shell(radius, thickness, sld_core, sld_shell, sld_solvent, volfraction, fractal_dim, cor_length, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thickness=0.0, pd_cor_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0.0, inf]) Sphere core radius.	60.0
thickness	Å	([0.0, inf]) Sphere shell thickness.	10.0
sld_core	1e-6/Å^2	([-inf, inf]) Sphere core scattering length density.	1.0
sld_shell	1e-6/Å^2	([-inf, inf]) Sphere shell scattering length density.	2.0
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	3.0
volfraction		([0.0, inf]) Volume fraction of building block spheres.	0.05
fractal_dim		([0.0, 6.0]) Fractal dimension.	2.0
cor_length	Å	([0.0, inf]) Correlation length of fractal-like aggregates. Optional parameters:	100.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01

Name	Unit	Description	Default
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thickness		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_cor_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_fractal_core_shell.comp`.

11.39. The `SasView_fuzzy_sphere` McXtrace Component

SasView fuzzy_sphere model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_fuzzy_sphere component, generated from fuzzy_sphere.c in sasmodels.

Example: SasView_fuzzy_sphere(sld, sld_solvent, radius, fuzziness, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Particle scattering length density.	1
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	3
radius	Å	([0, inf]) Sphere radius.	60
fuzziness	Å	([0, inf]) std deviation of Gaussian convolution for interface (must be << radius). Optional parameters:	10
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0

Name	Unit	Description	Default
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_fuzzy_sphere.comp`.

11.40. The `SasView_gauss_lorentz_gel` McXtrace Component

SasView gauss_lorentz_gel model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_gauss_lorentz_gel component, generated from gauss_lorentz_gel.c in sasmodels.

Example: SasView_gauss_lorentz_gel(gauss_scale, cor_length_static, lorentz_scale, cor_length_dynam
model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int
target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_cor_length_static=0.0, pd_cor_length_dynamic=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
gauss_scale		([-inf, inf]) Gauss scale factor.	100.0
cor_length_static	Å	([0, inf]) Static correlation length.	100.0
lorentz_scale		([-inf, inf]) Lorentzian scale factor.	50.0
cor_length_dynamic	Å	([0, inf]) Dynamic correlation length.	20.0
		Optional parameters:	

Name	Unit	Description	Default
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_cor_length_static		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_cor_length_dynamic		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_gauss_lorentz_gel.comp`.

11.41. The `SasView_gaussian_peak` McXtrace Component

SasView gaussian_peak model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_gaussian_peak component, generated from gaussian_peak.c in sasmodels.

Example: SasView_gaussian_peak(peak_pos, sigma, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0,)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
peak_pos	1/Å	([-inf, inf]) Peak position.	0.05
sigma	1/Å	([0, inf]) Peak width (standard deviation). Optional parameters:	0.005
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5

Name	Unit	Description	Default
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0

Links

- Component source code found in file `SasView_gaussian_peak.comp`.

11.42. The `SasView_gel_fit` McXtrace Component

SasView gel_fit model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_gel_fit component, generated from gel_fit.c in sasmodels.

Example: SasView_gel_fit(guinier_scale, lorentz_scale, rg, fractal_dim, cor_length, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_rg=0.0, pd_cor_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
guinier_scale	cm^{-1}	([-inf, inf]) Guinier term scale.	1.7
lorentz_scale	cm^{-1}	([-inf, inf]) Lorentz term scale.	3.5
rg	Å	([2, inf]) Radius of gyration.	104.0
fractal_dim		([0, inf]) Fractal exponent.	2.0
cor_length	Å	([0, inf]) Correlation length. Optional parameters:	16.0

Name	Unit	Description	Default
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_rg		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_cor_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_gel_fit.comp`.

11.43. The `SasView_guinier` McXtrace Component

SasView guinier model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_guinier component, generated from guinier.c in sasmodels.

Example: SasView_guinier(rg, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_rg=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
rg	Å	([-inf, inf]) Radius of Gyration. Optional parameters:	60.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0

Name	Unit	Description	Default
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_rg		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_guinier.comp`.

11.44. The `SasView_guinier_porod` McXtrace Component

SasView guinier_porod model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_guinier_porod component, generated from guinier_porod.c in sasmodels.

Example: SasView_guinier_porod(rg, s, porod_exp, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_rg=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
rg	Å	([0, inf]) Radius of gyration.	60.0
s		([0, inf]) Dimension variable.	1.0
porod_exp		([0, inf]) Porod exponent. Optional parameters:	3.0

Name	Unit	Description	Default
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_rg		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_guinier_porod.comp`.

11.45. The `SasView_hardsphere` McXtrace Component

SasView hardsphere model component as sample description.

Identification

- **Author:** Jose Robledo

- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_hardsphere component, generated from hardsphere.c in sasmodels.

Example: SasView_hardsphere(radius_effective, volfraction, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_effective=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_effective	Å	([0, inf]) effective radius of hard sphere.	50.0
volfraction		([0, 0.74]) volume fraction of hard spheres. Optional parameters:	0.2
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0

Name	Unit	Description	Default
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_effective		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_hardsphere.comp`.

11.46. The `SasView_hayter_msa` McXtrace Component

SasView hayter_msa model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_hayter_msa component, generated from hayter_msa.c in sasmodels.

Example: SasView_hayter_msa(radius_effective, volfraction, charge, temperature, concentration_salt, dielectconst, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_effective=0.0, pd_charge=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_effective	Å	([0, inf]) effective radius of charged sphere.	20.75
volfraction	None	([0, 0.74]) volume fraction of spheres.	0.0192
charge	e	([1e-06, 200]) charge on sphere (in electrons).	19.0
temperature	K	([0, 450]) temperature, in Kelvin, for Debye length calculation.	318.16

Name	Unit	Description	Default
concentration_salt	M	([0, inf]) conc of salt, moles/litre, 1:1 electolyte, for Debye length.	0.0
dielectconst	None	([-inf, inf]) dielectric constant (relative permittivity) of solvent, kappa, default water, for Debye length. Optional parameters:	71.08
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_effective		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_charge		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_hayter_msa.comp`.

11.47. The `SasView_hollow_cylinder` McXtrace Component

SasView hollow_cylinder model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_hollow_cylinder component, generated from hollow_cylinder.c in sasmodels.

Example: SasView_hollow_cylinder(radius, thickness, length, sld, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thickness=0.0, pd_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0, inf]) Cylinder core radius.	20.0
thickness	Å	([0, inf]) Cylinder wall thickness.	10.0
length	Å	([0, inf]) Cylinder total length.	400.0
sld	1e-6/Å^2	([-inf, inf]) Cylinder sld.	6.3
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent sld. Optional parameters:	1
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0

Name	Unit	Description	Default
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thickness		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_hollow_cylinder.comp`.

11.48. The `SasView_hollow_cylinder_aniso` McXtrace Component

SasView hollow_cylinder model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_hollow_cylinder component, generated from hollow_cylinder.c in sasmodels.

Example: SasView_hollow_cylinder_aniso(radius, thickness, length, sld, sld_solvent, theta, Phi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thickness=0.0, pd_length=0.0, pd_theta=0.0, pd_Phi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0, inf]) Cylinder core radius.	20.0
thickness	Å	([0, inf]) Cylinder wall thickness.	10.0
length	Å	([0, inf]) Cylinder total length.	400.0
sld	1e-6/Å^2	([-inf, inf]) Cylinder sld.	6.3
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent sld. Optional parameters:	1
theta			90
Phi			0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0

Name	Unit	Description	Default
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thickness		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_hollow_cylinder_aniso.comp`.

11.49. The `SasView_hollow_rectangular_prism` McXtrace Component

SasView hollow_rectangular_prism model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_hollow_rectangular_prism component, generated from hollow_rectangular_prism.c in sasmodels.

Example: SasView_hollow_rectangular_prism(sld, sld_solvent, length_a, b2a_ratio, c2a_ratio, thickness, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005,

R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length_a=0.0, pd_thickness=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Parallelepiped scattering length density.	6.3
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
length_a	Å	([0, inf]) Shortest, external, size of the parallelepiped.	35
b2a_ratio	Å	([0, inf]) Ratio sides b/a.	1
c2a_ratio	Å	([0, inf]) Ratio sides c/a.	1
thickness	Å	([0, inf]) Thickness of parallelepiped. Optional parameters:	1
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0

Name	Unit	Description	Default
focus_r	m	case of circular focusing, focusing radius.	0
pd_length_a		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thickness		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_hollow_rectangular_prism.comp`.

11.50. The `SasView_hollow_rectangular_prism_aniso` McXtrace Component

SasView hollow_rectangular_prism model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_hollow_rectangular_prism component, generated from hollow_rectangular_prism.c in sasmodels.

Example: SasView_hollow_rectangular_prism_aniso(sld, sld_solvent, length_a, b2a_ratio, c2a_ratio, thickness, theta, Phi, Psi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length_a=0.0, pd_thickness=0.0, pd_theta=0.0, pd_Phi=0.0, pd_Psi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Parallelepiped scattering length density.	6.3

Name	Unit	Description	Default
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
length_a	Å	([0, inf]) Shortest, external, size of the parallelepiped.	35
b2a_ratio	Å	([0, inf]) Ratio sides b/a.	1
c2a_ratio	Å	([0, inf]) Ratio sides c/a.	1
thickness	Å	([0, inf]) Thickness of parallelepiped. Optional parameters:	1
theta			0
Phi			0
Psi			0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_length_a		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0

Name	Unit	Description	Default
pd_thickness		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Psi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_hollow_rectangular_prism_aniso.comp`.

11.51. The `SasView_hollow_rectangular_prism_thin_walls` McXtrace Component

SasView hollow_rectangular_prism_thin_walls model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_hollow_rectangular_prism_thin_walls component, generated from hollow_rectangular_prism_th in sasmodels.

Example: SasView_hollow_rectangular_prism_thin_walls(sld, sld_solvent, length_a, b2a_ratio, c2a_ratio, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length_a=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Parallelepiped scattering length density.	6.3
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
length_a	Å	([0, inf]) Shorter side of the parallelepiped.	35
b2a_ratio	Å	([0, inf]) Ratio sides b/a.	1
c2a_ratio	Å	([0, inf]) Ratio sides c/a. Optional parameters:	1
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_length_a		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_hollow_rectangular_prism_thin_walls.comp`.

11.52. The SasView_lamellar_hg McXtrace Component

SasView lamellar_hg model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_lamellar_hg component, generated from lamellar_hg.c in sasmodels.

Example: SasView_lamellar_hg(length_tail, length_head, sld, sld_head, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length_tail=0.0, pd_length_head=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
length_tail	Å	([0, inf]) Tail thickness (total = H+T+T+H).	15
length_head	Å	([0, inf]) Head thickness.	10
sld	1e-6/Å^2	([-inf, inf]) Tail scattering length density.	0.4
sld_head	1e-6/Å^2	([-inf, inf]) Head scattering length density.	3.0
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	6
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005

Name	Unit	Description	Default
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_length_tail		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length_head		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_lamellar_hg.comp`.

11.53. The `SasView_lamellar_hg_stack_caille` McXtrace Component

SasView lamellar_hg_stack_caille model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_lamellar_hg_stack_caille component, generated from lamellar_hg_stack_caille.c in sasmodels.

Example: SasView_lamellar_hg_stack_caille(length_tail, length_head, Nlayers, d_spacing, Caille_parameter, sld, sld_head, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length_tail=0.0, pd_length_head=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
length_tail	Å	([0, inf]) Tail thickness.	10
length_head	Å	([0, inf]) head thickness.	2
Nlayers		([1, inf]) Number of layers.	30
d_spacing	Å	([0.0, inf]) lamellar d-spacing of Caille S(Q).	40.0
Caille_parameter		([0.0, 0.8]) Caille parameter.	0.001
sld	1e-6/Å^2	([-inf, inf]) Tail scattering length density.	0.4
sld_head	1e-6/Å^2	([-inf, inf]) Head scattering length density.	2.0
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	6
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5

Name	Unit	Description	Default
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_length_tail		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length_head		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_lamellar_hg_stack_caille.comp`.

11.54. The `SasView_lamellar_stack_caille` McXtrace Component

SasView lamellar_stack_caille model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_lamellar_stack_caille component, generated from lamellar_stack_caille.c in sasmodels.

Example: SasView_lamellar_stack_caille(thickness, Nlayers, d_spacing, Caille_parameter, sld, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_thickness=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
thickness	Å	([0, inf]) sheet thickness.	30.0
Nlayers		([1, inf]) Number of layers.	20
d_spacing	Å	([0.0, inf]) lamellar d-spacing of Caille S(Q).	400.0
Caille_parameter	1/Å^2	([0.0, 0.8]) Caille parameter.	0.1
sld	1e-6/Å^2	([-inf, inf]) layer scattering length density.	6.3
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	1.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_thickness		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_lamellar_stack_caille.comp`.

11.55. The `SasView_lamellar_stack_paracrystal` McXtrace Component

SasView lamellar_stack_paracrystal model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_lamellar_stack_paracrystal component, generated from lamellar_stack_paracrystal.c in sasmodels.

Example: SasView_lamellar_stack_paracrystal(thickness, Nlayers, d_spacing, sigma_d, sld, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_thickness=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
thickness	Å	([0, inf]) sheet thickness.	33.0
Nlayers		([1, inf]) Number of layers.	20
d_spacing	Å	([0.0, inf]) lamellar spacing of paracrystal stack.	250.0
sigma_d	Å	([0.0, inf]) Sigma (polydispersity) of the lamellar spacing.	0.0
sld	1e-6/Å^2	([-inf, inf]) layer scattering length density.	1.0
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	6.34
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0

Name	Unit	Description	Default
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_thickness		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_lamellar_stack_paracrystal.comp`.

11.56. The `SasView_line` McXtrace Component

SasView line model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_line component, generated from line.c in sasmodels.

Example: SasView_line(intercept, slope, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0,)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
intercept	1/cm	([-inf, inf]) intercept in linear model.	1.0
slope	Å/cm	([-inf, inf]) slope in linear model. Optional parameters:	1.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0

Links

- Component source code found in file `SasView_line.comp`.

11.57. The `SasView_linear_pearls` McXtrace Component

SasView linear_pearls model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_linear_pearls component, generated from linear_pearls.c in sasmodels.

Example: SasView_linear_pearls(radius, edge_sep, num_pearls, sld, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0, inf]) Radius of the pearls.	80.0
edge_sep	Å	([0, inf]) Length of the string segment - surface to surface.	350.0
num_pearls		([1, inf]) Number of the pearls.	3.0
sld	1e-6/Å^2	([-inf, inf]) SLD of the pearl spheres.	1.0
sld_solvent	1e-6/Å^2	([-inf, inf]) SLD of the solvent. Optional parameters:	6.3
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005

Name	Unit	Description	Default
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_linear_pearls.comp`.

11.58. The `SasView_lorentz` McXtrace Component

SasView lorentz model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_lorentz component, generated from lorentz.c in sasmodels.

Example: SasView_lorentz(cor_length, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_cor_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
cor_length	Å	([0, inf]) Screening length. Optional parameters:	50.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_cor_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_lorentz.comp`.

11.59. The `SasView_mass_fractal` McXtrace Component

SasView mass_fractal model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_mass_fractal component, generated from mass_fractal.c in sasmodels.
Example: SasView_mass_fractal(radius, fractal_dim_mass, cutoff_length, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_cutoff_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0.0, inf]) Particle radius.	10.0
fractal_dim_mass		([1.0, 6.0]) Mass fractal dimension.	1.9
cutoff_length	Å	([0.0, inf]) Cut-off length. Optional parameters:	100.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0

Name	Unit	Description	Default
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_cutoff_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_mass_fractal.comp`.

11.60. The `SasView_mass_surface_fractal` McXtrace Component

SasView mass_surface_fractal model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_mass_surface_fractal component, generated from mass_surface_fractal.c in sasmodels.

Example: SasView_mass_surface_fractal(fractal_dim_mass, fractal_dim_surf, rg_cluster, rg_primary, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_rg_cluster=0.0, pd_rg_primary=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
fractal_dim_mass		([0.0, 6.0]) Mass fractal dimension.	1.8
fractal_dim_surf		([0.0, 6.0]) Surface fractal dimension.	2.3
rg_cluster	Å	([0.0, inf]) Cluster radius of gyration.	4000.0
rg_primary	Å	([0.0, inf]) Primary particle radius of gyration. Optional parameters:	86.7
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_rg_cluster		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_rg_primary		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_mass_surface_fractal.comp`.

11.61. The `SasView_mono_gauss_coil` McXtrace Component

SasView mono_gauss_coil model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_mono_gauss_coil component, generated from mono_gauss_coil.c in sasmodels.

Example: SasView_mono_gauss_coil(i_zero, rg, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_rg=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
i_zero	1/cm	([0.0, inf]) Intensity at q=0.	70.0
rg	Å	([0.0, inf]) Radius of gyration. Optional parameters:	75.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0

Name	Unit	Description	Default
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_rg		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_mono_gauss_coil.comp`.

11.62. The `SasView_multilayer_vesicle` McXtrace Component

SasView multilayer_vesicle model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_multilayer_vesicle component, generated from multilayer_vesicle.c in sasmodels.

Example: SasView_multilayer_vesicle(volfraction, radius, thick_shell, thick_solvent, sld_solvent, sld, n_shells, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thick_shell=0.0, pd_thick_solvent=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
volfraction		([0.0, 1]) volume fraction of vesicles.	0.05
radius	Å	([0.0, inf]) radius of solvent filled core.	60.0
thick_shell	Å	([0.0, inf]) thickness of one shell.	10.0
thick_solvent	Å	([0.0, inf]) solvent thickness between shells.	10.0
sld_solvent	1e-6/Å^2	([-inf, inf]) solvent scattering length density.	6.4
sld	1e-6/Å^2	([-inf, inf]) Shell scattering length density.	0.4
n_shells		([1.0, inf]) Number of shell plus solvent layer pairs (must be integer). Optional parameters:	2.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0

Name	Unit	Description	Default
pd_thick_shell		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_solvent		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_multilayer_vesicle.comp`.

11.63. The `SasView_onion` McXtrace Component

SasView onion model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_onion component, generated from onion.c in sasmodels.

Example: SasView_onion(sld_core, radius_core, sld_solvent, n_shells, sld_in[n_shells], sld_out[n_shells], thickness[n_shells], A[n_shells], model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_core=0.0, pd_thickness[n_shells]=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default

Links

- Component source code found in file `SasView_onion.comp`.

11.64. The `SasView_parallelepiped` McXtrace Component

SasView parallelepiped model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_parallelepiped component, generated from parallelepiped.c in sasmodels.

Example: SasView_parallelepiped(sld, sld_solvent, length_a, length_b, length_c, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length_a=0.0, pd_length_b=0.0, pd_length_c=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Parallelepiped scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
length_a	Å	([0, inf]) Shorter side of the parallelepiped.	35
length_b	Å	([0, inf]) Second side of the parallelepiped.	75
length_c	Å	([0, inf]) Larger side of the parallelepiped. Optional parameters:	400
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01

Name	Unit	Description	Default
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_length_a		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length_b		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length_c		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_parallelepiped.comp`.

11.65. The `SasView_parallelepiped_aniso` McXtrace Component

SasView parallelepiped model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_parallelepiped component, generated from parallelepiped.c in sasmodels.

Example: SasView_parallelepiped_aniso(sld, sld_solvent, length_a, length_b, length_c, theta, Phi, Psi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length_a=0.0, pd_length_b=0.0, pd_length_c=0.0, pd_theta=0.0, pd_Phi=0.0, pd_Psi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Parallelepiped scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
length_a	Å	([0, inf]) Shorter side of the parallelepiped.	35
length_b	Å	([0, inf]) Second side of the parallelepiped.	75
length_c	Å	([0, inf]) Larger side of the parallelepiped. Optional parameters:	400
theta			60
Phi			60
Psi			60
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1

Name	Unit	Description	Default
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_length_a		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length_b		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_length_c		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Psi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_parallelepiped_aniso.comp`.

11.66. The `SasView_peak_lorentz` McXtrace Component

SasView peak_lorentz model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_peak_lorentz component, generated from peak_lorentz.c in sasmodels.

Example: SasView_peak_lorentz(peak_pos, peak_hwhm, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0,)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
peak_pos	1/Å	([-inf, inf]) Peak postion in q.	0.05
peak_hwhm	1/Å	([-inf, inf]) HWHM of peak. Optional parameters:	0.005
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0

Links

- Component source code found in file `SasView_peak_lorentz.comp`.

11.67. The `SasView_pearl_necklace` McXtrace Component

SasView pearl_necklace model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_pearl_necklace component, generated from pearl_necklace.c in sasmodels.

Example: SasView_pearl_necklace(radius, edge_sep, thick_string, num_pearls, sld, sld_string, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thick_string=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0, inf]) Mean radius of the chained spheres.	80.0
edge_sep	Å	([0, inf]) Mean separation of chained particles.	350.0
thick_string	Å	([0, inf]) Thickness of the chain linkage.	2.5
num_pearls	none	([1, inf]) Number of pearls in the necklace (must be integer).	3
sld	1e-6/Å^2	([-inf, inf]) Scattering length density of the chained spheres.	1.0
sld_string	1e-6/Å^2	([-inf, inf]) Scattering length density of the chain linkage.	1.0
sld_solvent	1e-6/Å^2	([-inf, inf]) Scattering length density of the solvent. Optional parameters:	6.3

Name	Unit	Description	Default
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_string		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_pearl_necklace.comp`.

11.68. The `SasView_poly_gauss_coil` McXtrace Component

SasView poly_gauss_coil model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_poly_gauss_coil component, generated from poly_gauss_coil.c in sasmodels.

Example: SasView_poly_gauss_coil(i_zero, rg, polydispersity, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_rg=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
i_zero	1/cm	([0.0, inf]) Intensity at q=0.	70.0
rg	Å	([0.0, inf]) Radius of gyration.	75.0
polydispersity	None	([1.0, inf]) Polymer Mw/Mn. Optional parameters:	2.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5

Name	Unit	Description	Default
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_rg		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_poly_gauss_coil.comp`.

11.69. The `SasView_polymer_excl_volume` McXtrace Component

SasView polymer_excl_volume model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_polymer_excl_volume component, generated from polymer_excl_volume.c in sasmodels.

Example: SasView_polymer_excl_volume(rg, porod_exp, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_rg=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
rg	Å	([0, inf]) Radius of Gyration.	60.0

Name	Unit	Description	Default
porod_exp		([0, inf]) Porod exponent. Optional parameters:	3.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_rg		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_polymer_excl_volume.comp`.

11.70. The `SasView_polymer_micelle` McXtrace Component

SasView polymer_micelle model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_polymer_micelle component, generated from polymer_micelle.c in sasmodels.

Example: SasView_polymer_micelle(ndensity, v_core, v_corona, sld_solvent, sld_core, sld_corona, radius_core, rg, d_penetration, n_aggreg, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_core=0.0, pd_rg=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
ndensity	1e15/cm^3	([0.0, inf]) Number density of micelles.	8.94
v_core	Å^3	([0.0, inf]) Core volume .	62624.0
v_corona	Å^3	([0.0, inf]) Corona volume.	61940.0
sld_solvent	1e-6/Å^2	([0.0, inf]) Solvent scattering length density.	6.4
sld_core	1e-6/Å^2	([0.0, inf]) Core scattering length density.	0.34
sld_corona	1e-6/Å^2	([0.0, inf]) Corona scattering length density.	0.8
radius_core	Å	([0.0, inf]) Radius of core (must be >> rg).	45.0
rg	Å	([0.0, inf]) Radius of gyration of chains in corona.	20.0
d_penetration		([-inf, inf]) Factor to mimic non-penetration of Gaussian chains.	1.0
n_aggreg		([-inf, inf]) Aggregation number of the micelle. Optional parameters:	6.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0

Name	Unit	Description	Default
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_core		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_rg		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_polymer_micelle.comp`.

11.71. The `SasView_porod` McXtrace Component

SasView porod model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_porod component, generated from porod.c in sasmodels.

Example: SasView_porod(, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0,)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0

Links

- Component source code found in file `SasView_porod.comp`.

11.72. The `SasView_power_law` McXtrace Component

SasView power_law model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_power_law component, generated from power_law.c in sasmodels.
Example: SasView_power_law(power, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0,)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
power		([-inf, inf]) Power law exponent. Optional parameters:	4.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1

Name	Unit	Description	Default
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0

Links

- Component source code found in file `SasView_power_law.comp`.

11.73. The `SasView_pringle` McXtrace Component

SasView pringle model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_pringle component, generated from pringle.c in sasmodels.

Example: SasView_pringle(radius, thickness, alpha, beta, sld, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thickness=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0, inf]) Pringle radius.	60.0
thickness	Å	([0, inf]) Thickness of pringle.	10.0
alpha		([-inf, inf]) Curvature parameter alpha.	0.001
beta		([-inf, inf]) Curvature paramter beta.	0.02
sld	1e-6/Å^2	([-inf, inf]) Pringle sld.	1.0

Name	Unit	Description	Default
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent sld. Optional parameters:	6.3
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thickness		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_pringle.comp`.

11.74. The `SasView_raspberry` McXtrace Component

SasView raspberry model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_raspberry component, generated from raspberry.c in sasmodels.

Example: SasView_raspberry(sld_lg, sld_sm, sld_solvent, volfraction_lg, volfraction_sm, surface_fraction, radius_lg, radius_sm, penetration, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_lg=0.0, pd_radius_sm=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld_lg	1e-6/Å^2	([-inf, inf]) large particle scattering length density.	-0.4
sld_sm	1e-6/Å^2	([-inf, inf]) small particle scattering length density.	3.5
sld_solvent	1e-6/Å^2	([-inf, inf]) solvent scattering length density.	6.36
volfraction_lg		([-inf, inf]) volume fraction of large spheres.	0.05
volfraction_sm		([-inf, inf]) volume fraction of small spheres.	0.005
surface_fraction		([-inf, inf]) fraction of small spheres at surface.	0.4
radius_lg	Å	([0, inf]) radius of large spheres.	5000
radius_sm	Å	([0, inf]) radius of small spheres.	100
penetration	Å	([-1, 1]) fractional penetration depth of small spheres into large sphere. Optional parameters:	0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0

Name	Unit	Description	Default
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_lg		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius_sm		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_raspberry.comp`.

11.75. The `SasView_rectangular_prism` McXtrace Component

SasView rectangular_prism model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld			6.3
sld_solvent			1
length_a			35
b2a_ratio			1
c2a_ratio			1
model_scale			1.0
model_abs			0.0
xwidth			0.01
yheight			0.01
zdepth			0.005
R			0
target_x			0
target_y			0
target_z			1
target_index			1
focus_xw			0.5
focus_yh			0.5
focus_aw			0
focus_ah			0
focus_r			0
pd_length_a			0.0

Links

- Component source code found in file `SasView_rectangular_prism.comp`.

11.76. The `SasView_rectangular_prism_aniso` McXtrace Component

SasView rectangular_prism model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_rectangular_prism component, generated from rectangular_prism.c in sasmodels.

Example: SasView_rectangular_prism_aniso(sld, sld_solvent, length_a, b2a_ratio, c2a_ratio, theta, Phi, Psi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length_a=0.0, pd_theta=0.0, pd_Phi=0.0, pd_Psi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Parallelepiped scattering length density.	6.3
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
length_a	Å	([0, inf]) Shorter side of the parallelepiped.	35
b2a_ratio		([0, inf]) Ratio sides b/a.	1
c2a_ratio		([0, inf]) Ratio sides c/a. Optional parameters:	1
theta			0
Phi			0
Psi			0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1

Name	Unit	Description	Default
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_length_a		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Psi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_rectangular_prism_aniso.comp`.

11.77. The `SasView_rpa` McXtrace Component

SasView rpa model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_rpa component, generated from rpa.c in sasmodels.
Example:

SasView_rpa(case_num, N[4], Phi[4], v[4], L[4], b[4], K12, K13, K14, K23, K24, K34,

model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0,)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default

Links

- Component source code found in file `SasView_rpa.comp`.

11.78. The `SasView_sc_paracrystal` McXtrace Component

SasView sc_paracrystal model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_sc_paracrystal component, generated from sc_paracrystal.c in sasmodels.

Example: SasView_sc_paracrystal(dnn, d_factor, radius, sld, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
dnn	Å	([0.0, inf]) Nearest neighbor distance.	220.0
d_factor		([-inf, inf]) Paracrystal distortion factor.	0.06
radius	Å	([0.0, inf]) Radius of sphere.	40.0

Name	Unit	Description	Default
sld	1e-6/Å^2	([0.0, inf]) Sphere scattering length density.	3.0
sld_solvent	1e-6/Å^2	([0.0, inf]) Solvent scattering length density. Optional parameters:	6.3
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_sc_paracrystal.comp`.

11.79. The SasView_sc_paracrystal_aniso McXtrace Component

SasView sc_paracrystal model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_sc_paracrystal component, generated from sc_paracrystal.c in sasmodels.

Example: SasView_sc_paracrystal_aniso(dnn, d_factor, radius, sld, sld_solvent, theta, Phi, Psi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_theta=0.0, pd_Phi=0.0, pd_Psi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
dnn	Å	([0.0, inf]) Nearest neighbor distance.	220.0
d_factor		([-inf, inf]) Paracrystal distortion factor.	0.06
radius	Å	([0.0, inf]) Radius of sphere.	40.0
sld	1e-6/Å^2	([0.0, inf]) Sphere scattering length density.	3.0
sld_solvent	1e-6/Å^2	([0.0, inf]) Solvent scattering length density. Optional parameters:	6.3
theta			0
Phi			0
Psi			0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01

Name	Unit	Description	Default
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Psi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_sc_paracrystal_aniso.comp`.

11.80. The `SasView_sphere` McXtrace Component

SasView sphere model component as sample description.

Identification

- **Author:** Jose Robledo

- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_sphere component, generated from sphere.c in sasmodels.

Example: SasView_sphere(sld, sld_solvent, radius, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Layer scattering length density.	1
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	6
radius	Å	([0, inf]) Sphere radius. Optional parameters:	50
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5

Name	Unit	Description	Default
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_sphere.comp`.

11.81. The `SasView_spinodal` McXtrace Component

SasView spinodal model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_spinodal component, generated from spinodal.c in sasmodels.

Example: SasView_spinodal(gamma, q_0, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0,)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
gamma		([-inf, inf]) Exponent.	3.0
q_0	1/Å	([-inf, inf]) Correlation peak position. Optional parameters:	0.1

Name	Unit	Description	Default
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0

Links

- Component source code found in file `SasView_spinodal.comp`.

11.82. The `SasView_squarewell` McXtrace Component

SasView squarewell model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_squarewell component, generated from squarewell.c in sasmodels.

Example: SasView_squarewell(radius_effective, volfraction, welldepth, wellwidth, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_effective=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_effective	Å	([0, inf]) effective radius of hard sphere.	50.0
volfraction		([0, 0.08]) volume fraction of spheres.	0.04
welldepth	kT	([0.0, 1.5]) depth of well, epsilon.	1.5
wellwidth	diameters	([1.0, inf]) width of well in diameters (=2R) units, must be > 1. Optional parameters:	1.2
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0

Name	Unit	Description	Default
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_effective		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_squarewell.comp`.

11.83. The `SasView_stacked_disks` McXtrace Component

SasView stacked_disks model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_stacked_disks component, generated from stacked_disks.c in sasmodels.

Example: SasView_stacked_disks(thick_core, thick_layer, radius, n_stacking, sigma_d, sld_core, sld_layer, sld_solvent, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_thick_core=0.0, pd_thick_layer=0.0, pd_radius=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
thick_core	Å	([0, inf]) Thickness of the core disk.	10.0
thick_layer	Å	([0, inf]) Thickness of layer each side of core.	10.0
radius	Å	([0, inf]) Radius of the stacked disk.	15.0
n_stacking		([1, inf]) Number of stacked layer/core/layer disks.	1.0
sigma_d	Å	([0, inf]) Sigma of nearest neighbor spacing.	0

Name	Unit	Description	Default
sld_core	1e-6/Å^2	([-inf, inf]) Core scattering length density.	4
sld_layer	1e-6/Å^2	([-inf, inf]) Layer scattering length density.	0.0
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	5.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_thick_core		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_layer		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_stacked_disks.comp`.

11.84. The `SasView_stacked_disks_aniso` McXtrace Component

SasView stacked_disks model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_stacked_disks component, generated from stacked_disks.c in sasmodels.

Example: SasView_stacked_disks_aniso(thick_core, thick_layer, radius, n_stacking, sigma_d, sld_core, sld_layer, sld_solvent, theta, Phi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_thick_core=0.0, pd_thick_layer=0.0, pd_radius=0.0, pd_theta=0.0, pd_Phi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
thick_core	Å	([0, inf]) Thickness of the core disk.	10.0
thick_layer	Å	([0, inf]) Thickness of layer each side of core.	10.0
radius	Å	([0, inf]) Radius of the stacked disk.	15.0
n_stacking		([1, inf]) Number of stacked layer/core/layer disks.	1.0
sigma_d	Å	([0, inf]) Sigma of nearest neighbor spacing.	0
sld_core	1e-6/Å^2	([-inf, inf]) Core scattering length density.	4
sld_layer	1e-6/Å^2	([-inf, inf]) Layer scattering length density.	0.0
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density. Optional parameters:	5.0

Name	Unit	Description	Default
theta			0
Phi			0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_thick_core		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thick_layer		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0

Name	Unit	Description	Default
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_stacked_disks_aniso.comp`.

11.85. The `SasView_star_polymer` McXtrace Component

SasView star_polymer model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_star_polymer component, generated from star_polymer.c in sasmodels.

Example: SasView_star_polymer(rg_squared, arms, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_rg_squared=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
rg_squared	$Å^2$	([0.0, inf]) Ensemble radius of gyration SQUARED of the full polymer.	100.0
arms		([1.0, 6.0]) Number of arms in the model. Optional parameters:	3
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0

Name	Unit	Description	Default
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_rg_squared		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_star_polymer.comp`.

11.86. The `SasView_stickyhardsphere` McXtrace Component

SasView stickyhardsphere model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_stickyhardsphere component, generated from stickyhardsphere.c in sasmodels.

Example: SasView_stickyhardsphere(radius_effective, volfraction, perturb, stickiness, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_effective=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_effective	Å	([0, inf]) effective radius of hard sphere.	50.0
volfraction		([0, 0.74]) volume fraction of hard spheres.	0.2
perturb		([0.01, 0.1]) perturbation parameter, tau.	0.05
stickiness		([-inf, inf]) stickiness, epsilon. Optional parameters:	0.2
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0

Name	Unit	Description	Default
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_effective		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_stickyhardsphere.comp`.

11.87. The `SasView_superball` McXtrace Component

SasView superball model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_superball component, generated from superball.c in sasmodels.

Example: SasView_superball(sld, sld_solvent, length_a, exponent_p, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length_a=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Superball scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
length_a	Å	([0, inf]) Cube edge length of the superball.	50

Name	Unit	Description	Default
exponent_p		([0, inf]) Exponent describing the roundness of the superball. Optional parameters:	2.5
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_length_a		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_superball.comp`.

11.88. The `SasView_superball_aniso` McXtrace Component

SasView superball model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_superball component, generated from superball.c in sasmodels.

Example: SasView_superball_aniso(sld, sld_solvent, length_a, exponent_p, theta, Phi, Psi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_length_a=0.0, pd_theta=0.0, pd_Phi=0.0, pd_Psi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Superball scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
length_a	Å	([0, inf]) Cube edge length of the superball.	50
exponent_p		([0, inf]) Exponent describing the roundness of the superball. Optional parameters:	2.5
theta			0
Phi			0
Psi			0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005

Name	Unit	Description	Default
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_length_a		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Psi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_superball_aniso.comp`.

11.89. The `SasView_surface_fractal` McXtrace Component

SasView surface_fractal model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_surface_fractal component, generated from surface_fractal.c in sasmodels.

Example: SasView_surface_fractal(radius, fractal_dim_surf, cutoff_length, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_cutoff_length=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius	Å	([0, inf]) Particle radius.	10.0
fractal_dim_surf		([1, 3]) Surface fractal dimension.	2.0
cutoff_length	Å	([0.0, inf]) Cut-off Length. Optional parameters:	500.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0

Name	Unit	Description	Default
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_cutoff_length		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_surface_fractal.comp`.

11.90. The `SasView_teubner_strey` McXtrace Component

SasView teubner_strey model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_teubner_strey component, generated from teubner_strey.c in sasmodels.
Example: SasView_teubner_strey(volfraction_a, sld_a, sld_b, d, xi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0,)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
volfraction_a		([0, 1.0]) Volume fraction of phase a.	0.5
sld_a	1e-6/Å^2	([-inf, inf]) SLD of phase a.	0.3
sld_b	1e-6/Å^2	([-inf, inf]) SLD of phase b.	6.3
d	Å	([0, inf]) Domain size (periodicity).	100.0
xi	Å	([0, inf]) Correlation length. Optional parameters:	30.0

Name	Unit	Description	Default
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0

Links

- Component source code found in file `SasView_teubner_strey.comp`.

11.91. The `SasView_triaxial_ellipsoid` McXtrace Component

SasView triaxial_ellipsoid model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_triaxial_ellipsoid component, generated from triaxial_ellipsoid.c in sasmodels.

Example: SasView_triaxial_ellipsoid(sld, sld_solvent, radius_equat_minor, radius_equat_major, radius_polar, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_equat_minor=0.0, pd_radius_equat_major=0.0, pd_radius_polar=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Ellipsoid scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
radius_equat_minor	Å	([0, inf]) Minor equatorial radius, Ra.	20
radius_equat_major	Å	([0, inf]) Major equatorial radius, Rb.	400
radius_polar	Å	([0, inf]) Polar radius, Rc. Optional parameters:	10
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5

Name	Unit	Description	Default
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_radius_equat_minor		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius_equat_major		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius_polar		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_triaxial_ellipsoid.comp`.

11.92. The `SasView_triaxial_ellipsoid_aniso` McXtrace Component

SasView triaxial_ellipsoid model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_triaxial_ellipsoid component, generated from triaxial_ellipsoid.c in sasmodels.

Example: SasView_triaxial_ellipsoid_aniso(sld, sld_solvent, radius_equat_minor, radius_equat_major radius_polar, theta, Phi, Psi, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius_equat_minor=0.0, pd_radius_equat_majo pd_radius_polar=0.0, pd_theta=0.0, pd_Phi=0.0, pd_Psi=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) Ellipsoid scattering length density.	4
sld_solvent	1e-6/Å^2	([-inf, inf]) Solvent scattering length density.	1
radius_equat_minor	Å	([0, inf]) Minor equatorial radius, Ra.	20
radius_equat_major	Å	([0, inf]) Major equatorial radius, Rb.	400
radius_polar	Å	([0, inf]) Polar radius, Rc. Optional parameters:	10
theta			60
Phi			60
Psi			60
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0

Name	Unit	Description	Default
pd_radius_equat_minor		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius_equat_major		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_radius_polar		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_theta		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Phi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_Psi		(0,360) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_triaxial_ellipsoid_aniso.comp`.

11.93. The `SasView_two_lorentzian` McXtrace Component

SasView two_lorentzian model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_two_lorentzian component, generated from two_lorentzian.c in sasmodels.

Example: SasView_two_lorentzian(lorentz_scale_1, lorentz_length_1, lorentz_exp_1, lorentz_scale_2, lorentz_length_2, lorentz_exp_2, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_lorentz_length_1=0.0, pd_lorentz_length_2=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
lorentz_scale_1		([-inf, inf]) First power law scale factor.	10.0
lorentz_length_1	Å	([-inf, inf]) First Lorentzian screening length.	100.0
lorentz_exp_1		([-inf, inf]) First exponent of power law.	3.0
lorentz_scale_2		([-inf, inf]) Second scale factor for broad Lorentzian peak.	1.0
lorentz_length_2	Å	([-inf, inf]) Second Lorentzian screening length.	10.0
lorentz_exp_2		([-inf, inf]) Second exponent of power law. Optional parameters:	2.0
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0
pd_lorentz_length_1		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0

Name	Unit	Description	Default
pd_lorentz_length_2		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_two_lorentzian.comp`.

11.94. The `SasView_two_power_law` McXtrace Component

SasView two_power_law model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_two_power_law component, generated from two_power_law.c in sasmodels.

Example: SasView_two_power_law(coefficent_1, crossover, power_1, power_2, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1, target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0,)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
coefficent_1		([-inf, inf]) coefficent A in low Q region.	1.0
crossover	1/Å	([0, inf]) crossover location.	0.04
power_1		([0, inf]) power law exponent at low Q.	1.0
power_2		([0, inf]) power law exponent at high Q.	4.0
		Optional parameters:	
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0

Name	Unit	Description	Default
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0

Links

- Component source code found in file `SasView_two_power_law.comp`.

11.95. The `SasView_vesicle` McXtrace Component

SasView vesicle model component as sample description.

Identification

- **Author:** Jose Robledo
- **Origin:** FZJ / DTU / ESS DMSC
- **Date:**

Description

SasView_vesicle component, generated from vesicle.c in sasmodels.

Example: SasView_vesicle(sld, sld_solvent, volfraction, radius, thickness, model_scale=1.0, model_abs=0.0, xwidth=0.01, yheight=0.01, zdepth=0.005, R=0, int target_index=1,

target_x=0, target_y=0, target_z=1, focus_xw=0.5, focus_yh=0.5, focus_aw=0, focus_ah=0, focus_r=0, pd_radius=0.0, pd_thickness=0.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
sld	1e-6/Å^2	([-inf, inf]) vesicle shell scattering length density.	0.5
sld_solvent	1e-6/Å^2	([-inf, inf]) solvent scattering length density.	6.36
volfraction		([0, 1.0]) volume fraction of shell.	0.05
radius	Å	([0, inf]) vesicle core radius.	100
thickness	Å	([0, inf]) vesicle shell thickness. Optional parameters:	30
model_scale		Global scale factor for scattering kernel. For systems without inter-particle interference, the form factors can be related to the scattering intensity by the particle volume fraction.	1.0
model_abs		Absorption cross section density at 2200 m/s.	0.0
xwidth	m	([-inf, inf]) Horiz. dimension of sample, as a width.	0.01
yheight	m	([-inf, inf]) vert . dimension of sample, as a height for cylinder/box	0.01
zdepth	m	([-inf, inf]) depth of sample	0.005
R	m	Outer radius of sample in (x,z) plane for cylinder/sphere.	0
target_x	m	relative focus target position.	0
target_y	m	relative focus target position.	0
target_z	m	relative focus target position.	1
target_index		Relative index of component to focus at, e.g. next is +1.	1
focus_xw	m	horiz. dimension of a rectangular area.	0.5
focus_yh	m	, vert. dimension of a rectangular area.	0.5
focus_aw	deg	, horiz. angular dimension of a rectangular area.	0
focus_ah	deg	, vert. angular dimension of a rectangular area.	0
focus_r	m	case of circular focusing, focusing radius.	0

Name	Unit	Description	Default
pd_radius		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable.	0.0
pd_thickness		(0,inf) defined as (dx/x), where x is de mean value and dx the standard devition of the variable	0.0

Links

- Component source code found in file `SasView_vesicle.comp`.

12. Contributed components

The `contrib` category holds components contributed directly by McXtrace users, rather than authored and maintained by the core McXtrace team. The McXtrace team provides the same technical support for these as for any other component, but the original authors – credited individually in each component's source header and reproduced below – carry primary responsibility for the underlying physics and its validation. Always check with `mxdoc` rather than an older document for a component's current category.

12.1. Contributed optics: crystals, masks, and mirrors

12.2. The `Bragg_crystal_simple` McXtrace Component

Perfect Bragg reflecting slab

Identification

- **Author:** Jose I. Robledo
- **Origin:** FaMAF - UNC, Argentina
- **Date:** February 2016

Description

Rectangle of matter perfectly reflecting the incident X-ray beam that fulfills Bragg's law for a set of scattering vectors in the vicinity of the theoretical Q given a d-spacing. The rectangle is in the x-y plane.

Example: Bragg_crystal_simple(yheight=0.05, xwidth=0.02, DM=3.1356, err_Q= 0.000075, r0=1.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
xmin	m	Lower x bound (overridden by xwidth/yheight)	1.0
xmax	m	Upper x bound (overridden by xwidth/yheight)	1.0

Name	Unit	Description	Default
ymin	m	Lower y bound (overridden by xwidth/yheight)	1.0
ymax	m	Upper y bound (overridden by xwidth/yheight)	1.0
xwidth	m	Width in the x direction	0.0
yheight	m	Height in the y direction	0.0
r0		Maximum reflectivity	1
DM	$Å^{-1}$	d-spacing of the crystal	0
err_Q		dQ/Q relative error of the modulus of Q vector. Approximates the Darwin width of the crystal.	0.0001

Links

- Component source code found in file `Bragg_crystal_simple.comp`.

12.3. The `Bragg_crystal_BC` McXtrace Component

Perfect, reflecting crystal with common cubic structures (diamond, fcc, or bcc, and others if symmetry form factor multipliers provided explicitly)

Identification

- **Author:** Marcus H Mendenhall, NIST <marcus.mendenhall@nist.gov>
- **Origin:** NIST
- **Date:** May, 2017

Description

Bragg_crystal_BC.comp is intended to supercede Bragg_Crystal.comp.

For details see: The optics of focusing bent-crystal monochromators on X-ray powder diffractometers with application to lattice parameter determination and microstructure analysis, Marcus H. Mendenhall, David Black and James P. Cline, J. Appl. Cryst. (2019). 52, https://doi.org/10.1107/S1600576719010951

Reads atomic formfactors from a data input file. The Bragg_Crystal code reflects ray in an ideal geometry, does not include surface imperfections or mosaicity

The crystal code reflects ray in an ideal geometry, i.e. does not include surface imperfections or mosaicity. The crystal planes from which the reflection is made lies in the X-Z plane on the unbent crystal rotated by an angle alpha about the Y axis with respect to the crystal surface.

The crystal itself is set in the X-Z plane positioned such that the long axis of the crystal surface coincides with the Z-axis, withs normal pointing in the poisitivce Y-direction.

N.B. The component does not work for rays hitting the back of the monochromator.

Bragg_crystal_BC.comp is written by Marcus H. Mendenhall, NIST, Gaithersburg, MD, USA It is based on the full vector math and exact solution of the dispersion relation in Batterman and Cole, Reviews of Modern Physics 36 number 3, page 681, July 1964

This code has been validated against both experimental data (2 channel-cut 3-bounce Si 440 crystals together in non-dispersive mode, at Cu kalpha) and against theoretical rocking rocking curves from XOP for Si220 at Sc kalpha and Si440 at Cu kalpha.

Non-copyright notice: Contributed by the National Institute of Standards and Technology; not subject to copyright in the United States. This is not an official contribution, in that the results are in no way certified by NIST.

Example: Bragg_crystal_BC(length=0.05, width=0.02, V=160.1826, h=1, k=1, l=1, alphay=1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
length	m	z depth (length) of the crystal.	0.05
width	m	x width of the crystal.	0.02
V	Å^3	unit cell volume	160.1826
form_factors	str	Data file with atomic form factors	"FormFactors.txt"
material		Si, Ge (maybe also GaAs?)	"Si.txt"
alphax	1	x-component of the normal (unit vector) to the crystal planes. The crystal surface itself has normal [0,1,0].	0.0
alphay	1	y-component of the normal (unit vector) to the crystal planes.	1.0
alphaz	1	z-component of the normal (unit vector) to the crystal planes.	0.0
R0		Reflectivity. Overrides the computed Darwin reflectivity. Probably only useful for debugging.	0
debye_waller_B	Å^2	Debye-Waller temperature factor, M=B*(sin(theta)/lambda)^2*(2/3), default=silicon at room temp.	0.4632

Name	Unit	Description	Default
crystal_type		1 => Mx_crystal_explicit: provide explicit real and imaginary form factor multipliers structure_factor_scale_r, structure_factor_scale_i; 2 => Mx_crystal_diamond: diamond; 3 => Mx_crystal_fcc: fcc; 4 => Mx_crystal_fcc: bcc	1
h		Miller index of reflection	1
k		Miller index of reflection	1
l		Miller index of reflection	1
structure_factor_scale_r		real part of complex explicit override of structure factor multiplier for crystal structure if Bragg_crystal_explicit	0.0
structure_factor_scale_i		imaginary part of complex explicit override of structure factor multiplier for crystal structure if Bragg_crystal_explicit	0.0
verbose		if non-zero: Output more information (warnings and messages) to the console.	0

Links

- Component source code found in file `Bragg_crystal_BC.comp`.
- material datafile obtained from http://physics.nist.gov/cgi-bin/ffast/ffast.pl

12.4. The `Bragg_crystal_bent_BC` McXtrace Component

Bent, perfect crystal with common cubic structures (diamond, fcc, or bcc, and others if symmetry form factor multipliers provided explicitly)

Identification

- **Author:** Marcus H Mendenhall, NIST <marcus.mendenhall@nist.gov>
- **Origin:** NIST
- **Date:** Dec 2016

Description

Bragg_crystal_bent_BC.com is intended to supercede Bragg_Crystal_bent.comp For details see: The optics of focusing bent-crystal monochromators on X-ray powder diffractometers with application to lattice parameter determination and microstructure anal-

ysis, Marcus H. Mendenhall,* David Black and James P. Cline, J. Appl. Cryst. (2019). 52, https://doi.org/10.1107/S1600576719010951

Reads atomic formfactors from a data input file. The Bragg_Crystal code reflects ray in an ideal geometry, does not include surface imperfections or mosaicity

The crystal code reflects ray in an ideal geometry, i.e. does not include surface imperfections or mosaicity. The crystal planes from which the reflection is made lies in the X-Z plane on the unbent crystal rotated by an angle alpha about the Y axis with respect to the crystal surface.

The crystal itself is set in the X-Z plane positioned such that the long axis of the crystal surface coincides with the Z-axis, withs normal pointing in the poisitivce Y-direction.

The asummetry angle alpha is defined so that positive alpha reduces the Bragg angle to the plane i.e. alpha=Thetain grazes the planes. if alpha!=0, one should restrict to rays which have small kx values, since otherwise the alpha rotation is not around the diffraction axis.

The mirror is positioned such that the a-axis of the mirror ellipsoid is on the z-axis, the b-axis is along the y-axis and the c is along the x-axis. The reference point of the mirror is the ellipsoid centre, offset by one half-axis along the y-axis. (See the component manual for a drawing).

Notation follows Tadashi Matsushita and Hiro-O Hashizume, X-RAY MONOCHROMATORS. Handbook on Synchrotron Radiation,North-Holland Publishing Company, 1:263–274, 1983.

NOTE: elliptical coordinate code and documentation taken from Mirror_elliptic.comp distributed in McXtrace v1.2 written by: Erik Knudsen. However, the coordinates are rotated to be consistent with Perfect_Crystal.comp and NIST_Perfect_Crystal.comp Idealized elliptic mirror with surface ellipse and lattice ellipses independent, to allow construction of Johansson optics, for example.

Non-copyright notice: Contributed by the National Institute of Standards and Technology; not subject to copyright in the United States. This is not an official contribution, in that the results are in no way certified by NIST.

Example: Bragg_crystal_bent_BC(length=0.05, width=0.02, V=160.1826, h=1, k=1, l=1, alpha=0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
x_a	m	1st short half axis (along x). Commonly set to zero, which really implies infinite value, so crystal is an elliptic cylinder.	0
y_b	m	2nd short half axis (along y), which is also the presumed near-normal direction, reflection near the y-z plane.	1.0

Name	Unit	Description	Default
z_c	m	Long half axis (along z). Commonly a=0. b=c, which creates a circular cylindrical surface.	1.0
lattice_x_a	m	Curvature matrix component around a for underlying lattice, for bent/ground/rebent crystals	0
lattice_y_b	m	Curvature matrix component around b for underlying lattice, for bent/ground/rebent crystals	1.0
lattice_z_c	m	curvature matrix component around c for underlying lattice, for bent/ground/rebent crystals THERE HAS BEEN NO TESTING for the case in which lattice_x_a != x_a.	1.0
length	m	z depth (length) of the crystal.	0.05
width	m	x width of the crystal.	0.02
V	$Å^3$	unit cell volume	160.1826
form_factors	str	Data file with atomic form factors	"FormFactors.txt"
material		Si, Ge (maybe also GaAs?)	"Si.txt"
alpha	rad	Asymmetry angle (alpha=0 for symmetric reflection, i.e. the Bragg planes are parallel to the crystal surface)	0.0
R0		Reflectivity. Overrides the computed Darwin reflectivity. Probably only useful for debugging.	0
debye_waller_B	$Å^2$	Debye-Waller temperature factor, M=B*(sin(theta)/lambda)^2*(2/3), default=silicon at room temp.	0.4632
crystal_type		1 => Mx_crystal_explicit: provide explicit real and imaginary form factor multipliers structure_factor_scale_r, structure_factor_scale_i; 2 => Mx_crystal_diamond: diamond; 3 => Mx_crystal_fcc: fcc; 4 => Mx_crystal_fcc: bcc	1
h		Miller index of reflection	1
k		Miller index of reflection	1
l		Miller index of reflection	1
structure_factor_scale_r		real part of complex explicit override of structure factor multiplier for crystal structure if Bragg_crystal_explicit	0.0

Name	Unit	Description	Default
structure_factor_scale_i	1	imaginary part of complex explicit override of structure factor multiplier for crystal structure if Bragg_crystal_explicit	0.0
verbose		if non-zero: Output more information (warnings and messages) to the console.	0

Links

- Component source code found in file `Bragg_crystal_bent_BC.comp`.
- material datafile obtained from http://physics.nist.gov/cgi-bin/ffast/ffast.pl

12.5. The `Laue_crystal_BC` McXtrace Component

Perfect, Laue crystal with common cubic structures (diamond, fcc, or bcc, and others if symmetry form factor multipliers provided explicitly)

Identification

- **Author:** Marcus H Mendenhall, NIST <marcus.mendenhall@nist.gov>
- **Origin:** NIST
- **Date:** June, 2017

Description

NIST_Laue_crystal_BC.comp is written by Marcus H. Mendenhall, NIST, Gaithersburg, MD, USA It is based on the full vector math and exact solution of the dispersion relation in Batterman and Cole, Reviews of Modern Physics 36 number 3, page 681, July 1964 Perfect crystal with common cubic structures (diamond, fcc, or bcc, and others if symmetry form factor multipliers provided explicitly)

Reads atomic form factors from a data input file. The Laue_Crystal code reflects rays in an ideal geometry, does not include surface imperfections or mosaicity.

The crystal is positioned such that the long axis of the crystal surface coincides with z-axis and the outer normal to the crystal surface is along +y.

The ratio of the transmitted beam and forward-diffracted Borrman-effect beam is a hack. The sum of the two is exactly right, but the actual ratio depends critically on geometry, and I just put in a wild estimate to allow one to demonstrate what the Borrmann effect looks like. If this is turned on, the displacement of the transmitted beam and forward diffracted beam at the back side of the crystal will be correctly computed. This displacement is only exact for symmetrical Laue; asymmetrical computation requires more effort, and is probably not worth it. The sampling of these processes are

controlled by the 3 variables transmission_sampling, forward_diffraction_sampling, and Laue_sampling. Since 99% of uses of this will have the transmitted beam turned off, and use Laue diffraction mode, the values should be just 0,0,1. If the general behavior of the transmitted beams is interesting, use 1,1,1 which samples all beams equally. Results weights are adjusted for this, so computed intensities won't be affected.

Non-copyright notice: Contributed by the National Institute of Standards and Technology; not subject to copyright in the United States. This is not an official contribution, in that the results are in no way certified by NIST.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
length	m	zdepth (length) of the crystal.	0.05
width	m	width of the crystal.	0.02
thickness	m	thickness of crystal (along y-axis, the surface normal)	1e-4
V	Å^3	unit cell volume	160.1826
form_factors		"FormFactors.txt" from McXtrace install, usually	"FormFactors.txt"
material		Si ("Si.txt"), Ge ("Ge.txt")	"Si.txt"
alphax		x component of normal (unit vector) to crystal planes. Vector is usually [0,0,1] for symmetric Laue. Crystal surface itself has normal [0,1,0].	0.0
alphay		y component of normal (unit vector) to crystal planes.	0.0
alphaz		z component of normal (unit vector) to crystal planes.	1.0
debye_waller_B	Å^2	Debye-Waller temperature factor, M=B*(sin(theta)/lambda)^2*(2/3), default=silicon at room temp, 0.4632	0.4632
crystal_type		1 => Mx_crystal_explicit: provide explicit real and imaginary form factor multipliers structure_factor_scale_r, structure_factor_scale_i, 2 => Mx_crystal_diamond: diamond 3 => Mx_crystal_fcc: fcc 4 => Mx_crystal_fcc: bcc	1
h		1st Miller index of reflection	1
k		2nd Miller index of reflection	1
l		3rd Miller index of reflection	1

Name	Unit	Description	Default
structure_factor_scale_r		real part of complex explicit override of structure factor multiplier for crystal structure if Bragg_crystal_explicit	0.0
structure_factor_scale_i		imaginary part of complex explicit override of structure factor multiplier for crystal structure if Bragg_crystal_explicit	0.0
transmission_sampling		enable sampling of transmission diffraction mode.	1.0
forward_diffraction_sampling		enable sampling of forward diffraftion mode.	1.0
laue_sampling		enable sampling of Laue diffraction mode.	1.0

Links

- Component source code found in file `Laue_crystal_BC.comp`.
- material datafile obtained from http://physics.nist.gov/cgi-bin/ffast/ffast.pl

12.6. The `Attenuating_mask` McXtrace Component

Attenuating_mask

Identification

- **Author:** Matteo Busi, Erik B Knudsen
- **Origin:** DTU Physics
- **Date:** November 2017

Description

This component models a mask of energy dependent attenuation. This consists of a rectangular grid of size "xwidth*yheight", composed of multiple disks of finite thickness "zdepth" of an attenuating material "att_file", width "blocks_width" and period "blocks_dist"(i.e. distance between the center of each disk). If holed_mask mode is turned the model of the component is the opposite, i.e. the mask is composed of an attenuating slab of finite thickness and of size "xwidth*yheight", with apertures of desired width and period.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
att_file	".txt"	File that contains the object information. (Default: "W.txt")	"W.txt"
xwidth	m	Horizontal width of the mask. (Default: 1e-1)	1e-1
yheight	m	Vertical height of the mask. (Default: 1e-1)	1e-1
zdepth	m	Thickness of the absorbing mask. (Default: 3e-3)	10e-6
blocks_xwidth	m	Width of the absorbing blocks in the x-direction. (Default: 3e-3)	1e-3
blocks_xdist	m	Distance between absorbing blocks in the x-direction. (Default: 10e-3)	2.5e-3
blocks_yheight	m	Height of the absorbing blocks in the y-direction. (Default: 3e-3)	1e-3
blocks_ydist	m	Distance between absorbing blocks in the y-direction. (Default: 10e-3)	2.5e-2
holed_mask	1	Set to 1 if the mask is a holed grid. (Default: 0)	0

Links

- Component source code found in file `Attenuating_mask.comp`.

12.7. The `Mirror_toroid_pothole` McXtrace Component

Toroidal shape mirror (in XZ)

Identification

- **Author:** Erik B Knudsen
- **Origin:** DTU Physics
- **Date:** Jul 2016

Description

This is an implementation of a toroidal mirror which may be curved in two dimensions. To avoid solving quartic equations, the intersection is computed as a combination of two intersections. First, the ray is intersected with a cylinder to catch (almost) the

small radius curvature. Secondly, the ray is the intersected with an ellipsoid, with the curvatures matching that of the torus.

The first incarnation (Mirror_toroid.comp) the mirror curves outwards (a bump), but this incarnation (Mirror_toroid_pothole) curves inwards (a pothole).

Example: Mirror_toroid_pothole(radius=0.1, radius_o=1000, xwidth=5e-2, zdepth=2e-1,R0=1, coating="")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
zdepth	m	Length of mirror.	0.1
xwidth	m	Width of mirror.	0.01
radius	m	Curvature radius	
radius_o	m	Curvature radius, outwards	
R0	1	Reflectivity of mirror.	0
coating	str	Datafile containing either mirror material constants or reflectivity numbers.	""

Links

- Component source code found in file `Mirror_toroid_pothole.comp`.

12.8. Contributed monitors and detectors

12.9. The `Detector_pn` McXtrace Component

Version: McXtrace 1.2

Block of a attenuating material

Identification

- **Author:** Maria Thomsen (mariath@fys.ku.dk)
- **Origin:** NBI, KU
- **Date:** Jan 24, 2011

Description

A scintillator detector model taking photoabsorption efficiency into account. As such it consitututes a more physical version of the PSD_monitor. Only direct absorption is taken into account.

Example: Detector_pn(restore_xray=restore_flag,filename="detector_Si", material_datafile="Si.txt", xwidth=1e-2, yheight=1e-2,zdepth=1e-5)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
material_datafile	str	File where the material parameters for the scintillator may be found. Format is similar to what may be found off the NIST website.	"Be.txt"
nx	m	Number of pixel columns.	90
ny	m	Number of pixel rows.	90
filename	str	Name of file in which to store the detector image.	""
restore_xray		If set, the monitor does not influence the xray state.	0
xwidth	m	Width of block.	1e-2
yheight	m	Height of block.	1e-2
zdepth	m	Thickness of block.	1e-6

Links

- Component source code found in file `Detector_pn.comp`.
- material datafile obtained from http://physics.nist.gov/cgi-bin/ffast/ffast.pl

12.10. The `PSD_monitor_rad` McXtrace Component

Position-sensitive monitor with radially averaging.

Identification

- **Author:** Henrich Frielinghaus
- **Origin:** FZ-Juelich/FRJ-2/IFF/KWS-2
- **Date:** Sept 2004

Description

Radial monitor that allows for radial averaging. Comment: The intensity is given as two files: 1) a radial sum 2) a radial average (i.e. intensity per area)
Example: PSD_monitor_rad(rmax=0.2, nr=100, filename="Output.psd", filename_av="Output_av.psd")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
nr	1	Number of concentric circles	100
filename	text	Name of file in which to store the detector image	0
filename_av	text	Name of file in which to store the averaged detector image	0
rmax	m	Outer radius of detector	0.2

Links

- Component source code found in file `PSD_monitor_rad.comp`.

12.11. The `SAXSQMonitor` McXtrace Component

Release: McXtrace 1.0

A circular detector measuring the radial average of intensity as a function of the momentum transform in the sample.

Identification

- **Author:** Martin Cramer Pedersen (mcpe@nbi.dk)
- **Origin:** KU-Science
- **Date:** May 2, 2012

Description

A circular detector measuring the radial average of intensity as a function of the momentum transform in the sample. The q-range is set up to qMax = 4 * PI * sin(TwoThetaMax / 2.0) / LambdaMin;

Example: SAXSQMonitor(RadiusDetector = 0.1, DistanceFromSample = 0.5, LambdaMin = 1, Lambda0 = 1.54, NumberOfBins = 2000) Example: SAXSQMonitor(RadiusDetector = 0.1, qMax = 5, Lambda0 = 1.54, NumberOfBins = 2000)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
RFilename	str	File used for storing I(r).	"RDetector"
qFilename	str	File used for storing I(q).	"QDetector"
NumberOfBins	1	Number of bins in the r (and q).	100
restore_xray		If set to 1, the component restores the original x-ray.	0
RadiusDetector	m	Radius of the detector (in the xy-plane).	
DistanceFromSample		Distance from the sample to this component.	
LambdaMin	Å	Max sensitivity in lambda - used to compute the highest possible value of momentum transfer, q.	1.0
Lambda0	Å	If given, the momentum transfers of all rays are computed from this value. Otherwise, instrumental effects are negated Lambda0=2PI/k.	0.0
qMax	Å-1	Max momentum for the Q-monitor. use either qMax or LambdaMin.	0

Links

- Component source code found in file `SAXSQMonitor.comp`.

12.12. Contributed SAXS (small-angle X-ray scattering) samples

12.13. The `SAXSCurve` McXtrace Component

A component mimicking the scattering from a given I(q)-curve by using linear interpolation between the given points.

Identification

- **Author:** Martin Cramer Pedersen (mcpe@nbi.dk)
- **Origin:** KU-Science
- **Date:** May 2, 2012

Description

A box-shaped component simulating the scattering from any given I(q)-curve. The component uses linear interpolation to generate the points necessary to compute the scattering of any given photon.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
DeltaRho	cm/Å^3	Excess scattering length density of the particles.	1.0e-14
Volume	Å^3	Volume of the particles.	10000.0
Concentration	mM	Concentration of sample.	0.01
AbsorptionCrosssection	1/m	Absorption cross section of the sample.	0.0
xwidth	m	Dimension of component in the x-direction.	
yheight	m	Dimension of component in the y-direction.	
zdepth	m	Dimension of component in the z-direction.	
SampleToDetectorDistance	m	Distance from sample to detector (for focusing the scattered x-rays).	
DetectorRadius	m	Radius of the detector (for focusing the scattered x-rays).	
FileWithCurve	str	Datafile with the given I(q).	"Curve.dat"

Links

- Component source code found in file `SAXSCurve.comp`.

12.14. The `SAXSSpheres` McXtrace Component

A sample of monodisperse spherical particles in solution.

Identification

- **Author:** Martin Cramer Pedersen (mcpe@nbi.dk)
- **Origin:** KU-Science
- **Date:** May 2, 2012

Description

A simple component simulating the scattering from a box-shaped, thin solution of monodisperse, spherical particles.

Example: SAXSSpheres(xwidth = 0.01, yheight = 0.01, zdepth = 0.01, R = 50.0, SampleToDetectorDistance = 0.5, DetectorRadius = 0.1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
R	Å	Radius of the spherical particles.	100.0
Concentration	mM	Concentration of sample.	0.01
DeltaRho	cm/Å^3	Excess scattering length density of the particles.	1.0e-14
AbsorptionCrosssection	1/m	Absorption cross section of the sample.	0.0
xwidth	m	Dimension of component in the x-direction.	
yheight	m	Dimension of component in the y-direction.	
zdepth	m	Dimension of component in the z-direction.	
SampleToDetectorDistance	m	Distance from sample to detector (for focusing the scattered x-rays).	
DetectorRadius	m	Radius of the detector (for focusing the scattered x-rays).	

Links

- Component source code found in file `SAXSSpheres.comp`.

12.15. The `SAXSCylinders` McXtrace Component

A sample of monodisperse cylindrical particles in solution.

Identification

- **Author:** Martin Cramer Pedersen (mcpe@nbi.dk)
- **Origin:** KU-Science
- **Date:** May 2, 2012

Description

A component simulating the scattering from a box-shaped, thin solution of monodisperse, cylindrical particles.

Example: SAXSCylinders(xwidth = 0.01, yheight = 0.01, zdepth = 0.01, SampleToDetectorDistance = 0.48, DetectorRadius = 0.1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
R	Å	Semiaxis of the cross section of the cylinder.	40.0
Height	Å	Height of the cylinder.	100.0
Concentration	mM	Concentration of sample.	0.01
DeltaRho	$cm/Å^3$	Excess scattering length density of the particles.	1.0e-14
AbsorptionCrosssection	1/m	Absorption cross section of the sample.	0.0
xwidth	m	Dimension of component in the x-direction.	
yheight	m	Dimension of component in the y-direction.	
zdepth	m	Dimension of component in the z-direction.	
SampleToDetectorDistance	m	Distance from sample to detector (for focusing the scattered x-rays).	
DetectorRadius	m	Radius of the detector (for focusing the scattered x-rays).	

Links

- Component source code found in file `SAXSCylinders.comp`.

12.16. The `SAXSEllipticCylinders` McXtrace Component

A sample of monodisperse cylindrical particles with elliptic cross section in solution.

Identification

- **Author:** Martin Cramer Pedersen (mcpe@nbi.dk)
- **Origin:** KU-Science
- **Date:** May 2, 2012

Description

A component simulating the scattering from a box-shaped, thin solution of monodisperse, cylindrical particles with elliptic cross section.

Example: SAXSEllipticCylinders(xwidth = 0.01, yheight = 0.01, zdepth = 0.01, SampleToDetectorDistance = 0.48, DetectorRadius = 0.1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
R1	Å	First semiaxis of the cross section of the elliptic cylinder.	20.0
R2	Å	Second semiaxis of the cross section of the elliptic cylinder.	40.0
Height	Å	Height of the cylinder.	100.0
Concentration	mM	Concentration of sample.	0.01
DeltaRho	cm/Å^3	Excess scattering length density of the particles.	1.0e-14
AbsorptionCrosssection	1/m	Absorption cross section of the sample.	0.0
xwidth	m	Dimension of component in the x-direction.	
yheight	m	Dimension of component in the y-direction.	
zdepth	m	Dimension of component in the z-direction.	
SampleToDetectorDistance	m	Distance from sample to detector (for focusing the scattered x-rays).	
DetectorRadius	m	Radius of the detector (for focusing the scattered x-rays).	

Links

- Component source code found in file `SAXSEllipticCylinders.comp`.

12.17. The `SAXSShells` McXtrace Component

Release: McXtrace 1.0

A sample of monodisperse shell-like particles in solution.

Identification

- **Author:** Martin Cramer Pedersen (mcpe@nbi.dk)
- **Origin:** KU-Science
- **Date:** May 11, 2012

Description

A simple component simulating the scattering from a box-shaped, thin solution of monodisperse, shell-like particles.

Example: Sample1 = SAXSShells(xwidth = 0.01, yheight = 0.01, zdepth = 0.01, SampleToDetectorDistance = 0.5, DetectorRadius = 0.1, R = 50.0, Thickness = 20.0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
R	Å	Average radius of the particles.	100.0
Thickness	Å	Thickness of the shell - so that the outer radius is R + Thickness and the inner is R - Thickness.	5.0
Concentration	mM	Concentration of sample.	0.01
DeltaRho	cm/Å^3	Excess scattering length density of the particles.	1.0e-14
AbsorptionCrosssection	1/m	Absorption cross section of the sample.	0.0
xwidth	m	Dimension of component in the x-direction.	
yheight	m	Dimension of component in the y-direction.	
zdepth	m	Dimension of component in the z-direction.	
SampleToDetectorDistance	m	Distance from sample to detector (for focusing the scattered x-rays).	
DetectorRadius	m	Radius of the detector (for focusing the scattered x-rays).	

Links

- Component source code found in file `SAXSShells.comp`.

12.18. The `SAXSLiposomes` McXtrace Component

A sample of polydisperse liposomes in solution (water).

Identification

- **Author:** Martin Cramer Pedersen (mcpe@nbi.dk)
- **Origin:** KU-Science

- **Date:** May 2, 2012

Description

A component simulating the scattering from a box-shaped, thin solution (water) of liposomes described by a pentuple-shell model.

Example: SAXSLiposomes(xwidth = 0.01, yheight = 0.01, zdepth = 0.01, SampleToDetectorDistance = 0.48, DetectorRadius = 0.1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
Radius	Å	Average thickness of the liposomes.	800.0
Thickness	Å	Thickness of the bilayer.	38.89
SigmaRadius		Relative Gaussian deviation of the radius in the distribution of liposomes.	0.20
nRadius		Number of bins in Radius for polydisperse distribution.	100
VolumeOfHeadgroup	$Å^3$	Volume of one lipid headgroup - default is POPC.	319.0
VolumeOfCH2Tail	$Å^3$	Volume of the CH2-chains of one lipid - default is POPC.	818.8
VolumeOfCH3Tail	$Å^3$	Volume of the CH3-tails of one lipid - default is POPC.	108.6
ScatteringLengthOfHeadgroup		Scattering length of one lipid headgroup - default is POPC.	4.62E-11
ScatteringLengthOfCH2Tail		Scattering length of the CH2-chains of one lipid - default is POPC.	6.71E-11
ScatteringLengthOfCH3Tail		Scattering length of the CH3-tails of one lipid - default is POPC.	5.08E-12
Concentration	mM	Concentration of sample.	0.01
AbsorptionCrosssection	1/m	Absorption cross section of the sample.	0.0
xwidth	m	Dimension of component in the x-direction.	
yheight	m	Dimension of component in the y-direction.	
zdepth	m	Dimension of component in the z-direction.	
SampleToDetectorDistance		Distance from sample to detector (for focusing the scattered x-rays).	

Name	Unit	Description	Default
DetectorRadius	m	Radius of the detector (for focusing the scattered x-rays).	

Links

- Component source code found in file `SAXSLiposomes.comp`.

12.19. The `SAXSNanodiscs` McXtrace Component

A sample of monodisperse phospholipid bilayer nanodiscs in solution (water).

Identification

- **Author:** Martin Cramer Pedersen (mcpe@nbi.dk)
- **Origin:** KU-Science
- **Date:** May 2, 2012

Description

A component simulating the scattering from a box-shaped, thin solution (water) of monodisperse phospholipid bilayer nanodiscs.

Example: SAXSNanodiscs(xwidth = 0.01, yheight = 0.01, zdepth = 0.01, SampleToDetectorDistance = 0.48, DetectorRadius = 0.1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
AxisRatio		Axis ratio of the bilayer patch.	1.4
NumberOfLipids		Number of lipids per nanodisc.	130.0
AreaPerLipidHeadgroup	$Å^2$	Area per lipid headgroup - default is POPC.	65.0
HeightOfMSP	Å	Height of the belt protein - default is MSP1D1.	24.0
VolumeOfOneMSP	$Å^3$	Volume of one belt protein - default is MSP1D1.	26296.5
VolumeOfHeadgroup	$Å^3$	Volume of one lipid headgroup - default is POPC.	319.0
VolumeOfCH2Tail	$Å^3$	Volume of the CH2-chains of one lipid - default is POPC.	818.8

Name	Unit	Description	Default
VolumeOfCH3Tail	Å^3	Volume of the CH3-tails of one lipid - default is POPC.	108.6
ScatteringLengthOfOneMSP	m	Scattering length of one belt protein - default is MSP1D1.	3.34E-9
ScatteringLengthOfHeadgroup	m	Scattering length of one lipid headgroup - default is POPC.	4.62E-11
ScatteringLengthOfCH2Tail	m	Scattering length of the CH2-chains of one lipid - default is POPC.	6.71E-11
ScatteringLengthOfCH3Tail	m	Scattering length of the CH3-tails of one lipid - default is POPC.	5.08E-12
Roughness		Factor used to smear the interfaces of the nanodisc.	3.5
Concentration	mM	Concentration of sample.	0.01
AbsorptionCrosssection	1/m	Absorption cross section of the sample.	0.0
xwidth	m	Dimension of component in the x-direction.	
yheight	m	Dimension of component in the y-direction.	
zdepth	m	Dimension of component in the z-direction.	
SampleToDetectorDistance	m	Distance from sample to detector (for focusing the scattered x-rays).	
DetectorRadius	m	Radius of the detector (for focusing the scattered x-rays).	

Links

- Component source code found in file `SAXSNanodiscs.comp`.

12.20. The `SAXSNanodiscsFast` McXtrace Component

Release: McXtrace 1.0

A sample of monodisperse phospholipid bilayer nanodiscs in solution (water).

Identification

- **Author:** Martin Cramer Pedersen (mcpe@nbi.dk)
- **Origin:** KU-Science
- **Date:** May 2, 2012

Description

A component very similar to SAXSNanodiscs.comp - however, the scattering profile is only computed once, and linear interpolation is then used to simulate the instrument.

Example: SAXSNanodiscsFast(xwidth = 0.01, yheight = 0.01, zdepth = 0.01, SampleToDetectorDistance = 0.48, DetectorRadius = 0.1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
AxisRatio		Axis ratio of the bilayer patch.	1.4
NumberOfLipids		Number of lipids per nanodisc.	130.0
AreaPerLipidHeadgroup	$Å^2$	Area per lipid headgroup - default is POPC.	65.0
HeightOfMSP	Å	Height of the belt protein - default is MSP1D1.	24.0
VolumeOfOneMSP	$Å^3$	Volume of one belt protein - default is MSP1D1.	26296.5
VolumeOfHeadgroup	$Å^3$	Volume of one lipid headgroup - default is POPC.	319.0
VolumeOfCH2Tail	$Å^3$	Volume of the CH2-chains of one lipid - default is POPC.	818.8
VolumeOfCH3Tail	$Å^3$	Volume of the CH3-tails of one lipid - default is POPC.	108.6
ScatteringLengthOfOneMSP	m	Scattering length of one belt protein - default is MSP1D1.	3.34E-9
ScatteringLengthOfHeadgroup	m	Scattering length of one lipid headgroup - default is POPC.	4.62E-11
ScatteringLengthOfCH2Tail	m	Scattering length of the CH2-chains of one lipid - default is POPC.	6.71E-11
ScatteringLengthOfCH3Tail	m	Scattering length of the CH3-tails of one lipid - default is POPC.	5.08E-12
Roughness		Factor used to smear the interfaces of the nanodisc.	3.5
Concentration	mM	Concentration of sample.	0.01
AbsorptionCrosssection	1/m	Absorption cross section of the sample.	0.0
xwidth	m	Dimension of component in the x-direction.	
yheight	m	Dimension of component in the y-direction.	
zdepth	m	Dimension of component in the z-direction.	

Name	Unit	Description	Default
SampleToDetectorDistance	m	Distance from sample to detector (for focusing the scattered x-rays).	
DetectorRadius	m	Radius of the detector (for focusing the scattered x-rays).	
qMin	$Å^{-1}$	Lowest q-value, for which a point is generated in the scattering profile	0.001
qMax	$Å^{-1}$	Highest q-value, for which a point is generated in the scattering profile	1.0
NumberOfQBins		Number of points generated in inital scattering profile.	200

Links

- Component source code found in file `SAXSNanodiscsFast.comp`.

12.21. The `SAXSNanodiscsWithTags` McXtrace Component

A sample of monodisperse phospholipid bilayer nanodiscs in solution (water) - with histidine tag still on the belt proteins.

Identification

- **Author:** Martin Cramer Pedersen (mcpe@nbi.dk)
- **Origin:** KU-Science
- **Date:** May 2, 2012

Description

A component simulating the scattering from a box-shaped, thin solution (water) of monodisperse phospholipid bilayer nanodiscs - with histidine tags still on the belt proteins.

Example: SAXSNanodiscsWithTags(xwidth = 0.01, yheight = 0.01, zdepth = 0.01, SampleToDetectorDistance = 0.48, DetectorRadius = 0.1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
AxisRatio		Axis ratio of the bilayer patch.	1.4
NumberOfLipids		Number of lipids per nanodisc.	130.0

Name	Unit	Description	Default
AreaPerLipidHeadgroup	Å^2	Area per lipid headgroup - default is POPC.	65.0
HeightOfMSP	Å	Height of the belt protein - default is MSP1D1.	24.0
RadiusOfGyrationForHisTag	Å	Radius of gyration for the his-tag.	10.0
VolumeOfOneMSP	Å^3	Volume of one belt protein - default is MSP1D1.	26296.5
VolumeOfHeadgroup	Å^3	Volume of one lipid headgroup - default is POPC.	319.0
VolumeOfCH2Tail	Å^3	Volume of the CH2-chains of one lipid - default is POPC.	818.8
VolumeOfCH3Tail	Å^3	Volume of the CH3-tails of one lipid - default is POPC.	108.6
VolumeOfOneHisTag	Å^3	Volume of one his-tag.	2987.3
ScatteringLengthOfOneMSP	m	Scattering length of one belt protein - default is MSP1D1.	3.34E-9
ScatteringLengthOfHeadgroup	m	Scattering length of one lipid headgroup - default is POPC.	4.62E-11
ScatteringLengthOfCH2Tail	m	Scattering length of the CH2-chains of one lipid - default is POPC.	6.71E-11
ScatteringLengthOfCH3Tail	m	Scattering length of the CH3-tails of one lipid - default is POPC.	5.08E-12
ScatteringLengthOfOneHisTag	m	Scattering length of one his-tag.	3.89E-10
Roughness		Factor used to smear the interfaces of the nanodisc.	3.5
Concentration	mM	Concentration of sample.	0.01
AbsorptionCrosssection	1/m	Absorption cross section of the sample.	0.0
xwidth	m	Dimension of component in the x-direction.	
yheight	m	Dimension of component in the y-direction.	
zdepth	m	Dimension of component in the z-direction.	
SampleToDetectorDistance		Distance from sample to detector (for focusing the scattered x-rays).	
DetectorRadius	m	Radius of the detector (for focusing the scattered x-rays).	

Links

- Component source code found in file `SAXSNanodiscsWithTags.comp`.

12.22. The `SAXSNanodiscsWithTagsFast` McXtrace Component

Release: McXtrace 1.0

A sample of monodisperse phospholipid bilayer nanodiscs in solution (water) - with histidine tag still on the belt proteins.

Identification

- **Author:** Martin Cramer Pedersen (mcpe@nbi.dk)
- **Origin:** KU-Science
- **Date:** May 2, 2012

Description

A component very similar to SAXSNanodiscsWithTags.comp - however, the scattering profile is only computed once, and linear interpolation is then used to simulate the instrument.

Example: SAXSNanodiscsWithTagsFast(xwidth = 0.01, yheight = 0.01, zdepth = 0.01, SampleToDetectorDistance = 0.48, DetectorRadius = 0.1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
AxisRatio		Axis ratio of the bilayer patch.	1.4
NumberOfLipids		Number of lipids per nanodisc.	130.0
AreaPerLipidHeadgroup	Å^2	Area per lipid headgroup - default is POPC.	65.0
HeightOfMSP	Å	Height of the belt protein - default is MSP1D1.	24.0
RadiusOfGyrationForHisTag	Å	Radius of gyration for the his-tag.	12.7
VolumeOfOneMSP	Å^3	Volume of one belt protein - default is MSP1D1.	26296.5
VolumeOfHeadgroup	Å^3	Volume of one lipid headgroup - default is POPC.	319.0
VolumeOfCH2Tail	Å^3	Volume of the CH2-chains of one lipid - default is POPC.	818.8
VolumeOfCH3Tail	Å^3	Volume of the CH3-tails of one lipid - default is POPC.	108.6
VolumeOfOneHisTag	Å^3	Volume of one his-tag.	2987.3
ScatteringLengthOfOneMSP	m	Scattering length of one belt protein - default is MSP1D1.	3.34E-9

Name	Unit	Description	Default
ScatteringLengthOfHeadgroup	m	Scattering length of one lipid headgroup - default is POPC.	4.62E-11
ScatteringLengthOfCH2Tail	m	Scattering length of the CH2-chains of one lipid - default is POPC.	6.71E-11
ScatteringLengthOfCH3Tail	m	Scattering length of the CH3-tails of one lipid - default is POPC.	5.08E-12
ScatteringLengthOfOneHisTag	m	Scattering length of one histidine tag.	3.89E-10
Roughness		Factor used to smear the interfaces of the nanodisc.	3.5
Concentration	mM	Concentration of sample.	0.01
AbsorptionCrosssection	1/m	Absorption cross section of the sample.	0.0
xwidth	m	Dimension of component in the x-direction.	
yheight	m	Dimension of component in the y-direction.	
zdepth	m	Dimension of component in the z-direction.	
SampleToDetectorDistance		Distance from sample to detector (for focusing the scattered x-rays).	
DetectorRadius	m	Radius of the detector (for focusing the scattered x-rays).	
qMin	$Å^{-1}$	Lowest q-value, for which a point is generated in the scattering profile	0.001
qMax	$Å^{-1}$	Highest q-value, for which a point is generated in the scattering profile	1.0
NumberOfQBins		Number of points generated in inital scattering profile.	200

Links

- Component source code found in file `SAXSNanodiscsWithTagsFast.comp`.

12.23. The `SAXSPDB` McXtrace Component

Release: McXtrace 1.0

A sample describing a thin solution of proteins. This components must be compiled with the -lgsl and -lgslcblas flags (and possibly linked to the appropriate libraries).

Identification

- **Author:** Martin Cramer Pedersen (mcpe@nbi.dk) and Søren Kynde (kynde@nbi.dk)

- **Origin:** KU-Science
- **Date:** May 2, 2012

Description

This components expands the formfactor amplitude of the protein on spherical harmonics and computes the scattering profile using these. The expansion is done on amino-acid level and does not take hydration layer into account. The component must have a valid .pdb-file as an argument.

This component is very slow. You should rather use the SAXSPDBFast sample component.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
RhoSolvent	Å	Scattering length density of the buffer.	9.4e-14
Concentration	mM	Concentration of sample.	0.01
AbsorptionCrosssection	1/m	Absorption cross section of the sample.	0.0
xwidth	m	Dimension of component in the x-direction.	
yheight	m	Dimension of component in the y-direction.	
zdepth	m	Dimension of component in the z-direction.	
SampleToDetectorDistance	m	Distance from sample to detector (for focusing the scattered x-rays).	
DetectorRadius	m	Radius of the detector (for focusing the scattered x-rays).	
PDBFilepath		Path to the file describing the high resolution structure of the protein.	"PDBfile.pdb"

Links

- Component source code found in file `SAXSPDB.comp`.

12.24. The `SAXSPDBFast` McXtrace Component

Release: McXtrace 1.0

A sample describing a thin solution of proteins using linear interpolation to increase computational speed. This components must be compiled with the -lgsl and -lgslcblas flags (and possibly linked to the appropriate libraries).

Identification

- **Author:** Martin Cramer Pedersen (mcpe@nbi.dk) and Søren Kynde (kynde@nbi.dk)
- **Origin:** KU-Science
- **Date:** May 2, 2012

Description

This components expands the formfactor amplitude of the protein on spherical harmonics and computes the scattering profile using these. The expansion is done on amino-acid level and does not take hydration layer into account. The component must have a valid .pdb-file as an argument.

This is fast implementation of the SAXSPDB sample component.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
RhoSolvent	Å	Scattering length density of the buffer.	9.4e-14
Concentration	mM	Concentration of sample.	0.01
AbsorptionCrosssection	1/m	Absorption cross section of the sample.	0.0
xwidth	m	Dimension of component in the x-direction.	
yheight	m	Dimension of component in the y-direction.	
zdepth	m	Dimension of component in the z-direction.	
SampleToDetectorDistance		Distance from sample to detector (for focusing the scattered x-rays).	
DetectorRadius	m	Radius of the detector (for focusing the scattered x-rays).	
qMin	$Å^{-1}$	Lowest q-value, for which a point is generated in the scattering profile	0.001
qMax	$Å^{-1}$	Highest q-value, for which a point is generated in the scattering profile	0.5
NumberOfQBins		Number of points generated in inital scattering profile.	200
PDBFilepath		Path to the file describing the high resolution structure of the protein.	”PDBfile.pdb”

Links

- Component source code found in file `SAXSPDBFast.comp`.

13. Obsolete components

Components in this chapter have been superseded, typically by a more general or more robust replacement, and are kept solely so that existing instrument files referencing them continue to compile and run. **Do not use these in new instrument development.** Where a direct replacement exists it is noted below; run `mxdoc` on the component name for the authoritative, current pointer.

- `Perfect_crystal` – superseded by the `contrib` chapter's `Bragg_crystal_simple`/`Bragg_crysta` family.
- `Lens_parab_rough`, `Lens_parab_Cyl_rough` – early, roughness-aware refractive lens models, superseded by the current `Lens_parab`/`Lens_parab_Cyl` components (see the Optics chapter).
- `Lens_Kinoform` – an early kinoform refractive lens model.

This chapter is intentionally text-only: none of the components above ever had a full manual entry of their own (`Perfect_crystal`, `Lens_Kinoform` and the `_rough` lens variants only ever carried short documentation stubs), so there is nothing further to reproduce here beyond the summary above. Run `mxdoc` on a component name for its full, current documentation where one exists.

14. AstroX components: X-ray optics for astronomical instrumentation

The `astrox` category collects components specific to the AstroX branch of McXtrace, modelling grazing-incidence X-ray optics of the kind used in astronomical X-ray telescopes (nested Wolter-type mirror shells, pore optics, and micropore/microchannel plate optics), together with a dedicated extended source component suited to astronomical (rather than laboratory or synchrotron) source geometries.

Each optical element is generally available in three variants, reflecting the underlying mirror-shell geometry:

- `_c`: conical approximation,
- `_p`: parabolic/hyperbolic (true Wolter-I) profile,
- `_h`: hyperbolic profile (second Wolter-I reflection).

14.1. Sources

14.2. The `Source_extended` McXtrace Component

Release: McXtrace 1.4

A plane source emitting x-rays to emulate a distant extended source, such as a nebula or galaxy

Identification

- **Author:** Arne 'S Jegers
- **Origin:** Technical University of Denmark
- **Date:** May 6, 2019

Description

A rectangular x-ray source that samples ray intensity from an image, and deflects the emitted ray to reflect having been emitted from the sampled part of the extended source. Rays that are sampled from the same point on the image are collimated, but can be emitted from anywhere on the rectangular source. The image should be provided as a 2D-ascii table, whose header includes the following entries:

w_pixels and h_pixels: The width and height of the image in pixels x_ref and y_ref: x and y reference values of the FITS image r_max: The maximum distance from the point (x_ref, y_ref) to any corner of the image iCD11, iCD12, iCD21 and iCD22: The values of the FITS image's CD matrix

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
spectrum_file	str	File with the energy spectrum to sample from (Table format)	NULL
yheight	m	Height of rectangle in (x,y,0) plane where x-rays are generated.	0
xwidth	m	Width of rectangle in (x,y,0) plane where x-rays are generated.	0
dist	m	Currently not used by the component	0
E0	keV	Mean energy of xrays.	0
dE	keV	Energy half spread of x-rays (flat or gaussian sigma).	0
lambda0	Å	Mean wavelength of x-rays.	0
dlambda	Å	Wavelength half spread of x-rays.	0
flux	pht/s	total flux radiated from the source	0
gauss	1	Gaussian (1) or Flat (0) energy/wavelength distribution	0
incoherent		Source is fully incoherent	1
phase	rad	Set phase to something given.	0
image_path	string	Path to file containing the heat map of the source	""

Links

- Component source code found in file `Source_extended.comp`.

14.3. Mirror shells

14.4. The `Shell_c` McXtrace Component

Single conical shell as part of a Wolter optic.

Identification

- **Author:** Erik B Knudsen and Desiree D. M. Ferreira

- **Origin:** DTU Physics, DTU Space
- **Date:** Feb. 2016

Description

A single shell is simulated. The top and bottom are curved cylindrically azimuthally, whereas they are straight sagitally. The primary parameter specifies whether this is a primary or secondary mirror. The azimuthal curvature is defined by the parameter radius. This refers to the top plate of the shell. I.e the top and bottom plates have radius of curvature <radius> and <radius-yheight> respectively.

To intersect the Wolter I plates we take advatage of the azimuthal symmetry and only consider the radial component of the photon's wavevector.

Example: Shell_c(radius_m=0.535532,Z0=12,yheight=1e-2,length=0.5,primary=0, R_d=1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_m	m	Ring radius of the upper (reflecting) plate of the shell at the optic centre.	
Z0	m	Distance between optics centre plane and focal spot (essentially focal length).	
yheight	m	Height of the shell.	
gap	m	Gap between the plate and the intersection plane with the hyperbolic section. (currently ignored)	0
chamferwidth	m	Width of side walls.	0
length	m	Length along the optical axis	0
mirror_reflec		Data file containing reflectivities of the reflector surface (TOP).	""
bottom_reflec		Data file containing reflectivities of the bottom surface (BOTTOM).	""
R_d		Default reflectivity value to use if no reflectivity file is given. Useful f.i. is one surface is reflecting and the others absorbing.	1
primary		If non-zero, the shell is considered a primary reflector, and extends towards negative z. I.e. the entry plane is behind the z=0-plane. If zero, the shell is considered secondary and extends from the z=0-plane and towards positive z.	1

Name	Unit	Description	Default
dalpha	deg	Offset to the alpha angle computed from the focal length. Useful for targeting the modified conical geometry (currently ignored).	0
waviness	rad	Waviness of the shell reflecting surface. The slope error is assumed to be uniformly distributed in the interval [-waviness:waviness].	0
longw		If non-zero, waviness is 1D and along the shell axis.	0

Links

- Component source code found in file `Shell_c.comp`.

14.5. The `Shell_p` McXtrace Component

Single parabolic shell as part of a Wolter optic.

Identification

- **Author:** Erik B Knudsen and Desiree D. M. Ferreira
- **Origin:** DTU Physics, DTU Space
- **Date:** Feb. 2016

Description

A single shell is simulated. The top and bottom are curved cylindrically azimuthally. The sagital profile is defined by a parabola, which passes through the radii raidus_m at z=0, and radius_p at zentry (<0).

To intersect the Wolter I plates we take advatage of the azimuthal symmetry and only consider the radial component of the photon's wavevector.

Imperfect mirrors may be modelled using one of 4 models. In all cases the surface normal of the mirror at the ideal mirror intersection point is perturbed before the exit vector is computed. 1. Longitudinal 1D. A perturbation angle is chosen from a uniform distribution with width waviness. 2. Isotropic 2D. The surface normal is perturbed by choosing an angle on a disc with radius waviness 3. Externally measured/computed data. We interpolate in a data-file consisting of blocks of dtheta/theta with 1 block per energy. dtheta is a sampled angle offset from the nominal Fresnel grazing angle theta. 4. Double gaussian. dtheta is chosen from one of two gaussian distributions. Either

specular or off-specular, where the widths (sigmas) are given by the tables in the file "wave_file". If the off-specular case the behaviour is similar to 2D uniform case.

In the case of 3, the format of the data file should be: #e_min=0.1 #e_max=15 #e_step=0.01 #theta_min=0.01 #theta_max=1.5 #theta_step=0.01 #dtheta_min=-0.02 #dtheta_max=0.02 #dtheta_step=0.001

```
1.0  0.9  0.8  0.75  ...
0.99 0.89 0.79 0.749 ...
```

... #block 2 (energy data point 2)

```
1.0  0.9  0.8  0.75  ...
0.99 0.89 0.79 0.749 ...
```

...

I.e. one 2D data block per energy data point where rows represent the steps in nominal incident angle, and columns represent the sampled granularity of the off-specular scattering.

Example: Shell_p(radius_p=0.535532, radius_m=0.533113, zdepth=0.5, Z0=12, yheight=1e-2, R_d=1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_p	m	Ring radius of the upper (reflecting) plate of the shell at the edge furthest away from the focal point.	
radius_m	m	Ring radius of the upper (reflecting) plate of the shell at the intersection with the hyperbolic section.	
Z0	m	Distance between optics centre plane and focal spot (essentially focal length).	
yheight	m	Height of the shell.	
chamferwidth	m	Width of side walls.	0
gap	m	Gap between the plate and the intersection plane with the hyperbolic section.	0
zdepth	m	Currently not used by the component	0
mirror_reflec		Data file containing reflectivities of the reflector surface (TOP).	""
bottom_reflec		Data file containing reflectivities of the bottom surface (BOTTOM).	""

Name	Unit	Description	Default
wave_file	str	Data file with the waviness distribution, used when wave_model is non-zero	""
R_d		Default reflectivity value to use if no reflectivity file is given. Useful f.i. is one surface is reflecting and the others absorbing.	1
wave_model		Flag to choose waviness model. 1. longitudinal uniform, 2. 2D-uniform, 3. lorentzian sagittal, 4. double gaussian sagittal. See above for details.	0
waviness	rad	Waviness of the pore reflecting surface. The slope error is assumed to be uniformly distributed in the interval "[-waviness:waviness]".	0
verbose		If !=0 output extra info during simulation.	0

Links

- Component source code found in file `Shell_p.comp`.

14.6. The `Shell_h` McXtrace Component

Single Pore as part of the Silicon Pore Optics (SPO) as envisioned for the ATHENA+ space telescope.

Identification

- **Author:** Erik B Knudsen and Desiree D. M. Ferreira
- **Origin:** DTU Physics, DTU Space
- **Date:** Feb. 2016

Description

A single shell is simulated. The top and bottom are curved cylindrically azimuthally, and according to the Wolter I optic lengthwise (sagitally). This is the hyperbolic part. The azimuthal curvature is defined by the radius parameters.

To intersect the Wolter I plates we take advantage of the azimuthal symmetry and only consider the radial component of the photon's wavevector.

Imperfect mirrors may be modelled using one of 4 models. In all cases the surface normal of the mirror at the ideal mirror intersection point is perturbed before the exit

vector is computed. 1. Longitudinal 1D. A perturbation angle is chosen from a uniform distribution with width waviness. 2. Isotropic 2D. The surface normal is perturbed by choosing an angle on a disc with radius waviness 3. Externally measured/computed data. We interpolate in a data-file consisting of blocks of dtheta/theta with 1 block per energy. dtheta is a sampled angle offset from the nominal Fresnel grazing angle theta. 4. Double gaussian. dtheta is chosen from one of two gaussian distributions. Either specular or off-specular, where the widths (sigmas) are given by the tables in the file "wave_file". If the off-specular case the behaviour is similar to 2D uniform case.

In the case of 3, the format of the data file should be: #e_min=0.1 #e_max=15 #e_step=0.01 #theta_min=0.01 #theta_max=1.5 #theta_step=0.01 #dtheta_min=-0.02 #dtheta_max=0.02 #dtheta_step=0.001

```
1.0  0.9  0.8  0.75  ...
0.99 0.89 0.79 0.749 ...
```

... #block 2 (energy data point 2)

```
1.0  0.9  0.8  0.75  ...
0.99 0.89 0.79 0.749 ...
```

...

I.e. one 2D data block per energy data point where rows represent the steps in nominal incident angle, and columns represent the sampled granularity of the off-specular scattering.

Example: Shell_h(radius_m=0.535532, radius_h=0.533113, zdepth=0.5, Z0=FL, yheight=1e-2, R_d=1)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_m	m	Ring radius of the upper (reflecting) plate of the pore at the intersection with the parabolic section.	
radius_h	m	Ring radius of the upper (reflecting) plate of the pore at the edge closest to the focal point.	
Z0	m	distance between intersection plane and the focal spot(essentially the focal length).	
yheight	m	Height of the pore. (Thus the inner radius is radius_{m,h}-yehight	
chamferwidth	m	Width of side walls.	0

Name	Unit	Description	Default
gap	m	gap between intersection with parabolic section and actual plate.	0
zdepth	m	Currently not used by the component	0
mirror_reflec		Data file containing reflectivities of the reflector surface (TOP).	""
bottom_reflec		Data file containing reflectivities of the bottom surface (BOTTOM).	""
wave_file	str	Data file with the waviness distribution, used when wave_model is non-zero	""
R_d		Default reflectivity value to use if no reflectivity file is given. Useful f.i. is one surface is reflecting and the others absorbing.	1
wave_model		Flag to choose waviness model. 1. longitudinal uniform, 2. 2D-uniform, 3. lorentzian sagittal, 4. double gaussian sagittal. See above for details.	0
waviness	rad	Waviness of the pore reflecting surface. The slope error is assumed to be uniformly distributed in the interval [-waviness:waviness].	0
verbose		If !=0 output extra info during simulation.	0

Links

- Component source code found in file `Shell_h.comp`.

14.7. Ring optics

14.8. The `Ring_c` McXtrace Component

Date: Feb. 2017 Version: 1.1 Release: McXtrace 1.2 Origin: DTU Physics, DTU Space
Stack of conical shells as part of a Wolter optic.

Identification

- **Author:** Erik B Knudsen and Desiree D. M. Ferreira
- **Origin:** DTU Physics, DTU Space
- **Date:** Feb. 2016

Description

A stack of conical shells is simulated. To intersect the Wolter I plates we take advatage of the azimuthal symmetry and only consider the radial component of the photon's wavevector.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
ring_nr	1	Ring number; selects the data block read from size_file	
radius_m	m	Ring radius of the upper (reflecting) plate of the shell at the optic centre.	
Z0	m	Distance between optics centre plane and focal spot (essentially focal length).	
xwidth	m	Currently not used by the component	
yheight	m	Height of the shell.	
gap	m	Gap between the plate and the intersection plane with the hyperbolic section. (currently ignored)	0
chamferwidth	m	Width of side walls.	0
length	m	Length along the optical axis	0
mirror_reflec		Data file containing reflectivities of the reflector surface (TOP).	""
bottom_reflec		Data file containing reflectivities of the bottom surface (BOTTOM).	""
R_d		Default reflectivity value to use if no reflectivity file is given. Useful f.i. is one surface is reflecting and the others absorbing.	1
primary		If non-zero, the shell is considered a primary reflector, and extends towards negative z. I.e. the entry plane is behind the z=0-plane. If zero, the shell is considered secondary and extends from the z=0-plane and towards positive z.	1
dalpha	deg	Offset to the alpha angle computed from the focal length. Useful for targeting the modified conical geometry (currently ignored).	0

Name	Unit	Description	Default
waviness	rad	Waviness of the shell reflecting surface. The slope error is assumed to be uniformly distributed in the interval [-waviness:waviness].	0
longw		If non-zero, waviness is 1D and along the shell axis.	0

Links

- Component source code found in file `Ring_c.comp`.

14.9. The `Ring_p` McXtrace Component

Date: Feb. 2017 Version: 1.1 Release: McXtrace 1.2 Origin: DTU Physics, DTU Space
Stack of conical shells as part of a Wolter optic. Parabolic version.

Identification

- **Author:** Erik B Knudsen and Desiree D. M. Ferreira
- **Origin:** DTU Physics, DTU Space
- **Date:** Feb. 2016

Description

A stack of conical shells is simulated. Parabolic version. To intersect the Wolter I plates we take advatage of the azimuthal symmetry and only consider the radial component of the photon's wavevector.

Example: Ring_p(pore_th=0, ring_nr=3, Z0=FL, yheight=0.605e-3, mirror_reflec="mirror_coating R_d=0, size_file="ref_design_breaks.txt")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
ring_nr	1	Ring number; selects the data block read from size_file	
pore_th	m	Thickness of the pore walls (plates)	
Z0	m	Distance between optics centre plane and focal spot (essentially focal length).	
yheight	m	Height of the pore.	

Name	Unit	Description	Default
chamferwidth	m	Width of side walls.	0
gap	m	Gap between the plate and the intersection plane with the hyperbolic section.	0
zdepth	m	Currently not used by the component	0
mirror_reflec		Data file containing reflectivities of the reflector surface (TOP).	""
bottom_reflec		Data file containing reflectivities of the bottom surface (BOTTOM).	""
size_file	str	Table file with the per-ring geometry, read with block number ring_nr	""
R_d		Default reflectivity value to use if no reflectivity file is given. Useful f.i. is one surface is reflecting and the others absorbing.	1
waviness	rad	Waviness of the pore reflecting surface. The slope error is assumed to be uniformly distributed in the interval [-waviness:waviness].	0
longw		If non-zero, waviness is 1D and along the pore axis.	0

Links

- Component source code found in file `Ring_p.comp`.

14.10. The `Ring_h` McXtrace Component

Date: Feb. 2017 Version: 1.1 Release: McXtrace 1.2 Origin: DTU Physics, DTU Space
Stack of conical shells as part of a Wolter optic. Hyperbolic version.

Identification

- **Author:** Erik B Knudsen and Desiree D. M. Ferreira
- **Origin:** DTU Physics, DTU Space
- **Date:** Feb. 2016

Description

A stack of conical shells is simulated. Hyperbolic version. To intersect the Wolter I plates we take advatage of the azimuthal symmetry and only consider the radial component of the photon's wavevector.

Example: Ring_h(pore_th=0, ring_nr=3, Z0=12, yheight=0.83e-3, mirror_reflec="mirror_coating_u
R_d=0, size_file="ref_design_breaks.txt")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
ring_nr	1	Ring number; selects the data block read from size_file	
pore_th	m	Thickness of the pore walls (plates)	
Z0	m	distance between intersection plane and the focal spot(essentially the focal length).	
yheight	m	Height of the pore. (Thus the inner radius is radius_{m,h}-yheight	
chamferwidth	m	Width of side walls.	0
gap	m	gap between intersection with parabolic section and actual plate.	0
zdepth	m	Currently not used by the component	0
mirror_reflec		Data file containing reflectivities of the reflector surface (TOP).	""
bottom_reflec		Data file containing reflectivities of the bottom surface (BOTTOM).	""
size_file	str	Table file with the per-ring geometry, read with block number ring_nr	""
R_d		Default reflectivity value to use if no reflectivity file is given. Useful f.i. is one surface is reflecting and the others absorbing.	1
waviness	rad	Waviness of the pore reflecting surface. The slope error is assumed to be uniformly distributed in the interval [-waviness:waviness].	0
longw		If non-zero, waviness is 1D and along the pore axis.	0

Links

- Component source code found in file `Ring_h.comp`.

14.11. Pore optics

14.12. The Pore_c McXtrace Component

Single Pore as part of the Silicon Pore Optics (SPO) as envisioned for the ATHENA+ space telescope.

Identification

- **Author:** Erik B Knudsen and Desiree D. M. Ferreira
- **Origin:** DTU Physics, DTU Space
- **Date:** Feb. 2016

Description

A single pore is simulated, which may have thick walls. The top and bottom are curved cylindrically azimuthally, whereas they are straight sagitally. The primary parameter specifies whether this is a primary or secondary mirror. If primary they mirror extends backwards. The azimuthal curvature is defined by the parameter radius. This refers to the center of the pore. I.e the top and bottom plates have radius of curvature <radius+yheight/2> and <radius-yheight/2> respectively.

To intersect the Wolter I plates we take advatage of the azimuthal symmetry and only consider the radial component of the photon's wavevector.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_m	m	Ring radius of the upper (reflecting) plate of the pore at the optic centre.	
Z0	m	Distance between optics centre plane and focal spot (essentially focal length).	
xwidth	m	Width of the pore.	
yheight	m	Height of the pore.	
gap	m	Gap between the plate and the intersection plane with the hyperbolic section. (currently ignored)	0
chamferwidth	m	Width of side walls.	0
length	m	Length along the optical axis	0
mirror_reflec		Data file containing reflectivities of the reflector surface (TOP).	""

Name	Unit	Description	Default
bottom_reflec		Data file containing reflectivities of the bottom surface (BOTTOM).	""
side_reflec		Data file containing reflectivities of the side walls (LEFT and RIGHT).	""
R_d		Default reflectivity value to use if no reflectivity file is given. Useful f.i. is one surface is reflecting and the others absorbing.	1
primary		If non-zero, the pore is considered a primary reflector, and extends towards negative z. I.e. the entry plane is behind the z=0-plane. If zero, the pore is considered secondary and extends from the z=0-plane and towards positive z.	1
dalpha	deg	Offset to the alpha angle computed from the focal length. Useful for targeting the modified conical geometry (currently ignored).	0
waviness	rad	Waviness of the pore reflecting surface. The slope error is assumed to be uniformly distributed in the interval [-waviness:waviness].	0
longw		If non-zero, waviness is 1D and along the pore axis.	0

Links

- Component source code found in file `Pore_c.comp`.

14.13. The `Pore_p` McXtrace Component

Single Pore as part of the Silicon Pore Optics (SPO) as envisioned for the ATHENA+ space telescope.

Identification

- **Author:** Erik B Knudsen and Desiree D. M. Ferreira
- **Origin:** DTU Physics, DTU Space
- **Date:** Feb. 2016

Description

A single pore is simulated, which may have thick walls. The top and bottom are curved cylindrically azimuthally, and according to the Wolter I optic lengthwise (sagitally). This is the parabolic part. The azimuthal curvature is defined by the radius parameters. This refers to the center of the pore. I.e the top and bottom plates have radius of curvature <radius+yheight/2> and <radius-yheight/2> respectively.

To intersect the Wolter I plates we take advatage of the azimuthal symmetry and only consider the radial component of the photon's wavevector.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_p	m	Ring radius of the upper (reflecting) plate of the pore at the edge furthest away from the focal point.	
radius_m	m	Ring radius of the upper (reflecting) plate of the pore at the intersection with the hyperbolic section.	
Z0	m	distance between intersection plane and the focal spot(essentially the focal length).	
xwidth	m	Width of the pore.	
yheight	m	Height of the pore. (Thus the inner radius is radius_{m,h}-yehight.	
gap	m	gap between intersection with parabolic section and actual plate.	0
chamferwidth	m	Width of side walls.	0
zdepth	m	Currently not used by the component	0
mirror_reflec		Data file containing reflectivities of the reflector surface (TOP).	""
bottom_reflec		Data file containing reflectivities of the bottom surface (BOTTOM).	""
side_reflec		Data file containing reflectivities of the side walls (LEFT and RIGHT).	""
R_d		Default reflectivity value to use if no reflectivity file is given. Useful f.i. is one surface is reflecting and the others absorbing.	1

Name	Unit	Description	Default
waviness	rad	Waviness of the reflecting surface. The slope error is assumed to be uniformly distributed in the interval [-waviness:waviness].	0
longw	1	If non-zero, waviness is 1D and along the pore axis.	0

Links

- Component source code found in file `Pore_p.comp`.

14.14. The `Pore_h` McXtrace Component

Single Pore as part of the Silicon Pore Optics (SPO) as envisioned for the ATHENA+ space telescope.

Identification

- **Author:** Erik B Knudsen and Desiree D. M. Ferreira
- **Origin:** DTU Physics, DTU Space
- **Date:** Feb. 2016

Description

A single pore is simulated, which may have thick walls. The top and bottom are curved cylindrically azimuthally, and according to the Wolter I optic lengthwise (sagitally). This is the hyperbolic part. The azimuthal curvature is defined by the radius parameters. This refers to the center of the pore. I.e the top and bottom plates have radius of curvature <radius+yheight/2> and <radius-yheight/2> respectively.

To intersect the Wolter I plates we take advatage of the azimuthal symmetry and only consider the radial component of the photon's wavevector.

Example: Pore_h(radius_m=0.286148, radius_h=0.284333, zdepth=0.101504, Z0=12, xwidth=0.83e-3, yheight=0.605e-3, mirror_reflec="mirror_coating_unity.txt", R_d=0)

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
radius_m	m	Ring radius of the upper (reflecting) plate of the pore at the intersection with the parabolic section.	
radius_h	m	Ring radius of the upper (reflecting) plate of the pore at the edge closest to the focal point.	
Z0	m	distance between intersection plane and the focal spot(essentially the focal length).	
xwidth	m	Width of the pore.	
yheight	m	Height of the pore. (Thus the inner radius is radius_{m,h}-yehight.	
chamferwidth	m	Width of side walls.	0
gap	m	gap between intersection with parabolic section and actual plate.	0
zdepth	m	Currently not used by the component	0
mirror_reflec		Data file containing reflectivities of the reflector surface (TOP).	""
bottom_reflec		Data file containing reflectivities of the bottom surface (BOTTOM).	""
side_reflec		Data file containing reflectivities of the side walls (LEFT and RIGHT).	""
R_d		Default reflectivity value to use if no reflectivity file is given. Useful f.i. is one surface is reflecting and the others absorbing.	1
waviness	rad	Waviness of the reflecting surface. The slope error is assumed to be uniformly distributed in the interval [-waviness:waviness].	0
longw	1	If non-zero, waviness is 1D and along the pore axis.	0

Links

- Component source code found in file `Pore_h.comp`.

14.15. Micropore/microchannel plate (MM) optics

14.16. The MM_c McXtrace Component

Single Pore as part of the Silicon Pore Optics (SPO) as envisioned for the ATHENA+ space telescope.

Identification

- **Author:** Erik B Knudsen and Desiree D. M. Ferreira
- **Origin:** DTU Physics, DTU Space
- **Date:** Feb. 2016, Feb. 2017

Description

A single pore is simulated, which may have thick walls. The top and bottom are curved cylindrically azimuthally, and according to the Wolter I optic lengthwise (sagitally). A parameter specifies whether this is hyperbolic or parabolic. The azimuthal curvature is defined by the parameter radius. This refers to the center of the pore. I.e the top and bottom plates have radius of curvature <radius+yheight/2> and <radius-yheight/2> respectively.

To intersect the Wolter I plates we take advatage of the azimuthal symmetry and only consider the radial component of the photon's wavevector.

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
pore_th	m	Thickness of the pore walls (plates)	
pore_width	m	Width of a single pore	
ring_nr	1	Ring number; selects the data block read from size_file	
radius_m	m	Ring radius of the upper (reflecting) plate of the pore at the optic centre.	
Z0	m	Distance between optics centre plane and focal spot (essentially focal length).	
xwidth	m	Width of the pore.	
pore_height	m	Height of the pore	
gap	m	Gap between the plate and the intersection plane with the hyperbolic section. (currently ignored)	0
chamfer_width	m	Width of side walls.	0

Name	Unit	Description	Default
length	m	Length along the optical axis	0
mirror_reflec		Data file containing reflectivities of the reflector surface (TOP).	""
bottom_reflec		Data file containing reflectivities of the bottom surface (BOTTOM).	""
side_reflec		Data file containing reflectivities of the side walls (LEFT and RIGHT).	""
size_file	str	Table file with the per-ring geometry, read with block number ring_nr	""
non_specular_file	str	Data file describing non-specular reflectivity	""
R_d		Default reflectivity value to use if no reflectivity file is given. Useful f.i. is one surface is reflecting and the others absorbing.	1
primary		If non-zero, the pore is considered a primary reflector, and extends towards negative z. I.e. the entry plane is behind the z=0-plane. If zero, the pore is considered secondary and extends from the z=0-plane and towards positive z.	1
dalpha	deg	Offset to the alpha angle computed from the focal length. Useful for targeting the modified conical geometry (currently ignored).	0
waviness	rad	Waviness of the reflecting surface. The slope error is assumed to be uniformly distributed in the interval [-waviness:waviness].	0
longw		If non-zero, waviness is 1D and along the pore axis.	0

Links

- Component source code found in file `MM_c.comp`.

14.17. The `MM_p` McXtrace Component

Date: Feb. 2017 Version: 1.1 Release: McXtrace 1.2 Origin: DTU Physics, DTU Space
Single Pore as part of the Silicon Pore Optics (SPO) as envisioned for the ATHENA+ space telescope.

Identification

- **Author:** Erik B Knudsen and Desiree D. M. Ferreira
- **Origin:** DTU Physics, DTU Space
- **Date:** Feb. 2016

Description

A single pore is simulated, which may have thick walls. The top and bottom are curved cylindrically azimuthally, and according to the Wolter I optic lengthwise (sagitally). A parameter specifies whether this is hyperbolic or parabolic. The azimuthal curvature is defined by the parameter radius. This refers to the center of the pore. I.e the top and bottom plates have radius of curvature <radius+pore_height/2> and <radius-pore_height/2> respectively.

To intersect the Wolter I plates we take advatage of the azimuthal symmetry and only consider the radial component of the photon's wavevector.

Example: MM_p(pore_th=0, ring_nr=1, Z0=12, pore_width=0.83e-3 , pore_height=0.605e-3, mirror_reflec="mirror_coating_unity.txt", R_d=0, size_file="ref_design_breaks.txt")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
pore_th	m	Thickness of the pore walls (plates)	
ring_nr	1	Ring number; selects the data block read from size_file	
Z0	m	Distance between optics centre plane and focal spot (essentially focal length).	
pore_height	m	Height of the pore.	
pore_width	m	Width of a single pore	
chamfer_width	m	Width of side walls.	0
gap	m	Gap between the plate and the intersection plane with the hyperbolic section.	0
zdepth	m	Currently not used by the component	0
mirror_reflec		Data file containing reflectivities of the reflector surface (TOP).	""
bottom_reflec		Data file containing reflectivities of the bottom surface (BOTTOM).	""
side_reflec		Data file containing reflectivities of the side walls (LEFT and RIGHT).	""
size_file	str	Table file with the per-ring geometry, read with block number ring_nr	""

Name	Unit	Description	Default
non_specular_file	str	Data file describing non-specular reflectivity	""
R_d		Default reflectivity value to use if no reflectivity file is given. Useful f.i. is one surface is reflecting and the others absorbing.	1
waviness	rad	Waviness of the reflecting surface. The slope error is assumed to be uniformly distributed in the interval [-waviness:waviness].	0
longw	1	If non-zero, waviness is 1D and along the pore axis.	0

Links

- Component source code found in file `MM_p.comp`.

14.18. The `MM_h` McXtrace Component

Date: Feb. 2017 Version: 1.1 Release: McXtrace 1.2 Origin: DTU Physics, DTU Space

Single Pore as part of the Silicon Pore Optics (SPO) as envisioned for the ATHENA+ space telescope.

Identification

- **Author:** Erik B Knudsen and Desiree D. M. Ferreira
- **Origin:** DTU Physics, DTU Space
- **Date:** Feb. 2016

Description

A single pore is simulated, which may have thick walls. The top and bottom are curved cylindrically azimuthally, and according to the Wolter I optic lengthwise (sagitally). This is the hyperbolic part. The azimuthal curvature is defined by the radius parameters.

To intersect the Wolter I plates we take advatage of the azimuthal symmetry and only consider the radial component of the photon's wavevector.

Example: MM_h(pore_th=0, ring_nr=1, Z0=12, pore_width=0.83e-3, pore_height=0.605e-3, chamfer_width=0.17e-3, mirror_reflec="mirror_coating_unity.txt", R_d=0, size_file="ref_design_breaks.txt")

Input parameters

Parameters in **boldface** are required; the others are optional.

Name	Unit	Description	Default
pore_th	m	Thickness of the pore walls (plates)	
ring_nr	1	Ring number; selects the data block read from size_file	
Z0	m	Distance between intersection plane and the focal spot (essentially focal length).	
pore_height	m	Height of the pore. (Hence the inner radii are radius_{m,h}-pore_height)	
pore_width	m	Width of a single pore	
chamfer_width	m	Width of side walls.	0
gap	m	Gap between intersection with parabolic section and actual plate.	0
zdepth	m	Currently not used by the component	0
mirror_reflec		Data file containing reflectivities of the reflector surface (TOP).	""
bottom_reflec		Data file containing reflectivities of the bottom surface (BOTTOM).	""
side_reflec		Data file containing reflectivities of the side walls (LEFT and RIGHT).	""
size_file	str	Table file with the per-ring geometry, read with block number ring_nr	""
non_specular_file	str	Data file describing non-specular reflectivity	""
R_d		Default reflectivity value to use if no reflectivity file is given. Useful f.i. is one surface is reflecting and the others absorbing.	1
waviness	rad	Waviness of the reflecting surface. The slope error is assumed to be uniformly distributed in the interval [-waviness:waviness].	0
longw	1	If non-zero, waviness is 1D and along the pore axis.	0

Links

- Component source code found in file `MM_h.comp`.

A. The Union components

The **Union components** are a set of McXtrace components, originally developed for McStas by Mads Bertelsen [Ber17; BJL17] and subsequently ported to McXtrace, that take a fundamentally different approach to sample and sample-environment simulation than ordinary McXtrace components. This chapter explains the concepts behind the Union components and how to use them from an instrument file; the full parameter reference for every individual Union component is auto-generated from the component headers and found in the Component Manual's Union chapter.

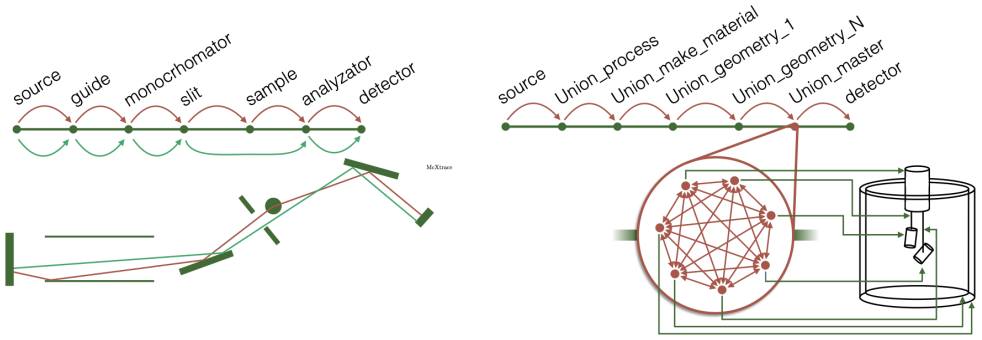
The McXtrace port covers the core of the framework – processes, materials, geometries, masks, exit volumes and the master component – but not (yet) every feature of the McStas original: there is currently no logger, absorption-logger, conditional or surface (reflection/refraction) support, and no ray-history tagging system. Everything described in this chapter is available; where a feature exists in McStas but not McXtrace, this is noted explicitly.

A.1. Motivation

An ordinary McXtrace instrument is a linear sequence of self-contained components: an x-ray enters a component, something happens to it (or not), and it moves on to the next component in the sequence. This is a good match for the intended beam path through an instrument, but it means that

- a single sample component must contain *all* the physics relevant to that sample – absorption and every scattering process – as one monolithic block of code, and
- multiple scattering *between* physically separate parts of a sample environment (a sample inside a capillary, inside a sample changer, ...) is essentially impossible to capture, since each part would need to be a separate component and McXtrace components do not natively talk to each other.

The Union components solve this by splitting the sample simulation task into several cooperating component classes – *processes*, *materials*, *geometries* and a *master* – and performing the actual ray tracing for all of them together, inside the master component, with full native multiple scattering between every defined volume. Because physics (processes) and geometry are separated, adding a new scattering process is a comparatively small task, and existing processes can be freely combined and reused across different shapes.



Traditional McXtrace instrument

McXtrace instrument file using Union components

Figure A.1.: Left: the linear succession of components in a traditional McXtrace instrument file. Right: the same instrument using Union components – all ray tracing for the sample environment happens inside a single master component, which works around the normal McXtrace component sequence to achieve native multiple scattering between an arbitrary number of volumes. (Originally illustrated for McStas; the same principle applies unchanged to McXtrace.)

It is entirely possible, and common, to mix ordinary McXtrace components with Union components in the same instrument file – the Union machinery only takes over at its `Union_master` components (section A.2).

A.1.1. Volumes, priority and overlap

The central concept of the Union components is the *volume*: a geometrical shape (box, cylinder, sphere or cone) combined with a material definition (absorption plus zero or more scattering processes), placed in space using the ordinary `AT/ROTATED` mechanism. Volumes may be placed in any order in the instrument file, and, unlike ordinary McXtrace geometry, volumes are allowed to *overlap*.

Each volume is given a unique `priority` value. Wherever two or more volumes overlap in space, the material of the volume with the *highest* priority applies. This single rule makes it straightforward to build geometries that would otherwise require careful, error-prone manual splitting into non-overlapping pieces: a hollow capillary wall is simply a material cylinder with a vacuum cylinder of higher priority placed inside it, and a layered sample with several different materials is built from concentric or adjacent shapes, each its own volume and material.

A.2. Component classes

A Union simulation is assembled from components of several distinct classes, always used in the same overall order:

1. **Process components** – define a scattering process (section A.2.2).

2. `Union_make_material` – collect processes and an absorption cross section into a named material (section A.2.3).

3. **Geometry components** – place a volume with a given material in the simulation (section A.2.4).

4. `Union_master` – performs the actual ray tracing for all volumes defined since the previous master (section A.2.5).

A.2.1. Loading the Union library

Union components need no setup or cleanup components. Each one loads the Union library itself, and the library keeps the internal bookkeeping lists (of processes, materials, geometries, ...) that the components communicate through, freeing them once the last Union component has finished.

An instrument that uses Union components must contain at least one `Union_master`; without one it fails to compile, with an error naming `Union_components_need_a_Union_master_in_the_instrumen` Union components placed after the last `Union_master` have no effect, and the simulation warns about each of them when it finishes.

`Union_init` and `Union_stop`, which older instruments place first and last among their Union components, are no longer needed. They are kept without effect, apart from a deprecation warning, so that such instruments still compile. For the same reason every Union component still accepts the string parameter `init`, which older instruments set to the name of their `Union_init` component; its value is ignored.

A.2.2. Scattering processes

A process component has no physical shape of its own. It describes a single physical scattering mechanism: a function giving the inverse penetration depth (macroscopic cross section) as a function of incoming wavevector, and a function describing what happens to a ray in a scattering event. Table A.1 lists the processes currently available. Note that x-ray-specific processes (`Compton_xrl_process`, `KN_xrl_process`, `Rayleigh_xrl_process`) take a `density` and an atomic number `atomno` rather than a cross section, and use the external `xraylib` library for the underlying physics.

Every process has an `interact_fraction` setting parameter, in the range $[0, 1]$ or -1 to disable, which can be used to artificially force a certain fraction of scattering events to that process (an importance sampling technique, entirely analogous to `p_interact` discussed below); the resulting ray weight is corrected automatically so the simulation remains unbiased. If left at -1 for every process in a material, the interact fractions default to the actual, physical scattering probabilities.

A minimal example defining a Compton scattering process for copper:

```
1 COMPONENT compton1 = Compton_xrl_process(density=8.92, atomno=29)
2 AT (0,0,0) ABSOLUTE
```

Component	Description
`Incoherent_process`	Isotropic, elastic incoherent scattering (cross-section based, ported from McStas).
`Compton_xrl_process`	Compton (incoherent) scattering via `xraylib`, using `density` and atomic number `atomno`.
`KN_xrl_process`	Klein-Nishina Compton scattering via `xraylib`.
`Rayleigh_xrl_process`	Rayleigh (coherent, elastic) scattering via `xraylib`.
`Powder_process`	Bragg scattering from a powder (based on `PowderN`, given a reflection list via `reflections`).
`Template_process`	Documented template for writing a new process.

Table A.1.: Scattering processes currently available for the McXtrace Union components.

A process component's position in the instrument file is irrelevant to the physics (it has no shape); its *name* is what matters, since materials refer to processes by name.

A.2.3. Union_make_material

`Union_make_material` collects one or more previously defined processes into a named material, together with the absorption cross section (as the inverse penetration depth, `my_absorption`, in m^{-1}):

```
COMPONENT cmpton1 = Union_make_material(
    process_string="compton1", my_absorption=0)
AT (0,0,0) ABSOLUTE
```

As x-ray absorption and scattering are often already captured by the `xrl`-based processes' own physics, it is common to see `my_absorption=0` in practice, as in the example above and the complete worked example in section A.4. If `process_string` is left unset, all processes defined since the previous `Union_make_material` are collected automatically; giving it explicitly is recommended, since it makes the resulting material self-documenting. Two material names are reserved and always available without being explicitly defined: `"Vacuum"` (no processes, no absorption) and `"Exit"` (behaves like vacuum, but see section A.2.4 below). Setting `absorber=1` instead of giving a `process_string` creates a purely absorbing material with no scattering processes at all.

A.2.4. Geometry components

A geometry component places a volume of a given shape and material in the simulation, using the material name from `Union_make_material` and the ordinary `AT`/`ROTATED` keywords for position and orientation. The currently available shapes are `Union_box`, `Union_cylinder`, `Union_sphere` and `Union_cone` (which can also represent a plain cylinder by setting `radius_top`=`radius_bottom`, or a full cone by leaving one of them at its

default of 0). No ray tracing happens in a geometry component itself – it only registers the volume with the following `Union_master`.

Every geometry component shares a common set of setting parameters, summarized in table A.2, in addition to its own shape-specific dimensions (e.g. `radius`/`yheight` for `Union_cylinder`; `xwidth`/`yheight`/`zdepth` for `Union_box`; `radius_top`/`radius_bottom`/`yheight` for `Union_cone`).

Parameter	Description
`material_string`	Name of a material from `Union_make_material`, or `"Vacuum"`/`"Exit"`.
`priority`	Unique priority value; the volume with the highest priority wins where volumes overlap (section A.1.1).
`p_interact`	Forces this probability [0-1] for a scattering event when a ray is inside the volume, regardless of path length (importance sampling; see note below).
`visualize`	Set to 0 to hide this volume in `mcdisplay`.
`mask_string`, `mask_setting`	Turn this volume into a *mask* restricting one or more other volumes (section A.2.4).
`number_of_activations`	Number of subsequent `Union_master` components this volume should be simulated by (section A.2.4).
`target_index`/ `target_x,y,z`, `focus_aw`/`focus_ah`, `focus_xw`/`focus_xh`, `focus_r`	Standard McXtrace focusing parameters, used by any process assigned to this volume's material that supports focusing.

Table A.2.: Setting parameters common to all McXtrace Union geometry components.

Unlike the McStas Union components, the McXtrace geometry components do not yet have any `surface`/`cut_surface`-style parameters – there is no reflection/refraction (surface) support in the McXtrace port (section A.1).

p_interact and multiple scattering

Unlike `p_interact` in an ordinary McXtrace sample component, `p_interact` on a Union volume applies at *every* step of the multiple scattering loop, not just once. Setting it to 50%, for instance, gives a 25% chance of *two* scattering events in that volume. It is therefore best kept well below 1; values close to 1 lead to very high probabilities of high-order multiple scattering and correspondingly large statistical weight corrections.

Masks

A mask volume restricts one or more other (*masked*) volumes: a ray only sees the masked volume's material where the masked volume and the mask volume(s) both cover the same space. This gives additional geometric freedom beyond what overlap-by-priority alone provides. A masked volume can have several masks; `mask_setting` (`"All"` or `"Any"`, default `"All"`) controls whether every mask or just any one of them must cover a point for the masked volume to apply there. A mask volume is set via `mask_string` (a comma-separated list of the geometry names it masks) rather than `material_string`.

Exit volumes and number_of_activations

Normally, once a ray enters the region of space covered by a `Union_master`, it stays inside the Union ray-tracing loop until it escapes all defined volumes. Sometimes it is useful to break out of this loop early – for example to place an ordinary McXtrace monitor or component somewhere inside an otherwise Union-simulated geometry. Assigning a volume the special material `"Exit"` achieves this: once a ray enters an exit volume, the current master stops and the *next* McXtrace component in the instrument sequence runs as normal.

`number_of_activations` (default 1) controls how many subsequent `Union_master` components will simulate a given volume, which is useful when a static piece of geometry should be present across more than one master section of the instrument without redefining it.

A.2.5. Union_master

`Union_master` is where the actual ray tracing happens – it is the component that is placed into the ordinary linear McXtrace component sequence, and by default it simulates every volume defined since the previous master (or, for the first master, since the start of the instrument). If a volume is defined after the last `Union_master` in the file, it will never be simulated – remember to place a master after your geometry.

```
1 COMPONENT master = Union_master()
2 AT (0,0,0) RELATIVE sample_position
```

`Union_master` also has an `allow_inside_start` setting (set to 1 if rays are expected to start already inside a volume at this master) and `verbal`/`list_verbal`/`finally_verbal` settings for diagnostic terminal output; see the Component Manual entry for the complete list. Unlike McStas's Union components, there is currently no separate GPU-enabled master variant for McXtrace.

A.3. The propagation algorithm

This section gives a conceptual outline of how `Union_master` actually propagates rays; see the original thesis [Ber17], written for the McStas Union components but describing the same underlying algorithm, for the full derivation and pseudocode.

Within a volume, the distance to the next scattering position is drawn from the usual exponential distribution set by the total inverse penetration depth (sum over the volume's absorption and all its processes' scattering contributions). If this distance is less than the distance to the volume's boundary, a scattering event occurs there; otherwise, the ray is propagated to the boundary and the simulation continues in whichever volume lies on the other side.

Naively, determining which volume lies on the other side of a boundary would require testing every other defined volume for intersection and containment, for every propagation step – prohibitively expensive once more than a handful of volumes are involved. Instead, in the `initialize` section of each `Union_master`, the whole ensemble of volumes is analysed once to build, for every volume, the (usually small) set of other volumes that actually need an intersection test from there, and the set of volumes the ray could possibly end up in immediately after leaving through its own boundary. This turns an $O(N)$-per-step cost into something close to $O(1)$ in typical geometries. Figure A.2 sketches the resulting behaviour for a simple case (illustrated for a neutron ray, but the same principle applies unchanged to McXtrace's x-rays).

A.3.1. Known limitations

- **Coincident surfaces.** If two volume boundaries coincide exactly, the algorithm cannot reliably determine which volume a ray enters. It is safe, and normal, to *overlap* volumes – but two volumes should never be placed so that a face merely touches without any overlap or gap; leaving even a micron-scale gap or overlap avoids the problem entirely.

- **No gravity.** The underlying intersection algorithms are linear, so Union volumes propagate in straight lines regardless of any gravity setting elsewhere in the instrument (in practice rarely relevant for x-ray instruments).

- **Feature parity with McStas.** As noted in section A.1, loggers, absorption loggers, conditionals, surfaces (reflection/refraction) and ray-history tagging are not currently available for McXtrace's Union components.

A.4. A complete example

The following excerpt, adapted from the `PowderCompton_union` example instrument (section A.5), combines Compton scattering in a sphere with powder diffraction in a box:

```
1 COMPONENT compton1 = Compton_xrl_process(density=8.92, atomno=29)
2 AT (0,0,0) ABSOLUTE
3
4 COMPONENT pow2_proc = Powder_process(reflections=refs)
5 AT (0,0,0) ABSOLUTE
6
7 COMPONENT cmpton1 = Union_make_material(
```

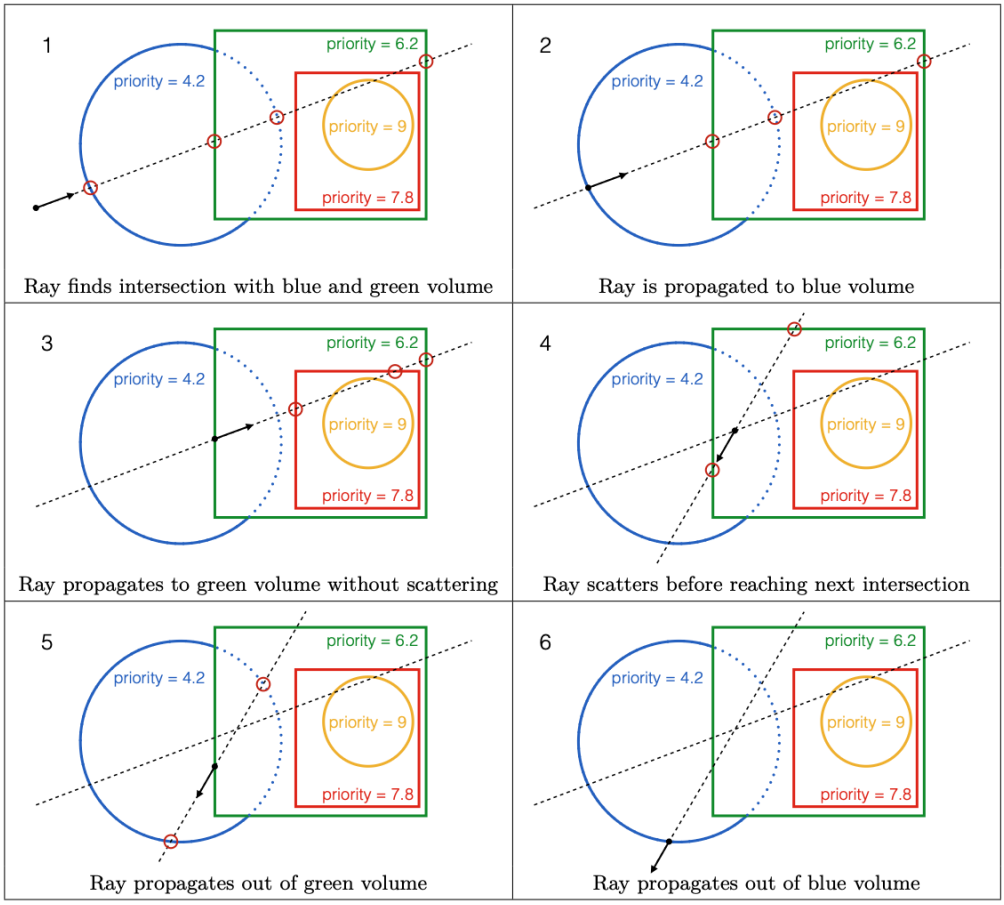


Figure A.2.: Illustration of the Union propagation algorithm for a single scattering event between two overlapping volumes. Only intersections actually relevant given the ray's current volume are calculated at each step (the orange volume, never reachable from the ray's actual path here, is never tested).

```
8       process_string="compton1", my_absorption=0)
9   AT (0,0,0) ABSOLUTE
10
11  COMPONENT pow2 = Union_make_material(
12      process_string="pow2_proc", my_absorption=0)
13  AT (0,0,0) ABSOLUTE
14
15  /* ... source, slits etc. as normal ... */
16
17  COMPONENT usphere = Union_sphere(radius=0.25e-3, material_string="cmpton1",
18      priority=2)
19  AT (0,-0.3e-3,0) RELATIVE sample_position
20
21  COMPONENT ubox = Union_box(xwidth=1e-3, yheight=0.75e-3, zdepth=0.5e-3,
```

```
22      material_string="pow2", priority=1)
23 AT (0,0.8e-3,0) RELATIVE sample_position
24
25 COMPONENT master = Union_master()
26 AT (0,0,0) RELATIVE sample_position
27
28 COMPONENT fpi = PSD_monitor_4PI(radius=1.25, nx=101, ny=101, filename="fpi"
       )
29 AT (0,0,0) RELATIVE sample_position
```

Note the two volumes overlap in neither space nor priority here (a sphere and a box placed side by side) – the priority mechanism only matters where volumes actually share space, as discussed in section A.1.1.

A.5. Further examples

The example instrument library ships Union examples in the `Tests_union` category, including `PowderCompton_union` (the basis for the worked example above, combining Compton and powder scattering), `Test_KN_Comp_Rayl_union` (comparing the Klein-Nishina, Compton and Rayleigh processes) and `Geometry_test`. See the User Manual's chapter on instrument examples for how to browse these. A number of the individual component headers, documented in the Component Manual's Union chapter, contain further short usage examples.

B. Libraries and conversion constants

The McXtrace Library contains a number of built-in functions and conversion constants which are useful when constructing components. These are stored in the `share` directory of the `MCXTRACE` library.

Within these functions, the 'Run-time' part is available for all component/instrument descriptions. The other parts are dynamic, that is they are not pre-loaded, but only imported once when a component requests it using the `%include` McXtrace keyword. For instance, within a component C code block, (usually SHARE or DECLARE):

```
1  %include "read_table-lib"
```

will include the 'read_table-lib.h' file, and the 'read_table-lib.c' (unless the `--no-runtime` option is used with `mcxtrace`). Similarly,

```
1  %include "read_table-lib.h"
```

will *only* include the 'read_table-lib.h'. The library embedding is done only once for all components (like the SHARE section). For an example of implementation, see **Res_monitor**.

In this Appendix, we present a short list of both each of the library contents and the run-time features.

B.1. Run-time calls and functions (`mcxtrace-r`)

Here we list a number of preprogrammed macros and functions which may ease the task of writing component and instrument definitions. By convention macros are in upper case whereas functions are in lower case.

B.1.1. Photon propagation

Propagation routines perform all necessary operations to transport x-rays from one point to an other. Except when using the special `ALLOW_BACKPROP;` call prior to executing any `PROP_*` propagation, the x-rays which have negative propagation lengths are removed automatically.

- **ABSORB**. This macro issues an order to the overall McXtrace simulator to interrupt the simulation of the current x-ray history and to start a new one.
- **PROP_Z0**. Propagates the x-ray to the $z = 0$ plane, by adjusting (x, y, z), ϕ, and t accordingly from knowledge of the x-ray wavevector (kx, ky, kz). If the propagation length is negative, the x-ray is absorbed, except if a `ALLOW_BACKPROP;` preceeds it.

For components that are centered along the z-axis, use the `_intersect` functions to determine intersection time(s), and then a `PROP_DL` call.

- **PROP_X0, PROP_Y0**. These macros are analogous to `PROP_Z0` except they propagate to the $x = 0$ and $y = 0$ planes respectively.

- **PROP_DL**(dl). Propagates the x-ray by the length dl, adjusting (x, y, z), ϕ, t accordingly, from knowledge of the x-ray wavevector.

- **ALLOW_BACKPROP**. Indicates that the *next* propagation routine will not remove the x-ray, even if negative propagation lengths are found. Subsequent propagations are not affected.

- **SCATTER**. This macro is used to denote a scattering event inside a component. It should be used to indicate that a component has interacted with the x-ray (e.g. scattered or detected). This does not affect the x-ray state (see, however, **Beamstop**), and it is mainly used by the `MCDISPLAY` section and the `GROUP` modifier. See also the SCATTERED variable (below).

B.1.2. Coordinate and component variable retrieval

- **MC_GETPAR**(*comp*, *outpar*). This may be used in e.g. the FINALLY section of an instrument definition to reference the parameters of a component.

- **NAME_CURRENT_COMP** gives the name of the current component as a string.

- **POS_A_CURRENT_COMP** gives the absolute position of the current component. A component of the vector is referred to as POS_A_CURRENT_COMP.i where i is x, y or z.

- **ROT_A_CURRENT_COMP** and **ROT_R_CURRENT_COMP** give the orientation of the current component as rotation matrices (absolute orientation and the orientation relative to the previous component, respectively). A component of a rotation matrix is referred to as ROT_A_CURRENT_COMP[m][n], where m and n are 0, 1, or 2 standing for x, y and z coordinates respectively.

- **POS_A_COMP**(*comp*) gives the absolute position of the component with the name *comp*. Note that *comp* is not given as a string. A component of the vector is referred to as POS_A_COMP(*comp*).i where i is x, y or z.

- **ROT_A_COMP**(*comp*) and **ROT_R_COMP**(*comp*) give the orientation of the component *comp* as rotation matrices (absolute orientation and the orientation relative to its previous component, respectively). Note that *comp* is not given as a string. A component of a rotation matrice is referred to as ROT_A_COMP(*comp*)[m][n], where m and n are 0, 1, or 2.

- **INDEX_CURRENT_COMP** is the number (index) of the current component (starting from 1).
- **POS_A_COMP_INDEX**(*index*) is the absolute position of component *index*. POS_A_COMP_INDEX (INDEX_CURRENT_COMP) is the same as POS_A_CURRENT_COMP. You may use POS_A_COMP_INDEX (INDEX_CURRENT_COMP+1) to make, for instance, your component access the position of the next component (this is usefull for automatic targeting). A component of the vector is referred to as POS_A_COMP_INDEX(*index*).*i* where *i* is x, y or z.
- **POS_R_COMP_INDEX** works the same as above, but with relative coordinates.
- **STORE_XRAY**($index, x, y, z, kx, ky, kz, phi, t, Ex, Ey, Ez, p$) stores the current x-ray state in the trace-history table, in local coordinate system. *index* is usually INDEX_CURRENT_COMP. This is automatically done when entering each component of an instrument.
- **RESTORE_XRAY**($index, x, y, z, kx, ky, kz, phi, t, Ex, Ey, Ez, p$) restores the x-ray state to the one at the input of the component *index*. To ignore a component effect, use RESTORE_XRAY (INDEX_CURRENT_COMP, $x, y, z, kx, ky, kz, phi, Ex, Ey, Ez, p$) at the end of its TRACE section, or in its EXTEND section. These x-ray states are in the local component coordinate systems.
- **SCATTERED** is a variable set to 0 when entering a component, which is incremented each time a SCATTER event occurs. This may be used in the `EXTEND` sections to determine whether the component interacted with the current x-ray.
- **extend_list**(n, &*arr*, &*len*, *elemsize*). Given an array *arr* with *len* elements each of size *elemsize*, make sure that the array is big enough to hold at least n elements, by extending *arr* and *len* if necessary. Typically used when reading a list of numbers from a data file when the length of the file is not known in advance.
- **mcset_ncount**(n). Sets the number of x-ray histories to simulate to n.
- **mcget_ncount**(). Returns the number of x-ray histories to simulate (usually set by option `-n`).
- **mcget_run_num**(). Returns the number of x-ray histories that have been simulated until now.

B.1.3. Coordinate transformations

- **coords_set**(x, y, z) returns a Coord structure (like POS_A_CURRENT_COMP) with x, y and z members.
- **coords_get**(P, &x, &y, &z) copies the x, y and z members of the Coord structure P into x, y, z variables.

- **coords_add**(a, b), **coords_sub**(a, b), **coords_neg**(a) enable to operate on coordinates, and return the resulting Coord structure.
- **rot_set_rotation**(*Rotation t*, ϕ_x, ϕ_y, ϕ_z) Get transformation matrix for rotation first ϕ_x around x axis, then ϕ_y around y, and last ϕ_z around z. t should be a 'Rotation' ([3][3] 'double' matrix).
- **rot_mul***(Rotation t1, Rotation t2, Rotation t3)* performs $t3 = t1.t2$.
- **rot_copy***(Rotation dest, Rotation src)* performs $dest = src$ for Rotation arrays.
- **rot_transpose***(Rotation src, Rotation dest)* performs $dest = src^t$.
- **rot_apply***(Rotation t, Coords a)* returns a Coord structure which is $t.a$

B.1.4. Mathematical routines

- **NORM**(x, y, z). Normalizes the vector (x, y, z) to have length 1.
- **scalar_prod**$(a_x, a_y, a_z, b_x, b_y, b_z)$. Returns the scalar product of the two vectors (a_x, a_y, a_z) and (b_x, b_y, b_z).
- **vec_prod**(&a_x,&a_y,&a_z, b_x,b_y,b_z, c_x,c_y,c_z). Sets (a_x, a_y, a_z) equal to the vector product $(b_x, b_y, b_z) \times (c_x, c_y, c_z)$.
- **rotate**(&x,&y,&z,v_x,v_y,v_z,φ,a_x,a_y,a_z). Set (x, y, z) to the result of rotating the vector (v_x, v_y, v_z) the angle φ (in radians) around the vector (a_x, a_y, a_z).
- **normal_vec**(n_x, n_y, n_z, x, y, z). Computes a unit vector (n_x, n_y, n_z) normal to the vector (x, y, z).*
- **solve_2nd_order**(*t_0,*t_1, A, B, C). Solves the 2nd order equation $At^2+Bt+C = 0$ and puts the solutions in *t_0 and *t_1. The smallest positive solution into pointer *t_0. If t_1=`NULL` it is ignored and the second solution is discarded.

B.1.5. Output from detectors

Details about using these functions are given in the McXtrace User Manual.

- **DETECTOR_OUT_0D**(...). Used to output the results from a single detector. The name of the detector is output together with the simulated intensity and estimated statistical error. The output is produced in a format that can be read by McXtrace front-end programs.
- **DETECTOR_OUT_1D**(...). Used to output the results from a one-dimensional detector. Integrated intensities error etc. is also reported as for DETECTOR_OUT_0D.
- **DETECTOR_OUT_2D**(.....). Used to output the results from a two-dimentional detector. Integrated intensities error etc. is also reported as for DETECTOR_OUT_0D.

- **DETECTOR_OUT_0D_MEAN**, **DETECTOR_OUT_1D_MEAN** and **DETECTOR_OUT_2D_MEAN**(...). Used to output the mean of a quantity instead of an intensity, see DETECTOR_OUT_0D, 1D and 2D.
- **mcinfo_simulation***(FILE *f, mcformat, char *pre, char *name)* is used to append the simulation parameters into file f (see for instance **Res_monitor**). Internal variable $mcformat$ should be used as specified. Please contact the authors for further information.

B.1.6. Ray-geometry intersections

- **inside_rectangle**(x, y, xw, yh). Return 1 if $-xw/2 \leq x \leq xw/2$ AND $-yh/2 \leq y \leq yh/2$. Else return 0.
- **box_intersect**(&l_1, &l_2, x, y, z, k_x, k_y, k_z, d_x, d_y, d_z). Calculates the (0, 1, or 2) intersections between the x-ray path and a box of dimensions d_x, d_y, and d_z, centered at the origin for a x-ray with the parameters (x, y, z, k_x, k_y, k_z). The intersection lengths are returned in the variables l_1 and l_2, with $l_1 < l_2$. In the case of less than two intersections, t_1 (and possibly t_2) are set to zero. The function returns true if the x-ray intersects the box, false otherwise.
- **cylinder_intersect**(&l_1, &l_2, x, y, z, k_x, k_y, k_z, r, h). Similar to **box_intersect**, but using a cylinder of height h and radius r, centered at the origin.
- **sphere_intersect**(&l_1, &l_2, x, y, z, k_x, k_y, k_z, r). Similar to **box_intersect**, but using a sphere of radius r.
- **ellipsoid_intersect**(&l_1, &l_2, x, y, z, k_x, k_y, k_z, a,b,c,Q,). Similar to **box_intersect**, but using an ellipsoid with half-axis a,b,c oriented by the rotation matrix Q. If $Q = I$, a is along the x-axis, b along y and c along z

B.1.7. Random numbers

By default McXtrace uses a 64-bit KISS ("Keep It Simple Stupid") algorithm for generating pseudo random numbers, which runs on both CPU and GPU/OpenACC. The classic Mersenne Twister[MN98] algorithm remains available as a CPU-only, build-time alternative (see the User Manual's appendix on random numbers).

- **rand01**(). Returns a random number distributed uniformly between 0 and 1.
- **randnorm**(). Returns a random number from a normal distribution centered around 0 and with $\sigma = 1$. The algorithm used to sample the normal distribution is explained in Ref. [Pre+86, ch.7].
- **randpm1**(). Returns a random number distributed uniformly between -1 and 1.
- **randtriangle**(). Returns a random number from a triangular distribution between -1 and 1.

- **randvec_target_circle**(&v_x, &v_y, &v_z, &$d\Omega$, aim$_x$, aim$_y$, aim$_z$, r_f). Generates a random vector (v_x, v_y, v_z), of the same length as (aim$_x$, aim$_y$, aim$_z$), which is targeted at a *disk* centered at (aim$_x$, aim$_y$, aim$_z$) with radius r_f (in meters), and perpendicular to the *aim* vector.. All directions that intersect the circle are chosen with equal probability. The solid angle of the circle as seen from the position of the x-ray is returned in $d\Omega$. This routine was previously called **randvec_target_sphere** (which still works).

- **randvec_target_rect_angular**(&v_x, &v_y, &v_z, &$d\Omega$, aim$_x$, aim$_y$, aim$_z$,h, w, Rot) does the same as randvec_target_circle but targetting at a rectangle with angular dimensions h and w (in **radians**, not in degrees as other angles). The rotation matrix *Rot* is the coordinate system orientation in the absolute frame, usually ROT_A_CURRENT_COMP.

- **randvec_target_rect**(&v_x, &v_y, &v_z, &$d\Omega$, aim$_x$, aim$_y$, aim$_z$,$height, width, Rot$) is the same as randvec_target_rect_angular but *height* and *width* dimensions are given in meters. This function is useful to e.g. target at a guide entry window or analyzer blade.

B.2. Reading a data file into a vector/matrix (Table input, `read_table-lib`)

The `read_table-lib` library provides functionalities for reading text (and binary) data files. To use this library, add a `%include "read_table-lib"` in your component definition DECLARE or SHARE section. Tables are structures of type `t_Table` (see `read_table-lib.h` file for details):

```
1    /* t_Table structure (most important members) */
2    double *data;      /* Use Table_Index(Table, i j) to extract [i,j]
         element */
3    long    rows;      /* number of rows */
4    long    columns;   /* number of columns */
5    char   *header;    /* the header with comments */
6    char   *filename;  /* file name or title */
7    double  min_x;     /* minimum value of 1st column/vector */
8    double  max_x;     /* maximum value of 1st column/vector */
```

Available functions to read *a single* vector/matrix are:

- **Table_Init**(&*Table*, *rows*, *columns*) returns an allocated Table structure. Use $rows = columns = 0$ not to allocate memory and return an empty table. Calls to Table_Init are *optional*, since initialization is being performed by other functions already.

- **Table_Read**(&*Table*, *filename*, *block*) reads numerical block number *block* (0 to catenate all) data from *text* file *filename* into *Table*, which is as well initialized in the process. The block number changes when the numerical data changes its size,

or a comment is encoutered (lines starting by '`# ; % /`'). If the data could not be read, then $Table.data$ is NULL and $Table.rows = 0$. You may then try to read it using Table_Read_Offset_Binary. Return value is the number of elements read.

- **Table_Read_Offset**(&$Table$, $filename$, $block$, &$offset$, n_{rows}) does the same as Table_Read except that it starts at offset $offset$ (0 means begining of file) and reads n_{rows} lines (0 for all). The $offset$ is returned as the final offset reached after reading the n_{rows} lines.

- **Table_Read_Offset_Binary**(&$Table$, $filename$, $type$, $block$, &$offset$, n_{rows}, $n_{columns}$) does the same as Table_Read_Offset, but also specifies the $type$ of the file (may be "float" or "double"), the number n_{rows} of rows to read, each of them having $n_{columns}$ elements. No text header should be present in the file.

- **Table_Rebin**(&$Table$) rebins all $Table$ rows with increasing, evenly spaced first column (index 0), e.g. before using Table_Value. Linear interpolation is performed for all other columns. The number of bins for the rebinned table is determined from the smallest first column step.

- **Table_Info**($Table$) print information about the table $Table$.

- **Table_Index**($Table, m, n$) reads the $Table[m][n]$ element.

- **Table_Value**($Table, x, n$) looks for the closest x value in the first column (index 0), and extracts in this row the n-th element (starting from 0). The first column is thus the 'x' axis for the data.

- **Table_Free**(&$Table$) free allocated memory blocks.

- **Table_Value2d**($Table$, X, Y) Uses 2D linear interpolation on a Table, from (X,Y) coordinates and returns the corresponding value.

Available functions to read *an array* of vectors/matrices in a *text* file are:

- **Table_Read_Array**($File$, &n) read and split $file$ into as many blocks as necessary and return a `t_Table` array. Each block contains a single vector/matrix. This only works for text files. The number of blocks is put into n.

- **Table_Free_Array**(&$Table$) free the $Table$ array.

- **Table_Info_Array**(&$Table$) display information about all data blocks.

The format of text files is free. Lines starting by '`# ; % /`' characters are considered to be comments, and stored in $Table.header$. Data blocks are vectors and matrices. Block numbers are counted starting from 1, and changing when a comment is found, or the column number changes. For instance, the file 'MCXTRACE/data/Rh.txt' (Material data for Rhodium) looks like:

```
#Rh (Z  45)
#Atomic weight: A[r]  102.9055
#Nominal density: rho 1.2390E+01
#    14 edges. Edge energies (keV): K 23.22, L-I 3.41, ...
#-------------------------------------------------------
#Form Factors, Attenuation and Scattering Cross-sections
#Z=45, E = 0.001 - 433 keV
#
#     E(keV)     f[1]       f[2]     [mu/rho]
 1.069000E-02  1.894E+00  4.805E+00  1.838E+05
 1.142761E-02  2.096E+00  5.102E+00  1.826E+05
 1.221612E-02  2.327E+00  5.401E+00  1.808E+05
 1.305903E-02  2.585E+00  5.699E+00  1.784E+05
 1.396010E-02  2.872E+00  5.993E+00  1.755E+05
   ...
```

Binary files should be of type "float" (i.e. REAL*32) and "double" (i.e. REAL*64), and should *not* contain text header lines. These files are platform dependent (little or big endian).

The *filename* is first searched into the current directory (and all user additional locations specified using the `-I` option, see the 'Running McXtrace ' chapter in the User Manual), and if not found, in the `data` sub-directory of the `MCXTRACE` library location. This way, you do not need to have local copies of the McXtrace Library Data files (see table 1.1).

A usage example for this library part may be:

```
t_Table Table;        // declare a t_Table structure
char file[]="Rh.txt"; // a file name
double x,y;

Table_Read(&Table, file, 1); // initialize and read the first numerical
    block
Table_Info(Table);           // display table informations
...
x = Table_Index(Table, 2,5); // read the 3rd row, 6th column element
                             // of the table. Indexes start at zero in C
                             .
y = Table_Value(Table, 1.45,1); // look for value 1.45 in 1st column (x
    axis)
                             // and extract 2nd column value of that row
Table_Free(&Table);          // free allocated memory for table
```

Additionally, if the block number (3rd) argument of **Table_Read** is 0, all blocks will be catenated. The **Table_Value** function assumes that the 'x' axis is the first column (index 0). Other functions are used the same way with a few additional parameters, e.g. specifying an offset for reading files, or reading binary data.

This other example for text files shows how to read many data blocks:

```
t_Table *Table;       // declare a t_Table structure array
long     n;
double y;
```

```
4
5   Table = Table_Read_Array("file.dat", &n); // initialize and read the all
        numerical block
6   n = Table_Info_Array(Table);      // display informations for all blocks (
        also returns n)
7
8   y = Table_Index(Table[0], 2,5);   // read in 1st block the 3rd row, 6th
        column element
9                                     // ONLY use Table[i] with i < n !
10  Table_Free_Array(Table);          // free allocated memory for Table
```

You may look into, for instance, the source files for **Lens_parab** or **Filter** for other implementation examples.

B.3. Constants for unit conversion etc.

The following predefined constants are useful for conversion between units

Name	Value	Conversion from	Conversion to
DEG2RAD	$2\pi/360$	Degrees	Radians
RAD2DEG	$360/(2\pi)$	Radians	Degrees
MIN2RAD	$2\pi/(360 \cdot 60)$	Minutes of arc	Radians
RAD2MIN	$(360 \cdot 60)/(2\pi)$	Radians	Minutes of arc
FWHM2RMS	$1/\sqrt{8\log(2)}$	Full width half maximum	Root mean square (standard deviation)
RMS2FWHM	$\sqrt{8\log(2)}$	Root mean square (standard deviation)	Full width half maximum
PI	3.14159265...	π	
CELE	1.602176487e-19	Elementary charge (C)	
M_C	299792458	Speed of light in vacuum (m/s)	
MELECTRON	9.10938291e-31	Electron mass (kg)	
ALPHA	7.2973525698e-3	Fine structure constant, α	
RE	2.8179402894e-5	Thomson scattering length (AA)	
E2K	0.506773091264796	Energy (keV)	Wavenumber (1/AA)
K2E	1.97326972808327	Wavenumber (1/AA)	Energy (keV)

C. The McXtrace terminology

This is a short explanation of phrases and terms which have a specific meaning within McXtrace. We have tried to keep the list as short as possible running the calculated risk that the reader may occasionally miss an explanation. In this case, you are more than welcome to contact the McXtrace core team.

- **Arm** A generic McXtrace component which defines a frame of reference for other components.
- **Component** One unit (*e.g.* optical element) in an x-ray beamline. These are considered as Types of elements to be instantiated in an Instrument description.
- **Component Instance** A named Component (of a given Type) inserted in an Instrument description.
- **Definition parameter** An input parameter for a component. For example the radius of a sample component or the divergence of a collimator. Technically, a definition parameter is translated into a literal constant, which prevents it from being edited at runtime.
- **Input parameter** For a component, either a definition parameter or a setting parameter. These parameters are supplied by the user to define the characteristics of the particular instance of the component definition. For an instrument, a parameter that can be changed at simulation run-time.
- **Instrument** An assembly of McXtrace components defining an x-ray beamline.
- **Kernel** The McXtrace language definition and the associated compiler
- **Output parameter** An output parameter for a component. For example the counts in a monitor. An output parameter may be accessed from the instrument in which the component is used using `MC_GETPAR`.
- **Run-time** C code, contained in the files `mcxtrace-r.c` and `mcxtrace-r.h` included in the McXtrace distribution, that declare functions and variables used by the generated simulations.
- **Setting parameter** Similar to a definition parameter, but with the restriction that the type of the parameter must be declared unless it is a number. In technical terms, a setting parameter is translated into an actual variable (as opposed to a definition parameter) which may be dynamically updated.

Bibliography

[Mcx] See http://www.mcxtrace.org (cit. on pp. 12, 16).

[Tra] See http://trac.edgewall.com (cit. on p. 12).

[Nis] See http://www.nist.gov/pml/data/ffast/index.cfm (cit. on pp. 15, 46, 58).

[Mcz] See http://trac.mccode.org/report (cit. on p. 16).

[Sha] See http://www.esrf.eu/computing/scientific/raytracing/ (cit. on p. 17).

[ANM11] J Als-Nielsen and D McMorrow. *Elements of modern X-ray physics*. Wiley, 2011 (cit. on p. 23).

[Bac75] G.E. Bacon. *Neutron Diffraction*. Oxford University Press, 1975 (cit. on pp. 68, 80).

[Bau+07] Sondes Trabelsi Bauer et al. "Simulation of X-ray beamlines with the new ray tracing tool¡ i¿ X¡/i¿ Trace". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 582.1 (2007), pp. 90–92 (cit. on p. 17).

[Bea67] J A Bearden. "X-ray wavelengths". In: *Reviews of Modern Physics* 39.1 (1967), p. 78 (cit. on p. 35).

[BK+13] Erik Bergbäck Knudsen et al. "McXtrace". English. In: *Journal of Applied Crystallography* 46.3 (2013), pp. 679–696. ISSN: 00218898, 16005767. DOI: 10.1107/S0021889813007991 (cit. on p. 40).

[Ber17] Mads Bertelsen. "Software for simulation and design of neutron scattering instrumentation". PhD thesis. University of Copenhagen, 2017 (cit. on pp. 349, 354).

[BJL17] Mads Bertelsen, Henrik Jacobsen, and Kim Lefmann. "Expanding the McStas sample simulation logic with McStas Union components". In: *Review of Scientific Instruments* 88 (2017), p. 085106. DOI: 10.1063/1.4994697 (cit. on p. 349).

[GRR92] Grimmett, G. R., and Stirzakerand D. R. *Probability and Random Processes, 2nd Edition*. Clarendon Press, Oxford, 1992 (cit. on p. 18).

[Jam80] F. James. In: *Rep. Prog. Phys.* 43 (1980), p. 1145 (cit. on pp. 17, 18, 22).

[Knu+14] E. B. Knudsen et al. *Users' and Programmers' Guide to the X-ray tracing Package McXtrace, version 1.2*. DTU Physics, 2014 (cit. on pp. 12, 16, 67).

[Kra23] H. A. Kramers. “XCIII. On the theory of X-ray absorption and of the continuous X-ray spectrum”. In: *Philosophical Magazine Series 6* 46.275 (Nov. 1923), pp. 836–871. ISSN: 1941-5982. DOI: 10.1080/14786442308565244 (cit. on p. 35).

[KO79] MO Krause and JH Oliver. “Natural widths of atomic K and L levels, Ka X-ray lines and several KLL Auger lines”. In: *J. Phys. Chem. Ref. Data* 8 (1979), pp. 329–338 (cit. on p. 35).

[LN99] K. Lefmann and K. Nielsen. “McStas, a general software package for neutron ray-tracing simulations”. In: *Neutron News* 10 (1999), pp. 20–23. DOI: 10.1080/10448639908233684 (cit. on p. 12).

[MN98] Makoto Matsumoto and Takuji Nishimura. “Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator”. In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 8.1 (1998), pp. 3–30 (cit. on p. 362).

[Pre+86] W. H. Press et al. *Numerical Recipes in C*. Cambridge University Press, 1986 (cit. on p. 362).

[Rio+11] M Sanchez del Rio et al. “SHADOW3: a new version of the synchrotron X-ray optics modelling package”. In: *Journal of Synchrotron Radiation* 18.5 (2011), p. 0 (cit. on pp. 17, 112).

[Sch08] F Schäfers. “The BESSY raytrace program RAY”. In: *Modern Developments in X-Ray and Neutron Optics* (2008), pp. 9–41 (cit. on p. 17).

[SKS96] A Snigirev, V Kohn, and I Snigireva. “A compound refractive lens for focusing high-energy X-rays”. In: *Nature* (1996) (cit. on p. 50).

[Squ78] G.L. Squires. *Thermal Neutron Scattering*. Cambridge University Press, 1978 (cit. on p. 81).

[TK01] Takashi Tanaka and Hideo Kitamura. “*SPECTRA*: a synchrotron radiation calculation code”. In: *Journal of Synchrotron Radiation* 8.6 (2001), pp. 1221–1228. DOI: 10.1107/S090904950101425X (cit. on p. 24).

[WCC94] C Welnak, G J Chen, and F Cerrina. “SHADOW: A synchrotron radiation and X-ray optics simulation tool”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 347.1-3 (1994), pp. 344–347 (cit. on p. 17).

Index

AstroX, **327**

Bugs, 16

Concentric components, 78
Contrib, **296**

Data files, 15
DFT, 90
Diffraction, 73, 78, 88

Environment variable
  BROWSER, 16
  MCXTAS, 365
  MCXTRACE, 16, 358

imaging, 44
Incoherent elastic scattering, 88

Keyword
  %include, 358
  EXTEND, 16, 23, 111, 359
  GROUP, 359
  MCDISPLAY, 359
  OUTPUT PARAMETERS, 111
  SHARE, 358

Library, **358**
  Components
    astrox, 327
    contrib, 296
    data, 15, 365
    misc, 112
    monitors, 91
    obsolete, 326
    optics, 38
    samples, 67
    sasmodels, 137
    share, 358
    sources, 23
    union, 117
  mcxtrace-r, *see* Library/Run-time
  read_table-lib (Read_Table), 15, **363**
  Run-time, **358**
    ABSORB, 358
    ALLOW_BACKPROP, 358
    MC_GETPAR, 111, 359
    NAME_CURRENT_COMP, 359
    POS_A_COMP, 359
    POS_A_CURRENT_COMP, 359
    PROP_DL, 358
    PROP_Z0, 358
    RESTORE_XRAY, 359
    ROT_A_COMP, 359
    ROT_A_CURRENT_COMP, 359
    SCATTER, 358
    SCATTERED, 359
    STORE_XRAY, 359
  run-time
    PROP_X0, 358
    PROP_Y0, 358
  Shared, *see* Library/Components/share

Monitor
  4*pi*, 98
  photon counter, 92
Monitors, **91**
  Banana shape, 109
  Custom monitoring (user variables, Monitor_nD), 110
  Energy monitor, 93
  Position sensitive detector (PSD), 95
  Position sensitive detector (PSD), coherent, 97
  Position Sensitive Detector (PSD), Energy Selective, 99
  Position sensitive monitor recording mean energy, 110
  PowerPSD, 100
  The All-in-One monitor (Monitor_nD), 106
  Wavelength monitor, 94
Monte Carlo method, 17
  Accuracy, 22

Direction focusing, 20
Stratified sampling, 21
Multiple scattering, 88

Obsolete, **326**
Optics, **38**, **50**, **56**
Beam stop, 42
Filter, 46
lens, 48, 51–53, 55
mask, 44
mirror, 58, 59, 61
Mirror plane, 62
multilayer, mirror, 62
Point in space (Arm, Optical bench, Coordinate system), 38
Progress bar, 113
Slit, 40

Removed x-ray events, 359
Removed xray events, 15

Sample environments, 78
Samples, **67**
Absorption, 71
Dilute colloid medium, 73
liquid, diffraction, 90
Powder, multiple diffraction line, 78
SAXS, 73
Single crystal diffraction, 88
SasView, **137**
Small angle scattering, 73
Sources, **23**
Continuous source with specified divergence, 30
Flat surface source, 28
Point source, 26
Shadow, 115
Source_gaussian, 32
Virtual source, 116
X-ray tube laboratory source, 34

time resolved, 90
Tomography, 71
Tools
mcdoc, 16

Union, **117**, **349–357**
algorithm, 354
exit volume, 354
geometry, 352
masks, 354
master, 354
materials, 352
processes, 351
Union_init, 351
Union_stop, 351