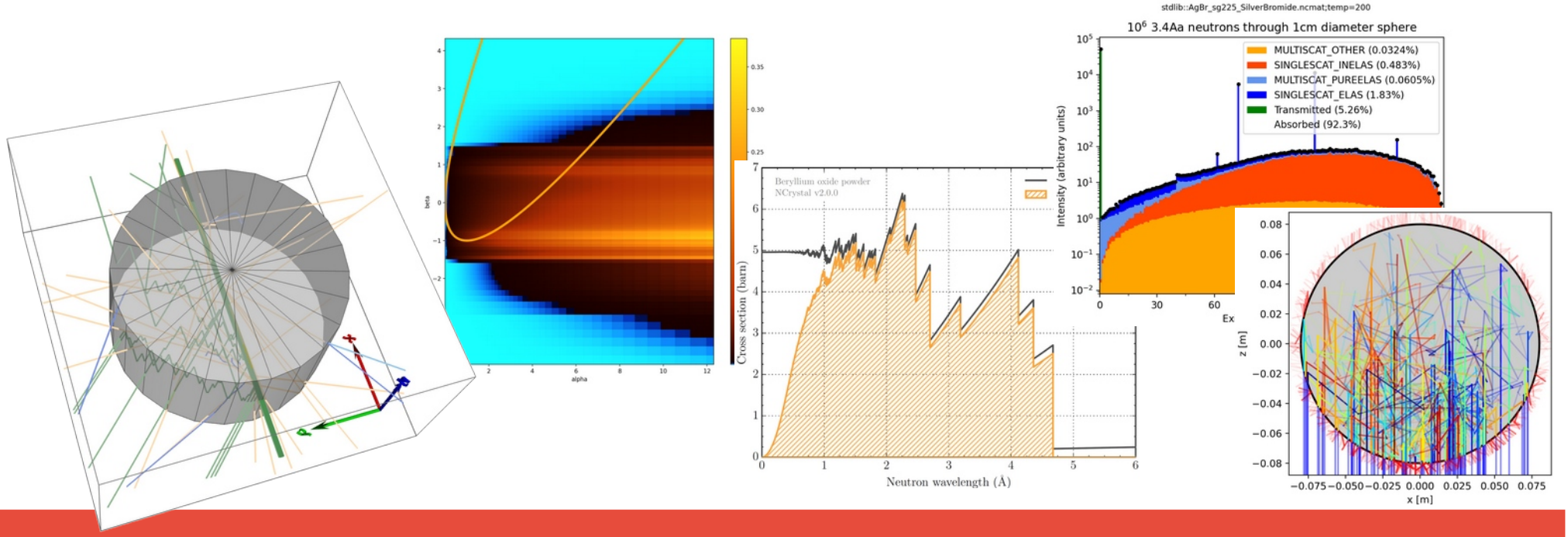


NCrystal

Update on the NCrystal project for modelling thermal neutron interactions

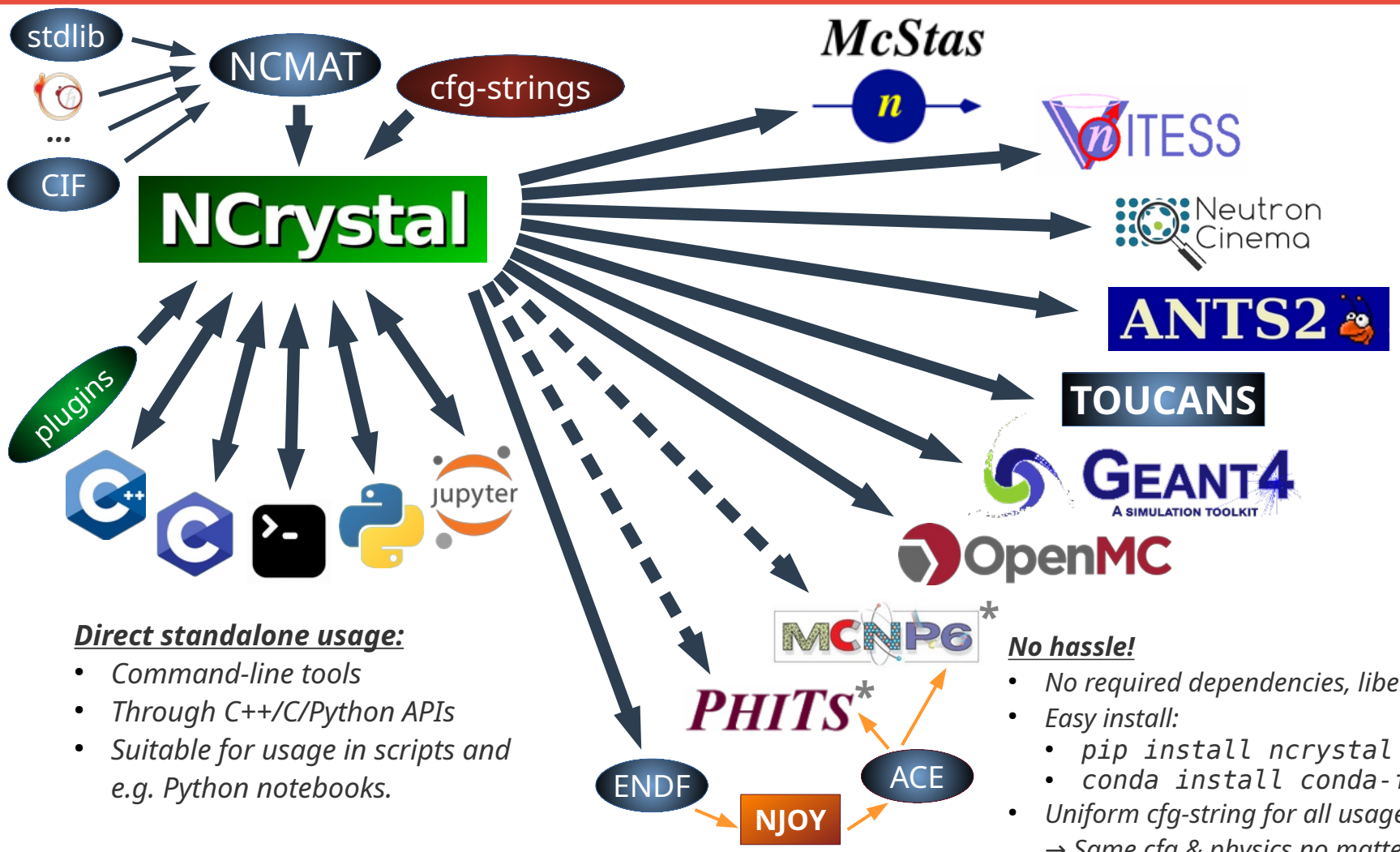
ICANS XXV
Clarion Hotel Malmö Live
April 13-17, 2026

T. Kittelmann†, J. I. Marquez Damian‡, D. D. Di Julio‡, P. Willendrup†*, S. Xu§
† European Spallation Source ERIC, Data Management and Scientific Computing centre
‡ European Spallation Source ERIC, Spallation Physics Group
* Technical University of Denmark, Physics
§ Lund University, Department of Physics, Division of Synchrotron Radiation Research

EUROPEAN SPALLATION SOURCE · DTU · brightness · HighNess · APRENDE

NCrystal: Open Source backend providing thermal neutron scattering to MC codes

NCrystal
Thermal Neutron Transport



Direct standalone usage:

- *Command-line tools*
- *Through C++/C/Python APIs*
- *Suitable for usage in scripts and e.g. Python notebooks.*

No hassle!

- *No required dependencies, liberal license (Apache 2)*
- *Easy install:*
  - `pip install ncrystal`
  - `conda install conda-forge::ncrystal`
- *Uniform cfg-string for all usage*
  → Same cfg & physics no matter usage context
- *Multi-thread safe, support Linux/macOS/BSD/Windows*

***: Can not share bindings for projects that are not open-source**

Core NCrystal engine

Building blocks for entire NCrystal ecosystem

NCrystal
Thermal Neutron Transport

Inputs

cfg-strings

```
"Al_sg225.ncmat"
"Y2SiO5_sg15_YSO.ncmat;temp=200K"
"Rubber_C5H8.ncmat;temp=200;comp=inelas"
"LiquidHeavyWaterD2O_T293.6K.ncmat"
"""phases<0.1*PbS_sg225_LeadSulfide.ncmat
   &0.9*Epoxy_Araldite506_C18H20O3.ncmat>"""
"gasmix::0.7xCO2+0.3xAr/1.5atm/250K"
```

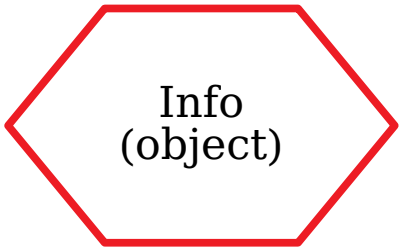
data

Preferred native format is the NCMAT (.ncmat) format.

Can reside in actual files, or as content in memory.

Can be hand-crafted, auto-converted, or generated on-demand by plugins.

Outputs



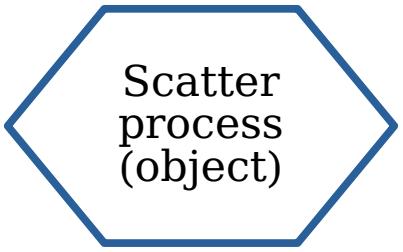
Densities, compositions

Phases

Scattering lengths

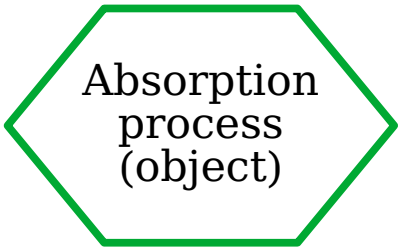
Crystal structures, HKL factors

Dynamics info like phonon DOS



Provides integrated scattering cross sections when queried with neutron state.

Can perform Monte Carlo sampling of outgoing states.

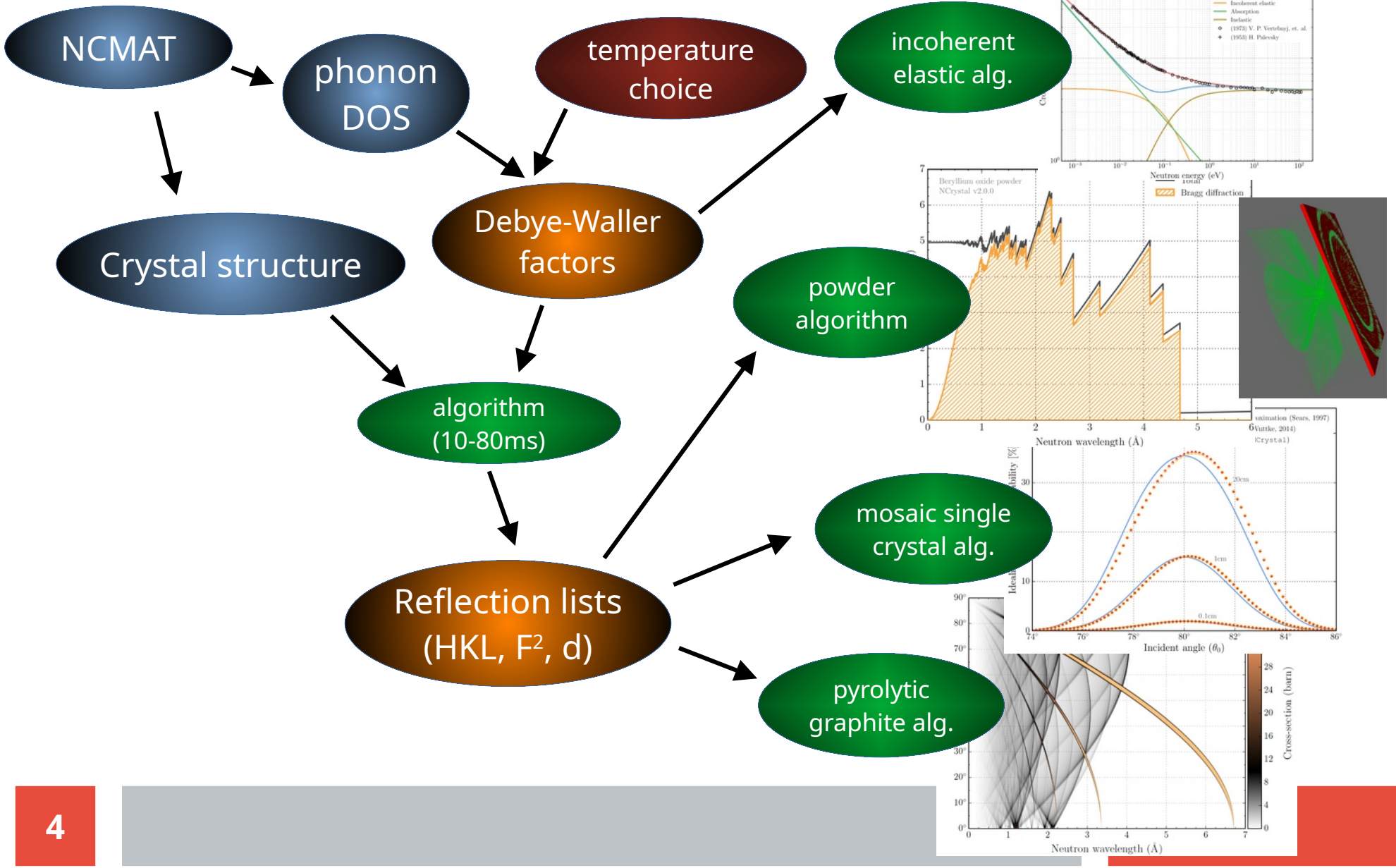


Provides integrated absorption cross sections when queried with neutron state.

Presently only implements simple 1/v model.

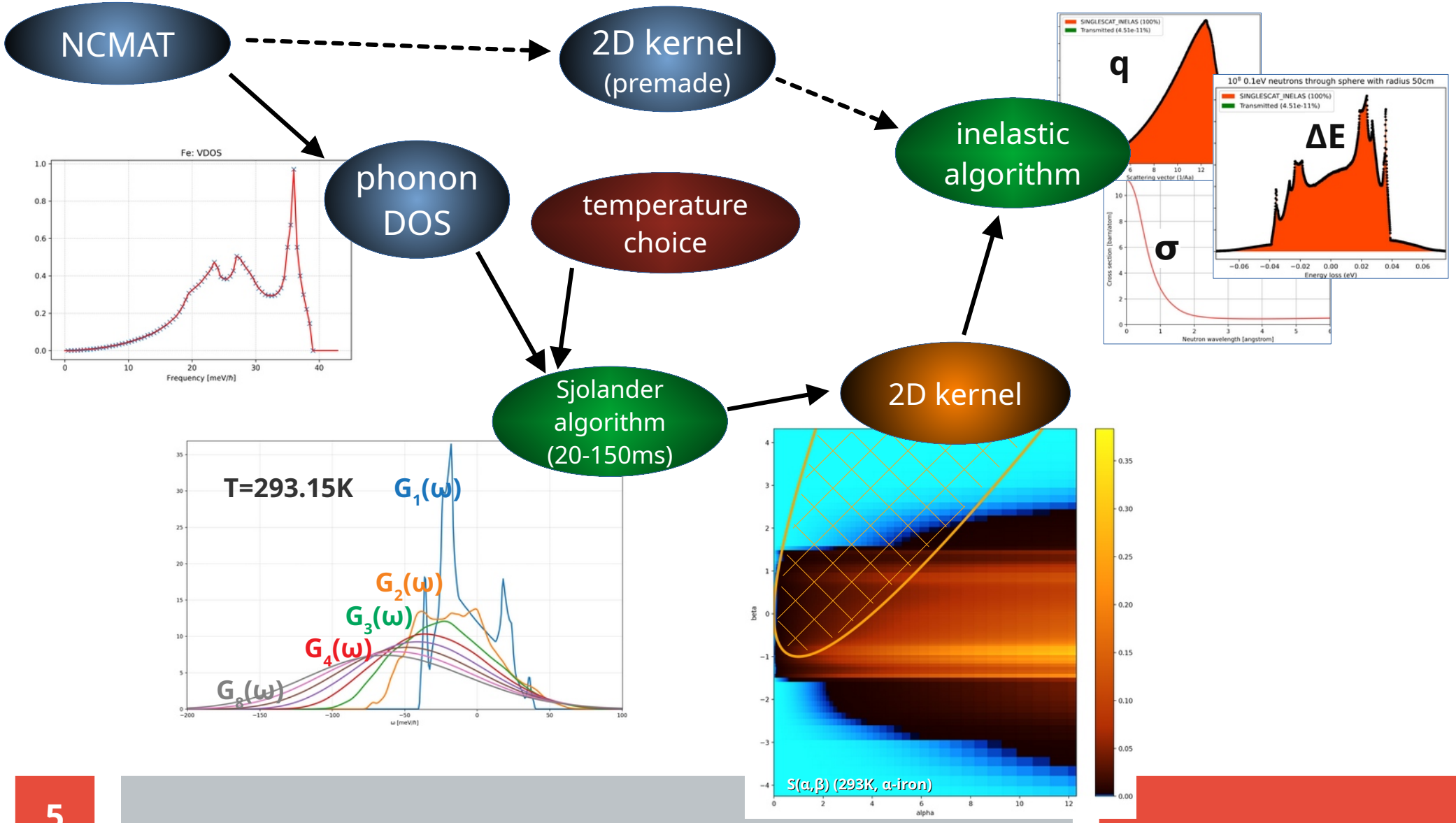
Brief overview: Elastic processes

NCrystal
Thermal Neutron Transport



Brief overview: Inelastic processes

NCrystal
Thermal Neutron Transport



Other physics

NCrystal
Thermal Neutron Transport

- **Will not go into details here. But also supported:**
  - Multiphase materials, enrichments, impurities, ...
  - SANS (only hard spheres model out of the box for now)
  - Easy one-liner materials:

```
"gasmix::BF3/2atm/25C/B_is_0.95_B10_0.05_B11"
"gasmix::CO2"
"gasmix::He/He_is_He3/10bar"
"gasmix::air"
"gasmix::air/-10C/0.8atm/0.30relhumidity"
"solid::Gd2O3/7.07gcm3/Gd_is_0.9_Gd157_0.1_Gd155"
```

- **Plugins for immature/experimental/project-specific capabilities:**
  - Extra materials (e.g. for novel moderators).
  - Extinction, nanodiamonds, magnetic interactions, coherent-inelastic models, ...
  - Find plugins at https://github.com/mctools/ncrystal/wiki/CuratedPlugins

NCrystal data library

Browse online: https://github.com/mctools/ncrystal/wiki/Data-library
Browse with command: `$> nctool -b`

NCrystal
Thermal Neutron Transport

AcrylicGlass_C5O2H8.ncmat
AgBr_sg225_SilverBromide.ncmat
Ag_sg225.ncmat
Al2O3_sg167_Corundum.ncmat
Al4C3_sg166_AluminiumCarbide.ncmat
AlN_sg186_AluminumNitride.ncmat
Al_sg225.ncmat
Ar_Gas_STP.ncmat
Au_sg225.ncmat
BaF2_sg225_BariumFluoride.ncmat
BaO_sg225_BariumOxide.ncmat
Ba_sg229.ncmat
Be3N2_sg206_BerylliumNitride.ncmat
BeF2_sg152_Beryllium_Fluoride.ncmat
BeO_sg186.ncmat
Be_sg194.ncmat
Bi_sg166.ncmat
CaCO3_sg62_Aragonite.ncmat
CaF2_sg225_CalciumFlouride.ncmat
CaH2_sg62_CalciumHydride.ncmat
CaO2H2_sg164_CalciumHydroxide.ncmat
CaO_sg225_CalciumOxide.ncmat
Ca_sg225.ncmat
Ca_sg229_Calcium-gamma.ncmat
CaSiO3_sg2_Wollastonite.ncmat
CeO2_sg225_CeriumOxide.ncmat
Cr_sg229.ncmat
C_sg194_pyrolytic_graphite.ncmat
C_sg227_Diamond.ncmat
Cu2O_sg224_Cuprite.ncmat
Cu_sg225.ncmat
Dy2O3_sg206_DysprosiumOxide.ncmat
Epoxy_Araldite506_C18H20O3.ncmat
Fe_sg225_Iron-gamma.ncmat
Fe_sg229_Iron-alpha.ncmat
GaN_sg186_GalliumNitride.ncmat
GaSe_sg194_GalliumSelenide.ncmat
Ge3Bi4O12_sg220_BismuthGermanate.ncmat
Ge_sg227.ncmat
He_Gas_STP.ncmat
HfO2_sg14_HafniumOxide.ncmat
Ho2O3_sg206_HolmiumOxide.ncmat
Kapton_C22H10N2O5.ncmat
KBr_sg225_PotassiumBromide.ncmat
KF_sg225_PotassiumFlouride.ncmat
KOH_sg4_PotassiumHydroxide.ncmat
Kr_Gas_STP.ncmat
K_sg229.ncmat
LaBr3_sg176_LanthanumBromide.ncmat
Li2O_sg225_LithiumOxide.ncmat
Li3N_sg191_LithiumNitride.ncmat
LiF_sg225_LithiumFlouride.ncmat
LiH_sg225_LithiumHydride.ncmat
LiquidHeavyWaterD2O_T293.6K.ncmat
LiquidWaterH2O_T293.6K.ncmat
Lu2O3_sg206_LutetiumOxide.ncmat
Lu2SiO5_sg15.ncmat
Mg2SiO4_sg62_MagnesiumSilicate.ncmat
MgAl2O4_sg227_MAS.ncmat
MgCO3_sg167_MagnesiumCarbonate.ncmat
MgD2_sg136_MagnesiumDeuteride.ncmat
MgF2_sg136_MagnesiumFlouride.ncmat
MgH2_sg136_MagnesiumHydride.ncmat
MgO2H2_sg164_MagnesiumHydroxide.ncmat
MgO_sg225_Periclase.ncmat
Mg_sg194.ncmat
Mo_sg229.ncmat
Na4Si3Al3O12Cl_sg218_Sodalite.ncmat
NaBr_sg225_SodiumBromide.ncmat
NaCl_sg225_SodiumChloride.ncmat
NaF_sg225_SodiumFlouride.ncmat
NaI_sg225_SodiumIodide.ncmat
Na_sg229.ncmat
Nb_sg229.ncmat
Ne_Gas_STP.ncmat
Ni_sg225.ncmat
Nylon11_C11H21NO.ncmat
Nylon12_C12H23NO.ncmat
Nylon610_C16H30N2O2.ncmat
Nylon66or6_C12H22N2O2.ncmat
PbF2-beta_sg225_BetaLeadFlouride.ncmat
PbO-alpha_sg129_Litharge.ncmat
PbO-beta_sg57_Massicot.ncmat
Pb_sg225.ncmat
PbS_sg225_LeadSulfide.ncmat
Pd_sg225.ncmat
PEEK_C19H12O3.ncmat
Polycarbonate_C16O3H14.ncmat
Polyester_C10H8O4.ncmat
Polyethylene_CH2.ncmat
Polylactide_C3H4O2.ncmat
Polypropylene_C3H6.ncmat
Polystyrene_C8H8.ncmat
Pt_sg225.ncmat
PVC_C2H3Cl.ncmat
Rb_sg229.ncmat
Rubber_C5H8.ncmat
Sc_sg194.ncmat
SiC-beta_sg216_BetaSiliconCarbide.ncmat
SiO2-alpha_sg154_AlphaQuartz.ncmat
SiO2-beta_sg180_BetaQuartz.ncmat
Si_sg227.ncmat
Sn_sg141.ncmat
SrF2_sg225_StrontiumFluoride.ncmat
SrH2_sg62_StrontiumHydride.ncmat
Sr_sg225.ncmat
Th3N4_sg166_ThoriumNitride.ncmat
ThO2_sg225_ThoriumDioxide.ncmat
Th_sg225.ncmat
TiO2_sg136_Rutile.ncmat
TiO2_sg141_Anatase.ncmat
Ti_sg194.ncmat
TlBr_sg221_ThaliumBromide.ncmat
Tm2O3_sg206_ThuliumOxide.ncmat
UF6_sg62_UraniumHexaflouride.ncmat
UO2_sg225_UraniumDioxide.ncmat
void.ncmat
V_sg229.ncmat
W_sg229.ncmat
Xe_Gas_STP.ncmat
Y2O3_sg206_Yttrium_Oxide.ncmat
Y2SiO5_sg15_YSO.ncmat
Y3Al5O12_sg230_YAG.ncmat
Y_sg194.ncmat
ZnF2_sg136_ZincFlouride.ncmat
ZnO_sg186_ZincOxide.ncmat
Zn_sg194.ncmat
ZnS_sg216_Sphalerite.ncmat
ZrF4-beta_sg84.ncmat
ZrO2_sg137_Zirconia.ncmat
ZrO2_sg14_Zirconia.ncmat
Zr_sg194.ncmat

134 materials (v4.3.0):
Crystals (110), amorphous solids (16), liquids, gasses, ...

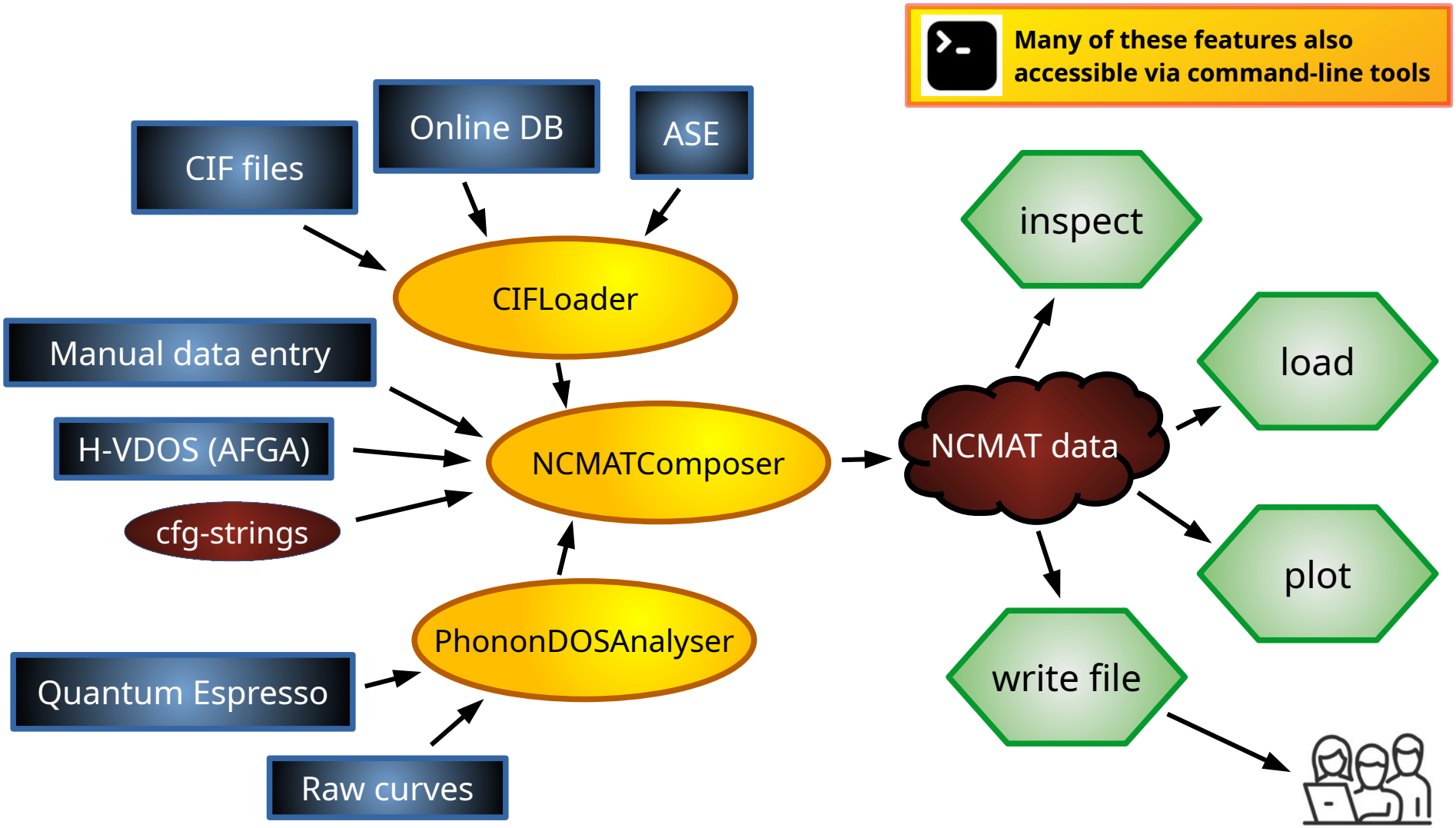
Small (few kB) file sizes:
- **Entire collection normally embedded in NCrystal shared library.**

Easy universal cfg:
"Al_sg225.ncmat;temp=250K"
"Rubber_C5H8.ncmat;comp=inelas"
- **Same physics in all applications!**

Rich Python API for material compositon

Jupyter notebooks tutorials: https://github.com/mctools/ncrystal-notebooks

NCrystal
Thermal Neutron Transport



Selected updates on recent work

Updates for development, testing, and platform support

NCrystal
Thermal Neutron Transport

New!

- Most intensive task last few years: large infrastructure revamp!
- **Introduced official support for Windows!**
- Development now completely @ GitHub!
- Using **standard development workflow** (PRs, branches, ...)
- Extensive unit & integration testing:
  - Easy to launch test suite locally (from repo or via **ncrystal-verify** pkg)
  - Runs in **GitHub CI with very extensive coverage**.
    - Platforms: Windows, macOS, Linux, FreeBSD.
    - Intel, ARM. Clang, GCC, MSVC, Intel OneAPI.
    - All modern C/C++/Python versions supported.

GitHub

Main message for users:

```
$> conda install -c conda-forge ncrystal
$> pip install ncrystal
```

now gives fully functional, updated, and pre-tested packages on all major platforms!!

Updated application bindings

- **All builds of McStas and OpenMC now work with NCrystal on all supported platforms.**
  - NCrystal installed via its own package as a dependency, and McStas or OpenMC finds it automatically at runtime by invoking the `ncrystal-config` command.
  - Dependency on NCrystal is runtime rather than buildtime for full flexibility.
- **McStas case was waiting for NCrystal@Windows which is now there.**
- **OpenMC is using the C++ API and a dedicated solution of stable "virtual APIs" had to be implemented**
  - OpenMC binary now finds and loads the NCrystal binary at runtime.
- **Geant4 bindings now live in standalone `ncrystal-geant4` package.**
  - A bit rough around the edges, but now supports multithreaded Geant4.

- **Other bindings recently updated or work in progress by community (i.e. not us) that we are aware of: VITESS, MCNP, FLUKA, ANTS3**

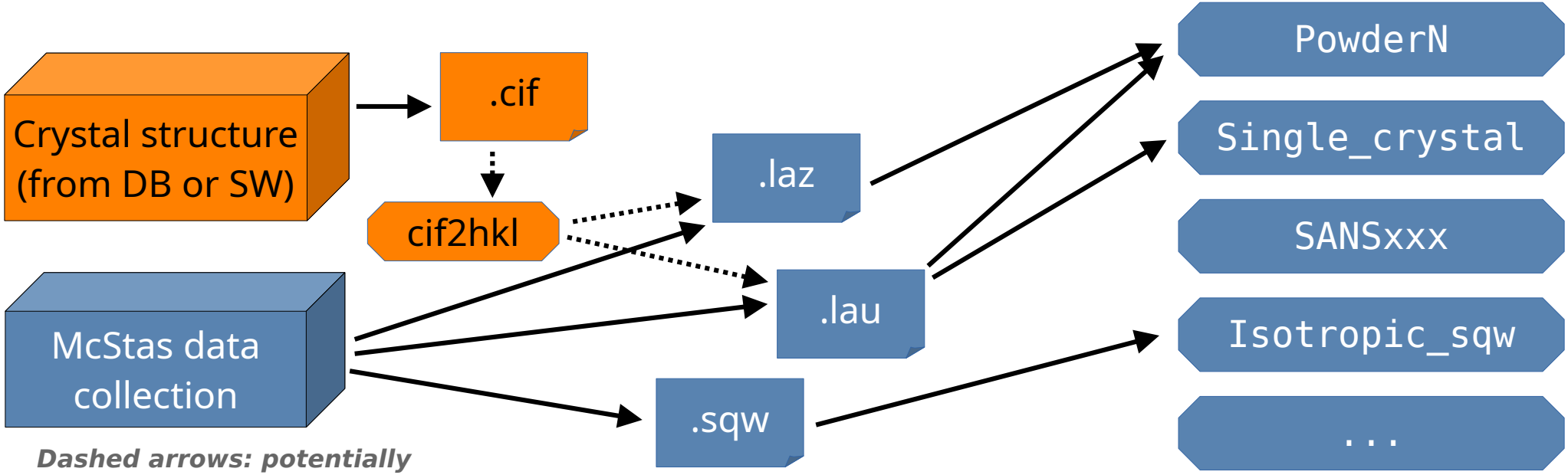
Materials in McStas eco-system

NCrystal integration continously improving

McStas

NCrystal

Thermal Neutron Transport



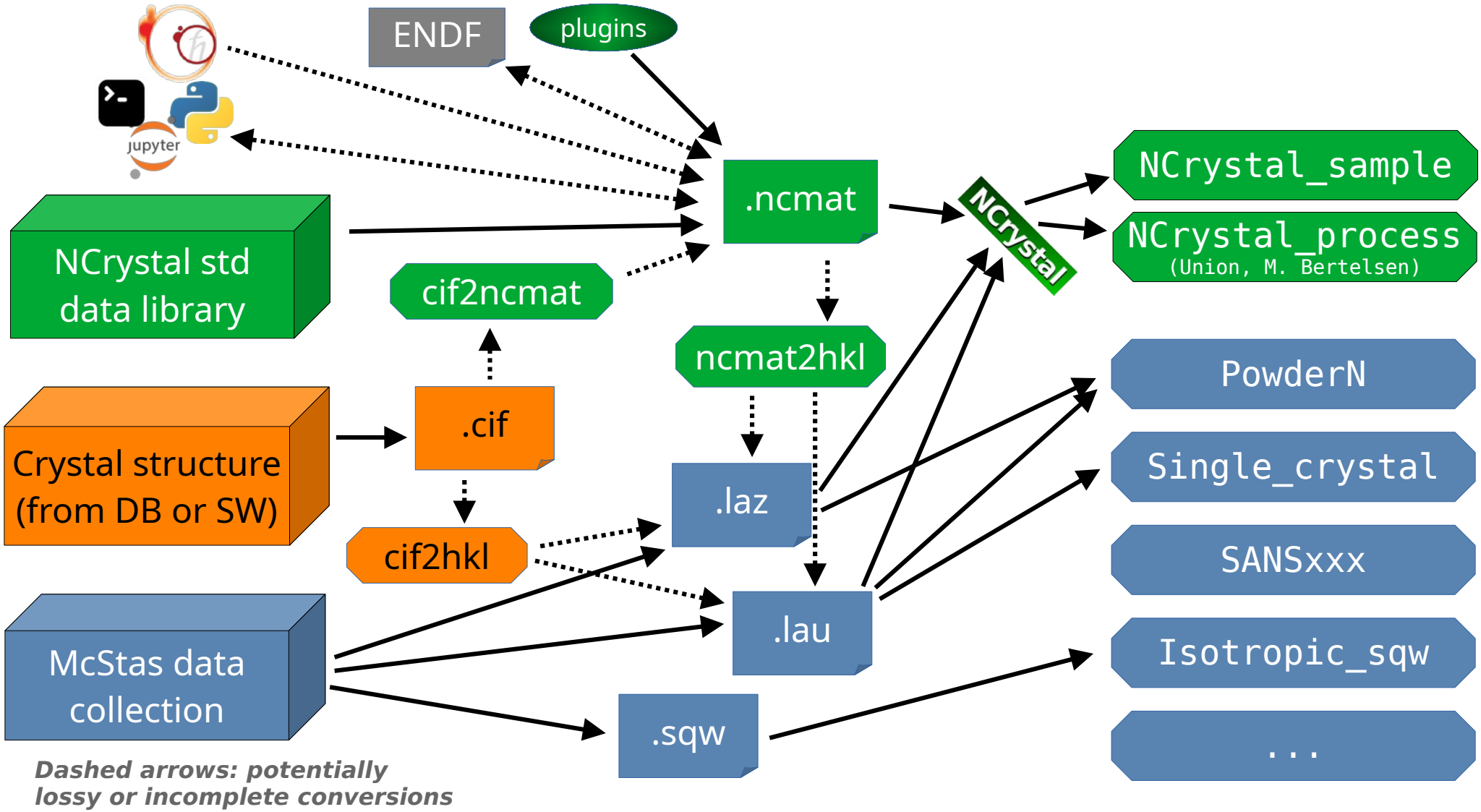
Dashed arrows: potentially lossy or incomplete conversions

Materials in McStas eco-system

NCrystal integration continously improving

McStas

NCrystal
Thermal Neutron Transport



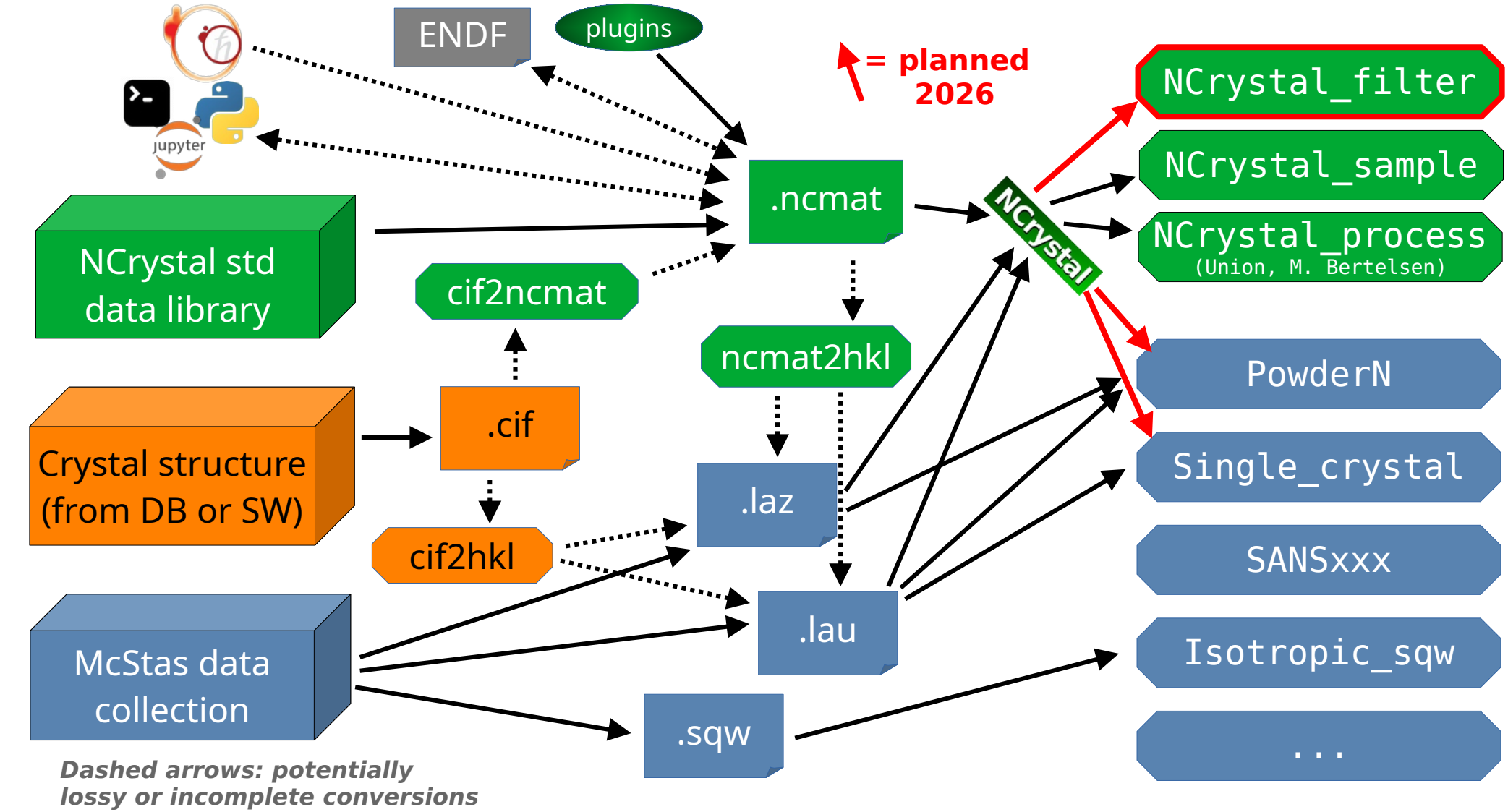
Dashed arrows: potentially lossy or incomplete conversions

Materials in McStas eco-system

NCrystal integration continously improving

McStas

NCrystal
Thermal Neutron Transport



Dashed arrows: potentially lossy or incomplete conversions

New ncmat2endf utility

APRENDE

NCrystal
Thermal Neutron Transport

Some users can not use NCrystal directly, instead relying on ENDF files.

ncmat2endf gives them access to NCrystal data library and material creation tool chain

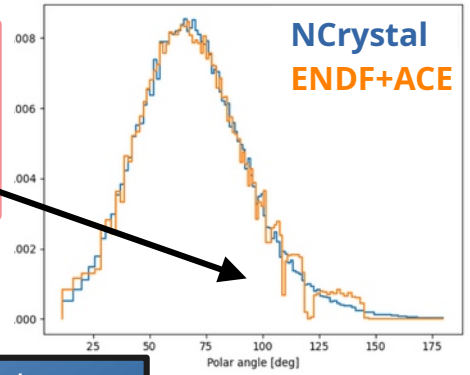
```
$> ncrystal_ncmat2endf "Al_sg225.ncmat;temp=293.6K" --name "Al-metal" -f --now
        -m alab:IAEA -m auth:"Marie Curie" -m matnum:Al:113 -m libname:IAEA-TSL
Initialise nuclear data...
Write ENDF file tsl_Al_in_Al-metal.endf ...
Files created:
  tsl_Al_in_Al-metal.endf : Al with fraction 1
Suggested material density: 2.698645518 g/cm^
```

```
1 from NCrystal.ncmat2endf import ncmat2endf
2 m = ncmat2endf.EndfMetaData()
3 m.update_from_dict({'alab':'IAEA', 'auth': 'Marie Curie',
4                     'matnum':{'Al':113}, 'libname':'IAEA-TSL'})
5 res = ncmat2endf.ncmat2endf('Al_sg225.ncmat;temp=293.6K',
6                             endf_metadata=m,
                              material_name='Al-metal', force=True)
```

```
Created with ncmat2endf                                          113 0  0    0
 1.300000+4 2.674975+1          -1          0          0          0 113 1451    1
 0.000000+0 0.000000+0           0          0          0          6 113 1451    2
 1.000000+0 5.000000+0           0          0         12          1 113 1451    3
 0.000000+0 0.000000+0           0          0         60          3 113 1451    4
 13-Al     IAEA        EVAL-APR26 Marie Curie                       113 1451    5
 REFERENCE             DIST-APR26 REV0-APR26                        113 1451    6
----IAEA-TSL           MATERIAL  113                                113 1451    7
-----THERMAL NEUTRON SCATTERING DATA                                113 1451    8
------ENDF-6 FORMAT                                                 113 1451    9
**************************************************************     113 1451   10
                                                                   113 1451   11
 This file was converted from the following NCrystal [1,2]         113 1451   12
 cfg-string:                                                       113 1451   13
                                                                   113 1451   14
                  "Al_sg225.ncmat;temp=293.6"                      113 1451   15
                                                                   113 1451   16
 using NCrystal 4.3.0                                              113 1451   17
 and endf-parserpy [3] 0.14.3                                      113 1451   18

                --- file continues ---
```

Warning: Not as high-fidelity as using NCrystal directly (nor as convenient).

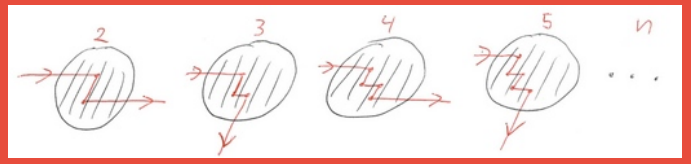


65 ENDF-6 files generated with NCrystal are now available in the JEFF 4.0 nuclear data library

See also ICANS poster (Tuesday):
"*Neutron scattering models and nuclear data at ESS*" J. I. Marquez Damian, et. al.)

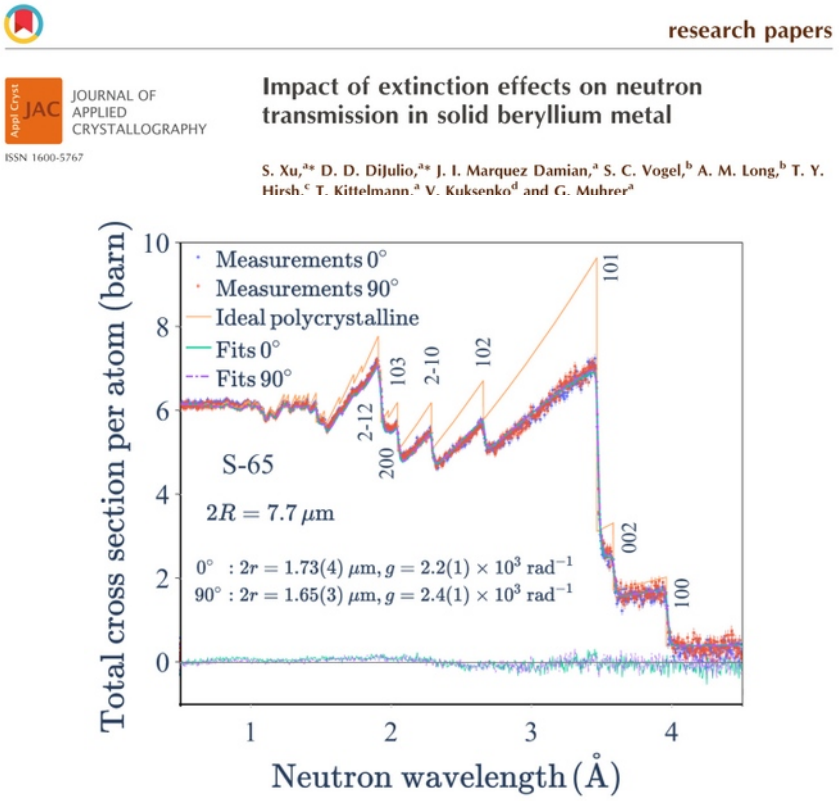
ncmat2endf uses endf-parserpy (G. Schnabel, IAEA)

Work in progress: Extinction models



NCrystal
Thermal Neutron Transport

- **Extinction reduce apparent strength of Bragg diffraction.**
  - Simplest case: A second scattering realigns neutron with original direction, "undoing" the initial scattering.
  - Much more complicated in practice: primary vs. secondary extinction, grain/domain geometry effects, etc.
  - Plethora of models to pick from, no single "fits-all" choice.
- **NCrystal "CrysExtn" plugin from ~1.5 years ago by S. Xu and collaborators implemented many such models.**
  - Already put to use, collecting experience with models.



See also ICANS contribution (Thursday):
"*Quantitative data analysis for neutron scattering experiments using NCrystal*" S. Xu, et. al.)

Also used by collaborators (T. Hirsch, et. al.) in a new toolkit for Bragg edge fitting: ***nbragg*** **https://nbragg.readthedocs.io/**

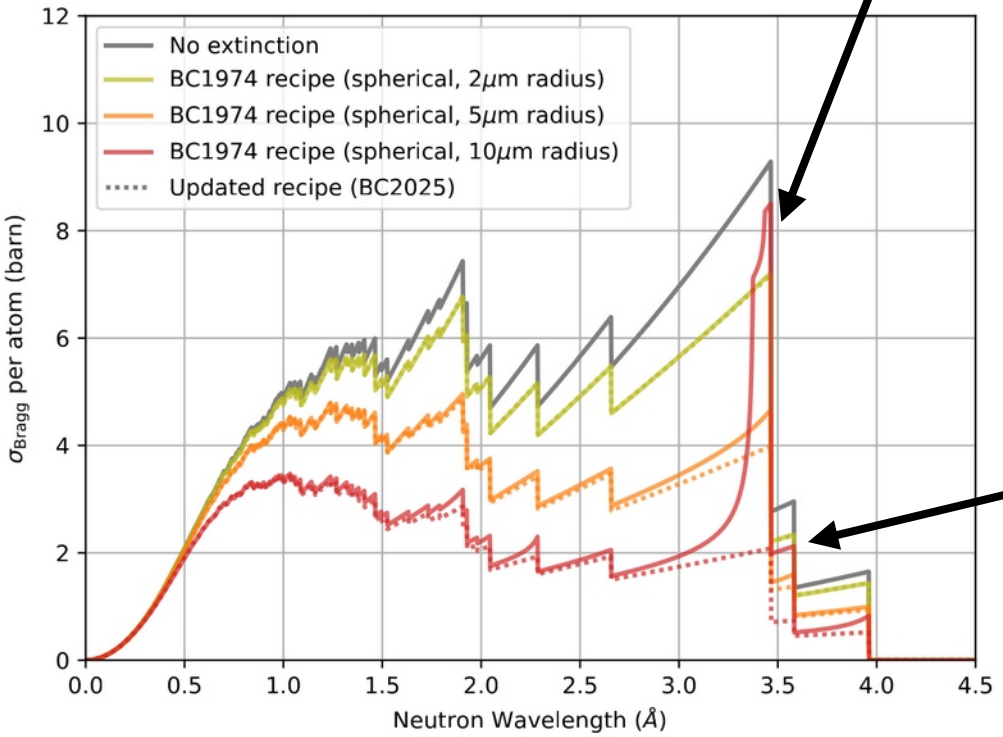
Developments ongoing now:
→ Curate & improve extinction models
→ integrate fully into NCrystal itself

Extinction model rabbit hole:

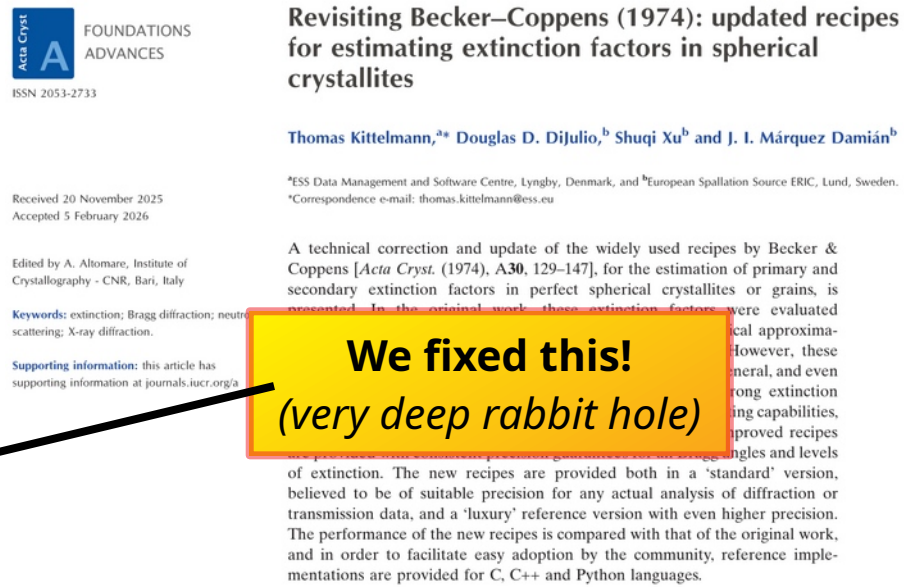
Updating the Becker-Coppens 1974 recipe for extinction.

NCrystal
Thermal Neutron Transport

After implementing the Becker-Coppens extinction recipes in NCrystal, we observed **non-sensical results** for strong extinction and large scattering angles.



Cause was actually in the numerical evaluation (not the physics models) of certain integrals in the original work.



FOUNDATIONS ADVANCES

ISSN 2053-2733

Received 20 November 2025
Accepted 5 February 2026

Edited by A. Altomare, Institute of Crystallography - CNR, Bari, Italy

Keywords: extinction; Bragg diffraction; neutron scattering; X-ray diffraction.

Supporting information: this article has supporting information at journals.iucr.org/a

Revisiting Becker–Coppens (1974): updated recipes for estimating extinction factors in spherical crystallites

Thomas Kittelmann,[a]* Douglas D. DiJulio,[b] Shuqi Xu[b] and J. I. Márquez Damián[b]

[a]ESS Data Management and Software Centre, Lyngby, Denmark, and [b]European Spallation Source ERIC, Lund, Sweden. *Correspondence e-mail: thomas.kittelmann@ess.eu

A technical correction and update of the widely used recipes by Becker & Coppens [*Acta Cryst.* (1974), A**30**, 129–147], for the estimation of primary and secondary extinction factors in perfect spherical crystallites or grains, is presented. In the original work, these extinction factors were evaluated ical approxima- However, these neral, and even rong extinction ing capabilities, proved recipes ngles and levels of extinction. The new recipes are provided both in a 'standard' version, believed to be of suitable precision for any actual analysis of diffraction or transmission data, and a 'luxury' reference version with even higher precision. The performance of the new recipes is compared with that of the original work, and in order to facilitate easy adoption by the community, reference implementations are provided for C, C++ and Python languages.

We fixed this!
(very deep rabbit hole)

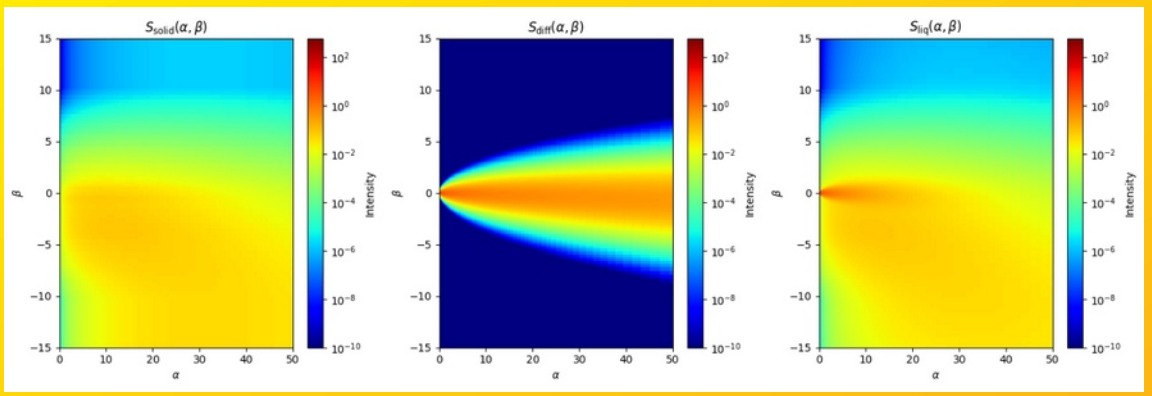
So now we can resume the work on extinction models in NCrystal.
Stay tuned!

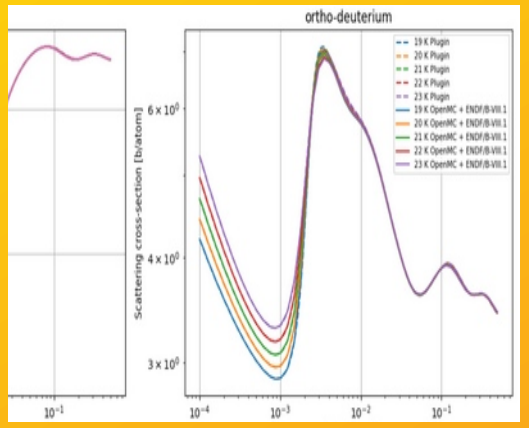
Work in progress: Liquids

NCrystal
Thermal Neutron Transport

- **Support for liquids is currently only possible via pre-built 2D kernels.**
- **Work is ongoing to improve this situation!**
- **Expected side-effect: improved modelling of coherent scattering in amorphous solids.**

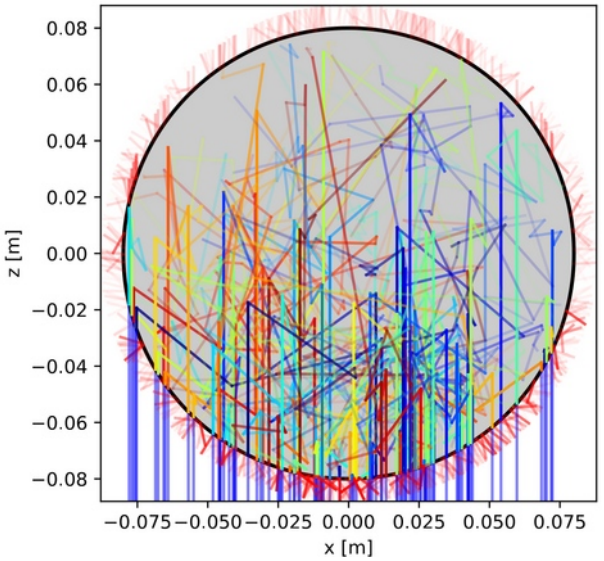
<u>Presentation after the break (this room):</u>
"*A liquids plugin for NCrystal*" (Douglas Di Julio, et. al.)

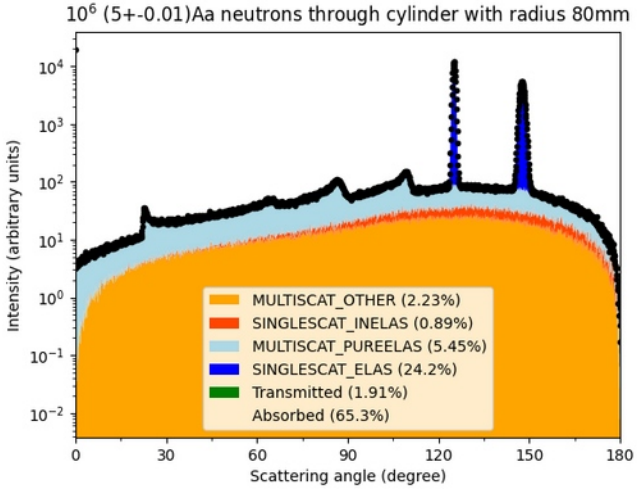




The NCrystal MiniMC

(from release 4.3.0, April 6, 2026)





The NCrystal MiniMC

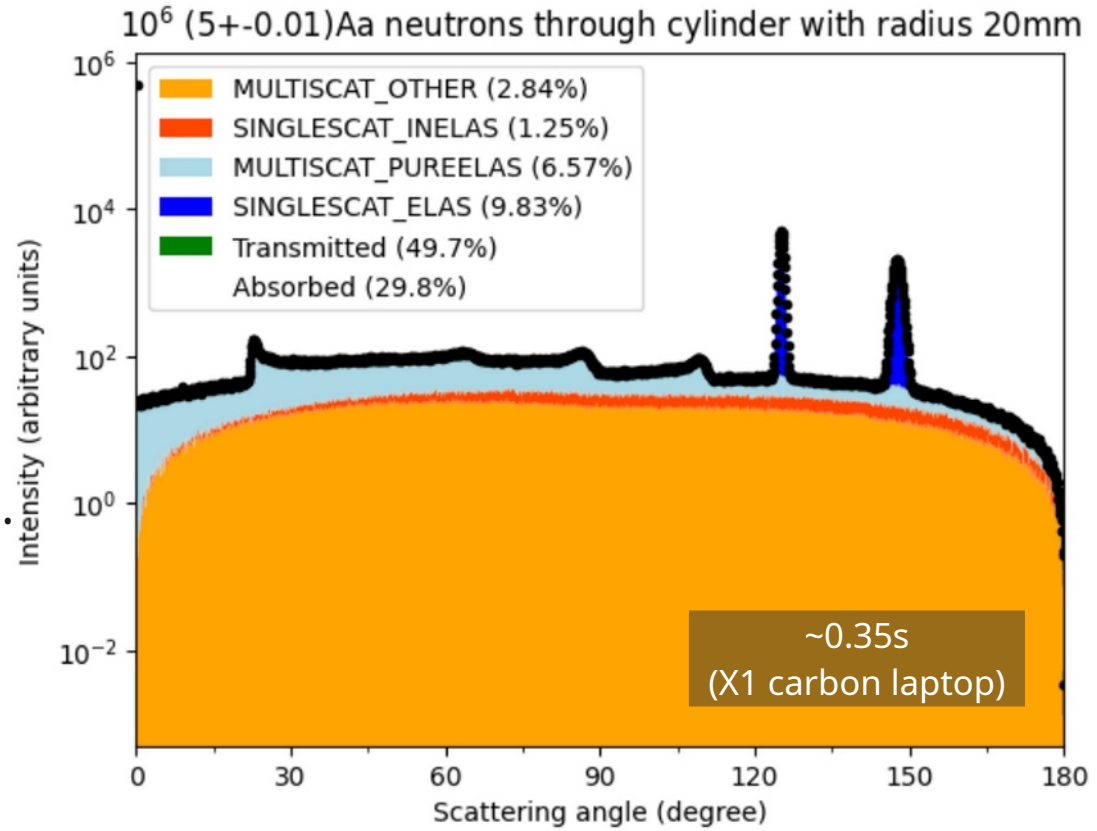
NCrystal
Thermal Neutron Transport

Ultimately NCrystal is intended as a backend providing thermal neutron physics for McStas/OpenMC/Geant4/VITESS/...

But simple simulations with geometry and multiple scattering effects can now be performed with just NCrystal, using the built-in "MiniMC".

- Provides a common reference.
- Tool for users for debugging physics, configurations, etc.
- Potential usage in analysis frameworks.
- Tool for developers for monitoring performance in CI, debugging issues, investigating physics performance.

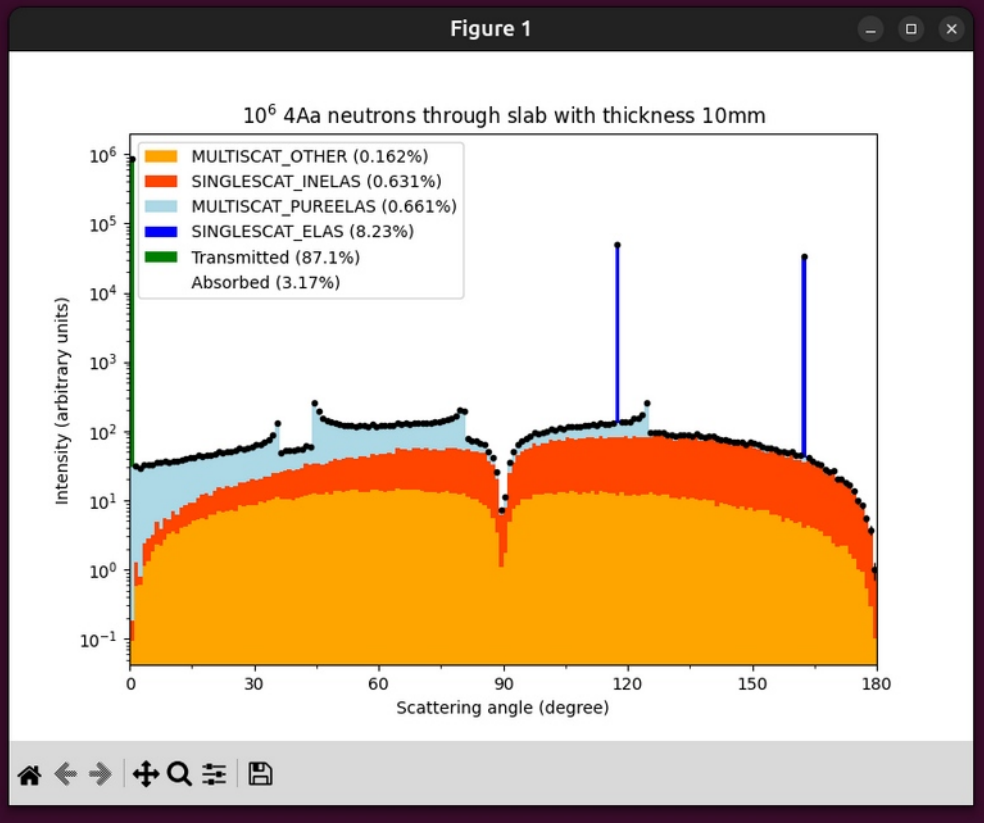
```
import NCrystal.minimc as mmc
results = mmc.run('ZnO_sg186_ZincOxide.ncmat;temp=300K',
                  geomcfg= 'cyl;r=0.02',
                  srccfg='circular;r=-0.02;wl=5+-0.01;n=1e6',
                  enginecfg='tally=q,de,theta;tallybins=+')
results.tally('theta').plot()
```



MiniMC: Command-line interface

NCrystal
Thermal Neutron Transport

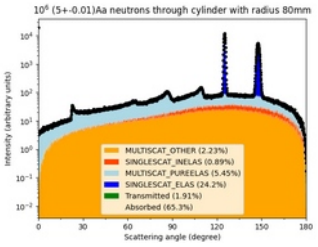
```
$> ncrystal_minimc "Al_sg225.ncmat;temp=300K" "4Aa pencil on 10mm slab"
Plotting tally: theta
```

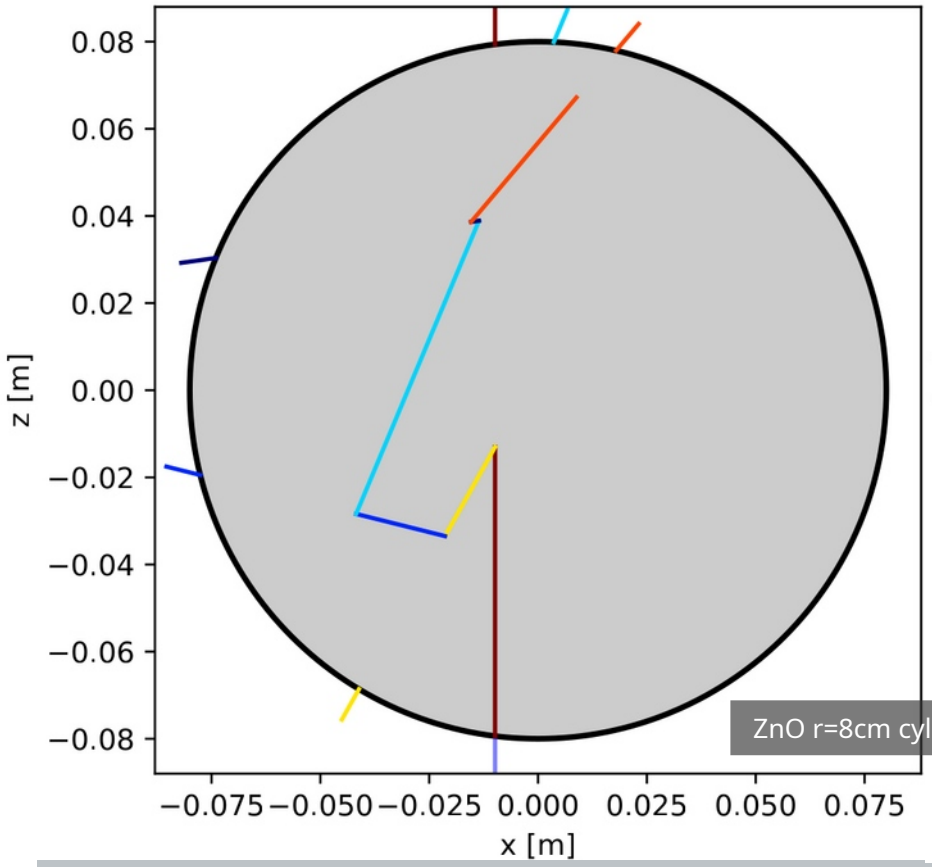


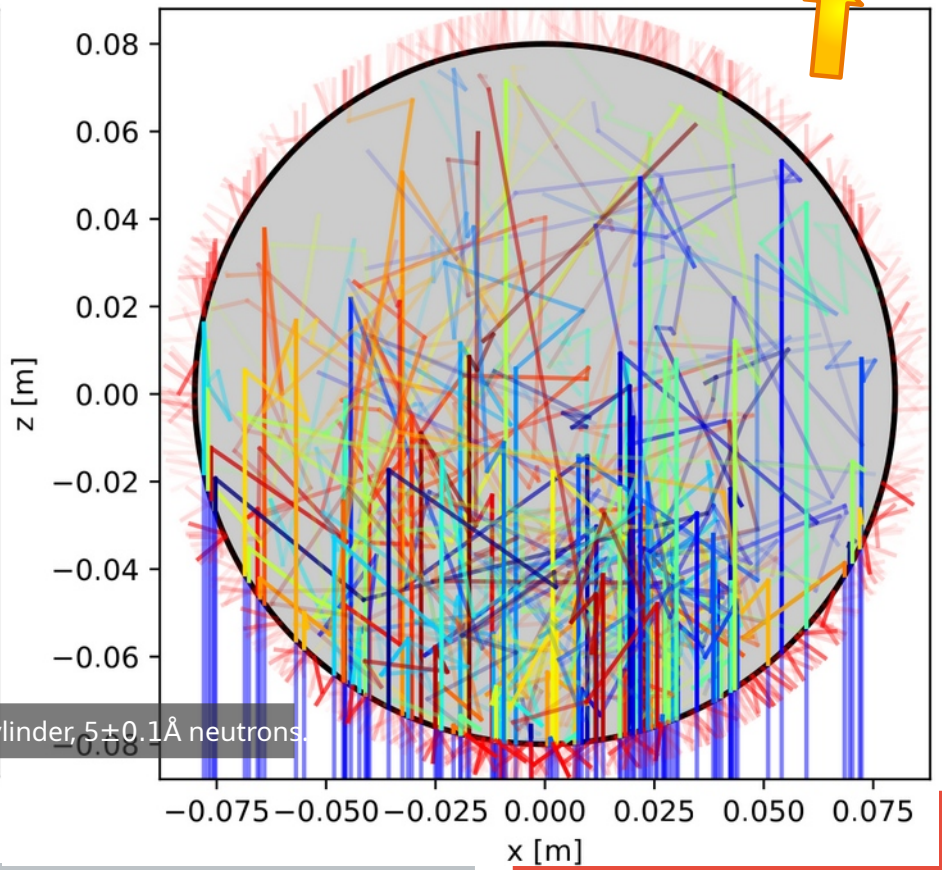
The MiniMC engine

NCrystal
Thermal Neutron Transport

→ Goal: Speed and low errors on multiple scattering effects.
→ Neutrons split many times in transmitted (tallied) and scattered parts.
→ Terminated via roulette scheme.
→ Fluxes correctly adjusted (also for absorption).







ZnO r=8cm cylinder 5±0.1Å neutrons.

MiniMC techniques for fast results

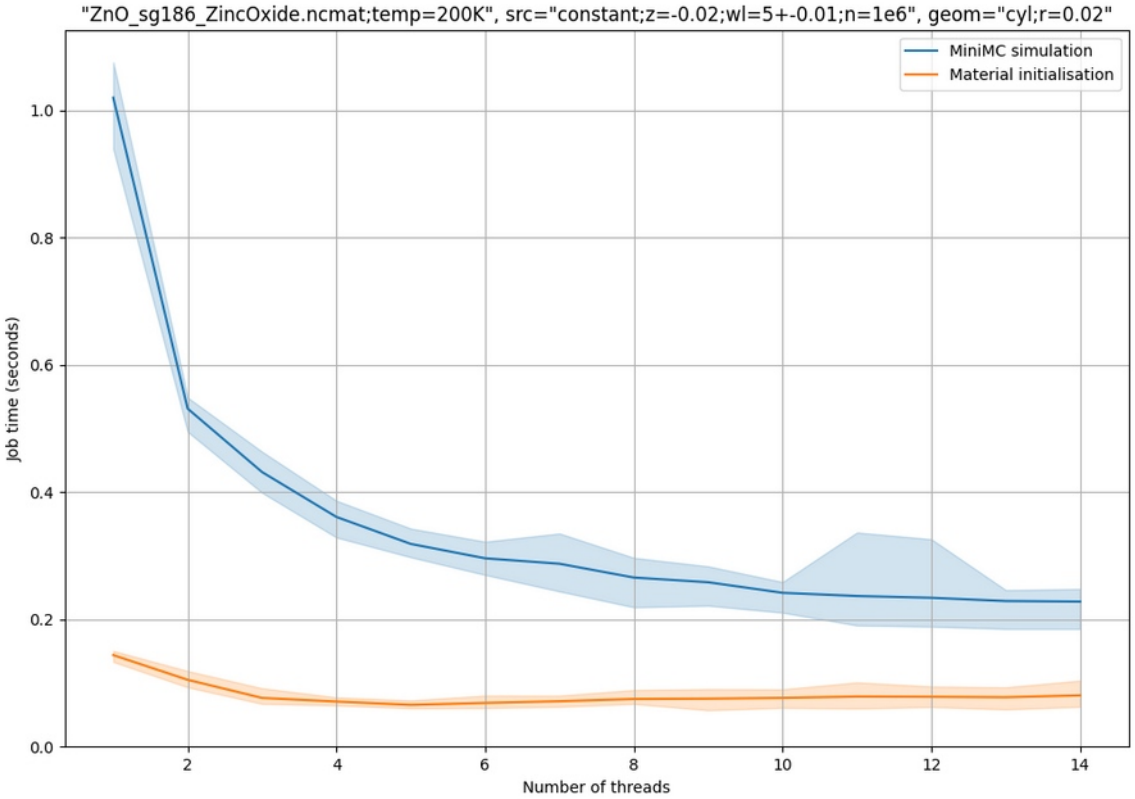
NCrystal
Thermal Neutron Transport

Several techniques:

→ Use splitting/forced-scatter+transmission/roulette (previous slide)

→ Handle “baskets” of 4096 neutrons at a time for vectorised (SIMD) code.

→ Use multi-threading.

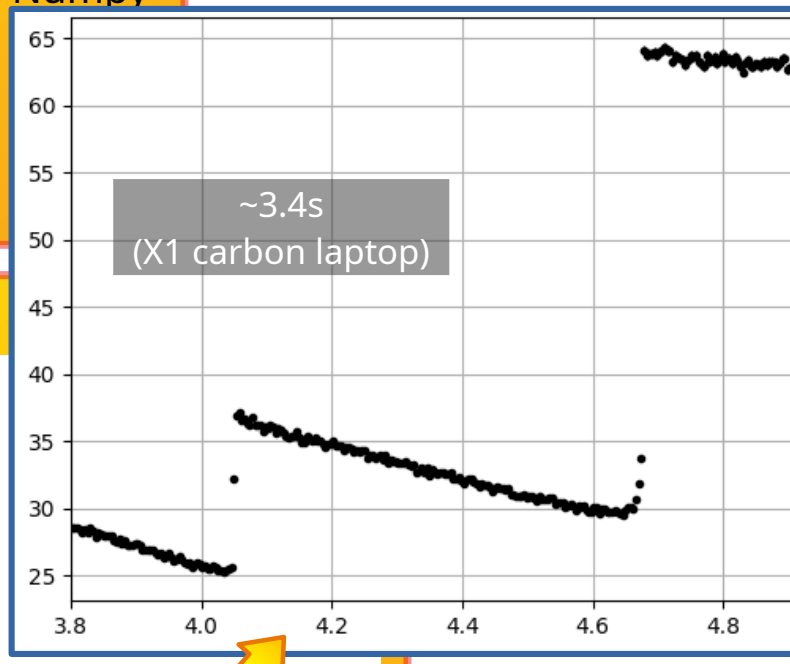


MiniMC: Custom tallying

NCrystal
Thermal Neutron Transport

Pre-defined tally quantities might not be enough for all users.
→ Allow accessing tallied neutrons directly from Python.
→ For efficiency, this is done via call-back function which gets long Numpy arrays of particle data.
→ But not so long as to cause memory issues :-)
→ Simulation can even be terminated dynamically by returning “True” from the callback function.

3.4s



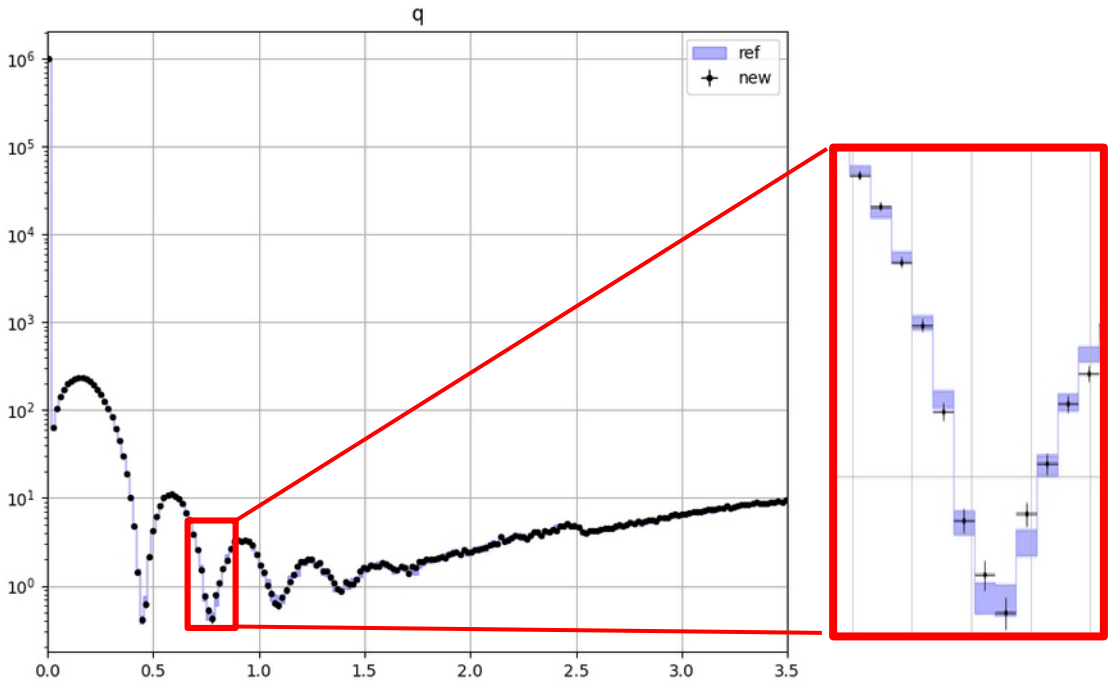
Example: P_transmission (<10deg) as a function of λ

```
from NCrystal import ekin2wl
import NCrystal.minimc as minimc
from NCrystal.hist import HistFiller1D
nbins, nsim, wllow, wlhigh = 300, 1e7, 3.8, 5.0
hist = HistFiller1D( nbins, wllow, wlhigh )
def cb( data ):
    mask = data['uz'] > 0.9848# = cos(10degree)
    hist.fill( ekin2wl(data['ekin0'][mask]), data['w'][mask] )
minimc.run( 'Al_sg225.ncmat', callback = cb,
            geomcfg = "slab;dz=0.05",
            srccfg = f"constant;z=-0.05;n={nsim};wl={wllow}-{wlhigh}" )
hist.to_hist1d().scale(100.0*nbins/nsim).plot(color='none')
```

MiniMC : used for CI monitoring

NCrystal
Thermal Neutron Transport

→ Having the MiniMC engine built in, allows our automated tests to monitor for statistically significant physics performance changes.
→ Highly unlikely that physics-breaking changes could sneak past this.
→ But without tons of spurious test failures on tiny changes.
→ Like all tests: shipped as part of ncrystal-verify package, so users can feel confident that their NCrystal installation is OK.



More info about NCrystal

NCrystal
Thermal Neutron Transport

Forum for questions/discussions, issue tracker, wiki, data library page at:
https://github.com/mctools/ncrystal

General info:
DOI 10.1016/j.cpc.2019.07.015

Details about elastic models:
DOI 10.1016/j.cpc.2021.108082

Embedded documentation:
Entire Python API has doc-strings.
Command-line tools have -h / --help flags

Details about inelastic sampling:
DOI 10.1016/j.jcp.2018.11.043

Jupyter tutorials at:
https://github.com/mctools/ncrystal-notebooks/

Better website to come
(hopefully in 2026)

BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

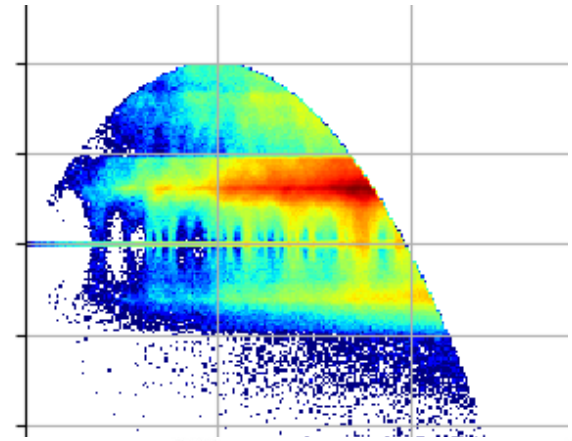
Backup slides

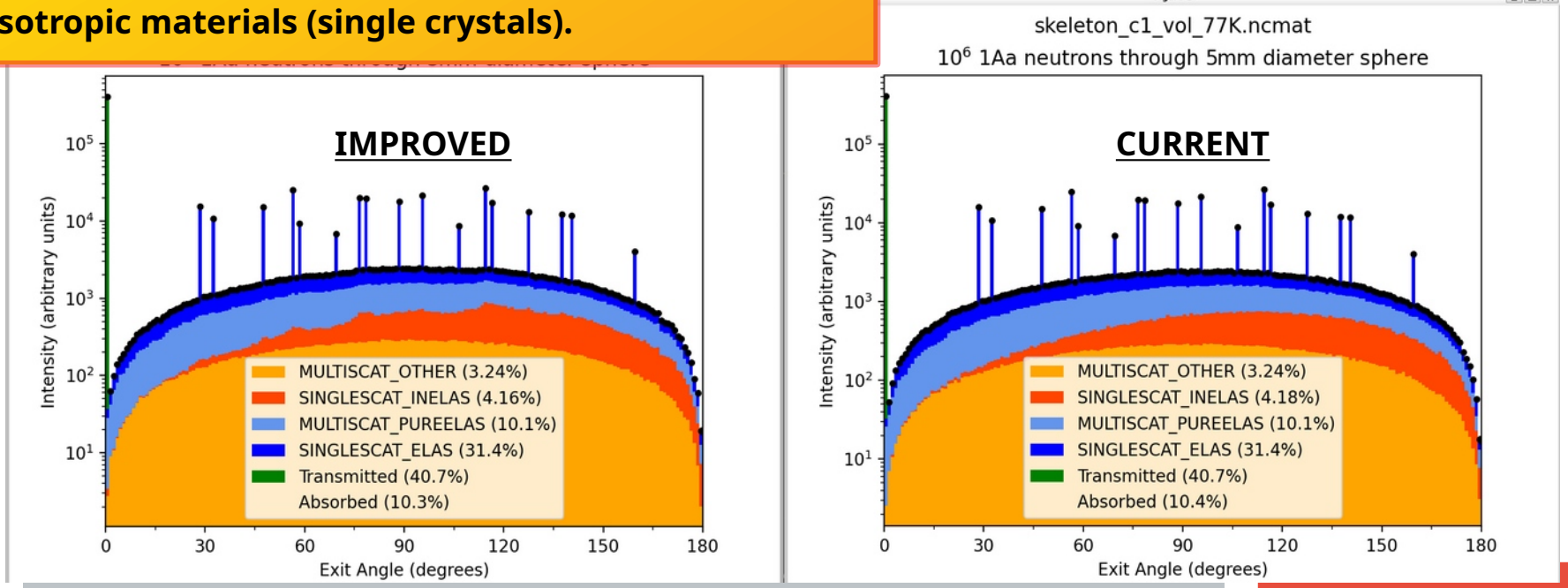
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

WIP: Coherent inelastic effects

NCrystal
Thermal Neutron Transport

- CSNS group (Xiao Xiao Cai, et al.) working on tools for providing high quality coherent-inelastic kernels for NCrystal.
- A proof-of-concept plugin for NCrystal for using those already exists.
- Such files would be temperature-specific and much larger, but for many use-cases this would be very interesting.
- Group also looking at extending this to non-isotropic materials (single crystals).





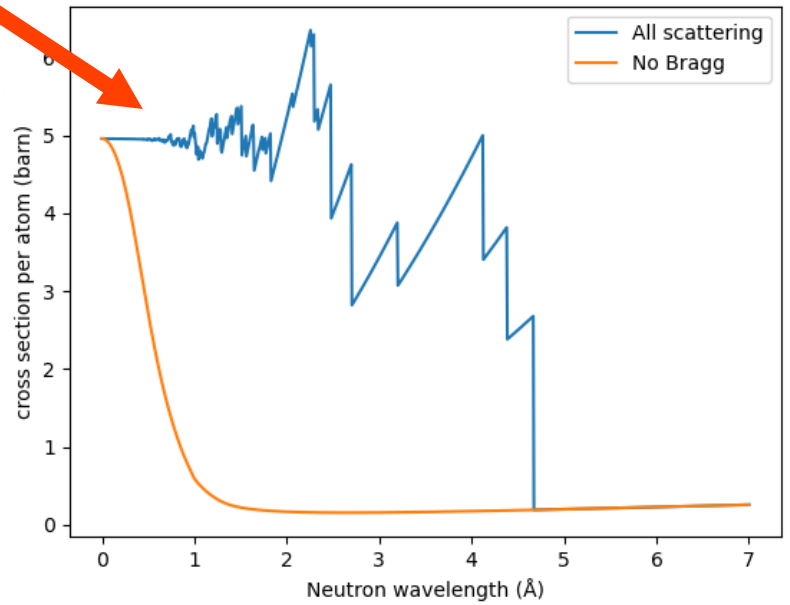
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

Python API

NCrystal
Thermal Neutron Transport

```
import NCrystal as NC
import matplotlib.pyplot as plt
import numpy
scBeO = NC.createScatter('BeO_sg186.ncmat')
scBeO_nobragg = NC.createScatter('BeO_sg186.ncmat;bragg=0')
wls = numpy.linspace(0.0,7.0,1000)
plt.plot( wls, scBeO.xsect(wl=wls), label='All scattering' )
plt.plot( wls, scBeO_nobragg.xsect(wl=wls), label='No Bragg' )
plt.xlabel('Neutron wavelength (Å)')
plt.ylabel('cross section per atom (barn)')
plt.legend()
plt.show()
```

Simply use NCrystal cfg-strings when creating materials



Jupyter tutorials available at:
https://github.com/mctools/ncrystal-notebooks/

conda install -c conda-forge ncrystal
or **pip install ncrystal**

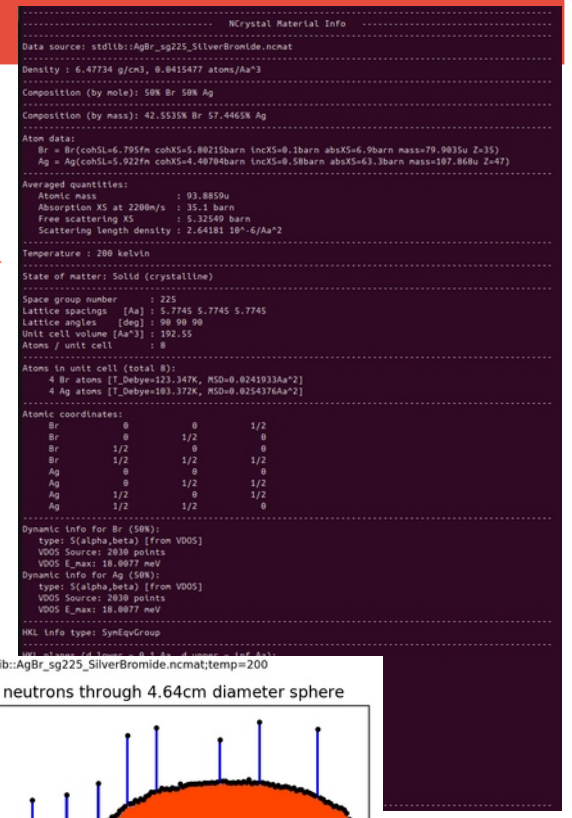
Command-line interface (CLI)

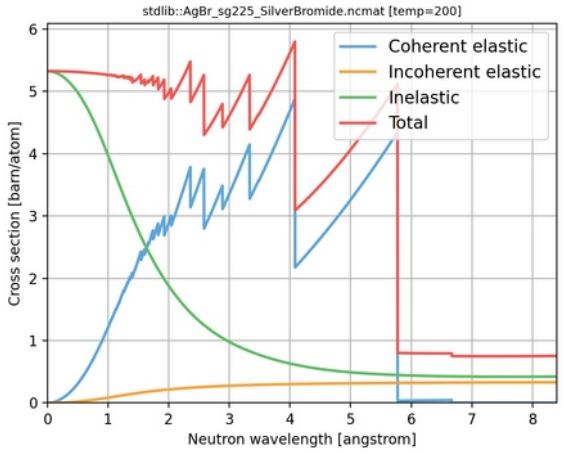
NCrystal
Thermal Neutron Transport

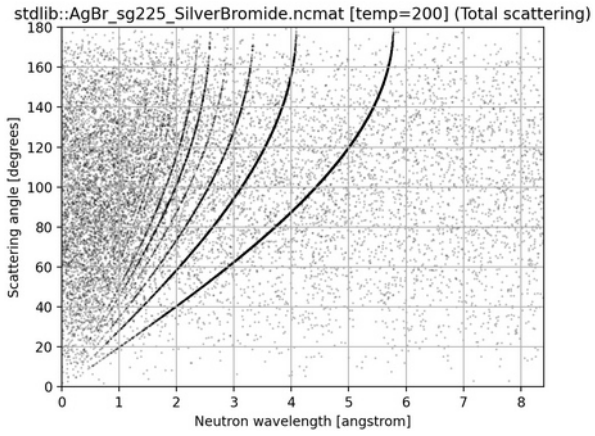
The `nctool` command allows inspection of physics resulting from a given cfg-string, browsing of data files, and more.

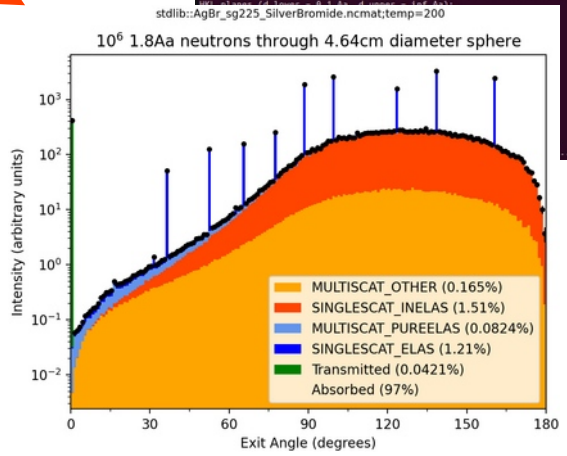
```
$> nctool --dump "AgBr_sg225_SilverBromide.ncmat;temp=200K"
```

```
$> nctool  "AgBr_sg225_SilverBromide.ncmat;temp=200K"
```









<u>Other scripts provided, mostly for conversions:</u>
`ncrystal_cif2ncmat, ncrystal_endf2ncmat, ncrystal_ncmat2hkl, ncrystal_hfg2ncmat, ncrystal-config, ncrystal_minimc, ...`

The Python API provides everything provided by the cmdline scripts, but sometimes a quick command in the terminal is convenient.

NCrystal in McStas

McStas n — NCrystal — Thermal Neutron Transport

Standard McStas example (`.instr`)

```
COMPONENT mysample = NCrystal_sample(cfg="Al2O3_sg167_Corundum.ncmat",
                                     radius=0.01, yheight=0.05)
AT (0, 0, 0) RELATIVE PREVIOUS
```

McStasScript + Union example (Python)

```
import mcstasscript.tools.ncrystal_union as ncunion
ncunion.add_ncrystal_union_material(instr,
                                    name="myAl",
                                    cfgstr="Al_sg225.ncmat;temp=10C")
#... usual mcstasscript code for creating volume "myvol" here
myvol.set_parameters(radius=0.01,
                     yheight=0.01,
                     material_string='"myAl"',
                     priority=1)
```

Simply use NCrystal cfg-strings when creating materials

To use Union+NCrystal directly in .instr files, the `ncrystal_mcstasunion` script easily generates the required code (can be used with .instr `SHELL` keyword for automated execution).

```
$> ncrystal_mcstasunion MyAl "Al_sg225.ncmat;temp=25C"
COMPONENT MyAl_ncrystal_proc = NCrystal_process(
    cfg = "Al_sg225.ncmat;temp=25C"
)
AT (0,0,0) ABSOLUTE

COMPONENT MyAl = Union_make_material(
    process_string = "MyAl_ncrystal_proc",
    my_absorption = 1.39136803716641
)
AT (0,0,0) ABSOLUTE
```

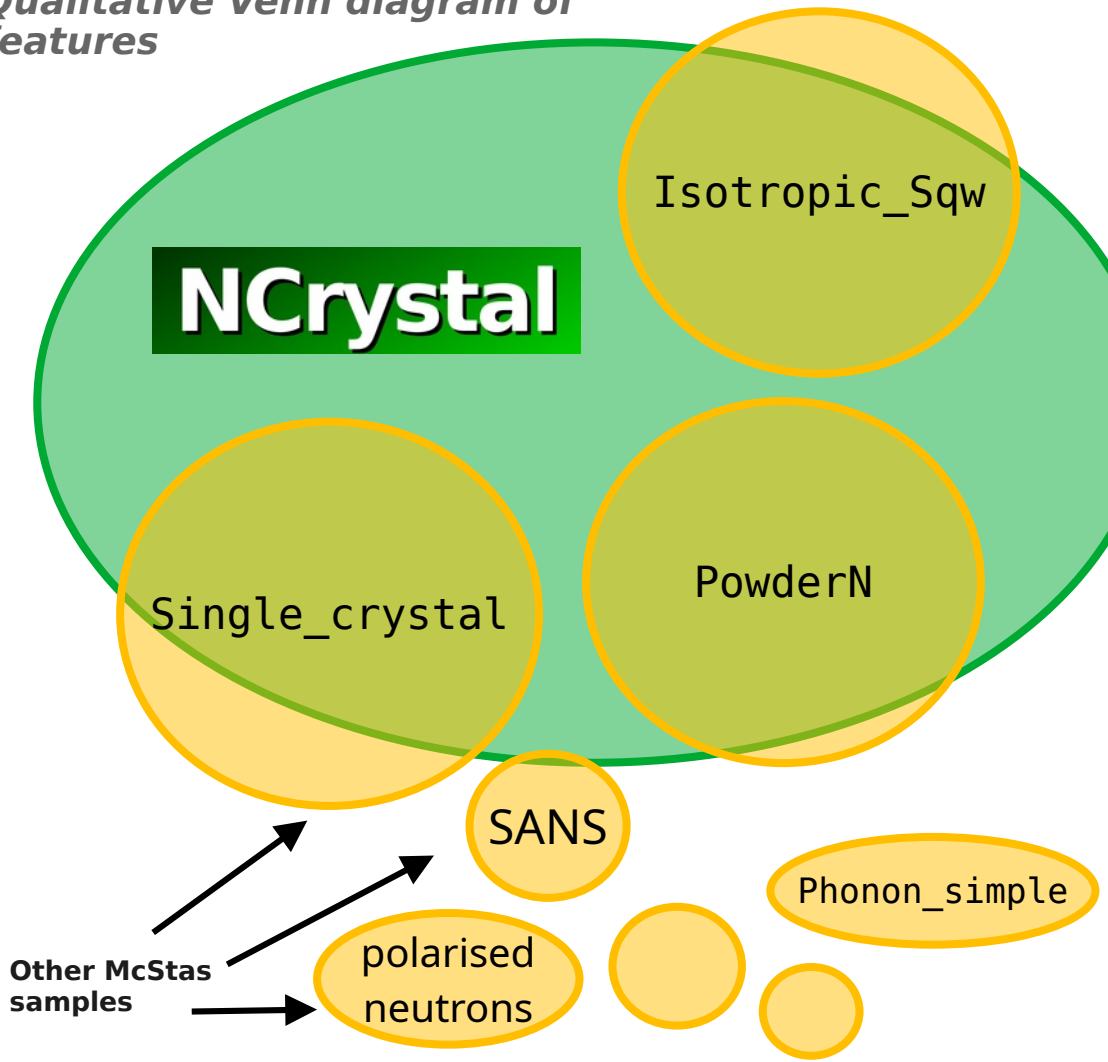
NOTE: This is admittedly a bit "hacky", and we plan to improve this!!

NCrystal versus other McStas "sample" components

McStas n

NCrystal
Thermal Neutron Transport

Qualitative Venn diagram of features



- **Some reasons to use NCrystal:**
  - Standalone features for debugging, configuring, or creating materials.
  - Aims to be a standard used across multiple codes (e.g. Vitess, OpenMC, Geant4, ...).
  - Large pre-existing material library.
  - Highly configurable materials (multi-phase support, easy to play with temperatures, enrichments, etc.).
  - Inclusive total cross sections (always accounting for both inelastic/elastic and coherent/incoherent, multiple scattering) ⇒ More realistic transmission.
  - In some cases NCrystal is only option for the physics models in McStas: HOPG, extinction, etc.
  - ...
- **Missing in NCrystal:**
  - Focusing options.
  - GPU support (runs in McStas funnel mode)
  - Surface physics
  - Huge component library of McStas contains many physics models without analogues in NCrystal.
  - ...

NCrystal + OpenMC/Geant4/...

Combining two regimes of neutron scattering

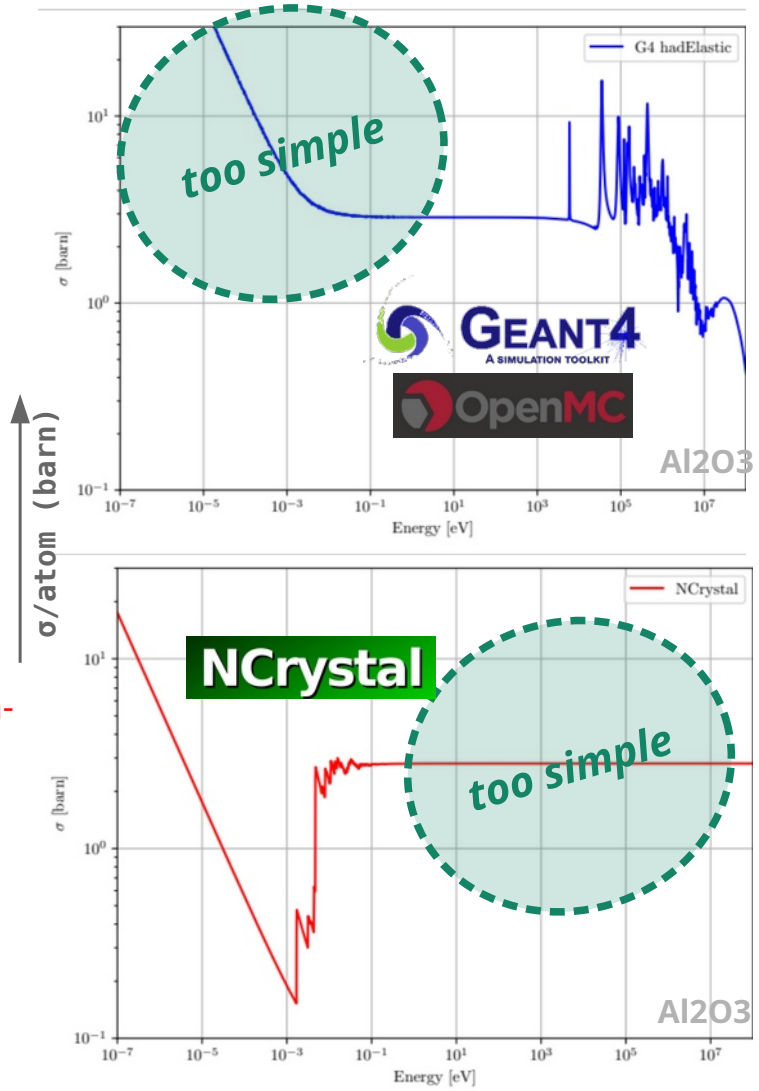
NCrystal
Thermal Neutron Transport

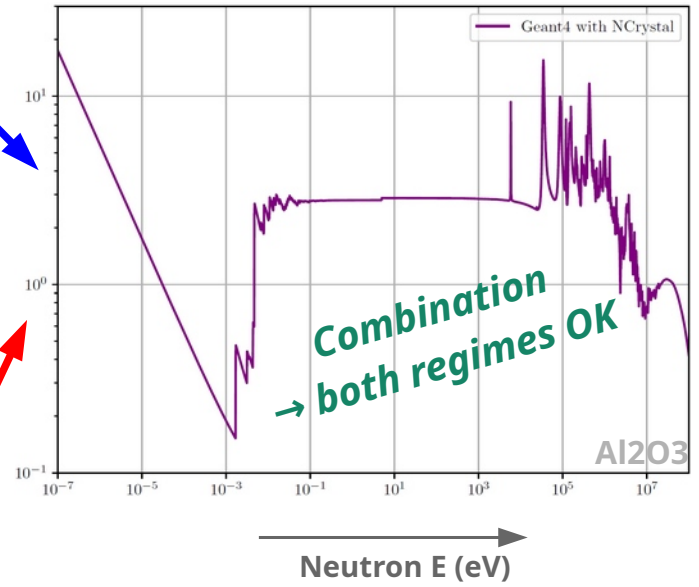
Higher energy (>>eV):

- Complex neutron-nuclei interactions with energy dependent strength.
- Not sensitive to material structure.

Low energy (<<eV):

- Simple (point-like) neutron-nuclei interactions with constant strength.
- Very sensitive to material structure.





BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

NCrystal in OpenMC

OpenMC

NCrystal
Thermal Neutron Transport

```python
import openmc
# Materials
openmc_mat = openmc.Material.from_ncrystal('Polyethylene_CH2.ncmat;temp=50C')
# Geometry
s1 = openmc.Sphere(r=10, boundary_type='vacuum')
c1 = openmc.Cell(region=-s1, fill=openmc_mat)
geometry = openmc.Geometry([c1])
# Execution settings
settings = openmc.Settings()
settings.source = openmc.Source(energy=openmc.stats.Discrete(x=[10.0], p=[1.0]))
settings.run_mode = 'fixed source'
settings.batches = 10
settings.particles = 10000
# Write xml files
model = openmc.model.Model(geometry=geometry, settings=settings)
model.export_to_xml()
```

Simply use NCrystal cfg-strings when creating materials

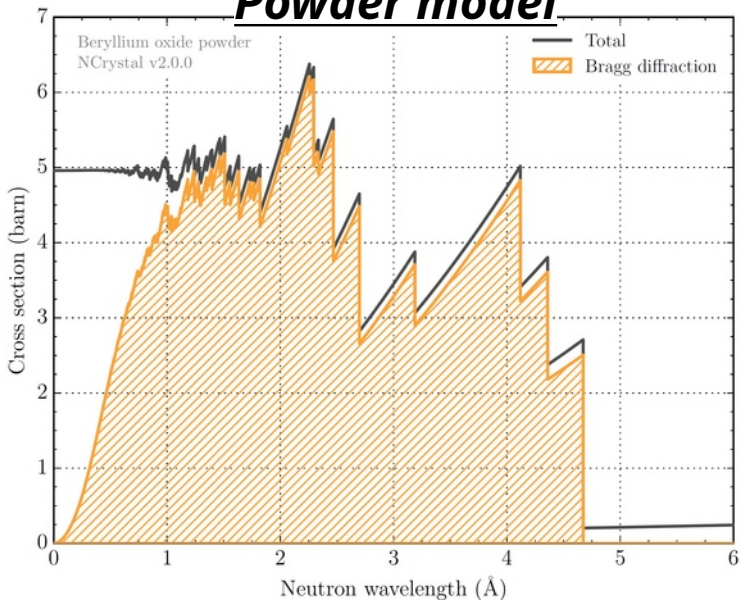
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

Bragg diffraction

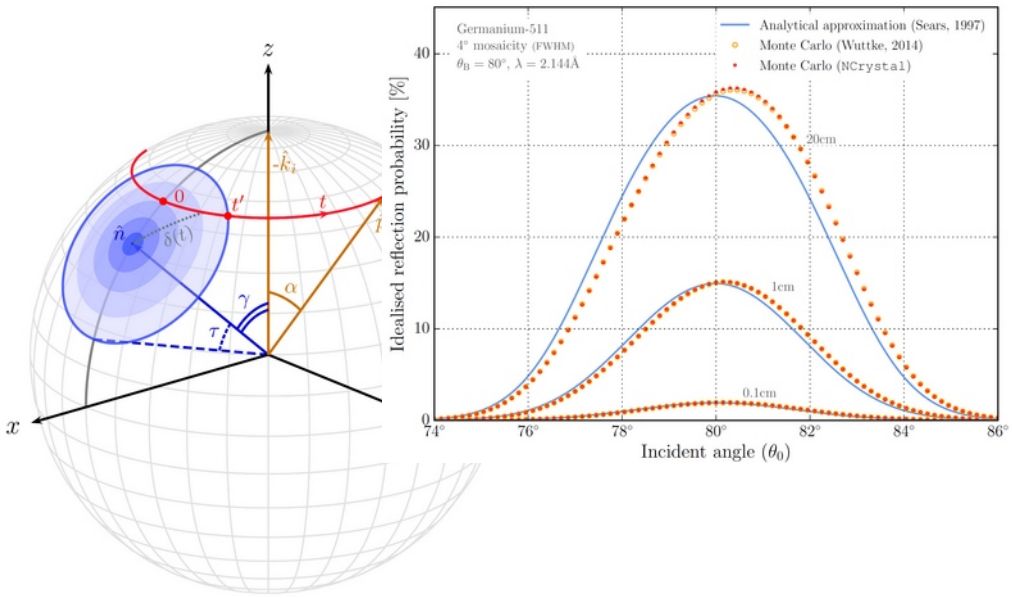
NCrystal
Thermal Neutron Transport

- HKL factors **initialised on-the-fly** from unit cell (<50ms).
  - Includes enough low-d-spacing planes to ensure correct transmission at shorter-wavelengths.
- Contains **power model**, **Gaussian mosaic single crystal model**, and **single crystal model for pyrolytic graphite**.
  - Single crystal models support back-scattering and any size of mosaic spread.
- Plugins for **texture** and **primary extinction** effects exist (to be adopted for NCrystal core).

Powder model



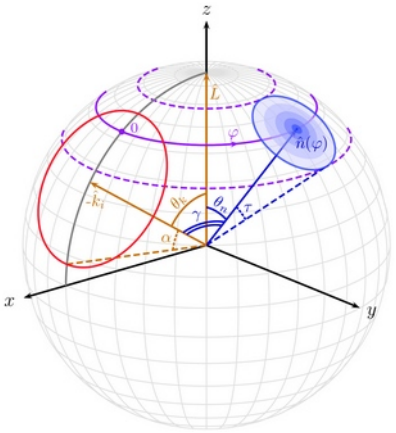
Single crystal model

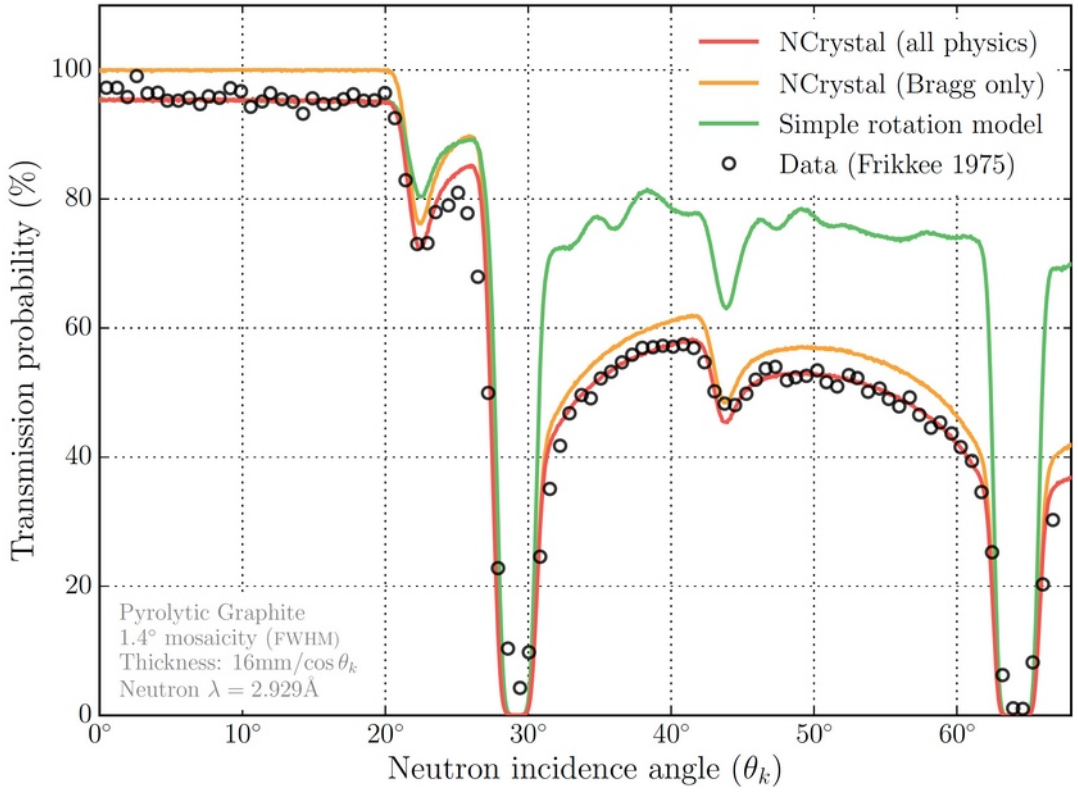


Special single crystal model for pyrolitic graphite

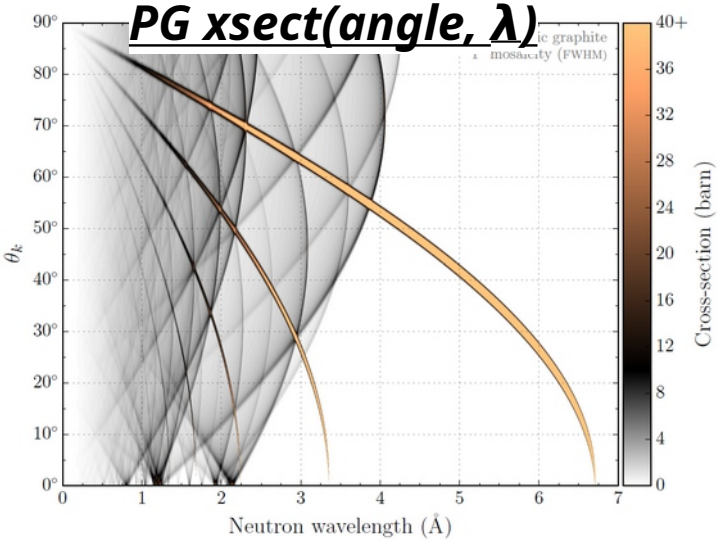
NCrystal
Thermal Neutron Transport

Tricky model to implement, but can actually reproduce transmission data!





PG xsect(angle, λ)

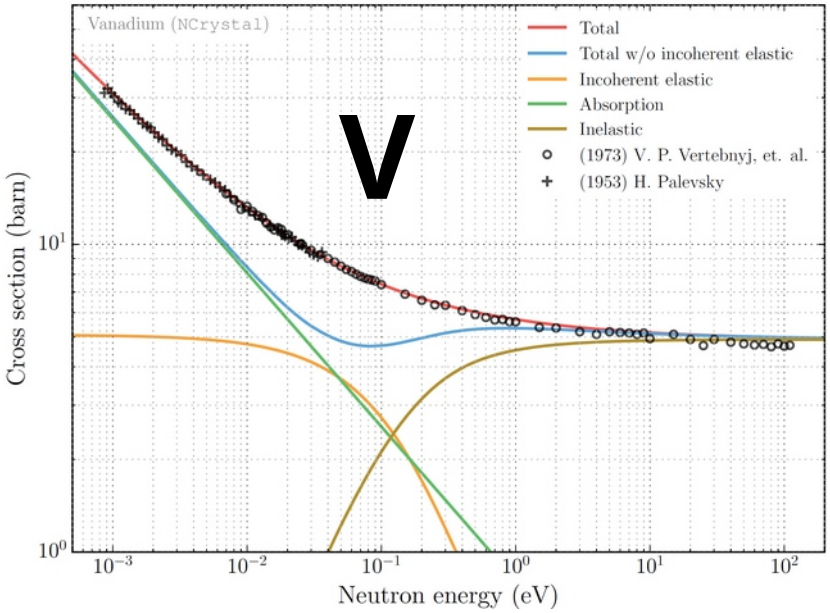


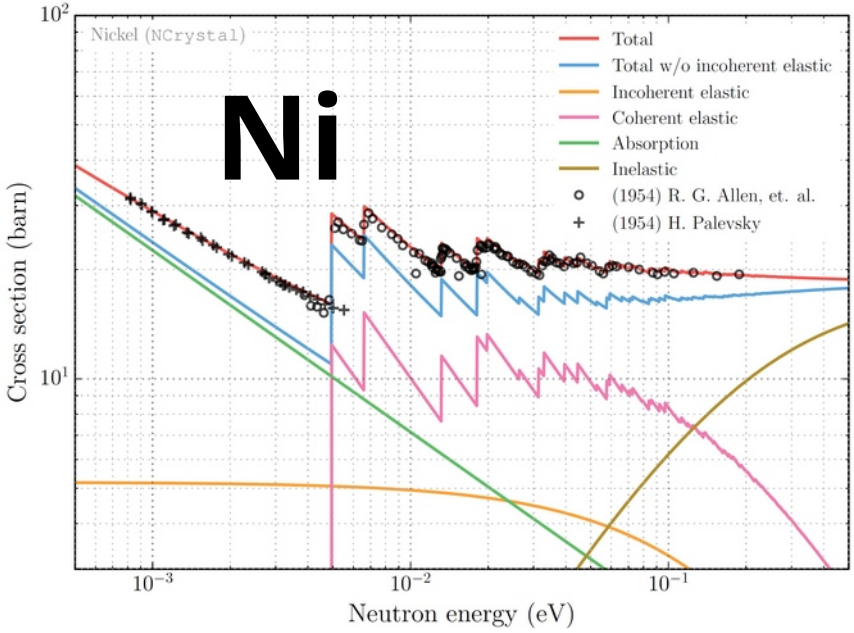
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

Incoherent-elastic model

Simple if you know the Debye-Waller factors

NCrystal
Thermal Neutron Transport

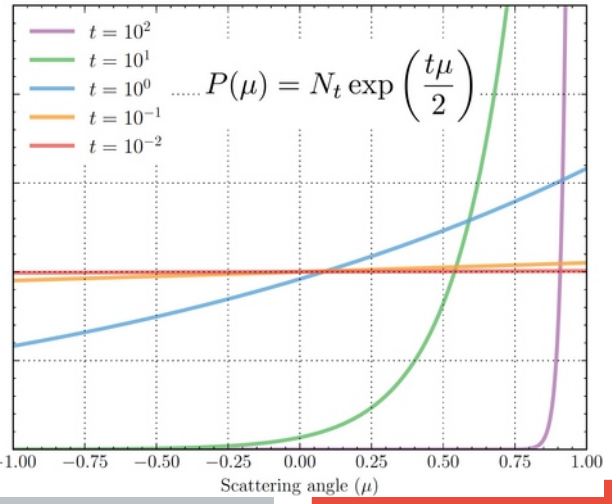




- Physics described by **sum of Debye-Waller factors**.
- Governed by ratio between neutron wavelength and atomic displacements:

$$t \equiv (2k\delta)^2 = \left(\frac{4\pi\delta}{\lambda}\right)^2$$

$$\sigma^{\mathrm{inc,el}}(k) = \sigma_{\mathrm{inc}} \frac{1-\exp(-t)}{t}$$



BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

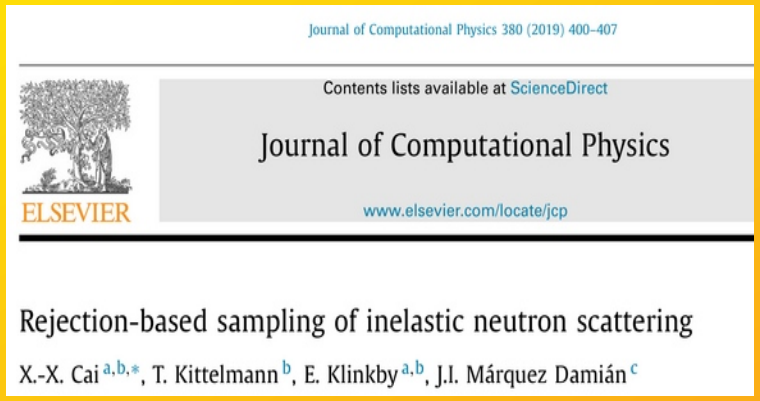
Inelastic physics:
Using the 2D scattering kernels

NCrystal
Thermal Neutron Transport

Custom algorithms for using the 2D kernels

→ neutron total cross sections $\sigma(k_f)$

→ precise MC sampling of (q,ω) for $k_i \rightarrow k_f$


Journal of Computational Physics 380 (2019) 400–407

Contents lists available at ScienceDirect

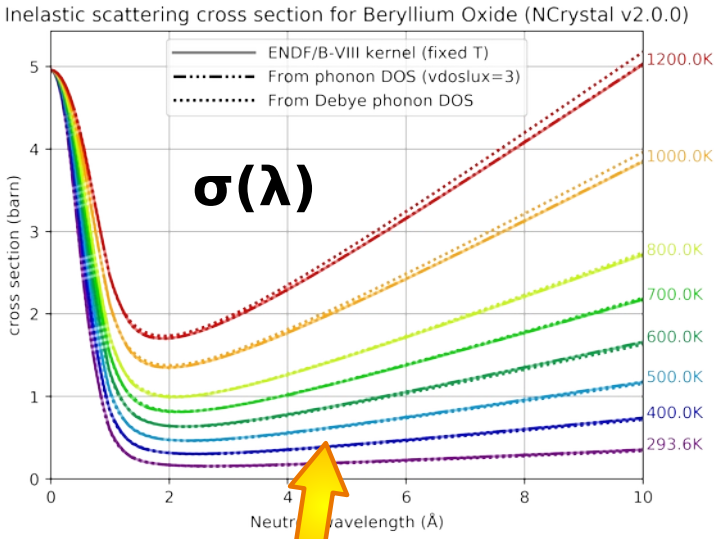
Journal of Computational Physics

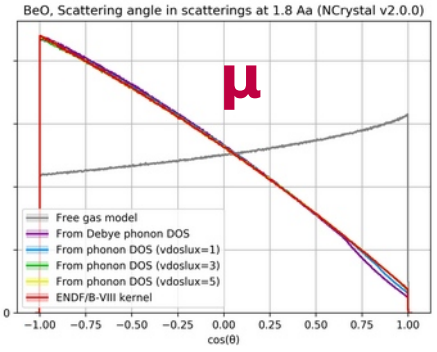
www.elsevier.com/locate/jcp

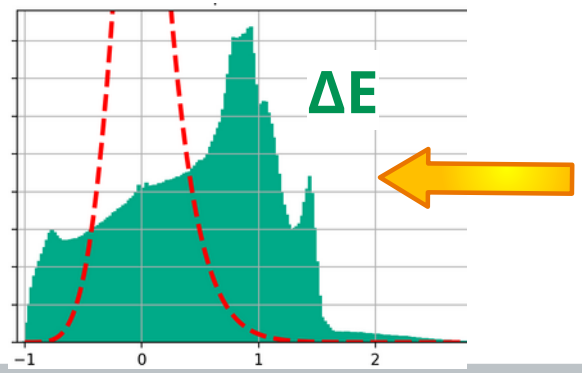
ELSEVIER

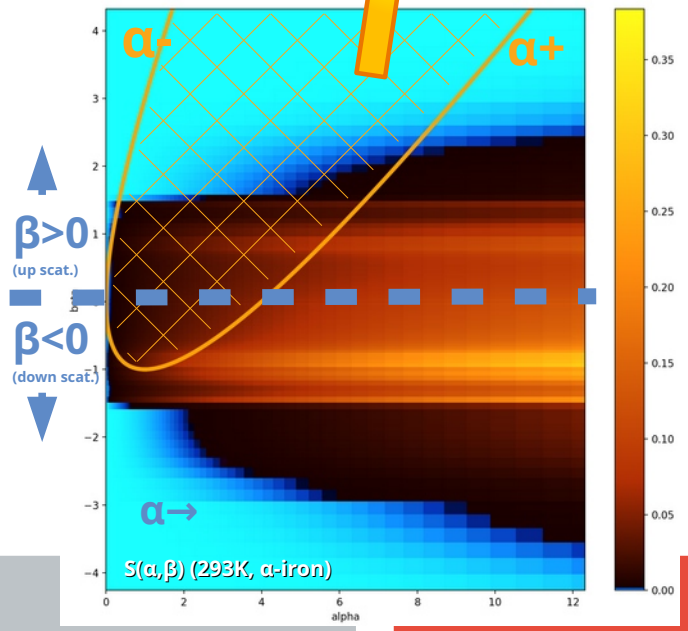
Rejection-based sampling of inelastic neutron scattering

X.-X. Cai [a,b,*], T. Kittelmann [b], E. Klinkby [a,b], J.I. Márquez Damián [c]









Easily model any hydrogen-rich amorphous solids (CLI tool)

NCrystal
Thermal Neutron Transport

- DFT/MD modelling of amorphous materials can be difficult and time consuming.
- Recent paper (Romanelli et. al., 2021) provides trustworthy and cheap alternative for hydrogen-rich materials.
- Relies on universality of hydrogen vibrations in different materials: Overall hydrogen VDOS can be composed from list of hydrogen bindings.
- We provide script for setting up NCMAT files with this.

Journal of Physics: Condensed Matter

PAPER

Thermal neutron cross sections of amino acids from average contributions of functional groups

Giovanni Romanelli[1], Dalila Onorati[9,2], Pierfrancesco Ulpiani[3], Stephanie Cancelli[4], Enrico Perelli-Cippo[4], José Ignacio Márquez Damián[5], Silvia C Capelli[1], Gabriele Croci[4,6], Andrea Muraro[6], Marco Tardocchi[6] Show full author list

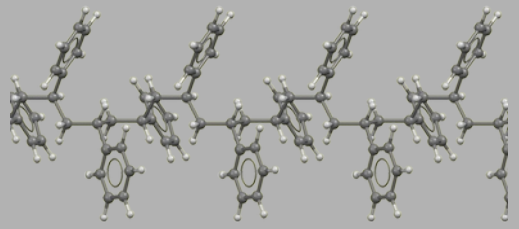
Published 31 May 2021 • © 2021 IOP Publishing Ltd

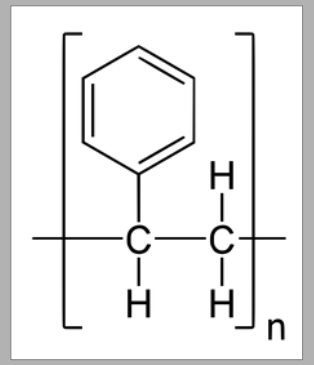
Journal of Physics: Condensed Matter, Volume 33, Number 28

Citation Giovanni Romanelli *et al* 2021 *J. Phys.: Condens. Matter* **33** 285901

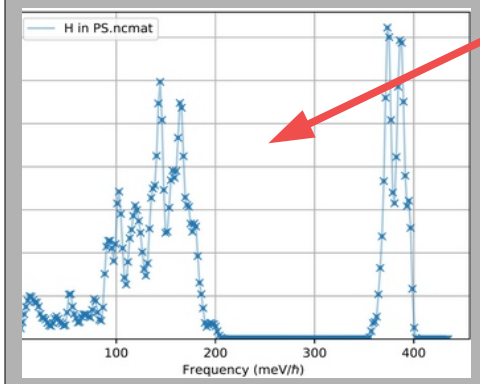
Example (polystyrene):

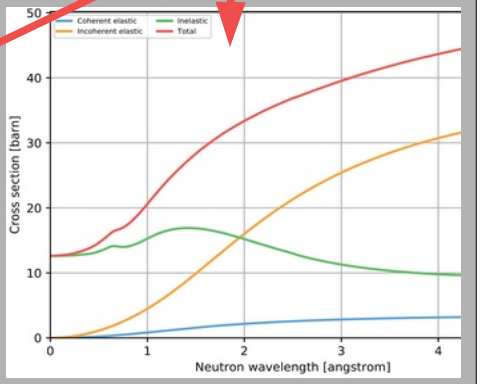
- 1 aromatic ring with 5 H
- 1 CH2 group
- 1 aliphatic CH binding






```
$> ncrystal_hfg2ncmat --formula C8H8 \
                      --spec 5xCHaro+1xCHali+1xCH2 \
                      --density 0.99 -o PS.ncmat
```



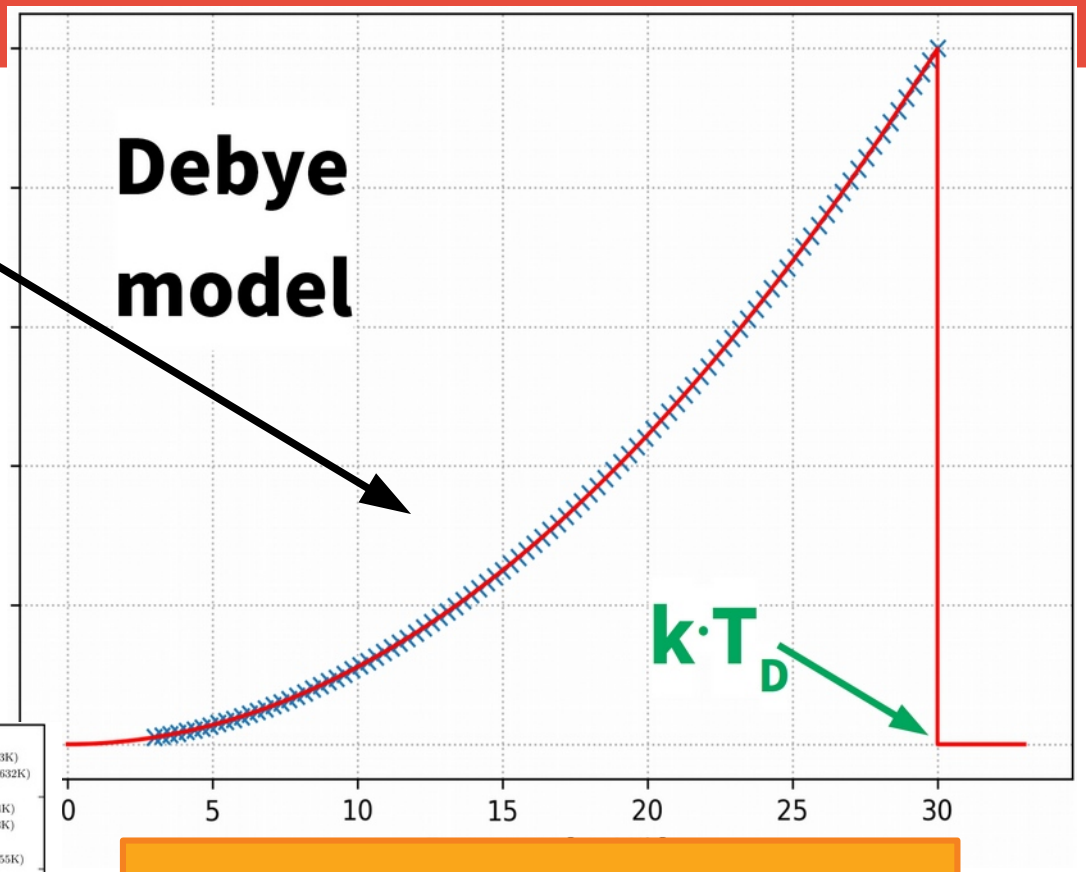


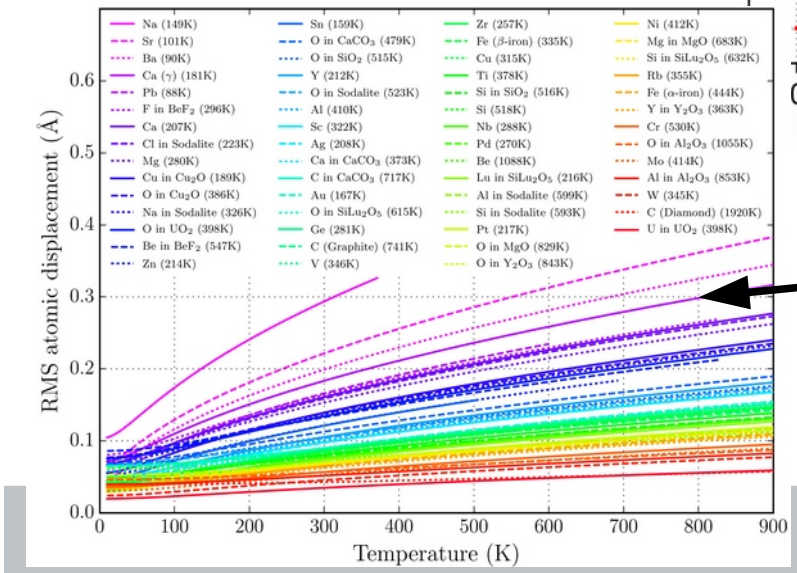
How we handle materials with no phonon DOS specified?

NCrystal
Thermal Neutron Transport

Idealised DOS (Debye Model) is constructed and fed into same infrastructure as any other DOS.

Lacks details of course, but gives consistent kinematics and handles multi-phonon physics ~OK.





T-dependent atomic displacements (δ), from Debye temperature (T_D)

Of course, a real DOS gives more realistic δ.