NCrystal : a library for thermal neutron transport

Thomas Kittelmann,
European Spallation Source ERIC
Data Management and Software Centre

Tutorials also by D. D. Julio and J. I. Márquez Damián
European Spallation Source ERIC
Spallation Physics Group

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

Programme

NCrystal
Thermal Neutron Transport

- **The NCrystal project: Background, history, introduction**
  - Jupyter: Python API and core NCrystal concepts
- **Using NCrystal as backend engine**
  - Jupyter: Scatter patterns with the builtin MiniMC framework
- **NCrystal material formats and data lib.**
  - Jupyter: Data infrastructure and standard data library
- **NCrystal physics algorithms**
  - (Recap of thermal neutron scattering theory)
  - NCrystal elastic physics algorithms
  - Jupyter: Creating materials and the NCMATComposer (first half)
  - NCrystal inelastic physics algorithms
  - Jupyter: Creating materials and the NCMATComposer (second half)
- **Bits and pieces: Other features, plans, etc.**
  - Afternoon: More Jupyter-based tutorials with various examples and use-cases.

jupyter

Jupyter tutorials at:

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

This includes links to open in the cloud at google Colab

These slides also available at:
https://indico.esss.lu.se/event/3439/

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: background, history, brief introduction.

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

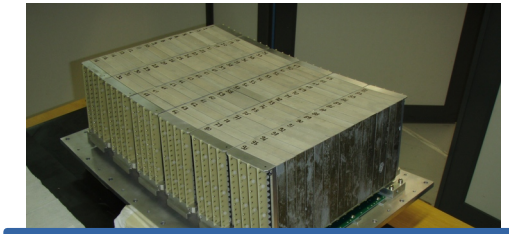
The NCrystal project: Background

https://mctools.github.io/ncrystal/

NCrystal
Thermal Neutron Transport

Original motivation back in ~2014 (X.X. Cai and T. Kittelmann):

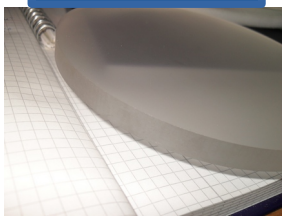
Augment Geant4 with proper modelling of thermalised neutrons in crystalline materials (and avoid the usual free-gas treatment)



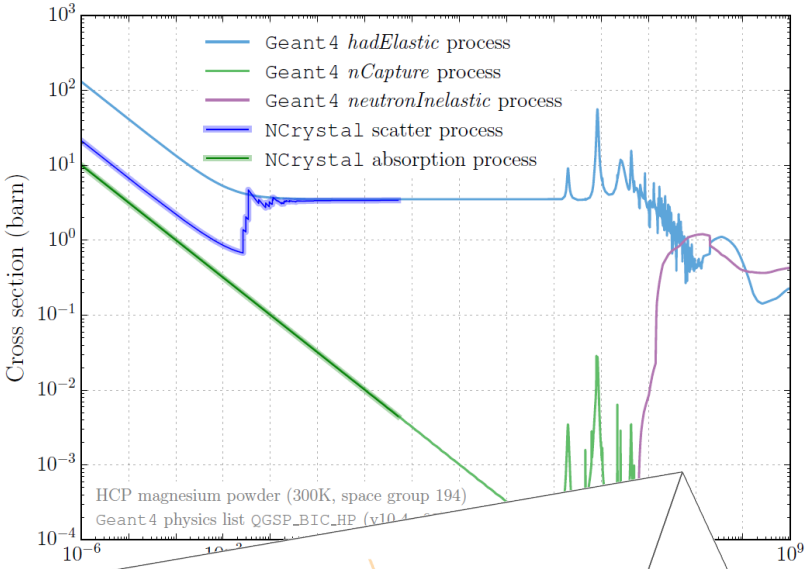
Detector frames, vessels, supports (polycrystalline metals)

Crystalline samples

Filters (single- or poly-crystals)

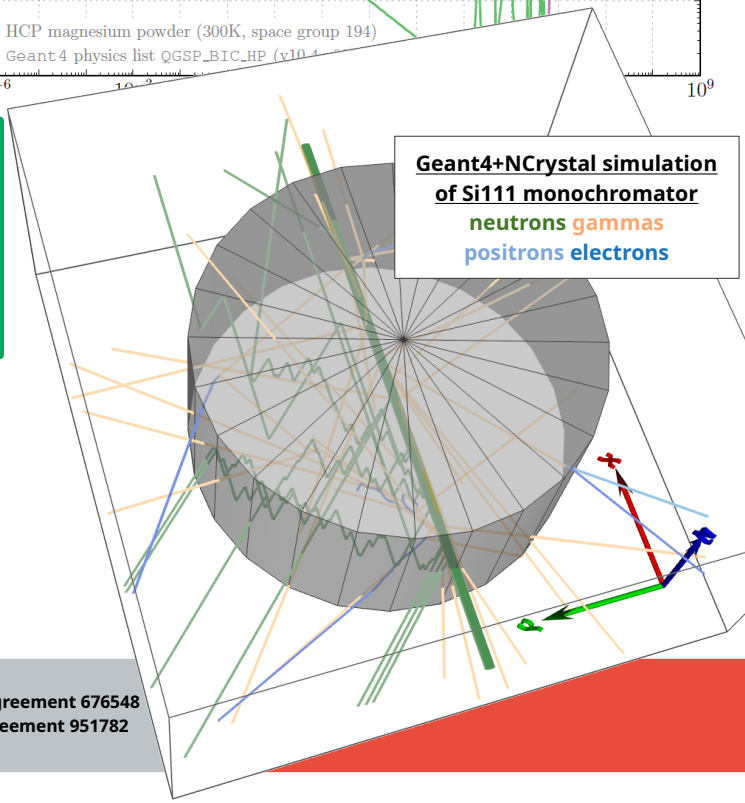


Monochromators, analysers (single crystals. layered crystals)



Advanced earlier efforts in older "NXSG4" plugin
- T. Kittelmann & M Boin 2015 *Comput. Phys. Commun.* **189**, 114-118
- Geant4-specific plugin for polycrystals, no inelastic, no tools/bindings – just a thin wrapper around nxslib by M. Boin.

Scope has since expanded beyond Geant4!
And the physics scope has extended beyond crystals and beyond Bragg diffraction!



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

A brief history of NCrystal
(no, this will not be on the test)

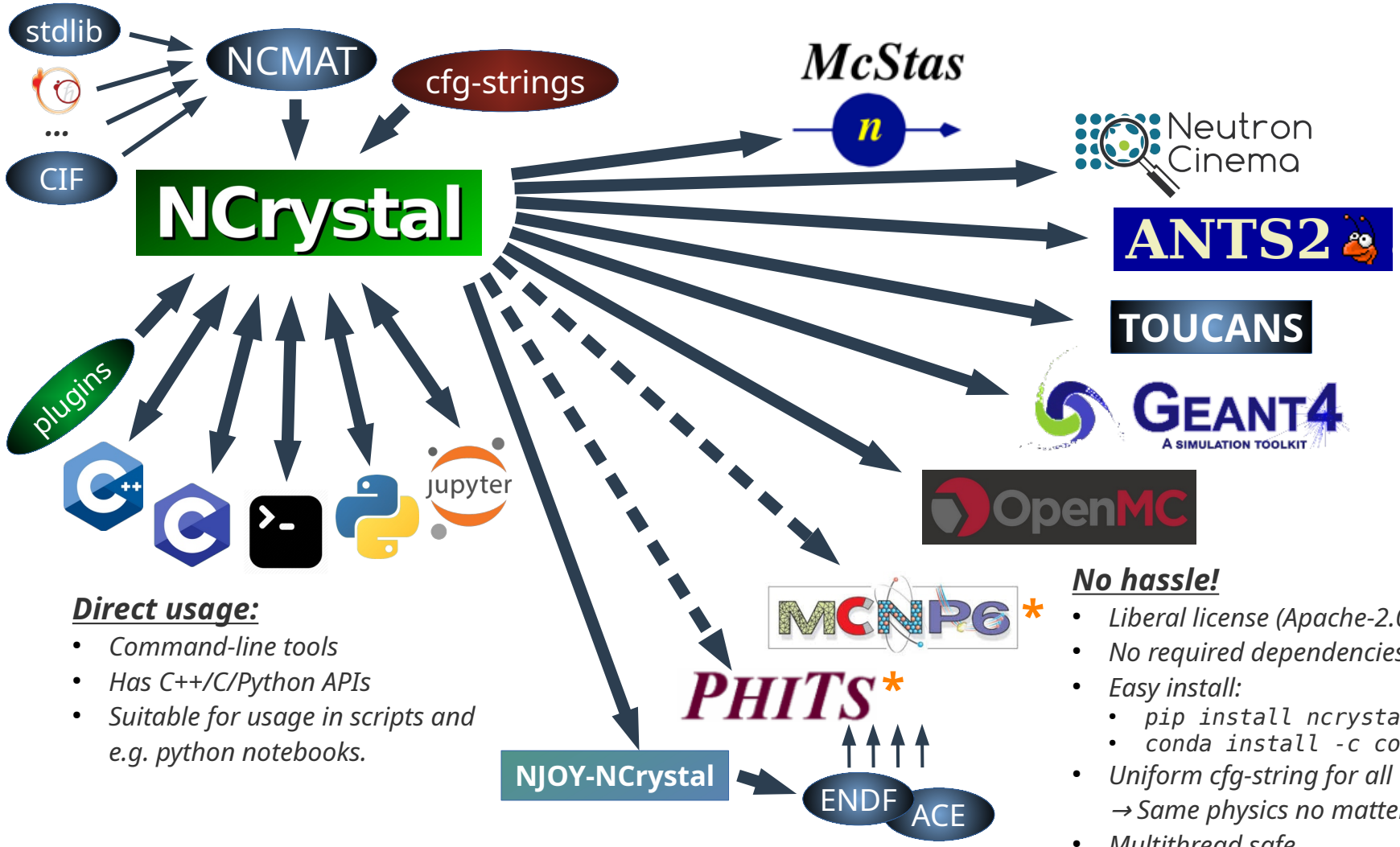
NCrystal
Thermal Neutron Transport

- (prehistoric: NXSG4 in 2014 by T. Kittelmann + M. Boin)
- ~2015: T. Kittelmann + X. X. Cai of the ESS Detector Group and DTU NuTech join separate efforts aimed at adding thermal neutron physics to Geant4. Quickly decide to make it a standalone project and add support for McStas as well. Both join the Geant4 collaboration to get NCrystal integrated, and also work with P. Willendrup on McStas integration.
- 2019 (v1.0): After years of development and writing the first publication, release 1.0, focusing on Bragg diffraction and crystalline materials. HKL structure factors are calculated on the fly at startup. Contains minimal C++/C/Python API and hooks for Geant4 and McStas.
- 2020 (v2.0): with inelastic physics: scattering kernels and a unique capability to expand phonon spectra to scattering kernels on the fly.
- 2020 (fall) (v2.1-2.4): Support atomic/isotopic mixtures, virtual files, and a system for third-party plugins with specific new physics.
- March 2021 (v2.5): Major C++11 rewrite for safe modern C++. Becomes multi-thread safe and gets a flexible data-source structure.
- April 2021 (v2.6): K. Ramic, J. I. Marquez Damian, and D.DiJulio join the efforts, and most files in the data library gets phonon DOS curves added. We also begin to estimate atomic displacements from such curves.
- May 2021 (v2.7): The data library grows enormously (now arguably world leading!) thanks again to collaboration with the same people. Add support for amorphous materials, add cmdline tools for adding new materials.
- April 2022 (v3.0): Large update with multi-phase materials, support for SANS physics, new cfg-parameters like “density”, “one-liner materials”, etc.
- June 2022 (v3.1): Focus on UCN (ultra cold neutron) production in inelastic collisions, ensuring artifact-free modelling and possibilities for biasing.
- August 2022 (v3.2): Support for easily configuring gas-mixtures.
- Aug-Dec 2022 (v3.3-3.5): Improve CMake layer and introduce ncrystal-config command. Improve integrations with McStas (with P. Willendrup and M. Bertelsen) and Geant4. Bindings for OpenMC in OpenMC 13.3 release. NCrystal appears on conda-forge. CI tests improves with help of M. Klausz.
- May 2023 (v3.6): Large improvement of python API, many utilities for creating new materials from a variety of sources. Publish Jupyter tutorials.
- July 2023 (v3.7): NCrystal appears on PyPI, “pip install ncrystal” now works! ESS DMSC supports NCrystal.
- December 2024 (v3.8): NCrystal components move into McStas itself, NCrystal+McStas integration improves via conda (work with P. Willendrup).
- August 2024 (v3.9): NCrystal ships with “MiniMC” for fast simple diffraction patterns.
- October 2024 - : NCrystal work supported by APRENDE EURATOM grant (T.K. with J. I. Marquez Damian, and D.DiJulio).
- Along the way, supported by: two EU projects, EURATOM, got physics in external plugins (Thanks N. Rizzi, S. Xu, CSNS), and were supported in various fashions by many people not mentioned already, in particular: T. R. Nielsen, V. Santoro, R.H.Wilton, K.Kanaki, A. Morozov, E. Klinkby, E. Knudsen, ...
- Many ongoing ideas and developments at ESS (in various groups), and CSNS in particular.

Not such an old project, but a lot of activity and many people contributing in many ways!
Always looking for more contributions!

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

NCrystal
Thermal Neutron Transport



Direct usage:

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

No hassle!

- *Liberal license (Apache-2.0).*
- *No required dependencies beyond C++/Python*
- *Easy install:*
  - `pip install ncrystal`
  - `conda install -c conda-forge ncrystal`
- *Uniform cfg-string for all usage*
  → Same physics no matter where used.
- *Multithread safe.*
- *Used on Linux/OSX/BSD, Windows by EOY 2024.*

***: Can not share bindings for non open-source codes**

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

Standard physics in NCrystal
(crystals/liquids/amorphous solids/gasses)

NCrystal
Thermal Neutron Transport

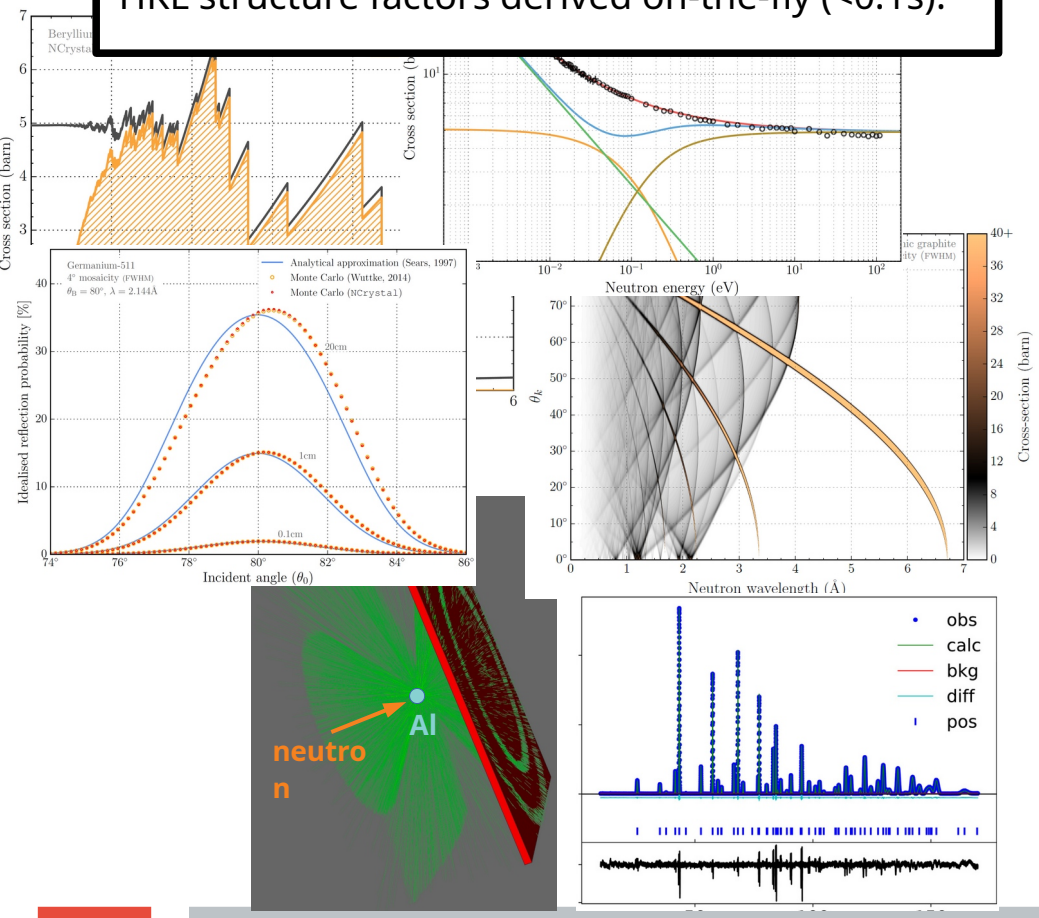
Elastic (ΔE=0) components

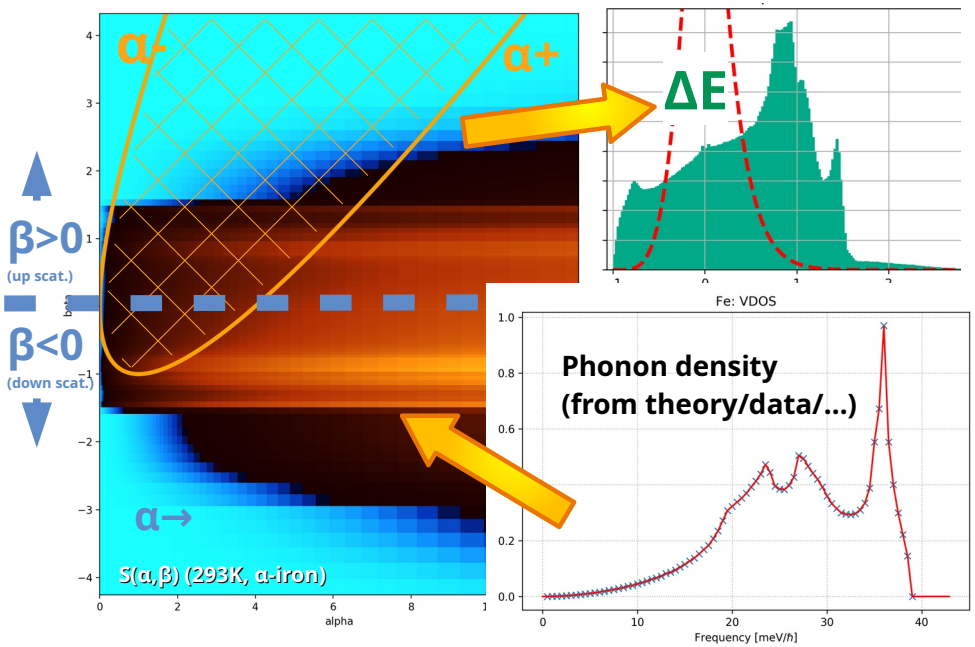
Bragg diffraction, incoherent, single crystals, isotropic materials (powders), HOPG.
HKL structure factors derived on-the-fly (<0.1s).

Inelastic (ΔE≠0) components

Scattering kernel based:

- Initialise from external kernel
- Or from phonon density curve (~0.1s).
- Using incoherent approximation (for now!)





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

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

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

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

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

Core NCrystal concepts

All other features, tools, hooks, etc. are built upon these.

NCrystal
Thermal Neutron Transport

Inputs

cfg-strings

```
"Al_sg225.ncmat"
"Y2SiO5_sg15_YSO.ncmat;dcutoff=0.5Aa"
"gasmix::0.7xCO2+0.3xAr/1.5atm/250K"
"""phases<0.1*PbS_sg225_LeadSulfide.ncmat
   &0.9*Epoxy_Araldite506_C18H2O3.ncmat>"""
"Rubber_C5H8.ncmat;temp=200;vdoslux=4"
"LiquidHeavyWaterD2O_T293.6K.ncmat"
```

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

Info (object)

Densities, compositions

Phases

Scattering lengths

Crystal structures, HKL factors

Dynamics info like phonon DOS

Scatter process (object)

Provides integrated scattering cross sections when queried with neutron state.

Can perform Monte Carlo sampling of outgoing states.

Absorption process (object)

Provides integrated absorption cross sections when queried with neutron state.

Presently only implements simple 1/v model.

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

Cfg-strings

Complete list: https://github.com/mctools/ncrystal/wiki/CfgRefDoc

NCrystal
Thermal Neutron Transport

Apply after "filename" part, separated with semicolons:
`"Al_sg225.ncmat;density=2.6gcm3;temp=250K;comp=elas"`

- **Temperature:**
  - Examples: `temp=100`, `temp=100K`, `temp=-10C`, `temp=100F`
  - All materials have a temperature. The `temp` parameter does exactly what you think it does. By default it assumes the value is in kelvin, but a unit can be added (must be one of `K`, `C`, or `F`.
- **Density**
  - Examples: `density=2.0gcm3`, `density=3.4kgm3`, `density=0.9x`.
  - Also does what you think it does. The last example scales the density by a factor of 0.9.
- **Scattering component toggling**
  - Current recognised component names are `coh_elas` (alias `bragg`), `incoh_elas`, `sans`, and `inelas`. `elas` refers to all components except `inelas`.
  - Syntax: `<compname>=0` (disable component), `comp=<compname1>,..,<compnamen>` (disable all but the listed components.
  - Examples: `...;comp=inelas,sans` (only inelastic and SANS),`...;inelas=0` (without inelastic), `...;inelas=0;comp=inelas` (actually removes all components)
- **Modify atomic compositions**
  - Examples: `atomdb=H is D`, `atomdb=Al:is:0.9:Al:0.1:Cr`, `atomdb=Si29:28.97649466525u:4.7fm:0.001b:0.101b`, ...
- **Single crystal parameters**
  - Single crystal models and orientations are primarily controlled by the parameters `mos`, `dir1`, `dir2`, and `dirtol`. Refer to the documentation linked above, and see examples in the notebooks.

`Whitespace in cfg-strings is allowed (and ignored).`

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

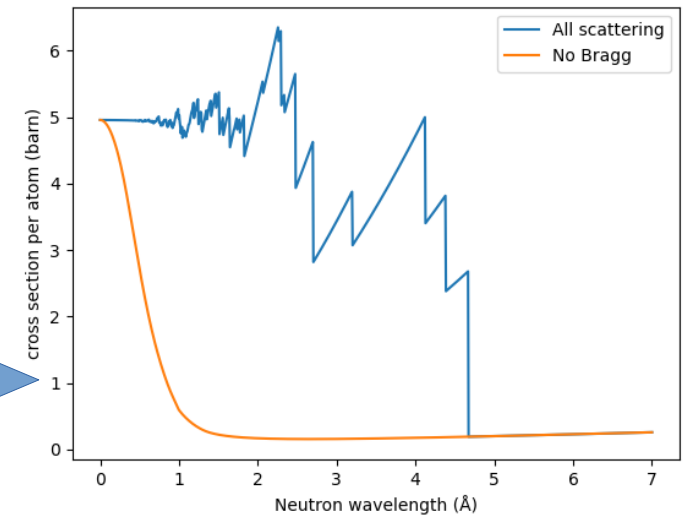
Python API

NCrystal
Thermal Neutron Transport

```
#Plot beryllium-oxide cross sections
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()
```

Universal cfg-strings
→ Same cfg in McStas, Geant4, ...

...or just do: `NC.load('Beo_sg186.ncmat').plot()`




```
#Can also extract more detailed info:
info_BeO = NC.createInfo('BeO_sg186.ncmat')
info_BeO.dump()
print('Density [g/cm3]: %g'%info_BeO.density)
print('Number density [atoms/Å³]: %g'%info_BeO.numberdensity)
for fraction,atom in info_BeO.composition:
    print(f'Has {fraction*100}% {atom}')
for hkl in [ hkl for hkl in info_BeO.hklObjects()
             if ( 1.1 < hkl.d < 1.17) ]:
    print(f'N={hkl.mult} d={hkl.d:g}Å F²={hkl.f2:g}barn',
          list(zip(hkl.h,hkl.k,hkl.l)))
```

Prints all info to terminal

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

```
Density [g/cm3]: 3.00836
Number density [atoms/Å³]: 0.144868
Has 50.0% Be=Be(cohSL=7.79fm cohXS=7.62579barn incXS=0.0018barn absXS=0.0076barn
Has 50.0% O=O(cohSL=5.803fm cohXS=4.2317barn incXS=0.0008barn absXS=0.00019barn m
N=6 d=1.16827Å F²=1.66027barn [(2, 0, 0), (2, -2, 0), (0, 2, 0)]
N=12 d=1.14858Å F²=3.49424barn [(2, -1, 2), (2, -1, -2), (1, 1, 2), (1, 1, -2), (
N=12 d=1.1288Å F²=0.772761barn [(2, 0, 1), (2, 0, -1), (2, -2, 1), (2, -2, -1), (
```

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

Command-line interface (CLI)

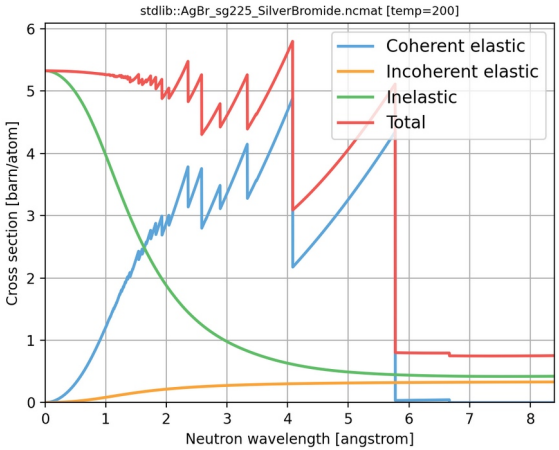
NCrystal
Thermal Neutron Transport

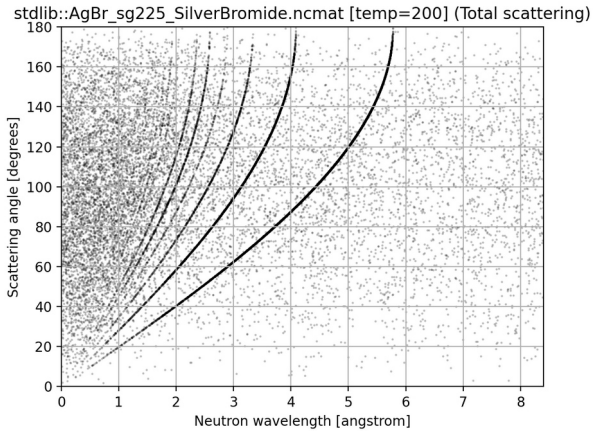
The `nctool` command allows inspection of physics resulting from a given cfg-string, browsing of data files, and more.

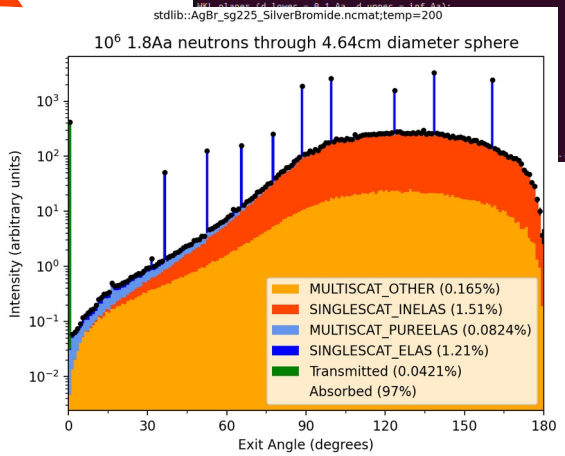
```
$> nctool --dump 'AgBr_sg225_SilverBromide.ncmat;temp=200K'
```

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

```
----------------------------------  NCrystal Material Info  ----------------------------------
Data source: stdlib::AgBr_sg225_SilverBromide.ncmat
Density : 6.47734 g/cm3, 0.0415477 atoms/Aa^3
Composition (by mole): 50% Br 50% Ag
Composition (by mass): 42.5535% Br 57.4465% Ag
Atom data:
   Br = Br(cohSL=6.795fm cohXS=5.80215barn incXS=0.1barn absXS=6.9barn mass=79.9035u Z=35)
   Ag = Ag(cohSL=5.922fm cohXS=4.40704barn incXS=0.58barn absXS=63.3barn mass=107.868u Z=47)
Averaged quantities:
   Atomic mass                 : 93.8859u
   Absorption XS at 2200m/s    : 35.1 barn
   Free scattering XS          : 5.32549 barn
   Scattering length density : 2.64181 10^-6/Aa^2
Temperature : 200 kelvin
State of matter: Solid (crystalline)
Space group number       : 225
Lattice spacings   [Aa] : 5.7745 5.7745 5.7745
Lattice angles     [deg] : 90 90 90
Unit cell volume [Aa^3] : 192.55
Atoms / unit cell        : 8
Atoms in unit cell (total 8):
     4 Br atoms [T_Debye=123.347K, MSD=0.0241933Aa^2]
     4 Ag atoms [T_Debye=103.372K, MSD=0.0254376Aa^2]
Atomic coordinates:
     Br            0            0          1/2
     Br            0          1/2            0
     Br          1/2            0            0
     Br          1/2          1/2          1/2
     Ag            0            0            0
     Ag            0          1/2          1/2
     Ag          1/2            0          1/2
     Ag          1/2          1/2            0
Dynamic info for Br (50%):
   type: S(alpha,beta) [from VDOS]
   VDOS Source: 2030 points
   VDOS E_max: 18.0077 meV
Dynamic info for Ag (50%):
   type: S(alpha,beta) [from VDOS]
   VDOS Source: 2030 points
   VDOS E_max: 18.0077 meV
HKL info type: SymEqvGroup
```







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

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

Time to look at the first notebook:

”Introduction to NCrystal and the Python API”

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

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

Using NCrystal as a backend engine (examples)

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 McStas

- **Works "out of the box" as long as NCrystal is installed.**
  - Automatically installs NCrystal if installing McStas from conda.
  - Automatically installs NCrystal if installing McStas from "app bundle" for Windows/MacOS, or Debian packages.
- **Several ways to use:**
  - With dedicated `NCrystal_sample.comp` gives multiple-scattering and absorption effects in single volume (box/cylinder/sphere).
  - As a physics engine in McStas Union's advanced geometry system.

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 McStas

Standard McStas non-Union 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

Can of course also use Union+NCrystal in `.instr` and `NCrystal_sample` in McStasScript.

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 + OpenMC/Geant4

Combining two regimes of neutron scattering

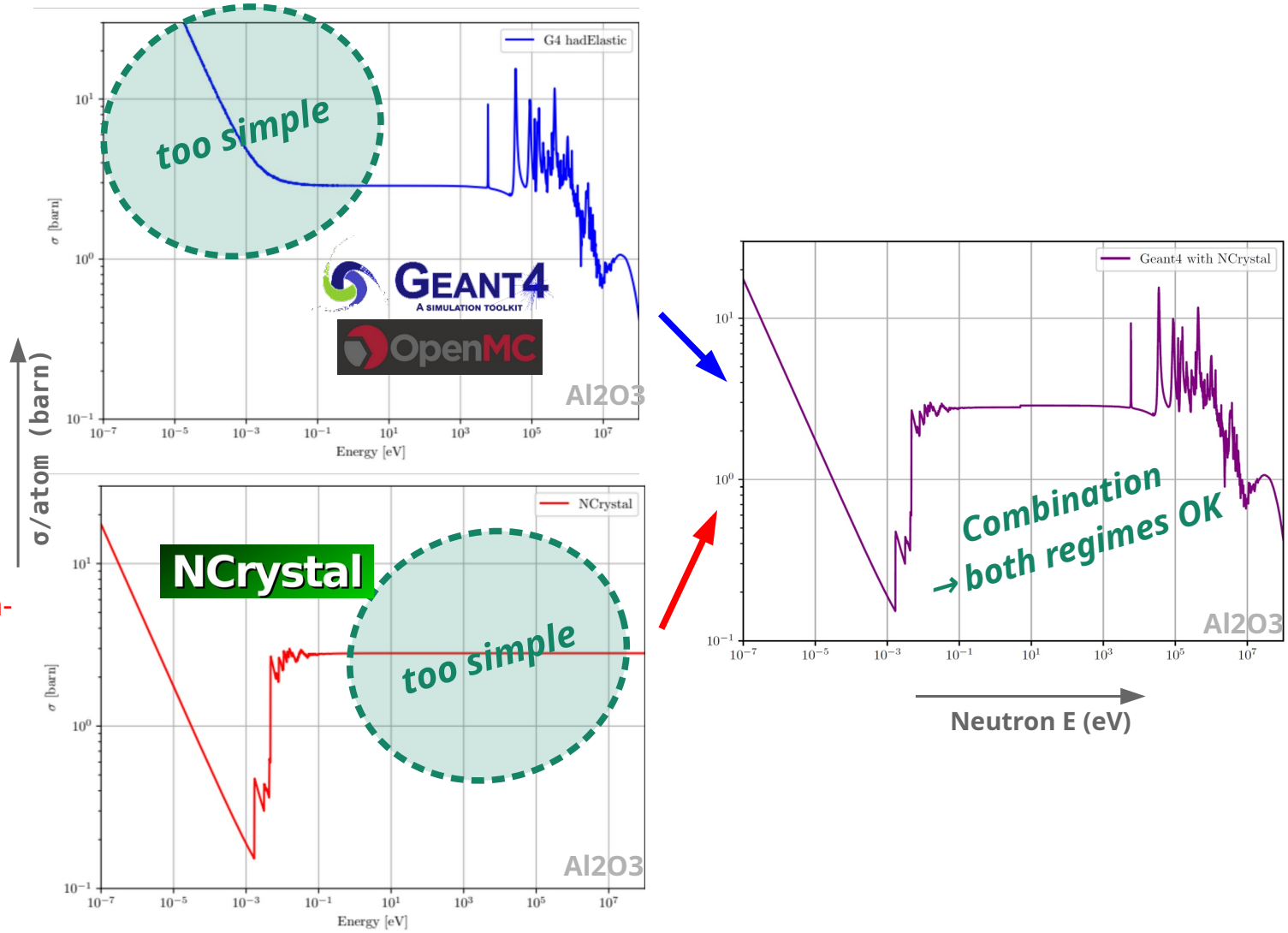
NCrystal
Thermal Neutron Transport

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

```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()
#Now launch openmc
#Check resulting materials.xml (we could of course actually RUN OpenMC here instead)
import pathlib
print(pathlib.Path('materials.xml').read_text())
```

Simply use NCrystal cfg-strings when creating materials

Currently needs OpenMC built with `-DOPENMC_USE_NCRYSTAL=On`

Plan is to enable this in OpenMC conda packages.

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 Geant4

```
//Include the relevant header:
#include "G4NCrystal/G4NCrystal.hh"

//...

//Create materials directly from cfg-strings:
G4Material * mat_aluminium = G4NCrystal::createMaterial("Al_sg225.ncmat");

//...

//Currently the NCrystal process must then be injected in the following way:
g4runManager->Initialize();
G4NCrystal::installOnDemand();
g4runManager->BeamOn(1000)
```

Simply use NCrystal cfg-strings when creating materials

Somewhat unusual initialisation (plan to improve this)

Currently needs custom build of NCrystal with: -DNCRYSTAL_ENABLE_GEANT4=On

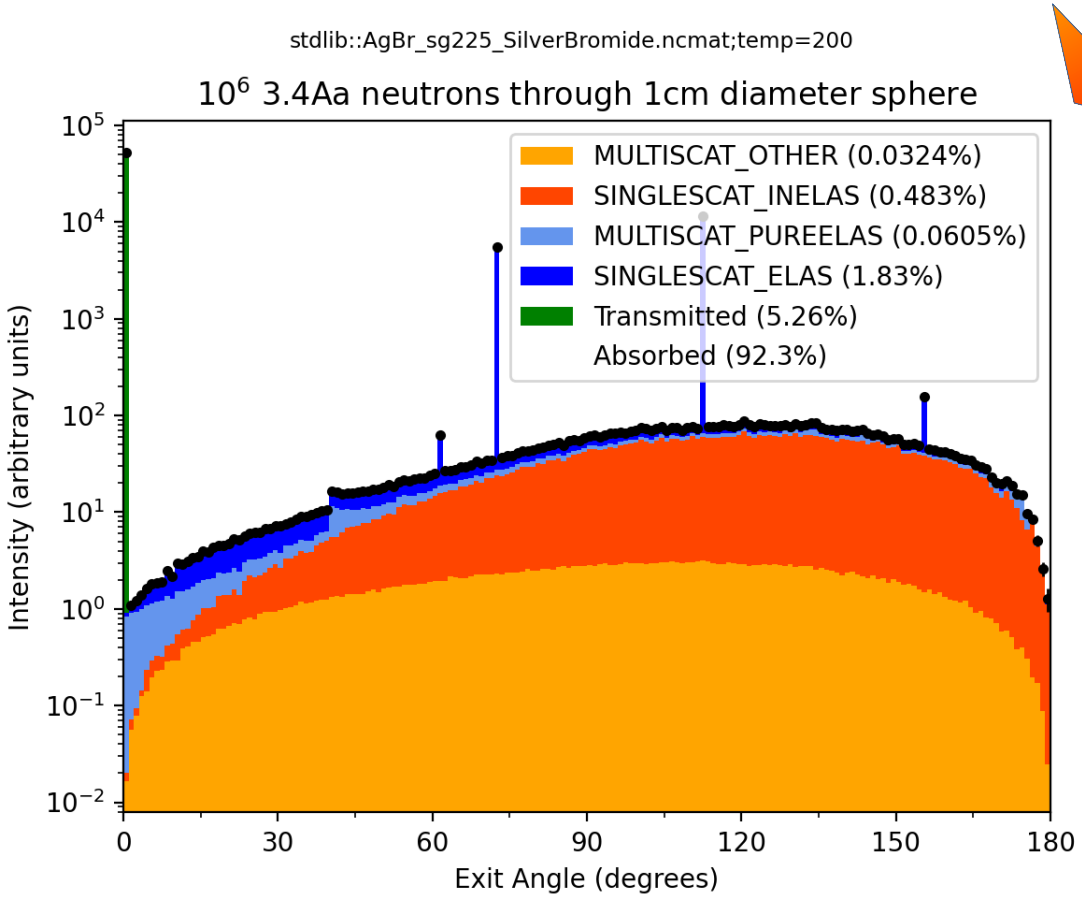
Plan to migrate to standalone package and make it work out of the box in conda.

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

Built-in "Mini MC" for scatter patterns with multiple-scattering effects

NCrystal
Thermal Neutron Transport

```
$> nctool --mc 3.4Aa 1cm  'stdlib::AgBr_sg225_SilverBromide.ncmat;temp=200K'
```



Runs multithreaded and vectorised, for fast results (ultimately intended for possible usage in diffraction analysis)

<u>Time for this particular simulation on my laptop:</u>
Load material: 140ms
Run simulation: 170ms
Total: 310ms

Note: 10^6 src neutrons → >10^7 tally entries due to some variance reduction tricks.

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

Try it in the notebook:

”Using the built-in MiniMC framework for generating scatter patterns”

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

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 material formats and data library

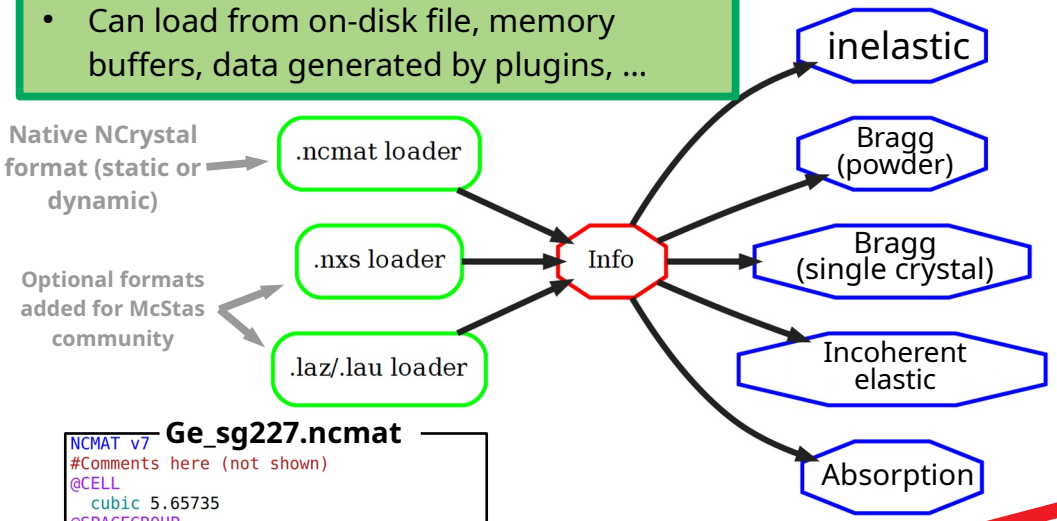
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

The NCMAT data format for materials

https://github.com/mctools/ncrystal/wiki/NCMAT-format

NCrystal
Thermal Neutron Transport

- NCrystal supports multiple input formats
- But NCMAT (.ncmat) is the native one.
- Can load from on-disk file, memory buffers, data generated by plugins, ...



Ge_sg227.ncmat

```
NCMAT v7
#Comments here (not shown)
@CELL
  cubic 5.65735
@SPACEGROUP
  227
@ATOMPOSITIONS
  Ge 1/2 1/2 1/2
  Ge 1/4 1/4 3/4
  Ge 1/2 0   0
  Ge 1/4 3/4 1/4
  Ge 0   1/2 0
  Ge 3/4 1/4 1/4
  Ge 0   0   1/2
  Ge 3/4 3/4 3/4
@DYNINFO
  element  Ge
  fraction 1
  type     vdos
  vdos_egrid .0037709 .0377897
  vdos_density .0064216 .0067183 .0070150
    .0074164 .0078883 .0078611 .0078339
    #(~50 lines here not shown)
    .032778 .02264 .012923 .002484 0
```

NCMAT loader

Info

```
-----------------------------------  NCrystal Material Info  -----------------------------------
Data source: Ge_sg227.ncmat
-----------------------------------------------------------------------------------------------
Density : 5.3288 g/cm3, 0.0441826 atoms/Aa^3
-----------------------------------------------------------------------------------------------
Composition (by mole): 100% Ge
-----------------------------------------------------------------------------------------------
Composition (by mass): 100% Ge
-----------------------------------------------------------------------------------------------
Atom data:
   Ge = Ge(cohSL=8.185fm cohXS=8.41874barn incXS=0.18barn absXS=2.2barn mass=72.6322u Z=32)
-----------------------------------------------------------------------------------------------
Averaged quantities:
   Atomic mass                : 72.6322u
   Absorption XS at 2200m/s   : 2.2 barn
   Free scattering XS         : 8.3648 barn
   Scattering length density  : 3.61634 10^-6/Aa^2
-----------------------------------------------------------------------------------------------
Temperature : 293.15 kelvin
-----------------------------------------------------------------------------------------------
State of matter: Solid (crystalline)
-----------------------------------------------------------------------------------------------
Space group number      : 227
Lattice spacings   [Aa] : 5.65735 5.65735 5.65735
Lattice angles    [deg] : 90 90 90
Unit cell volume [Aa^3] : 181.067
Atoms / unit cell       : 8
-----------------------------------------------------------------------------------------------
Atoms in unit cell (total 8):
     8 Ge atoms [T_Debye=295.348K, MSD=0.00692135Aa^2]
-----------------------------------------------------------------------------------------------
Atomic coordinates:
     Ge            0            0          1/2
     Ge            0          1/2            0
     Ge          1/4          1/4          3/4
     Ge          1/4          3/4          1/4
     Ge          1/2            0            0
     Ge          1/2          1/2          1/2
     Ge          3/4          1/4          1/4
     Ge          3/4          3/4          3/4
-----------------------------------------------------------------------------------------------
Dynamic info for Ge (100%):
   type: S(alpha,beta) [from VDOS]
   VDOS Source: 425 points
   VDOS E_max: 37.7897 meV
-----------------------------------------------------------------------------------------------
HKL info type: SymEqvGroup
-----------------------------------------------------------------------------------------------
HKL planes (d_lower = 0.1 Aa, d_upper = inf Aa):
  H   K   L  d_hkl[Aa] Mult. FSquared[barn]
  1   1   1    3.26627    8      20.896
  2   2   0    2.00018   12     40.0457
  3   1   1    1.70576   24     19.5165
  4   0   0    1.41434    6     37.4019
  3   3   1    1.29789   24     18.2281
```

Conversions from/to other formats supported by Python/CLI tools: CIF, ENDF, .laz/.lau, ...

Neutron interactions (scatter processes), Based on this “Info” object.

NCrystal data library

Browse with performance plots at: https://github.com/mctools/ncrystal/wiki/Data-library

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_BeryllliumNitride.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_BetaQuart
Si_sg227.ncmat
Sn_sg141.ncmat
SrF2_sg225_StrontiumFluor
SrH2_sg62_StrontiumHydrid
Sr_sg225.ncmat
Th3N4_sg166_ThoriumNitrid
ThO2_sg225_ThoriumDioxide.ncmat
Th_sg225.ncmat
TiO2_sg136_Rutile.ncmat
TiO2_sg141_Anatase.ncma
Ti_sg194.ncmat
TlBr_sg221_ThaliumBromi
Tm2O3_sg206_ThuliumOxid
UF6_sg62_UraniumHexaflo
UO2_sg225_UraniumDioxid
void.ncmat
V_sg229.ncmat
W_sg229.ncmat
Xe_Gas_STP.ncmat
Y2O3_sg206_Yttrium_Oxid
Y2SiO5_sg15_YSO.ncmat
Y3Al5O12_sg230_YAG.ncmat
Y_sg194.ncmat
ZnF2_sg136_ZincFlouride.
ZnO_sg186_ZincOxide.ncma
Zn_sg194.ncmat
ZnS_sg216_Sphalerite.ncm
ZrF4-beta_sg84.ncmat
ZrO2_sg137_Zirconia.ncma
ZrO2_sg14_Zirconia.ncmat
Zr_sg194.ncmat
```

132 materials (v3.9.3):

Crystals (108), amorphous solids (16), liquids, gasses, …

Easy universal cfg:

"Al_sg225.ncmat;temp=250K"
"Rubber_C5H8.ncmat;comp=inelas"

- Same physics in all applications!
- Cfg variables documented at:

github.com/mctools/ncrystal/wiki/CfgRefDoc

Small (few kB) file sizes:

- Optionally embed in binary and avoid need for actual files.

Easy to create more:

- Hand-write NCMAT file (human readable ASCII, format well-defined & versioned) or use new NCMATComposer.
- Convert from ENDF, CIF, online crystal DB carbohydrate chemical formula, Quantum Espresso output, …
- Request help on GitHub/ncrystal.

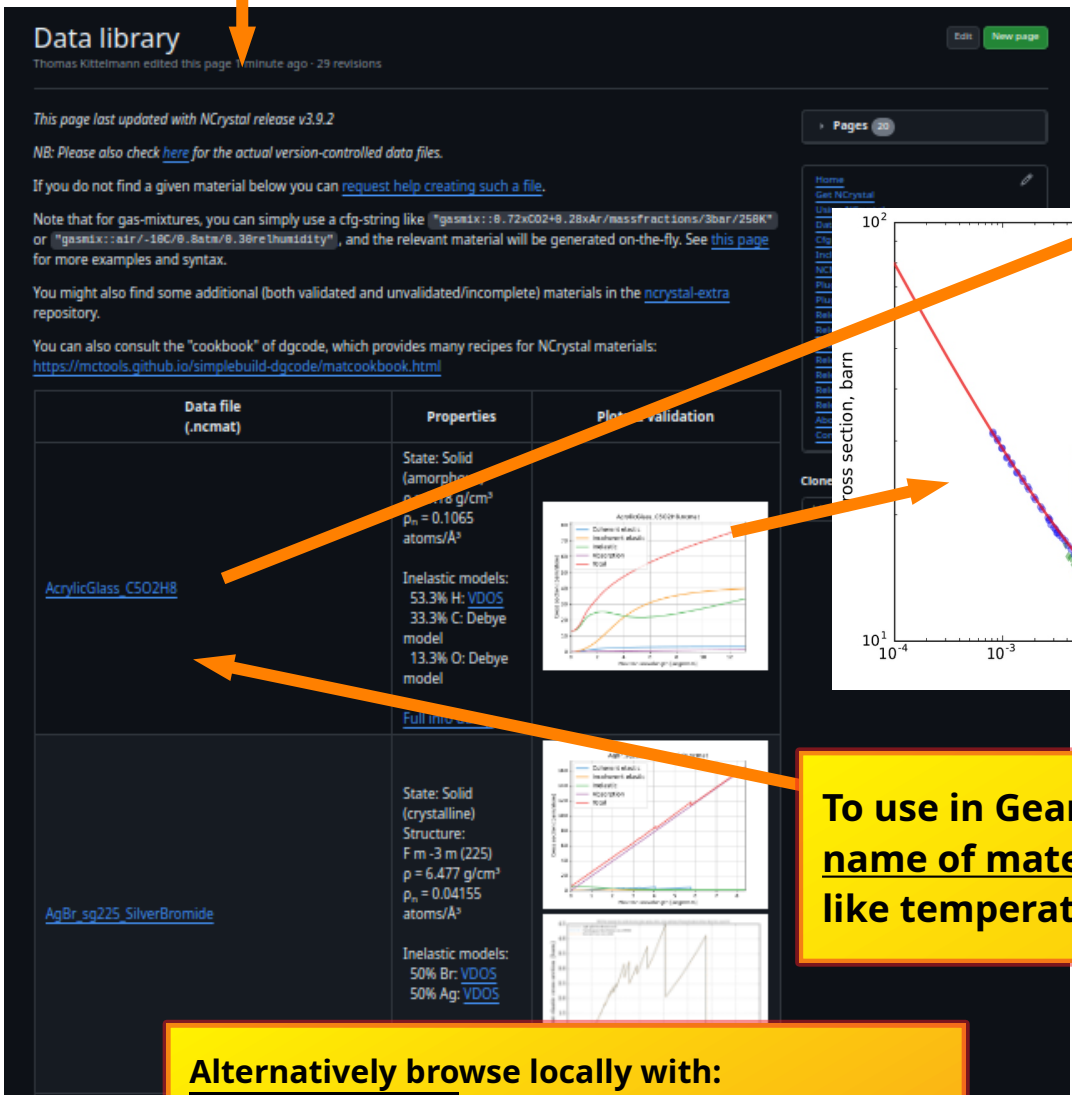
Can be converted to other formats:

- To .laz/.lau for McStas
- To ENDF via the NJOY-NCrystal project
- But limited by target format physics capabilities!

NCrystal data library wiki page

https://mctools.github.io/ncrystal/ → wiki → Data-library

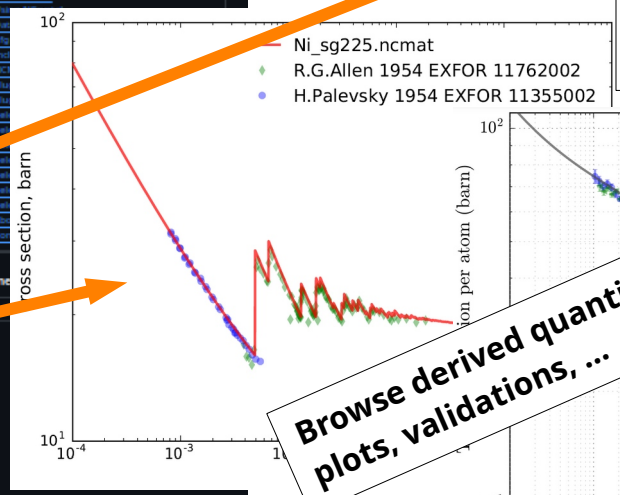
NCrystal
Thermal Neutron Transport

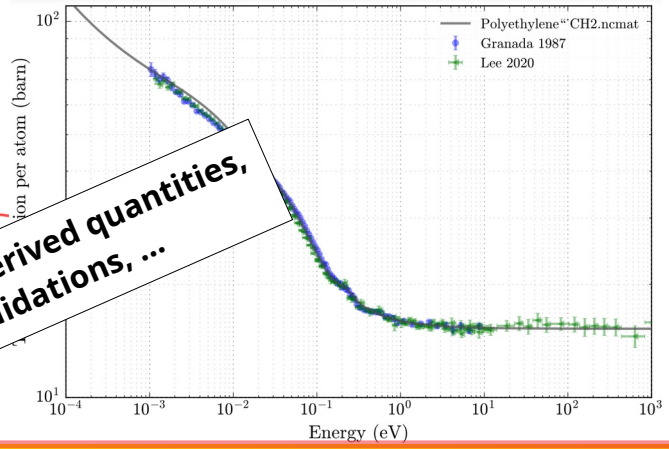



```
NCMAT v1
#Some comment here...
@CELL
    lengths 4.04958 4.04958 4.04958
    angles 90. 90. 90.
@SPACEGROUP
    225
@ATOMPOSITIONS
    Al 0. 0.5 0.5
    Al 0. 0. 0.

    Al 0.5 0. 0.5
@DEBYETEMPERATURE
```

Al_sg225.ncmat





Browse derived quantities, plots, validations, ...

To use in Geant4/McStas/OpenMC/Python/..., simply supply name of material file, along with relevant parameters like temperature, orientation of single crystal, etc.

Alternatively browse locally with:
`nctool –browse`
And then:
`nctool [-dump|--extract] SomeFile.ncmat`

NOTE: M. Klausz is working on a much improved less static datalib website.

In-memory data

With or without a “filename”

NCrystal
Thermal Neutron Transport

Register as virtual file with filename:

```
import NCrystal as NC
content="""NCMAT v3
@CELL
 lengths 4.04958 4.04958 4.04958
 angles 90 90 90
@SPACEGROUP
  225
@ATOMPOSITIONS
  Al 0   1/2 1/2
  Al 0     0   0
  Al 1/2 1/2   0
  Al 1/2   0 1/2
@DEBYETEMPERATURE
  Al   410.4
"""

NC.registerInMemoryFileData( "MyAl.ncmat",content)
sc = NC.createScatter("MyAl.ncmat")
```

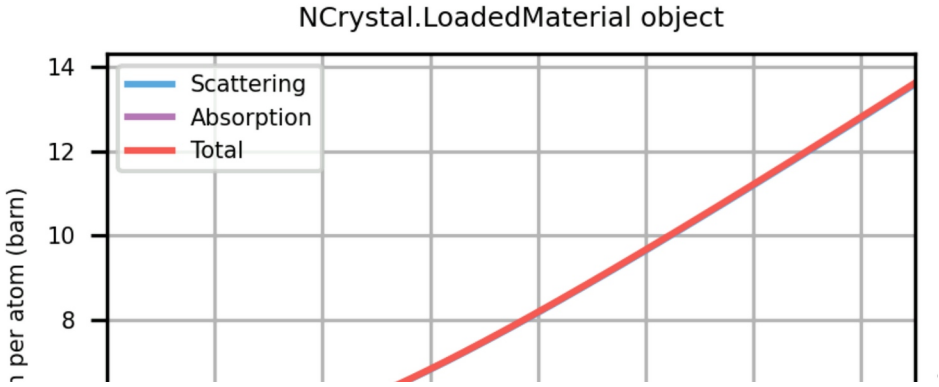
Accessing data is of course possible:

```
nctool –extract Al_sg225.ncmat
NC.createTextData('Al_sg225.ncmat')
```

NCrystal conda/pip packages use this to embed standard data lib files into libNCrystal.
(using CMake -DNCRYSTAL_ENABLE_DATA=EMBED)

Load NCMAT data directly:

```
a_string_with_ncmat_data="""NCMAT v7
#Don't use this material for anything
@DENSITY
  1.2345 g_per_cm3
@DYNINFO
  element  C
  fraction 1
  type     freegas
"""
NC.load(a_string_with_ncmat_data).plot()
```



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

Quick one-liner materials

Especially useful for gas mixtures and highly absorbing materials

NCrystal
Thermal Neutron Transport

- NCrystal's flexible data infrastructure allows plugins to provide one-liner materials, i.e. materials completely defined by their cfg-string without the need for actual NCMAT data.
- This is best for materials without the need for e.g. unit cell positions or DOS curves.
- The one-liner goes in the "filename" part of a cfg-string.
- Dedicated plugins (currently "freegas", "solid", "gasmix") analyse the "filename" and produces corresponding NCMAT data on-the-fly (run "nctool --browse" for examples).
- Examples:

```
"solid::B4C/2.52gcm3/B_is_0.95_B10_0.05_B11"
"gasmix::0.72xCO2+0.28xAr/massfractions/1.5atm/250K"
"gasmix::0.7xCO2+0.3xAr/0.001relhumidity"
"gasmix::0.7xCO2+0.3xAr/1.5atm/250K"
"gasmix::BF3/2atm/25C/B_is_0.95_B10_0.05_B11"
"gasmix::CO2"
"gasmix::He/1.64kgm3"
"gasmix::He/10bar"
"gasmix::He/He_is_He3/10bar"
"gasmix::air"
"gasmix::air/-10C/0.8atm/0.30relhumidity"
```

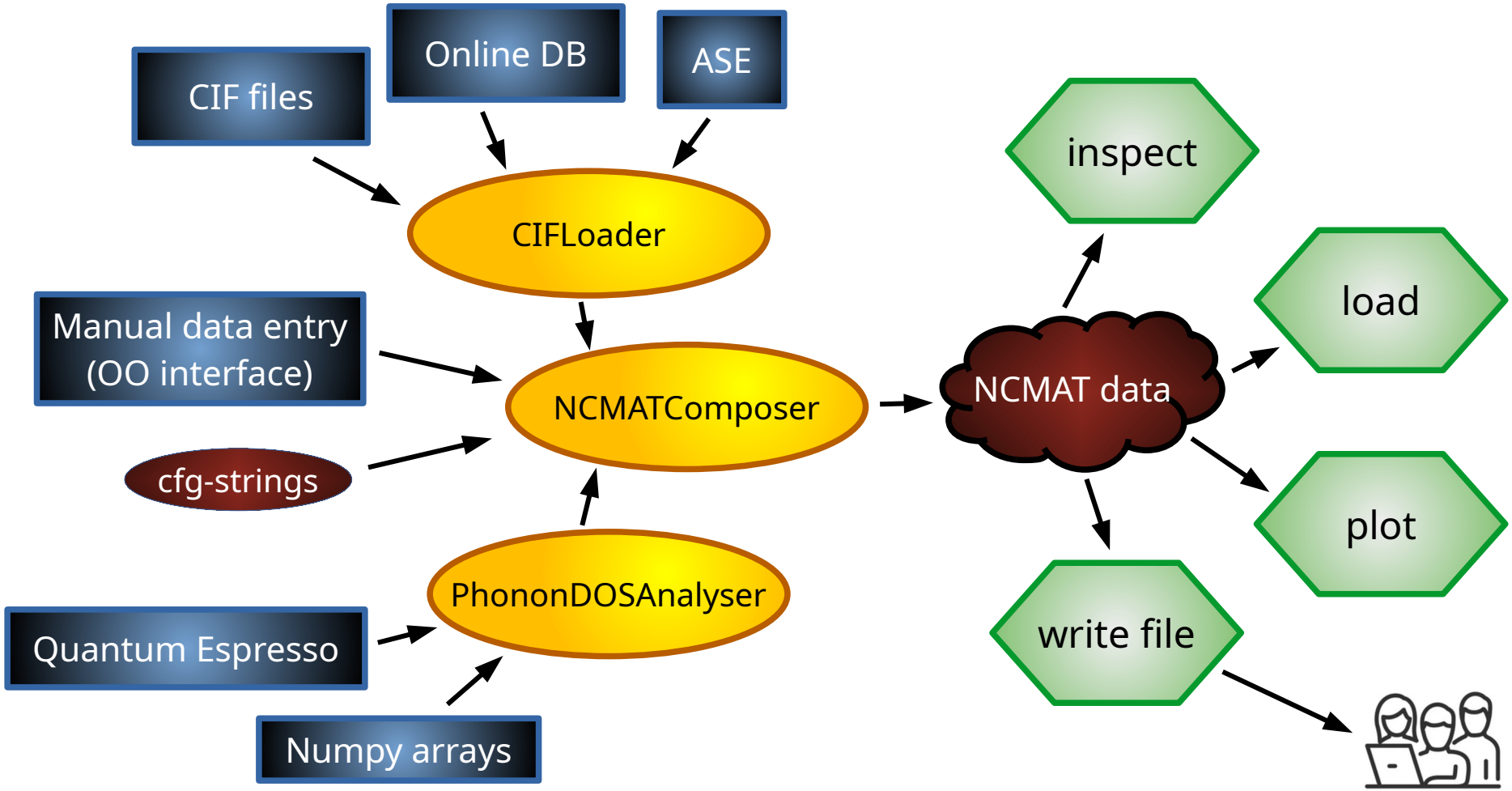

BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

Python tools for material compositon

The subject of several Jupyter notebooks.

NCrystal
Thermal Neutron Transport

- **NCrystal includes Python tools for creating new 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

Try it in the notebook:

”Data infrastructure and standard data library”

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

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 physics algorithms

A small recap of Thermal Neutron scattering theory

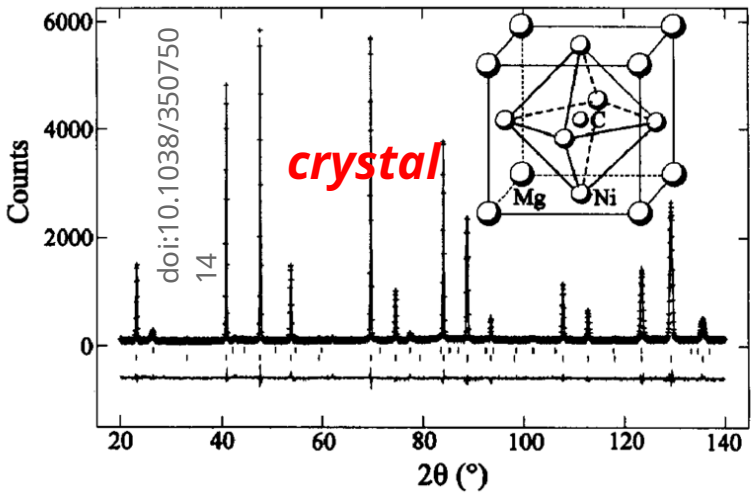
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

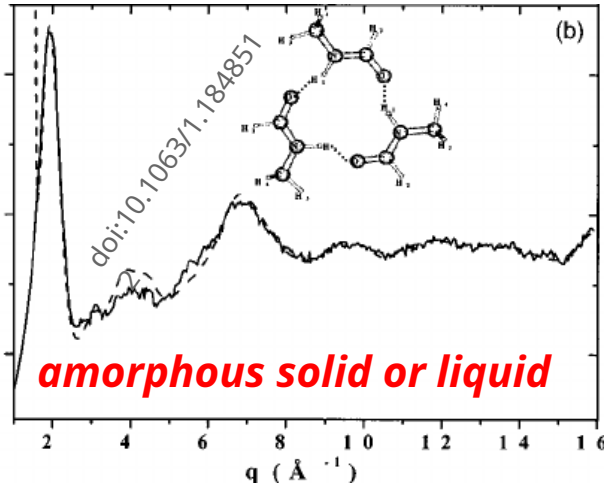
Thermal neutron scattering: Rich connection to material structure

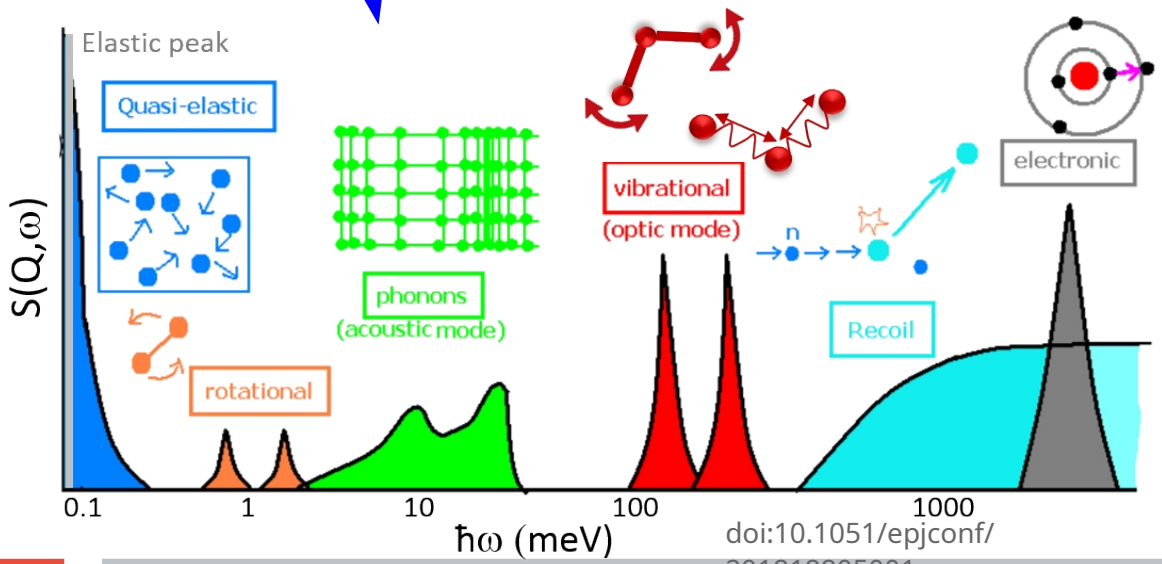
NCrystal
Thermal Neutron Transport

Thermal neutron wavelength
≈ interatomic distances.
⇒ **Sensitive to atomic positions**

Thermal neutron energy
≈ material energy levels
⇒ **sensitive to material dynamics**







doi:10.1051/epjconf/201818805001

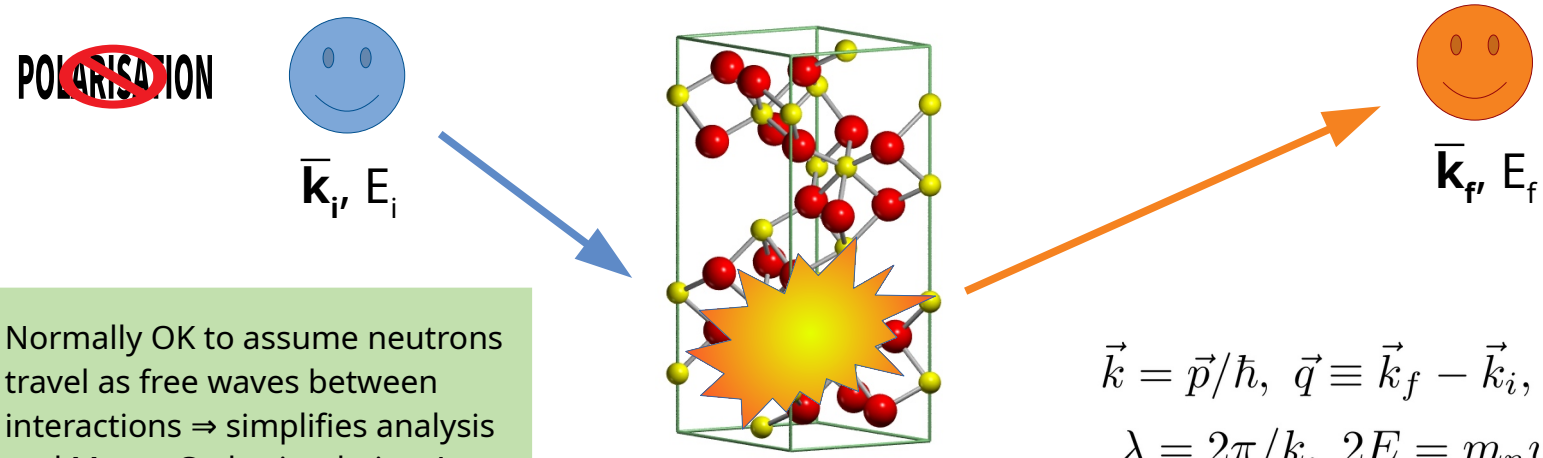
Thermal neutron scattering a uniquely useful tool for understanding material structure

On the flipside it is impossible to ignore material structure in MC simulations!

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

Thermal neutron scattering
(ignoring polarisation)

NCrystal
Thermal Neutron Transport



Normally OK to assume neutrons travel as free waves between interactions ⇒ simplifies analysis and Monte Carlo simulations!

$$\vec{k} = \vec{p}/\hbar,\ \vec{q} \equiv \vec{k}_f - \vec{k}_i,\ \Delta E \equiv E_f - E_i \equiv -\hbar\omega$$
$$\lambda = 2\pi/k,\ 2E = m_n v^2 = p^2/m_n = \hbar^2 k^2/m_n$$

Probability to scatter to given $\bar{k}_f$ given by differential cross section:

$$\frac{d^2\sigma_{\vec{k}_i \Rightarrow \vec{k}_f}}{d\Omega_f dE_f} = \frac{k_f}{k_i}\frac{1}{2\pi\hbar}\sum_{j,j'=1}^{N} b_j b_{j'} \int_{-\infty}^{\infty} dt \langle e^{-i\vec{q}\cdot\vec{\boldsymbol{R}}_{j'}(0)} e^{i\vec{q}\cdot\vec{\boldsymbol{R}}_j(t)}\rangle e^{-i\omega t}$$

N nuclei in target system

$S(\bar{q},\omega)$

Scattering function
Depends on layout and dynamics of target particles. Does *not* depend on state of incident neutron.

Scattering length of *j*th nucleus (depends on isotope & spin state)

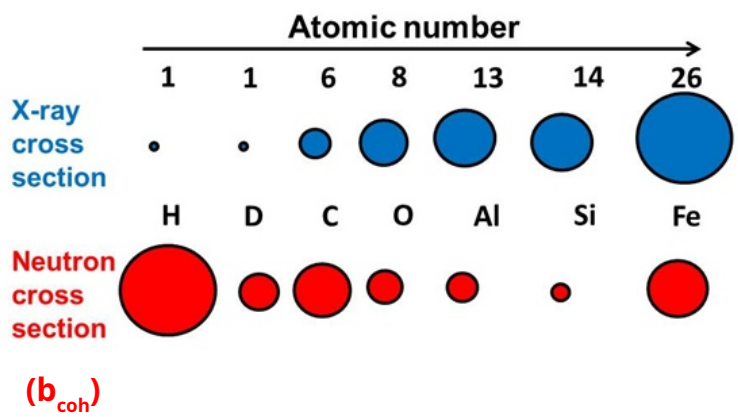
Correlate position of nucleus *j* at time 0 with position of nucleus *j'* at time t.
⇒ Neutron X.S. depends on material structure!!

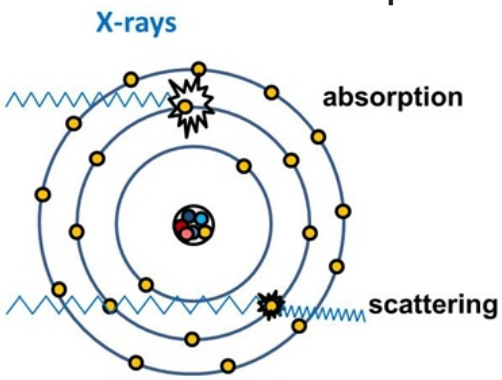
Scattering cross section depends on interference from scattering on different atoms, not just sum of 1-atom scatterings!!

Coherent scat. len. (b_{coh}) for select elements

NCrystal
Thermal Neutron Transport

X-ray strength increases with Z, neutron strength various across isotopes:





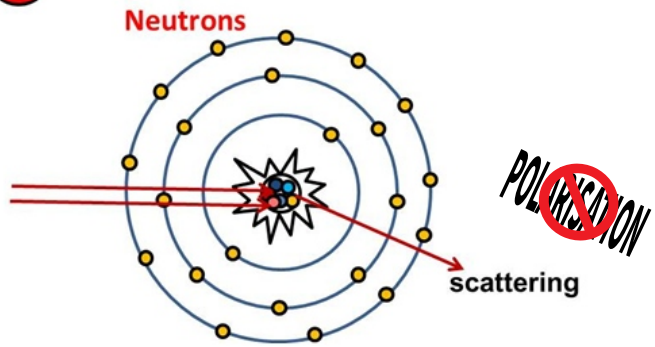
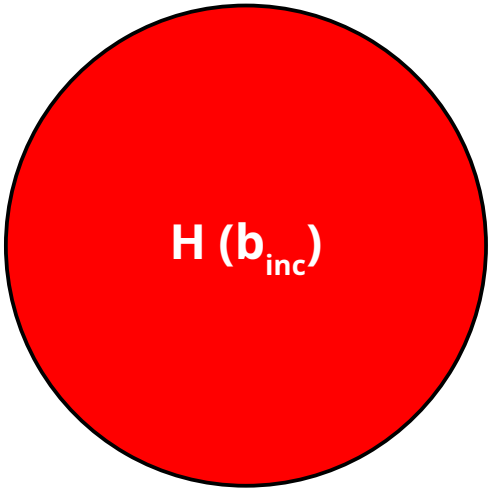


image from http://worldsciencereport.blogspot.com/2018/05/neutron-scattering.html

Note: hydrogen has huge *incoherent* scat. len. (b_{inc}):



→ Neutron scattering sensitive to structure which is otherwise inaccessible to x-rays.
→ Sample deuteration (H→D) allows tuning hydrogen scattering strength

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

Split in coherent / incoherent

NB: This is unique to neutrons, absent for x-ray scattering!

NCrystal
Thermal Neutron Transport

Most target systems can be split into statistically equivalent subsystems (e.g. unit cells for crystals). Average contribution per sub-system:

Average

shorthand for $\langle e^{-i\vec{Q}\cdot\vec{\boldsymbol{R}}_{j'}(0)} e^{i\vec{Q}\cdot\vec{\boldsymbol{R}}_{j}(t)}\rangle$

$$S(\vec{Q},\omega) \equiv \frac{1}{2\pi\hbar}\sum_{j,j'=1}^{N} \overline{b_j b_{j'}} \int_{-\infty}^{\infty} dt\,\langle j', j\rangle e^{-i\omega t}$$

(only nuclear charge (Z) important for structure, not isotope/spin state)

Using $\overline{b_j b_{j'}} = \begin{cases} \overline{b_j}\cdot\overline{b_{j'}}, & \text{for } j \neq j' \\ \overline{b_j^2}, & \text{for } j = j' \end{cases}$ and reordering terms:

$$S(\vec{Q},\omega) = S_{\text{coh}}(\vec{Q},\omega) + S_{\text{inc}}(\vec{Q},\omega)$$

$$S_{\text{coh}}(\vec{Q},\omega) \equiv \frac{1}{2\pi\hbar}\sum_{j,j'=1}^{N} \underbrace{\overline{b_j}\cdot\overline{b_{j'}}}_{\stackrel{\text{def}}{=} b_{\text{coh}}} \int_{-\infty}^{\infty} dt\,\langle j', j\rangle e^{-i\omega t}$$

$$S_{\text{inc}}(\vec{Q},\omega) \equiv \frac{1}{2\pi\hbar}\sum_{j=1}^{N} \underbrace{\left(\overline{b_j^2} - \left(\overline{b_j}\right)^2\right)}_{\stackrel{\text{def}}{=} (b_{\text{inc}})^2} \int_{-\infty}^{\infty} dt\,\langle j, j\rangle e^{-i\omega t}$$

Always 0 in X-ray scattering!

Coherent:

- Keep pair-correlations as in full S(q,w)
- Plug in per-element scat. lengths, which are the *averages* of isotopic/spin scatlens

Incoherent:

- No pair-correlation, no interference! (but still *indirect* dep. on mat. structure!)
- Just sum up separate contributions.
- Plug in per-element scat. lengths which are the *variances* of isotopic/spin scatlens

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

Elastic versus inelastic scatterings

(in the sense of ΔE=0 and ΔE≠0)

NCrystal
Thermal Neutron Transport

- Mathematically speaking, elastic scattering means factors of delta-functions: δ(ΔE)
- δ(ΔE) appears as result of static (i.e. time-independent) correlations between atomic positions:

$$S(\vec{Q},\omega) \equiv \frac{1}{2\pi\hbar}\sum_{j,j'=1}^{N}\overline{b_j b_{j'}}\int_{-\infty}^{\infty} dt\,\langle j', j\rangle e^{-i\omega t}$$

$$\langle e^{-i\vec{Q}\cdot\vec{\boldsymbol{R}}_{j'}(0)} e^{i\vec{Q}\cdot\vec{\boldsymbol{R}}_{j}(t)}\rangle$$

(positional pair-correlations)

Fourier transform of static term gives δ-function in energy

$$\int_{-\infty}^{\infty} dt e^{-i\omega t} = 2\pi\hbar\delta(\hbar\omega) = 2\pi\hbar\delta(E_f - E_i)$$

Static correlations between atomic positions, and thus ΔE=0 scatterings is a feature of solid materials (crystalline or amorphous).

Does not strictly speaking occur in liquids or gasses, although correlations in such might give rise to "quasi-elastic" scatterings peaking around ΔE=0.

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

In summary: four $S(\bar{q},\omega)$ components

NCrystal
Thermal Neutron Transport

$$S(\bar{q},\omega) = S_{coh,inel}(\bar{q},\omega) + S_{inc,inel}(\bar{q},\omega) + S_{coh,el}(\bar{q},\omega) + S_{inc,el}(\bar{q},\omega)$$

solid systems only

Depending on material and neutron energy, any of these four components can dominate!

For isotropic materials: $S(\bar{q},\omega) \rightarrow S(q,\omega)$, allowing to treat scattering via 2D function (*scattering kernels*).

In practice, only inelastic scattering is modelled via 2D scattering kernels. Elastic scattering involves delta-functions and is best described by dedicated algorithms.

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 elastic physics algorithms

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

$S_{coh,el}(\bar{q},\omega)$ in crystals: Bragg diffraction!

NCrystal
Thermal Neutron Transport

Microscopic scatter function:

sum over "crystal planes"

$$S^{el}_{coh}(\vec{Q},\omega) = \frac{(2\pi)^3\delta(\hbar\omega)}{V_{uc}} \sum_{hkl} \delta(\vec{Q}-\vec{\tau}_{hkl})\,|F(\vec{\tau}_{hkl})|^2$$

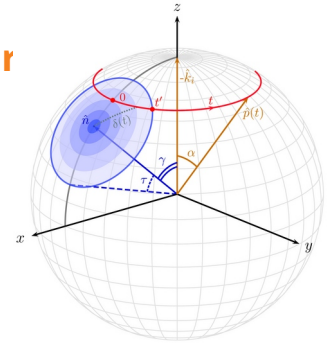
$$F(\vec{Q}) \equiv \sum_i \overline{b_i} e^{-W_i(\vec{Q})} e^{i\vec{Q}\cdot\vec{p}_i}$$

$2W=\delta^2q^2$, δ=atomic displacement

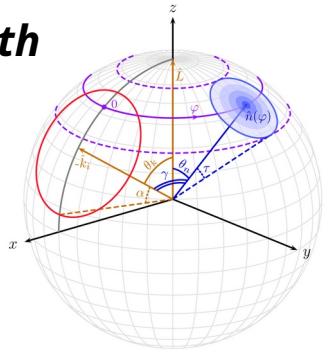
unit cell volume

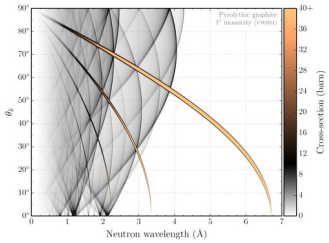
plane normal with magnitude $2\pi/d_{hkl}$

unit cell structure factor depends on layout of atoms in unit cell



Macroscopic values are found from convoluting microscopic values with crystal grain distributions → geometrical problem!



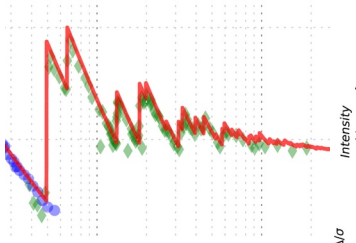


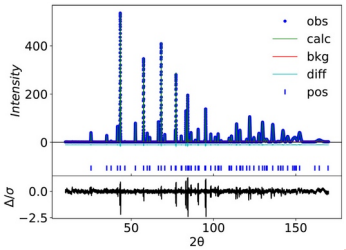
Supported geometries in NCrystal:

- *Completely disoriented layout ("powder approximation")*
- *Gaussian deviations from completely oriented ("Mosaic single crystals")*
- *Layered crystals ("rotated mosaic single crystals", pyrolythic graphite)*

Not supported yet:

- *Bent/deformed crystals, corrections for very small grain sizes.*
- *Textured crystals (as in most metals/polycrystals). But powder approximation OK for many use-cases! People have expressed interest in improving this though, get in touch if you want to contribute :-)*





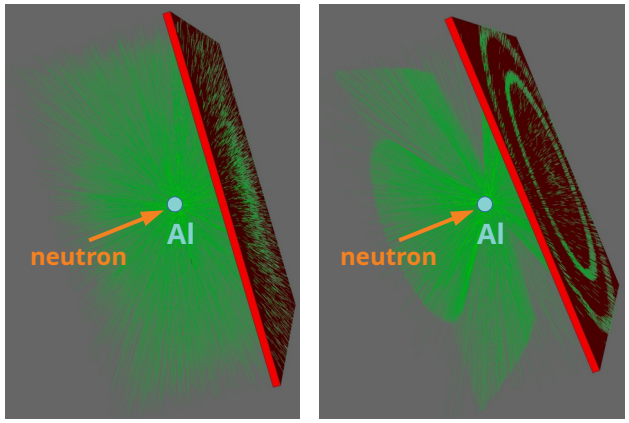
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

Bragg diffraction in powders (and texture-free polycrystals)

NCrystal
Thermal Neutron Transport

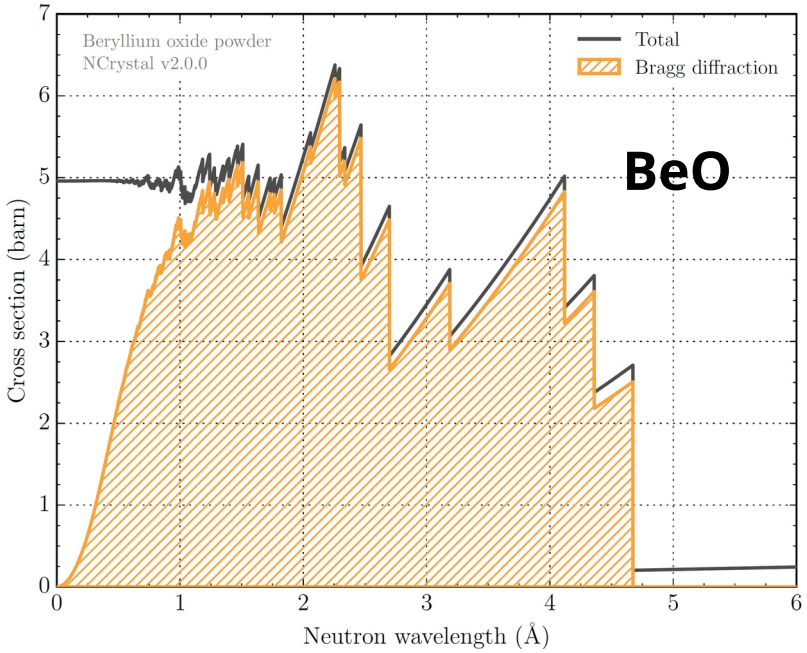
Based on provided HKL planes with d-spacings and structure factors, the implementation is straight-forward.

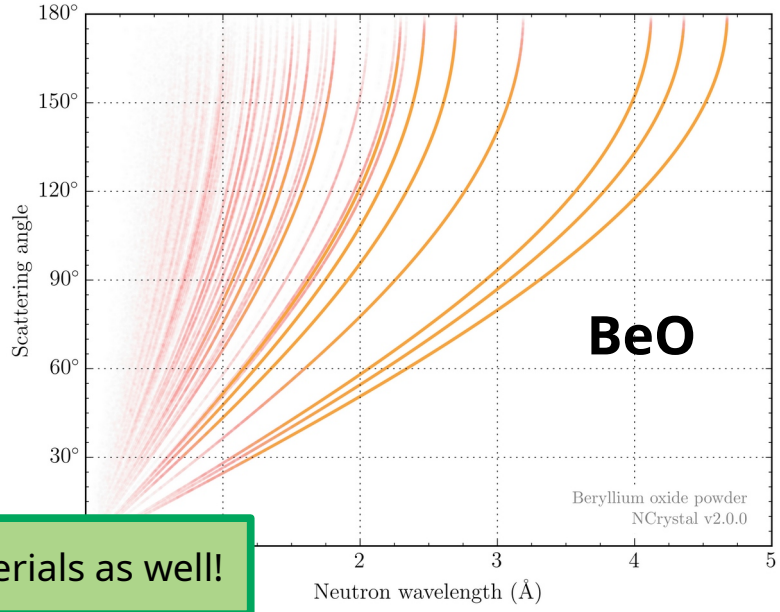
Care is taken to be extremely fast O(10ns/call), even in case of huge number of planes.
Currently no texture/grain-size effects.



Geant4 free-gas model (wrong MFP, wrong scatter)

Geant4 with NCrystal ⇒Debye-Scherrer cones



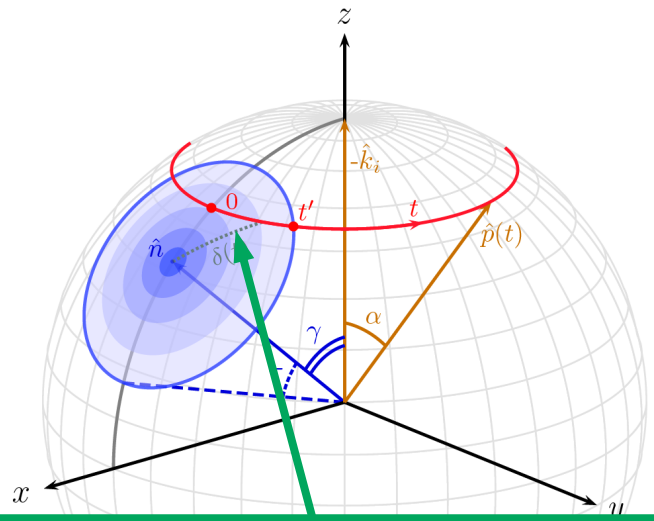


There is interest in extending this to textured materials as well!

Single Crystals with Gaussian mosaicity

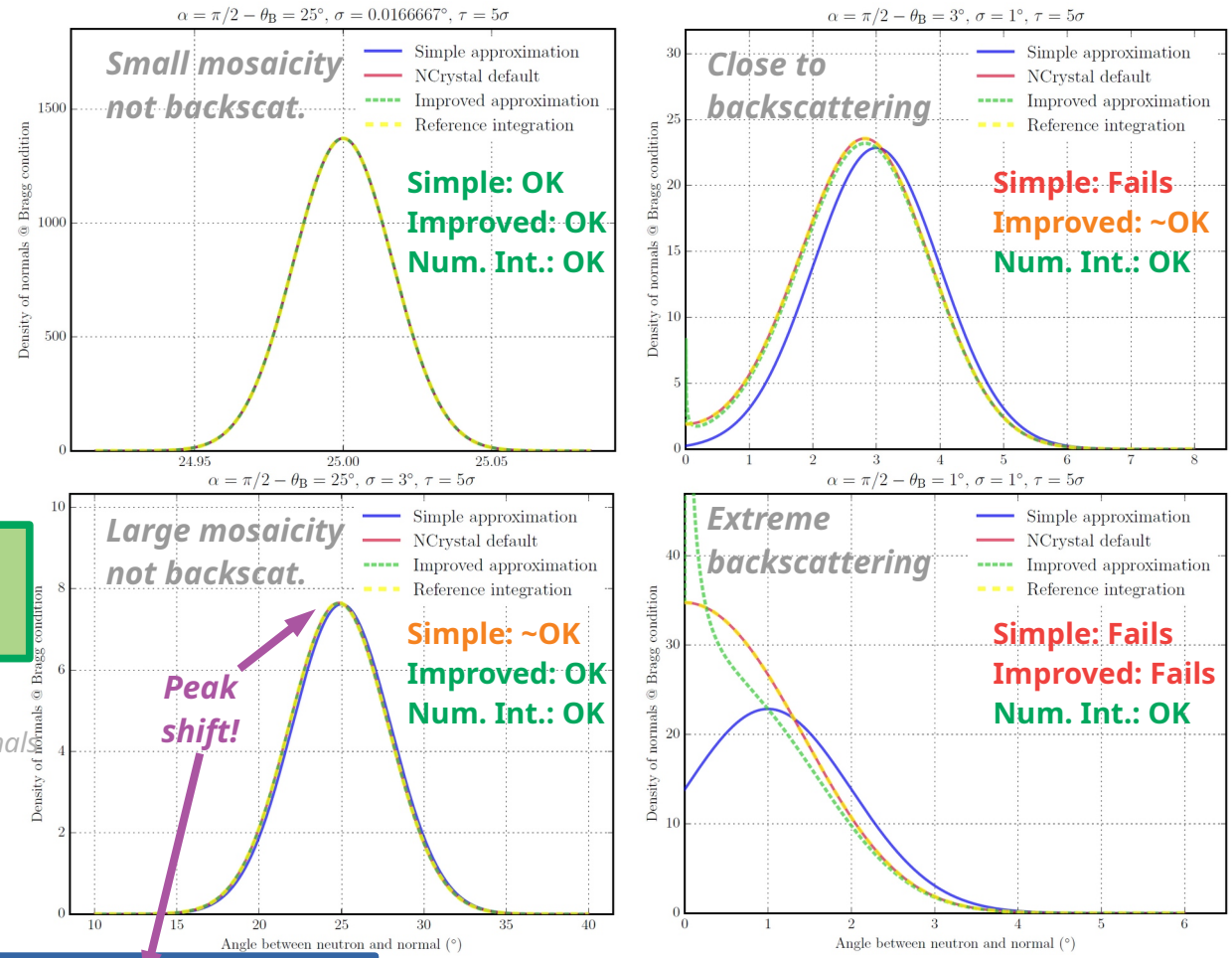
Can model monochromators, analysers, filters, samples
Handles also large mosaicities and backscattering!

NCrystal
Thermal Neutron Transport



The tricky part* is the integration of mosaic density along circle of Bragg condition.

*: Once contributing normals Have been identified.



Simple closed-form approx. valid for small mosaicity (and not backscattering):

$$\sigma_{\text{Bragg}}(\alpha,\gamma) = Q \times \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\tfrac{1}{2}\delta_0^2/\sigma^2\right] \times \operatorname{erf}\left[\sqrt{\frac{\tau^2-\delta_0^2}{2\sigma^2}}\right] \times \sqrt{\frac{\sin\alpha}{\sin\gamma}} \times \frac{N}{1/(2\pi\sigma^2)}$$

$$\delta_0 = |\alpha - \gamma|$$

Our improved form extends validity to much larger mosaicities

Code picks appropriate method from:

- Our closed form approximation
- Full numerical integration

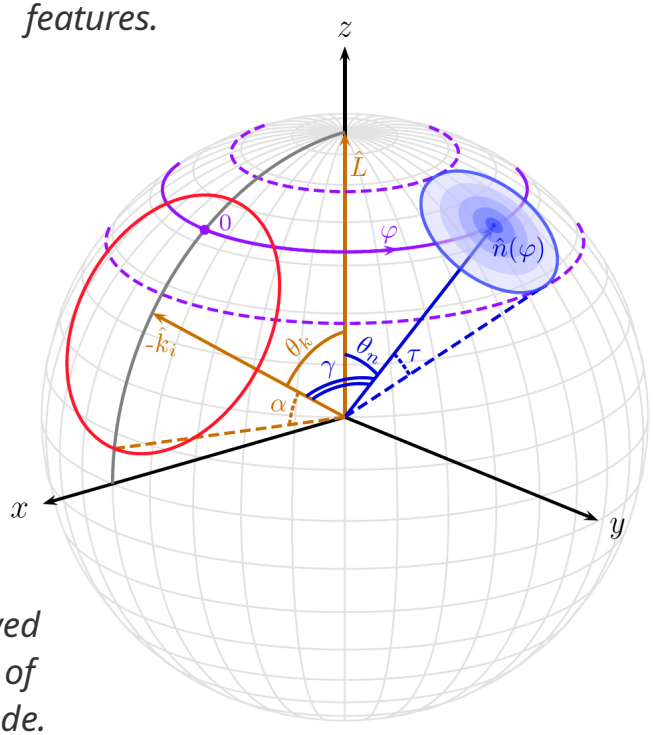
Special anisotropic model for Pyrolytic Graphite

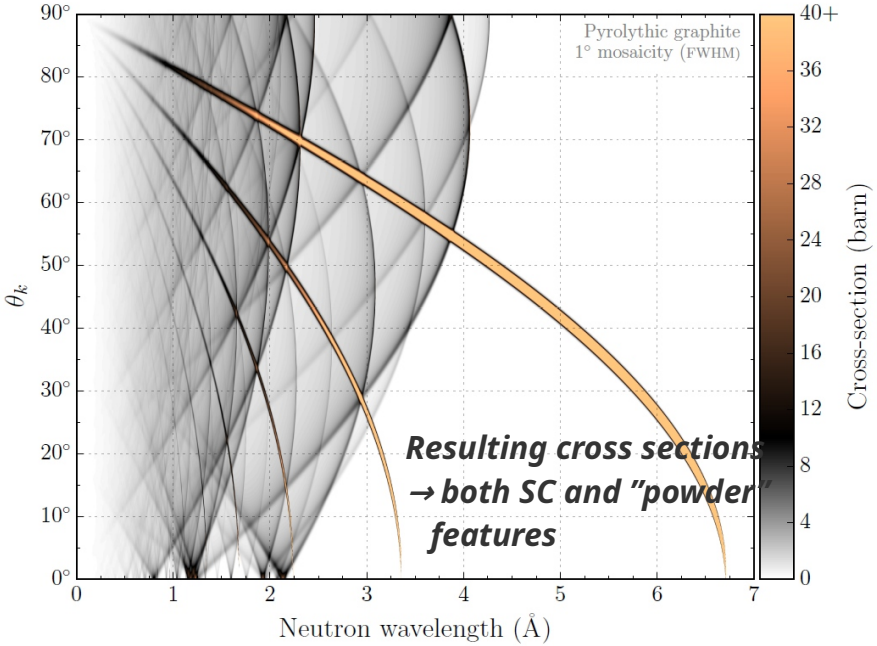
PG often used as filters, monochromator, analyser

NCrystal
Thermal Neutron Transport

Layered crystal model:

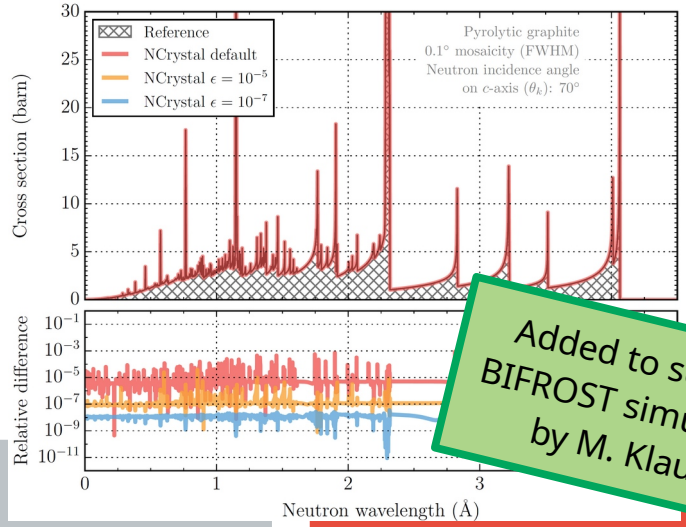
- *Usual Gaussian mosaic distribution is "smeared out" by rotation*
- *Exhibits both single-crystal and powder features.*





Features:

- *Cross-sections determined by efficient pre-search followed by fast Romberg integration of non-layered single crystal code.*
- *Features realistic transmission probabilities and multiple-scattering effects (incl. "zig-zag walk")*



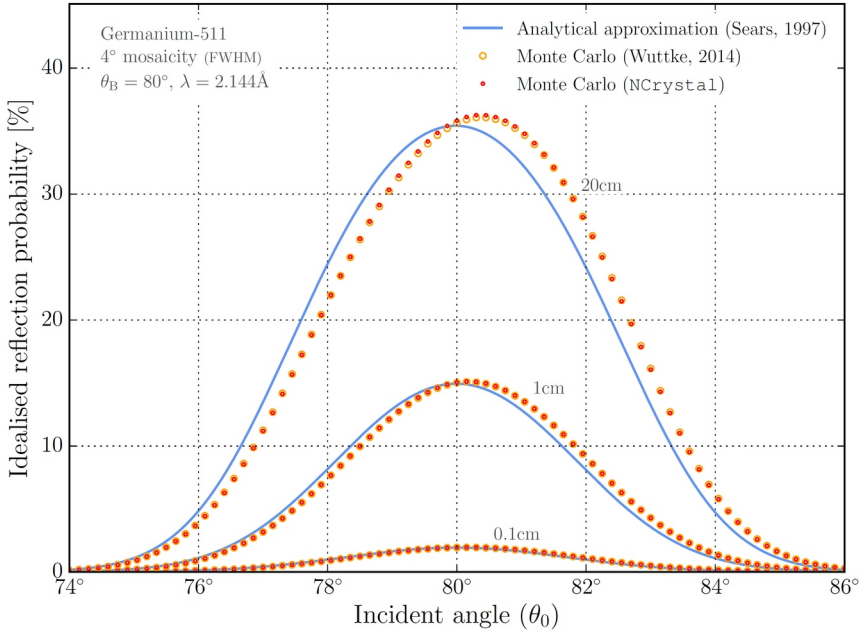
Added to support BIFROST simulations by M. Klausz.

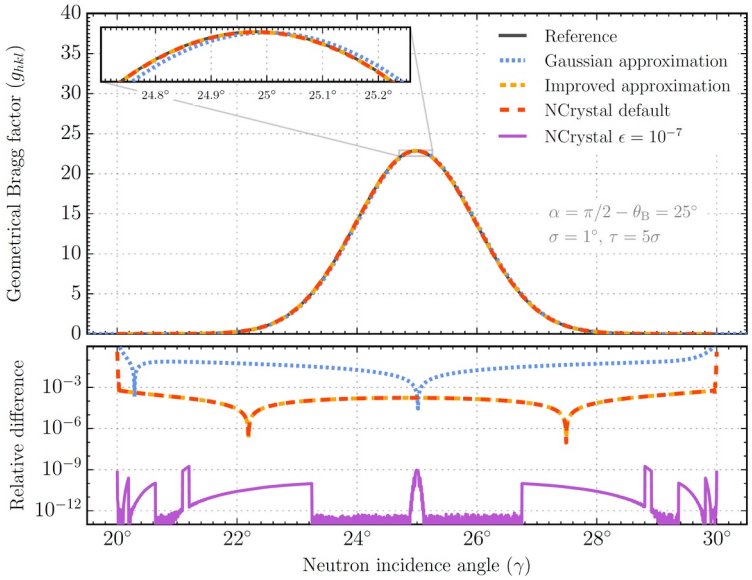
Single crystal model validated

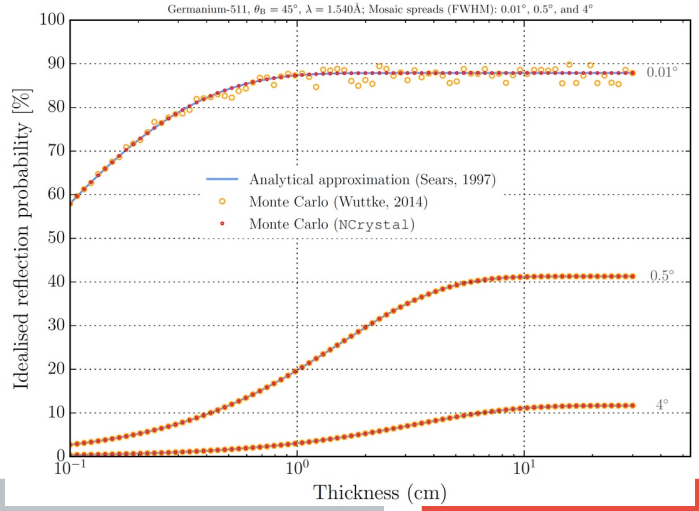
NCrystal
Thermal Neutron Transport

Validation includes:

- Against existing codes (Wuttke2014) or analytical results (Sears1997) in their domains of validity.
- Against (very) slow but simple+precise implementation (using mpmath high-precision math module)
- Technical validations (zig-zag, “powdered”)







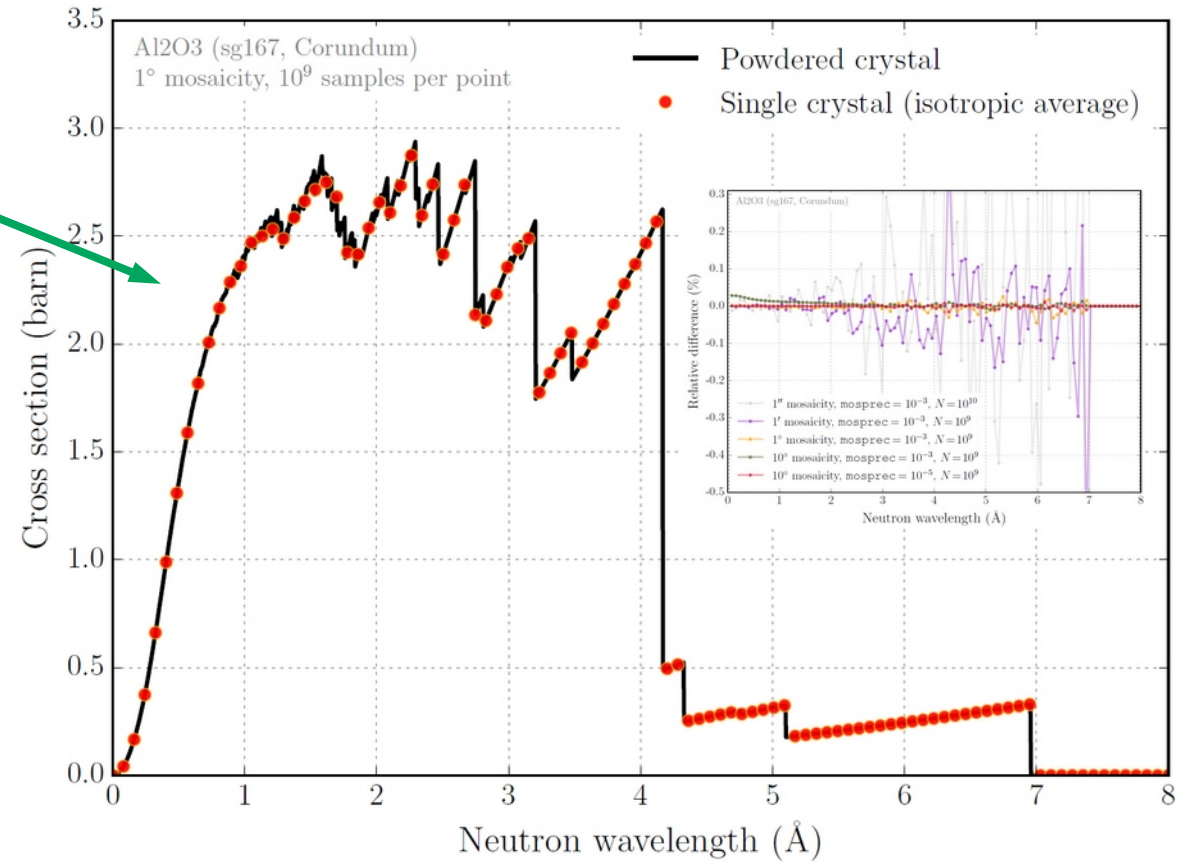
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

Technical checks for consistency of single crystal codes (should also hold for texture models)

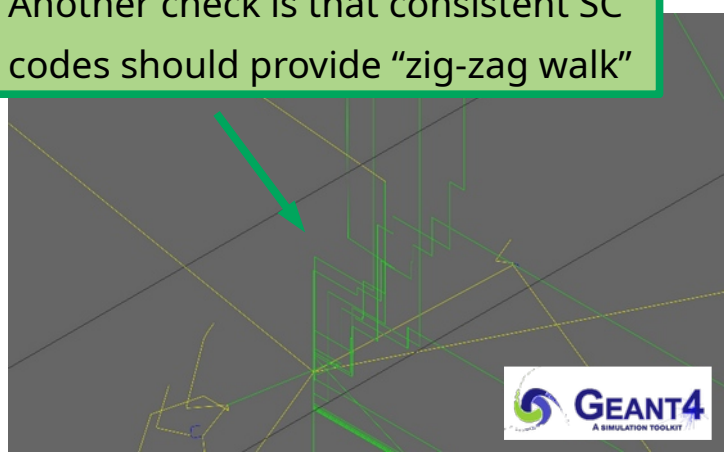
NCrystal
Thermal Neutron Transport

In principle an isotropically illuminated single crystal should on average give powder-like cross-sections.

In practice, a lot of edge-cases and details have to be treated correctly in the SC code before this happens!



Another check is that consistent SC codes should provide "zig-zag walk"

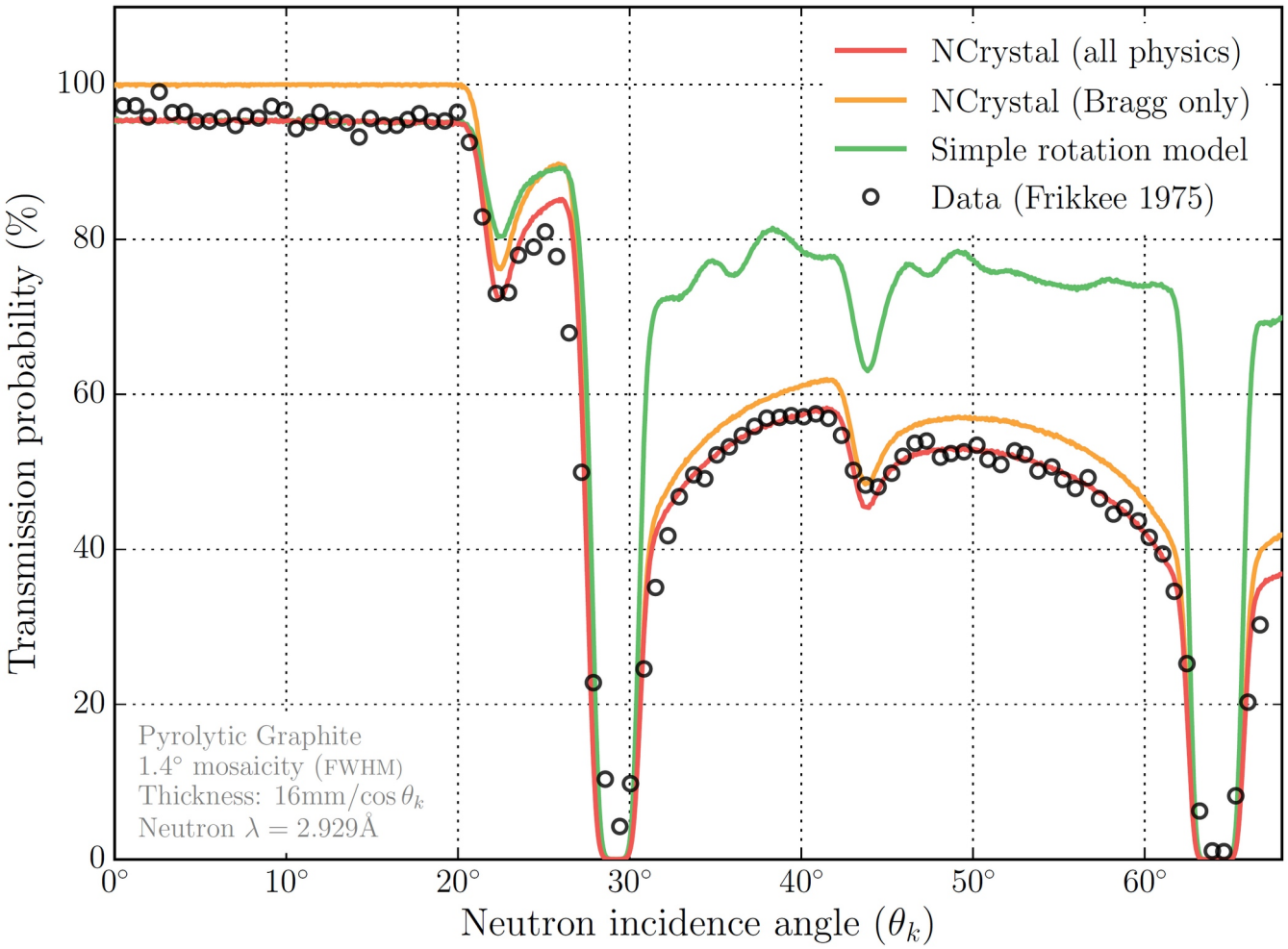


Thanks to the DMSC cluster for help with this brute force validation.

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

Can reproduce PG transmission spectra!

NCrystal
Thermal Neutron Transport



Validation also includes:

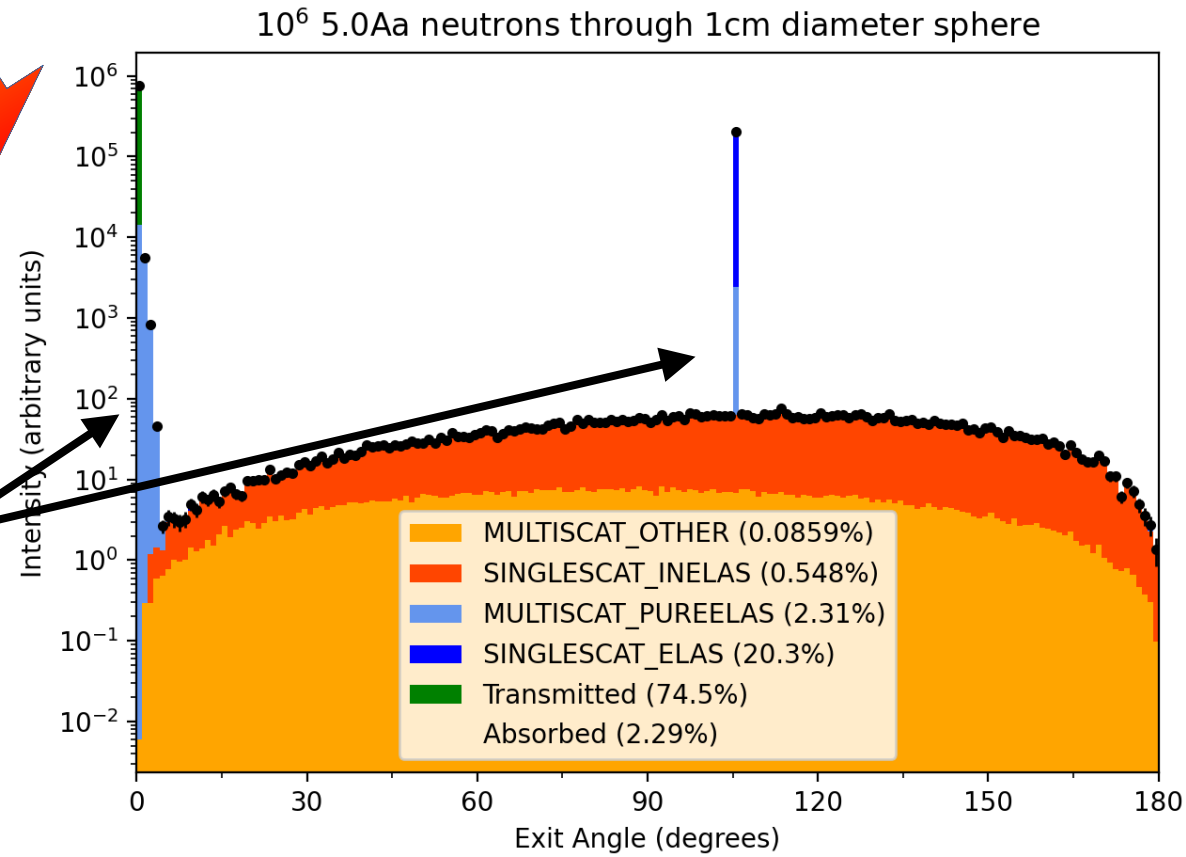
- Comparison against (very very) slow but simple+precise implementation.
- Verification that cross section maxima structure matches predictions (Frikkee1975).
- Technical validations (zig-zag, “powdered”).

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

Embedded “Mini MC” also supports single crystals (here Si111 monochromator)

NCrystal
Thermal Neutron Transport

```
$> nctool --mc 5.0Aa 1cm \
     “Si_sg227.ncmat;dir1=@crys_hkl:1,1,1@lab:0,3,1;dir2=@crys_hkl:0,0,1@lab:0,0,1;dirtol=180deg;mos=1deg”
```



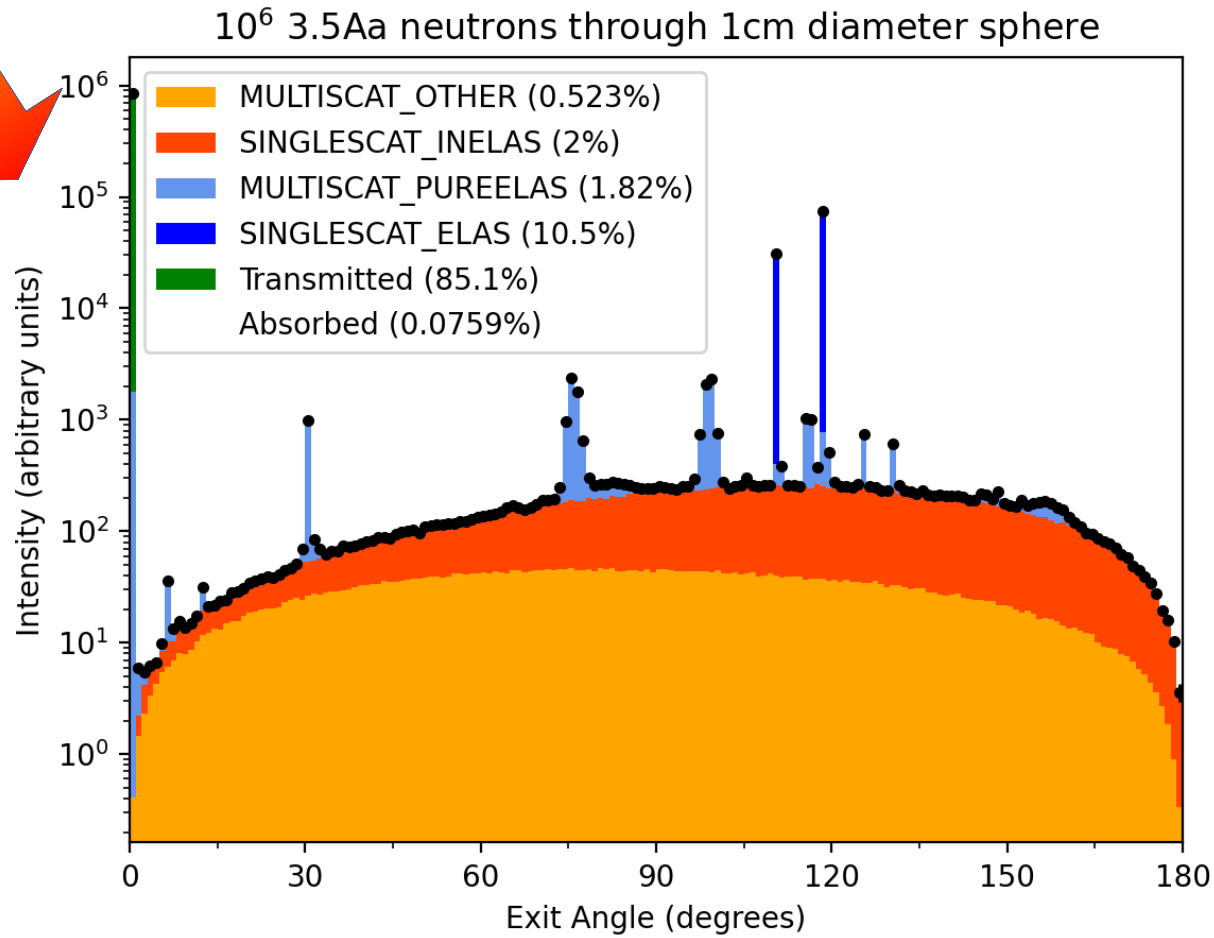
“Zig-zag” multiple scattering effect enhances both transmission and the SC Bragg reflection

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

Pyrolytic graphite in “Mini MC”

NCrystal
Thermal Neutron Transport

```
$> nctool --mc 3.5Aa 1cm """C_sg194_pyrolytic_graphite.ncmat
     ;dir1=@crys_hkl:0,0,2@lab:0,3,1;mos=0.2deg;
     ;dir2=@crys_hkl:1,1,1@lab:0.1,0,1;dirtol=180deg"""
```



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 scattering

NCrystal
Thermal Neutron Transport

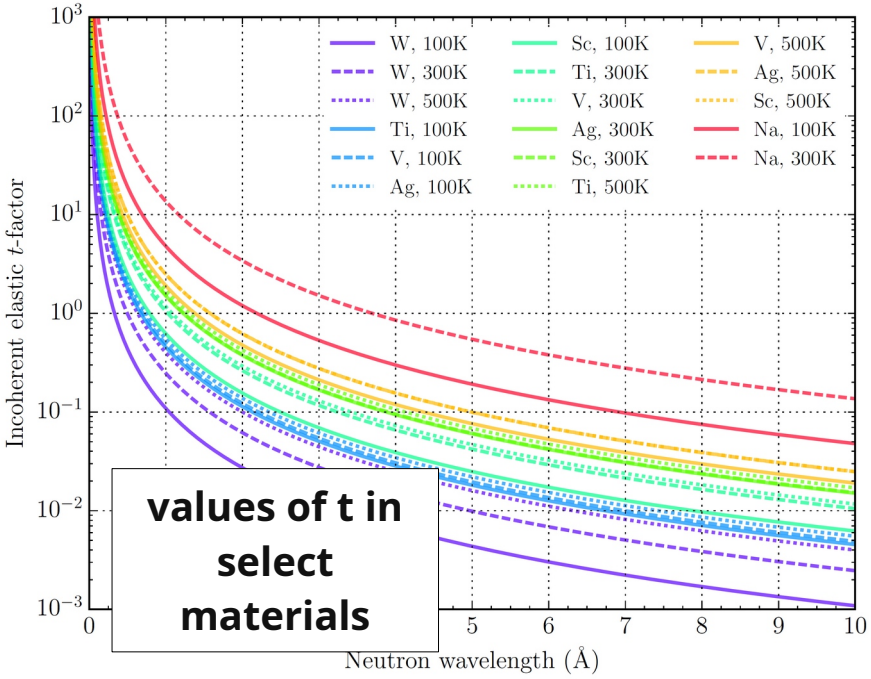
$$\frac{d\sigma^{\mathrm{inc,el}}_{\vec{k}_i\to\vec{k}_f}}{d\Omega_f} = \frac{1}{N}\sum_{j=1}^{N}\frac{\sigma_j^{\mathrm{inc}}}{4\pi}e^{-2W_j(\vec{Q})}$$

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

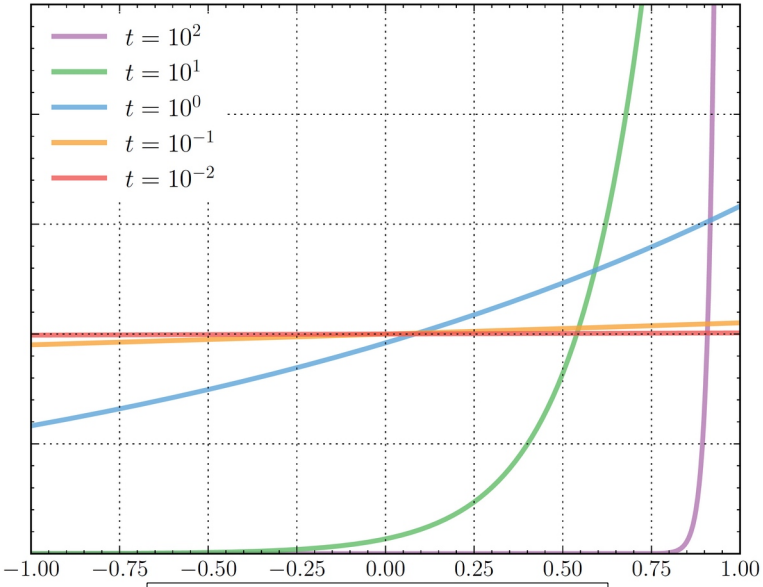
$$P(\mu) = N_t \exp\left(\frac{t\mu}{2}\right)$$

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

Get Debye-Waller factors (or δ²) from phonon DOS (or Debye temp.).



values of t in select materials



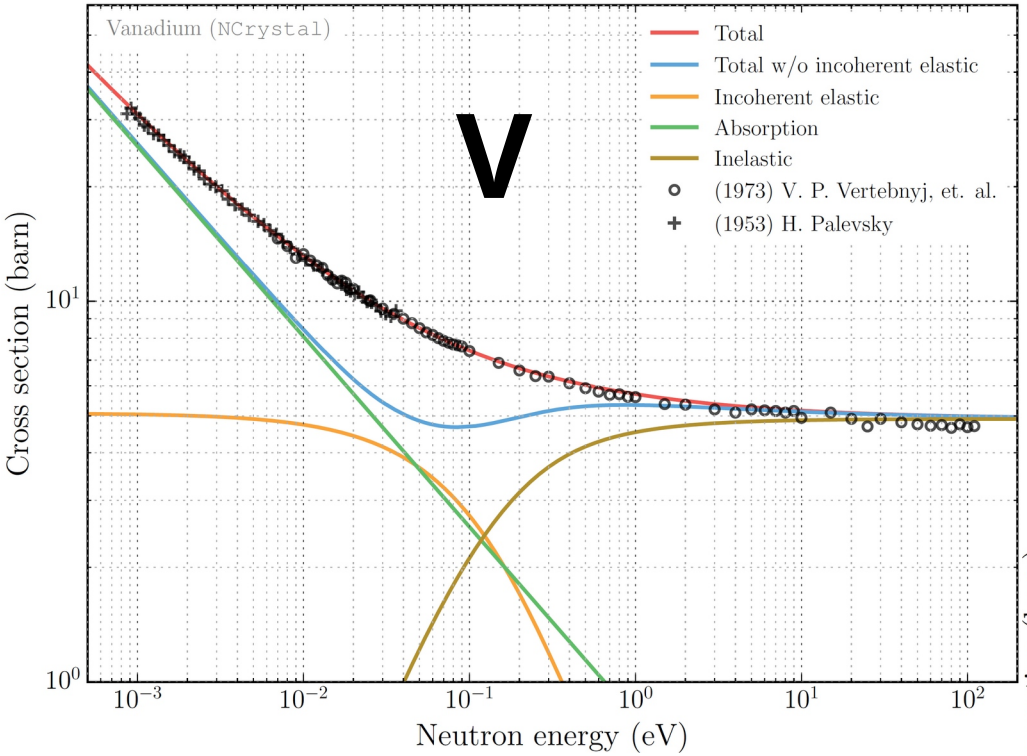
cos(θ)=μ distribution (isotropic=flat)

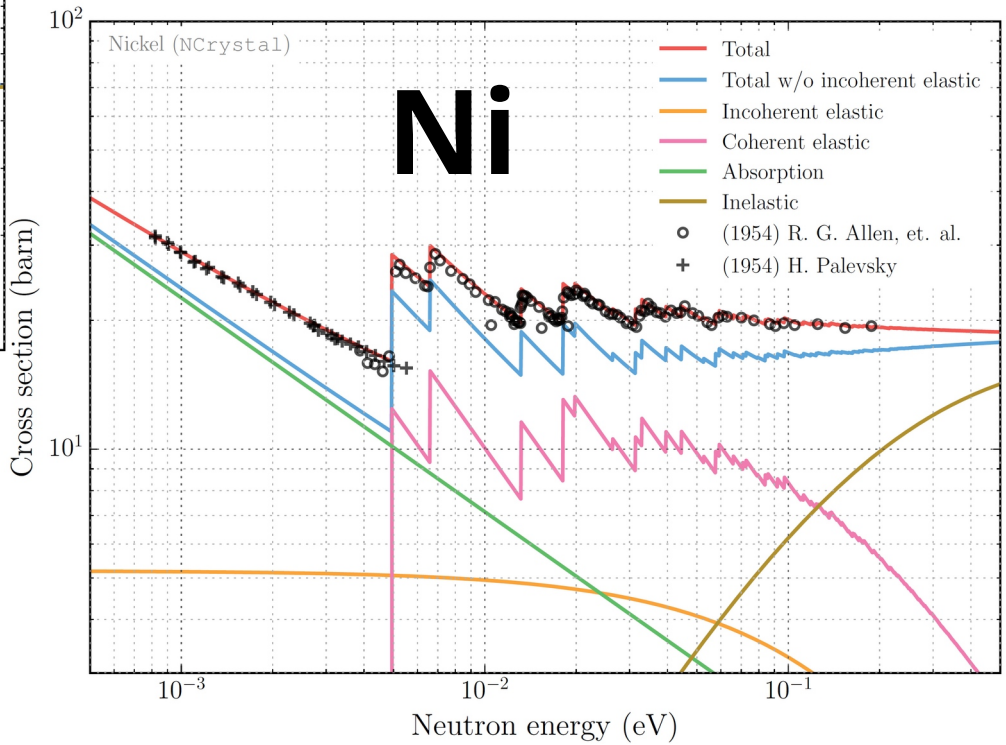
→ Not always completely isotropic!!

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
→ crucial for many materials

NCrystal
Thermal Neutron Transport





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

First half of the notebook: "Creating materials and the NCMATComposer"

(to "Specifying dynamic information")

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

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 inelastic physics algorithms

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 : Scattering kernels

NCrystal
Thermal Neutron Transport

$$\frac{d^2\sigma_{\vec{k}_i \Rightarrow \vec{k}_f}}{d\Omega_f dE_f} = \frac{k_f}{k_i} S(\vec{Q}, \omega)$$

$$S(\vec{Q}, \omega) = S_{\text{coh}}(\vec{Q}, \omega) + S_{\text{inc}}(\vec{Q}, \omega)$$

$$S_{\text{coh}}(\vec{Q}, \omega) \equiv \frac{1}{2\pi\hbar} \sum_{j,j'=1}^{N} \overline{b_j} \cdot \overline{b_{j'}} \int_{-\infty}^{\infty} dt \langle j', j \rangle e^{-i\omega t}$$

$$S_{\text{inc}}(\vec{Q}, \omega) \equiv \frac{1}{2\pi\hbar} \sum_{j=1}^{N} \left(\overline{b_j^2} - \left(\overline{b_j}\right)^2 \right) \int_{-\infty}^{\infty} dt \langle j, j \rangle e^{-i\omega t}$$

Under some assumptions $S(\overline{Q},w)$ can be described with a single "smooth" 2D function (one per atom type):

- Elastic scattering is dealt with separately (as it is in NCrystal).
- *Isotropic material* ($\overline{Q}$ dependency becomes scalar Q)
- *Incoherent approximation*: Off-diagonal entries in S_{coh} wash out when integrating over isotropic grain distribution, so shape(S_{coh}) ≈shape(S_{inc}).

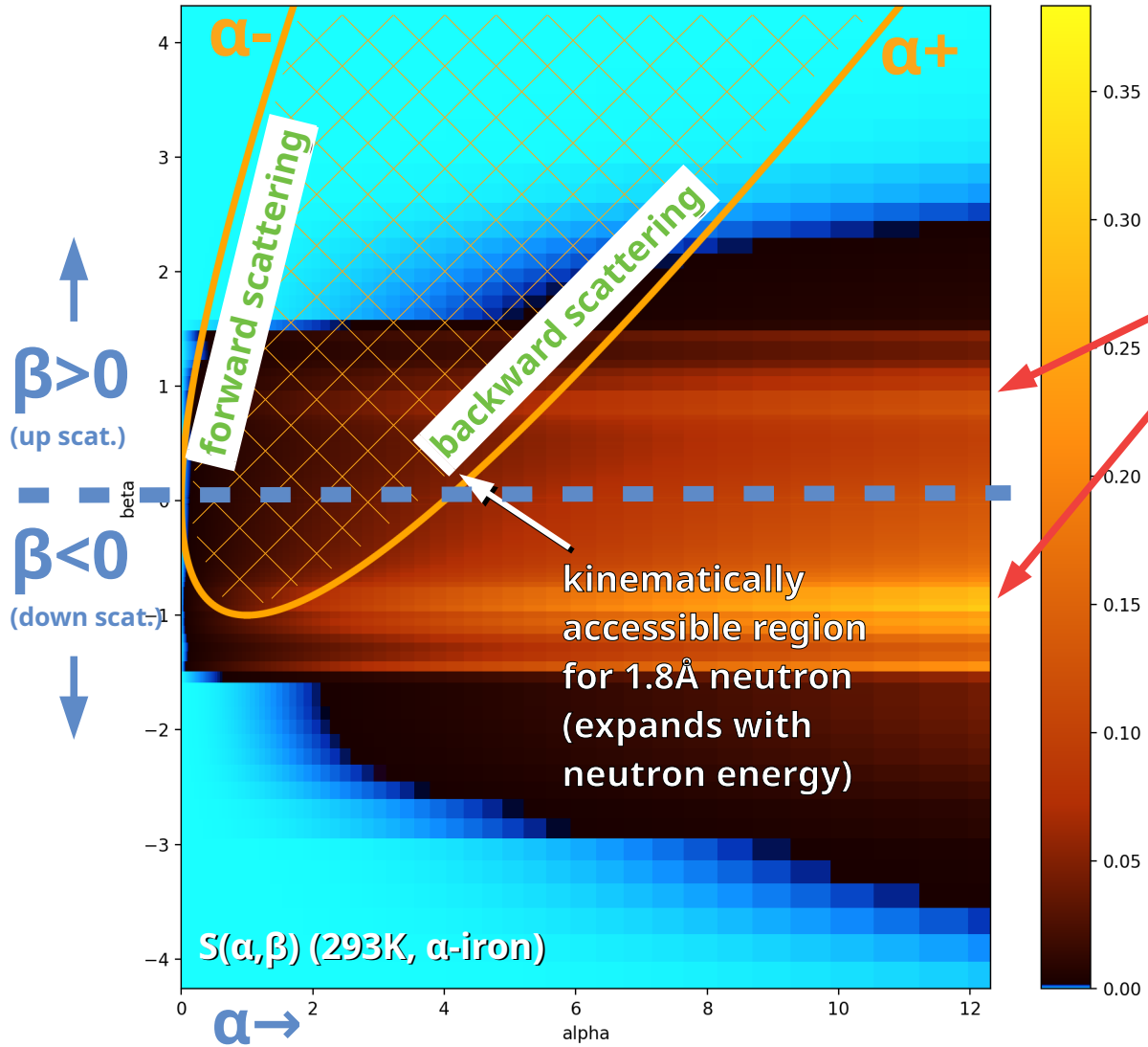
Tabulate this function on a grid → **scattering kernel**.

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

Scattering kernel and connection to neutron

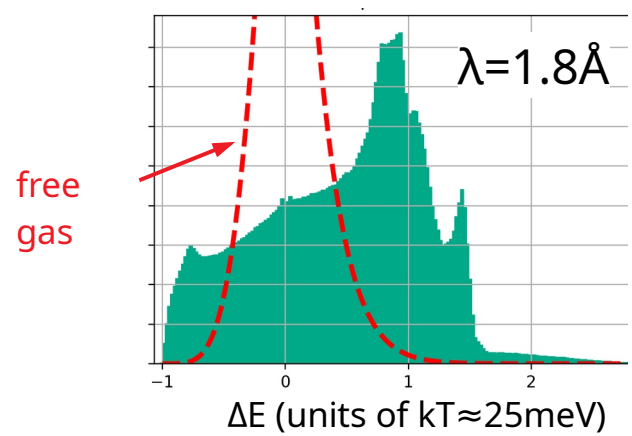
S(q,ω) ~ S(α,β); α: dimensionless q^2, β: dimensionless ω (~ΔE)

NCrystal
Thermal Neutron Transport



"Detailed balance":
Boltzmann factor + identical strength of phonon absorption & emission → Symmetry around β=0, given by S(α,-β)=exp(β)·S(α,β)

β-projection of kinematically accessible region → energy transfer spectrum (green):



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

Formulation in dimensionless variables: α,β

(q,ω) preferred in neutron scattering community, (α,β) preferred in nuclear industry (incl. MCNP, ENDF, ...)

NCrystal
Thermal Neutron Transport

α: dimensionless q^2
β: dimensionless ΔE

$$\beta = \frac{E_f - E}{kT} = -\frac{\hbar\omega}{kT}$$

$$\alpha = \frac{\hbar^2}{2m_n kT} q^2 = \frac{E + E_f - 2\mu\sqrt{EE_f}}{kT} = \frac{2E}{kT} + \beta - 2\mu\sqrt{\frac{E}{kT}\left(\frac{E}{kT} + \beta\right)}$$

$\mu = \cos\theta_{\mathrm{scat}}$

Scattering lengths taken outside definition of S:

$$\frac{\mathrm{d}^2\sigma}{\mathrm{d}E_f \mathrm{d}\Omega} = \sqrt{\frac{E_f}{E}} \frac{\sigma_b}{4\pi} \frac{S(\alpha,\beta)}{k_b T}$$

Total cross section, with explicit kinematic limits:

μ=-1, complete backwards scattering

$$\sigma(E) = \frac{\sigma_{\mathrm{b}} kT}{4E} \int_{-E/kT}^{\infty} \int_{\alpha_-(E,\beta)}^{\alpha_+(E,\beta)} S(\alpha,\beta) d\alpha d\beta$$

constant affects σ(E), but not (α,β)-sampling

neutron lose all its energy

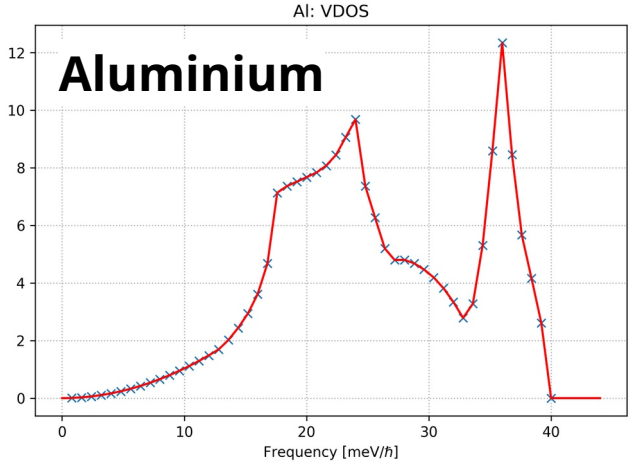
μ=+1, complete forward scattering

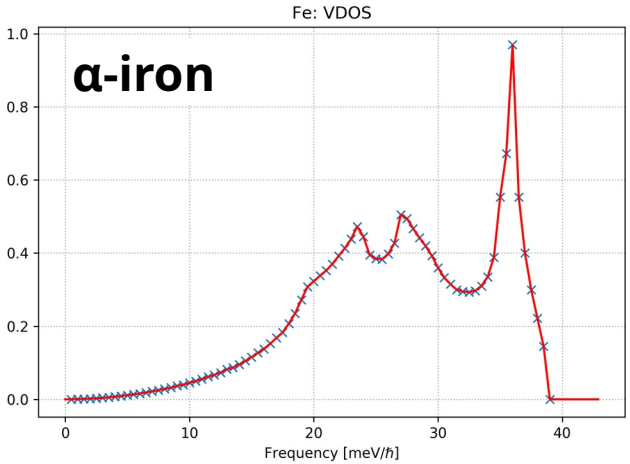
$$\alpha_\pm(E,\beta) = \frac{2E}{kT} + \beta \pm 2\sqrt{\frac{E}{kT}\left(\frac{E}{kT} + \beta\right)}$$

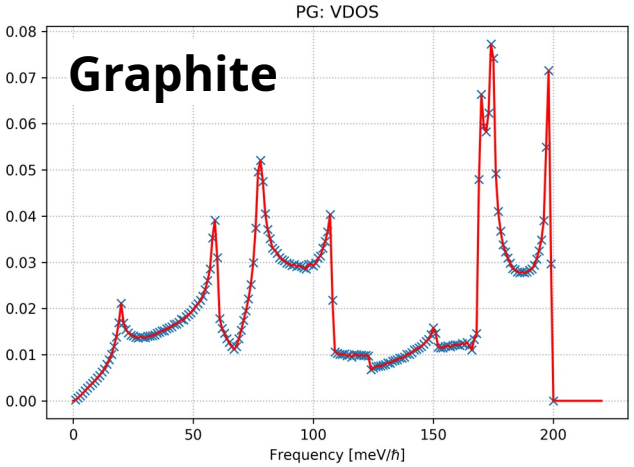
kinematically accessible region is a parabola in the (α,β)-plane

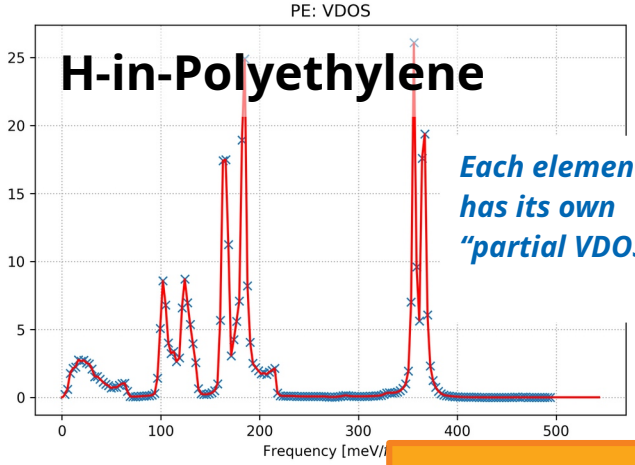
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

Solids: Scattering kernels are connected to phonon frequency spectrums (aka Vibrational Density Of States, VDOS)

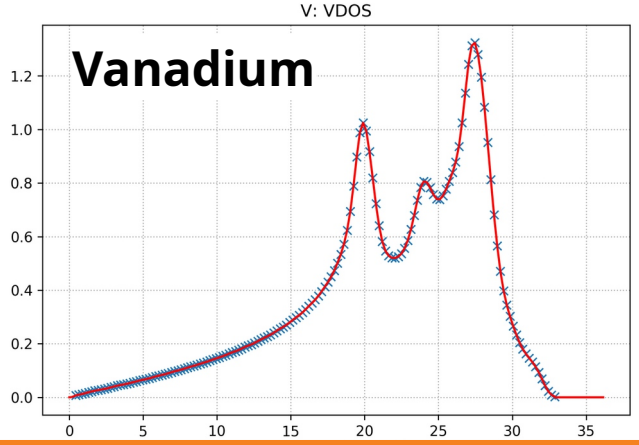




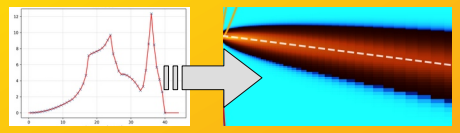




Each element has its own "partial VDOS"



NCrystal can expand VDOS to scattering kernels on-the-fly in O(100ms).



Captures material structure info relevant for *inelastic* neutron scatterings (isotropic materials, incoherent approximation)
+ gives displacements (Debye-Waller factors) needed for *elastic* scatterings.

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

Phonon spectrum (VDOS) sources

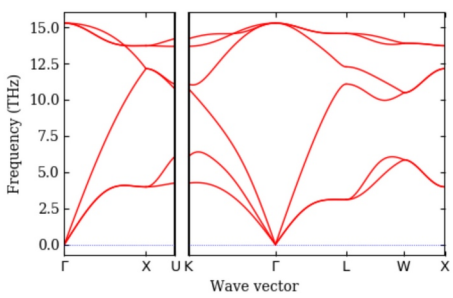
NCrystal
Thermal Neutron Transport

VDOS is not specific to neutrons!
→ Many resources exists

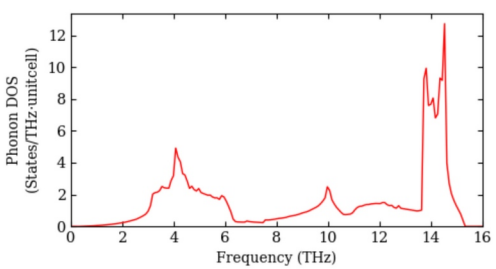
Materials id 149 / Si / Fd-3m (227)

- Date page updated: 2018-4-17
- Space group type: Fd-3m (227) / F 4d 2 3 -1d
- Number of formula units (Z): 8
- Phonon raw data: `mp-149-20180417.tar.lzma`
- Link to Materials Project: https://www.materialsproject.org/materials/mp-149/

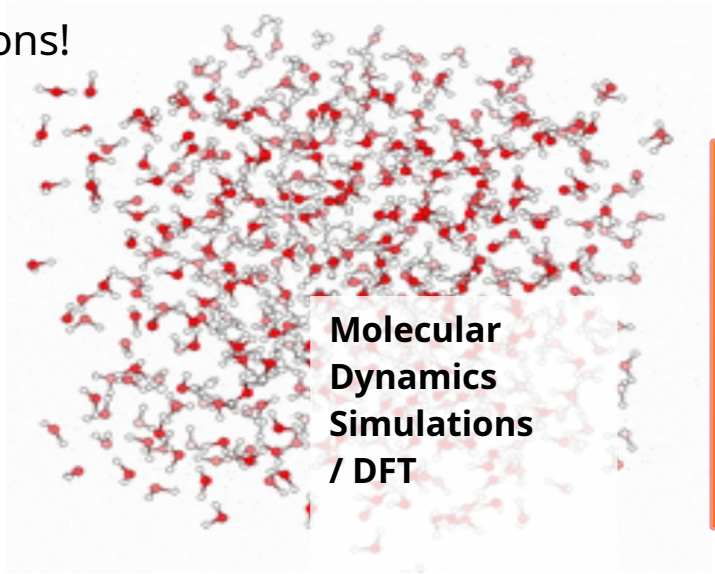
Phonon band structure



Phonon DOS



http://phonondb.mtl.kyoto-u.ac.jp/



VDOS can also be measured experimentally...

... or dug out from old research papers!

Or a combination, potentially using other SW (QuantumEspresso / VASP / phononpy / Oclimax / ...)

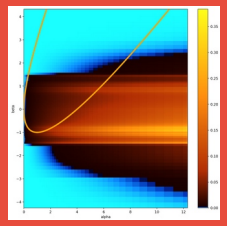
Davide Campi's lectures at the 2023 HighNESS school contains more information: https://indico.ess.eu/event/3096/

QUANTUMESPRESSO

Many VDOS curves need a bit of processing and cleanup to be used.

This can be done in Python using the PhononDOSAnalyser class provided with NCrystal.

NCrystal has unique features for *using* scattering kernels



NCrystal
Thermal Neutron Transport

- Given S(α,β) values on a grid (α_i,β_i) and a neutron with energy, E, we must define suitable interpolation scheme to provide S(α,β) for any (α,β) value, and be able to:

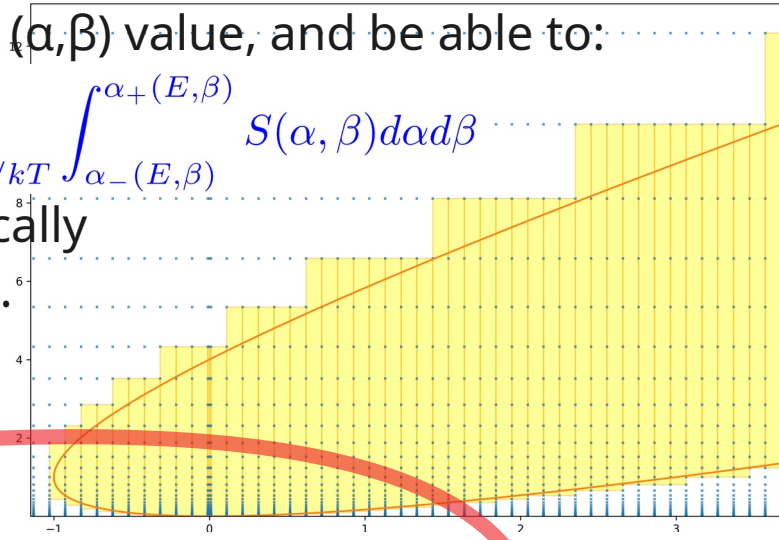
  1) Estimate scattering cross section: $\sigma(E) = \frac{\sigma_b kT}{4E} \int_{-E/kT}^{\infty} \int_{\alpha_-(E,\beta)}^{\alpha_+(E,\beta)} S(\alpha,\beta) d\alpha d\beta$

  2) Sample (α,β) values randomly within the kinematically accessible region, with density proportional to S(α,β).

- This must be done accurately and with reasonable computing resources! **Tricky part is sampling.**

- NCrystal has novel method for accurate+fast sampling, without ACE-like discretisation, with attention to near-endpoint sampling (crucial for ultra-cold neutron moderator studies).

- **We have ideas to further improve this** (make it faster, remove remaining artifacts).



Journal of Computational Physics 380 (2019) 400–407

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Rejection-based sampling of inelastic neutron scattering

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

[a] Technical University of Denmark, Denmark
[b] European Spallation Source ERIC, Sweden
[c] Nuclear Data Group, Neutron Physics Department, Centro Atómico Bariloche, CNEA, Argentina

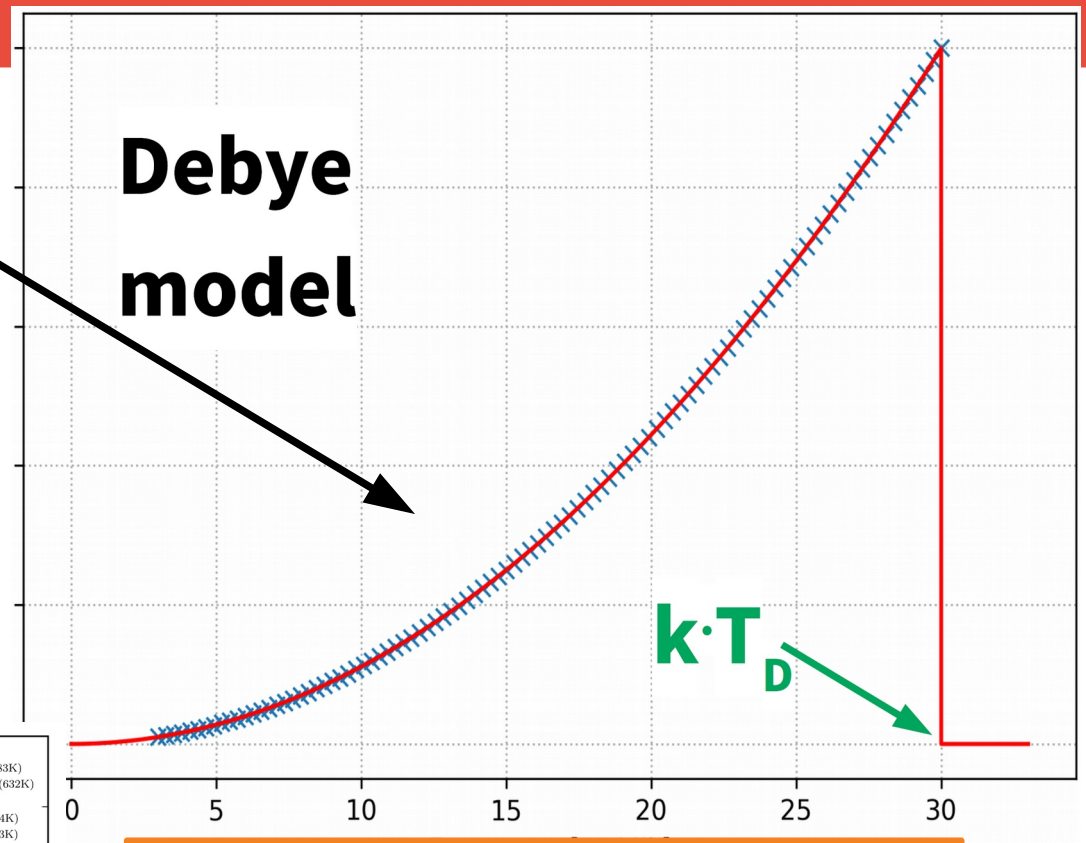
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

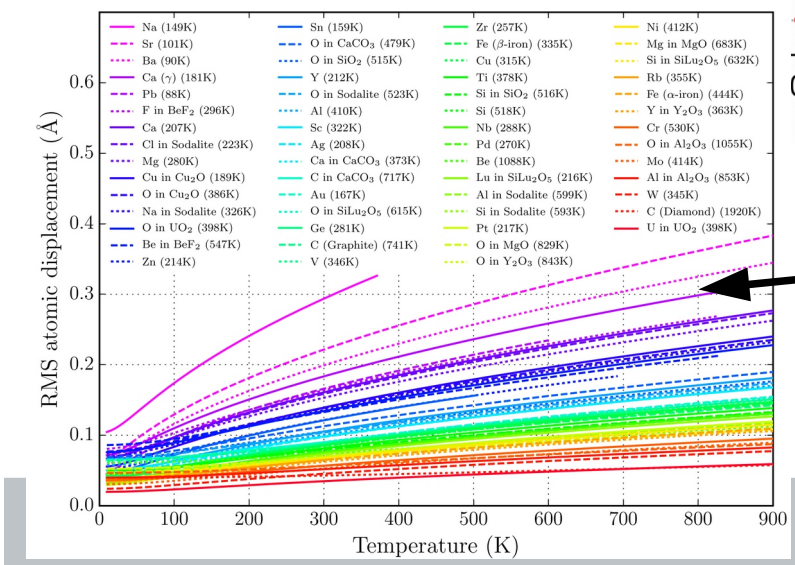
How we handle materials with no phonon DOS specified?

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.

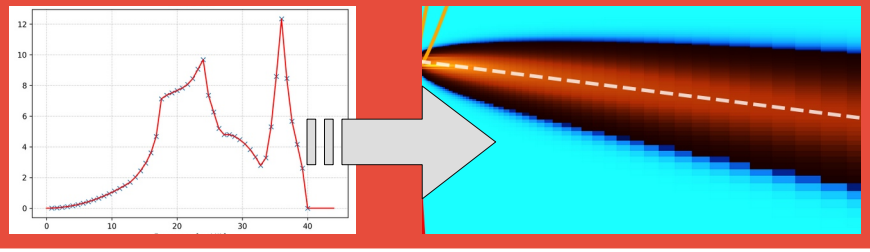




T-dependent atomic displacements (δ), from Debye temperature (T_D)

Of course, a real DOS gives more realistic δ.

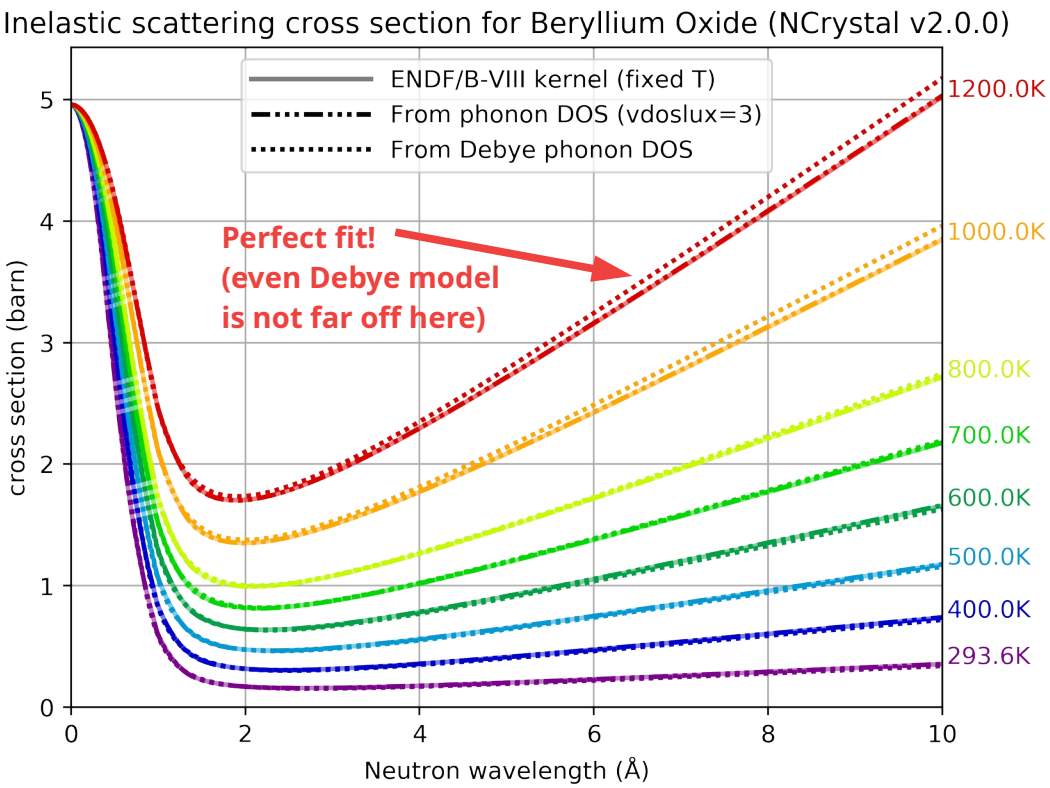
VDOS → S(α,β) [solids]

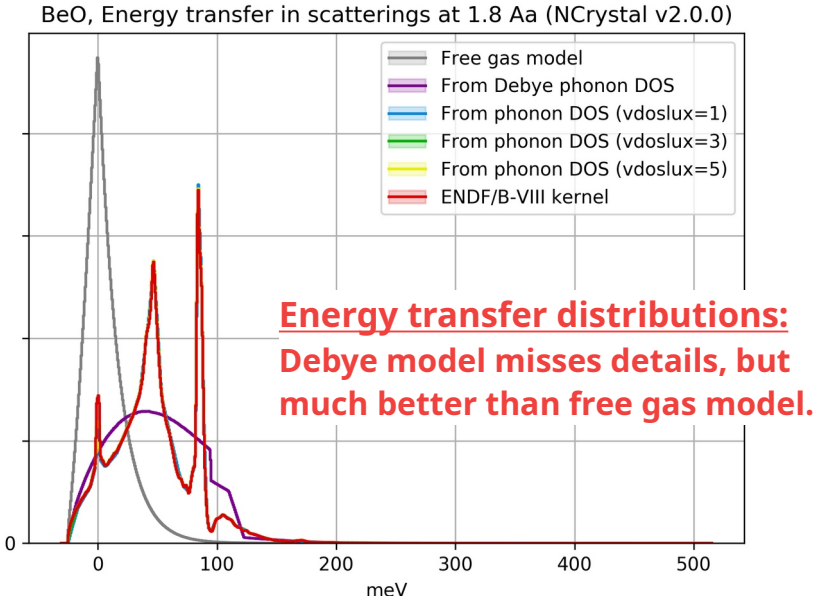


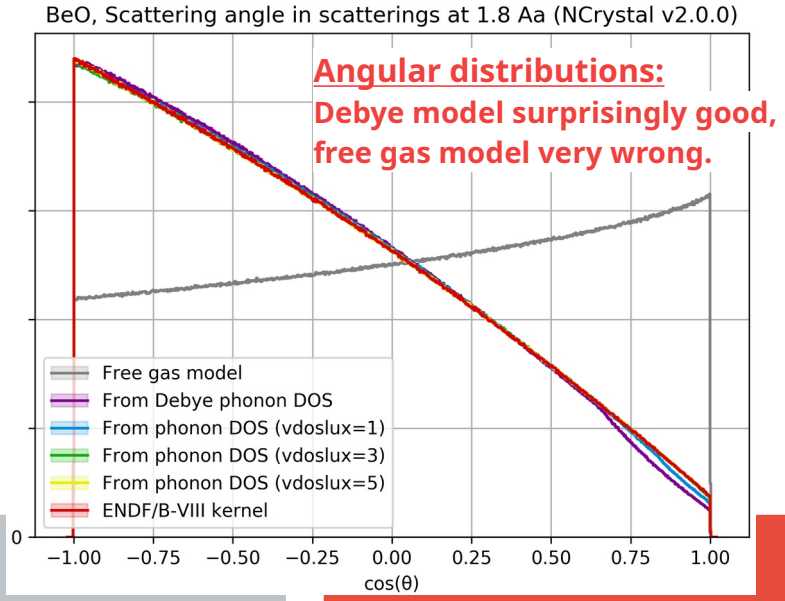
NCrystal
Thermal Neutron Transport

Identical results as when using LEAPR/ENDF kernels:

- Comparing at fixed temperature values from ENDF/B-VIII, using the same VDOS curve.
- NCrystal "luxury" level (cfg param "vdoslux", default value 3) mostly affects scattering kernel grid size+granularity.

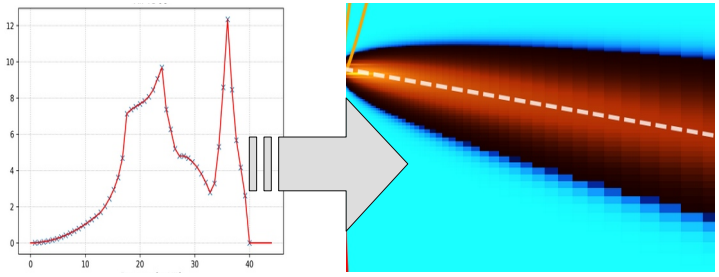






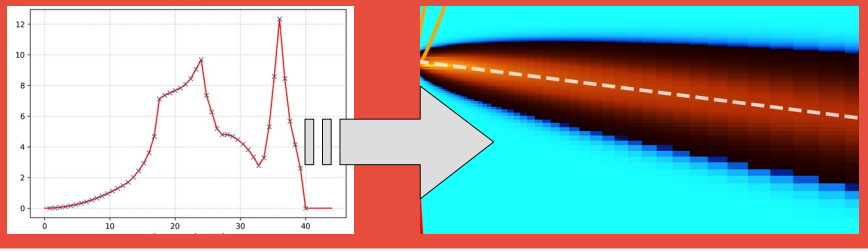
VDOS → Scattering kernel

(the Sjölander method)

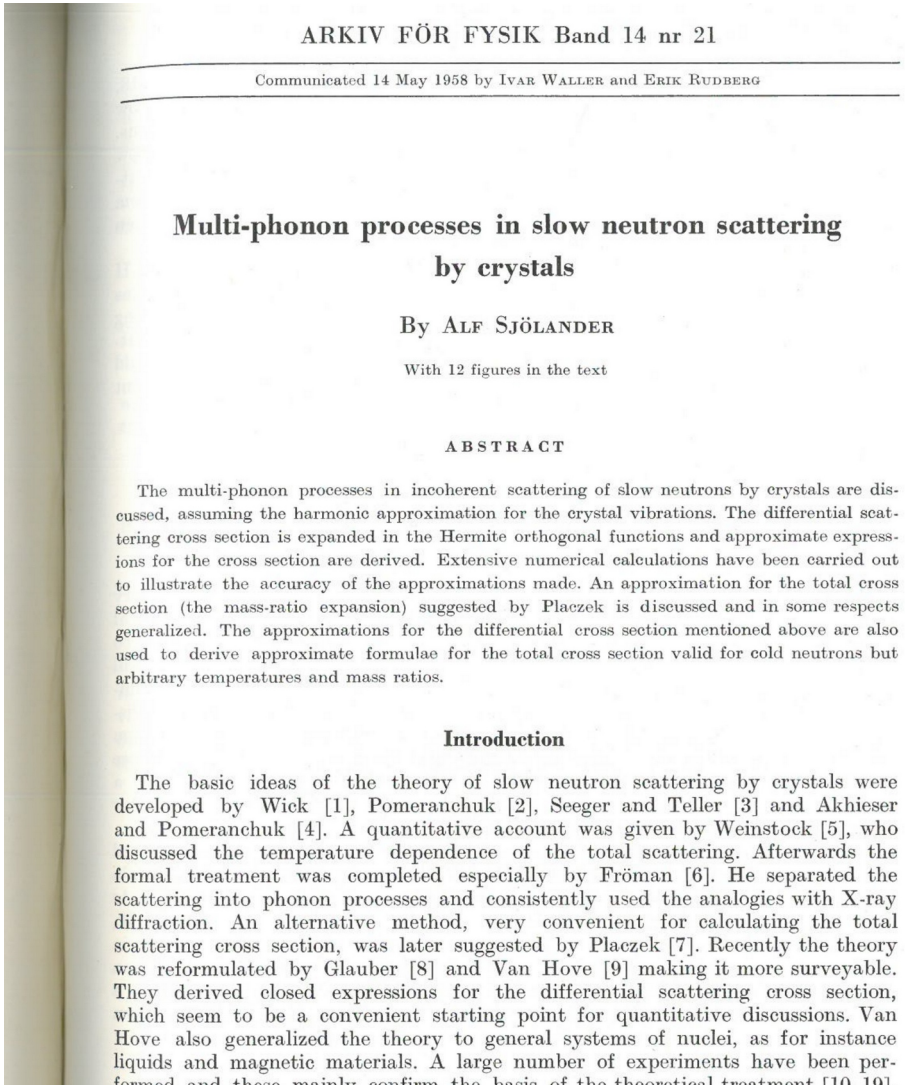


BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

VDOS → S(α,β)



NCrystal
Thermal Neutron Transport



ARKIV FÖR FYSIK Band 14 nr 21

Communicated 14 May 1958 by Ivar Waller and Erik Rudberg

Multi-phonon processes in slow neutron scattering by crystals

By Alf Sjölander

With 12 figures in the text

ABSTRACT

The multi-phonon processes in incoherent scattering of slow neutrons by crystals are discussed, assuming the harmonic approximation for the crystal vibrations. The differential scattering cross section is expanded in the Hermite orthogonal functions and approximate expressions for the cross section are derived. Extensive numerical calculations have been carried out to illustrate the accuracy of the approximations made. An approximation for the total cross section (the mass-ratio expansion) suggested by Placzek is discussed and in some respects generalized. The approximations for the differential cross section mentioned above are also used to derive approximate formulae for the total cross section valid for cold neutrons but arbitrary temperatures and mass ratios.

Introduction

The basic ideas of the theory of slow neutron scattering by crystals were developed by Wick [1], Pomeranchuk [2], Seeger and Teller [3] and Akhieser and Pomeranchuk [4]. A quantitative account was given by Weinstock [5], who discussed the temperature dependence of the total scattering. Afterwards the formal treatment was completed especially by Fröman [6]. He separated the scattering into phonon processes and consistently used the analogies with X-ray diffraction. An alternative method, very convenient for calculating the total scattering cross section, was later suggested by Placzek [7]. Recently the theory was reformulated by Glauber [8] and Van Hove [9] making it more surveyable. They derived closed expressions for the differential scattering cross section, which seem to be a convenient starting point for quantitative discussions. Van Hove also generalized the theory to general systems of nuclei, as for instance liquids and magnetic materials. A large number of experiments have been performed and these mainly confirm the basis of the theoretical treatment [10, 19]

Well-established method!
Used in NJOY/LEAPR. Most ENDF S(α,β) kernels were created this way.

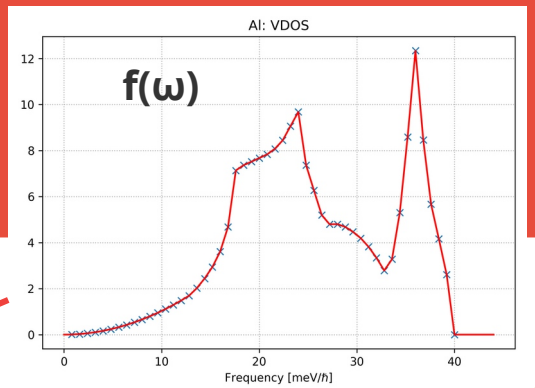
Key feature: Capability is built into NCrystal
And so fast (<0.1s) it can be invoked on-the-fly.

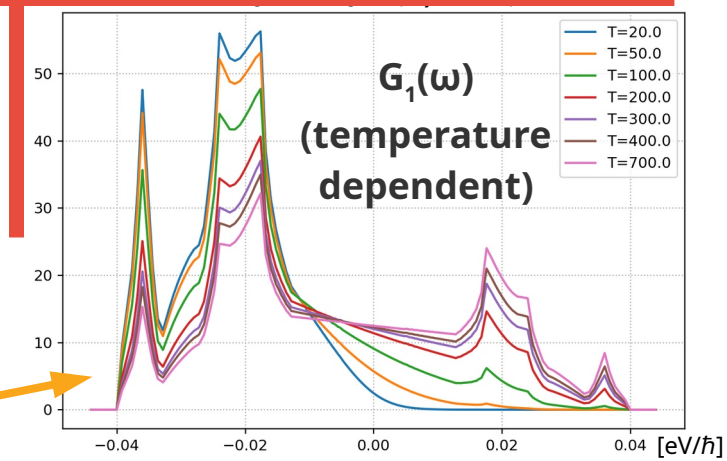
Gives us:

- **Flexibility.** Work directly from VDOS input, avoid usage of non-trivial third-party SW.
- **More materials.** VDOS are much more easily obtained than full kernels.
- **Small data files!**
  Can easily include everywhere.
- **Temperature dependency built in:**
  Unlike static S(α,β) which is only valid for a specific temperature.

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

VDOS→S(α,β) Sjölander's recipe



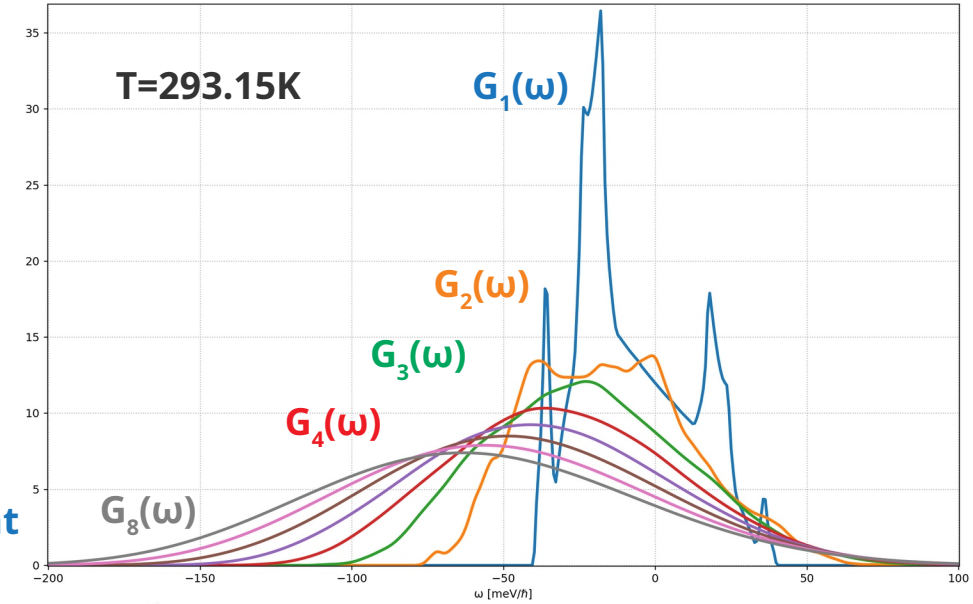


$$G_1(\omega) = g(\omega) = \frac{f(\omega)}{\omega\,\gamma(0)}\,\frac{\coth\left(\frac{\hbar\omega}{2kT}\right)-1}{2},$$

$$G_2(\omega) = \int_{-\infty}^{\infty} g(\omega-\omega')\,G_1(\omega')\,d\omega',$$

. . . . . .

$$G_{n+1}(\omega) = \int_{-\infty}^{\infty} g(\omega-\omega')\,G_n(\omega')\,d\omega'.$$



2W=δ²q², δ=atomic displacement

σ_b

$$\frac{d^2\sigma_{\text{incoh}}^{\text{inel}}}{d\Omega\,d\omega} = A\,\frac{k}{k_0}\,e^{-2W}\sum_{n=1}^{\infty}\frac{(2W)^n}{n!}\,G_n(\omega-\omega_0).$$

$$\sqrt{\frac{E_f}{E}}\,\frac{\sigma_b}{4\pi}\,\frac{S(\alpha,\beta)}{k_bT}$$

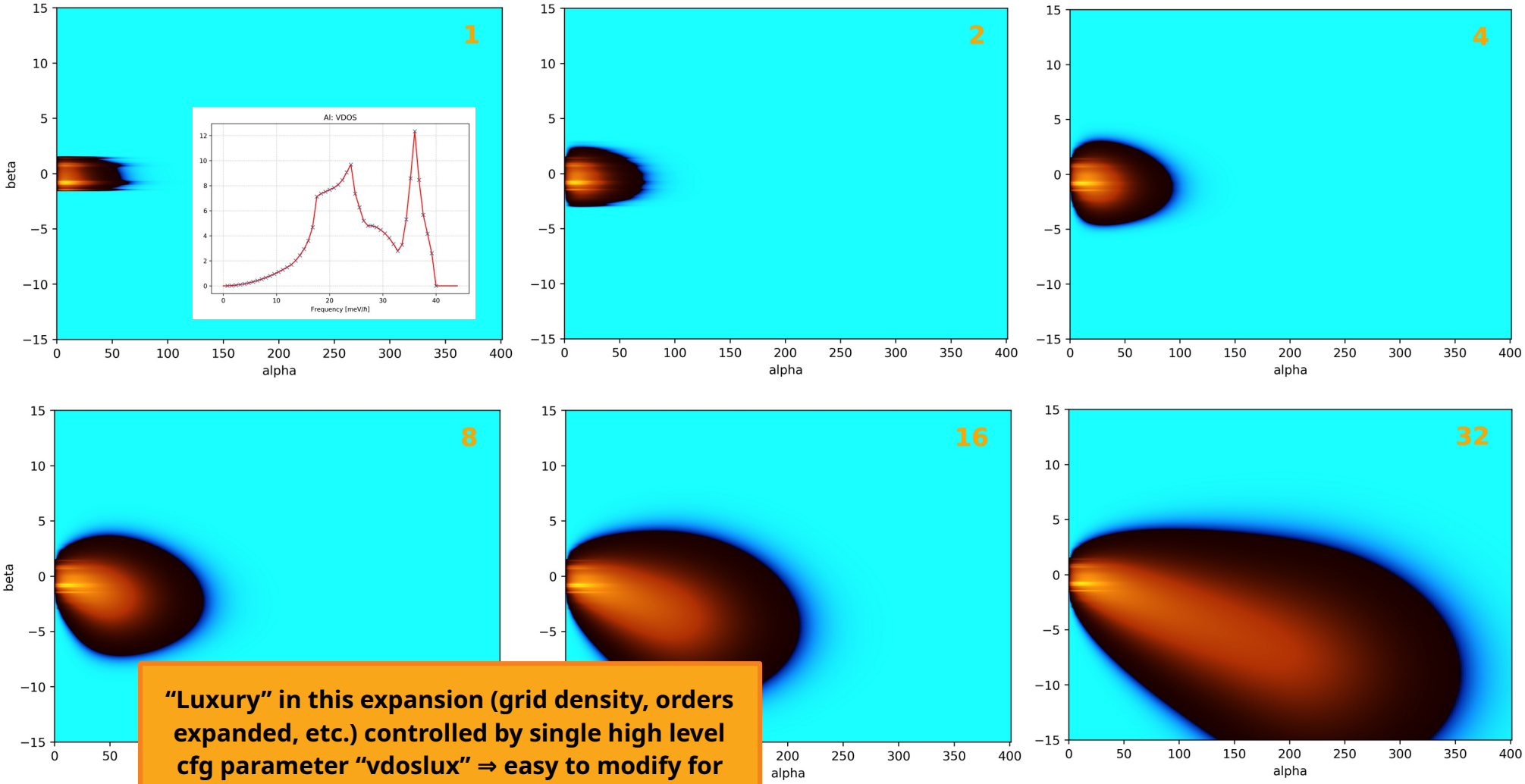
$-\hbar(\omega-\omega_0)/kT=\Delta E/kT=\beta$

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

VDOS→S(α,β): Aluminium

https://github.com/mctools/ncrystal/wiki/VDOSAnimations

NCrystal
Thermal Neutron Transport



"Luxury" in this expansion (grid density, orders expanded, etc.) controlled by single high level cfg parameter "vdoslux" ⇒ easy to modify for any user (see backup slide for details). Default value is of course sensible.

jupyter **"Connecting phonon DOS curves to inelastic scattering" notebook at** https://github.com/mctools/ncrystal-notebooks/

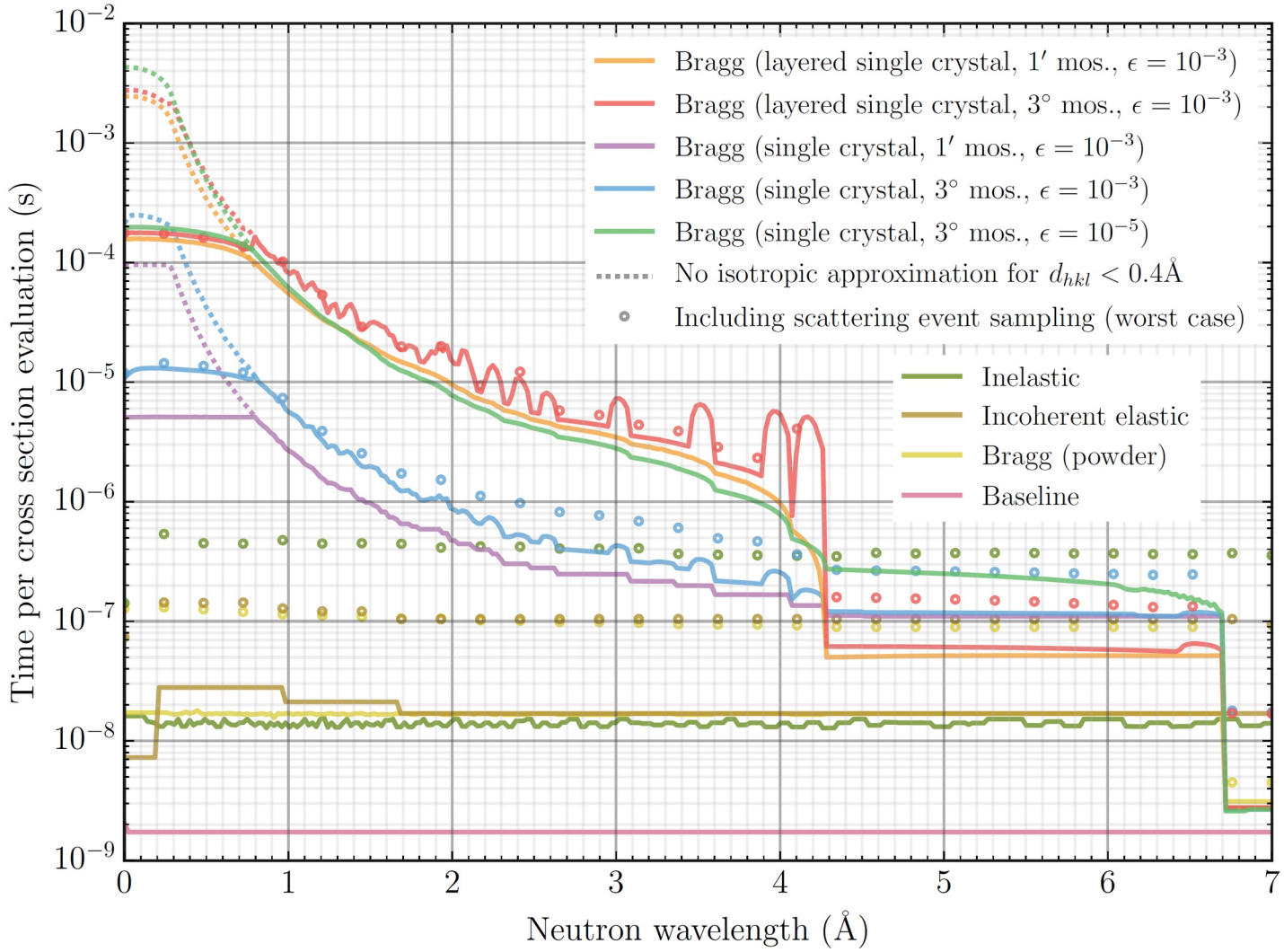
Miscellaneous subjects:

- Computational speed
- Amorphous solids
- Flexible atomic definitions
- Multiphase materials + SANS
- Coherent elastic phonons
- Support for developed plugins

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

Strong focus on computational speed

NCrystal
Thermal Neutron Transport



Rough conclusions for MC simulations with thin samples:

- **O(1-100MHz) neutrons *per thread* in powder**
- **O(0.1-10MHz) neutrons *per thread* in single crystal (depends on λ)**

Material *initialisation* time also very strong focus.

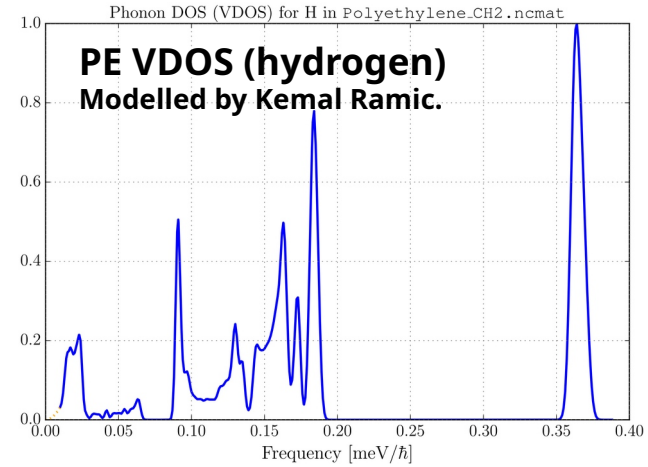
- **V3.9.0 introduced several speedups and the option for multithreaded initialisation.**
- **Most materials initialise in <200ms (many <10ms)**

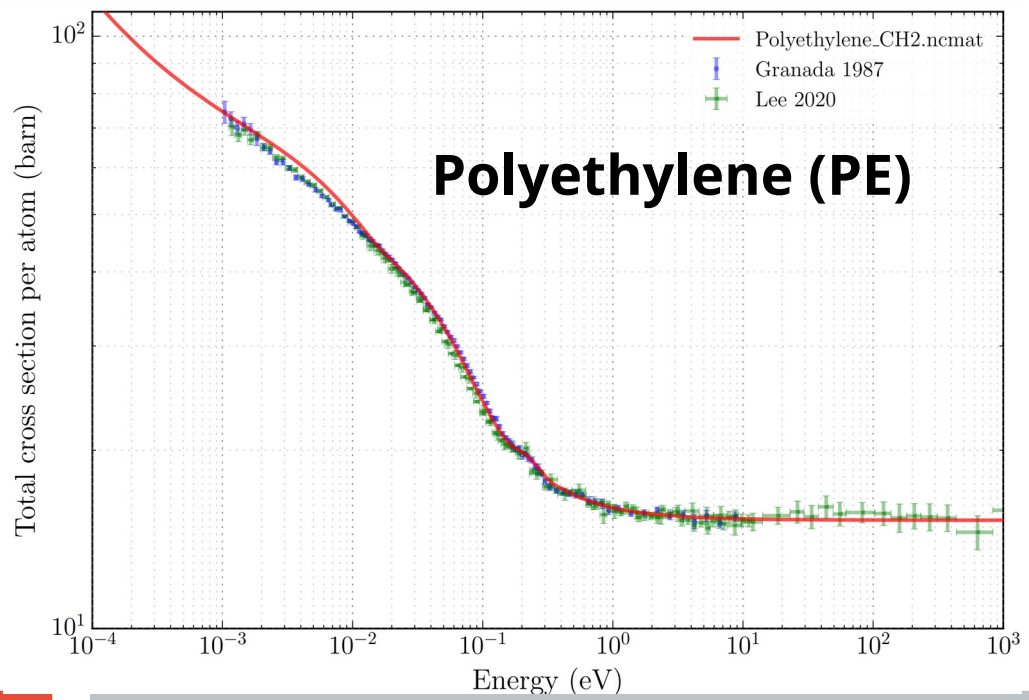
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

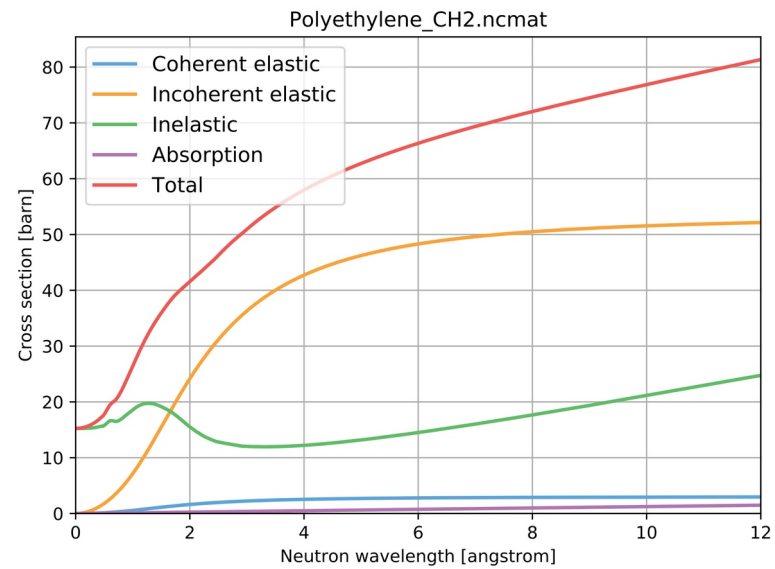
Amorphous solids

NCrystal
Thermal Neutron Transport

- **Uses same inelastic and incoherent-elastic approach as for crystalline solids.**
- **Coherent-elastic scattering via incoherent approximation for now, best for materials with lots of incoherent scattering like H-rich materials (but see next slide).**







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

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

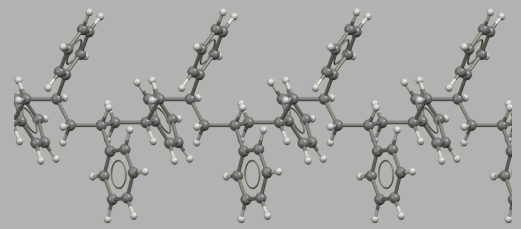
Citation Giovanni Romanelli *et al* 2021 *J. Phys.: Condens. Matter* **33** 285901

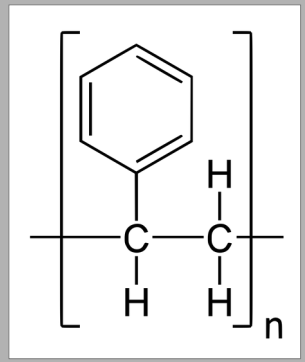
DOI 10.1088/1361-648X/abfc13

References

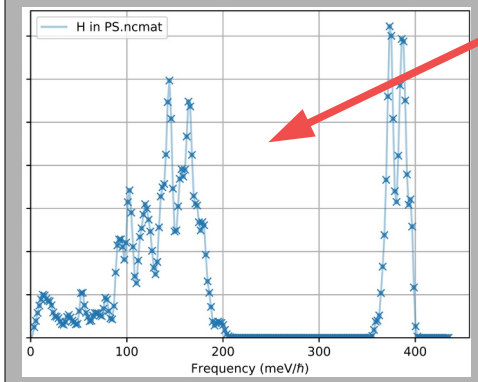
Example (polystyrene):

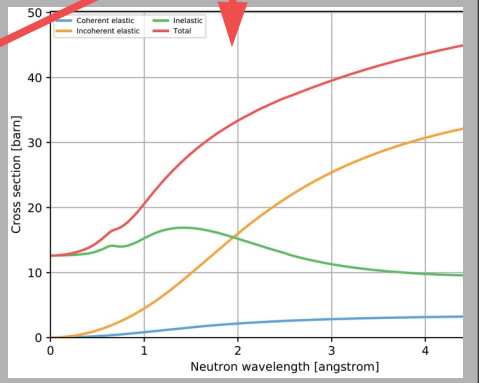
- 1 aromatic ring with 5 H
- 1 CH2 group
- 1 aliphatic CH binding





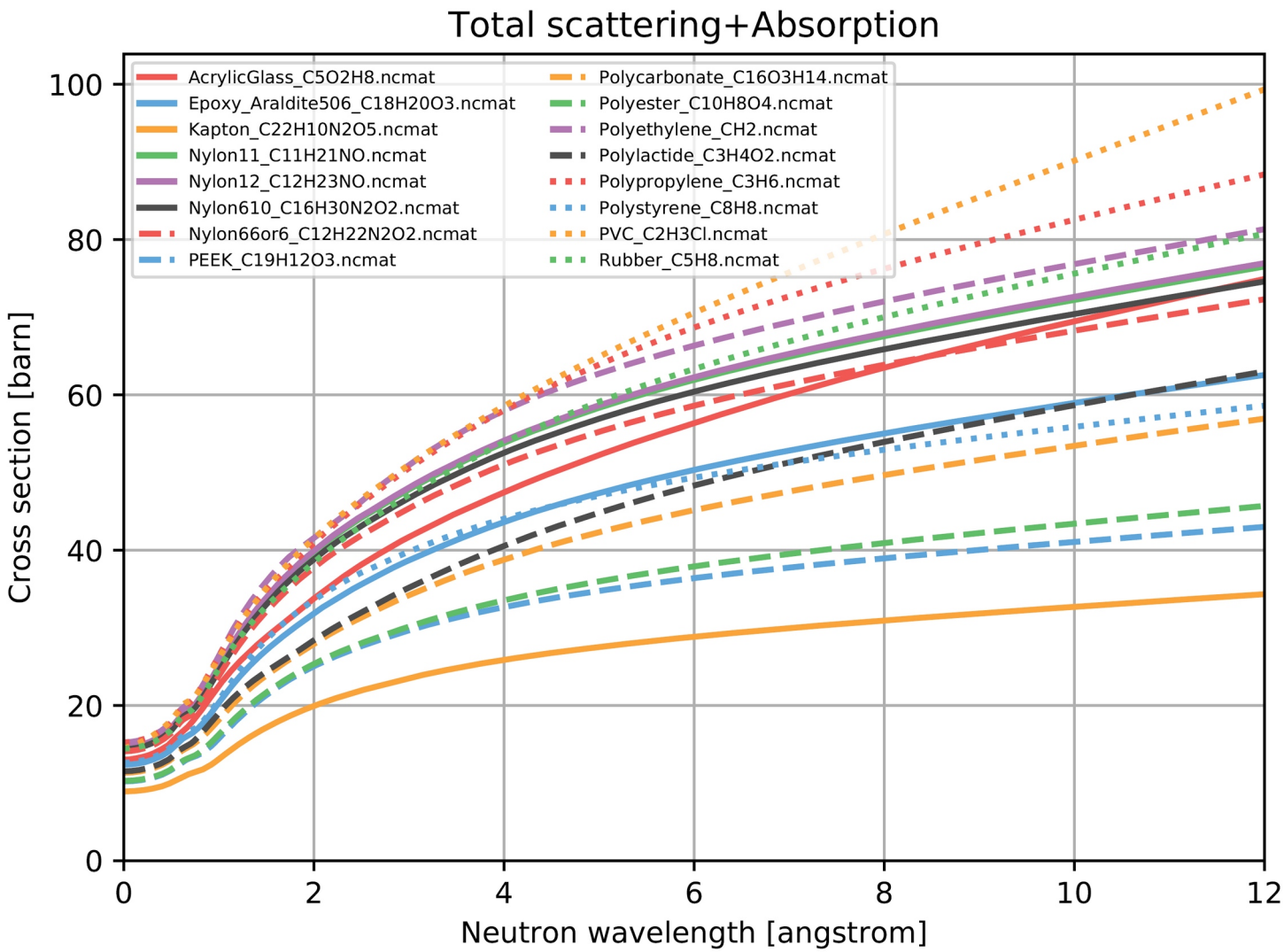

```
$> ncrystal_hfg2ncmat --formula C8H8 \
                      --spec 5xCHaro+1xCHali+1xCH2 \
                      --density 0.99 -o PS.ncmat
```





Amorphous materials in data library

NCrystal
Thermal Neutron Transport



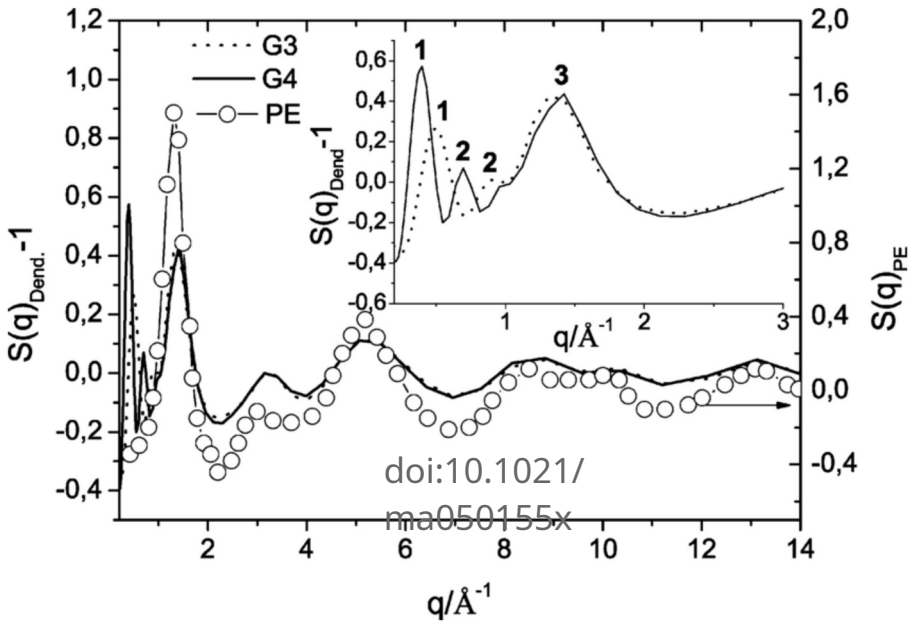
- **Polyethylene and AcrylicGlass (a.k.a. Plexiglass/Lucite) based on VDOS from other sources.**
- **14 others based on ncrystal_hfg2ncmat script (AFGA).**
- **Let us know if we are missing something useful!**

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

Coherent effects in amorphous materials: static structure factors

NCrystal
Thermal Neutron Transport

- We plan to eventually also optionally include static structure factors S(q) in our treatment of amorphous solids, but for now they are modelled under the *incoherent approximation.*
- The *incoherent approximation* is very good for esp. hydrogen-rich materials, but amorphous materials without strong incoherent cross sections might suffer in realism currently.



NCrystal development at ESS was recently chosen to be supported by the APRENDE EURATOM grant. This will hopefully allow us to include these effects for increased realism.

The tricky part is to include them along with the current inelastic kernel, without ending up with double-counting (likely we will use Sköld's method).

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

Flexible atomic (re)definitions

NCrystal
Thermal Neutron Transport

NCrystal supports atoms which are not just natural elements!

- **Ships with database of 80+ natural elements and 261+ isotopes.**
- **Possibility to customise:**
  - **In NCMAT data**
  - **In cfg-string parameter**
  - **With the NCMATComposer**

```
NCMAT v3
@ATOMDB
  #Override data for whatever reason:
  H 1.008u -3.7fm 80.3b 0.3b
  #Provide absent data:
  Rn222 222.017u 123fm 0.456b 789b
  #Enrich Boron:
  B mix 0.95 B10 0.05 B11
  #Add dopants on Al positions:
  Al mix 0.99 Al 0.01 Cr
  #Alternatively use "variable names"
  #(for usage elsewhere in the file):
  X mix 0.2 Al 0.4 Cr 0.4 Th
  #Or simply assign:
  B is B10
```

```
auto sc = NCrystal::createScatter("Al2O3_sg167_Corundum.ncmat;atomdb=Al:mix:0.99:Al:0.01:Cr");
```


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

Multiphase materials

NCrystal
Thermal Neutron Transport

- **Multiphase materials can be defined inside NCMAT data (i.e. with the NCMATComposer), or in a cfg-string:**
  - `“phases<FRAC1*CFGSTR1&...&FRACN*CFGSTRN>;COMMONCFG”`
- **Example (enriched B4C pellets in epoxy):**
  - `“phases<0.01*solid::B4C/2.52gcm3/B_is_0.95_B10_0.05_B11 &0.99*Epoxy_Araldite506_C18H20O3.ncmat>;temp=250K”`
- **NB: Using volume fractions, not mass fractions (for now).**
- **NB: Syntax designed so you can always append e.g. “;temp=250K” to a cfg-string, and have it work.**

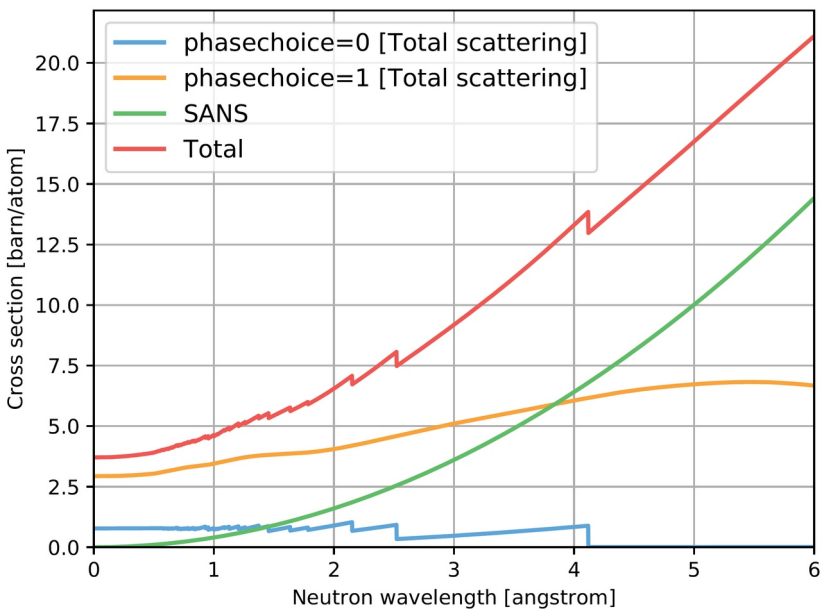
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

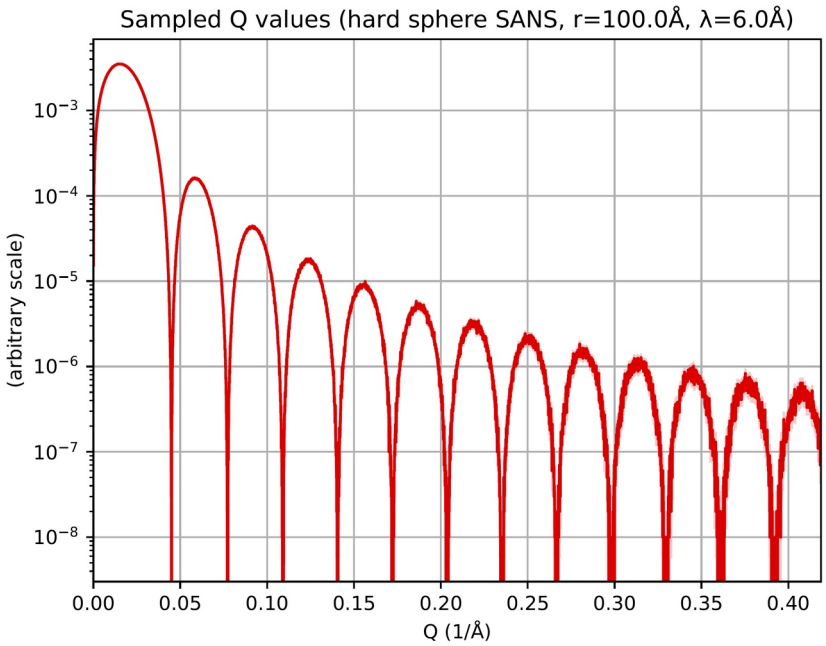
SANS

Small Angle Neutron Scattering

NCrystal
Thermal Neutron Transport

- **Closely connected to multiphase support, NCrystal contains a framework for SANS physics (= phase interference).**
- **For now, only a basic hard-sphere SANS model can be enabled, as proof of concept. Planning to at least add this and general I(Q) support properly to the NCMAT format**





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

Extend NCrystal with 3rd-party plugins

NCrystal
Thermal Neutron Transport

NCrystal includes a plugin mechanism, making it possible add custom physics models

- This can help people with their specific simulation use-case, and (in an ideal world) high quality models can eventually be adopted into the main NCrystal code.
- Extending NCrystal will naturally require C++ knowledge.
- Such plugins can be developed in separate github repos, with standard mechanism for how to include them in a given NCrystal setup.
- More details on: https://github.com/mctools/ncrystal/wiki/Plugins

Nanodiamond plugin:

Rizzi, N., et al. (2023).
Benchmarking of the NCrystal SANS Plugin for Nanodiamonds.
Nuclear Science and Engineering, 198(1), 92–100.
https://doi.org/10.1080/00295639.2023.2196926
https://indico.ess.eu/event/3096/contributions/17717/ (slides)

Magnetic scattering plugin:

Xu, S. , et al. (2024).
Physical model of neutron scattering by clathrate hydrate and C60 hosting paramagnetic oxygen molecules.
Journal of Physics: Condensed Matter, Volume 36, Number 38.
https://doi.org/10.1088/1361-648X/ad5947
https://indico.ess.eu/event/3096/contributions/17717/ (slides)

Other plugins in development:

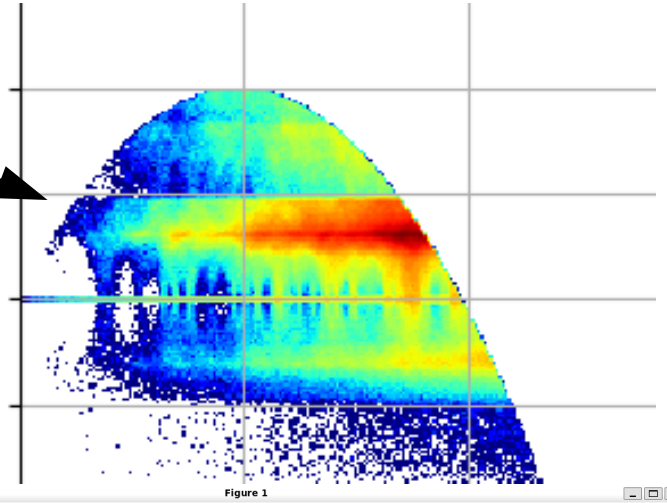
- **March-Dollase Texture plugin by S. Xu et al.**
- **Coherent phonon plugin by X. X. Cai et al.**

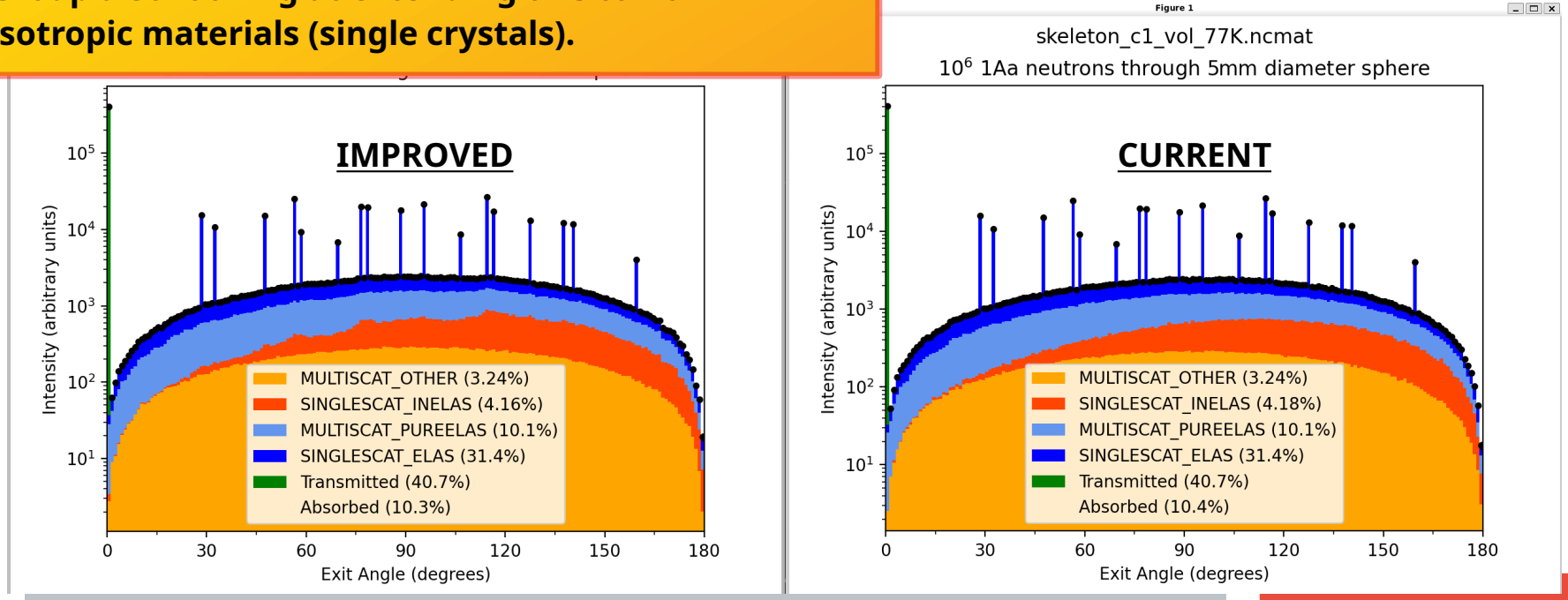
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

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

Second half of the notebook: "Creating materials and the NCMATComposer"

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

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

(of these 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

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

Control DOS→scat. kernel expansion through cfg parameter "vdoslux"

NCrystal
Thermal Neutron Transport

- Controls all aspects of DOS→kernel expansion with only one high level user-accessible parameter.
- Exposing the underlying multitude of parameters to end-users would do no good in practice since no-one would understand how to change them in a self-consistent way.

- **vdoslux=0**: Extremely crude, 100x50 grid, Emax=0.5eV (costs 0.1MB mem, 0.02s init time)
- **vdoslux=1**: Crude, 200x100 grid, Emax=1eV (costs 0.5MB mem, 0.04s init time)
- **vdoslux=2**: Decent, 400x200 grid, Emax=3eV (costs 2MB mem, 0.08s init time)
- **vdoslux=3** : Good, 800x400 grid, Emax=5eV (costs 8MB mem, 0.2s init time)
- **vdoslux=4**: Very good, 1600x800 grid, Emax=8eV (costs 30MB mem, 0.8s init time)
- **vdoslux=5**: Extremely good, 3200x1600 grid, Emax=12eV (costs 125MB mem, 5s init time)

The default if using Debye model instead of actual input data (vdoslux gets reduced by 3 for these mats.)

Typical level in ENDF kernels

The default!

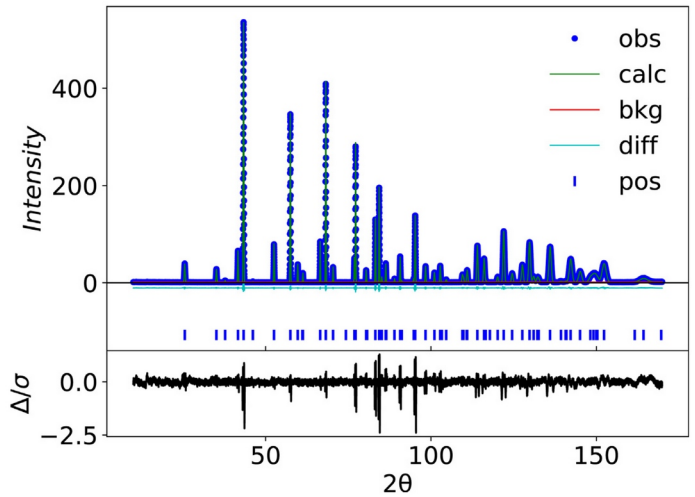
Overkill, exists for validation purpose

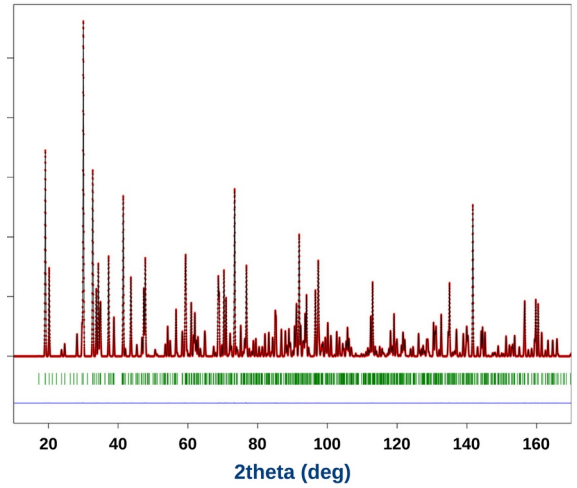
Users adviced to leave at default (3), or change with ±1 to 2 or 4.

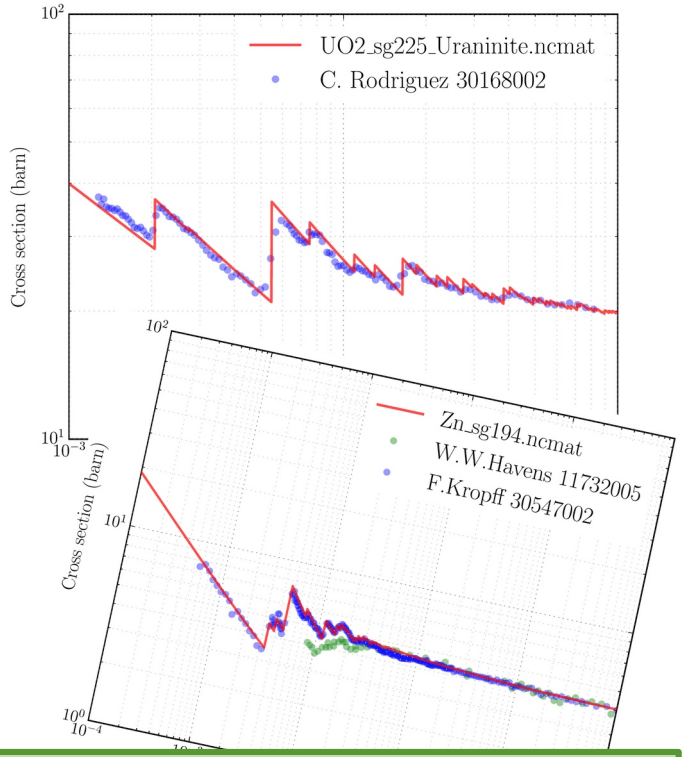
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

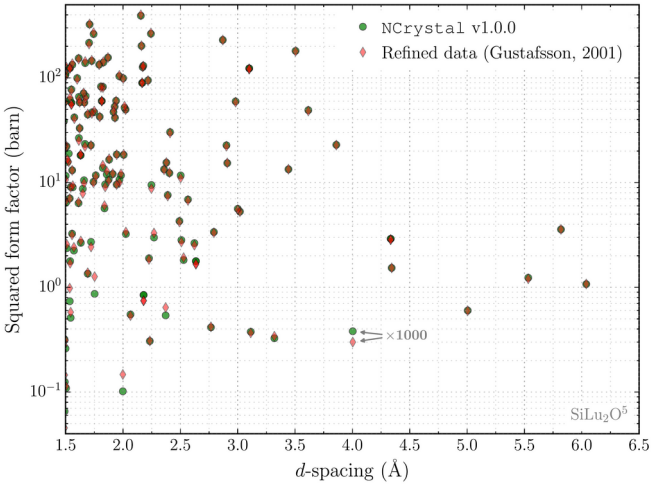
HKL structure factors initialised on-the-fly (validated thoroughly)

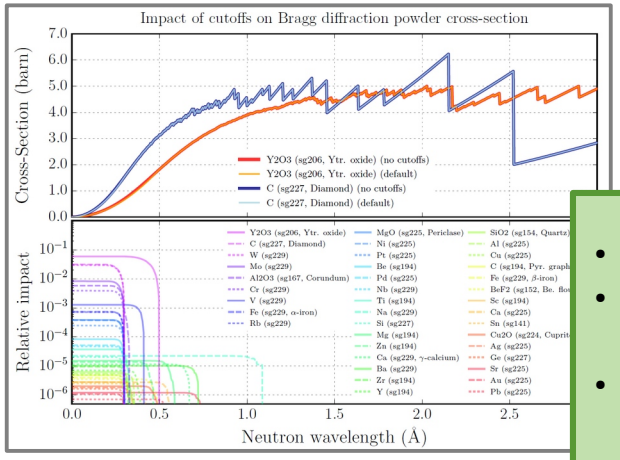
NCrystal
Thermal Neutron Transport











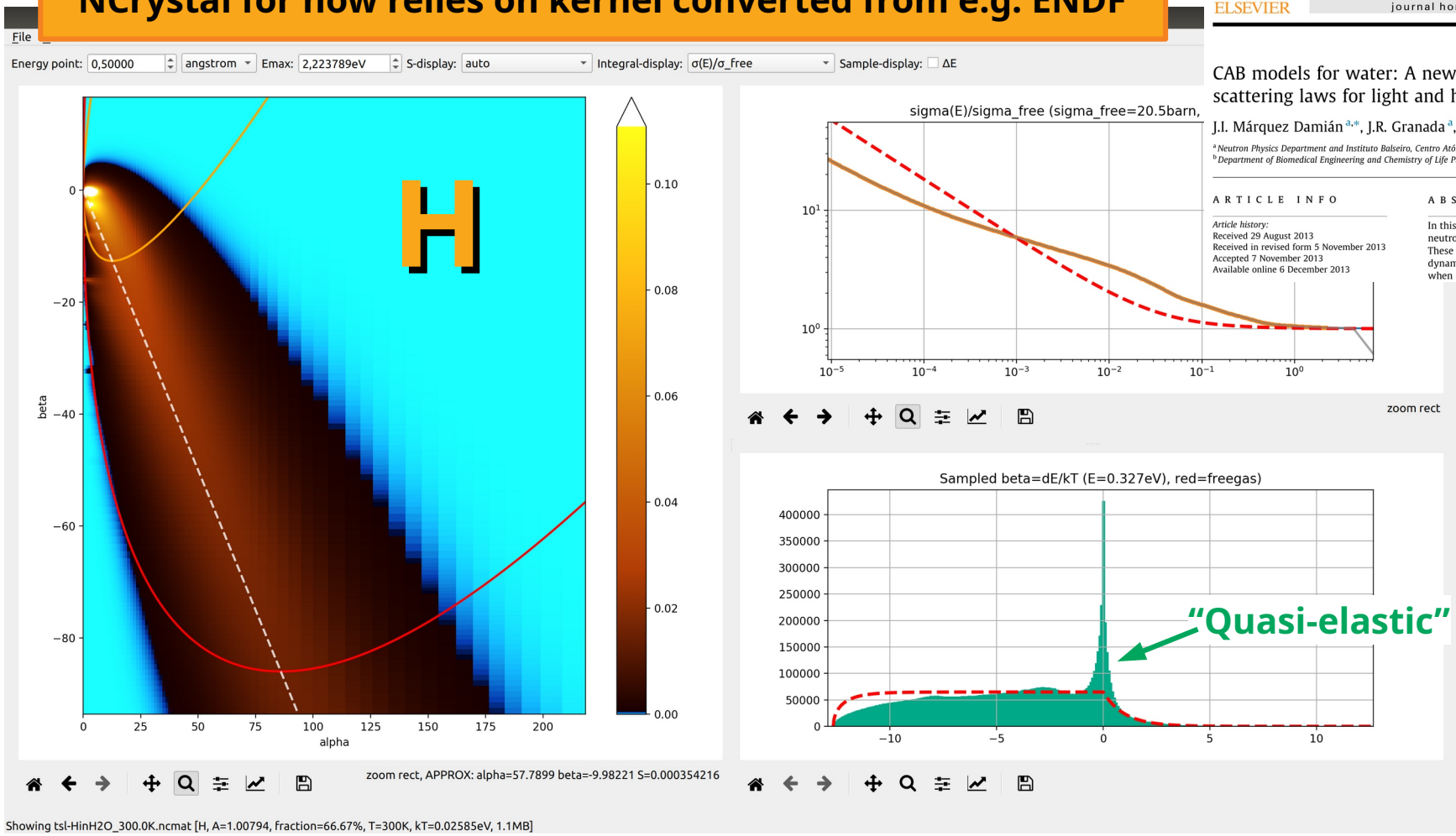
- Fast (few ms) init. of **all** relevant planes.
- Comparisons to measured structure factors and total cross sections.
- NCrystal+McStas simulated scattering patterns analysed with GSAS-II/Fullprof (recovering input crystal parameters).
- Comparison with NXS library predictions.

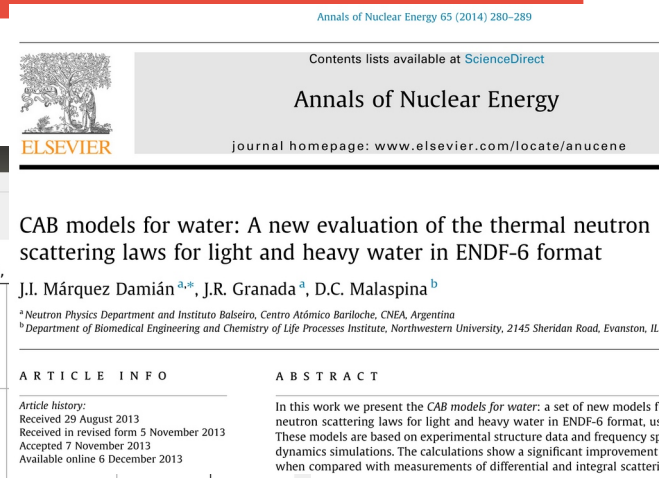
BrightnESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 676548
HighNESS is funded by the European Union Framework Program for Research and Innovation Horizon 2020, under grant agreement 951782

Support for liquids rely on externally provided kernels, here water (converted from ENDF8)

NCrystal
Thermal Neutron Transport

VDOS→S(α,β) does not work directly for liquids!
NCrystal for now relies on kernel converted from e.g. ENDF





Oxygen is modelled as free gas

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

Single crystal with Gaussian mosaicity uses cheap pre-check to speed up

NCrystal
Thermal Neutron Transport

```
double NC::GaussMos::calcCrossSections( InteractionPars& ip,
                                        const NC::Vector& indir,
                                        const std::vector<NC::Vector>& deminormals,
                                        std::vector<NC::GaussMos::ScatCache>& cache,
                                        VectD& xs_commul ) const
{
  nc_assert(ip.isValid()&&ip.m_wl>0);
  nc_assert(indir.isUnitVector());
  std::vector<Vector>::const_iterator it(deminormals.begin()), itE(deminormals.end());
  double xsoffset = xs_commul.empty() ? 0.0 : xs_commul.back();
  double xssum(0.0);
  const double cptsq = ip.m_cos_perfect_theta_sq;
  const double cta = m_gos.getCosTruncangle();
  for(;it!=itE;++it) {
    const Vector& normal = *it;
    const double dot = normal.dot(indir);
    double sdotcptsq = (1.0 - dot * dot)*cptsq;
    double ds = dot * ip.m_sin_perfect_theta;

    //First a combined check, which usually allows us to skip both normal and
    //anti-normal:
    double A0 = ncmax( 0.0, cta - ncabs(ds) );
    if ( sdotcptsq <= A0*A0 )
      continue;

    //At least one of the two normals should contribute, so deal with them:
    double Am = ncmax( 0.0, cta - ds );
    if ( sdotcptsq > Am*Am ) {
      //anti-normal is within truncated Gauss
      double xs = calcRawCrossSectionValue(ip, dot );
      if (xs) {
        xs_commul.push_back(xsoffset + (xssum += xs));
        cache.emplace_back(-normal, ip.m_inv2dsp);
      }
    }
    double Ap = ncmax( 0.0, cta + ds );
    if ( sdotcptsq > Ap*Ap ) {
      //normal is within truncated Gauss
      double xs = calcRawCrossSectionValue(ip, -dot );
      if (xs) {
        xs_commul.push_back(xsoffset + (xssum += xs));
        cache.emplace_back(normal, ip.m_inv2dsp);
      }
    }

  }
  return xssum;
}
```

This function gives xsect for all planes with a given d-spacing (only called if d-spacing < λ/2)

Must check all normals in group.

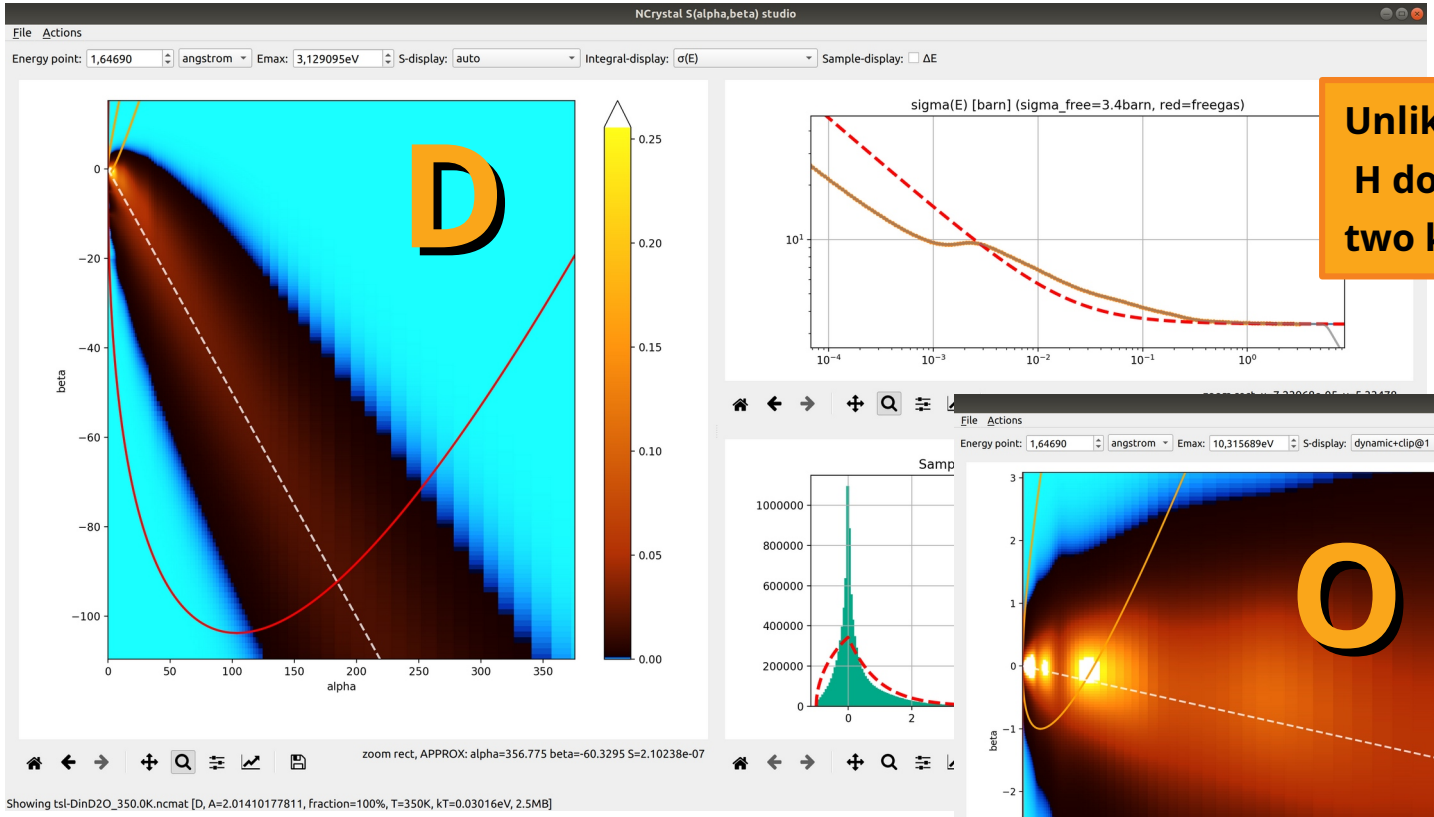
Precheck allows most planes to be skipped since they do not contribute.

Precise evaluation of the non-zero contribution is still where most time is spent, even when only very few planes survives the cheap pre-check!

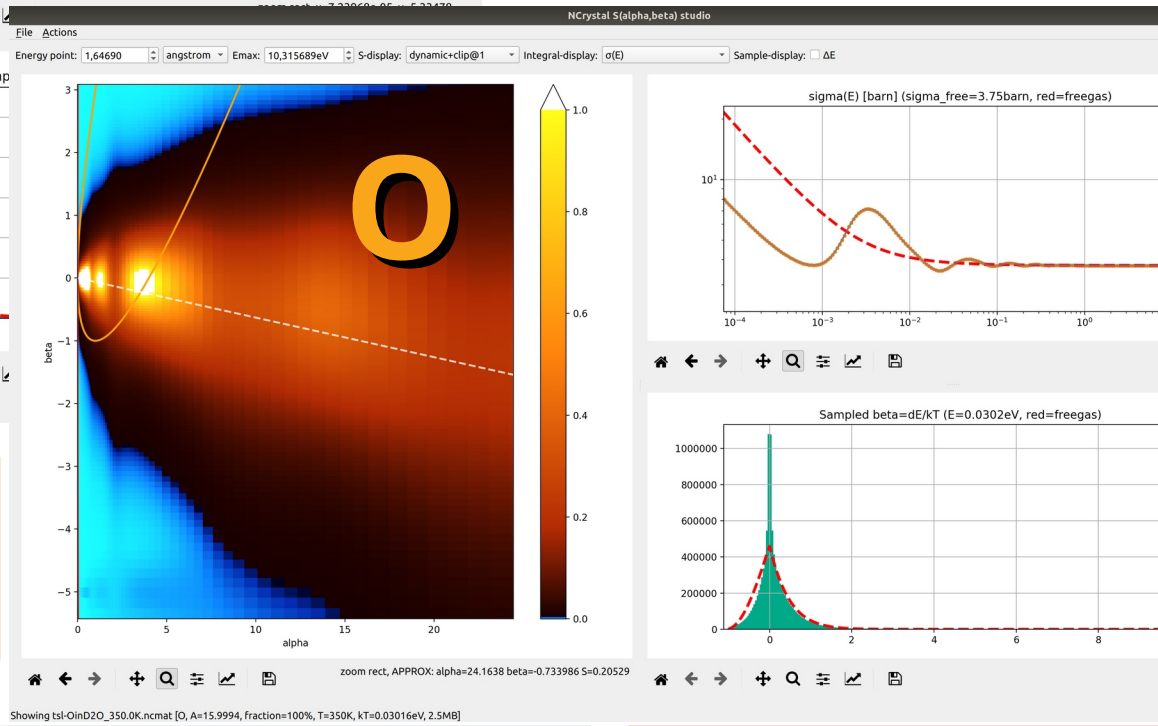
Time to sample interactions is much less crucial, since it mostly deals with only a single plane.

… and heavy water (also converted from ENDF8)

NCrystal
Thermal Neutron Transport



Unlike liquid H_20 (where scattering on H dominates completely), D_2O needs two kernels



NCrystal of course supports non-natural "enrichments" like D = ^{2}H

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