



NCrystal: A library for thermal neutron transport

Thomas Kittelmann (ESS)
(~40min, this presentation)

Peter Willendrup (DTU/ESS)
(~20min McStas-NCrystal demo)



brightness



HighNess

Incomplete (sorry!) list of people who provided useful input, testing, bug reports or other support for NCrystal (alphabetic order):

M. Bertelsen, J. I. Márquez Damián, D. Di Julio, E. Dian, R. Hall-Wilton, K. Kanaki, M. Klausz, E. Klinkby, E.B.Knudsen, A. Morozov, K. Ramic, N. Rizzi, V. Santoro, P. Willendrup, ...

- **Introduction to the NCrytal framework**
- **Physics models**
 - Bragg diffraction (powder)
 - Bragg diffraction (single crystals, isotropic or pyrolytic graphite)
 - Inelastic physics (scattering kernels, phonon DOS)
 - Incoherent elastic physics
- **Recent / planned developments**
 - New materials, new physics, ...

The NCrystal framework

The NCrystal project: History

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

NCrystal

X. X. Cai & T. Kittelmann

Original motivation:

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

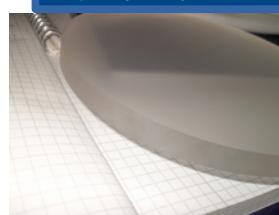


Detector frames, vessels, supports
(polycrystalline metals)

Crystalline samples



Filters
(single- or
poly-crystals)



Monochromators,
analysers
(single crystals.
layered crystals)

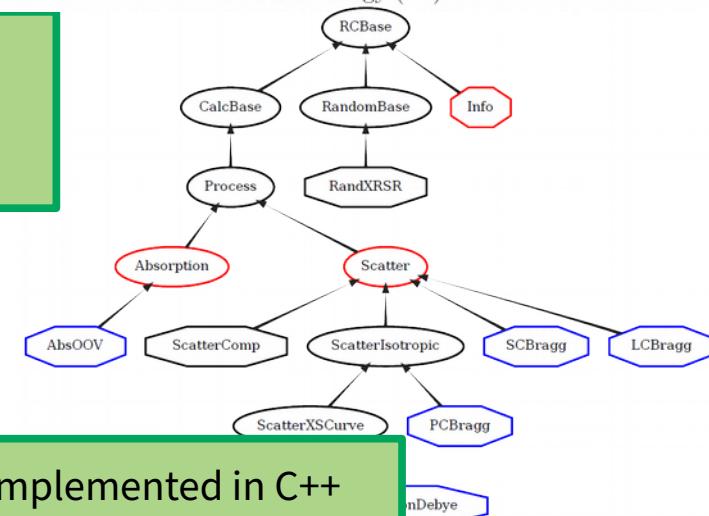
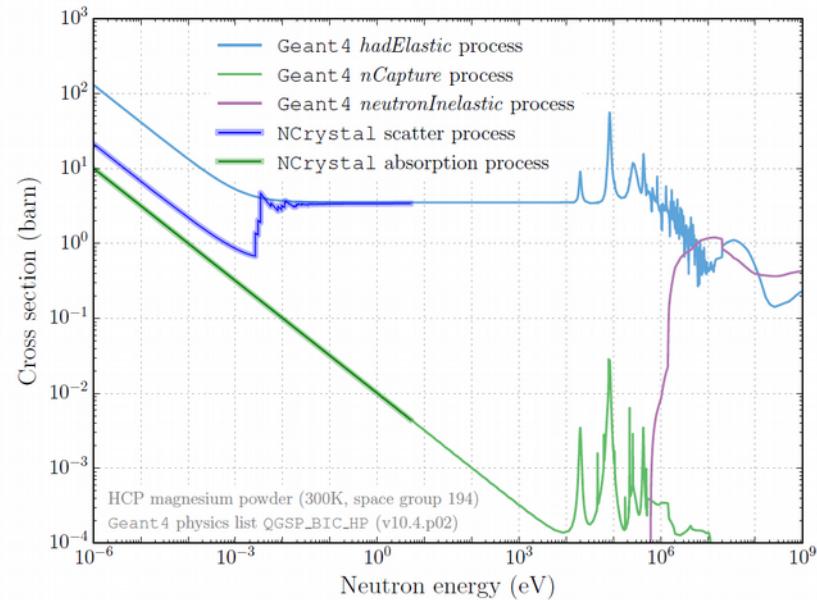
Advanced earlier efforts in older “NXSG4” plugin

- 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 expanded beyond Geant4

- We can reach a wider and more relevant community
- We also have use-cases outside Geant4 ourselves
- As a service to the community in general

Physics scope has expanded as well, as
will be discussed in this presentation.



Implemented in C++

- But users not forced to deal with C++ code

Many available interfaces

Can share configurations and data files across them all

NCrystal

X. X. Cai & T. Kittelmann

NCrystal



Direct/standalone usage

- Command-line tools
- Programmatic access from C++/C/Python



NCrystal aims to be hassle-free:

- Robust models, not “expert-only”.
- Fast models.
- No required dependencies.
- Linux, OSX, Windows
- C++11, C89, Python3.5 or later versions.
- Open source (github, Apache 2.0 license)
- Multi-thread safe.
- RNG streams, error reporting etc. can be controlled by calling code.

NJOY

- Can use NCrystal results to generate ENDF data files. (thanks to J. I. Márquez Damián and Kemal Ramic)

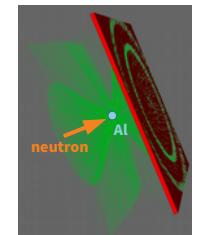


- Available out-of-the box in McStas (thanks to P. Willendrup)
- But can optionally install and use newer NCrystal release.
- Supports multiple scattering + a few basic sample shapes
- Also supported in McStas Unions (thanks to M. Bertelsen)

ANTS2

- Available out-of-the-box (thanks to A. Morozov)

GEANT4
A SIMULATION TOOLKIT



- Plugin ships with NCrystal.
- ESS Detector Group's Geant4 framework has NCrystal built in.
- Goal is availability out-of-the-box for all.

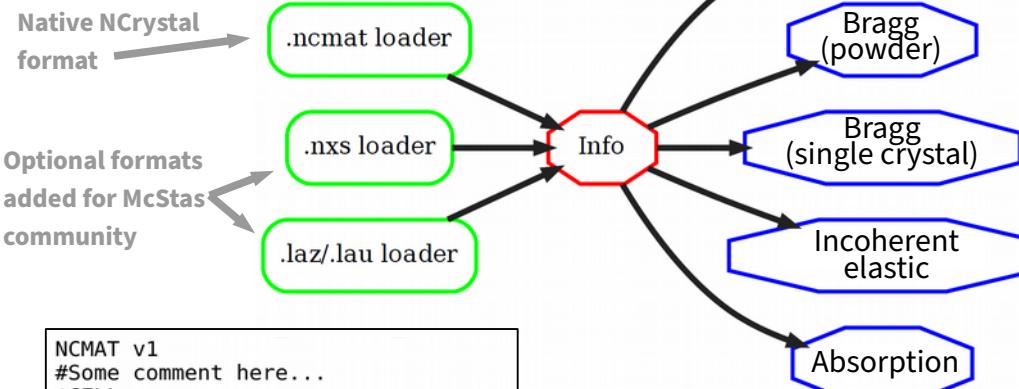


Source of material definitions

NCrystal

X. X. Cai & T. Kittelmann

- Supports multiple input formats
- Can load from on-disk file or in-memory buffers



```

NCMAT v1
#Some comment here...
@CELL
lengths 5.65735 5.65735 5.65735
angles 90. 90. 90.
@SPACEGROUP
227
@ATOMPOSITIONS
Ge 0.75 0.75 0.25
Ge 0.5 0.5 0.
Ge 0.75 0.25 0.75
Ge 0.5 0. 0.5
Ge 0.25 0.75 0.75
Ge 0. 0.5 0.5
Ge 0.25 0.25 0.25
Ge 0. 0. 0.
@DEBYETEMPERATURE
Ge 281.4
  
```

Ge_sg227.ncmat

.ncmat
loader

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 per unit cell (total 8):
8 Ge atoms [T_Dehby=281.437K, MSD=0.00760282Aa]

Atomic coordinates:

	0	0	0
Ge	0	0.5	0.5
Ge	0.25	0.25	0.25
Ge	0.25	0.75	0.75
Ge	0.5	0	0.5
Ge	0.5	0.5	0
Ge	0.75	0.25	0.75
Ge	0.75	0.75	0.25

Density : 5.32937 g/cm3

Temperature : 293.15 kelvin

Neutron cross-sections:
Absorption at 2200m/s : 2.2 barn
Free scattering : 8.36483 barn

HKL planes (d_lower = 0.15 Aa, d_upper = inf Aa):

H	K	L	d_hkl[Aa]	Multiplicity	Fsquared[barn]	Expanded-HKL-list
1	-1	-1	3.26627	8	20.8434	1,-1,-1 -1,1,1 1,
0	2	-2	2.00018	12	39.7773	0,2,-2 0,-2,2 0,2
1	-3	-1	1.70576	24	19.3369	1,-3,-1 -1,3,1 1,
0	0	4	1.41434	6	36.9022	0,0,4 0,0,-4 0,4,
1	-3	-3	1.29789	24	17.9392	1,-3,-3 -1,3,3 1,
2	-4	-2	1.1548	24	34.235	2,-4,-2 -2,4,2 2,
1	-5	-1	1.08876	32	16.6426	1,-5,-1 -1,5,1 1,
0	4	-4	1.00009	12	31.7605	0,4,-4 0,-4,4 0,4,
1	-5	-3	0.956267	48	15.4397	1,-5,-3 -1,5,3 1,
0	2	-6	0.894506	24	29.4649	0,2,-6 0,-2,6 0,2,
3	-5	-3	0.862738	24	14.3238	3,-5,-3 -3,5,3 3,
4	-4	-4	0.816568	8	27.3353	4,-4,-4 -4,4,4 4,
1	-7	-1	0.792187	48	13.2885	1,-7,-1 -1,7,1 1,
2	-6	-4	0.755995	48	25.3595	2,-6,-4 -2,6,4 2,

- Native NCMAT format:
<https://github.com/mctools/ncrystal/wiki/NCMAT-format>
- Well defined and versioned

NCrystal data library

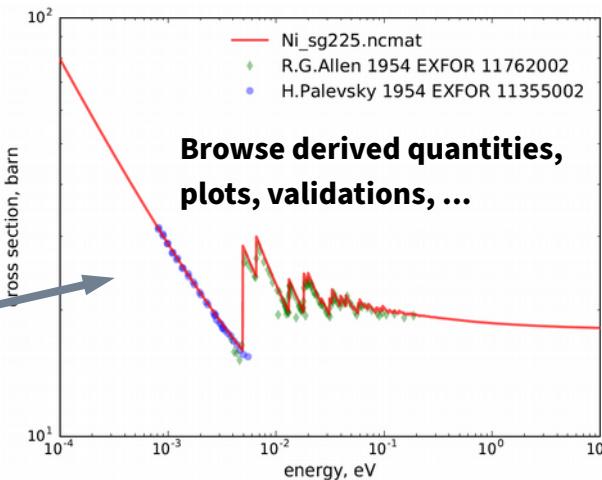
<https://mctools.github.io/ncrystal/> → wiki → Data-library

NCrystal

X. X. Cai & T. Kittelmann

Ge_sg227	F d -3 p=5.31 g/cm ³ ...
Mg_sg194	P 63/m p=1.7 g/cm ³ ...
Mo_sg229	I m -3 m (229) p=10.22 g/cm ³ ...
Na_sg229	I m -3 m (229) p=0.96663 g/cm ³ ...
Nb_sg229	I m -3 m (229) p=8.5827 g/cm ³ ...
Ni_sg225	F m -3 m (225) p=8.9092 g/cm ³ ...
Pb_sg225	F m -3 m (225) p=11.344 g/cm ³ ...
Pd_sg225	F m -3 m (225) p=12.01 g/cm ³ ...
	F m -3 m (225)

NCrystal comes with library of validated data files,
for describing a large number of crystal
structures relevant to neutron scattering.



```
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 0.5 0.
Al 0.5 0. 0.5
@DEBYETEMPERATURE
Al 410.35
```

Al_sg225.ncmat

NB: Debye temperature not
needed since NCrystal v2.6.0.
Debye-Waller factors are now
estimated from phonon DOS
curves if available.

To use in Geant4 or McStas (or ...), simply supply
name of material file, along with relevant parameters
like temperature, orientation of single crystal, etc.

Example usage (from standalone Python)



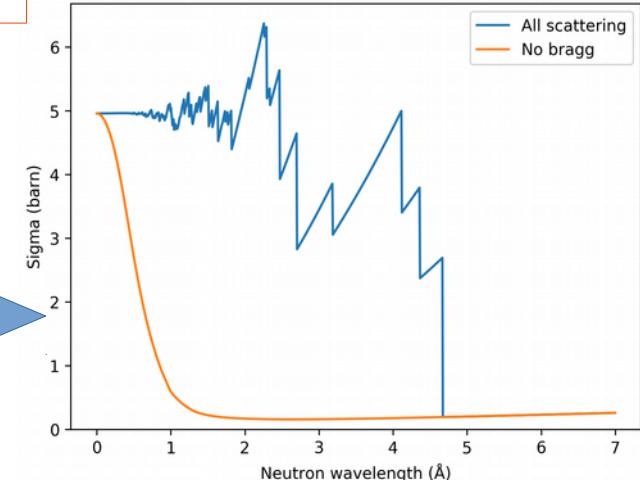
NCrystal

X. X. Cai & T. Kittelmann

```
#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;coh_elas=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 (\u00c5)')  
plt.ylabel('Sigma (barn)')  
plt.legend()  
plt.show()
```

```
#Can also extract more detailed info:  
info_BeO = NC.createInfo("BeO_sg186.ncmat")  
info_BeO.dump() Prints all info to terminal  
print ('Density [g/cm3]: ',info_BeO.getDensity())  
for fraction,atom in info_BeO.composition:  
    print(f'Has {fraction*100}% {atom}')  
for h,k,l,mult,dspacing,fsq in info_BeO.hklList():  
    if 1.0 < dspacing < 1.2:  
        print(f'd={dspacing:g}Aa, strength={mult*fsq:g}')
```

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



Additionally NCrystal can also perform Monte-Carlo sampling of scattering events

```
Density [g/cm3]: 3.022865728266587  
Has 50.0% Be=Be(cohSL=7.79fm cohXS=7.62579barn incXS=0.0018barn absXS=0.0076barn mass=9.01218amu Z=4)  
Has 50.0% O=O(cohSL=5.805fm cohXS=4.23462barn incXS=0barn absXS=0.00019barn mass=15.9994amu Z=8)  
d=1.16827Aa, strength=9.96888  
d=1.14706Aa, strength=41.9296  
d=1.12844Aa, strength=9.252  
d=1.08975Aa, strength=0.259484  
d=1.02967Aa, strength=10.2016
```

More info about NCrystal framework

Code, issue tracker, small wiki, data library page at GitHub:

<https://github.com/mctools/ncrystal>

NCrystal

Cai & T. Kittelmann

ARTICLE IN PRESS
Computer Physics Communications
Contents lists available at ScienceDirect
[Computer Physics Communications](http://www.elsevier.com/locate/cpc)
journal homepage: www.elsevier.com/locate/cpc



Detailed paper:

DOI [10.1016/j.cpc.2019.07.015](https://doi.org/10.1016/j.cpc.2019.07.015)

¹ Paper and its associated software (T. Kittelmann).
² Corresponding author.
E-mail address: thomas.kittelmann@esrf.fr (T. Kittelmann).

<https://doi.org/10.1016/j.cpc.2019.07.015>
© 2019 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

Second publication in the pipeline, with details of all elastic physics models (submitted).

<https://arxiv.org/abs/2012.04294>

Physics models

Bragg diffraction in powders (and texture-free polycrystals)

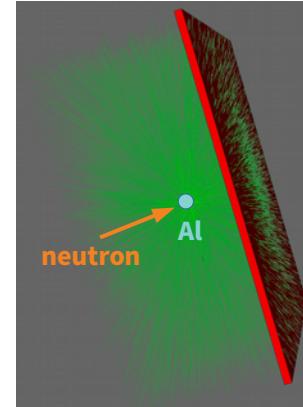
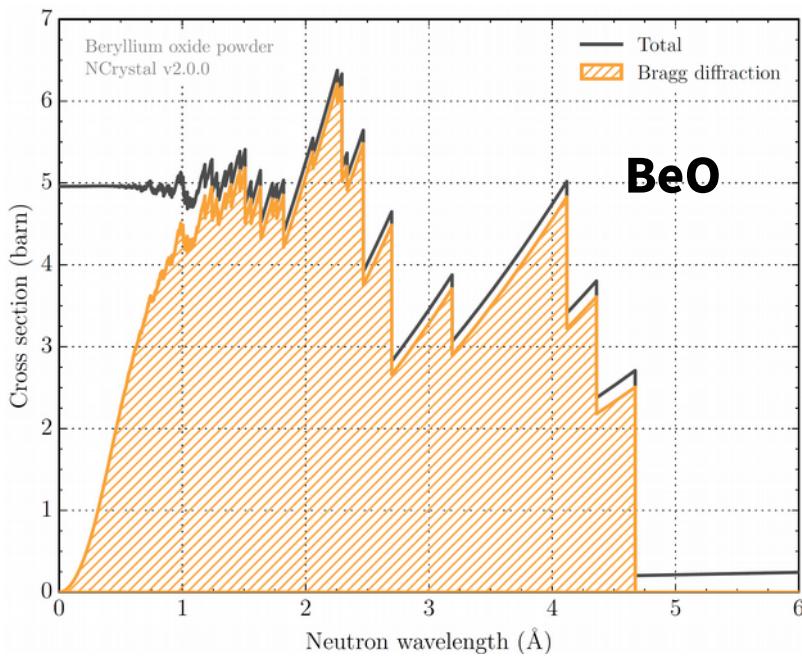
NCrystal

x. x. Cai & T. Kittelmann

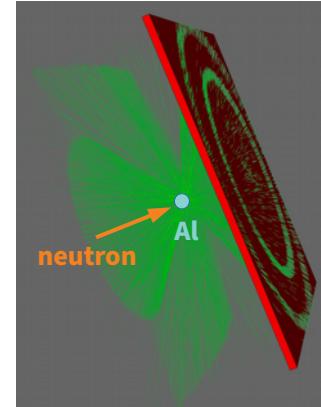
Based on provided HKL planes with d-spacings and structure factors, the implementation is straight-forward.

Care is taken to be extremely fast $O(10\text{ns}/\text{call})$, even in case of huge number of planes.

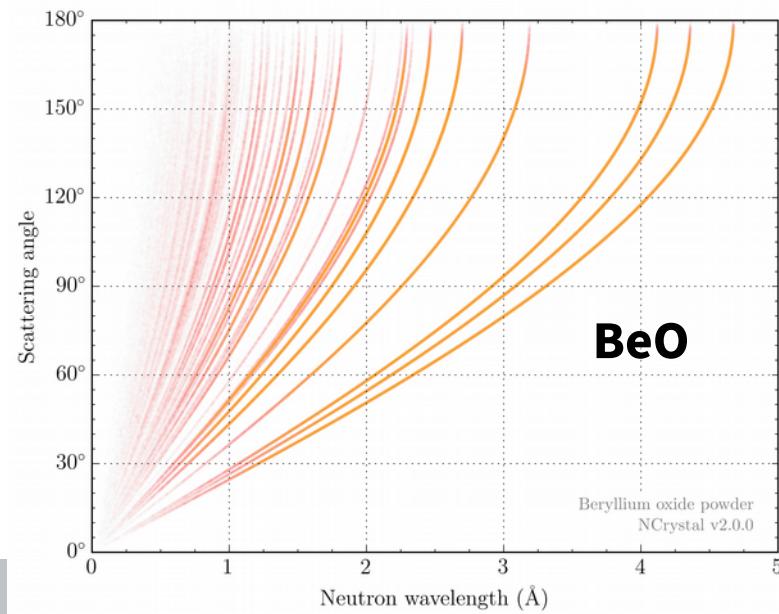
Currently no texture/grain-size effects.



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



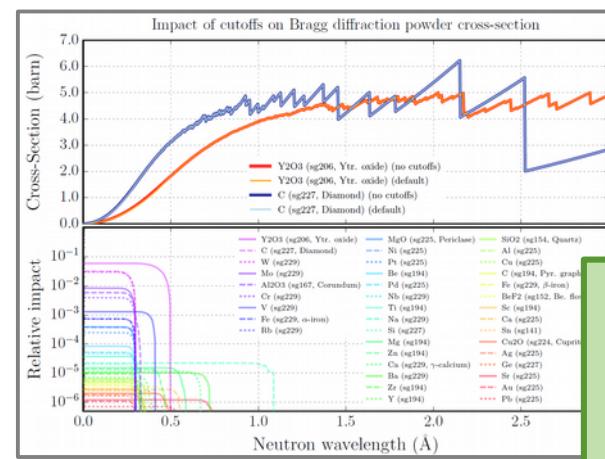
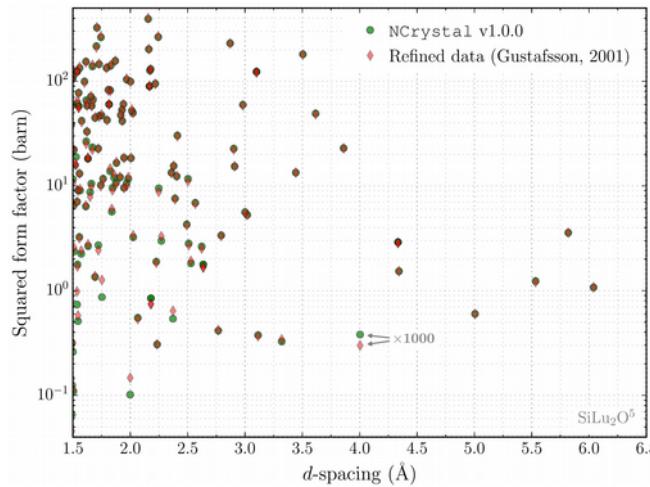
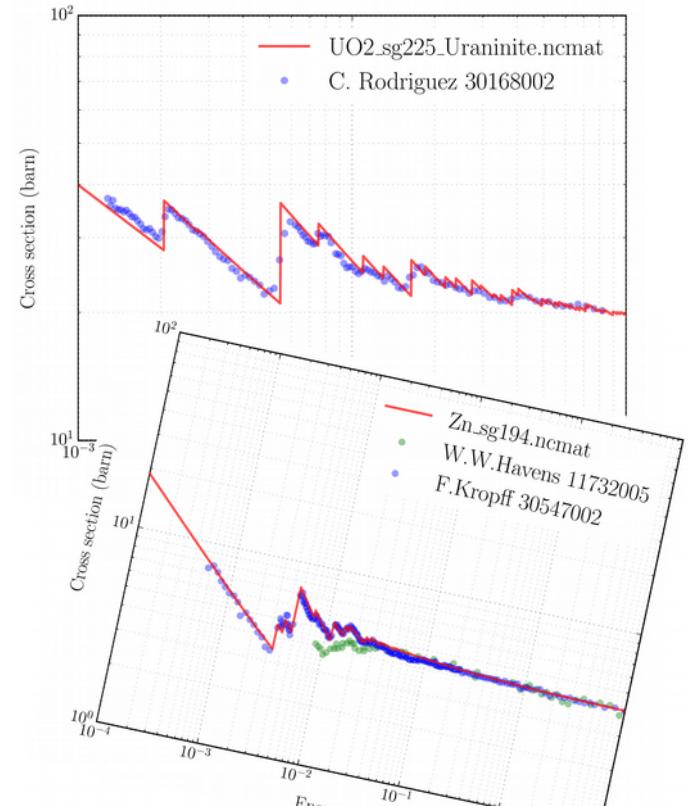
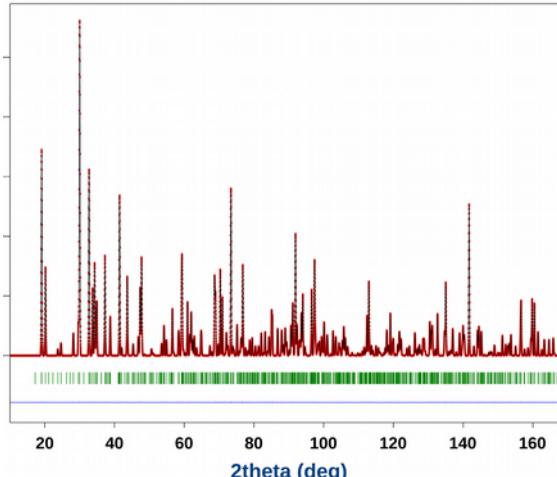
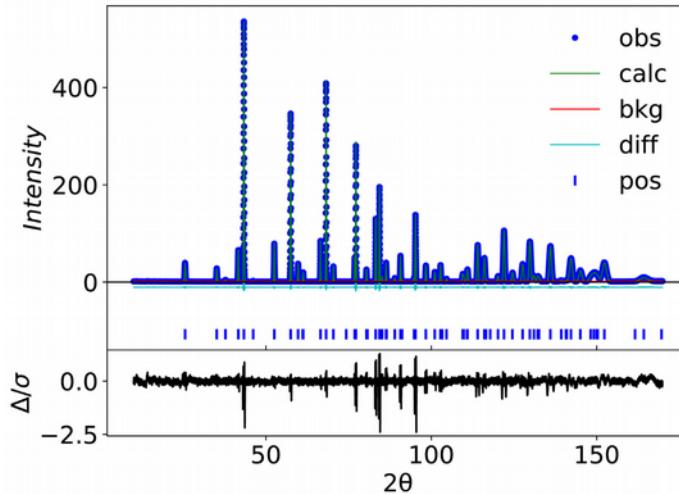
Geant4 with NCrystal
⇒Debye-Scherrer cones



Structure factors and powder model validated thoroughly.

NCrystal

X. X. Cai & T. Kittelmann



- 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.

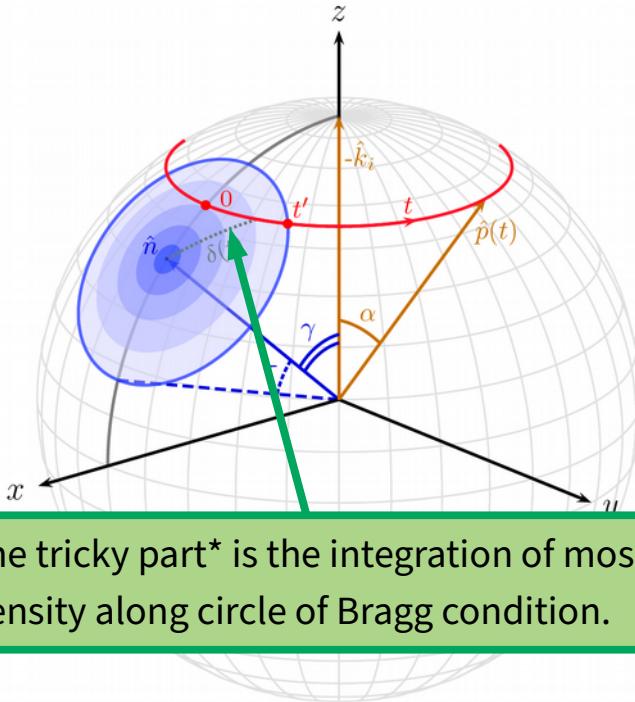
Single Crystals with Gaussian mosaicity

Can model monochromators, analysers, filters, samples

Handles also large mosaicities and backscattering!

NCrystal

X. X. Cai & T. Kittelmann



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

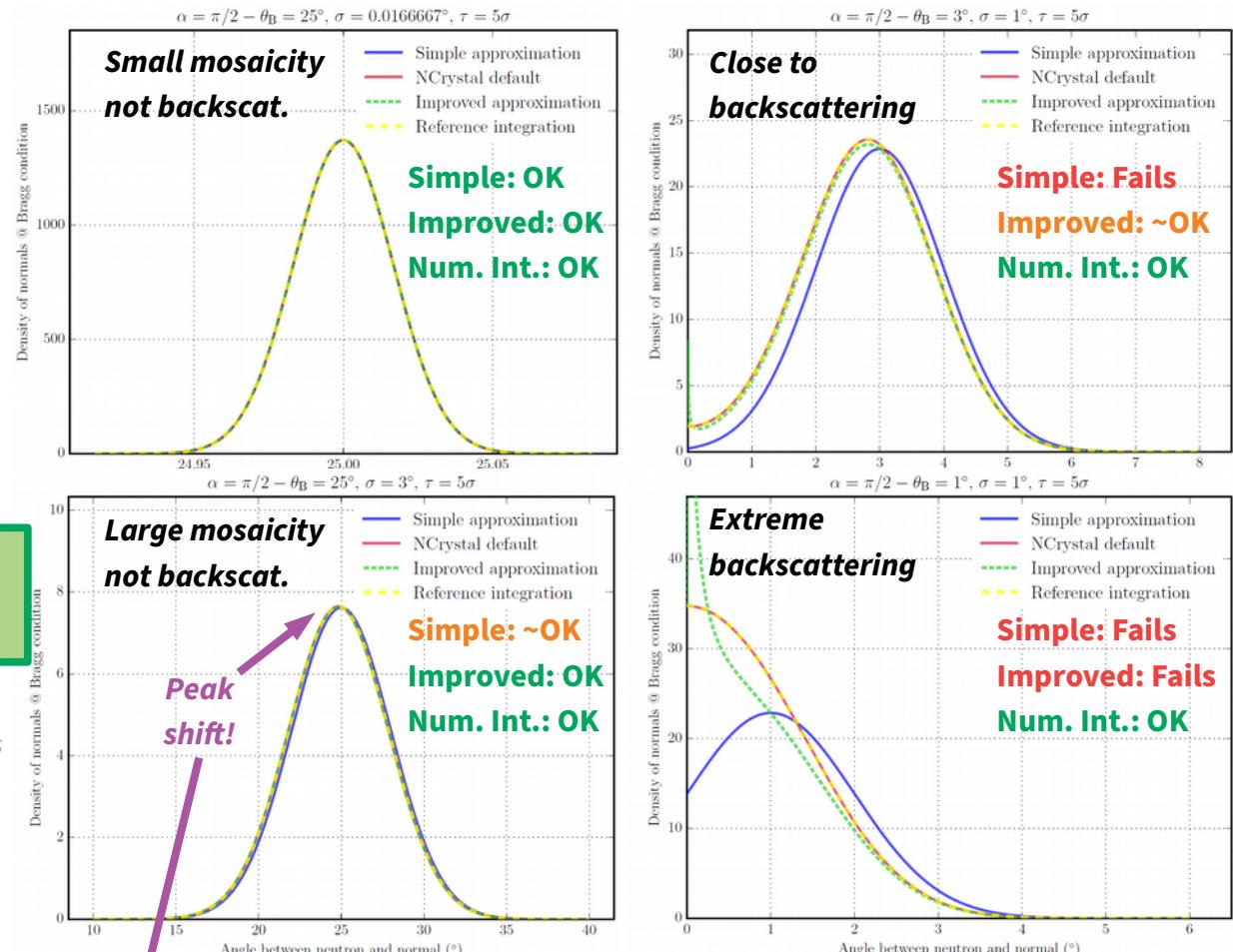
*: Once contributing normals
Have been identified.

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

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

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

Improved form extends validity
to much larger mosaicities

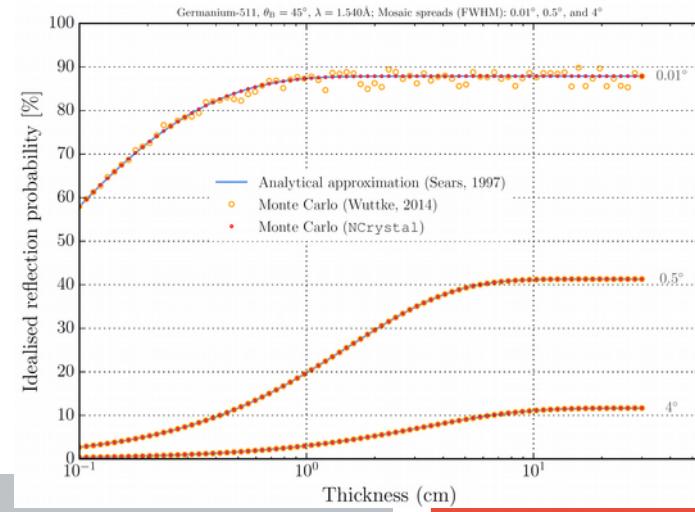
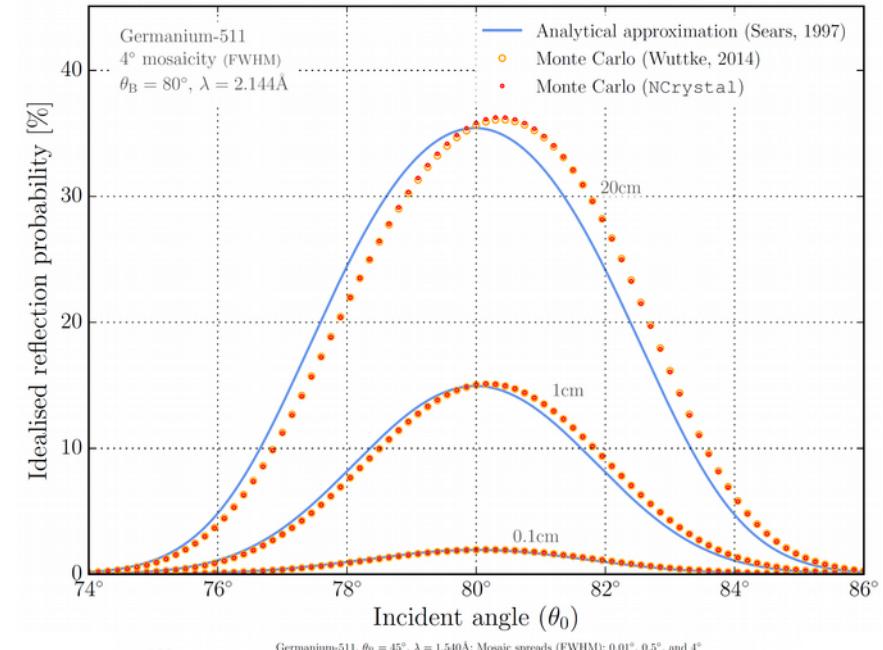
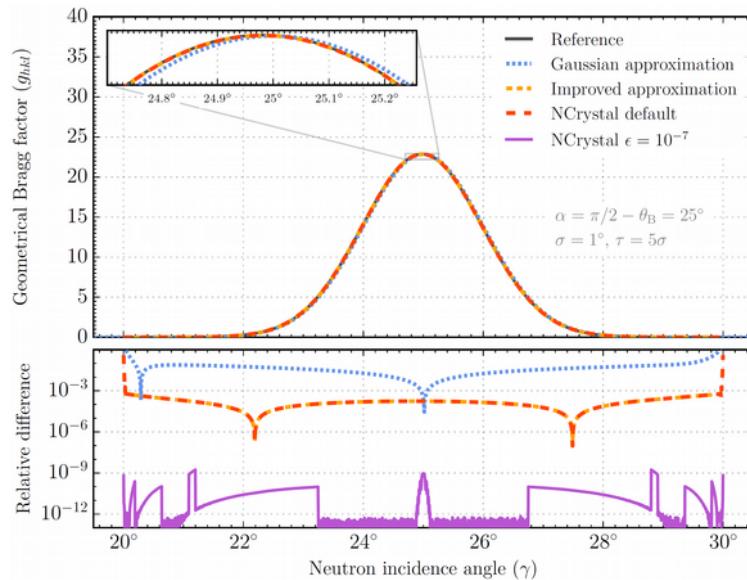


Single crystal model validated

NCrystal
X. X. Cai & T. Kittelmann

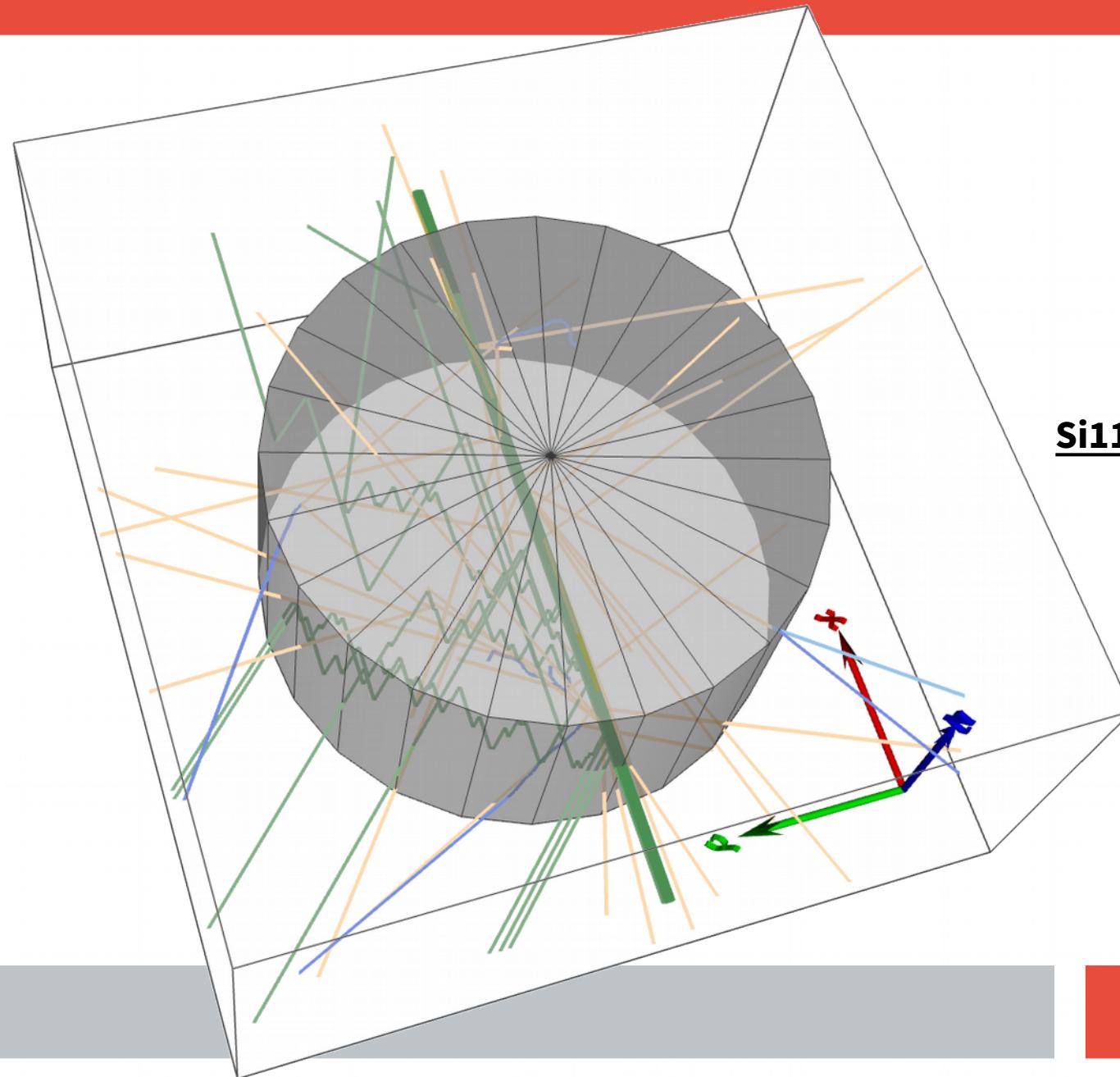
Validation includes:

- 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”)



Geant4+NCrystal → Unprecedented realism in monochr./analyser/filter/sample simulations

NCrystal
X. X. Cai & T. Kittelmann



Si111 monochromator

neutrons
gammas
positrons
electrons

Special anisotropic model for Pyrolytic Graphite

PG often used as filters, monochromator, analyser

NCrystal

X. X. Cai & T. Kittelmann

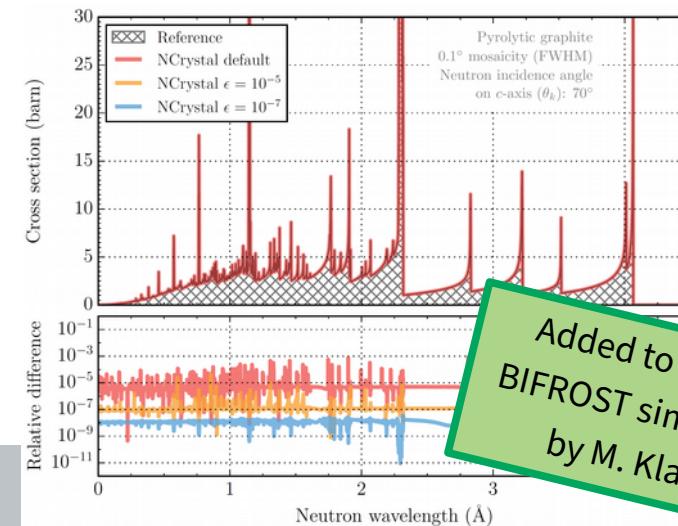
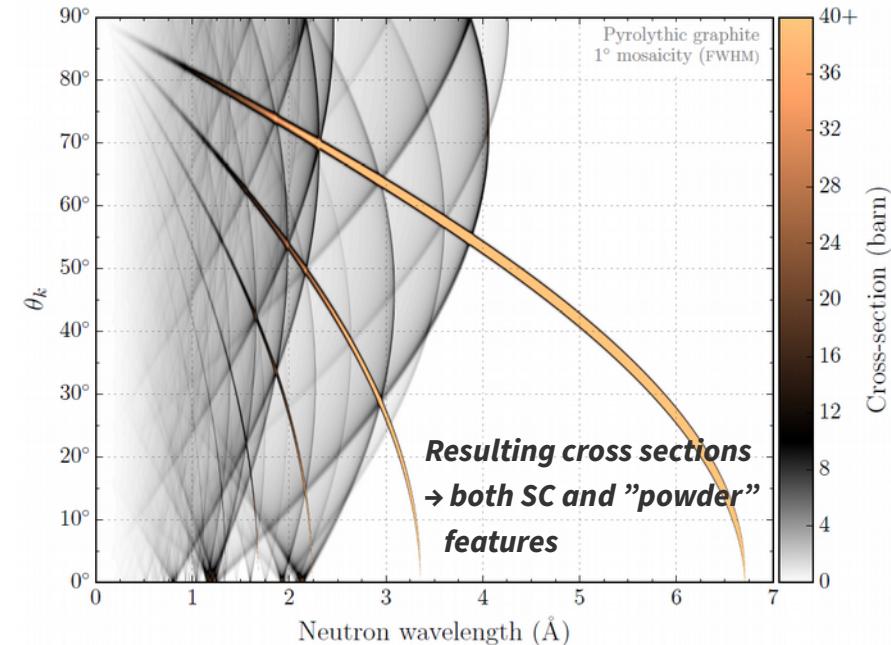
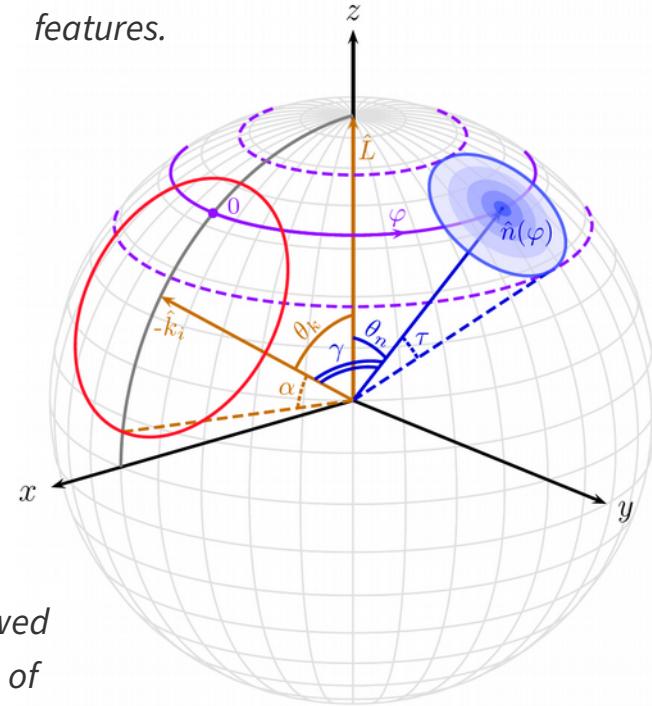


Features:

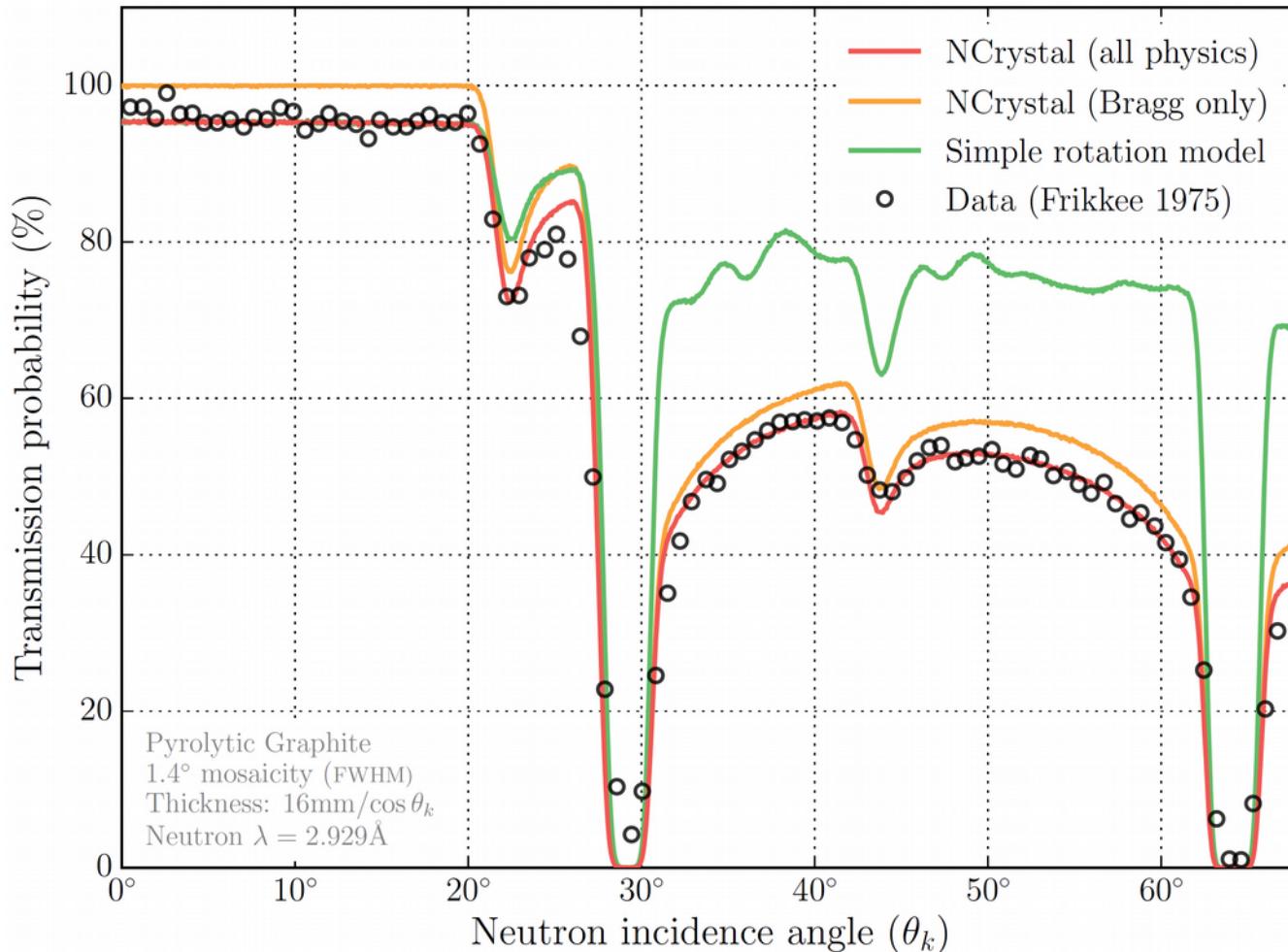
- Cross-sections determined by efficient pre-search followed by fast Romberg integration of non-layered single crystal code.
- Features realistic transmission probabilities and multiple-scattering effects (incl. "zig-zag walk")

Layered crystal model:

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



Can reproduce PG transmission spectra!



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”).

Inelastic physics : Scattering kernels

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

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

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

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

Under some assumptions $S(Q, \omega)$ 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* (Q dependency becomes scalar)
- *Incoherent approximation*: Off-diagonal entries in S_{coh} wash out when integrating over isotropic grain distribution, so $\text{shape}(S_{\text{coh}}) \approx \text{shape}(S_{\text{inc}})$.

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

Formulation in dimensionless variables: α, β

NCrystal

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

X. X. Cai & T. Kittelmann

α : dimensionless q^2

β : dimensionless ΔE

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

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

Scattering lengths taken outside definition of S:

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

Total cross section, with explicit kinematic limits:

constant affects $\sigma(E)$,
but not (α, β) -sampling

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

neutron lose all its energy

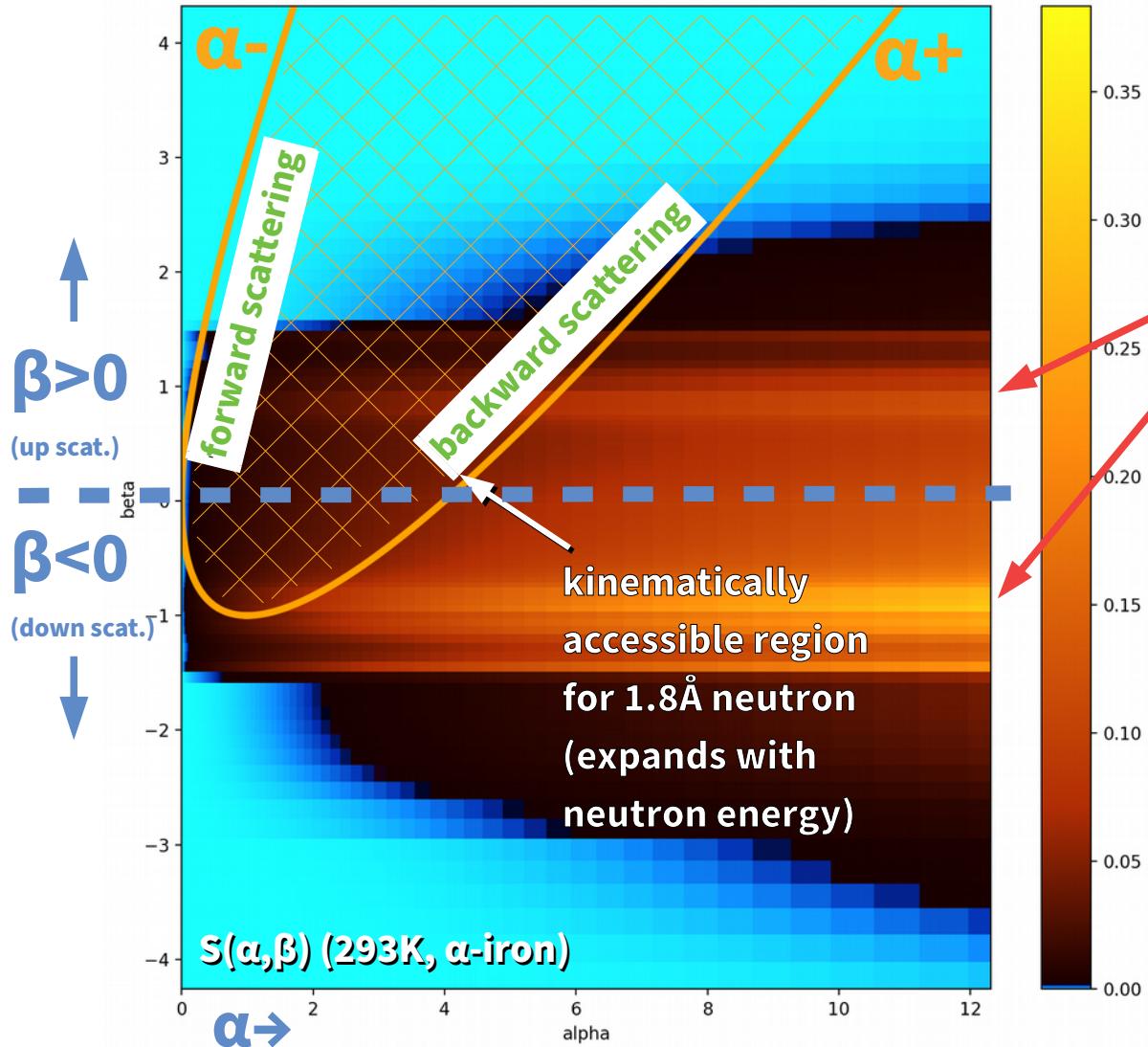
$\mu=-1$, complete backwards scattering

$\mu=+1$, complete forward scattering

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

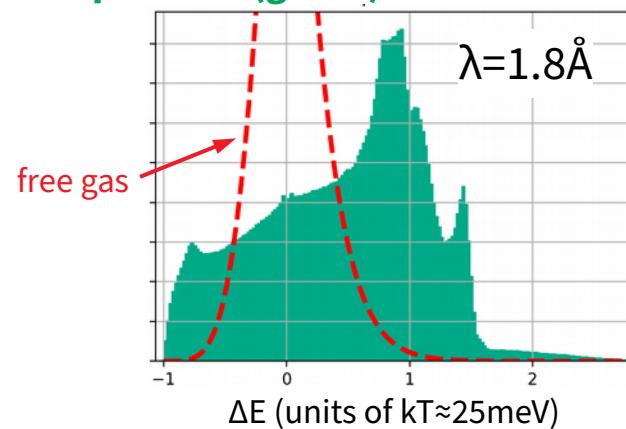
kinematically accessible
region is a parabola in
the (α, β) -plane

$S(\alpha, \beta)$ and connection to neutron



"Detailed balance":
 Boltzmann factor + identical strength of phonon absorption & emission \rightarrow Symmetry around $\beta=0$, given by $S(\alpha, -\beta) = \exp(\beta) \cdot S(\alpha, \beta)$

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



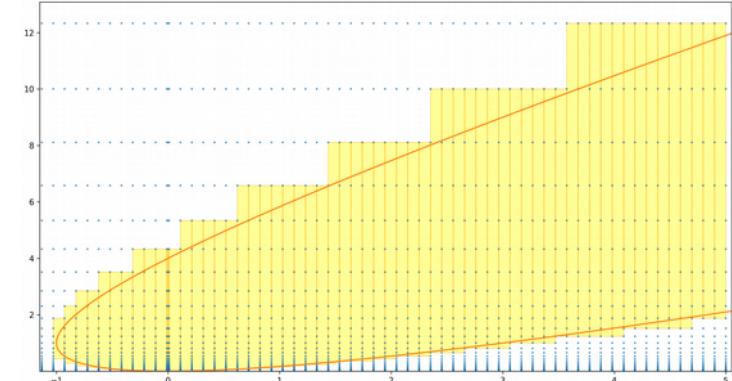
Technical details of scattering kernel treatment in NCystal

NCystal

X. X. Cai & T. Kittelmann

- No details today, but includes:

- How to integrate a kernel to get $\sigma(E)$
- How to sample a kernel in case of a scattering
(we improved this!)
- How to extrapolate beyond reach of table (needs free-gas model implementation)
- Works well. Have ideas for additional improvements.



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



ARTICLE INFO

Article history:

Received 19 June 2018

Received in revised form 12 November 2018

Accepted 30 November 2018

Available online 13 December 2018

ABSTRACT

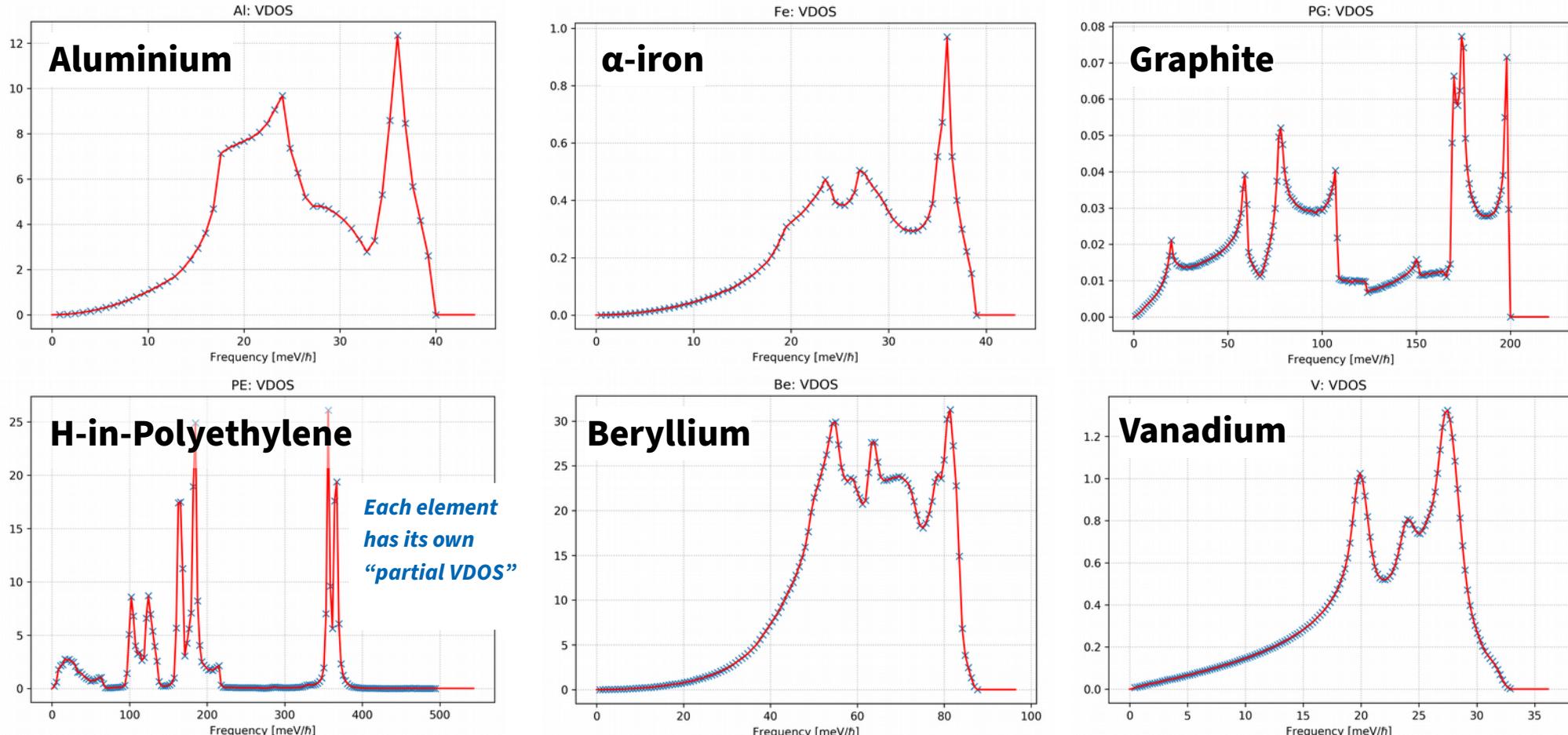
Distributions of inelastically scattered neutrons can be quantum dynamically described by a scattering kernel. We present an accurate and computationally efficient rejection method for sampling a given scattering kernel of any isotropic material. The proposed method produces continuous neutron energy and angular distributions, typically using just a single interpolation per sampling. We benchmark the results of this method against those from accurate analytical models and one of the major neutron transport codes. We also show

Connecting scattering kernels to phonons

Unlike scat. kernels, phonon spectra widely available

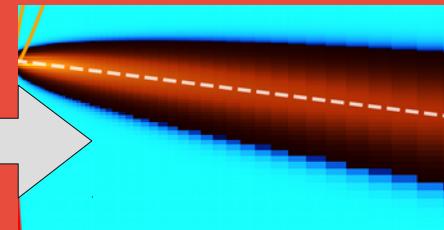
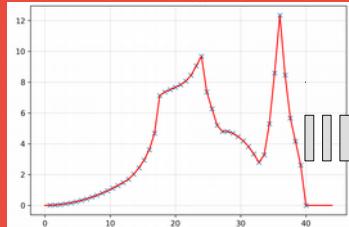
NCrystal

X. X. Cai & T. Kittelmann



Captures material structure info relevant for inelastic neutron scatterings (isotropic materials, incoherent approximation)

VDOS \rightarrow S(α, β)



NCrystal
X. X. Cai & T. Kittelmann

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 [10-10].

Well-established method!

Used in NJOY/LEAPR. Most ENDF S(α, β) kernels were created this way.

Idea: Build this capability into NCrytal and make it fast (<1s) so can invoke 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:**
Static S(α, β) is only valid for specific T.

VDOS \rightarrow $S(\alpha, \beta)$ Sjölander's recipe

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

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

• • • • • • • • • • • • • • •

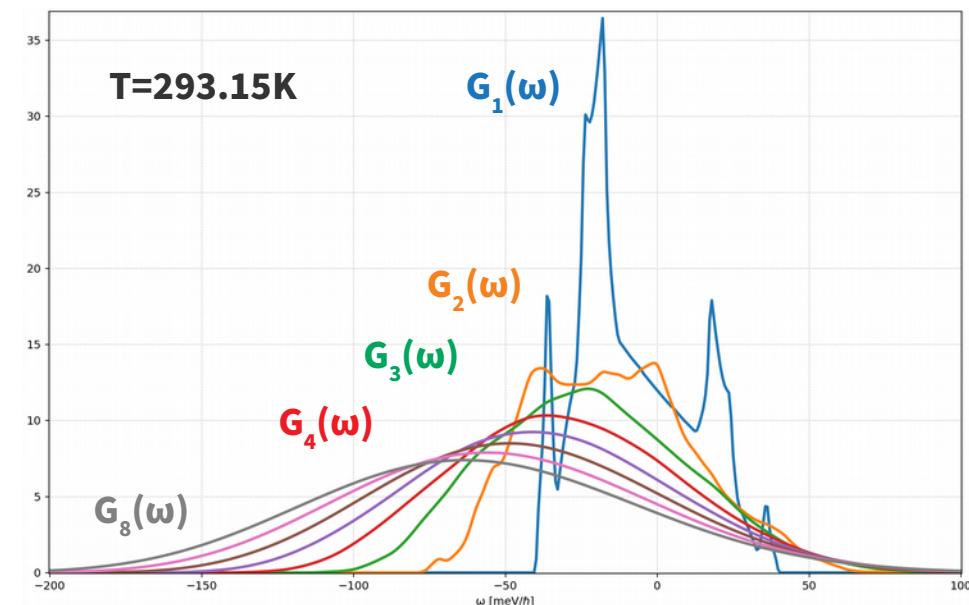
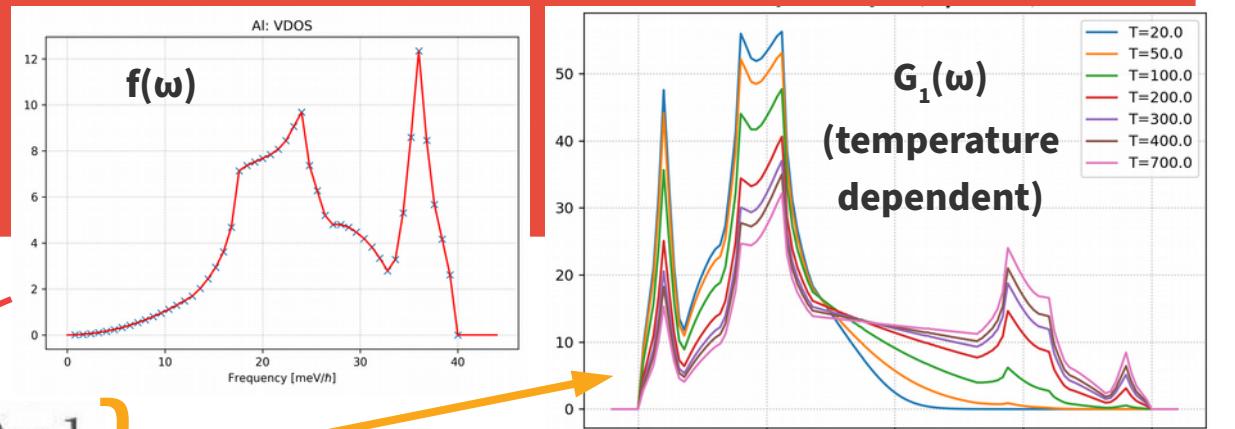
$$G_{n+1}(\omega) = \int_{-\infty}^{\infty} g(\omega - \omega') G_n(\omega') d\omega'.$$

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

σ_b

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

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

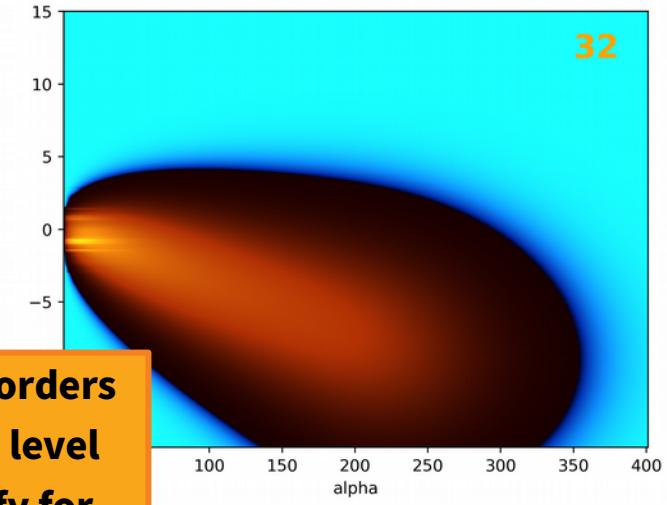
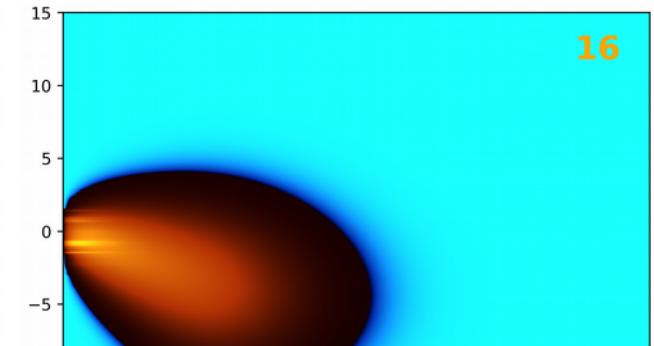
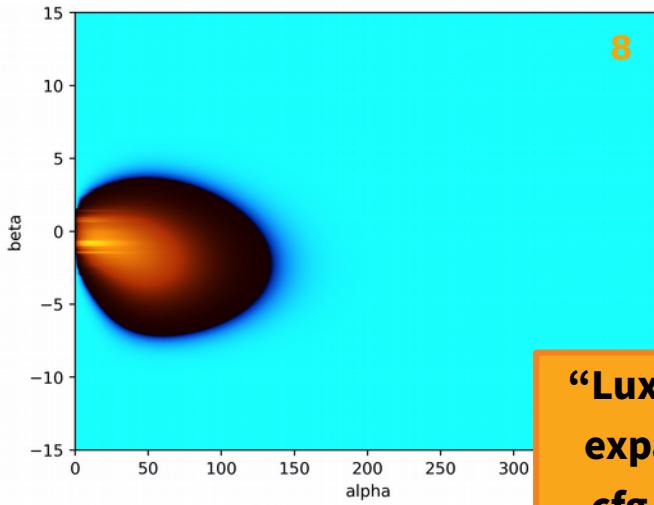
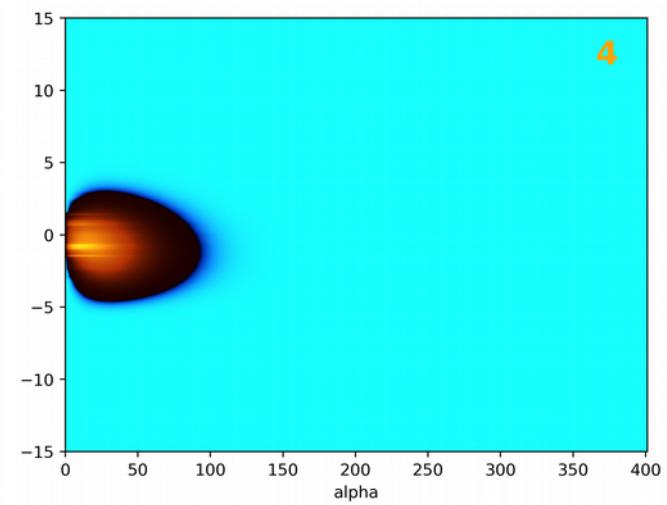
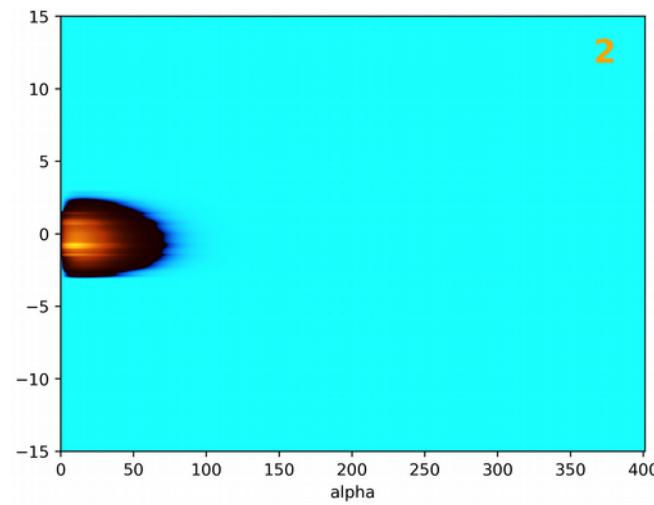
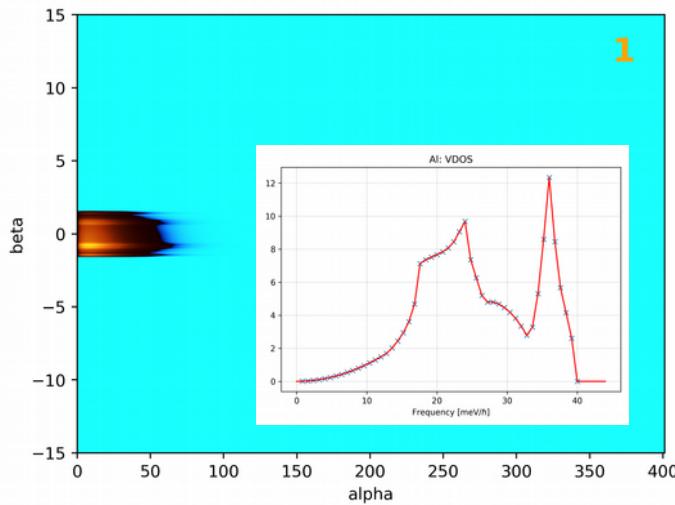


VDOS \rightarrow S(α, β): \rightarrow Aluminium

<http://cern.ch/tkittel/vdosanim>

NCrystal

X. X. Cai & T. Kittelmann



"Luxury" in this expansion (grid density, orders expanded, etc.) controlled by single high level cfg parameter "vdoslux" \Rightarrow easy to modify for any user (see backup slide for details).

Default value is of course sensible.

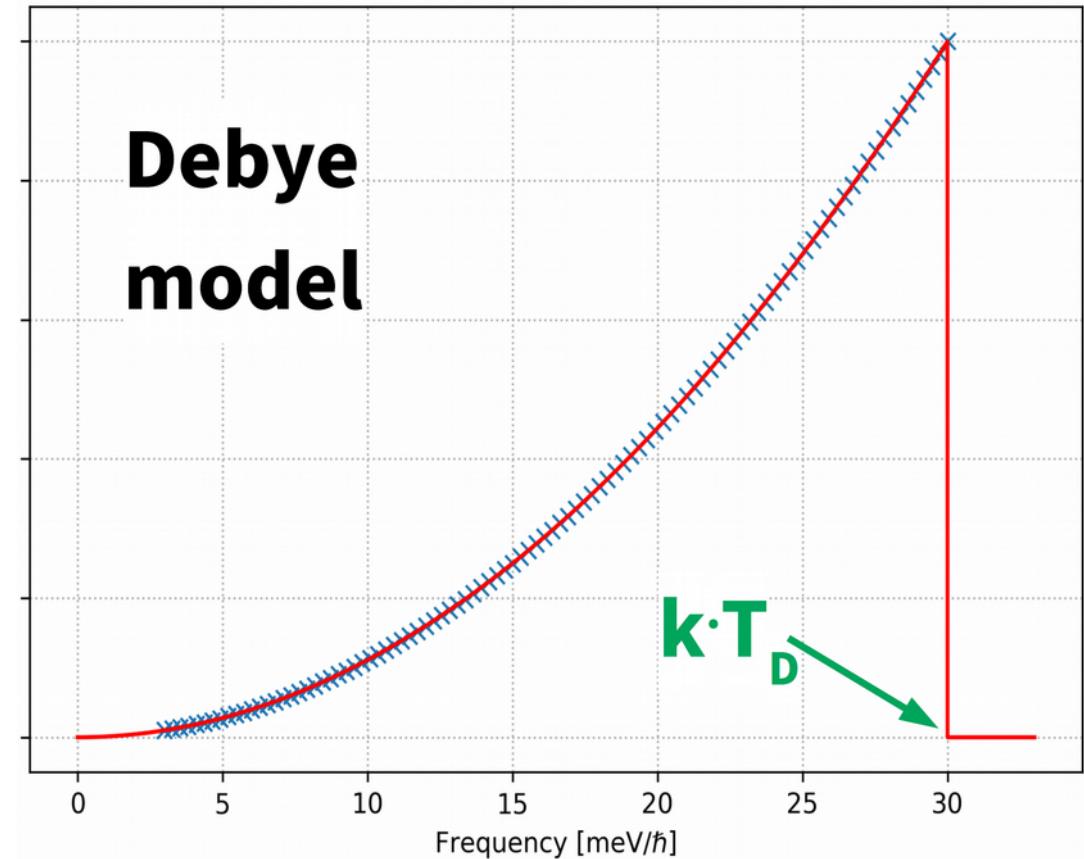
How we handle materials with no DOS specified?

NCrystal

X. X. Cai & T. Kittelmann

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 rather well.



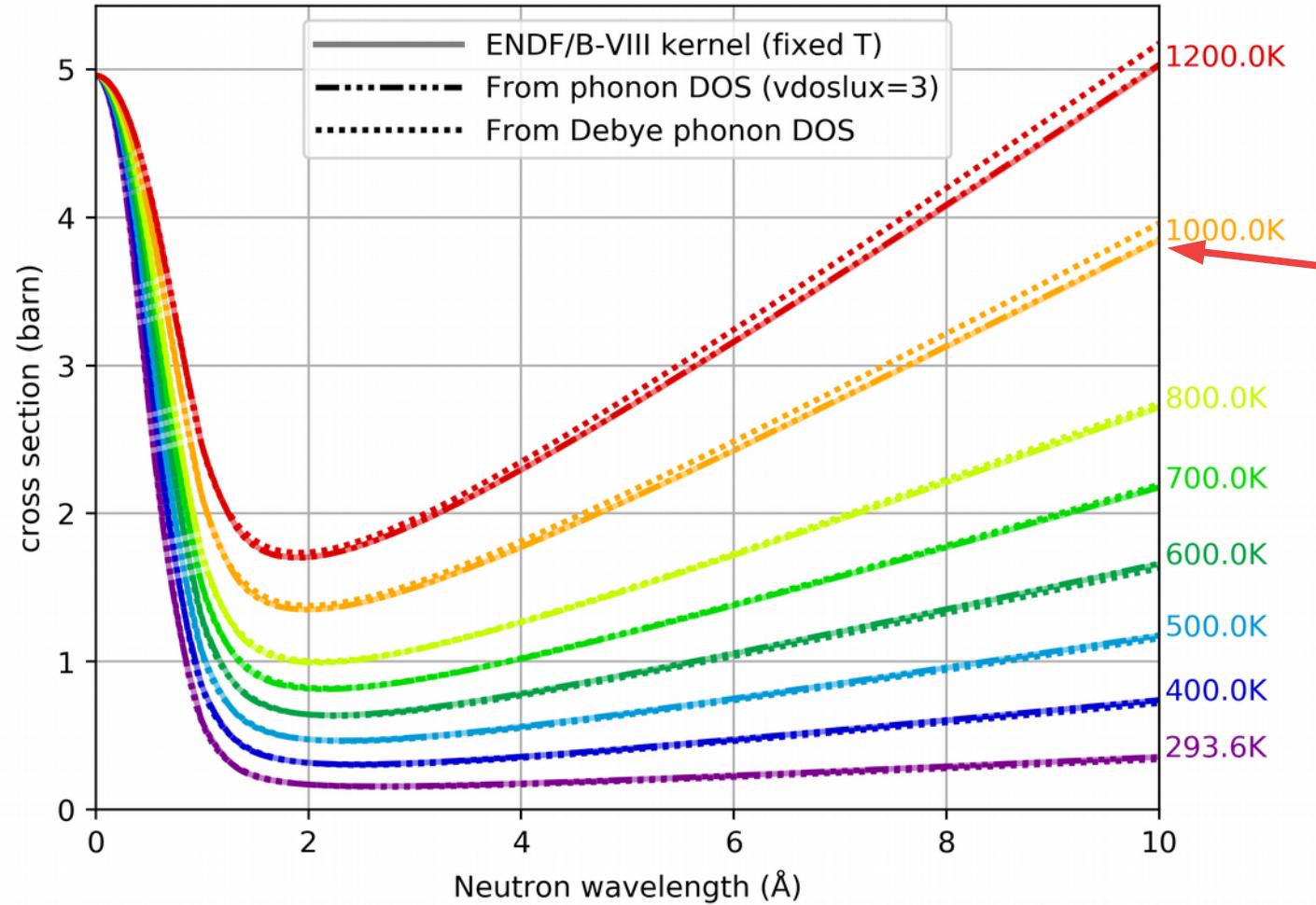
Example inelastic cross sections

(compared with ENDF kernel which is only available at 8 fixed temperatures)

NCrystal

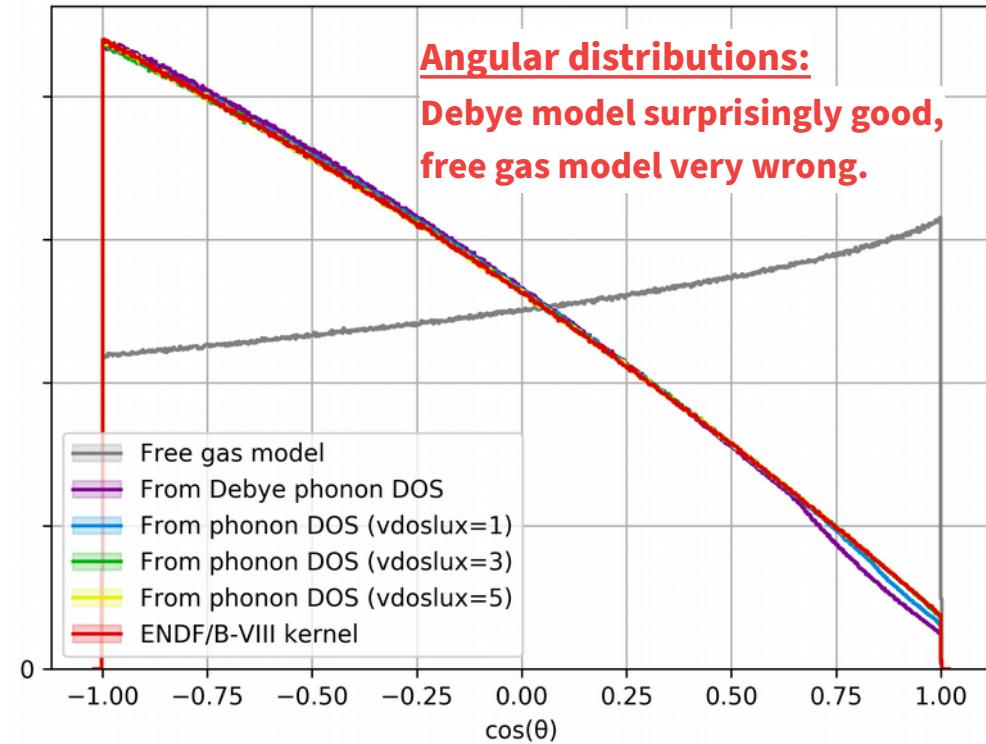
X. X. Cai & T. Kittelmann

Inelastic scattering cross section for Beryllium Oxide (NCrystal v2.0.0)

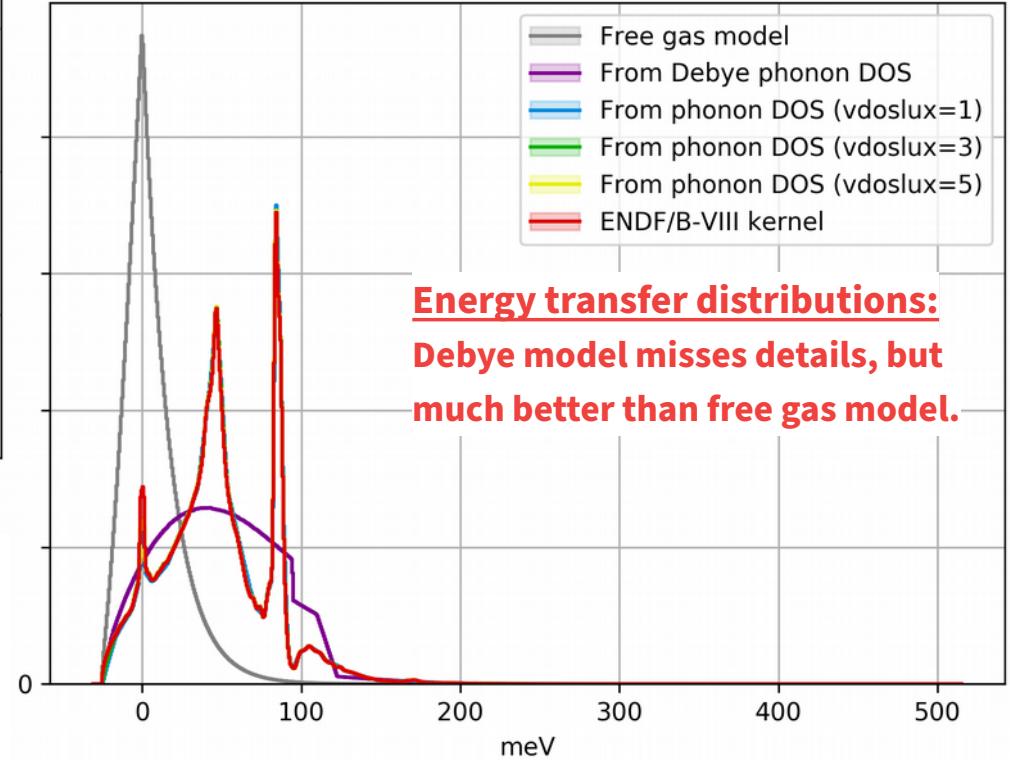


Example inelastic scattering distributions

BeO, Scattering angle in scatterings at 1.8 Åa (NCrystal v2.0.0)



BeO, Energy transfer in scatterings at 1.8 Åa (NCrystal v2.0.0)



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

NCrystal
X. X. Cai & T. Kittelmann

Annals of Nuclear Energy 65 (2014) 280–289

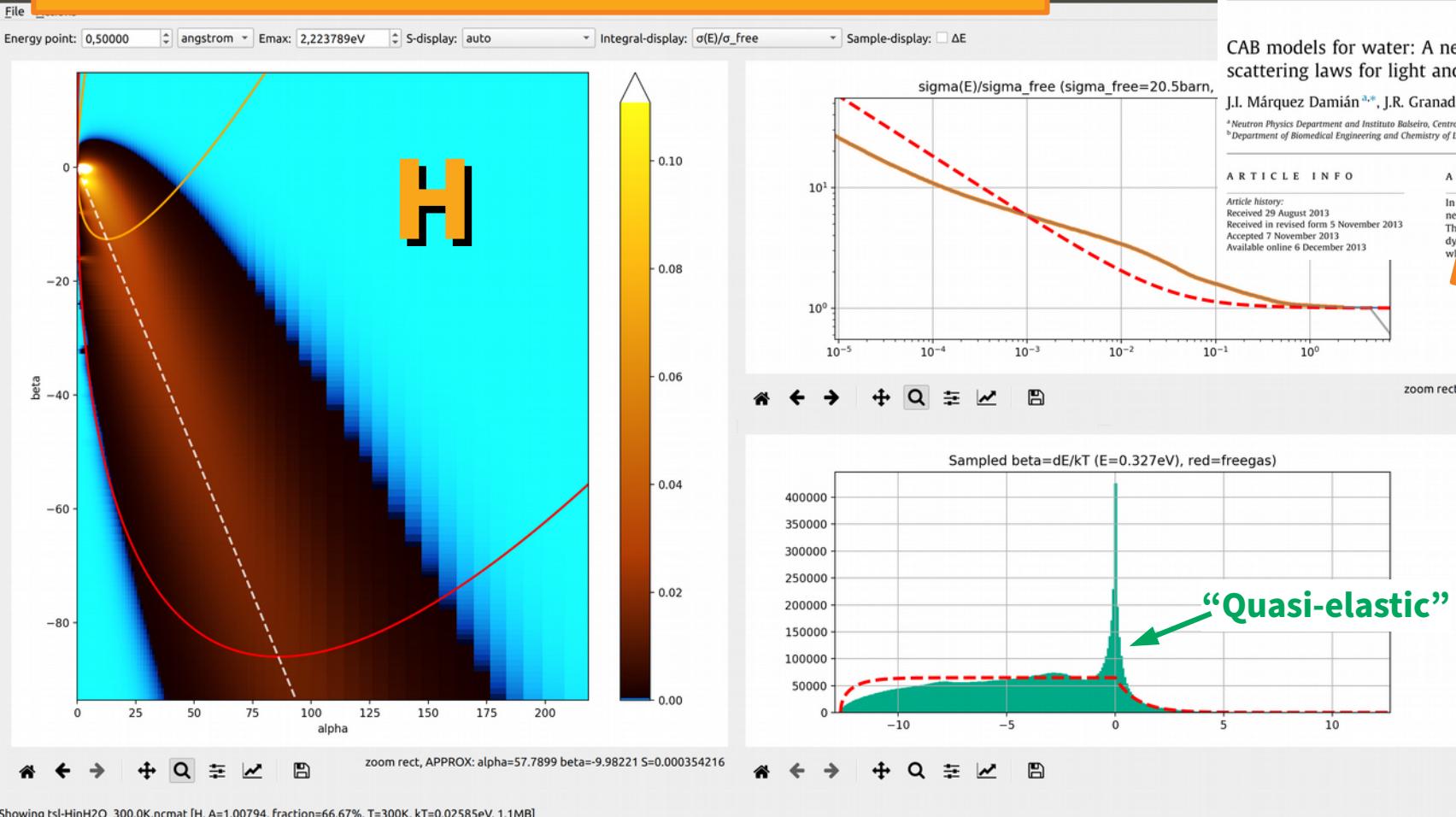


Contents lists available at ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene

**VDOS \rightarrow S(α, β) does not work directly for liquids!
NCrystal will rely on kernel converted from ENDF.**



CAB models for water: A new evaluation of the thermal neutron scattering laws for light and heavy water in ENDF-6 format

J.I. Márquez Damián ^{a,*}, J.R. Granada ^a, D.C. Malaspina ^b

^a Neutron Physics Department and Instituto Balseiro, Centro Atómico Bariloche, CNEA, Argentina

^b Department of Biomedical Engineering and Chemistry of Life Processes Institute, Northwestern University, 2145 Sheridan Road, Evanston, IL, USA

ARTICLE INFO

Article history:
Received 29 August 2013
Received in revised form 5 November 2013
Accepted 7 November 2013
Available online 6 December 2013

In this paper we present CAB models for water: a set of new models for the thermal neutron scattering laws for light and heavy water in ENDF-6 format, using the Cabannes-Bonamy (CAB) theory. The new models are able to predict the dynamic scattering laws for both light and heavy water, which are very different from the standard S₀ model. The new models are able to predict the dynamic scattering laws for both light and heavy water, which are very different from the standard S₀ model.

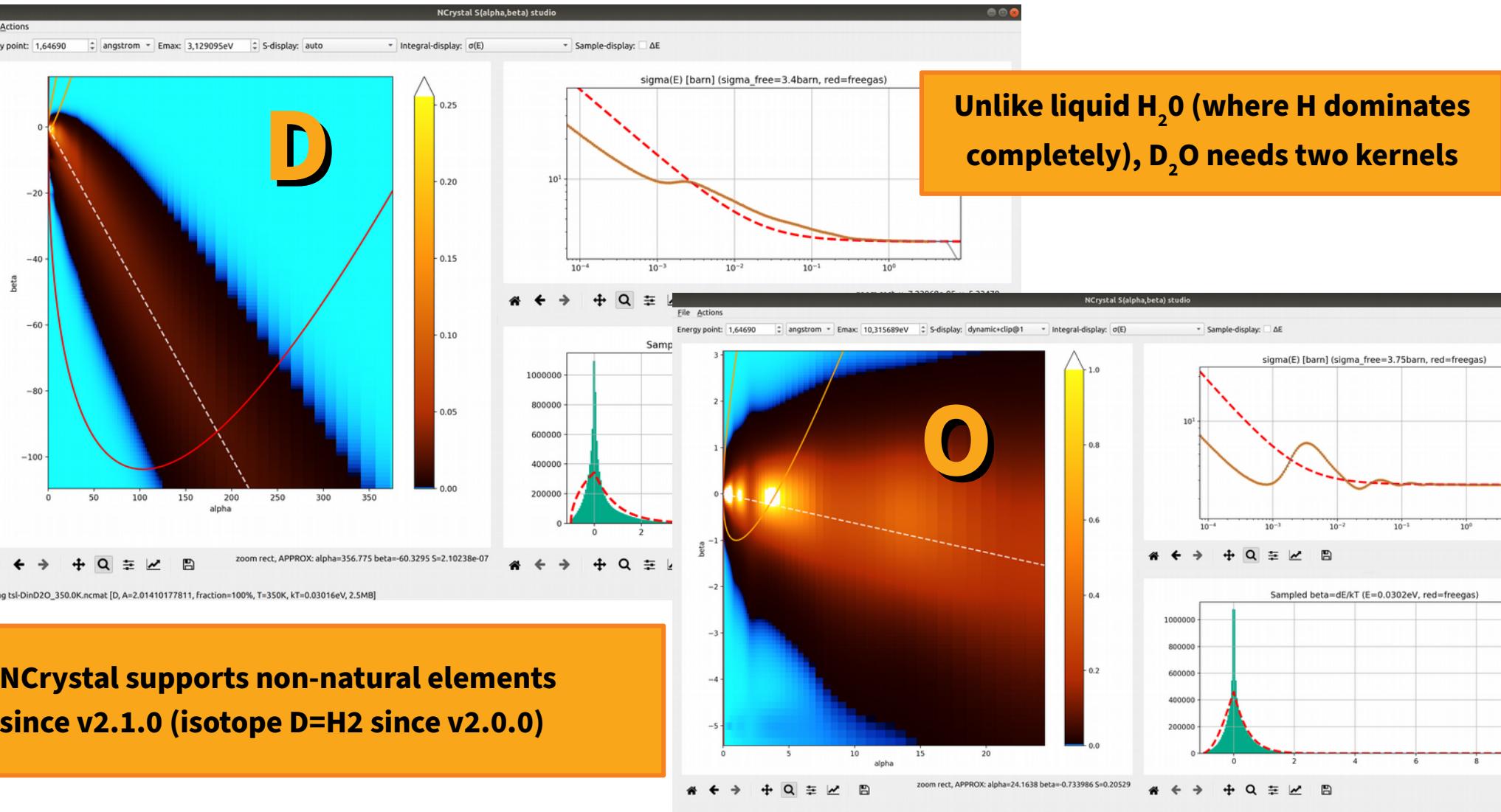
ENDF water recently updated

Oxygen is modelled as free gas

... and heavy water (also converted from ENDF8)

NCrystal

X. X. Cai & T. Kittelmann



Incoherent-elastic scattering

NCrystal

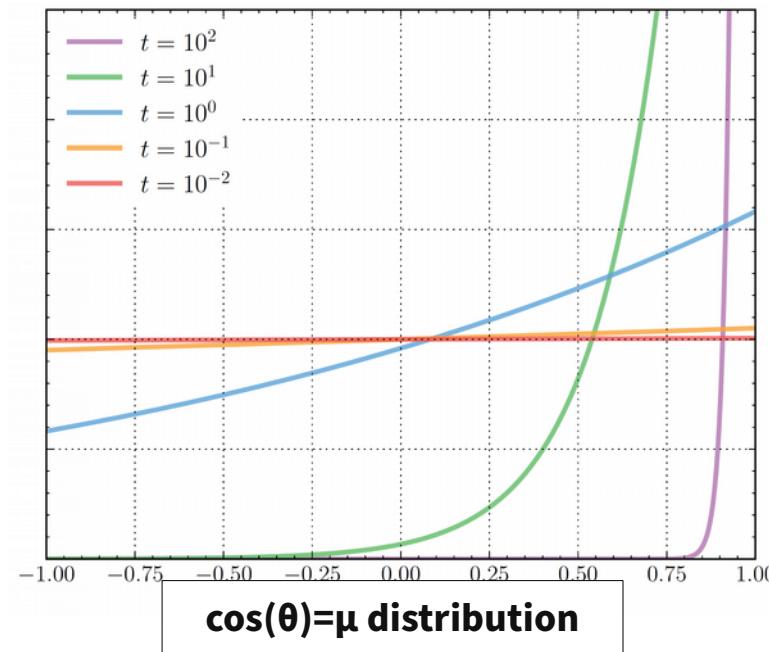
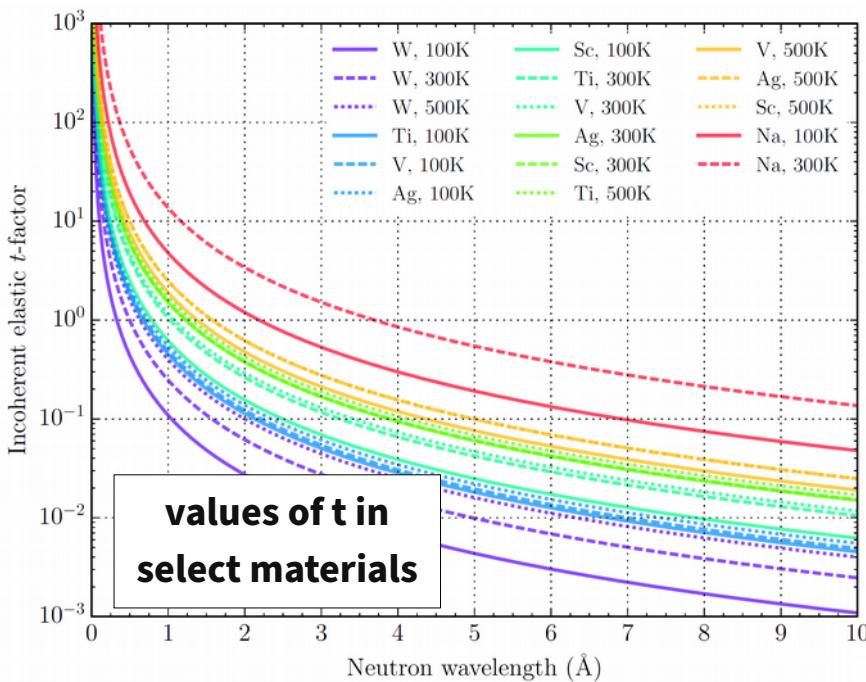
X. X. Cai & T. Kittelmann

$$\frac{d\sigma_{\vec{k}_i \rightarrow \vec{k}_f}^{\text{inc,el}}}{d\Omega_f} = \frac{1}{N} \sum_{j=1}^N \frac{\sigma_j^{\text{inc}}}{4\pi} e^{-2W_j(\vec{Q})} \longrightarrow \sigma^{\text{inc,el}}(k) = \sigma_{\text{inc}} \frac{1 - \exp(-t)}{t}$$

Get Debye-Waller factors (or δ^2) from phonon DOS (or Debye temp.).

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

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

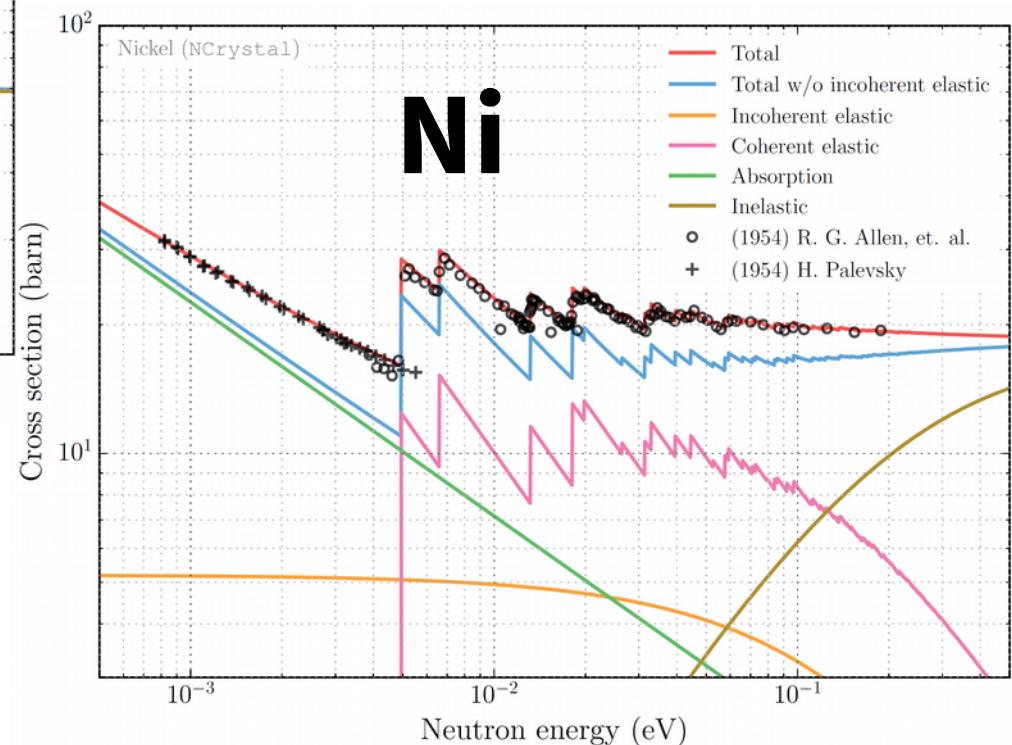
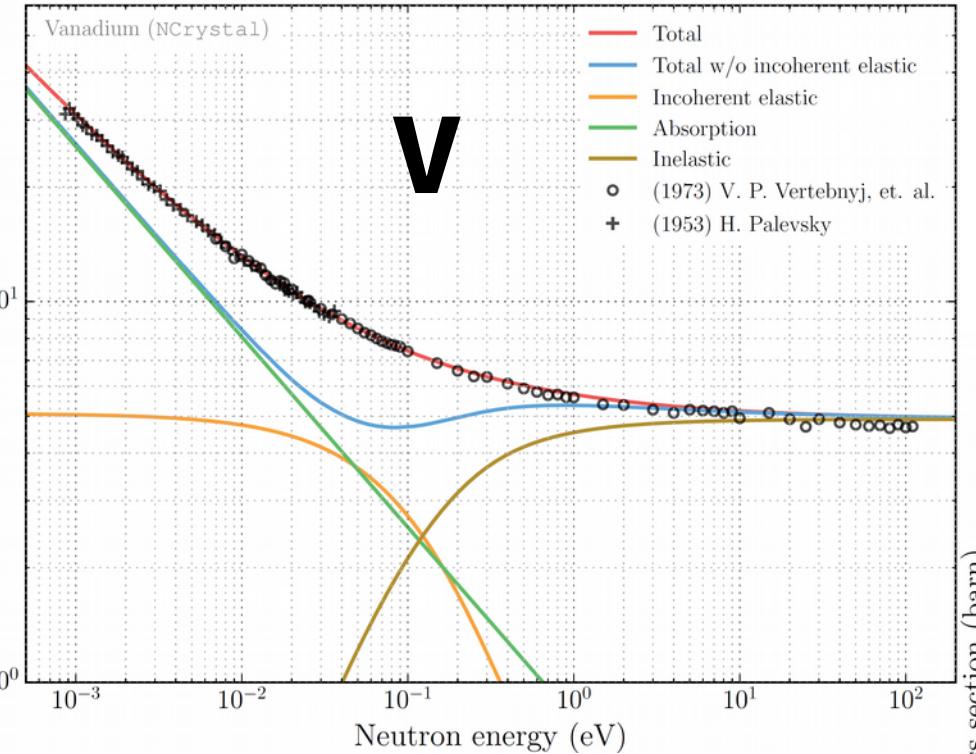


Incoherent-elastic model validations

→ so far only with total cross sections

NCrystal

X. X. Cai & T. Kittelmann



Recent/planned improvements

Flexible atomic (re)definitions

NCrystal
X. X. Cai & T. Kittelmann

NCrystal v2.1.0 introduced support for atoms which are not just natural elements!

- Ships with database of 80+ natural elements and 261+ isotopes.
- Possibility to customise (in .ncmat files or cfg vars).

Plan to add multi-phase support in the near future as well.

- Can improve many materials.
- Can be used to model crystal grains suspended in a liquid.

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");
```

Extend NCrystal with plugins

NCrystal v2.1.0+v2.2.0 made it easier to add custom physics models to NCrystal

- 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 still require C++ knowledge, but now it can be done with less nasty hacks :-)
- 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>
- Already we have a few plugins starting development in this way:
 - Plugin with empirical nanodiamond model (Nicola Rizzi, DTU)
 - SANS (Zhiyang Wang & Xiao Xiao Cai, CSNS).
 - Improved Bragg diffraction models (texture, ...) (Matteo Busi, PSI)

In-memory data files

Optional: can build entire data library into the compiled NCrystal binary lib

NCrystal

X. X. Cai & T. Kittelmann

```
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")
```

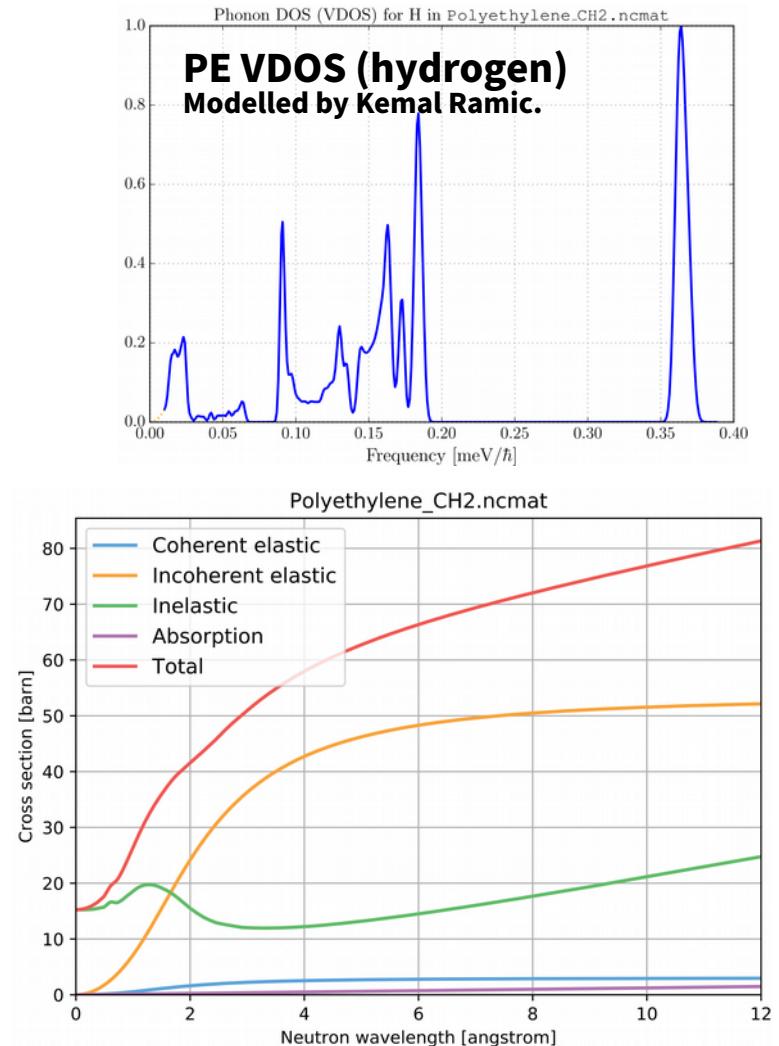
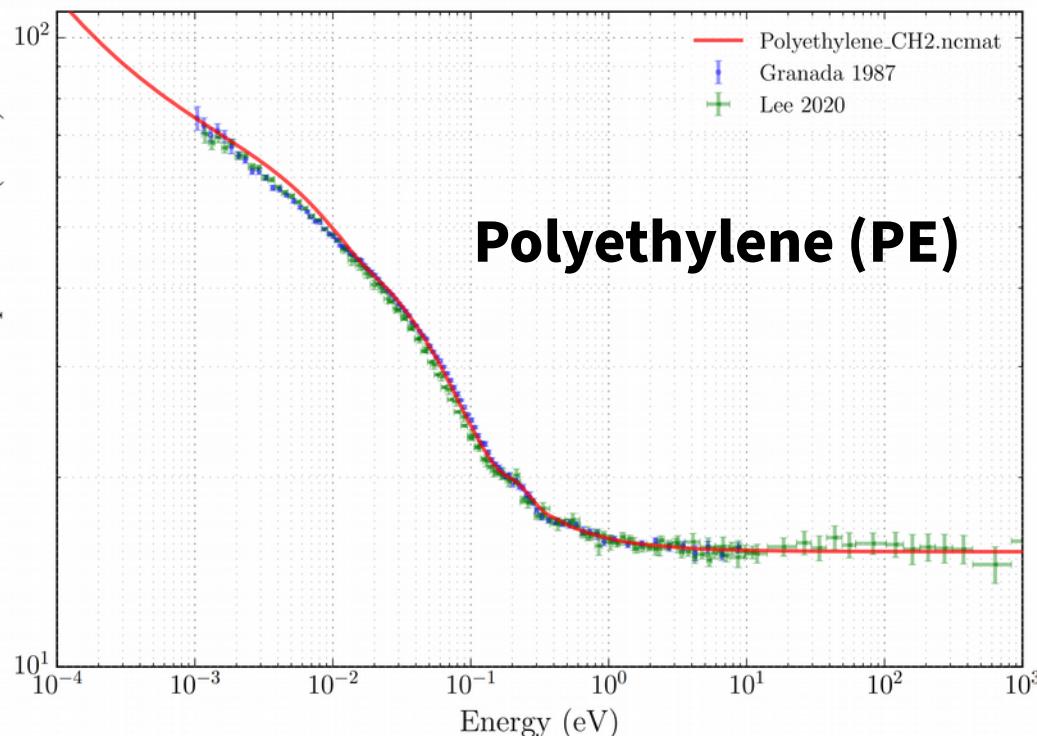
Amorphous solids

New feature in NCrystal v2.7.0

NCrystal

X. X. Cai & T. Kittelmann

- Uses same inelastic and incoherent-elastic approach as for crystalline solids.
- Coherent-elastic scattering via incoherent approximation. Hope to add proper support for static structure factor at some point.



Hydrogen-rich amorphous solids

New feature in NCystal v2.7.0

NCystal

X. X. Cai & T. Kittelmann

- DFT/MD modelling of amorphous materials can be difficult and time consuming.
- Recent paper (Romanelli et. al., arxiv 2102.06147) 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.

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

Giovanni Romanelli,¹ Dalila Onorati,^{2,a)} 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 Giuseppe Gorini,⁴ Carla Andreani,^{2,7} and Roberto Senesi^{2,8}

¹ISIS Neutron and Muon Source, UKRI-STFC, Rutherford Appleton Laboratory, Harwell Campus, Didcot, Oxfordshire OX11 0QX, United Kingdom

²Università degli Studi di Roma "Tor Vergata", Dipartimento di Fisica and NAST Centre, Via della Ricerca Scientifica 1,

Roma 00133, Italy

³Università degli Studi di Roma "Tor Vergata", Dipartimento di Scienze e Tecnologie Chimiche, Via della Ricerca Scientifica 1,

Roma 00133, Italy

⁴Università di Milano-Bicocca, Piazza della Scienza 3, Milano, Italy

⁵European Spallation Source ERIC, P.O. Box 176, 22100 Lund, Sweden

⁶Istituto per la Scienza e Tecnologia dei Plasmi, CNR, via Cozzi 53, 20125 Milano, Italy

⁷CNR-ISRM, Area della Ricerca di Roma Tor Vergata, Via del Fosso del Cavaliere 100, 00133 Roma, Italy

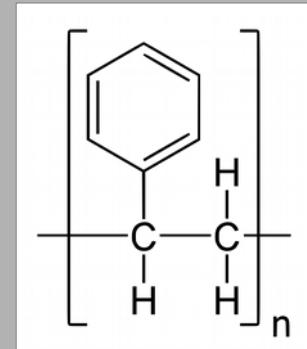
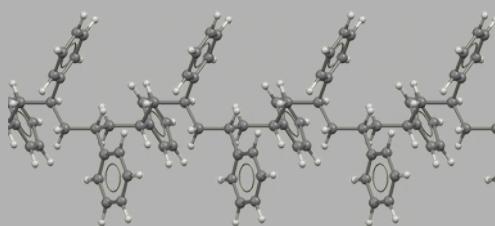
⁸CNR-IPCF, Sezione di Messina, Viale Ferdinando Stagno d'Alcontres 37, Messina, 98158, Italy

(Dated: 12 February 2021)

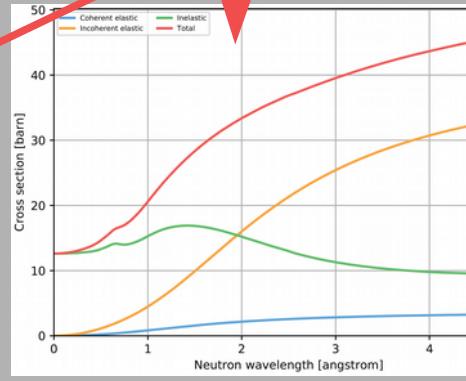
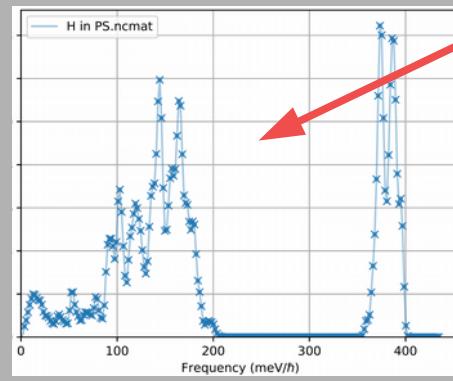
The experimental thermal neutron cross sections of the twenty proteinogenic amino acids have been measured over the incident-neutron energy range spanning from 1 meV to 10 keV and data have been interpreted using the multi-phonon expansion based on first-principles calculations. The scattering cross section, dominated by the incoherent inelastic contribution from the hydrogen atoms, can be rationalised in terms of the average contributions of different functional groups, thus neglecting their correlation. These results can be used for modelling the total neutron cross sections of complex organic systems like proteins, muscles, or human tissues from a limited number of starting input functions. This simplification is of crucial importance for fine-tuning of transport simulations used in medical applications, including boron neutron capture therapy as well as secondary neutrons-emission induced during proton therapy. Moreover, the parametrized neutron cross sections allow a better treatment of neutron scattering experiments, providing detailed sample self-attenuation corrections for a variety of biological and soft-matter systems.

Example (polystyrene):

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



```
$> ncystal_hfg2ncmat --formula C8H8 \
--spec 5xCharO+1xChAlI+1xCH2 \
--density 0.99 -o PS.ncmat
```

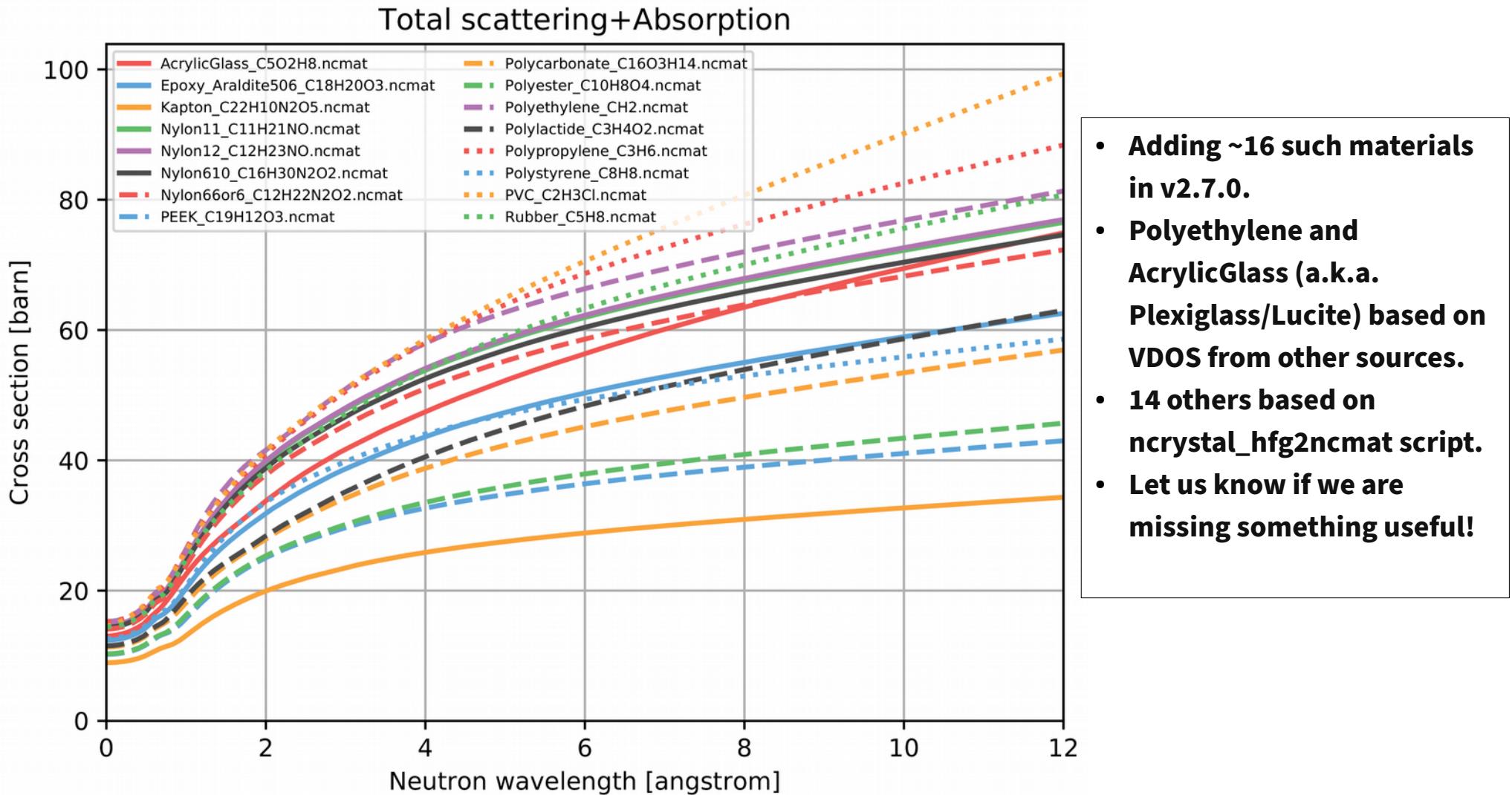


Amorphous materials in data library

New feature in NCystal v2.7.0

NCystal

X. X. Cai & T. Kittelmann



Outlook

(see also <https://github.com/mctools/ncrystal/issues>)

NCrystal

X. X. Cai & T. Kittelmann

- NCrystal v2.7.0 will include many new (relevant & validated) crystalline materials in the data library, prepared by Kemal Ramic, ESS Spallation Group. With new amorphous materials this will triple the number of materials (~50 to >150).
- Other new physics under development:
 - Multiphase support (for complex materials, and for phase-contrast physics like SANS).
 - SANS (Zhiyang Wang & Xiao Xiao Cai, CSNS).
 - S(Q) static structure factor for coherent physics in amorphous materials (TBD).
 - Coherent-single phonon physics (Xiao Xiao Cai, CSNS).
 - Plugin with empirical nanodiamond model (Nicola Rizzi, DTU)
 - More realistic bragg diffraction models, grain size/texture/... (Matteo Busi, PSI)
 - New materials from HighNESS WP2 (ESS, University of Milano-Bicocca)
- Work on new or improved hooks for: Geant4, OpenMC, MCNP, PHITS, ...
- NCrystal-NJOY project (Kemal Ramic and J. Ignacio Marquez Damian) to be published and released to GitHub, along with ENDF (or ACE) files for all files in NCrystal data library. Will not support single-crystal or other new physics.
- Other technical and workflow improvements (move entire development process to GitHub, improve kernel sampling, improve free-gas model, ...).



The end - thank you for your time

Over to Peter for a live demo
of using NCrystal in McStas...



Backup slides



Control DOS→scat. kernel expansion through cfg parameter “vdoslux”

NCrystal
X. X. Cai & T. Kittelmann

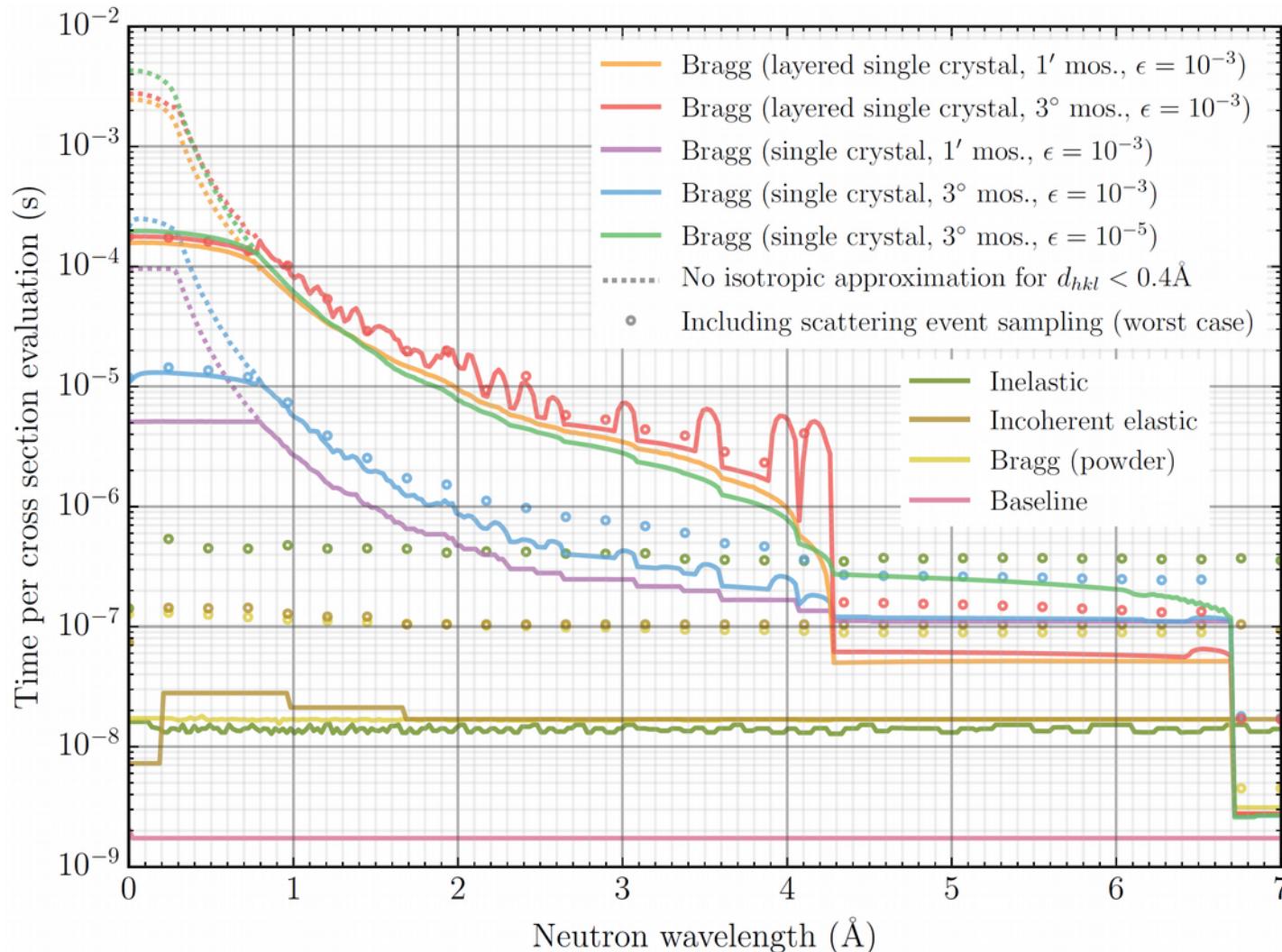
- Controls all aspects of DOS→kernel expansion with only one high level user-accessible parameter.
 - Exposing the underlying multitude of parameters to end-users would do no good in practice since no-one would understand how to change them in a self-consistent way.
- **vdoslux=0:** Extremely crude, 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) ~level of premade ENDF kernels
- **vdoslux=3 :** Good, 800x400 grid, Emax=5eV (costs 8MB mem, 0.2s init time) The default!
- **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) Overkill, exists for validation purpose

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

Speed is important! (at least for many applications)

NCrystal

X. X. Cai & T. Kittelmann



Rough conclusions for MC simulations with thin samples:

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

Have several ideas for additional speedups.
(just needs time to implement...)

Extending NCrystal: New example

Silly goal of example: Replace NCrystal's incoherent-elastic model with different model (isotropic scattering, constant sigma).

```
//////////  
//A "great" physics model (incoherent elastic scattering as isotropic+constantXS):  
class SimpleIncElasScatter : public NCrystal::ScatterIsotropic {  
public:  
    SimpleIncElasScatter( double sigma ) ← Model takes just one  
        : ScatterIsotropic("SimpleIncElasScatter"),           input parameter (sigma)  
        m_sigma(sigma)  
    {  
    }  
    double crossSectionNonOriented( double ekin ) const final  
    {  
        return m_sigma;  
    }  
    void generateScatteringNonOriented( double E, double& theta, double& deltaE ) const final  
    {  
        deltaE = 0;  
        theta = std::acos(getRNG()->generate()*2.0-1.0); //isotropic scatter angle  
    }  
protected:  
    virtual ~SimpleIncElasScatter() = default;  
private:  
    double m_sigma;  
};
```

@CUSTOM sections in NCMAT files

(contents transferred to NCrystal::Info C++ objects)

NCrystal

X. X. Cai & T. Kittelmann

New in v2.1.0

```
NCMAT v3
#Aluminium
#*snip* (some comments with references not shown)
@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
@CUSTOM_SIMPLEINCELAS
    #Activate custom physics model and force incoherent-elastic scattering to be completely isotropic with constant sigma(E)=5barn:
    5.0
@DYNINFO
    element Al
    fraction 1
    type     vdos
    vdos_egrid .0048501456072871 .039822248144042
    vdos_density .013232 .0141193 .0150065 .0158938 .0167811 .0176684
    .0185557 .0195536 .020725 .0218964 .0230678 .0242392 .0254106
    #*snip* (~30 lines not shown)
    .458806 .434668 .410583 .386357 .362131 .337836 .313452 .289068
    .261524 .233749 .20274 .16795 .125961 .0770796 .0135243 0
```

No longer need to hack core
NCrystal code to get input
parameters from the user and
into your C++ class.

Factory code: Teaching NCrystal when/how to use your model

NCrystal

X. X. Cai & T. Kittelmann

```
namespace NC=NCrystal;
class SimpleIncElasScatterFactory : public NC::FactoryBase {
    int canCreateScatter( const NC::MatCfg& cfg ) const final {
        if (cfg.get_incoh_elas()) {
            //user did not disable incoherent-elastic component.
            if /*input has @CUSTOM_SIMPLEINCELAS section*/ {
                return 9999;//high priority => our factory will get selected
            }
            return 0;//indicate that our factory does not want to handle this
        }
        const NC::Scatter * createScatter( const NC::MatCfg& cfg ) const final
        {
            //Dig out sigma from @CUSTOM_SIMPLEINCELAS section:
            // As simple as extracting the value from a
            // vector<vector<string>> with contents [[ "5.0" ]]
            double sigma = parseCustomSimpleIncElas(cfg);

            //Instantiate our "great" physics model:
            auto sc_simpleincelas = new SimpleIncElasScatter(sigma);

            //Combine with other physics (Bragg+inelastic, unless disabled):
            auto cfg2 = cfg.clone();
            cfg2.set_incoh_elas(false);
            auto sc_other( globalCreateScatter(cfg2) );
            auto sc_comp = new NC::ScatterComp;//composition class
            sc_comp->addComponent(sc_other);
            sc_comp->addComponent(sc_simpleincelas);
            return sc_comp;
        }
    };
};
```

This is why we needed a complete example :-)

In real-life models, most lines of code will be spent on the actual physics, not this boiler-plate.

NC::registerFactory(new SimpleIncElasScatterFactory);

Tada! NCrystal is now extended.



```
//Register custom factory with NCrystal:  
NC::registerFactory(new SimpleIncElasScatterFactory);  
  
//Now, NCrystal will create the "great" new physics, in files  
//with the @CUSTOM_SIMPLEINC section:  
auto sc_customAl = NC::createScatter("MyCustomAl.ncmat;temp=350K");  
  
//But create standard physics from other files:  
auto sc_stdAl = NC::createScatter("Al_sg225.ncmat;temp=350K");
```

**Assuming everything is built/linked properly,
NCrystal will now have your new feature.
For usage in C++/Python/McStas/Geant4/...**