EUROPEAN
SPALLATION
SOURCE

Implementation and verification of `ncplugin-MagScat` in `NCrystal`

Work Package 2

S. Xu, S. Laporte, D. Di Julio, J. I. Marquez Damian, T. Kittelmann,
M. Bernasconi, D. Campi, G. Gorini, V. Santoro

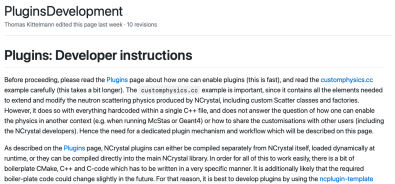
PRESENTED BY S. XU

26-05-2023

Introduction

To include other physics or take into account some specific properties of materials, `NCrystal` [1, 2, 3] supports custom plugin development since version 2.2.0 released in December 2020. It provides a plugin template in which users can implement external physical models and the developed plugin can be compiled directly into the main `NCrystal` library.

- `ncplugin-SANSND` which implements small angles neutron scattering for nanodiamonds (Talk by Nicola)



PluginsDevelopment

Thomas Kittelmann edited this page last week · 10 revisions

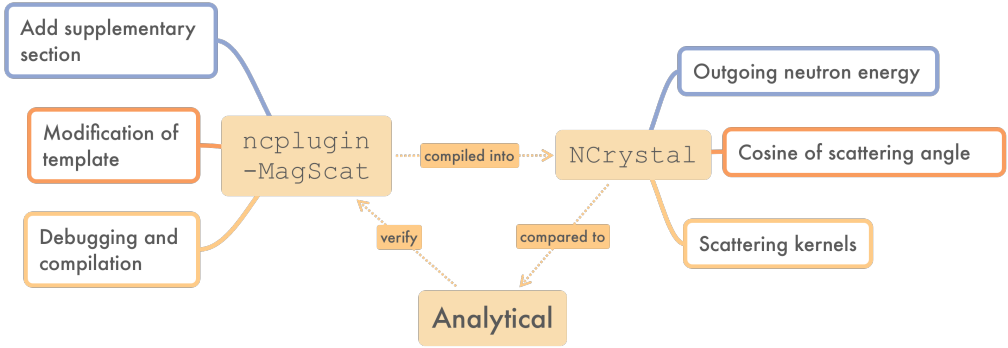
Plugins: Developer instructions

Before proceeding, please read the Plugins page about how one can enable plugins (this is fast), and read the customphysics.cc example carefully (this takes a bit longer). The `customphysics.cc` example is important, since it contains all the elements needed to extend and modify the neutron scattering physics produced by NCrystal, including custom Scatter classes and factories. However, it does so with everything hardcoded within a single C++ file, and does not answer the question of how one can enable the physics in another context (e.g. when running McStas or Geant4) or how to share the customisations with other users (including the NCrystal developers). Hence the need for a dedicated plugin mechanism and workflow which will be described on this page.

As described on the Plugins page, NCrystal plugins can either be compiled separately from NCrystal itself, loaded dynamically at runtime, or they can be compiled directly into the main NCrystal library. In order for all of this to work easily, there is a bit of boilerplate CMake, C++ and C-code which has to be written in a very specific manner. It is additionally likely that the required boiler-plate code could change slightly in the future. For that reason, it is best to develop plugins by using the ncplugin-template

This work was funded by HighNESS project at European Spallation Source ERIC under HORIZON 2020 grant agreement ID: 951782.

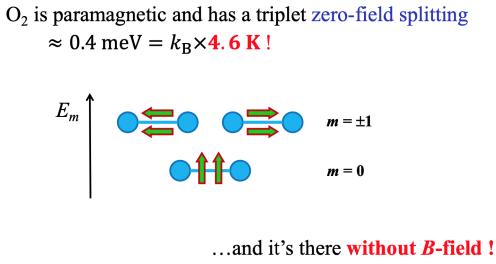
Data flow of implementation and verification



Neutron slowdown by paramagnetic oxygen

Neutron slowdown by paramagnetic oxygen described in Zimmer's paper [4]:

$$\frac{\mathrm{d}^2\sigma_{\mathrm{mag}}}{\mathrm{d}\Omega\,\mathrm{d}E'} = b_{\mathrm{m}}^2 \left(\sqrt{\frac{E'}{E}} S_{\mathrm{mag},\pm}(Q,\omega) + S_{\mathrm{mag},0}(Q,\omega) \right). \quad (1)$$



(screenshot of Zimmer's presentation)

- Molecular oxygen possesses a paramagnetic spin triplet ground state $m = \pm 1, 0$ characterising the spin state projection along the symmetry axis of the molecule
- Transitions of the magnetic levels (spin flip) result in the change of neutron kinetic energy due to the molecular zero-field splitting

Neutron slowdown by paramagnetic oxygen

Based on Zimmer's paper, the neutron magnetic scattering kernels or dynamic structure factors $S_{\mathrm{mag}}(Q,\omega)$ are derived:

$$\textbf{Inelastic: } S_{\mathrm{mag},\pm}(Q,\omega) = \exp\left(-(\langle u^2\rangle + \frac{\ln(2)}{\Gamma_{\mathrm{mag}}^2})Q_{\pm}^2\right) g_{\pm}(T)\delta(\hbar\omega \pm D). \quad (2)$$

$$\textbf{Elastic: } S_{\mathrm{mag},0}(Q,\omega) = \exp\left(-(\langle u^2\rangle + \frac{\ln(2)}{\Gamma_{\mathrm{mag}}^2})Q_0^2\right) g_0(T)\delta(\hbar\omega). \quad (3)$$

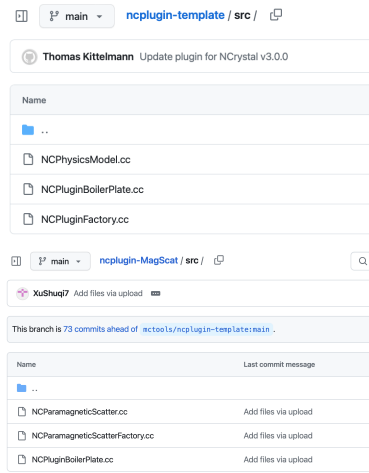
- b_{m}: magnetic scattering length
- Γ_{mag}: half width at half maximum (HWHM) of the magnetic form factor
- D: zero-field splitting constant
- i: integer with 1 for magnetic up-scattering, -1 for down-scattering, and 0 for elastic scattering, omitted for total magnetic scattering

Implementation of `ncplugin-MagScat`

Add a supplementary section:

```
24    #NCRYSTALMATCFG[temp=1K]
25    @CUSTOM_MAGSCAT
26      3.66  1.5  0.4e-3  -1
27    @CELL
28      cubic 6.78
29    @SPACEGROUP
30      223
31    @ATOMPOSITIONS
32      O   0.04360   0.04360   0.04360
33      O   0.03400   0.25160   0.41510
34    @DYNINFO
35      element  O
36      fraction 1.
37      type     vdosdebye
38      debye_temp 104
```

Modify the plugin template:



Compilation and verification

So when configuring NCrystal with CMake, one might specify this like:

```
cmake <other options here> -DNCRYSTAL_BUILTIN_PLUGINS="jchadwick:CoolStuff::deve
```

Verification using built-in modules of `NCrystal` (advanced modules since v3.6, talk by Thomas):

Create magnetic scatters ¶

[3]:
```
T = 2 # K
```

[4]:
```
O2_down = NC.createScatter('ncplugin-MagScat_O2_sg223_SolidOxygen-gamma_mag_down.ncmat;temp={}K;inelas=0;elas=0'.format(T))
O2_up   = NC.createScatter('ncplugin-MagScat_O2_sg223_SolidOxygen-gamma_mag_up.ncmat;temp={}K;inelas=0;elas=0'.format(T))
O2_mela = NC.createScatter('ncplugin-MagScat_O2_sg223_SolidOxygen-gamma_mag_elas.ncmat;temp={}K;inelas=0;elas=0'.format(T))
O2_mag  = NC.createScatter('ncplugin-MagScat_O2_sg223_SolidOxygen-gamma_mag.ncmat;temp={}K;inelas=0;elas=0'.format(T))
O2_info = NC.createInfo('ncplugin-MagScat_O2_sg223_SolidOxygen-gamma.ncmat;temp={}K'.format(T))
```

```
NCrystal::NCMATLoader WARNING: Loading NCMAT data which has @CUSTOM_ section(s). This is OK if intended.
NCrystal::NCMATLoader WARNING: Loading NCMAT data which has @CUSTOM_ section(s). This is OK if intended.
NCrystal::NCMATLoader WARNING: Loading NCMAT data which has @CUSTOM_ section(s). This is OK if intended.
NCrystal::NCMATLoader WARNING: Loading NCMAT data which has @CUSTOM_ section(s). This is OK if intended.
```

Sample isotropic scatters

[5]:
```
N = 10000000 # number of sampled particles
```

[6]:
```
E1 = 1.e-2 # eV
Ep1, mu1 = O2_down.sampleScatterIsotropic(E1, repeat=N)
```

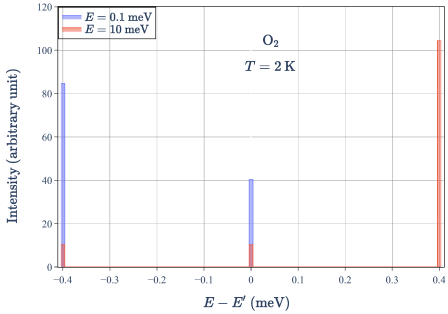
[7]:
```
E2 = 1.e-3 # eV
Ep2, mu2 = O2_up.sampleScatterIsotropic(E2, repeat=N)
```

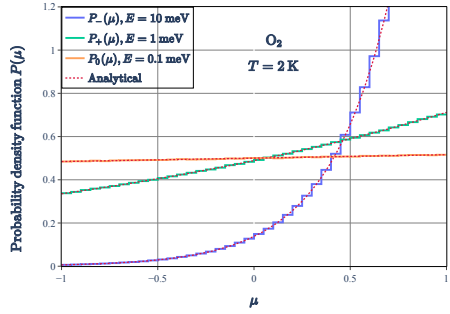
[8]:
```
E3 = 1.e-4 # eV
Ep3, mu3 = O2_mela.sampleScatterIsotropic(E3, repeat=N)
```

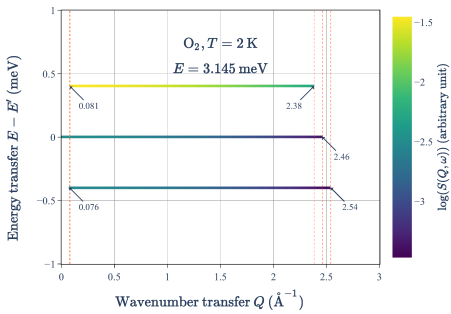
Verification

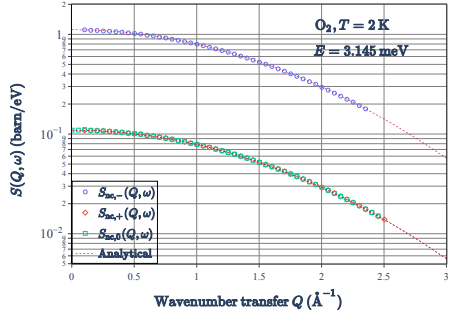
Table: Tests for the verification of `ncplugin-MagScat`.

Test	E (meV)	Magnetic scattering	Number of neutrons	Parameters
1	0.1 10	total	10^7	Intensity
2	0.1 1 10	elastic up- down-	10^7	$P(\mu)$
3	3.145	total	10^8	$S(Q,\omega)$
4	3.145	elastic up- down-	10^8	$S(Q,\omega)$


$E = 0.1$ meV
$E = 10$ meV
O_2
$T = 2$ K
Intensity (arbitrary unit)
$E - E'$ (meV)

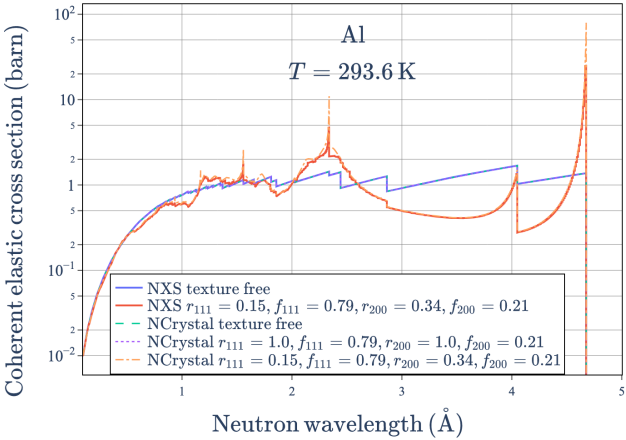

$P_-(\mu), E = 10$ meV
$P_+(\mu), E = 1$ meV
$P_0(\mu), E = 0.1$ meV
Analytical
O_2
$T = 2$ K
Probability density function $P(\mu)$
μ


$O_2, T = 2$ K
$E = 3.145$ meV
0.081
2.38
2.46
0.076
2.54
Energy transfer $E - E'$ (meV)
Wavenumber transfer Q (Å^{-1})
$\log(S(Q,\omega))$ (arbitrary unit)


$O_2, T = 2$ K
$E = 3.145$ meV
$S_{zz,-}(Q,\omega)$
$S_{zz,+}(Q,\omega)$
$S_{zz,0}(Q,\omega)$
Analytical
$S(Q,\omega)$ (barn/eV)
Wavenumber transfer Q (Å^{-1})

Plugin for texture

Implementation of the March-Dollase texture model in a `NCrystal` plugin:



- Good agreement obtained between the developed plugin and the `NXS` code [5]
- Implementation of scattering angle on going

Summary

- `NCrystal` supports custom plugin development with detailed documentation and existing examples.
- The plugin template facilitates the implementation of new physics.
- In this work, we use the built-in modules of `NCrystal` to verify the implementation with JupyterLab. Nevertheless, users can still benefit from the provided `testcode` toolkit for verification.
- Since `NCrystal` can be hooked with many Monte-Carlo neutron transport codes such as `OpenMC`, `McStas`, `CINEMA` and `Geant4`, new physical models can be tested more easily with custom plugins.

Thanks for your time.
Questions?

References

X.-X. Cai, T. Kittelmann, E. Klinkby, and J.I. Márquez Damián.
Rejection-based sampling of inelastic neutron scattering.
Journal of Computational Physics, 380:400–407, 2019.

X.-X. Cai and T. Kittelmann.
Ncrystal: A library for thermal neutron transport.
Computer Physics Communications, 246:106851, 2020.

T. Kittelmann and X.-X. Cai.
Elastic neutron scattering models for ncrystal.
Computer Physics Communications, 267:108082, 2021.

Oliver Zimmer.
Neutron conversion and cascaded cooling in paramagnetic systems for a high-flux source of very cold neutrons.
Phys. Rev. C, 93:035503, Mar 2016.