

# Methods for processing and use of thermal neutron scattering data in OpenMC

Paul K. Romano<sup>1,\*</sup> and Adam G. Nelson<sup>1</sup>

<sup>1</sup>Argonne National Laboratory, 9700 S. Cass Avenue, Lemont, IL 60439, United States

**Abstract.** OpenMC, an open source Monte Carlo particle transport code, relies on its own nuclear data format that is based on the HDF5 file format. The Python API within OpenMC includes an `openmc.data` module, which handles generation of thermal scattering HDF5 libraries starting from either ACE or ENDF files. When starting from ENDF files, NJOY is automatically executed to produce intermediate ACE files. In most cases, the resulting HDF5 data is equivalent to that stored in an ACE file. For incoherent elastic scattering, however, a novel sampling method is derived from the double-differential cross section and can be used to preserve the true continuous angular distribution. In this case, the characteristic bound cross section and Debye–Waller integral are taken directly from an ENDF file. Testing on the H in ZrH evaluation from ENDF/B-VIII.0 has demonstrated how the new sampling method matches the true distribution whereas the traditional treatment based on discrete cosines can leave significant gaps in the angular distribution.

## 1 Introduction

The Monte Carlo (MC) method is commonly used to solve particle transport problems that arise in nuclear reactor physics. For nuclear reactor designs with thermal neutron spectra—which includes most operating nuclear reactors in the world—accurately treating thermal neutron scattering phenomena in transport simulations plays a key role in accurately predicting the neutron flux spectrum, reaction rates, and other observables.

Many production MC codes, such as MCNP [1] and Serpent [2], rely on ACE-format [3] nuclear data libraries for obtaining integrated cross sections, secondary angle/energy distributions, and other data. As a result of this, these codes inherit any limitations that are imposed by the ACE format itself. OpenMC [4], a community-developed open source MC neutron/photon transport code, adds one additional step to the nuclear data processing sequence, wherein ACE libraries are converted to HDF5 files that are then read by OpenMC’s transport solver. The adoption of a different data format was motivated by 1) the need to extend the data library beyond the information available in the ACE format, 2) the ability to differentiate between temperature-dependent and temperature-independent data, which results in significant space savings, and 3) the fact that HDF5 files are cross-platform binary files.

To enable seamless conversion of ACE (and ENDF) files into the HDF5 format required by OpenMC, an `openmc.data` module was added to OpenMC’s Python API. The overall design of the `openmc.data` module was the subject of a previous work [5]. In this paper, we focus on the processing of thermal scattering data for use

in OpenMC and how it allows OpenMC to overcome the limitations imposed by the ACE format.

## 2 Methodology and Results

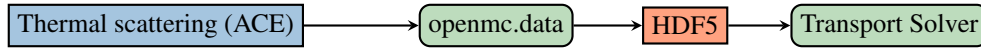
OpenMC is comprised of a neutron/photon transport solver written in C++ and a Python API that is used for programmatic generation of input files, workflow automation, depletion calculations, as well as nuclear data file manipulation and conversion. The `openmc.data` Python module contains all nuclear data-related functionality, including classes for working with ENDF and ACE files as well as a hierarchy of Python classes that store the major categories of data needed by the transport solver. For thermal scattering data, the `openmc.data.ThermalScattering` class sits at the top of the class hierarchy and has methods for reading data from ENDF thermal scattering sublibrary files, reading data from ACE-format thermal scattering files, and writing/reading data to/from HDF5 files.

### 2.1 Data Processing Sequences

If one has an ACE thermal scattering file that has already been generated by NJOY or another code, the processing sequence, illustrated in fig. 1, is straightforward. The `openmc.data` module loads information from the ACE file into the `ThermalScattering` class. This class can then export this data to an HDF5 file, which is eventually loaded by the OpenMC transport solver. In this case, the data stored in the HDF5 file is exactly the same as the data that appears in the original ACE file.

As mentioned earlier, the ACE format itself imposes some limitations on how the underlying data is stored. In

\*e-mail: [promano@anl.gov](mailto:promano@anl.gov)



**Figure 1.** Processing sequence for generating a thermal scattering HDF5 data library starting from a thermal scattering ACE file.

particular, for thermal scattering data, the angle and energy distributions are sometimes stored as a series of discrete values that preserve statistical properties of the true distribution but are otherwise crude approximations, especially for continuous-energy MC transport simulations. MCNP, NJOY, and the ACE format were updated in 2012 [6] to be able to use a continuous representation of incoherent inelastic scattering distributions. Even when thermal scattering ACE files are available, though, it is often desirable to start from a source ENDF file to have more control over the processing sequence; for example, an existing ACE file might have been generated using discrete angles and energies rather than the newer continuous representation. For such cases, the `openmc.data` module can automate execution of NJOY [7] to first produce an ACE file that can then be read by the `ThermalScattering` class; this processing sequence is illustrated fig. 2. Information can also be pulled directly from the ENDF file in order to augment or replace data from the ACE file. This is specifically used for incoherent elastic data, as discussed in section 2.2.

When automating NJOY execution, it is necessary for OpenMC to populate an NJOY input file. Most information needed for the NJOY input can be determined from the ENDF file. However, there are a few exceptions. NJOY requires a list of associated isotopes for a thermal scattering file (`iza` parameter for ACER card 8a) and the number of atom types in a mixed moderator (`nmix` parameter for ACER card 9). In the absence of additional information, these would have to be specified by a user. As a convenience, OpenMC maintains a list of recognized thermal scattering materials that is used to set these parameters when executing NJOY. This list used to rely on the MAT number present in an ENDF thermal scattering sublibrary file. However, as the number of thermal scattering evaluations has grown in recent years, there are now many conflicting MAT numbers, and this method can no longer be relied upon to uniquely determine an evaluation. Instead, OpenMC relies on the `ZSYMAM` field in an ENDF evaluation, which is just a short text description of the material, e.g., “H(H2O)”. This method is more reliable, but it does require some manual intervention by the OpenMC developers whenever a new thermal scattering evaluation appears that hasn’t been handled previously.

## 2.2 Incoherent elastic scattering

NJOY is capable of producing a continuous angular distribution for incoherent inelastic scattering, and for coherent elastic scattering the Bragg edges and structure factors are preserved, which allows for an exact treatment of the secondary angle/energy distribution in OpenMC. However, for incoherent elastic scattering, NJOY still produces an angular distribution that is composed of a series of equally-likely discrete cosines. Without special treatment,

these discrete cosines can produce unphysical artifacts in a transport simulation. Instead of using discrete cosines from an ACE file, OpenMC uses a novel, continuous sampling method based on the differential cross section, which can be expressed analytically as

$$\frac{d\sigma}{dE'd\mu}(E \rightarrow E', \mu, T) = \frac{\sigma_b}{2} e^{-2EW'(T)(1-\mu)} \delta(E - E') \quad (1)$$

where  $\sigma$  is the cross section,  $E$  and  $E'$  are the incident/outgoing energy of the neutron,  $\mu$  is the cosine of the scattering angle,  $\sigma_b$  is the characteristic bound cross section, and  $W'(T)$  is the the Debye–Waller integral as a function of temperature,  $T$ . A probability density function for  $\mu$  can be determined by integrating over  $E'$  and dividing by the total cross section, yielding

$$p(\mu)d\mu = \frac{ce^{-c(1-\mu)}}{1 - e^{-2c}} d\mu \quad (2)$$

where  $c = 2EW'(T)$ . Integrating again, we obtain the following cumulative distribution function:

$$P(\mu) = \int_{-1}^{\mu} d\eta p(\eta) = \frac{e^{c(\mu+1)} - 1}{e^{2c} - 1}. \quad (3)$$

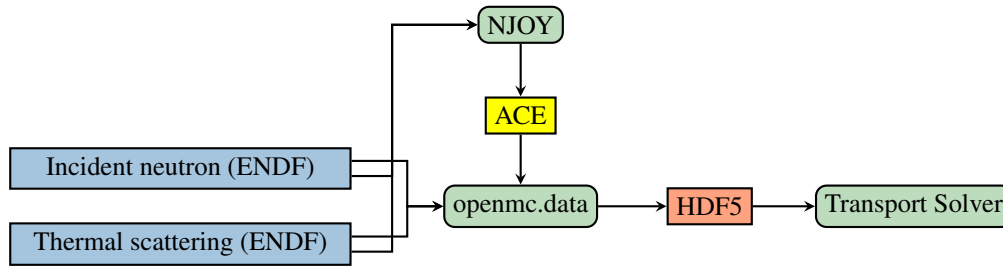
We can then use inverse transform sampling, that is, setting  $P(\mu) = \xi$  where  $\xi$  is a uniformly distributed random number and solving for  $\mu$ , which gives

$$\mu = \frac{1}{c} \log\left(1 + \xi\left(e^{2c} - 1\right)\right) - 1. \quad (4)$$

The sampling method based on eq. (4) requires that  $\sigma_b$  and  $W'(T)$  be stored in the HDF5 data file; to support this, OpenMC pulls the appropriate data directly from the ENDF thermal scattering sublibrary file to augment the ACE data produced by NJOY.

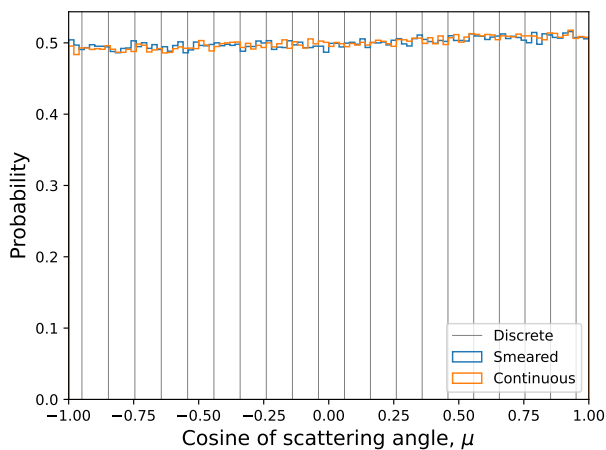
To demonstrate the efficacy of the proposed sampling scheme, the angular distribution for the H in ZrH thermal scattering evaluation from ENDF/B-VIII.0 at 296 K was sampled 1,000,000 times, first using the discrete cosine representation that is stored in an ACE file and then using the sampling scheme from eq. (4). This distribution was sampled for two different incident energies,  $1.25 \times 10^{-3}$  eV and  $1.56 \times 10^{-1}$  eV. Figure 3 shows a histogram of the sampled  $\mu$  values at  $1.25 \times 10^{-3}$  eV. This incident energy is sufficient low that the distribution of  $\mu$  is nearly uniform. Note that when using a discrete cosine representation, OpenMC will “smear” the values between neighboring grid points to produce a smoother distribution<sup>1</sup>, which, at this energy, matches the true uniform distribution quite well. However, at the higher energy of  $1.56 \times 10^{-1}$  eV, the angular distribution is distinctly forward peaked, as shown

<sup>1</sup>This technique is also implemented in MCNP although to the best of our knowledge, it is not documented in any of the MCNP manuals or associated publications.

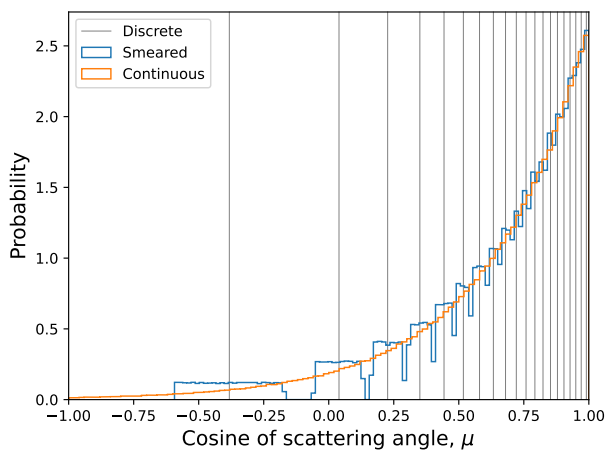


**Figure 2.** Processing sequence for generating a thermal scattering HDF5 data library starting from an ENDF thermal scattering sublibrary file, including automation of NJOY.

in fig. 4. In this case, the discrete cosine representation with smearing fails to reproduce the continuous distribution and has notable gaps over ranges of  $\mu$ . The distribution sampled using eq. (4) preserves the full continuum of  $\mu$  values, including  $\mu$  values close to -1, which represent backscattering.



**Figure 3.** Histogram of sampled  $\mu$  values for H in ZrH at a temperature of 296 K and incident energy  $E=1.25 \times 10^{-3}$  eV



**Figure 4.** Histogram of sampled  $\mu$  values for H in ZrH at a temperature of 296 K and incident energy  $E=1.56 \times 10^{-1}$  eV

### 2.3 Thermal Scattering Classes

The nuclear data class hierarchy in OpenMC (within both the Python and C++ codebase) is designed such that any special classes needed for thermal scattering follow the same interfaces as other, non-thermal-scattering classes. Having a consistent set of interfaces enables maximal code reuse and simplifies the physics implementation. To be more specific, integrated cross sections for thermal scattering reactions are represented as one-dimensional functions that are callable, take  $E$  as an argument, and return the corresponding cross section. Special one-dimensional functions exist for coherent and incoherent elastic scattering that use the minimal set of information in order to calculate a cross section at a given energy. A similar concept applies to angle–energy distributions, which also have a well-defined interface that special thermal scattering classes conform to; in this case, the classes have a `sample` method that takes  $E$  as an argument and returns sampled  $E'$  and  $\mu$  values.

The full list of thermal scattering classes for cross sections and angle–energy distributions is shown in table 1. Most of the classes correspond directly to representations that appear in the ACE format. However, the `IncoherentElasticXS` and `IncoherentElasticAE` classes for incoherent elastic scattering rely on the Debye–Waller integral, which is not present in the ACE format.

### 2.4 Other features

OpenMC has several other attractive features related to thermal scattering. It supports the use of “partial”  $S(\alpha, \beta, T)$  tables, wherein the user can indicate that thermal scattering data should only apply to some fraction of the atoms in a material. This makes it possible to spatially homogenize materials, some of which should use thermal scattering data and some of which should not. For multitemperature problems, stochastic interpolation is performed between temperatures for which thermal scattering data is tabulated, which is the same method used for resolved and unresolved resonance region data. Lastly, OpenMC has recently been updated to support new ENDF-6 formatted evaluations that contain both coherent and incoherent elastic scattering (so-called “mixed” elastic scattering).

**Table 1.** Thermal scattering classes in the `openmc.data` module.

Class	Description	ACE Equivalent
<i>One-dimensional functions</i>		
CoherentElasticXS	Coherent elastic cross section based on Bragg edges and structure factors	✓
IncoherentElasticXS	Incoherent elastic cross section based on characteristic bound cross section and Debye–Waller integral	—
<i>Angle–energy distributions</i>		
CoherentElasticAE	Coherent elastic distribution based on Bragg edges and structure factors	✓
IncoherentElasticAE	Incoherent elastic distribution based on eq. (4)	—
IncoherentElasticAEDiscrete	Incoherent elastic distribution based on discrete cosines	✓
MixedElasticAE	Mixed coherent and incoherent elastic distribution based on any of the above representations	✓
IncoherentInelasticAE	Incoherent inelastic distribution with continuous distribution in energy	✓
IncoherentInelasticAEDiscrete	Incoherent inelastic distribution based on discrete energies and cosines	✓

### 3 Conclusions

In this paper, we have discussed techniques for processing thermal scattering data for use in OpenMC. The `openmc.data` module has processing sequences for generating a thermal scattering HDF5 file starting from a thermal scattering ACE file as well as from a ENDF thermal scattering sublibrary file, in which case NJOY is executed under the hood. In most cases, the HDF5 data is equivalent to that stored in an ACE file. For incoherent elastic scattering, however, a novel sampling method derived from the double-differential cross section can be used to preserve the true continuous angular distribution. In this case, the characteristic bound cross section and Debye–Waller integral are taken directly from an ENDF file. Testing on the H in ZrH evaluation from ENDF/B-VIII.0 demonstrated how the new sampling method matches the true distribution whereas the traditional treatment based on discrete cosines can leave significant gaps in the angular distribution. Lastly, the thermal scattering classes present in OpenMC have been designed to conform to the same interface as other classes in the codebase, maximizing code reuse and simplifying the physics implementation.

### Acknowledgements

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation’s exascale computing imperative.

### References

- [1] C.J. Werner, B.J. S., C.J. Solomon, Jr., F.B. Brown, G.W. Mckinney, M.E. Rising, D.A. Dixon, R.L. Martz, H.G. Hughes, III, L.J. Cox et al., Tech. Rep. LA-UR-18-20808, Los Alamos National Laboratory, Los Alamos, New Mexico (2018)
- [2] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, T. Kaltiaisenaho, Ann. Nucl. Energy **82**, 142 (2015)
- [3] J.L. Conlin, P. Romano, Tech. Rep. LA-UR-19-29016, Los Alamos National Laboratory, Los Alamos, New Mexico (2016)
- [4] P.K. Romano, N.E. Horelik, B.R. Herman, A.G. Nelson, B. Forget, Ann. Nucl. Energy **82**, 90 (2015)
- [5] P.K. Romano, S.M. Harper, EPJ Web. Conf. **146**, 06011 (2017)
- [6] J.L. Conlin, D.K. Parsons, F.B. Brown, R.E. MacFarlane, R.C. Little, M.C. White, Tech. Rep. LA-UR-12-00155, Los Alamos National Laboratory, Los Alamos, New Mexico (2012)
- [7] R. MacFarlane, D.W. Muir, R.M. Boicourt, A.C. Kahler, J.L. Conlin, Tech. Rep. LA-UR-17-20093, Los Alamos National Laboratory, Los Alamos, New Mexico (2017)

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan. <https://energy.gov/downloads/doe-public-access-plan>