Just a little NJOY update

W. Haeck, N. Gibson, M. Staley, C. Josey, A. C. Kahler, J. L. Conlin

November 15-19, 2021
Agenda

1. Getting NJOY2016 ready for ENDF/B-VIII.1
   1. Overview of features to be supported
   2. ACE format changes
   3. NJOY2016.66

2. Modernisation updates
   1. ENDFtk
   2. ACEtk
   3. GNDStk
Making a new ENDF/B library …

• Every new ENDF/B generation changes formats and adds new data

• The future library: ENDF/B-VIII.1 (somewhere in 2024)
  − Mixed mode thermal scattering (coherent and incoherent elastic scattering)
  − Background R-matrix elements for resonance parameters in MF2 MT151
  − Improved photonuclear data

• If these changes impact the ACE format, MCNP needs to be updated too
  − These changes are prioritised due to the involvement of MCNP
  − Changes are made in collaboration with the MCNP development team

• MCNP6.3 will have experimental support for these new ENDF/B-VIII.1 features
Thermal scattering

• Nuclear data evaluations identify multiple categories of thermal scattering:
  − Coherent elastic: important in crystalline solids (graphite, metals, etc)
  − Incoherent elastic: important in solids with hydrogen (polyethylene, ZrH, etc.)
  − Coherent and incoherent inelastic: all solid and liquid materials (hydrogen in water)

• Prior to ENDF/B-VIII.1: either coherent or incoherent elastic scattering
  − Coherent and incoherent are not exclusive and neglecting one is an approximation
  − ENDF/B-VIII.1 will introduce mixed mode elastic scattering

• This feature is reflected in the ACE format itself
  − Only one elastic thermal scattering data block, which is either coherent or incoherent
  − We needed to add an optional second block when both are given
The original thermal scattering format in ACE

- The thermal scattering format is relatively simple
  - Two main blocks: one for inelastic and one for elastic
  - The elastic block is either coherent (IDPNC=4) or incoherent (IDPNC=3)
  - Formatting parameters given in the NXS array

### Parameters

<table>
<thead>
<tr>
<th>Length</th>
<th>IDPNI</th>
<th>NIL</th>
<th>NIEB</th>
<th>IDPNC</th>
<th>NCL</th>
<th>IFENG</th>
</tr>
</thead>
</table>

- **Inelastic**
  - ITIE – Energies
  - ITIX – Cross sections
  - ITXE – Angular data

- **Elastic**
  - (coherent or incoherent)
  - ITCE – Energies
  - ITCX – Cross sections
  - ITCA – Angular data

-1 for coherent elastic

Not present for coherent elastic
The original thermal scattering format in ACE

- The old format is still VALID
  - When mixed mode is used, there will be an additional elastic block (IDPNC=5)
  - Coherent elastic is always given first, incoherent elastic is given after that
  - An additional formatting parameter: NCLI for the second elastic block only
    - NCL will always be -1 for IDPNC=5

<table>
<thead>
<tr>
<th>Length</th>
<th>IDPNI</th>
<th>NIL</th>
<th>NIEB</th>
<th>IDPNC</th>
<th>NCL = -1</th>
<th>IFENG</th>
<th>NCLI</th>
</tr>
</thead>
</table>

- Inelastic
  - ITIE – Energies
  - ITIX – Cross sections
  - ITXE – Angular data

- Coherent elastic
  - ITCE – Energies
  - ITCX – Cross sections
  - ITCA – Angular data

- Incoherent elastic
  - ITCEI – Energies
  - ITCXI – Cross sections
  - ITCAI – Angular data
Example: H-H2O - inelastic only

H IN H2O @400K FROM ENDF/B-VIII.0 - CHECK TAPE 1
Thermal cross sections

Cross section (barns) vs. Energy (MeV)

- inelastic
Example: Al27 metal – coherent elastic and inelastic
Example: Zr-ZrH – incoherent elastic and inelastic

![Graph showing thermal cross sections for Zr in ZrH at 400K from ENDF/B-VIII.0 - Check Tape 1. The graph displays cross sections in barns as a function of energy in MeV. The graph includes lines for inelastic, incoherent elastic, and total cross sections.](image-url)
Example: D-$^7$LiD – mixed mode elastic and inelastic
Photonuclear data

• Traditional photonuclear data
  – Secondary photon distributions traditionally given using the LAW=1 LANG=1 format
  – Traditionally using a single Legendre coefficient (i.e. isotropic distribution)
  – This assumption was hardcoded in NJOY2016’s ACER module

• And then the IAEA-2019 library was released (August 2020)
  – Secondary distributions are using anisotropic Legendre expansion

• NJOY2016 had to be updated
  – A temporary fix was introduced to keep the distributions isotropic
  – A permanent fix now translates the distributions properly into ACE LAW=61
  – Only MCNP6.3 is capable of using these new photonuclear files
Example: Mono-energetic photon beam on a Pu239 disk

- Neutron spectrum tallied outside the disk in a 0.1 mm sphere
  - Using ENDF/B-VIII.0 photonuclear data
Example: Mono-energetic photon beam on a Pu239 disk

- Comparing ENDF/B-VIII.0 with IAEA-2019
  - Only MCNP6.3 can run the IAEA-2019 ACE files
This is a big update, the major changes are:

- The photonuclear data format changes (ACE LAW 44 to 61)
- Mixed mode elastic scattering in thermal scattering laws

**ACER**

- Photonuclear ACE format update (including plots and output file)
- Formatting/processing the thermal scattering data (including plots and output file)
- The XSS array and its size is now set in the common acecm module
- Added locator checking and unknown law checking when writing out ACE files
  - NJOY2016 will now error out when locators are inconsistent or when an unknown law is used
  - Previously only for incident neutron and charged particle CE files
  - Extended to photonuclear and thermal scattering files
- Some changes in the input file for mixed mode elastic (card 9: ielas=2)
- Charged particle updates

When this happens, there is an issue that can cause MCNP problems. These things previously went undetected.
NJOY2016.66

• MODER
  - MF7 and MF28 update

• THERMR
  - Most of the thermal scattering processing happens here
  - Between 1 and 3 thermal scattering MT numbers are added to the PENDF file

• ERROR
  - Fixed a crash caused by MF34 covariances using multiple subsections
  - Each subsection is now processed but we are still working on outputting the results

• Additional non-regression tests
  - There are now 71 cases in the NJOY2016 test suite
  - Tests 67-70 provide tests for all combinations of thermal scattering data
What does the future bring?

• NJOY2016 will be maintained for the foreseeable future
  – NJOY2016 is essentially the production code at LANL
  – New formats for ENDF/B-VIII.1 will be supported:
    ▪ Thermal scattering: mixed coherent and incoherent elastic scattering
    ▪ External R-matrix elements used in some new resonance evaluations

• NJOY21: shift from a module based to a component based modernisation
  – Modernised modules are built from components
    ▪ Components provide formats (ENDF, ACE) or processing operations (resonance reconstruction)
    ▪ Components can be developed and deployed faster than modules
    ▪ Components provide features that modules do not provide
  – Using a C++ and Python API at the same time
Processing components are format agnostic

• In the beginning there was only ENDF …
  – As a result, NJOY2016 is very closely linked to ENDF
  – Introducing the new GNDS format in NJOY2016 is practically impossible

• NJOY21 processing components MUST be format agnostic
  – Internal data structures that reflect generic data can be built from scratch
  – Build these data structures using ENDF or GNDS evaluated data, or other user data
### NJOY21 formatting components

<table>
<thead>
<tr>
<th>NJOY21 format components</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ENDFtk</td>
<td>Evaluated nuclear data format (the legacy one)</td>
</tr>
<tr>
<td>GNDStk</td>
<td>Evaluated nuclear data format (the new one)</td>
</tr>
<tr>
<td>ACEtk</td>
<td>Application library format for MCNP</td>
</tr>
</tbody>
</table>

- **ENDFtk**: almost everything in the ENDF format, including internal NJOY sections
  - Some of the covariance sections are still missing
  - GENDF support (for GROUPR and ERRORR) will be added soon
  - [https://github.com/njoy/ENDFtk](https://github.com/njoy/ENDFtk)

- **ACEtk and GNDStk**: will be our focus for FY22
  - [https://github.com/njoy/ACEtk](https://github.com/njoy/ACEtk)
  - [https://github.com/njoy/GNDStk](https://github.com/njoy/GNDStk)
## Current status of ACEtk

### Incident neutron and charged particle continuous energy files

<table>
<thead>
<tr>
<th>Block</th>
<th>Description</th>
<th>C++</th>
<th>Python</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESZ</td>
<td>Energy grid and principal cross sections (total, elastic, absorption, heating)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NU</td>
<td>Average number of neutrons per fission</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MTR, MTRP, MTRH</td>
<td>Available reactions (excluding elastic)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>LQR</td>
<td>Reaction Q values</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>TYR, TYRH</td>
<td>Reference frame and multiplicity</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SIG</td>
<td>Cross section data</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SIGP, SIGH</td>
<td>Particle production cross sections</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>AND, ANDP, ANDH</td>
<td>Angular distribution data for secondary particles (no correlated angular)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DLW, DLWP, DLWH</td>
<td>Secondary particle energy distribution data (includes correlated angular)</td>
<td>Yes/No</td>
<td>Yes/No</td>
</tr>
<tr>
<td>GPD, HPD, YP, YH</td>
<td>Total secondary particle production cross section and multiplicities</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PTYPE, NTRO, IXS</td>
<td>Auxiliary arrays for secondary particle production</td>
<td>Yes/No</td>
<td>Yes/No</td>
</tr>
<tr>
<td>UNR</td>
<td>Unresolved resonance probability tables</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Current status of GNDStk

• The GNDStk development approach
  – A tree/node based core interface that is GNDS standard agnostic
  – A GNDS standard interface layer
    ▪ An autogenerated interface that follows the GNDS standard specifications
    ▪ Both a C++ and python interface is generated
    ▪ Allowing for customization to produce a slimmed down and abstracted interface

• GNDStk v0.1.0 is a prototype using a slimmed down GNDS 1.9 standard

• We’re in the process of implementing the GNDS 2.0 standard
  – The autogenerated interface should be ready by now (at least I hope it is)
  – Customisation will begin for resonance parameters and cross section data
import ENDFtk, ACEtk
import matplotlib.pyplot as plot

# open an Pu239 ENDF file and extract the total cross section
tape = ENDFtk.tree.Tape.from_file( 'U235.endf' )
section = tape.materials.front().file( 3 ).section( 1 ).parse()
energies1 = section.energies
total = section.cross_sections

# open the associated Pu239 ACE file and extract the total cross section
ace = ACEtk.ContinuousEnergyTable.from_file( 'U235.ace' )
index = ace.MTR.index( 18 )
energies2 = [ energy * 1e+6 for energy in ace.ESZ.energies ]
fission = ace.SIG.cross_sections( index )

# plot the cross sections
plot.plot( energies1, total )
plot.plot( energies2, fission )
plot.xscale( 'log' )
plot.yscale( 'log' )
plot.xlabel( 'Incident neutron energy [eV]' )
plot.ylabel( 'Cross section [barn]' )
plot.show()
A first application: plotting
Conclusions

• NJOY2016 will be maintained for the foreseeable future
  – New formats for ENDF/B-VIII.1 will be supported
  – NJOY2016.66 already implements a few of these
  – Experimental support for the ACE format changes in MCNP6.3

• NJOY21 modernisation
  – Component based versus module based
  – C++ and python interfaces
  – Currently focus is on format components: ENDFtk, ACEtk, GNDStk