NJOY work in FY21

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

February 15, 2022

Table of Contents

Our priorities for the future

- 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 ORNL evaluations
    - New photonuclear data
- Deliver useable NJOY21 components sooner rather than later
  - Modernised modules are built from components
    - Components provide formats (ENDFtk) and processing operations (resonance reconstruction)
    - Modules provide current NJOY capabilities: RECONR, BROADR, etc.
    - Components can be developed faster than modules
  - Using a C++ and Python API to allow for better interaction
  - Regular releases with testing and validation

NJOY2016 in FY21

- Mixed mode thermal scattering
- New photonuclear data
- NJOY2016 updates

Thermal scattering

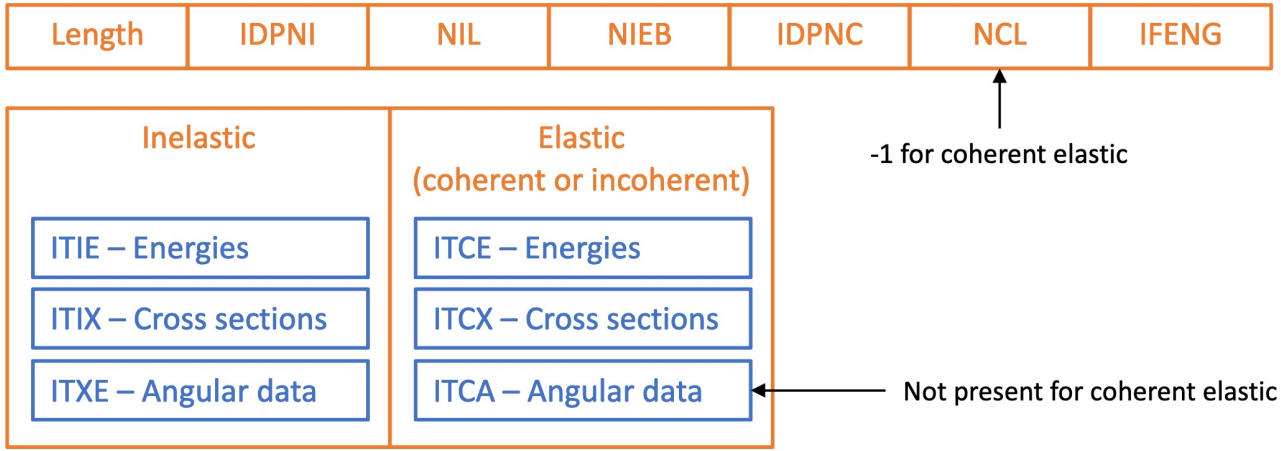
- Nuclear data evaluations identify multiple categories of thermal scattering:
  - Coherent elastic: important in crystalline solids (graphite, metals, etc)
  - Incoherent elastic: important in solids with hydrogen (polyethylene, ZrH, etc.)
  - Coherent and incoherent inelastic: all solid and liquid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



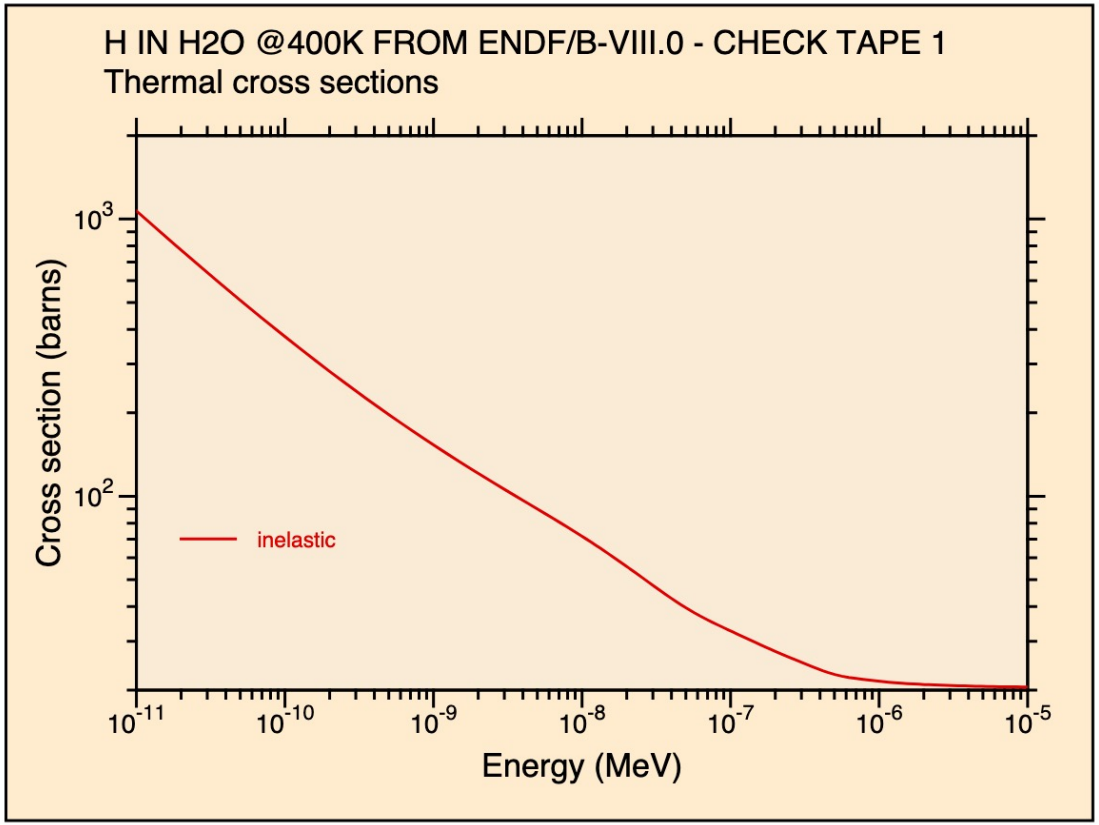
The new thermal scattering format in ACE

- 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

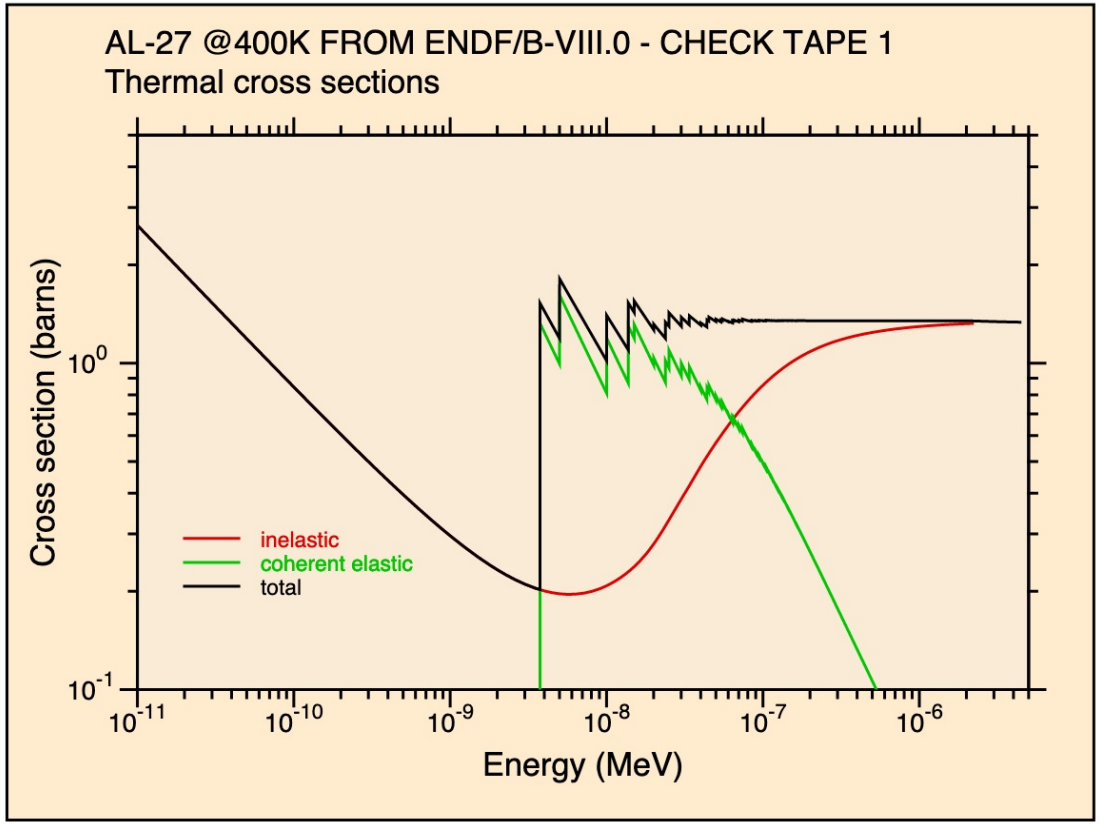
Length	IDPNI	NIL	NIEB	IDPNC	NCL = -1	IFENG	NCLI

Inelastic	Coherent elastic	Incoherent elastic
ITIE – Energies	ITCE – Energies	ITCEI – Energies
ITIX – Cross sections	ITCX – Cross sections	ITCXI – Cross sections
ITXE – Angular data	ITCA – Angular data	ITCAI – Angular data

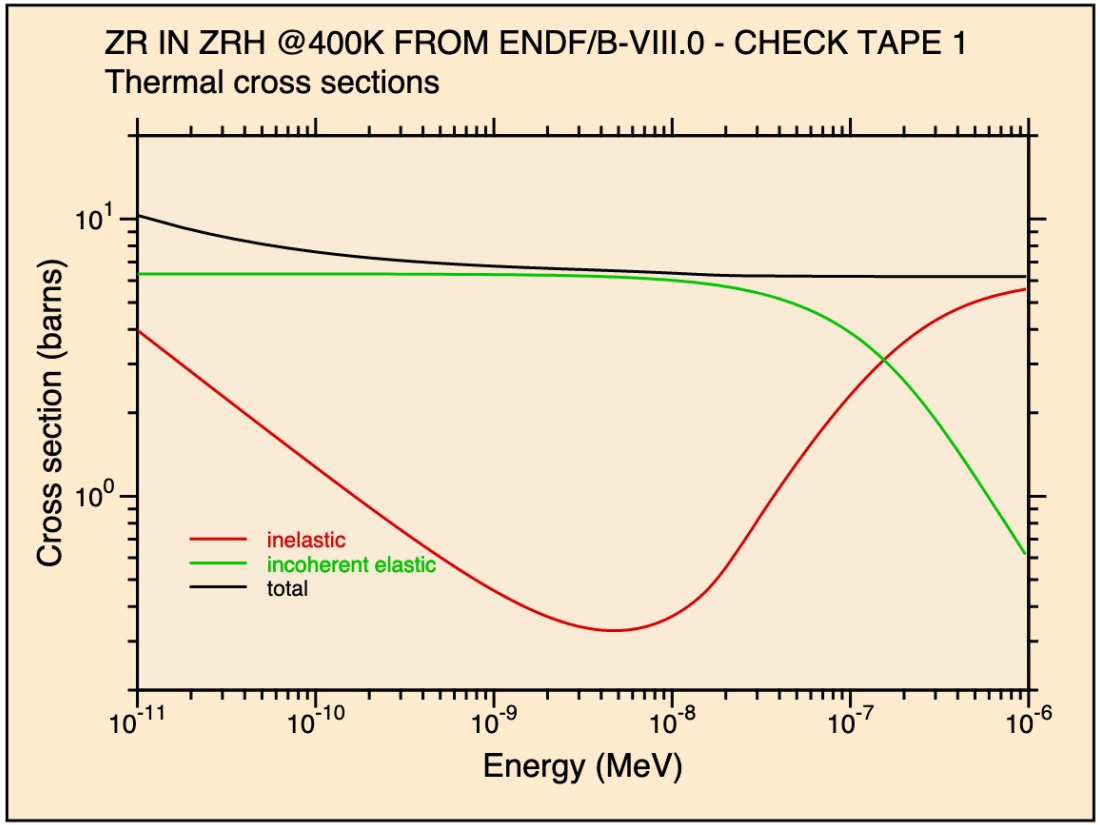
Example: H-H2O - inelastic only



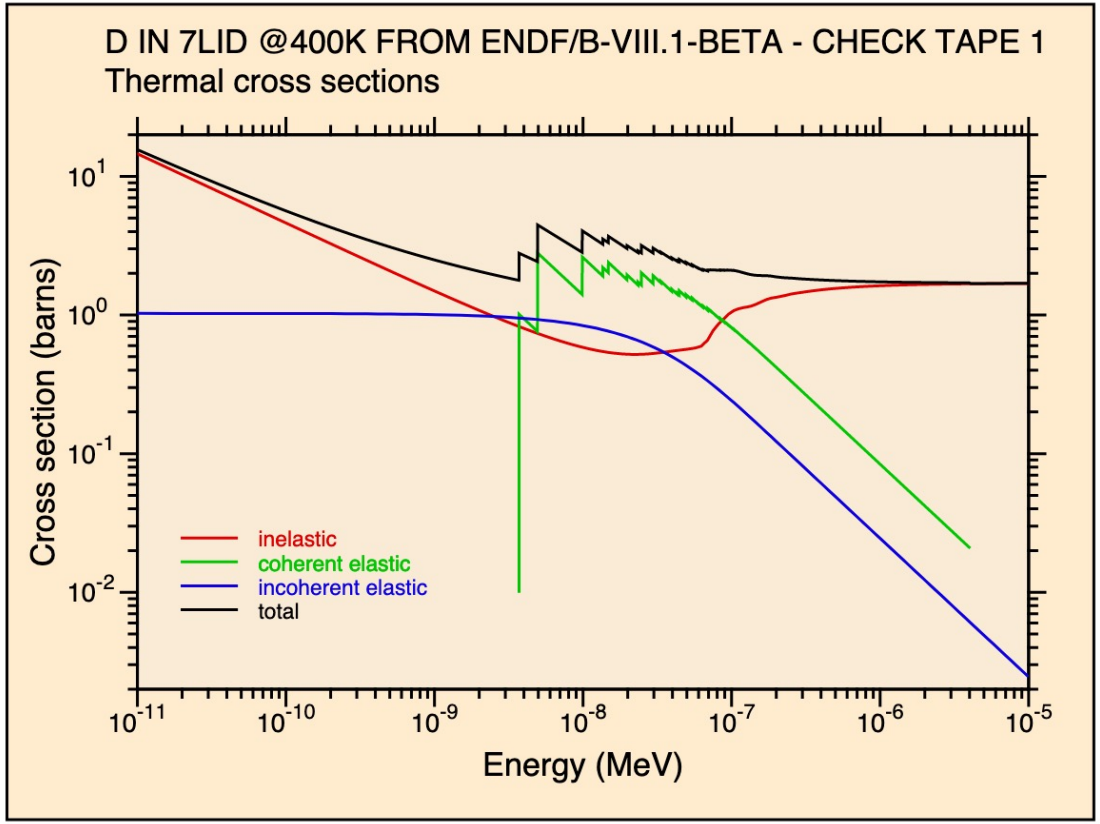
Example: Al27 metal – coherent elastic and inelastic



Example: Zr-ZrH – incoherent elastic and inelastic



Example: D-7LiD – mixed mode elastic and inelastic



Photonuclear data

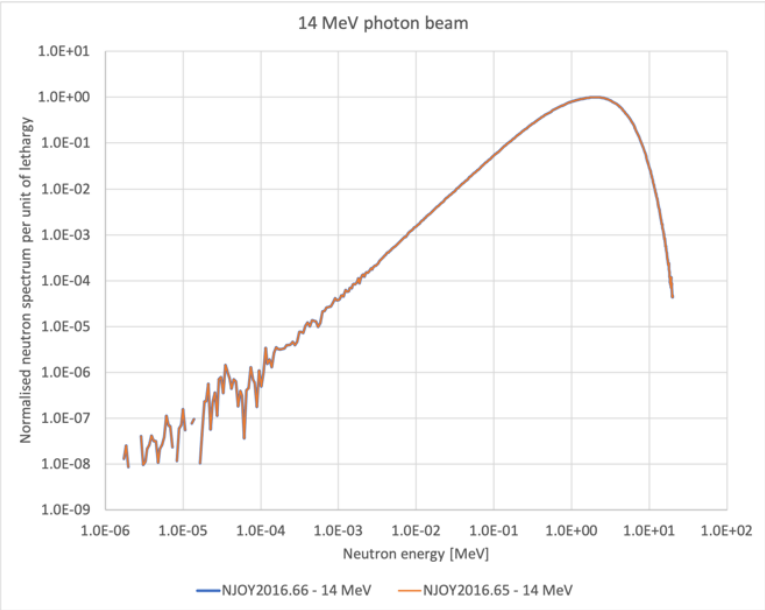
- Traditional photonuclear data
  - Secondary photon distributions traditionally given using the LAW=1 LANG=1 format
  - Traditionally using a single Legendre coefficient (i.e. isotropic distribution)
  - This assumption was hardcoded in NJOY2016's ACER module

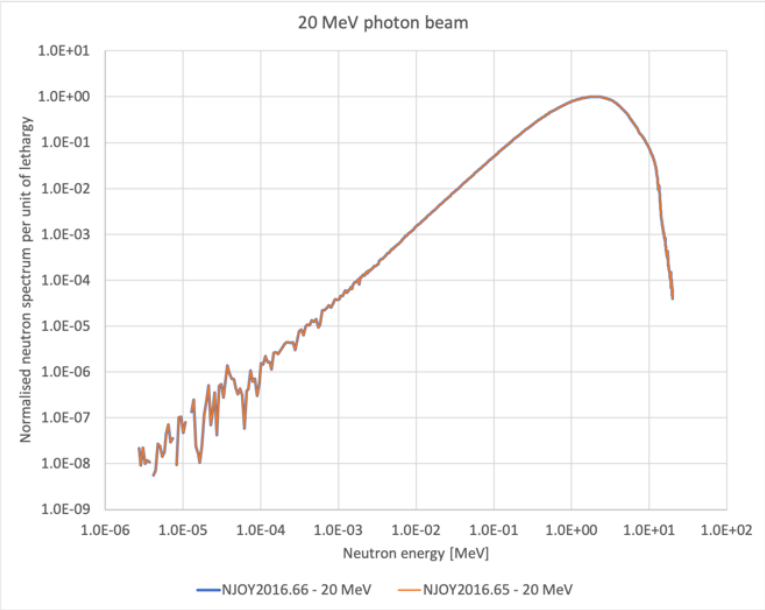
- And then the IAEA-2019 library was released (August 2020)
  - Secondary distributions are using anisotropic Legendre expansion

- 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

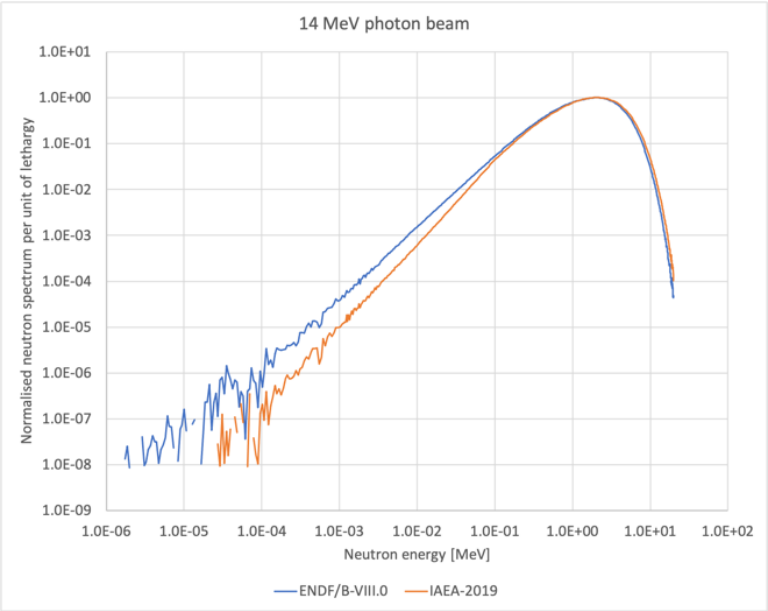
- Neutron spectrum tallied outside the disk in a 0.1 mm sphere
  - Using ENDF/B-VIII.0 photonuclear data

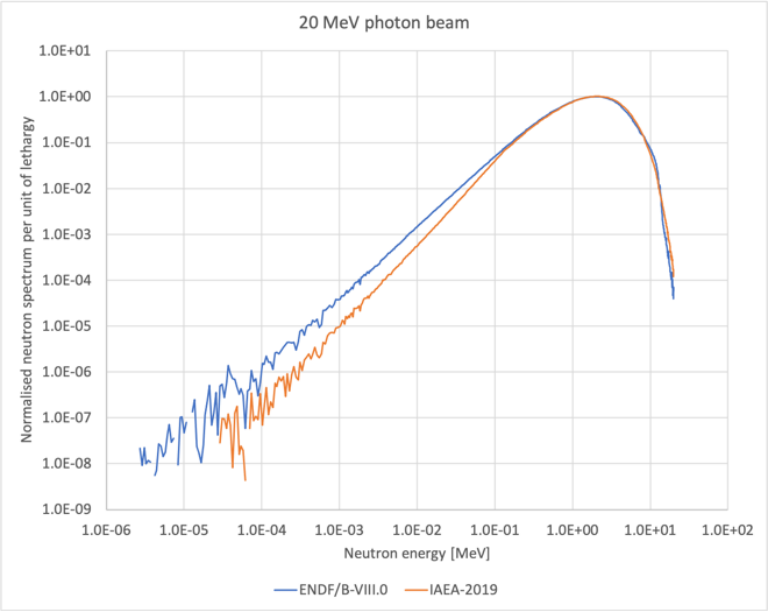




Example: Mono-energetic photon beam on a Pu239 disk

- Comparing ENDF/B-VIII.0 with IAEA-2019
  - Only MCNP6.3 can run the IAEA-2019 ACE files





NJOY2016 updates

- 5 NJOY2016 updates were released (updates 2016.60 to 2016.64)
  - Minor updates to fix specific bugs
  - Updating the physical constants to be in line with the updated ENDF manual
- Major work on thermal scattering and photonuclear data to be released
  - NJOY2016.66 update is undergoing QA review
  - All work performed in FY21 but released in FY22

- Other noteworthy updates in NJOY2016.66
  - ACER will error out when locators are inconsistent or when an unknown law is used
    - For incident neutron and charged particle CE files, photonuclear and thermal scattering files
    - For example: the original ENDF/B-VIII.0 B10 evaluation
  - ERRORR now processes multiple subsections in MF34 (not in output yet though)
  - The NJOY2016 test suite now has 71 cases in it

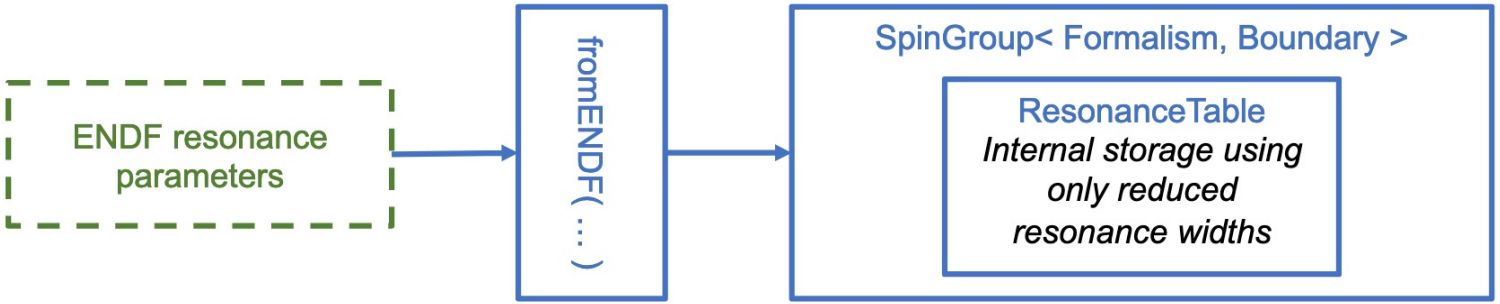
NJOY modernisation work in FY21

- Format and processing components
- ENDFtk, ACEtk and GNDStk

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 format components: ENDFtk

- FY21 work focused on completing this component as much as possible

- ENDFtk v0.3.0 release in July 2021
  - Added support for isotope production (MF9 and MF10)
  - Added support for secondary photon data (MF12 to MF15)
  - First NJOY21 component to come with a consistent C++ AND python interface
- ENDFtk development version at the end of FY21
  - Added support for photo-atomic and atomic relaxation (MF23 to MF28)
  - Added support for covariance data (MF33 and MF34)
  - Inserting, replacing and removing pieces from an ENDF tape
- Remaining pieces to be implemented in FY22
  - MF32 and MF35, some internal NJOY ENDF components (e.g. GEND files)

NJOY21 format components: ACEtk

- Development work on ACEtk was restarted near the end of FY21

- Read, write, modify and create ACE tables
  - We want to be able to extract pieces from the ACE table and insert it elsewhere
  - Locators get handled under the hood without user intervention
  - Simple interface that does not require (too much) knowledge of the ACE format

- ACEtk development version at the end of FY21
  - Limited continuous energy capability (cross sections and angular distributions)
  - A consistent C++ and python interface built upon ENDFtk experience
- ACEtk development will be one of our focus points in FY22

NJOY21 format components: 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 release in July 2021
  - A prototype using an extremely slimmed down GNDS 1.9 standard

- We want to deliver the GNDS 2.0 standard interface in FY22

A first application: plotting

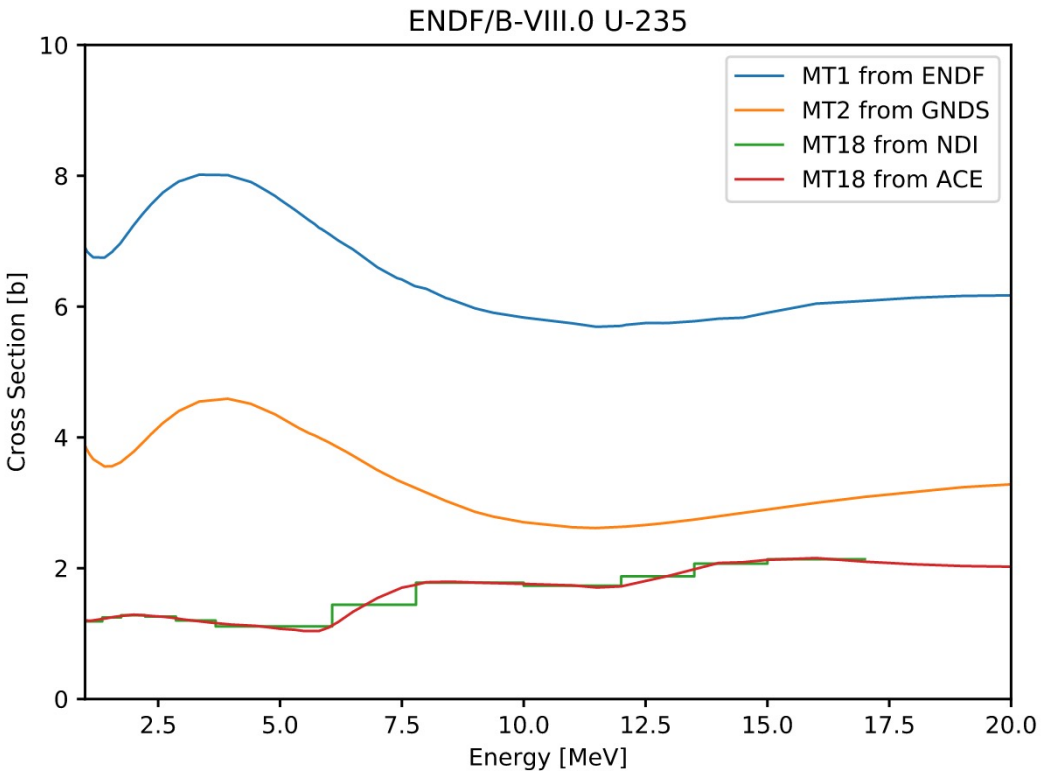
```
# import the required components
import ENDFtk, ACEtk
import matplotlib.pyplot as plot

# 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



Acknowledgements

- This work was supported by the Nuclear Criticality Safety Program, funded and managed by the National Nuclear Security Administration for the Department of Energy.