Los Alamos NATIONAL LABORATORY

NJOY's road toward GNDS

W. Haeck, N. Gibson, M. Staley

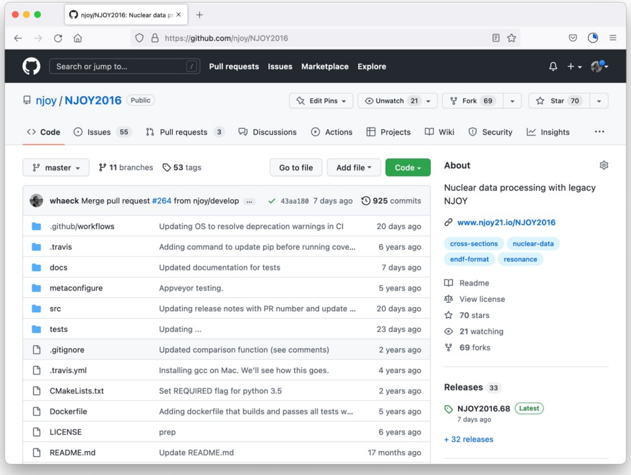
May 16, 2023

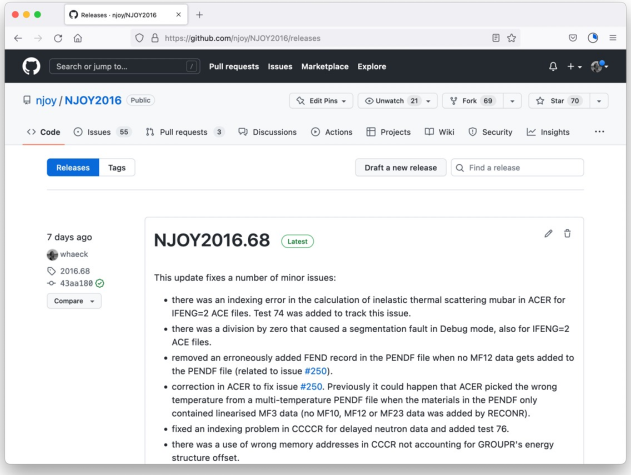
Agenda

1. A quick NJOY2016 update
2. The NJOY modernisation progress
3. GNDStk update

Maintaining our production version

- Get it at https://github.com/njoy/NJOY2016





- Latest version is NJOY2016.70 (April 2023)
  - We aim to release updates every three months – even if the changes are minor
  - This coincides with quarterly reports that we give to our funding sources

Our main objective: smooth processing of ENDF/B-VIII.1

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

- The future library: ENDF/B-VIII.1
  - Mixed mode thermal scattering (coherent and incoherent elastic scattering)
  - Improved photonuclear data
  - Background R-matrix elements for resonance parameters in MF2 MT151
  - General R-matrix formalism (KRM = 4) in MF2 MT151

- Caveat: if these 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

Overview of some of the latest NJOY2016 changes

- ACER processing using multi-temperature PENDF files (NJOY2016.68)
  - PENDF files could contain multiple FEND records due to incorrect logic in RECONR
  - ACER sometimes picked the wrong temperature from a multi-temperature PENDF file
    - Only happens when the materials in the PENDF contained only linearised MF3 data (no MF10, MF12 or MF23 data was added by RECONR).

- Bondarenko cross sections from probability tables (NJOY2016.69)
  - The NJOY output file provides Bondarenko data calculated during the sampling process and calculated from the final probability tables
  - For low dilution values, the Bondarenko data calculated from the final probability tables are closer to the results from UNRESR so those are now written to MF2 MT152
  - Note: deterministic data processing should use UNRESR

Overview of some of the latest NJOY2016 changes

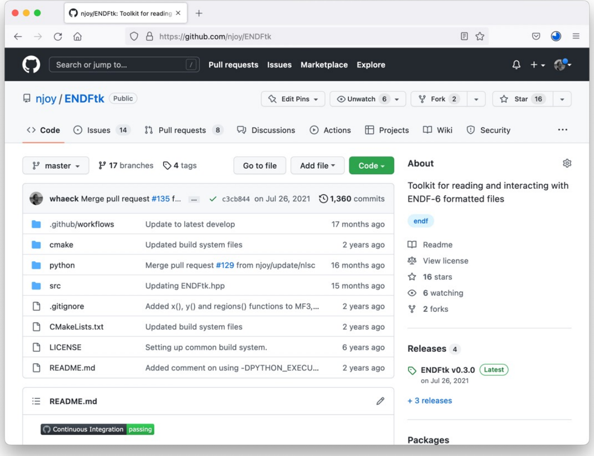
- Heating values using MF6 LAW2 in MT5 (NJOY2016.69)
  - ACER assumed that the yield of the secondary particle is 1, which is correct in all cases except when MT5 is used as a lumped reaction

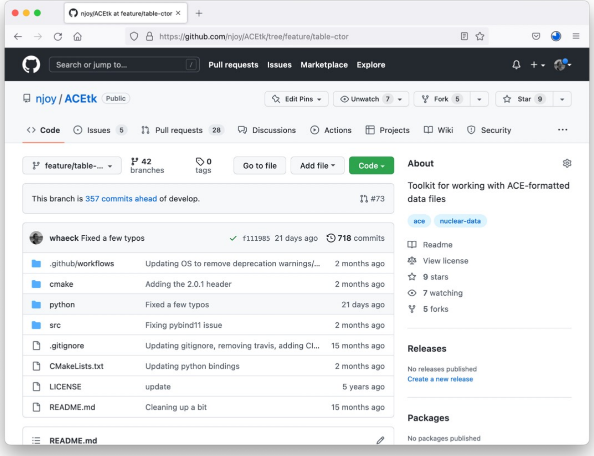
- Segmentation fault in ERRORR (NJOY2016.69)
  - Some ENDF/B-VIII.0 and ENDF/B-VIII.1 evaluations failed in ERRORR
  - ERRORR would segfault for LRF=7 resonance evaluations when MF33 was present without MF32

- Multiple calls to ERRORR for MF34 and MF35 data now work (NJOY2016.70)
  - ERRORR needs to be called for each sub-subsection and incident energy group
  - The issue was related to arrays being allocated but not deallocated in the previous ERRORR run in NJOY's Sammy routines for evaluations using MF2 LRF=7.

NJOY modernisation strategy

- NJOY21: shift from a module based to a component based modernisation
  - Modernised modules are built from components
    - Components provide formats (e.g. ENDF, ACE, GNDS) or processing operations (e.g. scion)
    - Components can be developed and deployed faster than modules
  - Using a C++ and Python API at the same time
  - Regular releases with testing and validation





ENDFtk and ACEtk development is almost completed

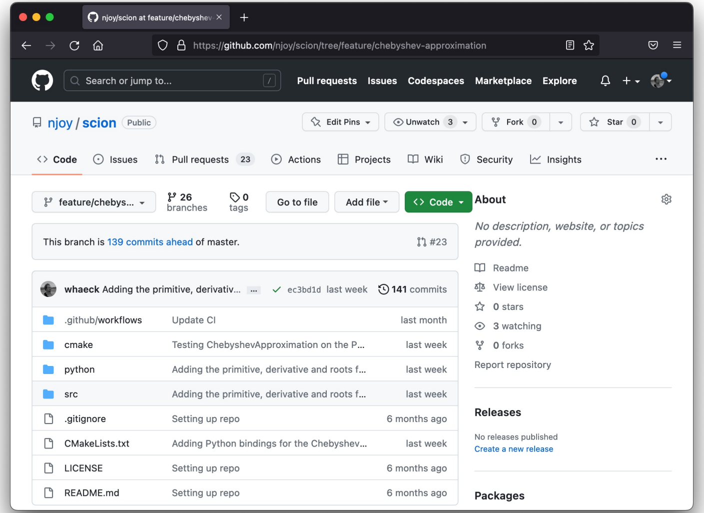
- ENDFtk: https://github.com/njoy/ENDFtk
  - We now have full support for all data, including covariance data
  - We have added functionality for manipulating ENDF files
    - Inserting, replacing and removing materials, files and sections
    - Updating the directory of the ENDF file
- ACEtk: https://github.com/njoy/ACEtk
  - We now have full support for the following ACE file types:
    - Incident neutron and charged particle ACE files
    - Photoatomic and photonuclear ACE files
    - Thermal scattering ACE files
    - Dosimetry ACE files
- Look out for v1.0 releases of both toolkits soon!

So we can read and write data, now what?

- Most NJOY modules need to perform a common set of operations:
  - Interpretation of various data representations (tables, analytical functions, etc.)
  - Linearisation of various data representations
  - Unionisation of data on a common energy grid, etc.
  - Differentiation and integration of the data

- This will be the job of SCION
  - SCientific interpretatION, linearisatION, differentiatION, integratION and more IONs
  - It will provide a format agnostic data interface

This one's important, more on this later



Current capabilities in SCION

- Functional interpretation of tabulated data using various interpolation schemes
- Functional interpretation of polynomial based expansions
  - Normal power series, Legendre series and Chebyshev series
  - Root finding for the general case f(x) = a using the companion matrix
  - Integration and differentiation can be performed using a functional interface

- Generic linearisation of functions
  - Extensible interface for convergence and panel splitting
- Common mathematical capabilities
  - Horner and Clenshaw recursion for polynomial evaluation
  - Newton-Raphson for root finding
  - Special mathematical functions

Current capabilities in SCION

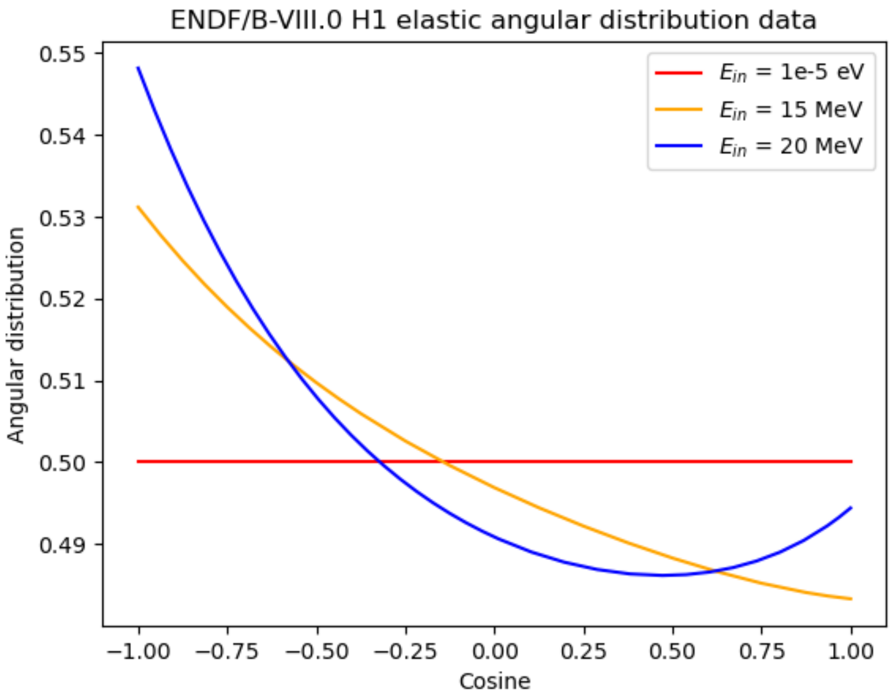
```
import scion
import ENDFtk
import matplotlib.pyplot as plot

# convert to the proper Legendre coefficients
def convert( data, index ) :
    coefficients = data.angular_distributions[index].coefficients
    converted = [ 0.5 ]
    for i in range( 1, len( coefficients ) + 1 ) :
        converted.append( ( 2 * i + 1 ) / 2 * coefficients[i - 1] )
    return converted

# open the H1 data and extract the elastic angular distribution data
u235 = ENDFtk.tree.Tape.from_file( 'h1.endf' )
elastic = u235.materials.front().file( 4 ).section( 2 ).parse()

# retrieve the angular distributions at different energies
d1 = scion.math.LegendreSeries( convert( elastic, 0 ) ).linearise()
d2 = scion.math.LegendreSeries( convert( elastic, -11 ) ).linearise()
d3 = scion.math.LegendreSeries( convert( elastic, -1 ) ).linearise()

# plot the data
plot.figure()
plot.plot( d1.x, d1.y, label = '$E_{in}$ = 1e-5 eV', color = 'red' )
plot.plot( d2.x, d1.y, label = '$E_{in}$ = 15 MeV', color = 'orange' )
plot.plot( d3.x, d2.y, label = '$E_{in}$ = 20 MeV', color = 'blue' )
plot.xlabel( 'Cosine' )
plot.ylabel( 'Angular distribution' )
plot.show()
```



A format agnostic nuclear data interface in SCION

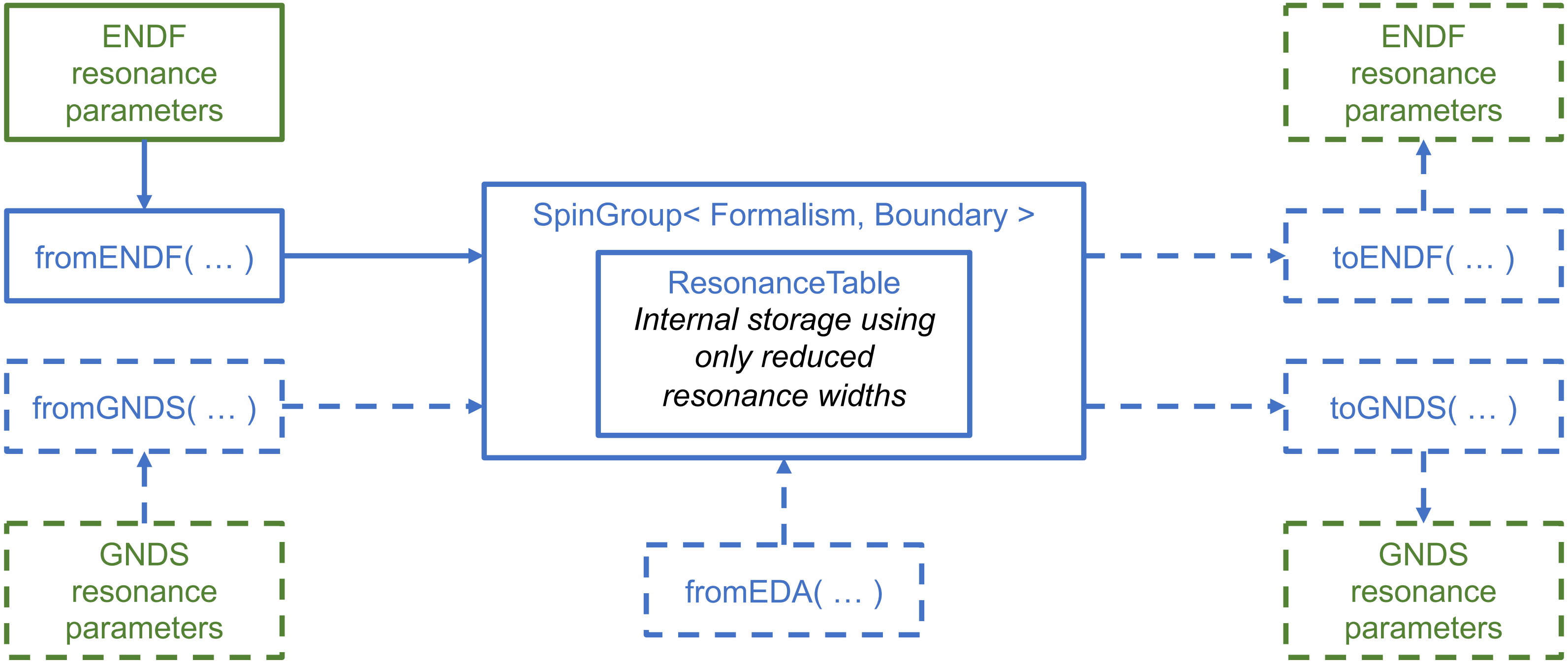
- A nuclear data user should not have to worry about format details
- For example:
  - ENDF MF4 Legendre data does not include the order 0 coefficient (equal to 1/2)
  - ENDF MF6 LAW1 Legendre data includes the order 0 coefficient but is not normalised

- How to achieve format abstraction?
  - Solution 1: there can be only ONE!
    - Translate every format to the one you use

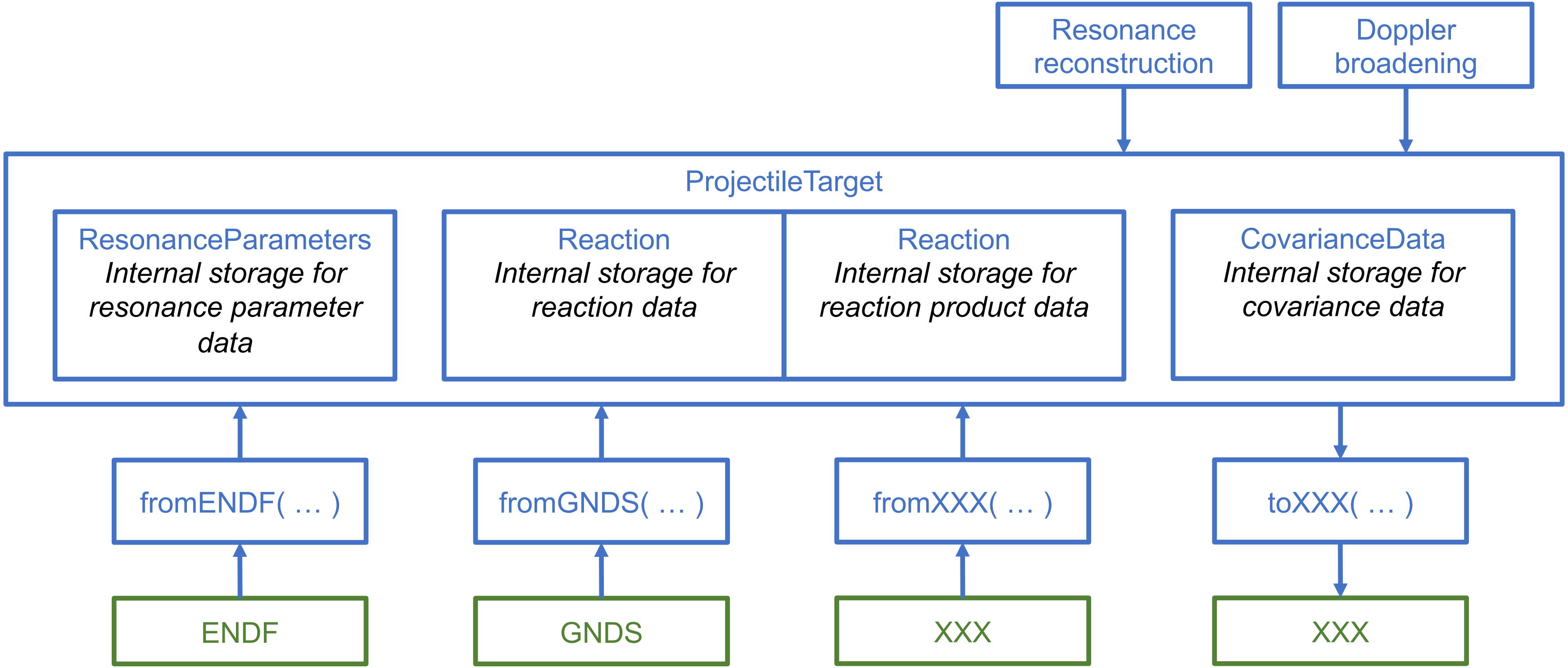
    This is a really short-sighted solution
  - Solution 2: an interface capable of using any format
    - Data is represented in its native format but extracted using a common interface
  - Solution 3: format agnostic data structures
    - Data is represented in its most generic form and can be translated to/from different formats

    This is the one we should use

Example in resonance reconstruction



A format agnostic nuclear data interface in SCION



Current status of GNDStk

- GNDStk core interface
  - A standard independent generic tree and node based interface
  - Using the core interface requires an almost perfect knowledge of the GNDS structure
  - Queries can be verbose due to GNDS complexity

- GNDStk standard interface
  - A simpler high-level interface linked to GNDS standards
  - It provides a simpler and less verbose interface (both C++ and python)
  - It can be generated automatically and can be customised later
  - Non-standard GNDS hierarchies still require the core interface

Current status of GNDStk

- Autogenerating the standard interface has proven to be quite difficult
  - The GNDS specifications were not written for this type of use
  - Circular dependencies in the original GNDS specifications
- We now have a GNDS 2.0 “compliant” interface in C++ and Python
  - The original GNDS 2.0 specifications had to be rewritten
  - It should be able to read all existing GNDS 2.0 files
- There are outstanding issues that we need to solve
  - Compilation time is currently prohibitive
  - Documentation is missing because of the autogeneration

Conclusions

- NJOY2016 will be maintained for the foreseeable future
  - ENDF/B-VIII.1 processing is our current goal
  - More changes on photonuclear data processing are coming

- NJOY modernisation is on track
  - ENDFtk and ACEtk v1.0 releases are “imminent”
  - Development of SCION has started and a format agnostic interface is in the works

- GNDStk development continues but it’s … complicated
  - We now have a GNDS 2.0 “compliant” interface in C++ and Python
  - Outstanding issues need to be solved before we can formally release it