Integrating TAGNDS in Autotalys

C. M. Mattoon

IAEA Consultancy meeting on GNDS/FUDGE/TAGNDS
May 23, 2023

LLNL-PRES-849215

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

GNDS-2.0 is an official standard! Next challenge: increase code support for creating and using GNDS evaluated data

- Nuclear data community is making progress:
  - FUDGE checking capabilities already being used to improve library quality
  - Multiple institutions are updating processing codes to handle GNDS
  - LLNL transport codes using processed data in GNDS
  - GEANT-4 soon to be updated to support GNDS-2.0
- Another major need: tools to help evaluators leverage GNDS
  - Several institutions (LLNL, NNDC, IAEA and some universities) are starting to use GNDS for assembling and patching evaluations
  - Still needed: broader toolkit to help evaluators assemble GNDS 'from scratch'

Translating from ENDF-6 is a good first step, but we also need direct GNDS production

Benefit of GNDS:
Easier to navigate files and fix data

- ENDF-6 supports searching by MF/MT, but not within sections. GNDS separates each product, multiplicity, distribution, etc.

```
<reaction label="sumOfRemainingOutputChannels" ENDF_MT="5">
  <crossSection>...</crossSection>
  <outputChannel genre="sumOfRemainingOutputChannels">
    <Q>...</Q>
    <products>
      <product pid="n" label="n">...</product>
      <product pid="H1" label="H1">...</product>
      <product pid="H2" label="H2">...</product>
      ...  <!-- 130 products total -->
      <product pid="Pt193" label="Pt193">...</product>
      <product pid="Pt193_e5" label="Pt193_e5">...</product>
      <product pid="Pt194" label="Pt194">...</product>
    </products>
  </outputChannel>
</reaction>
```

Pt193 from ENDF-VIII (adopted from TENDL2017), ~24,000 lines in MF6/MT5

- Standard tools like XPath assist with searching through GNDS:

```
# unix command-line query to count number of products in the MT=5 reaction:
> xmllint --xpath 'count(//reaction[@label="sumOfRemainingOutputChannels"]//product)' n-078_Pt_193.xml
130
```

Benefit of switching to GNDS:
Store evaluated and processed data together

- GNDS file may store multiple ‘styles’ of data, plus details of how each style was generated
- Goal: simplify sharing processed data between institutions

```
<reactionSuite projectile="n" target="He4" evaluation="ENDF/B-8.0"
               format="1.10" projectileFrame="lab">
  <styles>
    <evaluated label="eval" date="2010-09-01" library="ENDF/B" version="8.0.0">
      <temperature value="0.0" unit="K"/>...</evaluated>
    <heated label="heated_000" derivedFrom="eval">
      <temperature value="2.586e-08" unit="MeV/k"/></heated>
    <heatedMultiGroup label="MultiGroup_000" derivedFrom="heated_000" .../>
    <heated label="heated_001" derivedFrom="MonteCarlo_cdf" date="2019-01-07">
      <temperature value="1e-07" unit="MeV/k"/></heated>
    <heatedMultiGroup label="MultiGroup_001" derivedFrom="heated_001" .../>
  </styles>
  ...
  <reactions>
    <reaction label="n + He4" ENDF_MT="2">
      <crossSection>
        <XYs1d label="eval">...</XYs1d>
        <XYs1d label="heated_000">...</XYs1d>
        <gridded1d label="MultiGroup_000">...</gridded1d>
        <XYs1d label="heated_001">...</XYs1d>
        <gridded1d label="MultiGroup_001">...</gridded1d>
      </crossSection>
      ...
```

ENDF-GNDS translator helps migrate libraries

Translation must preserve original physics content/meaning

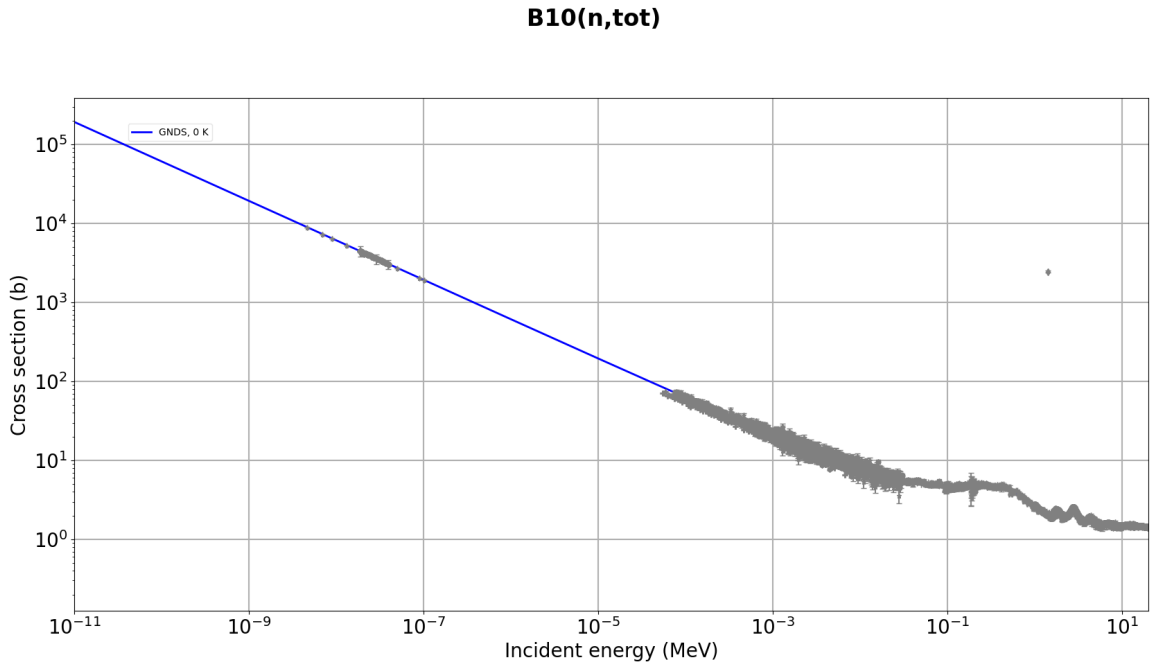
- Translator is included with FUDGE. Sample use:

```
> python ~/fudge/brownies/bin/rePrint.py ~/ENDF-VIII.0/neutrons/n-029_Cu_063.endf
> ls *.xml
test.endf6-covar.xml  test.endf6.xml                      # GNDS files produced by translator
> ls *.noLineNumbers
test.endf6.noLineNumbers  test.endf6.orig.noLineNumbers   # ENDF-6 files, one produced by translating
                                                          # GNDS back to ENDF-6. Compare the two files
                                                          # to test fidelity of translation
```

- Some evaluations still cannot be translated, however. Common causes of translation errors:
  - Bad data in the original evaluation. For example,
    - cross section values not sorted in ascending order
    - inconsistencies between MF2 and MF32
  - Infrequently-used ENDF-6 options that are not yet supported by the translator. For example,
    - Adler-Adler resonance parameters
    - Isotope-specific resonance parameters listed in an elemental evaluation

FUDGE includes several utilities to help view and check evaluations

- checkGNDS.py: run physics checks on reactionSuite and covarianceSuite
- energySpectrum.py: summarize and plot outgoing spectra for specified incident energy and product
- plotEvaluation.py: plot various observables including experimental data (uses x4i)



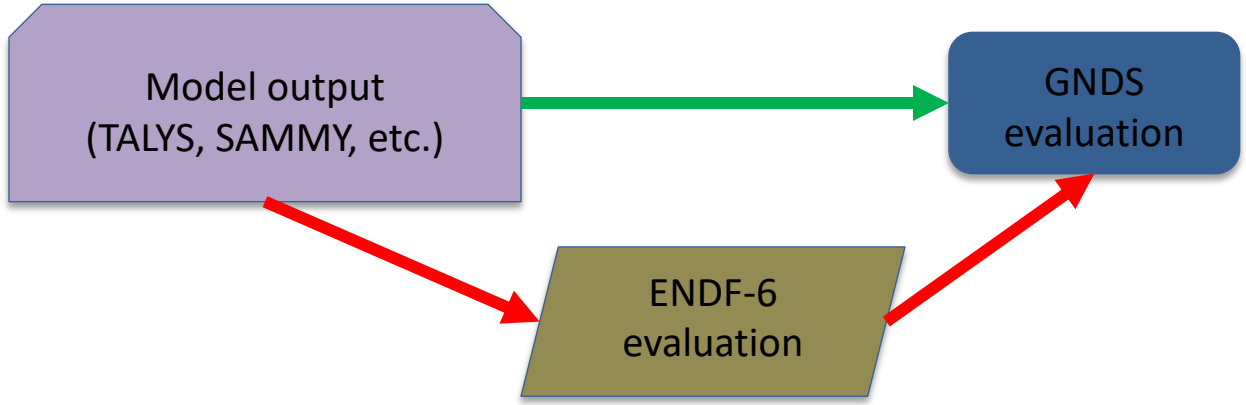
GNDS translator is strict, provides useful errors
Found/fixed bugs before ENDF-VII.1 and VIII.0 releases

- Four data issues weren't fixed in time for ENDF-VII.1
  - Incorrect primary capture gamma in p-001_H_002.endf
  - Incorrect RTYP in three decay files (Li8, N16 and Au186)
  - **All fixed in ENDF-VIII**

- Other libraries still need some fixes to facilitate translating to GNDS
  - Godfree to address this more tomorrow

ENDF-6 translation is important, but we also need ability to directly generate GNDS

- Avoids limitations inherent in ENDF/B format
- Requires some initial effort, but eventually the direct route to GNDS should be simpler and more maintainable



- Thanks to IAEA support, we now have a TALYS-GNDS translator
  - **TAGNDS** translator reads TALYS output, generates and populates GNDS classes in FUDGE, writes final result to XML

TAGNDS was originally written for TALYS only but has since been expanded to work with autotalys framework.

- TAGNDS was written in Python
  - Requires FUDGE v6.1 or later (https://github.com/LLNL/fudge), Python 3.7+
- Currently about 2800 lines of code
- Two main scripts:
  - *generateGNDS.py* for TALYS outputs only,
  - *autotalysGNDS.py* for translating full autotalys results

```
python3 tagnds/bin/autotalysGNDS.py /path/to/AutoTalys/run/directory
```

- Both produce XML GNDS-2.0 files following the TENDL naming convention:
  - E.g. n-Al26_m1.gnds.xml, p-Mn55.gnds.xml.

Some background on Autotalys:

- Provides a single command to generate TENDL-style evaluation:
  1. run TALYS with reasonable default input parameters,
  2. optionally re-run TALYS to renormalize to experimental data,
  3. optionally run TARES, TAFIS and TANES to add resonance parameters and fission observables,
  4. run TEFAL to generate one or more ENDF-6 formatted files,
  5. run various other codes on resulting files: plotting, basic processing, etc.

- Sample input:

```
autotalys -element Al -mass 27 -Liso 1 -Ltarget 1 -proj n -ntalys 3 -bins 20 -low -thin -s20 -sdefault
-acf -eaf -njoy -residual -isomer -recoil -noclean -binsrand 20 -plot -subfission
```

Running autotalys input generates several directories:

```
ls Al027m
input             # directory with TALYS and TEFAL inputs
lib               # final result: files in ENDF-6 and ACE format
notasman
plots
tables
work              # working directory with most output files
```

- Work directory contains all TALYS outputs:
  - xs*.tot and xs*.gam with cross sections
  - gam*.tot with discrete gamma information
  - sp*.tot and sp*.rec files with outgoing energy spectra
  - *spec.tot files with pre-equilibrium fractions (used in Kalbach-Mann distributions)
  - optionally a ‘cov’ directory for covariances
  - Additional MF* files with partially assembled ENDF files

Autotalys requires resources beyond TALYS

- ‘libraries.t6’ directory: supplemental libraries to help fill in information not handled by Hauser-Feshbach formalism
  - resonance parameters, nubar, PFNS and so on
- Also contains EXFOR datasets and partial or total evaluations from ENDF, JEFF and others
  - Much of this is in ENDF-6 format, translation already supported by FUDGE

```
 53M    libraries.t6/FY
1.1G    libraries.t6/a
1.7G    libraries.t6/d
2.5G    libraries.t6/fnsbase
2.9G    libraries.t6/g
1.3G    libraries.t6/h
  0B    libraries.t6/libraries
 24G    libraries.t6/n
 62M    libraries.t6/nubarbase
4.6G    libraries.t6/p
4.9G    libraries.t6/plots
6.1G    libraries.t6/resbase
105M    libraries.t6/resfiles
2.6M    libraries.t6/stat
1.3G    libraries.t6/t
```

Workflow within TAGNDS:

- Pull all available information out of TALYS output files
  - cross sections, Q-values, angular distributions for two-body reactions, spectra and pre-equilibrium parameters for N-body reactions, radioactive production data, etc.
- Read resonance parameter data if available
  - Still using ENDF MF=2 format here since data are stored that way in libraries.t6
- Add fission info (MT=452, 455, 456) if available
- Note: fission and capture require special handling!
  - TALYS supports breaking them up into multiple reactions, but downstream codes (and current resonance parameterizations) probably aren't ready to deal with multiple fission / capture reactions
  - Utilities 'gatherCaptureReactions' and 'gatherFissionReactions' convert to more familiar style

Running autotalysGNDS:

```
python3 ~/apps/tagnds/bin/autotalysGNDS.py -h
usage: Read output files from a completed autotalys run,
and generate GNDS-formatted evaluation from the results including all available data.
Uses FUDGE to manage and write GNDS.
       [-h] [-o OUTPUT] [-v] [--library LIBRARY] [--style STYLE]
       [--energyUnit ENERGYUNIT] [--crossSectionUnit CROSSSECTIONUNIT]
       [--autotalysInstallDir AUTOTALYSINSTALLDIR]
       workDir

positional arguments:
  workDir               Directory where autotalys was run

optional arguments:
  -h, --help            show this help message and exit
  -o OUTPUT, --output OUTPUT
                        output file name. Defaults to
                        $projectile_$target.gnds.xml
  -v, --verbose         enable verbose output
  --library LIBRARY     Library name, e.g. 'TENDL'
  --style STYLE         Label for the evaluated style
  --energyUnit ENERGYUNIT
                        desired unit for storing incident / outgoing energies
  --crossSectionUnit CROSSSECTIONUNIT
                        desired unit for storing cross sections
  --autotalysInstallDir AUTOTALYSINSTALLDIR
                        AUTOTALYS installation directory, required unless
                        'AUTOTALYS' env. variable is set
```

autotalys + autotalysGNDS.py has been tested against an initial range of protares

	n	p	d	t	h	a	g
F19	X	X	X	X	X	X	X
Al26_m	X	X	X	X	X	X	X
Mn55	X	X	X	X	X	X	X
Fe56	X	X	X	X	X	X	X
Am242	X	X	X	X	X	X	X

- Selected these to cover a wide ZA range, fissile and not, ground and excited-state targets.
  - Other useful isotopes to add to this test matrix?
- Start with simple tests:
  - Does autotalysGNDS.py finish and produce an output file?
  - Does resulting file conform to GNDS schema?

Although TAGNDS produces a GNDS file in each case, results need more inspection

- Many warning messages may appear during translation
- These are mostly recoverable, but they may indicate that TAGNDS is having to guess how to interpret TALYS output

```
Warning: no distributions found for dn.LL24! Assuming isotropic.
…
Warning: dropping non-zero cross section point(s) below threshold for reaction '3np’
…
Warning: trouble parsing data from sp000001.tot. Attempting to use column positions
…
Warning: no distribution found for ejectile 'h' in reaction type 'h2a’
…
Warning: skipping incident energy '0' in file nnleg.L00. Format error?
…
Warning: no distribution found for ejectile 'a' in reaction type '100001fission'
```

More detail on some warning messages:

- "Warning: no distributions found for dn.LL24! Assuming isotropic."
  - Refers to 2-body reaction to specific excited state in the residual. Cross section file 'dn.L24' was found, but angular distribution 'dnleg.L24' or 'dnang.L24' not found

```
ls Am242/work/dn*
Am242/work/dn.L00    Am242/work/dn.L09    Am242/work/dn.L18    Am242/work/dnang.L00 Am242/work/dnang.L09 Am242/work/dnleg.L07
Am242/work/dn.L01    Am242/work/dn.L10    Am242/work/dn.L19    Am242/work/dnang.L01 Am242/work/dnang.L10 Am242/work/dnleg.L08
Am242/work/dn.L02    Am242/work/dn.L11    Am242/work/dn.L20    Am242/work/dnang.L02 Am242/work/dnleg.L00 Am242/work/dnleg.L09
Am242/work/dn.L03    Am242/work/dn.L12    Am242/work/dn.L21    Am242/work/dnang.L03 Am242/work/dnleg.L01 Am242/work/dnleg.L10
Am242/work/dn.L04    Am242/work/dn.L13    Am242/work/dn.L22    Am242/work/dnang.L04 Am242/work/dnleg.L02
Am242/work/dn.L05    Am242/work/dn.L14    Am242/work/dn.L23    Am242/work/dnang.L05 Am242/work/dnleg.L03
Am242/work/dn.L06    Am242/work/dn.L15    Am242/work/dn.L24    Am242/work/dnang.L06 Am242/work/dnleg.L04
Am242/work/dn.L07    Am242/work/dn.L16    Am242/work/dn.con    Am242/work/dnang.L07 Am242/work/dnleg.L05
Am242/work/dn.L08    Am242/work/dn.L17    Am242/work/dn.tot    Am242/work/dnang.L08 Am242/work/dnleg.L06
```

More detail on some warning messages:

- "Warning: dropping non-zero cross section point(s) below threshold for reaction '3np'"
  - Refers to file xs310000.tot
  - Perhaps due to a binning effect?

```
# d + 242Am    : (d,3np)              Total
# Q-value     =-1.44093E+01
# E-threshold= 1.45292E+01
# # energies =     22
#      E             xs          gamma xs   xs/res.prod.xs
 2.50000E-01 0.00000E+00 0.00000E+00 0.00000E+00
 5.00000E-01 0.00000E+00 0.00000E+00 0.00000E+00
 ...
 1.20000E+01 0.00000E+00 0.00000E+00 0.00000E+00
 1.30000E+01 0.00000E+00 0.00000E+00 0.00000E+00
 1.40000E+01 6.07047E-20 5.31856E-20 1.43513E-13
 1.50000E+01 4.33368E-17 1.14615E-16 6.46327E-12
 1.60000E+01 2.18050E-14 4.34609E-14 3.50092E-10
 ...
```

More detail on some warning messages:

- "Warning: trouble parsing data from sp000000.tot. Attempting to use column positions"
  - Usually indicates negative probability for some incident / outgoing energy

```
# t +  26Al(m): (t,g)              Spectra
# E-incident =   17.00000
#
# # energies =    78
#  E-out    gamma       neutron     proton      deuteron    triton      helium-3    alpha
   0.001 1.31837E-07 1.57009E-04 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
   ...
  28.000 3.98783E-06 9.06897E-08 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 8.59389E-09
  29.000 7.46994E-06 2.14706E-08 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
  30.000-3.52215E-05 4.01245E-09 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
```

More detail on some warning messages:

- "Warning: skipping incident energy '0' in file nnleg.L00. Format error?"
  - Incident energies sometimes written without sufficient precision

```
# n + 242Am Elastic scattering Legendre coefficients
# E-incident =    0.00000
#
# # coeff.    =    8
#  L        Total           Direct          Compound        Normalized     ENDF-6
  0     8.76148E+02     8.68747E+02     7.40104E+00     7.95775E-02     1.00000E+00
  1     1.96953E-04     1.96953E-04     0.00000E+00     1.78886E-08     2.24794E-07
  2     1.48016E-10     1.39313E-10     8.70383E-12     1.34438E-14     1.68940E-13
  3     5.77316E-15     5.77316E-15     0.00000E+00     5.24356E-19     6.58925E-18
  ...
# n + 242Am Elastic scattering Legendre coefficients
# E-incident =    0.00001
#
# # coeff.    =   10
#  L        Total           Direct          Compound        Normalized     ENDF-6
  0     8.75581E+02     8.68199E+02     7.38122E+00     7.95776E-02     1.00000E+00
  1     4.97182E-03     4.97182E-03     0.00000E+00     4.51866E-07     5.67831E-06
  2     9.25821E-08     8.71425E-08     5.43964E-09     8.41438E-12     1.05738E-10
  3     2.23377E-13     2.23377E-13     0.00000E+00     2.03017E-17     2.55119E-16
  ...
```

Another useful test: compare TAGNDS result with TEFAL + endf2gnds translation result

- Ideally this should be a one-to-one comparison: same TALYS run was used to produce both TEFAL and autotalysGNDS outputs
  - Mostly true for incident neutrons, less so for other projectiles

- Main remaining issue: TEFAL appears to be making additional corrections and adding data missing from direct TALYS output.

TAGNDS separates reaction channels where possible, TEFAL may combine them – e.g. using MT=4 instead of 51-91

- Possibly related to the ‘no distributions found, using isotropic’ warning
- Example for α + Fe56:
  - TAGNDS produces reactions for MT=51 – 73 and MT=91, TEFAL combines into MT=4
  - TAGNDS produces multiple capture reactions: MT=900-932 + MT=999, TEFAL combines into MT=102

TEFAL adds distribution data below the first incident energy computed by TALYS. Where from?

- Example 1: n + Am242 elastic angular distributions

Via TEFAL

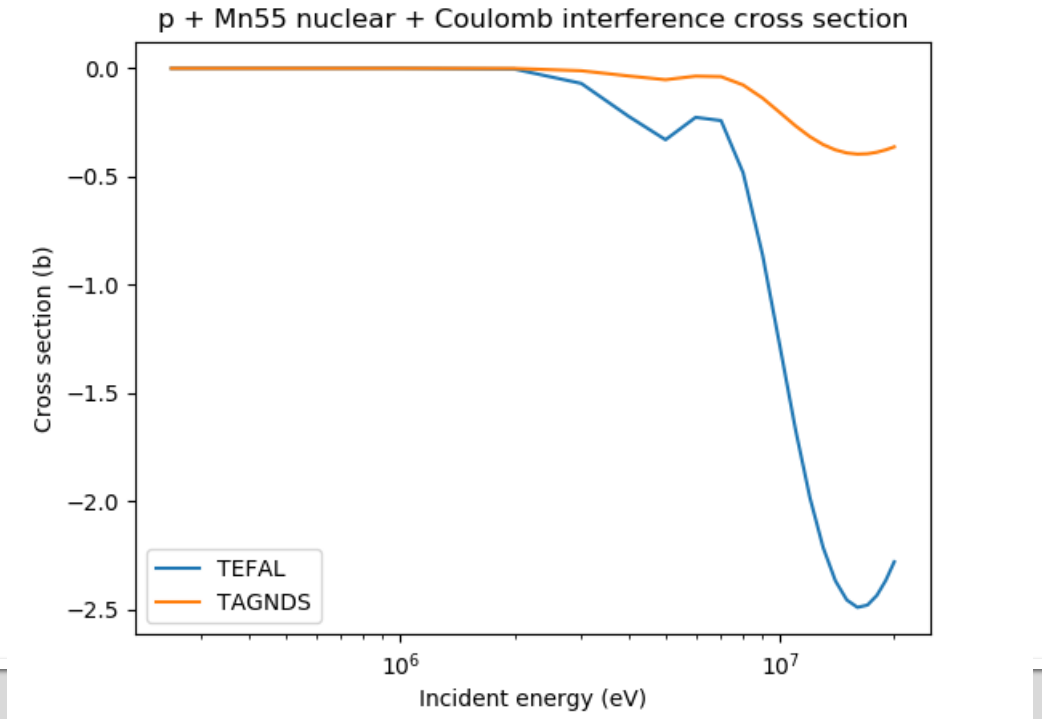
```
<angularTwoBody label="eval" productFrame="centerOfMass">
  <XYs2d>
    <axes>
      <axis index="2" label="energy_in" unit="eV"/>
      <axis index="1" label="mu" unit=""/>
      <axis index="0" label="P(mu|energy_in)" unit=""/></axes>
    <function1ds>
      <Legendre outerDomainValue="1e-5"><values>1 0 0</values></Legendre>
      <Legendre outerDomainValue="0.0253"><values>1 2.24794e-7 1.6894e-13</values></Legendre>
      <Legendre outerDomainValue="1e2"><values>1 2.24794e-7 1.6894e-13</values></Legendre>
      <Legendre outerDomainValue="1e3"><values>1 7.76771e-4 1.12057e-6</values></Legendre>
```

Via TAGNDS

```
<angularTwoBody label="eval" productFrame="centerOfMass">
  <XYs2d>
    <axes>
      <axis index="2" label="energy_in" unit="eV"/>
      <axis index="1" label="mu" unit=""/>
      <axis index="0" label="P(mu|energy_in)" unit=""/></axes>
    <function1ds>
      <Legendre outerDomainValue="10"><values>1 5.67831e-6 1.05738e-10 2.55119e-16 4.50397e-17 -1.79939e-17 ... </values></Legendre>
      <Legendre outerDomainValue="1e2"><values>1 5.8359e-5 1.0609e-8 2.72485e-13 5.8495e-17 1.46522e-18 ... </values></Legendre>
      <Legendre outerDomainValue="2e2"><values>1 1.18622e-4 4.25176e-8 2.18503e-12 2.60909e-16 -6.61557e-18 ...</values></Legendre>
```

Still some differences in Coulomb elastic cross sections

- TEFAL is doing some rescaling?
- Plot shows just the nuclear and interference contributions to cross section, omitting pure Rutherford scattering:



What about other modeling / evaluation codes?

- TAGNDS is written for TALYS, but the same approach should work for other modeling codes.
- Where possible I separated the parts responsible for reading TALYS output from the parts responsible for generating GNDS
- Use TAGNDS as a basis to support other codes?
  - YAHFC (LLNL Monte Carlo Hauser-Feshbach code by E. Ormand)
  - CCONE
  - COH3
  - EMPIRE
  - Others?

Basic Autotalys + TAGNDS capability is in place, some additional work needed to ensure direct GNDS translation is complete

- 'autotalysGNDS.py' script can be executed at the end of an autotalys run to generate GNDS, but resulting files still need to be checked against TEFAL

- Comparing GNDS files produced by TAGNDS to files from TEFAL + endf2gnds is a useful but time-intensive process.
  - *Better tools for summarizing differences between GNDS files needed!*

- Major differences seem to come from two main sources:
  - TEFAL choosing to combine individual channels (e.g. summing 51-91 into MT=4)
  - TEFAL making further physics transformations after reading TALYS outputs

Once GNDS files are generated, next step is to ensure they can be processed and used!

- XML schema validation reveals some minor problems with TAGNDS results
  - Mostly in documentation
- Test translation back to ENDF-6?
  - Requires some FUDGE work: need to improve defaults for when data can be stored more than one way in ENDF-6
- Try processing TAGNDS results!
- LLNL has also processed most of TENDL-2021 in GNDS format (produced via TEFAL and endf2gnds.py)

Lawrence Livermore
National Laboratory