IAEA

From model code outputs to GNDS -TAGNDS -

C. M. Mattoon and J.-Ch. Sublet

Lawrence Livermore National Laboratory

International Atomic Energy Agency

Objectives

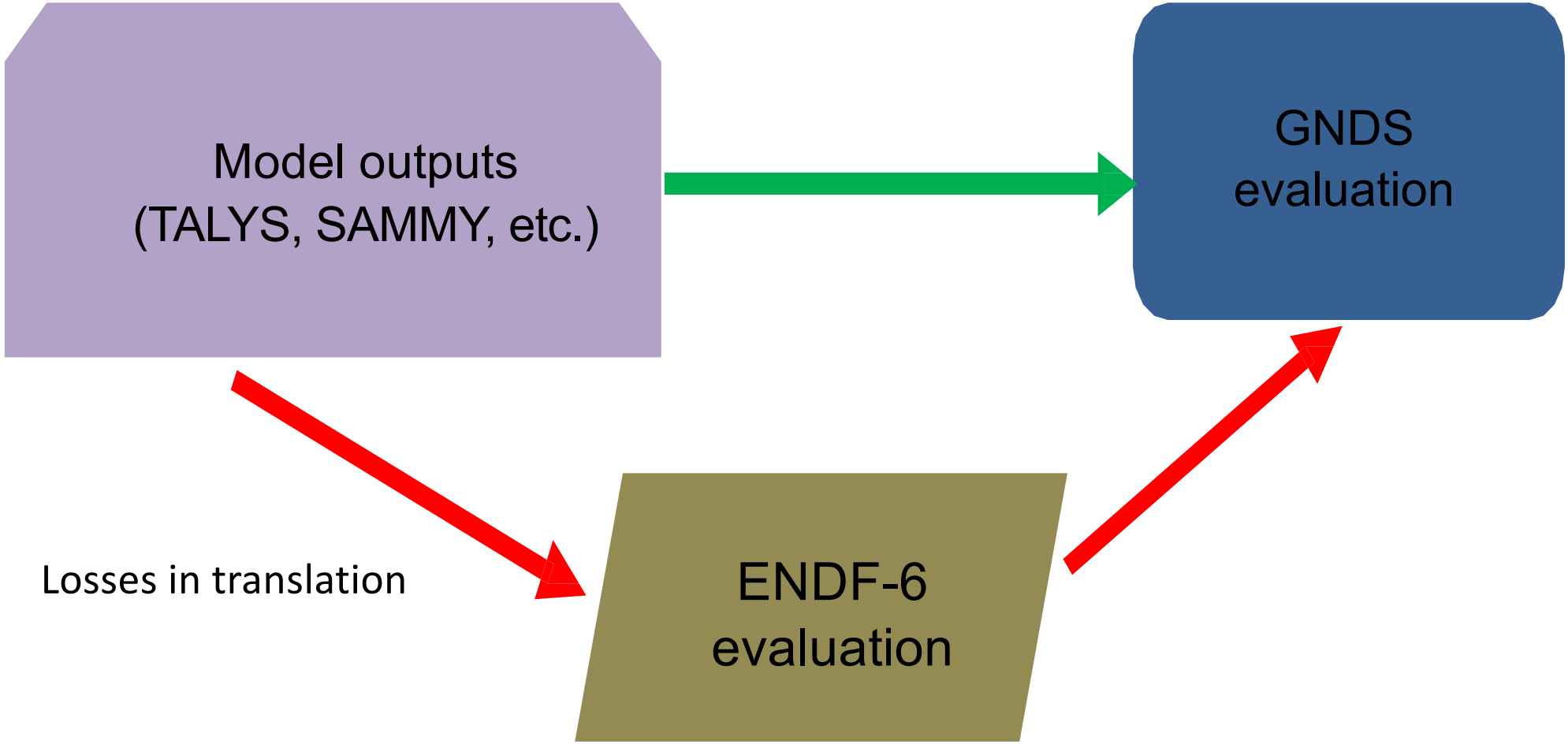
- ENDF-6 to GNDS translator
- GNDS translator is strict, provides useful information
- Found/fixed bugs before library releases
- TENDL's translation issues
  - 1/3 incident neutron fail to translate properly (out of 2813)
  - All isomeric targets fail 513 m (1^{st}) and 30 n (2^{nd}) .
    - Most seem to have the same cause: inconsistencies between residual excitation energies listed in MF8/9/10
  - Sub-actinide fission cross sections using ZAP=0 instead of -1
    - Translator has a command-line option to allow this
  - Super-heavy elements: need to update translator for Z > 100
  - Assorted other problems:
    - non-zero MF8 ELFS for ground state
    - Lumped-sum covariances not summing over any reactions
    - etc.

Objectives

- We would like to contribute testing before the next TENDL release, expect that these issues should be easy to fix
- ENDF2GNDS exists, but the fit is tight
- Translation is important, but we also need the ability to directly generate GNDS, THE objective
  - Avoids limitations inherent in ENDF/B format
- Requires some initial effort, but eventually the direct route to GNDS should be simpler, sustainable, maintainable
- With IAEA's support, we now have a TALYS-GNDS translator
  - Tentatively called TAGNDS, the translator reads TALYS output (not one of its 7 output forms) into GNDS classes in FUDGE, then serializes the result to XML

Schematics

- TALYS-GNDS parser: TAGNDS



TAGNDS (TALYS to GNDS) overview

- Written in Python - compatible with v2.7 or v3.5 and later
  - Requires the development version of FUDGE. Another release is coming shortly, in the meantime a pre-release copy is available
- Currently about 1400 lines of code (including blank lines)
- Main driver is the 'generateGNDS.py' script. Sample usage:

```
python tagnds/bin/generateGNDS.py /path/to/TALYS/run/directory
```

- By default generateGNDS.py writes out a GNDS/XML file
- following the TENDL naming convention:
  - E.g. n-Al26_m1.gnds.xml, p-Mn55.gnds.xml, etc.

Options when running generateGNDS.py

```
➜  TALYS2GNDS python tagnds/bin/generateGNDS.py -h
usage: This tool reads output files from a completed TALYS run,
and generates a GNDS-formatted evaluation from the results.
It uses the LLNL code FUDGE to manage and write GNDS data.
       [-h] [-o OUTPUT] [-v] [--library LIBRARY] [--style STYLE]
       [--energyUnit ENERGYUNIT] [--crossSectionUnit CROSSSECTIONUNIT]
       [--talysInstallDir TALYSINSTALLDIR]
       talysDir

positional arguments:
  talysDir              Directory containing TALYS outputs

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
  --talysInstallDir TALYSINSTALLDIR
                        TALYS installation directory, required unless 'TALYS'
                        environment variable is set
```

Quick guide to using TAGNDS

- Install FUDGE
- Add FUDGE and TAGNDS to the PYTHONPATH
- Run TALYS on an input file
  - Input should enable options ‘endf’, ‘outSpectra’, ‘outLegendre’, etc.
  - Redirect output to a file called ‘output’ in the same directory, i.e. talys < input > output or do not it delete if using autotalys
- Use generateGNDS.py to translate results

First simple matrix TAGNDS testing

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

- TAGNDS testing has started focusing on a small sample of target nuclides
- Selected these to cover a wide ZA range, fissile or not, ground and excited-state targets.
  - Other useful isotopes to add to test matrix?
- Simple tests for now:
  - Does generateGNDS.py finish and produce an output file?
  - Does resulting file conform to GNDS ‘schema’?

Further TAGNDS testing

- For 630 evaluations targets nuclides, Z=1-100, Hydrogen to Fermium, including as target some 38 m (1st) isomeric states. This subset represents the union of all targets for n-induced that exists in the currently traditional libraries: JEFF-3.3, JENDL-4.0 and ENDF/B-VIII.0.
  - Fails to translate ~1/3 also, warnings

Pu237:MF1 MT455 claims 1 delayed neutron groups, but MF5 MT455 claims 6 groups **(23)**
Pu238:MF1 MT455 claims 6 delayed neutron groups, but MF5 MT455 claims 8 groups **(1)**

MF8 residual level energy = 0.0 for level 0 of ZA = 47107 not close to MF10's value = 93125.0 for MT = 4. **(173)** == > QI need to be equal to 0.0 in the MF10

Easy but necessary correction in the evaluated files

  - Fudge issue(s) with metastable, as target not product **(38)**
- All the above processed usefully from s30 files for the Monte Carlo codes: MCNP, SERPENT, OpenMC, the inventory code FISPACT-II and the materials sciences code SPECTRA-PKA

Surprises in TALYS-1.9 output

- Most (but not all) numbers are separated by one or more spaces, so TAGNDS can ‘split’ on whitespace

- Prefer Python str.split() method rather than hard-coding column indices, in case indices change in future versions

- Problem occasionally shows up in *spec*.tot files:

```
# p + 242Am: alpha    spectrum
# E-incident =    4.000
#
# # energies =    63
# E-out    Total        Direct    Pre-equil.  BU ratio    Stripping   Knock-out   Break-up
   0.800 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
   0.900 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
   ...
   1.500 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
   1.600 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
   1.700 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
   1.800 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00-0.00000E+00-0.00000E+00 0.00000E+00
   1.900 4.34403E-44 0.00000E+00 4.34403E-44 0.00000E+00 4.34403E-44 0.00000E+00 0.00000E+00
```

Surprises in TALYS-1.95 output

- xs*.tot lists threshold along with cross section… but first non-zero cross section point is occasionally below threshold

xs100000.tot:

```
# p +  26Al(m): (p,n)                Total
# Q-value    =-5.62318E+00
# E-threshold= 7.70665E+00
# # energies =    27
#     E           xs          gamma xs   xs/res.prod.xs
 5.00000E-03 0.00000E+00 0.00000E+00 0.00000E+00
 ...
 4.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
 5.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
 6.00000E+00 4.84898E+00 0.00000E+00 1.00000E+00
 7.00000E+00 1.16141E+01 0.00000E+00 1.00000E+00
```

- Conclusion: Q-value and cross section are consistent, energy threshold is not, TALYS fixed grid !!

Ongoing: TAGNDS against TENDL translation

- Numbers may differ due to changes in TALYS, but the overall structure of the evaluation should be the same
  - Mostly true for incident neutrons, less so for other projectiles

- Comparisons helped uncover several faulty assumptions made in the translator
  - E.g. assuming that outgoing N-body spectral data should use lin-lin interpolation rather than histogram style

- TEFAL logic for switching to different representations at the transition energies s0, s20 or s30, energy in MeV

Remaining work to complete TAGNDS

- Currently only reading TALYS output, full system will require adding resonances, fission observables, non Hauser-Feshbach forms, etc.

- Capture and fission are broken up into partial reactions
  - GNDS supports this, but for backwards compatibility with ENDF-6 they must be summed to total reactions

- User needs more control over level of detail
  - i.e. whether to separate reactions or sum them into MF=5

- Expanded testing
  - More targets, increased energy range, run physics checks, translate GNDS results to ENDF-6, apply processing, etc.

What about other modelling

- TAGNDS is written for TALYS, but the same approach should work for other modelling code output
- Where possible the parts responsible for reading TALYS output is separated from the parts responsible for generating GNDS
- Use TAGNDS as a basis to support other model codes?
- Use resonances data forms (widths) databases (novel TARES output: measured and statistical resonances alike)

After generating a GNDS evaluation, what's next?

- Nuclear data processing is a necessary step before data can useful for applications simulation
- Processing tasks include resonance reconstruction, Doppler broadening, computing average/bin outgoing product energies/momenta, grouping, assembling, tightening, etc.
- Compared with the ENDF-6 format frame GNDS can and will contains pre-processed data forms along side the raw forms
- LLNL code FUDGE (**F**or **U**pdating **D**ata and **G**enerating **E**valuations) has been updated to support processing GNDS data
- Other processing tool/protocol need to be created to independently assess the adequacy and robustness of the interpreted forms, assuming that the GNDS physical forms are perfect or sufficient

GNDS and associated infrastructure are maturing!

- TAGNDS provides a framework for translating modelling code output directly to GNDS
- SG-43 is working together with several institutions to make the transition to GNDS smoother
  - Providing APIs for data access
  - Updating processing and physics checking codes
  - Translating existing libraries into GNDS
- Working to translate current evaluated libraries, reporting issues to library maintainers
- Endeavor to work with other users to expand GNDS + new code infrastructure to meet their needs