Thermal Scattering and Self-Shielding in GNDS/FUDGE/GIDI+

Caleb Mattoon
Nuclear Data and Theory Group, NACS/PHY

NCSP TPR
December 3, 2024

LLNL-PRES-2001285
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

NCSP/AM helped fund FUDGE development for URR & TNSL processing and testing

- Early FUDGE development was focused on higher projectile energies & temperatures, but LLNL Criticality Safety team and NCSP helped push us to prioritize thermal / epithermal capabilities

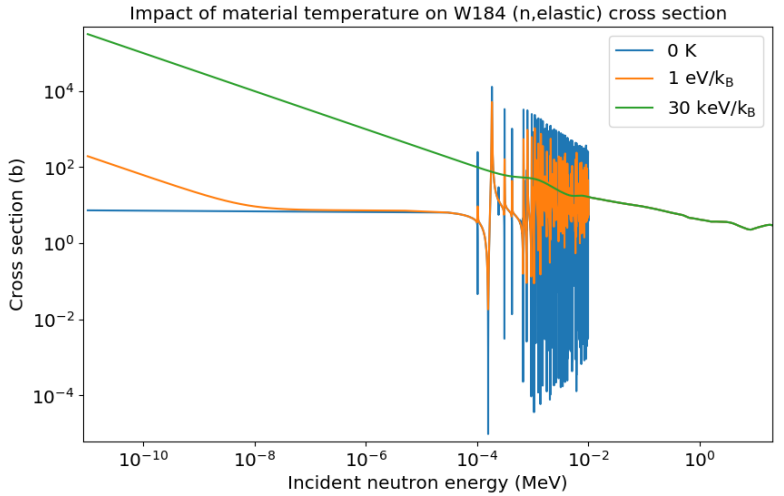
- Specifications for storing URR and TNSL data in GNDS were evolving at the same time, influenced by discussions at WPEC subgroups 38 (GNDS), 48 (TNSL advances) and 51 (URR methodology)

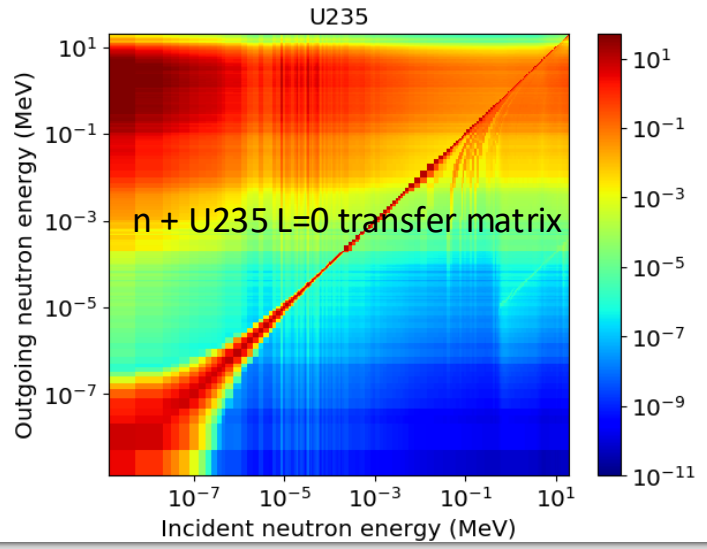
LLNL codes for managing and processing GNDS nuclear data

- FUDGE: For Updating Data and Generating Evaluations
  - Python-based code for reading, writing, modifying, viewing and processing nuclear data
  - Computationally intensive routines written in C and C++

- GIDI+: General Interaction Data Interface+
  - Suite of C++ APIs for accessing GNDS data for use in transport codes
  - Includes API for sampling GNDS data as needed by Monte Carlo codes

- Both codes are open source and used externally



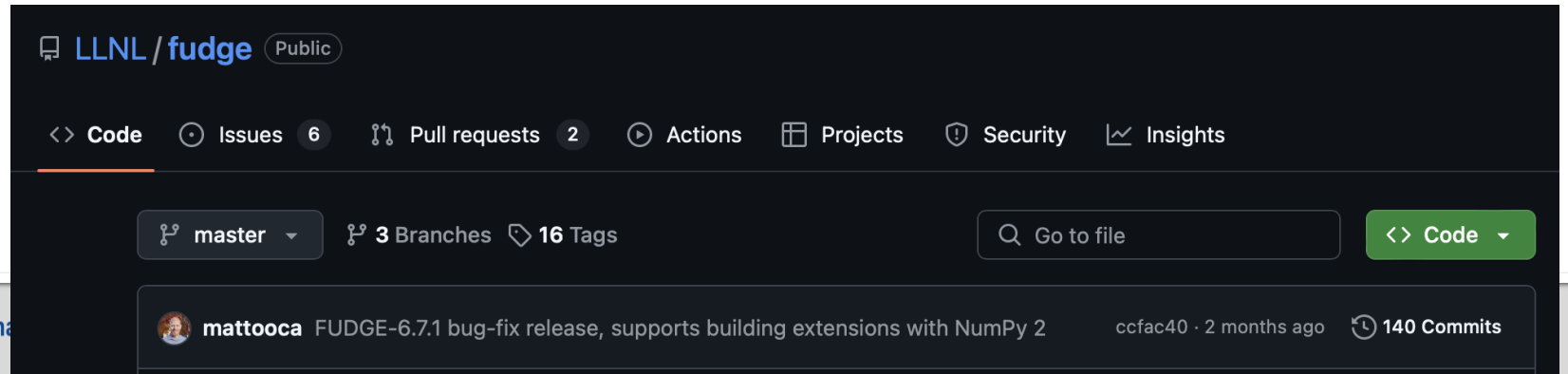


FUDGE-6.7.1 was released on Github in September

- Available at https://github.com/LLNL/fudge
  - Requires Python3.7 or later, numpy, matplotlib, C++ compiler for extensions.
  - Install with pip or with make
  - Release schedule: around 4 public versions per year

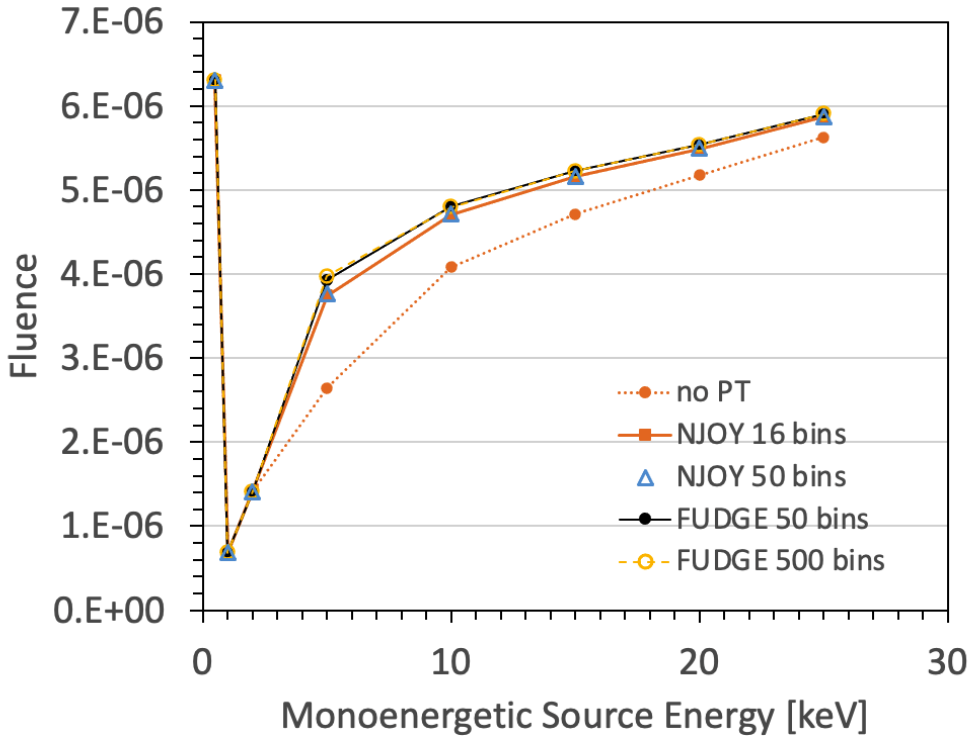
- Recent updates:
  - Full support for translating and processing ENDF/B-VIII.1
  - Improved physics checking tools, focus on diagnosing and fixing energy balance

- Next release coming soon. Focus areas:
  - Improved support for the GRIN project
  - Better tools to support evaluators
  - TNSL processing improvements

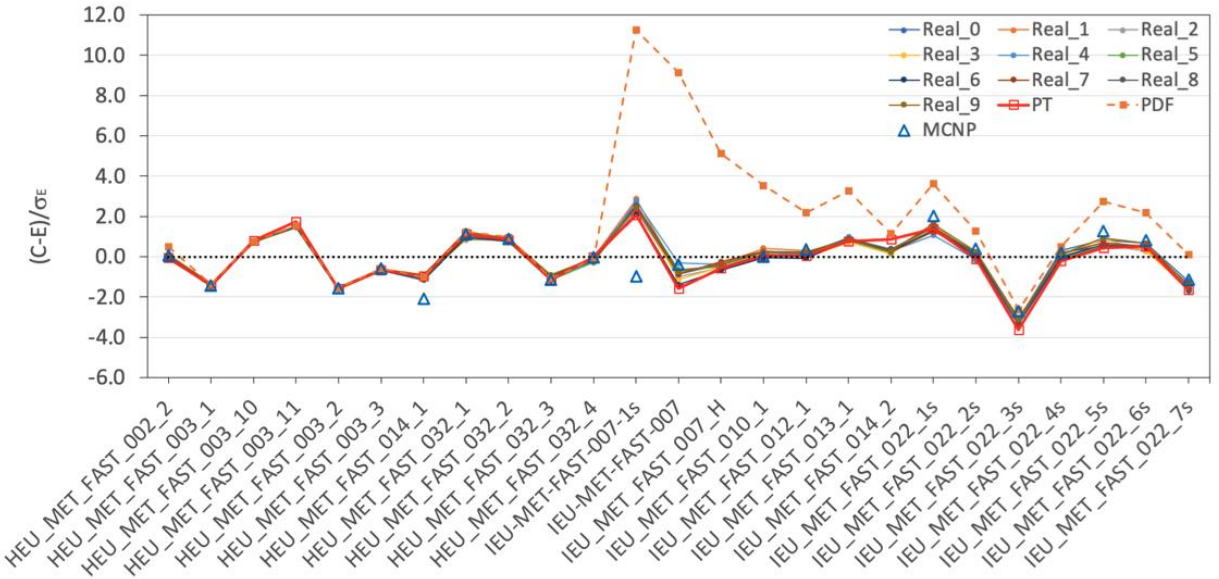


FUDGE-generated URR probability tables are similar to NJOY and FRENDY, with good criticality and shielding performance.

Simulated neutron transmission through a slab of Pu239:
FUDGE PTs give ~5% higher transmission at 5 & 10 keV



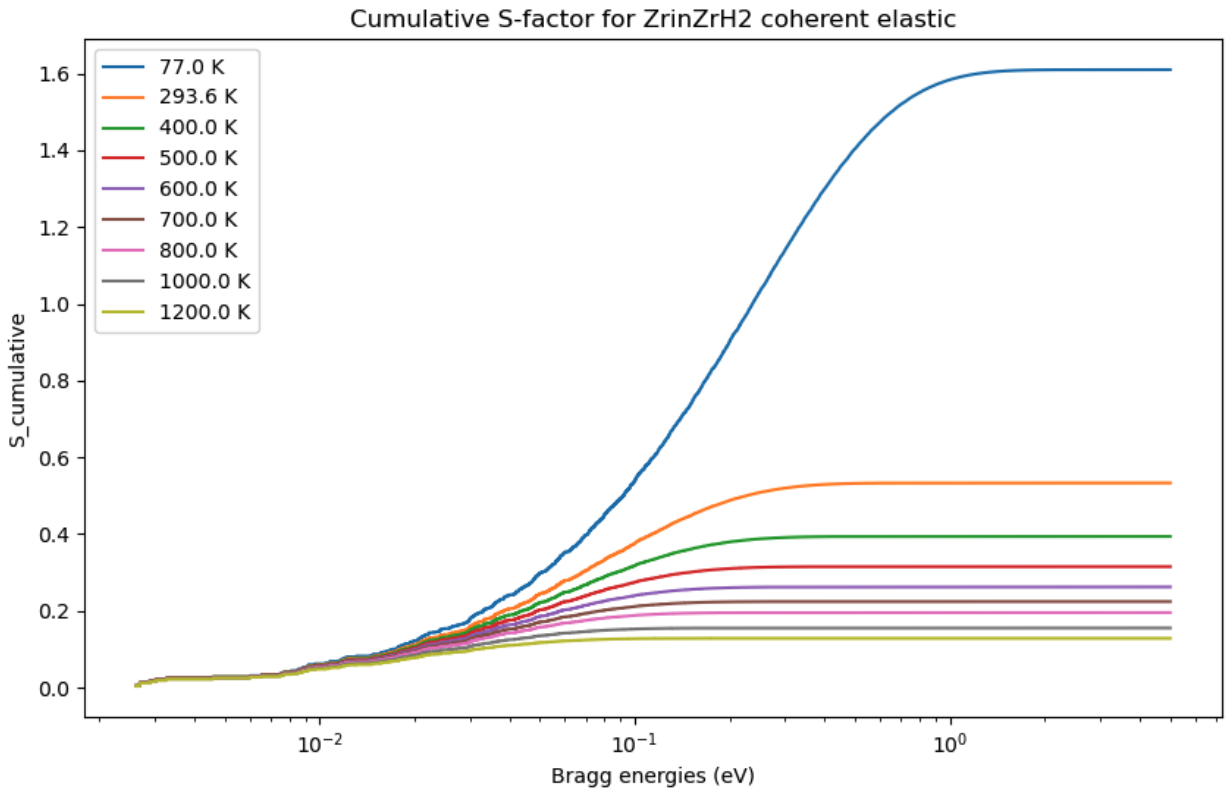
Impact of self-shielding on critical assemblies,
comparing probability tables and individual URR realizations



Results published by M.-A. Descalle in PHYSOR 2024 proceedings: https://www.ans.org/pubs/proceedings/article-55566/

ENDF/B-VIII.1 significantly expanded the TNSL sub-library. Some new evaluations are challenging to process and use!

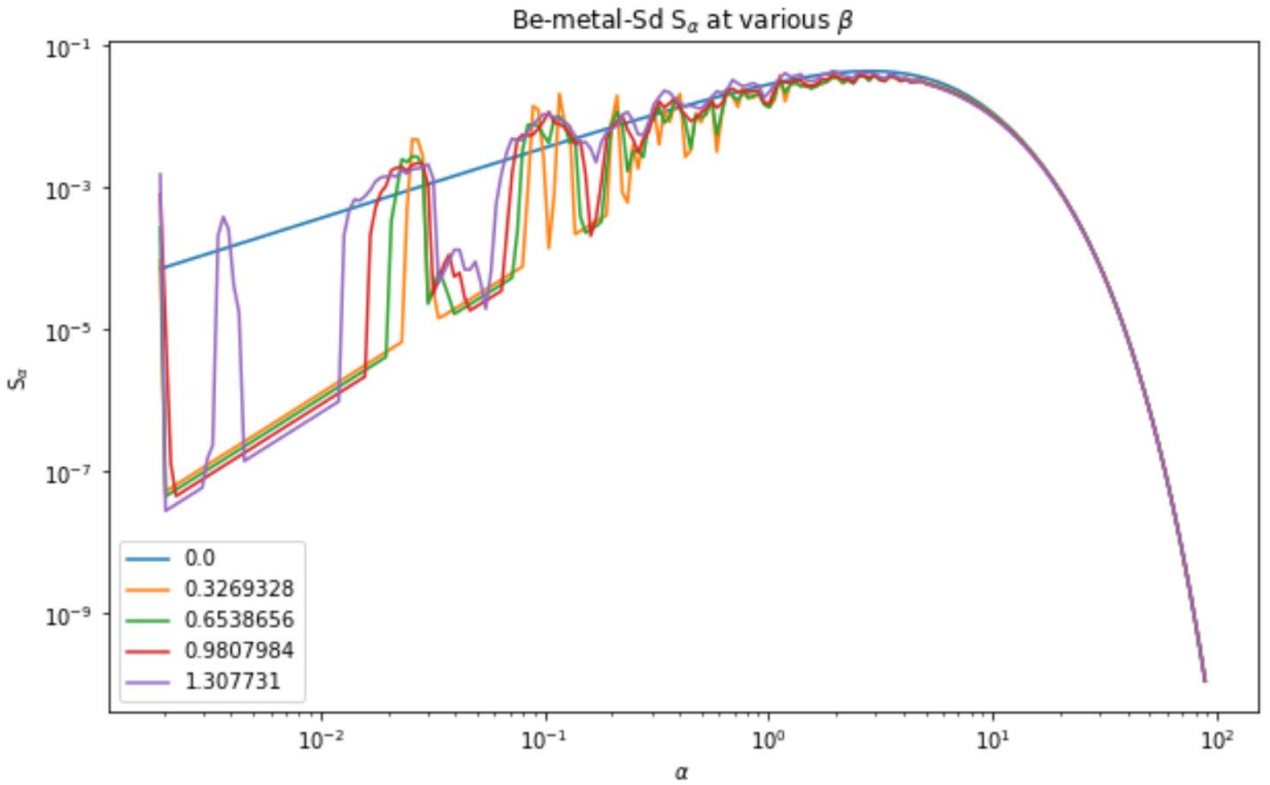
- Coherent elastic: new evaluations include many more Bragg edges than we've seen before:
  - ZrH2 evaluation lists over 54K Bragg edges, nearly half of which make no difference to the cumulative S-factor
  - CinCF2 with 34K Bragg edges,
  - Several U metal evaluations with 7-8K edges
- Required processing changes for better memory management + more efficient sampling



https://git.nndc.bnl.gov/endf/library/thermal_scatt/-/issues/85

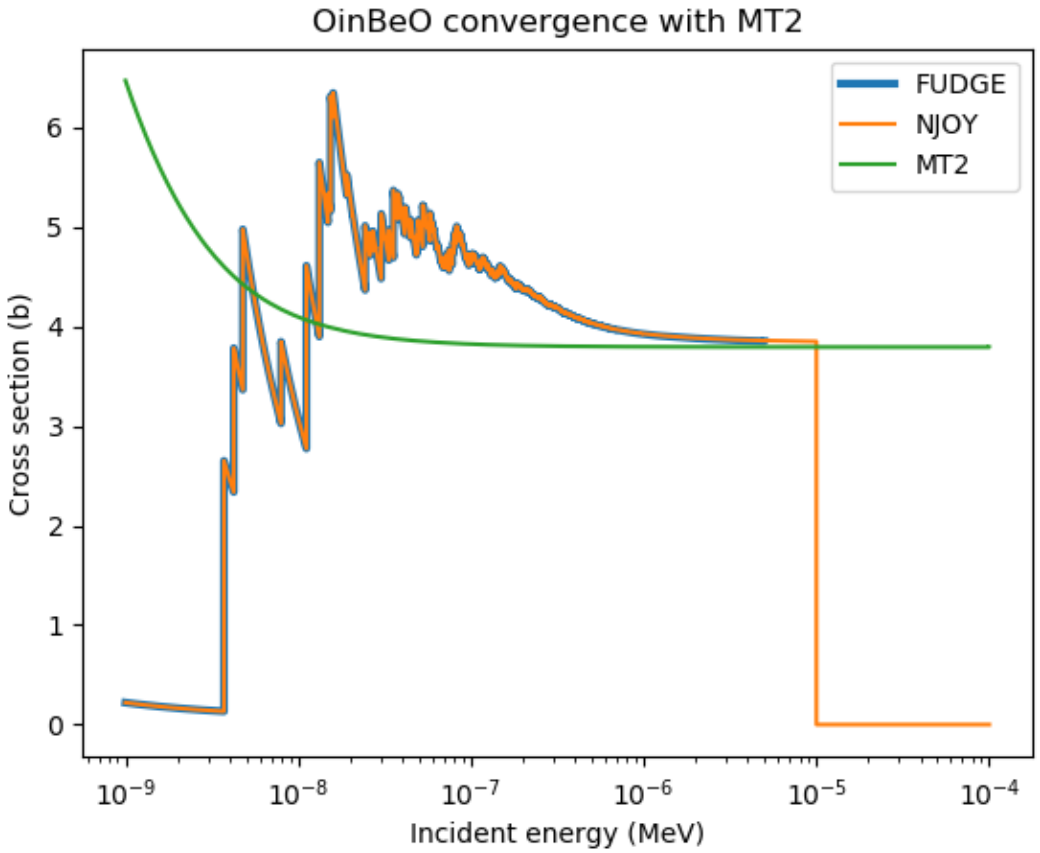
ENDF/B-VIII.1 significantly expanded the TNSL sub-library. Some new evaluations are challenging to process and use!

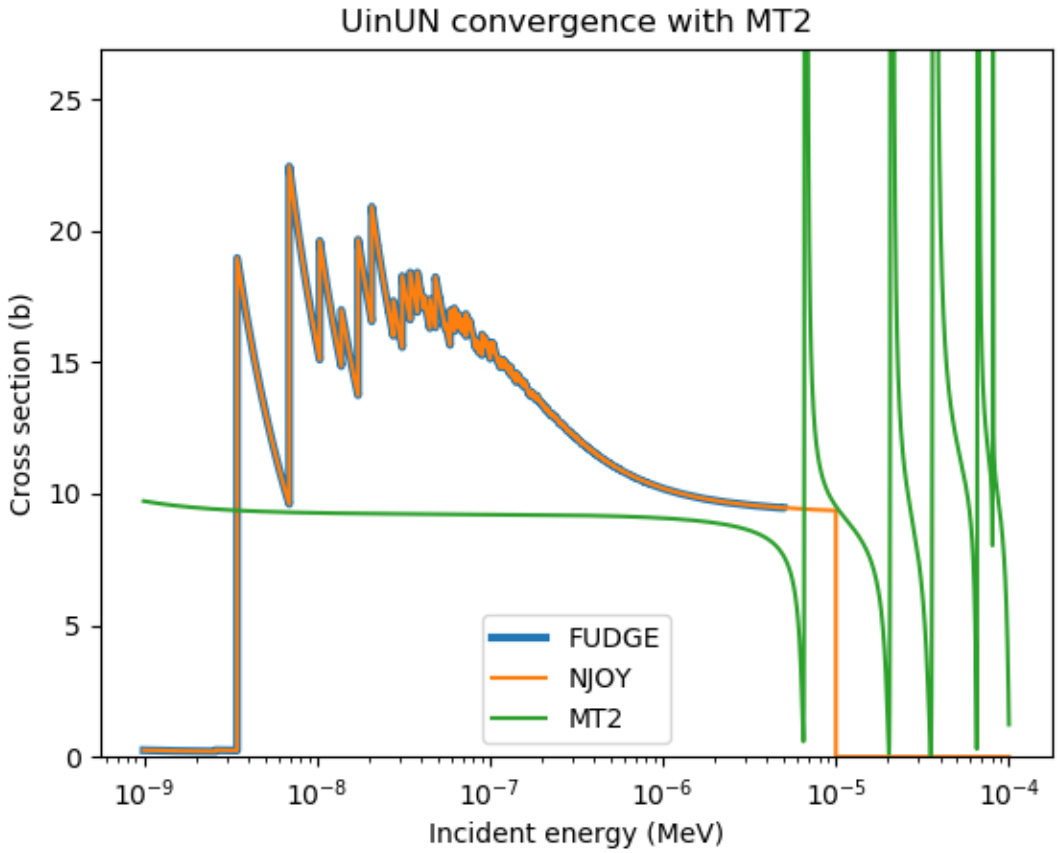
- Incoherent inelastic: new Sd evaluations for Be metal and graphite include one-phonon coherent inelastic contribution.
  - α-resonances at some values of β require changing how we extrapolate S to smaller α
  - Extrapolation choices can have an impact on the integrated cross section and a bigger impact on scattering into forward angles



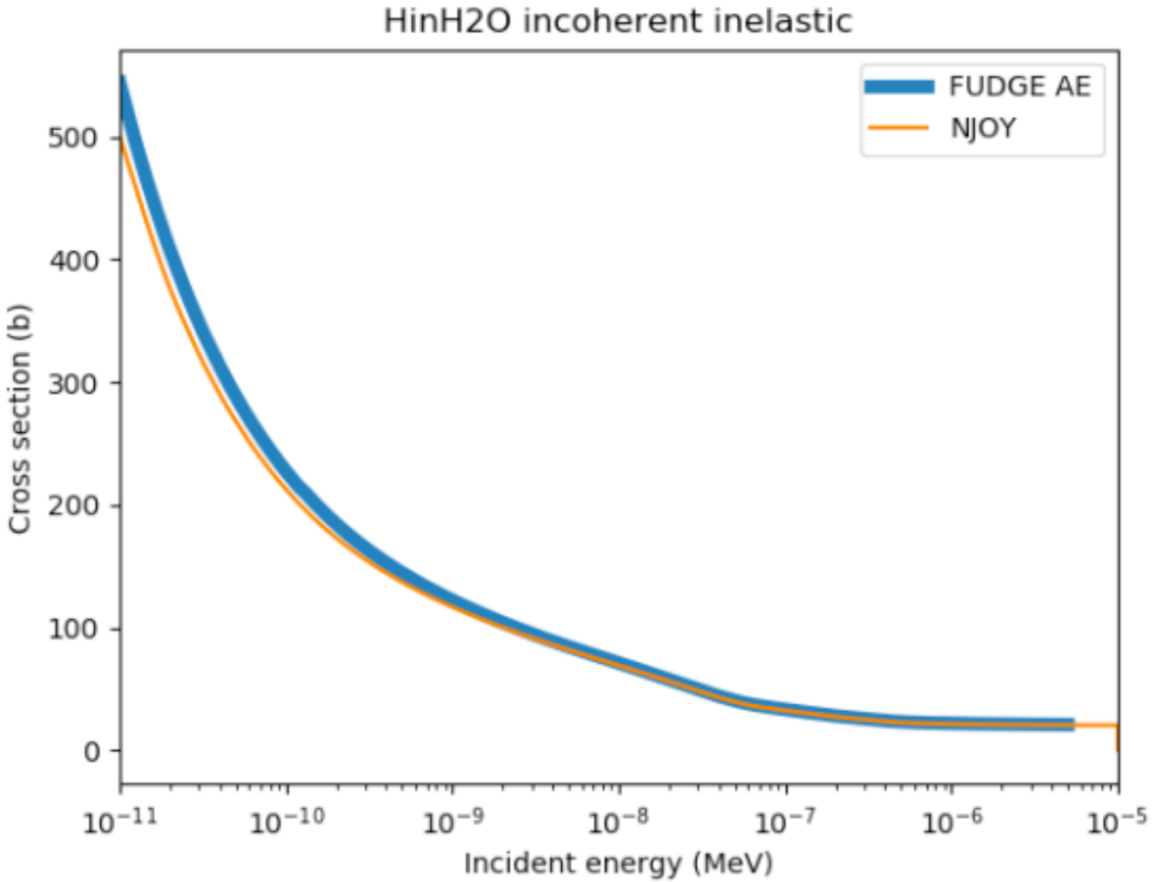
https://git.nndc.bnl.gov/endf/library/thermal_scatt/-/issues/84

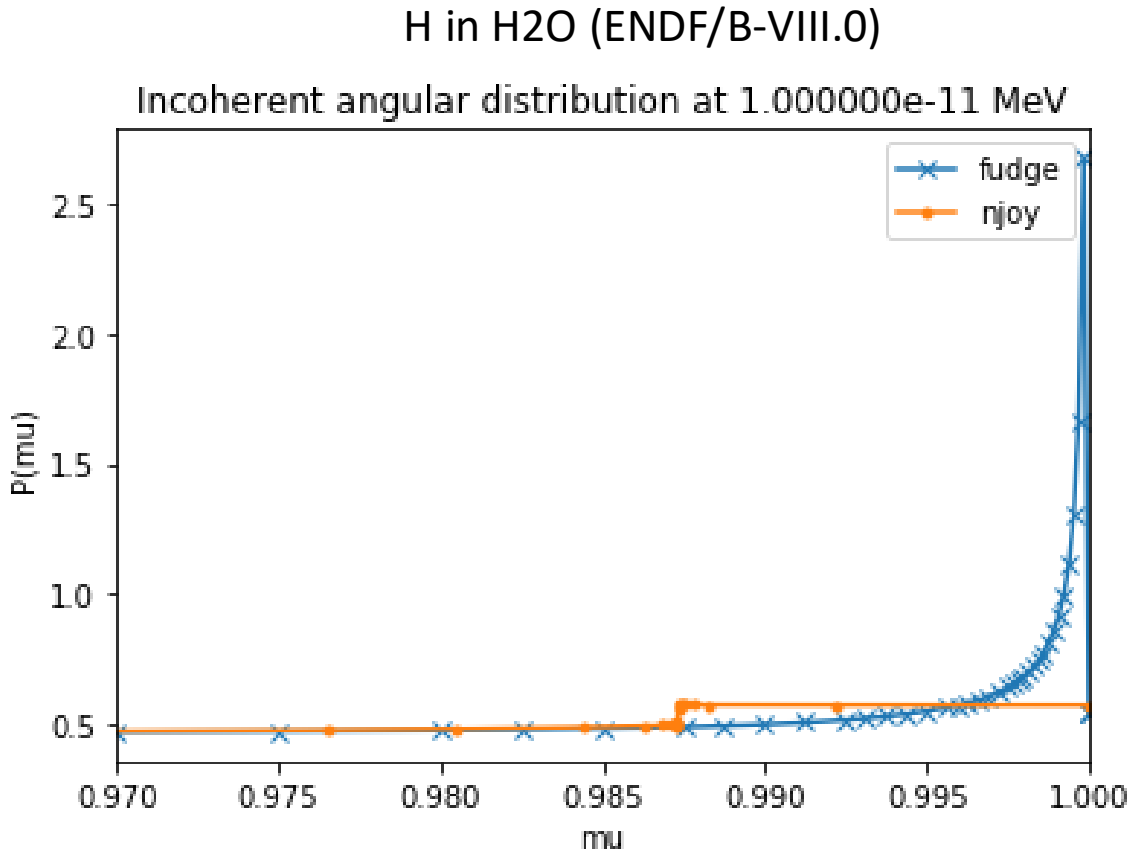
FUDGE TNSL processing results were compared extensively against THERMR: integrated cross sections





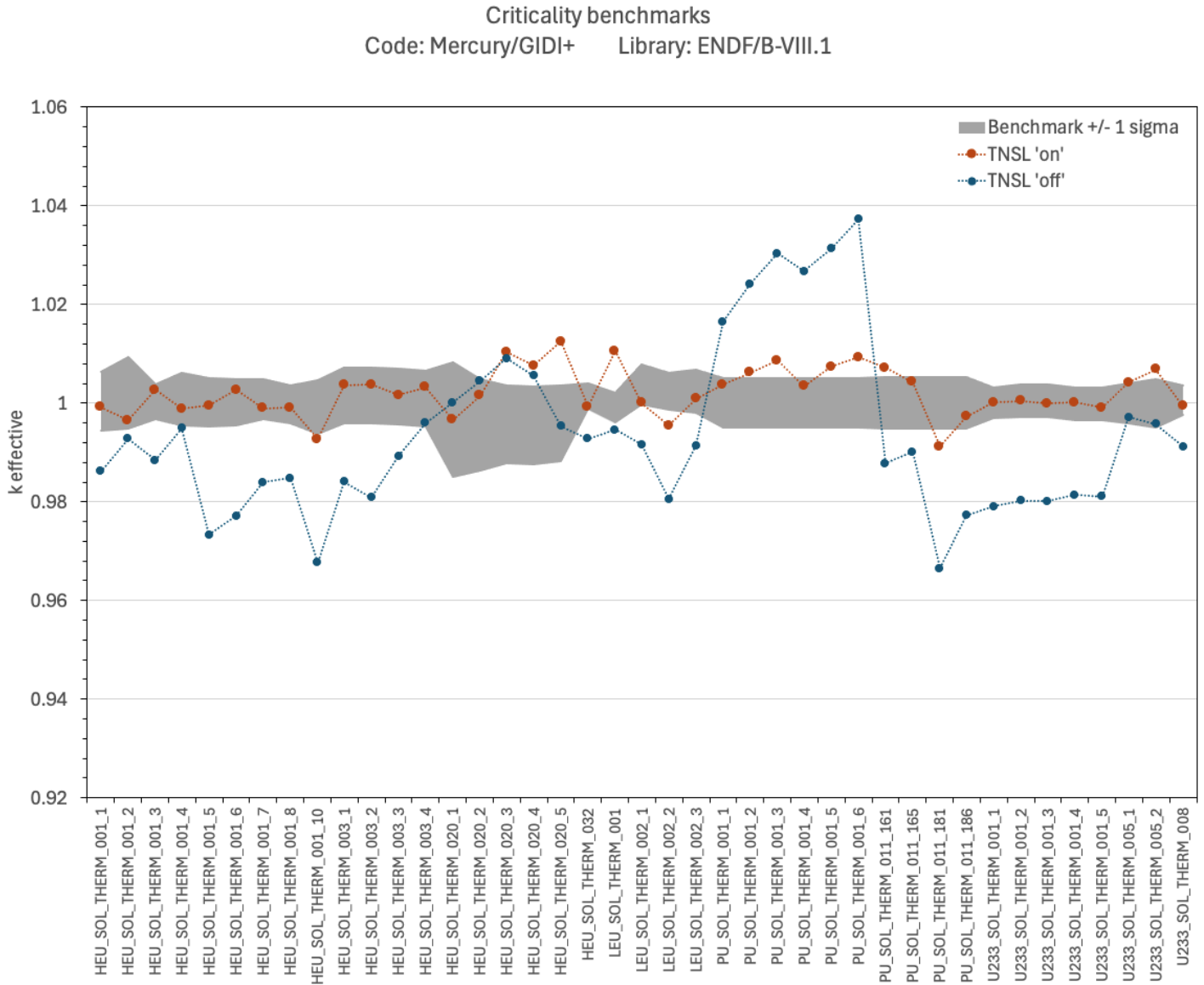
FUDGE TNSL processing results were compared extensively against THERMR: integrated cross sections and distributions





Differences at forward angles are due to different extrapolation outside tabulated S(α,β) grid

Metis (LLNL V&V test suite) was used to test FUDGE-generated TNSL data against a range of thermal solution benchmarks



The FUDGE team also supports GIDIplus: C++ API for reading and sampling processed GNDS data

- GIDIplus is open source, available from https://github.com/LLNL/gidiplus
  - Latest public release is 3.28.22, new update coming soon
- LLNL codes are updating to use GIDIplus by default
  - Effort involves extensive testing and review by Mercury (Monte Carlo) and Ardra (deterministic) transport code teams
- Older version of GIDIplus is used in the GEANT-4 package G4LEND
  - We recently updated G4LEND as part of the GRIN project. Updates expected to be part of the next G4 release

In addition to GIDI+, FUDGE supports exporting processed data to several formats

- toACE.py: export processed Monte Carlo data to an ACE file
  - Currently supports incident and outgoing neutron data, TNSL, probability tables
  - Related script toACE_multiTemperature.py for generating ACE files at various temperatures
- Legacy LLNL formats: MCF for Monte Carlo codes, NDF for SN
- Recent addition: exporting some processed data directly to COG library format

Thanks to NCSP for helping push us to implement new FUDGE capabilities!

- Now that the capabilities are available, other groups at LLNL have gotten involved in testing and are interested in further development
- We plan to take part in more URR and TNSL exercises soon, with particular focus on helping expand GNDS-related capabilities
- Continuing to work with library developers (ENDF, JEFF, JENDL, …) to make evaluations available in both ENDF-6 and GNDS


Lawrence Livermore
National Laboratory