LLNL's GND nuclear data structure and processing

16 Nov. 2011

Bret Beck and Caleb Mattoon

Physical and Life Sciences

Lawrence Livermore National Laboratory

LLNL-PRES-513408

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

Outline

- GND, new nuclear data structure development
  - Motivation
  - Design status
  - Infrastructure
    - General
    - Processing

Collaborators

Caleb Mattoon, Nidhi Patel,
Neil Summers and Gerry Hedstrom
LLNL

Dave Brown
BNL

Motivation for new nuclear data format

- Three flavors of nuclear data
  - Experimental (e.g., ExFor)
  - Evaluated (e.g, ENDF and ENDL (LLNL))
  - Processed
    - Deterministic (e.g., ndf (LLNL) and NDI (LANL))
    - Monte Carlo (e.g., mcf (LLNL) and ACE (MCPN/LANL))

The GND structure is designed to include evaluated and process data.

LLNL legacy formats overview

	Started	Format	Form
Evaluated	1960	FORTRAN Card based	Pointwise
Processed Deterministic	~ 1970	Cray? binary	Pointwise
Processed Monte Carlo	~ 1970	Cray? binary then PDB - LLNL developed	Pointwise

- LLNL legacy formats
  - Limit types of data that can be stored
    - All pointwise, no ENDF function equivalent types
  - Bloat data (ENDL), example on next slide
  - Slow to access for large files (PDB), example on next slide

LLNL is divorced from others

- ENDF was started ~1964, ~4 years after ENDL
  - LLNL has been detached from other since
- From Endf To Endl (FETE) code developed in the last 10 years
  - Some ENDF data bloats in ENDL format

    Example:

    ENDF (Kalbach/Mann) 740kB

    GND/XML 560kB

    ENDL 20 MB
  - Bloated data not always bloated enough
- PDB limitations:
  - A merge of two large pdb files took 1400 minutes

Up shot, LLNL needs new evaluated, deterministic and Monte Carlo nuclear data formats.

GND structure

- One goal, structure represent physics
  - Per reaction data
    - Cross section
    - Per product data
      - Multiplicity
      - Distribution data
      - Energy and momentum deposition data
- Defining a structure (GND) and not a format

  GND/XML, GND/HDF5, GND/?
- ~3 to 4 FTE years of development to date (design, development, translation, infrastructure, processing and C accessing)
- Supports ENDL and ENDF data types

From 3 to 1 (kind of)

- Initially, evaluated, deterministic and Monte Carlo formats differed
- Realized that most of the structure is the same only the end data are different (called **forms**)
  - Reaction
    - cross section
    - outputChannel
      - product1
      - product2
  - For example: cross section **forms** can be:
    - resonance parameters for evaluated
    - pointwise for Monte Carlo
    - grouped for deterministic

FUDGE: Infrastructure for GND

- FUDGE (For Updating Data and Generating ENDL)
  - Read, modify, plot, process and write
  - Top level written in python
    - C, C++, FORTRAN wrappers when speed is an issue
  - Converter for
    - ENDL to GND, ENDF to GND, GND to ENDF
      - Can convert all of ENDF/B-VII.1/neutron
        - Except ~6 bad evaluations
      - Can convert most of Photon and charge particle evaluations
    - GND/XML to GND/HDF5 converter
  - FUDGE is available to all
    - We have released two beta versions
    - Releasing 1.0 (see Me, Caleb Mattoon or Dave Brown)

FUDGE/GND is Object Oriented

- Methods to
  - Read/write GND
  - Plot data: many data classes will have a plot method
  - Math operations on data
  - Processing data
  - Checking data
  - etc.

FUDGE processing infrastructure

- Processing converts evaluated data into derived data or data useful for transport codes
- Derived data: calculated from existing data
  - Examples, energy and momentum deposition data
    - Energy deposition data

$$< E'(E) >= \int d\mu \int dE'\, E'\, f(E \mid \mu, E')$$

      - Good for checking product spectral data, $f(E \mid \mu, E')$, via energy balance
- Transport data processing
  - Heating cross section
  - Grouping data
  - Transfer matrices for deterministic transport

Status of processing

- Energy and momentum deposition: Done by end of Nov.
- Heating cross section: Done
- Grouping data: Done
- Transfer matrices for deterministic transport
  - Routines written C++: Done
  - Need to interface python to C++ routines:
    - 50% Done
    - Complete by beginning of 2012
  - Elastic upscatter model
    - Includes thermal motion of target.
    - Estimate completion April 2012.

C accessing routines for transport codes

- Developing access library called GIDI
  - General Interaction Data Interface

Conclusion

- We have developed a new modern nuclear data structure called GND
- We have written a lot of infrastructure for GND
- We will share what we have done with other
- We want for cooperate with others to make GND and the infrastructure better
- We would like feedback

For information on how to get GND/Infrastructure please see me, Caleb Mattoon or Dave Brown