IRSN
INSTITUT
DE RADIOPROTECTION
ET DE SÛRETÉ NUCLÉAIRE

ENDF++ - Nuclear Data Activities at IRSN

NJOY User Meeting - November 18, 2008

Wim Haeck, IRSN, wim.haeck@irsn.fr

Joachim Miss, IRSN

Nuclear data activities at IRSN

- **No “pure” nuclear data but activities with strong ties to nuclear data**

- **Experimental programs: data & code validation**
  - Fission product experimental program
  - Plutonium temperature effect experimental program
  - Structural materials experimental program

- **Development of calculation tools: nuclear data related developments**
  - VESTA: a generic interface for depletion calculations
  - ENDF++: nuclear data manipulation
  - MORET: Monte Carlo calculations

VESTA

- **A generic Monte Carlo evolution interface**
  - Provide a generic software framework for depletion calculations
  - Highly customisable to a user's needs
  - For more information: http://promethee.irsn.org//doku.php?id=tmp:vesta

- **VESTA has very specific nuclear data needs like any depletion code**
  - Cross section data read from ENDF files generated by NJOY which are also used by MCNP(X)
  - ENDF libraries contain additional data needed for depletion:
    - Fission yield data and radio-active decay data
    - Branching ratios for the production of isomeric states
    - Fission Q-value data
  - Tools to easily manipulate data are required: ENDF++

ENDF++: C++ software for nuclear data manipulation

- **Under developed in the framework of our main activities**
- **All the advantages of object oriented design**
  - Modularity, generality and reusability
  - Optimisation and maintenance
  - They can function as a standalone application or can be easily integrated into other dedicated applications
  - High level design with low level interface

- **ENDF++ is currently:**
  - An ENDF parser to read/format/write nuclear data
  - A number of function objects or functors:
    - Linearisation of point wise data
    - Integration of linearised functions
    - Unionisation of linearised functions
  - Dedicated classes for resonance reconstruction: SLBW, MLBW and AA

Parsing ENDF files with ENDF++

- **Read, write and extract ENDF data**
  - MF1: fission neutron yield, fission Q-values, etc.
  - MF2: resonance data
  - MF3/MF9/MF10/MF23: cross section and multiplicity data
  - MF4: angular distribution data
  - MF7: thermal neutron scattering data
  - MF8: radioactive decay data, fission yield data, etc.

- **Format nuclear data in the ENDF format**
  - MF1: fission neutron yield, fission Q-values, etc.
  - MF3/MF9/MF10/MF23: cross section and multiplicity data
  - MF4: angular distribution data (e.g. JEF/DOC 1255)
  - MF7 MT4: S(α,β) data (e.g. JEF/DOC 1213)

- **Easily extendable to other data contained in the ENDF file**

Parsing ENDF files with ENDF++

- **The parser is currently mainly used inside VESTA, to read and manipulate the required nuclear data for a depletion calculation**
- **Other current & possible future applications for this parser will include:**
  - Quality Assurance for basic evaluations
    - Test for formatting errors
    - Test for inconsistencies between various sub-libraries (e.g. LIS/LISO in neutron and decay sub-libraries)
  - Help in automated NJOY processing
    - Checks important parameters for nuclide tailored NJOY input
    - E.g. the temperature, whether or not the nuclide has resolved/unresolved resonances, etc.
  - Quality Assurance for processed evaluations
    - ENDF format is used by NJOY to pass on data between modules
    - Provide testing for undetected reconstruction errors (e.g. JEFF 3.1 ^{58}Co capture)
  - Merging data together into a single ENDF tape
  - Etc.

Functors within ENDF++

- **Functors can be developed independently with relative ease in little time**
  - Tasks should be split up as much as possible
    - A functor that integrates data should not have to linearise it
  - Nuclear data can be a difficult subject but the individual subtasks are not
  - Anybody can contribute!

- **Functors should have a basic, simple and *generic* interface**
  - Only low level data is manipulated: integers, real values, arrays, etc.
  - No “advanced” concepts or links with existing data formats

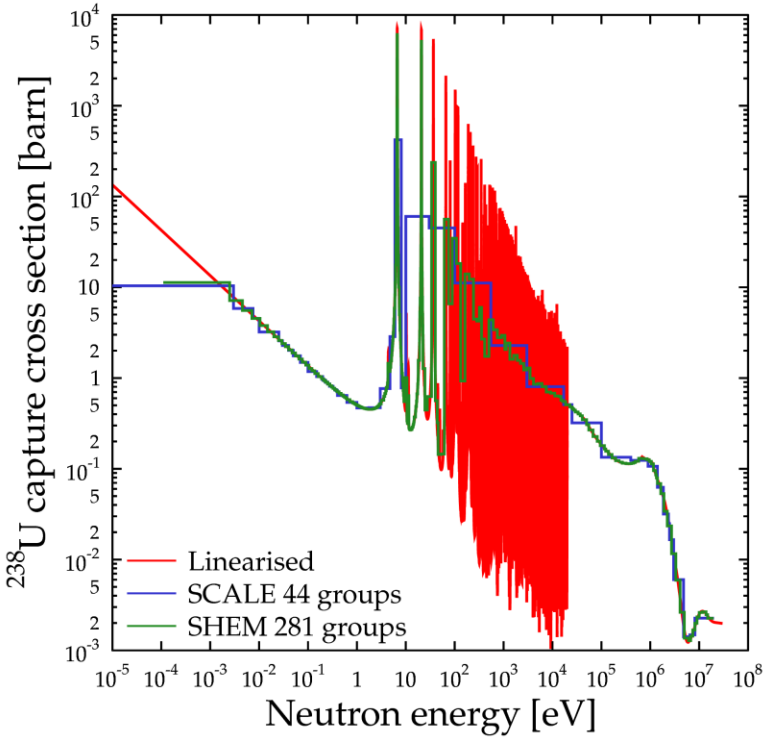
- **Great optimisation potential**
  - Multiple implementations are possible and the best can be chosen
  - Gains the potential of automatic testing and validation

Functors within ENDF++

- **One example: integrating linearised data**

$$f_g = \int_{E_{g-1}}^{E_g} f(E)\,dE$$

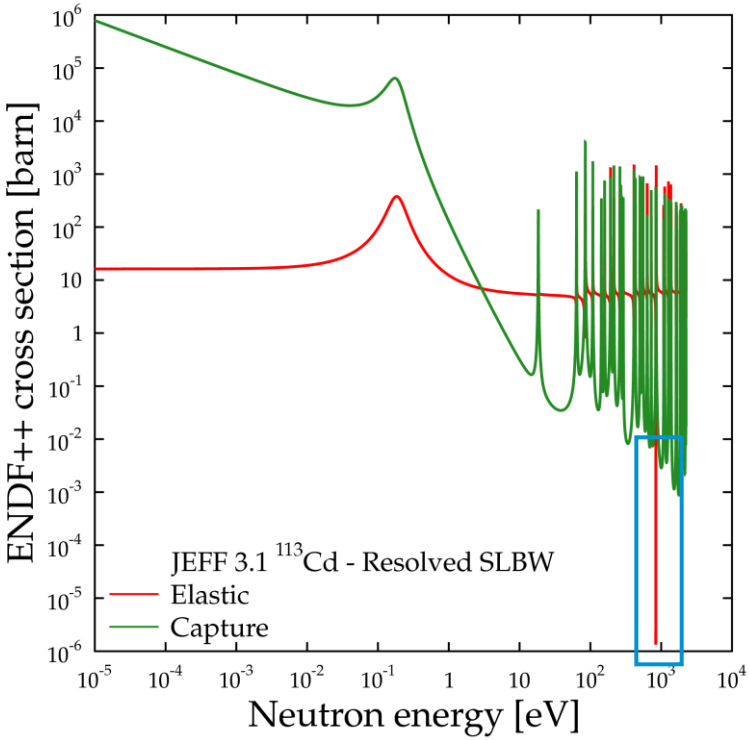
$$f_g^h = \int_{E_{g-1}}^{E_g} h(E) f(E)\,dE$$



- For simple integrals
- For calculating group cross sections
- For weighted integrals:
  - isomeric production branching ratios
  - Weighted multi-group cross sections

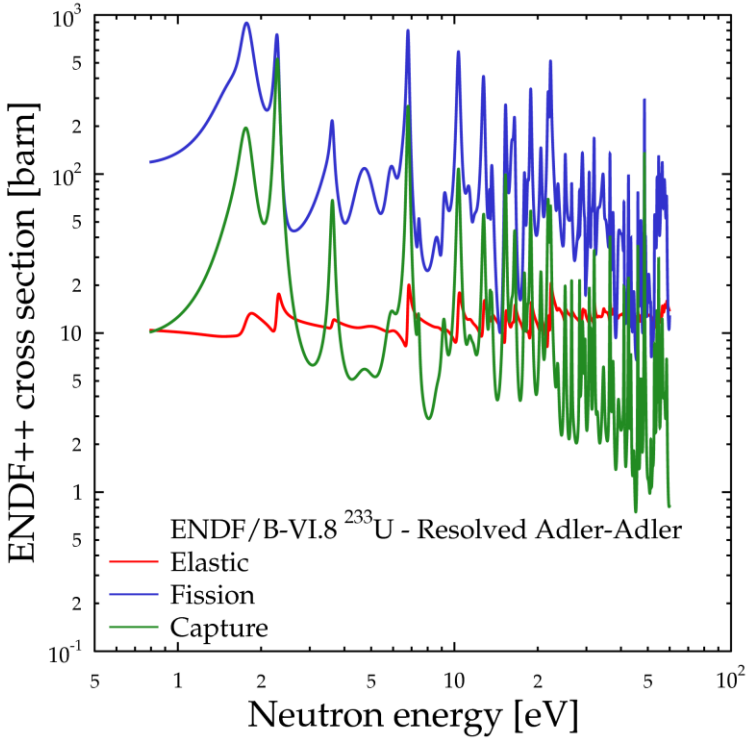
Resonance reconstruction

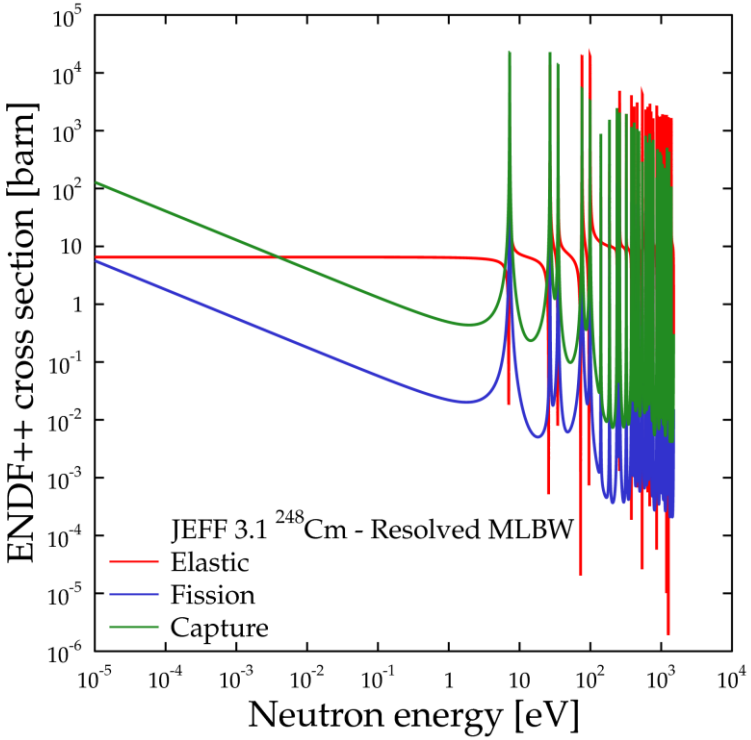
- **The continuous energy version of MORET5 uses a "universal" energy grid**
  - This leads to an important decrease in the code's CPU time
  - But: the energy grid needs to be optimised to avoid memory problems
- **One solution would be to optimise this grid already at the level of cross section reconstruction**
- **Current ENDF++ resolved resonance reconstruction capability:**
  - Single Level Breit Wigner
  - Multi Level Breit Wigner
  - Adler-Adler



Resonance reconstruction

- **We're currently testing these resonance reconstruction objects:**
  - Comparison with NJOY 99.259 and PREPRO 2007 are underway





Conclusion and outlook

- **ENDF++ provides us with a software framework:**
  - For nuclear data testing and Quality Assurance
  - For easy access to data in the ENDF format
  - For “basic” operations on that data

- **Ultimately, ENDF++ will contain all the basic building blocks for a more complicated task like cross section reconstruction:**
  - Reading and extracting the basic data
  - Reconstructing resonance data
  - Linearising the data
  - Thinning/optimising the resulting energy grid
  - Doppler broadening

- **We’re favorable towards free distribution of these tools and/or collaborative development with interested parties**