AMPX Status Report

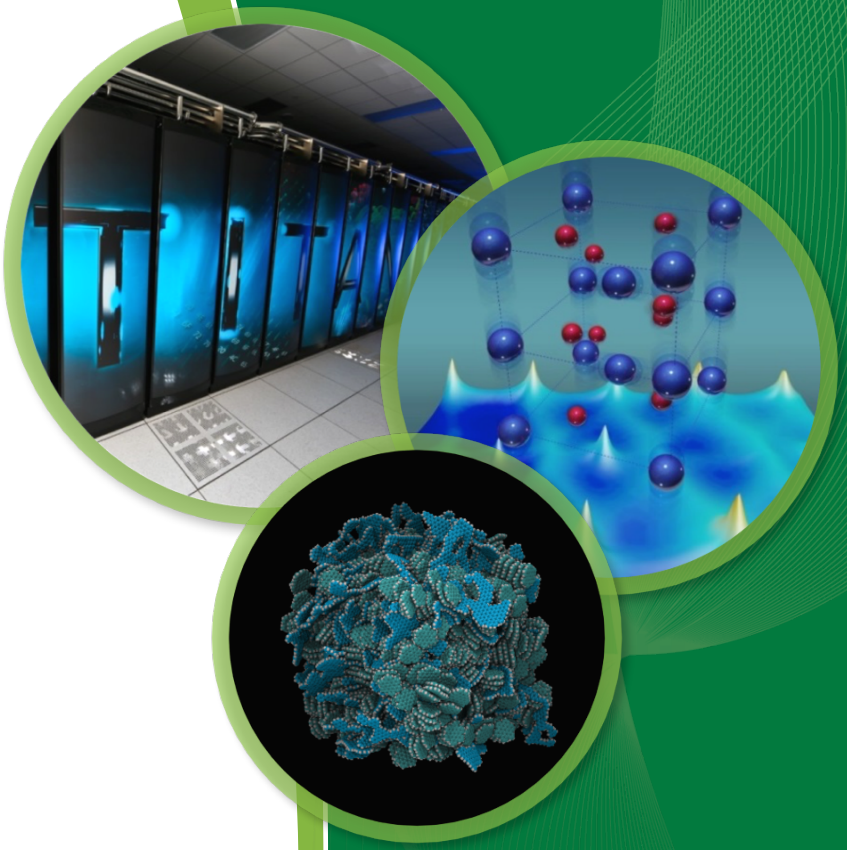
D. Wiarda

A. M. Holcomb

M. E. Dunn

C. Celik

M. L. Williams



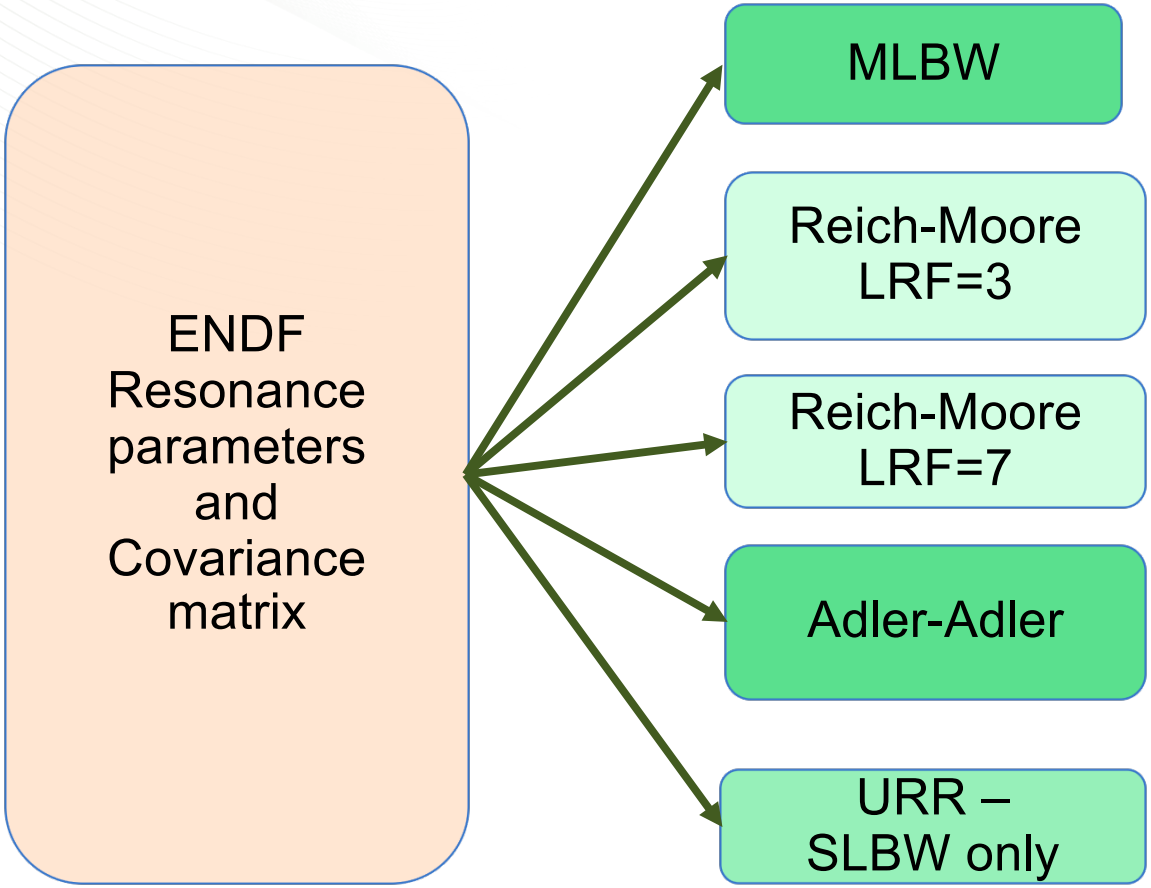
ORNL is managed by UT-Battelle
for the US Department of Energy

OAK RIDGE National Laboratory

Introduction

- Update to infrastructure API classes
  - Resonance parameter API
  - API to group-average derivatives
  - API to determine energy grid in RR and URR
  - In-memory COVERX resource
- Update to PUFF – code to process covariance matrices
- Update to POLIDENT – code to process 1-D cross section data
- Process ENDF/VIII.0-β data

Resonance API

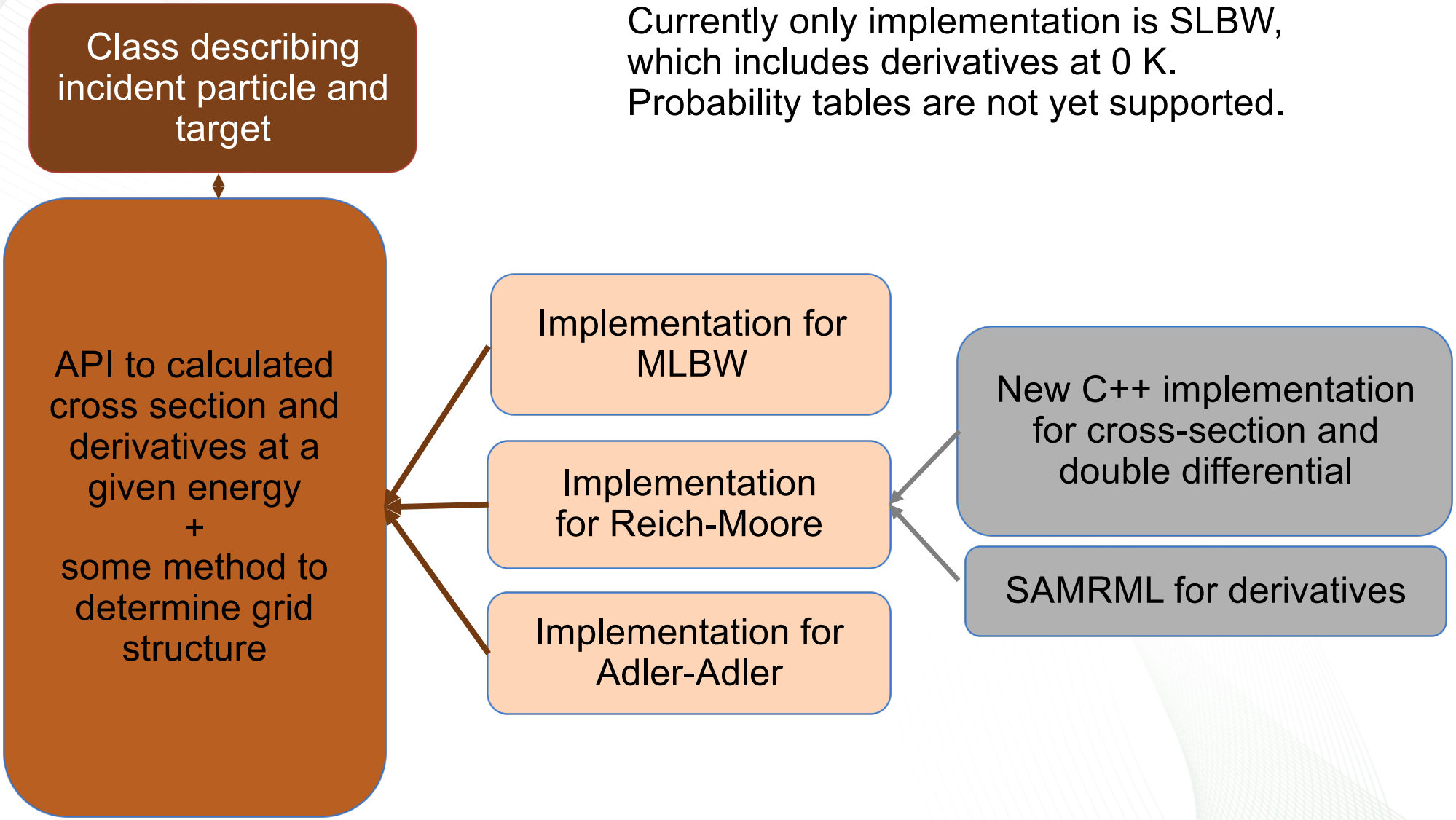


- SLBW and MLBW parameter are stored in the same class with a flag indicating which formalism to use
- Resonance parameters for Reich-Moore for LRF=3 are initially stored in a different class, but are converted to a LRF=7 class before calculation
- If derivatives are desired, all formalisms (except URR) are converted to LRF=7 so SAMRML can be used under the hood
- All resonance parameter classes can contain a covariance matrix. If converting to a different formalism, the covariance matrix is re-organized accordingly

A GND reader will be added to read the resonance parameters into memory

Resonance API classes

An API also exists for the URR.
Currently only implementation is SLBW, which includes derivatives at 0 K.
Probability tables are not yet supported.



API to calculate group averaged derivatives

- Read the ENDF data into the in-memory resonance parameter class
- A helper function returns an implementation of the resonance API class based on the resonance parameters passed in
- An implementation of the AMPX integration routine (using fourth-order Runge–Kutta) with the function that returns values at a given E overridden.
- PUFF calls the above implementation with a flux and implementation of the resonance API class
- Apply the sandwich formula to get the covariance matrix with respect to the cross section on the user desired group structure.

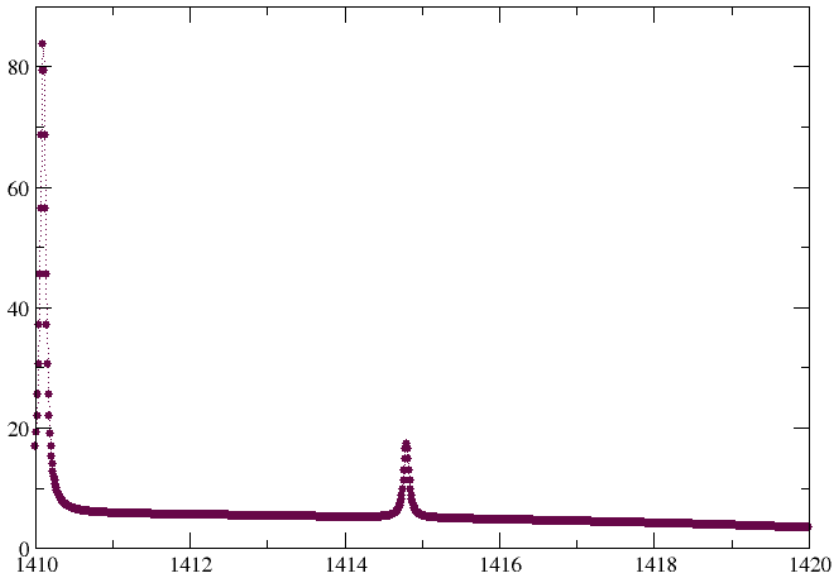
API routines that determine grid in RR

Since the routine uses the Resonance API it is independent of the resonance formalism. All routines have been updated to use C++

Determine an initial grid:

- Divide the energy range into intervals of 10 eV, but increase interval size until at least one resonance is captured in the interval.
- Determine:
  - Average level spacing <D> in interval – function in Resonance API
  - Average neutron width $\langle\Gamma_n\rangle$ in interval – function in Resonance
  - Pick initial step size

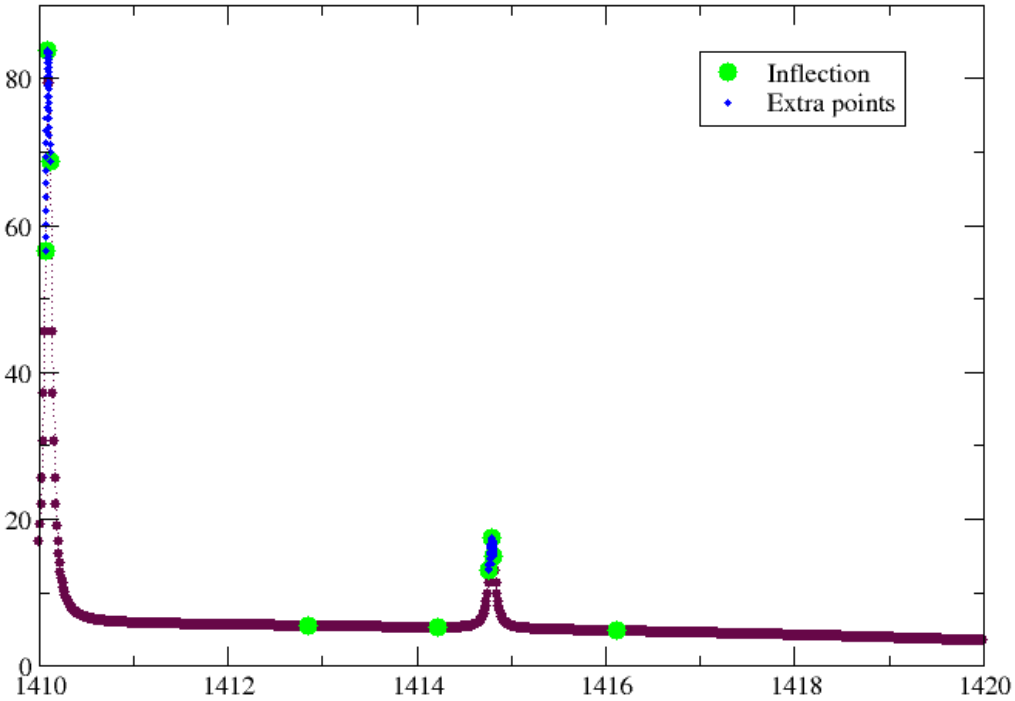
$$0.1\sqrt{\frac{\langle\Gamma_n\rangle}{\langle D\rangle}}$$



^{156}Gd Total
<D>= 4.7 eV
$\langle\Gamma_n\rangle$=9.825E-4

Calculate first and second derivative numerical from points given

- Add inflection points - first derivative changes sign
- Add minima and maxima – second derivative changes sign



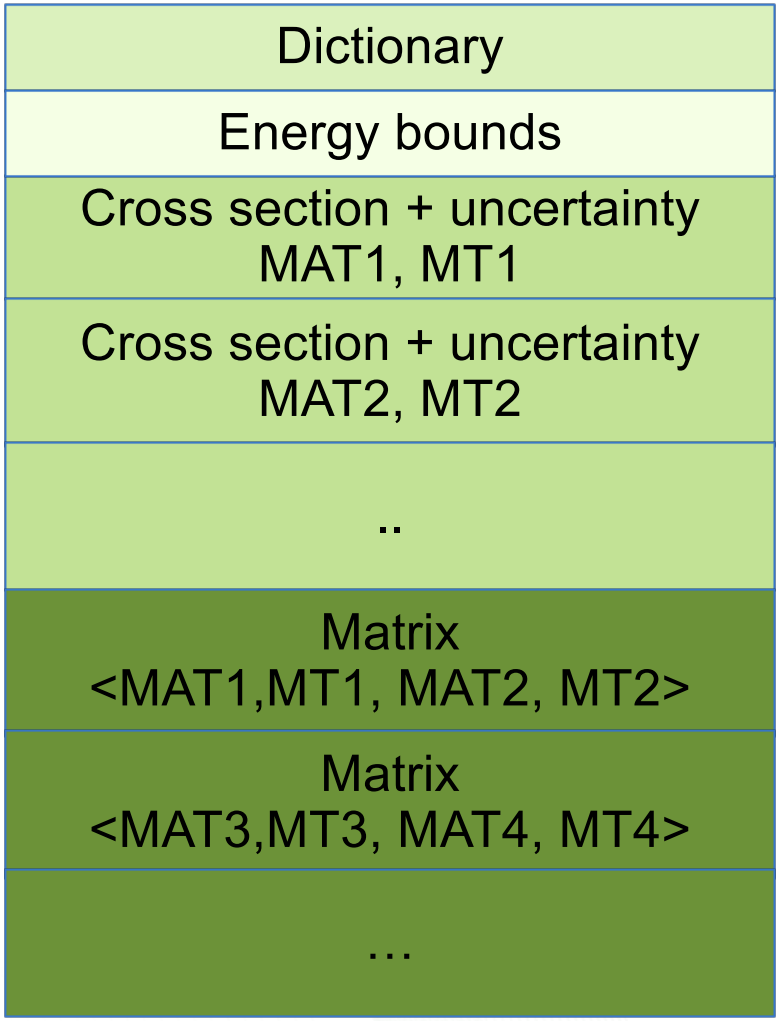
Add additional points between maxima and inflection points to either side. Points are spaced equidistant from one another in that range, number of points (10, 15, or 20) is determined by an empirical formula.

Final step:
Add additional points by a halving scheme and thin the data with a user supplied epsilon.

Note: If data are too dense (i.e. more than 1500 points per interval), additional energy points are only added if they are not different in single precision

In-memory resource for COVERX data

- Data are read into memory
- File format (ascii, binary) is automatically detected, endianness no longer matters
- Methods to retrieve, add, delete cross section and matrices by material and reaction or by index
- Methods to convert between absolute and relative uncertainties and matrices
- Methods to convert to correlation matrices
- Class is written in C++ with Fortran bindings.
- Writer class that saves the data in ascii or binary format to disk



Updated PUFF

- Resonance parameter covariance matrices use the resonance API
- New keyword based user input (old input is still supported)
- Better error reporting (done in the ENDF reading level)
- More output information
- Add redundant cross section if not present in ENDF
- Use the COVERX resource throughout the code to handle all temporary covariance matrices
- Add additional unit tests and keep old regression tests.
- Easier handling of cross-material covariance data

PUFF input

- PUFF takes cross section data from:
  - Point-wise cross section data (group averaged cross sections are calculated on union grid)
  - MG cross section library (cross sections are extended to the union grid as needed)
  - COVERX file from a previous run(cross sections are extended to the union grid as needed)
- PUFF takes flux from:
  - point-wise flux data (group averaged flux is calculated on union grid)
  - group-averaged flux data (extended group averaged cross sections are calculated on union grid)
- Covariance matrices for materials for which correlations are given in ENDF
  - COVERX file from a previous run for the cross material
    (Previously a special file containing uncertainties was saved by PUFF)
- If indicated in the input, PUFF prints out all the ENDF covariance matrices on the evaluator grid in more human readable format than ENDF.

COVERX comparison program (COVCOMP)

- Convert all covariance matrices to correlation matrices
- Convert all uncertainties to relative uncertainties

This gives a smaller dynamic range of allowable parameters

Report errors for this stage if:

- Conversion is not possible due to missing data
- Duplicate matrices or cross section exist

Compare the data and report differences:

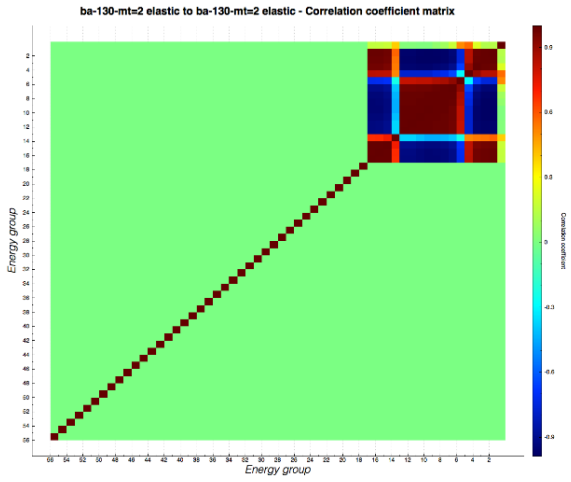
- Cross section data or matrix data only exist on one of the two files
- Relative difference in cross section is larger than a user supplied epsilon (eps)
- Relative difference in cross section uncertainty is larger than a user supplied epsilon (eps)
- Absolute difference between correlation is larger than a user supplied epsilon (corr)

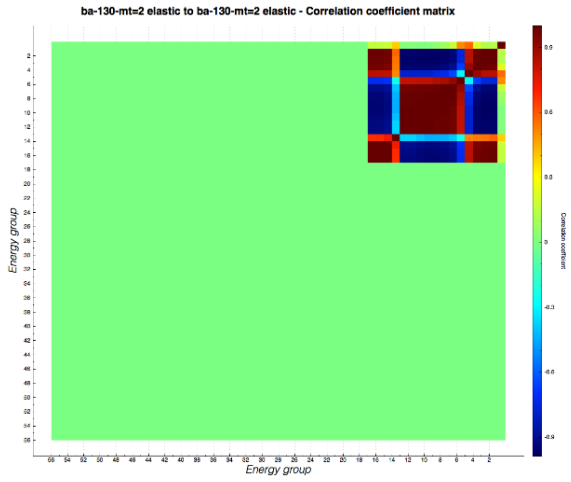
Can be called via an API or as a module.

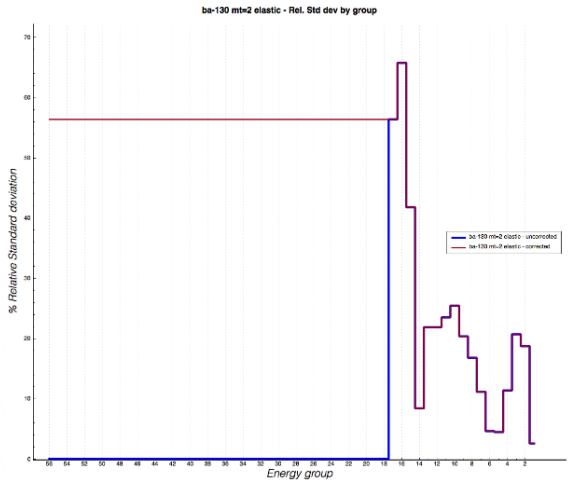
Used for PUFF regression tests.

COVERX correction program (cognac)

- Convert all covariance matrices to correlation matrices, convert all uncertainties to relative uncertainties, report errors if conversion is not possible
- SCALE only requires a smaller set of reactions, so we only keep covariance matrices and cross section data for those (MT= 1, 2, 4, 16, 18, 102, 103, 104, 105, 106, 107, 452, 455, 456, 1018). This is a user input.
- Set all relative uncertainties that are larger than 1 to 1. This mostly happens near the threshold due to numerics.
- Set all correlation values smaller than -1 to -1 and all correlation values larger than 1 to 1.
- If relative uncertainty is zero, but cross section is not zero, extend uncertainty to lower energies (diagonal elements only)







OAK RIDGE National Laboratory

POLIDENT (point-wise 0K data) update

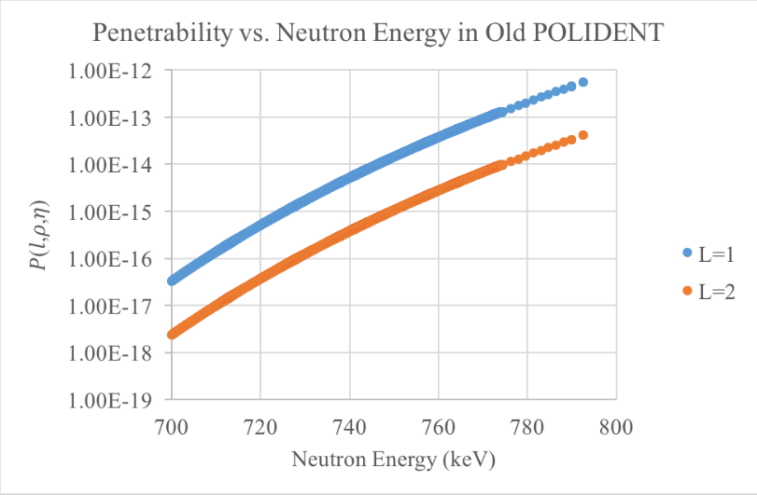
- POLIDENT was updated to take advantage of all the new API code:
  - Use the new C++ ENDF reading routines
  - Use the new resonance API
  - Use the API to determine the gridding in the RR and URR
  - Use templated classes to combine File 2 and File 3 data
- New keyword based user input (old input still supported)
- Use the new SCALE sequence/module architecture
- More consistent handling of discontinuities at the end of the RR and URR.
- Better error reporting

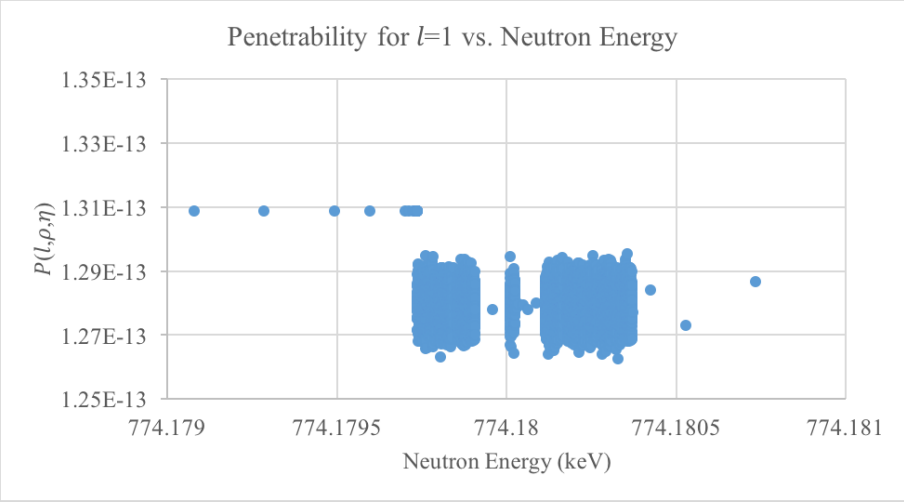
Numerical instability in the Coulomb function implementation

For incident neutrons, the expressions for the penetration factor are analytic, smoothly-varying functions of energy.

For incident charged particles, the penetrability must be calculated from approximate solutions to the Coulomb wave equation. When the Coulomb repulsion is very large in comparison to the effective incident energy, the computation of the wave functions, and thus the penetrability, becomes unstable: small changes in the incident neutron energy will lead to unphysical changes in the penetrability.

Due to using a halving scheme in determining the grid, POLIDENT could not determine the energy grid for the new Ca40 evaluations:





OAK RIDGE National Laboratory

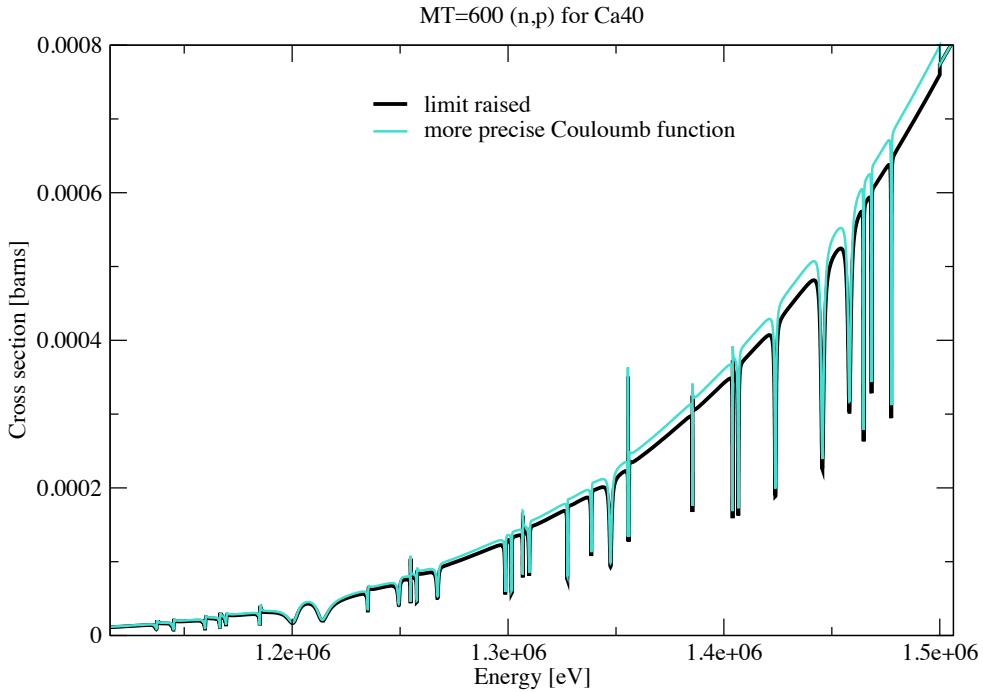
The instability is present in POLIDENT, SAMRML and SAMMY:

- Currently a different algorithm is used in different ranges for the Coulomb function
- In SAMMY an input flag can select the more precise algorithm over the whole range.

For the version of POLIDENT released with SCALE 6.2 and 6.2.1 we changed the limit for the use of the more precise algorithm. This allows processing of Ca40

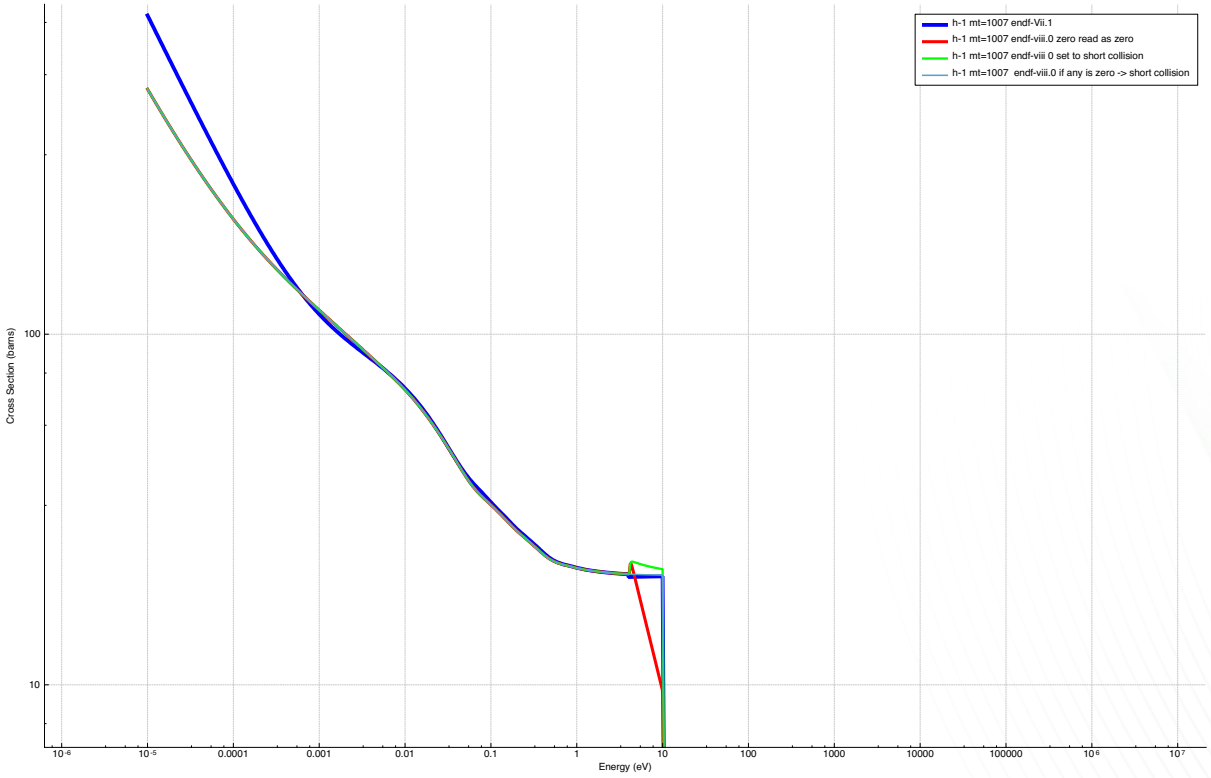
For the modernized version of POLIDENT, we use a more precise algorithm:

N. Michel, “Precise Coulomb wave functions for a wide range of complex ℓ, η and z,” Computer Physics Communications, Volume 176, Issue 3, 1 February 2007, Pages 232-249, ISSN 0010-4655, http://dx.doi.org/10.1016/j.cpc.2006.10.004



H in H_2O in ENDF/B-VIII.0-beta2

- $S(\alpha,\beta)$ in the ENDF files are tabulated as a function of α and β.
- $S(\alpha,\beta)$ values can be very small. Therefore the precision achievable in the ENDF file is not high enough – thus values printed as 0, are not actually 0.
- ENDF/B-VIII.0-beta2 extended the range to 10eV, making the above problem more pronounced.
- We updated Y12, the code that processes $S(\alpha,\beta)$ into double differential distributions, to use the short-collision approximation if any of the values we interpolate in is 0. This is the same approach that NJOY takes.



ENDF/B-VIII.0 processing

- ENDF/B-VIII.0-β2 and ENDF/B-VIII.0-β3 were processed with AMPX
- Some inconsistencies in ENDF/B-VIII.0-β2 were reported to NNDC and fixed by NNDC.
- Results are presented in Mark Williams talk.

AMPX modernization
What has been accomplished

- Modernization of resonance parameter processing
- Modernization of ENDF reading (point-wise covariance matrices for cross section data with respect to energy can be read and processed from GND file generated by fudge)
- Modernization of PUFF-IV
- Modernization of codes to compare and correct covariance libraries.
- Modernization of POLIDENT

AMPX modernization
What is planned for the future

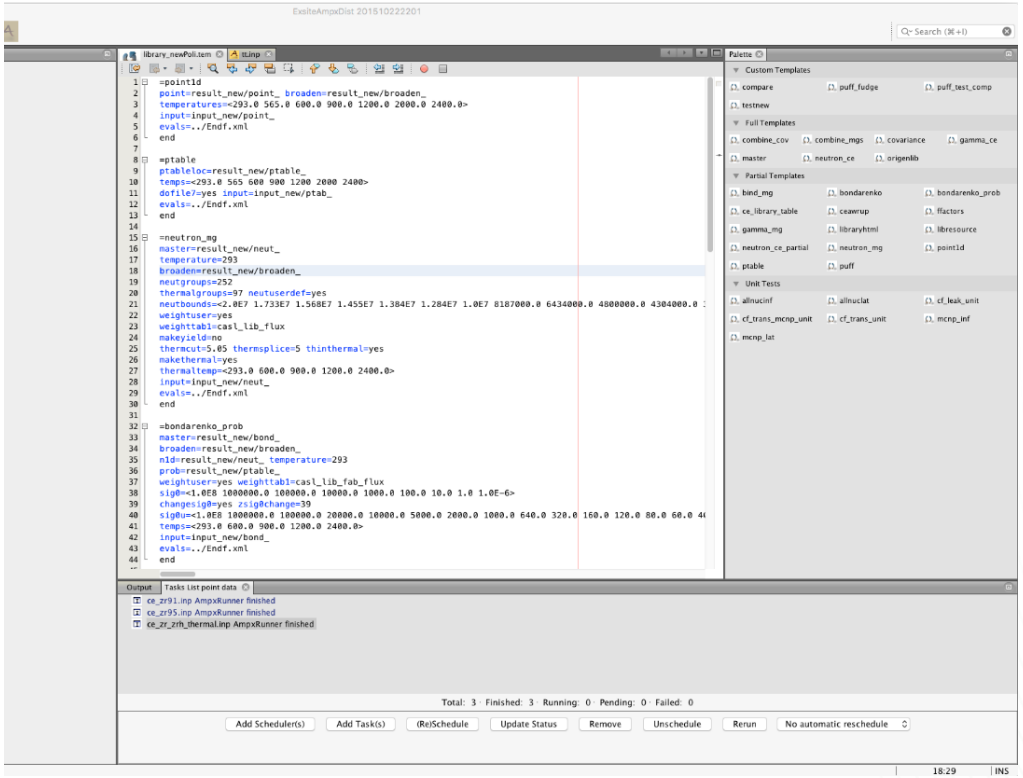
Module	Functional Capability	Estimated Completion Data
JAMAICAN	Produce CE collision kinematics PDFs and CDFs for Monte Carlo libraries	6 month
PLATINUM	Assemble CE library for Monte Carlo calculations	12 month
BROADEN	Doppler broaden 1D CE data	24 month

PLATINUM and JAMAICAN will use an in-memory resource for CE Monte Carlo libraries shared between SCALE and AMPX

Work will continue on implementing the use of the GND format.

AMPX is available from RSICC

- AMPX sources are included in the SCALE 6.2 release
- Pre-compiled AMPX is available with SCALE 6.2.1 release
- Includes the GUI to expand templates and a directory with sample templates
- Scale 6.2.1 includes the new version of PUFF-IV
- AMPX documentation is available online at:
  http://scale.ornl.gov/Publications/AMPX-6.pdf



Summary

- New C++ infrastructure API classes were added:
  - ENDF reading of RR and URR data
  - Resonance processing
  - Grid creation in RR and URR
  - Group-averaging of resonance parameters with respect to energy
- Update PUFF – code to process covariance matrices
- Update POLIDENT - code to process 1-D cross section data
- AMPX is available within the SCALE 6.2 and SCALE 6.2.1 package from RSICC