AMPX and SAMMY Status Report

Dorothea Wiarda
Andrew M. Holcomb
Jesse Brown
Goran Arbanas
Chris W. Chapman
Kang Seog Kim

CSEWG
Brookhaven
November 30, 2020

GNDS and AMPX

- GNDS support in AMPX has three layers:
  - Low level access to XML file, Element and Attribute access classes.
  - A base Container class that also handles access to external files.
  - Classes for Data container classes, like Values, Array and Table.
- GNDS layer classes generated from the JSON description files using a python program.
- A layer that fills AMPX in-memory classes from the GNDS layer classes.

Added classes to the last layer to fill kinematic data and started comparing the results for the ENDF-B/VIII library in GNDS 1.9 format.

F-Factors beyond Narrow Resonance

Use modules PMC and CENTRM to calculate a shielded cross section, with a CE flux for a homogenous model

$$\left(\sigma_t^{(j)}(E,T)+\sigma_0\right)\varphi(\mathrm{E},\sigma_0,T)=\int_E^{E/\alpha^{(j)}}\frac{\sigma_s^{(j)}(E',\mathrm{T})\varphi(E',\sigma_0,\mathrm{T})}{(1-\alpha^{(j)})E'}dE'+\sigma_0\int_E^{\infty}\frac{\varphi(E',\sigma_0\,\mathrm{T})}{E'}dE'$$

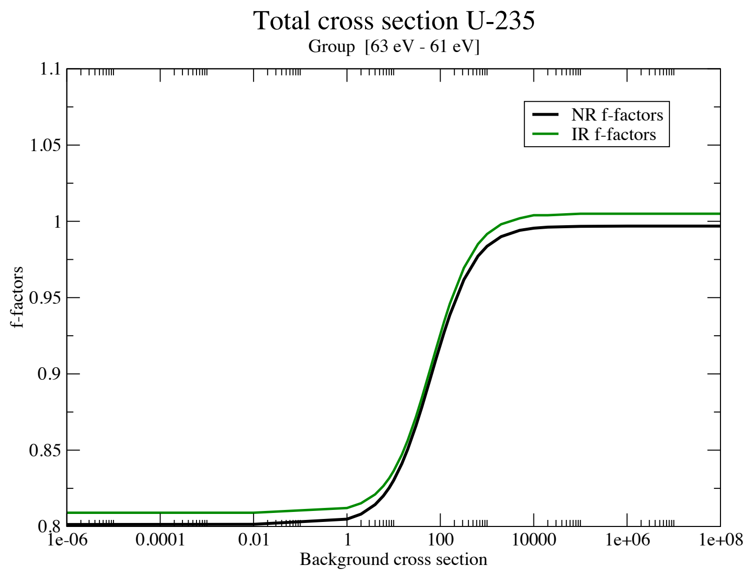
$1-\alpha^{(j)}$ = the maximum fractional energy loss in an elastic collision with nuclide *j*;

$\sigma_0=\frac{N^{(H)}}{N^{(j)}}\sigma_p^{(H)}$ (the background cross section);

$\sigma_p^{(H)}$ = the potential cross section for hydrogen;

If the desired scattering nuclide is not fissionable, a small amount of U-235 is added to the homogenous model

All our MG libraries have this for Z>= 40



Open slide master to edit

Heterogeneous f-factors (ROUX)

Similar to homogeneous f-factors, but:

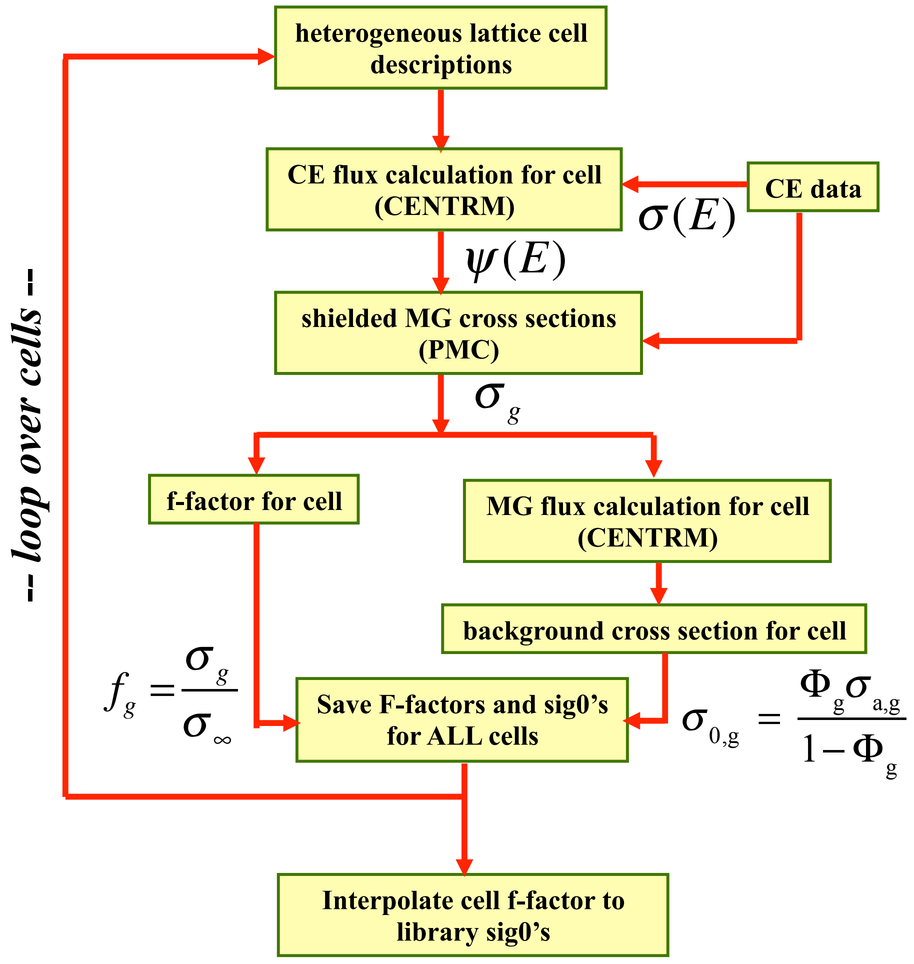
- Heterogenous models are now used
- Background cross section can not be determined in advance
- A suite of predefined models is used

Our current version:

- Uses an early (unstable) version of XSPROC
- Requires to give every cell used explicitly
- Has no checks on whether the range is covered

In progress:

- Made the code more modular and C++, so that it is easier to use different shielding codes
- Better user input, cell parameters (pitch, number density) can be drawn from a distribution



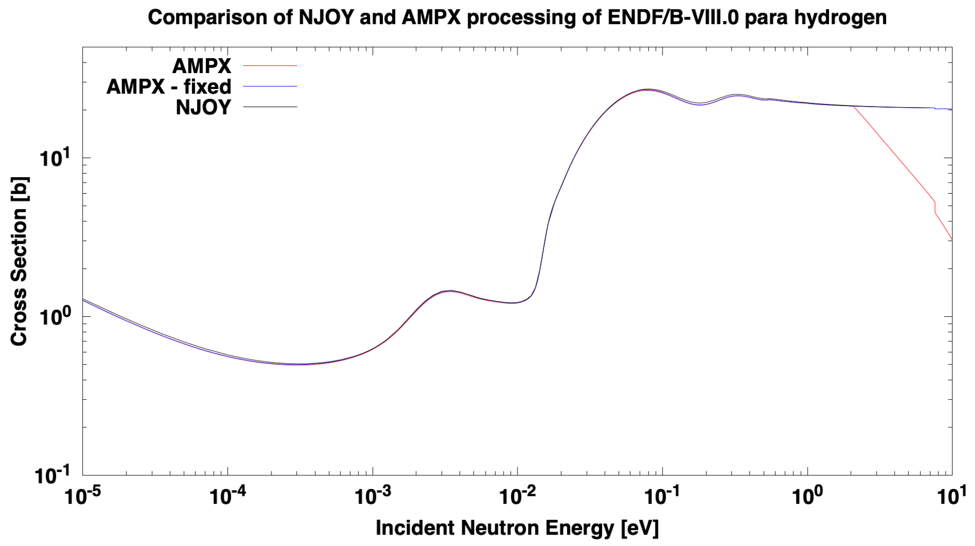
-- loop over cells --
heterogeneous lattice cell descriptions
CE flux calculation for cell (CENTRM)
CE data
$\sigma(E)$
$\psi(E)$
shielded MG cross sections (PMC)
σ_g
f-factor for cell
MG flux calculation for cell (CENTRM)
background cross section for cell
$f_g = \frac{\sigma_g}{\sigma_\infty}$
Save F-factors and sig0's for ALL cells
$\sigma_{0,g} = \frac{\Phi_g \sigma_{a,g}}{1-\Phi_g}$
Interpolate cell f-factor to library sig0's

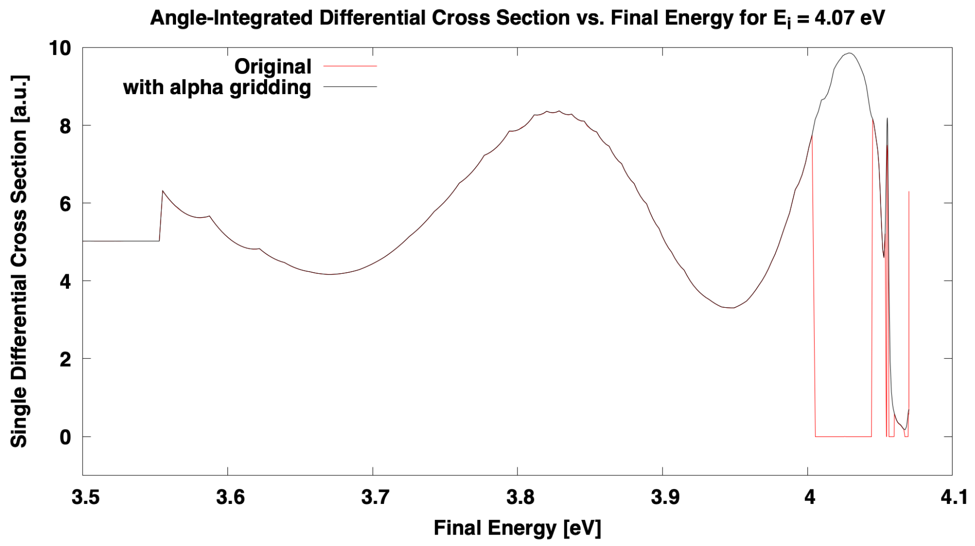
Other AMPX activities

- Virtual 3-day AMPX training course as part of the SCALE training week.
- Working on updated covariance libraries (see Andrew Holcomb’s talk)
- Working on an updated 258-group MG library, with a thermal cut-off at 10 eV.
  - SCALE CE library use thermal moderator data up to 10 eV
  - SCALE MG libraries use thermal moderator data up to 5 eV, due to a limitation in the SCALE shielding code.

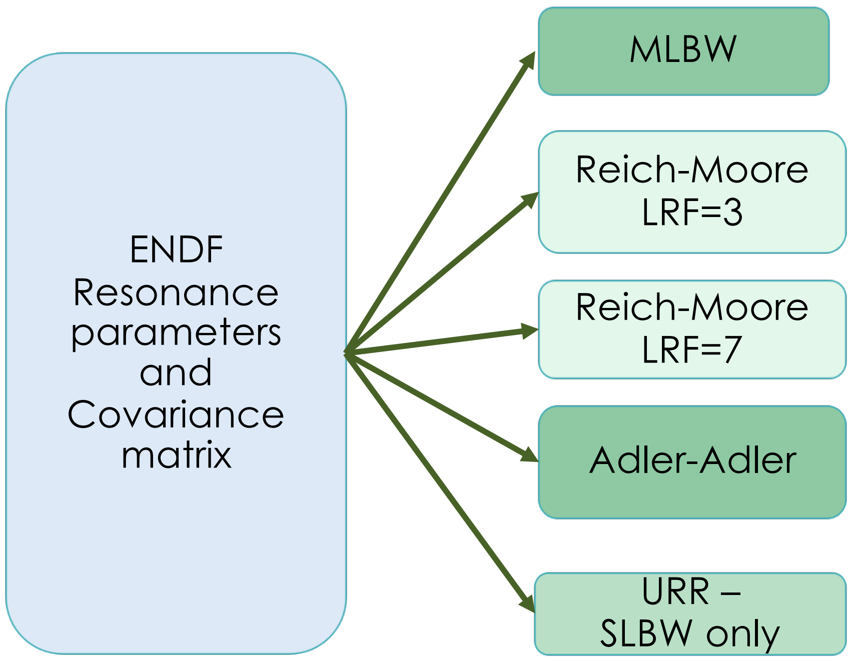
Thermal Moderators

- Short Collision Time (SCT) subroutine
  - Previously calculated SCT S(a,b) before cross-section calculation
    - Normally fine, but lead to issues with large values of beta / low temperatures (large number x small number)
  - Now called during cross-section calculation & calculates SCT DDXS, which fixes the large number x small number issue
- Changed angular gridding
  - Preconditioned mu gridding is now based on values of alpha in addition to linearly-spaced values
  - Ensures sharp features are captured in the calculations



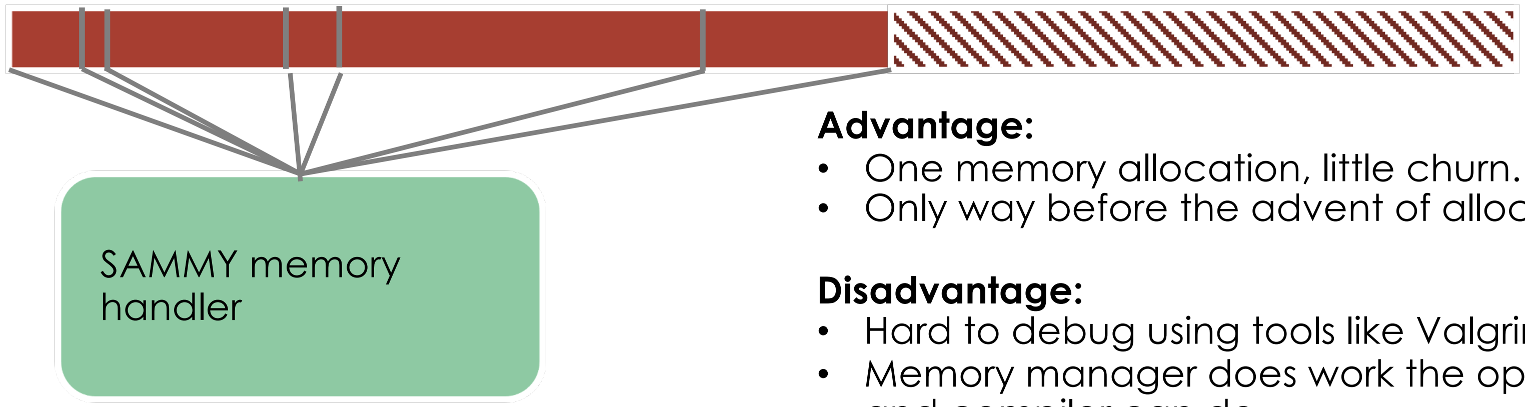


Resonance parameter storage



- SAMMY and AMPX now share the same C++ in-memory class for resonance parameters. This includes covariance information.
- Eliminated some temporary file I/O
- SAMMY still uses its own reading/writing of parameters in ENDF format. We intend to switch to the AMPX reading and writing routines. If we do, this will enable access to GNDS formatted files.
  - We have delayed to implement this as we concentrated on making the SAMMY code more modular.
- We also intend to share the same code to calculate cross section data and derivatives

Container array



Advantage:
- One memory allocation, little churn.
- Only way before the advent of allocate

Disadvantage:
- Hard to debug using tools like Valgrind.
- Memory manager does work the operating system and compiler can do
- Array is fixed and needs to be dimensioned for largest desired problems.

We have eliminated all use of the Container array in SAMMY and can now run modern error detection tools. This has turned out to be a complicated task due to frequent recycling of *container* array space, made more burdensome by the cryptic naming of thousands of arrays used in SAMMY
Added advantage: The code is a bit more modular and additional changes are easier to implement.

Numerous defects in the code, due to the use of the container array, have been detected, reported, and fixed.

Energy grids in SAMMY

Energy grid C++ class

E_1	**Exp. Data 1**	**Theory 1**	$\frac{\partial \sigma}{\partial p_1}$	…
…	…	…	…	…
E_N	**Exp. Data 2**	**Theory 2**	$\frac{\partial \sigma}{\partial p_1}$	…

🟦	implemented
🟧	In progress

The class contains a data covariance matrix

- SAMMY uses two energy grids:
  - Experimental data grid
  - Auxiliary grid used for resolution broadening
- Energy points for both grids now use the C++ class
- Switch to use derivatives and other data in C++ class is ongoing.
- Switch to data covariance matrix is ongoing

Adjusted Parameters in SAMMY

Switched to C++ in-memory storage for all SAMMY adjusted parameter information

covariance	Covariance matrix for input parameters
uCovariance	Covariance matrix for u-parameters (internal SAMMY form of the input parameters)
uParamValues	Values for the adjusted parameters (in u-parameter space)
numParameters	Number of parameters to be adjusted
numPups	Number of propagated uncertainty parameters

We now have all parameters needed for parameter adjustments in C++ in-memory classes.

We can therefore begin to separate code that does the adjusting and the theory (plus resolution and Doppler broadening).

We developed an API for adjusting that will allow us to switch in different models and adjustment procedures.

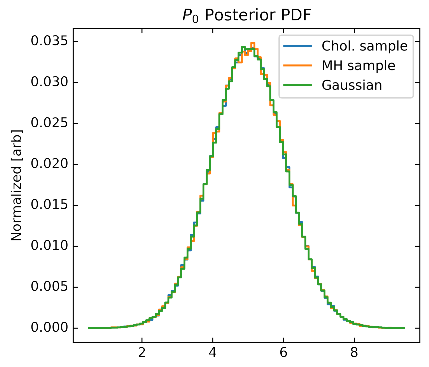
We are in the process of writing the interface to SAMMY for this API.

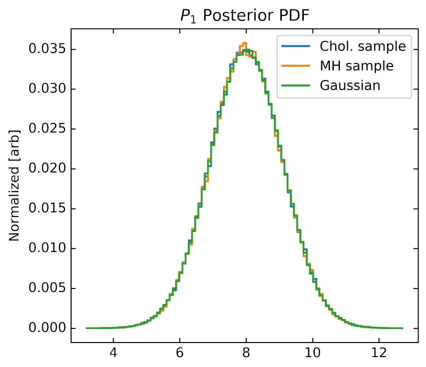
MC evaluation project: Metropolis-Hastings algorithm

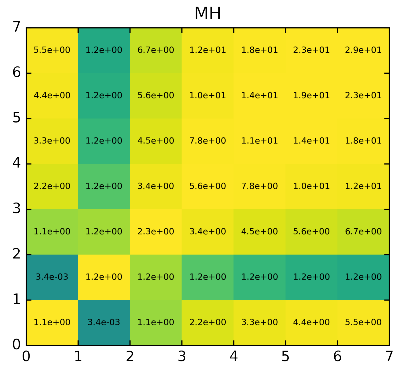
Linear:
$T(P) = P_0 \cdot x + P_1$

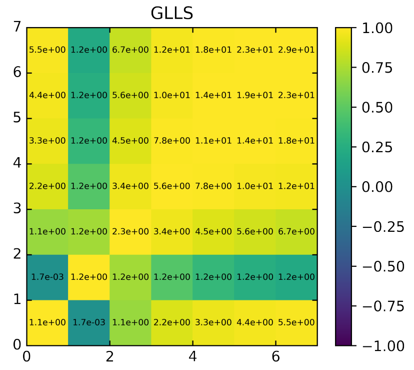
- Models: Linear, Breit-Wigner
- Linear approximation fails
- In-development: SAMMY model

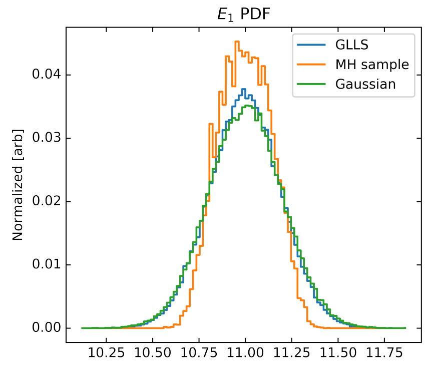
Breit-Wigner:
$T(P,E) = \sum_i \frac{1}{\pi\Gamma_i} \frac{\Gamma_i^2}{(E - E_i) + \Gamma_i^2}$

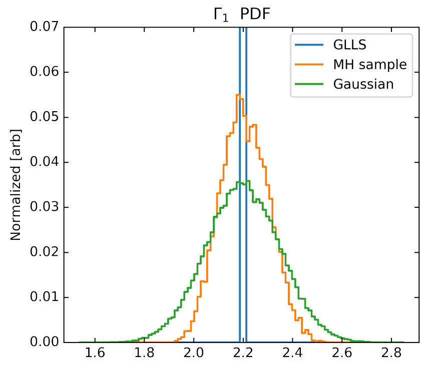


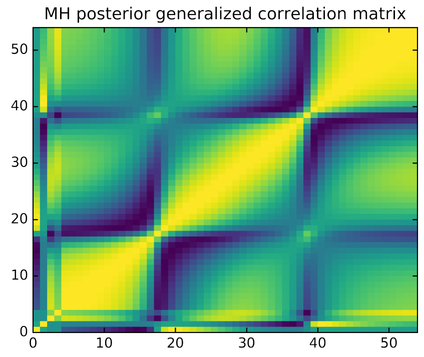


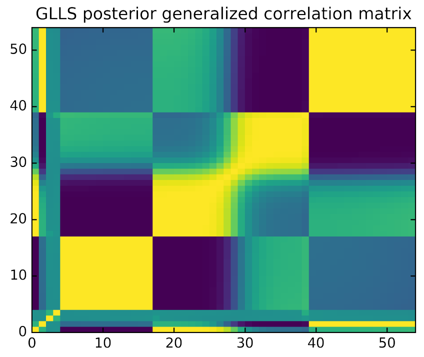












OAK RIDGE National Laboratory

Open slide master to edit

New R-matrix parameterization of direct reactions for improved ENDF nuclear data evaluations by SAMMY

- Neutron-Nucleus reactions are broadly classified as:
  - Direct: neutron excites simple states of a target nucleus, yielding smooth cross section background
  - Resonant: neutron excites complex nuclear states manifested by resonances seen in cross sections
- Phenomenological *R*-matrix formalism accounts for resonant reactions only
- Therefore, there is a need to include direct reactions into *R*-matrix formalism
- Consequently, we have generalized *R*-matrix formalism to account for direct reactions
- This amounts to redefining the scattering matrix, $\boldsymbol{U}_R$, of the R-matrix formalism, as:

$$\boldsymbol{U}_{D+R} = \boldsymbol{M}_D^{-1}\boldsymbol{U}_R\boldsymbol{M}_D$$

where $\boldsymbol{M}_D$ is a matrix whose elements parameterize direct reactions

- $\boldsymbol{U}_{D+R}$ accounts for interference between direct and resonant reactions
- It also enables inclusion of direct neutron capture processes into R-matrix formalism
- These new features will be implemented into modernized version of SAMMY

This work was supported by the Nuclear Criticality Safety Program, funded and managed by the National Nuclear Security Administration for the Department of Energy.