SAMMY Modernization

Goran Arbanas

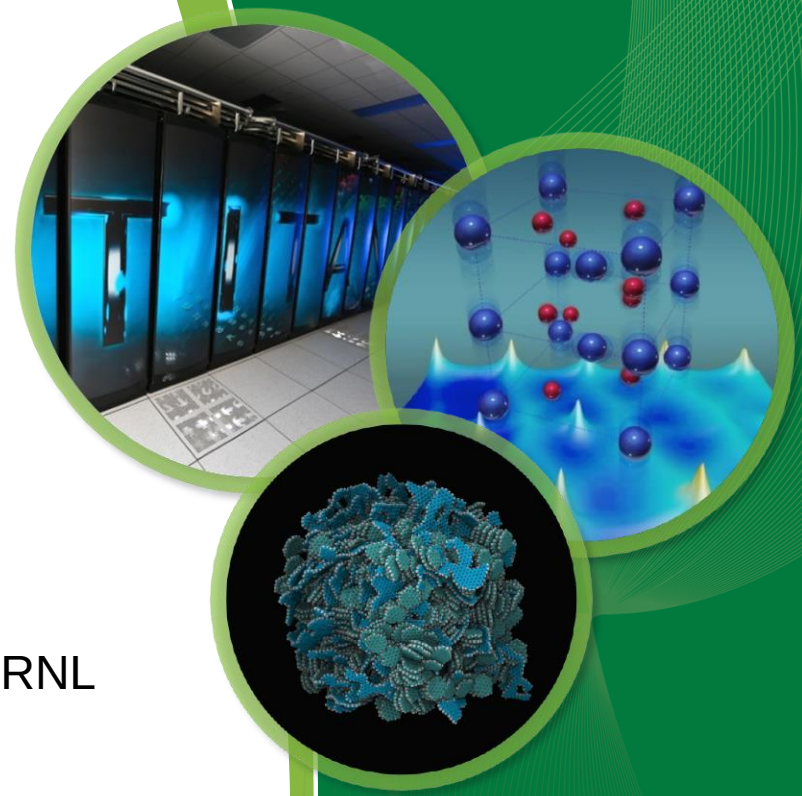
Vladimir Sobes

Andrew Holcomb

Marco Pigni

Dorothea Wiarda

Nuclear Data and Criticality Safety Group, ORNL



Consultant's Meeting on
R-Matrix Codes for Charged-Particle Reactions
in the Resolved Resonance Region,
IAEA Headquarters,
Vienna, Austria
December 5-7, 2016

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

OAK RIDGE National Laboratory

Talk outline

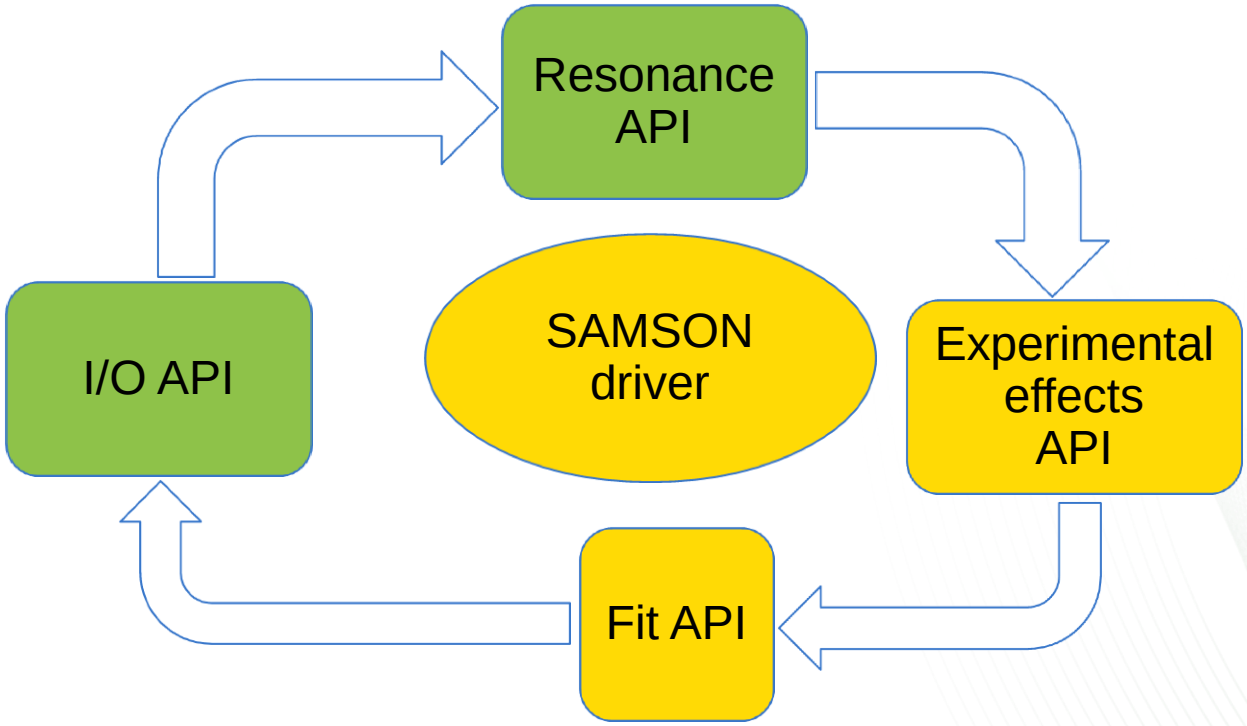
- Overview of SAMMY 8.1
- High-level API Design Overview for SAMMY Modernization
- Recent developments

New features in SAMMY 8.1

- SAMINT: integral benchmark exp.'s inform res. param. eval's.
  - Implemented by Vlad Sobes
- SAMMY integrated into SCALE SQA in AMPX footsteps
  - Automated cmake/ctest suite, revision control repository, FogBugz
  - Platforms supported Linux/gfortran, Mac/gfortran, Windows/ifort
- New detector resolution functions in collaboration with RPI
- Updated physical constants
  - SAMMY and SAMRML compute consistently now
- Corrected a misplaced index causing incorrect matrix multiplication for non-diagonal data covariance matrix
  - (uncovered and corrected by Vlad Sobes)
- Several other bug fixes
  - Revealed by compiler or platform idiosyncrasies

SAMSON high-level API diagram

- Define API's before implementation
  - Enables variety of methods for each API
  - Leverage I/O and Resonance API from C++ SAMRML
    - Delineate I/O formats from program logic (e.g. SCALE data resource)



Experimental Effects (EE) API

- Convolution of: Doppler broad., Target, and Detector effects…
  - Each one implenting the EE API

Doppler broadening (FGM, DDXS, S(a,b))

Neutron transport: SHIFT API (DBRC, LH, multiple scatter.)

Resolution Function (via e.g. MCNP)

- SHIFT API for on-the fly neutron transport aspects
  - Would enable fitting integral benchmark experiments (IBE)
  - In SCALE development (in 3-6 months)
  - MPI enabled
  - It could use MCNP input
- In principle the entire experimental setup could be simulated
  - Fitting to raw data may be possible; varying opinions
  - Raw data may become publicly available

Fit API: GLS, Monte Carlo, MinMax, ...

- Generalized Least Squares (GLS)
  - Compact expressions by Froehner (Sect. 2.2 of JEFF Report 18, 2000)
    - Incorporates covariance between various data sets
- (Total) Monte-Carlo, MinMax
  - May yield more realistic uncertainties than GLS.
- Fit differential, integral benchmark experiment data
  - Separately or together (cf. SAMINT)

Fit API: Preliminary interface

Interface
Fit

setData
Sets an instance of Data interface

initialize
After setting Data object initialize internal data structures

execute
Do the actual fitting

finalize
Clean up any internal resources

Interface
Array

getNumDim
Get the number of dimensions

getSize(int dim)
Get the array size for dimension m

getValue(int i1, int i2, …)
Get the value for the indicated indices. In C++ we would pass in a vector of length getNumDim

setValue(int i1,int i2, …_
set value

Interface
Data

getNumberParams
Get the number of parameters

getNumData
Get the number of experimental

getData
Get the list of experimental data (1-dim Array)

getParam
Get the list of initial params (1-dim Array)

getCovMatrix
Get the full covariance matrix (2-dim Array)

getTheory
Get theoretical values based on current parameters (1-dim Array)

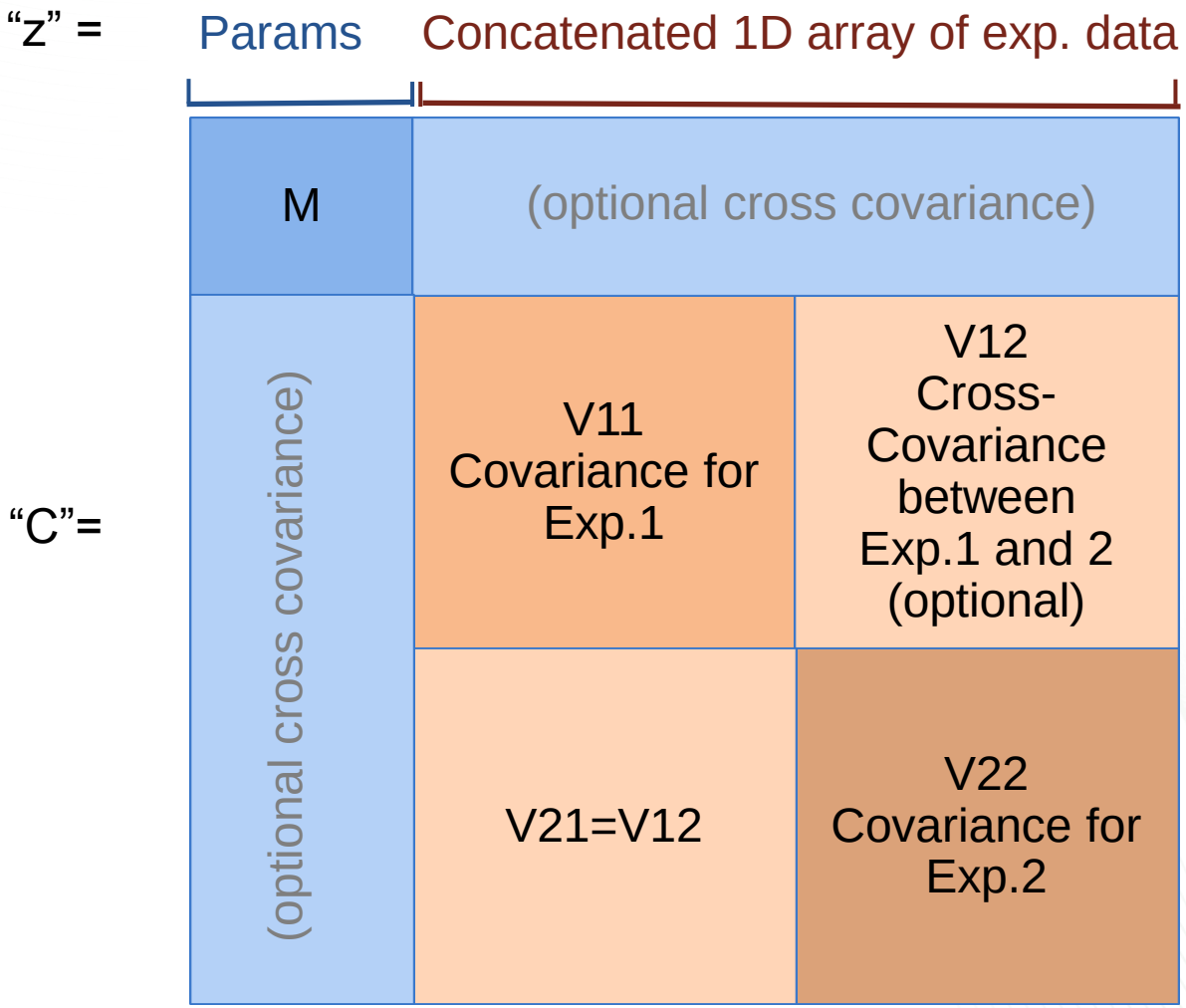
setParam
Set the current parameters (1-dim Array)

setCovMatrix
Set the full covariance matrix (2-dim Array)

- Actual instances are instantiated by a factory class.
- Data will have method to get the derivatives (2-dim Array: getNumberParams x getNumData). There will be a function that computes derivatives numerically.
- Fit calls setParams, getTheory, setCovMatrix repeatedly in the course of fitting the data

Fit API: GLS implementation

- Parameters and exp. data cast into 1D array by an implementation of Data
  - for generic use inside SCALE framework
  - Froehner's formulation and notation:



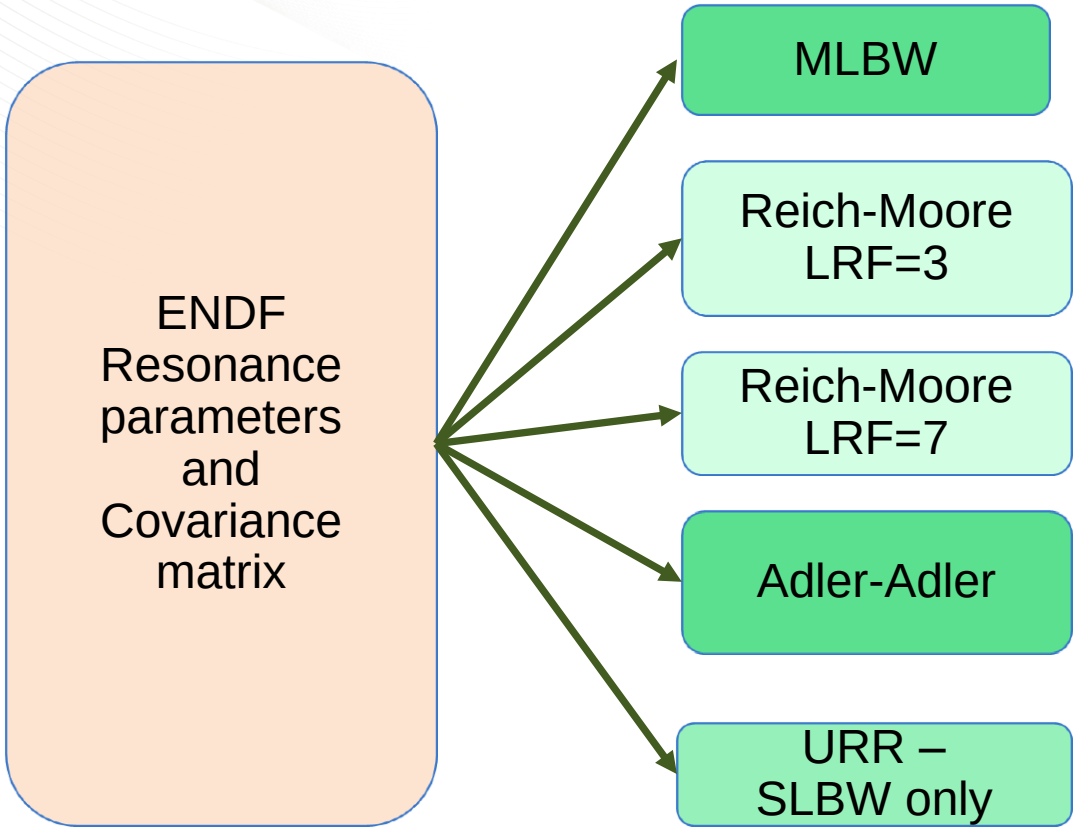
Fit API: GLS implementation

GLS

- Generalized Least Squares (GLS)
  - Compact expressions by Froehner (Sect. 2.2 of JEFF Report 18, 2000)
    - Incorporates covariance between various data sets

- Consider cpp-array library (CPC **185**,1681, 2014)
  - Parallelized via underlying BLAS library (Intel MKL, cuBLAS, MAGMA, …)
  - Compact notation for matrix operations, e.g. parameter set update
  - P = P + inv(transpose(S) * inv(C) * S) * transpose(S) * inv(C) * (Π – T)
- BLAS advantages
  - drastically speeds up large matrix operations in SAMMY & shortens code
    - (Arbanas, Dunn, Wiarda, M&C2011)

Leverage AMPX Resonance API

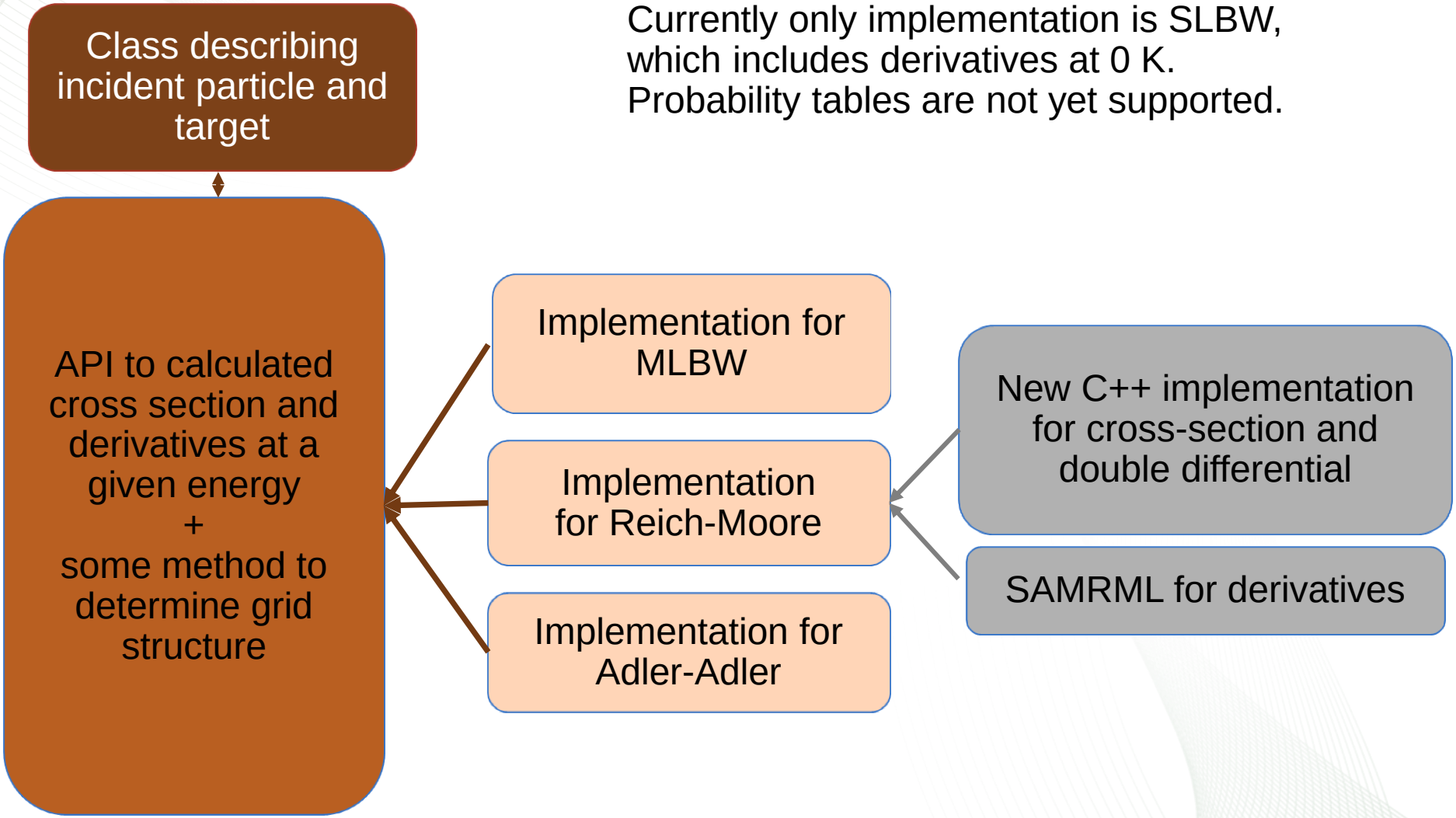


- SLBW and MLBW parameter are stored in the same class with a flag indicating which formalism to use
- Resonance parameters for Reich-Moore for LRF=3 are initially stored in a different class, but are converted to a LRF=7 class before calculation
- If derivatives are desired, all formalisms (except URR) 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

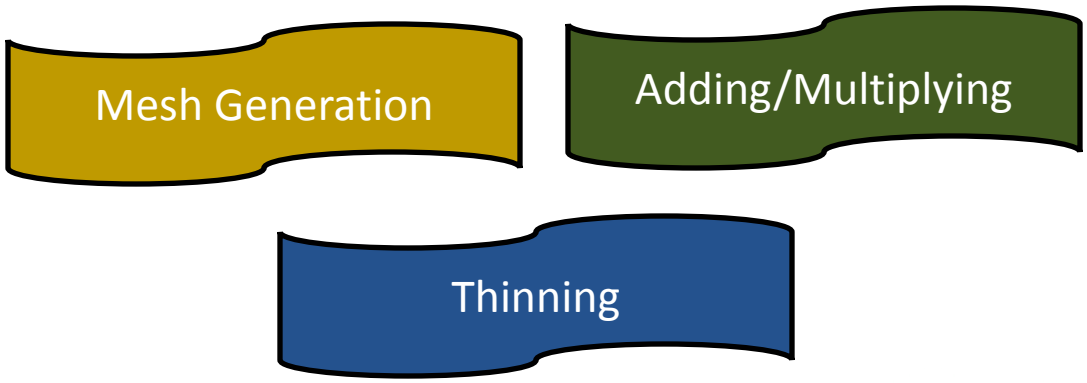
Leverage AMPX Resonance API classes

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



Leverage AMPX mesh generation API:

Iterator
Remove
Insert
Interpolate
can interpolate ?



Recent Developments:

- Generalized Reich-Moore Approximation (ND2016)
  - Total capture equivalent to full R-matrix, unitarity, Brune transform O.K.
- Convert SAMMY resonance parameters to formal R-matrix
  - SAMMY evaluations set $S_c(E)-B_c=0$ boundary condition (b.c.)
    - Advantage: R-matrix resonance energies near resonance peaks
  - Andrew Holcomb programmed conversion to/from formal R-matrix
    - Implement full alternative R-matrix (in progress by A. Holcomb)
- S-matrix pole representation of R-matrix (Hwang, Froehner)
  - via Brune transform of R-matrix param.'s for $L_c(E)-B_c=0$ b.c.
  - Useful for on-the-fly Doppler broadening in neutron transport app.'s
  - Nicolas Michel's complex Coulomb w.f. library (CPC, 176 (2007) 232)
- Effects of closed channels below and at thresholds (cusp, …)
- Can R-matrix formalism support a normalization uncertainty?
  - Model uncertainties often much smaller than data normalization unc.'s
  - How large are effects of unaccounted channels

Conclusions and outlook

- SAMMY 8.1 Beta released rsicc.ornl.gov
  - 8.1 final release will include some recently suggested modifications
- SAMMY modernization continuing in a SCALE SQA framework
- API top-to-bottom design
  - code sharing with AMPX modules
    - e.g. I/O, SAMRML modernized into C++ by Doro Wiarda and Andrew Holcomb
- Collecting SAMMY feature requests from users
  - Collaboration with RPI and others has lead to SAMMY improvements

Backup slides

RM History and use

Reich and Moore, Phys. Rev. 111, 929 (1958)

- Reich-Moore divides full *R*-matrix into particle and γ-ray blocks
  - Formal expressions derived for reduced R-matrix of particle channels
  - The effect of γ-ray channels on particle-channels approx. by a diagonal
    - Level-level interference among γ-ray channels neglected

F. Froehner, JEFF Report 18 (2000)

$$R_{cc'} = \sum_{\lambda} \frac{\gamma_{\lambda c}\gamma_{\lambda c'}}{E_\lambda - E - i\Gamma_{\lambda\gamma}/2} \qquad (c, c' \notin \gamma)\,, \qquad (194)$$

corresponding SLBW level matrix. Otherwise it is so exact that although the reduced collision matrix cannot be unitary - because of transitions into eliminated channels - the overall collision matrix can still be considered as unitary, i. e. as conserving probability flux, so that the capture cross section may alternatively be obtained as the difference

$$\sigma_{c\gamma} = \sigma_c - \sum_{c' \notin \gamma} \sigma_{cc'}\,, \qquad (196)$$

with σ_c calculated from the reduced collision matrix element U_{cc} according to Eq. 153.

$$\sigma_c \equiv \sum_{c'} \sigma_{cc'} = 2\pi \lambda\!\!\!^{-2}_{c} g_c (1 - \mathrm{Re}\ U_{cc})\,, \qquad (153)$$

Motivation for generalized Reich-Moore

- Conversion of RM R-matrix parameters
  - Between formal and alternative (a.k.a. physical) ones (C. Brune, 2002)
- Investigate whether Reich-Moore (RM) approx. is unitary
  - It may appear not to be as particle-channel R-matrix is complex (not real)
  - RM derivation was revisited to investigate unitarity
  - In this process a generalization of RM was found
    - This generalization is manifestly unitary
    - Corollary: Conventional RM is also unitary
    - It provides basis for Fritz Froehner's prescription used by SAMMY
- Other potential benefits Generalized RMA (gRMA)
  - May provide better fits to total capture (and other cross sections)
    - Because gRMA reproduces total cross section formally
  - May shed light on resonant-interference effects neglected by conv. RM
    - Including statistical properties of capture widths

Derivation uses full R-matrix via level-matrix A

$$A^{-1} = e - E1 + \gamma(L - B)\gamma^T$$

- Separate channel space into particle and γ-ray channels:

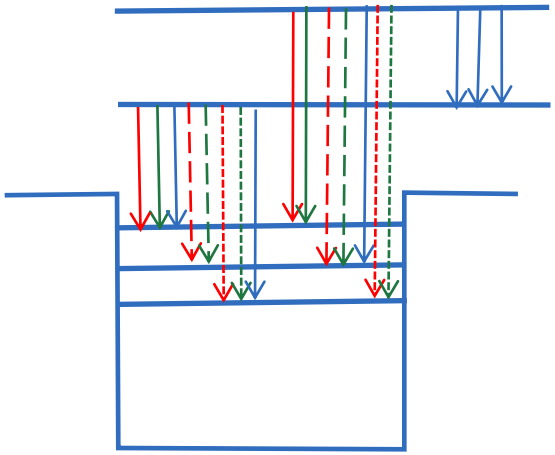
$$\gamma \equiv (\gamma_c, \gamma_\gamma)$$

$$L - B \equiv \begin{pmatrix} L_c - B_c & 0 \\ 0 & L_\gamma - B_\gamma \end{pmatrix}$$

$$A^{-1} = e - E1 + \gamma_\gamma(L_\gamma - B_\gamma)\gamma_\gamma^T + \gamma_c(L_c - B_c)\gamma_c^T$$

Full R-matrix g-ray channels

- γ-ray channels
  - defined by EM multipolarity, helicity, and final state quantum numbers
  - Selection rules based on f.s. quantum numbers, γ-ray multipolarity
  - Electric: E1, E2, E3, …
  - Magnetic: M1, M2, M3, …
- Level-level interference takes place via identical γ-ray channels
  - Use conventional approximation S-B=0
    - Or use Brune alternative R-matrix parameters for which S-B=0 always

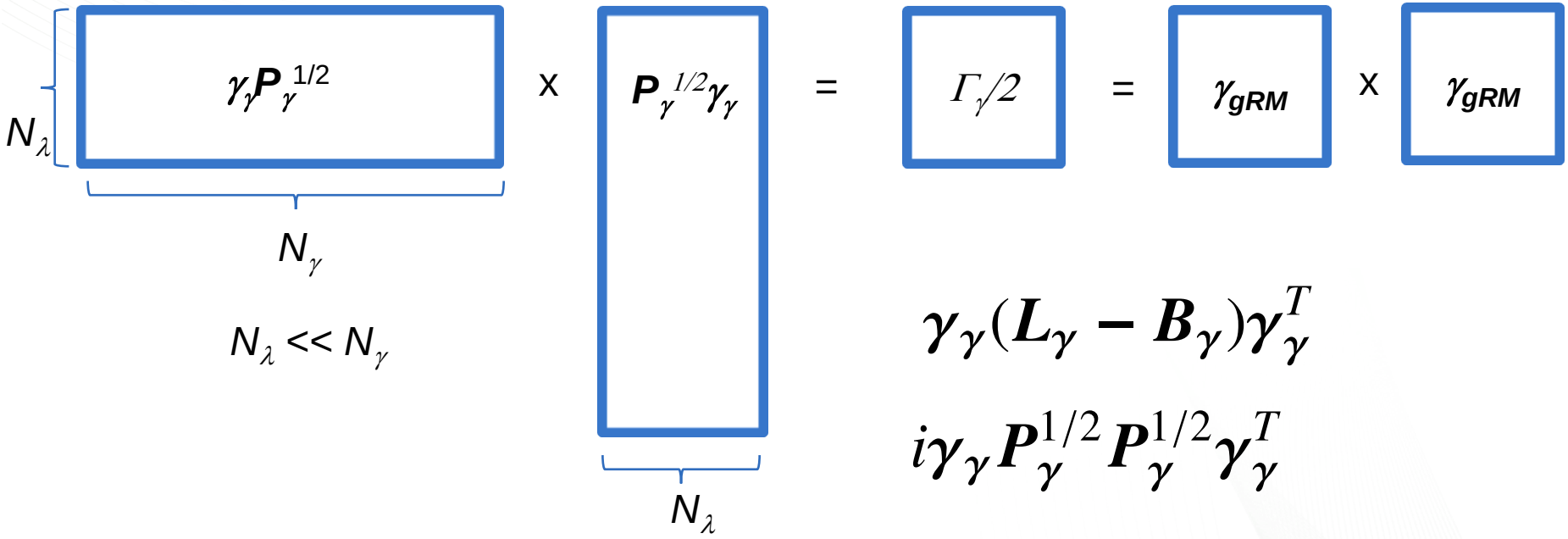


$$i\boldsymbol{\gamma}_\gamma \boldsymbol{P}_\gamma^{1/2} \boldsymbol{P}_\gamma^{1/2} \boldsymbol{\gamma}_\gamma^T$$

Generalized RM Derivation

- Consider capture-width matrix Γ_γ inside the level matrix **A**

γ_γ = N_λ x N_γ matrix of physical capture channel widths
γ_{gRM} = N_λ x N_λ matrix of gRMA capture channel widths



$N_\lambda << N_\gamma$

$$\gamma_\gamma(L_\gamma - B_\gamma)\gamma_\gamma^T$$

$$i\gamma_\gamma P_\gamma^{1/2} P_\gamma^{1/2}\gamma_\gamma^T$$

- Since *total capture* cross section depends on Γ_γ, it could be fit equally well by N_λ as it could by all N_γ capture channels
  - True for *total* capture only (individual γ-channels require full R-matrix)

Total capture of gRMA equals that of full R-matrix

- Working with alternative R-matrix parameters since S(E)-B=0

$$\boldsymbol{A}^{-1} = \boldsymbol{e} - E\mathbf{1} + \boldsymbol{\gamma}_\gamma(\boldsymbol{L}_\gamma - \boldsymbol{B}_\gamma)\boldsymbol{\gamma}_\gamma^T + \boldsymbol{\gamma}_c(\boldsymbol{L}_c - \boldsymbol{B}_c)\boldsymbol{\gamma}_c^T$$

$$\boldsymbol{A}^{-1} \approx \boldsymbol{e} - E\mathbf{1} + i\boldsymbol{\gamma}_\gamma \boldsymbol{P}_\gamma^{1/2}\boldsymbol{P}_\gamma^{1/2}\boldsymbol{\gamma}_\gamma^T + \boldsymbol{\gamma}_c(\boldsymbol{L}_c - \boldsymbol{B}_c)\boldsymbol{\gamma}_c^T$$

$$= \boldsymbol{e} - E\mathbf{1} + i\boldsymbol{\Gamma}^{(\gamma)}/2 + \boldsymbol{\gamma}_c(\boldsymbol{L}_c - \boldsymbol{B}_c)\boldsymbol{\gamma}_c^T$$

- The total capture cross section is proportional to:

$$\boldsymbol{U}_{c\gamma}\boldsymbol{U}_{c\gamma}^{*T} = \boldsymbol{\Omega}_c \boldsymbol{P}_c^{1/2}\boldsymbol{\gamma}_c^T \mathbf{A}\boldsymbol{\gamma}_\gamma \boldsymbol{P}_\gamma^{1/2}\boldsymbol{\Omega}_\gamma\boldsymbol{\Omega}_\gamma^* \boldsymbol{P}_\gamma^{*1/2}\boldsymbol{\gamma}_\gamma \mathbf{A}\boldsymbol{\gamma}_c^T \boldsymbol{P}_c^{*1/2}\boldsymbol{\Omega}_c^*$$

$$= \boldsymbol{\Omega}_c \boldsymbol{P}_c^{1/2}\boldsymbol{\gamma}_c^T \mathbf{A}(\boldsymbol{\gamma}_\gamma \boldsymbol{P}_\gamma^{1/2}\boldsymbol{P}_\gamma^{1/2}\boldsymbol{\gamma}_\gamma^T)\mathbf{A}\boldsymbol{\gamma}_c^T \boldsymbol{P}_c^{*1/2} \quad (14)$$

$$= \boldsymbol{\Omega}_c \boldsymbol{P}_c^{1/2}\boldsymbol{\gamma}_c^T \mathbf{A}(\boldsymbol{\Gamma}^{(\gamma)}/2)\mathbf{A}\boldsymbol{\gamma}_c^T \boldsymbol{P}_c^{*1/2}\boldsymbol{\Omega}_c^* \quad (15)$$

- → since $\boldsymbol{\Gamma}^{(\gamma)}$ is the same for full R-matrix and gRMA

Full R-matrix vs. conventional RM: total capture

- full R-matrix equivalent to gRMA; 2-level SAMMY example:

Table II B2.1. Parameter values used to illustrate Reich-Moore vs. full R-matrix calculations

	λ	Energy (MeV)	$\bar{\Gamma}_{\lambda\gamma}$ (eV)	$\Gamma_{\lambda n}$ (eV)	Sign $\times \Gamma_{\lambda\gamma}$ (eV)[a]
Reich Moore	1	1.0	1.0	10000.	
	2	1.1	1.1	11000.	
Pseudo-full R-matrix # 1	1	1.0	10^{-8}	10000.	1.0
	2	1.1	10^{-8}	11000.	1.1
Pseudo-full R-matrix # 2	1	1.0	10^{-8}	10000.	1.0
	2	1.1	10^{-8}	11000.	−1.1

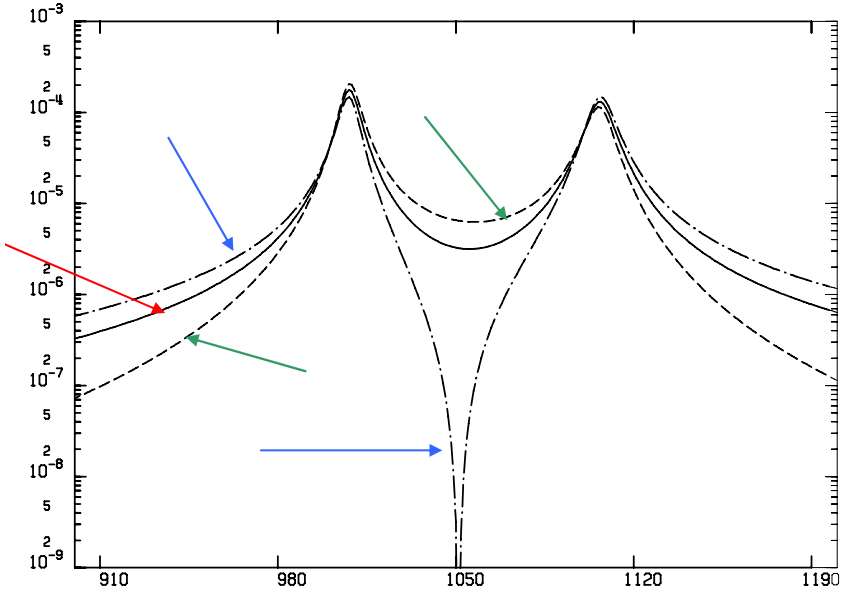


Figure II B2.1. Reich-Moore approximation vs. full R-matrix for artificial example of test case tr110.

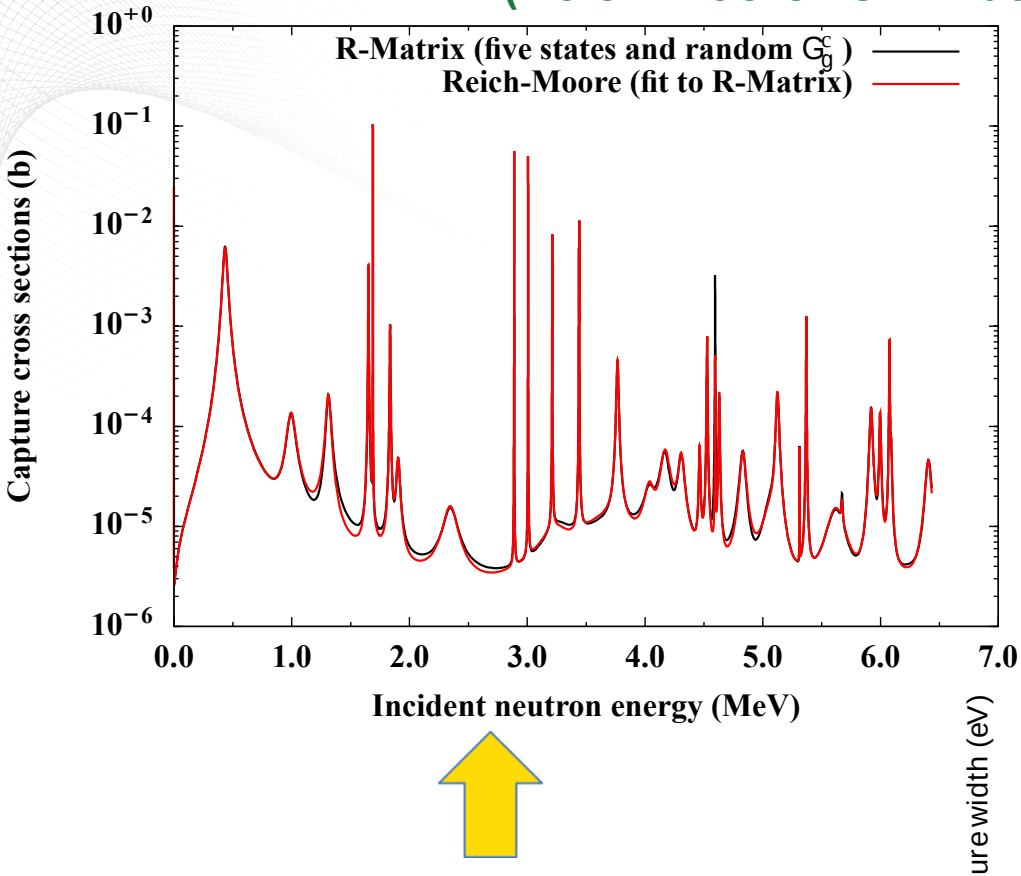
Solid line = Reich Moore

Dot-dash = full R-matrix # 1

Dash = full R-matrix # 2

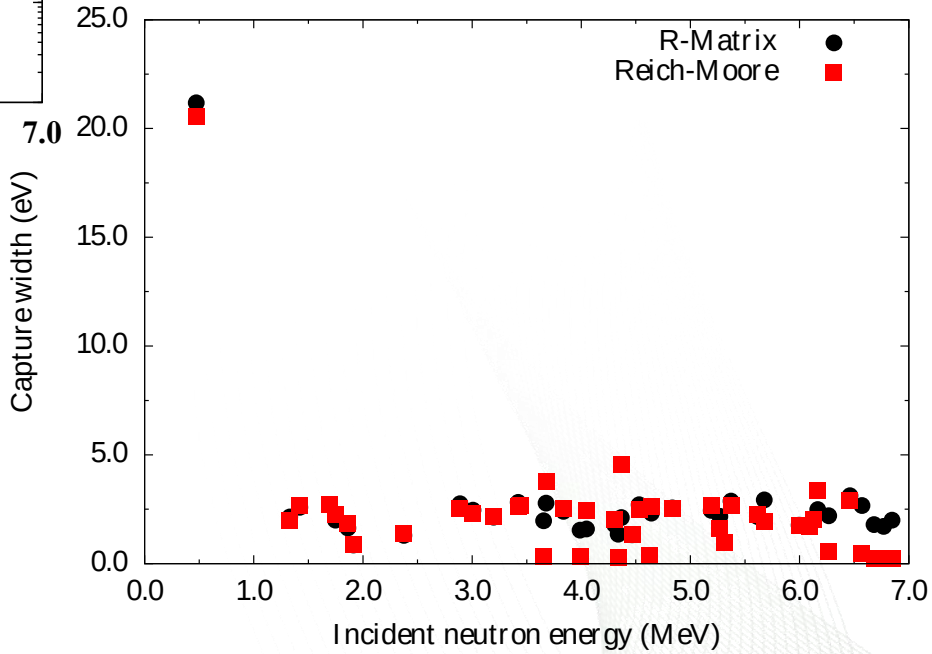
^{16}O Full R-matrix vs. conventional RM toy case

(Reich-Moore vs R-Matrix calculations)



Good agreement between Reich-Moore and R-Matrix capture widths

$$G_{mm} = 2\sum_c g_m g_m$$



- R-matrix case (five states for ^{17}O) generated by randomly sampling capture widths (in black)
- The R-matrix capture cross sections were used to obtain capture widths in Reich-Moore approximation and related capture cross sections (in red)
- The fit of the capture cross sections did not impact other reaction channels

History

- Developed by Dr. Nancy Larson since 1970's through 2008
- Includes SAMMY + 25 auxiliary codes
  - e.g. SAMRML
- Architecture
  - Large Fortran (77) container array for memory management
- 185 multi-step test cases + 10 tutorial examples
- Comprehensive Documentation:
  - http://info.ornl.gov/sites/publications/files/Pub13056.pdf
- Employed for resolved resonance evaluations in ENDF
- Distributed via RSICC https://rsicc.ornl.gov/
  - SAMMY 8.1 is forthcoming

Capabilities

- Multi-level Multi-channel R-matrix code
- Bayesian fitting of R-matrix resonance parameters (RP)
  - a.k.a. Generalized Least Squares
  - yields covariance matrix of RP
- Data reduction:
  - Experimental Facility Resolution functions: ORELA, RPI, GELINA
  - Normalization, background
- Detector resolution functions
  - Configurable for variety of detectors
- Doppler broadening
  - Solbrig's kernel, Leal-Hwang method
- Multiple scattering effects, and other target effects
- Charged projectiles (p, α)
- Unresolved Resonance Range (FITACS by F. Froehner)

Capabilities: R-matrix *and* Reich-Moore Approx.

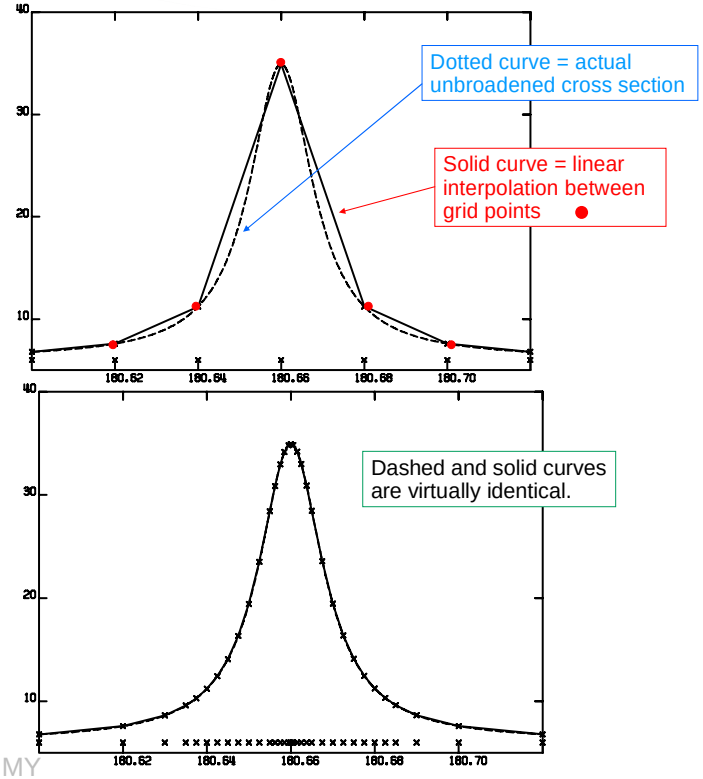
- Reich-Moore approximation (RMA)
  - For channels approximated statistically via Random Phase Hypothesis
  - Applied mostly for capture channels
    - penetrability *P=1* for capture in RMA (and in exact R-matrix below)
- R-matrix w/o RMA
  - Make capture channels as reaction channels in SAMMY input files
  - Marco Pigni's talk:
    - Quantify accuracy of RMA relative to R-matrix on ^{16}O using 5 γ-rays/level

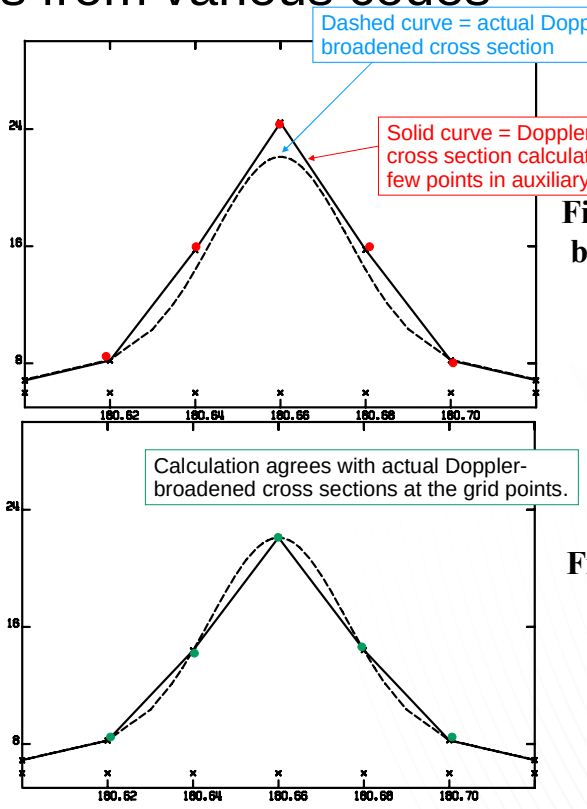
Capabilities: Resolution broadening

- Experimental Resolution broadening
  - Convolution of 4 components:
    - The electron burst
      - a square function of time
    - Neutron sources:
      - tantalum target
      - water moderator
    - Neutron detectors
      - model NE-110 recoil proton detector
      - lithium glass detector
    - Time-of-flight channel width
      - a square function of time
- Doppler broadening (DB)
  - numerical convolution of cross sections by Solbrig kernel over *E*
  - Double-differential c.s. DB-ed approximately by SAMMY now
    - our group developed and published an exact method
  - Leal-Hwang: efficient, used by SCALE

Capabilities: configurable energy mesh

- Auxiliary energy mesh for computations
  - Includes data energy points and additional points
    - especially at and near resonance peaks to trace their shapes completely
    - Needed because resonance are narrower at T=0 K before Doppler broadening to room temperature for data
  - Useful for comparing results from various codes





Figures from SAMMY Users' Guide:

Figure III A1.2. Incorrect Doppler-broadened cross section calculated with too few points in the auxiliary grid.

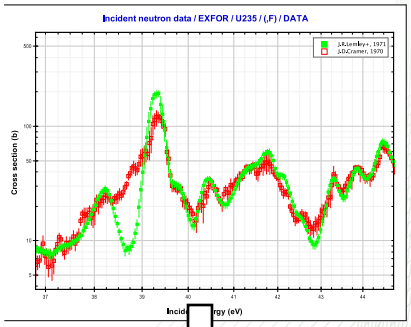
Figure III A1.4. Doppler-broadened cross section calculated with an adequate number of points in the auxiliary grid.

SAMINT: Nuclear Data Adjustment Based on Integral Benchmark Experiments (IBE)

- SAMINT is an auxiliary program designed to allow SAMMY to adjust nuclear data parameters based on integral data.
- Enables coupling of differential and integral data evaluation in a continuous-energy framework
- Informs the evaluation of resolved resonance parameters
  - Based on performance in simulations of IBE's
- Leverages RNSD (ORNL) expertise in sensitivity studies of nuclear IBE's and applications
  - SCALE modules TSUNAMI and TSURFER

Integral Experiments to Aid Nuclear Data Evaluation

- SAMINT can be used to extract information from integral benchmarks to aid the nuclear data evaluation process.
- Near the end of the evaluation based on differential experimental data, integral data can be used to:
  - Resolve remaining ambiguity between differential data sets
  - Guide the evaluator to troublesome energy regions
  - Inform the evaluator of the most important nuclear data parameters to integral benchmark calculations
  - Improve the nuclear data covariance matrix evaluation

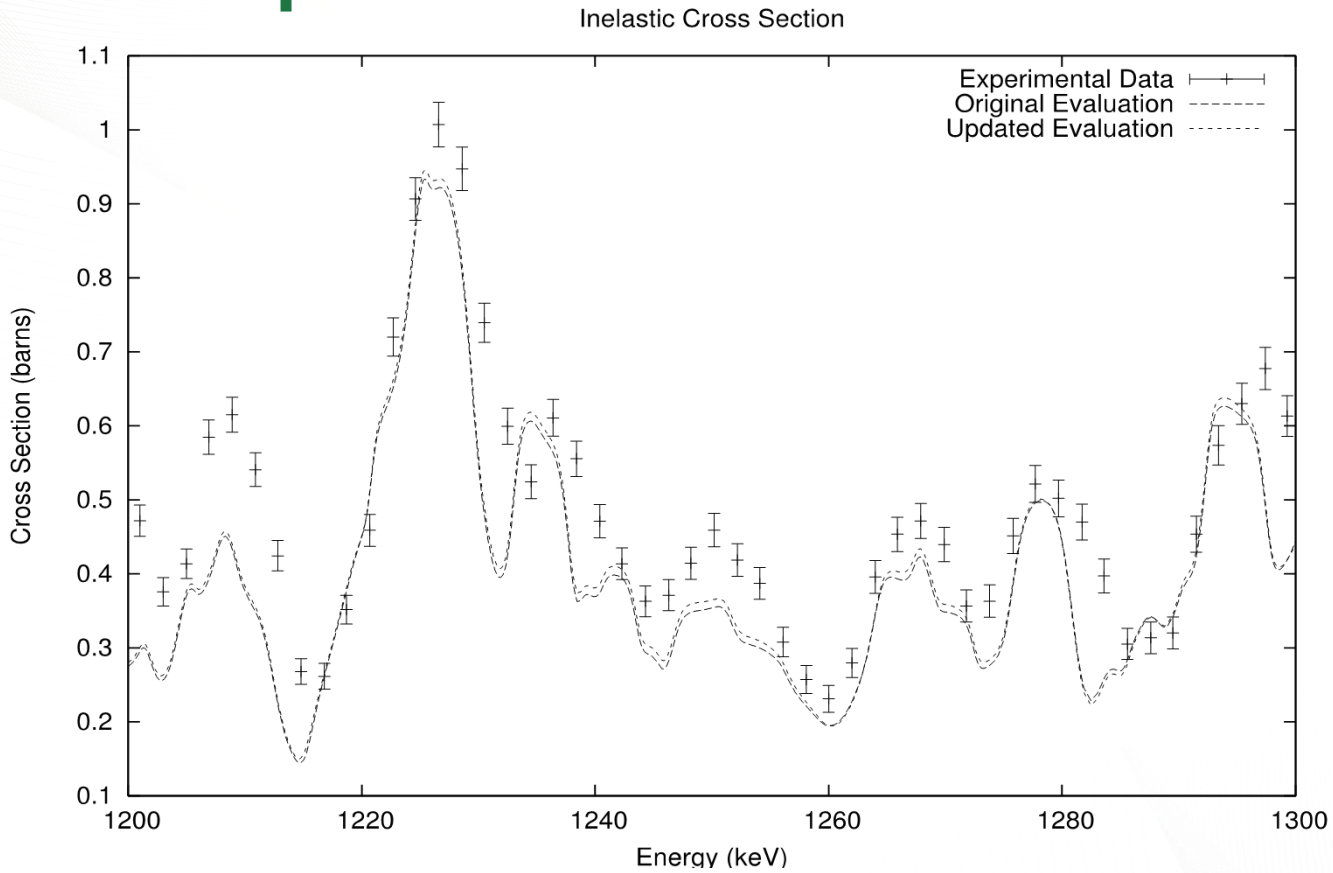


SAMMY

ENDF

OAK RIDGE National Laboratory

Cross Section Changes: Finer Scale than Differential Experimental Data



Inelastic cross section of ^{56}Fe before (χ^2 = 23.6023) and after (χ^2 = 22.9036) the adjustment based on integral experimental data plotted on top of differential experimental data of Perey, presented with one standard deviation error bars.

Connections to AMPX and SCALE

- AMPX: data processing suite for SCALE data libraries
  - Dorothea Wiarda, Andrew Holcomb, Michael Dunn (ORNL)
  - Shipped with SCALE 6.2 via https://rsicc.ornl.gov
- SCALE: nuclear modeling suite for design, safety, licensing
  - http://scale.ornl.gov Brad Rearden (ORNL) Manager
- SAMMY modernization follows AMPX and SCALE
  - Mercurial version control system
  - Cmake automated build and ctest automated testing
  - Fogbugz Bug tracking system
- SAMMY Module SAMRML “shared” with AMPX:
  - Stripped-down SAMMY for computing resonant cross sections
    - no parameter fitting, no Doppler or resolution broadening
  - also used in data processing codes: AMPX, NJOY, PREPRO
  - Modernized into C++ by Andrew Holcomb
    - Provides a framework for modernizing SAMMY

Modernization: SQA

- Version control of source code and test cases using Mercurial
- Bug tracking and workflow
- Cmake: auto configuration build (make –j)
- 25 executables built automatically on several platforms
- Ctest: auto testing tool
  - Test whether result are within a prescribed tolerance (1E-4)
    - SAMMY files tested: LST, PAR, LPT…
    - Makes it much easier to notice discrepancies.
  - 178 test cases from SAMMY 8.0.0 and
  - 4 new test cases for SAMINT
  - 1 new test case for RPI Lithium Glass detector resolution function
  - (All test cases include subcases.)

Supported Platforms and Compilers

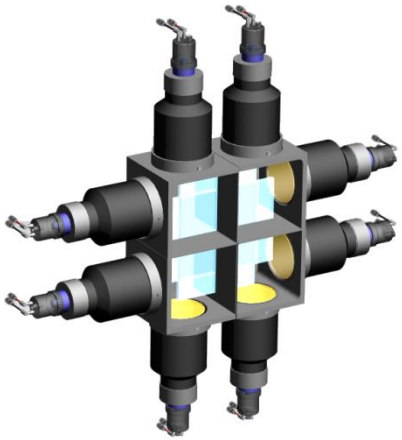
- Mac: gfortran
- Linux: gfortran, ifort
- Windows: ifort
  - Revealed few remaining issues that were corrected

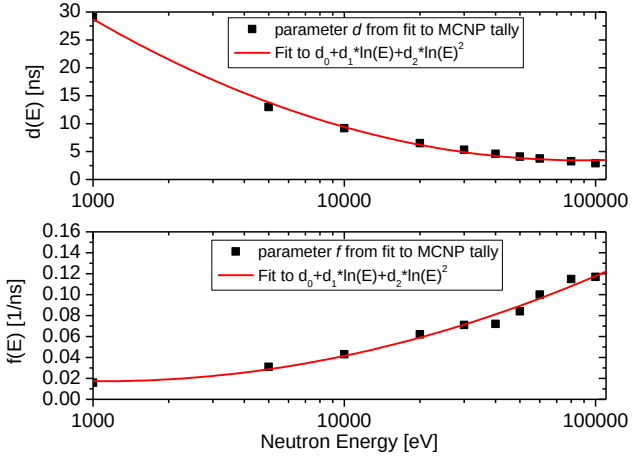
Modernization cont.'d

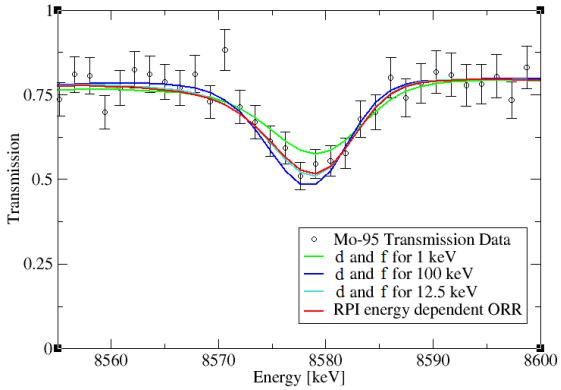
- Proposed SAMMY re-organization
  - Delineate modules that compute cross sections from those that fit resonance parameters to the data
  - i.e. keep SAMRML a standalone module called by a fitting program
  - Fitting method could remain Bayesian or Generalized Least Squares

^{6}Li-glass Neutron Detector Array MELINDA

- Improved parameterization
  - Based on MCNP simulations by Amanda Youmans (RPI)







Comparing the original SAMMY detector resolution function to the modified function with Mo-95 data

Before:

$$I(l) = \begin{cases} Dg & for\ 0 < t < d \\ De^{-f(t-d)} & for\ d < t \end{cases}$$

After:

$$d(E) = d_0 + \ln(E) * (d_1 + d_2 * \ln(E))$$

$$f(E) = f_0 + \ln(E) * (f_1 + f_2 * \ln(E))$$

Liquid Scintillator Detector liquid $CH_{1.212}$

- Based on MCNP simulations by Amanda Youmans (RPI)

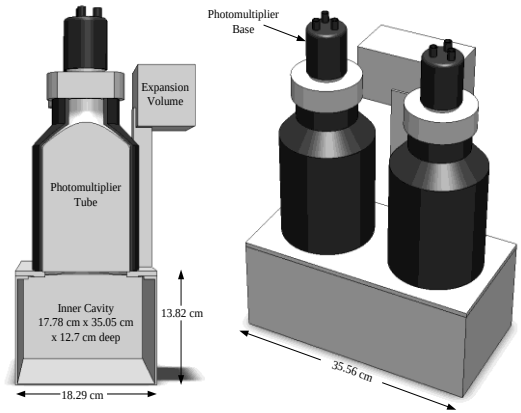
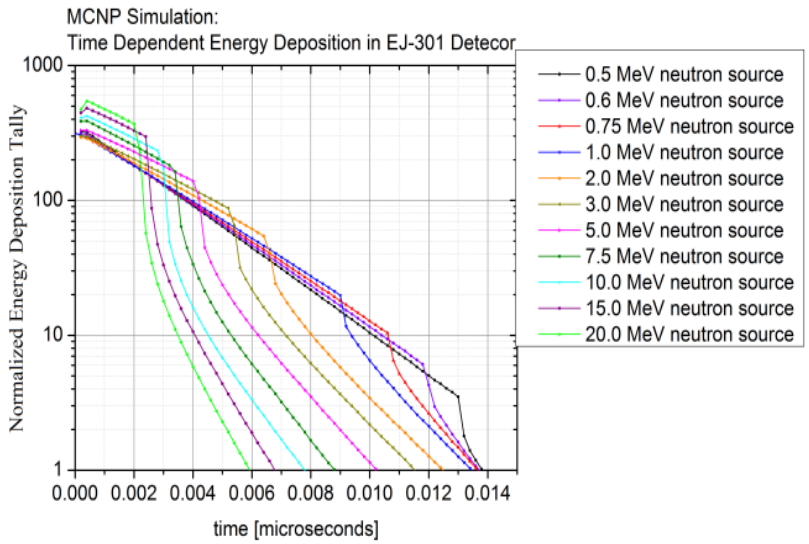
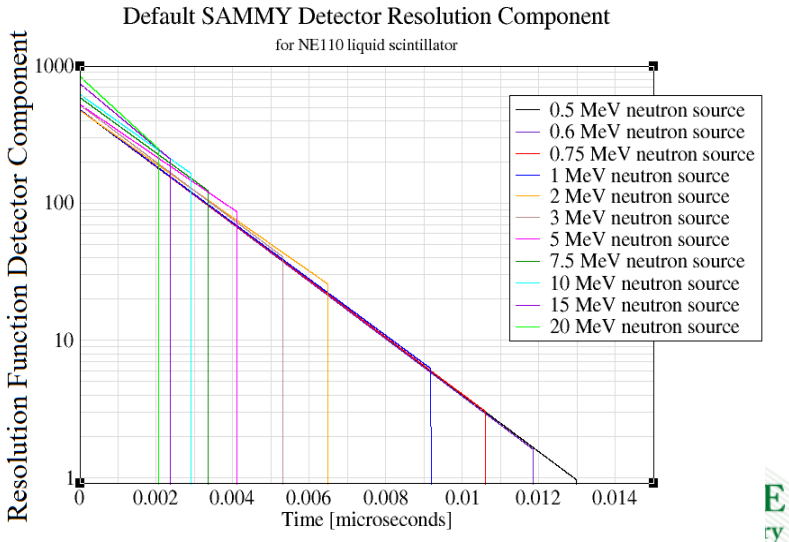


Table IV: Cross Sections for EJ-301 SAMMY Input	
Energy [eV]	Cross section [b]
500,000	7.967348
600,000	6.963397
750,000	5.958675
1,000,000	4.953201
2,000,000	3.101132
3,000,000	2.329274
5,000,000	1.683733
7,500,000	1.554593
10,000,000	1.172541
15,000,000	0.877705
20,000,000	0.663136





Minor improvements in SAMMY 8.1

- Updated physical constants
  - Consistent with SAMRML
  - SAMMY and SAMRML yield identical results now
- Corrected a misplaced index causing incorrect matrix multiplication for non-diagonal data covariance matrix
  - (uncovered and corrected by Vlad Sobes)
- Several other bug fixes
  - Revealed by compiler or platform idiosyncrasies

Recent Developments

- Convert SAMMY resonance parameters to formal R-matrix
  - SAMMY evaluations set $S_c(E)-B_c=0$ boundary condition (b.c.)
    - Advantage: resonance energies coincide with resonance peaks
    - Disadvantage: slight deviations from formal R-matrix
    - SAMMY does support formal R-matrix $B_c=-1$ b.c.
  - Andrew Holcomb programmed conversion to/from formal R-matrix
    - Converted ^{16}O $S_c(E)-B_c=0$ into formal parameters for Marco Pigni's talk
    - and is extending it to $L_c(E)-B_c=0$ in complex plane
- S-matrix pole representation of R-matrix
  - via e.g. Brune transform of R-matrix param.'s for $L_c(E)-B_c=0$ b.c.
  - Useful for on-the-fly Doppler broadening in neutron transport app.'s
    - Developed by Hwang (ANL) and Fritz Froehner (INR, Karlsruhe)
    - Ongoing collaboration Vlad Sobes and N.E. at M.I.T. (Pablo Ducru)
      - Talk by Vlad Sobes this Friday
  - Use Nicolas Michel's complex Coulomb w.f. library (CPC, 176 (2007) 232)