EPJ Web of Conferences **106**, 07006 (2016)
DOI: 10.1051/epjconf/201610607006
© Owned by the authors, published by EDP Sciences, 2016

CALMAR: A New Versatile Code Library for Adjustment from Measurements

G. Grégoire[a], C. Fausser, C. Destouches, and N. Thiollay

CEA, DEN, DER, Dosimetry, Sensors, Instrumentation Laboratory, 13108 Saint Paul Lez Durance, France

Abstract. CALMAR, a new library for adjustment has been developed. This code performs simultaneous shape and level adjustment of an initial prior spectrum from measured reactions rates of activation foils. It is written in C++ using the ROOT data analysis framework,with all linear algebra classes. STAYSL code has also been reimplemented in this library. Use of the code is very flexible : stand-alone, inside a C++ code, or driven by scripts. Validation and test cases are under progress. Theses cases will be included in the code package that will be available to the community. Future development are discussed. The code should support the new Generalized Nuclear Data (GND) format. This new format has many advantages compared to ENDF.

1. Introduction

Adjustment methods are well known in dosimetry for years [1–3], but few solve the double determination problem, level and shape of spectrum. When scaling is implemented, uncertainty computation generally assumes independence between scaling and shape determination. Some codes clearly points out that the scaling factor uncertainty use is a bit tricky [4]. So a new adjustment code, CALMAR has been developed to perform both objectives, including rigorous uncertainty computations.

2. Implemented Method

The main objective for developing a new adjustment code was the determination of the two parameters for a multigroup spectrum:

- the shape, classically computed by well known adjustment codes listed in the ASTM standard [1].
- the level of this spectrum: absolute level of the reactor is not necessarily known from alternate measurements than foils activity.

Basically, the CALMAR implemented algorithm finds an optimum scaling factor to be applied to the prior spectrum. Then the code determines the shape of the spectrum with a least square formula.

[a] Corresponding author: gilles.gregoire@cea.fr

Table 1. Notations for equations.

symbol	meaning
C	scaling factor
ϕ^0	prior spectrum
Γ_{ϕ^0}	prior spectrum covariance matrix
$\hat{\phi}$	adjusted spectrum
τ^m	measured reaction rates
Γ_m	measured reaction rates covariance matrix
Σ	cross sections
Γ_{Σ_i}	covariance for i^{th} cross section
$\tau = \Sigma\phi$	computed rates, without scaling factor
Γ_{τ^c}	variance of computed rates

2.1 Notations

The following notations are used in the equations (Table 1).

2.2 Algorithm

The basic principle of the CALMAR algorithm is a chi-square minimization of Eq. (1).

$$\chi^2 = (\tau^m - C\tau)^T W_{\Sigma M}(\tau^m - C\tau) + (\phi - \phi^0)^T \Gamma_{\phi^0}^{-1}(\phi - \phi^0). \tag{1}$$

The weighting matrix $W_{\Sigma M}$ (Eq. (2)) is a combination of the experimental uncertainties and computed rates uncertainties (only cross section part).

$$W_{\Sigma M} = \left[\Gamma_m + C^2 \left(\Gamma_{\tau^c}|\phi\right)\right]^{-1}. \tag{2}$$

The Eq. (1) is a chi-square realisation and has its minimum equal to (3)

$$\chi^2_{min} = (\tau^m - C\tau^0)^T W (\tau^m - C\tau^0) \tag{3}$$

where W is the complete weighting matrix (Eq. (4))

$$W = \left[\Gamma_m + C^2 \Gamma_{\tau^c}\right]^{-1}. \tag{4}$$

Computation of the scaling factor C
The C scaling factor is determined by minimizing (3). If we consider that $\Gamma_{\tau^c}|\phi$ is independent from $\hat{\phi}$, then C can be determined in one step. In our algorithm, we update the cross section part of the uncertainty. The Γ_{τ^c} is then computed as Eq. (5)

$$\Gamma_{\tau^c} = \Sigma\Gamma_{\phi^0}\Sigma^T + \mathrm{diag}\left(\hat{\phi}\Gamma_{\Sigma_i}\hat{\phi}^T\right) + \mathrm{diag}\left[\mathrm{trace}\left(\Gamma_{\phi^0}\Gamma_{\Sigma_i}\right)\right] \tag{5}$$

Computation of the shape
Once the scaling factor is computed, the shape is computed with a least square formula, assuming a scaled prior spectrum. The shape is then computed according to Eq. (6)

$$\hat{\phi} = \phi^0 + C\Gamma_{\phi^0}\Sigma W(\tau^m - C\tau^0). \tag{6}$$

Iterative process
The adjustment procedure is iterative, due to Γ_{τ^c} dependency to $\hat{\phi}$. Procedure is stopped when convergence is obtained on C scaling factor. The process is shown on Fig. 1.

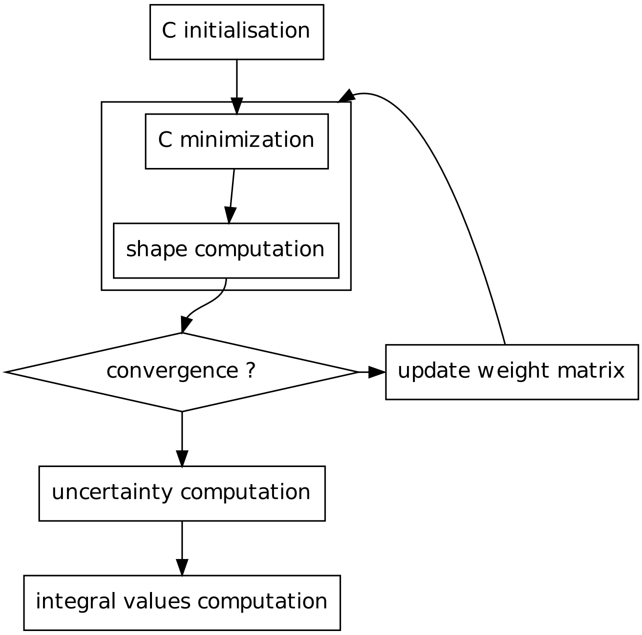

Figure 1. CALMAR iterative process.

Uncertainties

Uncertainties are computed using the conditional expectation formalism. The STAYSL code and its derivatives make use of the Schur complement (or block matrix inversion lemma) for computing the adjusted covariance matrix. In CALMAR algorithm, the inversion lemma is used many times to determine:

- scaling factor variance ($\Gamma_{\hat{C}}$)
- the covariance matrix of the shape ($\Gamma_{\hat{\phi}}$) with the remaining available experimental uncertainties. One part has been already taken into account to determine the scaling factor.
- the covariance matrix between the shape and the scaling factor ($\Gamma_{\hat{C},\hat{\phi}}$).

The expectation and the covariance of the scaled spectrum is computed using these three previous items. One pays attention to the fact that C and $\hat{\phi}$ are correlated, so expectation of the product should be computed as follow (7)

$$\mathbb{E}\left[\hat{C}\hat{\phi}\right] = \mathbb{E}\left[\hat{C}\right]\mathbb{E}\left[\hat{\phi}\right] + \Gamma_{\hat{C},\hat{\phi}}. \tag{7}$$

The covariance of the product is deduced from higher-order moments formula for multivariate distributions.

3. Code Structure

3.1 Development Environment

The CALMAR library has been developed using C^{++} language, with the of help the ROOT framework. ROOT [5] is a C^{++} framework dedicated to data analysis. It is developed by the CERN to analyse high energy physics experiments. The main characteristics are:

- framework of C^{++} classes

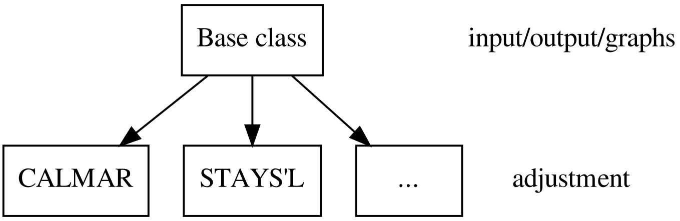

Figure 2. CALMAR library hierarchy class.

- data analysis of huge datasets
- C^{++} interpreter
- linear algebra
- minimization algorithms
- lots of graphs possibilities.

3.2 Structure

The library has been constructed as a C^{++} class hierarchy (Fig. 2). The base class is dealing with input/output, memory management and graphs functionalities (Fig. 3). Constructors of the base class are dedicated to each use of the code. CALMAR uses dynamic memory management so energy grid is free and number of reactions is only limited by memory. CALMAR makes an intensive use of ROOT's linear algebra classes and minimization functions.

Each derived class implements a different adjustment method: the CALMAR method and the original STAYS'L method. To add a new adjustment method, a new derived class should be created, with only one computation function to overload.

3.3 Execution of the Code

The use of the CALMAR library is versatile and three modes are available now:

- stand-alone code with text data files
- function calls to the library within a C^{++} code. In this case no files are needed.
- C^{++} script through the ROOT interpreter.

The last mode is the more interesting because the script is in fact an input job file: it contains all the directives. Some more outputs can be added to the job file to compute some more rates (gas production, DPA, ...). The coupling with other codes is also possible, with all the support of ROOT framework. For example, the TRIPOLI4 Monte-Carlo code [6] supports various output formats including XML and ROOT trees. The CALMAR script can process easily these data as an input for the adjustment code.

4. Validation and Test Cases

4.1 Validation

Basic numerical validation is done to check algorithm stability against the following items: convergence speed, role of matrix condition number for accuracy. Sensitivity studies are forecast to validate the solution in case of ill-posed problems. Obtaining a full rank prior covariance matrix always needs manual tuning [7]. Cross section covariances does not have the same level of quality for all reactions even a particular attention has been made in the last version of IRDFF [8].

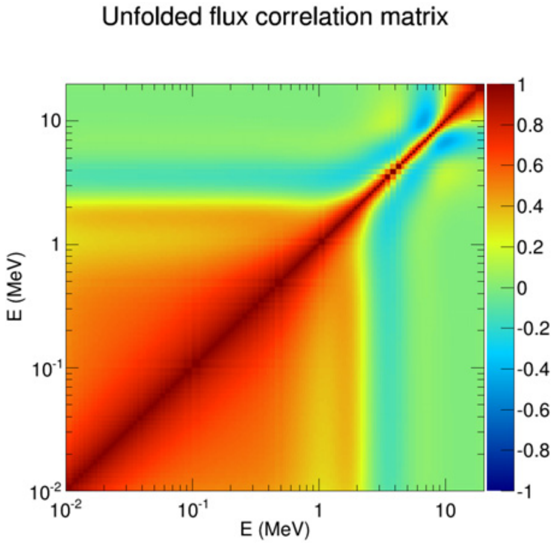

Figure 3. Sample output of adjusted flux correlation.

4.2 Test Cases

A validation over experimental datasets is in preparation. Code will be tested over experimental datasets of foils from thermal to fast. A dataset obtained in CALIBAN reactor is described in [9]. A comparison with the STAYS'L code will be done. These test cases will be included in the CALMAR package. This type of package has been provided with the IRDF90 library as Neutron Metrology File (NMF-90, [10]) including codes and test cases. In our case, we will provide updated datasets with the new IRDFF library.

5. Nuclear Data Format

Today, cross section processing is done by an external tool with the same functionalities as the X333 tool provided with IRDF90 [11]. The intended developments of the CALMAR library are to include ENDF reading routines in order to directly process the IRDFF library. Nuclear data processing is still consuming time, even for end-users and not really standardized. The general NJOY[12] tool has many parameters, increasing the risk of a mistake for dosimetry end-users. The paragraph describing NJOY in [13] is quite symptomatic of code complexity and a post processor (NJpp) is still needed to interface other codes. On the other side, specialized tools provided with dosimetry library (X333) are not maintained (updated for new format) any more. JANIS is user friendly tool but lacks automation features. Rewriting a specialised tool is time consuming, with the risk of misunderstanding some part of the ENDF102[14] format. Format error in some older libraries should be also supported.

As a dosimetry user, the need is a modular tool to access to nuclear data, with different level of access: end-user, intermediate and even evaluator. Modern programming languages allows this flexibility. This is one of the features of the new Generalized Nuclear Data (GND, [15]) format for libraries. This new format has the following features:

- GND format is XML (eXtended Markup Language). XML can be automatically checked with a grammar file: mistakes on the format can be avoided.
- Many standard tools are available for XML (editors, viewers): GND is more human readable.

- GND developers not only define a file format but already define access routines. These routines defines an Application Programming Interface (API). This API brings flexibility and code interfacing possibilities with high level reading, writing, graphing routines. It is written now in python language but a C/C++ API will be provided next.

Dosimetry tools should benefit from these developments, as the validation of the API is done by GND developers.

6. Conclusion

A new spectrum adjustment library, CALMAR, has been developed to perform simultaneous adjustment and scaling. This library is versatile in its use, thanks to the C++ ROOT framework that allows interactive sessions. After testing and validation, it will be available to the community with selected cases. Future developments will include the GND format, avoiding cross section post-processing tools.

The authors would like to thank Jean-Marc Martinez from CEA Saclay for his support on mathematical aspects, especially in uncertainty modelling.

References

[1] ASTM-E944 (ASTM International, West Conshohocken, PA, 2013), `http://www.astm.org`

[2] W. McElroy, S. Berg, T. Crockett, R. Hawkins, Tech. Rep. AFWL-TR-67-41-Vol1 Air Force Weapons Laboratory, Kirtland, NM (1967)

[3] F. Perey, Tech. Rep. TM-6062 ENDF-254, ORNL (1977)

[4] E. Szondi, H. Nolthenius, E. Zsolnay, Tech. Rep. BME-NTI-223-95, Institute of Nuclear Techniques, Technical University of Budapest (1995)

[5] I. Antcheva, al., Computer Physics Communications **182**, 1384 (2011)

[6] E. Brun, E. Dumonteil, F. Hugot, N. Huot, C. Jouanne, Y. Lee, F. Malvagi, A. Mazzolo, O. Petit, J. Trama et al., *Overview of TRIPOLI-4 version 7 continuous-energy Monte Carlo transport code*, in *Proceedings of the ICAPP 2011 Conference* (Nice, France, 2011)

[7] J. Williams, D. Vehar, P. Griffin, D. King, *Covariances Matrices for calculated neutron spectra and measured dosimeter responses*, in *13th Int. Symp. on Reactor Dosimetry*, edited by W. Scientific (Alkmaar,Nederlands, 2008), pp. 343–351, ISBN 978-981-4271-10-3

[8] E. Zsolnay, R.C. Noy, H. Nolthenius, A. Trkov, Tech. Rep. IAEA-NDS-0616, IAEA (2012)

[9] P. Casoli, al., *Characterization of the CALIBAN Critical Assembly Neutron Spectra using several Adjustment Methods based on Activation Foils Measurement*, in *15th Int. Symp. on Reactor Dosimetry* (Aix en Provence,France, 2014), to be published

[10] N. Kocherov, Tech. Rep. INDC-NDS-347, IAEA (1996)

[11] E. Szondi, H. Nolthenius, Tech. Rep. BME-NTI 222/95, Institution of Nuclear Techniques, Technical University of Budapest (1995)

[12] D. Muir, R. Boicourt, A. Kahler, Tech. Rep. LA-UR-12-27079, Los Alamos National Laboratory (2012)

[13] L. Greenwood, C. Johnson, Tech. Rep. PNNL-22253, Pacific Northwest National Laboratory, Richland, WA (2013)

[14] M. Herman, Tech. Rep. BNL-90365-2009, National Nuclear Data Center Brookhaven National Laboratory (2009)

[15] C. Mattoon, B. Beck, N. Patel, N. Summers, G. Hedstrom, D. Brown, Nuclear Data Sheets **113**, 3145 (2012), special Issue on Nuclear Reaction Data