A modified Generalized Least Squares method for large scale nuclear data evaluation

Georg Schnabel, Helmut Leeb

► To cite this version:

Georg Schnabel, Helmut Leeb. A modified Generalized Least Squares method for large scale nuclear data evaluation. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2017, 841, pp.87 - 96. 10.1016/j.nima.2016.10.006 . hal-01465923

HAL Id: hal-01465923

<https://hal.science/hal-01465923>

Submitted on 13 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A modified Generalized Least Squares method for large scale nuclear data evaluation

Georg Schnabel^{a,b,*}, Helmut Leeb^b

^a *Irfu, CEA, Université Paris-Saclay, 91191 Gif-sur-Yvette, France*

^b *Atominstytut, TU Wien, Vienna, Austria*

ARTICLE INFO

Keywords:

Nuclear data evaluation
Generalized Least Squares
Bayesian statistics

ABSTRACT

Nuclear data evaluation aims to provide estimates and uncertainties in the form of covariance matrices of cross sections and related quantities. Many practitioners use the Generalized Least Squares (GLS) formulas to combine experimental data and results of model calculations in order to determine reliable estimates and covariance matrices. A prerequisite to apply the GLS formulas is the construction of a prior covariance matrix for the observables from a set of model calculations. Modern nuclear model codes are able to provide predictions for a large number of observables. However, the inclusion of all observables may lead to a prior covariance matrix of intractable size. Therefore, we introduce mathematically equivalent versions of the GLS formulas to avoid the construction of the prior covariance matrix. Experimental data can be incrementally incorporated into the evaluation process, hence there is no upper limit on their amount. We demonstrate the modified GLS method in a tentative evaluation involving about three million observables using the code TALYS. The revised scheme is well suited as building block of a database application providing evaluated nuclear data. Updating with new experimental data is feasible and users can query estimates and correlations of arbitrary subsets of the observables stored in the database.

1. Introduction

The increasing performance of computers and the availability of computer clusters enable nowadays simulations of complex nuclear systems such as reactors. These simulations guide design choices regarding efficiency and safety. An important prerequisite for such simulations are reliable estimates of reaction cross sections and emission spectra. There is also an increasing demand for uncertainty information in the form of covariance matrices. Usually, perturbation theory is used to propagate these uncertainties through simulations in order to obtain the corresponding uncertainties of the integral observables (e.g. [1]). An alternative to perturbation theory is the total Monte Carlo method [2]. The latter is able to deal with non-Gaussian distributions but is computationally more demanding [3]. Therefore, perturbation theory using covariance matrices remains a viable option for uncertainty propagation.

Estimates and covariance matrices are generated with the help of an evaluation method. An evaluation method is a statistical procedure which combines results of model calculations and experimental data to obtain an improved set of estimates and covariance matrices for the observables. Several evaluation methods were proposed (e.g. [4–13]) which differ in their details but commonly employ the framework of

Bayesian statistics. For an overview of the different methods see e.g. [14] and [15, pp. 61–74].

The basic formulas used by most of these methods are often referred to as Generalized Least Squares (GLS) formulas. There are two approaches to include information of a nuclear model into the GLS formulas. In the linearization approach, one linearizes the model and assumes a multivariate normal distribution for the model parameters (e.g. [5,7]). In the sampling approach, one creates a sample of model calculations with varied model parameters and constructs a multivariate normal distribution for the predicted observables (e.g. [9,11]). The linearization approach requires only a small number of model parameters to be updated. The advantage of the sampling approach is that it allows evaluations to better resemble experimental data because the final evaluation does not necessarily need to correspond to a model prediction with a specific choice of model parameters [15, p. 37–41]. However, a disadvantage of the sampling approach is its restriction on the number of observables which can be taken into account.

Nuclear model codes such as TALYS [16,17] are able to predict a number of observables in the order of millions if one considers a broad range of incident energies (e.g. between 1 and 200 MeV). Updating such a number of observables and the associated covariance matrix is troublesome using the sampling approach in combination with the

* Corresponding author at: CEA/Saclay, DRF/Irfu/SPhN, 91191 Gif-sur-Yvette Cedex, France

standard GLS formulas.

The main obstacle for the large scale application of the GLS formulas is the prior covariance matrix for the observables. As an example, the GANDR code suite [18] developed at the International Atomic Energy Agency [19] allows the consistent evaluation of 91 000 observables on a personal computer, which involves a 91 000×91 000 covariance matrix taking about 30 gigabytes of storage space. If the number of observables were increased to millions, the covariance matrix would inflate to a size in the order of 10 terabytes. Consequently, the computation time would surpass an acceptable level.

In this paper, we present a revised computation scheme to evaluate the GLS formulas which avoids the explicit construction of the prior covariance matrix. We exploit the fact that the $L \times L$ covariance matrix for L observables is constructed from a much smaller number N of model calculations (e.g. $L = 10^6$ and $N = 10^3$). Concerns may be raised that estimating a covariance matrix of larger dimension than the number of model calculations is problematic from a statistical point of view. However, model predictions are determined by a limited number of parameters, hence the intrinsic dimension of the covariance matrix is much lower than its nominal dimension. The number of model calculations must be only sufficiently larger than the intrinsic dimension to justify the application of the GLS formulas.

In order to represent an updated state of knowledge, neither updated estimates of observables nor corresponding covariance matrices have to be stored. It suffices to keep (1) a vector $\vec{w}_i$ containing a weight for each of the N model calculations, (2) a $N \times N$ covariance matrix $\mathbf{W}_i$ referring to model calculations instead of observables, and (3) the results of model calculations.

Experimental data can be included in several stages, which is a significant extension to the work presented in [20]. The time need to update the information of millions of observables based on experimental data is in the order of seconds on a modern personal computer. Estimates of observables and relevant pieces of the full evaluated covariance matrix can be efficiently reconstructed from $\vec{w}_i$, $\mathbf{W}_i$ and the model calculation results.

These features make the revised GLS scheme well suited to be implemented as a database application. Estimates, uncertainties and correlations of arbitrary subsets of observables can be computed on demand. Due to the large size of a full covariance matrix for all observables, libraries such as JEFF [21] or TENDL [22] provide covariances only for the most significant reaction channels. On the other hand, the flexibility of the envisaged database application would give the user the freedom to retrieve any covariance matrix which is important for his application.

Finally, we note that the idea to avoid the construction of a possibly large matrix and instead to work with the vectors from which it would be constructed has already been successfully applied in other fields and contexts, see e.g. the Ensemble Kalman Filter [23,24] and the L-BFGS method for optimization [25,26].

The structure of this paper is as follows. Section 2 provides a concise sketch of the GLS method and elaborates on the required modifications to enable the evaluation of a large number of observables. Section 3 compares the time and storage requirement of the standard approach and the modified approach. Section 4 contains details about the application of the revised GLS method in an evaluation of neutron-induced reactions on ^{181}Ta with 854 experimental data points and about three million observables predicted by the nuclear models code TALYS. Section 5 underlines important aspects of the revised evaluation scheme. Section 6 provides conclusions and an outlook.

2. Method

2.1. Basic method

We give a brief sketch of the GLS method, which will serve as the

basis to discuss the modifications for a large number of observables. A derivation of the GLS formulas is given in Appendix A.

The purpose of the GLS method is to infer best estimates of observables and an associated covariance matrix by taking into account both the results of model calculations and experimental data. In the GLS method, both experimental uncertainties and the variations of model predictions due to variations of parameters are described by multivariate normal distributions.

The GLS method takes place in three steps: First, the results of model calculations are used to construct the so-called *prior*, defined by a prior mean vector and a prior covariance matrix. Second, one assembles the measured values of experiments into a measurement vector and constructs an associated covariance matrix based on the knowledge about statistical and systematic measurement errors. And third, the quantities of the first and second steps are combined using the GLS formulas to obtain the *posterior*, which comprises a posterior mean vector and a posterior covariance matrix. Following, we discuss each step in more detail and provide the mathematical formulas.

Construction of the prior. Information from model calculations has to be represented by a vector of best estimates $\vec{\sigma}_0$ and an associated covariance matrix $\mathbf{A}_0$. In order to determine these two quantities, one performs a certain number N of model calculations with different model parameter sets $\{\vec{p}_i\}_{i=1..N}$. For example, using TALYS the vectors $\vec{p}_i$ contain the parameters of the optical model in the global parametrization of Koning–Delaroche [27]. The values of the parameter vectors are usually drawn from a uniform or a multivariate normal distribution. Different procedures have been suggested to determine sensible ranges (in the case of a uniform distribution) or a center vector and a covariance matrix (in the case of a multivariate normal distribution) for the parameter vector. These procedures either make use of constraints stemming from physics considerations [11] or specify the parameter ranges in such a way that the associated predictions cover the majority of available experimental data [28].

Once the model prediction vector $\vec{x}_i$ is available for each model parameter vector $\vec{p}_i$ in the set $\{\vec{p}_i\}_{i=1..N}$, the prior mean vector $\vec{\sigma}_0$ and the prior covariance matrix $\mathbf{A}_0$ are calculated by

$$\vec{\sigma}_0 = \frac{1}{N} \sum_{i=1}^N \vec{x}_i \quad \text{and} \quad \mathbf{A}_0 = \frac{1}{N} \sum_{i=1}^N (\vec{x}_i - \vec{\sigma}_0)(\vec{x}_i - \vec{\sigma}_0)^T. \quad (1)$$

Another possible way to set up $\vec{\sigma}_0$ is to make a model calculation based on the mean vector of the parameter distribution and to use the result as prior mean vector $\vec{\sigma}_0$ for the observables.

One can see from these formulas that the vector $\vec{\sigma}_0$ is of the same dimension as the model prediction vectors $\vec{x}_i$. The position of a value in these vectors indicates to which observable this value refers. Observables predicted by a nuclear model are typically cross sections, angle-differential cross sections and spectra. However, any continuous observable predicted by a model may be used.

Experimental data. The knowledge from experimental data has to be cast into a measurement vector $\vec{\sigma}_{\text{exp}}$ and an associated experiment covariance matrix $\mathbf{B}$. The measurement vector contains the values obtained in the experiments. Of course, one has to keep track to which observables these values relate. The level of trust in the measured values is used to set up a covariance matrix $\mathbf{B}$. The squared statistical uncertainties due to a limited number of detected events have to be placed in the diagonal of the covariance matrix. Systematic errors, which tie together uncertainties of different values, lead to contributions to both diagonal and off-diagonal elements. For example, an uncertainty about the proper calibration of the employed detector translates to a systematic uncertainty of all cross sections of different atomic nuclei measured with the detector. More information about the construction of covariance matrices can be found in [29].

Performing the update. The information from model calculations represented by a prior center vector $\vec{\sigma}_0$ and a prior covariance matrix $\mathbf{A}_0$ provides one source of information; the information from experiments given by a measurement vector $\vec{\sigma}_{\text{exp}}$ and an experiment

covariance matrix $\mathbf{B}$ another one. The application of the Bayesian update formulas allows us to combine these quantities to obtain the posterior center vector $\vec{\sigma}_i$ and the posterior covariance matrix $\mathbf{A}_i$. If new measurement data are available, one can use the obtained posterior as the new prior. Hence we will use the notation $\vec{\sigma}_i$ and $\mathbf{A}_i$ to denote the prior of the current iteration. The Bayesian update formula to calculate the posterior mean vector $\vec{\sigma}_{i+1}$ of the current iteration reads (see Appendix)

$$\vec{\sigma}_{i+1} = \vec{\sigma}_i + \mathbf{A}_i \mathbf{S}^T (\mathbf{S} \mathbf{A}_i \mathbf{S}^T + \mathbf{B})^{-1} (\vec{\sigma}_{\text{exp}} - \mathbf{S} \vec{\sigma}_i). \quad (2)$$

The observables in $\vec{\sigma}_i$ are usually defined on a different energy/angle grid than those in $\vec{\sigma}_{\text{exp}}$. The sensitivity matrix $\mathbf{S}$ maps the observables from the model grid associated with $\vec{\sigma}_i$ to the grid associated with $\vec{\sigma}_{\text{exp}}$. Depending on the choice of $\mathbf{S}$, different mapping schemes such as linear or spline interpolation are possible. In the opinion of the authors, linear interpolation has to be preferred over other interpolation schemes because it leads to very sparse matrices $\mathbf{S}$, which is computationally beneficial. The reduced smoothness compared to e.g. splines can be compensated by using more grid points.

The posterior covariance matrix $\mathbf{A}_{i+1}$ with a reduced uncertainty is given by (see Appendix)

$$\mathbf{A}_{i+1} = \mathbf{A}_i - \mathbf{A}_i \mathbf{S}^T (\mathbf{S} \mathbf{A}_i \mathbf{S}^T + \mathbf{B})^{-1} \mathbf{S} \mathbf{A}_i. \quad (3)$$

These update formulas are convenient tools in nuclear data evaluation due to their simplicity. The numerically most complex operation is the inversion of an $M \times M$ matrix with M being the number of measurements and thus the dimension of $\vec{\sigma}_{\text{exp}}$. The number L of observables in the vector $\vec{\sigma}_i$ is from a computational perspective rather uncritical, because the corresponding $L \times L$ covariance matrix $\mathbf{A}_i$ is only involved in a matrix product. However, admissible computation time and storage space pose practical limits on the number of observables. For instance, a TALYS calculation up to 200 MeV yields millions of predicted observables. The storage requirement of the associated covariance matrix $\mathbf{A}_i$ would be in the order of 10 terabytes.

2.2. Modified method

Following we describe the modified method, which enables evaluations with a much larger number of observables than the basic method. The methods are mathematically equivalent, which is shown in Section 2.4. Like the basic method, the input quantities of the modified method are the mean prediction vector $\vec{\sigma}_0$ (see Eq. (1)), the model prediction vectors $\vec{x}_k$ and the measurement vector $\vec{\sigma}_{\text{exp}}$ with the associated covariance matrix $\mathbf{B}$. Three execution blocks can be distinguished in the modified method: constructing the prior, performing the update, and making predictions.

Constructing the prior. The quantities being updated in the modified method are a vector of weights $\vec{w}_i$ and an associated covariance matrix $\mathbf{W}_i$. The dimensions are N and $N \times N$, respectively, with N being the number of model calculations. The index i indicates that updates can be performed in several stages, where a new experimental data set is included in each stage. In the very first update, the assignments are $\vec{w}_0 = \vec{0}$ and $\mathbf{W}_0 = \mathbf{1}$, where $\mathbf{1}$ is the $N \times N$ identity matrix.

Performing the update. Each update step requires the calculation of the matrix

$$\mathbf{V} = (\vec{v}_1, \vec{v}_2, \dots, \vec{v}_N) \quad (4)$$

with $v_k = \mathbf{S}(\vec{x}_k - \vec{\sigma}_0)$. The matrix $\mathbf{S}$ maps observables from the model grid to the energies and angles of the measurement vector $\vec{\sigma}_{\text{exp}}$. Thus, the i th column of $\mathbf{V}$ contains the prediction of a model calculation based on parameter set $\vec{p}_i$ for the experimental data set. Depending on the number of experimental data points M and the number of observables L predicted by the model code, the $M \times L$ matrix $\mathbf{S}$ can attain a significant size. However, $\mathbf{S}$ is usually very sparse and this feature allows for both efficient storage and computation. We elaborate

on this point in more detail in Section 2.3. The $M \times N$ matrix $\mathbf{V}$ fits in typical evaluation scenarios completely into main memory. If not, the matrix products in the following equations can be performed block wise, reading the blocks of $\mathbf{V}$ sequentially from the hard drive.

Introducing the abbreviation

$$\mathbf{X}_i = \left(\frac{1}{N} \mathbf{V} \mathbf{W}_i \mathbf{V}^T + \mathbf{B} \right)^{-1}, \quad (5)$$

we can write the update formulas as

$$\vec{w}_{i+1} = \vec{w}_i + \mathbf{W}_i \mathbf{V}^T \mathbf{X}_i \left(\vec{\sigma}_{\text{exp}} - \mathbf{S} \vec{\sigma}_0 - \frac{1}{N} \mathbf{V} \vec{w}_i \right), \quad (6)$$

and

$$\mathbf{W}_{i+1} = \mathbf{W}_i - \frac{1}{N} \mathbf{W}_i \mathbf{V}^T \mathbf{X}_i \mathbf{V} \mathbf{W}_i. \quad (7)$$

Noteworthy, the modified method updates the $N \times N$ matrix $\mathbf{W}_i$ whereas the basic method updates the $L \times L$ matrix $\mathbf{A}_i$. Evaluations may involve $N=1000$ model calculations and $L=10^6$ observables. The smaller size of the matrix to be updated in the modified method is the main reason why the modified method is much faster than the basic method. We compare the differences between the methods with respect to execution time and storage requirement thoroughly in Section 3.

Making predictions. Let $\mathbf{S}^P$ be the matrix that maps the observables from the model grid to the energies and angles of interest. Using the weight vector $\vec{w}_i$, the mean prediction vector $\vec{\sigma}_0$ and the model prediction vectors $\vec{x}_k$, the predictions for the observables of interest are

$$\vec{\sigma}_i^P = \mathbf{S}^P \vec{\sigma}_0 + \frac{1}{N} \mathbf{V}^P \vec{w}_i, \quad (8)$$

where $\mathbf{V}^P$ is defined as in Eq. (4) with $\mathbf{S}$ replaced by $\mathbf{S}^P$. The associated covariance matrix reads

$$\mathbf{A}_i^P = \frac{1}{N} \mathbf{V}^P \mathbf{W}_i (\mathbf{V}^P)^T. \quad (9)$$

In contrast to the basic method, there is no need to evaluate the complete covariance matrix $\mathbf{A}_i$ during updating. Blocks may only be computed when they are themselves of interest.

2.3. Efficient mapping

Both modified update equations (6) and (7) contain the matrix $\mathbf{V}$ whose columns are given by $\mathbf{S} \vec{v}_k = \mathbf{S} \vec{x}_k - \mathbf{S} \vec{\sigma}_0$. The sensitivity matrix $\mathbf{S}$ is of dimension $M \times L$ where M is the number of elements in $\vec{\sigma}_{\text{exp}}$ and L is the number of elements in $\vec{\sigma}_0$. For instance, assuming $L=10^6$ and $M=5000$, the matrix $\mathbf{S}$ needs about 40 gigabytes of storage space. However, $\mathbf{S}$ is usually very sparse and one could exploit this sparsity to reduce the storage size. Assuming linear interpolation for cross sections, which are functions of incident energy, and bilinear interpolation for angle-differential cross sections and spectra, which are functions of incident energy and emission energy or scattering angle, the sensitivity matrix contains not more than four non-zero elements in each row. Storing just the positions and the values of the non-zero elements, instead of $M \times L$ numbers only $3 \times 4 \times M$ numbers have to be stored. The sparse representation also speeds up the computation of $\mathbf{S} \vec{x}_k$ and $\mathbf{S} \vec{\sigma}_0$ by the same factor.

Furthermore, it is possible to avoid completely the explicit determination of $\mathbf{S}$. The only purpose of $\mathbf{S}$ is to interpolate observables from the model grid to the experiment grid. The multiplication with $\mathbf{S}$ can therefore be replaced by directly performing the required interpolation.

Because the task to compare model predictions with experimental data is quite common, scripts to interpolate results of model calculations to the energies and angles of the experimental data are often available. Building on existing scripts, the modified GLS method can be quickly implemented in a high-level language like R or MatLab.

2.4. Derivation of the modified method

The modified method presented in Section 2.2 is the direct consequence of a reformulation which is outlined in the following. For a concise notation, we introduce the shifted prediction vectors $\vec{u}_k = \vec{x}_k - \vec{\sigma}_0$ and reiterate the abbreviation $\vec{v}_k = \mathbf{S}\vec{u}_k$. We bundle these quantities to matrices of dimension $L \times N$ and $M \times N$ respectively,

$$\mathbf{U} = (\vec{u}_1, \vec{u}_2, \dots, \vec{u}_N) \quad \text{and} \quad \mathbf{V} = (\vec{v}_1, \vec{v}_2, \dots, \vec{v}_N). \quad (10)$$

Key idea of the modified GLS method is to avoid the explicit computation of the prior covariance matrix. Therefore, we introduce an $N \times N$ matrix $\mathbf{W}_i$ and rewrite the prior covariance matrix $\mathbf{A}_i$ of some iteration i as

$$\mathbf{A}_i = \frac{1}{N} \sum_{j=1}^N \sum_{k=1}^N \vec{u}_j \mathbf{W}_{i,jk} \vec{u}_k^T = \frac{1}{N} \mathbf{U} \mathbf{W}_i \mathbf{U}^T. \quad (11)$$

In the very first update, the choice $\mathbf{W}_0 = \mathbf{I}$ yields the prior covariance $\mathbf{A}_0$ given in Eq. (1). In later update iterations, the respective matrix $\mathbf{W}_i$ will in general also contain non-diagonal elements. We use this decomposition of $\mathbf{A}_i$ to express the inverted matrix occurring in Eqs. (2) and (3) as

$$\mathbf{X}_i = (\mathbf{S} \mathbf{A}_i \mathbf{S}^T + \mathbf{B})^{-1} = \left(\frac{1}{N} \mathbf{V} \mathbf{W}_i \mathbf{V}^T + \mathbf{B} \right)^{-1}. \quad (12)$$

Using the introduced notation, we can rewrite Eq. (2)

$$\begin{aligned} \vec{\sigma}_{i+1} &= \vec{\sigma}_i + \frac{1}{N} \mathbf{U} \mathbf{W}_i \mathbf{V}^T \mathbf{X}_i (\vec{\sigma}_{\text{exp}} - \mathbf{S} \vec{\sigma}_i) \\ &= \vec{\sigma}_i + \frac{1}{N} \sum_{k=1}^N \vec{u}_k [\vec{u}_k^T \mathbf{U} \mathbf{W}_i \mathbf{V}^T \mathbf{X}_i (\vec{\sigma}_{\text{exp}} - \mathbf{S} \vec{\sigma}_i)]. \end{aligned} \quad (13)$$

The expression in the squared bracket at the end is just a number. Therefore, the difference $\vec{\sigma}_{i+1} - \vec{\sigma}_i$ can be described as a linear combination of the shifted model prediction vectors $\vec{u}_k$.

This feature allows us to express $\vec{\sigma}_i$ of any iteration i in the form

$$\vec{\sigma}_i = \vec{\sigma}_0 + \frac{1}{N} \sum_{k=1}^N \vec{u}_k w_{ik} = \vec{\sigma}_0 + \frac{1}{N} \mathbf{U} \vec{w}_i, \quad (14)$$

where the vectors $\vec{w}_i$ have to be determined during updating. In the first update iteration, we have $\vec{w}_0 = \vec{0}$. Inserting Eq. (14) into Eq. (13) leads to

$$\begin{aligned} \vec{\sigma}_{i+1} - \vec{\sigma}_0 &= \frac{1}{N} \mathbf{U} \vec{w}_i + \frac{1}{N} \mathbf{U} \mathbf{W}_i \mathbf{V}^T \mathbf{X}_i \left(\vec{\sigma}_{\text{exp}} - \mathbf{S} \vec{\sigma}_0 - \frac{1}{N} \mathbf{V} \vec{w}_i \right) \\ &= \frac{1}{N} \mathbf{U} \left\{ \vec{w}_i + \mathbf{W}_i \mathbf{V}^T \mathbf{X}_i \left(\vec{\sigma}_{\text{exp}} - \mathbf{S} \vec{\sigma}_0 - \frac{1}{N} \mathbf{V} \vec{w}_i \right) \right\} = \frac{1}{N} \mathbf{U} \vec{w}_{i+1}. \end{aligned} \quad (15)$$

The result of this update formula can be written as the product of $\mathbf{U}$ and an updated vector of weights $\vec{w}_{i+1}$. Therefore, it suffices to consider the reduced update equation

$$\vec{w}_{i+1} = \vec{w}_i + \mathbf{W}_i \mathbf{V}^T \mathbf{X}_i \left(\vec{\sigma}_{\text{exp}} - \mathbf{S} \vec{\sigma}_0 - \frac{1}{N} \mathbf{V} \vec{w}_i \right), \quad (16)$$

which is exactly Eq. (6) in the outline of the modified method in Section 2.2.

We proceed with the calculation of $\mathbf{A}_{i+1}$. Inserting Eq. (11) into the update Eq. (3) gives

$$\mathbf{A}_{i+1} = \frac{1}{N} \mathbf{U} \mathbf{W}_i \mathbf{U}^T - \frac{1}{N^2} \mathbf{U} \mathbf{W}_i \mathbf{V}^T \mathbf{X}_i \mathbf{V} \mathbf{W}_i \mathbf{U}^T \quad (17)$$

$$\mathbf{A}_{i+1} = \frac{1}{N} \mathbf{U} \left(\mathbf{W}_i - \frac{1}{N} \mathbf{W}_i \mathbf{V}^T \mathbf{X}_i \mathbf{V} \mathbf{W}_i \right) \mathbf{U}^T = \frac{1}{N} \mathbf{U} \mathbf{W}_{i+1} \mathbf{U}^T. \quad (18)$$

Thus, we just consider the update equation

$$\mathbf{W}_{i+1} = \mathbf{W}_i - \frac{1}{N} \mathbf{W}_i \mathbf{V}^T \mathbf{X}_i \mathbf{V} \mathbf{W}_i, \quad (19)$$

whose structure resembles the original update equation (3). This update equation appears as Eq. (7) in the outline of the modified method in Section 2.2. Similar to the original update equations (2) and (3), also the modified equations (16) and (19) allow the inclusion of experimental data in several stages.

Predictions of observables can be computed by taking recourse to Eq. (14). Let $\mathbf{S}^P$ be the matrix to map from the model grid to the energies and angles of interest. Multiplying Eq. (14) from the left by $\mathbf{S}^P$ yields

$$\vec{\sigma}_i^P = \mathbf{S}^P \vec{\sigma}_0 + \frac{1}{N} \mathbf{V}^P \vec{w}_i \quad (20)$$

with $\vec{\sigma}_i^P = \mathbf{S}^P \vec{\sigma}_i$. Similarly, multiplying Eq. (18) from the left by $\mathbf{S}^P$ and from the right by $(\mathbf{S}^P)^T$ and substituting $(i+1) \rightarrow i$ gives

$$\mathbf{A}_i^P = \frac{1}{N} \mathbf{V}^P \mathbf{W}_i (\mathbf{V}^P)^T \quad (21)$$

with $\mathbf{A}_i^P = \mathbf{S}^P \mathbf{A}_i (\mathbf{S}^P)^T$ commonly known as *sandwich formula*.

3. Comparison of methods

Flow charts of the standard and the revised GLS scheme are shown in Fig. 1. The most important difference between them is that updating and making predictions are clearly separated processes in the revised scheme. To enable updating with experimental data in several stages in the standard scheme, the full covariance matrix $\mathbf{A}_i$ has to be updated within each stage. If the employed nuclear model predicts e.g. $L = 10^6$ different observables, the $L \times L$ posterior covariance matrix would need about 4 terabytes of storage space. While this amount of storage space may be available on a well-equipped personal computer, the calculation of the posterior covariance matrix and writing it to disk takes considerable time. How much time will be studied in the course of this section.

In contrast to that, an update in the revised scheme consists of updating the auxiliary quantities $\vec{w}_i$ and $\mathbf{W}_i$ of dimension N and $N \times N$, respectively, with N being the number of performed model calculations. Typically, N is in the order of 10^3 and therefore the storage requirement of $\mathbf{W}_i$ about 400 megabytes—a remarkable reduction compared to the earlier mentioned 4 terabytes of $\mathbf{A}_i$.

The auxiliary quantities $\vec{w}_i$ and $\mathbf{W}_i$ in combination with the vector $\vec{\sigma}_0$ and the model prediction vectors $\vec{x}_k$ completely characterize the current state of knowledge. On their basis, predictions of observables and associated covariance matrices *can be* calculated whenever needed. We emphasize again the important difference to the standard scheme, in which the full set of predictions and the associated covariance matrix *must be* calculated during updating.

Having outlined the conceptional differences between the two schemes, we want to test their performance in practice, such as their time need. Because the time need depends on the employed computer as well as the used programming language, we feel obliged to provide these technical details. The calculations presented in the next two subsections were performed on a computer equipped with an *Intel i5-3550* processor with four cores at 3.3 GHz, 32 gigabytes of main memory, and a solid-state drive. The throughput of the solid-state drive has been measured to be around 350 megabytes a second. We used the programming language *R* [30] to implement the matrix formulas. This language relies on the *Basic Linear Algebra Subprogram* routines to perform basic linear algebra operations. We employed the optimized *OpenBLAS* [31] implementation which makes use of all available cores in parallel. In order to work with matrices whose size exceeds available main memory, we employed the *bigmemory* [32] package. Using this package, matrices are stored in files but within *R* they can be treated as if they would be located in the main memory.

Fig. 1. Flow charts of the standard GLS approach and the revised GLS approach. The most important differences ($L \equiv$ number of predicted observables, $N \equiv$ number of model calculations): The matrix W_i is just $(N/L)^2$ the size of A_i . Updating and making predictions are disentangled in the revised scheme.

Table 1

Comparison of the time for the prior generation and the storage space requirement of the revised scheme and the standard scheme for different choices of the number N of model calculations and the number L of predicted observables. In the revised scheme, the matrices U and W_0 are considered, whereas in the standard scheme the matrix A_0 is taken into account. The size of the vector $\vec{\sigma}_0$ appearing in the prior generation process of both schemes is negligible compared to the former mentioned matrices and hence not considered in the timings. Numbers marked with an asterisk are extrapolations.

N	L	Revised scheme		Standard scheme	
		Time (s)	size (MByte)	Time (s)	Size (MByte)
1000	1000	0.081	15.266	0.072	7.645
1000	5000	0.285	45.784	1.124	190.811
1000	10 000	0.555	83.931	4.144	763.092
1000	20 000	1.092	160.225	17.224	3052.063
1000	40 000	2.170	312.813	72.295	12 207.642
2000	1000	0.162	45.792	0.126	7.645
2000	5000	0.595	106.827	1.737	190.811
2000	10 000	1.182	183.121	6.637	763.092
2000	20 000	2.207	335.709	25.760	3052.063
2000	40 000	4.357	640.884	114.611	12 207.642
2000	10^6	436.860	15 289.310	*71 763.00	*7 629 411.00

3.1. Prior generation

Both schemes contain a prior generation phase. In the standard scheme, this phase includes the construction of the prior mean vector $\vec{\sigma}_0$ and the associated prior covariance matrix A_0 . The prior generation in the revised scheme comprises the construction of $\vec{\sigma}_0$, the determination of the model prediction vectors $\vec{x}_k$ as well as the initializations of the vector $\vec{w}_0 = \vec{0}$ and the matrix $W_0 = 1$. We also include the generation of the matrix $U = (\vec{u}_1, \vec{u}_2, \dots, \vec{u}_N)$ with $\vec{u}_k = \vec{x}_k - \vec{\sigma}_0$ into the prior generation of the modified method. This matrix remains the same for each update and allows us to slightly speed up the calculation of V (see Eq. (4)). Once $\vec{\sigma}_0$ and U are available, the now redundant model prediction vectors $\vec{x}_k$ can be discarded.

We are interested in the required time of both schemes to generate the prior as a function of the number L of observables and the number N of model calculations. We do not account for the time need to perform the model calculations but assume that their results are already available in the form of N prediction vectors $\vec{x}_i$. Because the origin of the numbers in the prediction vectors does not influence the efficiency of the method, we initialized their elements with random numbers drawn from a uniform distribution in the range between zero and one. The computed quantities were always written to the solid-

state drive even if they would fit completely into main memory. Thereby, we accounted for the real use-case in which data have to be stored permanently for later usage.

Table 1 summarizes the obtained timings. The time and storage requirement as a function of the number L of observables grow more rapidly for the standard scheme than for the revised scheme. Since the matrix U is of dimension $N \times L$, time need and storage space scale linearly with L for the revised scheme. In contrast to that, building the $L \times L$ matrix A_0 in the standard scheme causes the time need and storage space to scale quadratically with L . This important difference permits a much larger number of observables in the revised scheme. For example, the last row of Table 1 shows that a prior on the basis of $N=2000$ model calculations and $L = 10^6$ observables can be created and stored in about eight minutes using the revised scheme. Extrapolating from the table, the prior generation for the same setup in the standard scheme can be expected to take about 20 h and needs about 8 terabytes of storage space (or 4 terabytes if exploiting the symmetry of A_i).

3.2. Updating

We proceed with the comparison of the time need for updating. The revised scheme requires the update of the quantities $\vec{w}_i$ and W_i of dimension N and $N \times N$, respectively. In the standard scheme, the quantities $\vec{\sigma}_i$ and $\vec{A}_i$ of dimension L and $L \times L$ have to be updated.

We use linear interpolation to map the observables from the model grid to the experiment grid. Linear interpolation leads to a $M \times L$ sensitivity matrix S with exactly two non-zero elements in each row, where M is the number of experimental data points in the vector $\vec{\sigma}_{\text{exp}}$. We exploit this sparsity in both schemes and only store positions and values of non-zero elements. This measure significantly reduces storage space. Furthermore, the information about the positions of non-zero elements enables to perform only the element-wise products that contribute. For example, only $2ML$ instead of ML^2 element-wise multiplications are needed to compute the product of the $M \times L$ matrix S with the $L \times L$ matrix A_i .

The priors for the two schemes were constructed as described in Section 3.1. Because the position and the specific choice for the non-zero values in the matrix S do not affect the efficiency of the method, we selected randomly two elements in each row and assigned to them random values drawn from a uniform distribution between zero and one.

Table 2 shows the time need for updating of both schemes. The most important difference is that the time need grows quadratically

Table 2

Comparison of the time need for updating between the revised scheme and the standard scheme as a function of the number L of observables and the number M of measurements. In the revised scheme, the time depends also on the number N of model calculations. Numbers with an asterisk are extrapolations.

L	M	Time (s)			
		Revised scheme			Standard scheme
		$N=1000$	$N=2000$	$N=4000$	
1000	1000	0.473	1.032	3.894	0.518
5000	1000	0.366	1.030	3.877	2.784
10 000	1000	0.370	1.024	3.906	9.896
20 000	1000	0.370	1.053	3.906	40.883
40 000	1000	0.372	1.043	3.900	190.015
1000	2000	1.043	2.314	6.784	2.355
5000	2000	1.025	2.280	6.796	7.663
10 000	2000	1.039	2.271	6.818	24.044
20 000	2000	1.035	2.311	6.773	93.235
40 000	2000	1.046	2.293	7.168	485.011
10^6	2000	1.041	2.294	*7.000	*304 132.000

Table 3

Comparison of the time need for making predictions between the revised scheme and the standard scheme as a function of the number L of observables. The time need of the revised scheme depends additionally on the number N of model calculations, and those of the standard scheme on the number M of experimental data points. Numbers with an asterisk are extrapolations.

L	Time (s)			
	Revised scheme		Standard scheme	
	$N=1000$	$N=2000$	$M=1000$	$M=2000$
1000	0.008	0.016	0.300	1.794
5000	0.046	0.092	0.639	2.956
10 000	0.095	0.195	1.092	4.491
20 000	0.195	0.397	1.944	7.496
40 000	0.407	1.052	3.645	13.790
10^6	14.401	28.794	*86.000	*309.000

with L in the standard scheme whereas it is independent from L in the revised scheme. This feature allows us to update the information of millions of observables within seconds in the revised scheme. In contrast to that, an update involving $L = 10^6$ observables would take about four days using the standard scheme.

3.3. Making predictions

Contrary to the revised scheme, making an update with measurements in the standard scheme implies computing the predictions of all L observables. As has been shown in the last subsection, updating a large number of observables in the standard scheme is time consuming. However, it is possible to skip the calculation of A_{i+1} and to only calculate $\vec{\sigma}_{i+1}$. Such an incomplete update can be performed much faster than a full update, but without A_{i+1} no further updates are possible afterward.

Here, we want to compare the time needs of an incomplete update in the standard scheme (see Eq. (2)) and the prediction of all L observables in the revised scheme (see Eq. (8) with $S^p = \mathbf{1}_{L \times L}$). The timings are reported in Table 3. The number of predicted observables is the same in both schemes. Because experimental information enters an incomplete update (see Eq. (2)), the timings of the standard scheme are also dependent on the number M of measurement points. The time need for a prediction in the revised scheme depends besides L on the number N of model calculations.

As can be seen from Table 3, the time need for predicting scales linearly with the number L of observables for both schemes. However, comparing the case $N=2000$ in the revised scheme with the case $M=1000$ in the standard scheme, the slope of the increase in time need is larger in the standard scheme. Noteworthy, the combined time for an update and a subsequent prediction in the revised scheme is still significantly smaller than the time for an incomplete update in the standard scheme.

Finally, we emphasize that it is possible to compute the predictions only for a subset of the observables. This means that the predictions of e.g. 1000 observables can be calculated in about 0.01 s, irrespective of the total number of observables considered in the update procedure.

4. Demonstration

The last section showed that the formulas of the revised scheme can be much faster evaluated than those of the standard scheme. In this section, we provide details about a tentative evaluation of neutron-induced reactions on ^{181}Ta using the revised scheme. Especially, information needed for updating and predicting was directly retrieved from the output files of the nuclear model, hence besides $\vec{w}_i$ and W_i no additional data structures were introduced.

We performed 2000 TALYS [17] calculations with random varia-

tions of the neutron, proton and alpha optical model parameters. The parameters were sampled uniformly in the range between 80% and 120% relative to their default values. In total, 72 model parameters were adjusted—24 optical model parameters for each particle type. The same grid of 150 incident energies between 1 MeV and 200 MeV was used for all calculations. The calculations were performed on a computer cluster in parallel. TALYS was configured to provide detailed output, which led to the generation of about 4×10^4 text files in each calculation. The files generated in each calculation required 300 megabytes of storage space and contained predictions for about 3×10^6 quantities. Such a number of quantities clearly surpasses the domain of application of the standard formulas, especially in the case of stage-wise updates. We bundled the resulting files of each calculation to a compressed *tar* archive. Each archive took about 30 megabytes. The total storage requirement for all archives was 60 gigabytes. For updating and making predictions, the relevant files were extracted whenever needed.

The linear interpolation from the model grid to the experiment grid was directly performed during the process of reading the content of the result files. In other words, instead of calculating $\vec{v}_k = \mathbf{S}(\vec{x}_k - \vec{\sigma}_0)$ (see Eq. (4)), we calculated $\vec{v}_k = \mathbf{S}\vec{x}_k - \mathbf{S}\vec{\sigma}_0$; and instead of constructing the sensitivity matrix $\mathbf{S}$ and multiplying it with the vectors $\vec{x}_k$ and $\vec{\sigma}_0$, we directly interpolated from these vectors. Furthermore, the vectors $\vec{x}_k$ and $\vec{\sigma}_0$ containing the predictions of model calculation never have to be explicitly constructed because their values are already available in the result files of TALYS. To interpolate a model calculation to an experimental observable, it suffices to locate the two relevant values in the result files. Thus, besides $\vec{w}_i$ and $\mathbf{W}_i$, only the output files of the model calculations were stored.

It is important to mention that storing the text files produced by the model calculations as compressed *tar* archives is convenient but not efficient. Unpacking and parsing the files introduces a significant overhead to the update procedure. Thus, the time need will be larger than reported in the previous section, where timings were based on an efficient binary representation of the matrix $\mathbf{U}$. However, using directly the output files of TALYS offers great flexibility. In order to introduce a new observable into the update procedure, it suffices to implement a function to read that observable from the TALYS result files.

We performed an update with the experimental data stated in [33]. The 854 experimental data points are distributed over 14 reaction channels. It took about 10 min to evaluate Eqs. (6) and (7). This time span includes unpacking and reading 14 files from each of the 2000 archives to construct the matrix $\mathbf{V}$ defined in Eq. (4). The update formulas can be evaluated in seconds, hence the overall time need is mainly determined by unpacking and reading the files.

The obtained quantities $\vec{w}_i$ and $\mathbf{W}_i$ enable the computation of arbitrary observables according to Eq. (8). As an example, we computed the evaluated estimates and associated covariances for the $(n, 2n)^{180}\text{Ta}$ cross section and the neutron spectrum at 10 MeV incident energy. These cross sections are displayed in Fig. 2 and their correlations in Fig. 3. The computation of the matrix $\mathbf{V}^p$ including the extraction of 151 files from each *tar* archive took about 14 min. This example shows that predictions, uncertainties, and correlations can be calculated for any observable predicted by the model in reasonable time.

One might wonder whether the computation of all 3×10^6 evaluated estimates contained in about 4×10^4 files is feasible. To answer this question, we computed 45 300 evaluated estimates of emission spectra at different incident energies with the neutron, proton, deuteron, triton, helium-3 nucleus, and alpha nucleus as emitted particles. The computation involves interpolations of the values from 900 files from each calculation. Using the already unpacked files, the computation of the evaluated estimates required 30 min. Both in terms of number of files and in terms of number of observables, there is about a factor 50 to the computation of all evaluated quantities. Thus, the time needed to obtain the estimates of all 3×10^6 observables can be estimated to be

roughly one day. As shown in Section 3.3, using an efficient binary representation, for example in the form of the matrix $\mathbf{U}$, would allow the prediction of all 3×10^6 observables in about 2 min.

5. Discussion

The revised scheme presents a fundamentally different view on the GLS method than the standard formulas. Rather than updating the values of observables $\vec{\sigma}_0$, one updates a vector of weights $\vec{w}_0$. Each weight in this vector is associated with a specific calculation. Similarly, instead of computing all elements of a possible large posterior covariance matrix $\mathbf{A}_i$ for the observables, one only computes a matrix $\mathbf{W}_i$ which contains the covariances between calculations.

The revised scheme clearly demonstrates which information is considered in the update process. The updated quantities $\vec{w}_i$ and $\mathbf{W}_i$ are functions of the results of the model calculations interpolated to the grid of the measurements. Predicted observables not directly related to the measurements play no role in the update process. The information of measurements included in earlier updates affects the current update only through the induced modifications in $\vec{w}_0$ and $\mathbf{W}_0$. It must be emphasized that the latter two quantities are global information at the level of complete calculations. These two quantities in combination with $\vec{\sigma}_0$ and the model prediction vectors $\vec{x}_k$ completely specify the current state of knowledge.

One could go even one step further. The information in $\vec{\sigma}_0$ and the model prediction vectors $\vec{x}_k$ is fully determined by the predictions of a model $\mathcal{M}$ performed with some parameter sets $(\vec{p}_k)_{k=1\dots N}$. Thus, in principle, it is not necessary to keep the results of model calculations after an update. Required information from model calculations for updating and making predictions can be recomputed at any time. Whether the results of model calculations should be stored or rather recomputed depends on the time requirement of the computation, the size of the output, and the available storage space.

These features allow us to add missing observables at a later point, even if updates have already been performed. In contrast to that, in the standard scheme all relevant observables must be specified at the very beginning before the generation of the prior. Every eventual measurement and any prediction of interest in the future has to be anticipated. If at some point, a prediction of some observable not taken into account is needed, the whole chain from prior generation over updating with measurement data has to be redone.

The increased flexibility in the revised scheme in combination with its speed advantage make it suitable to be implemented as a database application. As has been shown in Section 3.2, the information for millions of observables can be updated in seconds. The predictions of about 10^6 observables can be calculated in about 30 s, see Section 3.3. The time need to predict a subset of them, e.g. 1000 observables, is in the order of 0.01 s. These timings were measured on a personal computer. Therefore, these operations could even be performed faster on a server with more powerful hardware, for instance a system where the results of model calculations are distributed over several solid-state drives.

6. Summary and outlook

We presented an alternative computation scheme for the GLS formula which avoids the explicit calculation of the prior covariance matrix and instead works directly with the set of model calculations. The outlined computation scheme is mathematically equivalent to the standard scheme. In typical evaluation scenarios, the feasible number of observables is increased by two or three orders of magnitude compared to the standard scheme. We applied the new scheme in a tentative evaluation of neutron-induced reactions of ^{181}Ta involving about 3×10^6 observables.

The revised scheme allows us to add missing observables at a later point, even after updates with experimental data. Updating and making

Fig. 2. Prior cross section (red) and evaluated cross section (blue). The left graph shows the (n,2n) cross section leaving the target nucleus in the ground state. The right graph shows the continuous part of the neutron emission spectrum for incoming neutrons with energy 10 MeV. Depicted data are taken from EXFOR [34]. Original sources are [35,36,37,38,39]. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

predictions are separate processes, and both of these processes can be performed quickly. These features make the revised scheme suited to be implemented as a database application with public access. Users could query evaluated predictions, uncertainties and correlations of any observable which can be predicted by the employed nuclear model. The flexibility of such a database would be a valuable complement to existing data services such as the JEFF or the TENDL library.

Finally, the ability to deal with a large number of observables is a key requirement for the evaluation of high-energy reactions. Evaluated uncertainties may provide essential information for the uncertainty quantification of novel technologies, such as the currently planned

MYRRHA reactor and future fusion and transmutation devices.

Acknowledgments

We thank Jean-Christoph David and Anthony Marchix for valuable feedback on the paper. This work was performed within the work package WP11 of the CHANDA project (605203) of the European Commission. It is partly based on results achieved within the Impulsprojekt IPN2013-7 financed by the Austrian Academy of Sciences and the Partnership Agreement F4E-FPA-168.01 with Fusion for Energy (F4E).

Appendix A. Derivation of the GLS formulas

We derive the GLS formulas by taking recourse to Bayesian statistics (e.g. [40]). An alternative derivation adopting a minimum-variance viewpoint is presented in [4].

Prior knowledge about the true cross section vector $\vec{\sigma}_{\text{true}}$ is specified in terms of a probability density function (pdf). In the GLS method, this

Fig. 3. Change of correlation structure due to updating with experimental data. Both graphs show the correlation between the (n,2n) cross section leaving the target nucleus in the ground state and the emitted neutron spectra for incoming neutrons with energy 10 MeV. The left graph shows the correlations before and the right graph after taking into account experimental data.

prior pdf is given as a multivariate normal distribution

$$\pi_0(\vec{\sigma}_{\text{true}}) \propto \exp\left\{-\frac{1}{2}(\vec{\sigma}_{\text{true}} - \vec{\sigma}_0)^T \mathbf{A}_0^{-1}(\vec{\sigma}_{\text{true}} - \vec{\sigma}_0)\right\}. \quad (\text{A.1})$$

The center vector $\vec{\sigma}_0$ and the covariance matrix $\mathbf{A}_0$ define its shape. Section 2 outlined their determination based on model calculations. We do not need to know normalization constants in the following discussion.

The dispersion of the measurement vector $\vec{\sigma}_{\text{exp}}$ around the true cross section vector is also described by a multivariate normal distribution,

$$\ell(\vec{\sigma}_{\text{exp}}|\vec{\sigma}_{\text{true}}) \propto \exp\left\{-\frac{1}{2}(\vec{\sigma}_{\text{exp}} - \mathbf{S}\vec{\sigma}_{\text{true}})^T \mathbf{B}^{-1}(\vec{\sigma}_{\text{exp}} - \mathbf{S}\vec{\sigma}_{\text{true}})\right\}. \quad (\text{A.2})$$

This pdf is called the likelihood. The covariance matrix $\mathbf{B}$ reflects statistical and systematic errors associated with the measurements. The sensitivity matrix $\mathbf{S}$ maps the values of $\vec{\sigma}_{\text{true}}$ to the grid of $\vec{\sigma}_{\text{exp}}$.

Both pdfs can be combined using the Bayesian update formula:

$$\pi_1(\vec{\sigma}_{\text{true}}|\vec{\sigma}_{\text{exp}}) \propto \ell(\vec{\sigma}_{\text{exp}}|\vec{\sigma}_{\text{true}})\pi_0(\vec{\sigma}_{\text{true}}). \quad (\text{A.3})$$

The posterior pdf π_1 represents an improved knowledge about $\vec{\sigma}_{\text{true}}$. The exponents of both prior pdf and likelihood are quadratic forms with respect to $\vec{\sigma}_{\text{true}}$. The exponent of the posterior is the sum of the exponents of the prior pdf and the likelihood and therefore also a quadratic form. Thus, the posterior pdf is a multivariate normal distribution,

$$\pi_1(\vec{\sigma}_{\text{true}}) \propto \exp\left\{-\frac{1}{2}(\vec{\sigma}_{\text{true}} - \vec{\sigma}_1)^T \mathbf{A}_1^{-1}(\vec{\sigma}_{\text{true}} - \vec{\sigma}_1)\right\}. \quad (\text{A.4})$$

The center vector $\vec{\sigma}_1$ and the covariance matrix $\mathbf{A}_1$ can be determined by comparing the exponents of the left and right hand side of Eq. (A.3):

$$\begin{aligned} \vec{\sigma}_{\text{true}}^T \mathbf{A}_1^{-1} \vec{\sigma}_{\text{true}} + \vec{\sigma}_{\text{true}}^T \mathbf{A}_1^{-1} \vec{\sigma}_1 + \vec{\sigma}_1^T \mathbf{A}_1^{-1} \vec{\sigma}_{\text{true}} + \vec{\sigma}_1^T \mathbf{A}_1^{-1} \vec{\sigma}_1 &= \vec{\sigma}_{\text{true}}^T \mathbf{A}_0^{-1} \vec{\sigma}_{\text{true}} + \vec{\sigma}_{\text{true}}^T \mathbf{A}_0^{-1} \vec{\sigma}_0 + \vec{\sigma}_0^T \mathbf{A}_0^{-1} \vec{\sigma}_{\text{true}} + \vec{\sigma}_0^T \mathbf{A}_0^{-1} \vec{\sigma}_0 + C \vec{\sigma}_{\text{true}}^T \mathbf{S}^T \mathbf{B}^{-1} \mathbf{S} \vec{\sigma}_{\text{true}} + \vec{\sigma}_{\text{true}}^T \mathbf{S}^T \mathbf{B}^{-1} \mathbf{S} \vec{\sigma}_{\text{exp}} \\ &+ \vec{\sigma}_{\text{exp}}^T \mathbf{B}^{-1} \mathbf{S} \vec{\sigma}_{\text{true}} + \vec{\sigma}_{\text{exp}}^T \mathbf{B}^{-1} \vec{\sigma}_{\text{exp}}. \end{aligned} \quad (\text{A.5})$$

Using normalized pdfs for the prior and the likelihood, the Bayesian update formula yields a normalized posterior pdf. Terms independent of $\vec{\sigma}_{\text{true}}$ in the exponents of the pdfs only affect the normalization constant. Therefore, the sides of Eq. (A.5) are allowed to differ by a constant C and we only have to consider terms with $\vec{\sigma}_{\text{true}}$. Matching the terms with $\vec{\sigma}_{\text{true}}$ of the left and right hand side of Eq. (A.5) leads to

$$\mathbf{A}_1^{-1} = \mathbf{A}_0^{-1} + \mathbf{S}^T \mathbf{B}^{-1} \mathbf{S} \quad (\text{A.6})$$

$$\mathbf{A}_1^{-1} \vec{\sigma}_1 = \mathbf{A}_0^{-1} \vec{\sigma}_0 + \mathbf{S}^T \mathbf{B}^{-1} \vec{\sigma}_{\text{exp}} \quad (\text{A.7})$$

Applying the Woodbury matrix identity for inversion [41], the posterior covariance matrix can be expressed as

$$\mathbf{A}_1 = \mathbf{A}_0 - \mathbf{A}_0 \mathbf{S}^T (\mathbf{S} \mathbf{A}_0 \mathbf{S}^T + \mathbf{B})^{-1} \mathbf{S} \mathbf{A}_0. \quad (\text{A.8})$$

Multiplying Eq. (A.7) from the left with this representation of $\mathbf{A}_1$ yields

$$\vec{\sigma}_1 = \vec{\sigma}_0 - \mathbf{A}_0 \mathbf{S}^T (\mathbf{S} \mathbf{A}_0 \mathbf{S}^T + \mathbf{B})^{-1} \mathbf{S} \vec{\sigma}_0 + \mathbf{A}_0 \mathbf{S}^T (\mathbf{B}^{-1} - (\mathbf{S} \mathbf{A}_0 \mathbf{S}^T + \mathbf{B})^{-1} \mathbf{S} \mathbf{A}_0 \mathbf{S}^T \mathbf{B}^{-1}) \vec{\sigma}_{\text{exp}} \quad (\text{A.9})$$

We rearrange the bracketed expression in the second line:

$$\mathbf{B}^{-1} - (\mathbf{S} \mathbf{A}_0 \mathbf{S}^T + \mathbf{B})^{-1} \mathbf{S} \mathbf{A}_0 \mathbf{S}^T \mathbf{B}^{-1} = (\mathbf{S} \mathbf{A}_0 \mathbf{S}^T + \mathbf{B})^{-1} ((\mathbf{S} \mathbf{A}_0 \mathbf{S}^T + \mathbf{B}) \mathbf{B}^{-1} - \mathbf{S} \mathbf{A}_0 \mathbf{S}^T \mathbf{B}^{-1}) = (\mathbf{S} \mathbf{A}_0 \mathbf{S}^T + \mathbf{B})^{-1}. \quad (\text{A.10})$$

Using this identity, we can write Eq. (A.9) as

$$\vec{\sigma}_1 = \vec{\sigma}_0 + \mathbf{A}_0 \mathbf{S}^T (\mathbf{S} \mathbf{A}_0 \mathbf{S}^T + \mathbf{B})^{-1} (\vec{\sigma}_{\text{exp}} - \mathbf{S} \vec{\sigma}_0). \quad (\text{A.11})$$

Eqs. (A.8) and (A.11) are the final forms of the GLS formulas as stated in Eqs. (2) and (3).

References

- [1] NEA, Uncertainty and Target Accuracy Assessment for Innovative Systems Using Recent Covariance Data Evaluations, NEA/OECD, Paris, 2008.
- [2] A.J. Koning, D. Rochman, Towards sustainable nuclear energy: putting nuclear physics to work, *Ann. Nucl. Energy* 35 (11) (2008) 2024–2030. <http://dx.doi.org/10.1016/j.anucene.2008.06.004>.
- [3] D. Rochman, A.J. Koning, S.C. van der Marck, A. Hogenbirk, D. van Veen, Nuclear data uncertainty propagation: total Monte Carlo vs. covariances, *J. Korean Phys. Soc.* 59 (23) (2011) 1236. <http://dx.doi.org/10.3938/jkps.59.1236>.
- [4] D.W. Muir, Evaluation of correlated data using partitioned least squares: a minimum-variance derivation, *Nucl. Sci. Eng.* 101 (1) (1989) 88–93.
- [5] T. Kawano, K. Shibata, Covariance Evaluation System, Technical Report JAERI-Data/Code 97-037, JAERI, Japan, 1997.
- [6] D. Smith, L.A. Naberejnev, *Large Errors and Severe Conditions* (2002).
- [7] M. Herman, P. Obložinský, D. Rochman, T. Kawano, L. Leal, Fast Neutron Covariances for Evaluated Data Files, Technical Report BNL-75893-2006-CP, Brookhaven National Laboratory, May 2006.
- [8] E. Bauge, S. Hilaire, P. Dossantos-Uzarralde, Evaluation of the Covariance Matrix of Neutronic Cross Sections with the Backward-Forward Monte Carlo Method, EDP Sciences, Les Ulis, France; London, UK, 2007. <http://dx.doi.org/10.1051/ndata:07339>.
- [9] H. Leeb, D. Neudecker, T. Srdinko, Consistent procedure for nuclear data evaluation based on modeling, *Nucl. Data Sheets* 109 (12) (2008) 2762–2767. <http://dx.doi.org/10.1016/j.nds.2008.11.006>.
- [10] D.L. Smith, A unified Monte Carlo approach to fast neutron cross section data evaluation, in: *Proceedings of the 8th International Topical Meeting on Nuclear Applications and Utilities Of Accelerators*, Pocatello, July 2008, pp. 736–743.
- [11] D. Neudecker, The full Bayesian evaluation technique and its application to isotopes of structural materials (Ph.D. thesis), Technische Universität Wien, Austria, 2012.
- [12] R. Capote, D.L. Smith, A. Trkov, M. Meghzifene, A new formulation of the unified Monte Carlo approach (UMC-B) and cross-section evaluation for the dosimetry reaction $^{55}\text{Mn}(n,g)^{56}\text{Mn}$, *J. ASTM Int.* 9 (3) (2012) 104115. <http://dx.doi.org/10.1520/JAI104115>.
- [13] A.J. Koning, Bayesian Monte Carlo method for nuclear data evaluation, *Nucl. Data Sheets* 123 (2015) 207–213. <http://dx.doi.org/10.1016/j.nds.2014.12.036>.
- [14] R. Capote, D. Smith, A. Trkov, Nuclear data evaluation methodology including estimates of covariances, *EPJ Web Conf.* 8 (2010) 04001. <http://dx.doi.org/10.1051/epjconf/20100804001>.
- [15] G. Schnabel, Large scale Bayesian nuclear data evaluation with consistent model defects (Ph.D. thesis), Technische Universität Wien, Vienna, 2015. URL (<http://katalog.ub.tuwien.ac.at/AC12392895>).
- [16] A.J. Koning, S. Hilaire, M.C. Duijvestijn, TALYS-1.0, EDP Sciences, Les Ulis, France; London, UK, 2007. <http://dx.doi.org/10.1051/ndata:07767>.
- [17] A.J. Koning, S. Hilaire, S. Goriely, TALYS-1.6 – A Nuclear Reaction Program (December) (2013) (<http://www.talys.eu>).
- [18] D.W. Muir, A. Trkov, I. Kodeli, R. Capote, V. Zerkin, The Global Assessment of Nuclear Data, GANDR, EDP Sciences, Les Ulis, France; London, UK, 2007. <http://dx.doi.org/10.1051/ndata:07635>.
- [19] International Atomic Energy Agency. URL (<https://www.iaea.org/>).

- [20] G. Schnabel, H. Leeb, A. New, Module for large scale Bayesian evaluation in the fast neutron energy region, *Nucl. Data Sheets* 123 (2015) 196–200. <http://dx.doi.org/10.1016/j.nds.2014.12.034>.
- [21] Nuclear Energy Agency – JEFF and EFF Projects. URL (https://www.oecd-neo.org/dbforms/data/eva/evatapex/jeff_32/).
- [22] A. Koning, D. Rochman, J. Kopecky, J. Sublet, M. Fleming, E. Bauge, S. Hilaire, P. Romain, B. Morillon, H. Duarte, S. van der Marck, S. Pomp, H. Sjostrand, R. Forrest, H. Henriksson, O. Cabellos, S. Goriely, J. Leppanen, H. Leeb, A. Plompen, R. Mills, TENDL-2015 Nuclear Data Library. URL (https://tendl.web.psi.ch/tendl_2015/tendl2015.html).
- [23] G. Evensen, Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, *J. Geophys. Res.* 99 (C5) (1994) 10143. <http://dx.doi.org/10.1029/94JC00572>.
- [24] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*, 2nd ed., Springer, Dordrecht, 2009.
- [25] H. Matthies, G. Strang, The solution of nonlinear finite element equations, *Int. J. Numer. Methods Eng.* 14 (11) (1979) 1613–1626. <http://dx.doi.org/10.1002/nme.1620141104>.
- [26] J. Nocedal, Updating quasi-Newton matrices with limited storage, *Math. Comput.* 35 (151) (1980) 773. <http://dx.doi.org/10.2307/2006193>.
- [27] A.J. Koning, J.P. Delaroche, Local and global nucleon optical models from 1 keV to 200 MeV, *Nucl. Phys. A* 713 (3–4) (2003) 231–310. [http://dx.doi.org/10.1016/S0375-9474\(02\)01321-0](http://dx.doi.org/10.1016/S0375-9474(02)01321-0).
- [28] A.J. Koning, Bayesian Monte Carlo method for nuclear data evaluation, *Eur. Phys. J. A* 51(12) (2015). <http://dx.doi.org/10.1140/epja/i2015-15184-x>.
- [29] D. Smith, N. Otuka, Experimental nuclear reaction data uncertainties: basic concepts and documentation, *Nucl. Data Sheets* 113 (12) (2012) 3006–3053. <http://dx.doi.org/10.1016/j.nds.2012.11.004>.
- [30] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008 (ISBN 3-900051-07-0). URL (<http://www.R-project.org>).
- [31] OpenBLAS: An Optimized BLAS Library (<http://www.openblas.net/>).
- [32] M.J. Kane, J. Emerson, S. Weston, Scalable strategies for computing with massive data, *J. Stat. Softw.* 55 (14) (2013) 1–19 (URL (<http://www.jstatsoft.org/v55/i14/>)).
- [33] H. Leeb, G. Schnabel, T. Srdinko, V. Wildpaner, Bayesian evaluation including covariance matrices of neutron-induced reaction cross sections of ^{181}Ta , *Nucl. Data Sheets* 123 (2015) 153–158. <http://dx.doi.org/10.1016/j.nds.2014.12.027>.
- [34] N. Otuka, E. Dupont, V. Semkova, B. Pritychenko, A. Blokhin, M. Aikawa, S. Babykina, M. Bossant, G. Chen, S. Dunaeva, R. Forrest, T. Fukahori, N. Furutachi, S. Ganesan, Z. Ge, O. Gritzay, M. Herman, S. Hlavač, K. Katō, B. Lalremruata, Y. Lee, A. Makinaga, K. Matsumoto, M. Mikhaylyukova, G. Pikulina, V. Pronyaev, A. Saxena, O. Schwerer, S. Simakov, N. Soppera, R. Suzuki, S. Takács, X. Tao, S. Taova, F. Tárkányi, V. Varlamov, J. Wang, S. Yang, V. Zerkov, Y. Zhuang, Towards a more complete and accurate experimental nuclear reaction data library (EXFOR): international collaboration between Nuclear Reaction Data Centres (NRDC), *Nucl. Data Sheets* 120 (2014) 272–276. <http://dx.doi.org/10.1016/j.nds.2014.07.065>.
- [35] F. Peiguo, Chin. J. Nucl. Phys. 7 (3) (1985) 242.
- [36] L. Han-Lin, Z. Wen Rong, F.P. Guo, Measurement of the neutron cross sections for the reactions $^{169}\text{Yb}(n, 2n)^{168}\text{Yb}$, $^{169}\text{Yb}(n, 3n)^{167}\text{Yb}$, and $^{181}\text{Ta}(n, 2n)^{180m}\text{Ta}$, *Nucl. Sci. Eng.* 90 (3) (1985) 304–310. <http://dx.doi.org/10.13182/NSE85-3>.
- [37] A.A. Filatenkov, Leningrad Reports, Technical Report 258, Khlopin Radium Institute, 2001.
- [38] C. Zhu, Y. Chen, Y. Mou, P. Zheng, T. He, X. Wang, L. An, H. Guo, Measurements of $(n, 2n)$ reaction cross sections at 14 MeV for several nuclei, *Nucl. Sci. Eng.* 169 (2) (2011) 188–197. <http://dx.doi.org/10.13182/NSE10-35>.
- [39] C. Bhatia, M.E. Gooden, W. Tornow, A.P. Tonchev, Ground-state and isomeric-state cross sections for $^{181}\text{Ta}(n, 2n)^{180}\text{Ta}$ between 8 and 15 MeV, *Phys. Rev. C* 87(3) (2013). <http://dx.doi.org/10.1103/PhysRevC.87.031601>.
- [40] E.T. Jaynes, *Probability Theory: The Logic of Science*, Cambridge University Press, Cambridge, UK; New York, NY, 2003.
- [41] M.A. Woodbury, Inverting Modified Matrices, Statistical Research Group, Memo. Rep. no. 42, Princeton University, Princeton, N.J., 1950.