BNL-99143-2013-IR

Upgrade the EMPIRE code as needed for evaluation and production of covariances for actinides

Gustavo Nobre, Annalia Palumbo, Sam Hoblit, Mike Herman

National Nuclear Data Center,
Brookhaven National Laboratory, Upton NY 11973

January 3, 2013

National Nuclear Data Center
Brookhaven National Laboratory
P.O. Box 5000
Upton, NY 11973-5000
www.nndc.bnl.gov

U.S. Department of Energy
Office of Science, Office of Nuclear Physics

Notice: This manuscript has been authored by employees of Brookhaven Science Associates, LLC under Contract No. DE-AC02-98CH10886 with the U.S. Department of Energy. The publisher by accepting the manuscript for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof..

Contents

Abstract

Treatment of actinides in the EMPIRE code has been updated by adding capabilities necessary for effective assimilation of nuclei that undergo fission. The upgrades comprise calculations of Prompt Fission Neutron Spectra (PFNS), and the average cosine of elastically scattered neutrons ($\bar{\mu}$) and include sensitivity and covariance computations for both quantities. In addition, EMPIRE outputs the average number of neutrons released per fission event ($\bar{\nu}$) that is read from the existing evaluated file and scaled by a factor. Kalman filter is used to fit calculated PFNS, $\bar{\mu}$, and $\bar{\nu}$ to the experimental data and to produce covariances for the quantities and respective model parameters. These updates expand EMPIRE assimilation possibilities by adding PFNS, $\bar{\nu}$, and $\bar{\mu}$ to the physical quantities that might be perturbed along with the cross sections during the adjustment. We describe PFNS models used in the EMPIRE code, the modifications required to accommodate these models, and detail updates to the EMPIRE scripts used for fitting, and illustrate the new features applying them to specific nuclei. Finally, we report on actual assimilation of ^{239}Pu performed by calculating sensitivity of the JEZEBEL k_{eff} to the full set of reaction model parameters (EMPIRE+MCNP) and using Kalman filter to perform adjustment of the parameters. This work was performed according to the goals set forth in milestones M3FT-12BN0210011 and M3FT-12BN0210012.

Introduction

The task described in this report continues previous years effort to establish viable and statistically solid methodology for generating nuclear data covariances combining nuclear reaction modeling and experimental data. The previous work focused on the production of an extensive set of sensitivities to the reaction model parameters defining fission cross sections. These sensitivities are to be used in the KALMAN code to combine theory with measurements in order to estimate fission cross section covariances. In the current phase, the EMPIRE code has been extended to allow for calculation of the prompt neutron fission spectra (PFNS) along with their covariances that are expected to play significant role in the total uncertainty budget. In addition, we also include covariances for fission neutron multiplicities ($\bar{\nu}$) and elastic angular distributions ($\bar{\mu}$) to cover all most important degrees of freedom in modeling the fission channel. These steps will allow for more reliable, better physically founded covariances that will support nuclear data adjustment and benefit the new assimilation technique that is being developed by the BNL/INL collaboration.

In the present report we describe Prompt Fission Neutron Spectra (PFNS) models implemented in the current extension of the EMPIRE code constituting milestone M3FT-12BN0210012. In the second phase, the EMPIRE code is extended to allow for calculation of covariances for PFNS that are expected to play significant role in the total uncertainty budget but have not yet been explored. Both steps will allow for more reliable, better physically founded covariances that will support nuclear data adjustment and benefit the new assimilation technique that is being developed by the BNL/INL collaboration.

Let us start with a short description of the EMPIRE code, which has been extended in the present work by implementing PFNS models. EMPIRE is a modular system of nuclear reaction codes, comprising various nuclear models and designed for calculations over a broad range of energies and incident particles. The system can be used for theoretical investigations of nuclear reactions as well as for nuclear data evaluation work. Photons, nucleons, deuterons, tritons, helions (^{3}He), α's, and light or heavy ions can be selected as projectiles. The energy range starts just above the resonance region in the case of a neutron projectile, and extends up to few hundred MeV for heavy ion induced reactions. The code accounts for the major nuclear reaction models, such as the optical model, Coupled Channels and DWBA (ECIS06 and OPTMAN), Multi-step Direct (ORION + TRISTAN), NVWY Multi-step Compound, exciton model (PCROSS), hybrid Monte Carlo simulation (DDHMS), and the full featured Hauser-Feshbach model including width fluctuations and the optical model for fission. Heavy-ion fusion cross sections can be calculated within the simplified coupled channels approach (CCFUS). A comprehensive library of input parameters based on the RIPL-3 library covers nuclear masses, optical model parameters, ground state deformations, discrete levels and decay schemes, level densities, fission barriers, and γ-ray strength functions. Effects of the dynamic deformation of a fast rotating nucleus can be taken into account in the calculations (BARFIT, MOMFIT).

The results can be converted into the ENDF-6 format using the accompanying EMPEND code. Modules of the ENDF Utility Codes and the ENDF Pre-Processingt codes are applied for ENDF file verification. The package contains the full EXFOR library of experimental data in computational format C4 that are automatically retrieved during the calculations.

EMPIRE contains the resonance module that retrieves data from the electronic version of the Atlas of Neutron Resonances by Mughabghab (not provided with the EMPIRE distribution) to produce resonance section and related covariances for the ENDF-6 formatted files. EMPIRE can be used to determine covariances of the calculated data using either sensitivity matrices along with the KALMAN code or employing Monte Carlo approach to produce model generated covariances. In both cases experimental data can be taken into account, either directly (KALMAN) or by feeding the EMPIRE calculated Monte Carlo modeling covariance as a prior to the least square fitting GANDR system.

Publication quality graphs can be obtained using the powerful and flexible plotting package ZVView. Interactive plots with ZVView comparing experimental results with calculations can be produced with ENDVER modules.

The backbone of the EMPIRE system are bash-shell UNIX scripts that provide for seamless console operation of EMPIRE on Linux, Mac OS X, and Microsoft Windows with GNU gfortran compiler installed. Additionally, the graphical interface, provides for an easy operation of the system on Linux, Mac OS X and virtual Linux machines running on Microsoft Windows.

Section 1

Upgrade of EMPIRE code with PFNS capabilities

The prompt fission neutrons play an important role in neutronics calculations for nuclear applications. It has been shown that estimated uncertainties in the prompt fission neutron spectrum can significantly impact the results of transport simulations for critical assemblies as well as reactor sensitivity calculations [1]. The main difficulty with the modeling of prompt fission neutron spectra (PFNS) is that their shape is not well defined experimentally. In addition, it is also important to know the PFNS as a function of both the fissioning nucleus and its excitation energy. Due to this sensitivity, PFNS has to be included in the adjustment to avoid unphysical modification of other quantities, e.g., fission cross sections, that might result from compensating deficiencies in the PFNS. In the current version of the EMPIRE code we have implemented PFNS calculations using Los Alamos and Kornilov models. We have decided to include both formulations since there is a controversy regarding the lower energy part of the spectra - the Los Alamos model tends to predict lower values versus the Kornilov model. So far integral testing was mixed and there is no clear evidence which of the two models is more adequate.

1.1 Los Alamos model for prompt fission neutron spectra

According to the Los Alamos model [2], both the Watt and Maxwellian spectra neglect two important physical effects: 1) the distribution of fission-fragment residual nuclear temperature that results from the initial distribution of fission-fragment excitation energy and the subsequent cooling of the fragments as neutrons are emitted and 2) the energy dependence of the cross section for the inverse process of compound nucleus formation. Furthermore, the Maxwellian spectrum also neglects the center-of-mass motion of the fission fragments from which the neutrons are emitted; therefore, the agreement between spectra and data is achieved by adjusting parameters to values that are somewhat unphysical. The Los Alamos model addresses these inconsistencies by taking the distribution of fission-fragment residual nuclear temperature to be triangular in shape, extending linearly from zero to a maximum value T$_m$, and calculating the energy-dependent compound nucleus cross section for representative average fission fragments by use of an optical model. This permits $N(E)$ to be calculated easily for any fissioning nucleus at any excitation energy. This model is the basis for evaluation of prompt fission neutrons spectra in most currently evaluated nuclear data libraries. This relatively simple and compact formalism, with only a handful of adjustable parameters, has been very successful in predicting the prompt fission neutrons spectra for neutron-induced as well as spontaneous fission reactions for a wide range of actinides and incident neutron energies. More refined and/or elaborate versions of the Los Alamos model have been developed in recent years for evaluation purposes, but they all rely on the implementation of this model in the first place. The Los Alamos (LA) model is described at length in Ref. [2]. We will only summarize its main features here.

The prompt fission neutron spectrum $N(E)$ is calculated as a function of the fissioning nucleus and its excitation energy. Weisskopf statistical evaporation theory [3] is used to explain the emission of neutrons from an excited compound nucleus (CN) at a given temperature T, and a triangular distribution of initial fission fragments residual temperatures is assumed.

1.1.1 Center-of-mass neutron energy spectrum

The neutron energy spectrum in the center-of-mass of a fission fragment is given by

$$\phi(\epsilon) = \frac{2\sigma_c(\epsilon)\epsilon}{T_m^2} \int_0^{T_m} k(T) e^{-\epsilon/T} T \, dT \tag{1.1}$$

with the temperature-dependent normalization constant $k(T)$

$$k(T)^{-1} = \int_0^{\infty} \sigma_c(\epsilon) e^{-\epsilon/T} \epsilon \, d\epsilon. \tag{1.2}$$

$\sigma_c(\epsilon)$ is the energy-dependent cross section for the inverse process of CN formation. Equation (1.1) was obtained by integrating over a triangular distribution of temperatures with a maximum temperature T_m.

1.1.2 Laboratory neutron energy spectrum

In the laboratory system, the neutron energy spectrum $N(E)$ for a fission fragment moving with a kinetic energy per nucleon E_f is

$$N(E) = \frac{1}{4\sqrt{E_f}} \int_{\left(\sqrt{E}-\sqrt{E_f}\right)^2}^{\left(\sqrt{E}+\sqrt{E_f}\right)^2} \frac{\phi(\epsilon)}{\sqrt{\epsilon}} \, d\epsilon, \tag{1.3}$$

where E is the laboratory neutron energy.

Inserting (1.1) into (1.3), the laboratory neutron energy spectrum $N(E)$ becomes

$$N(E) = \frac{1}{2\sqrt{E_f}T_m^2} \int_{\left(\sqrt{E}-\sqrt{E_f}\right)^2}^{\left(\sqrt{E}+\sqrt{E_f}\right)^2} \sigma_c(\epsilon)\sqrt{\epsilon}\, d\epsilon \times \int_0^{T_m} k(T) e^{-\epsilon/T} T\, dT, \tag{1.4}$$

with $k(T)$ given by Eq. (1.2).

Considering the most probable fragmentation only, the average center-of-mass neutron energy spectrum $\phi(\epsilon)$ is therefore given by an average over the spectra for the light and heavy fragments as

$$\phi(\epsilon) = \frac{1}{2}\left(\phi_L(\epsilon) + \phi_H(\epsilon)\right), \tag{1.5}$$

and similarly for the laboratory neutron energy spectrum $N(E)$

$$N(E) = \frac{1}{2}\left(N_L(E) + N_H(E)\right). \tag{1.6}$$

In the last two equations, it is implicitly assumed that half of the emitted neutrons come from the light fragment and the other half from the heavy fragment.

1.1.3 Average prompt neutrons multiplicity

The average prompt fission neutrons multiplicity $\overline{\nu}_p$ is simply obtained from energy conservation:

$$\overline{\nu}_p = \frac{\langle E^* \rangle - \left\langle E_\gamma^{tot} \right\rangle}{\langle S_n \rangle + \langle \epsilon \rangle}, \tag{1.7}$$

where $\langle E^* \rangle$ is the average total excitation energy and is equal to

$$\langle E^* \rangle = \langle E_r \rangle + B_n + E_n - \left\langle E_f^{tot} \right\rangle. \tag{1.8}$$

$\langle E_r \rangle$ is the average fission energy release, B_n is the neutron binding energy of the target nucleus, E_n is the neutron incident energy, $\left\langle E_f^{tot} \right\rangle$ is the average total fission-fragment kinetic energy, $\left\langle E_\gamma^{tot} \right\rangle$ is the average total energy carried away through gamma-ray emission, $\langle S_n \rangle$ is the average neutron separation energy of the fission fragments, and $\langle \epsilon \rangle$ is the average energy of the outgoing neutron in the center-of-mass reference frame.

1.1.4 Multiple-chance fission

At higher incident neutron energies, above the neutron binding energy, the excited compound system may emit one or several neutrons before undergoing fission. Although the neutrons emitted prior to fission are not directly related to the ones emitted from the fission fragments, in the LA model they are treated in a similar way. In the case of multiple-chance fission, the prompt fission neutron spectrum in the laboratory system can be written as

$$N(E) = \frac{\sum_i P_f^i \left(\sum_{j=1}^{i-1} \phi_j(E) + \overline{\nu}_i N_i(E) \right)}{\sum_i P_f^i (i - 1 + \overline{\nu}_i)} \tag{1.9}$$

where $\phi_j(E)$ is the pre-fission evaporation spectrum for the $(j+1)^{th}$-chance fission channel, and P_f^i is the i^{th}-chance fission probability.

The average prompt neutron multiplicity can be obtained similarly

$$\overline{\nu} = \frac{\sum_i P_f^i (i - 1 + \overline{\nu}_i)}{\sum_i P_f^i} \tag{1.10}$$

The i^{th} prompt neutron multiplicity $\overline{\nu}_i$ is given by

$$\overline{\nu}_i = \frac{\langle E_i^* \rangle - \langle E_{\gamma_i}^{tot} \rangle}{\langle S_n^i \rangle + \langle \epsilon_i \rangle}, \tag{1.11}$$

where the average total excitation energy for the i^{th}-chance fission is

$$\begin{aligned} \langle E_i^* \rangle \;\; &= \;\; \langle E_r^i \rangle + B_n(A) + E_n - \langle E_f^i \rangle \\ & \qquad - \sum_{j=1}^{i-1} \left(B_n(A - j + 1) + \langle \epsilon_j \rangle \right) \end{aligned} \tag{1.12}$$

Here, $\langle \epsilon_j \rangle$ it the mean kinetic energy of the neutron evaporated from the $(Z, A - j + 1)$ nucleus.

1.2 Kornilov model for prompt fission neutron spectra

The Kornilov model [4] is actually a systematics that allows to calculate PFNS with an accuracy comparable, and sometimes even better, that the more physical Los Alamos model described above. The essential argument for including the Kornilov model in the EMPIRE code in addition to the Los Alamos model is the fact the both models differ in shape of the PFNS. The Kornilov model tends to provide results that are higher at low emission energies than the Los Alamos model. This difference has been shown to influence integral testing although not always in the right direction. Therefore, it has been considered important to use both models in the assimilation procedure, which eventually might help to decide which model performs better.

The Kornilov model makes use of two Watt spectra to describe emission of neutrons with energy E from a fragment i (low mass $(i = l)$ and high mass $(i = h)$) created in the fission event induced by a neutron with energy E_o

$$W_i(E, E_{\nu i}, T_i(E_o)) = M(E, T_i(E_o)) \exp\left(-\frac{E_{\nu i}}{T_i(E_o)}\right) \frac{\sinh\left(\sqrt{bE}\right)}{\sqrt{bE}} \tag{1.13}$$

$$b = \frac{4E_{\nu i}}{T_i^2(E_o)},$$

where Maxwellian $M(E, T_i(E_o))$ is given by

$$M(E, T_i(E_o)) = \frac{2\sqrt{E}}{T_i(E_o)\sqrt{\pi T_i(E_o)}} \exp\left(-\frac{E}{T_i(E_o)}\right). \tag{1.14}$$

This formula is obtained by assuming that neutron spectra are described by a Maxwellian $M(E, T_i(E_o))$ in the center of mass system (CM) and transforming this spectra to the laboratory frame of reference (Lab). The temperature $T_i(E_o)$ of each fission fragment is made dependent on the incident neutron energy E_o. As in the Los Alamos model, it is assumed that neutron energy spectrum is an average over the spectra emitted from the both fragments.

$$F(E) = 0.5(W_l(E, E_{\nu l}, T_l(E_o)) + W_h(E, E_{\nu h}, T_h(E_o))) \tag{1.15}$$

The $E_{\nu i}$ are calculated as

$$E_{\nu l} = \frac{A_h}{A_l A} \alpha T_{\mathrm{KE}} \tag{1.16}$$

$$E_{\nu h} = \frac{A_l}{A_h A} \alpha T_{\mathrm{KE}},$$

where A_l, A_h, and A are masses of the two fission fragments and of the fissioning nucleus, α is the adjustable parameter reducing available total kinetic energy T_{KE}, which in turn is calculated following Malinovskii *et. al.* [5], where a heavy fragment with mass A_h has atomic number Z_h given by

$$Z_h = \frac{Z}{A} A_h - 0.5 \tag{1.17}$$

and accordingly for the low mass fragment we have

$$Z_l = Z - Z_h \tag{1.18}$$

$$A_l = A - A_h. \tag{1.19}$$

The default value of the α parameter in Eq. 1.16 is set according to Tab. 1.1 (1.0 is used for the nuclei not contained in Tab. 1.1). In addition, α can be controlled through the input in order to scale the total kinetic energy T_{KE} entering the calculations. Values for T_l and T_h are calculated by

$$T_{l,h}^x = T_{l,h}^{\mathrm{Cf}} \sqrt{\frac{U^x A^{\mathrm{Cf}}}{U^{\mathrm{Cf}} A^x}} \tag{1.20}$$

with $T_l^{\mathrm{Cf}} = 0.902$ MeV and $T_h^{\mathrm{Cf}} = 0.7675$ MeV, while $U = E_r + B_n + E_0 + T_{\mathrm{KE}} = U_0 + E_0$, where E_r is the energy release, B_n is the neutron binding energy, E_0 the projectile energy, and T_{KE} the total kinetic energy of the fragments. We used $U_0^{\mathrm{Cf}} = 32.9$ MeV.

Table 1.1: Value of the parameter α from Eq. 1.16 used to adjust the total kinetic energy for various materials.

Material	α
^{232}Th	0.947
^{233}U	0.920
^{235}U	0.936
^{238}U	0.880
^{237}Np	0.873
^{252}Cf	0.809

1.3 Implementation of the PFNS in EMPIRE code

Both Los Alamos and Kornilov models were coded as EMPIRE subroutines and linked to the code resources so that all necessary input data are made automatically available to the subroutines. The PFNS specific input data are read in through the standard EMPIRE input subroutine and can be adjusted individually. The standard EMPIRE approach allowing to adjust model parameters in function of incident energy is also used for the PFNS parameters that can thus be made energy dependent. Four parameters are defined in Empire for fitting PFNS as follows.

- PFNTKE: This parameter is used to scale the total kinetic energy of the fission fragments, T_{KE}.
- PFNALP: This parameter is used to scale the value of α as defined in Eq. 1.16, which scales the energy of both the light and heavy fragments $E_{\nu l}$ and $E_{\nu h}$.
- PFNRAT: This parameter is used to adjust the ratio of the kinetic energy of the light to heavy fragments, $E_{\nu l}/E_{\nu h}$.
- PFNERE: Used to scale the total fission energy release E_r as defined in section 1.2.

All four of these parameters take a default value of 1.0; if unspecified the default values in the Empire code are used.

The current implementation does not include neutrons emitted from the compound nucleus before the fission thus it is formally restricted to the first chance fission. This restriction makes the current implementation to be applicable at incident energies up to 3-4 MeV, the typical threshold for the (n,nf) reaction, which is generally sufficient for most reactor applications since fission spectrum practically disappears above 2 MeV.

EMPIRE has been extended to automatically retrieve experimental PFNS data from the EXFOR library and translate them into a form useful for plotting. Comparison plots showing results of calculations along with the experimental data can be generated from the EMPIRE GUI. An option is provided to plot calculated and experimental data normalized to the Maxwellian distribution at an arbitrary temperature. When choosing this option each experimental data set is divided by the Maxwellian value and normalized so that the integral of the spectrum is unity. The same procedure is applied also to the calculated results. Therefore, only shapes of the PFNS are compared while the absolute value is defined by the product of a fission cross section and multiplicity of fission neutrons at a given incident energy. An example plot obtained using this system is shown in Fig. 1.1. The same figure illustrates also sensitivity of the PFNS calculations to the perturbation of the total kinetic energy of the fission fragments T_{KE} by 5%.

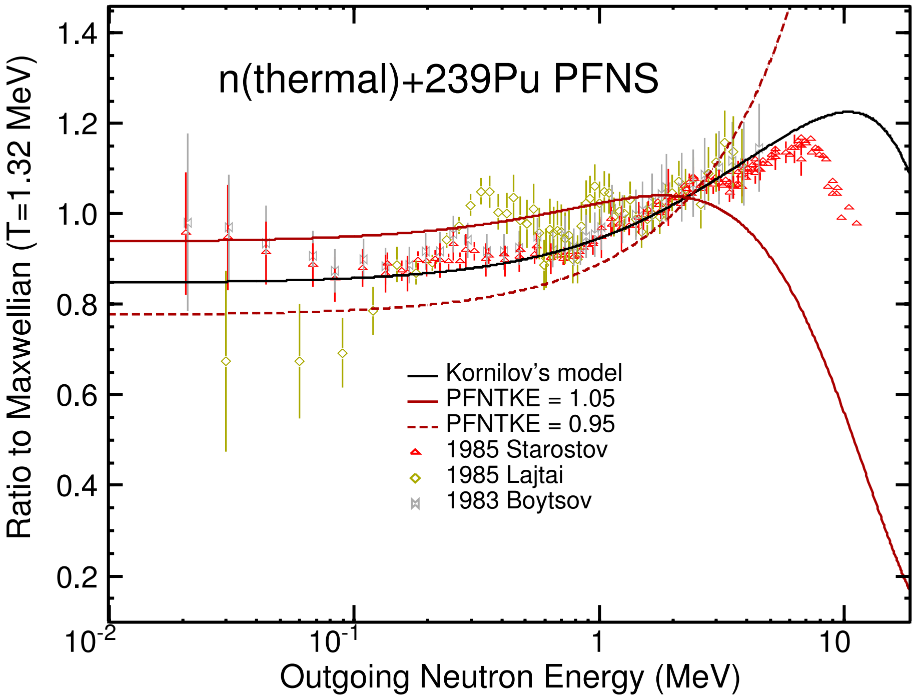


Figure 1.1: Calculated prompt fission neutron spectrum for thermal neutron induced fission on ^{239}Pu compared with the experimental data. The effect of changing total kinetic energy of the fragments by 5% in both direction is shown in red. The results are normalized to the Maxwellian distribution with $T = 1.32$ MeV.

Section 2

Using KALMAN fitting code

The multi-parameter, linearly fitting code KALMAN has proven to be very useful in improving evaluation procedure. Starting from an initial input, KALMAN is able to determine an "optimal" set of parameters that minimizes the overall χ^2. In order to do this, KALMAN varies the values of a given set of parameters and analyzes the effects produced in the cross sections.

Below, when describing filenames and instructions for the command line, the string '*' was adopted to correspond to the main project name, that is, the root of EMPIRE's input filename (the filename without the extension '`.inp`')

The Kalman utility within the Empire package was initially developed with the purpose of fitting only cross sections. However, the KALMAN code itself is completely oblivious to what quantity it is fitting since it simply receives as inputs only the experimental data sets for which the χ^2 will be minimized, the original functions before the fit, the parameters that modify each function and their sensitivities. It is thus the role of the auxiliary codes and of the *kalman* script to manipulate the information given by EMPIRE and write them in the appropriate format for the KALMAN code. Having this in mind, the Kalman utility was generalized in order to be able to also fit prompt fission neutron spectra (PFNS). Four PFNS-specific parameters were introduced into EMPIRE code: PFNTKE, PFNALP, PFNRAT, and PFNERE (see section 1.3). Each incident energy produces a spectrum that is a function of the neutron outgoing energy. In order to accommodate bi-dimensional fits, such as that involved with PFNS, the whole Kalman process had to be modified. In the descriptions below the differences and similarities in format, structure and processing between fitting cross sections and PFNS will be detailed.

2.1 Sensitivity calculation

In order to assess the direction of the gradient in the multi-parameter space that minimizes χ^2, the fitting code has to evaluate how much the cross sections change when a certain parameter is varied in a given direction, i.e. the sensitivity of the cross section in respect to each parameter variation. This information is contained in the *sensitivity matrix* file.

The Kalman filter technique is used both in the resonance and in the fast neutron region. It is based on minimum variance estimation and naturally combines covariances of model parameters, of experimental data and of cross sections. This universality is a major advantage of the method. KALMAN uses measurements along with their uncertainties to constrain covariances of the model parameters via the sensitivity matrix. Then, the final cross section covariances are calculated from the updated covariances for model parameters. This procedure consistently accounts for the experimental uncertainties and the uncertainties of the model parameters ensuring that the final cross section uncertainties are at least as good as the smaller of the two. We emphasize that under the term 'reaction model' we mean also the resonance region described by models such as the Multi-Level Breit-Wigner formalism.

The key ingredient of the method is the sensitivity matrix, which represents complex nuclear reaction calculations. If we denote the combination of nuclear reaction models as an operator $\hat{\mathbf{M}}$ that transforms the vector of model parameters $\mathbf{p}$ into a vector of cross sections $\boldsymbol{\sigma}(\mathbf{p})$ for a specific reaction channel, then the sensitivity matrix $\mathbf{S}$ can be interpreted as the linear term in the expansion of the operator $\hat{\mathbf{M}}$,

$$\begin{aligned}\hat{\mathbf{M}}\mathbf{p} &= \boldsymbol{\sigma}(\mathbf{p}) \\ \hat{\mathbf{M}}(\mathbf{p}+\delta\mathbf{p}) &= \boldsymbol{\sigma}(\mathbf{p}) + \mathbf{S}\delta\mathbf{p} + \ldots \end{aligned} \tag{2.1}$$

We use 'hat' to stress that $\hat{\mathbf{M}}$ is the operator rather than a matrix. In practice, the elements $s_{i,j}$ of the sensitivity matrix are calculated numerically as partial derivatives of the cross sections σ at the energy E_i with respect to the parameter p_j,

$$s_{i,j} = \frac{\partial\sigma(E_i,\mathbf{p})}{\partial p_j}\,. \tag{2.2}$$

In case of covariance determination, the initial values of the parameters, $\mathbf{p}_0$, are already optimized, *i.e.*, when used in the model calculations they provide the evaluated cross sections. Their covariance matrix $\mathbf{P}_0$ is assumed to be diagonal while the uncertainties of the parameters are estimated using systematics, independent measurements or educated guesses. The model-based covariance matrix (prior) for the cross sections, $\mathbf{C}_0$, can be obtained through a simple error propagation formula,

$$\mathbf{C}_0 = \mathbf{S}\mathbf{P}_0\mathbf{S}^{\mathrm{T}}\,, \tag{2.3}$$

where superscript T indicates a transposed matrix.

The experimental data, if available, are included through a sequential update of the parameter vector $\mathbf{p}$ and the related covariance matrix $\mathbf{P}$ as

$$\begin{aligned}\mathbf{p}_{n+1} &= \mathbf{p}_n + \mathbf{P}_n\mathbf{S}^{\mathrm{T}}\mathbf{Q}_{n+1}(\boldsymbol{\sigma}^{\mathrm{exp}}_{n+1} - \boldsymbol{\sigma}(\mathbf{p}_n)) \\ \mathbf{P}_{n+1} &= \mathbf{P}_n - \mathbf{P}_n\mathbf{S}^{\mathrm{T}}\mathbf{Q}_{n+1}\mathbf{S}\mathbf{P}_n\,. \end{aligned} \tag{2.4}$$

Here,

$$\mathbf{Q}_{n+1} = (\mathbf{C}_n + \mathbf{C}^{\mathrm{exp}}_{n+1})^{-1}\,, \tag{2.5}$$

where $n = 0, 1, 2, ...$ and $n + 1$ denotes update related to the sequential inclusion of the $(n + 1)^{th}$ experimental data set. In particular, the subscript $1 \equiv 0 + 1$ denotes updating model prior $(n = 0)$ with the first experiment. Vector $\mathbf{p}_{n+1}$ contains the improved values of the parameters starting from the vector $\mathbf{p}_n$, and $\mathbf{P}_{n+1}$ is the updated covariance matrix of the parameters $\mathbf{p}_{n+1}$. The $\mathbf{C}^{\mathrm{exp}}_{n+1}$ is the cross section covariance matrix for the $(n + 1)^{\mathrm{th}}$ experiment. The updated (posterior) covariance matrix for the cross sections is obtained by replacing $\mathbf{P}_0$ with $\mathbf{P}_{n+1}$ in Eq. (2.3),

$$\mathbf{C}_{n+1} = \mathbf{S}\mathbf{P}_{n+1}\mathbf{S}^{\mathrm{T}} \,. \tag{2.6}$$

The updating procedure described above is often called Bayesian, although Eqs. (4-2.6) can be derived without any reference to the Bayes theorem.

The experimental covariance matrix, $\mathbf{C}^{\mathrm{exp}}_n$, is usually non-diagonal, due to the correlations among various energy points E_i. Assuming that systematic experimental uncertainties are fully correlated, the matrix elements are expressed through the statistical, $\Delta^{\mathrm{sta}}\sigma^{\mathrm{exp}}_n$, and systematic, $\Delta^{\mathrm{sys}}\sigma^{\mathrm{exp}}_n$, experimental uncertainties. This yields

$${}_n c^{\mathrm{exp}}_{i,i} = (\Delta^{\mathrm{sta}}\sigma^{\mathrm{exp}}_n(E_i))^2 + (\Delta^{\mathrm{sta}}\sigma^{\mathrm{exp}}_n(E_i))^2 \tag{2.7}$$

and, for $i \neq k$,

$${}_n c^{\mathrm{exp}}_{i,k} = \Delta^{\mathrm{sys}}\sigma^{\mathrm{exp}}_n(E_i) \times \Delta^{\mathrm{sys}}\sigma^{\mathrm{exp}}_n(E_k) \,. \tag{2.8}$$

An important technical issue, which has to be addressed in most of the covariance methods, is ensuring that the energy grid, E_i, for the model calculations and experimental data is the same to enable matrix operations in Eqs. (4-2.6). In the KALMAN code this is achieved by bi-spline interpolation of model cross sections and sensitivity matrices.

The above description can easily be generalized to account for correlations among different experiments. To this end one should construct a single vector containing all experimental points and the related covariance matrix, which now may contain blocks correlating different experiments. Only one update is needed in such a case but the covariance matrices are much bigger (in the current implementation of the Kalman filter the model-based covariance matrix is expanded to match the experimental one).

The quality and consistency of the evaluated cross sections can be assessed by scalar quantity

$$\chi^2 = \sum_{n=1}^{N} (\boldsymbol{\sigma}^{\mathrm{exp}}_n - \boldsymbol{\sigma}(\mathbf{p}_{\mathrm{N}}))^{\mathrm{T}} (\mathbf{C}^{\mathrm{exp}}_n)^{-1} (\boldsymbol{\sigma}^{\mathrm{exp}}_n - \boldsymbol{\sigma}(\mathbf{p}_{\mathrm{N}})) \,, \tag{2.9}$$

where $\mathbf{p}_{\mathrm{N}}$ is the final set of model parameters corresponding to the inclusion of N experiments. A value of χ^2 per degree of freedom exceeding unity indicates underestimation of the evaluated uncertainties. It is a fairly common practice to multiply such uncertainties by a square root of χ^2 per degree of freedom to address this issue.

The evaluator may choose to perform a sequential update using experimental data for several/all reactions or just for a single one. In the former case, all considered reactions are correlated and unique set of parameters along with the related covariance matrix are produced. On the other hand, poor experimental data in one reaction channel can negatively influence predictions for other channels.

We note that EMPIRE-KALMAN system is a general and powerful tool for evaluation of nuclear reactions. In addition to covariance calculations it may also be used to adjust model parameters to reproduce experimental cross sections and other observables within the selected reaction models and initial uncertainties of model parameters. Therefore, the Kalman filter can be used throughout the whole evaluation procedure to ensure consistency between cross sections, model parameters, and related covariance matrices.

The sensitivity matrix file ('`*-mat.sen`' for cross sections, '`*-pfns-full-mat.sen`') for PFNS, may be obtained through two different manners: using one of Empire's own subroutines; or using a python script to break the calculation into smaller parts and submit them to be run in parallel. Both methods follow the same concept and they are described in the following subsections. Actually, the sensitivity contained in those files is not the true sensitivity, but rather the so-called *realistic sensitivity*, which is the true sensitivity divided by the central value and multiplied by twice the interval of variation. The reason for this is that the realistic sensitivity provides much more information regarding the relative importance of the parameters.

2.1.1 Sequential calculations

By setting input parameter `KALMAN` to the value of 1 in EMPIRE's input, instead of performing a regular run of EMPIRE calculations, it will read the sensitivity input file in order to calculate the sensitivity matrices. This is very useful for simple fits and when access only to a sequential machine is available. However, when fitting more than just a few parameters, the sequential approach may take a prohibitive amount of time to conclude the calculation. As additional limitations, EMPIRE's sensitivity routines have not yet been prepared to generate sensitivity matrices for PFNS and for a certain class of cross-section parameters. For these reasons, it is advised for the user to perform parallel calculations instead, in order to generate sensitivity matrix files.

2.1.2 Parallel calculations (*sensitivity.py* python script)

One may use the *sensitivity.py* script, present in the '`empire/empy/qsubEmpire`' directory, to generate sensitivity-matrix file. This method splits the full calculation into single energy runs, submitting each incident energy for execution by a different processor node, shortening dramatically the total execution time. Firstly, it creates a directory, named '`*_orig/`', to where all the original input files are copied, for the calculation of the cross sections corresponding to central values of the parameters. Afterwards, the script reads the sensitivity input file to determine which parameters are going to be varied. Then, for each parameter, it creates two directories with names of the form '**_keyword_I1_I2_I3_I4*`plus/`' and '**_keyword_I1_I2_I3_I4*`minus/`', where *keyword* is the parameter name, and *I1-I4* are the four parameter options. All input files are also copied to those directories. However, in each directory, the input file is modified by changing the value of the parameter corresponding to the directory name. If the directory has the suffix ‘`plus`’ then the relative increment defined in the sensitivity input is added to the original parameter value. Likewise, for the directories with the suffix ‘`minus`’, then the same relative increment is subtracted from the original parameter value. Inside each of those directories, the script *qsubEmpire.py*, located in ‘`empy/qsubEmpire/`’ directory, is called. It then splits each run into separate directories with the same input parameters but containing only one value of incoming energy, and finally submits each of those as different jobs to different CPU’s for calculation.

The *sensitivity.py* script has basically five execution options:

- **-i**: Initializes the inputs and prepare to run. Executing with this option simply creates the ‘*_orig/’ directory and all the parameter directories described above. However, *qsubEmpire.py* script is not called, so the jobs are not submitted.
- **-r**: Calls *qsubEmpire.py* script to submit jobs for running. This option should only be executed after previous initialization (**-i**)
- **-a** or –**analyze**: After running (option **-r**), this option may be used to gather all the cross-section information from files ‘`*.xsc`’ in the different directories that were created and calculate the sensitivity matrix, creating the sensitivity matrix file ‘`*-mat.sen`’.
- **-p** or –**pfns**: After running (option **-r**), this option may be used to reorganize and gather all the PFNS information from files ‘`*-pfns.out`’ in the different directories that were created and calculate the PFNS sensitivity matrix, creating the PFNS sensitivity matrix file ‘`*-pfns-full-mat.sen`’.
- **-c** or –**clean**: This option is used to delete all the auxiliary directories that might have been created by a previous initialization or run of *sensitivity.py*

If executing *sensitivity.py* without specifying any options, it assumes options `-i` and, sequentially, `-r`. There are some examples of the option usage below:

```
$EMPIREDIR/empy/qsubEmpire/sensitivity.py u235 -i
$EMPIREDIR/empy/qsubEmpire/sensitivity.py u235 -r
$EMPIREDIR/empy/qsubEmpire/sensitivity.py u235
```

(this is equivalent of executing, in order, both commands above)

```
$EMPIREDIR/empy/qsubEmpire/sensitivity.py u235 -a
$EMPIREDIR/empy/qsubEmpire/sensitivity.py u235 -c
```

2.2 Kalman script

For KALMAN to run properly, its inputs (sensitivities, experimental data to be fitted, list of parameters, central values for cross sections and parameters, etc.) must be found in the appropriate files and in the expected format. Preparing those files manually for each run of KALMAN may prove to be a cumbersome and prone-to-error task. For this reason, it is convenient to use the *kalman* script, from the '`scripts`' directory. This script connects the different ingredients needed for the fit and organizes the results in a clear way, with the help of auxiliary codes.

The *kalman* script is called in a very simple manner by providing values for a set of variables in the command line, right after the script call. These variables are referred in the script as `file`, `MT`, `MAT`, `EXPDAT`, `PFNS`, and `EINC`, in this order. The first option corresponds to the project name, that is, EMPIRE's regular input file name without the extension '`.inp`'. The second option is the MT number that should be fitted by KALMAN. The next string corresponds to the material number (MAT) of the target. The variable `EXPDAT` defines which experimental sets will be used in the fit, by assuming the values 0, 1, or 2. `EXPDAT`=0 implies that all experimental data will be ignored (no fit); `EXPDAT`=1 indicates that only experimental data for the selected MT will be considered; while `EXPDAT`=2 informs the KALMAN code that agreement to all experimental data available in the c4 file should be considered for fitting. At this point, it is relevant to state that, when `EXPDAT`=2, the data agreement of the cross section (or PFNS) for a given reaction (or incident energy) may not improve, or may even get worse after the fitting. This is due to the fact that the overall χ^2 is being minimized and worsening of agreement with data for a given reaction (or incident energy) is likely being compensated by improvement of the agreement for all other reactions (or incident energies) When `PFNS` has value 0 (default), KALMAN will fit only cross sections, while when `PFNS`=1 it will fit prompt fission neutron spectra (PFNS). When `PFNS`=1, the variable `EINC` corresponds to the incident energy for which PFNS will be fitted. The options `PFNS` and `EINC` may be omitted when fitting cross sections since they serve only to fit PFNS. For the time being, these fits are independent and there is no correlation between cross sections and PFNS.

Examples of valid calls of *kalman* script:

```
kalman pu239 18 9437 2
kalman u235 18 9228 2 1 2.53E-8
```

In order to run properly, KALMAN needs to find a list of files in the working directory, otherwise it will abort and print an error message. Those files are: the ENDF file ('`*.endf`'), the sensitivity input file ('`*-inp.sen`'), the sensitivity matrix file ('`*-mat.sen`'), and the EMPIRE's cross section file ('`*.xsc`'). If fitting PFNS instead of cross sections, the last two files aforementioned are not necessary. However, KALMAN will then need the content of files '`*-pfns.out`' and '`*-pfns-full-mat.sen`'.

After finishing running, the *kalman* script creates several output files containing important information regarding the fits. Such files, with a brief the description, are listed below:

- '`*-out.kal`': Standard output from KALMAN. Contains information about the data sets used for fitting, the χ^2 obtained, the final values for the fitted parameters and their corresponding uncertainties, and the correlation between parameters.

- '`*-expcorr.kal`': Provides the fraction of the experimental uncertainty that corresponds to systematical errors for each data set. This file is initially generated by the code '`genkal`', as file '`fort.12`', having the default value of 0.200 for all experiments.
- '`*-`*MT*`-err.kal`', where *MT* corresponds to the same script variable `MT` which was selected in the command line: Contains the covariances and uncertainties of the fitted curve.
- '`*-par.kal`': Model parameter correlations
- '`*-xsc.kal`': Final curves found by KALMAN for all reactions (or incident energies, in the PFNS case)
- '`*-cov.kal`': Contains the uncertainties (in units of %) and correlations for all incident (or outgoing in the PFNS case) energies.
- '`*-expxsc.kal`': Contains the experimental data used in the fit.
- '`*-parcorr.kal`': Contains a list of the varied parameters, with their initial values (usually set to 1.0) and ranges of variation. Additionally, it also contains, correlation matrix for the parameters, which is usually diagonal in the first default calculation. This matrix might be changed or updated if the parameter correlation matrix is know. Then, true sensitivity matrices for each reaction (or each incident energy, in the PFNS case) are printed.
- '`*-pfns.kal`': File containing the PFNS for the fitted incident energies.
- '`*-`*MT*`-cov.gpd`', where *MT* corresponds to the same script variable `MT` which was selected in the command line: File to plot a three-dimensional graph of the covariances using `gnuplot`
- '`*-`*MT*`-c4.gpd`', where *MT* corresponds to the same script variable `MT` which was selected in the command line: File that contains the experimental data for the reaction corresponding to the *MT* number used in the KALMAN fit. May be plotted together with '`*-`*MT*`-xsc.gpd`' using '`gnuplot`'.
- '`*-PFNS-`*INDEX*`-c4.gpd`': File that contains the pfns experimental data for the incident energy corresponding to the index *INDEX*, as listed in file '`*-pfns.kal`'. May be plotted using '`gnuplot`'.
- '`*-`*MT*`-xsc.gpd`', where *MT* corresponds to the same script variable `MT` which was selected in the command line: Contains both the curves before and after the KALMAN fit. It may be plotted together with '`*-`*MT*`-c4.gpd`' using `gnuplot`.

Plots of the fit can be made using the information contained in files '`*.gpd`' with the help of the widely-available plotting tool '`gnuplot`'. Plots of the parameter correlations resulting from the KALMAN fits can be made with '`util/kalman/corr.gp`'; correlations of the PFNS parameters with ('`util/kalman/corr-pfns.gp`'. This plotting process is described in Section 2.2.4.

2.2.1 Experimental Data

For fitting either cross sections or PFNS, KALMAN needs access to the experimental data so it can calculate and compare the χ^2 at each point. The codes '`c4tokal`' and '`pnt2kal`' read experimental data in computational format (c4 files) and write in a way that is understandable by the KALMAN code. The former process cross section data while the latter PFNS data.

Handling Cross-Section Experimental Data - c4tokal

The code '`c4tokal`' reads the cross-section experimental data from the file '`*-kal.c4`', if it exists. Otherwise, it looks for the existence of file '`*.c4`'. The reason for the additional '`*-kal.c4`' file is that it is common, during the fitting process, that one might want to exclude data points or even whole data sets from the fit. This way the user can edit the file '`*-kal.c4`' while keeping the original and more complete '`*.c4`' unaltered.

EMPIRE's cross-section file '`*.xsc`' is read by '`c4tokal`' only to determine the number of different reactions printed out by Empire and their corresponding nomenclature. In addition, the sensitivity input file '`*-inp.sen`' is read to obtain the number of parameters to be varied in the fit.

After gathering all this information, '`c4tokal`' creates a series of output files, which will be read by KALMAN. Below is a brief description of each of those files:

- **fort.75**: contains data points, their absolute errors, and the MT number
- **fort.10**: contains only the data points
- **fort.11**: contains only relative errors for each energy
- **fort.12**: contains correlations between different experiments
- **KALMAN.INP**: automatically-generated input for the main KALMAN fitting code

Handling PFNS Experimental Data - pnt2kal

When compared to cross-section fiting, preparing PFNS experimental data require additional steps due to the fact that PFNS is usually presented after normalization to Maxwellian. This normalization is often done because PFNS vary largely across the values of outgoing energy. Since the value of the temperature used in the Maxwellian normalization (T_{maxw}) is relatively arbitrary, different T_{maxw} may be used to focus on different parts of the spectra.

Instead of calling '`c4tokal`, kalman script calls the code `pnt2kal` when fitting PFNS. This code reads the normalized PFNS experimental data from an existing '`*.pnt`' file and converts it to the input format needed by KALMAN. If this file does not exist, '`pnt2kal` runs the code LSTTAB to read the PFNS data from the '`*.c4`' file and generate the corresponding normalized '`*.pnt`' file. In this case, the code PLOTLST is initially called to identify the index of the PFNS data in the c4 file (with output sent to file 'output.plotlst'). Then, the Maxwellian temperature T_{maxw}, to which the data will be normalized, is read from EMPIRE's PFNS output file, '*-pfns.out'. With that value, an input file for LSTTAB, 'input.lsttab', is written. The files '*-log.plotc4', '*.c4', and '*-s.endf', together with the value of T_{maxw}, read from '*-pfns.out' are necessary to run LSTTAB. When LSTTAB is called by 'pnt2kal, its standard output is written in the file 'output.lsttab' and the standard pnt file 'LSTTAB.PNT' is renamed to '*.pnt'.

When the '*.pnt' is ready, the PFNS data and all its relevant information are read at once and stored in a structured variable. From this, the values of incident energies (E_{inc}) for which there are available data are extracted. Those values are compared to the ones used for the original Empire calculation of the PFNS, to check if PFNS were calculated at all incident energies found in the experimental '*.pnt' file. If not, 'pnt2kal' aborts since it makes no sense to fit data without the initial calculation of its central values. In this case, an error message is printed with all the values of E_{inc} and indicating which matches could be found in the PFNS output from EMPIRE.

An important difference between integrating cross-section and PFNS fitting with EMPIRE is that, in the former case, the initial central values are already in an easily-readable format in the file '*.xsc', with the cross sections for each E_{inc} in subsequent rows, and with the different reactions distributed in columns. In the PFNS case, however, the spectra are printed as sequential blocks for the different incident energies, with each block containing the distribution as a function of the outgoing energy. For KALMAN to be able to fit PFNS properly, the spectra have to be rewritten in a format similar to that of the cross sections in the file '*.xsc'. The code 'pnt2kal does that by reading the spectra from file '*-pfns.out' and writing them in a reformatted file, with name '*-pfns.fmt'. In this new file the PFNS are written as function of the outgoing energy, with the different E_{inc} distributed in columns. This way, we have the correspondence expressed in Table 2.1

Table 2.1: Correspondence between cross-section and PFNS files

Quantity	cross sections	PFNS
File extension	'.xsc'	'-pfns.fmt'
Rows	incident energy	outgoing energy
Columns	reaction	incident energy

As mentioned, the file '*-pfns.fmt' contains the PFNS at all incident energies calculated by EMPIRE. However, it is only necessary to provide to KALMAN those spectra at the same incident energies for which there are experimental data available in the '*.pnt' file. For this reason, 'pnt2kal reads '*-pfns.fmt' and rewrites it in file '*-pfns.kal', keeping only the columns corresponding to E_{inc} found in '*.pnt'. This trimming process is also done to the original PFNS sensitivity file, '*-pfns-full-mat.sen', with the outcome being printed in the file '*-pfns-mat.sen'. For doing this, the number of parameters is retrieved from the sensitivity input file '*-inp.sen'.

Finally, 'pnt2kal' creates and writes to files 'fort.75', 'fort.10', 'fort.11', 'fort.12', and 'KALMAN.INP'. The contents of those files are the same as in the case of cross-section fitting (section 2.2.1), with the corresponding substitutions expressed in Table 2.1. In addition to those files, ‘pnt2kal’ also writes to a temporary file, named ‘ENERGYANDINDEX.TMP’, that contains some information needed later on by the *kalman* script, such as number of incident energies, and index and value of E_{inc} to be fitted. This file is deleted at the end of kalman script.

2.2.2 Central Values - genkal

The ‘genkal’ code converts the cross sections and sensitivities for all reactions into the format needed by the KALMAN code. The cross sections are read from file ‘*.xsc’ and are rewritten in file ‘fort.50’ in such a way to present the central values for the cross sections as sequential blocks, each block corresponding to one of the reactions. The sensitivity matrices are read from ‘*-mat.sen’ and rewritten into file ‘fort.52’. The initial intervals within each parameter may be varied by KALMAN are also written in the file ‘fort.52’.

The central values and sensitivities in the case of PFNS fitting are converted to KALMAN format by the same code ‘genkal’ that does that for cross sections. File ‘fort.50’ is written after reading the central spectra from file ‘*-pfns.kal’. It will have the PFNS rewritten as sequential blocks, with each block now corresponding to the spectrum distribution for each incident energy. The PFNS sensitivity matrix will be read from file ‘*-pfns-mat.sen’ and stored again in the file ‘fort.52’. As in the cross section case, the relative intervals for parameter variation will also be written in ‘fort.52’.

In both cross-section and PFNS cases, ‘fort.52’ is renamed to ‘*-parcorr.kal’ if such file does not exist beforehand. If file ‘*-parcorr.kal’ exists, then it is copied over ‘fort.52’ so, following the general EMPIRE policy, the existing version is used rather than the newly-generated one. The reason for this is that, after a given run of Kalman, a user might want to edit this file in order to modify some values, such as the range of variation for a given parameter.

2.2.3 KALMAN code

If ‘genkal’ and ‘c4tokal’ (or ‘pnt2kal’, in the PFNS case) have run properly, then all the files needed for KALMAN to run will be ready. However, before executing KALMAN, the *kalman* script checks if the files ‘*-expxsc.kal’, ‘*-expcorr.kal’, and ‘*-parcorr.kal’ exist. If so, the script renames them in such a way that the fitting code will use the existing files instead of the newly-generated ones. As mentioned above, this allows the user to manually modify those files according to specific needs, between two consecutive runs of KALMAN.

Afterwards, the *kalman* script opens the standard KALMAN input file, KALMAN.INP, previously created by either ‘c4tokal’ or ‘pnt2kal’, using a text editor. At this point, the user may choose to edit this file to modify the default options before the actual run. Once the user is satisfied with the input file, should save it and close the editor window. This will trigger execution of the KALMAN code and start the actual fitting process.

2.2.4 Post-KALMAN processing

After the KALMAN code completed fitting the experimental data sets and has found an optimal set of parameters that minimizes the χ^2, the kalman script does a few more operations. Firstly, it renames the output files generated by KALMAN '`fort.13`', '`fort.14`', and '`fort.15`' into '`*-xsc.kal`', '`*-cov.kal`', and '`*-par.kal`', respectively. Secondly, it calls the code '`kalend`' which reads the KALMAN-generated covariance and writes it in the ENDF-6 format into file '`*-`*MT*`-err.kal`'. Additionally, it also creates the files '`corrplot.d`' and '`xscplot.d`', for plotting.

Finally, before finishing, *kalman* script uses `gnuplot` to plot the calculated covariances and pre- and post-fit curves to compare with experimental data. To do this, it uses the `gnuplot`-instruction file '`util/kalman/corr.gp`' ('`util/kalman/corr-pfns.gp`', in the case of fitting PFNS). Both files '`corr.gp`' and '`corr-pfns.gp`' are set to plot the covariances directly from file '`corrplot.d`'. The curves before and after fitting are set to be taken from file '`xscplot.d`', for either cross-section or PFNS calculations. Both files '`corrplot.d`' and '`xscplot.d`' are directly generated by '`kalend`'. The file containing the experimental data for plotting ('`*-`*MT*`-c4.gpd`' for cross sections, or when plotting PFNS, '`*-PFNS-`*INDEX*`-c4.gpd`') was previously generated either by '`c4tokal`' (in the case of cross sections) or by '`pnt2kal`' (in the PFNS case). However, both '`corr.gp`' and '`corr-pfns.gp`' are set to take the experimental data from file '`expxscplot.d`'. Therefore, right before calling '`gnuplot`', *kalman* script copies file '`*-`*MT*`-c4.gpd`' (or '`*-PFNS-`*INDEX*`-c4.gpd`', depending to which quantity has been fitted) to file '`expxscplot.d`'.

Once the '`gnuplot`' command is executed, the three-dimensional graph of the covariance should be plotted. Figure 2.1 shows examples of such graphs for the case of fission (top pannel) and capture (bottom pannel) of neutrons inciding on ^{235}U. After the user closes the covariance-graph window and hits enter on the terminal, the graph of prior- and post-KALMAN fitting curves is plotted. Figure 2.2 shows the agreements with fission (top pannel) and capture (bottom pannel) experimental data of the calculated cross sections before (blue curve) and after (green curve) the fit.

It is important to mention that this post-fitting calculation does not necessarily correspond to the exact result that one should get when the new values of the parameters are put back into EMPIRE's input file for a new run of EMPIRE. This happens because the Kalman filter is a linear fitting code, which means that, the effect of perturbing a given parameter will be calculated by linearly interpolating (or extrapolating) the sensitivity matrix. Hence the importance of choosing wisely the intervals of variation, both when obtaining the sensitivity matrix and when editing '`*-parcorr.kal`' to avoid falling out of the linearity regime. The final effect of fitting parameters through KALMAN will thus only be known after the new parameter values are put back into EMPIRE's input file and new cross sections or PFNS are calculated.

2.3 Fitting example: ^{235}U

As an example, we present below the entire process of generating the central values, sensitivities, and KALMAN fits, with a complete set of commands, for the target material ^{235}U.

Once EMPIRE's initial input files have been defined and assuming EMPIRE's main input file is named as '`u235.inp`', one can run EMPIRE, generate ENDF-6 formatted file, and process it for plotting by issuing the following commands:

```
$EMPIREDIR/scripts/runE u235
$EMPIREDIR/scripts/format u235 9228
$EMPIREDIR/scripts/process u235
```

or, alternatively:

```
$EMPIREDIR/scripts/run u235 9228
```

In order to calculate sensitivities, the sensitivity input file '`u235-inp.sen`' must be created, containing the list of parameters to be varied. This file may contain both cross-section and PFNS parameters. Since, at the time being, those fits are independent and uncorrelated, the cross-section sensitivity file ('`u235-mat.sen`') will contain zero sensitivities for the PFNS parameters and, likewise, the PFNS sensitivity file ('`u235-pfns-full-mat.sen`') will contain vanishing sensitivities for the cross-section parameters. To calculate the sensitivities and generate the sensitivity matrix files for both cross sections and PFNS one may use the commands:

```
$EMPIREDIR/empy/qsubEmpire/sensitivity.py u235
$EMPIREDIR/empy/qsubEmpire/sensitivity.py u235 -a
$EMPIREDIR/empy/qsubEmpire/sensitivity.py u235 -p
```

At this stage, all the files needed to start the fitting process have been generated. If the user chooses to fit cross sections taking into consideration experimental data for all reactions available in the `c4` file ('`u235.c4`'), and wishes to see the correlation and prior/post-fit graphs for fission reaction, for example, he/she may use the command:

```
kalman u235 18 9228 2
```

The result of this KALMAN run is shown in Figures 2.1 and 2.2, top pannels. If, however, the user chooses instead to see the KALMAN predictions for capture, the following command may be used:

```
kalman u235 102
```

Figures 2.1 and 2.2, bottom panels, show the results of this run.

To fit PFNS experimental data for all incident energies contained in the `c4` file ('`u235.c4`'), but to plot the KALMAN predictions for thermal energy, one may use the command:

```
kalman u235 18 9228 2 1 2.53E-8
```

After the final parameters, which may be found in this case in file ‘`u235-out.kal`’, are put back into EMPIRE’s main input file, ‘`u235.inp`, EMPIRE may be executed again with command:

```
$EMPIREDIR/scripts/run u235 9228
```

to obtain the final results, for cross sections and PFNS, after the KALMAN fit Figures 2.3, 2.4, and 2.5 show examples of results for cross-section and PFNS fits for ^{235}U using Kalman filter integrated with EMPIRE.

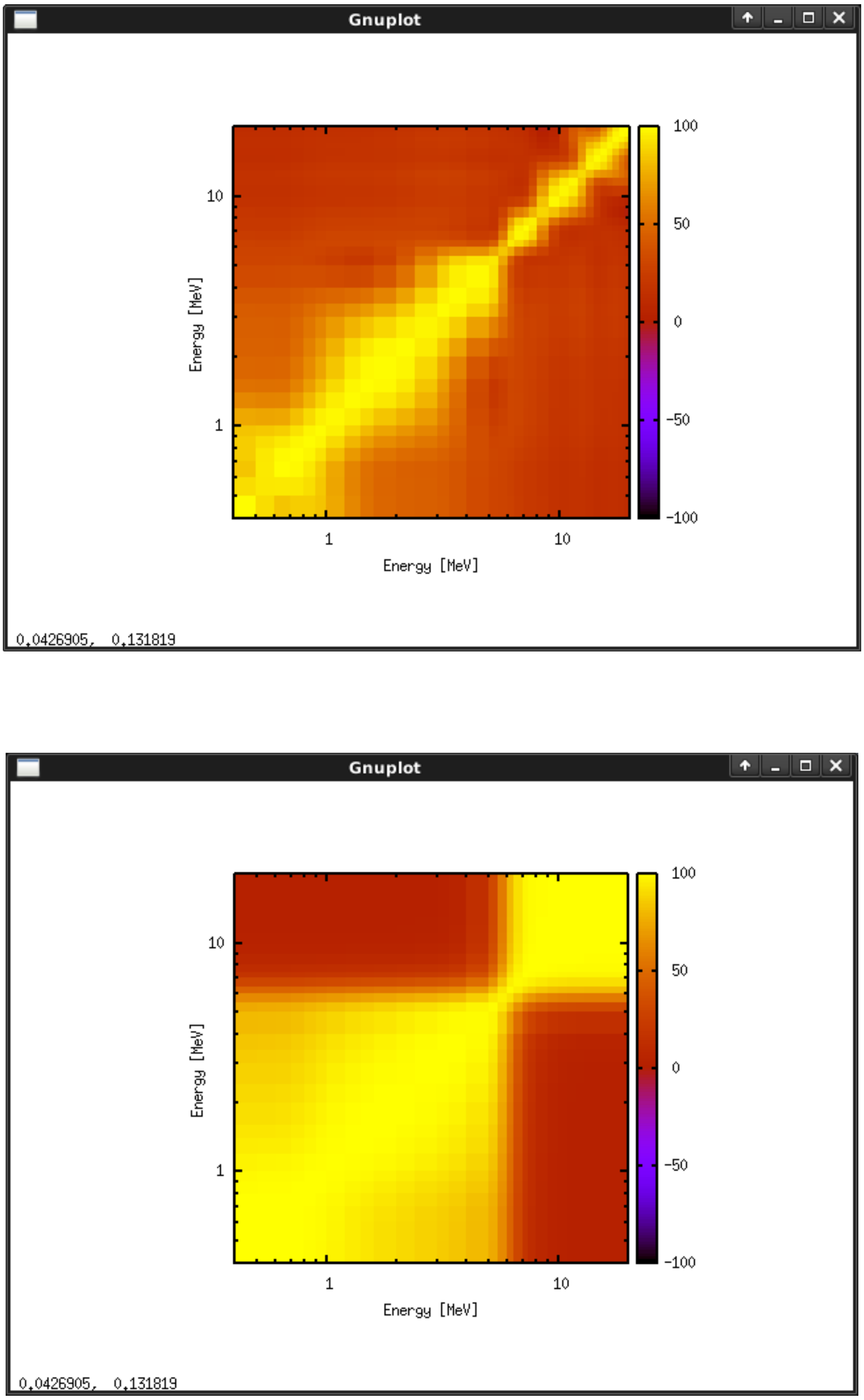
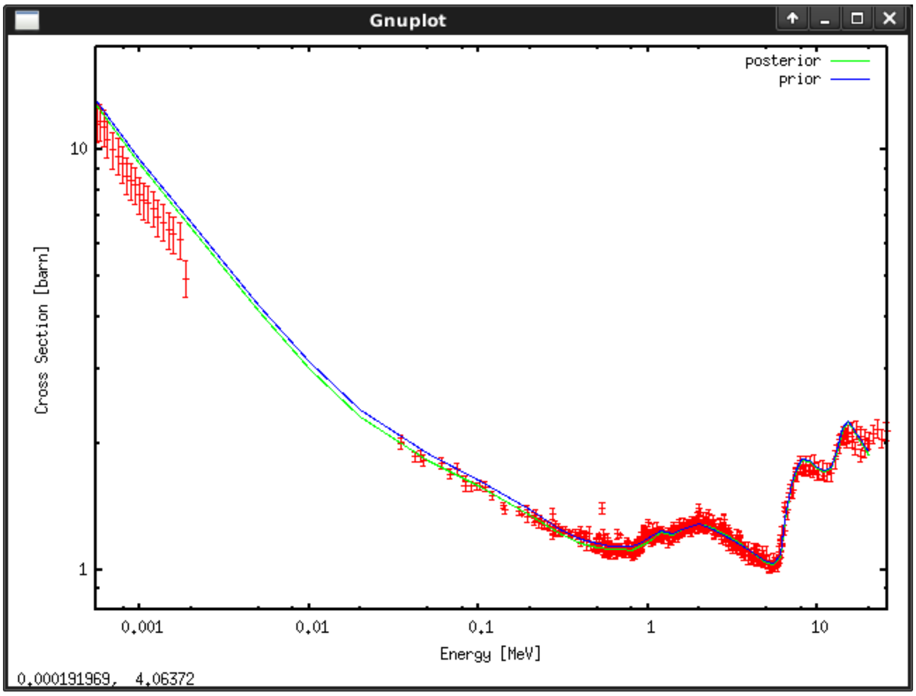


Figure 2.1: Correlation matrices for neutron-induced fission (top) and capture (bottom) reactions ($MT =$ 18 and 102, respectively) for material ^{235}U.



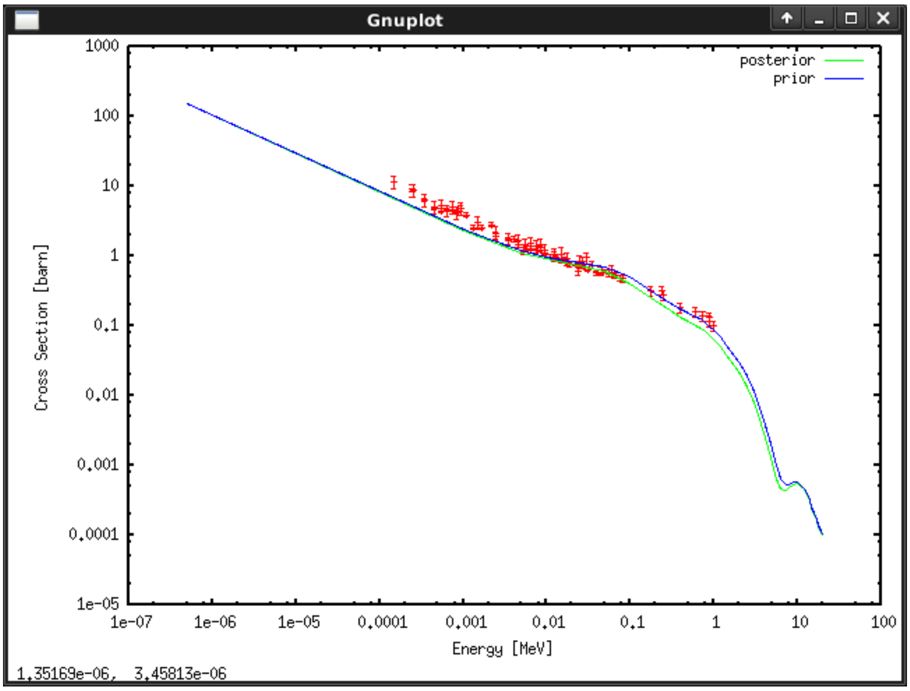


Figure 2.2: Cross section for the neutron-induced fission (top) and capture (bottom) reactions ($MT = 18$ and 102, respectively) for material ^{235}U, comparing the agreement with experimental data between the curves prior and post KALMAN fitting.

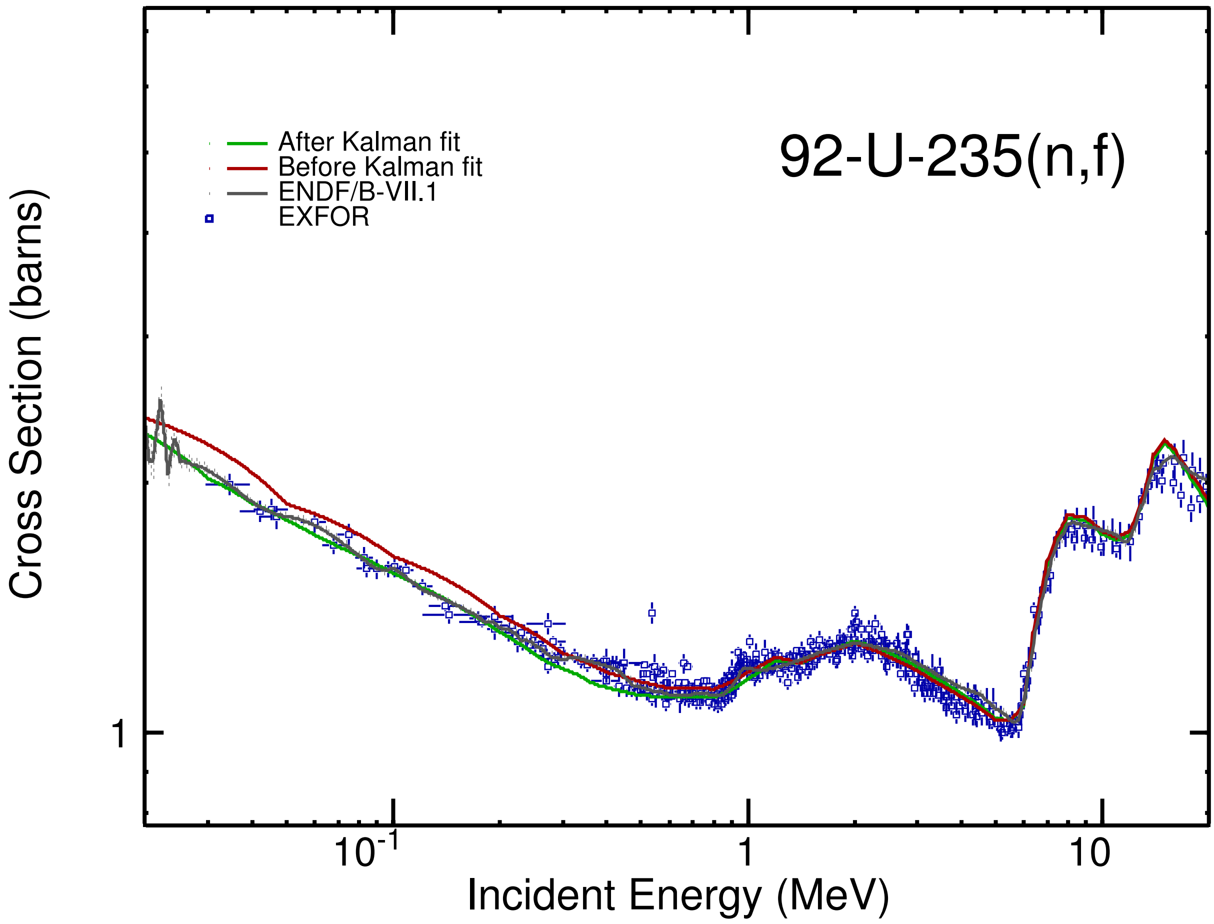


Figure 2.3: Comparison between fission cross sections calculated by EMPIRE using input parameters obtained before and after fitting using Kalman filter. Evaluation from ENDF/B-VII.1 library and experimental data from EXFOR are also plotted.

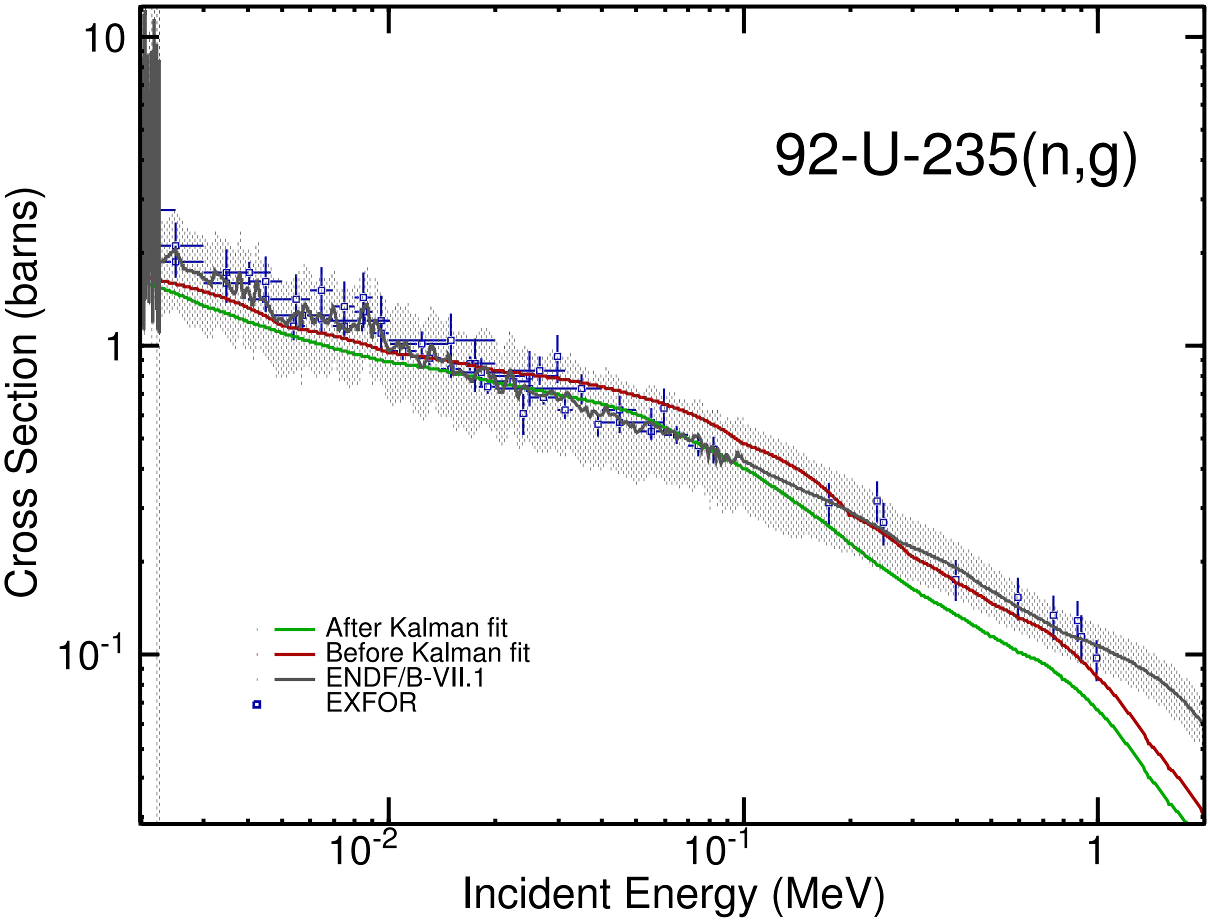


Figure 2.4: Comparison between capture cross sections calculated by EMPIRE using input parameters obtained before and after fitting using Kalman filter. Evaluation from ENDF/B-VII.1 library and experimental data from EXFOR are also plotted. Even though the agreement of the post-fit curve seems worse than that of the pre-fit one, this is compensated by improvements at other reactions, minimizing the overall χ^2. This may happen when the option to fit data for all reactions is selected.

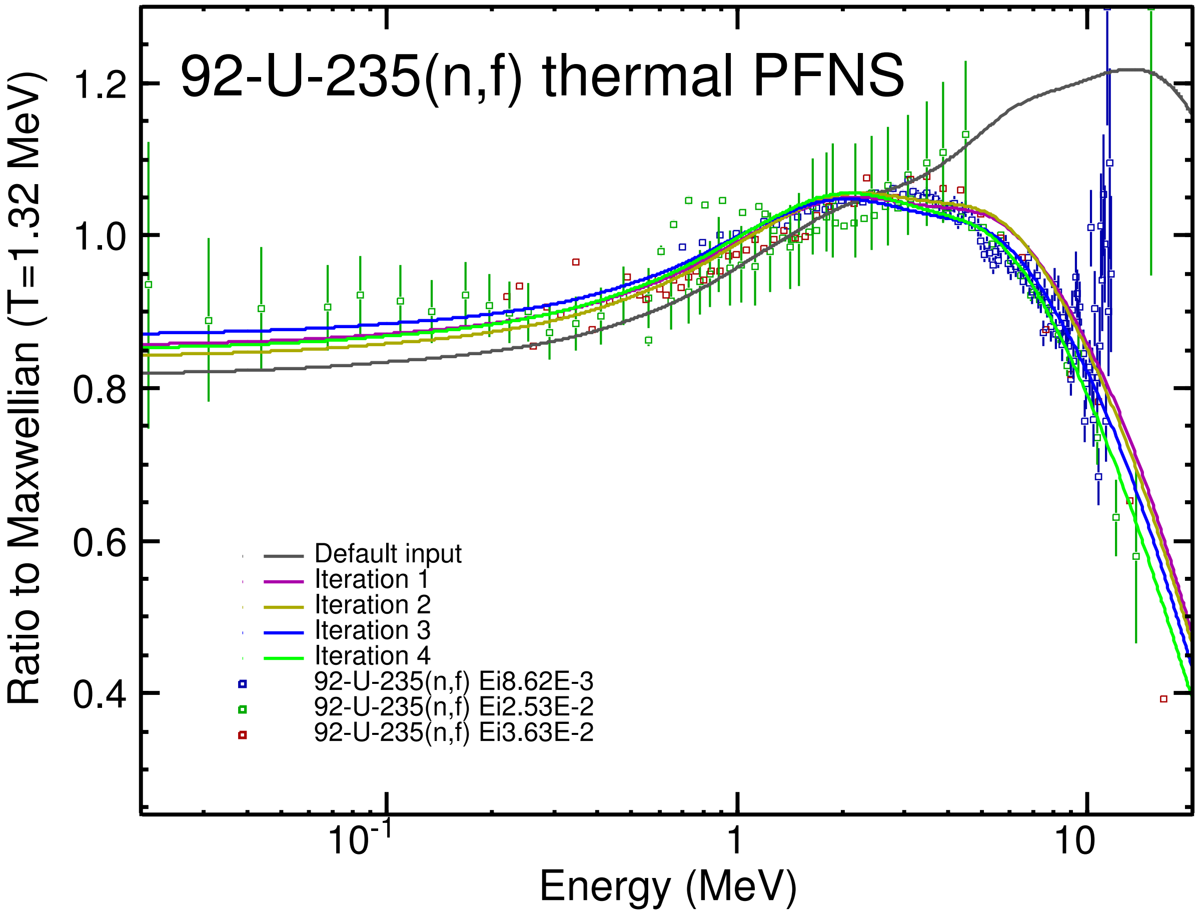


Figure 2.5: Prompt fission neutron spectra (PFNS) for thermal neutrons calculated by EMPIRE using a default input and input files containing PFNS parameters after different numbers of KALMAN-EMPIRE iterations. For iterations 3 and 4, additionally to iterating a new sequence of KALMAN and EMPIRE runs, some spurious data points were removed from the fitting process.

Section 3

Fitting Mubar and Nubar in Empire with Kalman

The fitting code KALMAN, used within the EMPIRE nuclear modeling system, is a fitting package based on the Kalman fitting technique where a set of empire model parameters are adjusted to fit a set of data [6]. The required input is only the data (observables) being fit, the predicted value for those data from the model, and the "sensitivity" of each observable to each fitted parameter, where the "sensitivity" is simply the first derviative of the observable with respect to the fitted parameter. These sensitivities are calculated numerically. They are prepared as a matrix, where one dimension is the parameter index and the other is the index of the observable. For each parameter x_i, the sensitivity of observable $OB(j)$ is calculated by adjusting the parameter up and down by a given amount, and then calculating the observables for these adjusted parameters, which are designated with a superscript + or −. The resulting sensitivity of the jth observable to the ith parameter is then simply given by

$$\text{Sens}(j)_i = (OB(j)^+ - OB(j)^-)/(x_i^+ - x_i^-) \tag{3.1}$$

This matrix of sensitivities, along with the predicted values of the observables without adjustment (often referred to as the "central values," as the parameters are set to original values between the "+" and "-" values, hence "central") are then used to adjust the parameters using the Kalman method to minimize the difference between the fitted value and the prepared set of observables in the standard C4 data format, using the usual definition of χ^2 as the sum of differences between the model predictions Y_i and measured quantities y_i divided by the measurement uncertainty σ_i. The sum of the squares of these ratios forms χ^2,

$$\chi^2 = \sum_{i=1}^{n} ((Y_i - y_i)/\sigma_i)^2 \tag{3.2}$$

where the summation is over all measured observables n. For further details about the Kalman approximation used in fitting see [6].

This approach has been successfully used with the Empire package for fitting cross sections. It includes, but is not limited to, total, elastic, inelastic, (n,2n), fission, and capture. The user creates a file which lists each empire parameter to be fitted along with a displacement which limits the range of the parameter when fitted by the Kalman code and also to determine the "+" and "-" values of Eq. 3.1 when calculating the sensitivity to the parameters for each fitted cross section. As such, these values should be chosen to be as large as possible as not to restrict the fit, but also not so large as to introduce nonlinearities into the calculation of the sensitivities. This amount varies for each parameter and its functional form within the EMPIRE code.

Here we discuss an extension of this fitting that will allow one to fit in addition to the usual cross sections, the prompt neutron fission spectra (PFNS), the average cosine of the scattering angle for elastic neutron scattering ($\bar{\mu}$), and the average number of neutrons produced in fission ($\bar{\nu}$). For a detailed description of the PFNS fitting, see section 2. The addition of $\bar{\mu}$ and $\bar{\nu}$ fitting in empire/kalman is described here.

3.1 Mubar

When fitting EMPIRE calculations to experimental observables with Kalman, the data are prepared for the fitting code by the kalman script which controls the processing of the C4 data and the standard EMPIRE cross sections and sensitivities into the format used by the KALMAN code. A mapping function is used to map each reaction from EMPIRE to the corresponding MT as specified for that reaction in the C4 data file. This is done for various MF3 cross sections, but not for angular distributions, etc. As a first step toward including sensitivity to the angular distributions we describe the addition of the capability of fitting the $\bar{\mu}$ as calculated by EMPIRE to experimental values. This involves 2 steps: (i) fitting of the experimental elastic angular distributions to produce an experimental value for $\bar{\mu}$ with uncertainties and (ii) the addition of $\bar{\mu}$ as another "reaction" in the file of cross sections produced by EMPIRE for fitting with Kalman (the XSC file). Here we describe the preparation of both the experimental and calculated values.

The experimental differential cross sections (C4 MF4/MT2) for elastic scattering are fit with a Legendre polynomial expansion to extract the coefficient from the fit for the first Legendre coefficient, which corresponds to $\bar{\mu}$ or the average scatting cosine (here, the 0th Legendre parameter corresponds to the elastic scattering cross section). To fit these Legendre coefficients two similar approaches have been developed.

In the first approach, ang_mu.f code, developed by A. Trkov at BNL, scans the input C4 data file and each section of elastic scattering differential cross sections are fitted using standard least-squares with the sum of the squares of the differences between the experimental cross sections and the Legendre fit being minimized. This method will iterate the fit, automatically increasing the order of the fit until the maximum relative difference between the data and the fit are less than the maximum relative error in the diff. cross section divided by $\sqrt{2}$, but never less than 7%. The order of the fit is also never allowed to exceed $l = 22$ or 2/3 the number of points in the angular distribution, whichever is smaller. This will produce a value for $\bar{\mu}$ for each elastic differential cross section encountered in the C4 file from the 0^{th} and 1^{st} Legendre moments as

$$\bar{\mu} = \frac{c_1}{3c_0} \tag{3.3}$$

where the factor of 3 comes from re-normalizing the angular distribution by $2l + 1$, with $l = 1$. These fitted Legendre coefficients are written to the C4 file with MF154/MT2 which are paired with the calculated values for $\bar{\mu}$ from Empire, which are assigned to MF3/MT251. The files are processed and prepared for the Kalman fitting code by the routine c4tokal.f90.

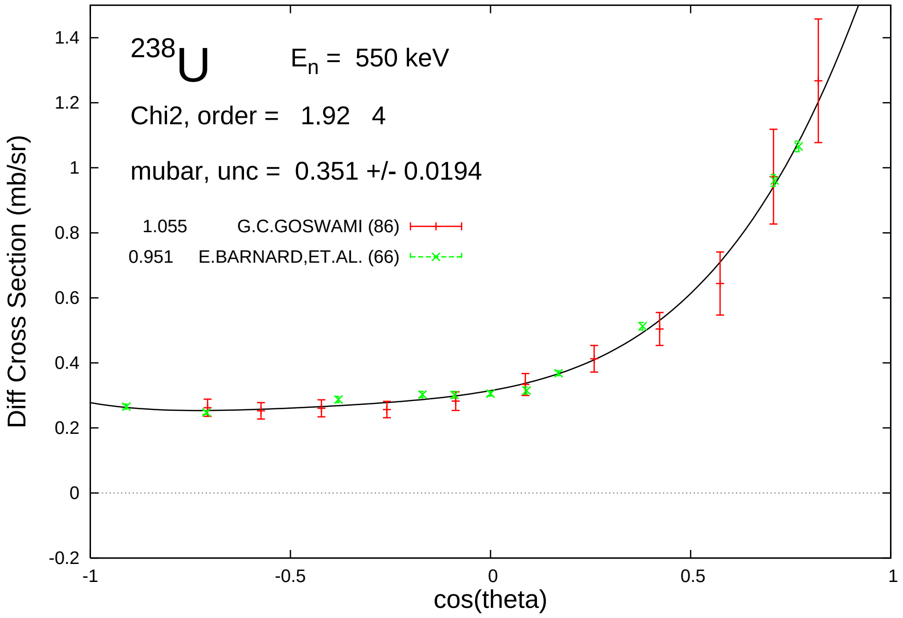


Figure 3.1: Legendre fit to differential cross section data at E = 550 keV. There were 2 data sets available at this energy. Each data set was adjusted by a scale factor that was fitted along with the Legendre coefficients by including a penalty term in the χ^2 (see Eq. 3.4). This resulted in a good fit to the data with only a 4th order Legendre fit with a final reduced χ^2 of 1.92.

The Legendre fitting of the angular distributions will suffer if the number of data points is small or the data do not cover enough angular range to successfully constrain the fit. In an attempt to increase the coverage, when available, a second version of the Legendre fitting was developed that will group the data by incident neutron energy, which then may fit more than one C4 data section at once, increasing the number of points and angular coverage which may result in a better fit. In this version, the uncertainty of the individual data points are used in creating the χ^2, as in Eq. 3.2. Each data set is assigned a systematic uncertainty and a scale factor is used to bring the various data sets into agreement, which is important to achieve a good fit when fitting more than one data set at a time. A scale factor f for each data set is added to the definition of χ^2 and a penalty term added for each data set weighted set's systematic error σ_f as shown in Eq. 3.4 (see ref. [7]).

$$\chi^2 = \sum_{i=1}^{n}((Y_i - fy_i)/f\sigma_i)^2 + ((f-1)/\sigma_f)^2 \tag{3.4}$$

The order of the Legendre fit was also automatically increased, where the order of the fit would increase until $\chi^2 < 2.5$ but never allowed to exceed 2/3 of the total number of points being fit. This method yielded uncertainties on the coefficients of the Legendre polynomials which could then be directly propagated to an uncertainty in $\bar{\mu}$ as defined in Eq. 3.3. An example of this fitting procedure is shown in Figure 3.1. Here 2 different data sets were each assigned a 5% systematic error. The fit scaled the data set of Goswami, et. al. up by 5.5% and the data set by E.Barnard, et. al., down approximately 5%. The resulting angular distribution was fit using only a 4^{th} order Legendre polynomial with a reduced χ^2 of 1.92.

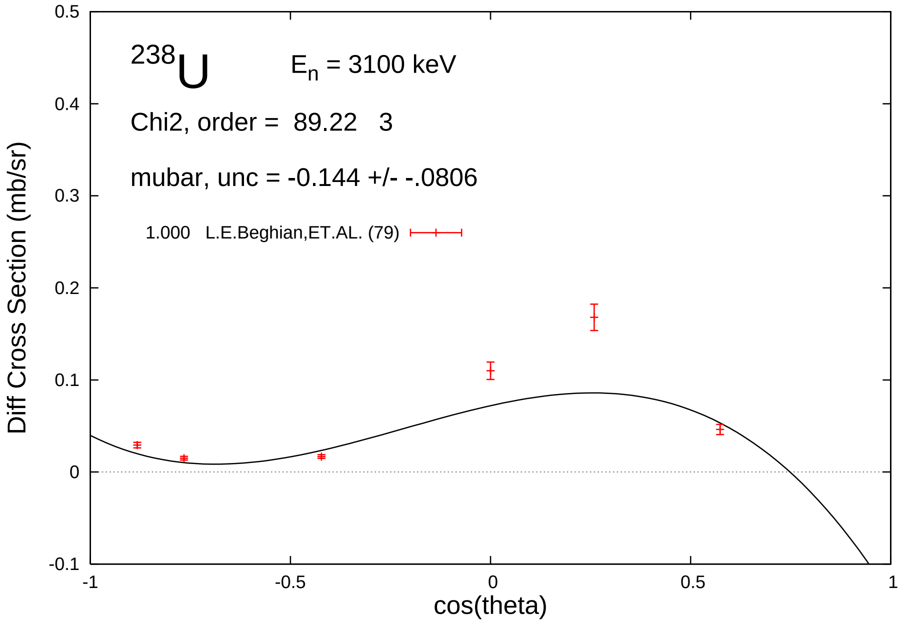


Figure 3.2: Legendre fit to differential cross section data at $E = 3.1MeV$. An example of a data set with too few points and too small angular coverage. The fit fails with negative cross sections being produced, resulting in a nonsensical value for $\bar{\mu}$.

These procedures for fitting $\bar{\mu}$ are fairly automated, but care must be taken and each fit must be visually checked. A common cause for a bad fit is simply insufficient angular coverage or too few points to constrain the fit, or both. An example of this type of failure is shown in Fig. 3.2. Here, there are few data points and none at forward angles. The resulting fit has a negative cross section at forward angles. The resulting value for $\bar{\mu}$ is meaningless should not be used.

Another case where there are often problems are at high energy ($\mathrm{E_n}$ > 10 MeV), where the angular distribution is very forward peaked. Fitting these distributions usually requires a very high order Legendre polynomial, which can easily become unconstrained at very forward or backward angles, resulting in a bad value for $\bar{\mu}$. An example of such a fit at $E_n = 15$ MeV is shown in Fig. 3.3. At high energy, it is often preferable to not

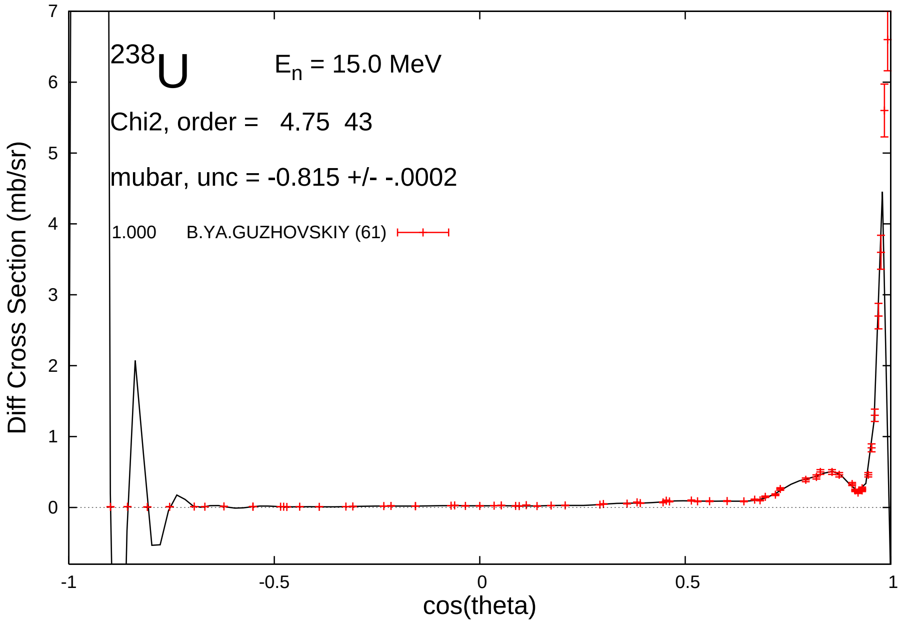


Figure 3.3: Legendre fit to differential cross section data at $E = 15$ MeV. Here the forward angle peak of the angular distribution requires a very high order Legendre polynomial for a good fit which results in a relatively good χ^2 but the high oscillations at the rather unconstrained backward angles result in a bad value for $\bar{\mu}$ of -0.815.

fit the angular distribution but rather calculate $\bar{\mu}$ directly, determining the the average cosine of the differential cross section as

$$\bar{\mu} = \frac{\int_{-1}^{1} \frac{d\sigma(\theta)}{d\Omega} \cos\theta \, d\cos\theta}{\int_{-1}^{1} \frac{d\sigma(\theta)}{d\Omega} \, d\cos\theta}. \tag{3.5}$$

For the distribution shown in Fig. 3.3, calculating $\bar{\mu}$ using Eq. 3.5 results in a value of 0.82 ± 0.03, an acceptable value, rather than the result from the high-order Legendre fit of -0.815 ± 0.0002, which is clearly wrong.

Of course, to successfully use Eq. 3.5 the angular coverage of the differential cross section must be sufficient to perform the integral. If this is not the case then there is little choice but to use a Legendre fit. In any case, care must be always taken when determining the value of $\bar{\mu}$ for a given elastic differential cross section. Each fit must be visually inspected and only those that result in a good fit without problems (e.g., unconstrained or negative cross section, etc.) should be accepted for Kalman fitting. In the end, it is up to the evaluator to decide which method provides the best value for $\bar{\mu}$ from a given differential cross section. The accepted values with uncertainties are appended to the C4 file as MF154/MT2 data, which are then paired with the EMPIRE cross predictions for $\bar{\mu}$ for fitting with Kalman. An example of a set of $\bar{\mu}$ derived from ^{238}U elastic differential cross sections, with an EMPIRE calculation for $\bar{\mu}$ are shown in Fig. 3.4.

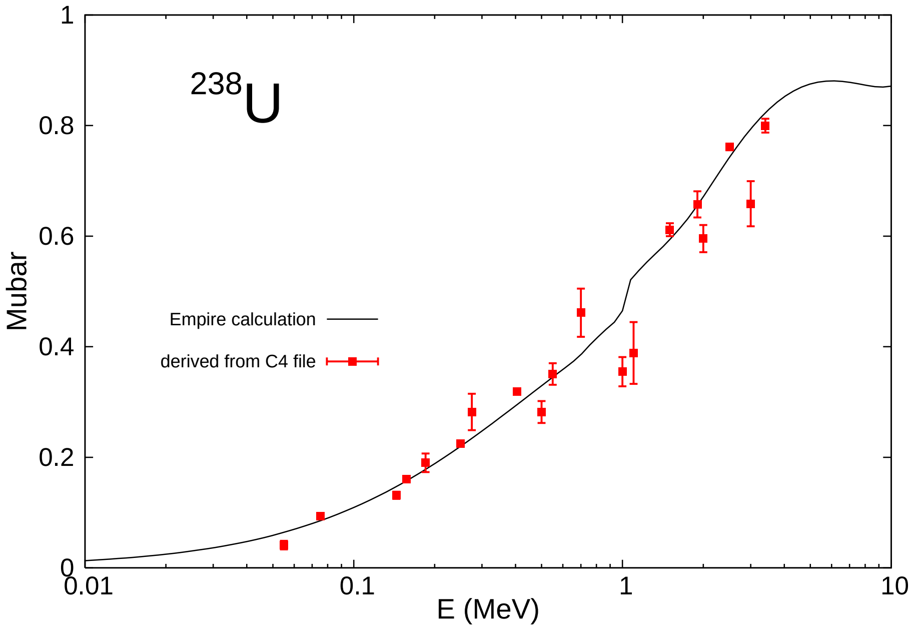


Figure 3.4: Values of $\bar{\mu}$ from fitted elastic differential cross sections for ^{238}U shown in red with uncertainties from the fit. The black curve is an EMPIRE calculation for $\bar{\mu}$. This curve is fit to the $\bar{\mu}$ data with the KALMAN code, which will adjust the EMPIRE parameters based on their calculated sensitivities for $\bar{\mu}$.

The addition of calculated $\bar{\mu}$ to the list of cross sections output by EMPIRE required a minor addition to the EMPIRE code. The differential cross sections are stored internal to EMPIRE in the center-of-mass frame from 0° to 180°, every 2°. As $\bar{\mu}$ is defined as the average cosine of the elastic scattered neutrons in the *Lab* frame, these differential cross sections are first boosted to Lab and then the distribution is integrated using Eq. 3.5 to calculate $\bar{\mu}$. This value is then written out to the *.xsc cross section file along with the other cross sections used for fitting with KALMAN. It will later be mapped to MF3/MT251, which is paired with $\bar{\mu}$ from the Legendre coefficients from fits to C4 data as MF154/MT2.

3.2 Nubar

The number of neutrons produced per fission, $\bar{\nu}$, can also be fit to data with the KALMAN code. Here, the the values for $\bar{\nu}$ to be fit are directly available in the C4 data file as MF1/MT452. These values need no further processing and are paired with the calculated $\bar{\nu}$ from EMPIRE by the c4tokal.f90 routine. Care must be taken with fitting $\bar{\nu}$ as EMPIRE does not calculate $\bar{\nu}$ but uses a distribution taken from an existing evaluated file, and scales it by a single parameter. Therefore, fitting $\bar{\nu}$should only be done when adjusting the scale parameter in EMPIRE to fit new $\bar{\nu}$ data not being used by empire itself for $\bar{\nu}$. In this case, the KALMAN code can be used. The only parameter that will be adjusted is the $\bar{\nu}$ scaling parameter PFNNUI. This can be fit along with other cross section data or not. The parameters for the cross sections do not affect $\bar{\nu}$, so there will be no correlations between the $\bar{\nu}$ scaling parameter PFNNUI and the other EMPIRE parameters. Of course, this single scaling parameter can also be set by hand in this case, by taking the ratio of the integral of the $\bar{\nu}$ spectrum used by EMPIRE to the integral of the $\bar{\nu}$ data being fit.

Section 4

Integral Assimilation with Empire

The fitting of EMPIRE predictions to experimental data is accomplished with the KALMAN fitting program where the sensitivities to various experimental observables are calculated for the EMPIRE parameters. These sensitivities are in effect the first derivative of the reactions at each energy with respect to the EMPIRE parameters being varied. The KALMAN code will then fit the varied EMPIRE parameters to experimental data based on these sensitivities. These fitted values, which in effect are linear extrapolations of the sensitivities) should be put back into EMPIRE and a new calculation run to check for the effects of non-linearities in the calculated cross sections. Usually it is best to repeat the cycle of determining the parameter sensitivities, fitting with KALMAN, then using the fitted parameters with EMPIRE until the fitted differences are small and/or within the limits of the uncertainties in the fitted parameters. Of course, this does not cover cases where Peelle's Pertinent Puzzle (PPP) [8] may arise, where the KALMAN code creates "fits" that are actually a worse fit, even using predictions based entirely on linear parameter sensitivities. In this case, manual intervention and modification of the EMPIRE parameters is usually required.

In addition to the fitting of cross section data (usually derived from the EXFOR library or from C4 file, see [9]) the EMPIRE predictions and parameter sensitivities can also be used to form sensitivities in integral experiments to the EMPIRE parameters. To accomplish this, one of two methods have been used. Below each is discussed.

4.1 Assimilation using processed cross sections in 33-group energy bins

The first method is to prepare sensitivity files similar to those prepared for KALMAN fitting which are used by collaborators at Argonne National Lab in simulations of integral experiments. This method usually involves creating ENDF-6 formatted files from the EMPIRE output files, checking these files for completeness and adding any required sections as needed, and then processing the ENDF files for the central values as well as for the "+" and "-" calculations, where each parameter is varied up and down by a specified amount to determine the sensitivity to the calculated cross sections for each parameter. Here, the output of each EMPIRE run is converted to ENDF-6 format and processed with the NJOY code [10] using the standard AFCI 33-group structure [11] to produce group-wise cross sections. The central values are also processed to produce an ACE cross section file for use with MCNP simulations [12]. These cross sections and ACE files are used to determine the sensitivity of the outcome of a particular integral experiment, either through Monte-Carlo or analytic means. The EMPIRE parameters are varied using the covariance matrix of the parameters as determined from a fit to differential data. The use of the covariances will modify the parameters in such a way as to minimize the effect of the change of the EMPIRE parameters to the differential cross sections. If the resulting cross sections as calculated by EMPIRE are in good agreement with the experimental differential data and the modifications to reproduce the integral experiment are not large, then this usually results in minor modifications to the differential data that are well within the uncertainties of the data, especially when systematic errors are taken into account. The result is an evaluation for the material where the cross sections provide a good description of the differential data and reproduce the results of a given integral experiment (or set of integral experiments) *without* requiring the cross sections to be adjusted. The goal would be to include as many integral experiments as possible which if consistent would result in a set of cross sections for the various important reactions that would not need modifications for every experiment - a global solution that would reproduce many differential and integral experiments.

The steps needed to prepare the files required for assimilation are as follows. The input files for Empire need to be prepared and fit to experimental differential data. This involves creating an input file, the sensitivity input file, as well as the various files needed to specify fission, etc. This should be iterated with the KALMAN code as needed to produce a good fit to the differential data. The final file of fitted EMPIRE parameters can be used to generate the files needed for assimilation. These files are produced by the python script *mkfils.py*. This file is a modified version of the script *sensitivity.py* which creates the sensitivities used during the fitting of differential data. The first step is exactly as when calculating sensitivities for KALMAN on the NNDC cluster. First, directories are setup for running EMPIRE varying each parameter up and down by the amount specified in the input sensitivity file. Once the directories are created jobs are started for each parameter varied at every energy in the input file. These jobs may take from a few hours to a few days to complete depending on the particular material being modeled. Once complete, the full output files are then reconstructed from the single-energy EMPIRE output files, and then these reconstructed files are processed with the NJOY code using the standard AFCI 33-group energy structure. The NJOY script calculates the group-wise cross sections using this energy grid at the specified temperature, usually 300K. A job is started for every NJOY computation, and when these are complete then the NJOY output files are scanned to produce set of cross sections for every EMPIRE parameter varied, producing cross sections for both the parameter varied up (+) and down (-). The central values are also processed, which is simply an EMPIRE run where the parameters are as fit to the differential data without any modifications. A job is also ran with a

Table 4.1: Summary of the files produced for an assimilation project. This set of files must be prepared for the integral testing.

File	Description
*_central_xsc.txt	Table of cross sections in 33-group energy structure. Processed using NJOY.
*-ENDFVII_xsd.txt	Cross sections for ENDF/B-VII.1
*-summary_minus.txt	Cross sections for each parameter varied down.
*-summary_plus.txt	Cross sections for each parameter varied up.
*-cov_matrix.dat	Covariance matrix of EMPIRE parameters from KALMAN fit to differential data.

reference ENDF file for this material, e.g., from the ENDF/B-VII.1 library [13]. The central value ENDF file must also be processed by the NJOY code to produce an ACE file of cross sections that will be used in the simulation of the integral experiment, and the covariance matrix of the EMPIRE parameters produced when fitting to experimental differential data is also supplied. In total, the six files for a given project must be supplied as listed in table 4.1.

4.2 Direct assimilation of integral experiments using modified ACE files

An alternative method was developed at BNL. It uses ENDF files produced for each "plus" and "minus" directly in a Monte-Carlo MCNP simulation of a particular integral experiment, which can then by used to directly determine the sensitivity to the result of an integral experiment (for example, k_{eff}) to the EMPIRE parameters. This method begins the same as the above method, running EMPIRE jobs for each parameter varied up and down by the amount specified in the sensitivity input file. Then the EMPIRE output file for each varied parameter is converted to an ENDF file. But then instead of processing the ENDF files to produce average cross sections, each file is processed with NJOY to produce an ACE file, which is then used in a Monte-Carlo simulation of an integral experiment, replacing the standard ACE file with the modified version. This is done for each parameter varied, for both the parameter varied "up" and "down." Then the sensitivity to the result of the simulation is calculated for each parameter, and the results of the sensitivities are then used in a Kalman fit, where there is usually only one experimental point, the result of the integral experiment (typically k_{eff}). Here, the sensitivity matrix is reduced to a simple vector of sensitivities to each EMPIRE parameter, and the integral experiment is then fit using the KALMAN code. In an attempt to modify the EMPIRE parameters in a way that preserves the agreement with the differential data, the output covariance matrix for the EMPIRE parameters from the differential KALMAN fit is used when fitting the integral data. The KALMAN code will then fit the integral experiment by varying the parameters in a way which minimizes the change in the differential measurements. The various input files for the KALMAN fitting code are prepared from the sensitivity input file and the results of the MCNP simulations for each varied parameter by the code kefkal.f90. It reads the sensitivity input file (*-inp.sen*) and then reads the MCNP output file for each EMPIRE parameter simulation, forming a sensitivity to each parameter. All the required input files are created, and upon successful completion the KALMAN code can be run and to produce a new fitted value for the result of the integral experiment, usually a value of 1.0 for k_{eff}. The resulting modified EMPIRE parameters should then be used in another cycle of EMPIRE to check for any non-linearities in the parameters.

Table 4.2: Results of direct assimilation of ^{239}Pu. EMPIRE parameters varied are listed with values before and after assimilation of integral experiment JEZABEL. Parameters which had the default value of 1.0 and were not varied during assimilation are not listed.

Parameter Name	pre-assimilation	post-assimilation
ATILNO-000	1.083	1.0851
ATILNO-001	0.907	0.9034
ATILNO-020	0.938	0.9380
ATILNO-030	0.988	0.9880
TUNEFI-010	0.833	0.8327
TUNE-000	2.228	2.2230
FUSRED-000	0.097	0.9700
RESNOR-000	1.320	1.3200
FISVF1-000	1.000	0.9995
FISVF1-010	1.000	1.0005
FISVF2-000	1.000	1.0042
FISVE1-000	1.000	0.9985
FISVE2-000	1.000	0.9995
FISHO1-000	1.000	0.9992
FISHO2-000	1.000	0.9992
FISAT1-000	0.917	0.9157
FISAT2-000	0.971	0.9717
FISAT2-010	0.981	0.9810
FISDL1-000	1.000	0.9999
FISDL2-000	1.000	0.9999
LDSHIF-000	1.100	1.0990
LDSHIF-010	1.063	1.0647
LDSHIF-020	0.917	1.9170
PFNALP-000	0.963	0.9613
PFNRAT-000	0.928	0.9279
PFNERE-000	0.999	1.0002
PFNTKE-000	0.984	0.9853

This method was tested using ^{239}Pu. For this material the integral experiment modeled was JEZEBEL, a solid sphere of ^{239}Pu [14]. The simulation also required ACE cross section tables for trace amounts of ^{240,241}Pu and Gallium, which were not varied and taken from

ENDF-B/VII.1. Using the ACE file for ^{239}Pu generated using the central values from the Kalman fit to differential data, a value of k_{eff} = 1.00516 ± 0.00008 was obtained. After fitting k_{eff} with the KALMAN code and then running EMPIRE with the fitted EMPIRE parameters (to account for any non-linearities) we obtained a final value of k_{eff} = 0.99959 ± 0.00008. As this value was within the quoted uncertainty for k_{eff} another iteration of assimilation was not required. The changes to the EMPIRE parameters were minor, as shown in Table 4.2, and well within the uncertainty of the parameters when fitted to differential data, and the resulting differences in the differential cross sections calculated by EMPIRE are small compared to the uncertainties in the differential cross sections, as shown for fission in Fig.4.1, where the changes in the EMPIRE calculations pre- and post-assimilation are shown with a small sample of available differential data. The change required for assimilation is very small in comparison to the uncertainties of the experimental data sets. Integral experiments serve as global constraints on the cross sections which can help guide the fitting of the empire parameters, and these constraints

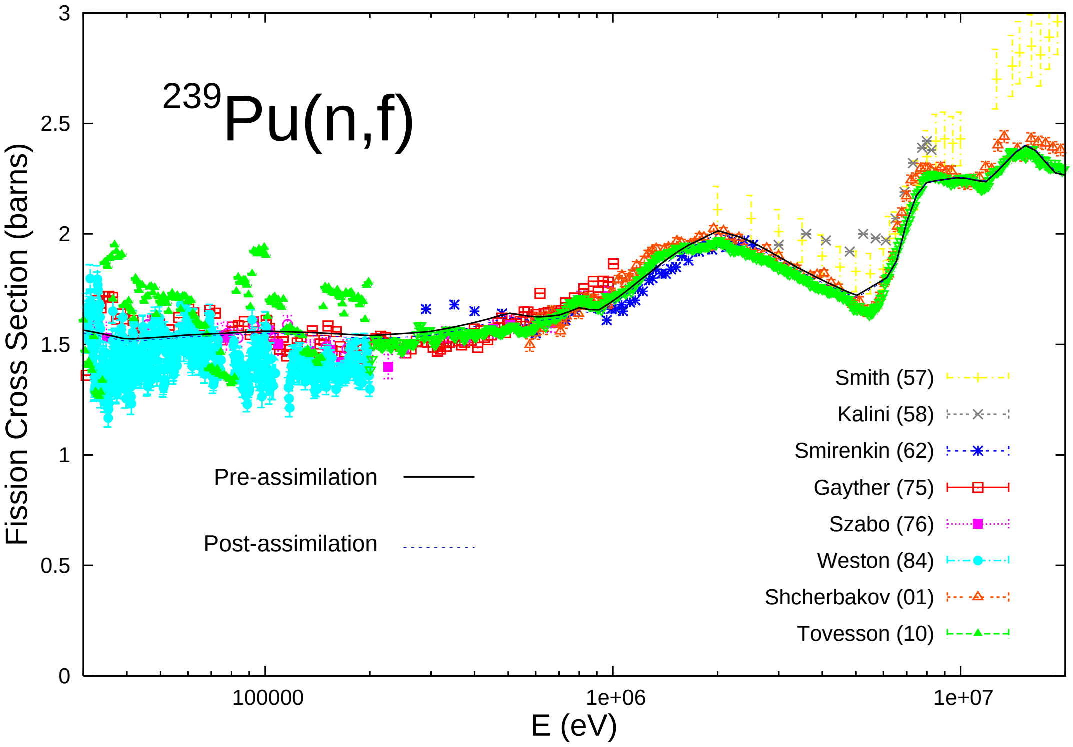


Figure 4.1: The pre-assimilation fit to differential fission data for ^{239}Pu data shown in solid black with the post-assimilation shown in dashed blue. A sample of the experimental data fitted with EMPIRE are shown for comparison. Note the small difference between the curves relative to the uncertainties and scatter of the differential data. These small differences from the pre- to post-assimilation cross sections are enough to bring k_{eff} into much better agreement with experiment.

are sensitive to minor modifications to the differential cross sections which are far smaller than the uncertainties currently available for most differential data sets. The fitting of integral experiments is an important additional tool to help determine a consistent set of values for the differential cross sections.

In summary, most of the preparation and creation of the sensitivities used for assimilation is managed by the python script *mkfils.py.* This script creates the various directories used when determining the sensitivity to a given EMPIRE parameter, submits various jobs to the NNDC cluster, and contains the results of the EMPIRE calculations with parameters varied around central values with the results stored in these directory trees, including NJOY calculations for both tables of cross sections and ACE files. A table of modifiers used with the *mkfils.py* script is given in table 4.2. The first task when creating the files for sensitivities is option -i and -r, where the various parameter directories are created and the EMPIRE jobs are submitted to the cluster. When the jobs have completed they can be then analyzed using the -a modifier, which uses the *runEmpire.py* script to collect the various single-energy runs into a single output file, which are then submitted to the cluster to create ENDF formatted files from the re-created EMPIRE output files. The -n modifier can then be used to produce tables of cross sections with NJOY using the standard 33-group energy structure, and the -z modifier to submit jobs that will create ACE files from the ENDF files for each varied parameter. Additionally, -k and -c are provided to clean up the directories, -k keeping the output files for the EMPIRE parameters, while -c removes the parameter subdirectories completely. Use the -c with caution.

Table 4.3: Summary of modifiers for assimilation using mkfils.py script.

Modifier	Description
-i	Initialize. Creates the directories for the various "plus" and "minus" empire runs.
-r	Run. Submit EMPIRE jobs for all the varied parameters to the NNDC cluster batch queue.
-a	Analyze. Re-assemble EMPIRE outputs from all the single-energy empire runs for each EMPIRE parameter directory. Then, submit a batch job that will create an ENDF file for each parameter directory. The making of the ENDF files is managed by the script *mkendf.sh.*
-n	Run NJOY jobs for each parameter directory that will process the ENDF file to produce weighted cross sections using the standard 33-group energy binning.
-z	Run NJOY jobs for each parameter directory that will create ACE files from the previously generated ENDF files.
-m	Run MCNP simulations for each directory using the ACE files created with the -z modifier. The MCNP input file should be in the main project directory and be named "mcnp.i". The non-varied ACE files used by the simulation should be placed in a subdirectory of the main project directory names xsec.
-k	Remove (clean) only the single-energy EMPIRE directories in all the EMPIRE parameter subdirectories. Also remove the log files from the single-energy EMPIRE runs. The summary output files in all the EMPIRE directories are not removed.
-c	Clean. Remove all the directories created by option -i. All EMPIRE parameter subdirectories are removed. This option will remove all results from the calculation of sensitivities for this project. Should be used with caution.

Conclusions

The EMPIRE code has been extended to enable assimilation of actinides. Presence of the fission channel requires that sensitivities and uncertainty estimates need to be provided for fission cross sections, prompt fission spectra (PFNS) and neutron multipicities ($\bar{\nu}$). The coding has been inserted that allows to vary extensive set of fission channel parameters and produce respective fission cross section sensitivities.

Two PFNS models (Los Alamos and Kornilov) have been implemented in the code and their performance tested. The energy dependence in the two models differs below 1 MeV emissive energy and it is expected that having both models included in the assimilation process might contribute to deciding which model performs better. The PFNS calculations have been coupled to the KALMAN code included in the EMPIRE system to provide possibility of fitting calculated PFNS to the experimental data and, at the same time, to estimate covariances for the PFNS model parameters as well as covariances for the prompt fission neutron spectra. To this end a module has been developed to process EXFOR library with experimental data, extract experiments that can be used for PFNS determination and compare them graphically with the EMPIRE calculations. To enhance visibility of potential differences this plotting can also be done after experimental and theoretical results were normalized to a Maxwellian at a given temperature.

Treatment of $\bar{\nu}$ is relatively simple - values read from the existing evaluation can be scaled preserving the original energy dependance. This first step should be sufficient at the current state of the assimilation methodology, although refinements are likely to be needed when the methodology becomes more mature.

Finally, EMPIRE has been extended to calculate the elastic angular distribution related quantity $\bar{\mu}$ along with its uncertainty. In order to compare these calculations with experimental data $\bar{\mu}$ specific processing of experimental data in the EXFOR library has been developed.

The above mentioned extensions put the code in a position of fulfilling all functions that are necessary for performing assimilation on actinides. In particular, fission cross sections, PFNS spectra and $\bar{\nu}$ can be adjusted simultaneously on the level of model parameters. As a proof of principle, assimilation of ^{239}Pu has been performed by adjusting full set of model parameters to the k_{eff} of the JEZEBEL integral experiment. It has been shown that minor adjustments are sufficient to bring MCNP calculated k_{eff} into agreement with the measured value preserving reaction theory constraints and maintaining agreement with the differential data. This relatively simple assimilation (only single integral experiment) shows that our reaction modeling is flexible enough to provide a physically consistent set of nuclear data that matches precision required by applications.

Bibliography

[1] P. Talou, B. Becker, T. Kawano, M. B. Chadwick, and Y. Danon. Advanced Monte Carlo modeling of prompt fission neutrons for thermal and fast neutron-induced fission reactions on ^{239}Pu. *Phys. Rev. C*, 83:064612, 2011.

[2] D. G. Madland and J. R. Nix. New calculation of prompt fission neutron-spectra and average prompt neutron multiplicities. *Nucl. Sci. and Eng.*, 81:213–271, 1982.

[3] V. Weisskopf. Statistics and nuclear reactions. *Phys. Rev.*, 52:295–303, Aug 1937.

[4] N. V. Kornilov, A. B. Kagalenko, and F.-J. Hambsch. Computing the spectra of prompt fission neutrons on the basis of a new systematics of experimental data. *Physics of Atomic Nuclei*, 62:173, 1999.

[5] V. V. Malinovskii et al. INDC(CCP)-277. Technical report, IAEA, 1987.

[6] M. Herman, M. T. Pigni, P. Obložinský, S. F. Mughabghab, C. M. Mattoon, R. Capote, Y. S. Cho, and A. Trkov. Development of Covariance Capabilities in EMPIRE Code. *Nuclear Data Sheets*, 109:2752–2761, 2008.

[7] G. D. Agostini. On the use of the covariance matrix to fit correlated data. *Nuclear Instruments and Methods*, A 346:306–311, 1994.

[8] S. Chiba and D. L. Smith. A suggested procedure for resolving an anomaly in least-squares data analysis known as Peelle's pertinent puzzle and the general implications for nuclear data evaluation. Technical Report ANL/NDM-121, Argonne National Lab, Argonne, Illinois, 1991.

[9] NRDC-Network. CSISRS/EXFOR library of experimental cross sections. www.nndc.bnl.gov/exfor, 2006.

[10] R.E. MacFarlane. NJOY nuclear data processing code, version 2006. Nuclear Data Sheets, to be published.

[11] M. W. Herman, P. Obložinský, C. M. Mattoon, M. Pigni, S. Hoblit, S.F. Mughabghab, A. A. Sonzogni, P. Talou, M. B. Chadwick, G. M. Hale, A. C. Kahler, T. Kawano, R. C. Little, and P. G. Young. COMMARA-2.0 Neutron Cross Section Covariance Library. Technical Report BNL-94380-2011, Brookhaven National Lab, 2011.

[12] R. A. Forster, L. J. Cox, R. F. Barrett, T. E. Booth, J. F. Briesmeister, F. B. Brown, J. S. Bull, G. C. Geisler, J. T. Goorley, R. D. Mosteller, S. E. Post, R. E. Prael, E. C. Selcow, and A. Sood. MCNP^{tm} Version 5. *Nuclear Instruments and Methods*, B 213:403, 2004.

[13] M. B. Chadwick, M. Herman, P. Obložinský, M. E. Dunn, Y. Danon, A. C. Kahler, D. L. Smith, B. Pritychenko, G. Arbanas, R. Arcilla, R. Brewer, D. A. Brown, R. Capote, A. D. Carlson, Y. S. Cho, H. Derrien, K. Guber, G. M. Hale, S. Hoblit, S. Holloway, T. D. Johnson, T. Kawano, B. C. Kiedrowski, H. Kim, S. Kunieda, N. M. Larson, L. Leal, J. P. Lestone, R. C. Little, E. A. McCutchan, R. E. MacFarlane, M. MacInnes, C. M. Mattoon, R. D. McKnight, S. F. Mughabghab, G. P. A. Nobre, G. Palmiotti, A. Palumbo, M. T. Pigni, V. G. Pronyaev, R. O. Sayer, A. A. Sonzogni, N. C. Summers, P. Talou, I. J. Thompson, A. Trkov, R. L. Vogt, S. C. van der Marck, A. Wallner, M. C. White, D. Wiarda, and P. G. Young. ENDF/B-VII.1 Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay Data. *Nuclear Data Sheets*, 112(12):2887–2996, 12 2011.

[14] R. D. Mosteller. Validation suite for mcnp. In *12th Biennial RPSD Topical Meeting*, Sante Fe, NM, April 2002.