A new software implementation of the Oslo method with complete uncertainty propagation

J. E. Midtbø*, F. Zeiser, E. Lima, A. C. Larsen, G. M. Tveten, T. Renstrom, M. Guttormsen and F. L. Bello Garrote

Department of Physics, University of Oslo, N-0316 Oslo, Norway

Abstract

The Oslo method comprises a set of analysis techniques designed to extract nuclear level density and average γ-decay strength from a set of excitation-energy tagged γ-ray spectra. Here we present a new software implementation of the entire Oslo method in Python, called Ompy. In addition to the functionality of the original analysis code, it includes novel components such as a method to propagate uncertainties throughout the various steps of the Oslo method using a Monte Carlo approach. The error propagation is applied to data from different mass regions, consistently reproducing previous results and refining the quantification of uncertainties.

1. Introduction

One long-standing challenge in nuclear physics is to precisely determine nuclear properties at excitation energies above the discrete region and up to the particle threshold(s). This region, often referred to as the quasi-continuum, represents an excitation-energy region where the quantum levels are very closely spaced, leading to a significant degree of mixing (complexity) of their wave functions, but they are still not fully overlapping as in the continuum region. For the quasicontinuum, it has proven fruitful to introduce average quantities to describe the excited nucleus: instead of specific levels, the level density as a function of excitation energy is used, and instead of specific reduced transitions strengths $B(XL)$ between a given initial and final state ($B(E1), B(M1), B(E2),...$), the average decay strength represented by the γ-ray strength function ($\gamma$SF) is applied.

In addition to their key role in describing fundamental nuclear properties, both the level density and the γSF are vital components for calculating cross sections and reaction rates for applications within nuclear astrophysics (nucleosynthesis) and for the design of next-generation nuclear power plants [1, 2, 3].

The Oslo method [4] allows for extracting the level density and the γSF simultaneously from a data set of particle-γ ray coincidences, and has been successfully applied to a range of nuclei of widely differing mass [5, 6, 7, 8, 9]. However, the Oslo method consists of several, highly non-linear steps. This makes an analytical propagation of statistical and systematic uncertainties very difficult, and thus hampers a reliable uncertainty quantification for the final results. Systematic uncertainties related to the absolute normalization of the level density and γSF have been discussed in Ref. [10], but uncertainty propagation from unfolding the γ spectra and the determination of the primary γ-ray distribution has so far not been addressed in a fully rigorous way. In lieu of this, an approximate uncertainty estimation has been used, which is described in Ref. [1].

In this work, we approach the problem of uncertainty propagation using a numerical technique. By generating an ensemble of perturbed input spectra, distributed according to the experimental uncertainties, and propagating each ensemble member through the Oslo method, we can gauge the impact of the method on the uncertainties. In the following, we discuss the various steps in the Oslo method and present our new uncertainty propagation method. The capability of the new method is illustrated by applying it to experimental data.
2. The Oslo method

The starting point for the Oslo method is an \( E_x-E_y \) coincidence matrix, i.e., a set of \( \gamma \)-ray spectra each stemming from an identified initial excitation energy \( E_i \). The standard version of the Oslo method constructs this matrix from coincidence measurements of \( \gamma \) rays and charged ejectiles following inelastic scattering or transfer reactions. An array of \( \gamma \) detectors measures the energy of the \( \gamma \) rays emitted, while a particle telescope determines the excitation energy from the outgoing charged rays emitted, while a particle telescope determines an array of \( \gamma \) detectors. (For a detailed description, see e.g. [10] and references therein.) In the \( \beta \)-Oslo method [11], \( E_i \) is instead determined from the sum of \( \gamma \) rays measured with a segmented total absorption spectrometer, while the associated energy matrix from a standard Oslo method experiment at the Oslo Cyclotron Laboratory is shown in panel a of Fig. 1.

The first step of the Oslo method is to unfold, i.e., deconvolute the \( \gamma \)-ray spectra for each excitation energy to compensate for the detector response (Compton scattering, \( e^+e^- \) production, etc.). This is done using the unfolding method described in Ref. [13]. We reiterate the main points of the procedure in Appendix A. The unfolded \( ^{164}\text{Dy} \) spectrum is shown in panel a of Fig. 1.

The second step is the determination of the first-generation, or primary, \( \gamma \)-ray spectrum for each excitation energy. Here, an iterative procedure is applied as described in Ref. [15]. We recapitulate some of the main points of the procedure in Appendix B. The resulting first-generation \( \gamma \)-ray matrix is shown in panel c of Fig. 1.

The final step of the Oslo method consists of fitting the first-generation spectra to a product of two one-dimensional functions, namely the nuclear level density \( \rho(E_x) \) and the \( \gamma \)-ray transmission coefficient \( T(E_i) \). The method relies on the relation

\[
P(E_x, E_i) \propto \rho(E_x - E_i)T(E_i),
\]

where \( P(E_x, E_i) \) is the first-generation spectrum \( FG(E_i)E_x \) normalized to unity for each \( E_x \) bin. Furthermore, if we assume that the \( \gamma \) decay at high \( E_x \) is dominated by dipole radiation, as strongly supported by data [16, 17, 18, 19, 20], the transmission coefficient is related to the dipole \( \gamma \)-ray strength function \( f(E_i) \) by the relation

\[
T(E_i) = 2\pi E_i^2 f(E_i).
\]

The assumptions underpinning the decomposition of Eq. (1) are:

- The compound nucleus picture: We assume that the \( \gamma \)-decay from the excited nuclear state is independent of how the excited state was formed. This goes back to Bohr’s theory for compound nuclei [21] and is supported by many experiments [22, 23, 24, 25, 26, 27, 6, 28, 16].
- Fermi’s golden rule: The probability of decay is, to first order in perturbation theory, proportional to the product of the density of final states \( \rho \) and the transition matrix elements \( |\langle f|\hat{O}|i\rangle|^2 \), which is again proportional to the \( \gamma \)-ray strength function \( f(E_i) \) [29, 30].
- The generalized Brink-Axel hypothesis: The gamma-ray strength function \( f(E_i) \) is independent of the initial and final states, i.e., it is the same for excitations and decays between any initial and final state that are separated by the energy \( E_i \), [31, 32, 33, 34, 35, 36].

The next step is to make the fit of \( P(E_x, E_i)T(E_i) \) to data. First, we select a suitable bin size \( \Delta E \), typically \( 100 \sim 300 \text{ keV} \) depending on the statistics, and rebin the first-generation matrix along both the \( E_x \) and \( E_i \) axes to this bin size. We obtain the matrix of experimental decay probabilities, \( P_{\text{exp}}(E_x, E_i) \), by normalizing the spectrum in each \( E_x \) bin to unity. For the fit of \( \rho \) and \( T \), we take the function value in each bin as a free parameter. For a given pair of trial functions \( (\rho, T) \), we construct the corresponding matrix \( P_{\text{fit}}(E_x, E_i) \) by

\[
P_{\text{fit}}(E_x, E_i) = N_{E_x}\rho(E_x - E_i)T(E_i),
\]

where \( N_{E_x} \) is a normalization coefficient so that \( \sum_{E_x} P_{\text{fit}}(E_x, E_i) = 1 \forall E_x \). We fit \( P_{\text{fit}} \) by a \( \chi^2 \) minimization approach, minimizing the weighted sum-of-squared errors

\[
\chi^2 = \sum_{E_x, E_i} \left( \frac{P_{\text{exp}}(E_x, E_i) - P_{\text{fit}}(E_x, E_i)}{\sigma_{T_{\text{exp}}}(E_x, E_i)} \right)^2.
\]
Figure 1: Raw (a), unfolded (b) and first-generation (c) matrices for the $^{164}$Dy dataset [13,6], as well as the respective standard deviation matrices (d, e and f) obtained with the ensemble propagation technique. The counts to the right of the $E_x = E_\gamma$ diagonal in panel a have been removed before the unfolding method is applied. See text for details.
Using a $\chi^2$ minimization is only strictly justified if the data being fit have a normal distribution. However, for sufficiently large values of the Poisson expectation value $\lambda$ (see below), the normal distribution is a good approximation to the Poisson distribution, hence the $\chi^2$ approach is justified.

It is important to use a weighted sum rather than simply a sum of the residuals, to suppress the influence of bins with large uncertainties. This in turn makes uncertainty estimation important. As already mentioned, a shortcoming of the original Oslo method implementation has been the estimation of the uncertainty $\sigma_{\text{exp}}(E_x, E_\gamma)$ in the denominator of the $\chi^2$ fit. Due to the lack of a complete statistical uncertainty propagation, one has had to resort to an approximate uncertainty estimation based on a Monte Carlo scheme similar to the present work, but where only the first-generation spectrum is perturbed. This is discussed in detail in Ref. \[4\]. In \texttt{OMpy}, we have access to a proper uncertainty matrix $\sigma_{\text{exp}}$. This, as well as the technicalities of the fitting procedure, will be discussed in more detail in the next section.

3. Uncertainty propagation by ensemble

We use an approach based on the Monte Carlo (MC) technique to estimate the statistical uncertainties in the Oslo method by an ensemble of randomly perturbed copies of the data set under study. To illustrate the method, we have chosen three experimental data sets, $^{163, 164}\text{Dy}$ and $^{56}\text{Fe}$. These data sets are from high statistics experiments, enabling us to focus on the uncertainties inherent in the Oslo method. They allow a comparison of the method’s performance on even and odd nuclei, as well as between high- and low-mass nuclei. Our results will be compared to the original analyses, published in Refs. \[8\] and \[10\], respectively.

The random variables are the experimental number of counts in each energy bin $i$ in the raw $E_x-E_\gamma$ coincidence matrix $R$. We assume that they are independent and follow a Poisson distribution with parameter $\lambda_i$. The Poisson distribution $P_\lambda$ is given as

$$P_\lambda \sim p(k|\lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (5)$$

We take the number of counts $k_i$ in bin $i$ of $R$ as an estimate for the Poisson parameter $\lambda_i$. Note that it is an unbiased estimator for $\lambda_i$, since the expectation value $\langle k \rangle = \lambda$. To generate a member matrix $R_l$ of the MC ensemble, we replace the counts in each bin $i$ by a random draw from the distribution $P_{\lambda_i}$. By this procedure, we obtain $N_{\text{ens}}$ matrices representing different realizations of the experiment. Defining $\vec{r}_l$ as the vector of all $N_{\text{ens}}$ realizations $l$ of bin $i$, we can calculate the sample standard deviation by

$$\sigma_i = \sqrt{\frac{1}{N_{\text{ens}}} \sum_{l=1}^{N_{\text{ens}}} (r_{li} - \langle \vec{r} \rangle)^2}. \quad (6)$$

Of course, in the case of the raw matrix $R$, the standard deviation is trivial because it is given by the Poisson distribution ($\sigma = \sqrt{\lambda}$). But the technique also allows us to estimate the standard deviation at later stages in the Oslo method — after unfolding, after the first-generation method and even after fitting the level density and $\gamma$-ray transmission coefficient. In Fig. \[1\] we show the standard deviations in the raw (d), unfolded (e) and first-generation (f) matrices of the $^{164}\text{Dy}$ dataset based on $N_{\text{ens}} = 100$ ensemble members.

With the first-generation matrix and its corresponding uncertainties at hand, we may proceed with the fitting of $\rho$ and $T$. As discussed in the previous section, we obtain the experimental probability matrix $P_{\text{exp}}(E_x, E_\gamma)$ by normalizing the first-generation matrix to unity in each $E_x$ bin. Similarly, we obtain the standard deviation matrix $\sigma_{P_{\text{exp}}}$ by normalizing the matrix of first-generation standard deviations with the same factor as the first-generation matrix itself. We use the Python package \texttt{uncertainties} to take proper care of correlations when normalizing \[37\]. We then carry out the $\chi^2$ minimization by numerical minimization. Note that this is different from the original Oslo method implementation, where the minimum is found by iteratively solving a set of equations to obtain a solution satisfying $\partial \chi^2/\partial \rho = 0, \partial \chi^2/\partial T = 0$ for each bin of $\rho$ and $T$ \[4\]. After testing several off-the-shelf minimizers, we have found that the modified Powell’s method in the \texttt{SciPy} package works well \[38, 39\]. It is a gradient-free method that is good for noisy data. In Fig. \[2\] we show the resulting fit to the $^{164}\text{Dy}$ dataset. The $\gamma$-ray strength function shown in panel b is obtained from $T$ using Eq. \[2\].

By a first glance, the curves bear little resemblance to a level density or $\gamma$SF. That is because the fit has not yet been normalized. In Ref. \[4\], it is shown that the $\chi^2$ fit is invariant under a Lie group $G$
of transformations by three continuous parameters $A, B$ and $\alpha$:

$$\rho(E_x), f(E_\gamma) \rightarrow A e^{\alpha E_x} \rho(E_x), B e^{\alpha E_\gamma} f(E_\gamma).$$

Thus, the solution shown in Fig. 2 is just one of an infinite set of solutions to the fit. However, we stress that the degrees of freedom are limited to those given by $G$—i.e., the relative values between adjacent data points are uniquely determined by the fit. To obtain the physical solution, the level density and $\gamma$SF need to be normalized to auxiliary data. Typically, one uses $s$-wave resonance spacings, $D_0$, from neutron capture experiments as well as discrete levels to fix the level density normalization, and augment this by average total radiative width data, $\langle \Gamma_\gamma \rangle$, to normalize the $\gamma$SF. It is important to note that the $\alpha$ parameter, which influences the slope (in a log plot) of the functions, is common to $\rho$ and $f$. Hence, their normalizations are coupled together in the Oslo method.

### 3.1. Uncertainty estimation of the fit parameters

We estimate the uncertainty in $\rho$ and $f$ by performing the fit separately to each member of the ensemble of first-generation matrices. We use the same $\sigma_{P_{\text{exp}}}$ in all the $\chi^2$ fits. From this ensemble of fits, we obtain the standard deviation in each data point. In Fig. 3 we show the resulting fits for the $^{164}$Dy data set. These have now had their $G$ parameters adjusted to match the original analysis of Ref. [6]. It is gratifying to see that overall, the analyses match each other closely. In Fig. 3, the level density below $\sim 2$ MeV exhibits the same structure of

---

**Figure 2:** Fit of $\rho(E_x)$ and $f(E_\gamma) = T(E_\gamma)/(2\pi E_\gamma^3)$ to the $^{164}$Dy primary matrix. No transformation has been applied to the fit. See text for details.

**Figure 3:** Fit of level density and gamma-ray strength function to the primary spectrum of $^{164}$Dy. The fit is the same as that shown in Fig. 2 but a transformation according to Eq. (7) has been applied. The transformation parameters were chosen by eye to match the normalized functions from the original work of Ref. [6], which are also shown.
bumps attributable to the discrete level structure, and at higher $E_x$ the curves are practically identical. Similarly for the $\gamma$SF in Fig. 3, the data points follow each other closely, especially in the region between 2 and 6 MeV. A difference to note is that while the data points in the original analysis tend to be quite aligned, even with rather large error bars (such as for $f$ in the region between 6 and 7 MeV), the neighbouring data points in the new fit show a larger scatter within their statistical uncertainties. This is a sign that the uncertainties are of the right order and are correctly represented. We note that the uncertainties estimated by OMpy are generally somewhat larger than in the original analysis.

Having established that OMpy works, we apply it to the other data sets as well. In Figs. 4 and 5, we show the results of fits to the $^{163}$Dy and $^{56}$Fe data sets, respectively. The $^{163}$Dy fit has very similar features as $^{164}$Dy. Again, we observe the best-fit values jumping inside their uncertainties between neighbouring data points, which indicates that the uncertainties are of the right order. Turning to the $^{56}$Fe fit, the story is different. There are striking differences between the uncertainties in the original analysis and the ones from OMpy. This is especially notable for the low-energy ($<2$ MeV) part of the level density. We believe that this is a signature of the difference between an uncertainty estimation based on perturbations of the primary spectrum only, and a complete uncertainty propagation through unfolding and the first-generation method. In the latter case, the primary matrices will pick up more variation between ensemble members. For regions of the fit where the number of primary counts is fluctuating strongly, as in the low-$E_x$ region of $^{56}$Fe, the counts may be shifted between neighbouring channels in the ensemble, resulting in significant differences in the fits to single data points. These fluctuations should be more significant in a low-mass nucleus like $^{56}$Fe, since it has larger spacings in its low-lying level structure.

For Fig. 5, we chose to represent the error bars as percentiles rather than standard deviations, to avoid the largest outliers. The lower and upper error bars are taken as the 15.9th and 84.1th percentile, respectively, corresponding to $\pm1\sigma$ uncertainty on a Gaussian distribution. However, the MC ensemble approach also opens for other useful ways to visualize the uncertainty in the data. In Fig. 4 we have plotted the fits of $\rho$ and $f$ for each member of the ensemble using partially transparent curves. We have also drawn the mean value of all the fits as a separate curve, representing the best estimate at each data point. The plots illustrate how the large error bars are caused by large variability in the fit value of some bins between ensemble members. It should be pointed out that, since the plots are in logarithmic scale, the error bars are visually inflated at low function values (so it arguably looks worse than it is).

A subtle point regarding the uncertainty estimation requires mention. Even though $\chi^2$ is invariant, and the solutions thus degenerate, with respect to transformations under $G$, we see empirically that the fit for a given data set always converges to the same $\rho$ and $f$, i.e., the same $A$, $B$ and $\alpha$ normalization parameter set. We assume that this is also true when the data set is slightly perturbed. However,
although we observe that they are very similar, it cannot be guaranteed that the ensemble of $\rho$ and $f$ fits that we use to determine the uncertainty in the fitted data points have exactly the same intrinsic normalization. To stabilise the solutions against the $G$ degeneracy, we use the fitted $\rho_0$, $T_0$ values of the unperturbed primary matrix as input parameters to the ensemble of fits for the uncertainty spread.

It would be worthwhile to search for other ways to quantify the $\rho$ and $f$ uncertainties that avoid this potential problem. It is however not trivial, since the only way to know the true normalization is to use auxiliary data. However, if all the ensemble members are normalized separately to auxiliary data, the intrinsic variation between the ensemble members gets mixed with the variation due to the individually adjusted normalization parameters for each member. One conceivable way to avoid the problem is to perform a global fit to both the primary matrix and auxiliary data at the same time. This is however a technical challenge, and probably has to be done in a Bayesian setting with prior probabilities imposed on the parameters.

3.2. The importance of unfolding

An important source of uncertainty in the Oslo method comes from the unfolding of the detector response in the $\gamma$-ray spectra. To illustrate the effect of an erroneous unfolding in an extreme case, we have investigated what happens if the unfolding is neglected altogether. This means that we go
straight from the raw matrix to the first-generation method. A comparison of fitted level densities and γ-ray strength functions to 164Dy primary γ matrices with and without unfolding is shown in Fig. 7. For this fit, we have not used the uncertainty propagation. Instead, we estimate the uncertainty matrix of the primary spectrum as $\sigma_{\text{exp}} = \sqrt{P_{\text{exp}}}$. It is instructive to see what happens: The level density is almost unchanged, while the γSF is very different. Keeping in mind that the absolute value of the γSF is a free parameter, a direct comparison of the two may not be correct. However, since the level density is nearly identical, it means that the slope parameter $\alpha$ is common between the two γSF’s. There are clear, qualitative differences between the two functions, most notably in the low-energy region below about 4 MeV. What appears in the unfolded γSF as a pronounced maximum around 3 MeV is washed out in the not-unfolded γSF by a very large enhancement that grows in and becomes very steep below about 1.5 MeV. This underlines the importance of having good control of the uncertainties from unfolding, as studies of structures in the low-energy region of the γSF is one of the most widely-used applications of the Oslo method.

It is possible to utilize $\text{OMpy}$’s error propagation functionality to estimate the systematic uncertainty due to unfolding. There are two main sources of systematic uncertainty: The iterative unfolding and Compton subtraction method itself, and the model of the detector response functions. The latter can be gauged by simulating an ensemble of different detector response functions that capture the breadth of physically reasonable response models, and applying the error propagation technique to them. The modelling itself requires use of auxiliary software such as $\text{GEANT4}$ [10, 11, 12]. The former uncertainty is more difficult to quantify. One avenue of approach would be to implement alternative unfolding algorithms (e.g. Refs. [13, 14]) and compare the results. This is outside the scope of the present work.

4. Conclusions and outlook

We have presented $\text{OMpy}$, a complete reimplementation of the Oslo method in Python. We have demonstrated its ability to perform Oslo method analyses, confirming previous results and refining error quantifications.

We have discussed the importance of a correct detector response unfolding, especially for the γSF. $\text{OMpy}$ opens the possibility for a thorough investigation of the systematic uncertainties related to the response functions. This will however require an effort to produce an ensemble of response functions using auxiliary software, and will be pursued in a subsequent work.

One of the reasons for writing $\text{OMpy}$ in Python is, in addition to the transparency of the code, the ease with which it can be modified and coupled to the vast amount of other Python packages available [38, 45]. As an example, it may be of interest to try other unfolding methods than the one currently implemented, and this can be achieved with minimal modifications (see Appendix D). It is also possible to use $\text{OMpy}$’s functions as inputs to other Python packages, e.g. to perform fits to Oslo-type data and auxiliary data sources simultaneously.
Appendix A. Unfolding

Here we explain the unfolding technique presented in Ref. [14], which is used both in the original Oslo method implementation and in OMpy. Let the detector response be modelled as a conditional probability distribution

$$p(E_\gamma | E'_\gamma), \quad (A.1)$$

encoding the probability that a \( \gamma \) ray with true energy \( E_\gamma \) is detected with energy \( E'_\gamma \). Given a true \( \gamma \)-ray spectrum \( U(E_\gamma) \), the folded spectrum \( F(E'_\gamma) \), i.e. the spectrum seen by the detector, is then given by

$$F(E_\gamma) = \int p(E_\gamma | E'_\gamma) U(E'_\gamma) \, dE'_\gamma. \quad (A.2)$$

By discretising into energy bins of width \( \Delta E_\gamma \), it becomes a matrix equation

$$\vec{F} = \vec{P} \vec{U}, \quad (A.3)$$

where \( \vec{P} \) is the response matrix of discrete probabilities \( p(E_\gamma, k|E'_\gamma, l) \Delta E_\gamma \). The unfolding procedure amounts to inverting this equation, to obtain \( \vec{U} \) from \( \vec{F} \). However, a straightforward matrix inversion is ill-advised, as it will produce large, artificial fluctuations in \( \vec{U} \). Instead, the approach taken in the Oslo method is to use an iterative technique that successively approximates \( \vec{U} \). Letting \( \vec{R} \) denote the measured spectrum, the algorithm is

1. Start with a trial function \( \vec{U}_0 = \vec{R} \) at iteration \( i = 0 \)
2. Calculate the folded spectrum \( \vec{F}_i = \vec{P} \vec{U}_i \)
3. Update the trial function to \( \vec{U}_{i+1} = \vec{U}_i + (\vec{R} - \vec{F}_i) \)
4. Iterate from 2 until \( \vec{F}_i \approx \vec{R} \). The criterion for terminating the iterations is taken as a weighted sum of the root-mean-square error of \( \vec{F}_i - \vec{R} \) and the level of fluctuations in \( \vec{U}_i \).

The fluctuations are estimated as \( \sum |U_{i, k} - \bar{U}_i| \), where \( \bar{U}_i \) is a smoothed version of the spectrum \( U_i \).

In addition to this, Ref. [14] presents a further refinement to the unfolding method known as Compton subtraction. It is used to further control the fluctuations in the unfolded spectrum. The basic concept behind it is to use the previously unfolded spectrum to decompose \( \vec{R} \) into parts corresponding to the full-energy, single and double escape and annihilation peaks, and the “rest” which comes from Compton scattering and similar processes. Each of these parts, save for the full-energy peak, are then smoothed with the detector resolution before they are subtracted from \( \vec{R} \). The resulting spectrum is then multiplied up to maintain the number of counts. The idea is that this gives an unfolded spectrum with the same statistical fluctuations as in the original spectrum \( \vec{R} \).

Appendix B. The first-generation method

Let \( FG(E_x)E_z \) denote the first-generation \( \gamma \)-ray spectrum, i.e., the intensity distribution of \( \gamma \)-ray decay from a given excitation energy \( E_x \), as function of \( \gamma \)-ray energy \( E_\gamma \). Generally, the nucleus will decay from \( E_x \) down to the ground state by emitting a cascade of \( \gamma \) rays, which forms the total \( \gamma \)-ray spectrum. The total, or all-generations \( \gamma \)-ray spectrum, denoted \( AG(E_x)E_z \), can be viewed as a superposition of the first-generation spectrum and a weighted sum of the all-generations spectra of excitation energies below,

$$AG(E_x)E_z = FG(E_x)E_z + \sum_{E_z' < E_z} n(E_z')E_z N(E_x)AG(E_z')E_z'. \quad (B.1)$$

Here, \( n(E_z')E_z \) is a normalization factor which corrects for the varying cross section to populate the \( E_z' \) bins, and \( w(E_z')E_z \) is a weight factor. The normalization factor can be estimated from the total \( \gamma \)-ray spectrum by the relation

$$n(E_z')E_z = \frac{M(E_z')N(E_x)}{M(E_x)N(E_z')}, \quad (B.2)$$

where \( M(E_x) \) and \( N(E_x) \) denote the average \( \gamma \)-ray multiplicity and the total number of counts, respectively, at excitation energy \( E_x \). The average multiplicity can again be estimated from the spectrum by the relation

$$M(E_x) = \frac{E_x}{\langle E_\gamma \rangle}. \quad (B.3)$$

where \( \langle E_\gamma \rangle \) is the weighted-average \( \gamma \)-ray energy at excitation energy \( E_x \). The weight function \( w(E_z') \) encodes the probability for the nucleus to decay from \( E_x \) to \( E_z' \), and is in fact nothing but the normalized first-generation spectrum for \( E_x \),

$$w(E_z')E_x = \frac{FG(E_x - E_z')E_x}{\sum_{E_z'} FG(E_x)E_x}. \quad (B.4)$$
By rewriting Eq. (B.2), we obtain

$$FG(E_\gamma)_{E_x} = AG(E_\gamma)_{E_x} - \sum_{E'_x < E_x} n(E'_x)_{E_x} \frac{FG(E_x - E'_x)_{E_x}}{\sum_{E'_x} FG(E'_x)_{E_x}} AG(E_\gamma)_{E_x}$$ \tag{B.5}$$

This is a self-consistent set of equations for the $FG$ spectra, which we solve by an iterative procedure, starting with a set of trial functions $FG(E_\gamma)_{E_x}$ and iterating until convergence is reached. In OpenPy, the trial functions are chosen as constant functions, \textit{i.e.} with the same value for all $E_\gamma$. In the original implementation of the first-generation method, the trial functions are instead chosen based on a Fermi gas level density model \cite{46}. We have checked with OpenPy that this gives identical results as with constant functions.

### Appendix C. Derivation of the Oslo method equation

Here, we derive the relationship between the distribution of primary $\gamma$ rays, and the strength function and level density. Consider Fermi’s golden rule \cite{29, 30}, which says that the probability of decay from a specific initial state $i$ into a quasi-continuum of final states $f$ is given to first order in perturbation theory as

$$\omega_{i \to f} = \frac{2\epsilon_0 N\gamma}{\hbar^2} \frac{(E_{\gamma})^{2\lambda+1}}{(2\lambda + 1)!!} B(X L; i \to f) \rho_{\text{avail}}(f),$$ \tag{C.1}$$

where $\rho_{\text{avail}}(f)$ is the density of available final states $f$ and $B(\sigma \lambda; i \to f)$ is the reduced transition probability.\footnote{This follows from the golden rule as shown in \textit{e.g.} appendix B of Ref. \cite{27}. The final state density has been modified since the decay goes into a quasi-continuum of nuclear levels.} The $\gamma$-ray strength function for a given multipolarity $X L$ and for nuclear states with a given excitation energy $E_x$, spin $J$ and parity $\pi$ is defined as \cite{13}

$$f_{XL}(E_{\gamma}, E_x, J, \pi) = \frac{(\Gamma_{\gamma, XL}(E_{\gamma}, E_x)\rho(E_x, J, \pi)}{E_x^{2\sigma+1}} \tag{C.2}$$

where $\langle \cdot \cdot \cdot \rangle$ denotes an average over individual transitions in the vicinity of $E_x$, $E_x$ (in practice defined by the energy binning resolution), $\rho_{J, \pi}(E_x)$ is the density of spin-$J$, parity-$\pi$ levels at energy $E_x$ and $\sigma_{XL}$ is a constant. Taking the average over Eq. \textit{(C.2)} in a vicinity around $E_x$, $E_{\gamma}$ and substituting, we obtain

$$\langle \omega_{i \to f} \rangle = \frac{2\epsilon_0 N\gamma}{\hbar^2} \frac{(E_{\gamma})^{2\lambda+1}}{(2\lambda + 1)!!} \sum_{E_x, J, \pi} \frac{f_{XL}(E_{\gamma}, E_x, J, \pi)}{\rho_{J, \pi}(E_x)} \rho_{\text{avail}}(E_f),$$ \tag{C.3}$$

where $E_f = E_x - E_{\gamma}$. In the Oslo method, since the decays happen at high $E_x$, we assume that dipole radiation dominates. This is well supported experimentally \cite{5, 49, 18, 19}. The selection rules dictate that dipole radiation changes the angular momentum $J$ by at most one unit. For $M_1$, the parity is unchanged, while for $E1$ it flips. This determines the density of available final states for the decay,\footnote{In the case of $J_f = 1/2$ the sum runs over $J_f = \{1/2, 3/2, \}$, and in the case of $J_f = 0$, the sum only runs over $J_f = 1$, since $J = 0 \to J = 0$ transitions are forbidden.} \textit{strictly speaking}, the factor is less than 3 when $J_f$ is less than 1. We assume that this is a small correction and can be neglected.

$$\rho_{\text{avail}}(E_f) = \sum_{J_f = J_i - 1}^{J_i + 1} 3\rho(E_f, J_f, \pi_f).$$ \tag{C.4}$$

The factor 3 comes from the distinction between levels and states.\footnote{For a nuclear level with a given $J_f$, there are $2J_f + 1$ distinct magnetic substates. However, in an electromagnetic dipole decay, the total $M$ quantum number cannot change by more than one unit, which means that only three of the magnetic substates are available, independent of $J_f$. We may then write the average total dipole transition rate $\langle \omega_i \rangle = \langle \omega_{E1} \rangle + \langle \omega_{M1} \rangle$ as

$$\langle \omega_{J, \pi, (E_x, E_{\gamma})} \rangle = \frac{CE_{\gamma}^3}{\rho_{J, \pi}(E_x)} \times \left( f_{E1}(E_{\gamma}, E_x, J_i, \pi_i) \sum_{J_f = J_i - 1}^{J_i + 1} \rho(E_f, J_f, -\pi_i) + f_{M1}(E_{\gamma}, E_x, J_i, \pi_i) \sum_{J_f = J_i - 1}^{J_i + 1} \rho(E_f, J_f, +\pi_i) \right),$$ \tag{C.5}$$
where all constants have been grouped together and named $C$ for brevity. Let us next define the total dipole strength function $f_1$ by

$$f_1 = f_{E1} + f_{M1}. \quad (C.6)$$

To factor the expression, we need to assume parity equilibration of the level density, i.e. $\rho(E_x, J, \pi) \approx \rho(E_x, J, -\pi)$. Then we can write

$$\langle \omega_{J_i, \pi_i}(E_x, E_{\gamma}) \rangle = \frac{CE_\gamma^3}{\rho_{J_i, \pi_i}(E_x)} f_1(E_{\gamma}, E_x, J_i, \pi_i) \times \sum_{J_f=J_i-1}^{J_i+1} \rho(E_f, J_f, \text{eq}), \quad (C.7)$$

where $\rho(E_x, J_f, \text{eq})$ denotes the level density of one parity, the notation emphasising the assumption of parity equilibration.

Assuming that $E_x$ is below the threshold energy for particle emission, and that dipole radiation dominates, Eq. (C.7) represents all possible decay modes. We may thus exploit probability conservation and write

$$P_{J_i, \pi_i}(E_x, E_{\gamma}) = \frac{\langle \omega_{J_i, \pi_i}(E_x, E_{\gamma}) \rangle}{\sum_{E_x} \langle \omega_{J_i, \pi_i}(E_x, E_{\gamma}) \rangle} \times \sum_{J_f=J_i-1}^{J_i+1} \rho_{J_f, \pi_f}(E_x - E_{\gamma}) \rho_{J_f, \pi_f}(E_x - E_{\gamma}) \times \sum_{J_f=J_i-1}^{J_i+1} \rho(E_f, J_f, \text{eq}), \quad (C.8)$$

where $D$ is another normalisation constant. Note that the density of initial states cancels out.

The final step is to remove the dependence on $J$ and $\pi$, since we cannot discriminate between them experimentally. By the generalised Brink-Axel hypothesis, as discussed in the previous chapter, the strength function is approximately independent of $E_x$, $J$ and $\pi$, so we may write

$$f_1(E_{\gamma}, E_x, J_i, \pi_i) \approx f_1(E_{\gamma}). \quad (C.9)$$

Inserting, this gives

$$P(E_x, E_{\gamma}) = DE_\gamma^3 f_1(E_{\gamma}) \sum_{J_i, \pi_i} g_{\text{pop}}(E_x, J_i, \pi_i) \times \sum_{J_f=J_i-1}^{J_i+1} \rho(E_x - E_{\gamma}, J_f, \text{eq}). \quad (C.11)$$

Lastly, we may write the partial level density $\rho(E_x - E_{\gamma}, J_f, \text{eq})$ as

$$\rho(E_x - E_{\gamma}, J_f, \text{eq}) = g_{\text{int}}(E_x - E_{\gamma}, J_i, \text{eq}) \rho(E_x - E_{\gamma}), \quad (C.12)$$

where $g_{\text{int}}$ denotes the intrinsic spin distribution of the nucleus and $\rho(E_x)$ the total nuclear level density. Factoring out, we obtain

$$P(E_x, E_{\gamma}) = DE_\gamma^3 f_1(E_{\gamma}) \rho(E_x - E_{\gamma}) z(E_x, E_{\gamma}), \quad (C.13)$$

where we have defined

$$z(E_x, E_{\gamma}) = \sum_{J_i, \pi_i} g_{\text{pop}}(E_x, J_i, \pi_i) \times \sum_{J_f=J_i-1}^{J_i+1} g_{\text{int}}(E_x - E_{\gamma}, J_i, \text{eq}). \quad (C.15)$$

In the Oslo method, it is assumed that $z(E_x, E_{\gamma}) \approx \text{constant}$. This is not completely correct, but the correction is believed to be small. Setting $z = \text{const}$.

$$P(E_x, E_{\gamma}) = DE_\gamma^3 f_1(E_{\gamma}) \rho(E_x - E_{\gamma}). \quad (C.16)$$

for a suitable normalisation constant $D$.

**Appendix D. Using OMPy**

OMPy, as well as the following example script, is available from our repository on GitHub. It is written to be flexible and to interface easily with the larger Python ecosystem. This is reflected in the design choices—for instance, all numeric arrays such as vectors and matrices are stored as Numpy objects, with all the functionality that comes with

[https://github.com/oslocyclotronlab/ompy](https://github.com/oslocyclotronlab/ompy)
In the following, we give a short introduction to the use of OMpy. More comprehensive documentation can be found online in the Github repository. OMpy is loaded into a Python environment by executing

```python
import ompy
```

Typically, the user will want to begin by loading an input matrix of \( E_x - E_\gamma \) coincidences. OMpy supplies a custom class called a Matrix, which enables storage of two-dimensional spectra in memory along with their energy calibration information. To load a coincidence matrix from a MAMA file, dubbed raw matrix to distinguish it from the unfolded and first-generation matrices that the Oslo method extracts, do

```python
fname_raw = "path/to/raw/matrix.m"
raw = ompy.Matrix(fname=fname_raw)
```

The Matrix class contains many methods, such as

```python
# Load a matrix from file into an existing Matrix instance:
raw.load(fname)
# Save the matrix to file:
raw.save(fname)
# Get the calibration of the energy axes, returned as a dictionary:
calib = raw.calibration()
# Cut away parts of the matrix defined by E_limits along axis:
raw.cut_rect(axis, E_limits)
# Zero all bins to the right of a diagonal line defined by coordinates:
raw.cut_diagonal([[Ex1, Eg1], [Ex2, Eg2]])
# Plot the matrix:
raw.plot()
# Plot a projection of an energy region E_limits onto one of the axes:
raw.plot_projection(E_limits, axis)
```

For one-dimensional spectra, OMpy supplies a corresponding class called Vector, with similar functionality.

With the raw matrix loaded and inspected, the user can proceed to do unfolding. This is done by the command

```python
unfolded = ompy.unfold(raw,
response_function,
Ex_min, Ex_max,
Eg_min, Eg_max,
diag_cut,
use_comptonsubtraction)
```

Here, response_function indicates which response function to use from a list of available experimental response functions, the Ex_min, etc., give the limits for the unfolding, diag_cut places a diagonal cut on the matrix, and use_comptonsubtraction determines whether to turn on the Compton subtraction method from Ref. [14]. Next, the first generation method can be applied by

```python
firstgen = ompy.first_generation_method(unfolded,
Ex_max, dE_gamma,
N_iterations,
multiplicity_estimation,
area_correction)
```

Here, Ex_max is the maximum excitation energy to include, dE_gamma is the amount by which \( E_\gamma \) is allowed to exceed \( E_x \) due to experimental resolution, N_iterations is the number of iterations to use, multiplicity_estimation determines the method to use for multiplicity estimation (either "statistical" or "total", see Ref. [15]), and area_correction determines whether to use the area correction described in Ref. [15].

The steps described above make up the core matrix manipulation routines of OMpy. We therefore provide a class which helps to streamline the process, called MatrixAnalysis.

```python
ma = ompy.MatrixAnalysis()
# Load raw matrix:
ma.raw.load(fname_raw)
# Unfold the raw matrix and place it in the variable ma.unfolded:
ma.unfold(*args)
# Apply first generation method and get result in ma.firstgen:
ma.first_generation_method(*args)
```

The class methods ma.unfold() and ma.first_generation_method() accept the same input arguments as their respective stand-alone functions. The advantage of the MatrixAnalysis class is that all parameter choices for the unfolding and first-generation method are stored within the class instance. For this reason, the MatrixAnalysis class is necessary when one wants to do uncertainty propagation, to ensure that the same settings are applied to each member of the ensemble. To perform the uncertainty propagation, instantiate the class ErrorPropagation with the instance of MatrixAnalysis as an input argument:
ep = ompy.ErrorPropagation(ma)

Then, an ensemble of \( N_{\text{ensemble members}} \) is generated by the command

\[
\text{ep.generate_ensemble}(N_{\text{ensemble members}})
\]

The standard deviations matrices for \text{raw}, \text{unfolded} and \text{firstgen} are automatically calculated and put in the variables \text{ep.std_raw}, \text{ep.std_unfolded} and \text{ep.std_firstgen}, respectively.

To fit \( \rho \) and \( T \) to the first-generation matrix (also requiring \text{ep.std_firstgen}), do

\[
# \text{Allocate arrays to store each ensemble member:}
# \rho_\text{ens} = \text{np.zeros}((N_{\text{ensemble fit}}, \text{len} (\text{rho.vector})))
# \text{T}_\text{ens} = \text{np.zeros}((N_{\text{ensemble fit}}, \text{len} (\text{T.vector})))
#
# As a trick, we copy the instance \text{ma} and replace its matrix every iteration:
import copy
\text{ma}_\text{curr} = \text{copy.deepcopy}(\text{ma})
#
# Loop through all and perform fit:
for \text{i}_\text{ens} in range(N_{\text{ensemble fit}}):
\text{ma}_\text{curr}.firstgen.matrix = \$
\text{ep.firstgen_ensemble[\text{i}_\text{ens}, : , :]}$
\text{fitter}_\text{curr} = \text{ompy.FitRhoT}(\text{ma}_\text{curr}.firstgen, \text{ep.std_firstgen}, \text{bin_width_out}, \text{Ex_min}, \text{Ex_max}, \text{Eg_min})
\text{rho}_\text{curr} = \text{fitter}_\text{curr}.\rho
\text{T}_\text{curr} = \text{fitter}_\text{curr}.T
\text{rho}_\text{ens}[\text{i}_\text{ens}, :] = \text{rho}_\text{curr}.\text{vector}
\text{T}_\text{ens}[\text{i}_\text{ens}, :] = \text{T}_\text{curr}.\text{vector}
\]

Here, \text{bin_width_out} gives the energy bin width of the resulting fit, and the other arguments determine the region of the first-generation matrix to fit to. To get the uncertainty on the fitted quantities, one can run the fitting for each member in the ensemble:

\[
# \text{Allocate arrays to store each ensemble member:}
# \rho_\text{ens} = \text{np.zeros}((N_{\text{ensemble fit}}, \text{len} (\text{rho.vector})))
# \text{T}_\text{ens} = \text{np.zeros}((N_{\text{ensemble fit}}, \text{len} (\text{T.vector})))
#
# As a trick, we copy the instance \text{ma} and replace its matrix every iteration:
import copy
\text{ma}_\text{curr} = \text{copy.deepcopy}(\text{ma})
#
# Loop through all and perform fit:
for \text{i}_\text{ens} in range(N_{\text{ensemble fit}}):
\text{ma}_\text{curr}.firstgen.matrix = \$
\text{ep.firstgen_ensemble[\text{i}_\text{ens}, : , :]}$
\text{fitter}_\text{curr} = \text{ompy.FitRhoT}(\text{ma}_\text{curr}.firstgen, \text{ep.std_firstgen}, \text{bin_width_out}, \text{Ex_min}, \text{Ex_max}, \text{Eg_min})
\text{rho}_\text{curr} = \text{fitter}_\text{curr}.\rho
\text{T}_\text{curr} = \text{fitter}_\text{curr}.T
\text{rho}_\text{ens}[\text{i}_\text{ens}, :] = \text{rho}_\text{curr}.\text{vector}
\text{T}_\text{ens}[\text{i}_\text{ens}, :] = \text{T}_\text{curr}.\text{vector}
\]

Appendix D.1. Adding new features
OMpy is written with modularity in mind. We want it to be as easy as possible for the user to add custom functionality and interface OMPy with other Python packages. For example, it may be of interest to try other unfolding algorithms than the one presently implemented. To achieve this, one just has to write a wrapper function that has the same input and output structure as the function unfold(), found in the file omy/unfold.py, and replace the calls to unfold() by the custom function.

It is our hope and goal that OMPy will be used as much as possible, and we are happy to provide support. Feedback and suggestions are also very welcome. We encourage users who implement new features to share them by opening a pull request in the Github repository.

References

[1] M. Arnould, S. Goriely, K. Takahashi, The \( r \)-process of stellar nucleosynthesis: Astrophysics and nuclear physics achievements and mysteries, Physics Reports 450 (4) (2007) 97–213. doi:10.1016/j.physrep.2007.06.002.
URL http://www.sciencedirect.com/science/article/pii/S0370157307002438

[2] F. Käppeler, R. Gallino, S. Bisterzo, W. Aoki, The \( s \)-process: Nuclear physics, stellar models, and observations, Reviews of Modern Physics 83 (1) (2011) 157–193. doi:10.1103/RevModPhys.83.157.
URL https://link.aps.org/doi/10.1103/RevModPhys.83.157

[3] M. B. Chadwick et al., ENDF/B-VII.1 Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay Data, Nuclear Data Sheets 112 (12) (2011) 2887–2996. doi:10.1016/j.nds.2011.11.002.
URL http://www.sciencedirect.com/science/article/pii/S009037521100113X

[4] A. Schiller, L. Bergholt, M. Guttormsen, E. Melby, J. Rekstad, S. Siem, Extraction of level density and \( \gamma \) strength function from primary \( \gamma \) spectra, Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment ArXiv: nucl-ex/9910009. doi:10.1016/S0168-9002(99)01187-0

[5] A. C. Larsen, N. Blasi, A. Bracco, F. Camera, T. K. Eriksen, A. Gørgen, M. Guttormsen, T. W. Hagen, S. Leoni, B. Million, H. T. Nyhus, T. Renstrøm, S. J. Rose, I. E. Ruid, S. Siem, T. Tormy, G. M. Tweten, A. V. Voinov, M. Wiedeking, Evidence for the Dipole Nature of the Low-Energy \( \gamma \) Enhancement in \( {\text{^{56}Fe}} \), Physical Review Letters 111 (24) (2013) 242504. doi:10.1103/PhysRevLett.111.242504.
URL https://link.aps.org/doi/10.1103/PhysRevLett.111.242504
[20] A. Voinov, S. M. Grimes, C. R. Brune, M. J. Hornish, T. N. Massey, A. Salas, Test of nuclear level density inputs for Hauser-Feshbach model calculations. Physical Review C 76 (4) (2007) 044602. doi:10.1103/PhysRevC.76.044602

[21] A. Voinov, B. M. Oginni, S. M. Grimes, C. R. Brune, M. Guttormsen, A. C. Larsen, T. N. Massey, A. Schiller, S. Siem, Nuclear excitations at constant temperature. Physical Review C 79 (3) (2009) 031301. doi:10.1103/PhysRevC.79.031301

[22] A. P. D. Ramirez, A. V. Voinov, S. M. Grimes, Y. Byun, C. R. Brune, T. N. Massey, S. Akhtar, S. Dhakal, C. E. Parker, Level density and mechanism of deuteron-induced reactions on $^{49,50,52}$Fe. Physical Review C 92 (1) (2015) 014303. doi:10.1103/PhysRevC.92.014303

[23] R. Sherr, F. P. Brady, Spectra of $(p,p')$ and $(p,p)$ Reactions and the Evaporation Model. Physical Review 124 (6) (1961) 1928–1943. doi:10.1103/PhysRev.124.1928

[24] B. V. Zhuravlev, A. A. Lychagin, N. N. Titarenko, V. G. Demenkov, V. I. Trykovka, Nuclear level densities in $^{44}$V, $^{45}$V, $^{47}$V, $^{49}$V, and $^{51}$V and $^{47}$Mn and $^{49}$Mn from neutron evaporation spectra. Physics of Atomic Nuclei 74 (3) (2011) 335–340. doi:10.1134/S1063778811030173

[25] M. Guttormsen, R. Chankova, U. Agvaanluvsan, E. Algin, L. A. Bernstein, F. Ingebretsen, T. L"onroth, S. Messelt, G. E. Mitchell, J. Rekstad, A. Schiller, S. Siem, A. C. Sunde, A. Voinov, S. Odérgård, Radiative strength functions in $^{93}$–$^{98}$Mo. Physical Review C 71 (4) (2005) 044307. doi:10.1103/PhysRevC.71.044307

[26] P. A. M. Dirac, N. H. D. Bohr, The quantum theory of the emission and absorption of radiation. Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character 114 (767) (1927) 243–265. doi:10.1098/rspa.1927.0039

[27] M. Fermi, Nuclear Physics. University of Chicago Press, Chicago, 1950. URL https://www.press.uchicago.edu/ucp/books/bookchicago/Bbco3651262.html

[28] D. M. Brink, PhD thesis, Oxford, [unpublished] (1955).

[29] A. Wallner, B. Strohmaier, H. Vonach, Extraction of neutron level densities from neutron spectra emitted in proton-induced reactions on lead isotopes and B. Physical Review C 51 (2) (1995) 614–622. doi:10.1103/PhysRevC.51.614

[30] F. Bečvář, P. Cejnar, R. E. Chrien, J. Kopecký, Test of photon strength functions by a method of two-step cascades. Physical Review C 46 (4) (1992) 1276–1287. doi:10.1103/PhysRevC.46.1276

[31] M. Krtička, F. Bečvář, J. Honzátko, I. Tomandl, M. Heil, F. Käppeler, R. Reifarth, F. Voss, K. Wissel, Evidence for $^{117}$I Scissors Resonances Built on the Levels in the Quasicontinuum of $^{115}$I. Physical Review Letters 92 (17) (2004) 172501. doi:10.1103/PhysRevLett.92.172501

[32] J. Kopecký, R. E. Chrien, Observation of the $^{117}$I giant resonance by resonance averaging in $^{108}$Pd. Nuclear Physics A 468 (2) (1987) 265–300. doi:10.1016/0375-9474(87)90518-5

[33] E. O. Lebigot, Uncertainties: a Python package for Scientific tools. URL http://www.uncertainties-python-package.org/

[34] E. Jones, T. Oliphant, P. Peterson, others, SciPy: Open source scientific tools for Python. 2001. URL http://www.scipy.org/

[35] M. J. D. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives. The Computer Journal 7 (2) (1964) 155–162. doi:10.1093/comjnl/7.2.155

[36] S. Agostinelli et al., Geant4—a simulation toolkit. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 506 (3) (2003) 250–303. doi:10.1016/S0168-9002(03)01368-8

[37] J. Allison et al., Recent developments in Geant4. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 516 (1) (2004) 2–15. doi:10.1016/j.nima.2003.12.065

[38] J. Allison et al., Geant4 developments and applications. IEEE Transactions on Nuclear Science 53 (1) (2006) 270–278. doi:10.1109/TNS.2006.869826

[39] J. Allison et al., Recent developments in Geant4. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 516 (1) (2004) 2–15. doi:10.1016/j.nima.2004.07.058

[40] E. Fermi, Nuclear Physics. University of Chicago Press, Chicago, 1950. URL https://www.press.uchicago.edu/ucp/books/bookchicago/Bbco3651262.html

[41] D. M. Brink, PhD thesis, Oxford, [unpublished] (1955).

[42] A. Wallner, B. Strohmaier, H. Vonach, Extraction of neutron level densities from neutron spectra emitted in proton-induced reactions on lead isotopes and B. Physical Review C 51 (2) (1995) 614–622. doi:10.1103/PhysRevC.51.614

[43] F. Bečvář, P. Cejnar, R. E. Chrien, J. Kopecký, Test of photon strength functions by a method of two-step cascades. Physical Review C 46 (4) (1992) 1276–1287. doi:10.1103/PhysRevC.46.1276

[44] J. Kopecký, R. E. Chrien, Observation of the $^{117}$I giant resonance by resonance averaging in $^{108}$Pd. Nuclear Physics A 468 (2) (1987) 265–300. doi:10.1016/0375-9474(87)90518-5

[45] E. O. Lebigot, Uncertainties: a Python package for calculations with uncertainties. 2018. URL http://www.uncertainties-python-package.org/

[46] E. Jones, T. Oliphant, P. Peterson, others, SciPy: Open source scientific tools for Python. 2001. URL http://www.scipy.org/

[47] M. J. D. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives. The Computer Journal 7 (2) (1964) 155–162. doi:10.1093/comjnl/7.2.155

[48] S. Agostinelli et al., Geant4—a simulation toolkit. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 506 (3) (2003) 250–303. doi:10.1016/S0168-9002(03)01368-8

[49] J. Allison et al., Geant4 developments and applications. IEEE Transactions on Nuclear Science 53 (1) (2006) 270–278. doi:10.1109/TNS.2006.869826

[50] J. Allison et al., Recent developments in Geant4. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 516 (1) (2004) 2–15. doi:10.1016/j.nima.2004.07.058

[51] E. Fermi, Nuclear Physics. University of Chicago Press, Chicago, 1950. URL https://www.press.uchicago.edu/ucp/books/bookchicago/Bbco3651262.html
[43] G. D’Agostini, Improved iterative Bayesian unfolding, arXiv:1010.0632 [physics], ArXiv:1010.0632. URL http://arxiv.org/abs/1010.0632

[44] G. Choudalakis, Fully Bayesian Unfolding, arXiv:1201.4612 [physics], ArXiv:1201.4612. URL http://arxiv.org/abs/1201.4612

[45] PyPI – the Python Package Index. URL https://pypi.org/ Accessed 24.04.2019.

[46] T. v. Egidy, D. Bucurescu, Systematics of nuclear level density parameters, Physical Review C 72 (4) (2005) 044311. doi:10.1103/PhysRevC.72.044311 URL https://link.aps.org/doi/10.1103/PhysRevC.72.044311

[47] P. Ring, P. Schuck, The Nuclear Many-Body Problem, Theoretical and Mathematical Physics, The Nuclear Many-Body Problem, Springer-Verlag, Berlin Heidelberg, 1980. URL https://www.springer.com/gp/book/9783540212055

[48] G. A. Bartholomew, E. D. Earle, A. J. Ferguson, J. W. Knowles, M. A. Lone, Gamma-Ray Strength Functions in: M. Baranger, E. Vogt (Eds.), Advances in Nuclear Physics: Volume 7, Springer US, Boston, MA, 1973, pp. 229–324. doi:10.1007/978-1-4615-9044-6_4 URL https://doi.org/10.1007/978-1-4615-9044-6_4

[49] A. C. Larsen, M. Gutormsen, R. Schwengner, D. L. Bleuel, S. Goriely, S. Harissopulos, F. L. Bello Garrote, Y. Byun, T. K. Eriksen, F. Giacoppo, A. Gorgen, T. W. Hagen, M. Klintefjord, T. Renstrøm, S. J. Rose, E. Sahin, S. Siem, T. G. Tornyi, G. M. Tveten, A. V. Voinov, M. Wiedeking, Experimentally constrained (p,γ)89Y and (n,γ)89Y reaction rates relevant to p-process nucleosynthesis, Physical Review C 93 (4) (2016) 045810. doi:10.1103/PhysRevC.93.045810 URL https://link.aps.org/doi/10.1103/PhysRevC.93.045810