A direct method for unfolding the resolution function from measurements of neutron induced reactions

(n_TOF Collaboration) Žugec, Petar; Colonna, N.; Sabate-Gilarte, M.; Vlachoudis, V.; Massimi, C.; Lerendegui-Marco, J.; Stamatopoulos, A.; Bacak, M.; Warren, S. G.

Journal article, Published version
Rad u časopisu, Objavljena verzija rada (izdavačev PDF)

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:036085

Download date / Datum preuzimanja: **2025-02-23**

Repository / Repozitorij:

Repository of the Faculty of Science - University of Zagreb

Contents lists available at ScienceDirect

Nuclear Inst. and Methods in Physics Research, A

journal homepage: www.elsevier.com/locate/nima

A direct method for unfolding the resolution function from measurements of neutron induced reactions

P. Žugec [a],[*], N. Colonna [b], M. Sabate-Gilarte [c],[d], V. Vlachoudis [c], C. Massimi [e],[f],
J. Lerendegui-Marco [d], A. Stamatopoulos [g], M. Bacak [c],[h],[i], S.G. Warren [c],
The n_TOF Collaboration [1]

[a] *Department of Physics, Faculty of Science, University of Zagreb, Croatia*
[b] *Istituto Nazionale di Fisica Nucleare, Sezione di Bari, Italy*
[c] *European Organization for Nuclear Research (CERN), Geneva, Switzerland*
[d] *Universidad de Sevilla, Spain*
[e] *Istituto Nazionale di Fisica Nucleare, Sezione di Bologna, Italy*
[f] *Dipartimento di Fisica e Astronomia, Università di Bologna, Italy*
[g] *National Technical University of Athens (NTUA), Greece*
[h] *Atominstitut der Österreichischen Universitäten, Technische Universität Wien, Austria*
[i] *Irfu, CEA, Université Paris-Saclay, Gif-sur-Yvette, France*

A R T I C L E   I N F O

A B S T R A C T

The paper explores the numerical stability and the computational efficiency of a direct method for unfolding the resolution function from the measurements of the neutron induced reactions. A detailed resolution function formalism is laid out, followed by an overview of challenges present in a practical implementation of the method. A special matrix storage scheme is developed in order to facilitate both the memory management of the resolution function matrix, and to increase the computational efficiency of the matrix multiplication and decomposition procedures. Due to its admirable computational properties, a Cholesky decomposition is at the heart of the unfolding procedure. With the smallest but necessary modification of the matrix to be decomposed, the method is successfully applied to system of $10^5 \times 10^5$. However, the amplification of the uncertainties during the direct inversion procedures limits the applicability of the method to high-precision measurements of neutron induced reactions.

1. Introduction

The resolution function of the neutron beam is an inherent characteristic of the neutron production facilities, such as the neutron time of flight facility n_TOF [1] at CERN. In particular, the resolution function $R(t, E)$ is the distribution of the neutron flight time t (over the flight path of a fixed length L) for neutrons of a given kinetic energy E. As opposed to the idealized one-to-one correspondence between time of flight and neutron energy, the flight time can vary due to several experimental effects: (1) the time spread of the primary beam of charged particles producing the neutron beam; (2) the variable moderation time in the target-moderator assembly; (3) the geometry of the neutron propagation along the beam line of finite length and diameter. In case of the n_TOF facility, the width of the primary 20 GeV proton beam

from the CERN Proton Synchrotron is 7 ns RMS. The target-moderator assembly consists of a massive lead spallation target, 40 cm in length and 60 cm in diameter, surrounded by a cooling system comprised of 1 cm layer of demineralized water and additional 4 cm of either borated or demineralized water. The n_TOF facility features two experiential areas: Experimental Area 1 (EAR1 [1]) at the horizontal distance of 185 m from the spallation target, and Experimental Area 2 (EAR2 [2–4]) at the vertical distance of 20 m from the same target. Therefore, two separate resolution functions must be taken into account, one for each experimental area.

At n_TOF the resolution function of the neutron beam has been given consideration ever since the very conception of the facility [5], throughout the start of its operation [6,7], to the present day [1,2,8,9].

Now, after entering n_TOF-Phase3 — the third phase in the operation of the n_TOF facility, marked by the successful completion, commissioning and the start of the operation of EAR2 — the resolution function considerations are being pursued with greater fervour than ever before [2,8,9]. These efforts are motivated by the requirement for the optimum quality in processing the high precision experimental data obtained at n_TOF, which are regularly a key input to the development of nuclear technologies and are of importance to the general scientific community [10,11].

The reliable evaluation of the resolution function throughout wide neutron energy range may only be obtained through the dedicated simulations of the neutron production and their propagation through the target-moderator assembly. However, the only way of evaluating the reliability of the simulated results is by benchmarking them against the experimental data, by means of applying the simulated resolution function to the estimated reaction yield of well known resonances (such as the neutron capture resonances of ^{25}Mg, ^{56}Fe, ^{197}Au) and comparing them with the available measurements [8,9]. Once the reliability of the resolution function has been confirmed, it may be used either in its numerical format, or it may be fitted to an appropriate analytical form. The form from Ref. [1] is used at n_TOF, while Ref. [12] lists some other forms widely used at other neutron production facilities.

Ultimately, the resolution function in either form needs to be applied to the experimental data, in order to decouple its effect from the measurements, in particular from the measured resonances in neutron induced reactions. One way of performing this task is by relying on specialized codes like SAMMY [12] or REFIT [13], which use the R-matrix formalism to fit the experimental data to the parameterized form of the underlying cross section. In the process the parameterized cross section is folded or convoluted with the self-shielding, multiple scattering, Doppler broadening and, ultimately, the resolution function effects in order to reproduce the observable and, indeed, measured reaction yield. This is certainly a very robust, stable and reliable method. Notwithstanding, in this work we explore an alternative approach, its applicability and limitations. While the SAMMY/REFIT approach is evidently a forward application of the resolution function to the assumed pure form of the underlying cross section, we seek to address the inverse problem: how to directly unfold the resolution function from the measurements, starting from the data already affected by it.

Section 2 presents the theoretical introduction to the resolution function, establishing the central problem to be solved. Section 3 reports on the successfully applied unfolding procedure. Section 4 addresses the propagation of uncertainties, which is a crucial issue for any of the direct unfolding procedures. Section 5 sums up the conclusions of this work. Appendix A presents a matrix storage scheme central to this work and lays out the important technical procedures to be performed prior to the unfolding itself. In addition, Appendix B addresses the applicability of the unfolding procedure in the presence of pronounced multiple scattering effects.

2. Theoretical framework

2.1. Resolution function

Let us consider the resolution function $R_X(X, E)$ expressed as a function of some general kinematic parameter X. For the moment we identify X with the neutron time of flight: $X = t$. Then the precise definition of the resolution function takes the form:

$$R_X(X, E) \equiv \frac{\mathrm{d}P_X(X, E)}{\mathrm{d}X} \tag{1}$$

where $\mathrm{d}P_X(X, E)$ is the probability[2] for the neutron of kinetic energy E to have the corresponding kinematic parameter X within the interval

---

[2] The resolution function is normalized over X for every value of E:

$$\int_{-\infty}^{\infty} R_X(X, E)\mathrm{d}X = 1 \quad \forall E.$$

$\mathrm{d}X$, i.e. to have the time of flight t within the time of flight interval $\mathrm{d}t$. Aside from the time of flight, one may express the resolution function in terms of the reconstructed energy $\mathcal{E}$, which is the equivalent neutron kinetic energy calculated from the relativistic time–energy relation:

$$\mathcal{E} = mc^2 \left\{ \left[1 - \left(\frac{L}{ct} \right)^2 \right]^{-1/2} - 1 \right\} \tag{2}$$

with m as the neutron mass, c as the speed of light in vacuum and L as the mean neutron flight path. Another commonly used parameterization is the one defining the effective neutron flight path λ:

$$\lambda \equiv v_E t - \lambda_0 = \frac{\sqrt{\left(E/mc^2 \right)^2 + 2E/mc^2}}{1 + E/mc^2} \times ct - \lambda_0 \tag{3}$$

where v_E is the neutron speed calculated from a true neutron kinetic energy E. In principle, the value of the shift parameter λ_0 is arbitrary, but two conventions are commonly used: $\lambda_0 = 0$ or $\lambda_0 = L$. Under the assumption of $\lambda_0 = 0$, the effective path length λ corresponds to the entire path length that the neutron of energy E would need to traverse to arrive at the measurement position at time t. On the other hand, when $\lambda_0 = L$ is assumed, then λ is interpreted as an effective path the neutron needs to traverse within the target-moderator assembly. Whichever kinematic parameter $X \in \{t, \mathcal{E}, \lambda\}$ is selected for expressing the resolution function, the probability conservation requires the following to hold:

$$R_t(t, E) \, |\mathrm{d}t| = R_{\mathcal{E}}(\mathcal{E}, E) \, |\mathrm{d}\mathcal{E}| = R_\lambda(\lambda, E) \, |\mathrm{d}\lambda| \tag{4}$$

allowing for a simple transformation between different forms of the resolution function.

A note should be taken of the relation between the resolution functions at different paths, in particular at the nominal neutron production point $L = 0$ and at some specific distance $L > 0$ corresponding to a measurement position. One may be tempted to use the function $R_t^{(L=0)}(\delta t, E)$ (with δt as effective production time relative to the nominal initial moment $t = 0$) in order to reconstruct the resolution function $R_t^{(L>0)}(t, E)$ at any flight path L simply by propagating the neutron in time by the time of flight $t_E = L/v_E$, with v_E as the neutron speed defined by Eq. (3). However, the inequality:

$$R_t^{(L>0)}(t, E) \neq R_t^{(L=0)}(t - t_E, E) \tag{5}$$

holds in general, for two reasons. One is that many neutrons reach the measurement position propagating under a slight angle θ relative to the shortest flight path, making their actual path length slightly longer ($L/\cos \theta$). The other reason is that the neutrons exiting the primary neutron source — directly giving rise to $R_t^{(L=0)}(\delta t, E)$ — may interact with any material along the flight path, thus being lost, delayed and/or producing secondary neutrons with kinematic parameters uncorrelated to their own initial ones. Therefore, the final resolution function $R_t^{(L>0)}(t, E)$ may have additional contributions, not inherited from $R_t^{(L=0)}(\delta t, E)$ at the neutron production point. For this reason, throughout this work we will only consider the resolution functions at the measurement position.

2.2. Continuous representation

Let us first lay down the formalism of continuous resolution functions, establishing the connection between the measured spectrum $S_X(X)$ expressed as a function of an experimentally accessible kinematic parameter X, and the underlying spectrum $S_E(E)$ dependent on the true neutron kinetic energy E. The considerations are equally valid whether

---

In principle, the lower limit of integration may not be 0, depending on the definition of the kinematic parameter X. For example, negative values of $X = t$ are introduced by an arbitrary selection of the nominal initial moment $t = 0$ for the neutron production. If the initial moment is selected such that no neutrons are produced prior to $t = 0$, only then may the lower limit of integration be set to 0.

the selected kinematic parameter is the neutron time of flight ($X = t$) or the reconstructed neutron energy ($X = \mathcal{E}$).

Let $d^2 N(X, E)$ be the number of reactions caused by the neutrons of energy E and detected with the kinematic parameter X. We define the corresponding spectrum $S(X, E)$ of detected counts as:

$$S(X, E) \equiv \frac{d^2 N(X, E)}{dX\, dE}. \tag{6}$$

We wish to reconstruct the spectrum $S_E(E)$, unaffected by the resolution function:

$$S_E(E) \equiv \frac{dN_E(E)}{dE} = \int_{-\infty}^{\infty} S(X, E)\, dX \tag{7}$$

starting from the measured spectrum $S_X(X)$:

$$S_X(X) \equiv \frac{dN_X(X)}{dX} = \int_0^{\infty} S(X, E)\, dE. \tag{8}$$

Since $dN_E(E) = S_E(E)\, dE$ is the total number of detected counts caused by the neutrons of energy E, the number of detections with a particular value of the kinematic parameter X equals:

$$d^2 N(X, E) = dN_E(E) \times dP_X(X, E) = S_E(E) R_X(X, E)\, dX\, dE \tag{9}$$

where we have taken advantage of the probability $dP_X(X, E)$ and the definition of the resolution function $R_X(X, E)$ from Eq. (1). Inserting this expression into Eq. (6) yields:

$$S(X, E) = S_E(E) R_X(X, E) \tag{10}$$

which, in turn, via Eq. (8) leads to:

$$S_X(X) = \int_0^{\infty} S_E(E) R_X(X, E)\, dE \tag{11}$$

otherwise known as a Fredholm integral equation of the first kind.

2.3. Discretized representation

In practical data analysis, the amount of data is finite and a discretized version of Eq. (11) is required:

$$S_X(X_i) = \sum_j S_E(E_j) R_X(X_i, E_j) \Delta E_j \tag{12}$$

with indices i and j denoting the corresponding histogram bins. In this case the discretized version of the resolution function is simply the average over both the source and the destination bin:

$$R_X^{(\Delta > 0)}(X_i, E_j) = \frac{1}{\Delta X_i \Delta E_j} \int_{\Delta X_i} \int_{\Delta E_j} R_X^{(\Delta \to 0)}(X, E)\, dX\, dE \tag{13}$$

with ΔX_i and ΔE_j as the corresponding bin widths. $R_X^{(\Delta > 0)}$ on the left side of Eq. (13) denotes the discretized representation, while $R_X^{(\Delta \to 0)}$ from the right side the continuous one. By constructing the vectors $\vec{S}_E$ and $\vec{S}_X$ of the histogrammed spectra such that $[\vec{S}_E]_i \equiv S_E(E_i)$ and $[\vec{S}_X]_i \equiv S_X(X_i)$, Eq. (12) may be rewritten in a compact matrix form:

$$\vec{S}_X = \mathbf{R}\, \vec{S}_E \tag{14}$$

with the resolution function matrix $\mathbf{R}$ defined as:

$$\mathbf{R}_{ij} \equiv R_X(X_i, E_j) \Delta E_j = \frac{\Delta E_j}{\Delta X_i} \Delta P_X(X_i, E_j). \tag{15}$$

Evidently, ΔP_X is the discretized version of the probability dP_X from Eq. (1).

Our ultimate goal — obtaining the spectrum $S_E(E)$ from the measured spectrum $S_X(X)$ — is now reduced to finding a solution of the linear system from Eq. (14). At this point it is useful to note that the matrix $\mathbf{R}$ may be rewritten as:

$$\mathbf{R} = \mathbf{X}^{-1}\mathbf{P}\mathbf{E} \tag{16}$$

where $\mathbf{X} \equiv \text{diag}[\Delta X_i]$ and $\mathbf{E} \equiv \text{diag}[\Delta E_i]$ are diagonal matrices and the probability matrix $\mathbf{P}$ is simply:

$$\mathbf{P}_{ij} \equiv \Delta P_X(X_i, E_j) = \frac{1}{\Delta E_j} \int_{\Delta E_j} dE \int_{\Delta X_j} dP_X(X, E). \tag{17}$$

The task of inverting $\mathbf{R}$ from Eq. (16): $\mathbf{R}^{-1} = \mathbf{E}^{-1}\mathbf{P}^{-1}\mathbf{X}$ is thus translated into that of inverting $\mathbf{P}$. The advantage of this approach is the unification of the numerical stability issues, regardless of the selection of the kinematic parameter X. If the reconstructed neutron energy ($X = \mathcal{E}$) is selected, then the relevant values of the fractional term from Eq. (15) are close to unity ($\Delta E_j / \Delta \mathcal{E}_i \approx 1$), making the average magnitude of $\mathbf{R}_{ij}$ values basically uniform over the entire range of E. On the other hand, when the neutron time of flight ($X = t$) is selected, there is a massive variation in magnitude if the data span the wide range in E, which is easily shown by taking advantage of the time–energy relation from Eq. (2):

$$\frac{\Delta E_j}{\Delta t_i} \approx \left| \frac{dt}{d\mathcal{E}} \right|_{\mathcal{E}=E_j}^{-1} = \frac{\sqrt{E_j (2 + E_j/mc^2)}^3}{L\sqrt{m}} \xrightarrow{E_j \ll mc^2} \frac{1}{L} \sqrt{\frac{8 E_j^3}{m}}. \tag{18}$$

Adopting $\mathbf{P}$ for the numerical treatment circumvents this inconsistency, since $\mathbf{P}_{ij}$ are the true probabilities, subject to the normalization $\sum_i \mathbf{P}_{ij} = 1$ for every j.

Having selected $\mathbf{P}$ in place of the initial $\mathbf{R}$, we may note that by inserting Eq. (16) into Eq. (14) one can make a transition from differential spectra $\vec{S}_E$ and $\vec{S}_X$ directly into the spectra of counts:

$$\begin{aligned} \vec{N}_E &\equiv \mathbf{E}\, \vec{S}_E \\ \vec{N}_X &\equiv \mathbf{X}\, \vec{S}_X. \end{aligned} \tag{19}$$

Evidently, the content of these vectors is just the amount of counts from each corresponding histogram bin: $[\vec{N}_E]_i = \Delta N_E(E_i)$ and $[\vec{N}_X]_i = \Delta N_X(X_i)$. Thus the central equation of interest becomes:

$$\vec{N}_X = \mathbf{P}\, \vec{N}_E \tag{20}$$

whose solution $\vec{N}_E$ we seek to compute and analyze.

Throughout the following sections we deal with many well established concepts in solving the linear system of equations. While the literature on the subject is vast, we point the reader to Refs. [14,15] for a highly comprehensive and practical overview of the subject.

3. Unfolding procedure

The potential challenges in solving Eq. (20) arise when we consider the size of the system we aim to handle. The neutron capture measurements from EAR1 often span the energy range from ~10 meV up to ~1 MeV, i.e. covering 8 orders of magnitude in energy. Considering that the binning used for these data often reaches 5000 bins per decade [16], we need to solve systems of the size $(4 \times 10^4) \times (4 \times 10^4)$, at the very least. Furthermore, the method should be of sufficient generality to make the transition between the spectra $\vec{N}_E$ and $\vec{N}_X$ not only of differently distributed bins, but also of the different number of bins, implying that $\mathbf{P}$ should not necessarily be a square matrix. In this case the so-called pseudoinverse of $\mathbf{P}$ may still be constructed and expressed as: $\mathbf{P}^{-1} = (\mathbf{P}^{\top}\mathbf{P})^{-1}\mathbf{P}^{\top}$, with $(\cdot)^{\top}$ standing for the matrix transposition. When applied, the pseudoinverse extracts the best solution to the system of equations, in a least-squares sense. However, this operation is only meaningful if the number of elements n_E in the solution vector $\vec{N}_E$ is not greater than the number of elements n_X in $\vec{N}_X$. In other words, if $\mathbf{P}$ is of the size $n_X \times n_E$, then $n_X \geq n_E$ should be satisfied in order for a unique solution $\vec{N}_E$ to exist [15].

Fig. 1 shows the resolution function for the EAR2 of the n_TOF facility, obtained by Geant4 simulations [9], which we have selected for presentation throughout this work. The top panel shows the probability matrix $\mathbf{P}$ as a function of an effective flight path length λ, which may be easily translated into the time of flight t by Eq. (3), or further into the reconstructed neutron energy $\mathcal{E}$ by Eq. (2). In presenting the graphical
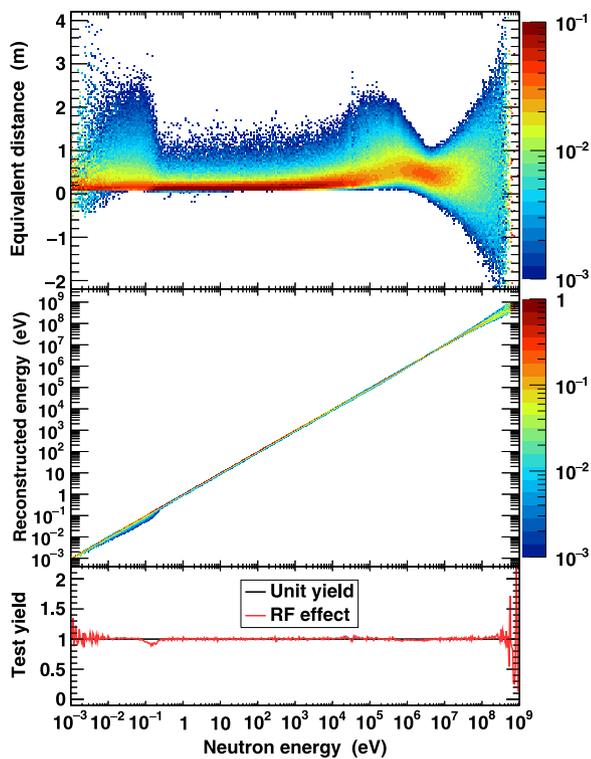
Fig. 1. Probability matrix **P**, representative of the resolution function for the EAR2 of the n_TOF facility. The numerical matrix has been adopted directly from the raw Geant4 simulations. Top panel: **P** as a function of an effective neutron flight inside the target-moderator assembly (equivalent distance, for short). Middle panel: **P** as function of the neutron energy reconstructed from the neutron time of flight. Bottom panel: the unit-spectrum folded by the adopted probability matrix.

examples from this work we show the reconstructed neutron energy, as this enables the direct comparison of the folded spectra with the ones from before and after unfolding. Thus, the middle panel from Fig. 1 shows the same probability matrix **P** as a function of $\mathcal{E}$. Alternatively, Fig. 13 from Ref. [2] presents an example of the (unnormalized) **P** as a function of the time of flight t (obtained, in that work, by FLUKA simulations). Finally, the bottom plot from Fig. 1 shows the effect of the adopted probability matrix when applied to the unit-spectrum. The rough shape of the folded spectrum reflects the fact that we have adopted the probability matrix directly from raw simulations, without much refinements. This serves our goals well, as we wish to investigate the numerical stability of the unfolding procedure even in case of a challenging **P**, which does not necessarily yield a smooth and "well-behaved" spectrum for unfolding.

Immediately evident from the middle panel of Fig. 1 (as well as Fig. 13 from Ref. [2]) is the banded structure of **P** — or, equivalently, of the resolution function matrix **R** — which we will take advantage of in order to increase the efficiency of the matrix computations and to reduce the matrix storage requirements. To this end we adopt the storage scheme from Appendix A.1, which we apply to all matrices referred to in this work. In addition, several important technical procedures need to be performed before solving the system of Eq. (20), all of which are described in Appendix A.2.

3.1. Solution to the system of equations

The most reliable method of solving the system from Eq. (20), in the sense of numerical stability and a prospect of success, is the Singular Value Decomposition (SVD) [14,15] of **P**. Moreover, SVD is directly applicable even when **P** is not square, which is the level of generality that we wish to maintain. However, for the systems of the size that

we aim to handle (of the order $10^4 \times 10^4$ or even $10^5 \times 10^5$) the SVD methods become prohibitively expensive to execute. Furthermore, while SVD may be used to directly construct the inverse $\mathbf{P}^{-1}$, for large systems this approach becomes of limited use. The main reason is that the inverse of banded matrices — bandedness being the feature of **P** that we heavily rely on to be able to store it efficiently — is dense in general, i.e. does not possess or even resemble the banded structure of the original matrix. Thus, significantly higher, even prohibitive memory resources would be needed just to store the inverse. Fortunately, there is a variety of alternative methods available for obtaining the solution of Eq. (20). Some entirely avoid the construction of any intermediate matrix, and some — if successful — construct intermediate matrices of favorable properties (preserving, for example, the banded structure), while utilizing far simpler computational schemes than SVD. The methods of the former class — requiring only the original matrix **P** — are usually iterative in nature and we reflect on them in Section 3.2. In this section, however, we focus on a method of the latter class, namely the Cholesky decomposition of a symmetric, positive definite matrix.

The system from Eq. (20) can always be reduced to a so-called system of normal equations by applying the transpose $\mathbf{P}^\top$, so that: $\mathbf{P}^\top \vec{N}_X = \mathbf{P}^\top \mathbf{P} \vec{N}_E$. Introducing the notation $\mathcal{P}$ for a new core-matrix, and $\vec{\mathcal{N}}_X$ for a modified spectrum of measured counts, the system to be solved takes the form:

$$\left. \begin{array}{l} \mathcal{P} \equiv \mathbf{P}^\top \mathbf{P} \\ \vec{\mathcal{N}}_X \equiv \mathbf{P}^\top \vec{N}_X \end{array} \right\} \quad \Rightarrow \quad \vec{\mathcal{N}}_X = \mathcal{P} \vec{N}_E. \tag{21}$$

The new core-matrix $\mathcal{P}$ is not only square, but also symmetric and positive definite, making it a prime candidate for a Cholesky decomposition. In addition, for banded **P**, $\mathcal{P}$ will also be banded, though with a wider bandwidth. One issue to take note of is that it is not usually recommended to substitute the system from Eq. (20) by the one from Eq. (21), as the matrix $\mathcal{P}$ has a greater condition number than **P**, bringing it closer to being ill-conditioned, thus making the decomposition and inversion procedures more susceptible to failure [14,15]. However, in light of our requirements (non-square matrix, the size of the system), not many alternatives remain available. Furthermore, the benefits of the Cholesky decomposition — if it can be successfully executed — are so overwhelming that any attempt at it is certainly justified. These benefits include: (1) extremely simple and efficient algorithm for performing the decomposition itself; (2) extremely efficient procedure for obtaining the solution to the linear system of equations from a decomposed matrix, by means of a forward and backward substitution; (3) the Cholesky decomposition inherits the banded structure of the decomposed matrix [14], not only further reducing the number of required operations, but also allowing for a compact storage of the decomposition result. If the decomposition is of sufficient numerical accuracy, the direct solution to Eq. (21) may be kept as the final result, or at least as a good initial guess for the iterative methods (Section 3.2), accelerating their convergence towards the true solution.

The Cholesky decomposition itself is of the form:

$$\mathcal{P} = \mathbf{L} \mathbf{L}^\top \tag{22}$$

with **L** as the lower triangular matrix. Employing the default algorithm [14,15], we indeed observe the failure of the decomposition when $\mathcal{P}$ reaches the size of approximately $10^3 \times 10^3$. The failure bears an unmistakable signature, as it is realized through the appearance of negative terms under the square roots involved in the calculation. However, even if the solution $\vec{N}_E$ to Eq. (21) is successfully found by means of the Cholesky decomposition, in the presence of the roundoff errors caused by the finite-precision arithmetic this solution does not solve exactly the starting Eq. (21), but rather the system $(\mathcal{P} + \delta\mathcal{P})\vec{N}_E$ where $\delta\mathcal{P}$ is a perturbation to $\mathcal{P}$ [17,18]. This implies that if the decomposition fails for $\mathcal{P}$, some perturbation $\delta\mathcal{P}$ could be manually added to it in order to promote it towards the closest machine-representable matrix for which the decomposition still succeeds. This matrix upgrade may go even further, to the point where the decomposition not only reconstructs
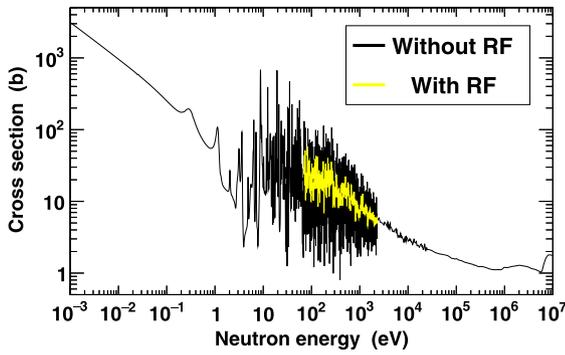
Fig. 2. Cross section for the ^{235}U(n,f) reaction, from the ENDF/B-VII.1 database. For visual purposes, the smoothing effect of the resolution function for the EAR2 of the n_TOF facility is displayed only within the dense resolved resonance region.

$\mathcal{P}$ as well as possible, but also yields the most accurate inverse $\mathcal{P}^{-1}$. In an attempt to promote the positive definiteness, we amplify the diagonal elements of $\mathcal{P}$ by a small adjustable factor ϵ, meaning that $\delta \mathcal{P} = \mathrm{diag}[\epsilon \, \mathcal{P}_{ii}]$. This modification may be directly implemented into the equations for calculating the elements of the Cholesky factor $\mathbf{L}$:

$$
\begin{aligned}
\mathbf{L}_{ii} &= \sqrt{\mathcal{P}_{ii}(1+\epsilon) - \sum_{k=0}^{i-1} \mathbf{L}_{ik}^2} \\
\mathbf{L}_{ij} &= \frac{1}{\mathbf{L}_{ii}} \left(\mathcal{P}_{ij} - \sum_{k=0}^{i-1} \mathbf{L}_{ik}\mathbf{L}_{jk} \right).
\end{aligned}
\tag{23}
$$

Indeed, this practice stabilizes the decomposition of $\mathcal{P}$ even for the values of ϵ close to the limit of the numerical precision ($\epsilon \approx 10^{-6}$ for type *float* and $\epsilon \approx 10^{-15}$ for type *double* from C++). However, for the matrices of the size up to $10^5 \times 10^5$ (and for our particular resolution function) we found the optimal value of $\epsilon = 10^{-4}$. In principle, this factor depends on the particular binning of the resolution function matrix. But if a unique value is sought, the optimization is best performed using a matrix as large as possible, since the smaller matrices not only suffer less from the stability issues but are also less sensitive to the value of ϵ. As suggested earlier, the method of optimization relies on observing the quality of the solution to Eq. (21), which visibly degrades for values of ϵ sufficiently far from the optimal one.

Fig. 2 shows the cross section of the ^{235}U(n,f) reaction, adopted from the ENDF/B-VII.1 database [19], that we have selected for presenting the results of the unfolding procedure. This cross section features all the common characteristics of a neutron reaction cross section — a smooth $1/\sqrt{E}$ thermal region, several wide low-energy resonances, a densely populated resolved resonance region and, finally, the unresolved resonance region. Therefore, this particular cross section will allow us to observe the response of the unfolding procedure to all the numerical conditions that may be expected from a typical neutron reaction dataset. In addition, Fig. 2 also shows (for visual purposes only) the portion of the folded spectrum inside the dense resolved resonance region, in order to emphasize the extreme smoothing effect that the resolution function commonly exhibits within this energy range.

We have applied the unfolding procedure to the spectra of maximum size of 10^5 bins (10^4 bins per decade within the energy range from 1 meV to 10 MeV), which meant using the probability matrix $\mathbf{P}$ of size $10^5 \times 10^5$. Fig. 3 shows the results for the ^{235}U(n,f) spectrum of this size. In order to reduce the visible dataset, in displaying the results we have kept only one in every ten points. As opposed to rebinning the dataset, which leads to a decrease in the relative fluctuations in the unfolded solution, rejecting not too large a portion of the points faithfully preserves the magnitude of these fluctuations (inasmuch as the remaining sample of points is representative of the entire population). The four plots from Fig. 3 focus on the smooth $1/\sqrt{E}$ range, wide low-energy resonances, densely populated resolved resonance region and

the unresolved resonance region. Somewhat surprisingly, the unfolding procedure seems to suffer most in the smoothest parts of the spectrum, yielding the greatest fluctuations around the true solution (shown separately by the respective bottom panels). However, taking into account the size of the system, we consider the quality of the unfolded solution to be commendable. It is worth noting that, implemented in C++, the entire unfolding procedure — which is by far dominated by the Cholesky decomposition — takes 15 min for the system of size $10^5 \times 10^5$, on a single core of an Intel Core i5-6500 3.2 GHz processor. On the other hand, the system of size $(5 \times 10^4) \times (5 \times 10^4)$ requires less than 2 min; the system of $(2 \times 10^4) \times (2 \times 10^4)$ takes 5 s, while the system of $10^4 \times 10^4$ is handled under a second.

3.2. Iterative methods

As an alternative to the direct methods of solving the linear system from Eq. (21) — or out of necessity if none of them could be made to work — one could always consider some of the iterative procedures. First among those is the *conjugate gradient method*, requiring that the matrix be symmetric and positive definite [14,15], which is a condition met by $\mathcal{P}$. If (and only if) successful, for the system of size $n_E \times n_E$ the method is supposed to converge to an exact solution in at most n_E iterations. However, under the finite precision arithmetics the success of the method may strongly depend on the initial guess for the solution. In practical situations, this guess can hardly be anything other than $\vec{\mathcal{N}}_X$ from the left side of Eq. (21). Unfortunately, we have observed that in our case the method starts diverging severely from the expected solution already for the system size of approximately 350×350. As such, we do not recommend it for solving this particular problem.

The next method worth considering is the *successive over-relaxation* [14,15]. For the system $\mathcal{P}\vec{x} = \vec{y}$ under consideration, its kth iterative step may be expressed in a closed-form as:

$$
x_i^{(k)} = (1-\omega)x_i^{(k-1)} + \frac{\omega}{\mathcal{P}_{ii}} \left(y_i - \sum_{j<i} \mathcal{P}_{ij} x_j^{(k)} - \sum_{j>i} \mathcal{P}_{ij} x_j^{(k-1)} \right)
\tag{24}
$$

with ω as the relaxation parameter such that $0 < \omega < 2$. For a particular value of $\omega = 1$ the method reduces to the so called *Gauss–Seidel method*. The benefit of values different than unity may be the increased rate of convergence. However, the optimal value for ω strongly depends on the particular system being solved. We have found that the over-relaxation method works well in solving Eq. (21) even for the systems of the largest considered size ($10^5 \times 10^5$), thus it may be recommended for its success in unfolding the resolution function.

In practical applications no *a priori* knowledge of the expected solution is available. For this reason, at every point we define a convergence criterion as the relative change r_i between successive iterations:

$$
r_i \equiv 2 \left| \frac{x_i^k - x_i^{(k-1)}}{x_i^k + x_i^{(k-1)}} \right|
\tag{25}
$$

and follow their evolution. Whenever the denominator happens to be 0, the corresponding r_i are simply ignored. Requiring that all r_i drop below a preset limit may be prohibitive, as there are always some sporadic far-away values to be expected. Alternatively, one could always require that some preset portion, e.g. 90% of their distribution drops below a certain threshold. On the other hand, for a basically exponential-like shape of the distribution with a longer tail, we found that a good estimator of the relevant distribution range is $7\mu\sigma$, with μ as the mean value of r_i values (excluding ignored ones) and σ as their root mean square.

Despite the general success of the successive over-relaxation in unfolding the resolution function, we have found its rate of convergence to be rather slow. A carefully selected value of ω may, indeed, speed up the convergence rate. However, there is no unique value to be recommended, as it depends even on the number of bins in the analyzed spectra, i.e. on the size of the constructed $\mathcal{P}$ matrix. Furthermore, we have observed the following: after reaching the same value of the convergence criterion using different values of ω, the final iterative
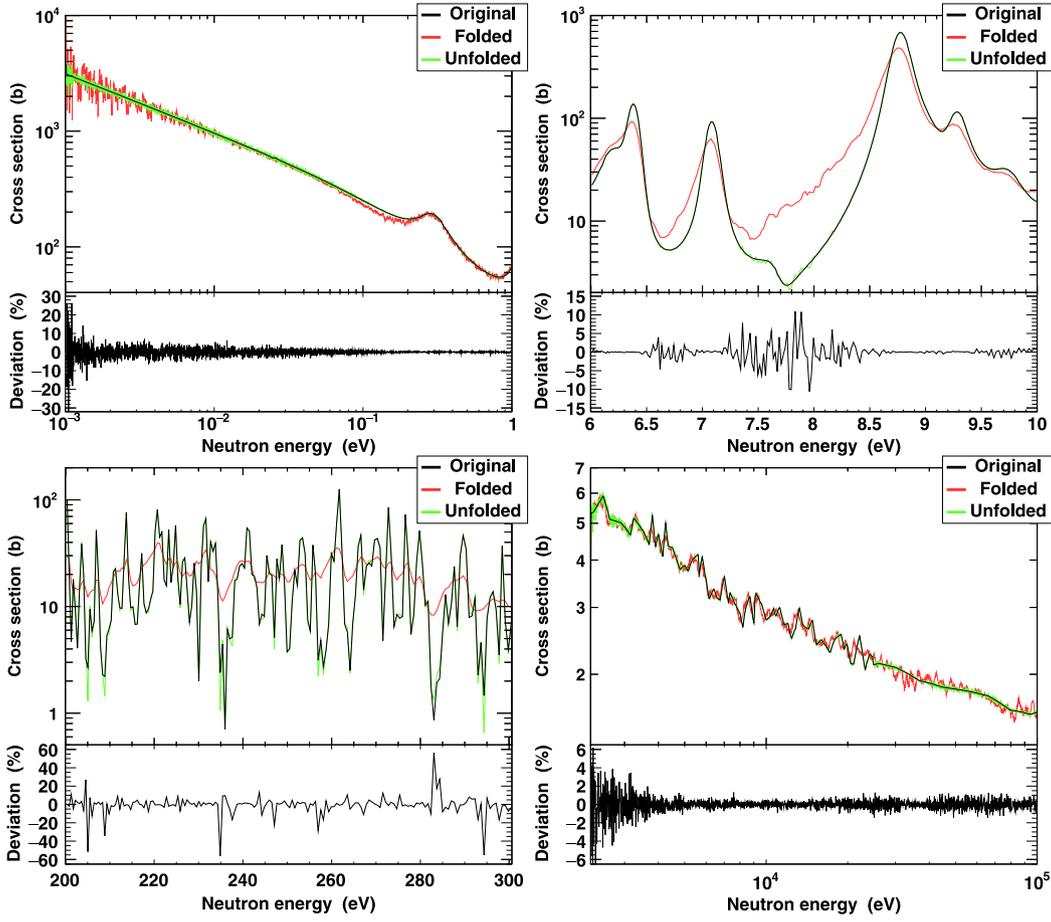
Fig. 3. Results of the resolution function unfolding procedure applied to the ^{235}U(n,f) cross section, within different representative energy regions. They are compared both to the spectrum folded by the resolution function for the EAR2 of the n_TOF facility, and the true solution. Isolated subplots show the relative fluctuations of the unfolded spectrum around the true solution.

results are sometimes at visibly different levels of agreement with the expected solution. This difference is manifested through the increased or reduced residual fluctuations around the expected solution, but the quality of the particular solution does not seem to bear any correlation with the optimality of ω (in the sense of the convergence rate).

One can, of course, use the solution obtained by the Cholesky decomposition from Section 3.1 as a starting point for any of the iterative procedures. If this option is available, i.e. if $\mathcal{P}$ was, indeed, successfully decomposed so as to at least produce a numerically stable approximation to the solution of Eq. (21), then another type of an iterative improvement also becomes available. For a system $\mathcal{P}\vec{x} = \vec{y}$ this improvement consists of a repeated application of the following procedure [14,15]:

$$\vec{x}_{\text{new}} = \vec{x}_{\text{old}} - \mathcal{P}^{-1}\left(\mathcal{P}\vec{x}_{\text{old}} - \vec{y}\right) \tag{26}$$

as many times as necessary, but rarely more than once. Eq. (26) is not to be interpreted as an algebraic identity yielding $\vec{x}_{\text{new}} = \mathcal{P}^{-1}\vec{y}$ at every step, but as a literal instruction for the numerical evaluation of the right-hand side. In that, the term $\delta\vec{x} = \mathcal{P}^{-1}\left(\mathcal{P}\vec{x}_{\text{old}} - \vec{y}\right)$ should be calculated in the same way as the initial solution, i.e. by the forward and backward substitutions, taking advantage of already completed decomposition of $\mathcal{P}$ in order to extract the solution of the system $\mathcal{P}\delta\vec{x} = \mathcal{P}\vec{x}_{\text{old}} - \vec{y}$ without explicitly constructing the inverse $\mathcal{P}^{-1}$. Having said that, we have not found any significant improvement to the initial solution obtained directly from the Cholesky decomposition, using any combination of the iterative procedures. Therefore, we can wholeheartedly recommend the standalone application of the procedure from Section 3.1 as selfsufficient.

4. Uncertainty propagation

The propagation of uncertainties is an important issue to be addressed in any kind of direct unfolding procedures. From the formal solution $\vec{N}_E = \mathbf{P}^{-1}\vec{N}_X$ to Eq. (20), it directly follows that the uncertainty $\sigma_i^{(E)} \equiv [\vec{\sigma}^{(E)}]_i$ of the ith component of the reconstructed solution equals:

$$\sigma_i^{(E)} = \sqrt{\sum_j \left(\mathbf{P}_{ij}^{-1}\sigma_j^{(X)}\right)^2} \tag{27}$$

with $\sigma_j^{(X)} \equiv [\vec{\sigma}^{(X)}]_j$ as the uncertainty of the jth component in $\vec{N}_X$, together with the following notation: $\mathbf{P}_{ij}^{-1} \equiv [\mathbf{P}^{-1}]_{ij}$. The difficulty with evaluating Eq. (27) lies in the numerical efficiency because $\vec{\sigma}^{(E)}$ *cannot* be expressed solely as a function of $\mathbf{P}^{-1}\vec{\sigma}^{(X)}$, which we *are* able to calculate efficiently by taking advantage of the Cholesky decomposition. In order to avoid an explicit calculation of $\mathbf{P}^{-1}$ — just as we were striving to do up to this point — we use Eq. (27) only as a starting point for developing the procedure that *can* take advantage of the already performed decomposition. We first recognize that $\mathbf{P}^{-1} = \mathcal{P}^{-1}\mathbf{P}^{\top}$, which leads to:

$$\sigma_i^{(E)} = \sqrt{\sum_j \left(\sum_k \mathcal{P}_{ik}^{-1}\mathbf{P}_{kj}^{\top}\sigma_j^{(X)}\right)^2}. \tag{28}$$

The benefit of defining the sequence of vectors $\vec{\chi}^{(j)}$:

$$[\vec{\chi}^{(j)}]_k \equiv \mathbf{P}_{kj}^{\top}\sigma_j^{(X)} \tag{29}$$

soon becomes evident, as this formal manipulation allows us to write:

$$\sigma_i^{(E)} = \sqrt{\sum_j \left([\boldsymbol{P}^{-1} \vec{\chi}^{(j)}]_i \right)^2} = \sqrt{\sum_j \left([\vec{\xi}^{(j)}]_i \right)^2} \tag{30}$$

where we have immediately introduced an additional sequence of vectors $\vec{\xi}^{(j)}$, each of them to be found as a solution to the equation:

$$\boldsymbol{P} \vec{\xi}^{(j)} = \vec{\chi}^{(j)}. \tag{31}$$

By again taking advantage of the already performed Cholesky decomposition of $\boldsymbol{P}$, all $\vec{\xi}^{(j)}$ may be found just by using forward and backward substitutions, without ever explicitly constructing $\boldsymbol{P}^{-1}$. In practical implementations $\vec{\xi}^{(j)}$ should not be *all* computed first and only then Eq. (30) be evaluated. Such procedure would require a memory of size $n_X \times n_E$ (n_X vectors, each with n_E components), equivalent to storing the entire $\mathbf{P}$ matrix, thus defeating one of the main benefits of introducing the storage scheme from Appendix A.1. Instead, each $\vec{\xi}^{(j)}$ should be immediately used for incrementing all n_E sums (one for every $\sigma_i^{(E)}$), allowing to store only one $\vec{\xi}^{(j)}$ at a time.

Unfortunately, this exact procedure is of $\mathcal{O}(n_X \times n_E \times B_{\boldsymbol{P}})$ computational complexity — $\mathcal{O}(n_E \times B_{\boldsymbol{P}})$ being the cost of solving each of the n_X systems from Eq. (31), with $B_{\boldsymbol{P}}$ as the bandwidth of $\boldsymbol{P}$ — and rapidly becomes prohibitive even for systems of modest size. Luckily, the exact procedure may be easily and reliably replaced by a rudimentary numerical simulation. Assuming that the original uncertainties are purely due to statistical fluctuations in the number of detected counts, a formally correct procedure would be to randomly generate across the whole spectrum a preselected number of K sets of deviations $\vec{\Delta}_X^{(k)}$ from the measured number of counts ($k \in [1, K]$), according to a Poisson distribution:

$$[\vec{\Delta}_X^{(k)}]_i = \text{Poisson}\left(\mu = [\vec{N}_X]_i \right) - [\vec{N}_X]_i. \tag{32}$$

However, sampling the Gaussian distribution may be simpler and also works perfectly well:

$$[\vec{\Delta}_X^{(k)}]_i = \text{Gauss}\left(\mu = 0; \sigma^2 = [\vec{N}_X]_i \right). \tag{33}$$

For each $k \in [1, K]$ a corresponding system:

$$\boldsymbol{P} \vec{\Delta}_E^{(k)} = \mathbf{P}^\top \vec{\Delta}_X^{(k)} \tag{34}$$

is to be solved for $\vec{\Delta}_E^{(k)}$ and the final set of uncertainties across the unfolded spectrum is to be calculated as:

$$\sigma_i^{(E)} = \sqrt{\frac{1}{K} \sum_{k=1}^{K} \left([\vec{\Delta}_E^{(k)}]_i \right)^2}. \tag{35}$$

The computational complexity of this procedure is $\mathcal{O}(K \times n_E \times B_{\boldsymbol{P}})$, a significant improvement when the size n_X of the initial dataset is large. Though one should not settle for less than $K = 100$ iterations, this value already yields a quite satisfactory level of precision in estimating the uncertainties that would otherwise be obtained by the exact procedure from Eq. (30).

In order to precisely evaluate severity by which the unfolding procedure affects the uncertainties, they need to be compared to the original uncertainties before unfolding. Here we need to take into account that the unfolded spectrum is always a function of the (true kinetic) neutron energy E. Therefore, a direct comparison between the uncertainties before and after unfolding can only be performed if the original spectrum has also been constructed as a function of neutron energy (in this case the reconstructed energy $\mathcal{E}$) and with the identical binning as the final unfolded spectrum $\vec{\sigma}^{(E)}$, regardless of the kinematic parameter X and of the binning otherwise used for displaying the data before unfolding. Using the notation $\vec{\sigma}^{(\mathcal{E}=E)}$ for thus constructed set of uncertainties, the relative amplification ρ_i of the uncertainties across the spectrum is:

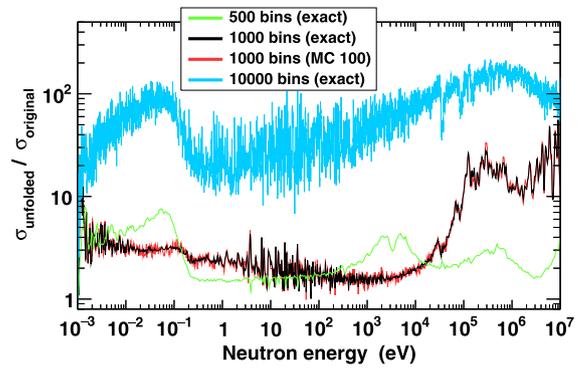$$\rho_i \equiv \frac{\sigma_i^{(E)}}{\sigma_i^{(\mathcal{E}=E)}}. \tag{36}$$

Fig. 4. Noise amplification induced by the direct unfolding procedure applied to the ^{235}U(n,f) cross section. For the spectrum of 1000 bins in total, simulated (MC: Monte Carlo) results obtained with 100 iterations are shown alongside the exact uncertainties.

For a given binning, and as long as the original uncertainties are purely statistical (due to the fluctuations in the number of detected counts), these bin-wise amplification factors are insensitive to the total integrated number of counts, making them robust estimators of the effect of the unfolding procedure.

Fig. 4 shows the uncertainty amplification for the ^{235}U(n,f) cross section from Figs. 2 and 3, for three different binning densities: 50, 100 and 1000 bins per decade, i.e. 500, 1000 and 10 000 bins in total. For the total of 1000 bins, the simulated results are also shown, obtained using $K = 100$ iterations in the context of Eq. (35). Unfortunately, this increase in the uncertainty is not caused by the numerical stability, which could be improved by a more sophisticated algorithm. Rather, it is the general and inherent feature of the direct unfolding methods, i.e. of the direct inversion procedures. Therefore, while the method may work impressively for the spectrum entirely devoid of noise — as demonstrated by Fig. 3 — it can be meaningfully applied to the real-world data only if the noise is still at or below some acceptable level, even after amplification.

5. Conclusions

We have explored a direct method of unfolding the resolution function from measurements of neutron induced reactions. The basic principle behind the method is straightforward and consists of solving a large linear set of equations, akin to a direct inversion of the resolution function matrix (RFM). The main concerns of the method are the numerical stability and the computational efficiency, since the systems of interest may become quite large, bringing the RFM closer and closer to being ill-conditioned. For large systems the computational efficiency requirement disqualifies some of the methods that would, in principle, be successful at solving the numerical stability issue, such as the Singular Value Decomposition. Fortunately, one may take advantage of the narrow banded structure of the RFM and combine the specialized matrix storage scheme, presented in this paper, with the Cholesky decomposition of the modified RFM in order to arrive at an efficient algorithm that was successfully applied to a system of size $10^5 \times 10^5$. However, a small modification of the large ill-conditioned RFM must be carried out in advance of the Cholesky decomposition, if the procedure is even to succeed, let alone be numerically stable during the inversion process. The modification consists of reinforcing the positive definiteness of the symmetric matrix $\boldsymbol{P}$, introduced in Eq. (21). There is more than a single unique choice for doing this, and we have opted for a simple amplification of diagonal elements by a multiplicative factor $1 + \epsilon$, using an optimal value of $\epsilon = 10^{-4}$.

We have also explored the amenability of the problem to the iterative, rather than direct procedures. While the conjugate gradient method fails to properly reconstruct the solution already in the case

of more than 350 equations, the successive overrelaxation method successfully handles the task, although at the price of a somewhat slow convergence. One could always use the solution obtained by the Cholesky decomposition as a starting point for any of the iterative methods. However, we have found this entirely unnecessary, due to the quality of the Cholesky solution already at the level where little could be gained from an iterative improvement.

A critical issue to be addressed regarding the direct unfolding methods is the propagation of uncertainties into the final solution. Unfortunately, the high sensitivity to the noise in the measured data is an inherent feature of such methods, as the noise is heavily amplified during the inversion procedure. Therefore, the level of the noise ultimately dictates if the procedure can be meaningfully applied. However, if the method *is* to be applied, the expected increase in the noise can be exactly evaluated (given the known or expected distribution of uncertainties across the measured spectrum), thus allowing to determine in advance the acceptable level of the noise in the measured data.

Acknowledgments

Appendix A. Matrix manipulations

A.1. Matrix storage

We will describe the storage scheme for banded matrices, which we apply to all matrices referred to in this work. Let $\mathbf{M}$ be the banded matrix of size $\mathcal{I} \times \mathcal{J}$. We will only store the consecutive sequence of elements from each column, outside of which all elements are 0. To this purpose we arrange the matrix content column-by-column into a one-dimensional array M, followed by the supporting arrays $i_{\min}$ and $i_{\max}$ containing the indices of the first and the last stored term from a given (jth) column. In this sense $\mathbf{M}_{i_{\min}[j],j}$ and $\mathbf{M}_{i_{\max}[j],j}$ are the first and the last nonzero elements from the jth column. While only these three arrays are necessary for a unique and unambiguous storage, it is extremely convenient to also maintain the arrays $j_{\min}$ and $j_{\max}$, containing the indices of the first and the last nonzero element from a given (ith) row. The main reason is that this practice allows for a significant speedup in the multiplication of banded matrices:

$$(\mathbf{AB})_{ij} = \sum_{k=\max\left\{j_{\min}^{(\mathbf{A})}[i],\, i_{\min}^{(\mathbf{B})}[j]\right\}}^{\min\left\{j_{\max}^{(\mathbf{A})}[i],\, i_{\max}^{(\mathbf{B})}[j]\right\}} \mathbf{A}_{ik}\mathbf{B}_{kj} \tag{A.1}$$

thus avoiding to loop over the off-band portions of either $\mathbf{A}$ or $\mathbf{B}$, where the product $\mathbf{A}_{ik}\mathbf{B}_{kj}$ is always 0.

A quick access to an arbitrary matrix element $\mathbf{M}_{ij}$ is facilitated by maintaining an additional array I, containing the indices of the positions in the array M itself, where the content of a new column starts. In this case the efficient access is achieved as:

$$\mathbf{M}_{ij} = \begin{cases} M\left[I[j] + i - i_{\min}[j]\right] & \text{if } i_{\min}[j] \le i \le i_{\max}[j] \\ 0 & \text{otherwise.} \end{cases} \tag{A.2}$$

Evidently, one may store the matrix row-by-row, in which case an array I needs to be replaced by a completely analogous array J, containing the indices of the positions within M where the content of a new row starts. Hence, the access takes the form:

$$\mathbf{M}_{ij} = \begin{cases} M\left[J[i] + j - j_{\min}[i]\right] & \text{if } j_{\min}[i] \le j \le j_{\max}[i] \\ 0 & \text{otherwise.} \end{cases} \tag{A.3}$$

However, the column-by-column storage lends itself more naturally to the procedure ahead. It is worth noting that both pairs of index arrays — $i_{\min}$ with $i_{\max}$ and $j_{\min}$ with $j_{\max}$ — may be simultaneously updated while the matrix $\mathbf{M}$ is being constructed, regardless of the selected storage

scheme. This fact may lead to the improvement in the computational efficiency when the construction of the next matrix element depends on the elements constructed up to that point, as during the various matrix decomposition operations.

In this work we adopt the convention of zero-offsetting all the array indices (implying that the first matrix term is $\mathbf{M}_{00} \leftrightarrow M[0]$). Recalling that the matrix $\mathbf{M}$ is of the size $\mathcal{I} \times \mathcal{J}$, we note the following: while the arrays $i_{\min}$ and $i_{\max}$ contain $\mathcal{J}$ elements each, their range of values spans between 0 and $\mathcal{I} - 1$. Similarly, while the arrays $j_{\min}$ and $j_{\max}$ contain $\mathcal{I}$ elements, their values are bounded between 0 and $\mathcal{J} - 1$. In the most general case, the only concern to be addressed is the arbitrary handling of the empty, i.e. all-zero rows or columns.

The very definition of the arrays I and J:

$$\begin{aligned} I[j] &= I[j-1] + \left(i_{\max}[j-1] - i_{\min}[j-1] + 1\right) \\ J[i] &= J[i-1] + \left(j_{\max}[i-1] - j_{\min}[i-1] + 1\right) \end{aligned} \tag{A.4}$$

with $I[0] = J[0] = 0$, is also computationally the most efficient procedure. Due to the adopted zero-offset convention, the final terms in these arrays are $I[\mathcal{J} - 1]$ and $J[\mathcal{I} - 1]$. However, in the sense of the previous definitions, the terms $I[\mathcal{J}]$ and $J[\mathcal{I}]$:

$$\begin{aligned} I[\mathcal{J}] &= \sum_{k=0}^{\mathcal{J}-1}\left(i_{\max}[k] - i_{\min}[k] + 1\right) \\ J[\mathcal{I}] &= \sum_{k=0}^{\mathcal{I}-1}\left(j_{\max}[k] - j_{\min}[k] + 1\right) \end{aligned} \tag{A.5}$$

determine the total number of the elements stored in the array M, not all of which are necessarily non-zeros. It should be noted that in general, between the two storage schemes — either by the columns or the rows — the number of stored elements is not the same: $I[\mathcal{J}] \ne J[\mathcal{I}]$, which is easily proven by considering the minimal matrix example $\mathbf{M} = [\emptyset\ 0\ \emptyset]$. If no dummy elements are used to fill-in all-zero columns or rows within the array M, here it holds: $I[\mathcal{J} = 3] = 2$ and $J[\mathcal{I} = 1] = 3$.

A.2. Matrix preparation

Before finding the solution to Eq. (20), several technical procedures need to be performed:

(1) construction of an appropriate matrix $\mathbf{P}$
(2) reduction of $\mathbf{P}$
(3) localization of the relevant invertible subportion of $\mathbf{P}$
(4) normalization of $\mathbf{P}$.

We address these procedures one by one.

Constructing P. The only reliable way to determine the resolution function over the wide range of energies covered at n_TOF is by dedicated simulations of the neutron production and transport process. The initial form $\mathbf{P}_0$ of the probability matrix is then obtained by directly histogramming the simulated data. However, this procedure cannot be relied on in case of an arbitrarily high binning, as the amount of statistics acquired from the simulations is limited. Therefore, for each specific binning the matrix $\mathbf{P}$ should be constructed by an appropriate interpolation, starting from an optimally binned $\mathbf{P}_0$. Regardless of the selection of the kinematic parameter $X \in \{t, \mathcal{E}, \lambda\}$ used for expressing the initial $\mathbf{P}_0$, the transition into the selected functional dependence of the final $\mathbf{P}$ may be efficiently performed in a course of a single interpolation process, without the need for constructing any intermediate forms of either $\mathbf{P}_0$ or $\mathbf{P}$.

Reducing P. When the data from simulations are histogrammed, there will always be few sporadic counts at the tails of the distribution (in the direction of the kinematic parameter X), which unnecessarily increase the column-wise bandwidth (the span between $i_{\min}[j]$ and $i_{\max}[j]$), but do not contribute in any significant way to the total resolution function. For this reason we recommend to reduce the stored content of $\mathbf{P}$ by cutting away the edges of the initially constructed matrix. This procedure is even more important if $\mathbf{P}$ was initially constructed
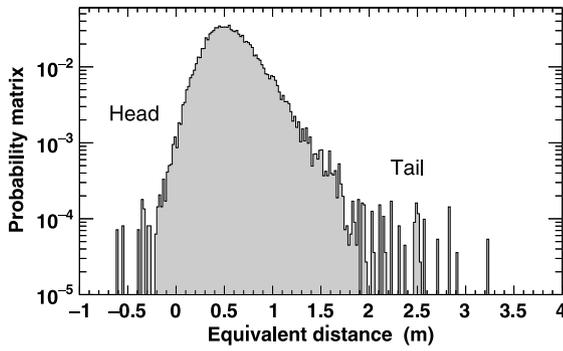
Fig. A.5. Example of the slice through the probability matrix for the EAR2 of the n_TOF facility, at the neutron kinetic energy of $E = 2$ MeV. The full probability matrix is shown by the top panel from Fig. 1.

not by directly histogramming the simulated data, but by evaluating the global analytical fit to the simulated data, such as the one proposed in Ref. [1] for modeling the n_TOF resolution function. In this case the analytical form may be evaluated at every point throughout the entire **P** matrix, destroying its banded structure and defeating any benefit of the storage scheme from Appendix A.1. Therefore, we keep only the most relevant portion of the resolution function. Some consideration is required, though, as **P** is regularly asymmetric along the X-direction. For $X = t$ the probability distribution has a long tail in the direction of longer flight times, while for $X = \mathcal{E}$ the tail extends towards the lower reconstructed energies. However, neither is the head of the distribution sharp, just more quickly decreasing than the tail. Therefore, the distribution needs to be cut at both sides, but in a way that the reduction at the tail is given precedence. For the clarity of terminology, Fig. A.5 illustrates the difference between the head and the tail of the distribution. Though the reduction is to be performed either in the time of flight or the reconstructed energy spectrum, Fig. A.5 shows the distribution as a function of the effective neutron flight path.

For the matrix reduction we employ the following simple algorithm. Let δ be the maximum allowed portion of probability to be discarded from each column of **P** (i.e. for each value of the true kinetic energy E). For each column let: $\Sigma_{\text{total}} = \sum_i \mathbf{P}_{ij}$ (at this point the columns of **P** have not yet necessarily been normalized to unity). Then for each column do the following:

(1) keep discarding the elements from the head of the distribution as long as the sum Σ_{head} of the discarded elements is lower than the half of the assigned portion: $\Sigma_{\text{head}} < (\delta/2) \times \Sigma_{\text{tot}}$

(2) keep discarding the elements from the tail as long as the total sum of discarded elements does not exceed the assigned portion: $\Sigma_{\text{head}} + \Sigma_{\text{tail}} \leq \delta \times \Sigma_{\text{tot}}$

(3) continue discarding the elements from the head as long as the assigned portion is not exceeded: $\Sigma_{\text{head}} + \Sigma_{\text{tail}} \leq \delta \times \Sigma_{\text{tot}}$.

At the end of the procedure it is possible that the greater contribution will be discarded from the head than from the tail of the distribution ($\Sigma_{\text{head}} > \Sigma_{\text{tail}}$). However, the tail is given a greater chance of dominating the total discarded content. For this work we have selected a value of $\delta = 1\%$. Note that this procedure may be performed during the column-by-column construction of **P**, meaning that one does not need to construct the entire matrix first and cut it afterwards, which enables the construction of matrices that might not fit the available memory resources before cutting. This possibility is greatly facilitated by the storage scheme from Appendix A.1.

Localizing P. One needs to ensure that **P** does not span the range outside the available resolution function data. Even a single row or column in **P** completely composed of zeros implies that **P** is singular,

hence uninvertible. In order to find the relevant invertible subportion of **P**, we first determine the "center of mass" coordinates $\bar{\imath}$ and $\bar{\jmath}$[3]:

$$(\bar{\imath}, \bar{\jmath}) = \frac{\sum_{i,j}(i,j) \times \mathbf{P}_{ij}}{\sum_{i,j} \mathbf{P}_{ij}}. \tag{A.6}$$

Starting from the position $(\bar{\imath}, \bar{\jmath})$, we search for the nearest rows and columns containing all zeros. Supposing that such rows and/or columns exist, we reduce **P** in size so that all the content beyond these rows and columns is discarded. The procedure needs to be repeated until no such rows/columns exist any more, since reducing the matrix in such a way may introduce new all-zero rows/columns when there were none before. This happens when the only non-zero entries from these rows/columns have all been discarded by the last iteration of the matrix localization. It should be noted that this procedure may also be efficiently performed in-place just by shifting the matrix content, without constructing any intermediate containers. This is again greatly facilitated by the storage scheme from Appendix A.1.

Normalizing P. Only at this point should the probability normalization ($\sum_i \mathbf{P}_{ij} = 1$ for every j) be performed. Otherwise, the total sum of probabilities in each column would be reduced by the portion of the discarded content (contributed both by the rejection factor δ and the portions of **P** discarded during the localization process), thus implying the nonconservation of counts when **P** or, conversely, $\mathbf{P}^{-1}$ is applied.

Appendix B. Multiple scattering effects

In the presence of the pronounced multiple scattering effects the direct resolution function unfolding cannot be meaningfully applied, due to their coupling not being multiplicatively separable. The inconvenience in obtaining the full and detailed parameterization of the multiple scattering effects stems from their dependence on every single sample — its cross section, shape, size, mass and material homogeneity — unlike the resolution function, which is uniquely determined by the neutron production facility.

In order to demonstrate this fact, let us consider the total number of detected counts $dN_E(E)$ caused by the neutrons of energy E. While they are expected to have been measured with the time of flight $t_E = L/v_E$, v_E being the neutron speed defined by Eq. (3), they are first delayed or advanced in time by τ_R due to the resolution function $R_t(t_E + \tau_R, E)$, only to be further offset by τ_M due to the multiple scattering effect $M_t(t_E + \tau_M, E)$. Therefore the total number of $d^3N(t, E)$ of counts detected with the time of flight t:

$$t = t_E + \tau_R + \tau_M \tag{B.1}$$

but offset specifically by τ_R and τ_M equals:

$$d^3N(t, E) = R_t(t_E + \tau_R, E) \times M_t(t_E + \tau_M, E) \times d\tau_R d\tau_M dN_E(E). \tag{B.2}$$

Adopting the substitution:

$$\left.\begin{array}{l} T_R \equiv t_E + \tau_R \\ T_M \equiv t_E + \tau_M \end{array}\right\} \quad \Rightarrow \quad t = T_R + T_M - t_E \tag{B.3}$$

and transitioning from T_M to t, Eq. (B.2) becomes:

$$d^3N(t, E) = R_t(T_R, E) \times M_t(t + t_E - T_R, E) \times dT_R dt \, dN_E(E). \tag{B.4}$$

Finally, to remove the experimentally indistinguishable sensitivity to the pure resolution function effect, separate from the multiple scattering contribution to the measured time of flight t, an integration over T_R needs to be performed:

$$d^2N(t, E) = dt \, dN_E(E) \int_{-\infty}^{\infty} R_t(T_R, E) \times M_t(t + t_E - T_R, E) \times dT_R \tag{B.5}$$

---

[3] If **P** were already normalized at this point, one might be tempted to reduce the denominator to $\sum_{i,j} \mathbf{P}_{ij} = n_E$ and even to obtain $\sum_{i,j} j \times \mathbf{P}_{i,j} = (n_E - 1)/2$. Alas, this does not work precisely when it matters: when the column is completely filled with zeros $\sum_i \mathbf{P}_{ij} = 1$ does not hold any more for a given j.

from where it is evident that in place of the resolution function from Eq. (9), a joint effect $Q_t(t, E)$ remains:

$$Q_t(t, E) \equiv \int_{-\infty}^{\infty} R_t(T_R, E) \times M_t(t + t_E - T_R, E) \times \mathrm{d}T_R. \quad (B.6)$$

Since the form is multiplicatively inseparable ($\mathbf{Q} \neq \mathbf{RM}$ for corresponding matrices), the resolution function cannot be directly separated from the measurements, without having a full parameterization of the multiple scattering effects at hand.

References

[1] C. Guerrero, A. Tsinganis, E. Berthoumieux, et al., Eur. Phys. J. A 49 (2013) 27.
[2] C. Weiß, E. Chiaveri, S. Girod, et al., Nucl. Instrum. Methods Phys. Res. A 799 (2015) 90.
[3] N. Colonna, E. Chiaveri, F. Gunsing, Nucl. Phys. News 25 (4) (2015) 19.
[4] S. Barros, I. Bergström, V. Vlachoudis, C. Weiß, J. Instrum. 10 (2015) P09003.
[5] C. Rubbia, S. Andriamonje, D. Bouvet-Bensimon, et al., A high resolution spallation driven facility at the CERN-PS to measure neutron cross sections in the interval from 1 eV to 250 MeV, CERN/LHC/98-02 and CERN/LHC/98-02-Add.1, 1998.
[6] C. Coceva, M. Frisoni, M. Magnani, A. Mengoni, Nucl. Instrum. Methods Phys. Res. A 489 (2002) 346.
[7] C. Borcea, P. Cennini, M. Dahlfors, et al., Nucl. Instrum. Methods Phys. Res. A 513 (2003) 524.
[8] S. Lo Meo, M.A. Cortés-Giraldo, C. Massimi, et al., Eur. Phys. J. A 51 (2015) 160.
[9] J. Lerendegui-Marco, S. Lo Meo, C. Guerrero, et al., Eur. Phys. J. A 52 (2016) 100.
[10] N. Colonna, S. Andriamonje, J. Andrzejewski, et al., Nucl. Instrum. Methods Phys. Res. B 269 (2011) 3251.
[11] A. Tsinganis, M. Barbagallo, E. Berthoumieux, et al., Phys. Procedia 64 (2015) 130.
[12] N.M. Larson, Updated Users' Guide for SAMMY: Multilevel R-Matrix Fits to Neutron Data Using Bayes' Equations, ORNL/TM-9179/R8, October 2008.
[13] M.C. Moxon, T.C. Ware, C.J. Dean, REFIT-2009 A Least-Square Fitting Program for Resonance Analysis of Neutron Transmission, Capture, Fission and Scattering Data Users' Guide for REFIT-2009-10 (UKNSF 2010 P243, April 2010).
[14] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, 1996.
[15] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes: The Art of Scientific Computing, third ed., Cambridge University Press, Cambridge, 2007.
[16] F. Mingrone, C. Massimi, G. Vannini, et al., Phys. Rev. C 95 (2017) 034604.
[17] J.H. Wilkinson, Proc. International Congress Math Moscow, 1968, p. 629.
[18] A. Kiełbasiński, Linear Algebra Appl. 88/89 (1987) 487.
[19] M.B. Chadwick, M. Herman, P. Obložinský, et al., Nucl. Data Sheets 112 (2011) 2887.