Converting point-wise nuclear cross sections to pole representation using regularized vector fitting

Xingjie Peng[a,b], Pablo Ducru[a], Shichang Liu[c], Benoit Forget[a], Jingang Liang[a], Kord Smith[a]

[a] *Nuclear Science & Engineering department, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, USA.*
[b] *Science and Technology on Reactor System Design Technology Laboratory, Nuclear Power Institute of China, Chengdu, Sichuan 610000, China.*
[c] *Department of Engineering Physics, Tsinghua University, Beijing 100084, China.*

Abstract

Direct Doppler broadening of nuclear cross sections in Monte Carlo codes has been widely sought for coupled reactor simulations. One recent approach proposed analytical broadening using a pole representation of the commonly used resonance models and the introduction of a local windowing scheme to improve performance [1, 2, 3, 4].

This pole representation has been achieved in the past by converting resonance parameters in the evaluation nuclear data library into poles and residues. However, cross sections of some isotopes are only provided as point-wise data in ENDF/B-VII.1 library. To convert these isotopes to pole representation, a recent approach has been proposed using the relaxed vector fitting (RVF) algorithm [5, 6, 7]. This approach however needs to specify ahead of time the number of poles.

This article addresses this issue by adding a poles and residues filtering step to the RVF procedure. This regularized VF (ReV-Fit) algorithm is shown to efficiently converge the poles close to the physical ones, eliminating most of the superfluous poles, and thus enabling the conversion of point-wise nuclear cross sections.

Keywords: nuclear cross sections, pole representation, Doppler broadening, Vector fitting, ReV-Fit.

1. Introduction

Nuclear cross sections are the fundamental input in the broad field of nuclear engineering which enables the calculation of reaction rates. In the design and operation of nuclear reactors, the energy dependence of nuclear cross sections is the source of many computational difficulties, as it displays resonant structure spanning various orders of magnitude. Additionally, this resonant structure leads to large temperature effects, called Doppler broadening, that must be accounted for accurately in reactor analysis.

Email addresses: pengxj@mit.edu (Xingjie Peng), p_ducru@mit.edu (Pablo Ducru), bforget@mit.edu (Benoit Forget)

In general, nuclear cross sections and their energy dependence are represented by R-matrix theory from which simplified models were derived such as the Multi-Level Breit-Wigner (MLBW), or Reich-Moore (RM). All these models express nuclear cross sections as parametric functions of energy, and the resonance parameters are recorded in nuclear data libraries, such as ENDF/B-VII.1 [8, 9]. Most Monte Carlo codes do not use these resonance parameters directly since they only represent the data at 0K, instead opting for a fine-grid point-wise representation that allows for accurate linear interpolation at a selected temperature.

As computing power increases, Monte Carlo methods have been widely considered for multiphysics simulations capturing temperature variations across the system. To enable these simulations, temperature-dependent nuclear cross sections are needed without substantially impacting the simulation runtime, nor significantly increasing the memory requirements. Much research has been carried out to minimize the number of reference temperatures needed for interpolating nuclear cross sections along the temperature dimension [10, 11, 12, 13, 14, 15, 16].

Another approach is to compute the temperature dependence directly during the Monte Carlo simulation. This is possible for simple cross section models such as the Single Level Breit-Wigner (SLBW) whose Doppler broadened Voigt profile in energy E is well known analytically. In the more general R-matrix case, this operation is however not evident. Yet, in the case of neutron channels without thresholds, for an approximation of the MLBW and the RM models, Hwang introduced a transformation of the parameter space in which the cross sections are cast as a sum of SLBW profiles with complex parameters — or poles (p_j) and residues (r_j) — called the *pole representation* [1, 17, 18]. For instance, under Hwang's hypotheses 0K reaction cross sections take the pole representation form where $\Re(z)$ gives the real part of z:

$$\sigma(E) = \Re\left[\frac{1}{E}\sum_j \frac{r_j}{\sqrt{E} - p_j}\right] \tag{1}$$

By replacing the point-wise piece-wise linear cross section on a fine energy grid with a rational fraction pole representation only requiring the poles (p_j) and residues (r_j), Hwang's pole representation can drastically reduce the memory needed to calculate cross sections [2].

Moreover, Hwang went on to show that it is possible to analytically Doppler broaden this pole representation, which then reduces to evaluating Faddeeva functions for each pole, plus a corrective term $C(E)$, which he calculated recursively and showed to be negligible for nuclear reactor physics purposes [1].

These Faddeeva functions can however be expensive to evaluate. A windowing procedure was thus introduced to increase the computational efficiency of Doppler broadening this pole representation[2, 3, 4]. The windowing algorithm only calls the nearby resonances within the window $\mathcal{W}(E)$ and locally curve-fits the aggregate effect of far-away resonances with a Laurent expansion in $\sqrt{E}$, with coefficients a_n, yielding

the following windowed pole representation [2, 3, 4]:

$$\sigma(E) \simeq \Re\left[\frac{1}{E}\sum_{j\in\mathcal{W}(E)}\frac{r_j}{\sqrt{E}-p_j}\right] + \sum_{n=-2}^{N} a_n E^{\frac{n}{2}} \tag{2}$$

With this piece-wise rational fraction expansion, Doppler-broadening calls to the Faddeeva functions can then be performed on the much reduced set of resonances $\left\{j \in \mathcal{W}(E)\right\}$ as shown in figure 1, yielding considerable performance gains, while a fast recursive algorithm was developed to analytically Doppler broaden the Laurent curve-fit [3, 4].

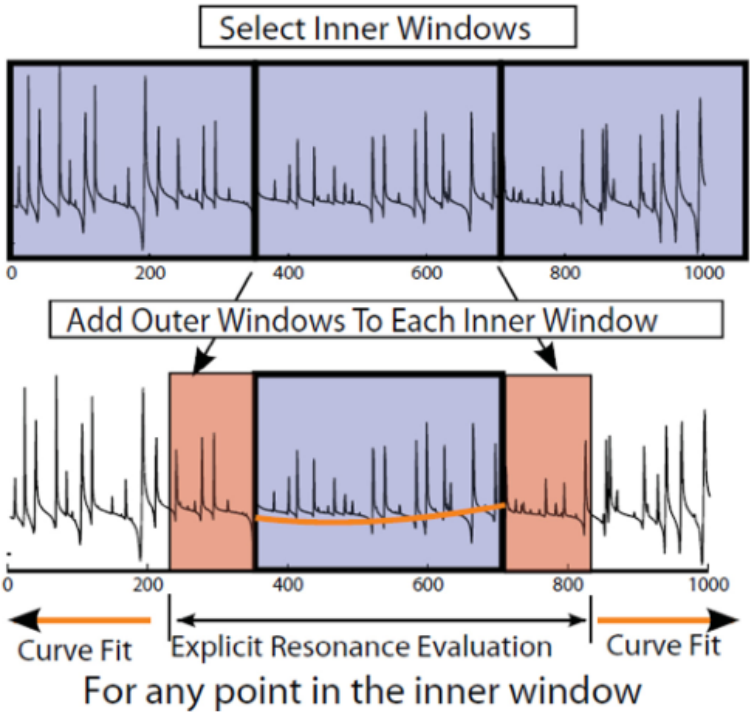


Figure 1: Concept of inner and outer window [3]

The conversion to pole representation has so far been performed by converting ENDF/B-VII.1 resonance parameters to poles and residues. However, in some cases, this conversion process is not possible. For instance, certain nuclides evaluations include point-wise corrections in addition to the resonance parameters. These limitations hinder the usefulness of the pole representation since multiple cross section representation would be required to simulate real systems.

Recently, a new method based on a widespread rational approximation algorithm, called vector fitting (VF) [5, 6], was proposed to calculate poles and residues from point-wise cross sections [7]. This approach was demonstrated successfully on a few nuclides, however, the basic VF algorithm often lead to over-fitting.

The previous approach requires an arbitrary determination of the number of poles needed, and too many poles leads to the creation of un-physical resonances.

This article presents a method to filter-out superfluous poles and systematically identify an optimal number of poles. An implementation of this regularized VF method, here called ReV-Fit algorithm, is presented in section 2, and is then applied in section 3 to convert point-wise nuclear cross section to pole representation with a physically reasonable number of poles.

Since rational approximations are of great use to many other physical applications, such as transfer functions and signal processing, further work on this ReV-Fit algorithm will find much use in the wider community [19, 20, 21].

2. Regularized vector fitting algorithm

The relaxed vector fitting (RVF) algorithm, most commonly used, is an iterative procedure aiming at approximating a known real function $H(z)$ for real z by a rational approximation:

$$H(z) \simeq \sum_{n=1}^{N_p} \frac{R_n}{z - p_n} + C + Dz \tag{3}$$

with unknown poles $\{p_n\}$ of size N_p, residues $\{R_n\}$ and coefficients C and D. $\{p_n\}$ and $\{R_n\}$ are either real quantities or come in complex conjugate pairs, while C and D are real.

This approximation belongs to the class of nonlinear least squares problems because there are unknowns $\{p_n\}$ in the denominator, and RVF solves this problem iteratively by converting it into a sequence of linear problems which consist of pole identification and residue identification steps.

For completeness, the RVF algorithm is briefly described below, however, interested readers should refer to [5, 6] for more details. Given a set of initial poles $\{\overline{p}_n\}$, the pole identification step requires in each iteration to solve the following linear problem in L_2 sense, on a set $\{z_k\}$ of N_k points:

$$s(z)H(z) \underset{\text{Squares}}{\overset{\text{Least}}{=}} \sum_{n=1}^{N_p} \frac{R_n}{z - \overline{p}_n} + C + Dz \tag{4}$$

where the scaling function $s(z)$ is defined as:

$$s(z) = \sum_{n=1}^{N_p} \frac{\widetilde{r}_n}{z - \overline{p}_n} + \widetilde{c} \tag{5}$$

Eq.(4) is linear in its unknowns $\{R_n\}$, $\{\widetilde{r}_n\}$, C, $\widetilde{c}$ and D, and an overdetermined linear problem can be obtained by applying Eq.(4) at $N_d \geq 2N_p + 3$ data points. Once the residues $\widetilde{r}_n$ and coefficients $\widetilde{c}$ have

been obtained, the new poles p_n of $H(z)$ are chosen as the zeros of $s(z)$ which can be calculated from the viewpoint of the following matrix eigenvalue problem [5, 6]:

$$\{p_n\} = \mathrm{eig}(\boldsymbol{A} - \boldsymbol{b}\widetilde{c}^{-1}\widetilde{\boldsymbol{r}}^T) \tag{6}$$

where $\boldsymbol{A}$ is a diagonal matrix containing the initial poles $\{\overline{p}_n\}$, $\boldsymbol{b}$ is a column-vector of ones, and $\widetilde{\boldsymbol{r}}^T$ is a row-vector holding the residues $\{\widetilde{r}_n\}$.

The poles identification step can be executed iteratively with the new poles $\{p_n\}$ replacing the previous poles $\{\overline{p}_n\}$ until convergence is achieved, and then the residues identification step should be performed. The residues $\{R_n\}$ and coefficients C and D can be calculated by solving the corresponding overdetermined linear problem formed by applying Eq.(3) at N_k data points:

$$\{R_n\} = \arg\min \left\{ \sum_{k=1}^{N_d} \left(\sum_{n=1}^{N_p} \frac{R_n}{z_k - p_n} + C + Dz_k - H(z_k) \right)^2 \right\} \tag{7}$$

with Eq. (7) defined for scalar functions. When RVF is applied to vector functions, the residues identification step can be executed separately for each scalar function.

Despite its robustness and widespread use, the RVF algorithm still presents major drawbacks. The most significant being that there is no easy way of selecting the order of the rational fit, and RVF requires specifying in advance the number of poles, N_p, which is problem dependent. Too large a number entails inefficiency in storage (N_p set too large) with the possibility of creating fictitious resonances, and too small a number reduces accuracy. RVF is thus prone to over-fitting, where too many poles are set to curve-fit to within target accuracy.

In this article, a regularized relaxed vector fitting algorithm (ReV-Fit) is proposed to solve the problem of order selection systematically. The essence of this ReV-Fit method is in adding a poles and residues filtering process when the convergence of the pole identification iterations is achieved, so as to eliminate the superfluous poles. This is done by finding the residues that do not contribute significantly to the accuracy of the curve fitting, and eliminating them and their corresponding poles. This process is then repeated until convergence is achieved on the number of poles, and the algorithm stabilizes.

In order to identify the residues that do not contribute significantly to the approximation, an L_1 norm LASSO regularization algorithm is introduced in the residue identification step to perform regression shrinkage and selection[22]. For the m-th iteration with $N_p^{(m)}$ converged poles obtained from the poles identification steps, this is performed by modifying the residue identification problem (7) into a LASSO optimization problem:

$$\{R_n\} = \arg\min \left\{ \sum_{k=1}^{N_d} \left(\sum_{n=1}^{N_p^{(m)}} \frac{R_n}{z_k - p_n} + C + Dz_k - H(z_k) \right)^2 + \lambda^{(m)} \sum_{n=1}^{N_p^{(m)}} |R_n| \right\} \tag{8}$$

where $\lambda^{(m)}$ is the regularization parameter of the m-th iteration. Residues are obtained by solving the LASSO problem in Eq. (8) and then the poles corresponding to zero residues are eliminated, so that $N_p^{(m+1)} \leq N_p^{(m)}$. Eq. (8) is expressed for scalar functions, and for a vector function which can be regarded as several scalar functions, the LASSO regularization step can be executed separately for each scalar function. Poles corresponding to zeros residues for all scalar functions simultaneously are eliminated.

When solving the LASSO problem of Eq. (8), the choice of the regularization parameter $\lambda^{(m)}$ determines the strength of the L_1 penalization. As $\lambda^{(m)}$ increases, the superfluous poles will be eliminated, reducing the over-fitting and increasing the overall accuracy of the rational approximation. However, if $\lambda^{(m)}$ becomes too large, it can end up driving the total number of poles $N_p^{(m+1)}$ too low for a given target accuracy to be achieved. $\lambda^{(m)}$ should be dynamically adapted and tailored to achieve an optimal performance at each new iteration of the residues filtering process, and the selection of $\lambda^{(m)}$ can be performed by resorting to information theory measures, such as the *Akaike's Information Criterion* (AIC) [23] and the *Bayesian Information Criterion* (BIC) [24], or their corrected verison AICc and BICc[25], and then choose $\lambda^{(m)}$ that yields the smallest corresponding AICc or BICc[24]:

$$\lambda^{(m)} \;=\; \arg\min_{\lambda} \left\{ \mathrm{AICc}^{(m)} \; or \; \mathrm{BICc}^{(m)} \right\} \tag{9}$$

where in our ReV-Fit algorithm, the $\mathrm{AICc}^{(m)}$ and $\mathrm{BICc}^{(m)}$ can be expressed separately as:

$$\begin{aligned} \mathrm{AICc}^{(m)} \;&\equiv\; \frac{2N_d N^{(m)}(\lambda)}{N_d - N^{(m)}(\lambda) - 1} \\ &+ N_d \ln \left\{ \sum_{k=1}^{N_d} \left(\sum_{n=1}^{N^{(m)}(\lambda)} \frac{R_n^{(m)}(\lambda)}{z_k - p_n^{(m)}(\lambda)} + C^{(m)}(\lambda) + D^{(m)}(\lambda) z_k - H(z_k) \right)^2 \right\} \end{aligned} \tag{10}$$

$$\begin{aligned} \mathrm{BICc}^{(m)} \;&\equiv\; \ln\left(N_d\right) N^{(m)}(\lambda) \ln(\ln(N^{(m)}(\lambda))) \\ &+ N_d \ln \left\{ \sum_{k=1}^{N_d} \left(\sum_{n=1}^{N^{(m)}(\lambda)} \frac{R_n^{(m)}(\lambda)}{z_k - p_n^{(m)}(\lambda)} + C^{(m)}(\lambda) + D^{(m)}(\lambda) z_k - H(z_k) \right)^2 \right\} \end{aligned} \tag{11}$$

where $N^{(m)}(\lambda)$ is the number of remaining poles after filtering when using λ as the LASSO regularization parameter in the m-th iteration, while $R_n^{(m)}(\lambda)$, $p_n^{(m)}(\lambda)$, $C^{(m)}(\lambda)$ and $D^{(m)}(\lambda)$ are the corresponding fitting coefficients. The number of poles for the $(m+1)$-th iteration is determined by selecting the $\lambda^{(m)}$ that yields

the smallest corresponding AICc or BICc in the poles and residues filtering process:

$$N_p^{(m+1)} = N^{(m)}(\lambda^{(m)}) \tag{12}$$

and the entire process is then repeated until the number of poles converges. Finally, RVF can be repeated several times on the final number of poles to obtain an optimal solution. Since the ReV-Fit algorithm can only eliminate superfluous poles, it should be initialized with a larger number of poles than those deemed physically reasonable: $N_p^{(0)} \gg N_{\text{phys.}}$.

In this study, the open source MATLAB code package VFIT3 [26] which implements the RVF algorithm was used to implement and test the ReV-Fit algorithm. A LASSO optimization process which uses the built-in MATLAB LASSO function was added into VFIT3 to solve the LASSO regularization problem in the form $\boldsymbol{F}\boldsymbol{x} = \boldsymbol{d}$, taking inputs as $(\boldsymbol{F}, \boldsymbol{d}, \lambda)$. Selection of the $\lambda^{(m)}$ regularization parameter was performed by solving the problem (8) for 19 values of $\lambda^{(m)}$ from 10^{-9} to 1, i.e. 10^{-9}, 5×10^{-9}, 10^{-8},..., 10^{-1}, 5×10^{-1}, 1, and choosing the $\lambda^{(m)}$ value that yielded the smallest corresponding AICc or BICc. The number of RVF iterations between successive LASSO steps is set at 10, and the convergence of the entire process is reached when $N_p^{(m+1)} = N_p^{(m)}$ for 20 successive LASSO steps. The implementation of the ReV-Fit algorithm is summarized in Algorithm 1.

Algorithm 1 ReV-Fit : Regularized relaxed Vector Fitting algorithm

Poles Number Initialization: Initialize pole number $N_p^{(0)}$
Poles initialization: Set $\left(p_n^{(0)}\right)_{n\in[\![1,N_p^{(0)}]\!]}$ initial guess for poles
Iteration index initialization: Set m as 0
repeat
  $i \leftarrow 0$
  repeat
    Poles identification: calculate $\left(p_n^{(i+1)}\right)_{n\in[\![1,N_p^{(m)}]\!]}$ from $\left(p_n^{(i)}\right)_{n\in[\![1,N_p^{(m)}]\!]}$ by solving Eq. (6)
    $i \leftarrow i+1$
  until $i = 10$ is reached.
  Poles and residues filtering: calculate filtered poles $\left(p_n^{(0)}\right)_{n\in[\![1,N_p^{(m+1)}]\!]}$ from $\left(p_n^{(10)}\right)_{n\in[\![1,N_p^{(m)}]\!]}$ by solving LASSO optimization problem (8) with optimal $\lambda^{(m)}$ determined by AICc or BICc
  $m \leftarrow m+1$
until Global convergence on $N_p^{(m)}$ i.e. $N_p^{(m)} = N_p^{(m-1)}$ for 20 successive LASSO steps
Final relaxed vector fitting:
$i \leftarrow 0$
repeat
  Poles identification: calculate $\left(p_n^{(i+1)}\right)_{n\in[\![1,N_p^{(m)}]\!]}$ from $\left(p_n^{(i)}\right)_{n\in[\![1,N_p^{(m)}]\!]}$
  $i \leftarrow i+1$
until $i = 10$ is reached.
Residues identification: calculate final residues by solving Eq. (7)

3. Converting nuclear cross sections to pole representation using the ReV-Fit algorithm

The first test of the ReV-Fit implementation is ^{199}Hg which was selected for its simple form of few s-wave resonances with known resonance parameters. This nuclide will provide verification on whether poles and residues from vector fitting are comparable to those generated from the conversion process. The second nuclide tested was ^{157}Gd which cannot be currently converted to pole representation due to the additional point-wise data provided in addition to resonance parameters. The ReV-Fit algorithm was applied to the quantity $E \cdot \sigma(E)$ for consistency with the form of Eq. (1). Accuracy is demonstrated by direct comparison of the cross sections on a fine energy mesh.

3.1. Quantifying the over-fitting problem in ^{199}Hg

Due to its clear and simple resonance structure, the ReV-Fit algorithm was tested on the radiative capture cross section of ^{199}Hg, focusing the energy range on the resolved resonance region which spans 0.01eV to 968eV.

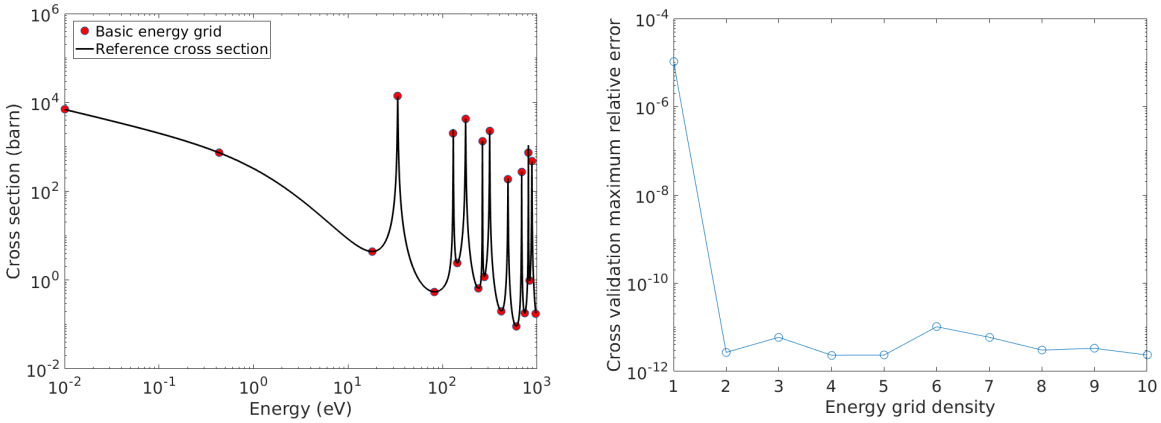
This test is divided into three steps as follows:

1. **Pole representation generation:** The WHOPPER code, developed by Hwang [1], is used to directly convert resonance parameters (from ENDF/B-VII.1) of the radiative capture cross section of ^{199}Hg into physical poles and residues [2]. Then RVF is used to replace all non-fluctuating poles into a few pseudo-poles: 27 reference poles are thus generated for the radiative capture cross section of ^{199}Hg: 10 pairs of conjugate poles near the positive real axis represent 10 resonances, and the remaining pseudo-poles capture the end effects. This set of poles and residues is considered our reference solution.
2. **Point-wise data generation:** Using these reference poles and residues, point-wise nuclear cross section data is generated using Eq. (1). In order to properly capture the resonances, an initial set of energy points is selected as shown in figure 2a. Then point-wise cross section data is generated by refining the basic energy grid, i.e. adding more energy points in order to obtain a target linear interpolation accuracy. Energy grid density (E_{gd}) is defined as:

$$E_{\mathrm{gd}} = \log_2 N_{\mathrm{grid}} \tag{13}$$

   where N_{grid} is the number of energy points between two initial energy points (red points in figure 2a), which are resonance peaks or inflection points in the absence of resonances.
3. **Pole reconstruction and comparison:** Based on either traditional RVF or our ReV-Fit algorithm, poles and residues can be reconstructed from the point-wise cross section generated in step 2, and then the accuracy of RVF or ReV-Fit can be verified by comparing the reconstructed poles to the

reference physical poles. Moreover, the point-wise cross sections generated by the two sets of poles can be compared.



(a) Basic energy grid from which is built the point – wise cross section.

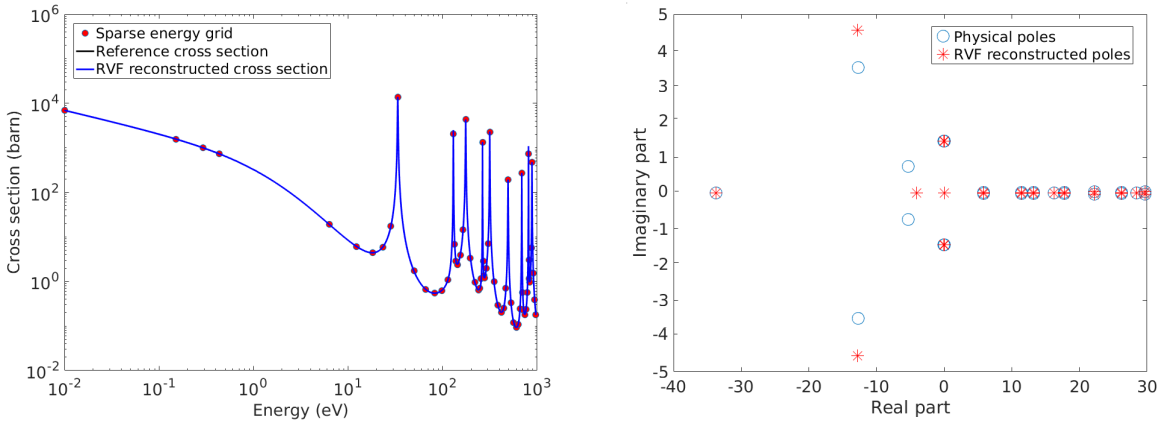
(b) Cross Validation error from RVF applied to different energy densities, provided the ^{199}Hg physically correct number of 27 log-spaced poles as initial guess.

Figure 2: Effect of energy mesh density on VF performance.

In order to test what minimal energy grid density should be provided, the RVF algorithm was given the physically correct number of 27 poles which were initially log-spaced in the energy range of interest as an initial guess, and fitted using different energy grid densities E_{gd}. The maximum relative error between the RVF curve-fit and the exact cross section was computed and the results were recorded in figure 2b. One can observe that if the density of the energy grid is too small, the RVF algorithm fails to converge to the correct poles and residues, even though it was given the correct number of poles. This phenomenon can be observed in figure 3.

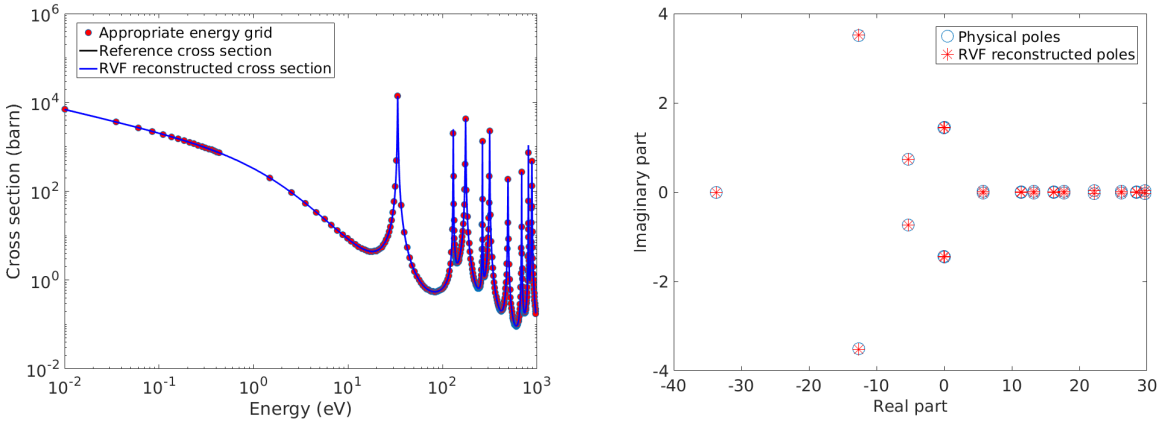
However, if the energy mesh is dense enough, the RVF algorithm is able, provided the ^{199}Hg physically correct number of 27 log-spaced poles as initial guess, to converge to the exact physical poles, as can be observed in figure 4.

Having determined the appropriate energy grid density for the problem, we then proceed to quantify the errors introduced by over-fitting using a cross validation technique. Cross validation gives an insight on how the fitting model will generalize to an independent dataset. The fitting energy grid density is fixed to 4, and the cross validation energy grid density is chosen as 16, and the number of log-spaced initial poles given to the VF algorithm is varied. One can observe in figure 5 that the CV (cross validation) error reaches a minimum at the physical number of poles, past which the VF algorithm starts to over-fit, which significantly damages the performance, losing 3 orders of magnitude of accuracy with just 3 additional poles.



(a) Exact cross section and RVF curve fit. **(b)** Exact poles vs. poles found by RVF.

Figure 3: The RVF algorithm was provided the ^{199}Hg physically correct number of 27 log-spaced poles as initial guess when energy grid density equals 1.



(a) Exact cross section and RVF curve fit. **(b)** Exact poles vs. poles found by VF.

Figure 4: The RVF algorithm was provided the ^{199}Hg physically correct number of 27 log-spaced poles as initial guess when energy grid density equals 4.

This over-fitting phenomenon can be observed in figure 6, where 120 poles were given to the RVF algorithm to fit the cross section. Observe that the RVF now misses many of the physical poles and places some poles on the real axis, generating narrow resonances between grid points. Since they are very narrow and between the energy mesh points, these resonances do not affect the AICc$^{(m)}$, but they do considerably hinder the cross-validation error if a very fine grid is used. This does highlight the importance of proper validation and of eliminating over-fitting to avoid the presence of these unphysical resonances.

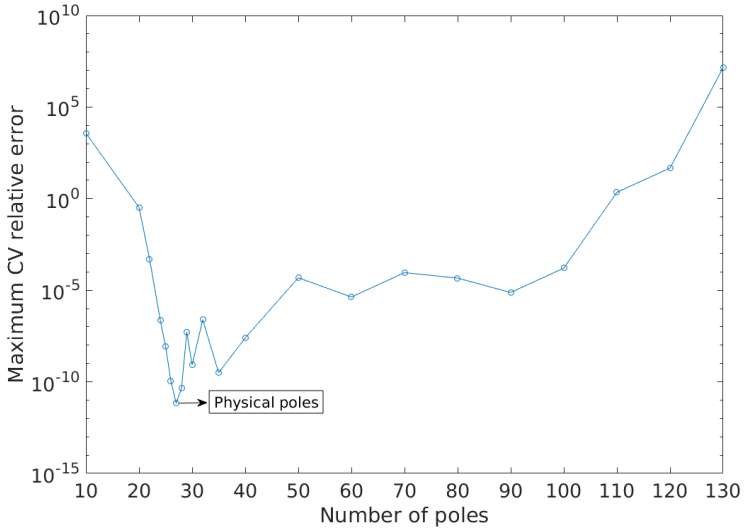
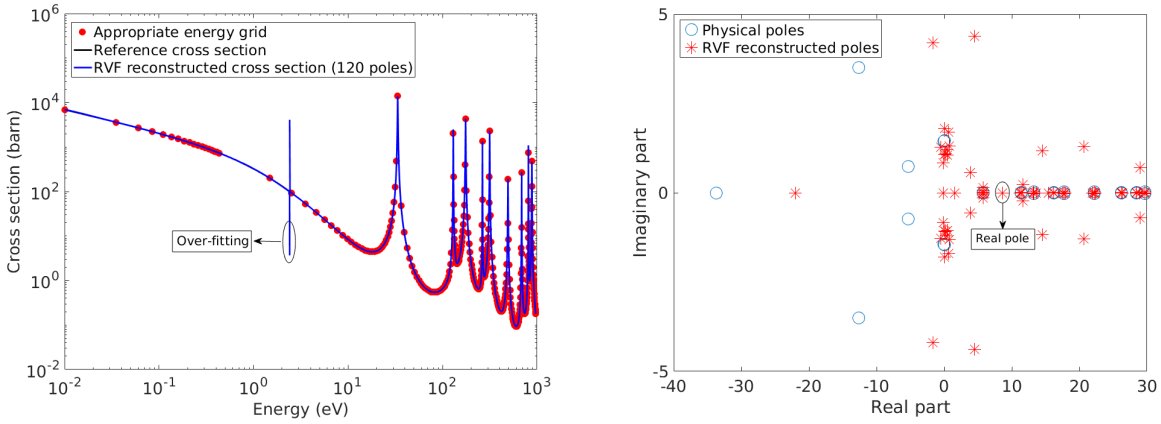


Figure 5: Change of cross validation maximum relative error for different number of poles



(a) Exact cross section and RVF curve fit. **(b)** Exact poles vs. poles found by VF.

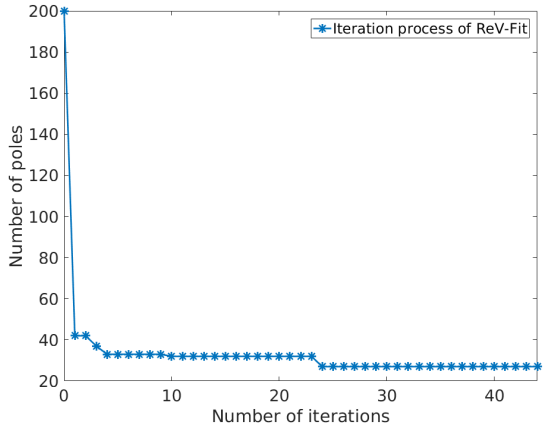
Figure 6: The RVF algorithm was provided 120 log-spaced poles as initial guess, on energy grid density equals 4.

3.2. ReV-Fit performance in eliminating over-fitting in ^{199}Hg

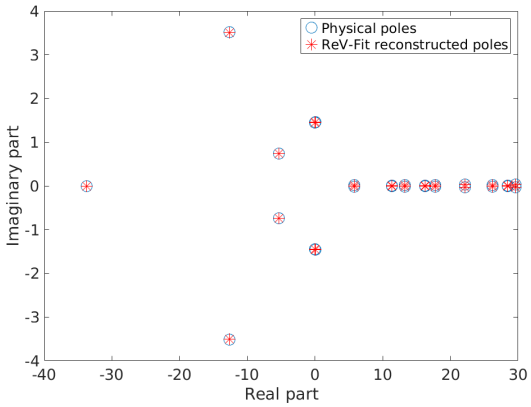
In section 3.1, we quantitatively showed that the RVF over-fitting problem can lead to severe loss of accuracy in cross-validation. In this section we demonstrate how our ReV-Fit algorithm is able to solve this over-fitting problem. The ReV-Fit algorithm was provided 200 log-spaced initial poles, and the energy grid density used for fitting is 4.322 (i.e. 20 energy points between each two basic energy points in figure 2a) while 16 for cross validation.

To assess the capability of the ReV-Fit algorithm using AICc in eliminating over-fitting, the evolution of the number of poles $N_p^{(m)}$ was monitored as the ReV-Fit algorithm progressed, and the results are recorded

in figure 7a. One can observe that, starting from 200 log-spaced poles, the number of poles $N_p^{(m)}$ decreases fast throughout the first ReV-Fit iterations. Remarkably, the ReV-Fit algorithm converges to the physically exact number of 27 poles, and finds their exact physical location, as can be seen in figure 7b. The final solution is presented in figure 8.

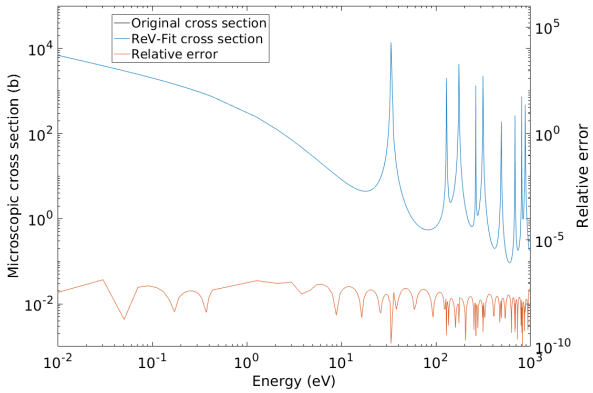


(a) Evolution of number of poles $N_p^{(m)}$ as poles and filtering iterations (m) progress using AICc.

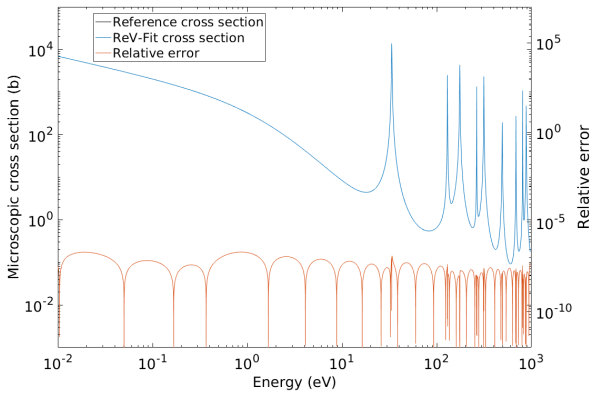


(b) Exact poles vs. poles found by ReV-Fit (AICc).

Figure 7: The ReV-Fit algorithm using AICc was provided 200 log-spaced poles as initial guess, on energy grid density equals $\log_2 20$.



(a) Exact cross section and ReV-Fit (AICc) curve fit on original grid.

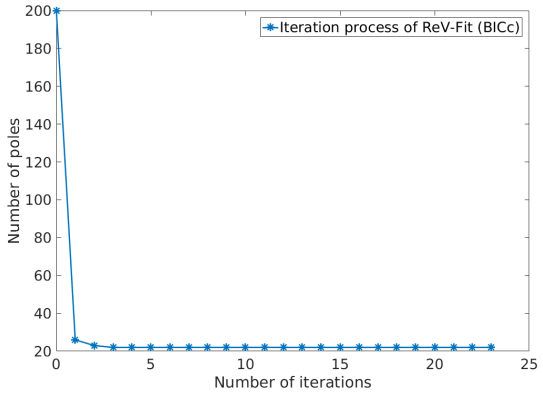


(b) Exact cross section and ReV-Fit (AICc) curve fit on cross-validation grid.

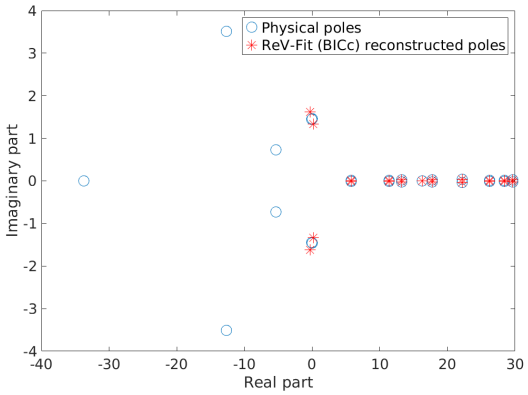
Figure 8: The ReV-Fit algorithm using AICc was provided 200 log-spaced poles as initial guess, on energy grid density equals $\log_2 20$.

For the situation using BICc in the ReV-Fit algorithm to reduce poles, the evolution of the number of poles $N_p^{(m)}$ is presented in figure 9a. The ReV-Fit algorithm converges to 22 poles, and finds the exact locations of resonant poles, as can be seen in figure 9b. The difference with the physical poles only exists

in the pseudo-poles capturing the background effects. The final solution is presented in figure 10, and the accuracy achieved is of 0.1% maximum relative error which is still acceptable in common applications.

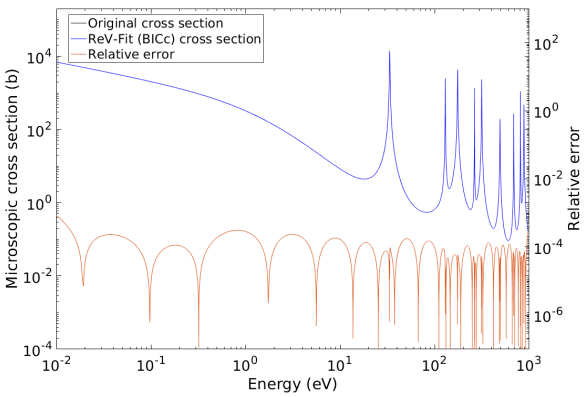


(a) Evolution of number of poles $N_p^{(m)}$ as poles and filtering iterations (m) progress using BICc.

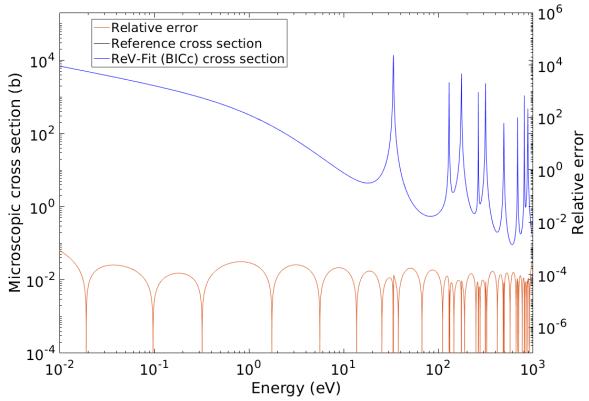


(b) Exact poles vs. poles found by ReV-Fit (BICc).

Figure 9: The ReV-Fit algorithm using BICc was provided 200 log-spaced poles as initial guess, on energy grid density equals $\log_2 20$.



(a) Exact cross section and ReV-Fit (BICc) curve fit on original grid.



(b) Exact cross section and ReV-Fit (BICc) curve fit on cross-validation grid.

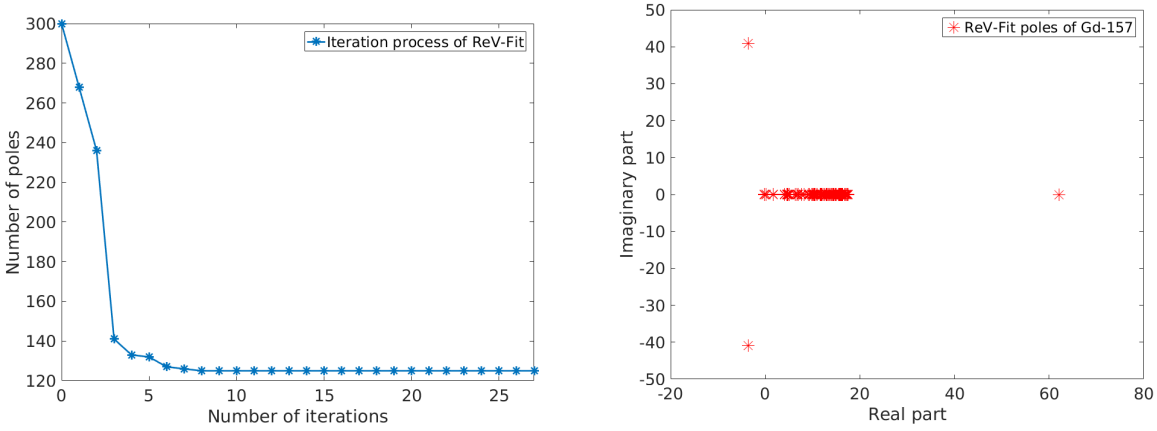
Figure 10: The ReV-Fit algorithm using BICc was provided 200 log-spaced poles as initial guess, on energy grid density equals $\log_2 20$.

This test case demonstrates that the ReV-Fit algorithm is capable of systematically eliminating over-fitting and finding the physical number of poles from which the underlying point-wise data was generated, while it appears that AICc performs better than BICc when the resonance structure is simple.

3.3. ^{157}Gd pole representation from ReV-Fit

The efficacy of the ReV-Fit algorithm to eliminate over-fitting was demonstrated on a nuclide with a known pole representation reference. The purpose of the ReV-Fit algorithm, however, is to generate pole representations for nuclides where the transformation is not possible. In this section, the algorithm is applied to ^{157}Gd since the poles of this nuclide cannot be obtained from the ENDF/B-VII.1 evaluation. This will demonstrate the ability of the ReV-Fit algorithm to construct a pole representation of cross sections when only point-wise data is available.

Gd isotopes are very important in reactor analysis since they are used as integral burnable neutron absorber in nuclear fuel to even out power production over time. These poisons are neutron absorbers which decay (or burn) under neutron exposure. The ^{157}Gd isotope is one of the more important nuclides due to its natural abundance and large absorption cross section. The ReV-Fit algorithm is applied to convert the point-wise cross section data of ^{157}Gd into pole representation. The point-wise data of ^{157}Gd is generated by NJOY [27], and the total number of energy points is 22,764. The ReV-Fit algorithm was fed 300 log-spaced initial poles, to account for the more complex resonance structure of ^{157}Gd.



(a) Evolution of number of poles $N_p^{(m)}$ as poles and filtering iterations (m) progress using BICc.

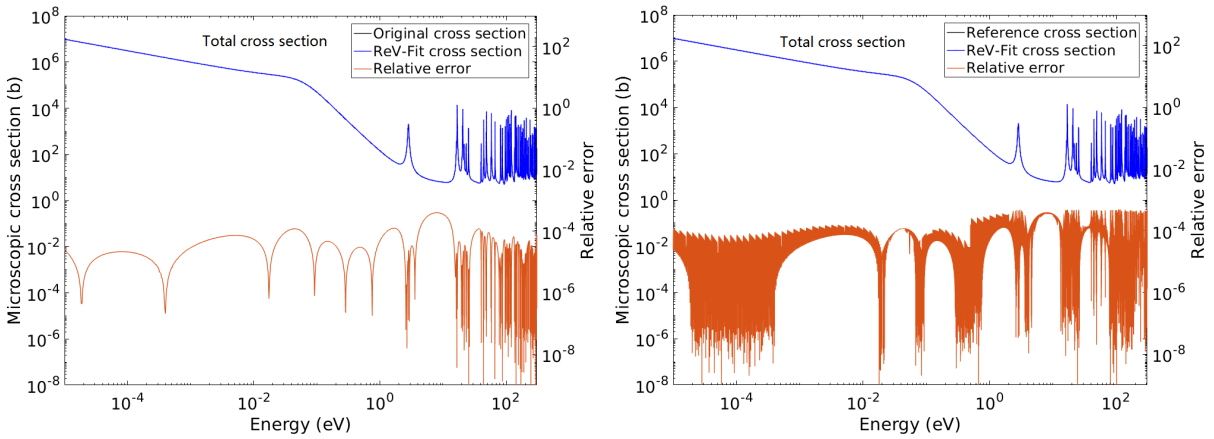
(b) Final ReV-Fit (BICc) poles on ^{157}Gd.

Figure 11: The ReV-Fit algorithm using BICc was provided 300 log-spaced poles as initial guess to reconstruct ^{157}Gd.

In the case of ^{157}Gd, the ReV-Fit algorithm using AICc failed to eliminate some of the real poles, leading to over-fitting. It has been reported in the literature that the AICc can be prone to over-fitting, perhaps in our case because the sample size (i.e. the number of energy points) was too large [25]. However, using BICc succeeded in eliminating the superfluous poles. The evolution of the number of poles $N_p^{(m)}$ using BICc was monitored and recorded in figure 11a, exhibiting similar results as in the ^{199}Hg case, and converging to 125

poles. The distribution of poles generated by ReV-Fit can be seen in figure 11b. Most of the poles occur in the form of conjugate pairs and near the location of physical resonances, while a pair of large imaginary poles accounts for the background behavior at low energies. A single pole is found on the real axis, but was not eliminated by the ReV-Fit algorithm because it is far beyond the energy range considered. This pole performs a similar duty to the use of external resonances (external to the resonance range) when evaluations are generated. These external resonances capture the smooth end effects of the resonance range.

The relative error between the ReV-Fit and exact cross sections was again measured on both the fitting and the cross validation energy grid, for the total and scattering cross sections, and the results are reported in figures 12 and 13, respectively, with a cross validation energy grid of one million log-spaced energy points in the RRR. The accuracy achieved is of the order of 0.1% maximum relative error over the entire energy range, which is sufficient to meet traditional accuracy requirements.

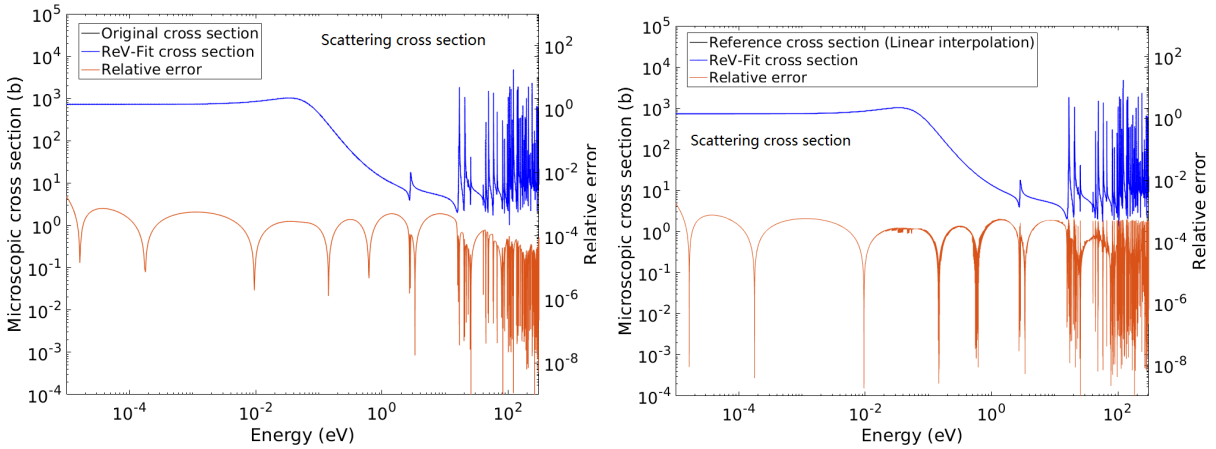


(a) Exact total cross section and ReV-Fit curve fit on original grid.

(b) Exact total cross section and ReV-Fit curve fit on cross-validation grid.

Figure 12: The ReV-Fit algorithm was provided 300 log-spaced poles as initial guess to reconstruct ^{157}Gd.

Because there are no reference poles of ^{157}Gd to be compared with, a brute-force searching method based on RVF was used to determine the poles that fit the cross section of ^{157}Gd best. By performing RVF on the same 22,764 energy points as our ReV-Fit algorithm, and manually changing the number of RVF poles one-by-one, the brute-force searching method selects the number of poles with the smallest cross validation error. The number of poles found by this brute-force method is 126, which is very close to the 125 obtained by ReV-Fit. The comparison of locations between poles generated by the brute-force method and the ReV-Fit can be seen in figure 14. The ReV-Fit method finds the exact locations of resonant poles compared with the brute-force searching method, the only difference being the location and number of pseudo-poles capturing the background effects. This case shows that ReV-Fit can easily identify poles with

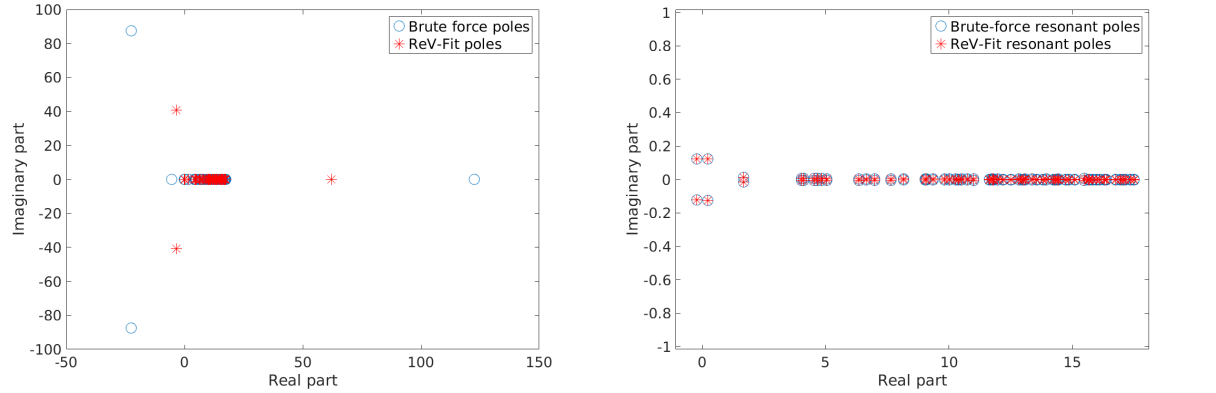


(a) Exact scattering cross section and ReV-Fit curve fit on original grid.

(b) Exact scattering cross section and ReV-Fit curve fit on cross-validation grid.

Figure 13: The ReV-Fit algorithm was provided 300 log-spaced poles as initial guess to reconstruct ^{157}Gd.

physical significance and provide a simple way of generating a pole representation for any nuclide in point-wise form. According to the two cases tested above, AICc is recommended for cases with small number of energy points or simple resonance structure, while BICc is recommended for those with large number of energy points, though further studies are needed. For general applications, both criterion can be tried by others who potentially implementing this ReV-Fit algorithm. In addition, theoretical analysis of different criterion could be important future work.



(a) Brute force poles vs ReV-Fit poles.

(b) Brute force resonant poles vs ReV-Fit resonant poles.

Figure 14: Comparison of poles found by brute-force method and poles found by ReV-Fit.

3.4. Remarks on ReV-Fit and pole representation

The capability of the ReV-Fit algorithm to avoid over-fitting and find physically accurate poles enables the possibility to cast any point-wise nuclear cross section into pole representation, and subsequently into windowed pole representation. This new capability opens the entire charted nuclides to fast direct temperature treatment in reactor physics, opening new avenues for multi-physics coupling in Monte Carlo simulations. From this view-point, it is interesting to know that the vector fitting algorithm has been showed to scale-up well in parallel computations [28], opening perspectives for parallel implementations of ReV-Fit, however this operation only needs to be performed once per evaluation as a pre-processing step.

The ReV-Fit algorithm may also be applied to simplify the windowed pole representation, by providing a rational approximation of the Laurent expansion in equation (2). In this approach, the windowed pole representation would always be cast, in each window $\mathcal{W}(E)$, in the simple form:

$$\sigma(E) \underset{\mathcal{W}(E)}{=} \Re\left[\frac{1}{E}\sum_j \frac{r_j}{\sqrt{E}-p_j}\right] \tag{14}$$

Well chosen constraints could also enforce continuity between windows using the ReV-Fit algorithm. Such a simple "piece-wise rational" form would be a significant improvement to the way nuclear cross sections have been treated so far in nuclear reactor physics, opening new opportunities for analytical solutions and increased computational performance.

4. Conclusions

Converting point-wise nuclear cross sections to pole representation enables efficient direct temperature treatment with minimal memory requirements. Prior work performed this operation using the RVF algorithm [5, 7] which led to over-fitting. In this article, we introduce an additional step to automatically reduce the number of poles N_p until it converges to a minimal number of poles providing a high degree of accuracy. This regularized VF algorithm (ReV-Fit) adds a standard LASSO sparsification step after the RVF to eliminate superfluous poles and residues.

Two different nuclides, ^{199}Hg and ^{157}Gd, were used to demonstrate the feasibility and accuracy of the ReV-Fit algorithm. In the ^{199}Hg case, the reference poles were generated from ENDF/B-VII.1 resonance parameters using the WHOPPER code. From $N_p^{(0)} = 200$ log-spaced initial poles, the ReV-Fit algorithm using AICc converged to the correct physical number of poles $N_p = 27$ and found their exact location, while ReV-Fit algorithm using BICc converged to $N_p = 22$ and found exact resonant poles location. The ReV-Fit algorithm was then applied to ^{157}Gd, which cannot be converted in pole representation from its ENDF/B-VII.1 resonance parameters due to its use of additional point-wise data to correct the resonance parameters.

The ReV-Fit algorithm using BICc reduced the number of poles to 125 and provided an accuracy equivalent to traditionally expected point-wise data. These results show that the ReV-Fit algorithm can effectively generate pole representation for many nuclides in point-wise format below the threshold thus allowing the use of the windowed pole representation method for more nuclides needed in reactor simulations. In addition, extension of this algorithm to energy range above threshold is an on-going work.

Generalization of our ReV-Fit algorithm could find many useful applications in various fields of computational physics where rational approximation is needed, such as model reduction or signal processing.

Acknowledgments

This research was partly supported by the Consortium for Advanced Simulation of Light Water Reactors (CASL), an Energy Innovation Hub for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No.DE-AC05-00OR22725. This research was also partially supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation's exascale computing imperative. The first author was also partly supported by China Scholarship Council.

References

[1] R. N. Hwang, A rigorous pole representation of multilevel cross sections and its practical applications, Nuclear Science and Engineering 96 (3) (1987) 192–209.

[2] B. Forget, S. Xu, K. Smith, Direct doppler broadening in monte carlo simulations using the multipole representation, Annals of Nuclear Energy 64 (2014) 78–85.

[3] C. Josey, B. Forget, K. Smith, Windowed multipole sensitivity to target accuracy of the optimization procedure, Journal of Nuclear Science and Technology 52 (7-8) (2015) 987–992.

[4] C. Josey, P. Ducru, B. Forget, K. Smith, Windowed multipole for cross section doppler broadening, Journal of Computational Physics 307 (2016) 715–727.

[5] B. Gustavsen, A. Semlyen, Rational approximation of frequency domain responses by vector fitting, IEEE Transactions on Power Delivery 14 (3) (1999) 1052–1061, DOI: 10.1109/61.772353.

[6] B. Gustavsen, Improving the pole relocating properties of vector fitting, IEEE Transactions on Power Delivery 21 (3) (2006) 1587–1592, DOI: 10.1109/TPWRD.2005.860281.

[7] S. Liu, X. Peng, C. Josey, J. Liang, B. Forget, K. Smith, K. Wang, Generation of the windowed multipole resonance data using vector fitting technique, Annals of Nuclear Energy 112 (2018) 30–41.

[8] M. Herman, A. Trkov, Endf-6 formats manual, Brookhaven National Laboratory, Brookhaven National Laboratory, Upton, NY (2005) 11973–5000.

[9] M. Chadwick, M. Herman, P. Obložinskỳ, M. E. Dunn, Y. Danon, A. Kahler, D. L. Smith, B. Pritychenko, G. Arbanas, R. Arcilla, 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.

[10] T. Trumbull, Treatment of Nuclear Data for Transport Problems Containing Detailed Temperature Distributions, Nucl. Tech. 156 (2006) 75–86.

[11] T. Viitanen, J. Leppanen, Explicit treatment of thermal motion in continuous-energy Monte Carlo tracking routines, Nucl. Sci. Eng. 171 (2012) 165–173.

[12] T. Viitanen, J. Leppanen, Optimizing the implementation of the target motion sampling temperature treatment technique - how fast can it get?, in: International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, Sun Valley, Idaho, 2013.

[13] G. Yesilyurt, W. Martin, F. Brown, On-the-Fly Doppler Broadening for Monte Carlo Codes, Nucl. Sci. Eng. 171 (3) (2012) 239–257.

[14] W. R. Martin, S. Wilderman, F. B. Brown, G. Yesilyurt, Implementation of on-the-fly doppler broadening in mcnp, in: International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, Sun Valley, Idaho, 2013.

[15] P. Ducru, K. Dibert, C. Josey, B. Forget, V. Sobes, K. Smith, Optimal temperature grid for accurate doppler kernel reconstruction, Transactions of the American Nuclear Society 115 (2017) 1148–1151.

[16] P. Ducru, C. Josey, K. Dibert, V. Sobes, B. Forget, K. Smith, Kernel reconstruction methods for doppler broadening, Journal of Computational Physics 335 (2017) 535–557, http://dx.doi.org/10.1016/j.jcp.2017.01.039.

[17] C. Jammes, R. N. Hwang, Conversion of single- and multilevel breit-wigner resonance parameters to pole representation parameters, Nuclear Science and Engineering 134 (2000) 37–49.

[18] R. N. Hwang, Recent developments pertinent to processing of endf/b-6 type resonnance cross section data, International Conference on the Physics of Nuclear Science and Technology.

[19] C.-U. Lei, Y. Wang, Q. Chen, N. Wong, A decade of vector fitting development: Applications on signal/power integrity, in: AIP Conference Proceedings, Vol. 1285, AIP, 2010, pp. 435–449.

[20] B. Gustavsen, A. Semlyen, Application of vector fitting to state equation representation of transformers for simulation of electromagnetic transients, IEEE Transactions on Power Delivery 13 (3) (1998) 834–842.

[21] Y. S. Mekonnen, J. E. Schutt-Aine, Broadband macromodeling of sampled frequency data using z-domain vector-fitting method, in: Signal Propagation on Interconnects, 2007. SPI 2007. IEEE Workshop on, IEEE, 2007, pp. 45–48.

[22] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society, Series B 58 (1) (1996) 267–288, dOI: 10.1111/j.1467-9868.2011.00771.x.

[23] H. Akaike, A new look at the statistical model identification, IEEE Transactions on Automatic Control 19 (6) (1974) 716–723.

[24] H. Zou, T. Hastie, R. Tibshirani, et al., On the "degrees of freedom" of the lasso, The Annals of Statistics 35 (5) (2007) 2173–2192.

[25] S. Chand, On tuning parameter selection of lasso-type methods-a monte carlo study, in: Applied Sciences and Technology (IBCAST), 2012 9th International Bhurban Conference on, IEEE, 2012, pp. 120–129.

[26] B. Gustavsen, Vfit3 toolbox, `https://www.sintef.no/projectweb/vectfit/downloads/vfut3/`.

[27] R. Macfarlane, D. W. Muir, R. Boicourt, A. C. Kahler III, J. L. Conlin, The njoy nuclear data processing system, version 2016, Tech. rep., Los Alamos National Laboratory (LANL) (2017).

[28] A. Chinea, S. Grivet-Talocia, On the parallelization of vector fitting algorithms, IEEE Transactions on Components,

Packaging and Manufacturing Technology 1 (11) (2011) 1761–1773, DOI: 10.1109/TCPMT.2011.2167973.