— Rational Networks —
Rational approximation of matrix functions by feature extraction & selection

Pablo Ducru[a]

[a] *Nuclear Science & Engineering department, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, USA.*

Abstract

Though deep networks have been at the forefront of the monumental progress in machine learning and artificial intelligence (AI) witnessed in the past decade, shallow networks still have there say when addressing certain types of problem. One such one-layer network is formed of rational functions. Rational functions are universal approximators (c.f. Runge's 1885 theorem), the form of which is sometimes sought as it can be especially well-suited for subsequent treatment. Moreover, myriad problems require modeling of resonant behavior. From transfer functions in electrical engineering, to physical resonances of quantum scattering interactions in spectroscopy, resonances are intrinsically characterized by a linear combination of complex poles and residues. This naturally calls for rational approximation, whereby one can learn the underlying rational functions describing a system.

Rational approximation is a century-old field, pioneered by Heni Padé, which has greatly expanded and been abundantly deployed over the years, either because the form of the problem required a rational solution, or because of the superior characteristics of rational functions over polynomial fitting. For instance "pole-zero modeling" has been central to electrical engineering and signal processing. These traditional rational approximation approaches all focus either on the coefficients of the numerator and denominator, or on its poles and roots (and sometimes on continued fractions):

$$f(z) = \frac{\sum_{n=1}^{N_n} a_n z^n}{\sum_{n=1}^{N_p} b_n z^n} \quad , \quad f(z) = f_0 \frac{\prod_{n=1}^{N_n} (z - \alpha_n)}{\prod_{(=1}^{N_p} (z - p_i)} \tag{1}$$

In this article, we look at this old problem of rational approximation from a novel statistical learning theory viewpoint. From partial fraction decomposition, learning these resonances is akin to a one-layer neural network, where the features (sometimes called atoms in the litterature) are the poles and the residues are the coefficients:

$$f(z) = \sum_{i=1}^{N_p} r_i \varphi_{p_i}(z) \quad , \quad \varphi_{p_i}(z) \coloneqq \frac{1}{z - p_i} \tag{2}$$

we shall henceforth call this a *rational network*.

Mathematically, there is no difference between what we here call a rational network and traditional

rational approximation theory. However, they differ in both spirit and form. In form, in contrast with (1), the rational networks approach focuses on the partial fraction expansion and seeks to learn the poles (p_i) and residues (r_i) of the rational function from which the output elements (y_k) have been drawn. In spirit, portraying this as a machine learning problem will provide us with useful statistical learning theory concepts – such as Reproductive Kernel Hilbert Spaces (RKHS) regularization – which we will use to address centuries-old problems confronting rational approximation.

Since the poles of the rational network are unknown, both feature extraction and selection must be performed. Because of this, training this rational network is not straightforward. Moreover, the poles introduce highly non-linear (actually resonant) behavior that impeach us to resort to traditional gradient-descent methods. Methods have been introduced in the past to circumvent this issue and establish iterative schemes to converge to the poles CITE [REMEZ ALGORITHM, SK ITERATION, VF. etc.]. In particular, Gustavsen's barycentric SK iteration by pole relocation was very successful and is the foundation of the widespread Vector Fitting (VF) algorithm CITE. In this article, we establish a comprehensive theory of the rational network, and combine the VF pole convergence scheme with various regularization methods from statistical learning theory to build a new algorithm — called Rational AToms Identification with Optimized Numerics ALGorithm (RATIONALG) — which introduces systematic methods for order selection and solves the traditional problem of over-fitting with poles and residues.

This article builds upon the widespread relaxed vector fitting algorithm to establish a new, regularized relaxed vector fitting (ReV-Fit) algorithm, introducing a systematic method to solve the problem of over-fitting with poles and residues.

TO DO

- Explain the embedded "kitchen sink" approach : 1. VF = feature extraction 2. Lasso = feature selection. 3. Compare this with other loss functions !
- show that all-in-one fails: you cannot extract the features while selecting them (solving one lasso will fail).
- Should pole filtering use the barycenter function ?
- Compare barycentric SK with gradient descent, and estimate acceleration of convergence
- Compare approximation theory case (not noisy signal -> convergence to exact poles capability) with noisy signal case (converge to mean poles ?)
- final table : gradient descent v.s. ReV-Fit ; deterministic v.s. noisy ; 1.2. alternations steps ratio, etc.

PUBLISH IN

- Journal: IEEE Patern analysis and Machine Intelligence.
- Conference: Try NIPS this year ! Deadline in8 weeks ! You can do this !

Keywords: ReV-Fit, Vector fitting, relaxed vector fitting, regularization, LASSO, over-fitting, curve-fitting, feature selection, signal processing, ARMA, MIMO, model reduction, poles and residues, rational approximation, nuclear cross sections, pole representation, Doppler broadening.

---

1. Introduction

MYRIAD fields, from machine learning to signal processing, computational physics or control theory, require approximating matrices — transfer functions, time series analysis, multi-input/multi-output system analysis, or video content analysis — that depend on a given variable, such as time, frequency, position or energy.

Often, one seeks to approximate the matrix function with a rational function, either because the physics of the problem dictate such pattern dependence, or because rational functions happen to be particularly convenient for subsequent treatment.

The establishment, in the late 19^{th} - early 20^{th} century, of the fundamental complex analysis theorems of Mittag-Leffler and Runge jointly guarantee that any meromorphic function can be uniformly approximated on a given domain of the complex plain by rational functions. Padé went on to show how such rational approximants could be constructed, and since then a plethora of methods have been proposed, many focusing on either determining the coefficients of the numerator and denominator polynomials, or taking the continued fraction approach [CITE]. In practice, such representations are however not always best suited for subsequent treatment as they are not a linear sum but a ratio of two polynomials. Instead, the partial fraction expansion form provides with a linear sum of monomials (from the entire part) and poles and residues (from the principal part), which often lend themselves well to ensuing manipulations. Also, in many cases (though there are exceptions [1]) poles of degree one suffice for modeling purposes, and simple poles thus form a feature space of elemental atoms for approximating the dependence of the given matrix function.

Vector fitting (VF) has become a widespread method to approximate matrix functions with a simple partial fraction expansion, with myriad applications in signal processing, statistical time series analysis

Email address: p_ducru@mit.edu (Pablo Ducru)

autoregressive-moving-average (ARMA) models, model reduction and surrogate macro-modeling, data analysis regression and optimization, feature selection in machine learning, or transfer functions associated with multi-input/multi-output (MIMO) dynamical systems [2, 3, 4, 5, 6]. Presently, the community that has emerged around this method of rational approximation most efficiently solves this inherently ill-posed problem by means of the relaxed vector fitting (RVF) algorithm [7, 8], or various derivative methods that seek to address specific shortcomings of VF [9, 10, 11, 12, 13, 14, 15].

Be it centuries-old Padé-type approximations or modern-day RVF and its derivatives, all rational approximation methods intrinsically suffer from an over-fitting problem: one has to arbitrarily prescribe ahead of time the number of poles with which to fit a function. This is not only inefficient in that either too many poles are prescribed – and thus stored – to achieve the same result, or too few are proposed to achieve the target accuracy, but over-fitting also severely hinders the generalization of the fit. This leads to a manual and costly meta-optimization on the number of poles

In the particular case of the RVF algorithm, prescribing too many poles makes certain become numerically unstable and requires engineering tweaks such as "pole flipping" during the procedure [8, 16, 17, 1, 18, 19]. Sometimes over-fitting takes the form of unstable poles which become real and produce very thin infinite resonances which fall between reference points and the cross validation grid, yielding enormous errors when generalizing to other cases.

Literature abounds with methods proposed to reduce the order of rational approximations (such as pole clustering or filtering [CITE]). In the particular case of VF, Drmac recently proposed the McMillan degree as a metric to estimate the order complexity and showed how a specified low-rank approximation could be enforced [14, 15], while various methods have been introduced to reduce the order of the rational approximation by filtering off spurious poles [20, 13, 17, 21]. However, all these previous methods rely on frequency-domain *ad-hoc* methods to detect spurious poles and manually eliminate them.

This article brings a new perspective to this old problem. Through the scope of regularization in machine learning, we view rational approximation as a standard non-linear Least-Squares (LS) statistical batch learning problem with a specific feature space comprised of the different poles.

We first recognize that the VF algorithm core contribution is to provide a fast method to converge the poles when their number is specified: in essence, this is a feature extraction step.

We then systematically address the VF algorithm over-fitting problem by eliminating the superfluous poles through a sparsity-inducing L_1-L_2 group least absolute shrinkage and selection operator convex-relaxation regularization (group LASSO penalization) on the residues. This feature selection step provides a systematic and problem-agnostic way of reducing the order of the rational approximation.

Our contribution is thus two-fold: from the VF viewpoint, we build upon the RVF relaxed vector fitting algorithm and apply standard machine learning feature selection methods to construct a robust and systematic *regularized relaxed vector fitting* (ReV-Fit) algorithm, which solves both the over-fitting and the stability problems. The ReV-Fit algorithm thus performs robust rational function approximation with systematic elimination of spurious poles. From the pattern analysis and machine learning view point, our ReV-Fit algorithm provides a systematic framework to solve the group LASSO regularized LS problem, taking advantage of the RVF poles convergence for feature extraction. Not only is this iterative procedure specifically tuned for fast and robust rational function, but it also performs both feature extraction and selection in the space of poles and residues functions (in a way somewhat reminiscent of Gaussian Model aggregation).

In section 2 we presented a review of vector fitting results, upon which section 3 builds to establish our ReV-Fit algorithm. The ReV-Fit algorithm is then tested on reference problems and section 4 reports its performance.

2. Review of vector fitting

Few articles develop the mathematical basis of the vector fitting algorithm [14, 22, 15]. We here present a thorough but synthetic approach to the latest works and improvements in the vector fitting theory and algorithm, in the wake of the works of Gustaven [8] and Drmac [14, 15]. We generalize some results to the case of RVF, and refer to the latter works for myriad insights into effective numerical implementation.

2.1. Relaxed vector fitting algorithm

The present widespread relaxed vector fitting algorithm is an iterative procedure aiming at approximating a known matrix function $\boldsymbol{H}(z)$ by a rational approximation $\boldsymbol{H_r}(z)$:

$$\boldsymbol{H}(z) \simeq \boldsymbol{H_r}(z) \;\coloneqq\; \sum_{n=1}^{N} \frac{\boldsymbol{R_n}}{z - p_n} + \boldsymbol{C} + \boldsymbol{A}z \tag{3}$$

Provided observed data points $(z_k)_{k\in[\![1,N_k]\!]}$ with weights $(\rho_k)_{k\in[\![1,N_k]\!]}$, the objective is to find the rational approximation that solves the non-linear weighted least square (LS) problem:

Find the rational approximant $\boldsymbol{H_r}(z)$ *that minimizes the empirical weighted least square error* $\widehat{\mathcal{E}}$

$$\widehat{\mathcal{E}} \;\coloneqq\; \sum_{k=1}^{N_k} \rho_k \left\| \boldsymbol{H_r}(z_k) - \boldsymbol{H}(z_k) \right\|_{\mathrm{L}_2}^2 \tag{4}$$

The L_2 norm $\|\cdot\|_{L_2}$ is the appropriate Euclidean norm for the vector space considered, which is the reason why the algorithm is called vector-fitting. In particular, it can simply be the modulus square in the case of scalars, or it can be the Frobenius norm associated to the Hilbert-Schmidt scalar product in the case $\boldsymbol{R_n}$, $\boldsymbol{C}$, and $\boldsymbol{A}$ are matrices.

To solve the difficult non-linear optimization problem (4), the vector fitting algorithm decomposes it in a series of simpler linear problems:

1. First, it converges the poles through a barycentric Sanathanan-Koerner iterative pole relocation procedure.
2. The residues are then retrieved in a final de-biasing step, which simply solves the linear weighter least squares problem (4) with fixed poles from 1.

1. **Poles convergence**: The core innovation of the VF algorithm is its relaxed barycentric Sanathanan-Koerner iterative pole relocation procedure.
   Instead of solving problem (4) directly, the relaxed barycentric Sanathanan-Koerner iterative pole relocation procedure takes the more robust route of iteratively solving for the following optimization problem (5):

$$\min \widehat{\mathcal{E}}_{\mathrm{SK}}^{(i)} \coloneqq \sum_{k=1}^{N_k} \rho_k \left\| \boldsymbol{H_r}^{(i)}(z_k) - b^{(i)}(z_k)\boldsymbol{H}(z_k) \right\|_{L_2}^2 \tag{5}$$

   where for each iteration (i) are constructed two rational fraction operators with the same poles $\left(p_n^{(i)}\right)_{n\in[\![1,N_p]\!]}$:

   - The rational approximant $\boldsymbol{H_r}^{(i)}(z)$:

$$\boldsymbol{H_r}^{(i)}(z) \coloneqq \sum_{n=1}^{N_p} \frac{\boldsymbol{R_n}^{(i)}}{z - p_n^{(i)}} + \boldsymbol{C}^{(i)} + \boldsymbol{A}^{(i)} z \tag{6}$$

   - and the relaxed barycenter $b^{(i)}(z)$ function:

$$b^{(i)}(z) \coloneqq \sum_{n=1}^{N_p} \frac{r_n^{(i+1)}}{z - p_n^{(i)}} + c^{(i+1)} \tag{7}$$

   The poles convergence procedure then breaks down the iterations into two separate linear problems: a) residue SK iteration, and b) pole relocation. These are such that their iteration ensures $b^{(i)}(z) \underset{i\to\infty}{\to} 1$ and thus $\boldsymbol{H_r}^{(i)}(z) \underset{i\to\infty}{\to} \boldsymbol{H_r}(z)$, which solves the initial problem (4).
   Convergence of the poles, though theoretically not guaranteed, is determined when the iteration steps become small enough (Cauchy criterion).

(a) **Residue SK iteration:** In this step, the poles $\left(p_n^{(i)}\right)_{n\in[\![1,N_p]\!]}$ are fixed and the residues and coefficients $\boldsymbol{K}^{(i)} \coloneqq \left(\boldsymbol{R_n}^{(i)}, \boldsymbol{C}^{(i)}, \boldsymbol{A}^{(i)}, r_n^{(i)}, c^{(i)}\right)$ are updated by solving what is now a linear least square problem. However, in order to avoid the trivial solution, as well as accelerating the convergence of $b^{(i)}(z) \underset{i\to\infty}{\longrightarrow} 1$, the following constraint is imposed upon the relaxed barycenter function $b^{(i)}(z)$ [8]:

$$\Re\left[c + \frac{1}{N_k}\sum_{k=1}^{N_k}\sum_{n=1}^{N_p}\frac{r_n}{z_k - p_n}\right] = 1 \tag{8}$$

When imposed upon the LS problem, this constrain is introduced with the Lagrange multiplier $\mu \geq 0$, and the residue SK iteration reduces to the following linear least square problem:

$$\begin{aligned}\boldsymbol{K}^{(i+1)} = \underset{\boldsymbol{K}^{(i)}}{\arg\min}\ \Bigg\{ &\sum_{k=1}^{N_k}\rho_k \left\| \sum_{n=1}^{N_p}\frac{\boldsymbol{R_n}^{(i)}}{z_k - p_n^{(i)}} + \boldsymbol{C}^{(i)} + z_k\boldsymbol{A}^{(i)} - \sum_{n=1}^{N_p}\frac{\boldsymbol{H}(z_k)}{z_k - p_n^{(i)}}r_n^{(i)} - \boldsymbol{H}(z_k)c^{(i)} \right\|_{\mathrm{L}_2}^2 \\ &-\mu\,\Re\left[c^{(i)} + \frac{1}{N_k}\sum_{k=1}^{N_k}\sum_{n=1}^{N_p}\frac{r_n^{(i)}}{z_k - p_n^{(i)}} - 1\right]\Bigg\}\end{aligned} \tag{9}$$

Solving for residues SK iteration:

Though the value of the Lagrange multiplier μ is undetermined to solve problem (9), it is easy to show that $\mu = 2\sum_{k=1}^{N_k}\rho_k\left(\|\boldsymbol{H}(z_k)\|_{\mathrm{L}_2}^2 - \Re\left[\langle \boldsymbol{H}(z_k), \boldsymbol{H_r}^{(i)}(z_k)\rangle\right]\right)$, which will tend towards zero as $\boldsymbol{H_r}^{(i)}(z_k) \underset{i\to\infty}{\overset{\mathrm{L}_2}{\longrightarrow}} \boldsymbol{H}(z_k)$. This backs Gustaven's claim that constraint (8) does not restrict system (9) other than avoiding the trivial solution. Gustaven observes that, in practice, setting a commensurate scaling to the Lagrange multiplier, such as $\mu \coloneqq \sum_{k=1}^{N_k}\rho_k\|\boldsymbol{H}(z_k)\|_{\mathrm{L}_2}$, avoids numerical scaling issues [8].

At the next iteration, only the new barycentric coefficients $\left(r_n^{(i+1)}, c^{(i+1)}\right)$ are used to relocate the poles: matrices $\left(\boldsymbol{R_n}^{(i+1)}, \boldsymbol{C}^{(i+1)}, \boldsymbol{A}^{(i+1)}\right)$ are actually discarded. This means a QR decomposition can formidably improve the performance of the algorithm, as shown in [**?**]. IMPLEMENT THIS QR VERSION ! THE $\left(r_n^{(i+1)}, c^{(i+1)}\right)$ ARE PROBABLY OBTAINABLE DIRECTLY BY HAND.

(b) **Pole relocation:** Once the residue SK iteration (a) has updated the coefficients $\boldsymbol{K}^{(i+1)}$, the pole relocation (b) step updates the poles, $\left(p_n^{(i)}\right)_{n\in[\![1,N_p]\!]} \to \left(p_n^{(i+1)}\right)_{n\in[\![1,N_p]\!]}$, as the roots of the barycentric function with new residues:

$$\left(p_n^{(i+1)}\right)_{n\in[\![1,N_p]\!]} = \left\{ z \;\middle|\; 0 \;=\; b^{(i)}(z) \coloneqq \sum_{n=1}^{N_p}\frac{r_n^{(i+1)}}{z - p_n^{(i)}} + c^{(i+1)} \right\} \tag{10}$$

This was Gustaven's improvement upon traditional Sanathanan-Koerner iterations, as it is both fast, stable, and at convergence guarantees $b^{(i)}(z) \underset{i\to\infty}{\to} 1$ (cf. [14], section 2.1).

Convergence of pole relocation: Convergence $b^{(i)}(z) \underset{i\to\infty}{\to} 1$ comes from contraint (8), and the fact that (10) entails the relaxed barycentric function can be written as:

$$b^{(i)}(z) \coloneqq c^{(i+1)} \frac{\prod_{n=1}^{N_p} \left(z - p_n^{(i+1)}\right)}{\prod_{n=1}^{N_p} \left(z - p_n^{(i)}\right)} \tag{11}$$

If poles converge $\left(p_n^{(i+1)}\right)_{n\in[\![1,N_p]\!]} \underset{i\to\infty}{\simeq} \left(p_n^{(i)}\right)_{n\in[\![1,N_p]\!]}$, expression (11) drives the residues towards zero, $\left(r_n^{(i)}\right)_{n\in[\![1,N_p]\!]} \underset{i\to\infty}{\to} 0$, in turn driving the real part of the relaxation constant $c^{(i)}$ towards unity through constraint (8): i.e. $\Re\left[c^{(i)}\right] \underset{i\to\infty}{\to} 1$. As the residues tend to zero and the real part of the relaxation amplitude coefficient $c^{(i)}$ tends towards unity, then the LS problem (9) will drive the imaginary part to zero: $\Im\left[c^{(i)}\right] \underset{i\to\infty}{\to} 0$. Thus, at convergence the relaxed barycenter function tends towards unity: $b^{(i)} \underset{i\to\infty}{\longrightarrow} 1$. This is a generalization of Drmac's proof in the case of non-relaxed VF [14].

Solving for pole relocation: Pole relocation problem (10) can be solved by numerical polynomial root-finding methods, such as the Aberth-Householder methods introduced in [**?**], or other standard polynomial root-finding techniques. Problem (10) can however also be cast into the following eigenvalue problem, unlocking standard eigenvalue methods:

$$\left(p_n^{(i+1)}\right)_{n\in[\![1,N_p]\!]} = \mathcal{S}p\left(\boldsymbol{P}^{(i)} - \frac{1}{c^{(i+1)}} \boldsymbol{r}^{(i+1)} \mathbf{1}_n^\top\right) \tag{12}$$

where $\boldsymbol{P} = \mathbf{diag}\left(p_n^{(i)}\right)$ is the diagonal matrix of the old poles, $\mathbf{1}_n$ is the N_p-length vector $\mathbf{1}_n = [1,\ldots,1]^\top$, and $\boldsymbol{r}^{(i+1)} = \left[r_1^{(i+1)},\ldots,r_{N_p}^{(i+1)}\right]^\top$ is the vector of residues of the relaxed barycenter function $b^{(i)}(z)$. Solving problems (12) or (10) is equivalent, since $\det\left(\boldsymbol{P}^{(i)} - z\mathbb{1} - \frac{1}{c^{(i)}}\boldsymbol{r}^{(i)}\mathbf{1}_n^\top\right) = 0$ is equivalent to $\det\left(c^{(i)}\mathbb{1} + \left(z\mathbb{1} - \boldsymbol{P}^{(i)}\right)^{-1}\boldsymbol{r}^{(i)}\mathbf{1}_n^\top\right) = 0$. This means $\forall j \in [\![1,N_p]\!],\ \exists\ v_j \neq 0 \mid c^{(i)}v_j + \left(\frac{r_n^{(i)}}{z_k - p_n^{(i)}}\right)\left(\sum_{j=1}^{N_p} v_j\right) = 0$, which, summing on j, yields the intended result $b^{(i)}(z) = 0$.

Convergence criterion: Drmac proposed the following criterion to stop poles convergence [14, 15]: iterations (i) come to a stop when the barycentric residues, $\left(r_n^{(i)}\right)$, which tend towards zero, are below a given tolerance threshold θ:

$$\theta^{(i)} \coloneqq \sum_{n=1}^{N_p} \left|\frac{\widetilde{r_n}^{(i)}}{\Re\left[p_n^{(i)}\right]}\right| \leq \theta \tag{13}$$

2. **Residues retrieval**: This last phase is much shorter and simply consists in solving directly problem

(4) with fixed poles, $\left(p_n^{(i_{\text{stop}})}\right)_{n\in[\![1,N_p]\!]}$, obtained from the pols convergence.

Though a linear LS, this operation can be notoriously ill conditioned due to the Cauchy matrices involved. Traditional Tichonov regularization methods can thus be used to ameliorate the numerical stability of this residues retrieval phase, with the penalization parameter chosen from the Morozov discrepancy principle [14] eq.(3.7).

The traditional relaxed vector fitting algorithm thus takes the following form [Algorithm 1]:

Algorithm 1 RVF: Relaxed Vector Fitting algorithm

POLES INITIALIZATION: Set $\left(p_n^{(0)}\right)_{n\in[\![1,N_p]\!]}$ initial guess for poles

1. POLES CONVERGENCE:

repeat

(a) **residue SK iteration:** fix $\left(p_n^{(i)}\right)_{n\in[\![1,N_p]\!]}$ and solve problem (9) to update the coefficients $\boldsymbol{K}^{(i+1)}$.

(b) **pole relocation:** calculate $\left(p_n^{(i+1)}\right)_{n\in[\![1,N_p]\!]}$ by solving problem (10), or equivalently problem (12).

until Convergence criterion (13) is reached

2. RESIDUES RETRIEVAL: fix poles from last iteration $\left(p_n^{(i_{\text{stop}})}\right)_{n\in[\![1,N_p]\!]}$ and solve problem (4) to obtain coefficients and residues of rational approximation $\boldsymbol{H_r}(z)$.

2.2. Relaxed vector fitting shortcomings

Despite its robustness and widespread use, the RVF algorithm still presents major drawbacks, some of intrinsic nature, and some of numerical nature.

- **Over-fitting:** The first and most significant drawback is that RVF requires specifying in advance the number of poles, N_p, with which the curve-fit will be performed. N_p is, however, intrinsically curve-dependent, so setting in it advance necessarily entails either inefficiency in storage (N_p set too large) or insufficiency in accuracy (N_p too small to achieve numerical accuracy target). This means that RVF is in practice prone to over-fitting, where too many poles are set to curve-fit. The problem with over-fitting is not just in memory requirement. Over-fitting also hinders the extrapolating performance beyond the set of measured data points (z_k). This is prohibitory when in quest of a curve fit with predictive power, or when building a lower-order surrogate model. Several proposals have been made to tackle this over-fitting problem and determine the right number of poles and residues [13, 21, 20]. Our ReV-Fit algorithm in a way generalizes them all by introducing standard optimization and machine learning regularization methods.

- **Unstable poles:** Another drawback introduced in part by over-fitting is that the RVF is prone to unstable poles. Some poles become exactly real, which is physically impossible and should be

synonym of a $\frac{0}{0}$ type of problem. Dealing with such unstable poles has so far been done through *ad hoc* engineering methods, such as "pole flipping". Various attempts have been made at dealing with this pole instability problem [16, 18, 17, 1].

- **Noisy data:** Poles instability is made worse when dealing with noisy data. It has been observed that the RVF may fail to converge in this case [23, 24]. This difficulty may be dealt with by properly accounting for these statistical fluctuations in the weights ρ_k, defining them as the inverse of the variance of the measures, $\rho_k = \frac{1}{\sigma_k^2}$ [25, 26].

- **Sensitivity to initial poles:** The rational approximation problem being ill-posed, the iterative VF algorithm will converge, if it converges at all — which is not guaranteed [23, 24] — to a local optimum that depends on the initial pole guesses. The orthonormal vector fitting OVF [11, 3] was introduced to lessen this dependence on initial guesses. By relaxing the barycentric function (7) — that is allowing for $\widetilde{c}^{(i)}$ to take any value at the beginning of the iterations, though we showed it ultimately converges to 1 — the RVF not only increased the speed of VF convergence, but also diminished the dependence on initial guesses as the poles are now more free to relocate [19, 27, 8]. Thus, in RVF, Gustaven suggests to spread the initial poles logarithmically, so as to cover the entire span on the fit, and then let the poles relocate to their correct positions. Other approaches have been proposed to improve this initial guess [28].

- **Ill conditioning:** By nature, the rational approximation problem is ill conditioned: the Cauchy matrices intervening are prone to high condition numbers [29]. This is intrinsically related to the vector fitting method, and considerable amount of work has been deployed to circumvent this ill-conditioning of the scheme [30, 31], for instance with a Tichonov regularizer in mimoVF [14, 15], or the Modal VF algorithm [9, 10]. Other methods could be deployed to address this high condition number issue, such as general preconditioning methods [32], in particular the widespread Preconditioned Conjugate Gradient (PCG) method [33]. More specifically tailored to the VF algorithm would be to deal with the explicit solutions of Cauchy matrices inversions [34].

In this article, we argue that the pole instability, sensitivity to initial guesses, selection of parameter order, and over-fitting problems are all linked. We introduce standard optimization and machine learning regularization procedures in an attempt to solve all these problems at once with our ReV-Fit algorithm. The remaining Cauchy matrix ill-conditioning problems can be resolved by standard operating procedures, and we show how some of these fit naturally in our ReV-Fit algorithm (section **??**, numerical instability: Elastic net regularization).

3. Regularized relaxed vector fitting: ReV-Fit

This article establishes our new method — here called *Regularized relaxed Vector Fitting* (ReV-Fit) — that purposes to systematically solve the over-fitting problem and related issues (exposed in section 2.2) the relaxed vector fitting (RVF) is intrinsically prone to.

The essence of this ReV-Fit algorithm is a kitchen sink approach: one starts with a way-too-high order approximation by specifying too many initial poles $N_p^{(0)} \gg N_p^{(\text{exact})}$; the algorithm then gradually eliminated spurious poles to converge to the right number and value of poles and residues.

This is achieved by iterating over two alternating phases: 1. pole convergence and 2. pole filtering. Iteration (m) of the ReV-Fit algorithm thus starts with $N_p^{(m)}$ poles and goes through the 2-step process:

1. pole convergence: a feature extraction phase using the RVF pole convergence procedure (residues SK iteration + relaxed barycentric pole relocation) that convergences $N_p^{(m)}$ poles: $\left(p_n^{(\text{conv})}\right)_{n\in[\![1,N_p^{(m)}]\!]}$
2. pole filtering: a feature selection phase where superfluous poles are eliminated by means of a sparsity-based L_1 regularization, yielding a new set of $N_p^{(m+1)} \leq N_p^{(m)}$ filtered poles: $\left(p_n^{(\text{filt})}\right)_{n\in[\![1,N_p^{(m+1)}]\!]}$

This process is then repeated until convergence is achieved on the number of poles (pole filtering) and their corresponding values (pole convergence). Upon convergence, the algorithm retrieves the corresponding residues in a final de-biasing step.

3.1. Pole filtering: sparsity-based L_1 *regularization*

Explain that threshold "skimming" has been proposed, but it is quite ad-hoc, and not universal. Then explain our method. Plus we do convex relaxation, making the problem simpler to solve.

At iteration (m) of the ReV-Fit algorithm, the pole convergence phase has yielded $N_p^{(m)}$ poles: $\left(p_n^{(\text{conv})}\right)_{n\in[\![1,N_p^{(m)}]\!]}$. The pole filtering phase then reduces the total number of poles $N_p^{(m+1)} \leq N_p^{(m)}$ by: 1. finding the residues that do not contribute significantly to the accuracy of the curve fitting (residues sparsification), and then 2. eliminating their corresponding poles (pole elimination), to produce the set of filtered poles $\left(p_n^{(\text{filt})}\right)_{n\in[\![1,N_p^{(m+1)}]\!]}$.

1. **Residues sparsification:** The converged poles $\left(p_n^{(\text{conv})}\right)_{n\in[\![1,N_p^{(m)}]\!]}$ are fixed (result of the feature extraction iteration), and the coefficients $\boldsymbol{K}^{(m)} \coloneqq \left(\boldsymbol{R_n}^{(m)}, \boldsymbol{C}^{(m)}, \boldsymbol{A}^{(m)}\right)$ are now obtained (feature selection) as the solution of the group LASSO sparsity-based $\mathrm{L}_1 - \mathrm{L}_2$ regularization problem (14), which enforces sparsity on the residue matrices $\boldsymbol{R}^{(m)} \coloneqq \left(\boldsymbol{R_1}, \ldots, \boldsymbol{R_n}, \ldots, \boldsymbol{R_{N_p^{(m)}}}\right)$:

$$\boldsymbol{K}^{(m+1)} = \underset{\boldsymbol{K}^{(m)}}{\arg\min}\ \mathcal{E}_{\text{LASSO}}^{(m)} \ \coloneqq\ \sum_{k=1}^{N_k} \rho_k \left\| \boldsymbol{H_r}^{(m)}(z_k) - \boldsymbol{H}(z_k) \right\|_{\mathrm{L}_2}^2 + \lambda^{(m)} \left\| \boldsymbol{R}^{(m)} \right\|_{\mathrm{L}_1-\mathrm{L}_2} \tag{14}$$

where $\left\|\boldsymbol{R}^{(m)}\right\|_{\mathrm{L}_1-\mathrm{L}_2} \coloneqq \sum_{n=1}^{N_p^{(m)}} \left\|\boldsymbol{R_n}^{(\mathrm{conv})}\right\|_{\mathrm{L}_2}$ is the group LASSO penalization.
This $\mathrm{L}_1-\mathrm{L}_2$ regularization (14) will drive low-incidence residues to zero, and we can then identify their associated poles as being superfluous and discard them. This will enable to both select the feature poles by dropping the spurious ones, and exhibit sparse matrix structure in the residues [NOT SO TRUE IF L1-L2]. TEST ALL COMPETING PROJECTS:

- Test LASSO = Sparce matrices but less poles elimination ?
- Test Group LASSO + Frobenius Norm = poles elimination with non sparse residues.
- Test Group LASSO + Nuclear Norm = poles elimination with low-rank residues approximations.
- Should we add the SK relaxed barycenric function here ? Might help ?

Solving the group Lasso regularization problem: There exists many iterative procedures to solve problem (14). The most basic one is the iterative forward backward splitting algorithm (FBS), which we here explicit in our case.
[PABLO FROM 9.520 class]

Numerical instability: Elastic net regularization
As mentioned in section 2.2, when the number of poles N_p is large the Cauchy matrices intervening in the L_2 optimization problems hitherto introduced can lead to severe numerical instability issues.
One way to palliate these shortcomings is to introduce a Tichonov L_2 regularizer, as proposed by Drmac [14, 15]. Such L_2 regularization fits naturally in our ReV-Fit algorithm: this corresponds to introducing a new meta-parameter μ to the LASSO residue sparsification problem (14), so as to provide numerical stability and obtaining the new coefficients $\mathbf{Coef_{res}}^{(m+1)}$ and residues $\boldsymbol{R}^{(m+1)}$ by solving instead the following Elastic net minimization problem:

$$\min \mathcal{E}^{(m)}_{\text{Elastic net}} \equiv \sum_{k=1}^{N_k} \rho_k \left\|\boldsymbol{H_r}^{(m)}(z_k) - \boldsymbol{H}(z_k)\right\|^2_{\mathrm{L}_2} + \lambda^{(m)} \left\|\boldsymbol{R}^{(m)}\right\|_{\mathrm{L}_1} + \mu^{(m)} \left\|\boldsymbol{H_r}^{(m)}(z_k)\right\|^2_{\mathrm{L}_2} \qquad (15)$$

This so called *Elastic Net* approach makes the space strictly convex, and thus standard convex optimization methods can be applied. However, this new-found ease in solving problem (15) might come at the cost of reduced selectivity. Indeed, the LASSO optimum usually drives redundant residues to machine-precision zero, while the Elastic net problem (15) will bring them to small, non-zero values, posing the question of what threshold to introduce to filter out the superfluous residues.

Again, a dynamic meta-optimization can be performed at each residue sparsification iteration (m) to best choose the values of $\mu^{(m)}$, which should be as small as possible to still correct the ill-conditioned matrices and numerical instability issues.

Meta-optimization: $\lambda^{(m)}$ selection When solving the LASSO problem (14), the choice of the meta-parameter $\lambda^{(m)}$ will determine the strength of the L_1 penalization: the greater the $\lambda^{(m)}$, the sparser the resulting residues $\boldsymbol{R}^{(m+1)}$. As hereafter explained in the pole elimination step 2, the poles corresponding to negligible residues are eliminated, so as to reduce the number of poles $N_p^{(m+1)} \leq N_p^{(m)}$. If the number of poles is too large, this leads to the over-fitting problem that damages accuracy by significantly reducing the generalization capability of the curve-fit. Thus, 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. This means that the meta-parameter $\lambda^{(m)}$ can itself be the subject of an optimization procedure, either to maximize the accuracy of the curve-fit through cross-validation, or to minimize the number of poles while still meeting some target accuracy requirement. This meta-optimization can take place at each new (m) iteration, meaning that as the ReV-Fit algorithm progresses, $\lambda^{(m)}$ is dynamically adapted and tailored to achieve an optimal performance at each new iteration of the residues sparsification step. This dynamic meta-optimization of $\lambda^{(m)}$ is best performed by choosing the $\lambda^{(m)}$ that minimizes the cross validation error on a cross-validation grid $(z_\ell^c)_{\ell \in [\![1, N_c]\!]}$:

$$\lambda^{(m)} \;=\; \arg\min \mathrm{CV}^{(m)} \tag{16}$$

where the cross validation can be performed on different metrics, depending on the problem. For instance, the L_2 norm cross validation is simply:

$$\mathrm{CV}^{(m)} \;=\; \sum_{\ell=1}^{N_c} \rho_\ell \left\| \boldsymbol{H_r}^{(m+1)}(z_\ell^c) - \boldsymbol{H}(z_\ell^c) \right\|_{\mathrm{L}_2}^2 \tag{17}$$

while choosing the L_∞ norm would yield a minimax problem.

Cross-validation is robust when computationally feasible, however, in the nuclear cross sections case studied bellow in section 4, the over-fitting problem introduces narrow but infinite resonances by adding real poles between the data points (c.f. figure 5). This entails that meta-optimizing $\lambda^{(m)}$ by selecting the best cross-validation performance is difficult (even through parametric bootstrapping methods as [35]), because the cross-validation data set $(z_\ell^c)_{\ell \in [\![1, N_c]\!]}$ has to be very fine to cover these over-fitting

resonances (or the cross validation data set could be drawn at random at each (m) iteration, but this would significantly increase the iterations to convergence).

We thus resort to traditional information theory measures, such as the *Akaike's Information Criterion* (AIC) [36], or its corrected-for-sample-size-bias version (AICc), and choose $\lambda^{(m)}$ such as [**?**]:

$$\lambda^{(m)} \;=\; \arg\min \mathrm{AICc}^{(m)} \tag{18}$$

where in our ReV-Fit algorithm, the $\mathrm{AICc}^{(m)}$ can be expressed, to an additive constant, as:

$$\mathrm{AICc}^{(m)} \;\equiv\; \frac{2N_k N_p^{(m+1)}}{N_k - N_p^{(m+1)} - 1} + \sum_{k=1}^{N_k} \rho_k \left\| \boldsymbol{H_r}^{(m+1)}(z_k) - \boldsymbol{H}(z_k) \right\|_{\mathrm{L}_2}^2 \tag{19}$$

where $N_p^{(m+1)}$ and $\boldsymbol{H_r}^{(m+1)}$ are the filtered poles & residues, yet without re-running a relaxed barycentric Sanathanan-Koerner pole converge — the same was true for the cross-validation (17). We also note that N_p, the number of poles, could be more accurately replaced with $N_p(1+D)$, where D is the dimension of the residue vectors (or matrices).

The ReV-Fit algorithm is here confronted with the traditional problem of choosing which criterion to penalize the excessive parameters that lead to over-fitting. A plethora of criteria are possible: AIC (and AICc); Schwarz's Bayesian Information Criterion (BIC) [37]; Hannan-Quinn information criterion (HQC) [38]; minimum description length (MDL) principle [39]; or the Widely applicable Bayesian Information Criterion (WBIC) [40, 41], for instance. Choosing which criterion to minimize in the meta-optimization step (18) is problem specific: the ReV-Fit user has to determine which criterion best suits the specific problem the algorithm is being used upon. We direct the reader to the extensive literature on this topic [42], and simply recall that the BIC penalizes more heavily additional poles and residues than the AIC does. The BIC is devised to consistently select the right model and dimensionality (poles and residues in our case) among a class of models if the data is exactly represented by one of the models. On the other hand, the AIC can be more performant when it comes to approximation and prediction, since AIC is established under the hypothesis that the true model requires an infinite amount of parameters, and seeks to minimize the loss of information when performing a finite-number-of-parameters approximation. Every criterion has its strength and weaknesses, which must be suited to the task. If possible, cross-validation is robust and does not require this criterion selection.

2. **Pole elimination:** The pole elimination step then simply consists of taking out the poles that correspond to residues that have been set to zero in the residues sparsification step (be it by LASSO,

Elastic net, or threshold sparsification). This means filtering-out a new, smaller set of relevant poles, $\left(p_n^{(i)}\right)_{n\in[\![1,N_p^{(m+1)}]\!]}$, with $N_p^{(m+1)} \leq N_p^{(m)}$.

3.2. Measuring convergence

In constructing out ReV-Fit algorithm, we have here introduced an additional iteration procedure to the RVF, posing the question of convergence on the (m) iterations and number of poles $N_p^{(m)}$.

$N_p^{(m)}$ being an integer number, the simplest approach is to stipulate that if $N_p^{(m)} = N_p^{(m+1)}$ for more than a given amount of consecutive iterations, the poles & residues filtering iteration has come to a halt.

Since the latter number of consecutive iterations to call for convergence is arbitrary, one can also measure the convergence by monitoring the stabilization of the information criterion $\mathrm{AICc}^{(m)}$ (or cross-validation $\mathrm{CV}^{(m)}$ when possible), terminating the poles & residues filtering iterations as:

$$\mathrm{AICc}^{(m)} \underset{m\to\infty}{\simeq} \mathrm{AICc}^{(m+1)} \tag{20}$$

3.3. Regularized relaxed vector fitting algorithm

After having introduced and explained the poles & residues filtering iterative procedure, we can now use it to construct the ReV-Fit algorithm [Algorithm 2].

- First, an excess of poles $N_p^{(0)} \gg N_{\mathrm{physical}}$ are given as initial guess: $\left(p_n^{(0)}\right)_{n\in[\![1,N_p^{(0)}]\!]}$. Usually, a log-spaced distribution is efficient to cover the span to be curve-fitted [7, 8].

- The poles are then pre-converged through the relaxed barycentric Sanathanan-Koerner poles convergence process — iterations (i). The convergence of this process can be ascribed through criterion (13). In the context of ReV-Fit, this criterion can eventually be relaxed compared to the RVF algorithm since it is only pre-converging the poles before the poles & residues filtering process.

- Once the poles have been pre-converged, the superfluous ones are filtered out through the poles & residues filtering process, described in section **??**. This process can be implemented in various ways: LASSO optimization (14); Elastic net (15); or hard threshold residues sparsification (**??**). All of these can be the object of further meta-optimization in selecting the regularization (or filtering) parameters (or thresholds), such as (18) or (**??**). In this article, we do not discuss specific algorithms for these meta-optimizations, which could be the subject of further developments. As a result of the poles & residues filtering iteration (m), the number of poles has diminished: $N_p^{(m+1)} \leq N_p^{(m)}$.

- This new set of filtered poles $\left(p_n^{(i)}\right)_{n\in[\![1,N_p^{(m+1)}]\!]}$ are then run again through a relaxed barycentric Sanathanan-Koerner poles convergence process (new set of (i) iterations), and then through a poles

& residues filtering step (m), until the entire process comes to convergence as the number of poles becomes stationary, $N_p^{(m+1)} = N_p^{(m)}$, and the chosen selection criterion $\mathrm{AICc}^{(m)}$ (or $\mathrm{CV}^{(m)}$) stabilizes (20).

- Finally, the residues are retrieved by solving problem (9) with the filtered-our poles $\left(p_n^{(i_{\mathrm{stop}})}\right)_{n\in[\![1,N_p^{(m_{\mathrm{stop}})}]\!]}$.

This constitutes the ReV-Fit algorithm [Algorithm 2].

Algorithm 2 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
repeat
1. POLES CONVERGENCE:
repeat
(a) **residue calculation:** fix $\left(p_n^{(i)}\right)_{n\in[\![1,N_p^{(m)}]\!]}$ and solve problem (9) to find the coefficients and residues $\mathbf{Coef_{res}}^{(i+1)}$.
(b) **pole relocation:** calculate $\left(p_n^{(i+1)}\right)_{n\in[\![1,N_p^{(m)}]\!]}$ by solving problem (10), or equivalently problem (12).
$i \leftarrow i+1$
until Local convergence criterion (13) is reached.
2. POLES & RESIDUES FILTERING:
(a) **residues sparsification:** fix poles from last iteration $\left(p_n^{(i_{\mathrm{stop}})}\right)_{n\in[\![1,N_p^{(m)}]\!]}$ and solve LASSO problem (14) with optimal $\lambda^{(m)}$ parameter from (18) — Elastic net (15) stabilized version or direct threshold elimination using (**??**) with optimal thresholds $\Xi^{(m)}$ from (**??**) can be used to circumvent numerical instability — to obtain coefficients and regularized residues of rational approximation $\boldsymbol{H_r}^{(m+1)}(z)$.
(b) **poles elimination:** reduce number of poles $N_p^{(m+1)} \leq N_p^{(m)}$ by eliminating the poles corresponding to null residues after sparsification: $\left(p_n^{(i)}\right)_{n\in[\![1,N_p^{(m+1)}]\!]}$
$m \leftarrow m+1$
until Global convergence on $N_p^{(m)}$ and poles & residues filtering criteria (20) are reached.
RESIDUES RETRIEVAL(de-biasing): fix poles from last iteration $\left(p_n^{(i_{\mathrm{stop}})}\right)_{n\in[\![1,N_p^{(m_{\mathrm{stop}})}]\!]}$ and solve problem (4) to obtain coefficients and residues of rational approximation $\boldsymbol{H_r}(z)$.

4. Application of ReV-Fit to nuclear cross sections in pole representation

A robust algorithm that can optimally generate rational fraction approximations of complex transfer matrices without over-fitting can find widespread use, for instance in signal treatment or physics. We here demonstrate our ReV-Fit algorithm's performance on the case of nuclear cross sections. Though this example does present challenging resonant structure, it does not deal with noisy data nor is applied to vectors or matrices. However, the ReV-Fit method is equally applicable to these cases, such as extracting MIMO surrogate matrix models from noisy data.

4.1. Nuclear cross sections

Nuclear cross sections are the basic input to calculating nuclear reaction rates, and as such are key to nuclear reactors physics, astrophysics, or medical imagery, among other fields. In reactor physics, nuclear cross section's energy dependence, displaying resonant structure spanning various orders of magnitude, is the source of many computational difficulties. A rational approximation to this complex energy dependence can be the source of significant progress on computational performance.

As computing power increases, Monte Carlo methods have been more widely used to simulate temperature dependence and coupling in nuclear reactors. 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 [43, 44].

Another approach is to compute the temperature dependence on-the-fly — that is directly during the Monte Carlo simulation — and to this effect was introduced the pole representation of nuclear cross sections [**?**]. Performing a pole decomposition of the fundamental R-matrix expressions for nuclear cross section, the pole representation expresses nuclear cross sections at 0 K as a the real part of a sum over poles (p_j) and residues (r_j), with a remaining function f that is locally analytic, and thus simple to estimate:

$$\sigma(E) = \Re\left[\frac{1}{E}\sum_j \frac{r_j}{\sqrt{E} - p_j}\right] + f(\sqrt{E}) \tag{21}$$

In addition to enabling the calculation of 0 K cross sections at any energy using the form (21), thus discarding the need for point-wise cross sections, a significant advantage of the pole representation is to provide an analytical way of Doppler broadening cross sections at different temperatures [**?**]. This means that the pole representation allows for an energy and temperature on-the-fly calculation of nuclear cross sections, thereby dramatically reducing nuclear cross section's memory overload [**?**].

Moreover, the windowing procedure was introduced to increase the efficiency of this pole representation. The windowing algorithm only explicitly calls local resonances and locally (that is in each window) curve-fits the aggregate effect of far-away resonances, combined with the remainder f. Doppler-broadening can then be performed on a much reduced set of resonances, yielding considerable performance gains [**? ?**].

One of the limiting steps for a more widespread use of the pole representation to calculate nuclear cross sections has been the difficulty of converting point-wise cross sections into poles and residues. Recently, a new method based on vector fitting (VF) was proposed to calculate poles and residues from the point-wise cross sections data, which can then be processed through the windowed pole representation method [**?**]. Such an approach could give the windowed pole representation method access to point-wise cross section,

thereby casting all needed nuclides in the same pole representation format. However, this VF approach required to arbitrarily prescribe ahead of time the number of poles N_p with which to fit the point-wise cross sections data, leading to all the shortcomings hereabove mentioned in section 2.2.

So as to palliate these shortfalls and the inherent over-fitting problem of the traditional VF approach, we developed the ReV-Fit algorithm, of which we implemented a version and applied it in this section to nuclear cross sections' pole representation. In order to match the pole representation form (21), the ReV-Fit algorithm was applied on the quantity $E \cdot \sigma(E)$, and the poles were paired by complex conjugates.

With our implementation of the ReV-Fit algorithm, we first processed nuclide ^{199}Hg of mercury, which has known poles and residues, and from these we generated point-wise cross sections so as to verify if the physical poles can be reconstructed correctly using our ReV-Fit method on the cross section data. We then considered the gadolinium ^{157}Gd isotope, for which only point-wise data is available, and applied the ReV-Fit algorithm to convert the point-wise data into pole representation. The accuracy of this ReV-Fit conversion to pole representation is then tested by direct comparison of the cross sections on a fine energy mesh.

4.2. Implementation of the ReV-Fit algorithm

As described in section 3.3, the ReV-Fit algorithm can come in different flavors, with the possibility to add various methods in the poles & residues filtering step to mitigate potential Cauchy-matrices ill-conditioning in the LASSO optimization (14), through Elastic Net (15) or hard thresholding residues sparsification (**??**)) — all of which can be subject to meta-optimization (18) or (**??**) with different possible metrics ($\mathrm{CV}^{(\mathrm{m})}$, $\mathrm{AICc}^{(\mathrm{m})}$, etc.) — and where different thresholds can be used for pre-convergence (13) or convergence (20).

We here describe the implementation of ReV-Fit we applied to the nuclear cross sections case. This implementation was deliberately chosen from readily available packages and software, for ease of implementation and reproducibility purposes.

- For the relaxed barycentric Sanathanan-Koerner poles convergence iterations, the original Relaxed Vector Fitting (RVF) package was used, VFIT3.zip, directely downloaded from the website (*https://www.sintef.no/projectweb/vectfit/downloads/vfut3/*). In this VFIT3 implementation of RVF, convergence criterion (13) is not implemented. Instead, one has to specify the number of pole relocation iterations [8]. In our implementation, we stuck to the suggested number of $(i^{\mathrm{stop}}) = 10$.

- For the poles & residues filtering step, the MATLAB VFIT3 code was modified to add an (m) iteration calling the built-in MATLAB "LASSO" function. This MATLAB function solves the LASSO regularization problem (14) in the form $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, taking as imputs $(\boldsymbol{A}, \boldsymbol{b}, \lambda)$.

- Selection of the $\lambda^{(m)}$ meta-parameter was done "by hand", solving for either $\mathrm{CV}^{(m)}$ or $\mathrm{AICc}^{(m)}$ for 19 values of $\lambda^{(m)}$ from 10^{-9} to 1, and choosing the $\lambda^{(m)}$ value that yielded the smallest corresponding value from equation (17) or (19), respectively. It is likely that an optimal choice of $\lambda^{(m)}$ would have reduced the number of iterations (m_{stop}) to convergence.

- The convergence criterion to terminate the ReV-Fit algorithm was simply chosen to be a stationary pole number $N_p^{(m+1)} = N_p^{(m)}$ for 20 successive iterations.

- Finally, the residues retrieval step was performed as in VFIT3, using the Moore-Penrose pseudo-inverse "\". Let us note that, as expected from section 2.1, item 2, this inversion applies to ill-conditioned Cauchy-matrices and is prone to high condition numbers and numerical instability. VFIT3 issues a warning statement in this case. For demonstration purposes, we still proceeded with the answer, aware that this might have hindered the accuracy of the final results. We refer to the various strategies proposed above to mitigate this high-condition number inversion problem (section 2.2).

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

There are three most naturally occurring isotopes of mercury, and they all have simple resonance structures. ^{202}Hg, representing 29.74% of natural occurrence, counts only 2 resonances; ^{200}Hg, representing 23,14% of natural occurrence, has 4 resonances and ^{199}Hg, representing 16.94% of natural occurrence, has 10 resonances. Due to its clear and simple resonance structure, we tested our ReV-Fit algorithm on the radiative capture cross section of ^{199}Hg, focusing the energy range on the resolved resonance region from 0.01eV to 968eV.

This test process is divided into three steps as follows:

1. **Pole representation generation:** The WHOPPER code, developed by Hwang [**?**], is used to directly convert resonance parameters (from ENDF/B-VII) of the radiative capture cross section of ^{199}Hg into physical poles and residues [**?**]. Then RVF is adopted 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 rest represent pseudo-poles. This set of poles and residues becomes our reference solution.
2. **Point-wise data generation:** Using these reference poles and residues, point-wise nuclear cross section data can be generated by means of equation (21). In order to represent resonance structure, a set of energy points must be fixed to construct the basic energy grid as shown in figure 1. Then point-wise cross section data can be generated by refining the basic energy grid, i.e. adding more energy points between basic energy points. Energy grid density (E_{gd}) is defined as:

$$E_{\text{gd}} = \log_2 N_{\text{grid}} \tag{22}$$

where N_{grid} is the number of energy points between two neighboring basic energy points (red points in left figure 1), which are resonance peeks or inflection points when in absence of resonances.

3. **Poles 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 tested 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.

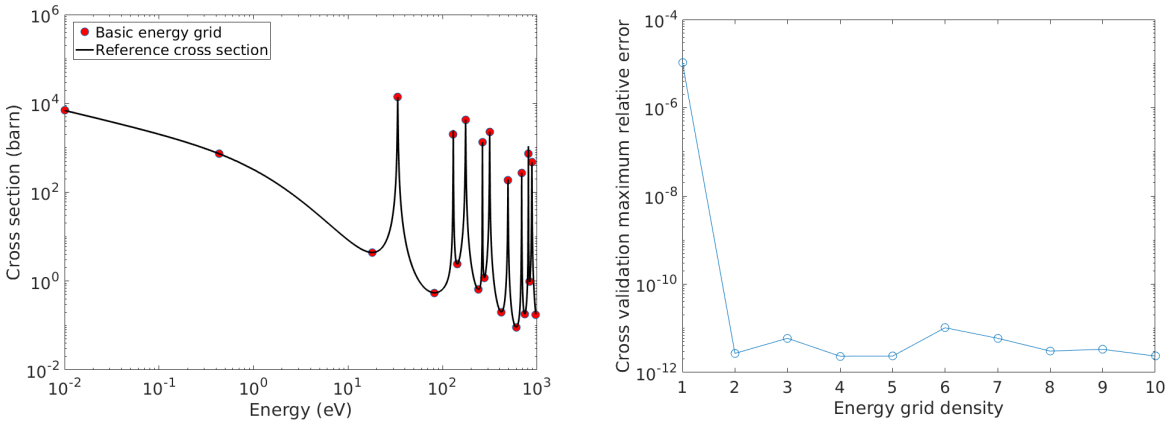

Figure 1: Left: Basic energy grid from which is built the point-wise cross section.
Right: Cross Validation error from RVF applied to different energy densities, provided the ^{199}Hg physically correct number of 27 log-spaced poles as initial guess.

In order to test what minimal energy grid density should be provided, the RVF algorithm was given the physically correct number of 27 log-spaced poles as initial guess, and run on 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 the right of figure 1. One can observe that if the density of the energy grid is too small, the RVF algorithm fails to converge to the correct poles & residues, even though it was given the correct number of poles. This phenomenon can be observed in figure 2.

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 3.

Having determined the appropriate energy grid density for the problem, we then proceed to quantify the errors introduced by over-fitting. The energy grid is fixed to a sufficient density of points, and the number

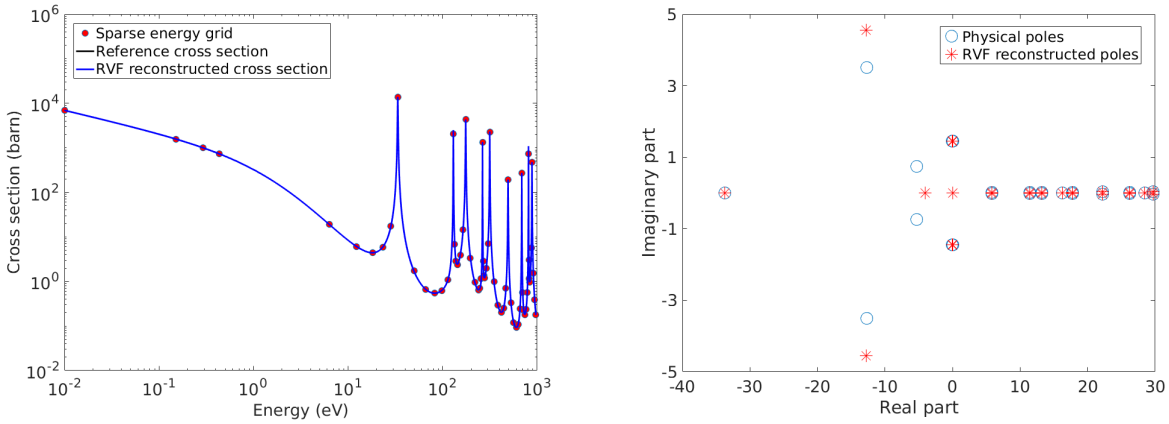

Figure 2: The RVF algorithm was provided the ^{199}Hg physically correct number of 27 log-spaced poles as initial guess, on energy grid density No.1. **Left:** Exact cross section and RVF curve fit. **Right:** Exact poles vs. poles found by RVF.

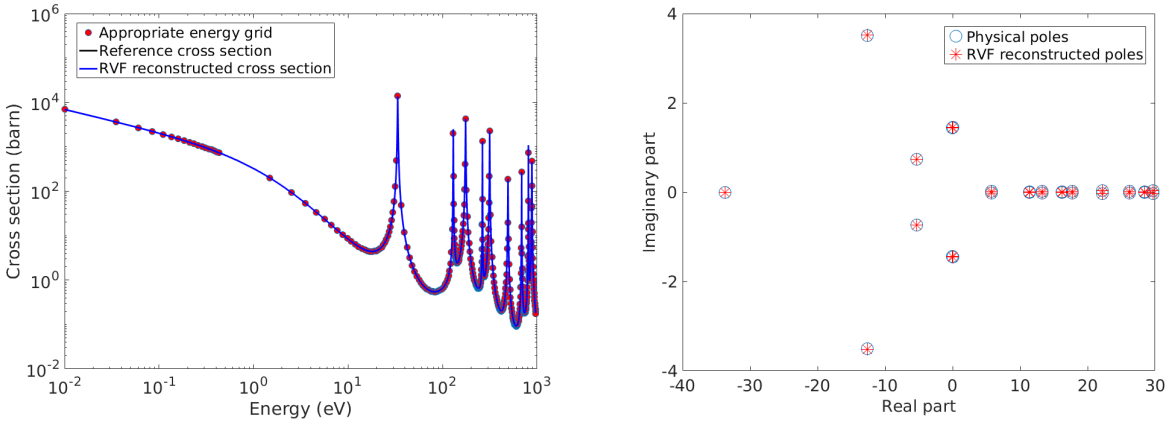

Figure 3: The RVF algorithm was provided the ^{199}Hg physically correct number of 27 log-spaced poles as initial guess, on energy grid density No.4. **Left:** Exact cross section and RVF curve fit. **Right:** Exact poles vs. poles found by RVF.

of log-spaced initial poles given to the RVF algorithm is varied. One can observe in figure 4 that the cross validation error reaches a minimum at the physical number of poles, past which the RVF algorithm starts to over-fit, which significantly damages the performance, loosing 3 orders of magnitude of accuracy with just 3 additional poles.

This over-fitting phenomenon can be seen in action in figure 5, 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, because it is "over crowded", and places some poles on the real axis, generating huge, very narrow resonances. Since they are narrow and between the energy mesh points, these resonances do not affect the likelihood of the results on the mesh, and thus not the AICc$^{(m)}$, but they do considerably hinder the cross-validation

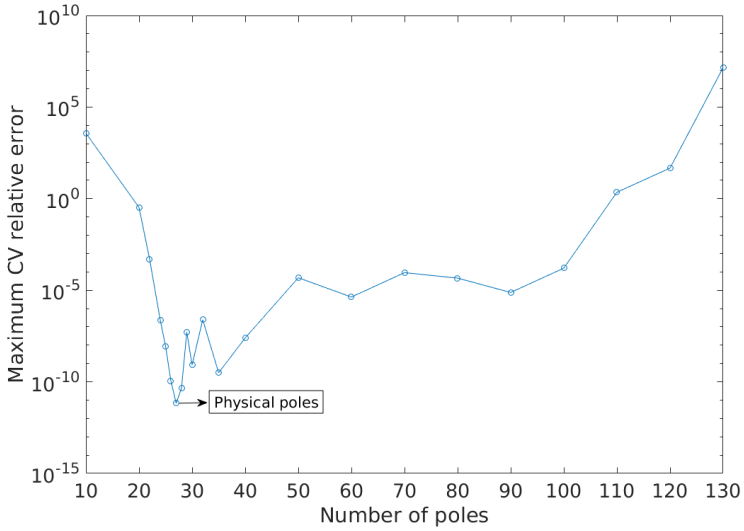

Figure 4: Change of cross validation maximum relative error for different number of poles

$CV^{(m)}$ performance. However, we can here observe that the cross-validation grid has to be very refined to measure these narrow real resonances, which can explain the use of an $AICc^{(m)}$ criterion instead of $CV^{(m)}$, as explained in section **??**.

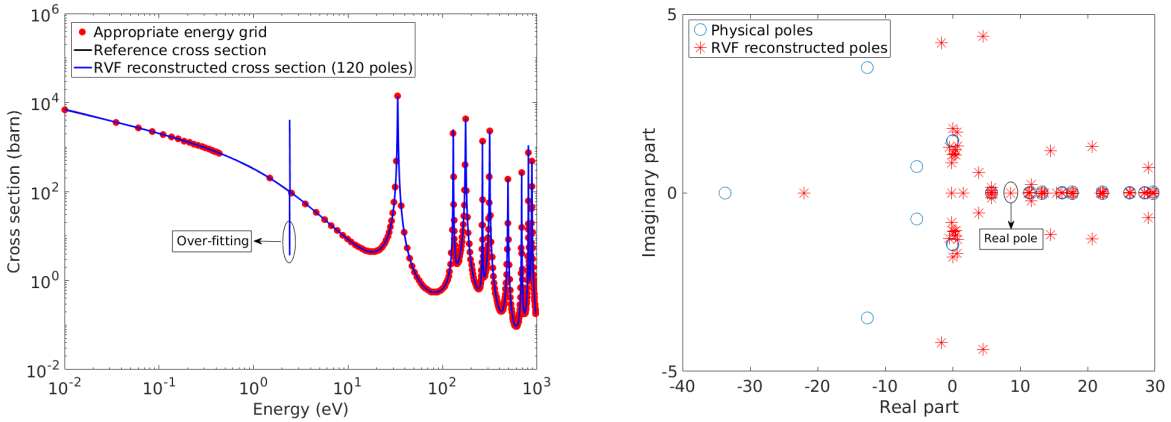

Figure 5: The RVF algorithm was provided 120 log-spaced poles as initial guess, on energy grid density No.4. **Left:** Exact cross section and RVF curve fit. **Right:** Exact poles vs. poles found by RVF.

4.4. ReV-Fit performance in eliminating over-fitting

In section 4.3, we quantitatively showed that the RVF over-fitting problem can lead to severe loss of extrapolating power. In this section we demonstrate on the test case of ^{199}Hg how our ReV-Fit algorithm is able to solve this over-fitting problem, even in with a simple implementation.

The performance of our ReV-Fit algorithm is tested on the ^{199}Hg cross section, whose exact poles we know. The energy grid density used for the ReV-Fit is $\log_2 20$, and the cross-validation energy grid density equals 4. 200 log-spaced initial poles were fed to the ReV-Fit algorithm.

To assess the effect of the choice of criterion, our ReV-Fit algorithm was run twice on the ^{199}Hg cross section with the same 200 initial log-spaced poles: once using, on a cross-validation mesh, $\mathrm{CV}^{(m)}$ as a criterion for $\lambda^{(m)}$ selection (16); and one using the $\mathrm{AICc}^{(m)}$ for $\lambda^{(m)}$ selection (18). As the ReV-Fit algorithm progresses, the evolution of the number of poles $N_p^{(m)}$, the corrected Akaike information criterion $\mathrm{AICc}^{(m)}$, and the cross validation error $\mathrm{CV}^{(m)}$, are monitored and compared for both choices of criterion, and the results recorded in figures 6 and 7.

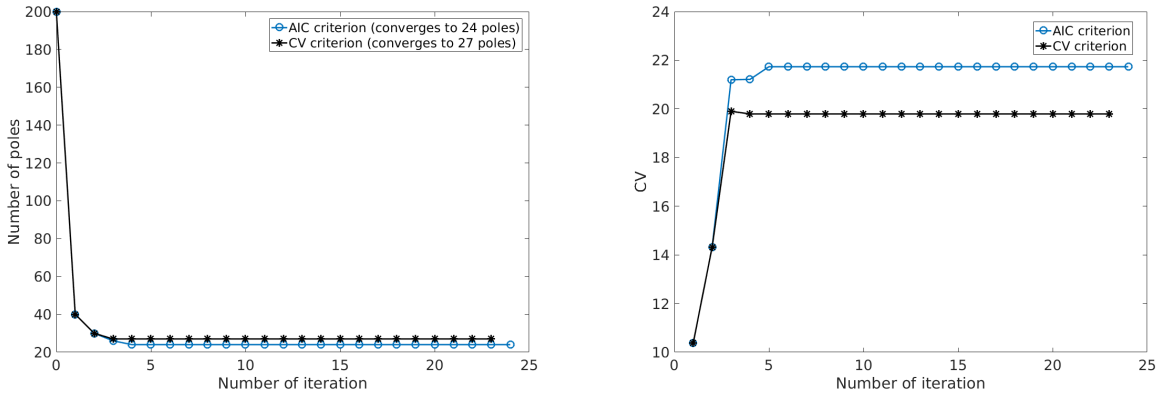

Figure 6: The ReV-Fit algorithm was provided 200 log-spaced poles as initial guess, on energy grid density equals $\log_2 20$, and run twice on ^{199}Hg: one with cross validation $\mathrm{CV}^{(m)}$ criterion, once with $\mathrm{AICc}^{(m)}$.
Left: Evolution of number of poles $N_p^{(m)}$ as poles and filtering iterations (m) progress. **Right:** Evolution of cross validation error $\mathrm{CV}^{(m)}$ as poles and filtering iterations (m) progress.

One can observe that, starting from 200 log-spaced poles, the number of poles decreases fast throughout the first ReV-Fit iterations (left figure 6). Remarkably, the cross-validation $\mathrm{CV}^{(m)}$ criterion systematically converges to the physically exact number of 27 poles, and finds their exact physical location, as it can be seen in figure 8. Perhaps surprisingly, the $\mathrm{AICc}^{(m)}$ over-restricted the number of poles, converging to 24 poles, despite the fact that the Aikaike Information Criterion is known to be sometimes prone to over-fitting. This is probably due to the particular type of over-fitting that the real poles introduce. In figure 7, one can observe that the value of $\lambda^{(m)}$ decreases over time and converges rapidly to the same value for both $\mathrm{CV}^{(m)}$ and $\mathrm{AICc}^{(m)}$ criteria. Thus, the "over-filtering" observed in the use of $\mathrm{AICc}^{(m)}$ criterion is due to the fact that the $\lambda^{(m)}$ value stays higher for two successive iterations in the case of $\mathrm{AICc}^{(m)}$. It is thus possible that a more systematic optimization method for $\lambda^{(m)}$ on a finer $\lambda^{(m)}$-grid might have yielded better agreement between $\mathrm{CV}^{(m)}$ and $\mathrm{AICc}^{(m)}$.

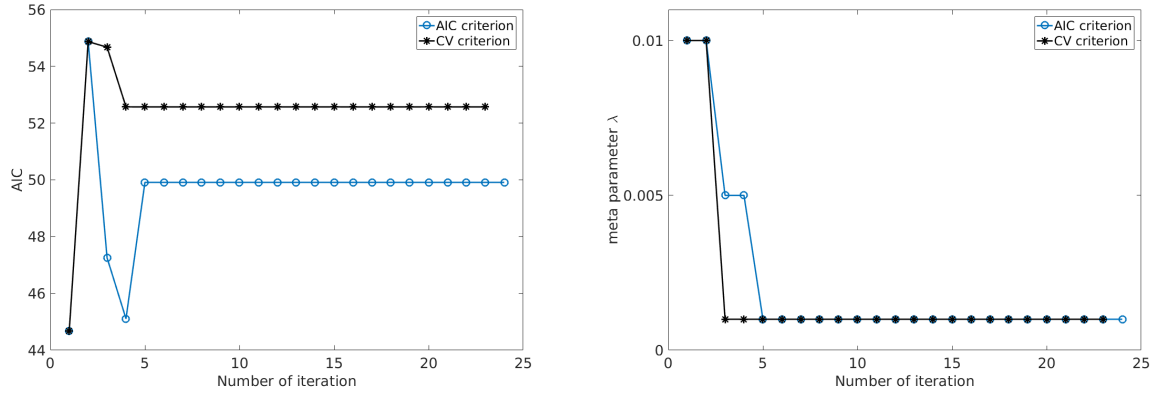

Figure 7: The ReV-Fit algorithm was provided 200 log-spaced poles as initial guess, on energy grid density equals $\log_2 20$, and run twice on ^{199}Hg: once with cross validation $\mathrm{CV}^{(m)}$ criterion, once with $\mathrm{AICc}^{(m)}$.
Left: Evolution of corrected Akaike information criterion $\mathrm{AICc}^{(m)}$ as poles and filtering iterations (m) progress.
Right: Evolution of meta-parameter $\lambda^{(m)}$ as poles and filtering iterations (m) progress.

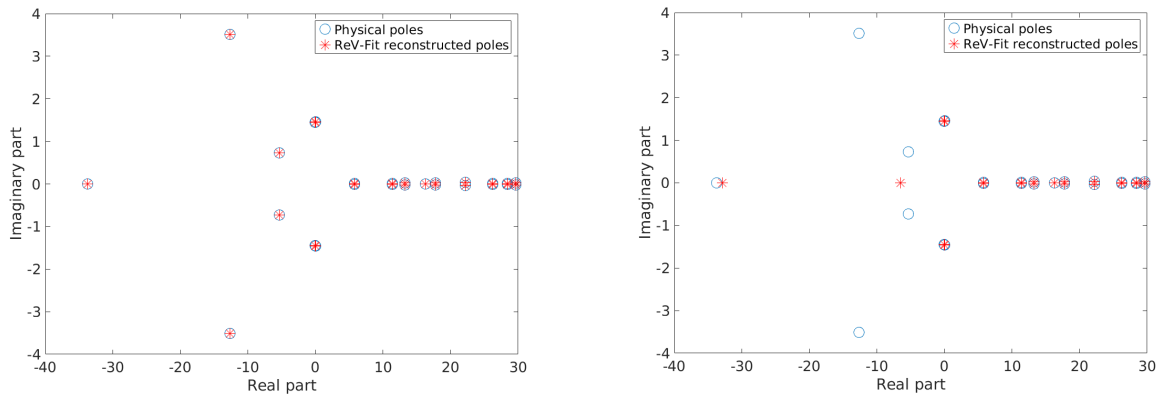

Figure 8: The ReV-Fit algorithm was provided 200 log-spaced poles as initial guess, on energy grid density equals $\log_2 20$. **Left:** Exact poles vs. poles found by ReV-Fit with CV. **Right:** Exact poles vs. poles found by ReV-Fit with AICc.

As it will be noted in section 4.5, this discrepancy between the criteria translates into a far better performance on the cross-validation $\mathrm{CV}^{(m)}$ criterion on final maximum relative error between the exact and the ReV-Fit cross section. We thus encourage the use of $\mathrm{CV}^{(m)}$ meta-optimization whenever computationally feasible.

This test-case on ^{199}Hg shows that the ReV-Fit algorithm is capable to systematically solve the over-fitting problem, finding the correct number N_p of poles, and converging to the exact residues, even though the number of the initial poles (200) was much larger than that of the physical model (27).

4.5. ReV-Fit for robust curve-fitting of nuclear cross sections

Now that we have established the performance of the ReV-Fit algorithm on the test case of mercury isotope ^{199}Hg, and showed that the use of the CV$^{(m)}$ criterion leads ReV-Fit to converge to the exact poles and residues, we here apply our ReV-Fit algorithm to establish a pole representation of both mercury ^{199}Hg and Gadolinium ^{157}Gd. Since the poles of the latter are not available from ENDF/B-VII, this will demonstrate the ability of the ReV-Fit algorithm to construct pole representation of cross sections from point-wise data.

4.5.1. Hg-199 case

The final ReV-Fit rational approximation to the mercury ^{199}Hg cross sections are reported in figures 9 and 10, respectively using the CV$^{(m)}$ and the AICc$^{(m)}$ criteria.

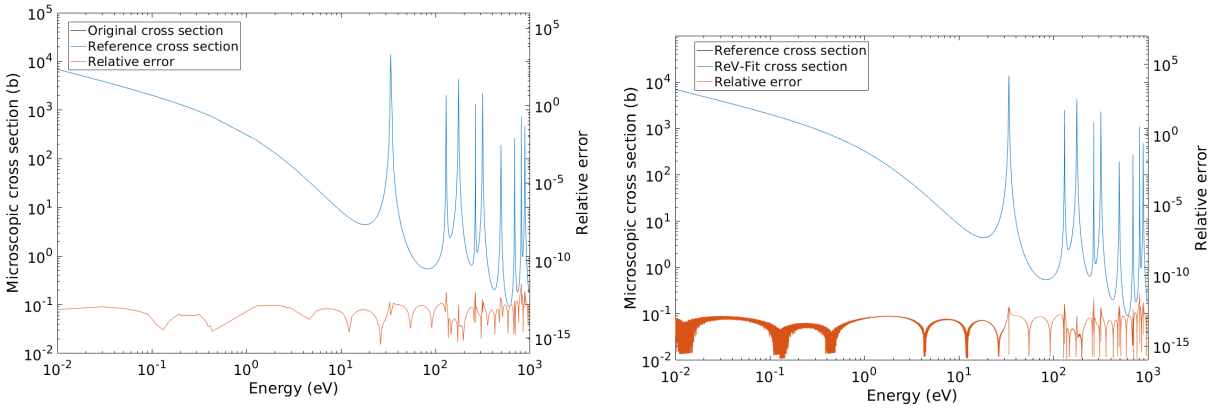

Figure 9: The ReV-Fit algorithm was provided 200 log-spaced poles as initial guess, on energy grid density equals $\log_2 20$, and meta-optimized using cross validation CV$^{(m)}$ criterion. **Left:** Exact capture cross section and ReV-Fit curve fit on original grid. **Right:** Exact capture cross section and ReV-Fit curve fit on cross-validation grid.

One can observe that in the case of cross-validation CV$^{(m)}$ meta-optimization, the maximum relative error between the exact and ReV-Fit cross sections — that is $\max_E \left\{\frac{\sigma_{\text{Exact}}(E)-\sigma_{\text{ReV-Fit}}(E)}{\sigma_{\text{Exact}}(E)}\right\}$ — is close to 10^{-12}, for both the fitting and cross-validation energy grids. In the case of AICc$^{(m)}$ meta-optimization, which had yielded only 24 poles, the ReV-Fit algorithm still achieves a 10^{-7} maximum relative error on the fitting energy grid, while 10^{-6} on the cross validation error. This performance is still way superior to traditionally used 0.1% accuracy threshold when constructing point-wise data, and it was thus deemed sufficient to use the AICc$^{(m)}$ criterion on the case of gadolinium isotope ^{157}Gd, in order to reduce the computational burden.

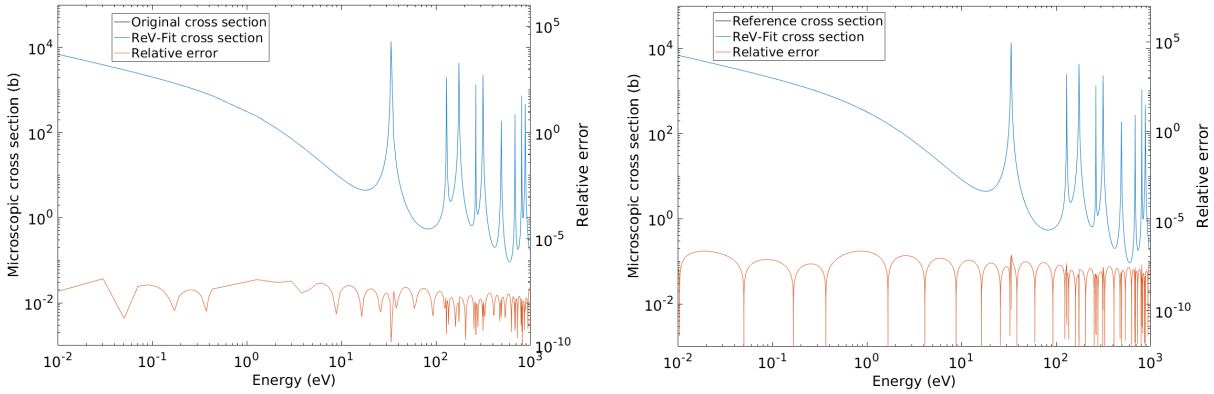

Figure 10: The ReV-Fit algorithm was provided 200 log-spaced poles as initial guess, on energy grid density equals $\log_2 20$, and meta-optimized using AICc AICc$^{(m)}$ criterion. **Left:** Exact cross section and ReV-Fit curve fit on original grid. **Right:** Exact cross section and ReV-Fit curve fit on cross-validation grid.

4.5.2. Gd-157 case

Burnable poisons are commonly used in nuclear fuel or coolant to even out the performance of the reactor over time, from fresh fuel being loaded to refuelling. These are neutron absorbers which decay under neutron exposure, compensating for the progressive build up of neutron absorbers in the fuel as it is burned. The best known is gadolinium due to its very high neutron absorption cross section of two isotopes: ^{155}Gd and ^{157}Gd. In this section, the ReV-Fit algorithm is applied to convert the point-wise cross section data of ^{157}Gd stored in ENDF/B-VII into pole representation. The ReV-Fit algorithm was fed 300 log-spaced initial poles, and, for computation reasons, was meta-optimized using the AICc$^{(m)}$ criterion instead of cross-validation CV$^{(m)}$.

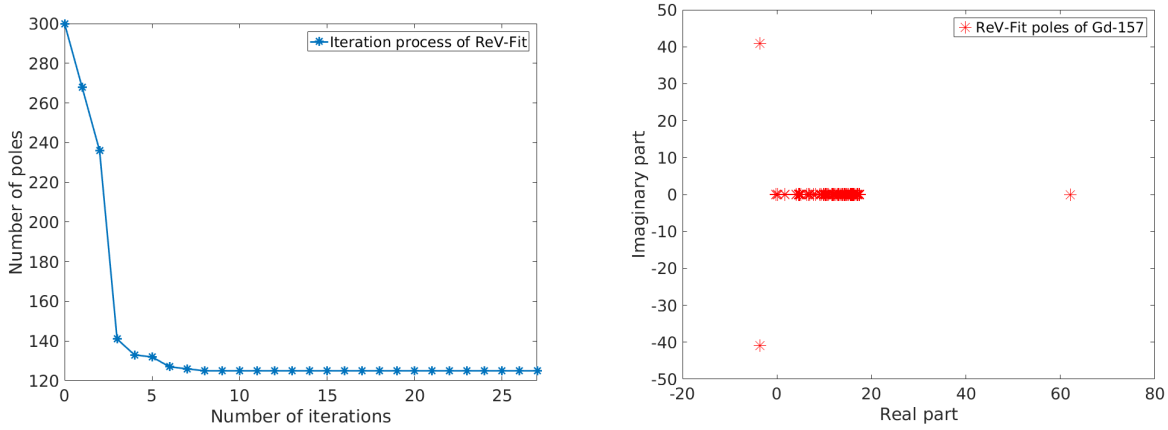

Figure 11: The ReV-Fit algorithm was provided 300 log-spaced poles as initial guess to reconstruct ^{157}Gd, with AICc$^{(m)}$ meta-optimization. **Left:** Evolution of number of poles $N_p^{(m)}$ as poles and filtering iterations (m) progress. **Right:** Final ReV-Fit poles on ^{157}Gd.

The evolution of the number of poles $N_p^{(m)}$ was monitored and recorded in figure 11 (left), 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 on the right part of figure 11. Most of the poles occur in the form of conjugate pairs and near the real axis and resonance, while a pair of far-away poles accounts for the background behavior at low energies. A single pole is found on the real axis, but was not eliminated by ReV-Fit because it is far beyond the energy range considered.

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.

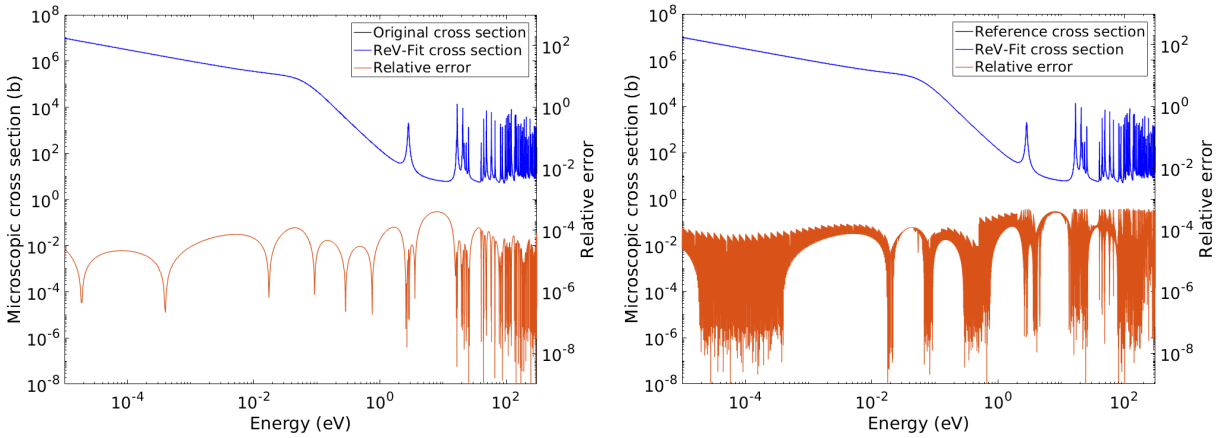

Figure 12: The ReV-Fit algorithm was provided 300 log-spaced poles as initial guess to reconstruct ^{157}Gd, with AICc$^{(m)}$ meta-optimization. **Left:** Exact total cross section and ReV-Fit curve fit on original grid. **Right:** Exact total cross section and ReV-Fit curve fit on cross-validation grid.

The accuracy achieved is of the order of 0.1% maximum relative error over the entire energy range. This is sufficient to meet traditional accuracy requirements, but it is likely that using a more computationally costly meta-optimization through cross-validation CV$^{(m)}$ criterion would have lead to more accurate ReV-Fit cross sections.

4.5.3. 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. This new capability opens the entire charted nuclides to fast on-the-fly 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 [45], opening perspectives for parallel implementations of ReV-Fit.

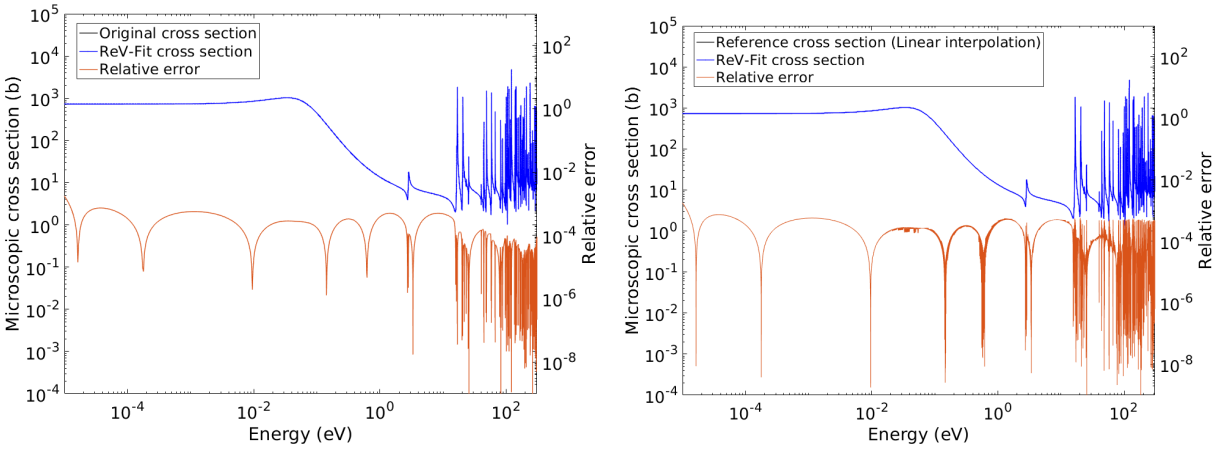

Figure 13: The ReV-Fit algorithm was provided 300 log-spaced poles as initial guess to reconstruct ^{157}Gd, with AICc$^{(m)}$ meta-optimization. **Left:** Exact scattering cross section and ReV-Fit curve fit on original grid. **Right:** Exact scattering cross section and ReV-Fit curve fit on cross-validation grid.

The ReV-Fit algorithm may also be applied to simplify the windowed pole representation, by providing a rational approximation of the background term f in equation (21), instead of the current Laurent expansion [**?**]. In this approach, the windowed pole representation would always be cast, in each window $\mathcal{V}(E)$, in the simple form:

$$\sigma(E) \underset{\mathcal{V}(E)}{=} \Re\left[\frac{1}{E}\sum_j \frac{s_j}{E-\epsilon_j}\right] \tag{23}$$

Well chosen constraints could also enforce continuity from window to window 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 computation performance.

5. Conclusions

Vector Fitting (VF) has been widely used as a rational fraction curve-fitting method for over a decade [7]. Because of its reliability and versatility, VF has been applied and extended to a number of areas. Later, the relaxed vector fitting algorithm (RVF) was introduced to improve the ability to relocate poles and to alleviate certain convergence problems [8]. Various other derivatives of VF were later introduced [10, 12, 15]. However, all previous versions of VF required to specify ahead of time the number of poles N_p with which to curve-fit. This major limitation exposed VF to over-fitting problems, hindering the extrapolating power of its curve-fits.

In this article, we introduce a new, regularized relaxed vector fitting (ReV-Fit) algorithm, that aims at solving RVF's over-fitting problem by automatically reducing the number of poles N_p until it converges to a physically accurate order. ReV-Fit builds upon the RVF algorithm and introduces standard machine learning regularization methods through a LASSO sparsification step that eliminates the superfluous poles and residues, providing a systematic way to deal with the problem of model order estimation and reduction in RVF.

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 physical poles could be 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 was shown to be able to converge to the correct physical number of poles $N_p = 27$, and find the exact physical pole positions, when the meta-optimization was performed through cross-validation. To ease the computing burden, the ReV-Fit algorithm was then applied on ^{157}Gd — or which only point-wise cross sections are provided in ENDF/B-VII.1 — using a corrected Akaike information criterion meta-optimization. The relative errors between the reference point-wise cross sections and the ReV-Fit reconstructed cross sections over both fitting and cross validation energy meshes were mostly below 0.1%, and are expected to be better if cross-validation is used for meta-optimization of the ReV-Fit. These results show that the ReV-Fit algorithm can be effectively applied to generate pole representations of nuclides for which no resonance parameters are provided in the ENDF/B-VII.1 evaluations.

This very promising performance of the ReV-Fit algorithm points towards applying it to various fields, such a signal treatment, macro-model reduction, or multi-input/multi-output (MIMO) dynamical systems, where the ReV-Fit algorithm could find myriad applications.

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. Peng, X.-j. was also partly supported by China Scholarship Council.

References

[1] D. Deschrijver, T. Dhaene, A note on the multiplicity of poles in the vector fitting macromodeling method, IEEE Transactions on Microwave Theory and Techniques 55 (4) (2007) 736–741, DOI: 10.1109/TMTT.2007.893651.

[2] C.-u. Lei, W. Y.-z., Q. Chen, N. Wong, A decade of vector fitting development: Applications on signal/power integrity, Vol. 1285, 2010, p. 435, http://dx.doi.org/10.1063/1.3510567.

[3] D. Deschrijver, T. Dhaene, Vol. 13 of Mathematics in Industry, Springer, 2008, section 16, Data-driven Model Order Reduction using Orthonormal Vector Fitting, pp. 341–359, iSBN 978-3-540-78840-9.

[4] F. Ferranti, D. Deschrijver, L. Knockaert, T. Dhaene, Vol. 72 of Lecture Notes in Electrical Engineering, Springer, 2011, section 7, Data-Driven Parameterized Model Order Reduction Using z-Domain Multivariate Orthonormal Vector Fitting Technique, pp. 141–148, iSBN 978-94-007-0088-8.

[5] E. Richards, 2013, https://books.google.com/books?id=SKF9nQAACAAJ.

[6] S. Grivet-Talocia, B. Gustavsen, John Wiley Sons, Inc, Hoboken, NJ, USA., 2015, Ch. 8, dOI: 10.1002/9781119140931.

[7] 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.

[8] 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.

[9] B. Gustavsen, C. Heitz, Modal vector fitting: A tool for generating rational models of high accuracy with arbitrary terminal conditions, IEEE Transactions on Advanced Packaging 31 (4) (2008) 664–672, DOI: 10.1109/TADVP.2008.927810.

[10] B. Gustavsen, C. Heitz, Fast realization of the modal vector fitting method for rational modeling with accurate representation of small eigenvalues, 2009, pp. 1–1, DOI: 10.1109/PES.2009.5275736.

[11] D. Deschrijver, B. Haegeman, T. Dhaene, Orthonormal vector fitting: A robust macromodeling tool for rational approximation of frequency domain responses, IEEE Transactions on Advanced Packaging 30 (2) (2007) 216–225, DOI: 10.1109/TADVP.2006.879429.

[12] D. Deschrijver, T. Dhaene, D. De Zutter, Robust parametric macromodeling using multivariate orthonormal vector fitting, IEEE Transactions on Microwave Theory and Techniques 56 (7) (2008) 1661–1667, DOI: 10.1109/TMTT.2008.924346.

[13] S. Grivet-Talocia, M. Bandinu, Improving the convergence of vector fitting for equivalent circuit extraction from noisy frequency responses, IEEE Transactions on Electromagnetic Compatibility 48 (1) (2006) 104–120, DOI: 10.1109/TEMC.2006.870814.

[14] Z. Drmac, S. Gugercin, C. Beattie, Quadrature-based vector fitting for discretized h2 approximation, SIAM Journal of Scientific Computing 37 (2) (2015) A625–A652, DOI: 10.1137/140961511.

[15] Z. Drmac, S. Gugercin, C. Beattie, Vector fitting for matrix-valued rational approximation, SIAM Journal of Scientific Computing 37 (5) (2015) A346–A379, DOI. 10.1137/15M1010774.

[16] L. Knockaert, D. De Zutter, Well-conditioned adaptive interpolation by a gaussian-modulated pole kernel with applications to vector fitting, 2007, pp. 57–60, DOI: 10.1109/SPI.2007.4512208.

[17] A. Ubolli, B. Gustavsen, A digital filtering approach for time domain vector fitting, 2011, pp. 25–28, DOI: 10.1109/SPI.2011.5898833.

[18] D. Deschrijver, L. Knockaert, T. Dhaene, Improving robustness of vector fitting to outliers in data, Electronics Letters 46 (17) (2010) 1200–1201, DOI: 10.1049/el.2010.1364.

[19] L. De Tommasi, B. Gustavsen, T. Dhaene, Robust transfer function identification via an enhanced magnitude vector fitting algorithm, IET Control Theory Applications 4 (7) (2010) 1169–1178, DOI: 10.1049/iet-cta.2009.0025.

[20] T. Dhaene, D. Deschrijver, Stable parametric macromodeling using a recursive implementation of the vector fitting algorithm, IEEE Microwave and Wireless Components Letters 19 (2) (2009) 59–61, DOI: 10.1109/LMWC.2008.2011304.

[21] S.-J. Se-Jung Moon, A. C. Cangellaris, Order estimation for time-domain vector fitting, 2009, pp. 69–72, DOI: 10.1109/EPEPS.2009.5338474.

[22] S. Lefteriu, A. C. Antoulas, Connecting vector fitting to barycentric interpolation and the loewner matrix, I2013 IEEE 22nd Conference on Electrical Performance of Electronic Packaging and Systems (2013) 133–136DOI:

10.1109/EPEPS.2013.6703483.

[23] G.-Y. Shi, On the nonconvergence of the vector fitting algorithm, IEEE Transactions on Circuits and Systems II: Express Briefs 63 (8) (2016) 718–722, DOI: 10.1109/TCSII.2016.2531127.

[24] S. Lefteriu, A. C. Antoulas, On the convergence of the vector-fitting algorithm, IEEE Transactions on Microwave Theory and Techniques 61 (4) (2013) 1435–1443, DOI: 10.1109/TMTT.2013.2246526.

[25] F. Ferranti, Y. Rolain, L. Knockaert, T. Dhaene, Variance weighted vector fitting for noisy frequency responses, IEEE Microwave and Wireless Components Letters 20 (4) (2010) 187–189, DOI: 10.1109/LMWC.2010.2042546.

[26] A. Beygi, A. Dounavis, An instrumental variable vector-fitting approach for noisy frequency responses, IEEE Transactions on Microwave Theory and Techniques 60 (9) (2012) 2702–2712, DOI: 10.1109/TMTT.2012.2206399.

[27] D. Deschrijver, L. Knockaert, T. Dhaene, A barycentric vector fitting algorithm for efficient macromodeling of linear multi-port systems, IEEE Microwave and Wireless Components Letters 23 (2) (2013) 60–62, DOI: 10.1109/LMWC.2013.2239282.

[28] I. R. Pordanjani, W.-s. Xu, Improvement of vector fitting by using a new method for selection of starting poles, Electric Power Systems Research 107 (2014) 206–212, http://dx.doi.org/10.1016/j.epsr.2013.10.005.

[29] F. Cucker, H.-I. Diao, Mixed and componentwise condition numbers for rectangular structured matrices, CALCOLO 44 (2007) 89–115, DOI: 10.1007/s10092-007-0130-3.

[30] M. M. Gourary, S. G. Rusakov, S. L. Ulyanov, M. M. Zharov, A technique to accelerate the vector fitting algorithm for interconnect simulation, 2010, pp. 127–130, DOI: 10.1109/EWDTS.2010.5742092.

[31] L. Zhang, W. Wang, Q. Li, W. H. Siew, An improved vector fitting method for parametric identification of transfer functions in emc measurement, 2007, pp. 174–177, DOI: 10.1109/ELMAGC.2007.4413459.

[32] M. Benzi, Preconditioning techniques for large linear systems: A survey, Journal of Computational Physics 182 (2002) 418–477, DOI: 10.1006/jcph.2002.7176.

[33] A. V. Knyazev, A preconditioned conjugate gradient method for eigenvalue problems and its implementation in a subspace, International Series of Numerical Mathematics 96 (1991) 143–154, DOI: 10.1007/978-3-0348-6332-2$_1$1.

[34] D. Calvetti, L. Reichel, On the solution of cauchy systems of equations, Electronic Transactions on Numerical Analysis 4 (1996) 125–137.

[35] E.-J. Wagenmakers, R. Ratcliff, P. Gomez, G. J. Iverson, Assessing model mimicry using parametric bootstrap, Journal of Mathematical Phychology 48 (2004) 28–50.

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

[37] G. Schwarz, Estimating the dimension of a model, The Annals of Statistics 6 (2) (1978) 461–464.

[38] E. J. Hannan, B. G. Quinn, The determination of the order of an autoregression, Journal of the Royal Statistical Society, series B 41 (2) (1979) 190–195, http://www.jstor.org/stable/2985032.

[39] J. Rissanen, Modeling by shortest data description, Automatica 14 (1978) 465–471.

[40] S. Watanabe, A widely applicable bayesian information criterion, Journal of Machine Learning Research 14 (2013) 867–397.

[41] S. Watanabe, Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory, Journal of Machine Learning Research 11 (2010) 3571–3594.

[42] K. P. Burnham, D. R. Anderson, Multimodel inference. understanding aic and bic in model selection, Sociological methods & research 33 (2) (2004) 261–304, DOI: 10.1177/0049124104268644.

[43] 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.

[44] 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.

[45] 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.