Optimal approximation with exponential sums by a maximum likelihood modification of Prony's method

Ran Zhang[*] Gerlind Plonka[†]

June 18, 2018

Abstract

We consider a modification of Prony's method to solve the problem of best approximation of a given data vector by a vector of equidistant samples of an exponential sum in the 2-norm. We survey the derivation of the corresponding non-linear eigenvector problem that needs to be solved and give its interpretation as a maximum likelihood method. We investigate numerical iteration schemes to solve this problem and give a summary of different approaches including the Levenberg-Marquardt algorithm which is a regularized Gauss-Newton method on the one hand and the Iterative Quadratic Maximum Likelihood (IQML) method as well as the Gradient Condition Reweighted Algorithm (GRA) on the other hand. We propose ourselves two further iteration schemes based on simultaneous minimization (SIMI) being more stable with regard to the choice of the initial vector, and where particularly the second scheme requires less computational cost. For parameter identification, we highly recommend a pre-filtering method to reduce the noise variance. We show that all considered iteration methods converge in numerical experiments.

Key words. Prony method, nonlinear eigenvalue problem, nonconvex optimization, structured matrices, nonlinear structured least squares problem
AMS Subject classifications. 65F15, 62J02, 15A18, 41A30

1 Introduction

The recovery of structured functions from noisy samples is a fundamental problem in signal processing. In particular, the efficient approximation by sums of exponentials plays a significant role in frequency analysis and parameter estimation but also for sparse signal approximation. Let us suppose that the signal is of the form or can be well approximated by an exponential sum

$$f(x) = \sum_{j=1}^{M} c_j e^{T_j x} = \sum_{j=1}^{M} c_j \, z_j^x \tag{1.1}$$

[*]Shanghai Key Laboratory for Contemporary Applied Mathematics, School of Mathematical Sciences, Fudan University, Shanghai, 200433, P.R. China. Email: 12110180047@fudan.edu.cn

[†]Institute for Numerical and Applied Mathematics, Göttingen University, Lotzestr. 16-18, 37083 Göttingen, Germany. Email: {plonka,i.keller}@math.uni-goettingen.de

with unknown coefficients $c_j \in \mathbb{C}$ and parameters $T_j \in \mathbb{C}$ with $\operatorname{Im} T_j \in [-\pi, \pi)$ or $z_j := e^{T_j} \in \mathbb{C}$, respectively. For parameter estimation problems, our goal is to recover all parameters c_j and T_j determining f from given perturbed observations $y_k = f(x_k) + \epsilon_k$, where $x_k = x_0 + kh$, $h > 0$, and ϵ_k are pairwise independent random variables with mean value zero and variance σ^2. For sparse signal approximation problems, we want to approximate a bounded sequence of function values by an exponential sum of the form (1.1) such that the error is minimized in some sense. The sparse approximation problem is very similar to the parameter estimation problem above, but here we do not require a priori knowledge about the error values ϵ_k.

One reason for the strong interest in signal approximation by exponential sums is the wide field of applications. Examples are synchrophasor estimation [41], estimation of mean curve lightning impulses [13], parameter estimation in electrical power systems [29], the localization of particles in inverse scattering [15] and sparse deconvolution methods in ultrasonic nondestructive testing [8].

The great importance of the topic can also be observed from the many reconstruction approaches related to the subject, as e.g. the annihilating filter method in signal processing [12], Padé approximation [45], the reconstruction of signals with finite rate of innovation [12] and linear prediction methods [24]. For a survey on these relations we refer to our paper [34]. Further important applications regard quadrature formulas for higher-dimensional integrals, see [9].

If $f(x)$ indeed possesses the exact structure in (1.1) and if M is given in advance, then the parameters c_j, T_j can be computed by the well-known Prony method from equidistant samples $f(k)$, $k = 0, \ldots, 2M-1$. The basic idea to solve the problem is to understand the exponential sum as the solution of a difference equation of order M with constant coefficients and to separate the evaluation of the parameter sets. First, one determines the characteristic polynomial (Prony polynomial) of the underlying difference equation and extracts the T_j from its zeros z_j. Then the c_j are obtained by solving an (overdetermined) Vandermonde equation system.

However, the classical Prony method is not numerically stable. Therefore, different numerical methods have been (partially independently) developed to recover the parameters in model (1), see e.g. multiple signal classification (MUSIC) by Schmidt [40], estimation of signal parameters via rotational invariance techniques (ESPRIT) by Roy and Kailath [39], the matrix pencil method by Hua and Sakar [16] and the approximate Prony method (APM) by Potts and Tasche [36]. The paper [37] contains a summary of all these algorithms and also studies their close relations.

Contributions of the paper and related work

In this paper, we are interested in the following problem. For a vector of given (noisy) measurements $\mathbf{y} = (y_k)_{k=0}^L$ at equidistant points $(x_k)_{k=0}^L$ with $L \geq 2M$ we want to compute the parameters $c_j \in \mathbb{C}$ and parameters $T_j \in \mathbb{C}$ with $\operatorname{Im} T_j \in [-\pi, \pi)$ such that

$$\Big\| \mathbf{y} - \Big(\sum_{j=1}^{M} c_j e^{T_j x_k} \Big)_{k=0}^{L} \Big\|_2 = \Big\| \mathbf{y} - \Big(\sum_{j=1}^{M} c_j z_j^{x_k} \Big)_{k=0}^{L} \Big\|_2 \tag{1.2}$$

is minimized.

The considered problem can be seen as a nonlinear structured least squares problem (NSLRA), see [22, 25, 26, 44] and references therein. It can be rewritten as a nonconvex minimization problem with rank reduction constraints, see e.g. [44]. One essential idea is to apply the variable projection [14], where the problem of minimizing with respect to the two parameter vectors $\mathbf{c} = (c_1, \ldots, c_M)$ and $\mathbf{z} = (z_1, \ldots, z_M)$ is reduced to a nonlinear minimization problem with regard to $\mathbf{z}$ only, while $\mathbf{c}$ can be simply obtained by a linear least squares method once $\mathbf{z}$ is known. The obtained nonlinear problem can then be tackled using standard optimization routines based on Gauss-Newton iterations or the Levenberg-Marquardt algorithm. We will summarize this approach in Section 4.1 and compare it with other concepts that are based on a non-linear eigenvalue problem formulation.

The minimization problem in (1.2) is also related to the problem of low-rank approximation of Hankel matrices. Taking $f_k = \sum_{j=1}^{M} c_j e^{T_j x_k}$ for $k = 0, \ldots, L$, one may consider instead of $\|\mathbf{y} - \mathbf{f}\|_2$ the matrix norm $\|\mathbf{H_y} - \mathbf{H_f}\|$, where $\mathbf{H_y}$ and $\mathbf{H_f}$ are Hankel matrices generated by $\mathbf{y}$ and $\mathbf{f}$. The special structure of $\mathbf{f}$ then implies that $\mathbf{H_f}$ has only rank M. Thus we arrive at the problem of best low-rank approximation with Hankel structure. Besides numerical approaches to solve this problem, see e.g. [17, 3] and references therein. Several papers considered the connection between low-rank approximation of Hankel matrices and AAK theory [1] being related with the approximation by exponential sums, see e.g. [5, 6, 2, 35]. However, we emphasize that these methods do not exactly solve the problem (1.2) but only a related approximation problem since the spectral norm of $\mathbf{H_y}$ (or the operator norm in AAK theory) cannot be exactly related to the Euclidean norm of $\mathbf{y}$.

For the special case when $c_j \in \mathbb{R}$ and $|z_j| = 1$ for $j = 1, \ldots, M$, iterative approaches have been proposed to solve (1.2) that try to improve the estimate of T_j directly at each iteration step, see e.g. [4, 7]. We also like to mention that approximation in the 1-norm has been considered in [42].

In this paper, we will employ a direct approach to tackle that problem (1.2) based on former ideas on maximum likelihood modifications of Prony's method, see [10, 11, 31]. We show that this approach can be also applied to sparse approximation of $\mathbf{y}$ where we do not need to assume that $\mathbf{y}$ is "close" to a vector that can be approximated by an exponential sum. However, if the components of $\mathbf{y}$ are perturbed observations of $f(x_k)$ in (1.1) then a statistical interpretation as a maximum likelihood method is possible. The modified Prony method leads to a general non-linear eigenvector problem. There exist essentially two different iterative algorithms in the literature to solve this nonlinear problem, the Iterative Quadratic Maximum Likelihood algorithm (IQML) [10, 11] and the Gradient Condition Reweighting Algorithm (GRA) [31]. For both methods the search for $\mathbf{z} = (z_1, \ldots, z_M)^T$ in a first step is converted to the search for a vector $\mathbf{p} \in \mathbb{C}^{M+1}$ of polynomial coefficients such that the knots z_j are the zeros of the corresponding polynomial of degree M. The IQML algorithm has been further studied in [23]. Both, IQML and GRA algorithm have been shown to converge locally under certain conditions, which cannot simply be checked in practice. Our numerical experiments showed however that the IQML algorithm converges very quickly for arbitrary starting vectors while the GRA algorithm only provides reasonable results if the starting vector is already very close to the solution vector. Similar results have been also reported in [21].

We propose two new iteration schemes that are based on simultaneous minimization (SIMI) and work equally well as IQML and GRA and are less sensitive with respect to the initial vector. The second proposed scheme completely avoids inverse covariance matrices in the iteration process. We particularly show the connections to the IQML and the GRA algorithm. Further, we propose a low-pass filtering procedure that can be employed as a preprocessing step for each of the considered iteration methods.

Our numerical experiments show that for stronger noise the pre-filtering step is crucial in order to obtain good parameter estimates. Furthermore, the pre-filtering step strongly reduces the computational effort for all considered iteration methods.

The paper is organized as follows. In Section 2, we shortly recall the Prony method and give some first ideas for its stabilization in case of noisy data, as e.g. given already by Pisarenko [33]. Section 3 is devoted to the maximum likelihood formulation of the Prony method in order to minimize (1.2). In Section 4 we survey the existing different iteration methods VARPRO (based on the Levenberg-Marquardt algorithm), GRA and IQML, and derive two new iteration schemes. Employing a suitable preprocessing step, particularly the second new iteration scheme requires considerably smaller computational costs than the other methods. Finally, we study different numerical examples to compare the efficiency of the iterative methods in Section 5.

2 Prony's method

The classical Prony method for parameter estimation works for exactly sampled data in the case of a known number M of terms, see e.g. [34]. For given data $x_0 \in \mathbb{R}$, $h > 0$, the samples of $f(x)$ in (1.1) are of the form

$$f_k = f(x_0 + hk) = \sum_{j=1}^{M} (c_j e^{T_j x_0}) e^{T_j kh} = \sum_{j=1}^{M} d_j z_j^k, \quad k = 0, \ldots, L,\ L \geq 2M, \tag{2.1}$$

where $d_j := c_j e^{T_j x_0}$ and $z_j := e^{T_j h}$, $j = 1, \ldots, M$, the goal is to recover all parameters c_j, T_j, $j = 1, \ldots, M$, or equivalently d_j, z_j, $j = 1, \ldots, M$, determining $f(x)$. With $\mathbf{f} := (f_0, f_1, \ldots, f_L)^T$, $\mathbf{d} := (d_1, \ldots, d_M)^T$, $\mathbf{z} := (z_1, \ldots, z_M)^T$, and the Vandermonde matrix

$$\mathbf{V}(\mathbf{z}) = \mathbf{V}_{L+1,M}(\mathbf{z}) := \begin{pmatrix} 1 & 1 & \ldots & 1 \\ z_1 & z_2 & \ldots & z_M \\ z_1^2 & z_2^2 & \cdots & z_M^2 \\ \vdots & \vdots & & \vdots \\ z_1^L & z_2^L & \ldots & z_M^L \end{pmatrix}, \tag{2.2}$$

the corresponding matrix equation reads

$$\mathbf{f} = \mathbf{V}(\mathbf{z})\, \mathbf{d}. \tag{2.3}$$

We want to extract $\mathbf{z}$ and $\mathbf{d}$ from $\mathbf{f}$. It is clear that once we found $\mathbf{z}$, the coefficient vector $\mathbf{d}$ can be obtained by solving the overdetermined linear system (2.3). In order to compute $\mathbf{z}$, we introduce the Prony polynomial being determined by its zeros $z_j \in \mathbb{C}$,

$$p(z) := \prod_{j=1}^{M} (z - z_j) = \sum_{k=0}^{M} p_k z^k = \sum_{k=0}^{M-1} p_k z^k + z^M,\ z \in \mathbb{C},$$

with coefficients $p_k \in \mathbb{C}$ and $p_M = 1$. Then, observing the relation

$$\begin{aligned}\sum_{k=0}^{M} p_k f_{k+m} &= \sum_{k=0}^{M} p_k \left(\sum_{j=1}^{M} d_j z_j^{k+m} \right) = \sum_{j=1}^{M} d_j z_j^m \left(\sum_{k=0}^{M} p_k z_j^k \right) \\ &= \sum_{j=1}^{M} d_j z_j^m p(z_j) = 0 \qquad (2.4)\end{aligned}$$

for $m = 0, \ldots, L - M$, we obtain the homogeneous linear system

$$\mathbf{H_f p} = \begin{pmatrix} f_0 & f_1 & \ldots & f_M \\ f_1 & f_2 & \ldots & f_{M+1} \\ \vdots & \vdots & & \vdots \\ f_{L-M} & f_{L-M+1} & \ldots & f_L \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_M \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad (2.5)$$

where the coefficient matrix $\mathbf{H_f} := (f_{j+k})_{j=0,k=0}^{L-M,M} \in \mathbb{C}^{(L-M+1)\times(M+1)}$ has Hankel structure and $\mathbf{p}$ contains the coefficients of the Prony polynomial. Assuming that the knots z_j, $j = 1, \ldots, M$, are pairwise different and that $d_j \neq 0$ for $j = 1, \ldots, M$, the structure of f implies the factorization

$$\begin{aligned}\mathbf{H_f} &= \mathbf{V}_{L-M+1,M}(\mathbf{z}) \operatorname{diag}(d_1, \ldots, d_M) \mathbf{V}_{M+1,M}(\mathbf{z})^T \\ &= \begin{pmatrix} 1 & 1 & \ldots & 1 \\ z_1 & z_2 & \ldots & z_M \\ z_1^2 & z_2^2 & \ldots & z_M^2 \\ \vdots & \vdots & & \vdots \\ z_1^{L-M} & z_2^{L-M} & \ldots & z_M^{L-M} \end{pmatrix} \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_M \end{pmatrix} \begin{pmatrix} 1 & z_1 & \ldots & z_1^M \\ 1 & z_2 & \ldots & z_2^M \\ 1 & z_3 & \ldots & z_3^M \\ \vdots & \vdots & & \vdots \\ 1 & z_M & \ldots & z_M^M \end{pmatrix},\end{aligned}$$

and therefore rank $(\mathbf{H}) = M$, since the two Vandermonde matrices and the diagonal matrix in this factorization have full rank M. Thus, a right-singular vector $\mathbf{p}$ of $\mathbf{H_f}$ to the singular value 0 is uniquely defined using the normalization $p_M = 1$. Knowing $\mathbf{p}$, we can extract the vector $\mathbf{z}$ from the Prony polynomial. Now, the function f can be completely recovered by performing the following three steps, as given in Algorithm 2.1.

Algorithm 2.1 (Prony method)
Input: M, x_0, h, f_k, $k = 0, \ldots, L$, with $L \geq 2M$.

1. *Compute the right singular vector $\mathbf{p} = (p_0, \ldots, p_{M-1}, 1)^T$ of $\mathbf{H_f} := (f_{j+k})_{j=0,k=0}^{L-M,M}$ corresponding to the zero singular value.*
2. *Compute the zeros z_j, $j = 1, \ldots, M$, of the Prony polynomial $p(z) = \sum_{k=0}^{M} p_k z^k$.*
3. *Compute the coefficients d_j, $j = 1, \ldots, M$ by solving the overdetermined system*

$$\sum_{j=1}^{M} d_j z_j^k = f_k, \qquad k = 0, \ldots, L,$$

using the least squares approach, and derive $c_j := e^{-T_j x_0} d_j = z_j^{x_0/h} d_j$.

Output: Parameter vectors $\mathbf{z}$, $\mathbf{c}$.

Note that in Algorithm 2.1 the recovery of $\mathbf{z}$ and $\mathbf{c}$ is even possible using only $2M$ equidistant samples. By rewriting the equation system (2.4) as

$$\sum_{k=0}^{M-1} p_k f_{k+m} = -f_{M+m}, \qquad m = 0, \ldots, M-1,$$

we can compute $\mathbf{p}$ by solving this equation system instead of solving the singular vector problem in step 1 of Algorithm 2.1.

If $L > 2M$, instead of employing the sampled data f_k, $k = 0, \ldots, L$ to compute $\mathbf{p}$ in (2.5) we can also apply an averaging operator first. Let

$$\widetilde{f}_k := \sum_{\ell=0}^{L-2M} w_\ell f_{\ell+k}, \qquad k = 0, \ldots, 2M, \tag{2.6}$$

with some low-pass filter $\mathbf{w} = (w_0, \ldots, w_{L-2M})$ satisfying $w_\ell \geq 0$ for all ℓ and $\sum_{\ell=0}^{L-2M} w_\ell = 1$. With $\widetilde{\mathbf{f}} := (\widetilde{f}_k)_{k=0}^{2M}$ we observe that

$$\mathbf{H}_{\widetilde{\mathbf{f}}} := (\widetilde{f}_{j+k})_{j,k=0}^{M} = \mathbf{L}^T \mathbf{H}_{\mathbf{f}}$$

with

$$\begin{aligned} \mathbf{L}^T &= \mathbf{L}^T_{L-M+1,M+1} \qquad (2.7) \\ &:= \begin{pmatrix} w_0 & w_1 & \ldots & w_{L-2M} & 0 & \ldots & 0 \\ 0 & w_0 & \ldots & & w_{L-2M} & & \vdots \\ \vdots & & \ddots & & & \ddots & 0 \\ 0 & \ldots & 0 & w_0 & & \ldots & w_{L-2M} \end{pmatrix} \in \mathbb{C}^{(M+1)\times(L-M+1)} \end{aligned}$$

is a quadratic matrix of size $(M+1) \times (M+1)$ with rank M, and we indeed have $\mathbf{H}_{\widetilde{\mathbf{f}}}\, \mathbf{p} = \mathbf{0}$ if $\mathbf{H}_{\mathbf{f}}\, \mathbf{p} = \mathbf{0}$. Therefore, the first step in the Algorithm 2.1 for the classical Prony method can be replaced by computing the normalized zero eigenvector $\mathbf{p} = (p_0, \ldots, p_{M-1}, 1)^T$ of $\mathbf{H}_{\widetilde{\mathbf{f}}}$.

Prony method for noisy measurements. In case of noisy input data,

$$\mathbf{y} = (y_k)_{k=0}^{L} = (f_k + \epsilon_k)_{k=0}^{L},$$

we have to consider the Hankel matrix $\mathbf{H}_{\mathbf{y}} := (y_{j+k})_{j=0,k=0}^{L-M,M}$ instead of $\mathbf{H}_{\mathbf{f}}$ that usually does not possess exactly the rank M. In this case, we study the singular vector $\widetilde{\mathbf{p}} = (\widetilde{p}_k)_{k=0}^{M}$ corresponding to the smallest singular value of $\mathbf{H}_{\mathbf{y}}$, i.e.,

$$\mathbf{H}_{\mathbf{y}}^* \mathbf{H}_{\mathbf{y}}\, \widetilde{\mathbf{p}} = \sigma_{M+1}^2\, \widetilde{\mathbf{p}},$$

in order to fix the coefficients of an approximate Prony polynomial $\widetilde{p}(z) = \sum_{k=0}^{M} \widetilde{p}_k z^k$ and to extract the parameters, see e.g. [36]. Employing here the normalization $\widetilde{p}_M = 1$, the least squares problem

$$\widetilde{\mathbf{p}} := \operatorname*{argmin}_{\substack{\mathbf{p} \in \mathbb{C}^{M+1} \\ p_M = 1}} \mathbf{p}^* \mathbf{H}_{\mathbf{y}}^* \mathbf{H}_{\mathbf{y}} \mathbf{p} \tag{2.8}$$

is solved. This approach is also known as the Pisarenko method if we take the normalization $\|\mathbf{p}\|_2 = 1$ instead of $p_M = 1$.

Let us now write the problem in a different way. Introducing the Toeplitz matrix $\mathbf{X}_{\mathbf{p}} \in \mathbb{C}^{(L+1)\times(L-M+1)}$ of the form

$$\mathbf{X}_{\mathbf{p}} := \begin{pmatrix} p_0 & & & \\ p_1 & p_0 & & \\ \vdots & p_1 & \ddots & \\ & \vdots & & p_0 \\ p_M & & & p_1 \\ & p_M & & \vdots \\ & & \ddots & \\ & & & p_M \end{pmatrix}, \tag{2.9}$$

we obtain

$$\mathbf{H}_{\mathbf{y}}\widetilde{\mathbf{p}} = \mathbf{X}_{\widetilde{\mathbf{p}}}^T \mathbf{y}, \tag{2.10}$$

and the problem in (2.8) for noisy data $\mathbf{y}$ is equivalent with

$$\widetilde{\mathbf{p}} := \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ p_M=1}}{\operatorname{argmin}} \mathbf{y}^* \, \overline{\mathbf{X}}_{\mathbf{p}} \mathbf{X}_{\mathbf{p}}^T \, \mathbf{y}. \tag{2.11}$$

We summarize this procedure in Algorithm 2.2.

Algorithm 2.2 (Prony method for noisy measurement vector y)
Input: M, x_0, h, y_k, $k = 0, \ldots, L$, with $L \geq 2M$.

1. *Form $\mathbf{H}_{\mathbf{y}} := (y_{j+k})_{j=0,k=0}^{L-M,M}$ and compute the right singular vector $\mathbf{p} = (p_k)_{k=0}^{M}$ of $\mathbf{H}_{\mathbf{y}}$ corresponding to its smallest singular value.*
2. *Compute the vector $\mathbf{z}$ of zeros z_j, $j = 1, \ldots, M$, of the Prony polynomial $p(z) = \sum_{k=0}^{M} p_k z^k$ by solving an eigenvalue problem for the corresponding companion matrix.*
3. *Compute the coefficients d_j, $j = 1, \ldots, M$ by solving the least squares problem*

$$\min_{\mathbf{d}\in\mathbb{C}^M} \|\mathbf{V}(\mathbf{z})\mathbf{d} - \mathbf{y}\|_2$$

and derive $c_j := e^{-T_j x_0} d_j = z_j^{x_0/h} d_j$.

Output: Parameter vectors $\mathbf{z}$, $\mathbf{c}$.

To improve the numerical stability of Algorithm 2, one can employ the approximate Prony method (AMP), see [36], that usually gives better solutions as ESPRIT. Having a larger number of measurements $L > 2M$, we can apply a preprocessing step that employs a low-pass FIR-filtering as given in (2.6) to obtain

$$\widetilde{y}_k = \sum_{\ell=0}^{L-2M} w_\ell y_{l+k}, \qquad k = 0, \ldots, 2M.$$

Then, with the Hankel matrix $\mathbf{H}_{\widetilde{\mathbf{y}}} := (\widetilde{y}_{j+k})_{j,k=0}^{M}$ of size $(M+1)\times(M+1)$, we compute

$$\mathbf{p}_0 := \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ p_M=1}}{\operatorname{argmin}} \mathbf{p}^* \mathbf{H}_{\widetilde{\mathbf{y}}}^* \mathbf{H}_{\widetilde{\mathbf{y}}} \mathbf{p} = \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ p_M=1}}{\operatorname{argmin}} \mathbf{p}^* \mathbf{H}_{\mathbf{y}}^* \overline{\mathbf{L}} \mathbf{L}^T \mathbf{H}_{\mathbf{y}} \mathbf{p} \tag{2.12}$$

instead of $\widetilde{\mathbf{p}}$ in (2.8) with $\mathbf{L}$ in (2.7). The main advantage of taking the averaged values $\widetilde{y}_k$ is a significantly reduced noise variance. For $y_k = f_k + \epsilon_k$ with i.i.d. random variables $\epsilon_k \in N(0, \sigma^2)$ and with $w_\ell = \frac{1}{L-2M+1}$, $\ell = 0, \ldots, L-2M$, we obtain $\widetilde{y}_k = \widetilde{f}_k + \widetilde{\epsilon}_k$ with

$$\widetilde{\epsilon}_k := \frac{1}{L-2M+1} \left(\sum_{\ell=k}^{k+L-2M} \epsilon_\ell \right) \in N\left(0, \frac{\sigma^2}{L-2M+1}\right), \qquad k = 0, \ldots, 2M.$$

Using this modification, the first step in the Algorithm 2 has to be replaced by:

1. Compute the singular vector $\mathbf{p} = (p_0, \ldots, p_{M-1}, 1)^T$ of $\mathbf{H}_{\widetilde{\mathbf{y}}} = \mathbf{L}^T \mathbf{H}_{\mathbf{y}}$ corresponding to its smallest singular value.

Note however, that the random variables $\tilde{\epsilon}_k$ are linearly dependent, where the dependency is caused by the filter $\mathbf{w}$. We obtain for $\boldsymbol{\epsilon} = (\epsilon_0, \ldots, \epsilon_L)^T$

$$E(\tilde{\boldsymbol{\epsilon}}\, \tilde{\boldsymbol{\epsilon}}^*) = E((\mathbf{L}^T \boldsymbol{\epsilon})(\mathbf{L}^T \boldsymbol{\epsilon})^*) = \mathbf{L}^T E(\boldsymbol{\epsilon}\, \boldsymbol{\epsilon}^*) \mathbf{L} = \mathbf{L}^T \mathbf{L}.$$

If we have $L + 1 = (2M+1)K$ measurement values y_k where $K > 1$ is an integer, then we can also construct local low-pass filters such that the filtered signal possesses i.i.d. Gaussian noise variables. Taking e.g.

$$\widetilde{y}_k := \frac{1}{K} \sum_{r=Kk}^{K(k+1)-1} y_r = \frac{1}{K} \sum_{r=Kk}^{K(k+1)-1} f_r + \frac{1}{K} \sum_{r=Kk}^{K(k+1)-1} \epsilon_r = \widetilde{f}_k + \widetilde{\epsilon}_k, \quad k = 0, \ldots, 2M, \tag{2.13}$$

the new variables $\widetilde{\epsilon}_k$ are linearly independent with mean value zero, and the noise variance is reduced to σ^2/K. Similarly, we can choose for example

$$\widetilde{y}_k = \frac{1}{K} \sum_{r=0}^{K-1} y_{k+(2M+1)r}, \qquad k = 0, \ldots, 2M, \tag{2.14}$$

for strongly decaying sequences $\mathbf{y}$.

Remark 2.3 *Algorithm 2 is equivalent with the Pisarenko method [33] and with the total least squares method considered by Rahman and Yu, [38]. If $\mathbf{H}_{\widetilde{\mathbf{y}}}^* \mathbf{H}_{\widetilde{\mathbf{y}}}$ has more than one singular value in the range $[0, \epsilon]$, than $\mathbf{p}$ is computed as a special linear combination of the corresponding singular vectors such that the norm of the partial vector $(p_0, \ldots, p_{M-1})^T$ of $\mathbf{p}$ is minimal. The APM method in [36] also uses two singular vectors corresponding to the two smallest singular values of $\mathbf{H}_{\widetilde{\mathbf{y}}}^* \mathbf{H}_{\widetilde{\mathbf{y}}}$ in $[0, \epsilon]$ and averages the zero values that are found from the two corresponding "Prony" polynomials.*

A slightly different more expensive iteration method is used for autoregressiv moving average (ARMA) models by Therrien and Velasco [43].

3 Maximum likelihood modification of Prony's method

Let us assume that $y_k = f(x_k) + \epsilon_k$. We want to approximate $\mathbf{y} = (y_k)_{k=0}^L$ by a sequence generated by an exponential sum with M terms. Formula (2.3) implies that we have to solve the nonlinear least squares problem

$$\min_{\mathbf{z},\mathbf{d}\in\mathbb{C}^M} \|\mathbf{y} - \mathbf{V}(\mathbf{z})\mathbf{d}\|_2^2 = \min_{\mathbf{z},\mathbf{d}\in\mathbb{C}^M} \sum_{k=0}^{L} |y_k - \sum_{j=1}^{M} d_j z_j^k|^2, \tag{3.1}$$

where we use the notations as in Section 2. If some a priori knowledge is known about $\mathbf{z}$ and $\mathbf{d}$ as e.g. $|z_j| < 1$ or $d_j \in \mathbb{R}$, we can restrict the range $\mathbb{C}^M$ for the parameter vectors to the suitable subspaces in the minimization process. Following the arguments in [10], we observe that for given $\mathbf{z}$, the minimization problem turns into a linear least squares problem

$$\min_{\mathbf{d}\in\mathbb{C}^M} \|\mathbf{y} - \mathbf{V}(\mathbf{z})\mathbf{d}\|_2^2$$

with the solution

$$\mathbf{d} = \mathbf{V}(\mathbf{z})^{+}\,\mathbf{y} = [\mathbf{V}^*(\mathbf{z})\mathbf{V}(\mathbf{z})]^{-1}\mathbf{V}(\mathbf{z})^*\mathbf{y}, \tag{3.2}$$

since $\mathbf{V}(\mathbf{z})$ can be assumed to have full rank M. Thus (3.1) can be rewritten as

$$\begin{aligned}
\min_{\mathbf{z}\in\mathbb{C}^M} \|\mathbf{y} - \mathbf{V}(\mathbf{z})\mathbf{V}(\mathbf{z})^{+}\,\mathbf{y}\|_2^2 &= \min_{\mathbf{z}\in\mathbb{C}^M} \|(\mathbf{I} - \mathbf{V}(\mathbf{z})\mathbf{V}(\mathbf{z})^{+})\,\mathbf{y}\|_2^2 \\
&= \min_{\mathbf{z}\in\mathbb{C}^M} \mathbf{y}^*(\mathbf{I} - \mathbf{V}(\mathbf{z})\mathbf{V}(\mathbf{z})^{+})^*(\mathbf{I} - \mathbf{V}(\mathbf{z})\mathbf{V}(\mathbf{z})^{+})\mathbf{y} \\
&= \min_{\mathbf{z}\in\mathbb{C}^M} \mathbf{y}^*\mathbf{y} - \mathbf{y}^*\mathbf{V}(\mathbf{z})[\mathbf{V}^*(\mathbf{z})\mathbf{V}(\mathbf{z})]^{-1}\mathbf{V}(\mathbf{z})^*\mathbf{y},
\end{aligned}$$

where we have used the definition of $\mathbf{V}(\mathbf{z})^{+}$ in (3.2) and the fact that $[\mathbf{V}^*(\mathbf{z})\mathbf{V}(\mathbf{z})]^{-1}$ is Hermitian. Thus,

$$\begin{aligned}
\widetilde{\mathbf{z}} &:= \operatorname*{argmin}_{\mathbf{z}\in\mathbb{C}^M} \left(\mathbf{y}^*\mathbf{y} - \mathbf{y}^*\mathbf{V}(\mathbf{z})[\mathbf{V}^*(\mathbf{z})\mathbf{V}(\mathbf{z})]^{-1}\mathbf{V}(\mathbf{z})^*\mathbf{y}\right) \\
&= \operatorname*{argmax}_{\mathbf{z}\in\mathbb{C}^M} \left(\mathbf{y}^*\mathbf{V}(\mathbf{z})[\mathbf{V}^*(\mathbf{z})\mathbf{V}(\mathbf{z})]^{-1}\mathbf{V}(\mathbf{z})^*\mathbf{y}\right).
\end{aligned} \tag{3.3}$$

We want to rephrase this nonlinear least squares problem for $\mathbf{z} = (z_1, \ldots, z_M)^T$ by means of the coefficients of the Prony polynomial $p(z) = \prod_{j=1}^{M}(z - z_j) = \sum_{k=0}^{M} p_k z^k$ with $\|\mathbf{p}\| = 1$.

Theorem 3.1 *For given data* $\mathbf{y} = (y_0, \ldots, y_L)^T$ *the parameter vectors* $\mathbf{z}$ *and* $\mathbf{d}$ *minimizing the nonlinear least squares problem*

$$\min_{\mathbf{z},\mathbf{d}\in\mathbb{C}^M} \|\mathbf{y} - \mathbf{V}(\mathbf{z})\mathbf{d}\|_2^2 = \min_{\mathbf{z},\mathbf{d}\in\mathbb{C}^M} \sum_{k=0}^{L} |y_k - \sum_{j=1}^{M} d_j z_j^k|^2$$

can be obtained by the following procedure.

1. Solve

$$\widetilde{\mathbf{p}} = \operatorname*{argmin}_{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|=1}} \mathbf{y}^*\overline{\mathbf{X}}_{\mathbf{p}}[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{X}_{\mathbf{p}}^T\mathbf{y} = \operatorname*{argmin}_{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|=1}} \mathbf{p}^*\mathbf{H}_{\mathbf{y}}^*[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p}. \tag{3.4}$$

2. Compute the vector of zeros $\mathbf{z} = (z_1, \ldots, z_M)^T$ *of the polynomial* $p(z) = \sum\limits_{k=0}^{M} p_k z^k$*.*

3. Compute

$$\mathbf{d} = \mathbf{V}(z)^+\,\mathbf{y} = [\mathbf{V}^*(\mathbf{z})\mathbf{V}(\mathbf{z})]^{-1}\mathbf{V}(\mathbf{z})^*\mathbf{y}.$$

Proof: We follow the ideas in [10] and give a short proof for the convenience of the reader. For a given vector $\mathbf{z} = (z_1, \ldots, z_M) \in \mathbb{C}^M$ of pairwise distinct knots let $\mathbf{p} = (p_0, \ldots, p_M)^T$ be the normalized coefficient vector of the corresponding Prony polynomial $p(z) = \prod_{j=1}^M (z - z_j) = \sum\limits_{k=0}^{M} p_k z^k$. Now, we observe that the matrices $\mathbf{X}_{\mathbf{p}}$ in (2.9) and $\mathbf{V}(\mathbf{z})$ in (2.2) satisfy

$$\mathbf{X}_{\mathbf{p}}^T\,\mathbf{V}(\mathbf{z}) = \mathbf{0}.$$

Note that $\operatorname{rank}(\mathbf{X}_{\mathbf{p}}) = L + 1 - M$ and $\operatorname{rank}(\mathbf{V}(\mathbf{z})) = M$. Thus, the $L + 1 - M$ columns of $\overline{\mathbf{X}}_{\mathbf{p}}$ span the orthogonal complement of the range of $\mathbf{V}(\mathbf{z})$ spanned by the columns of $\mathbf{V}(\mathbf{z})$ in $\mathbb{C}^{L+1}$, i.e., each vector $\mathbf{v} \in \mathbb{C}^{L+1}$ can be uniquely written as $\mathbf{v} = \mathbf{V}(\mathbf{z})\mathbf{v}_1 + \overline{\mathbf{X}}_{\mathbf{p}}\mathbf{v}_2$ with $\mathbf{v}_1 \in \mathbb{C}^M$, $\mathbf{v}_2 \in \mathbb{C}^{L+1-M}$. Therefore, we find for all $\mathbf{v} \in \mathbb{C}^{L+1}$ on the one hand

$$\mathbf{V}(\mathbf{z})[\mathbf{V}^*(\mathbf{z})\mathbf{V}(\mathbf{z})]^{-1}\mathbf{V}(\mathbf{z})^*\mathbf{v} = \mathbf{V}(\mathbf{z})[\mathbf{V}^*(\mathbf{z})\mathbf{V}(\mathbf{z})]^{-1}\mathbf{V}(\mathbf{z})^*\mathbf{V}(\mathbf{z})\mathbf{v}_1 = \mathbf{V}(\mathbf{z})\mathbf{v}_1$$

and on the other hand

$$\begin{aligned}[\mathbf{I} - \overline{\mathbf{X}}_{\mathbf{p}}[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{X}_{\mathbf{p}}^T]\,\mathbf{v} &= \mathbf{V}(\mathbf{z})\mathbf{v}_1 + \overline{\mathbf{X}}_{\mathbf{p}}\mathbf{v}_2 - \overline{\mathbf{X}}_{\mathbf{p}}[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}\mathbf{v}_2\\ &= \mathbf{V}(\mathbf{z})\mathbf{v}_1.\end{aligned}$$

We conclude that

$$\mathbf{V}(\mathbf{z})[\mathbf{V}^*(\mathbf{z})\mathbf{V}(\mathbf{z})]^{-1}\mathbf{V}(\mathbf{z})^* = \mathbf{I} - \overline{\mathbf{X}}_{\mathbf{p}}[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{X}_{\mathbf{p}}^T, \tag{3.5}$$

i.e., solving the maximization problem in (3.3) is equivalent with solving the minimization problem

$$\widetilde{\mathbf{p}} := \operatorname*{argmin}_{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ p_M=1}} \mathbf{y}^*\overline{\mathbf{X}}_{\mathbf{p}}[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{X}_{\mathbf{p}}^T\mathbf{y},$$

and extracting the vector $\widetilde{\mathbf{z}}$ of zeros of $\sum_{k=0}^M \widetilde{p}_k z^k$. The second representation in (3.4) is due to $\mathbf{X}_{\mathbf{p}}^T\mathbf{y} = \mathbf{H}_{\mathbf{y}}\mathbf{p}$. The remaining computation of $\mathbf{d}$ is the same as in (3.2). ∎

Remarks 3.2

1. Compared to the Prony method applied to noisy data that can be formulated as in (2.11)*, the maximum likelihood modification in* (3.4) *contains the further term* $[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}$ *that makes the minimization problem nonlinear.*

2. Theorem 3.1 *shows that similarly as for the Prony method, the determination of the parameter vectors* $\mathbf{z}$ *and* $\mathbf{d}$ *can be separated. Formula* (3.4) *is the variable* *projection formulation of the Hankel structured low-rank approximation, see e.g. [14, 26]. We emphasize that Theorem* 3.1 *can be applied to an arbitrary vector* $\mathbf{y}$*, i.e., we do not need prior knowledge that* $\mathbf{y}$ *is of the form* $(f(x_k) + \epsilon_k)_{k=0}^L$ *with* $f(x)$ *in* (1.1)*. In other words, no prior knowledge on* ϵ_k *is needed.*

3. A similar idea for fitting exponential models has been already given by Kumaresan et al. [20] even before [10]. An optimization problem similar to that in (3.4) *has been also derived by Hua and Sakar [16] and has been called whitened TLS-LP method.*

4. We note that the normalization of $\mathbf{p}$ *in* (3.4) *does not effect the objective function in* (3.4)*, see e.g. [30]. Indeed we have* $\mathbf{X}_{c\mathbf{p}} = c\mathbf{X}_{\mathbf{p}}$ *and therefore*

$$\overline{\mathbf{X}}_{c\mathbf{p}}[\mathbf{X}_{c\mathbf{p}}^T \overline{\mathbf{X}}_{c\mathbf{p}}]^{-1}\mathbf{X}_{c\mathbf{p}}^T = \overline{\mathbf{X}}_{\mathbf{p}}[\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{X}_{\mathbf{p}}^T$$

for all $c \neq 0$*. The normalization* $p_M = 1$ *is often used for the classical Prony method. The Pisarenko method, see [33] and in the modified Prony method by Osborne and Smith [30, 32] preferred the normalization* $\|\mathbf{p}\|_2 = 1$*. Observe that the vector* $\mathbf{z}$ *of zeros of* $p(z)$ *does not depend on the normalization of* $\mathbf{p}$*, but the Lagrange function used by Osborne and Smith [30, 32] changes.*

5. The minimization problem in (3.1) *can also be written equivalently as the NSLRA problem*

$$\min_{\hat{\mathbf{y}}, \mathbf{V}(\mathbf{z})} \|\mathbf{y} - \hat{\mathbf{y}}\|_2 \quad \textit{subject to } \hat{\mathbf{y}} = \mathbf{V}(\mathbf{z})\mathbf{d} \textit{ and rank } \mathbf{V}(\mathbf{z}) = M$$

with $\mathbf{y}, \hat{\mathbf{y}} \in \mathbb{C}^{L+1}$ *and* $\mathbf{V}(\mathbf{z})$ *in (2.2), or with the parameter vector* $\mathbf{p}$ *instead of* $\mathbf{z}$ *as*

$$\min_{\hat{\mathbf{y}}, \mathbf{X}_{\mathbf{p}}} \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 \quad \textit{subject to } \mathbf{X}_{\mathbf{p}}^T \hat{\mathbf{y}} = \mathbf{0} \textit{ and rank } \mathbf{X}_{\mathbf{p}} = L + 1 - M$$

with $\mathbf{X}_{\mathbf{p}}$ *in* (2.9)*, see e.g. [44].*

6. While the procedure derived in Theorem 3.1 *works for arbitrary data* $\mathbf{y}$*, it can be interpreted also statistically, see [30, 19]. Assume that* $y_k = f(x_k) + \epsilon_k$ *where* $\epsilon_k \in N(0, \sigma^2)$ *are i.i.d. Gaussian variables. Introducing the residual vector* $\mathbf{r} := \mathbf{H}_{\mathbf{y}}\mathbf{p} = \mathbf{X}_{\mathbf{p}}^T \mathbf{y}$*, where* $\mathbf{p}$ *is the (unknown) vector of the exact Prony polynomial coefficients satisfying* $\mathbf{H}_{\mathbf{f}}\mathbf{p} = \mathbf{X}_{\mathbf{p}}^T \mathbf{f} = \mathbf{0}$*, we observe that*

$$\mathbf{r} = (r_k)_{k=0}^{L-M} = \mathbf{X}_{\mathbf{p}}^T \mathbf{y} = \mathbf{X}_{\mathbf{p}}^T \mathbf{f} + \mathbf{X}_{\mathbf{p}}^T \boldsymbol{\epsilon} = \mathbf{X}_{\mathbf{p}}^T \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon} = (\epsilon_k)_{k=0}^L$*. Thus, while the components* r_k *of* $\mathbf{r}$ *have still mean value zero, we obtain for the covariance matrix of* $\mathbf{r}$

$$E(\mathbf{r}\mathbf{r}^*) = E(\mathbf{X}_{\mathbf{p}}^T \mathbf{y}\mathbf{y}^* \overline{\mathbf{X}}_{\mathbf{p}}) = E(\mathbf{X}_{\mathbf{p}}^T \boldsymbol{\epsilon}\, \boldsymbol{\epsilon}^* \overline{\mathbf{X}}_{\mathbf{p}}) = \mathbf{X}_{\mathbf{p}}^T E(\boldsymbol{\epsilon}\, \boldsymbol{\epsilon}^*) \overline{\mathbf{X}}_{\mathbf{p}} = \sigma^2 \mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}} \tag{3.6}$$

i.e., the errors r_k *are not longer independent. Therefore, Osborne and Smith [30, 32] propose to employ a reweighted residual vector*

$$\widetilde{\mathbf{r}} := (\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}})^{-1/2}\, \mathbf{r}$$

such that $E(\widetilde{\mathbf{r}}\widetilde{\mathbf{r}}^*) = \sigma^2 \mathbf{I}$*. Minimization of*

$$\|\widetilde{\mathbf{r}}\|_2^2 = \mathbf{r}^* (\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}})^{-1} \mathbf{r} = \mathbf{y}^* \overline{\mathbf{X}}_{\mathbf{p}} (\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}})^{-1} \mathbf{X}_{\mathbf{p}}^T \mathbf{y}$$

leads to the same Prony modification that we derived in (3.4).

7. Using the method of Lagrangian multipliers, the model in (3.4) *has been derived in [11] from the following reformulated problem: For given noisy data* $\mathbf{y}$*, solve*

$$\min_{\mathbf{s}\in\mathbb{C}^{L+1},\mathbf{p}\in\mathbb{R}^M} \|\mathbf{y}-\mathbf{s}\|_2^2 \quad \textit{subject to} \quad \mathbf{H_s p} = \mathbf{0} \textit{ and } \|\mathbf{p}\|_2^2 = 1.$$

4 Numerical algorithms for the ML-Prony method

In this section we will survey some existing numerical iterations to solve the problem in (3.4) and present two further iteration algorithms based on viewing the problem as a generalized eigenvalue problem.

We start with deriving a necessary condition for the vector $\widetilde{\mathbf{p}}$ in (3.4). This condition has been also found in different forms in earlier papers, see e.g. [30, 32, 31] but without giving direct matrix representations of the Jacobian and the gradient. Let

$$G(\mathbf{p}) := \mathbf{y}^* \overline{\mathbf{X}}_\mathbf{p} [\mathbf{X}_\mathbf{p}^T \overline{\mathbf{X}}_\mathbf{p}]^{-1} \mathbf{X}_\mathbf{p}^T \mathbf{y} = \|\mathbf{r}(\mathbf{p})\|_2^2 \tag{4.1}$$

with $\mathbf{r}(\mathbf{p}) := \overline{\mathbf{X}}_\mathbf{p} [\mathbf{X}_\mathbf{p}^T \overline{\mathbf{X}}_\mathbf{p}]^{-1} \mathbf{X}_\mathbf{p}^T \mathbf{y} = (\mathbf{X}_\mathbf{p}^+)^T \mathbf{X}_\mathbf{p}^T \mathbf{y} = \overline{\mathbf{X}}_\mathbf{p} \overline{\mathbf{X}}_\mathbf{p}^+ \mathbf{y} \in \mathbb{C}^{L+1}$ such that $\widetilde{\mathbf{p}}$ in (3.4) minimizes $G(\mathbf{p})$. Here $\mathbf{X}_\mathbf{p}^+$ denotes the Moore Penrose generalized inverse of $\mathbf{X}_\mathbf{p}$. Obviously, $\mathbf{r}(\mathbf{p})$ is closely related to $\widetilde{\mathbf{r}}$ in Remark 3.2, but we avoid to take square roots of matrices. For this special problem we can derive now the Jacobian of $\mathbf{r}(\mathbf{p})$ as follows.

Theorem 4.1 *Let* $\mathbf{p} = \mathbf{a} + \mathrm{i}\mathbf{b}$ *with* $\mathbf{a}$, $\mathbf{b} \in \mathbb{R}^{M+1}$*, and let for* $\breve{\mathbf{p}} = (\mathbf{a}^T, \mathbf{b}^T)^T$

$$\mathbf{J}(\breve{\mathbf{p}}) = \mathbf{J}(\mathbf{a},\mathbf{b}) := \left(\Big(\frac{\partial r_j(\mathbf{p})}{\partial a_k}\Big)_{j=0,k=0}^{L,M}, \Big(\frac{\partial r_j(\mathbf{p})}{\partial b_k}\Big)_{j=0,k=0}^{L,M} \right) \in \mathbb{C}^{(L+1)\times 2(M+1)}$$

be the Jacobian of the vector $\mathbf{r}(\mathbf{p}) = (r_j(\mathbf{p}))_{j=0}^L = (\mathbf{X}_\mathbf{p}^+)^T \mathbf{X}_\mathbf{p}^T \mathbf{y}$*. Then we have*

$$\mathbf{J}(\mathbf{a},\mathbf{b}) = (\mathbf{I} - \overline{\mathbf{X}}_\mathbf{p} [\mathbf{X}_\mathbf{p}^T \overline{\mathbf{X}}_\mathbf{p}]^{-1} \mathbf{X}_\mathbf{p}^T) \mathbf{X}_{\mathbf{v}(\mathbf{p})} \, (\mathbf{I}, \, -\mathrm{i}\mathbf{I}) + (\mathbf{X}_\mathbf{p}^+)^T \mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p})} \, (\mathbf{I}, \, \mathrm{i}\mathbf{I}) \, ,$$

where $\mathbf{v}(\mathbf{p}) := \overline{\mathbf{X}}_\mathbf{p}^+ \mathbf{y}$ *and* $\mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p})} = \mathbf{H}_\mathbf{y} - \mathbf{H}_{\mathbf{r}(\mathbf{p})}$ *with* $\mathbf{H}_{\mathbf{r}(\mathbf{p})}$ *being the Hankel matrix of size* $(L+1-M) \times (M+1)$ *generated by* $\mathbf{r}(\mathbf{p})$*. The gradient of* $G(\breve{\mathbf{p}}) := G(\mathbf{p})$ *in* (4.1) *reads*

$$\nabla G(\breve{\mathbf{p}}) = 2\,\mathbf{J}(\mathbf{a},\mathbf{b})^* \mathbf{r}(\mathbf{p}) = 2 \begin{pmatrix} \mathbf{I} \\ -\mathrm{i}\mathbf{I} \end{pmatrix} \mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p})}^* (\mathbf{X}_\mathbf{p}^T \overline{\mathbf{X}}_\mathbf{p})^{-1} \mathbf{H}_\mathbf{y} \mathbf{p}. \tag{4.2}$$

Further, we obtain

$$\begin{aligned} \mathbf{J}(\breve{\mathbf{p}})^T \overline{\mathbf{J}(\breve{\mathbf{p}})} \;&=\; \begin{pmatrix} \mathbf{I} \\ \mathrm{i}\mathbf{I} \end{pmatrix} \mathbf{X}_{\mathbf{v}(\mathbf{p})}^T (\mathbf{I} - \overline{\mathbf{X}}_\mathbf{p} (\mathbf{X}_\mathbf{p}^T \overline{\mathbf{X}}_\mathbf{p})^{-1} \mathbf{X}_\mathbf{p}^T) \overline{\mathbf{X}}_{\mathbf{v}(\mathbf{p})} \, (\mathbf{I}, \, -\mathrm{i}\mathbf{I}) \\ &\quad + \begin{pmatrix} \mathbf{I} \\ -\mathrm{i}\mathbf{I} \end{pmatrix} \mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p})}^T (\mathbf{X}_\mathbf{p}^T \overline{\mathbf{X}}_\mathbf{p})^{-1} \overline{\mathbf{H}}_{\mathbf{y}-\mathbf{r}(\mathbf{p})} \, (\mathbf{I}, \, \mathrm{i}\mathbf{I}) \, . \end{aligned}$$

Proof: First, we observe that $\frac{\partial \mathbf{X}_{\mathbf{p}}}{\partial a_k} = \mathbf{X}_k$ for $k = 0, \ldots, M$, where the matrix $\mathbf{X}_k \in \mathbb{C}^{(L+1)\times(L-M+1)}$ is of the form

$$\mathbf{X}_k := \begin{pmatrix} 0 & \dots & 0 \\ \vdots & & \\ 1 & & \\ & \ddots & \\ & & 1 \\ & & \vdots \\ 0 & \cdots & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{k\times(L-M+1)} \\ \mathbf{I}_{(L-M+1)\times(L-M+1)} \\ \mathbf{0}_{(M-k)\times(L-M+1)} \end{pmatrix},$$

and where $\mathbf{0}$ and $\mathbf{I}$ denote zero matrices and the identity matrix of given size. We obtain

$$\begin{aligned}
\frac{\partial}{\partial a_k}\mathbf{r}(\mathbf{p}) &= \frac{\partial}{\partial a_k}(\overline{\mathbf{X}}_{\mathbf{p}}[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{X}_{\mathbf{p}}^T\mathbf{y}) \\
&= \mathbf{X}_k[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{X}_{\mathbf{p}}^T\mathbf{y} + \overline{\mathbf{X}}_{\mathbf{p}}[-\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}(\mathbf{X}_k^T\overline{\mathbf{X}}_{\mathbf{p}} + \mathbf{X}_{\mathbf{p}}^T\mathbf{X}_k)[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{X}_{\mathbf{p}}^T\mathbf{y} \\
&\quad +\overline{\mathbf{X}}_{\mathbf{p}}(\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}})^{-1}\mathbf{X}_k^T\mathbf{y} \\
&= \mathbf{X}_k\mathbf{v}(\mathbf{p}) - (\mathbf{X}_{\mathbf{p}}^+)^T\mathbf{X}_k^T\mathbf{r}(\mathbf{p}) - (\mathbf{X}_{\mathbf{p}}^+)^T\mathbf{X}_{\mathbf{p}}^T\mathbf{X}_k\mathbf{v}(\mathbf{p}) + (\mathbf{X}_{\mathbf{p}}^+)^T\mathbf{X}_k^T\mathbf{y} \\
&= \mathbf{X}_{\mathbf{v}(\mathbf{p})}\mathbf{e}_k - (\mathbf{X}_{\mathbf{p}}^+)^T\mathbf{H}_{\mathbf{r}(\mathbf{p})}\mathbf{e}_k - (\mathbf{X}_{\mathbf{p}}^+)^T\mathbf{X}_{\mathbf{p}}^T\mathbf{X}_{\mathbf{v}(\mathbf{p})}\mathbf{e}_k + (\mathbf{X}_{\mathbf{p}}^+)^T\mathbf{H}_{\mathbf{y}}\mathbf{e}_k \\
&= (\mathbf{I} - (\mathbf{X}_{\mathbf{p}}^+)^T\mathbf{X}_{\mathbf{p}}^T)\mathbf{X}_{\mathbf{v}(\mathbf{p})}\mathbf{e}_k + (\mathbf{X}_{\mathbf{p}}^+)^T\mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p})}\mathbf{e}_k,
\end{aligned}$$

where $\mathbf{e}_k$ denotes the k-th unit vector. Here we have used that $\mathbf{X}_k\mathbf{v}(\mathbf{p}) = \mathbf{X}_{\mathbf{v}(\mathbf{p})}\mathbf{e}_k$ for $\mathbf{v}(\mathbf{p}) \in \mathbb{C}^{L+1-M}$ and $\mathbf{X}_k^T\mathbf{r}(\mathbf{p}) = \mathbf{H}_{\mathbf{r}(\mathbf{p})}\mathbf{e}_k$ as well as $\mathbf{X}_k^T\mathbf{y} = \mathbf{H}_{\mathbf{y}}\mathbf{e}_k$ for the two vectors $\mathbf{r}(\mathbf{p})$ and $\mathbf{y}$ of length $L+1$. The partial derivatives with respect to b_k are obtained similarly using $\frac{\partial \mathbf{X}_{\mathbf{p}}}{\partial b_k} = \mathrm{i}\mathbf{X}_k$. Taking these derivatives for all k we arrive at $\mathbf{J}(\breve{\mathbf{p}})$. For the gradient it now follows by $\overline{\mathbf{X}}_{\mathbf{p}}^+\overline{\mathbf{X}}_{\mathbf{p}}\overline{\mathbf{X}}_{\mathbf{p}}^+ = \overline{\mathbf{X}}_{\mathbf{p}}^+$ that

$$\begin{aligned}
\nabla G(\breve{\mathbf{p}}) &= 2\,\mathbf{J}(\breve{\mathbf{p}})^*\mathbf{r}(\mathbf{p}) \\
&= 2\left(\begin{pmatrix} \mathbf{I} \\ \mathrm{i}\mathbf{I} \end{pmatrix} \mathbf{X}_{\mathbf{v}(\mathbf{p})}^*(\mathbf{I} - \overline{\mathbf{X}}_{\mathbf{p}}[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{X}_{\mathbf{p}}^T) + \begin{pmatrix} \mathbf{I} \\ -\mathrm{i}\mathbf{I} \end{pmatrix} \mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p})}^*\overline{\mathbf{X}}_{\mathbf{p}}^+\right)\overline{\mathbf{X}}_{\mathbf{p}}\overline{\mathbf{X}}_{\mathbf{p}}^+\mathbf{y} \\
&= 2\begin{pmatrix} \mathbf{I} \\ -\mathrm{i}\mathbf{I} \end{pmatrix} \mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p})}^*\overline{\mathbf{X}}_{\mathbf{p}}^+\mathbf{y} = 2\begin{pmatrix} \mathbf{I} \\ -\mathrm{i}\mathbf{I} \end{pmatrix} \mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p})}^*(\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}})^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p}.
\end{aligned}$$

The representation for $\mathbf{J}(\breve{\mathbf{p}})^*\mathbf{J}(\breve{\mathbf{p}})$ follows similarly. ■

Corollary 4.2 *A normalized vector* $\mathbf{p} \in \mathbf{C}^{M+1}$ *that minimizes* $G(\mathbf{p})$ *necessarily satisfies the eigenvector equation*

$$\mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p})}^*(\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}})^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p} = \Big(\mathbf{H}_{\mathbf{y}}^*(\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}})^{-1}\mathbf{H}_{\mathbf{y}} - \mathbf{H}_{\mathbf{r}(\mathbf{p})}^*(\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}})^{-1}\mathbf{H}_{\mathbf{r}(\mathbf{p})}\Big)\mathbf{p} = \mathbf{0}. \quad (4.3)$$

Proof: The assertion directly follows from Theorem 4.1 and $\mathbf{H}_{\mathbf{r}(\mathbf{p})}\mathbf{p} = \mathbf{X}_{\mathbf{p}}^T\mathbf{r}(\mathbf{p})$ since we observe that

$$\mathbf{H}_{\mathbf{r}(\mathbf{p})}^*(\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}})^{-1}\mathbf{H}_{\mathbf{r}(\mathbf{p})}\mathbf{p} = \mathbf{H}_{\mathbf{r}(\mathbf{p})}^*(\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}})^{-1}\mathbf{X}_{\mathbf{p}}^T\mathbf{r}(\mathbf{p}) = \mathbf{H}_{\mathbf{r}(\mathbf{p})}^*(\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}})^{-1}\mathbf{X}_{\mathbf{p}}^T\mathbf{y}.$$

■

Let us now shortly review the algorithms that can be found already in the literature. All considered algorithms are iterative and aim at successive improvement of the coefficient vector $\mathbf{p}$. As a suitable initial vector we will use

$$\mathbf{p}_0 := \operatorname*{argmin}_{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}} \mathbf{p}^*\mathbf{H}_{\mathbf{y}}^*\mathbf{H}_{\mathbf{y}}\mathbf{p}. \tag{4.4}$$

Obviously, $\mathbf{p}_0$ is the eigenvector corresponding to the smallest eigenvalue of the positive semidefinite Hermitian matrix $\mathbf{H}_{\mathbf{y}}^*\mathbf{H}_{\mathbf{y}}$ obtained by the Pisarenko method (2.8). Since $\mathbf{y}$ is noisy, the obtained smallest singular value is usually nonzero.

Further, we will compare the algorithms taking either the full data vector $\mathbf{y} \in \mathbb{C}^{L+1}$ or a pre-smoothed data vector $\widetilde{\mathbf{y}} \in \mathbb{C}^{2M+1}$ obtained by (2.13) or (2.14), respectively.

4.1 Gauß-Newton and Levenberg-Marquardt iteration

We approximate $\mathbf{r}(\mathbf{p}+\boldsymbol{\delta})$ by its first order Taylor expansion. Here again we map $\mathbf{p} = \mathbf{a}+\mathrm{i}\mathbf{b}$ to $\breve{\mathbf{p}} := (\mathbf{a}^T, \mathbf{b}^T)^T \in \mathbb{R}^{2(M+1)}$ and $\boldsymbol{\delta} = \boldsymbol{\delta}_1+\mathrm{i}\boldsymbol{\delta}_2$ to $\breve{\boldsymbol{\delta}} := (\boldsymbol{\delta}_1^T, \boldsymbol{\delta}_2^T)^T \in \mathbb{R}^{2(M+1)}$. Then

$$\mathbf{r}(\breve{\mathbf{p}}+\breve{\boldsymbol{\delta}}) = \mathbf{r}(\breve{\mathbf{p}}) + \mathbf{J}(\breve{\mathbf{p}})\breve{\boldsymbol{\delta}},$$

where $\mathbf{J}(\breve{\mathbf{p}}) = \mathbf{J}(\mathbf{a},\mathbf{b})$ and $\mathbf{r}(\breve{\mathbf{p}}) = \mathbf{r}(\mathbf{p})$, and we compute

$$\begin{aligned} G(\breve{\mathbf{p}}+\breve{\boldsymbol{\delta}}) &\approx (\mathbf{r}(\breve{\mathbf{p}}) + \mathbf{J}(\breve{\mathbf{p}})\breve{\boldsymbol{\delta}})^*(\mathbf{r}(\breve{\mathbf{p}}) + \mathbf{J}(\breve{\mathbf{p}})\breve{\boldsymbol{\delta}}) \\ &= \mathbf{r}(\breve{\mathbf{p}})^*\mathbf{r}(\breve{\mathbf{p}}) + \mathbf{r}(\breve{\mathbf{p}})^*\mathbf{J}(\breve{\mathbf{p}})\breve{\boldsymbol{\delta}} + \breve{\boldsymbol{\delta}}^*\mathbf{J}(\breve{\mathbf{p}})^*\mathbf{r}(\breve{\mathbf{p}}) + \breve{\boldsymbol{\delta}}^*\mathbf{J}(\breve{\mathbf{p}})^*\mathbf{J}(\breve{\mathbf{p}})\breve{\boldsymbol{\delta}}. \end{aligned}$$

Minimization of this expression with regard to the vector $\breve{\boldsymbol{\delta}}$ gives

$$2\mathrm{Re}\,(\mathbf{J}(\breve{\mathbf{p}})^*\mathbf{r}(\breve{\mathbf{p}})) + 2\mathbf{J}(\breve{\mathbf{p}})^*\mathbf{J}(\breve{\mathbf{p}})\breve{\boldsymbol{\delta}} = \mathbf{0}.$$

The corresponding Gauss-Newton iteration $\mathbf{J}(\breve{\mathbf{p}})^*\mathbf{J}(\breve{\mathbf{p}})\breve{\boldsymbol{\delta}} = -\mathrm{Re}\,(\mathbf{J}(\breve{\mathbf{p}})^*\mathbf{r}(\breve{\mathbf{p}}))$ leads in our case at the jth step to

$$\mathbf{J}(\breve{\mathbf{p}}_j)^*\mathbf{J}(\breve{\mathbf{p}}_j)\breve{\boldsymbol{\delta}}_j = -\mathrm{Re}\,(\mathbf{J}(\breve{\mathbf{p}}_j)^*\mathbf{r}(\breve{\mathbf{p}}_j))$$

to get the improved vector $\breve{\mathbf{p}}_{j+1} = \breve{\mathbf{p}}_j + \breve{\boldsymbol{\delta}}_j$, where the needed expressions can be taken from Theorem 4.1. However, while the coefficient matrix $\mathbf{J}(\breve{\mathbf{p}}_j)^*\mathbf{J}(\breve{\mathbf{p}}_j)$ is obviously positive semidefinite, it is not always positive definite. Particularly, if $\mathbf{p}_j$ is a real vector, i.e. $\breve{\mathbf{p}}_j = (\mathbf{p}_j^T, \mathbf{0}^T)^T$, then

$$\begin{aligned} &\mathbf{J}(\breve{\mathbf{p}}_j)^*\mathbf{J}(\breve{\mathbf{p}}_j)\breve{\mathbf{p}}_j = \mathbf{J}(\mathbf{p}_j,\mathbf{0})^T\mathbf{J}(\mathbf{p}_j,\mathbf{0})\breve{\mathbf{p}}_j \\ = &\left(\begin{pmatrix} \mathbf{I} \\ -\mathrm{i}\,\mathbf{I} \end{pmatrix} \mathbf{X}_{\mathbf{v}(\mathbf{p}_j)}^*(\mathbf{I} - \mathbf{X}_{\mathbf{p}_j}(\mathbf{X}_{\mathbf{p}_j}^T\mathbf{X}_{\mathbf{p}_j})^{-1}\mathbf{X}_{\mathbf{p}_j}^T)\mathbf{X}_{\mathbf{v}(\mathbf{p}_j)}\right. \\ &\left. + \begin{pmatrix} \mathbf{I} \\ \mathrm{i}\,\mathbf{I} \end{pmatrix} \mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p}_j)}^*(\mathbf{X}_{\mathbf{p}_j}^T\mathbf{X}_{\mathbf{p}_j})^{-1}\overline{\mathbf{H}}_{\mathbf{y}-\mathbf{r}(\mathbf{p}_j)}\right)\mathbf{p}_j = \mathbf{0}. \end{aligned} \tag{4.5}$$

Indeed we observe for $\mathbf{p}_j \in \mathbb{R}^{M+1}$ and $\mathbf{v}(\mathbf{p}_j) = \overline{\mathbf{X}}_{\mathbf{p}_j}^+\mathbf{y} = \mathbf{X}_{\mathbf{p}_j}^+\mathbf{y}$ that

$$\begin{aligned} &\mathbf{X}_{\mathbf{v}(\mathbf{p}_j)}^*(\mathbf{I} - \mathbf{X}_{\mathbf{p}_j}(\mathbf{X}_{\mathbf{p}_j}^T\mathbf{X}_{\mathbf{p}_j})^{-1}\mathbf{X}_{\mathbf{p}_j}^T)\mathbf{X}_{\mathbf{v}(\mathbf{p}_j)}\mathbf{p}_j \\ = &\mathbf{X}_{\mathbf{v}(\mathbf{p}_j)}^*(\mathbf{I} - \mathbf{X}_{\mathbf{p}_j}(\mathbf{X}_{\mathbf{p}_j}^T\mathbf{X}_{\mathbf{p}_j})^{-1}\mathbf{X}_{\mathbf{p}_j}^T)\mathbf{X}_{\mathbf{p}_j}\mathbf{v}(\mathbf{p}_j) \\ = &\mathbf{X}_{\mathbf{v}(\mathbf{p}_j)}^*(\mathbf{I} - \mathbf{X}_{\mathbf{p}_j}(\mathbf{X}_{\mathbf{p}_j}^T\mathbf{X}_{\mathbf{p}_j})^{-1}\mathbf{X}_{\mathbf{p}_j}^T)\mathbf{X}_{\mathbf{p}_j}[\mathbf{X}_{\mathbf{p}_j}^T\mathbf{X}_{\mathbf{p}_j}]^{-1}\mathbf{X}_{\mathbf{p}_j}^T\mathbf{y} = \mathbf{0} \end{aligned}$$

and similarly $\mathbf{H}^*_{\mathbf{y}-\mathbf{r}(\mathbf{p}_j)}(\mathbf{X}^T_{\mathbf{p}_j}\mathbf{X}_{\mathbf{p}_j})^{-1}\overline{\mathbf{H}}_{\mathbf{y}-\mathbf{r}(\mathbf{p}_j)}\mathbf{p}_j = \mathbf{0}$.

Levenberg-Marquardt iteration. The *Levenberg-Marquardt algorithm* introduces a regularization changing the coefficient matrix at each iteration step to $\mathbf{J}(\breve{\mathbf{p}}_j)^*\mathbf{J}(\breve{\mathbf{p}}_j) + \lambda_j\mathbf{I}$ which is always positive definite for $\lambda_j > 0$. The iteration then reads

$$(\mathbf{J}(\breve{\mathbf{p}}_j)^*\mathbf{J}(\breve{\mathbf{p}}_j) + \lambda_j\mathbf{I})\,\breve{\boldsymbol{\delta}}_j = -\mathrm{Re}\,(\mathbf{J}(\breve{\mathbf{p}}_j)^*\mathbf{r}(\breve{\mathbf{p}}_j))\,.$$

In this algorithm, we need to fix the parameter λ_j which is usually taken very small. If we arrive at a (local) minimum of $G(\mathbf{p})$, then the right-hand side in the Levenberg-Marquardt iteration vanishes, and we obtain $\breve{\boldsymbol{\delta}}_j = \mathbf{0}$.

The optimization algorithm is very fast and tends to converge to the next local minimum. Therefore, the solution strongly depends on the initial vector $\breve{\mathbf{p}}_0$ that we take as given in (4.4). For existing software packages to implement this method we refer to [27].

4.2 Algorithms for the nonlinear eigenvector problem

We consider two other algorithms to solve (3.4) that are based on considering the underlying nonlinear eigenvector problem.

Gradient Condition Reweighting Algorithm (GRA). Employing the necessary condition (4.3) in Corollary 4.2, the GRA scheme of Osborne and Smith can be written in the form

$$\begin{aligned} \Big(\mathbf{H}^*_{\mathbf{y}-\mathbf{r}(\mathbf{p}_j)}(\mathbf{X}^T_{\mathbf{p}_j}\overline{\mathbf{X}}_{\mathbf{p}_j})^{-1}\mathbf{H}_{\mathbf{y}} - \mu_j\mathbf{I}\Big)\,\mathbf{p}_{j+1} &= \mathbf{0}, \\ \mathbf{p}^*_{j+1}\mathbf{p}_{j+1} &= 1. \end{aligned} \tag{4.6}$$

Comparison with Theorem 4.1 shows that the generalized eigenvalue equation above is equivalent with $\mathbf{B}_{\mathbf{p}_j}\mathbf{p}_{j+1} = \mu_j\mathbf{p}_{j+1}$, where $\mathbf{B}_{\mathbf{p}} = \mathbf{H}^*_{\mathbf{y}-\mathbf{r}(\mathbf{p})}(\mathbf{X}^T_{\mathbf{p}}\overline{\mathbf{X}}_{\mathbf{p}})^{-1}\mathbf{H}_{\mathbf{y}}$ such that $\mathbf{B}_{\mathbf{p}}\mathbf{p} = \nabla G(\mathbf{p})$. In the j-th step of the iteration, inverse iteration is applied to compute the eigenvector $\mathbf{p}_{j+1}$ of $\mathbf{B}_{\mathbf{p}_j}$ corresponding to the smallest eigenvalue by modulus μ_j. In this iteration, the necessary condition in Corollary 4.2 has been simplified by replacing $\mathbf{p}_{j+1}$ in the matrix $\mathbf{B}_{\mathbf{p}_j}$ by the vector $\mathbf{p}_j$. It is clear from Corollary 4.2 that μ_j tends to zero for $j \to \infty$. The algorithm stops when μ_{j+1} is small enough compared to $\|\mathbf{B}_{\mathbf{p}_j}\|$.

Algorithm 4.3 (GRA)
Input: M, y_k, $k = 0, \ldots, L$, with $L \geq 2M$.

1. *Initialization*
   - *Optional: Compute $\widetilde{\mathbf{y}}$ in* (2.13) *or* (2.14) *and replace in all further steps* $\mathbf{y}$ *by* $\widetilde{\mathbf{y}}$.
   - *Compute* $\mathbf{p}_0$ *in* (4.4).
2. *Iteration: For $j = 0\ldots$ till convergence*
   - *Compute* $\mathbf{p}_{j+1}$ *according to* (4.6)*, i.e., compute the eigenvector* $\mathbf{p}_{j+1}$ *of* $\mathbf{B}_{\mathbf{p}_j}$ *corresponding to its smallest eigenvalue by modulus.*
3. *Denote by* $\mathbf{p}$ *the vector obtained by that iteration.*
4. *Compute the vector* $\mathbf{z}$ *of zeros z_j, $j = 1, \ldots, M$, of the Prony polynomial $p(z) = \sum_{k=0}^{M} p_k z^k$ by solving an eigenvalue problem for the corresponding companion matrix.*

5. *Compute the coefficients d_j, $j = 1, \ldots, M$ by solving the least squares problem*

$$\mathbf{V}(z)\mathbf{d} = \mathbf{y}$$

and derive $c_j := e^{-T_j x_0} d_j = z_j^{x_0/h} d_j$.

Output: Parameter vectors $\mathbf{z}$, $\mathbf{c}$.

Iterative Quadratic Maximum Likelihood (IQML). Further, we present the iterative quadratic maximum likelihood (IQML) algorithm in [10, 11] and the algorithm ORA (Objective function Reweighting Algorithm) in [18]. In both methods the iteration

$$\mathbf{p}_{j+1} = \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1} \\ \|\mathbf{p}\|=1}}{\operatorname{argmin}} \mathbf{p}^* \mathbf{H}_{\mathbf{y}}^* [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \mathbf{H}_{\mathbf{y}} \mathbf{p}, \tag{4.7}$$

is proposed. Compared to the representation of the gradient in Theorem 4.1 and to the GRA iteration in (4.6) the IQML iteration just does not take the second term $\mathbf{H}_{\mathbf{r}(\mathbf{p})}^* (\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}})^{-1} \mathbf{H}_{\mathbf{y}} \mathbf{p}$ into account.

This iteration works well in practice, see Algorithm 4.4. However, it is not obvious whether the solution vector $\mathbf{p}_j$ is indeed a fixed point of the IQML iteration. We can apply this scheme also to the filtered data $\widetilde{\mathbf{y}}$.

Algorithm 4.4 (IQML)
Input: M, y_k, $k = 0, \ldots, L$, with $L \geq 2M$.

1. *Initialization*
   - *Optional: Compute $\widetilde{\mathbf{y}}$ in* (2.13) *or* (2.14) *and replace in all further steps* $\mathbf{y}$ *by* $\widetilde{\mathbf{y}}$.
   - *Compute* $\mathbf{p}_0$ *in* (4.4).
2. *Iteration: For $j = 0 \ldots$ till convergence*
   - *Compute $\mathbf{p}_{j+1}$ according to* (4.7)*, i.e., compute the right-singular vector $\mathbf{p}_{j+1}$ of $[\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1/2} \mathbf{H}_{\mathbf{y}}$ corresponding to its smallest singular value.*
3. *Denote by $\mathbf{p}$ the vector obtained by that iteration and compute the vector $\mathbf{z}$ of zeros z_j, $j = 1, \ldots, M$, of the Prony polynomial $p(z) = \sum_{k=0}^{M} p_k z^k$ by solving an eigenvalue problem for the corresponding companion matrix.*
4. *Compute the coefficients d_j, $j = 1, \ldots, M$ by solving the least squares problem*

$$\mathbf{V}(z)\mathbf{d} = \mathbf{y}$$

and derive $c_j := e^{-T_j x_0} d_j = z_j^{x_0/h} d_j$.

Output: Parameter vectors $\mathbf{z}$, $\mathbf{c}$.

4.3 New iteration schemes based on simultaneous minimization

Based on the ideas of Osborne, we want to consider an extended iteration scheme in order to relax the problem of getting stuck at the next local minimum. For two normalized vectors $\mathbf{p}$ and $\mathbf{q}$ in $\mathbb{C}^{M+1}$ we introduce the matrix

$$\mathbf{A}(\mathbf{p}, \mathbf{q}) := \overline{\mathbf{X}}_{\mathbf{p}} [\mathbf{X}_{\mathbf{q}}^T \overline{\mathbf{X}}_{\mathbf{q}}]^{-1} \mathbf{X}_{\mathbf{p}}^T.$$

Then, (3.4) can be written as $\widetilde{\mathbf{p}} = \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1} \\ \|\mathbf{p}\|_2=1}}{\operatorname{argmin}} \mathbf{y}^*\mathbf{A}(\mathbf{p},\mathbf{p})\mathbf{y}$. Our goal is now to improve $\mathbf{p}$ during an iteration by simultaneously minimizing $\mathbf{y}^*\mathbf{A}(\mathbf{p}_j,\mathbf{p})\mathbf{y}$ and $\mathbf{y}^*\mathbf{A}(\mathbf{p},\mathbf{p}_j)\mathbf{y}$ with respect to $\mathbf{p}$ to obtain $\mathbf{p}_{j+1}$. Therefore, we consider the new iteration scheme

$$\mathbf{p}_{j+1} := \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1} \\ \|\mathbf{p}\|_2=1}}{\operatorname{argmin}} \left(\mathbf{y}^*\mathbf{A}(\mathbf{p}_j,\mathbf{p})\mathbf{y} + \mathbf{y}^*\mathbf{A}(\mathbf{p},\mathbf{p}_j)\mathbf{y}\right), \tag{4.8}$$

and denote by

$$\begin{aligned} F(\mathbf{p}_{j+1},\mathbf{p}_j) &:= \mathbf{y}^*\mathbf{A}(\mathbf{p}_j,\mathbf{p}_{j+1})\mathbf{y} + \mathbf{y}^*\mathbf{A}(\mathbf{p}_{j+1},\mathbf{p}_j)\mathbf{y} \qquad (4.9)\\ = \ & (\mathbf{p}_j)^*\mathbf{H}_{\mathbf{y}}^*(\mathbf{X}_{\mathbf{p}_{j+1}}^T\overline{\mathbf{X}}_{\mathbf{p}_{j+1}})^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p}_j + (\mathbf{p}_{j+1})^*\mathbf{H}_{\mathbf{y}}^*(\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j})^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p}_{j+1}\end{aligned}$$

the obtained functional value. The iteration schemes based on (4.8) will be shortly called *simultaneous minimization approach* (SIMI). We start with the following Theorem that gives us a necessary condition for the sequence of vectors $(\mathbf{p}_j)_{j=0}^{\infty}$ similarly as in Corollary 4.2.

Theorem 4.5 *Let* $\mathbf{y} = (y_k)_{k=0}^L$ *be given with* $2M \le L$. *Then, the vector* $\mathbf{p}_{j+1}$ *computed in* (4.8) *necessarily satisfies the eigenvector equation*

$$\left(\mathbf{H}_{\mathbf{y}}^*\left(\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}\right)^{-1}\mathbf{H}_{\mathbf{y}} - \mathbf{X}_{\mathbf{w}_j}^T\overline{\mathbf{X}}_{\mathbf{w}_j}\right)\mathbf{p}_{j+1} = \mathbf{0}, \tag{4.10}$$

where $\overline{\mathbf{X}}_{\mathbf{w}_j}$ *is generated as in* (2.9) *with the vector* $\overline{\mathbf{w}}_j := (\mathbf{X}_{\mathbf{p}_{j+1}}^T\overline{\mathbf{X}}_{\mathbf{p}_{j+1}})^{-1}\mathbf{X}_{\mathbf{p}_j}^T\mathbf{y}$.

Proof: The proof is similar to that of Theorem 4.1 and Corollary 4.2. With $\mathbf{p} = \mathbf{a}+\mathrm{i}\mathbf{b} = (a_k)_{k=0}^M+\mathrm{i}(b_k)_{k=0}^M$ and $\breve{\mathbf{p}} = (\mathbf{a}^T,\mathbf{b}^T)^T \in \mathbb{R}^{2M+2}$ it follows from (4.8) necessarily that $\nabla_{\breve{\mathbf{p}}}F(\mathbf{p},\mathbf{p}_j) = \mathbf{0}$ for $\mathbf{p} = \mathbf{p}_{j+1}$. As before, the conditions

$$\frac{\partial F(\mathbf{p},\mathbf{p}_j)}{\partial a_k} = 0 \qquad \text{and} \qquad \frac{\partial F(\mathbf{p},\mathbf{p}_j)}{\partial b_k} = 0, \qquad k = 0,\ldots,M$$

imply with $\mathbf{X}_{\mathbf{p}} = \sum_{\ell=0}^M p_\ell\,\mathbf{X}_\ell$ that

$$\sum_{\ell=0}^M p_\ell\mathbf{y}^*\left(\mathbf{X}_k\left(\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}\right)^{-1}\mathbf{X}_\ell^T - \overline{\mathbf{X}}_{\mathbf{p}_j}\left(\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}\right)^{-1}\mathbf{X}_\ell^T\mathbf{X}_k\left(\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}\right)^{-1}\mathbf{X}_{\mathbf{p}_j}^T\right)\mathbf{y} = 0. \tag{4.11}$$

Taking $\overline{\mathbf{w}} := \left(\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}\right)^{-1}\mathbf{X}_{\mathbf{p}_j}^T\mathbf{y}$ and observing that $\mathbf{X}_k\left(\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}\right)^{-1}\mathbf{X}_{\mathbf{p}_j}^T\mathbf{y} = \mathbf{X}_k\,\overline{\mathbf{w}} = \overline{\mathbf{X}}_{\mathbf{w}}\mathbf{e}_k$, we can conclude from (4.11)

$$\sum_{\ell=0}^M p_\ell\left(\mathbf{y}^*\mathbf{X}_k\left(\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}\right)^{-1}\mathbf{X}_\ell^T\mathbf{y} - \mathbf{e}_\ell^T\mathbf{X}_{\mathbf{w}}^T\overline{\mathbf{X}}_{\mathbf{w}}\mathbf{e}_k\right) = 0$$

for $k = 0,\ldots,M$. Since $\mathbf{X}_k^T\mathbf{y} = \mathbf{H}_{\mathbf{y}}\mathbf{e}_k$, it follows that $\mathbf{p}_{j+1}$ necessarily satisfies the eigenvector equation

$$\mathbf{H}_{\mathbf{y}}^*\left(\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}\right)^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p}_{j+1} - \mathbf{X}_{\mathbf{w}_j}^T\overline{\mathbf{X}}_{\mathbf{w}_j}\mathbf{p}_{j+1} = \mathbf{0}.$$

Thus the assertion follows. ■

Remark 4.6 *Observe that the eigenvector equation in* (4.10) *is still an implicit equation since* $\mathbf{w}_j$ *contains* $\mathbf{p}_{j+1}$ *in its definition. In particular,* (4.10) *implies by multiplication with* $(\mathbf{p}_{j+1})^*$ *that*

$$\begin{aligned}
\mathbf{p}_{j+1}^* \mathbf{H}_{\mathbf{y}}^* (\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j})^{-1} \mathbf{H}_{\mathbf{y}} \mathbf{p}_{j+1} &= \mathbf{p}_{j+1}^* \mathbf{X}_{\mathbf{w}_j}^T \overline{\mathbf{X}}_{\mathbf{w}_j} \mathbf{p}_{j+1} \\
&= \mathbf{w}_j^* \mathbf{X}_{\mathbf{p}_{j+1}}^T \overline{\mathbf{X}}_{\mathbf{p}_{j+1}} \mathbf{w}_j \\
&= \mathbf{y}^* \overline{\mathbf{X}}_{\mathbf{p}_j} (\mathbf{X}_{\mathbf{p}_{j+1}}^T \overline{\mathbf{X}}_{\mathbf{p}_{j+1}})^{-1} \mathbf{X}_{\mathbf{p}_j}^T \mathbf{y} \\
&= (\mathbf{p}_j^* \mathbf{H}_{\mathbf{y}}^* (\mathbf{X}_{\mathbf{p}_{j+1}}^T \overline{\mathbf{X}}_{\mathbf{p}_{j+1}})^{-1} \mathbf{H}_{\mathbf{y}} \mathbf{p}_j
\end{aligned}$$

and thus

$$\mathbf{y}^* \mathbf{A}(\mathbf{p}_{j+1}, \mathbf{p}_j) \mathbf{y} = \mathbf{y}^* \mathbf{A}(\mathbf{p}_j, \mathbf{p}_{j+1}) \mathbf{y}. \tag{4.12}$$

This result is remarkable since $\mathbf{A}(\mathbf{p}_{j+1}, \mathbf{p}_j)$ *is similar to the pseudo inverse of* $\mathbf{A}(\mathbf{p}_j, \mathbf{p}_{j+1})$*.*

Theorem 4.7 *Let* $\mathbf{y} = (y_k)_{k=0}^L$ *be given with* $2M \le L$*. Suppose that the normalized vector* $\mathbf{p}_{j+1}$ *obtained by the iteration* (4.8) *or by the condition* (4.10) *respectively, is always uniquely defined. Then the sequence* $(F(\mathbf{p}_j, \mathbf{p}_{j+1}))_{j=0}^{\infty}$ *obtained by* (4.9) *converges to a limit* F^**. Moreover, the desired vector*

$$\widetilde{\mathbf{p}} = \operatorname*{argmin}_{\substack{\mathbf{p} \in \mathbb{C}^{M+1} \\ \|\mathbf{p}\|_2 = 1}} \mathbf{y}^* \overline{\mathbf{X}}_{\mathbf{p}} [\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}]^{-1} \mathbf{X}_{\mathbf{p}}^T \mathbf{y} \tag{4.13}$$

is a fixed point of the iteration (4.8)*.*

Proof: 1. First we observe that $\mathbf{A}(\mathbf{p}_j, \mathbf{p}_{j+1})$ and $\mathbf{A}(\mathbf{p}_{j+1}, \mathbf{p}_j)$ are Hermitian and positive semidefinite, therefore $F(\mathbf{p}_j, \mathbf{p}_{j+1})$ is for all $j \in \mathbb{N}$ bounded from below by 0. By definition of the functional in (4.9) we have

$$\begin{aligned}
F(\mathbf{p}_j, \mathbf{p}_{j+1}) &\le F(\mathbf{p}_j, \mathbf{p}_j) = 2\mathbf{y}^* \overline{\mathbf{X}}_{\mathbf{p}_j} [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \mathbf{X}_{\mathbf{p}_j}^T \mathbf{y} \\
&= 2\mathbf{y}^* \overline{\mathbf{X}}_{\mathbf{p}_j} (\overline{\mathbf{X}}_{\mathbf{p}_j})^+ \mathbf{y},
\end{aligned}$$

where $(\overline{\mathbf{X}}_{\mathbf{p}_j})^+$ denotes the Moore-Penrose inverse of $\overline{\mathbf{X}}_{\mathbf{p}_j}$. Further, since $\overline{\mathbf{X}}_{\mathbf{p}_j} \overline{\mathbf{X}}_{\mathbf{p}_j}^+$ only possesses the eigenvalues 0 and 1 it follows that

$$2\mathbf{y}^* \overline{\mathbf{X}}_{\mathbf{p}_j} \overline{\mathbf{X}}_{\mathbf{p}_j}^+ \mathbf{y} \le 2\|\mathbf{y}\|_2^2 < \infty.$$

Thus, the sequence $(F(\mathbf{p}_j, \mathbf{p}_{j+1}))_{j=0}^{\infty}$ is bounded from above. Further, the sequence is monotonically decreasing since by (4.8)

$$F(\mathbf{p}_j, \mathbf{p}_{j+1}) \le F(\mathbf{p}_j, \mathbf{p}_{j-1}) = F(\mathbf{p}_{j-1}, \mathbf{p}_j).$$

Therefore, this sequence converges to a limit $F^* = \lim_{j \to \infty} F(\mathbf{p}_j, \mathbf{p}_{j+1})$.

2. We show now that $\widetilde{\mathbf{p}}$ in (4.13) is indeed a fixed point of the iteration (4.8). By definition, $\widetilde{\mathbf{p}}$ satisfies the the necessary condition (4.3) that takes here the form

$$\left(\mathbf{H}_{\mathbf{y}}^* (\mathbf{X}_{\widetilde{\mathbf{p}}}^T \overline{\mathbf{X}}_{\widetilde{\mathbf{p}}})^{-1} \mathbf{H}_{\mathbf{y}} - \mathbf{X}_{\widetilde{\mathbf{v}}}^T \overline{\mathbf{X}}_{\widetilde{\mathbf{v}}} \right) \widetilde{\mathbf{p}} = \mathbf{0}$$

with $\widetilde{\mathbf{v}} = \overline{\mathbf{X}}^+_{\widetilde{\mathbf{p}}}\mathbf{y}$, since

$$\begin{aligned}
\mathbf{H}^*_{\mathbf{r}(\widetilde{\mathbf{p}})}[\mathbf{X}^T_{\widetilde{\mathbf{p}}}\overline{\mathbf{X}}_{\widetilde{\mathbf{p}}}]^{-1}\mathbf{H}_{\mathbf{r}(\widetilde{\mathbf{p}})}\widetilde{\mathbf{p}} &= \mathbf{H}^*_{\mathbf{r}(\widetilde{\mathbf{p}})}[\mathbf{X}^T_{\widetilde{\mathbf{p}}}\overline{\mathbf{X}}_{\widetilde{\mathbf{p}}}]^{-1}\mathbf{X}^T_{\widetilde{\mathbf{p}}}\,\mathbf{r}(\widetilde{\mathbf{p}}) = \mathbf{H}^*_{\mathbf{r}(\widetilde{\mathbf{p}})}\overline{\mathbf{X}}^+_{\widetilde{\mathbf{p}}}\,\overline{\mathbf{X}}_{\widetilde{\mathbf{p}}}\,\overline{\mathbf{X}}^+_{\widetilde{\mathbf{p}}}\,\mathbf{y} \\
&= \mathbf{H}^*_{\mathbf{r}(\widetilde{\mathbf{p}})}\,\overline{\mathbf{X}}^+_{\widetilde{\mathbf{p}}}\,\mathbf{y} = \mathbf{H}^*_{\mathbf{r}(\widetilde{\mathbf{p}})}\,\widetilde{\mathbf{v}} = \mathbf{X}^T_{\widetilde{\mathbf{v}}}\,\overline{\mathbf{r}(\widetilde{\mathbf{p}})} = \mathbf{X}^T_{\widetilde{\mathbf{v}}}\,\mathbf{X}_{\widetilde{\mathbf{p}}}\,\mathbf{X}^+_{\widetilde{\mathbf{p}}}\,\overline{\mathbf{y}} \\
&= \mathbf{X}^T_{\widetilde{\mathbf{v}}}\,\mathbf{X}_{\widetilde{\mathbf{p}}}\,\overline{\mathbf{v}} = \mathbf{X}^T_{\widetilde{\mathbf{v}}}\,\overline{\mathbf{X}}_{\widetilde{\mathbf{v}}}\,\widetilde{\mathbf{p}}. \qquad (4.14)
\end{aligned}$$

Thus, taking $\mathbf{p}_j = \widetilde{\mathbf{p}}$ in (4.10), it follows that $\mathbf{p}_{j+1} = \widetilde{\mathbf{p}}$, i.e., $\nabla_{\mathbf{p}} F(\mathbf{p}, \widetilde{\mathbf{p}}) = \mathbf{0}$ for $\mathbf{p} = \widetilde{\mathbf{p}}$. ■

In the following we want to propose two different iteration schemes that both approximate the iteration (4.8) to solve the nonlinear problem (3.4), where we successively update the vector $\mathbf{p}$. We start with $\mathbf{p}_0$ in (4.4).

First iteration scheme (SIMI-1).

We recall that

$$\begin{aligned}
\mathbf{y}^*\mathbf{A}(\mathbf{p}_j, \mathbf{p})\mathbf{y} &= (\mathbf{p}_j)^*\mathbf{H}^*_{\mathbf{y}}[\mathbf{X}^T_{\mathbf{p}}\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\,\mathbf{H}_{\mathbf{y}}\mathbf{p}_j \\
&= (\mathbf{p}_j)^*\mathbf{H}^*_{\mathbf{y}}[\mathbf{X}^T_{\mathbf{p}}\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\,[\mathbf{X}^T_{\mathbf{p}}\overline{\mathbf{X}}_{\mathbf{p}}]\,[\mathbf{X}^T_{\mathbf{p}}\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\,\mathbf{H}_{\mathbf{y}}\mathbf{p}_j. \qquad (4.15)
\end{aligned}$$

Approximating $[\mathbf{X}^T_{\mathbf{p}}\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}$ by $[\mathbf{X}^T_{\mathbf{p}_j}\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}$ in (4.15), we obtain

$$\begin{aligned}
\mathbf{y}^*\widetilde{\mathbf{A}}(\mathbf{p}_j, \mathbf{p})\mathbf{y} &= (\mathbf{p}_j)^*\mathbf{H}^*_{\mathbf{y}}[\mathbf{X}^T_{\mathbf{p}_j}\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\,[\mathbf{X}^T_{\mathbf{p}}\overline{\mathbf{X}}_{\mathbf{p}}]\,[\mathbf{X}^T_{\mathbf{p}_j}\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\,\mathbf{H}_{\mathbf{y}}\mathbf{p}_j \\
&= \mathbf{v}^*_j[\mathbf{X}^T_{\mathbf{p}}\,\overline{\mathbf{X}}_{\mathbf{p}}]\mathbf{v}_j
\end{aligned}$$

with $\mathbf{v}_j := [\mathbf{X}^T_{\mathbf{p}_j}\,\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\,\mathbf{H}_{\mathbf{y}}\mathbf{p}_j = \overline{\mathbf{X}}^+_{\mathbf{p}_j}\mathbf{y}$. Using this approximation we arrive at the first iteration scheme

$$\begin{aligned}
\mathbf{p}_{j+1} &:= \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmin}} \Big(\mathbf{y}^*\mathbf{A}(\mathbf{p}, \mathbf{p}_j)\mathbf{y} + \mathbf{y}^*\widetilde{\mathbf{A}}(\mathbf{p}_j, \mathbf{p})\mathbf{y}\Big) \\
&= \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmin}} \Big(\mathbf{p}^*\,\mathbf{H}^*_{\mathbf{y}}\,[\mathbf{X}^T_{\mathbf{p}_j}\,\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\,\mathbf{H}_{\mathbf{y}}\,\mathbf{p} + \mathbf{v}^*_j[\mathbf{X}^T_{\mathbf{p}}\,\overline{\mathbf{X}}_{\mathbf{p}}]\,\mathbf{v}_j\Big), \\
&= \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmin}} \Big(\mathbf{p}^*\,\mathbf{H}^*_{\mathbf{y}}\,[\mathbf{X}^T_{\mathbf{p}_j}\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\,\mathbf{H}_{\mathbf{y}}\,\mathbf{p} + \mathbf{p}^*[\mathbf{X}^T_{\mathbf{v}_j}\,\overline{\mathbf{X}}_{\mathbf{v}_j}]\,\mathbf{p}\Big). \qquad (4.16)
\end{aligned}$$

In the last equation, we have used that $\overline{\mathbf{X}}_{\mathbf{p}}\mathbf{v}_j = \mathbf{X}_{\mathbf{v}_j}\overline{\mathbf{p}}$ holds. Now each iteration step breaks down to finding the eigenvector to the smallest eigenvalue of the positive semidefinite matrix $\mathbf{H}^*_{\mathbf{y}}\,[\mathbf{X}^T_{\mathbf{p}_j}\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\,\mathbf{H}_{\mathbf{y}} + \mathbf{X}^T_{\mathbf{v}_j}\overline{\mathbf{X}}_{\mathbf{v}_j}$.

Remark 4.8 *The proposed iteration scheme in* (4.16) *is close to the GRA iteration in nature. The GRA algorithm employs the matrix* $\mathbf{H}^*_{\mathbf{y}-\mathbf{r}(\mathbf{p}_j)}(\mathbf{X}^T_{\mathbf{p}_j}\overline{\mathbf{X}}_{\mathbf{p}_j})^{-1}\mathbf{H}_{\mathbf{y}}$ *in* (4.6)*. But we have*

$$\mathbf{H}^*_{\mathbf{r}(\mathbf{p})}(\mathbf{X}^T_{\mathbf{p}}\overline{\mathbf{X}}_{\mathbf{p}})^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p} = \mathbf{H}^*_{\mathbf{r}(\mathbf{p})}(\mathbf{X}^T_{\mathbf{p}}\overline{\mathbf{X}}_{\mathbf{p}})^{-1}\mathbf{H}_{\mathbf{r}(\mathbf{p})}\mathbf{p} = \mathbf{X}^T_{\mathbf{v}}\overline{\mathbf{X}}_{\mathbf{v}}\mathbf{p}$$

similarly as in (4.14)*. Thus, for both iterations, the iteration matrix is composed of the two positive definite matrices* $\mathbf{H}^*_{\mathbf{y}}[\mathbf{X}^T_{\mathbf{p}_j}\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\mathbf{H}_{\mathbf{y}}$ *and* $\mathbf{X}^T_{\mathbf{v}_j}\overline{\mathbf{X}}_{\mathbf{v}_j}$*. But while* (4.16) *employs the sum of these matrices, GRA considers the matrix difference and searches for its eigenvector to the eigenvalue closest to zero. For small noise the vector* $\mathbf{v}_j$ *has a small norm and the second matrix becomes negligible. As we will see in the numerical experiments, the new iteration scheme is more robust with regard to the initial vector* $\mathbf{p}_0$ *than GRA.*

The first iteration scheme above has similar computation costs as GRA and can be quite time-consuming since we need to compute the iteration matrix and to apply for example inverse iteration to compute the desired smallest eigenvalue and the corresponding eigenvector. Therefore, we derive a further scheme that directly applies the shorter pre-smoothed vector $\widetilde{\mathbf{y}}$ from (2.13) or (2.14) and completely avoids inverse covariance matrices in the iteration scheme.

Second iteration scheme (SIMI-2).
We start with $\widetilde{\mathbf{y}}$ from (2.13) or (2.14) such that $\mathbf{H}_{\widetilde{\mathbf{y}}} \in \mathbb{C}^{(M+1)\times(M+1)}$ is a square matrix. We employ now a different approximation of the term $\widetilde{\mathbf{y}}^* \mathbf{A}(\mathbf{p}_j, \mathbf{p})\widetilde{\mathbf{y}}$. With $\mathbf{q}_j := \mathbf{H}_{\widetilde{\mathbf{y}}}\mathbf{p}_j$ we conclude that

$$\begin{aligned}
\underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmin}} \widetilde{\mathbf{y}}^* \mathbf{A}(\mathbf{p}_j, \mathbf{p})\widetilde{\mathbf{y}} &= \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmin}} \ \mathbf{p}_j^* \mathbf{H}_{\widetilde{\mathbf{y}}}^* \, [\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}]^{-1} \mathbf{H}_{\widetilde{\mathbf{y}}} \, \mathbf{p}_j \\
&= \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmin}} \ \mathbf{q}_j^* \, [\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}]^{-1} \, \mathbf{q}_j \\
&= \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmax}} \ \mathbf{q}_j^* \, [\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}] \, \mathbf{q}_j \\
&= \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmax}} \ \mathbf{p}^* [\mathbf{X}_{\mathbf{q}_j}^T \overline{\mathbf{X}}_{\mathbf{q}_j}] \, \mathbf{p}. \qquad (4.17)
\end{aligned}$$

On the other hand we get,

$$\begin{aligned}
\underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmin}} \widetilde{\mathbf{y}}^* \mathbf{A}(\mathbf{p}, \mathbf{p}_j)\widetilde{\mathbf{y}} &= \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmin}} \mathbf{p}^* \mathbf{H}_{\widetilde{\mathbf{y}}}^* \, [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \, \mathbf{H}_{\widetilde{\mathbf{y}}} \, \mathbf{p} \\
&= \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmax}} \mathbf{p}^* \, \mathbf{H}_{\widetilde{\mathbf{y}}}^{-1} \, [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}] \, (\mathbf{H}_{\widetilde{\mathbf{y}}}^*)^{-1} \, \mathbf{p}.
\end{aligned}$$

Therefore, we suggest the second iteration scheme

$$\mathbf{p}_{j+1} := \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmax}} \mathbf{p}^* \left[\mathbf{H}_{\widetilde{\mathbf{y}}}^{-1} [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}] (\mathbf{H}_{\widetilde{\mathbf{y}}}^*)^{-1} \, + \, [\mathbf{X}_{\mathbf{q}_j}^T \overline{\mathbf{X}}_{\mathbf{q}_j}] \right] \mathbf{p}. \qquad (4.18)$$

Here, each new iteration matrix can be obtained without determining inverses of covariance matrices. The inverse $\mathbf{H}_{\widetilde{\mathbf{y}}}^{-1}$ needs to be calculated only once in a preprocessing step, while we need only $\mathcal{O}(M^2)$ arithmetical operations to compute the iteration matrix

$$\mathbf{B}(\mathbf{p}_j) := \left[(\mathbf{H}_{\widetilde{\mathbf{y}}}^*)^{-1} [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}] \mathbf{H}_{\widetilde{\mathbf{y}}}^{-1} \, + \, \mathbf{X}_{\mathbf{q}_j}^T \overline{\mathbf{X}}_{\mathbf{q}_j} \right].$$

We summarize the procedure with one of the two iteration schemes in Algorithm 4.9.

Algorithm 4.9 (SIMI)
Input: M, x_0, h, y_k, $k = 0, \ldots, L$, *with* $L + 1 = (2M + 1)K$.

1. *Initialization*
   - *(Optional for the iteration* (4.16)*): Compute* $\widetilde{\mathbf{y}}$ *in* (2.13) *or* (2.14) *and replace in all further steps* $\mathbf{y}$ *by* $\widetilde{\mathbf{y}}$*.*

- *Compute* $\mathbf{p}_0$ *in* (4.4).
- *If the iteration* (4.18) *is applied then pre-compute* $\mathbf{H}_{\mathbf{y}}^{-1}$ *with high precision.*

2. *Iteration: For* $j = 0 \ldots$ *till convergence*
   - *Compute* $\mathbf{p}_{j+1}$ *according to* (4.16) *or* (4.18).
3. *Denote by* $\widetilde{\mathbf{p}}$ *the vector obtained by that iteration and compute the vector* $\mathbf{z}$ *of zeros* z_j, $j = 1, \ldots, M$, *of the Prony polynomial* $\widetilde{p}(z) = \sum_{k=0}^{M} \widetilde{p}_k z^k$ *by solving an eigenvalue problem for the corresponding companion matrix.*
4. *Compute the coefficients* d_j, $j = 1, \ldots, M$ *by solving the least squares problem*

$$\mathbf{V}(z)\mathbf{d} = \mathbf{y}$$

and derive $c_j := e^{-T_j x_0} d_j = z_j^{x_0/h} d_j$.

Output: Parameter vectors $\mathbf{z}$, $\mathbf{c}$.

In Algorithm 4.9, convergence is achieved if $\|\mathbf{p}_j - \mathbf{p}_{j+1}\|_2 < \epsilon$ for some predefined positive value ϵ. In our numerical results, we have employed $\epsilon = 10^{-8}$. Concerning the convergence properties of the two proposed iteration schemes we observe the following. Similarly as in the proof of Theorem 4.7 the achieved functional values in the two iterations are bounded from below and from above. Therefore the sequence of functional values possesses accumulation points. Since the functional values continuously depend on the iteration vectors, there can be only finitely many accumulation points and the Česaro mean of the sequence of functional values as well as the corresponding mean of the iteration vectors always converges.

5 Numerical examples

We want to compare the different iteration methods and show that they all converge in practice. We will consider the results of the least squares Prony method (Pisarenko method) (PM), the approximate Prony (APM) in [36], the GRA iteration in Algorithm 4.3, the IQML iteration in Algorithm 4.4, the VARPRO method based on Levenberg-Marquardt iteration using the software package of [27], and the two new SIMI-iterations (4.16) and (4.18) in Algorithm 4.9. For all algorithms we will also will use the smoothed data $\widetilde{\mathbf{y}}$ alongside the original data vector $\mathbf{y}$. The new scheme (4.18) is only applied to smoothed data $\widetilde{\mathbf{y}}$. Besides achieving a much smaller error variance already in the initialization, a further advantage is that the obtained Hermitian Toeplitz matrix $\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}$ is only of size $(M+1) \times (M+1)$ at each iteration step in GRA, IQML and (4.16). Taking our proposed iteration scheme (4.18), we even completely circumvent the computation of the inverse $(\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j})^{-1}$, and instead need to compute the $(M+1) \times (M+1)$ matrix $\mathbf{H}_{\widetilde{\mathbf{y}}}^{-1}$ only once in a preprocessing step.

For the first and second example we employ the right singular vector to the smallest singular value of $\mathbf{H}_{\mathbf{y}}$ or of $\mathbf{H}_{\widetilde{\mathbf{y}}}$ as initial vectors, respectively. Only in the last example we test different initial vectors. Noticing that all the methods used for computing the coefficients d_j (or c_j) are the same, we will only present the estimates of parameters T_j, the number of the iterations (NoI) and the *relative error* of the exponential sum,

$$e(f) = \frac{\max\limits_{k=0,\ldots,L} |f(x_k) - \widehat{f}(x_k)|}{\max\limits_{k=0,\ldots L} |f(x_k)|},$$

		PM	APM	GRA	IQML	VARPRO	SIMI-1	SIMI-2
							(4.16)	(4.18)
$L = 11$	NoI	\	\	6	5	4	5	
	$\widehat{T}$	-4.6939	-4.1239	-4.0941	-3.3699	-4.0068	-4.0089	
	rel. error	0.0955	0.0844	0.0742	0.0733	0.0815	0.0729	
	2-error	0.0266	0.0259	0.0249	0.0249	0.0262	0.0249	
with filter	NoI	\	\	7	6	3	4	6
	$\widehat{T}$	-3.9933	-3.9765	-4.0136	-4.2473	-3.9966	-4.0324	-4.0226
	rel. error	0.0914	0.0918	0.0900	0.0899	0.0929	0.0897	0.0899
	2-error	0.0262	0.0228	0.0262	0.0262	0.0277	0.0262	0.0262
$L = 29$	NoI	\	\	5	4	4	4	
	$\widehat{T}$	-6.0093	-4.0506	-4.0260	-3.9002	-4.0675	-3.9684	
	rel. error	0.1689	0.0758	0.0534	0.0531	0.0608	0.0531	
	2-error	0.0220	0.0183	0.0175	0.0175	0.0172	0.0175	
with filter	NoI	\	\	6	6	3	4	6
	$\widehat{T}$	-4.0366	-4.0298	-4.0396	-3.8199	-4.0446	-4.0474	-4.0434
	rel. error	0.0632	0.0632	0.0624	0.0625	0.0790	0.0626	0.0625
	2-error	0.0177	0.0188	0.0177	0.0177	0.0176	0.0177	0.0177

Table 5.1. Simulation results for perturbed signal values $y_k = \exp(-4x_k) + \epsilon_k$, $k = 0, 1, \ldots, L$ and the low-pass filtered data $\widetilde{y}_k, k = 0, 1, 2$ in Example 5.1.

where $\widehat{f}(x) = \sum_{j=1}^{M} \widehat{c}_j e^{\widehat{T}_j x}$ is the signal recovered by the corresponding algorithms. Further, we consider the normalized 2-*error*

$$\frac{1}{L+1}\Big(\sum_{k=0}^{L} |y_k - \widehat{f}(x_k)|^2\Big)^{1/2}$$

that measures the distance of the recovered signal to the measured signal $\mathbf{y}$.

In the first and the second example, we present the mean values $\hat{T}_j$ for T_j, the mean relative error $e(f)$ and the mean 2-error obtained from 100 simulated data sets. In the third example, we have considered single data vectors $\mathbf{y}$.

Example 5.1 *In a first small example from [18] we consider the data*

$$y_k = \exp(-4x_k) + \epsilon_k, \qquad k = 0, 1, \ldots, L$$

where $x_k = \frac{k}{L}$ and $\epsilon_k \sim N(0, 0.01)$, i.e. the deviation is $\sigma = 0.1$. We use either the full data vector $\mathbf{y}$ with $L = 11$ or $L = 29$ or the filtered data $\widetilde{\mathbf{y}}$ in (2.13)*. The bound for the highest number of iterations is set to* 10*. We compare the results for these two data sets for each algorithm in Table* 5.1*. The mean values of the normalized 2-error $\frac{1}{L+1}\left(\sum_{k=0}^{L} |f(x_k) - y_k)|^2\right)^{1/2}$ for the measured samples are* 0.0291 *and* 0.0181 *for $L = 11$ or $L = 29$, respectively.*

Comparing the iterative algorithms, we see that for filtered data IQML works slightly worse than the others. Comparing the mean 2*-error, we observe that all iterative schemes achieve a better* 2*-error than the Pisarenko method and also a better* 2*-error than obtained using the correct parameters. This means, that all iterative algorithms provide parameters $\widehat{T}$ and $\widehat{c}$ that approximate the the measured values better than the original sum with $T = -4$ and $c = 1$.*

		PM	APM	GRA	IQML	VARPRO	SIMI-1 (4.16)	SIMI-2 (4.18)
$\sigma = 0.001$	NoI	\	\	5	4	3	4	
	$\widehat{T}$	-3.5191 -8.2169	-4.0239 -6.9441	-3.9947 -7.0152	-3.9954 -7.0147	-4.0028 -6.9932	-3.9951 -7.0139	
	rel. error	0.0130	0.0054	0.0013	0.0013	0.0013	0.0013	
	2-error	$7.48e-04$	$3.31e-04$	$1.33e-04$	$1.33e-04$	$1.34e-04$	$1.33e-04$	
with filter	NoI	\	\	4	3	4	3	3
	$\widehat{T}$	-3.9958 -7.0113	-3.9960 -7.0107	-3.9964 -7.0092	-4.0351 -6.8632	-4.0128 -6.9619	-3.9963 -7.0094	-3.9964 -7.0093
	rel. error	0.0028	0.0028	0.0027	0.0027	0.0015	0.0027	0.0027
	2-error	$1.49e-04$	$1.29e-04$	$1.48e-04$	$1.48e-04$	$1.40e-04$	$1.48e-04$	$1.48e-04$
$\sigma = 0.01$	NoI	\	\	7	5	5	6	
	$\widehat{T}$	-1.5027 -86.868	-3.9500 -7.1545	-3.9677 -7.0955	-3.9077 -7.4468	-4.0217 -6.9288	-4.0159 -6.9478	
	rel. error	0.1716	0.0671	0.0122	0.0121	0.0126	0.0121	
	2-error	0.0110	0.0037	0.0014	0.0014	0.0014	0.0014	
with filter	NoI	\	\	6	5	6	5	5
	$\widehat{T}$	-3.9214 -7.2888	-3.9365 -7.2275	-3.9442 -7.2016	-4.1387 -6.4836	-3.9509 -7.2284	-3.9375 -7.2298	-3.9408 -7.2158
	rel. error	0.0323	0.0321	0.0297	0.0298	0.0201	0.0298	0.0298
	2-error	0.0016	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015
$\sigma = 0.05$	NoI	\	\	10	6	10	10	
	$\widehat{T}$	-1.7073 -41+154i	-1.3131 -83.2645	-2.2585 98.3110	-5.025 ± 0.82i	-2.7758 -19.0629	-4.676 ± 1.40i	
	rel. error	0.2021	0.3576	0.5998	0.0571	0.0859	0.0626	
	2-error	0.0117	0.0192	0.0279	0.0069	0.0071	0.0069	
with filter	NoI	\	\	6	5	6	6	5
	$\widehat{T}$	-3.7450 -7.8443	-4.832 ±0.80i	-3.7361 -8.3108	-4.322 ±1.42i	-3.353 ±2.14i	-4.0268 -6.8894	-2.543 ±2.20i
	rel. error	0.1502	0.1558	0.1490	0.1460	0.1362	0.1473	0.1460
	2-error	0.0080	0.0074	0.0078	0.0078	0.0076	0.0078	0.0078

Table 5.2. Simulation results for perturbed signal values $y_k = 2\exp(-4x_k) - \exp(-7x_k) + \epsilon_k$, $k = 0, 1, \ldots, L$ and the low-pass filtered data $\widetilde{y}_k$, $k = 0, 1, 2, 3, 4$ in Example 5.2.

Example 5.2 *We consider the example in* [31] *with* $M = 2$ *of the form*

$$y_k = 2\exp(-4x_k) - 1.5\exp(-7x_k) + \epsilon_k, \qquad k = 0, 1, \ldots, L$$

where $x_k = \frac{k}{L}$ *and* $\epsilon_k \sim N(0, \sigma^2)$. *We fix the length* L *of given values to* $L = 49$ *and consider different variances of noise,* $\sigma = 0.001$, $\sigma = 0.01$, *and* $\sigma = 0.05$. *The bound for the highest number of iterations is set to* 10. *The mean values of the normed 2-error for the measured samples are* $1.39e-04$, 0.0014 *and* 0.0071 *for* $\sigma = 0.001$, 0.01 *and* 0.05, *respectively. As it can be seen for strong noise, all algorithms give parameters being completely different from the original parameter set when applying a larger data vector, but the preprocessing step can strongly help to obtain reasonable results for parameter estimation, see Table* 5.2. *However, we observe that the algorithms achieve a* 2*-error being in the same range as the* 2*-error for the given data.*

Example 5.3 *Finally, we investigate a three-term model of the form*

$$y_k = \exp(0.95\, x_k) + \exp(0.5\, x_k) + \exp(0.2\, x_k) + \epsilon_k, \qquad k = 0, 1, \ldots, L$$

where $x_k = \frac{5k}{L}$, and $\epsilon_k \sim N(0, \sigma^2)$ with $\sigma \in \{0.0001, 0.001, 0.01\}$. Observe that in this case the exponentials $\exp(0.95)$, $\exp(0.5)$ and $\exp(0.2)$ are larger than 1 such that the sequence exponentially increases for $x_k > 1$. Again, the filtered data $\widetilde{y}_k, k = 0, \ldots, 6$ is also considered. We employed a fixed number of $L = 69$ samples. We have computed here only the the parameters of one noisy measured vector $\mathbf{y}$ (without any averaging of results). The values of the normed 2-error for the measured samples are $1.3371e-05$, $1.0789e-04$ and 0.0014 for $\sigma = 0.0001$, 0.001 and 0.01, respectively.

In this example we further investigated the influence of the initial vector $\mathbf{p}_0$ and replaced it by the singular vectors of $\mathbf{H}_{\mathbf{y}}^* \mathbf{H}_{\mathbf{y}}$ (and $\mathbf{H}_{\widetilde{\mathbf{y}}}^* \mathbf{H}_{\widetilde{\mathbf{y}}}$, respectively) to the second or third smallest singular value. The bound for the highest number of iterations has been set to 20. The results are given in Table 5.3. As one can see, the GRA iteration depends more strongly on the starting vector than the other algorithms. Further, for strong noise all algorithms provide in the last part of the table parameters for the frequencies T_j that are completely different from the original parameter vector $(0.95, 0.5, 0.2)^T$. But the 2-error shows that the found parameters indeed admit an approximation of the noisy data vector by a three-term exponential sum being equally good as the original parameter vector. Thus, from approximation point of view all algorithms work well.

Remark 5.4 *All considered iterations usually converge very quickly. However, for stronger noise the GRA iteration as well as VARPRO and the iteration (4.16) have used the full number of iterations up to the bound. Considering the computational costs, we can say that GRA and iteration (4.16) require a similar computational effort. The corresponding iteration matrices are composed by the same two positive definite matrices. The computation of one new iteration vector requires for GRA the computation of the eigenvector with eigenvalue closest to zero while for the iteration (4.16) we need to find the singular vector/eigenvector to the smallest eigenvalue. GRA and the iteration (4.16) are much more expensive than IQML and the iteration (4.18), which are close in spirit especially for the case of filtered vectors. In this case the second matrix in the iteration (4.18) is very small for small error. The iteration (4.18) is by far the fastest algorithm taking only about $1/20$ of the computation time of the GRA algorithm and $1/3$ of the computation time for the IQML iteration and $1/10$ of the computation time for the VARPRO in the third example.*

		APM	GRA	IQML	VARPRO	SIMI-1	SIMI-2
						(4.16)	(4.18)
$\sigma = 0.0001$	NoI	\	20	4	3	20	
$\mathbf{p}_0$ is last singular vector	$\widehat{T}$	2.3767 0.9347 0.3230	0.9492 0.4797 0.1808	0.9497 0.4930 0.1939	0.9496 0.4919 0.1938	0.9495 0.4879 0.1891	
	rel. error	$5.36e-05$	$8.11e-07$	$4.17e-07$	$4.97e-07$	$5.42e-07$	
	2-error	$5.58e-04$	$1.33e-05$	$1.29e-05$	$1.28e-05$	$1.29e-05$	
with filter	NoI	\	20	3	1	9	3
	$\widehat{T}$	0.9500 0.4993 0.1991	0.9501 0.5002 0.1994	0.9501 0.5002 0.1994	0.9504 0.5085 0.2063	0.9501 0.5002 0.1994	0.9501 0.5002 0.1994
	rel. error	$7.60e-07$	$1.07e-06$	$1.07e-06$	$1.57e-06$	$1.07e-06$	$1.07e-06$
	2-error	$1.31e-05$	$1.34e-05$	$1.34e-05$	$1.18e-05$	$1.34e-05$	$1.34e-05$
$\sigma = 0.001$	NoI	\	20	6	18	20	
$\mathbf{p}_0$ is third last singular vector	$\widehat{T}$	7.9810 0.9368 0.3288	0.9474 0.4406 0.1295	0.9475 0.4433 0.1345	0.9503 0.5108 0.2084	0.9474 0.4417 0.1315	
	rel. error	0.6449	$4.39e-06$	$4.30e-06$	$3.28e-06$	$4.36e-06$	
	2-error	4.3245	$1.05e-04$	$1.05e-04$	$1.40e-04$	$1.05e-04$	
with filter	NoI	\	12	4	20	20	4
	$\widehat{T}$	0.9480 0.4476 0.1351	0.8412 -0.12+3.6i -0.12-3.6i	0.9476 0.4439 0.1330	0.9510 0.5220 0.2146	0.9476 0.4440 0.1330	0.9476 0.4439 0.1330
	rel. error	$4.04e-06$	0.0158	$3.82e-06$	$2.44e-06$	$3.82e-06$	$3.82e-06$
	2-error	$1.07e-04$	0.1008	$1.06e-04$	$1.41e-04$	$1.06e-04$	$1.06e-04$
$\sigma = 0.01$	NoI	\	20	7	8	20	
$\mathbf{p}_0$ is second last singular vector	$\widehat{T}$	1.36+43i 0.9421 0.3569	0.9408 0.3477 -1.6154	0.9415 0.3526 -1.0412	0.9386 0.3358 -0.65+43i	0.9420 0.3561 -0.8045	
	rel. error	$3.71e-04$	$7.90e-05$	$6.19e-05$	$8.49e-05$	$5.39e-05$	
	2-error	0.0026	0.0013	0.0013	0.0013	0.0013	
with filter	NoI	\	20	9	6	20	9
	$\widehat{T}$	2.2492 0.9299 0.3165	0.9381 0.3340 -0.51+4.3i	2.9715 0.9310 0.3167	0.9355 0.3266 1.70+4.3i	3.1895 0.9313 0.3173	2.9715 0.9310 0.3167
	rel. error	$9.25e-05$	$1.22e-04$	$1.14e-04$	$1.87e-04$	$1.18e-04$	$1.14e-04$
	2-error	0.0015	0.0014	0.0016	0.0015	0.0016	0.0016

Table 5.3. Simulation results for perturbed signal values $y_k = \exp(0.95\,x_k) + \exp(0.5\,x_k) + \exp(0.2\,x_k) + \epsilon_k$, $k = 0, 1, \ldots, L$ and the low-pass filtered data $\widetilde{y}_k$, $k = 0, \ldots, 6$ in Example 5.3.

6 Acknowledgement

This work was funded by the NSFC (11571078), by a CSC Scholarship and by the German Research Foundation within the framework of the RTG 2088.

References

[1] V. M. Adamjan, D. Z. Arov, and M. G. Krein, Analytic properties of the Schmidt pairs of a Hankel operator and the generalized Schur-Takagi problem. *Mathematics of the USSR-Sbornik* **15**(1) (1971), 31–73.

[2] F. Andersson, M. Carlsson, and M. V. de Hoop, Sparse approximation of functions using sums of exponentials and AAK theory. *J. Approx. Theory* **163** (2011), 213–248.

[3] F. Andersson and M. Carlsson, Fixed-point algorithms for frequency estimation and structured low-rank approximation, *Appl. Comput. Harmon. Anal.*, to appear.

[4] Z.D. Bai, C.R. Rao, M. Chow, and D. Kundu, An efficient algorithm for estimating the parameters of superimposed exponential signals, *J. Statist. Plann. Inference* **110**(1) (2003), 23–34.

[5] G. Beylkin and L. Monzón, On approximation of functions by exponential sums, *Appl. Comput. Harmon. Anal.* **19** (2005), 17–48.

[6] G. Beylkin and L. Monzón, Approximation by exponential sums revisited, *Appl. Comput. Harmon. Anal.* **28** (2010), 131–148.

[7] J. Bian, H. Peng, J, Xing, Z. Liu, and H. Li, An efficient algorithm for estimating the parameters of superimposed exponential signals in multiplicative and additive noise, *Int. J. Appl. Math. Comput. Sci.* **23**(1), 117–129.

[8] F. Boßmann, G. Plonka, T. Peter, O. Nemitz, and T. Schmitte, Sparse deconvolution methods for ultrasonic NDT, *J. Nondestruct. Eval.* **31** (2012), 225–244.

[9] D. Braess and W. Hackbusch, On the efficient computation of high-dimensional integrals and the approximation by exponential sums, *in: Multiscale, Nonlinear and Adaptive Approximation, R.A. DeVore and A. Kunoth (eds.)*, Springer, Berlin, 2009.

[10] Y. Bresler and A. Macovski, Exact maximum likelihood parameter estimation of superimposed exponential signals in noise, *IEEE Trans. Acoust., Speech, Signal Process. ASSP* **34**(5) (1986), 1081–1089.

[11] Z. Doğan, C. Gilliam, T. Blue, and D. Van de Ville, Reconstruction of finite rate of innovation signals with model-fitting approach, *IEEE Trans. Signal Process.* **63**(22), (2015), 6024–6036.

[12] P.L. Dragotti, M. Vetterli, and T. Blu, Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang–Fix, *IEEE Trans. Signal Process.* **55** (2007), 1741–1757.

[13] E.A. Feilat, Prony analysis technique for estimation of the mean curve of lightning impulses. *IEEE Trans. Power Del.* 21(4) (2006), 2088–2090.

[14] G.H. Golub and V. Pereyra, The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM J. Numer. Anal.* **10**(2) (1973), 413-432.

[15] M. Hanke, One shot inverse scattering via rational approximation, *SIAM J. Imaging Science* **5**(1) (2012), 465–482.

[16] Y. Hua and T.K. Sarkar, On the total least squares linear prediction method for frequency estimation, *IEEE Trans. Acoust. Speech Signal Process., ASSP* **38**(12) (1990), 2186–2189.

[17] M. Ishteva, K. Usevich, and I. Markovsky. Factorization approach to structured low-rank approximation with applications. *SIAM J. Matrix Anal. Appl.*, **35**(3) (2014), 1180–1204.

[18] M.H. Kahn, M.S. Mackisack, M. R. Osborne and G. K. Smyth, On the consistency of Prony's method and related algorithms, *J. Comp. Graph. Statist.* **1** (1992), 329–349.

[19] A. Kukush, I. Markovsky, and S. Van Huffel, Consistency of the structured total least squares estimator in a multivariate errors-in-variables model, *J. Statist. Plann. Inference* **133**(2) (2005), 315–358.

[20] R. Kumaresan, L.L. Scharf, and A.K. Shaw, An algorithm for pole-zero modeling and spectral analysis, *IEEE Trans. Acoust. Speech Signal Process., ASSP* **34**(3) (1986), 637–640.

[21] D. Kundu and S. Nandi, Statistical Signal Processing: Frequency Estimation, Springer, New Dehli, 2012.

[22] P. Lemmerling, N. Mastronardi, and S. Van Huffel, Fast algorithm for solving the hankel/toeplitz structured total least squares problem. *Numer. Algor.* **23**(4) (2000), 37–392.

[23] J. Li, P. Stoica, and Z.-S. Liu, Comparative study of IQML and mode direction-of-arrival estimators, *IEEE Trans. Signal Process.* **46**(1) (1998), 149–160.

[24] D.G. Manolakis, V.K. Ingle, and S.M. Kogon, Statistical and Adaptive Signal Processing, McGraw-Hill, Boston, 2005.

[25] I. Markovsky, Structured low-rank approximation and its applications, *Automatica* **44**(4) (2008), 891–909.

[26] I. Markovsky, *Low-Rank Approximation: Algorithms, Implementation, Applications*, Springer, second edition edition, 2018.

[27] I. Markovsky and K. Usevich, Software for weighted structured low-rank approximation, *J. Comput. Appl. Math.* **256** (2014), 278–292.

[28] I. Markovsky, S. Van Huffel, and R. Pintelon, Block-Toeplitz/Hankel structured total least squares, *SIAM J. Matrix Anal. Appl.* **26**(4) (2005), 1083–1099.

[29] C. Ortbandt, C. Dzienis, R. Matussek and H. Schulte, Parameter estimation in electrical power systems using Prony's method, *Journal of Physics* **659**(1) (2015), pp. 012013.

[30] M. R. Osborne, Some Special nonlinear least square problems, *SIAM J. Numer. Anal.* **12** (1975), 571–592.

[31] M. R. Osborne and G.K. Smyth, A modified Prony algorithm for exponential function fitting, *SIAM J. Sci. Comput.* **16**(1) (1995), 119–138.

[32] M. R. Osborne and G.K. Smyth, A modified Prony algorithm for fitting functions defined by difference equations, *SIAM J. Sci. Stat. Comput.* **12** (1991), 362–382.

[33] V. F. Pisarenko, The retrieval of harmonics from a covariance function, *Geophysics, J. Roy. Astron. Soc.* **33** (1973), 347–366.

[34] G. Plonka and M. Tasche, Prony methods for recovery of structured functions, *GAMM-Mitt.* **37**(2) (2014), 239–258.

[35] G. Plonka and V. Pototskaia. Application of the AAK theory for sparse approximation of exponential sums. arXiv:1609.09603, 2016.

[36] D. Potts, and M. Tasche, Nonlinear approximation by sums of nonincreasing exponentials, *Appl. Anal.* **90** (2011), 609–626.

[37] D. Potts and M. Tasche, Parameter estimation for nonincreasing sums by Prony-like methods, *Linear Algebra Appl.* **439**(4) (2013), 1024–1039.

[38] A. Rahman and K.-B. Yu, Total least squares approach for frequency estimation using linear prediction, *IEEE Trans. Acoust. Speech Signal Process., ASSP* **35**(10) (1987), 1440–1454.

[39] R. Roy and T. Kailath, ESPRIT estimation of signal parameters via rotational invariance techniques, *IEEE Trans. Acoust. Speech Signal Process* **37** (1989), 984–995.

[40] R.O. Schmidt, A signal subspace approach to multiple emitter location and spectral estimation, PhD thesis, Stanford University, 1981.

[41] J.A. de la O Serna, Synchrophasor estimation using Prony's method, *IEEE Trans. Instrumentation Measurement* **62**(8) (2013), 2119–2128.

[42] M.-R. Skrzipek, Signal recovery by discrete approximation and a Prony-like method, *J. Comput. Appl. Math.* **326** (2017) 193–203.

[43] C. W. Therrien and C.H. Velsaco, An iterative method for ARMA signal modeling, *IEEE Trans. Signal Process.* **43**(1) (1995), 358–361.

[44] K. Usevich and I. Markovsky, Variable projection for affinely structured low-rank approximation in weighted 2-norms. *J. Comput. Appl. Math.*, **272** (2014), 430–448.

[45] L. Weiss and R.N. McDonough, Prony's Method, Z-Transforms, and Padé Approximation, *SIAM Rev.* **5** (1963), 145–149.