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

Ran Zhang* Gerlind Plonka†

February 5, 2019

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-convex minimization 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 numerical approaches. With the help of an explicitly derived Jacobian matrix, we review the Levenberg-Marquardt algorithm which is a regularized Gauss-Newton method and a new iterated gradient method (IGRA). We compare this approach with the Iterative Quadratic Maximum Likelihood (IQML). We propose two further iteration schemes based on simultaneous minimization (SIMI) approach. While being derived from a different model, the scheme SIMI-I appears to be equivalent to the Gradient Condition Reweighted Algorithm (GRA) by Osborne and Smyth. The second scheme SIMI-2 is more stable with regard to the choice of the initial vector. For parameter identification, we 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

In this paper, we are interested in the following problem. For a given vector of data $\mathbf{y} = (y_k)_{k=0}^{L}$ with $L \geq 2M$ we want to compute all parameters $d_j, z_j \in \mathbb{C}$ such that

$$\Big\|\mathbf{y} - \Big(\sum_{j=1}^{M} d_j z_j^k\Big)_{k=0}^{L}\Big\|_2 \tag{1.1}$$

*School of Mathematics, Shanghai University of Finance and Economics, Guoding Road 777, 200433, Shanghai, P.R. China. Email: zhang.ran@mail.shufe.edu.cn

†Institute for Numerical and Applied Mathematics, Göttingen University, Lotzestr. 16-18, 37083 Göttingen, Germany. Email: plonka@math.uni-goettingen.de, phone: + 49 551 39 22115

is minimized. Problem (1.1) occurs in two different scenarios. For parameter estimation problems, we assume that the given data are of the form $y_k = f(kh) + \epsilon_k$, where the signal $f(x)$ is an exponential sum

$$f(x) = \sum_{j=1}^{M} d_j \, e^{T_j x} \tag{1.2}$$

with unknown $d_j \in \mathbb{C} \setminus \{0\}$, $z_j := e^{T_j h}$, and $T_j \in \mathbb{C}$, $\mathrm{Im} T_j \in [-\pi/h, \pi/h)$, $j = 1, \ldots, M$. Further, we assume that ϵ_k are i.i.d. random variables with mean value zero and variance σ^2. In this case a statistical interpretation as a maximum likelihood method is possible. In the second scenario for sparse signal approximation problems, we want to approximate the vector $\mathbf{y}$ by a new vector whose components are exponential sums such that the error is minimized in the Euclidean norm.

One reason for the strong interest in signal approximation by exponential sums is the wide field of applications. Examples are synchrophasor estimation [36], estimation of mean curve lightning impulses [13], parameter estimation in electrical power systems [23], 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 reconstruction of signals with finite rate of innovation [12]. For a survey on relations of exponential analysis to annihilating filters, rational approximation and linear prediction we refer to our paper [30]. A further important application is the use of exponential sums in quadrature formulas for higher-dimensional integrals, see [9].

If $f(x)$ indeed possesses the exact structure in (1.2), then the parameters d_j, z_j can be computed by a Prony-like method from equidistant samples $f(kh)$, $k = 0, \ldots, 2M - 1$. 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 [35], estimation of signal parameters via rotational invariance techniques (ESPRIT) by Roy and Kailath [34], the matrix pencil method by Hua and Sakar [16] and the approximate Prony method (APM) by Potts and Tasche [32]. The paper [33] contains a summary of all these algorithms and also studies their close relations.

However, these numerical schemes do not solve the minimization problem (1.1) but assume that the given data are exactly of the form $y_k = f(kh)$ with f in (1.2). In the noisy case, these methods are not consistent for $L \to \infty$, see [18].

Contributions of the paper and related work

In this paper, we will employ a direct approach to tackle problem (1.1) based on former ideas on maximum likelihood modifications of Prony's method, see [10, 24, 25, 26]. First, we derive an equivalent formulation of (1.1) as a non-convex minimization problem, similarly as in [10, 24]. For that purpose we use variable projection in a first step in order to transfer (1.1) to a minimization problem with regard to the parameter vector $\mathbf{z} = (z_1, \ldots, z_M)^T$. In a second step, we rewrite the problem as a minimization problem with regard to $\mathbf{p}$, where $\mathbf{p} = (p_0, \ldots, p_M)^T$ is related to $\mathbf{z}$ by $\prod_{j=1}^{M}(z - z_j) = \sum_{k=0}^{M} p_k z^k$, see also [10, 24].

In Section 3, we survey some numerical algorithms to solve the obtained minimization problem. For this purpose, we derive an explicit form of the Jacobian matrix and of the gradient of the functional. These observations lead to simple presentations of

Gauß-Newton and Levenberg-Marquardt iteration schemes on the one hand and algorithms for the representation as a nonlinear eigenvalue problem on the other hand. Using the necessary condition for the gradient of the functional, we propose an iterative algorithm IGRA that is close in nature (but not equivalent) to the Gradient Condition Reweighting Algorithms (GRA) by Osborne and Smyth [26]. We also review the iterative quadratic maximum likelihood (IQML) algorithm in [10, 20, 11].

In Section 4, we derive a new iteration functional based on the minimization problem considered in Section 2. We can show that the desired solution vector is a fixed point of the obtained iteration scheme and that the corresponding iteration functional value always converges. The corresponding necessary condition for the gradient of the functional leads to the iteration scheme SIMI-1 which appears to be exactly equivalent to GRA for the so-called recurrence model in [25, 26]. We refer to [25, 26] for further results on asymptotic stability and local convergence of this iteration under special conditions. The second iteration scheme SIMI-2 uses a slightly different approximation, which results in the problem of finding an eigenvector to the smallest eigenvalue of a positive definite matrix at each iteration. We finally give a factorization of the matrices that are involved in the iteration schemes in order to ensure efficient calculation of the iteration matrices using the fast Fourier transform.

Our numerical experiments in Section 5 show for different examples that all compared iteration methods converge very fast and provide similar errors, while the found parameter vectors can be quite different. For parameter estimation, our numerical experiments show that the pre-filtering step is crucial for higher noise levels in order to obtain good parameter estimates. Furthermore, the pre-filtering step strongly reduces the computational effort for all considered iteration methods.

Finally, we would like to mention some further related work. For the special case when $d_j \in \mathbb{R}$ and $|z_j| = 1$ for $j = 1, \ldots, M$, iterative approaches have been proposed to solve (1.1) that try to improve the estimate of z_j directly at each iteration step, [4, 7]. For an approach where in (1.1) the 2-norm is replaced by the 1-norm we refer to [37].

The considered problem (1.1) can also be rewritten as a nonlinear structured least squares problem (NSLRA), [21, 39], see our remarks at the end of Section 2.

Further, (1.1) is related to the problem of low-rank approximation of Hankel matrices. Taking $f_k = \sum_{j=1}^{M} d_j z_j^k$ for $k = 0, \ldots, L$, one may consider instead of $\|\mathbf{y} - \mathbf{f}\|_2$ the spectral norm $\|\mathbf{H}_\mathbf{y} - \mathbf{H}_\mathbf{f}\|$, where $\mathbf{H}_\mathbf{y}$ and $\mathbf{H}_\mathbf{f}$ are Hankel matrices generated by $\mathbf{y}$ and $\mathbf{f}$. The special structure of $\mathbf{f}$ then implies that $\mathbf{H}_\mathbf{f}$ has only rank M. Thus we arrive at the problem of best low-rank approximation with Hankel structure, see [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 [5, 6, 2, 29]. However, we emphasize that these methods do not exactly solve the problem (1.1) but only a related approximation problem.

2 Maximum likelihood modification of Prony's method

Let $\mathbf{y} = (y_k)_{k=0}^{L} \in \mathbb{C}^{L+1}$ be a given sequence. Our goal is to approximate $\mathbf{y}$ by a sequence $\mathbf{f} = (f_k)_{k=0}^{L} \in \mathbb{C}^{L+1}$ generated by an exponential sum with $M \leq \frac{L}{2}$ terms of the form

$$f_k = \sum_{j=1}^{M} d_j z_j^k, \quad k = 0, \ldots, L, \tag{2.1}$$

with $d_j, z_j \in \mathbb{C}$, $j = 1, \ldots, M$, such that

$$\|\mathbf{y} - \mathbf{f}\|_2^2 = \sum_{k=0}^{L} |y_k - f_k|^2$$

is minimal. With $\mathbf{d} := (d_1, \ldots, d_M)^T$, $\mathbf{z} := (z_1, \ldots, z_M)^T$, and the Vandermonde matrix

$$\mathbf{V_z} := \begin{pmatrix} 1 & 1 & \ldots & 1 \\ z_1 & z_2 & \ldots & z_M \\ z_1^2 & z_2^2 & \ldots & z_M^2 \\ \vdots & \vdots & & \vdots \\ z_1^L & z_2^L & \ldots & z_M^L \end{pmatrix} \in \mathbb{C}^{(L+1)\times M}, \tag{2.2}$$

we can write

$$\mathbf{f} = \mathbf{V_z}\,\mathbf{d}. \tag{2.3}$$

Thus, the problem can be formulated as follows. For given $\mathbf{y} \in \mathbb{C}^{L+1}$ we want to solve the nonlinear least squares problem

$$\operatorname*{argmin}_{\mathbf{z},\mathbf{d}\in\mathbb{C}^M} \|\mathbf{y} - \mathbf{V_z}\mathbf{d}\|_2^2 = \operatorname*{argmin}_{\mathbf{z},\mathbf{d}\in\mathbb{C}^M} \sum_{k=0}^{L} |y_k - \sum_{j=1}^{M} d_j z_j^k|^2. \tag{2.4}$$

Throughout the paper we will assume that $\mathbf{V_z}$ has full rank M, i.e., we assume that z_j are pairwise distinct, and $\mathbf{y}$ cannot be exactly recovered by an exponential sum with less than M terms. In particular, we assume that all coefficients d_j are nonzero. If some further 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 suitable subsets in the minimization process.

Following the arguments in [10, 24, 25, 26], we observe that for given $\mathbf{z}$, the minimization problem turns into a linear least squares problem

$$\operatorname*{argmin}_{\mathbf{d}\in\mathbb{C}^M} \|\mathbf{y} - \mathbf{V_z}\mathbf{d}\|_2^2$$

with the solution

$$\mathbf{d} = \mathbf{V_z^+}\,\mathbf{y} = [\mathbf{V_z^*}\mathbf{V_z}]^{-1}\mathbf{V_z^*}\mathbf{y}, \tag{2.5}$$

where $\mathbf{V_z^+}$ denotes the Moore-Penrose inverse of $\mathbf{V_z}$ with full rank M. We introduce the projection matrix

$$\mathbf{P_z} := \mathbf{V_z}\,\mathbf{V_z^+}$$

satisfying the properties

$$\mathbf{P_z} = \mathbf{P_z^*}, \quad \mathbf{P_z^2} = \mathbf{P_z}, \quad \mathbf{P_z}\mathbf{V_z} = \mathbf{V_z}.$$

Then (2.4) can be rewritten as

$$\begin{aligned} \operatorname*{argmin}_{\mathbf{z}\in\mathbb{C}^M} \|\mathbf{y} - \mathbf{V_z}\mathbf{V_z^+}\,\mathbf{y}\|_2^2 &= \operatorname*{argmin}_{\mathbf{z}\in\mathbb{C}^M} \|(\mathbf{I} - \mathbf{P_z})\,\mathbf{y}\|_2^2 \\ &= \operatorname*{argmin}_{\mathbf{z}\in\mathbb{C}^M} \mathbf{y}^*(\mathbf{I} - \mathbf{P_z})^*(\mathbf{I} - \mathbf{P_z})\mathbf{y} \\ &= \operatorname*{argmin}_{\mathbf{z}\in\mathbb{C}^M} (\mathbf{y}^*\mathbf{y} - \mathbf{y}^*\mathbf{P_z}\mathbf{y}). \end{aligned}$$

Thus, we need to solve

$$\widetilde{\mathbf{z}} := \underset{\mathbf{z}\in\mathbb{C}^M}{\operatorname{argmax}} \left(\mathbf{y}^*\mathbf{P}_{\mathbf{z}}\mathbf{y}\right) \tag{2.6}$$

in order to find the optimal parameter vector $\tilde{\mathbf{z}}$. Afterwards, we can compute $\mathbf{d}$ simply by (2.5).

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 $p(z) = c\prod_{j=1}^M (z - z_j) = \sum_{k=0}^M p_k z^k$, where the constant c is chosen such that the arising vector $\mathbf{p} := (p_0, \ldots, p_M)^T$ of coefficients satisfies $\|\mathbf{p}\|_2 = 1$. For this purpose we introduce the two matrices $\mathbf{X}_{\mathbf{p}} \in \mathbb{C}^{(L+1)\times(L-M+1)}$ and $\mathbf{H}_{\mathbf{y}} \in \mathbb{C}^{(L-M+1)\times(M+1)}$ of the form

$$\mathbf{X}_{\mathbf{p}} := \begin{pmatrix} p_0 & & & \\ p_1 & p_0 & & \\ \vdots & p_1 & \ddots & \\ & \vdots & & p_0 \\ p_M & & & p_1 \\ & p_M & & \vdots \\ & & \ddots & \\ & & & p_M \end{pmatrix}, \quad \mathbf{H}_{\mathbf{y}} := \begin{pmatrix} y_0 & y_1 & \cdots & y_M \\ y_1 & y_2 & \cdots & y_{M+1} \\ \vdots & \vdots & & \vdots \\ y_{L-M} & y_{L-M+1} & \cdots & y_L \end{pmatrix}. \tag{2.7}$$

Then we have

$$\mathbf{H}_{\mathbf{y}}\mathbf{p} = \mathbf{X}_{\mathbf{p}}^T\mathbf{y} \in \mathbb{C}^{L-M+1}. \tag{2.8}$$

Moreover, let

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

be the corresponding projection matrix.

Theorem 2.1 *For given data $\mathbf{y} = (y_0, \ldots, y_L)^T$ the parameter vectors $\widetilde{\mathbf{z}}$ and $\widetilde{\mathbf{d}}$ 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

$$\widetilde{\mathbf{p}} = \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmin}} \mathbf{y}^*\overline{\mathbf{P}}_{\mathbf{p}}\mathbf{y} = \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmin}} \mathbf{p}^*\mathbf{H}_{\mathbf{y}}^*[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p}. \tag{2.9}$$

2. Compute the vector of zeros $\widetilde{\mathbf{z}} = (\widetilde{z}_1, \ldots, \widetilde{z}_M)^T$ of the polynomial $p(z) = \sum\limits_{k=0}^{M} \tilde{p}_k z^k$ obtained from $\tilde{\mathbf{p}} = (\tilde{p}_0, \ldots, \tilde{p}_M)^T$.
3. Compute

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

Proof: We follow the ideas in [10, 24, 25, 26] 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 coefficient vector of the corresponding Prony polynomial $p(z) = c\prod_{j=1}^{M}(z - z_j) = \sum\limits_{k=0}^{M} p_k z^k$ where c is taken such that $\|\mathbf{p}\|_2 = 1$. Now, we observe that the matrices $\mathbf{X_p}$ in (2.7) and $\mathbf{V_z}$ in (2.2) satisfy

$$\mathbf{X}_{\mathbf{p}}^T \mathbf{V}_{\mathbf{z}} = \mathbf{0}$$

and thus $\overline{\mathbf{P}}_{\mathbf{p}}\mathbf{P}_{\mathbf{z}} = \mathbf{0}$. Note that $\mathrm{rank}(\mathbf{X_p}) = \mathrm{rank}(\overline{\mathbf{P}}_{\mathbf{p}}) = L + 1 - M$ and $\mathrm{rank}(\mathbf{V_z}) = \mathrm{rank}(\mathbf{P_z}) = M$. Thus, we conclude

$$\mathbf{P}_{\mathbf{z}} = (\mathbf{I} - \overline{\mathbf{P}}_{\mathbf{p}}) \tag{2.10}$$

i.e., solving the maximization problem in (2.6) is equivalent with solving the minimization problem

$$\widetilde{\mathbf{p}} := \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1} \\ \|\mathbf{p}\|_2=1}}{\mathrm{argmin}}\, \mathbf{y}^* \overline{\mathbf{P}}_{\mathbf{p}} \mathbf{y},$$

and extracting the vector $\widetilde{\mathbf{z}}$ of zeros of $\sum_{k=0}^{M} \widetilde{p}_k z^k$. The second representation in (2.9) is due to $\mathbf{X}_{\mathbf{p}}^T\mathbf{y} = \mathbf{H_y}\mathbf{p}$. The remaining computation of $\widetilde{\mathbf{d}}$ is the same as in (2.5). ∎

In many applications, particularly for parameter identification, it is assumed that the given data satisfy the model (2.1), and the measurements y_k are noisy, i.e., $y_k = f_k + \epsilon_k$, $k = 0, \ldots, L$. Let us assume that ϵ_k are i.i.d. random variables with $\epsilon_k \in N(0, \sigma^2)$ and L is large. If we have $L + 1 = (2M + 1)K$ measurement values y_k, where $K > 1$ is an integer, then we can apply a local low-pass filter to $\mathbf{y}$ in a preprocessing step and obtain a filtered signal $\tilde{\mathbf{y}}$ with a measurement error $\tilde{\epsilon}$ possessing zero expectation and smaller variance. Taking e.g.

$$\widetilde{y}_k := \frac{1}{K}\sum_{r=Kk}^{K(k+1)-1} y_r = \frac{1}{K}\sum_{r=Kk}^{K(k+1)-1} f_r + \frac{1}{K}\sum_{r=Kk}^{K(k+1)-1} \epsilon_r = \widetilde{f}_k + \widetilde{\epsilon}_k, \quad k = 0, \ldots, 2M, \tag{2.11}$$

the new variables $\widetilde{\epsilon}_k$ are linearly independent with mean value zero, and the noise variance is reduced to σ^2/K. The filtered sequence $(\tilde{f}_k)_{k=0}^{2M}$ still satisfies the exponential model (2.1) , but this time with the parameters z_j^K instead of z_j, $j = 1, \ldots, M$, since

$$\tilde{f}_k = \frac{1}{K}\sum_{r=Kk}^{K(k+1)-1} f_r = \frac{1}{K}\sum_{r=0}^{K-1}\sum_{j=1}^{M} d_j z_j^{r+Kk} = \sum_{j=1}^{M} d_j \left(\frac{1 - z_j^K}{1 - z_j}\right) z_j^{Kk} = \sum_{j=1}^{M} \tilde{d}_j (z_j^K)^k,$$

where in the last computation we have assumed that $z_j \neq 1$. We need to ensure here that z_j is not a power of $\mathrm{e}^{2\pi \mathrm{i}/K}$ and that the values z_j^K, $j = 1, \ldots, M$, are still pairwise distinct. Moreover, ambiguities occur if we want to recover z_j from z_j^K. In practice, these ambiguities are resolved using a priori knowledge on the phase range of z_j. Instead of the filter (2.11), we can also use the following filter to find $\tilde{\mathbf{y}}$ of length $2M + 1$, with

$$\widetilde{y}_k = \frac{1}{K}\sum_{r=0}^{K-1} f_{k+(2M+1)r} + \frac{1}{K}\sum_{r=0}^{K-1} \epsilon_{k+(2M+1)r} = \tilde{f}_k + \tilde{\epsilon}_k, \quad k = 0, \ldots, 2M,$$

where the new noise variables $\tilde{\epsilon}_k$ also possess the reduced variance σ^2/K. Here, the filtered sequence $(\tilde{f}_k)_{k=0}^{2M}$ satisfies the exponential model (2.1) with the same parameters z_j, $j = 1, \ldots, M$, since

$$\tilde{f}_k = \frac{1}{K}\sum_{r=0}^{K-1} f_{k+(2M+1)r} = \frac{1}{K}\sum_{r=0}^{K-1}\sum_{j=1}^{M} d_j z_j^{k+(2M+1)r} = \sum_{j=1}^{M}\left(\frac{d_j}{K}\sum_{r=0}^{K-1} z_j^{(2M+1)r}\right) z_j^k.$$

To ensure that $\tilde{d}_j = \frac{d_j(1-z_j^{(2M+1)K})}{K(1-z_j^{(2M+1)})}$ does not vanish, we assume that $z_j^{(2M+1)K} \neq 1$ for $j = 1, \ldots, M$. Then, we can use $\tilde{\mathbf{y}}$ instead of $\mathbf{y}$ to evaluate the parameter vector $\mathbf{z}$, while still applying (2.5) to compute the parameter vector $\mathbf{d}$ in a second step.

Remarks 2.2

1. If the given data $\mathbf{y}$ can be exactly represented by a sum of exponentials, i.e., $\mathbf{y} = \mathbf{f}$ in (2.1)*, or if the errors $\epsilon_k = y_k - f_k$ have very small modulus, then a Prony-like method can be employed to identify the parameter vectors $\mathbf{z}$ and $\mathbf{d}$. In the exact data case it can be shown that the Hankel matrix $\mathbf{H_y}$ in* (2.7) *has rank M and that the vector of coefficients $\mathbf{p} = (p_0, \ldots, p_M)^T$ of the Prony polynomial $p(z) = c\prod_{j=1}^{M}(z - z_j) = \sum_{k=0}^{M} p_k z^k$ is the eigenvector of $\mathbf{H_y}$ to the eigenvalue $\mathbf{0}$. To construct the parameter vectors $\mathbf{z}$ and $\mathbf{d}$ we need to solve the eigenvector problem $\mathbf{H_y p} = \mathbf{0}$ to find $\mathbf{p}$, extract the zeros z_j of the obtained Prony polynomial $p(z)$ and find $\mathbf{d}$ by* (2.5)*. However, this procedure is not numerically stable. Already for small inaccuracies in the data, $\mathbf{H_y}$ has full rank $M+1$. A first simple idea for stabilization is to solve*

$$\widetilde{\mathbf{p}} := \operatorname*{argmin}_{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}} \mathbf{p}^*\mathbf{H}_\mathbf{y}^*\mathbf{H}_\mathbf{y}\mathbf{p}. \tag{2.12}$$

This approach is also known as the Pisarenko method [28]*. To improve the numerical stability of Prony's method, one can e.g. employ ESPRIT* [34] *or the approximate Prony method (AMP), see* [32]*. For a survey on Prony methods we refer to* [30].

2. Compared to the Pisarenko method in (2.12)*, the minimization problem in* (2.9) *contains the further term $[\mathbf{X}_\mathbf{p}^T\overline{\mathbf{X}}_\mathbf{p}]^{-1}$ that makes it non-convex. Theorem* 2.1 *shows that similarly as for the Prony-like methods, the determination of the parameter vectors $\mathbf{z}$ and $\mathbf{d}$ is separated. Formula* (2.9) *can be understood as the variable projection formulation of the Hankel structured low-rank approximation, see e.g.* [14, 21]*. We emphasize that Theorem* 2.1 *can be applied to an arbitrary vector $\mathbf{y}$. In many applications, one assumes that $y_k = f_k + \epsilon_k$ with f_k in* (1.2) *with some prior knowledge on the distribution of the error ϵ_k.*

3. Similar ideas for fitting exponential models have been also given by Kumaresan et al. [19] *and by Hua and Sakar* [16]*, where it has been called whitened TLS-LP method.*

4. We note that the normalization of $\mathbf{p}$ in (2.9) *does not effect the objective function in* (2.9)*, see e.g.* [24]*. Indeed we have $\mathbf{X}_{c\mathbf{p}} = c\mathbf{X}_\mathbf{p}$ and therefore*

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

for all $c \neq 0$. The classical Prony method often uses the normalization $p_M = 1$ instead of $\|\mathbf{p}\|_2 = 1$.

5. The minimization problem in (2.4) *can also be written as the NSLRA problem*

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

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*

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

with $\mathbf{X}_{\mathbf{p}}$ *in* (2.7)*, see e.g.* [39].

6. While the procedure derived in Theorem 2.1 *works for arbitrary data* $\mathbf{y}$*, it can be interpreted also statistically, see* [18]*. Assume that* $y_k = f_k + \epsilon_k$ *where* $\epsilon_k \in N(0, \sigma^2)$ *are i.i.d. Gaussian random 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* $\mathbf{r}$*,*

$$E(\mathbf{r}\mathbf{r}^*) = E(\mathbf{X}_{\mathbf{p}}^T \mathbf{y}\mathbf{y}^* \overline{\mathbf{X}}_{\mathbf{p}}) = E(\mathbf{X}_{\mathbf{p}}^T \boldsymbol{\epsilon}\, \boldsymbol{\epsilon}^* \overline{\mathbf{X}}_{\mathbf{p}}) = \mathbf{X}_{\mathbf{p}}^T E(\boldsymbol{\epsilon}\, \boldsymbol{\epsilon}^*) \overline{\mathbf{X}}_{\mathbf{p}} = \sigma^2 \mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}, \tag{2.13}$$

i.e., the errors r_k *are not longer independent. Therefore, we employ the 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*

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

leads to the Prony modification that we derived in (2.9)*.*

7. Using the method of Lagrangian multipliers, the model in (2.9) *has been derived in* [11] *from the following reformulated problem: For given noisy data* $\mathbf{y}$*, solve*

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

3 Numerical algorithms for the ML-Prony method

In this section we will survey some numerical approaches to solve the nonlinear minimization problem in (2.9). We start with deriving a new representation of the necessary condition for the vector $\widetilde{\mathbf{p}}$ in (2.9). Similar conditions have been also found in different forms in earlier papers in the case of real data $\mathbf{y} \in \mathbb{R}^{L+1}$, see e.g. [24, 25, 26], without giving direct matrix representations of the Jacobian and the gradient. Let for $\mathbf{p} \in \mathbb{C}^{M+1}$ with $\|\mathbf{p}\|_2 = 1$,

$$G(\mathbf{p}) := \mathbf{y}^* \overline{\mathbf{P}}_{\mathbf{p}} \mathbf{y} = \mathbf{y}^* \overline{\mathbf{X}}_{\mathbf{p}} [\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}]^{-1} \mathbf{X}_{\mathbf{p}}^T \mathbf{y} = \|\mathbf{r}(\mathbf{p})\|_2^2 \tag{3.1}$$

with $\mathbf{r}(\mathbf{p}) := \overline{\mathbf{P}}_{\mathbf{p}} \mathbf{y}$, where $\overline{\mathbf{P}}_{\mathbf{p}} = (\mathbf{X}_{\mathbf{p}}^+)^T \mathbf{X}_{\mathbf{p}}^T = \overline{\mathbf{X}}_{\mathbf{p}} \overline{\mathbf{X}}_{\mathbf{p}}^+ \in \mathbb{C}^{(L+1)\times(L+1)}$. Then (2.9) takes the form $\min_{\mathbf{p} \in \mathbb{C}^{M+1}, \|\mathbf{p}\|_2 = 1} G(\mathbf{p})$. We can derive now the Jacobian of $\mathbf{r}(\mathbf{p})$ as follows.

Theorem 3.1 *Let* $\mathbf{p} = \mathbf{a} + \mathrm{i}\mathbf{b}$ *with* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{M+1}$*, and let for* $\breve{\mathbf{p}} = (\mathbf{a}^T, \mathbf{b}^T)^T \in \mathbb{R}^{2M+2}$

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

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*

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

where $\mathbf{I}_{L+1}$ *and* $\mathbf{I}_{M+1}$ *denote the identity matrices of given size,* $\mathbf{v}(\mathbf{p}) := \overline{\mathbf{X}}_{\mathbf{p}}^{+} \mathbf{y} \in \mathbb{C}^{L-M+1}$ *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* (3.1) *reads*

$$\nabla G(\breve{\mathbf{p}}) = 2\, \mathbf{J}(\mathbf{a},\mathbf{b})^* \mathbf{r}(\mathbf{p}) = 2 \begin{pmatrix} \mathbf{I}_{M+1} \\ -\mathrm{i}\mathbf{I}_{M+1} \end{pmatrix} \mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p})}^* [\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}]^{-1} \mathbf{H}_{\mathbf{y}} \mathbf{p}. \tag{3.2}$$

Further, we obtain

$$\begin{aligned} \mathbf{J}(\breve{\mathbf{p}})^* \mathbf{J}(\breve{\mathbf{p}}) \; = \; & \begin{pmatrix} \mathbf{I}_{M+1} \\ \mathrm{i}\mathbf{I}_{M+1} \end{pmatrix} \mathbf{X}_{\mathbf{v}(\mathbf{p})}^* (\mathbf{I}_{L+1} - \overline{\mathbf{P}}_{\mathbf{p}}) \mathbf{X}_{\mathbf{v}(\mathbf{p})} \left(\mathbf{I}_{M+1}, \, -\mathrm{i}\mathbf{I}_{M+1} \right) \\ & + \begin{pmatrix} \mathbf{I}_{M+1} \\ -\mathrm{i}\mathbf{I}_{M+1} \end{pmatrix} \mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p})}^* [\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}]^{-1} \mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p})} \left(\mathbf{I}_{M+1}, \, \mathrm{i}\mathbf{I}_{M+1} \right). \end{aligned} \tag{3.3}$$

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

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

and where $\mathbf{0}$ and $\mathbf{I}$ denote zero matrices and the identity matrix of given size. With $\mathbf{v}(\mathbf{p}) = \overline{\mathbf{X}}_{\mathbf{p}}^{+} \mathbf{y}$ 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 of length $M+1$ for $k = 0, 1, ..., M$, i.e., $\mathbf{e}_k := (\delta_{j,k})_{j=0}^{M}$. 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 $k = 0, \ldots, M$ 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 3.2 *A normalized vector $\mathbf{p} \in \mathbb{C}^{M+1}$ that minimizes $G(\mathbf{p})$ in* (3.1) *necessarily satisfies the eigenvector equation*

$$\mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p})}^* [\mathbf{X}_\mathbf{p}^T \overline{\mathbf{X}}_\mathbf{p}]^{-1} \mathbf{H}_\mathbf{y} \mathbf{p} = \left(\mathbf{H}_\mathbf{y}^* [\mathbf{X}_\mathbf{p}^T \overline{\mathbf{X}}_\mathbf{p}]^{-1} \mathbf{H}_\mathbf{y} - \mathbf{H}_{\mathbf{r}(\mathbf{p})}^* [\mathbf{X}_\mathbf{p}^T \overline{\mathbf{X}}_\mathbf{p}]^{-1} \mathbf{H}_{\mathbf{r}(\mathbf{p})} \right) \mathbf{p} = \mathbf{0}. \tag{3.4}$$

Proof: If $\mathbf{p}$ with $\|\mathbf{p}\|_2 = 1$ minimizes $G(\mathbf{p})$, then $\nabla G(\mathbf{p}) = 0$. The assertion directly follows from (3.2) 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 review the algorithms to solve $\min\limits_{\mathbf{p} \in \mathbb{C}^{M+1}, \|\mathbf{p}\|_2 = 1} G(\mathbf{p})$ with $G(\mathbf{p}) = \|\mathbf{r}(\mathbf{p}\|_2$ in (3.1). All considered algorithms are iterative and aim at successive improvement of the coefficient vector $\mathbf{p}$. As a suitable initial vector one can use

$$\mathbf{p}_0 := \operatorname*{argmin}_{\substack{\mathbf{p} \in \mathbb{C}^{M+1} \\ \|\mathbf{p}\|_2 = 1}} \mathbf{p}^* \mathbf{H}_\mathbf{y}^* \mathbf{H}_\mathbf{y} \mathbf{p}. \tag{3.5}$$

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 (2.12). Since $\mathbf{y}$ is noisy, the obtained smallest singular value is usually nonzero.

All iteration algorithms that we investigate in this section and in the next section can also be applied using the pre-smoothed data vector $\widetilde{\mathbf{y}} \in \mathbb{C}^{2M+1}$ in (2.11) instead of $\mathbf{y}$.

3.1 Gauß-Newton and Levenberg-Marquardt iteration

We approximate $\mathbf{r}(\mathbf{p} + \boldsymbol{\delta})$ with $\mathbf{r}(\mathbf{p}) = \overline{\mathbf{P}}_\mathbf{p}\mathbf{y}$ in (3.1) by using 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 $\mathbf{r}(\breve{\mathbf{p}}) + \mathbf{J}(\breve{\mathbf{p}})\breve{\boldsymbol{\delta}}$ is the first order approximation of $\mathbf{r}(\breve{\mathbf{p}} + \breve{\boldsymbol{\delta}})$, where $\mathbf{J}(\breve{\mathbf{p}}) = \mathbf{J}(\mathbf{a}, \mathbf{b})$ and $\mathbf{r}(\breve{\mathbf{p}}) = \mathbf{r}(\mathbf{p})$. We compute

$$G(\breve{\mathbf{p}} + \breve{\boldsymbol{\delta}}) \approx (\mathbf{r}(\breve{\mathbf{p}}) + \mathbf{J}(\breve{\mathbf{p}})\breve{\boldsymbol{\delta}})^* (\mathbf{r}(\breve{\mathbf{p}}) + \mathbf{J}(\breve{\mathbf{p}})\breve{\boldsymbol{\delta}}).$$

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

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

The corresponding Gauss-Newton iteration leads in our case at the jth step to the system

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

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 3.1. 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 with $\mathbf{v}(\mathbf{p}_j) = \overline{\mathbf{X}}_{\mathbf{p}_j}^+ \mathbf{y} = \mathbf{X}_{\mathbf{p}_j}^+ \mathbf{y}$, we have from (3.3)

$$\begin{aligned} \mathbf{J}(\breve{\mathbf{p}}_j)^*\mathbf{J}(\breve{\mathbf{p}}_j)\breve{\mathbf{p}}_j &= \mathbf{J}(\mathbf{p}_j,\mathbf{0})^T\mathbf{J}(\mathbf{p}_j,\mathbf{0})\breve{\mathbf{p}}_j \\ = \; &\left(\begin{pmatrix} \mathbf{I}_{M+1} \\ \mathrm{i}\,\mathbf{I}_{M+1} \end{pmatrix} \mathbf{X}^*_{\mathbf{v}(\mathbf{p}_j)}(\mathbf{I}_{L+1} - \mathbf{P}_{\mathbf{p}_j})\mathbf{X}_{\mathbf{v}(\mathbf{p}_j)} \right. \\ &\left. + \begin{pmatrix} \mathbf{I}_{M+1} \\ -\mathrm{i}\,\mathbf{I}_{M+1} \end{pmatrix} \mathbf{H}^*_{\mathbf{y}-\mathbf{r}(\mathbf{p}_j)}[\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\mathbf{H}_{\mathbf{y}-\mathbf{r}(\mathbf{p}_j)} \right) \mathbf{p}_j = \mathbf{0} \end{aligned} \tag{3.6}$$

since

$$\mathbf{X}^*_{\mathbf{v}(\mathbf{p}_j)}(\mathbf{I}_{L+1} - \mathbf{P}_{\mathbf{p}_j})\mathbf{X}_{\mathbf{v}(\mathbf{p}_j)}\mathbf{p}_j = \mathbf{X}^*_{\mathbf{v}(\mathbf{p}_j)}(\mathbf{I}_{L+1} - \mathbf{P}_{\mathbf{p}_j})\mathbf{X}_{\mathbf{p}_j}\mathbf{v}(\mathbf{p}_j) = \mathbf{0}$$

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

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

$$\begin{aligned} \left(\mathbf{J}(\breve{\mathbf{p}}_j)^*\mathbf{J}(\breve{\mathbf{p}}_j) + \lambda_j\mathbf{I}\right)\breve{\boldsymbol{\delta}}_j &= -\mathrm{Re}\left(\mathbf{J}(\breve{\mathbf{p}}_j)^*\mathbf{r}(\breve{\mathbf{p}}_j)\right), \\ \breve{\mathbf{p}}_{j+1} &= \breve{\mathbf{p}}_j + \breve{\boldsymbol{\delta}}_j. \end{aligned}$$

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 (3.5). For existing software packages to implement this method we refer to [22].

3.2 Algorithms for the nonlinear eigenvector problem

We consider the necessary condition (3.4) of the form

$$\mathbf{H}^*_{\mathbf{y}-\mathbf{r}(\mathbf{p})}[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p} = \mathbf{0}$$

as a nonlinear eigenvalue problem.

Iterative Gradient Algorithm (IGRA). We denote

$$\mathbf{C}_{\mathbf{p}} := \mathbf{H}^*_{\mathbf{y}-\mathbf{r}(\mathbf{p})}[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{H}_{\mathbf{y}} = \mathbf{H}^*_{\mathbf{y}}[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{H}_{\mathbf{y}} - \mathbf{H}^*_{\mathbf{r}(\mathbf{p})}[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{H}_{\mathbf{r}(\mathbf{p})},$$

then (3.4) can be written as $\mathbf{C_p p} = \mathbf{0}$. Using this new representation of the gradient $\nabla G(\mathbf{p})$ in (3.2) we propose the iteration scheme

$$\begin{aligned} \left(\mathbf{C}_{\mathbf{p}_j} - \mu_j \mathbf{I}\right) \mathbf{p}_{j+1} = \Big(\mathbf{H}^*_{\mathbf{y}-\mathbf{r}(\mathbf{p}_j)} [\mathbf{X}^T_{\mathbf{p}_j} \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \mathbf{H}_{\mathbf{y}} - \mu_j \mathbf{I}\Big) \mathbf{p}_{j+1} = \mathbf{0}, \\ \mathbf{p}^*_{j+1} \mathbf{p}_{j+1} = 1. \end{aligned} \tag{3.7}$$

Here, at each iteration step the matrix $\mathbf{C_p}$ is approximated by $\mathbf{C}_{\mathbf{p}_j}$, where the vector $\mathbf{p}_j$ is found from the previous iteration. An initial vector $\mathbf{p}_0$ can be taken as in (3.5). At the j-th step, inverse iteration is applied to compute the eigenvector $\mathbf{p}_{j+1}$ of $\mathbf{C}_{\mathbf{p}_j}$ corresponding to the smallest eigenvalue by modulus μ_j. The algorithm stops if μ_j is small enough compared to $\|\mathbf{C}_{\mathbf{p}_j}\|$. The complete algorithm reads as follows.

Algorithm 3.3 (IGRA)
Input: M, y_k, $k = 0, \ldots, L$, *with* $L \geq 2M$.

1. *Initialization*
   - *Optional: Compute* $\widetilde{\mathbf{y}}$ *in* (2.11) *and replace in all further steps* $\mathbf{y}$ *by* $\widetilde{\mathbf{y}}$.
   - *Compute* $\mathbf{p}_0$ *in* (3.5).
2. *Iteration: For* $j = 0 \ldots$ *till convergence*
   - *Compute* $\mathbf{p}_{j+1}$ *according to* (3.7), *i.e., compute the eigenvector* $\mathbf{p}_{j+1}$ *of* $\mathbf{C}_{\mathbf{p}_j}$ *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*

$$\mathbf{V_z d} = \mathbf{y}.$$

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

Remark 3.4 *1. It can be simply observed that the desired solution vector* $\tilde{\mathbf{p}}$ *in* (2.9) *is a fixed point of the iteration* (3.7), *i.e., from* $\mathbf{p}_j = \tilde{\mathbf{p}}$ *if follows* $\mathbf{p}_{j+1} = \tilde{\mathbf{p}}$, *where in particular* $\mu_j = 0$ *by* (3.4).

2. In the scheme (3.7), $\mathbf{C}_{\mathbf{p}_j}$ *is the difference of the two Hermitian 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* $\mathbf{H}^*_{\mathbf{r}(\mathbf{p}_j)} [\mathbf{X}^T_{\mathbf{p}_j} \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \mathbf{H}_{\mathbf{r}(\mathbf{p}_j)}$. *The eigenvalues of this matrix difference are all real and lie in an interval bounded by* $\min_{\|\mathbf{x}\|_2=1} \mathbf{x}^* \mathbf{C}_{\mathbf{p}_j} \mathbf{x}$ *and* $\max_{\|\mathbf{x}\|_2=1} \mathbf{x}^* \mathbf{C}_{\mathbf{p}_j} \mathbf{x}$. *This interval always contains the value* 0 *since we have*

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

and thus $\mathbf{p}^*_j \mathbf{C}_{\mathbf{p}_j} \mathbf{p}_j = 0$.

3. Osborne and Smyth [24, 25, 26] *considered a similar algorithm called Gradient Condition Reweighting Algorithm (GRA) for real data. They employed the assumption* *that the given data are of the form $y_k = f_k + \epsilon_k$, where the errors ϵ_k are independent and with mean zero and variance σ^2. The algorithm considered in* [24] *for exponential data is close to the algorithm above in spirit but slightly differs with regard to the second matrix $\mathbf{H}^*_{\mathbf{r}(\mathbf{p}_j)}[\mathbf{X}^T_{\mathbf{p}_j}\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\mathbf{H}_{\mathbf{r}(\mathbf{p}_j)}$. Instead, for GRA the second matrix is of the form $\mathbf{X}^T_{\mathbf{v}_j}\overline{\mathbf{X}}_{\mathbf{v}_j}$ with $\mathbf{v}_j = \overline{\mathbf{X}}^+_{\mathbf{p}_j}\mathbf{y}$, see also Algorithm SIMI-1 in the next section.*

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

$$\mathbf{p}_{j+1} = \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|=1}}{\operatorname{argmin}}\, \mathbf{p}^*\mathbf{H}^*_{\mathbf{y}}[\mathbf{X}^T_{\mathbf{p}_j}\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p}, \tag{3.8}$$

is proposed. Compared to the representation of the gradient in Theorem 3.1 and to the IGRA iteration in (3.7) the IQML iteration just does not take the second term $\mathbf{H}^*_{\mathbf{r}(\mathbf{p})}[\mathbf{X}^T_{\mathbf{p}}\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p}$ into account.

This iteration works well in practice, see Algorithm 3.5. 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 3.5 (IQML)
Input: M, y_k, $k = 0, \ldots, L$, with $L \geq 2M$.

1. *Initialization*
   - *Optional: Compute $\widetilde{\mathbf{y}}$ in* (2.11) *and replace in all further steps $\mathbf{y}$ by $\widetilde{\mathbf{y}}$.*
   - *Compute $\mathbf{p}_0$ in* (3.5).
2. *Iteration: For $j = 0 \ldots$ till convergence*
   - *Compute $\mathbf{p}_{j+1}$ according to* (3.8)*, i.e., compute the right-singular vector $\mathbf{p}_{j+1}$ of $[\mathbf{X}^T_{\mathbf{p}_j}\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1/2}\mathbf{H}_{\mathbf{y}}$ 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*

$$\mathbf{V}_{\mathbf{z}}\mathbf{d} = \mathbf{y}.$$

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

4 New iteration schemes based on simultaneous minimization

Based on the ideas of Osborne and Smyth, 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

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

Then, (2.9) 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

$$\mathbf{p}_{j+1} := \underset{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}}{\operatorname{argmin}} \left(\mathbf{y}^*\mathbf{A}(\mathbf{p}_j,\mathbf{p})\mathbf{y} + \mathbf{y}^*\mathbf{A}(\mathbf{p},\mathbf{p}_j)\mathbf{y}\right), \tag{4.1}$$

and denote by

$$\begin{aligned} F(\mathbf{p}_{j+1},\mathbf{p}_j) &:= \mathbf{y}^*\mathbf{A}(\mathbf{p}_j,\mathbf{p}_{j+1})\mathbf{y} + \mathbf{y}^*\mathbf{A}(\mathbf{p}_{j+1},\mathbf{p}_j)\mathbf{y} \\ &= (\mathbf{p}_j)^*\mathbf{H}_{\mathbf{y}}^*[\mathbf{X}_{\mathbf{p}_{j+1}}^T\overline{\mathbf{X}}_{\mathbf{p}_{j+1}}]^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p}_j + (\mathbf{p}_{j+1})^*\mathbf{H}_{\mathbf{y}}^*[\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p}_{j+1} \end{aligned} \tag{4.2}$$

the obtained functional value. The iteration schemes based on (4.1) will be shortly called *simultaneous minimization schemes* (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 3.2.

Theorem 4.1 *Let $\mathbf{y} = (y_k)_{k=0}^L$ be given with $2M \le L$. Then, the vector $\mathbf{p}_{j+1}$ computed in (4.1) necessarily satisfies the eigenvector equation*

$$\left(\mathbf{H}_{\mathbf{y}}^*[\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\mathbf{H}_{\mathbf{y}} - \mathbf{X}_{\mathbf{w}_j}^T\overline{\mathbf{X}}_{\mathbf{w}_j}\right)\mathbf{p}_{j+1} = \mathbf{0}, \tag{4.3}$$

where $\mathbf{X}_{\mathbf{w}_j}$ is generated as in (2.7) with $\mathbf{w}_j$, where $\mathbf{w}_j := [\mathbf{X}_{\mathbf{p}_{j+1}}^T\overline{\mathbf{X}}_{\mathbf{p}_{j+1}}]^{-1}\mathbf{X}_{\mathbf{p}_j}^T\mathbf{y}$.

Proof: The proof is similar to that of Theorem 3.1 and Corollary 3.2. 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 (4.1) necessarily that $\nabla_{\breve{\mathbf{p}}}F(\mathbf{p},\mathbf{p}_j) = \mathbf{0}$ for $\mathbf{p} = \mathbf{p}_{j+1}$. As before, we employ the conditions

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

With $\mathbf{w} := [\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{X}_{\mathbf{p}_j}^T\mathbf{y}$, $\mathbf{X}_k\mathbf{y} = \mathbf{H}_{\mathbf{y}}\mathbf{e}_k$, and $\mathbf{X}_k\mathbf{w} = \mathbf{X}_{\mathbf{w}}\mathbf{e}_k$, where $\mathbf{e}_k \in \mathbb{C}^{M+1}$ denotes again the kth unit vector for $k = 0,\ldots,M$, we obtain

$$\begin{aligned} \frac{\partial F(\mathbf{p},\mathbf{p}_j)}{\partial a_k} &= \frac{\partial}{\partial a_k}\left[\mathbf{p}_j^*\mathbf{H}_{\mathbf{y}}^*[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p}_j + \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}\right] \\ &= \frac{\partial}{\partial a_k}\left[\mathbf{y}^*\overline{\mathbf{X}}_{\mathbf{p}_j}[\mathbf{X}_{\mathbf{p}}^T\overline{\mathbf{X}}_{\mathbf{p}}]^{-1}\mathbf{X}_{\mathbf{p}_j}^T\mathbf{y} + \mathbf{y}^*\overline{\mathbf{X}}_{\mathbf{p}}[\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\mathbf{X}_{\mathbf{p}}^T\mathbf{y}\right] \\ &= -\mathbf{y}^*\overline{\mathbf{X}}_{\mathbf{p}_j}[\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}_j}^T\mathbf{y} \\ &\quad +\mathbf{y}^*\mathbf{X}_k[\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\mathbf{X}_{\mathbf{p}}^T\mathbf{y} + \mathbf{y}^*\overline{\mathbf{X}}_{\mathbf{p}}[\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\mathbf{X}_k^T\mathbf{y} \\ &= -\mathbf{w}^*\mathbf{X}_k^T\overline{\mathbf{X}}_{\mathbf{p}}\mathbf{w} - \mathbf{w}^*\mathbf{X}_{\mathbf{p}}^T\mathbf{X}_k\mathbf{w} \\ &\quad +\mathbf{e}_k^T\mathbf{H}_{\mathbf{y}}^*[\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p} + \mathbf{p}^*\mathbf{H}_{\mathbf{y}}^*[\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{e}_k \\ &= 2\mathrm{Re}\left(-\mathbf{e}_k^T\mathbf{X}_{\mathbf{w}}^T\overline{\mathbf{X}}_{\mathbf{w}}\mathbf{p} + \mathbf{e}_k^T\mathbf{H}_{\mathbf{y}}^*[\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p}\right). \end{aligned}$$

Similar results are obtained for the imaginary part. We conclude that $\mathbf{p}_{j+1}$ necessarily satisfies the eigenvector equation

$$\mathbf{H}_{\mathbf{y}}^*[\mathbf{X}_{\mathbf{p}_j}^T\overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}\mathbf{H}_{\mathbf{y}}\mathbf{p}_{j+1} - \mathbf{X}_{\mathbf{w}_j}^T\overline{\mathbf{X}}_{\mathbf{w}_j}\mathbf{p}_{j+1} = \mathbf{0}.$$

Thus the assertion follows. ■

Remark 4.2 *Observe that the eigenvector equation in* (4.3) *is still an implicit equation since* $\mathbf{w}_j$ *contains* $\mathbf{p}_{j+1}$ *in its definition. In particular,* (4.3) *implies by multiplication with* $(\mathbf{p}_{j+1})^*$ *that*

$$\begin{aligned}
\mathbf{p}_{j+1}^* \mathbf{H}_{\mathbf{y}}^* [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \mathbf{H}_{\mathbf{y}} \mathbf{p}_{j+1} &= \mathbf{p}_{j+1}^* \mathbf{X}_{\mathbf{w}_j}^T \overline{\mathbf{X}}_{\mathbf{w}_j} \mathbf{p}_{j+1} = \|\mathbf{X}_{\mathbf{w}_j} \overline{\mathbf{p}}_{j+1}\|_2^2 \\
&= \mathbf{w}_j^* \mathbf{X}_{\mathbf{p}_{j+1}}^T \overline{\mathbf{X}}_{\mathbf{p}_{j+1}} \mathbf{w}_j \\
&= \mathbf{y}^* \overline{\mathbf{X}}_{\mathbf{p}_j} [\mathbf{X}_{\mathbf{p}_{j+1}}^T \overline{\mathbf{X}}_{\mathbf{p}_{j+1}}]^{-1} \mathbf{X}_{\mathbf{p}_j}^T \mathbf{y} \\
&= \mathbf{p}_j^* \mathbf{H}_{\mathbf{y}}^* [\mathbf{X}_{\mathbf{p}_{j+1}}^T \overline{\mathbf{X}}_{\mathbf{p}_{j+1}}]^{-1} \mathbf{H}_{\mathbf{y}} \mathbf{p}_j
\end{aligned}$$

and thus

$$\mathbf{y}^* \mathbf{A}(\mathbf{p}_{j+1}, \mathbf{p}_j) \mathbf{y} = \mathbf{y}^* \mathbf{A}(\mathbf{p}_j, \mathbf{p}_{j+1}) \mathbf{y}. \tag{4.4}$$

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

Let us now study the convergence of the iteration (4.3).

Theorem 4.3 *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* (4.1) *or by the condition* (4.3) *respectively, is always uniquely defined. Then the sequence* $(F(\mathbf{p}_j, \mathbf{p}_{j+1}))_{j=0}^\infty$ *obtained by* (4.2) *converges to a limit* F^*. *Moreover, the desired vector*

$$\widetilde{\mathbf{p}} = \operatorname*{argmin}_{\substack{\mathbf{p} \in \mathbb{C}^{M+1} \\ \|\mathbf{p}\|_2 = 1}} \mathbf{y}^* \overline{\mathbf{X}}_{\mathbf{p}} [\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}]^{-1} \mathbf{X}_{\mathbf{p}}^T \mathbf{y} \tag{4.5}$$

is a fixed point of the iteration (4.1).

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 (4.2) we have

$$F(\mathbf{p}_j, \mathbf{p}_{j+1}) \le F(\mathbf{p}_j, \mathbf{p}_j) = 2\mathbf{y}^* \overline{\mathbf{X}}_{\mathbf{p}_j} [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \mathbf{X}_{\mathbf{p}_j}^T \mathbf{y} = 2\mathbf{y}^* \overline{\mathbf{P}}_{\mathbf{p}_j} \mathbf{y} \le 2\|\mathbf{y}\|_2^2.$$

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 (4.1)

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

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 (4.5) is indeed a fixed point of the iteration (4.1). By definition, $\widetilde{\mathbf{p}}$ satisfies the necessary condition (3.4) that takes here the form

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

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

$$\begin{aligned}
\mathbf{H}_{\mathbf{r}(\widetilde{\mathbf{p}})}^* [\mathbf{X}_{\widetilde{\mathbf{p}}}^T \overline{\mathbf{X}}_{\widetilde{\mathbf{p}}}]^{-1} \mathbf{H}_{\mathbf{r}(\widetilde{\mathbf{p}})} \widetilde{\mathbf{p}} &= \mathbf{H}_{\mathbf{r}(\widetilde{\mathbf{p}})}^* [\mathbf{X}_{\widetilde{\mathbf{p}}}^T \overline{\mathbf{X}}_{\widetilde{\mathbf{p}}}]^{-1} \mathbf{X}_{\widetilde{\mathbf{p}}}^T \, \mathbf{r}(\widetilde{\mathbf{p}}) = \mathbf{H}_{\mathbf{r}(\widetilde{\mathbf{p}})}^* \overline{\mathbf{X}}_{\widetilde{\mathbf{p}}}^+ \, \overline{\mathbf{X}}_{\widetilde{\mathbf{p}}} \, \overline{\mathbf{X}}_{\widetilde{\mathbf{p}}}^+ \, \mathbf{y} \\
&= \mathbf{H}_{\mathbf{r}(\widetilde{\mathbf{p}})}^* \, \overline{\mathbf{X}}_{\widetilde{\mathbf{p}}}^+ \, \mathbf{y} = \mathbf{H}_{\mathbf{r}(\widetilde{\mathbf{p}})}^* \, \widetilde{\mathbf{w}} = \mathbf{X}_{\widetilde{\mathbf{w}}}^T \, \overline{\mathbf{r}(\widetilde{\mathbf{p}})} = \mathbf{X}_{\widetilde{\mathbf{w}}}^T \, \mathbf{X}_{\widetilde{\mathbf{p}}} \, \mathbf{X}_{\widetilde{\mathbf{p}}}^+ \, \overline{\mathbf{y}} \\
&= \mathbf{X}_{\widetilde{\mathbf{w}}}^T \, \mathbf{X}_{\widetilde{\mathbf{p}}} \, \overline{\widetilde{\mathbf{w}}} = \mathbf{X}_{\widetilde{\mathbf{w}}}^T \, \overline{\mathbf{X}}_{\widetilde{\mathbf{w}}} \, \widetilde{\mathbf{p}}.
\end{aligned} \tag{4.6}$$

Thus, for $\mathbf{p}_j = \widetilde{\mathbf{p}}$ in (4.3), 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 (4.1) to solve the nonlinear problem (2.9), where we successively update the vector $\mathbf{p}$. We start with the initial vector $\mathbf{p}_0$ in (3.5).

First iteration scheme (SIMI-1 or GRA).
Employing the necessary condition in (4.3) we define for a fixed normalized vector $\mathbf{p}_j$ the matrix

$$\mathbf{B}_{\mathbf{p}_j} := \mathbf{H}_{\mathbf{y}} [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \mathbf{H}_{\mathbf{y}} - \mathbf{X}_{\mathbf{v}_j}^T \overline{\mathbf{X}}_{\mathbf{v}_j}$$

with $\mathbf{v}_j := \overline{\mathbf{X}}_{\mathbf{p}_j}^{+} \mathbf{y} = [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \mathbf{X}_{\mathbf{p}_j}^T \mathbf{y}$ and propose the scheme

$$\begin{aligned} \left(\mathbf{B}_{\mathbf{p}_j} - \mu_j \mathbf{I}\right) \mathbf{p}_{j+1} &= \mathbf{0}, \\ \mathbf{p}_{j+1}^* \mathbf{p}_{j+1} &= 1. \end{aligned} \tag{4.7}$$

This iteration scheme is obtained from (4.3), when we approximate $[\mathbf{X}_{\mathbf{p}_{j+1}}^T \mathbf{X}_{\mathbf{p}_{j+1}}]^{-1} \mathbf{X}_{\mathbf{p}_j}$ by $[\mathbf{X}_{\mathbf{p}_j}^T \mathbf{X}_{\mathbf{p}_j}]^{-1} \mathbf{X}_{\mathbf{p}_j}$. The iteration scheme (4.7) is slightly different from the IGRA-iteration in (3.7). We observe that $\mathbf{B}_{\mathbf{p}_j}$ is again a difference of two positive definite matrices. While the first matrix $\mathbf{H}_{\mathbf{y}} [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \mathbf{H}_{\mathbf{y}}$ coincides with the first matrix in the IGRA iteration, the second matrix $\mathbf{X}_{\overline{\mathbf{v}}_j}^* \mathbf{X}_{\overline{\mathbf{v}}_j}$ is different from the matrix $\mathbf{H}_{\mathbf{r}(\mathbf{p}_j)}^* [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \mathbf{H}_{\mathbf{r}(\mathbf{p}_j)}$ in IGRA. However, when applied to the fixed point $\tilde{\mathbf{p}}$, the two matrices give the same result, see (4.6).

Remark 4.4 *It appears that SIMI-1 is equivalent to the GRA-algorithm proposed in* [24] *despite being derived in a different way. In* [25, 26]*, a similar method is considered, which is called difference version. The GRA algorithm in* [25, 26] *does not search for the vector* $\mathbf{p}$ *but for a different vector* $\boldsymbol{\gamma}$ *of parameters that is obtained by using a modification of Prony's algorithm based on the difference operator instead of the shift operator. This is possible since the exponential functions* z_j^x *are eigenfunctions of the shift operator as well as of the difference operator, see also* [27]*. There exists an invertible linear map that transfers* $\boldsymbol{\gamma}$ *to* $\mathbf{p}$*,* [25]*. A detailed study of the matrix* $\mathbf{B}_{\boldsymbol{\gamma}}$ *in* [25, 26] *led to some remarkable asymptotic results. In particular, Osborne and Smyth showed that for a fixed point* $\hat{\boldsymbol{\gamma}}$ *of the iteration* (4.7)*, the matrix* $\frac{1}{1+L}\mathbf{B}_{\hat{\boldsymbol{\gamma}}}$ *has a positive semidefinite limit for* $L \to \infty$ *and that with probability one, the zero eigenvalue of* $\mathbf{B}_{\hat{\boldsymbol{\gamma}}}$ *is asymptotically isolated, see* [26]*, Section 9. Their considerations about the local convergence of the iteration scheme employ the strong assumption that the functional* F *with* $\boldsymbol{\gamma}_{j+1} = F(\boldsymbol{\gamma}_j)$ *has a Frechet derivative with spectral radius smaller than* 1*, and that the fixed point of the iteration* (4.7) *is unique. The uniqueness of the fixed point can however be only shown asymptotically.*

Second iteration scheme (SIMI-2).
We recall that

$$\begin{aligned} \mathbf{y}^* \mathbf{A}(\mathbf{p}_j, \mathbf{p}) \mathbf{y} &= \mathbf{p}_j^* \, \mathbf{H}_{\mathbf{y}}^* [\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}]^{-1} \, \mathbf{H}_{\mathbf{y}} \mathbf{p}_j \\ &= \mathbf{p}_j^* \, \mathbf{H}_{\mathbf{y}}^* [\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}]^{-1} \, [\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}] \, [\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}]^{-1} \, \mathbf{H}_{\mathbf{y}} \mathbf{p}_j. \end{aligned} \tag{4.8}$$

Approximating $[\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}]^{-1}$ by $[\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1}$ in (4.8), we obtain

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

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

$$\begin{aligned}
\mathbf{p}_{j+1} &:= \operatorname*{argmin}_{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}} \Big(\mathbf{y}^* \mathbf{A}(\mathbf{p},\mathbf{p}_j)\mathbf{y} + \mathbf{y}^* \widetilde{\mathbf{A}}(\mathbf{p}_j,\mathbf{p})\mathbf{y}\Big) \\
&= \operatorname*{argmin}_{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}} \Big(\mathbf{p}^* \mathbf{H}_{\mathbf{y}}^* [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \mathbf{H}_{\mathbf{y}} \mathbf{p} + \mathbf{v}_j^* [\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}] \mathbf{v}_j\Big), \\
&= \operatorname*{argmin}_{\substack{\mathbf{p}\in\mathbb{C}^{M+1}\\ \|\mathbf{p}\|_2=1}} \Big(\mathbf{p}^* \mathbf{H}_{\mathbf{y}}^* [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \mathbf{H}_{\mathbf{y}} \mathbf{p} + \mathbf{p}^* [\mathbf{X}_{\mathbf{v}_j}^T \overline{\mathbf{X}}_{\mathbf{v}_j}] \mathbf{p}\Big). \qquad (4.9)
\end{aligned}$$

In the last equation, we have used that $\mathbf{v}_j^*[\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}] \mathbf{v}_j = \overline{\mathbf{v}_j^*[\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}] \mathbf{v}_j}$ and $\mathbf{X}_{\mathbf{p}} \overline{\mathbf{v}}_j = \overline{\mathbf{X}}_{\mathbf{v}_j} \mathbf{p}$ hold. Now each iteration step breaks down to finding the eigenvector to the smallest eigenvalue of the positive semidefinite matrix $\mathbf{H}_{\mathbf{y}}^* [\mathbf{X}_{\mathbf{p}_j}^T \overline{\mathbf{X}}_{\mathbf{p}_j}]^{-1} \mathbf{H}_{\mathbf{y}} + \mathbf{X}_{\mathbf{v}_j}^T \overline{\mathbf{X}}_{\mathbf{v}_j}$. We summarize the procedure with one of the two iteration schemes in Algorithm 4.5.

Algorithm 4.5 (SIMI)
Input: M, x_0, h, y_k, $k = 0, \ldots, L$, *with* $L + 1 = (2M + 1)K$.

1. *Initialization*
   - *(Optional): Compute* $\widetilde{\mathbf{y}}$ *in* (2.11) *and replace in all further steps* $\mathbf{y}$ *by* $\widetilde{\mathbf{y}}$.
   - *Compute* $\mathbf{p}_0$ *in* (3.5).
2. *Iteration: For* $j = 0 \ldots$ *till convergence*
   - *Compute* $\mathbf{p}_{j+1}$ *according to* (4.7) *or* (4.9).
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*

$$\mathbf{V}_{\mathbf{z}} \mathbf{d} = \mathbf{y}.$$

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

In Algorithm 4.5, 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 proposed iteration scheme SIMI-2 we observe the following. Similarly as in the proof of Theorem 4.3 the achieved functional values in the iteration scheme 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.

Finally we study the question, how to compute the inverse matrix $[\mathbf{X}_{\mathbf{p}}^T \overline{\mathbf{X}}_{\mathbf{p}}]^{-1}$ as well as the Moore-Penrose $\overline{\mathbf{X}}_{\mathbf{p}}^{+}$ for given $\mathbf{p} \in \mathbb{C}^{M+1}$ in an efficient way. For that purpose, let $\mathbf{F}_{L+1} := (\omega_{L+1}^{jk})_{j,k=0}^{L}$ be the Fourier matrix of size $(L+1) \times (L+1)$, where $\omega_{L+1} := \mathrm{e}^{-2\pi \mathrm{i}/(L+1)}$. Observe that the Fouriermatrix is almost unitary with $\mathbf{F}_{L+1}^{-1} = \frac{1}{L+1} \overline{\mathbf{F}}_{L+1}$.

Lemma 4.6 *For a given vector* $\mathbf{p} = (p_k)_{k=0}^M \in \mathbb{C}^{M+1}$ *the matrix* $\mathbf{X}_\mathbf{p}$ *in* (2.7) *can be factorized as*

$$\mathbf{X}_\mathbf{p} = \frac{1}{L+1} \overline{\mathbf{F}}_{L+1} \, \mathbf{D}_\mathbf{p} \, \mathbf{F}_{L+1,L-M+1} \tag{4.10}$$

where $\mathbf{D}_\mathbf{p}$ *denotes the diagonal matrix* $\mathbf{D}_\mathbf{p} := \mathrm{diag}(p(\omega_{L+1}^k))_{k=0}^L$ *with*

$$(p(\omega_{L+1}^k)_{k=0}^L = (\sum_{j=0}^M p_j \, \omega_{L+1}^{jk})_{k=0}^L = \mathbf{F}_{L+1,M+1}\mathbf{p},$$

and where $\mathbf{F}_{L+1,L-M+1}$ *and* $\mathbf{F}_{L+1,M+1}$ *denote truncated Fourier matrices containing only the first* $L-M+1$ *and* $M+1$ *columns, respectively. Further, we have*

$$[\mathbf{X}_\mathbf{p}^T \overline{\mathbf{X}}_\mathbf{p}]^{-1} = \frac{1}{L+1} \mathbf{F}_{L+1,L-M+1}^T [\mathbf{D}_\mathbf{p} \overline{\mathbf{D}}_\mathbf{p}]^{+} \overline{\mathbf{F}}_{L+1,L-M+1}.$$

If the vector $\mathbf{F}_{L+1,M+1}\mathbf{p}$ *only has nonzero components, then we also have*

$$\mathbf{X}_\mathbf{p}^{+} = \frac{1}{L+1} \mathbf{F}_{L+1,L-M+1}^{*} \mathbf{D}_\mathbf{p}^{-1} \mathbf{F}_{L+1}.$$

Proof: We consider the circulant matrix $\tilde{\mathbf{X}}_\mathbf{p}$ that is obtained by extension of $\mathbf{X}_\mathbf{p}$ in (2.7) to a square matrix of size $(L+1) \times (L+1)$. Then $\tilde{\mathbf{X}}_\mathbf{p}$ can be diagonalized by the Fourier matrix, i.e.,

$$\tilde{\mathbf{X}}_\mathbf{p} = \mathbf{F}_{L+1}^{-1} \, \mathbf{D}_\mathbf{p} \, \mathbf{F}_{L+1}$$

with the diagonal matrix $\mathbf{D}_\mathbf{p}$, as defined in Lemma 4.6, see e.g. [31], Section 3.3. Now, the factorization (4.10) is obtained by suitable truncation of the last Fourier matrix in the factorization of $\tilde{\mathbf{X}}_\mathbf{p}$. The formula for $[\mathbf{X}_\mathbf{p}^T \overline{\mathbf{X}}_\mathbf{p}]^{-1}$ directly follows from (4.10) using that $\mathbf{F}_{L+1,L-M+1}^{+} = \frac{1}{L+1} \mathbf{F}_{L+1,L-M+1}^{*}$. If moreover $\mathbf{D}_\mathbf{p}$ is invertible, then the factorization of $\mathbf{X}^{+}$ directly follows. ∎

5 Numerical results

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 method (APM) in [32], the SIMI-1 iteration (GRA) in (4.7), the IQML iteration in Algorithm 3.5, the VARPRO method based on Levenberg-Marquardt iteration using the software package of [22], and the two new iterations SIMI-2 in (4.9) and IGRA in Algorithm 3.3. For all algorithms we will also employ the smoothed data $\widetilde{\mathbf{y}}$ in (2.11) alongside the original data vector $\mathbf{y}$. Besides achieving a much smaller error variance in the smoothed data $\widetilde{\mathbf{y}}$, 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 IGRA, IQML and SIMI-iterations.

In all examples, we want to recover the parameters $T_j = \frac{1}{h} \log(z_j)$ and d_j of the signal $f(x) = \sum_{j=1}^M d_j \mathrm{e}^{T_j x}$ from noisy measurements $y_k = f(kh) + \epsilon_k$. With the previous notation we have $z_j = \mathrm{e}^{T_j h}$ where h is some fixed step size. The recovered signal is denoted by $\widehat{f}(x) = \sum_{j=1}^M \widehat{d}_j e^{\widehat{T}_j x}$. The recovery of d_j is done in the same way for all algorithms, therefore we present only the results for T_j. 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 vector, respectively. In Example 5.1 we particularly test the dependence of the results from the number of measurements L for fixed noise level. In Example 5.2 we consider different noise distributions and noise levels. In the third example we test different initial vectors for different levels of Gaussian noise. The last example contains complex parameters z_j and T_j, respectively. We will study the number of the iterations (NoI), the *relative error* $e(f)$ given by

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

and the normalized 2-*error*

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

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

In the first, the second and the last example, we present the mean values $\widehat{T}_j$ 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 this example from* [18] *we use $h = 1/L$ and consider the data*

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

where $\epsilon_k \sim N(0, 0.01)$, i.e. the deviation is $\sigma = 0.1$. We use either the full data vector $\mathbf{y}$ with $L = 11, 32, 128, 512$ or the filtered data $\widetilde{\mathbf{y}} \in \mathbb{R}^3$ in (2.11)*. The bound for the highest number of iterations is set to* 10*. We compare the results for each algorithm in Table* 1*. The mean values of the normalized* 2*-error $\frac{1}{L+1}\left(\sum_{k=0}^{L} |f(k/L) - y_k)|^2\right)^{1/2}$ achieved by the exact parameters are* 0.0291 ($L = 11$)*,* 0.0174 ($L = 32$)*,* 0.0088 ($L = 128$)*,* 0.0062 ($L = 254$) *and* 0.0044 ($L = 512$).

We observe in Table 1 *that the direct methods PM and APM do not profit from a higher number L of samples. Particularly PM obtains even worse results. These results verify that the non-iterative Prony methods are not consistent,* [18]*. All iterative methods achieve with their estimated parameters mean errors in the same range as the optimal parameters, and the errors decreases for growing L. For filtered data, all methods work equivalently well. For larger L, we have a stronger reduction of noise variance in $\widetilde{\mathbf{y}}$, see the remarks below* (2.11)*.*

Example 5.2 *We consider the example in* [26] *with $M = 2$ and $h = 1/L$ of the form*

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

Here, we are interested in the performance of the algorithms for $\epsilon_k \in N(0, \sigma^2)$ with different deviations, $\sigma \in \{0.001, 0.01, 0.05\}$. We show the results in Table 2 *for a fixed $L =$ 49. In addition to normal distribution, we show the results for $\epsilon_k \sim$ Lognormal$(0, \sigma^2)$*

with different deviations σ and $L = 254$ in Table 3*. Again we compare the* 2*-errors provided by the algorithms to the errors obtained by taking the exact parameters $T_1 = -4$ and $T_2 = -7$. The mean values of the normed* 2*-error for the measured samples with normal distribution noise are* $1.39e-04$ ($\sigma = 0.001$), 0.0014 ($\sigma = 0.01$) *and* 0.0071 ($\sigma = 0.05$)*. Those with Lognormal distribution are* $6.29e-05$ ($\sigma = 0.001$), 0.0006 ($\sigma = 0.01$) *and* 0.0022 ($\sigma = 0.005$)*. For smaller noise levels, all iterative methods work well and achieve even better* 2*-errors than the correct parameter vector. However, for $\sigma = 0.05$ completely different parameters are provided, while the obtained errors are very small. This shows, that many different parameter vectors allow an approximation of the given data with a similar* 2*-error.*

The results of Table 3 *show that the iterative methods can also cope with different noise distributions. For $\sigma = 0.05$, the methods IQML, VARPRO, as well as the new iterations SIMI-*2 *and IGRA find parameter vectors which are quite away from the original parameter vector $T = (-4, -7)$ but achieve a much smaller error of about* $3.4e-04$ *instead of* 0.0022*. SIMI-*1 *(GRA) does not work as well in this case. These results show however the strong ill-posedness of the parameter estimation problem, while very good approximation results are achieved.*

Example 5.3 *Now we investigate a three-term model with $h = 5/L$ of the form*

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

where $\epsilon_k \sim N(0, \sigma^2)$ 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. 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). With the correct parameters, we obtain the normed* 2*-error for the measured samples* $1.3371e-05$, $1.0789e-04$ *and* 0.0013 *for* $\sigma = 0.0001$, 0.001 *and* 0.01*, respectively.*

In this example we have 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* 4*. As one can see, the SIMI-*1 *(GRA) iteration depends more strongly on the starting vector than the other iterative 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.*

Example 5.4 *At last, a frequency estimation example in* [18] *will be studied,*

$$y_k = \cos(0.1k + 1) + \epsilon_k = \frac{\mathrm{e}^{\mathrm{i}}}{2}\mathrm{e}^{\mathrm{i}k/10} + \frac{\mathrm{e}^{-\mathrm{i}}}{2}\mathrm{e}^{-\mathrm{i}k/10} + \epsilon_k \qquad k = 0, 1, \ldots, L,$$

where $\epsilon_k \sim N(0, 0.01)$. We use either the full data vector $\mathbf{y}$ with $L = 14, 49, 254, 514$, or the filtered data $\widetilde{\mathbf{y}}$ in (2.11)*. 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*. The mean values of the normalized* 2*-error for the measured samples are* 0.0255 $(L = 14)$, 0.0139 $(L = 49)$, 0.0063 $(L = 254)$ *and* 0.0044 $(L = 514)$*. Convergence results can be obtained from the iteration algorithms with the full data. However, for the filtered data we suffer from aliasing effects caused by periodicity of* $\mathrm{e}^{\mathrm{i}x}$*. Here, we reconstruct* z_1^K *and* $z_2^K = z_1^{-K}$ *instead of* $z_1 = \mathrm{e}^{\mathrm{i}/10}$ *and* $z_2 = \mathrm{e}^{-\mathrm{i}/10}$*. For* $L = 254$ *we have* $K = 51$ *and for* $L = 514$ *we have* $K = 103$*, see* (2.11)*. While* z_1^K *and* z_2^K *can be still well reconstructed, we cannot extract* T_1 *and* $T_2 = -T_1$ *uniquely by restricting the phase to* $[-\pi, \pi]$*, since* $KT_1 = K/10$ *and* $KT_2 = -K/10$ *are not longer in* $[-\pi, \pi]$.

6 Conclusion

In this paper, we have surveyed different numerical methods to solve the problem of optimal recovery of signal vectors by vectors constructed with short exponential sums. This problem appears in many applications, where one needs to estimate exponential decays or requires a sparse approximation of the data using exponential sums. If the exponential function model is known beforehand and the measurements contain i.i.d random noise, then the considered model is consistent for $L \to \infty$, while the non-iterative methods PM and APM are not consistent, see [18]. Usually, the results of non-iterative methods can be strongly improved by employing a filtering in a pre-processing step. Pre-filtering may however cause aliasing effects.

One main goal was to present a uniform framework to solve the nonlinear minimization problem and to recover optimal parameters with respect to the 2-norm error. In particular, we are interested in iteration algorithms that are robust with regard to the choice of initial vectors and converge quickly. Using an explicit representation of the Jacobian matrix, we proposed the algorithm IGRA in Section 3, which is close in nature but not equivalent to the GRA algorithm by Osborne and Smyth [26]. Further, we proposed a new iteration scheme based on simultaneous minimization in Section 4. This approach leads to two schemes SIMI-1 and SIMI-2. SIMI-1 appears to be equivalent to GRA for the recurrence case in [25, 26]. The numerical experiments show that the two new schemes IGRA and SIMI-2 converge fast and are more robust with regard to the choice of starting vectors than VARPRO, see Example 5.3 and Table 4. Moreover, it can be seen from the numerical examples that the problem of parameter identification is ill-posed. We are able to find very good approximations of the given measurements using exponential sums with different parameter vectors.

7 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.

		PM	APM	SIMI-I (GRA)	IQML	VARPRO	SIMI-2 (4.9)	IGRA
$L = 11$	NoI	\	\	6	5	4	5	7
	$\widehat{T}$	-4.6939	-4.1239	-4.0941	-3.3699	-4.0068	-4.0089	-4.2321
	rel. error	0.0955	0.0844	0.0742	0.0733	0.0815	0.0729	0.0850
	2-error	0.0266	0.0259	0.0249	0.0249	0.0262	0.0249	0.0251
with filter	NoI	\	\	7	6	3	4	7
	$\widehat{T}$	-3.9933	-3.9765	-4.0136	-4.2473	-3.9966	-4.0324	-3.9905
	rel. error	0.0914	0.0918	0.0900	0.0899	0.0929	0.0897	0.0921
	2-error	0.0262	0.0228	0.0262	0.0262	0.0277	0.0262	0.0265
$L = 32$	NoI	\	\	6	5	3	4	6
	$\widehat{T}$	-6.0656	-3.915	-3.9702	-3.939	-4.0797	-3.9164	-4.4257
	rel. error	0.1773	0.0760	0.0532	0.0533	0.0611	0.0534	0.0564
	2-error	0.0216	0.0175	0.0168	0.0168	0.0168	0.0168	0.0175
with filter	NoI	\	\	7	6	2	4	6
	$\widehat{T}$	-4	-3.9931	-4.004	-4.0386	-4.0838	-4.0113	-4.0147
	rel. error	0.0676	0.0677	0.0670	0.0670	0.0762	0.0670	0.0690
	2-error	0.0171	0.0180	0.0171	0.0171	0.0172	0.0171	0.0179
$L = 128$	NoI	\	\	6	5	3	5	6
	$\widehat{T}$	-13.5582	-3.8959	-4.0196	-4.0052	-4.0018	-4.0013	-4.0456
	rel. error	0.4828	0.0818	0.0347	0.0347	0.0296	0.0348	0.0296
	2-error	0.0190	0.0092	0.0087	0.0087	0.0088	0.0087	0.0086
with filter	NoI	\	\	5	5	2	4	5
	$\widehat{T}$	-3.9893	-3.9871	-3.9887	-4.0123	-3.9948	-3.991	-3.9848
	rel. error	0.0438	0.0439	0.0424	0.0424	0.0329	0.0423	0.0376
	2-error	0.0088	0.0087	0.0088	0.0088	0.0088	0.0088	0.0087
$L = 254$	NoI	\	\	5	4	2	4	5
	$\widehat{T}$	-22.8268	-3.8254	-4.0040	-4.0082	-4.0172	-3.9944	-3.9959
	rel.error	0.6387	0.0903	0.0207	0.0207	0.0219	0.0207	0.0207
	2-error	0.0167	0.0067	0.0062	0.0062	0.0062	0.0062	0.0062
with filter	NoI	\	\	5	5	2	4	5
	$\widehat{T}$	-3.9918	-3.9908	-3.9964	-4.0261	-4.0165	-3.9975	4.0555
	rel.error	0.0239	0.0239	0.0243	0.0243	0.0245	0.0243	0.0243
	2-error	0.0062	0.0062	0.0062	0.0062	0.0062	0.0062	0.0062
$L = 512$	NoI	\	\	5	4	4	4	5
	$\widehat{T}$	-43.3615	-3.9179	-3.9989	-4.0031	-3.9880	-3.9937	-4.0396
	rel. error	0.7641	0.0764	0.0150	0.0150	0.0152	0.0150	0.0150
	2-error	0.0138	0.0046	0.0044	0.0044	0.0044	0.0044	0.0044
with filter	NoI	\	\	4	4	2	3	4
	$\widehat{T}$	-3.9958	-3.9952	-4.0006	-4.0339	-3.9755	-4.0013	-4.0075
	rel. error	0.0178	0.0178	0.0172	0.0172	0.0178	0.0172	0.0172
	2-error	0.0044	0.0044	0.0044	0.0044	0.0044	0.0044	0.0044

Table 1:
Simulation results for perturbed signal values $y_k = \exp(-4x_k) + \epsilon_k$, $\epsilon_k \sim N(0, 0.01)$, $k = 0, 1, \ldots, L$, and the low-pass filtered data $\widetilde{y}_k, k = 0, 1, 2$, in Example 5.1.

		PM	APM	SIMI-1 (GRA)	IQML	VARPRO	SIMI-2 (4.9)	IGRA
$\sigma = 0.001$	NoI	\	\	4	3	2	4	4
	$\widehat{T}$	-3.5124 -8.2330	-4.0558 -6.8757	-3.9984 -7.0059	-3.9860 -7.0394	-3.9990 -7.0036	-3.9989 -7.0044	-3.9939 -7.0011
	rel. error	0.0132	0.0051	0.0011	0.0011	0.0012	0.0011	0.0011
	2-error	7.66e-04	3.19e-04	1.36e-04	1.36e-04	1.34e-04	1.36e-04	1.36e-04
with filter	NoI	\	\	4	3	4	3	4
	$\widehat{T}$	-3.9949 -7.0203	-3.9951 -7.0195	-3.9963 -7.0148	-3.9997 -6.9967	-4.0180 -6.9335	-3.9962 -7.0152	-3.9906 -7.0420
	rel. error	0.0027	0.0027	0.0025	0.0025	0.0025	0.0025	0.0025
	2-error	1.52e-04	1.52e-04	1.50e-04	1.50e-04	1.46e-04	1.50e-04	1.50e-04
$\sigma = 0.01$	NoI	\	\	9	5	5	7	8
	$\widehat{T}$	-1.5070 -84.6605	-3.9369 -7.0896	-3.9919 -7.0168	-4.0374 -6.9200	-4.0048 -6.9828	-4.0424 -6.8672	-3.9857 -7.2087
	rel. error	0.1713	0.0667	0.0118	0.0118	0.0121	0.0118	0.0118
	2-error	0.0110	0.0037	0.0013	0.0013	0.0014	0.0013	0.0013
with filter	NoI	\	\	6	5	5	5	5
	$\widehat{T}$	-4.0232 -6.8825	-4.0496 -6.7948	-4.0561 -6.7756	-4.4972 -5.7285	-4.1783 -6.4773	-4.0453 -6.8130	-3.9047 -7.2656
	rel. error	0.0250	0.0252	0.0248	0.0249	0.0219	0.0249	0.0249
	2-error	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015
$\sigma = 0.05$	NoI	\	\	10	8	10	10	10
	$\widehat{T}$	-1.6878 -39+154i	-1.5845 -75.7982	-2.2097 75.4581	-4.8846 -5.2294	-2.7758 -19.0629	-4.4405 ±1.6933i	-2.6548 -21.1387
	rel. error	0.2068	0.3399	0.6233	0.0611	0.0859	0.0669	0.0801
	2-error	0.0119	0.0184	0.0285	0.0069	0.0071	0.0070	0.0071
with filter	NoI	\	\	9	7	6	7	9
	$\widehat{T}$	-3.5519 -10.0998	-3.7656 -8.2865	-3.7250 -8.7870	-1.5973 ±1.5356i	-3.353 ±2.14i	-3.5876 -9.9987	-3.4144 -12.1183
	rel. error	0.1467	0.1449	0.1394	0.1408	0.1362	0.1446	0.1394
	2-error	0.0081	0.0080	0.0079	0.0079	0.0076	0.0080	0.0079

Table 2:
Simulation results for perturbed signal values $y_k = 2\exp(-4x_k) - \exp(-7x_k) + \epsilon_k$, $\epsilon_k \sim N(0, \sigma^2)$, $k = 0, 1, \ldots, L$, with $L = 49$ and the low-pass filtered data $\widetilde{y}_k$, $k = 0, 1, 2, 3, 4$, in Example 5.2.

		PM	APM	SIMI-1 (GRA)	IQML	VARPRO	SIMI-2 (4.9)	IGRA
$\sigma = 0.001$	NoI	\	\	3	3	2	3	3
	$\widehat{T}$	-3.9683 -7.0417	-3.9684 -7.0415	-3.9449 -7.1083	-3.9449 -7.1082	-3.9449 -7.1083	-3.9449 -7.1082	-3.9449 -7.1083
	rel.error	0.0019	0.0019	0.0019	0.0019	0.0019	0.0019	0.0019
	2-error	1.07e-05	1.07e-05	6.4e-06	6.43e-06	6.43e-06	6.43e-06	6.43e-06
with filter	NoI	\	\	3	3	1	3	3
	$\widehat{T}$	-3.9261 -7.1888	-3.9262 -7.1888	-3.9224 -7.2052	-3.9224 -7.2052	-3.9258 -7.1903	-3.9224 -7.2052	-3.9224 -7.2050
	rel.error	0.0025	0.0025	0.0026	0.0026	0.0025	0.0026	0.0026
	2-error	1.44e-05	1.44e-05	1.68e-05	1.68e-05	1.46e-05	1.68e-05	1.68e-05
$\sigma = 0.01$	NoI	\	\	5	4	2	4	5
	$\widehat{T}$	-3.1808 -9.0047	-3.7189 -7.3791	-3.5649 -7.9637	-3.5667 -7.9546	-3.5649 -7.9634	-3.5685 -7.9452	-3.5643 -7.9663
	rel.error	0.0339	0.0193	0.0187	0.0187	0.0187	0.0187	0.0187
	2-error	4.47e-04	1.21e-04	6.39e-05	6.39e-05	6.38e-05	6.40e-05	6.40e-05
with filter	NoI	\	\	4	4	2	4	4
	$\widehat{T}$	-3.4646 -8.8142	-3.4655 -8.8064	-3.4475 -8.9900	-3.4473 -8.9919	-3.4605 -8.8554	-3.4471 -8.9939	-3.4498 -8.9622
	rel.error	0.0259	0.0258	0.0275	0.0275	0.0262	0.0275	0.0275
	2-error	1.58e-04	1.57e-04	1.85e-04	1.85e-04	1.64e-04	1.86e-04	1.85e-04
$\sigma = 0.05$	NoI	\	\	10	7	4	10	8
	$\widehat{T}$	-1.0785 -956.9269	-3.15 -7.7842	35.1825 -1.9647	-2.7327 -10.5473	-2.7190 -10.8301	-2.7570 -10.2465	-2.7134 -10.8505
	rel.error	0.2812	0.1352	0.9260	0.0936	0.0934	0.0937	0.0934
	2-error	0.0059	0.0020	0.0211	3.40e-04	3.38e-04	3.46e-04	3.39e-04
with filter	NoI	\	\	8	6	9	6	8
	$\widehat{T}$	-2.5766 -20.7+15.7i	-2.5821 -22.6+15.6i	-2.5596 -17.0+15.6i	-2.5594 -17.0+15.6i	-2.5671 18.9 +15.6i	-2.5564 -16.5+15.6i	-2.5691 -18.2+15.6i
	rel.error	0.2486	0.2448	0.2662	0.2678	0.2554	0.2695	0.2663
	2-error	0.0021	0.0021	0.0020	0.0020	0.0020	0.0020	0.0020

Table 3:
Simulation results for perturbed signal values $y_k = 2\exp(-4x_k) - \exp(-7x_k) + \epsilon_k$, $\epsilon_k \sim \text{Lognormal}(0, \sigma^2)$, $k = 0, 1, \ldots, L$, with $L = 254$ and the low-pass filtered data $\widetilde{y}_k$, $k = 0, 1, 2, 3, 4$, in Example 5.2.

		APM	SIMI-1 (GRA)	IQML	VARPRO	SIMI-2 (4.9)	IGRA
$\sigma = 0.0001$	NoI	\	20	3	5	20	20
$\mathbf{p}_0$ is last singular vector	$\widehat{T}$	0.9487 0.4779 0.1910	0.9490 0.4755 0.1765	0.9502 0.5040 0.2031	0.9501 0.5017 0.2012	0.9505 0.5127 0.2095	0.9505 0.5126 0.2095
	rel. error	1.20e-05	1.27e-06	3.10e-07	3.99e-07	6.36e-07	6.32e-07
	2-error	7.49e-05	1.26e-05	1.02e-05	1.03e-05	1.04e-05	1.04e-05
with filter	NoI	\	20	3	1	20	20
	$\widehat{T}$	0.9498 0.4943 0.1947	0.9497 0.4928 0.1942	0.9497 0.4928 0.1942	0.9506 0.5148 0.2109	0.9497 0.4928 0.1942	0.9497 0.4928 0.1942
	rel. error	8.36e-07	7.17e-07	7.17e-07	7.49e-07	7.16.e-07	7.17e-07
	2-error	1.07e-05	1.09e-05	1.09e-05	1.09e-05	1.09e-05	1.09e-05
$\sigma = 0.001$	NoI	\	20	6	18	20	20
$\mathbf{p}_0$ is third last singular vector	$\widehat{T}$	-2.4+43.4i 0.9409 0.3438	0.9507 0.5143 0.2095	0.9510 0.5215 0.2143	0.9475 0.4433 0.1345	0.9511 0.5226 0.2150	0.9512 0.5261 0.2172
	rel. error	1.08e-04	2.99e-06	2.94e-06	3.28e-06	2.95e-06	2.97e-06
	2-error	8.64e-04	1.14e-04	1.14e-04	1.40e-04	1.14e-04	1.14e-04
with filter	NoI	\	12	5	20	20	4
	$\widehat{T}$	0.9503 0.5113 0.2118	0.8412 -0.12+3.6i -0.12-3.6i	0.9511 0.5223 0.2146	0.9510 0.5220 0.2146	0.9511 0.5225 0.2148	0.9511 0.5220 0.2145
	rel. error	3.82e-06	0.0158	2.89e-06	2.44e-06	2.90e-06	2.88e-06
	2-error	1.16e-04	0.1008	1.14e-04	1.41e-04	1.14e-04	1.14e-04
$\sigma = 0.01$	NoI	\	20	12	20	20	20
$\mathbf{p}_0$ is second last singular vector	$\widehat{T}$	12.1346 1.0153 0.5318	0.9566 0.5658 0.2143	0.9389 0.4+43.4i 0.3378	0.9389 0.8+43.4i 0.3379	0.9528 0.4998 0.1648	0.9566 0.5650 0.2139
	rel. error	0.7328	3.11e-05	1.02e-04	1.09e-04	3.06e-05	3.11e-05
	2-error	4.6464	0.0013	0.0014	0.0014	0.0013	0.0013
with filter	NoI	\	20	5	8	20	18
	$\widehat{T}$	0.9431 0.3681 -0.3735	0.9372 0.3325 0.1+4.3i	0.9443 0.3733 -0.4417	0.9443 0.3738 -0.4324	0.9442 0.3728 -0.4516	0.9372 0.3325 0.1+4.3i
	rel. error	4.84e-05	1.88e-04	3.41e-05	3.38e-05	3.43e-05	1.88e-04
	2-error	0.0013	0.0015	0.0013	0.0013	0.0013	0.0015

Table 4:
Simulation results for perturbed signal values $y_k = \exp(0.95\,x_k) + \exp(0.5\,x_k) + \exp(0.2\,x_k) + \epsilon_k$, $k = 0, 1, \ldots, L$, with $L = 69$ and the low-pass filtered data $\widetilde{y}_k$, $k = 0, \ldots, 6$, in Example 5.3.

		PM	APM	SIMI-1 (GRA)	IQML	VARPRO	SIMI-2 (4.9)	IGRA
$L = 14$	NoI	\	\	9	6	5	7	8
	T	0.0002 -3.2774	1e-3-0.10i 1e-3+0.10i	-4e-3-0.08i -4e-3+0.08i	-6e4-0.09i -6e4+0.09i	5e-4-0.11i 5e-4+0.11i	3e-3-0.12i 3e-3+0.12i	0.01-0.12i 0.01+0.12i
	rel.error	0.8597	0.3567	0.1447	0.1344	0.1346	0.1351	0.1346
	2-error	0.0926	0.0386	0.0223	0.0216	0.0216	0.0217	0.0216
with filter	NoI	\	\	7	5	5	7	7
	$\widehat{T}$	-0.01-0.12i -0.01+0.12i	-5e-4-0.10i -5e-4+0.10i	2e-3-0.1i 2e-3+0.1i	0.3143 -1.1243	0.0165 -0.0492	6e-4-0.11i 6e-4+0.11i	0.02-0.11i 0.02+0.11i
	rel.error	0.1876	0.1659	0.1539	0.1546	0.1653	0.1553	0.1539
	2-error	0.0249	0.0234	0.0227	0.0227	0.0240	0.0227	0.0226
$L = 49$	NoI	\	\	6	4	3	5	5
	T	-1.92+3.14i -0.0064	9e-4-0.10i 9e-4+0.10i	2e-4-0.10i 2e-4+0.10i	4e-4-0.10i 4e-4+0.10i	2e-4-0.10i 2e-4+0.10i	1e-4-0.10i 1e-4+0.10i	-8e-3-0.10i -8e-3+0.10i
	rel.error	1.1464	0.6742	0.0494	0.0494	0.0494	0.0493	0.0494
	2-error	0.0850	0.0498	0.0134	0.0134	0.0134	0.0134	0.0134
with filter	NoI	\	\	5	4	3	4	5
	$\widehat{T}$	-2e-5-0.10i -2e-5+0.10i	2e-4-0.10i 2e-4+0.10i	9e-5-0.10i 9e-5+0.10i	4e-4-0.10i 4e-4+0.10i	-7e-4-0.10i -7e-4+0.10i	9e-5-0.10i 9e-5+0.10i	1e-3-0.10i 1e-3+0.10i
	rel.error	0.0546	0.0548	0.0549	0.0549	0.0519	0.0550	0.0549
	2-error	0.0135	0.0135	0.0135	0.0135	0.0136	0.0135	0.0135
$L = 254$	NoI	\	\	5	4	3	5	6
	T	-1.70+3.14i -0.02	-3e-3-0.10i -3e-3+0.10i	-4e-6-0.1i -4e-6+0.1i	-1e-5-0.1i -1e-5+0.1i	-4e-6-0.1i -4e-6+0.1i	-4e-6-0.10i -4e-6+0.10i	-6e-4-0.10i -6e-4+0.10i
	rel.error	1.1014	0.7395	0.0259	0.0259	0.0259	0.0259	0.0259
	2-error	0.0441	0.0242	0.0062	0.0062	0.0062	0.0062	0.0062
with filter	NoI	\	\	6	5	4	6	6
	$\widehat{T}$	-1e-4-0.02i -1e-4+0.02i	2e-6-0.02i 2e-6+0.02i	-3e-5-0.02i -3e-5+0.02i	2e-4-0.02i 2e-4+0.02i	3e-4-0.02i 3e-4+0.02i	-3e-5-0.02i -3e-5+0.02i	5e-4-0.02i 5e-4+0.02i
	rel.error	1.0373	1.0386	1.0384	1.0384	1.0412	1.0384	1.0384
	2-error	0.0445	0.0445	0.0445	0.0445	0.0445	0.0445	0.0445
$L = 514$	NoI	\	\	5	4	3	5	5
	T	-1.64+3.14i -0.02	4e-4-0.10i 4e-4+0.10i	4e-6-0.1i 4e-6+0.1i	8e-6-0.10i 8e-6+0.10i	4e-6-0.1i 4e-6+0.1i	4e-6-0.10i 4e-6+0.10i	-2e-5-0.10i -2e-5+0.10i
	rel.error	1.1026	0.8710	0.0188	0.0189	0.0188	0.0189	0.0188
	2-error	0.0311	0.0202	0.0044	0.0044	0.0044	0.0044	0.0044
with filter	NoI	\	\	5	4	4	5	5
	$\widehat{T}$	-8e-5-0.02i -8e-5+0.02i	-1e-5-0.02i -1e-5+0.02i	-2e-5-0.02i -2e-5+0.02i	-3e-5-0.02i -3e-5+0.02i	-2e-8-0.02i -2e-8+0.02i	-2e-5-0.02i -2e-5+0.02i	-1e-4-0.02i -1e-4+0.02i
	rel.error	1.0438	1.0431	1.0431	1.0431	1.0423	1.0431	1.0431
	2-error	0.0312	0.0312	0.0312	0.0312	0.0312	0.0312	0.0312

Table 5:
Simulation results for perturbed signal values $y_k = \cos(0.1x_k + 1) + \epsilon_k$, $\epsilon_k \sim N(0, 0.01)$, $k = 0, 1, \ldots, L$, and the low-pass filtered data $\widetilde{y}_k, k = 0, 1, 2, 3, 4$, in Example 5.4.

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.*, **46**(1) (2019), 40–65.

[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] 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.

[20] 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.

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

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

[23] 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.

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

[25] 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.

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

[27] T. Peter and G. Plonka, A generalized Prony method for reconstruction of sparse sums of eigenfunctions of linear operators, *Inverse Problems* **29** (2013), 025001.

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

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

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

[31] G. Plonka, D. Potts, G. Steidl, and M. Tasche, Numerical Fourier Analysis, Springer, 2019.

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

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

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

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

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

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

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

[39] 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.