Proceedings of the Project Review, Geo-Mathematical Imaging Group (Purdue University, West Lafayette IN), Vol. 1 (2011) pp. 21-36.

A FAST ALTERNATING PROJECTION METHOD FOR COMPLEX FREQUENCY ESTIMATION

FREDRIK ANDERSSON* AND MARCUS CARLSSON†

Abstract. We detect n complex frequencies by employing an method the iterative projects onto the Hankel matrices and rank n matrices.

1. Introduction. In this paper we will develop a method for the detection of complex frequencies from a signal by employing alternate projections on two matrix classes. The two matrix classes are complex symmetric rank n-matrices and Hankel matrices. By a complex frequency, we refer to the coefficient $\zeta \in \mathbb{C}$ in an exponential of the form $e^{\zeta t}$. For signal processing purposes, the case of purely harmonic oscillation ($\mathrm{Re}(\zeta) = 0$) is often considered.

There are several techniques for the estimation of (complex) frequencies from a signal. Two of the most commonly used ones are multiple signal classification (MUSIC) [13, 4] and Estimation using rotational invariance (ESPRIT) [12]. The MUSIC method is a generalization of the Pisarenko method [11]. Recently, complex frequency estimation has been used in the construction of close to optimal quadratures, for instance for bandlimited functions [3]. This work is related to the work of Adamjan, Arov and Krein [1], and the algorithms described in [3] have been investigated in more detail in [2].

To be more precise, let w be a positive function on $[-1, 1]$ and let $\omega : [-2, 2] \to \mathbb{R}^+$ be the weight obtained by

$$\omega = w * w. \tag{1.1}$$

Let $L^2(\omega)$ denote the set of functions for which $\|f\|_\omega^2 = \int_{-2}^{2} |f(t)|^2 \omega(t)\, dt < \infty$. Given $f \in L^2(\omega)$ and $n \in \mathbb{N}$, we are interested in computationally efficient methods for finding the best (or close to best) approximation of f by functions of the form $\sum_{k=1}^{n} c_k e^{\zeta_k t}$. Since the paper is motivated by concrete applications, we have for simplicity decided to develop the theory for finite sequences rather than functions on $[-2, 2]$, although, using techniques similar to those developed in [2], it would be possible to stay in the "continuous" world and consider discretization as an implementational issue.

2. Preliminaries.

2.1. Low rank complex symmetric matrices. We use the notation $\mathbb{M}_{2N+1,2N+1}$ to denote the Hilbert space of $(2N+1) \times (2N+1)$ matrices with complex values, equipped with the Frobenius norm

$$\|A\| = \sum_{j=-N}^{N} \sum_{j=-N}^{N} |A(j,k)|^2. \tag{2.1}$$

In places we will also associate an element $A \in \mathbb{M}_{2N+1,2N+1}$ with the linear operator $\mathbb{C}^{2N+1} \to \mathbb{C}^{2N+1}$ that is described by A.

Complex symmetric matrices satisfy the symmetry condition $A = A^T$. Note that this is different from the usual the ordinary (Hermitian) symmetry condition $A = A^*$. In similarity with symmetric matrices, which are almost always diagonalizable, complex symmetric matrices can generically be decomposed as

$$A = \sum_{m=1}^{2N+1} s_m u_m u_m^T, \quad s_m \in \mathbb{R}^+, \quad u_m \in \mathbb{C}^{2N+1},$$

*Centre for Mathematical Sciences, Lund University, Box 118, SE-22100, Lund, Sweden

†Department of Mathematics, Purdue University, 150 N. University Street, West Lafayette, IN 47907, USA, Now at Universidad de Santiago de Chile, Avenida Bernardo OńHiggins 3363, Estacion Central, Santiago, Chile, and Centre for Mathematical Sciences, Lund University, Box 118, SE-22100, Lund, Sweden

where the vectors $\{u_m\}_m$ are orthogonal. This decomposition of called a *Takagi* factorization. Note that in contrast to Hermitian case, the numbers $s_m > 0$. Moreover, the vectors u_m satisfy the relation

$$A\overline{u_m} = s_m u_m.$$

In [7], the vectors u_m are referred to as *con-eigenvectors* and the positive numbers s_m are referred to as *con-eigenvalues*. However, the con-eigenvectors are simply the singular vectors (obtained from the Singular Value Decomposition), and the con-eigenvalues are the singular values. This is easily seen by noting that

$$s_m^2 u_m = A^* A u_m.$$

The converse is not true, since it is easily seen that e.g. iu_m fails to be a con-eigenvector but is still a singular vector. However, in the case when the s_m's are distinct and $(u_m)_{m=0}^{2N}$ is any basis of singular vectors, then one can pick $\theta_m \in [0, 2\pi]$ such that $(e^{i\theta_m} u_m)_{m=0}^{2N}$ are con-eigenvectors. For this paper, we are only interested in the zeroes of the corresponding polynomials, and hence the θ_m's have no importance, but it will be computationally more convenient to extract the con-eigenvectors, and we have thus chosen to use this terminology.

By the above remarks, the Eckart Young theorem applies to give:

THEOREM 2.1. *Let $A \in \mathbb{M}_{2N+1,2N+1}$ be complex symmetric matrix with distinct con-eigenvalues. Given a positive integer $n \leq 2N+1$, then the best rank n approximation of A (in $\mathbb{M}_{2N+1,2N+1}$) is given by*

$$\sum_{m=1}^{n} s_m u_m u_m^T, \tag{2.2}$$

where s_m and u_m are the (decreasingly ordered) con-eigenvalues and con-eigenvectors of A, respectively. For proof, cf. [7, p 205].

We will also make use of approximations in weighted spaces. Given a positive weight $w \in \mathbb{R}^{2N+1}$, we denote by $\mathbb{M}^w_{2N+1,2N+1}$ the Hilbert space of matrices with the weighted Frobenius norm

$$\|A\|_w = \sum_{j,k=-N}^{N} w(j)|A(j,k)|^2 w(k).$$

THEOREM 2.2. *Let $A \in \mathbb{M}^w_{2N+1,2N+1}$, and let s_m and q_m denote con-eigenvalues and con-eigenvectors of $B = \operatorname{diag}(\sqrt{w})\ A\ \operatorname{diag}(\sqrt{w})$. Then the best approximation of A (in $\mathbb{M}^w_{2N+1,2N+1}$) is given by*

$$\sum_{m=1}^{n} s_m u_m u_m^T,$$

where $u_m(l) = q_m(l)/\sqrt{w(l)}$, $-N \leq l \leq N$.

Proof. By definition

$$\begin{aligned}
&\operatorname*{argmin}_{s_m,u_m} \|A - \sum_{m=1}^{n} s_m u_m u_m^T\|_w \\
&\quad = \operatorname*{argmin}_{s_m,u_m} \sum_{j,k} \left| \sqrt{w(j)} A(j,k) \sqrt{w(k)} - \sum_{m=1}^{n} s_m u_m(j) \sqrt{w(j)} u_m(k) \sqrt{w(k)} \right|^2 \\
&\quad = \operatorname*{argmin}_{s_m,u_m} \sum_{j,k} \left| B(j,k) - \sum_{m=1}^{n} s_m u_m(j) \sqrt{w(j)} u_m(k) \sqrt{w(k)} \right|^2
\end{aligned}$$

which is according to Theorem 2.1 is minimized be the con-eigenvectors $q_m(j) = u_m(j)\sqrt{w(j)}$, $-N \le j \le N$ and by choosing s_m as the con-eigenvalues of B. □

We denote by $\mathcal{M}^n = \mathcal{M}_N^{w,n} \subset \mathbb{M}^w_{2N+1,2N+1}$ the set of complex symmetric matrices with rank n,

$$\mathcal{M}^n \ni A = \sum_{m=1}^{n} s_m u_m u_m^T, \quad s_m \in \mathbb{R}^+, u_m \in \mathbb{C}^{2N+1}. \tag{2.3}$$

There are different ways to compute Takagi factorizations. We will describe one method that relies on the eigendecomposition of a real matrix.

PROPOSITION 2.3. *Let A be a complex symmetric matrix and let*

$$B = \begin{pmatrix} \mathrm{Re}(A) & -\mathrm{Im}(A) \\ -\mathrm{Im}(A) & -\mathrm{Re}(A) \end{pmatrix}.$$

Let d_m and v_m denote the eigenvalue–eigenvector pairs of B. Then it is possible to order the eigenvalues such that they appear in pairs of the form $d_m = -d_{-m}$, $m = 1, \ldots, 2N+1$, with $d_m \ge 0$, $m = 1, \ldots, 2N+1$, decreasing. Using this ordering, we have that $d_m = s_m$, and that $u_m(j) = v_m(j) + iv_m(2N+1+j)$. The proof is given as an exercise in [7].

2.2. Hankel matrices. A *Hankel* matrix A have constant values on the anti-diagonals, i.e., it satisfies the relation

$$A(j,k) = A(j',k'), \quad \text{if } j+k = j'+k'.$$

Every Hankel matrix $A \in \mathcal{H}$ can thus be generated from some vector f, element wise by

$$A(j,k) = f(j+k), \quad -N \le j,k \le N.$$

We will use the notation Hf to denote this mapping from vectors to Hankel matrices. Below we elaborate about between which spaces it is natural to consider this mapping.

The basis for Hankel matrices in $\mathbb{M}_{2N+1,2N+1}$ is given by

$$e_l(j,k) = \begin{cases} \frac{1}{\sqrt{2N+1-|l|}}, & \text{if } j+k = l; \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

for $-2N \le l \le 2N$, where the normalization factor originates from the number of elements along anti-diagonal l.

When considering Hankel matrices in weighted spaces we need to use proper normalization; the basis elements should be normalized with respect to the induced (matrix) weights along the anti-diagonal. We associate the (positive weights)

$$\omega(l) = \sum_{j+k=l} w(j)w(k), \quad -2N \le l \le 2N, \tag{2.5}$$

to the weights w, and note that this can be written as a discrete convolution $\omega = \tilde{w} * \tilde{w}$, where $\tilde{w}$ denotes the zero padded version of w.

A basis for Hankel matrices in the weighted spaces $\mathbb{M}^w_{2N+1,2N+1}$ is then given by

$$e_l^w(j,k) = \begin{cases} \frac{1}{\sqrt{\omega(l)}}, & \text{if } j+k = l; \\ 0, & \text{otherwise.} \end{cases}$$

for $-2N \le l \le 2N$. Note that in the case where $w = 1$ we get the "triangle weight" $\omega(l) = 2N+1-|l|$ which appeared implicitly in (2.4). If we choose to consider the mapping H as a mapping from ℓ^ω_{4N+1} to $\mathbb{M}_{2N+1,2N+1}$, and choose

$$a_l^\omega(j) = \begin{cases} \frac{1}{\sqrt{\omega(l)}}, & \text{if } j = l; \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

as a basis for ℓ^ω_{4N+1} then $H : \ell^\omega_{4N+1} \to \mathbb{M}_{2N+1,2N+1}$ defined by $Ha^\omega_l = e^w_l$ is a linear isometry, whose adjoint is the weighted averaging operator

$$H^*A(l) = \frac{1}{\omega(l)} \sum_{j+k=l} w(j)A(j,k)w(k), \tag{2.7}$$

and $H^*H = I$.

PROPOSITION 2.4. *Let $w \in \mathbb{R}^{2N+1}_+$ be given and let ω be the associated weight defined by (2.5). Moreover, suppose that $f \in \ell^\omega_{4N+1}$ is given and let Hf be the associated Hankel matrix. For any subset $\mathcal{S}$ of Hankel matrices then the problem*

$$\operatorname*{argmin}_{\tilde{H}\in\mathcal{S}} \|Hf - \tilde{H}\|_w$$

is equivalent to the problem

$$\operatorname*{argmin}_{g\in H^*(\mathcal{S})} \|f - g\|_{\ell^\omega_{4N+1}}.$$

The solutions are related by $Hg = \tilde{H}$.

Proof. This is a direct consequence of the fact that $H : \ell^\omega_{4N+1} \to \mathbb{M}^w_{2N+1,2N+1}$ is an isometry. ∎

3. Properties of low-rank Hankel matrices. We denote the subset of Hankel matrices in $\mathbb{M}^w_{2N+1,2N+1}$ by $\mathcal{H} = \mathcal{H}^w_N$, and the subset of rank n complex symmetric matrices in $\mathbb{M}^w_{2N+1,2N+1}$ by $\mathcal{M}^n = \mathcal{M}^{w,n}_N$. To simplify notation, we will drop the size and weight parameter when it is clear from the context what the parameters are.

We recall [5] that we can conceptually describe this set as a submanifold of lower dimension than the actual space or manifold. In a linear space, this concept is stronger than having measure zero, and such sets are never encountered in practice. For matrices that are Hankel and at the same time have rank n, we have the following result, which can be seen as a version of Kronecker's theorem for finite matrices.

THEOREM 3.1. *(Kronecker's Theorem) The set of matrices Hf in $\mathcal{M}^n \cap \mathcal{H}$ of the form*

$$f(l) = \sum_{p=1}^{n} c_p e^{\zeta_p l}, \quad c_p, \zeta_p \in \mathbb{C}. \tag{3.1}$$

is an open set in $\mathcal{M}^n \cap \mathcal{H}$ with a thin complement. We will exploit this fact in order approximate functions by sums of n exponentials. We choose some positive weight w that gives rise to a weight ω through (2.5). The problem of approximating f by g as a sum of n exponential in ℓ^w is then according to Theorem 2.2 equivalent to finding the matrix $Hg \in \mathcal{M}^n \cap \mathcal{H}$ that minimizes $\|Hg - Hf\|_w$.

Let us turn of focus to how to find c_m and ζ_m in (3.1) given $Hf \in \mathcal{M}^n \cap \mathcal{H}$. If u is a vector that is orthogonal to all u_m, $m = 1, \dots, n$, then it is clear from (2.2) that $(Hf)u = 0$. From (3.1) we then have

$$\begin{aligned}(Hf)u(j) &= \sum_k f(j+k)u(k) = \sum_k \sum_p c_p e^{\zeta_p(j+k)}u(k) \\ &= \sum_p c_p e^{\zeta_p j} \sum_k u(k)e^{\zeta_p k} = \sum_p c_p e^{\zeta_p j} P_u(e^{\zeta_p}) = 0,\end{aligned}$$

Where P_u denotes the polynomial generated by u. Since the relation above holds for all j it follows that $P_u(e^{\zeta_p}) = 0$. We can thus find the nodes ζ_p by finding the roots to a polynomial P_u, where u is orthogonal to the con-eigenvectors $\{u_m\}_{m=1}^n$ in (2.2).

The roots of P_u can be computed by computing the eigenvalues of the corresponding companion matrix in $\mathcal{O}(N^3)$ time. If $N >> n$ then it seems unnecessary to compute all $2N$ roots when only n

is going to be kept. This is both a computationally expensive and a numerically sensitive method. One way to improve the speed of the computations is to consider a submatrix of Hf. It is easy to see from (2.2) that it is sufficient to consider $n+1$ sequel elements of each of the vectors u_m, e.g. $\mathbb{C}^{n+1} \ni \tilde{u}_m$ such that $\tilde{u}_m(j) = u_m(j)$, $j = 0, \ldots, n$, and find and find a $\tilde{u}$ such that $\langle \tilde{u}, \tilde{u}_m \rangle = 0$, $m = 1, \ldots, n$. The roots of the polynomial $p_u(z) = \sum_{j=0}^{n} u(j) z^j$ will then coincide with e^{ζ_p}. This approach will be substantially faster ($\mathcal{O}(n^3)$) than computing all $2N$ roots. However this approach does not have good numerical stability. The reason for this is that we are only local data, i.e only $n+1$ elements from each con-eigenvector u_m.

A better method is to observe that if f is a pure sum of n exponentials, then from the fact that the corresponding Hankel matrix can be expressed in the form (2.2), each con-eigenvector u_m will also be a sum of the same exponentials. Let $U = (u_1, \ldots, u_m)$. It is then possible to write $U = VG$, where $V \in \mathbb{M}_{2N+1,n}$ is the Vandermonde matrix generated by e^{ζ_p} ($V(l,p) = e^{\zeta_p l}$) and some (invertible) $G \in \mathbb{M}_{n,n}$. We will use the notation $U_1 \in \mathbb{M}_{2N,n}$ for U with the first row dropped, and similarly U_{2N+1} for the dropping of the last row. Clearly, we have that $U_1 = V_1 G$ and $U_{2N+1} = V_{2N+1} G$. We also note that

$$V_{2N+1} = V_1 \mathrm{diag}(e^{\zeta_1}, \ldots, e^{\zeta_n}).$$

From the relations above, it follows that

$$(U_{2N+1}^* U_{2N+1})^{-1} (U_{2N+1}^* U_1) = U_{2N+1}^{\dagger} U_1 = G^{-1} \mathrm{diag}(e^{\zeta_1}, \ldots, e^{\zeta_n}) G,$$

and hence that we can compute the nodes ζ_p by computing the eigenvalues of $U_{2N+1}^{\dagger} U_1$. This method is numerically stable and can be computed in $\mathcal{O}(Nn^2 + n^3)$ time.

Once the nodes ζ_m are found, the problem of find c_m becomes linear, and similarly as for the nodes case it will be sufficient to consider n elements in sequel (e.g. $l = 0, \ldots, n-1$), and solve the corresponding linear system.

4. The root–MUSIC and ESPIRIT methods. We briefly recapitulate the two most widely used methods for "high accuracy" frequency estimation. Our description will follow the implementation given in [16].

In the previous section we noted that we can find the nodes for a function f of the form (3.1), by considering the null space of the Hankel matrix that is generated from f. Recall that it was sufficient to consider any submatrix of a size larger than $n+1 \times n+1$ to accomplish the same thing. It is hence easily seen that we can find the nodes by considering a singular value decomposition of a rectangular Hankel matrix, also generated from f. Let $H^r f \in \mathbb{M}_{4N+2-M,M}$, with $M > n$, be such a Hankel matrix and let $U \Sigma V^* = H^r f$. Then the nodes in (3.1) can in principle be found by finding the roots of either of the polynomials generated by V_m, $n < m \leq M$. The vectors V_m are eigenvectors to the matrix

$$\begin{aligned}(H^r f)^* (H^r f)(j,k) = Rf(j,k) &= \sum_{l=-2N}^{2N+2-M} \overline{f(j+l)} f(k+l), \quad 0 \leq j,k < M, \\ &= \sum_{m=0}^{M-1} \overline{F}_m F_m^T, \end{aligned} \tag{4.1}$$

where $F_m \in \mathbb{M}_{4N+2-M,1}$ with $F_m(l) = f(l+m)$. The matrix Rf is sometimes referred to as the *sample covariance matrix*. It may seem to be beneficial to work with Rf instead of with the full Hankel matrices, since it is in principle possible to choose M much smaller than $2N+1$. It appears tractable that we make eigenvalue decomposition on a smaller matrix, and that the root finding step is also done with smaller matrices. However, just as discussed previously, a too small M can lead to numerical instabilities, even when f is purely a sum of n exponentials. Moreover, the matrix Rf needs to be computed. It is not hard to see that this can be achieved in $\mathcal{O}(N \log N + M^3)$ time by

splitting H^r into two parts and employing FFT. For large M this is not particularly advantageous. Another drawback is the loss of precision when forming $(H^r f)^*(H^r f)$.

The discussion so far has been under the assumption that (3.1) is valid. The typical situation is that this is only partially true, the standard assumption is that f contains additive noise as well. In the noiseless case, we did see that we had a great deal of flexibility, as any V_m, $n < m \leq M$ could be selected to find the nodes. The root-MUSIC method exploits this property, and tries to use all of the vectors V_m, $n < m \leq M$ to reduce the influence of noise. In the root–MUSIC method, roots are found by solving

$$P_{\mathrm{MUSIC}}(z) = \sum_{m=n+1}^{M} P_{\overline{V_m}}(z) P_{\check{V}_m}(z) = 0,$$

where $\check{}$ reverses the order of the elements in a vector. Loosely speaking, this choice is motivated by the facts that the roots will appear in pairs when f is a linear combination of purely oscillatory exponentials. There will be $2M-2$ roots to $P_{\mathrm{MUSIC}}(z) = 0$. The pairs associate with the true nodes, will have $\mathrm{Re}(\zeta) \approx 0$, with one slightly larger than zero and one slightly smaller. For a more detailed justification on the choice of P_{MUSIC}, cf [16].

In the general case, where there is no constraint on the nodes ζ_p, n roots needs to be selected out of the $2M-2$ that are given from $P_{\mathrm{MUSIC}}(z) = 0$. In the simulations performed in the later sections, we have used the MUSIC code provided in [16], and added a selection step where we approximate f using all $2M-2$ nodes using a least squares approach, and then selecting the n nodes with highest coefficients. It appears unnecessary to compute nodes that has to be thrown away.

The ESPIRIT method avoid the step of computing unnecessary nodes. Instead, a similar approach as section 3 is used. For the noiseless case, it is readily verified that the eigenvalues of

$$(V_M^* V_M)^{-1}(V_M^* V_1)$$

will coincide with the eigenvalues of e^{ζ_m}, $m = 1, \ldots, n$. In the ESPRIT method the eigenvalues of the expression above is used to compute nodes also in the case where noise is present, cf. [12]

We end this by section by a few remarks about the connection to autocorrelation and Toeplitz matrices. For a function of the form (3.1) where the exponentials are purely harmonic (zero real part of ζ_p), it holds that

$$\lim_{N\to\infty} \frac{1}{N} Rf$$

is the self adjoint Toeplitz matrix generated by the autocorrelation of f. According to a Theorem by Carathéodory [17], if the self adjoint Toeplitz matrix generated by a function has rank n, then that function can be expressed a sum of n purely oscillatory exponentials. This motivates alternating projection schemes between the manifolds of Toeplitz matrices and low rank matrices, for the approximation of the autocorrelation of a function. However, the effect of a finite sample length can not be neglected, and the Toeplitz matrix generated from the autocorrelation of a pure sum of n oscillatory exponentials, will fail to have rank n. For that sake, the formulation we have chosen seems preferable.

5. Alternating projections. Given $f \in \ell^\omega_{4N+1}$, the problem of finding the best approximation in ℓ^ω_{4N+1} of the form $g_{opt} = \sum_{k=1}^n c_k e^{\zeta_k l}$ is hard. Instead, our aim is to find a g that is close to optimal. We will do this by employing *alternating projections*. By Proposition 2.4 we know that this problem is equivalent to

$$\operatorname*{argmin}_{Hg \in \mathcal{M}^n \cap \mathcal{H}} \|Hf - Hg\|_w. \tag{5.1}$$

Note that $\mathcal{H}$ is a linear subspace and that $\mathcal{M}^n$ is a manifold in $\mathbb{M}^w_{2N+1,2N+1}$, at least locally except for a thin set (i.e. the set of matrices with non-zero singular values of multiplicity ≥ 1, see

[5]). Moreover, $\mathcal{M}^n \cap \mathcal{H}$ is a manifold as well except for a thin set, and a chart for this manifold is given by Theorem 3.1. By alternatively projecting onto the subsets $\mathcal{M}^n$ and $\mathcal{H}$, the idea is than the so arising sequence Hg_k will converge to an intersection point Hg_∞, and moreover that Hg_∞ is in fact close to the optimal one, Hg_{opt}. This idea was investigated in a general framework in [5]. The main result of [5] roughly says that the above scheme indeed works if we start not too far away from $\mathcal{M}^n \cap \mathcal{H}$ and avoid the thin set of bad points related to $\mathcal{M}^n$ and $\mathcal{M}^n \cap \mathcal{H}$, (which in practice does not seem to be an issue). As an example we studied the case of projections between rank n matrices and Hankel matrices in non-weighted spaces. In this paper we elaborate on the approximation of function by exponentials in weighted spaces and how to construct fast algorithms to this end.

We now state the main result of [5] in the current framework. Let $P_{\mathcal{M}^n}$, $P_{\mathcal{H}}$ and $P_{\mathcal{M}\cap\mathcal{H}}$ denote the maps taking a given matrix B onto the closest point in the respective manifolds. Already here we hit some technical issues. We clearly have a formula for $P_{\mathcal{H}}$ since $\mathcal{H}$ is linear, so $P_{\mathcal{H}}$ is an orthogonal projection and explicit formula is given by (2.7). Concerning $P_{\mathcal{M}^n}$ we do have a formula for computing it, with the only head-ache that if B has singular values of higher multiplicity, then the map is not well defined. This is a common feature in algorithmic frameworks, and can be dealt with by introducing point-to-set maps, following [20]. However, this seems silly in the current framework since matrices with singular values of multiplicity ≥ 1 constitute a thin set, and is never encountered in practice. Moreover, in [5] we prove that both $P_{\mathcal{M}^n}$ and $P_{\mathcal{M}^n\cap\mathcal{H}}$ are well defined near "decent" points of $\mathcal{M}^n \cap \mathcal{H}$. With this in mind, we will from now on treat $P_{\mathcal{M}^n}$ and $P_{\mathcal{M}^n\cap\mathcal{H}}$ as well defined maps. Note that we do not have any formula for computing $P_{\mathcal{M}^n\cap\mathcal{H}}$, since then the current paper would be completely useless. Below follows a reformulation of the main result in [5] in the current setting.

THEOREM 5.1. *For all $A \in \mathcal{M}^n \cap \mathcal{H}$ except a thin subset, the following is true. Given any $\epsilon > 0$, there exists a $s > 0$ such that, for all Hf with $\|Hf - A\| \leq s$, the sequence of alternating projections given by $B_0 = Hf$ and*

$$B_{k+1} = \begin{cases} P_{\mathcal{M}^n}(B_k) & k \text{ is even} \\ P_{\mathcal{H}}(B_k) & k \text{ is odd} \end{cases} \tag{5.2}$$

(i) converges to a point $Hg_\infty \in \mathcal{M}^n \cap \mathcal{H}$
(ii) $\|Hg_\infty - Hg_{opt}\| \leq \epsilon \|Hf - Hg_{opt}\|$

A few remarks: (i) combined with Theorem 3.1 thus says that we will achieve an approximation of f of the form $g_\infty = \sum_{k=1}^n c_k e^{\zeta_k l}$. Moreover, note that if we had 0 on the right hand side of (ii), then $g_\infty = g_{opt}$. (ii) says that the error $\|g_\infty - g_{opt}\|_{\ell^2_\omega}$ can be made arbitrarily small relative to the distance $\|f - g_{opt}\|_{\ell^2_\omega}$. Finally, the full theorem in [5] has a third post, but to define this we need to discuss angles between manifolds, which we like to avoid. Basically, the third post says that there exists a number $0 < c < 1$, whose lower bound is related to the angle between $\mathcal{M}^n$ and $\mathcal{H}$ at A, such that

$$(iii)\ \|Hg_\infty - B_k\| < c^k \|Hf - Hg_{opt}\|.$$

For practical purposes, this is an important observation, since it says that the algorithm converges fast, i.e., it has c-linear convergence.

Let us now briefly discuss what happens if we are not close enough to $\mathcal{M} \cap \mathcal{H}$ that the above theorem kicks in. First of all, we have never encountered that the algorithm does not converge. Second of all, it is easy to see that both $P_{\mathcal{H}}$ and $P_{\mathcal{M}^n}$ are contractions, so $(B_k)_{k=0}^\infty$ is a bounded sequence. It then has a convergent subsequence by basic properties of compact sets. Moreover, the distance to $\mathcal{M}^n \cap \mathcal{H}$ is strictly decreasing, and hence the limit point of the convergent subsequence is in $\mathcal{M}^n \cap \mathcal{H}$. (However, there is of course no indication that the corresponding g_∞ is at all close to g_{opt}, so this is a rather pointless observation.) In literature treating similar topics as this article, one usually satisfies with concluding that the algorithm in question has the property that it generates a sequence with a convergent subsequence having a limit point in the desired set, and attributes this to Zangwill's theorem, [20]. Clearly, Theorem 5.1 provides much more information in our setting;

every point in $\mathcal{H} \cap \mathcal{M}^n$, except some thin subset, has a neighborhood such that, if any B_k enters that neighborhood, the sequence $(B_k)_{k=0}^{\infty}$ will converge. Since the sequence needs to have more than one accumulation-point if it does not converge, the only possibility for convergence not to take place is if $(B_k)_{k=0}^{\infty}$ wanders back and forth along the valleys of the thin pathological set, between the hills constituting of the open set formed by all nice neighborhoods mentioned above. This seems highly unlikely to the authors, but we leave it as an open question to rule out this possibility. Clearly, it would be interesting to have some concrete values of the parameters ϵ and s in Theorem 5.1, we will return to this issue in what follows.

Below is an algorithm that specifically describes how to apply the alternating projection scheme our case.

ALGORITHM 1.

1. *Let $g_0 = f$, $l = 0$*
2. *(Application of $P_{\mathcal{M}^n}$) Compute the n first con-eigenvalues s_m and the vectors u_m of Hg_l using Theorem 2.2. The projection $P_{\mathcal{M}^n} Hg_l$ is then given*

$$\sum_{m=0}^{n} s_m u_m u_m^T. \tag{5.3}$$

3. *(Application of $P_{\mathcal{H}}$) Compute*

$$g_{l+1} = H^* \left(\sum_{m=1}^{n} s_m u_m u_m^T \right)$$

4. *Increase l and repeat from (2) until convergence.*

6. Fast algorithms. There are two operations for which we will need fast numerical methods in the alternating projections approach for frequency detection; (low rank) con-eigenvalue decomposition, and the application of the averaging operator (5.3). It turns out that both operations can be implemented in a fast manner, but the first one will require some more effort than the second.

PROPOSITION 6.1. *The application of a Hankel matrix to a vector can be done in $\mathcal{O}(N \log(N))$ time by means of FFT.*

Proof. This is a standard result [6], and makes use of the fact that circular matrixes are diagonalized by the discrete Fourier transform, and that it easy to construct a circular matrix from a Hankel matrix by permutation and periodic extension. The $\mathcal{O}(N \log(N))$ time complexity can then be achieved by employing FFT. □

PROPOSITION 6.2. *The weighted averaging operator H^* in (5.3) can be applied to a rank 1 matrix in $\mathcal{O}(N \log(N))$ time.*

Proof. By definition

$$\begin{aligned} H^*(uu^T)(l) &= \frac{1}{\omega(l)} \sum_{j+k=l} w(j)u(j)u(k)w(k), \quad -2N \leq l \leq 2N \\ &= \frac{1}{\omega(l)} \sum_{j+k=l} v(j)v(k), \quad -2N \leq l \leq 2N, \end{aligned}$$

where $v = wu$. It is easy to see that the sum above can we written as a discrete convolution $(v(j)v(l-j))$ using zero padding to avoid boundary effects. The discrete convolutions can then be computed in $\mathcal{O}(N \log(N))$ time using FFT. □

6.1. Lanczos method for complex symmetric matrices. We will use a modified Lanczos method for finding the n first con-eigenvalues/con-eigenvectors. The Lanczos method is a way to do a unitary transformation of a symmetric matrix to tridiagonal form, i.e., given $A = A^*$, compute $T = Q^*AQ$, where Q is unitary and $T = T^*$ is tridiagonal. We need a similar decomposition

for complex symmetric matrices. The usage of a modified Lanczos method has been addressed in [8, 18, 15]. As we only need to compute the first n con-eigenvalues/con-eigenvectors, we develop a method tailored for that purpose.

The basic step in the Lanczos method is simple. However, it is notorious for the loss of precision, sometimes in a counterintuitive way. This issue must be addressed carefully. The columns in the unitary matrix Q are computed sequentially, in such a way that each new column is automatically orthogonal to all previous ones. In practice, finite numerical precision can ruin the orthogonality, and it can be completely lost in within just a few steps. Two methods that address this are *selective orthogonalization* [9] and *partial orthogonalization* [14]. We will make use of ingredients out of both these methods in our particular setup.

We seek decompositions of the form

$$T = Q^T A Q, \tag{6.1}$$

for a given $A \in \mathbb{M}_{2N+1,2N+1}$, $\mathbb{M}_{2N+1,2N+1} \ni Q = (q_1, \ldots, q_{2N+1})$ with ortonormal columns q_j, and $T \in \mathbb{M}_{2N+1,2N+1}$ of the form

$$T = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \ldots & 0 & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \ldots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & \alpha_{m-1} & \beta_{m-1} \\ 0 & 0 & 0 & \ldots & \beta_{m-1} & \alpha_m \end{pmatrix}. \tag{6.2}$$

There are many such decompositions. In order to investigate how to construct the α and β vectors, write $Q_m = (q_1, \ldots, q_m)$ and let T_m denote the $m \times m$ upper left submatrix of T. From the ortogonality of the columns of Q it then follows that $AQ = \overline{Q}T$, and

$$AQ_m = \overline{Q}_m T_m + \beta_m e_m^T q_{m+1} = \overline{Q}_m T_m + e_m^T r_{m+1}, \tag{6.3}$$

where $\mathbb{R}^m \ni e_m = (0, \ldots, 1)$. The latter equality can be formulated as

$$Aq_m - \overline{q}_m \alpha_m - \overline{q}_{m-1}\beta_{m-1} = \beta_m \overline{q}_{m+1} = r_{m+1}, \tag{6.4}$$

which provides an iterative scheme for computing β_m, q_{m+1} and a_{m+1};

$$\begin{aligned} \alpha_m &= q_m^T A q_m; \\ \beta_m &= \|r_{m+1}\|; \\ q_{m+1} &= \overline{r}_{m+1}/\beta_m, \quad \text{if } \beta_m \neq 0. \end{aligned}$$

The two last equation are immediate from $r_{m+1} = \beta_m q_{m+1}$ and the ortonormality condition $\|q_m\| = 1$. It is natural to choose β_m to be real valued. The assignment for α_m follows from the orthogonality constraints for q_{m+1}. To see this we check the orthogonality conditions between q_{m+1} and q_j, $j \leq m$,

$$\begin{aligned} &q_j^T r_m = \beta_m q_j^T q_{m+1} = \\ &q_j^T A q_m - \beta_{m-1} q_j^T \overline{q}_{m-1} - \alpha_m q_j^T \overline{q}_m = \begin{cases} q_j^T A q_m = T_m(j,m) = 0, & \text{if } j < m; \\ q_m^T A q_m - \alpha_m, & \text{if } j = m. \end{cases} \end{aligned}$$

Given that all $\beta_m \neq 0$, this procedure uniquely defines a decomposition of the form (6.1). For the case when $\beta_m = 0$ (or very small) for some m, we have that Q_m is an invariant subspace of A. Given a randomly selected starting vector q, this will (generically) only happen when A has rank m.

The immediate application of the decomposition (6.1) is that we can compute con-eigenvector and con-eigenvalues for T instead of for A (cf. discussion about con-similarity [7, p244,p251]), and due to the tridiagonal structure of T it may be beneficial. However, there is an additional important reason for why a Lanczos procedure is useful; it turns out that it is often sufficient to consider (6.3) and computing the coneigenvalues and eigenvectors of T_m instead of doing the full decomposition (6.1) and a coneigenvalue decomposition of T.

LEMMA 6.3. *Let $A = A^T$ be given and let T_m be the tridiagonal matrix in (6.3). Denote the con-eigenvectors of T_m by u_j and the corresponding con-eigenvalues by μ_j, $1 \leq j \leq m$. Then, for each j, there is a con-eigenvalue λ_j of A such that*

$$|\mu_j - \lambda_j| \leq \beta_m |u_j(m)|. \tag{6.5}$$

Proof. Consider the matrix product of u_j and the residual matrix $e_m^T r_{m+1}$ in (6.3)

$$\|AQ_m u_j - \overline{Q}_m T_m u_j\| = \|AQ_m u_j - \mu_j \overline{Q_m u_j}\| = \|e_m^T r_{m+1} u_j\| = \beta_m |u_j(m)|,$$

and introduce $w_j = Q_m u_j$. Since $\|u_j\| = 1$, it follows that $\|w_j\| = 1$. Denote the con-eigenvectors of A by v_l, and represent $w_j = \sum_l s_l v_l$, $\sum_l |s_l|^2 = 1$. We then have

$$\begin{aligned}\|AQ_m u_j - \overline{Q}_m T_m u_j\|^2 = \|Aw - \mu\overline{w}\|^2 = \|\sum_l s_l(\lambda_l - \mu_j)\overline{v}_l\|^2 \\ \geq \min_l |\lambda_l - \mu_j|^2 \sum_l |s_l|^2 = \min_l |\lambda_l - \mu_j|^2.\end{aligned}$$

□ This is a well known result for the case of Hermitian symmetry, see for instance [10, p. 69]. A similar result is given in [15, Proposition 2.2].

Lemma 6.3 provides a way to control the convergence of con-eigenvectors. When the quantities in (6.5) are small, then w_j will be a good approximation of the con-eigenvector to A that is associated with λ_j. In many cases, convergence for the first con-eigenvalues are reached for comparatively small m. In particular, for the case where the (con)spectrum of A has a large gap after, say n terms, it is typically only necessary to use m slightly larger than n. This will be the case for all but the first step in our alternating projection algorithm.

As mentioned before, in a straightforward Lanczos implementation the orthogonality of Q will quickly be lost due to finite precision arithmetics. Moreover, and somewhat counterintuitive, the loss of orthogonality will grow as the con-eigenvectors converge, cf. [9]. The simple remedy to this problem is to reorthogonalize q_{m+1} to all previous q_j at each iteration. However, this increases the algorithmic complexity of the method. Instead, we want to have a criterium on when reortogonalization is needed. The loss of orthogonalization is also indicative of con-eigenvalue converge, and hence reason for checking if the conditions of Lemma 6.3 is satisfied.

Two suggestions on reorthogonalization criteria are given in [14, 9]. We will follow the approach given in [14]. Since we are working with con-eigenvalues and con-eigenvectors instead eigenvalues and eigenvectors, we briefly provide the details.

Due to finite precision arithmetic, we model (6.4) as

$$\beta_m \overline{q}_{m+1} = Aq_m - \overline{q}_m \alpha_m - \overline{q}_{m-1}\beta_{m-1} + \epsilon_m, \tag{6.6}$$

where ϵ_m describes the error introduced by the finite precision. We now let q_j denote the vectors *computed* from the relation (6.6). Due to the errors ϵ_m, these vectors will not be orthogonal. Let $\omega_{j,k} = q_j^* q_k$. Then $\omega_{j,k}$ will satisfy the recursion relation

$$\begin{aligned}&\omega_{m+1,m+1} = 1, \quad \omega_{m+1,m} = q_{m+1}^* q_m = \psi_{m+1}, \\ &\omega_{m+1,j} = \frac{1}{\beta_m}\Big(\alpha_j\overline{\omega}_{m,j} + \beta_{j-1}\overline{\omega}_{m,j-1} + \beta_j\overline{\omega}_{m,j+1} - \alpha_m\omega_{m,j} - \beta_{m-1}\omega_{m-1,j}\Big) + \vartheta_{m,j},\end{aligned} \tag{6.7}$$

where $\vartheta_{m,j} = \beta_m^{-1}(q_j^T \epsilon_m - q_m^T \epsilon_j)$. The last equality follows from multiplying (6.6) by q_j^T, and subtracting the same quantity with the indices j and m interchanged. Since $A = A^T$ the quantity $q_j^T A q_m$ then cancels.

Using the recursion formula above, we can monitor the level of lost orthogonality without explicitly having to compute inner products of the columns of Q. In analogy with the empirical results in [14, 9], we simulate the error quantities as

$$\vartheta_{m,j} \in \mathbf{N}\left(0, 0.3\ \epsilon(\beta_m + \beta_j)\right),$$
$$\psi_{m+1} \in \mathbf{N}\left(0, 0.6\ \epsilon(2N+1)\frac{\beta_1}{\beta_m}\right),$$

where $\mathbf{N}(0, \sigma)$ denotes the complex normal distribution with standard deviation σ, zero mean and independent real and imaginary parts. Above, ϵ denotes the machine precision.

The maximum loss of precision that can be tolerated without loss of precision in the coefficients α_{m+1} and β_m is $\sqrt{\epsilon}$. Once some $\omega_{m+1,j}$ exceeds that level, it is necessary to reorthogonalize. As seen in (6.7), each $\omega_{m+1,j}$ is strongly influenced by its neighbors. Hence, it will not be efficient to only reortogonalize against the vectors q_j where $\omega_{m+1,j}$, since for isolated $j's$ the ortogonalization would immediately get lost in the next iteration. Instead, it is beneficial to reorthogonalize against a batch of q_j's. Hence (and in accordance with [14]) we reortogonalize against the set of q_j which have $|\omega_{m+1,j}| > \epsilon^{3/4}$ once $|\omega_{m+1,j}| > \sqrt{\epsilon}$ for some j.

After a reortogonalization has taken place, we need to reset the quantities $\omega_{m+1,j}$. Again following [14], we choose $\omega_{m+1,j} \in \mathbf{N}(0, 1.5\epsilon)$.

The final ingredient is a rule for when to utilize Lemma 6.3 for convergence monitoring of con-eigenvalues. Clearly $m \geq n$ in order to find n con-eigenvalues that has converged. Since the convergence of con-eigenvalues and the loss of orthogonality are coupled, we compute a Takagi factorization of T_m once loss of orthogonality is indicated by $|\omega_{m+1,j}| > \sqrt{\epsilon}$ for some j, given that $m \leq n$. Moreover, we can monitor the behavior of β_m to check for convergence. If β_m becomes very small for some m then it means that Q_m defines an almost invariant (con)subspace under A, which implies convergence of the (non-zero) con-eigenvalues. As always with numerical implementations, it can be difficult to determine how small β_m has to be in order to consider it to have almost vanished. What typically happens when the Lanczos procedure goes too far, is that the last value β jump dramatically in size. This behavior also serves as a good criterium for when to check for convergence by means of Lemma 6.3.

In the procedure above, we need to compute the Takagi factorization of T_m. The cost of that step when using Proposition 2.3 is $\mathcal{O}(m^3)$. However, due to the tridiagonal structure there are methods to compute this in $\mathcal{O}(m^2)$ time, cf. [8, 19, 18]. These methods are based on straight forward modifications of methods for eigenvalue decomposition of tridiagonal Hermitian matrices.

The most expensive step in the Lanczos procedure described above is the matrix vector multiplication Aq_m in (6.4). However, this step can be computed in $\mathcal{O}(N \log N)$ time by Proposition 6.1.

PROPOSITION 6.4. *The time complexity for computing the first n con-eigenvectors and con-eigenvalues of a Hankel matrix to accuracy ϵ using the modified Lanczos method described above is $\mathcal{O}(mN \log N + m^2)$, where m denotes the total number Lanczos steps, and where $m \geq n$, but where m is typically of the same order as n.*

7. Numerical simulations.

7.1. Performance analysis. In this section we compare the performance of our approach against the ESPRIT and root-MUSIC methods. We simulate functions of the form $f = f_0 + n$, where

$$f_0(k) = \sum_{p=1}^{n} c_p e^{\zeta_p k},$$

and where n is a noise component. The coefficients c_p is chosen as complex normal distributed variables, and the nodes $\zeta_p = 1/(4N+1)(50Z_p^r + iZ_p^i)$, where Z_p^r and Z_p^i are normally distributed.

The noise component is constructed by letting $\tilde{n}(k) = n_r(k) + in_i(k)$, where n_r and n_i are normally distributed noise, and where

$$n = \sqrt{\frac{\|f_0\|^2}{\|\tilde{n}\|^2} 10^{-\mathrm{SNR}/10}}\tilde{n},$$

for some *signal to noise* parameter SNR. By this construction, the signal to noise ratio will be exactly equal to the parameter SNR when measured in dB. Throughout the tests, we have chosen to work with a signal length of 509, i.e. $N = 127$. This is chosen such that FFT routines should run fast. All simulations has been run in a MATLAB environment, without any compiled optimizations. For the ESPRIT and root-MUSIC, we have used the routines provided in [16], with minor modifications to make them work for the case where $\mathrm{Re}(\zeta_p) \neq 0$. The accuracy parameter ϵ used in the alternating projection method has been chosen to be a factor 100 lower than the noise magnitude.

In figure 1, we show some simulation results for the different methods. We conduct a small number of simulations for $SNR = 10\ dB$ and $n = 10$, and consider the performance in terms of the errors generated by the different methods. We display the errors are displayed in two ways; in

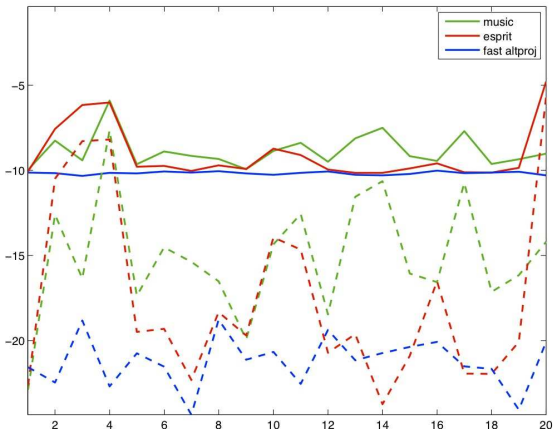

FIG. 1. *The solid curves show the error $\|g - f\|/\|f\|$ with g obtained with MUSIC (green), ESPRIT (red) and our propose method (blue), respectively, for SNR=10. The dashed ones show the counterparts for $\|g - f_0\|/\|f_0\|$.*

relation to the pure signal f and in relation to the noise one f_0.

We see that our proposed method systematically has a lower error, in both ways of measurement. We also note that for all methods we have substantially lower error when compared to the pure signal f_0 instead of the noisy one. Hence, all three methods successfully filter out a large part of the noise. It is also notable how close the error in relation to f is to the signal to noise ratio for our proposed method. This is also implied by Theorem 5.1. Basically, in the notation of Section 5, we have $g = g_0$, and it is reasonable to assume that $f_0 \approx g_{opt}$. This is because the noise has a high probability of being orthogonal to f_0, and that $P_{\mathcal{M}^n \cap \mathcal{H}}$ locally acts as an orthogonal projection, (which is further elaborated on in [5]). Thus, Figure 1 can be interpreted as the upper blue line shows $\|f - g_{opt}\|/\|f\|$, whereas the lower blue line gives an indication of the size of $\|g_\infty - g_{opt}\|/\|f\|$. In terms of Theorem 5.1 with $A = Hf_0$ of norm 1 and $s = 0.1$, this means that we can pick ϵ around 0.1 as well. Although the above images are constructed using standard ℓ^2-norm, not the weighted one required for Theorem 5.1 to kick in, it is interesting to observe that this is in line with the observations in [5]. There, using more carefully conducted examples to test Theorem 5.1, it seems that one can take $s \approx \epsilon$ when working with $n \approx 10$.

It is interesting to see how these result depend on the different parameters, i.e., the number of nodes n and the noise level SNR. In Figure 2 we have conducted more thorough investigations. For each $n = 1, 2, \ldots, 30$ we have done 100 simulations and computed average results. The averaging has been made in dB, in order to limit the effect of outlier results. As for Figure 1, we display errors

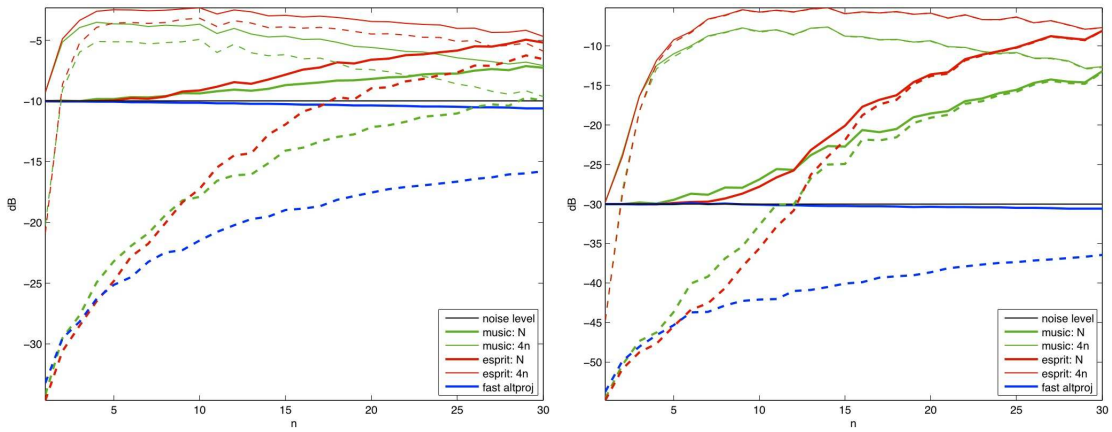

FIG. 2. *The left panel show results for SNR=10, and the right one results for SNR=30. The solid curves show the dependence on n of the average error* $\|g - f\|/\|f\|$ *with g obtained with MUSIC (green), ESPRIT (red) and our propose method (blue), respectively, for SNR=10 . The dashed ones show the counterparts for* $\|g - f_0\|/\|f_0\|$. *The average for each n is done over 100 simulations. The thin lines show the results obtained for MUSIC and ESPRIT for* $M = 4n$, *whereas the thick lines show results for* $M = N$.

in two ways, using solid lines for errors in comparison to the noise signal f and dashed lines for the comparison to the original one, f_0. We also display some of impact the choice of the size (M) of R in (4.1) has. The thin lines in red and green show errors for $M = 4n$ and the thick lines show the counterpart for $M = N = 127$.

There are a few interesting conclusions that can be drawn from the results depicted in Figure 2. First, we note that for the cases were n is low, all three methods perform comparably well, given that the size (M) of the sample covariance matrix used in MUSIC and ESPRIT is sufficiently large. However, as n increases, the alternating projection method starts to outperfom the other two. We can again note that the errors (compared to f) produced by the alternating projection method almost coincide with the signal to noise ratio. Moreover, in terms of Theorem 5.1, Figure 2 seems to indicate that $\epsilon \approx s$ is a good rule of thumb, although the ratio gets slightly worse as the complexity of the manifold $\mathcal{M}^n \cap \mathcal{H}$ increases with increasing n.

From the results we have seen so far we can conclude that the alternating projection method should be the method of choice unless n is very low, given that the prime concern is to minimize the estimation errors. The other criteria for method selection is speed. The computational times for the different methods is displayed in Figure 3. As mentioned before, the MUSIC and ESPRIT algorithms that are used are slightly modified versions of the ones given in [16]. In Figure 3 the fast alternating projection method is the fastest. The MUSIC and ESPRIT algorithms are substantially slower for high M. For the MUSIC algorithm, the most time consuming step is the root solving step. We note, however, that by using our the fast method for finding the first n con-eigenvector / con-eigenvalues, we can construct a method that would have much resemblance with the ESPRIT method as described in [12]. It seems to be advantageous to work directly with Hankel matrices rather than the covariance sample matrix of (4.1). Using such an approach, we would be able to construct a computational method that would provide similar results as the ESPRIT algorithm described in [16], but substantially faster than the one based on the sample covariance matrix. From the results in Figure 2 we can conclude that the results would not be as good as the ones obtained by the fast alternating projection method proposed here. However, it would be faster, as it would

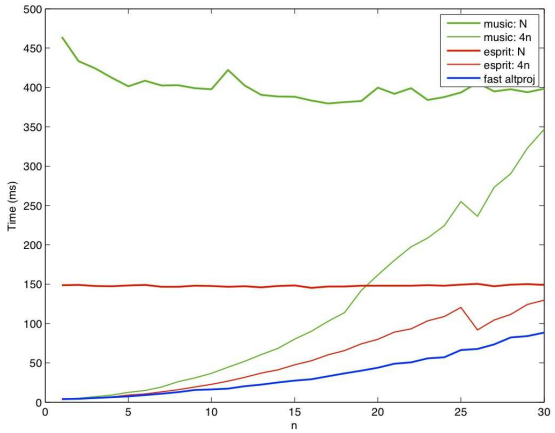

FIG. 3. *Average execution time for the different methods in milliseconds. The line notation as in Figure 2 is used.*

only involve one decomposition step. In other words, it would be equivalent to using the alternating projection scheme with only one iteration.

A natural question would then be how much faster such "fast" ESPRIT algorithm would be. A first guess would be that the speed ratio would be proportional to the number of alternating projections done before the target accuracy ϵ is reached. It turns out that the fast alternating projection method is faster than that. The reason for this is that fewer Lanczos iterations are required in each alternating projection iteration. In Figure 4 we display the ratio between the total

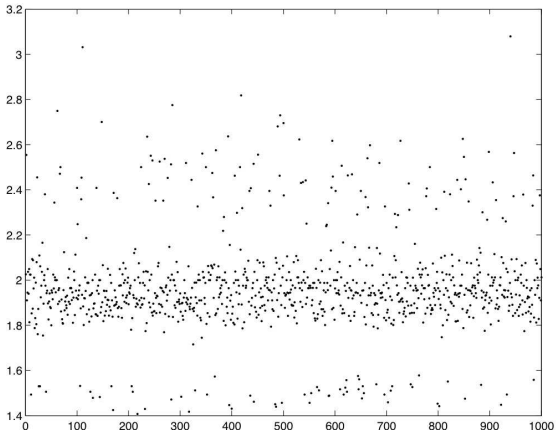

FIG. 4. *The ratio between the time for total number of iterations compared to the first one for the alternating projection method for 1000 simulations*

time and the time for the first iteration for the alternating projection method. We see that the ratio typically lies around 2. This means that the fast alternating projection method would only be about twice as expensive as a fast implementation of ESPRIT, while providing smaller errors. Again, we note that the proposed fast alternating projection method is substantially faster than the standard implementation of ESPRIT and MUSIC.

7.2. Approximations with Gaussians. As a final example, we show some results concerning the approximation of functions with Gaussians with fixed half-width, using the fast alternating projection method with Gaussian weights. There are two possible interesting cases. The first one

concerns the case where the functions are of the form

$$\sum_p c_p \mathrm{e}^{-\alpha(x-x_p)^2+\mathrm{i}\xi_p x}, \tag{7.1}$$

with fixed (and known) constant α. The second case concerns the approximation of functions using a Gaussian window, for example as done in time–frequency analysis. Using a non-linear approach may be beneficial compared to short-time Fourier transform representations with overlapping windows. However, we will in this section only show some results concerning (7.1).

In Figure 5 we show the result from one simulation using a function of the form (7.1), using 10 Gaussians. In order to approximate this function using exponentials, we choose the weights w such that $\sqrt{\omega_l}$ approximates $\mathrm{e}^{-\alpha l/2N}$, $l = -2N, \ldots, 2N$. To this end, we choose

$$w_k = \sqrt{\frac{2}{4N+1}\sqrt{\frac{8\alpha}{\pi}}}\mathrm{e}^{-4\alpha k/4N}, \quad k = -N, \ldots, N.$$

For sufficiently narrow Gaussians (large α), we will then have that $\sqrt{\omega_l} \approx \mathrm{e}^{-\alpha l/2N}$. Just as before

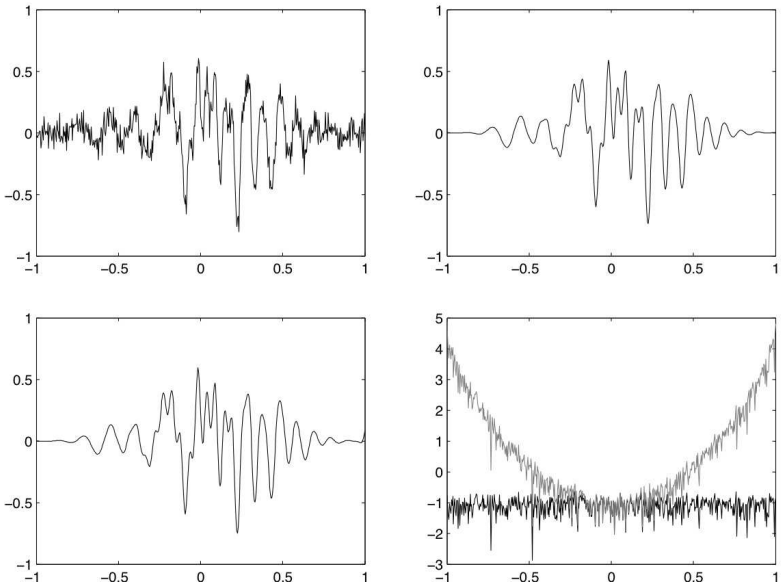

FIG. 5. *In the top left panel the noisy signal f is shown, the top right shows the original f_0, the bottom left shows the reconstruction, and the bottom right shows errors. The errors are shown unscaled in gray, and scaled with respect to $\sqrt{\omega}$ in black.*

.

we let f_0 be of the form (7.1) and use additive noise to obtain f. One simulation is shown in Figure 5 for $SNR = 10$. Before we start the alternating projection scheme, we divide f pointwise with $1/\sqrt{\omega_l}$. This will boost the amplitude at the endpoints of f substantially, but since we approximate using ω_l as a weight, we will obtain a uniform approximation. The noise will, however, not be uniform with this approach, but larger at the end-points.

The result from one simulation is shown in Figure 5. The noise signal is depicted in the top left panel, while the original is displayed in the top right panel. In the bottom left we see the obtained reconstruction. We can see that most of the features from the original signal is captured. In the bottom right panel we show pointwise errors; in black the error weighted with ω_l and in grey the unweighted pointwise error.

8. Conclusions. We have developed a method for the fast estimation of complex frequencies using an alternating projection scheme between Hankel matrices and rank n Takagi representations. The method has a time complexity of $\mathcal{O}(nN \log N + n^3)$. FFT routines are used both to get fast matrix–vector multiplications, and to project rank n representations to Hankel matrices. In order to compute the first n Takagi vectors, we employ a modified Lanczos scheme for complex symmetric matrices. The number of necessary alternating projection steps depends on an accuracy parameter, but in typical situations the total time is only a factor two larger than the time for the first iteration. The reason for that is that fewer Lanczos steps are needed as we the matrix we obtain is closer to being both Hankel, and rank n.

In our simulations we see that the proposed method performs better both with regards to speed and approximation accuracy, compared to standard implementations like root-MUSIC and ESPRIT. We also verify that the errors that we obtain behave in the same manner as theoretically predicted in [5]. The method works also for some weighted representations. A particular case of weights that can be used are Gaussian weights, for which case some numerical examples are provided.

REFERENCES

[1] V. M. Adamjan, D. Z. Arov, and M. G. Kreĭn. Infinite Hankel matrices and generalized problems of Carathéodory-Fejér and F. Riesz. *Funkcional. Anal. i Priložen.*, 2(1):1–19, 1968.

[2] Fredrik Andersson, Marcus Carlsson, and Maarten V de Hoop. Sparse approximation of functions using sums of exponentials and aak theory. *Journal of Approximation Theory*, 163(2):213–248, February 2011.

[3] Gregory Beylkin and Lucas Monzón. On approximation of functions by exponential sums. *Applied and Computational Harmonic Analysis*, 19(1):17–48, July 2005.

[4] G. Bienvenu. Influence of the spatial coherence of the background noise on high resolution passive methods. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '79.*, volume 4, pages 306 – 309, April 1979.

[5] Marcus Carlsson and Fredrik Andersson. Alternating projections on low-dimensional manifolds. *In progress.*

[6] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[7] Roger A. Horn and Charles R. Johnson. *Topics in matrix analysis*. Cambridge University Press, Cambridge, 1994. Corrected reprint of the 1991 original.

[8] Franklin T. Luk and Sanzheng Qiao. A fast eigenvalue algorithm for hankel matrices. *Linear Algebra Appl*, 316:171–182, 1998.

[9] B. N. Parlett and D. S. Scott. The Lanczos algorithm with selective orthogonalization. *Math. Comp.*, 33(145):217–238, 1979.

[10] Beresford N. Parlett. *The symmetric eigenvalue problem*. Prentice-Hall Inc., Englewood Cliffs, N.J., 1980. Prentice-Hall Series in Computational Mathematics.

[11] V. F. Pisarenko. The retrieval of harmonics from a covariance function. *Geophysical Journal of the Royal Astronomical Society*, 33(3):347–366, 1973.

[12] R. Roy and T. Kailath. ESPRIT-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):984–995, 1989.

[13] R. Schmidt. Multiple emitter location and signal parameter estimation. *Antennas and Propagation, IEEE Transactions on*, 34(3):276 – 280, March 1986.

[14] Horst D. Simon. The Lanczos algorithm with partial reorthogonalization. *Math. Comp.*, 42(165):115–142, 1984.

[15] V. Simoncini and E. Sjöström. An algorithm for approximating the singular triplets of complex symmetric matrices. *Numerical Linear Algebra with Applications*, 4(6):469–489, 1997.

[16] Petre Stoica and Randolph Moses. *Introduction to spectral analysis*. Prentice–Hall, 1997.

[17] G. Szegö. *Orthogonal Polynomials*. AMS, Providence, RI, 1975.

[18] Wei Xu and Sanzheng Qiao. A fast symmetric SVD algorithm for square Hankel matrices. *Linear Algebra Appl.*, 428(2-3):550–563, 2008.

[19] Wei Xu and Sanzheng Qiao. A twisted factorization method for symmetric SVD of a complex symmetric tridiagonal matrix. *Numer. Linear Algebra Appl.*, 16(10):801–815, 2009.

[20] W. I. Zangwill. *Nonlinear Programming*. Prentice Hall, Englewood Cliffs, N. J., 1969.