ESPRIT for multidimensional general grids.

Fredrik Andersson, Marcus Carlsson

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

Abstract

We present a new method for complex frequency estimation in several variables, extending the classical (1d) ESPRIT-algorithm. We also consider how to work with data sampled on non-standard domains (i.e. going beyond multi-rectangles).

Keywords: complex frequency estimation, Hankel, finite rank, Kronecker theorem, sums of exponentials

2010 MSC: 15B05, 41A63, 42A10.

1. Introduction

The ESPRIT-method [13] for (complex) frequency estimation is probably the most widely used, known for both its robustness and simplicity. Given equidistant measurements of a signal \( f \) and a “model order” \( K \), it seeks to approximate \( f \) by an exponential sum

\[
f(x) \approx \sum_{k=1}^{K} c_k e^{\zeta_k x}, \quad c_k, \zeta_k \in \mathbb{C},
\]

and also returns values of the parameters \( \zeta_k \). (The subsequent retrieval of the coefficients \( c_k \) is a linear problem which can be solved by the least squares method, also in the multi-dimensional setting.) In the noise free case, i.e. when \( f \) already is of the form given by the right hand side of (1.1), the method is exact, giving it a substantial advantage over methods working with predefined frequency grids.

Another exact method in the noise free case is Prony’s method, which forms the basis of the popular MUSIC-method for frequency estimation. There have been several suggestions on how to extend Prony’s method to several variables, cf. [4, 5, 6, 7, 9, 10, 11, 12, 14, 15, 16, 17] and the references therein. A common engineering approach is to use some reformulation of the two dimensional problem to one-dimensional problems and from there recover the two dimensional components. This causes a pairing problem that can lead to difficulties in practice. An approach that avoids this problem using the common roots of multivariate polynomials was presented in [3].

The multidimensional case poses several difficulties; Let us assume that \( f \) is of the form

\[
f(x) = \sum_{k=1}^{K} c_k e^{\zeta_k x}, \quad c_k \in \mathbb{C}, \zeta_k \in \mathbb{C}^d,
\]

Email address: fa@maths.lth.se, marcus.carlsson@math.lu.se

Preprint submitted to Elsevier
(possibly distorted by noise) and that we wish to retrieve the frequencies $\zeta_k = (\zeta_{k,1}, \ldots, \zeta_{k,d})$, where $\zeta_k \cdot x = \sum_{j=1}^d \zeta_{k,j} x_j$. We suppose for simplicity that $f$ has been sampled on an integer multi-cube $\{1, \ldots, N\}^d$. One strategy for solving the problem is of course to average over all dimensions but one, say the $p$:th one, (or extract a “fiber” in dimension $p$ from the data) and then use a 1-d technique for estimating $\zeta_{1,p}, \ldots, \zeta_{K,p}$, and then repeat this for the remaining dimensions. However, first of all this limits the amount of frequencies we are theoretically able to retrieve to $N-1$ (in the noise free case), and secondly it is not clear how to pair the $Kd$ frequencies $\zeta_{k,p}$ into $K$ multi-frequencies $\zeta_k = (\zeta_{k,1}, \ldots, \zeta_{k,d})$.

In this work, we will extend the ESPIRIT-algorithm to several variables, and in particular show that this method avoids the pairing problem. Moreover, we will also show how to work with data sampled on non-standard domains. This problem is also discussed in [6]. As it turns out, we are able to estimate $N^{d-1}(N-1)$ distinct multi-frequencies, avoiding degenerate cases. For a 10$^3$ data-tensor this amounts to that we may retrieve 900 (non-degenerate) complex frequencies (assuming no noise), as opposed to 9 using reduction to 1 dimensional techniques. In the numerical section we also test how the method performs with noise present, but we wish to underline that the main contribution of this paper is the deduction of an algorithm that is capable of exact retrieval in the noise-free case. Its performance as an estimator will be investigated elsewhere.

The paper is organized as follows, in Section 2 we go over the essentials of the classical ESPRIT algorithm in one variable, in Section 3 we extend this to the multi-cube case, using block Hankel matrices, and show how to solve the pairing problem. We then consider how to make this work for data sampled on non-cubical domains. This relies on so called general domain Hankel matrices, which are introduced in Section 4 and 5. The subsequent extension of the ESPRIT-type algorithm from Section 3 is given in Section 6. Numerical examples are given in Section 7.

2. One-dimensional ESPRIT

Let us start by assuming that $f$ is a given one-dimensional function which is sampled at integer points, and that

$$f(x) = \sum_{k=1}^K c_k e^{\zeta_k x},$$

for distinct values of $\zeta_k$, $k = 1, \ldots, K$. A Hankel matrix is a matrix $H$ which has constant values on its anti-diagonals. It is easy to see that each Hankel matrix can be generated from a function $h$ by setting $H(m,n) = h(m+n)$. Let us now consider the case where $H$ is generated by the function $f$ above, and let us assume that the number of columns and rows of $H$ are larger than $K+1$.

Let $\Lambda$ denote the vandermonde matrix generated by the numbers $e^{\zeta_k}$, i.e., let

$$\Lambda(j,k) = e^{\zeta_k j},$$

and let $\Lambda_k$ denote the columns of $\Lambda$. For the elements of the hankel matrix $H$ it thus holds that

$$H(m,n) = f(m+n) = \sum_{k=1}^K c_k e^{\zeta_k (m+n)} = \sum_{k=1}^K c_k e^{\zeta_k m} e^{\zeta_k n},$$

implying that $H$ can be written as

$$H = \sum_{k=1}^K c_k \Lambda_k \Lambda_k^T = \Lambda \text{diag}(c) \Lambda^T,$$

2
Figure 1: Illustration of the notation $\Lambda^+$ and $\Lambda^-$. For $\Lambda^+$ the first row is deleted, and for $\Lambda^-$ the last row is deleted. The elements marked in blue are to be deleted.

where $c = (c_1, \ldots, c_K)$. From this observation it is clear that the rank of $H$ is $K$.

Now, given a singular value decomposition $H = U\Sigma V^T$ (or a Autonne-Takagi factorization $H = U\Sigma U^T$, see e.g. Ch.4.4 [8]), it holds that the singular vectors $U$ (or $V$) are linear combinations of the columns of $\Lambda$, i.e.,

$$U = \Lambda B,$$

where $B$ is an invertible $K \times K$ matrix.

Let $D = \text{diag}(e^{\xi_1}, \ldots, e^{\xi_K})$. Let us also introduce the notation $\Lambda^+$ for the matrix that is obtained by deleting the first row from the matrix $\Lambda$, and similarly, let $\Lambda^-$ denote the matrix that is obtained by deleting the last row from $\Lambda$. There is a simple relationship between $\Lambda^+$ and $\Lambda^-$, namely

$$\Lambda^+ = \Lambda^- D.$$

Moreover, it holds that

$$U^+ = \Lambda^+ B = \Lambda^- DB$$
$$U^- = \Lambda^- B$$

Let us now consider

$$A = (U^-)^\dagger U^+ = ((U^-)^*)U^-)^{-1}(U^-)^*U^+)$$
$$= (B^*(\Lambda^-)^*\Lambda^- B)^{-1}B^*(\Lambda^-)^*\Lambda^- DB$$
$$= B^{-1}((\Lambda^-)^*\Lambda^-)^{-1}(B^*)^{-1}B^*(\Lambda^-)^*\Lambda^- DB$$
$$= B^{-1}DB.$$

Hence, the columns of $B^{-1}$ are eigenvectors of the matrix $A$ with $e^{\xi_k}$ being the corresponding eigenvalues. This implies that the exponentials $e^{\xi_k}$ can be recovered from $H$ by diagonalizing the matrix $A$, and this is the essence of the famous ESPRIT-method.

We end by making a remark about the connection between Hankel matrices with low-rank and functions that are sums of exponential functions. For every function $f$ being a sum of $K$ (distinct)
Algorithm 1 One-dimensional ESPRIT

1: Form Hankel matrix $H(m, n) = f(m + n)$ from samples of $f$.
2: Compute singular value decomposition $H = UΣV^*$. 
3: Form $U^+$ and $U^-$ by deleting the first and last rows from $U$, respectively.
4: Form $A = ((U^-)^*U^-)^{-1}(U^-)^*U^+$. 
5: Diagonalize $A = B^{-1}\text{diag}(λ_1, ..., λ_K)B$ by making an eigenvalue decomposition of $A$.
6: Recover $ζ_k = \log(λ_k)$

exponential functions the rank of the corresponding Hankel matrix is $K$ (given sufficiently many samples). The reverse state will typically be true, but there are exceptions. These exceptions are of degenerate character (like for instance the number of roots of a polynomial equation can degenerate in cases of multiple roots), and therefore this problem is often discarded in practice. For a longer discussion of these issues, see Section 2 and 11 in [2].

3. Multi-dimensional ESPRIT for block-Hankel matrices

We now consider the case where $f$ is a given $d$-dimensional function which is sampled at integer points, and that $f(x) = K \sum_{k=1}^{K} c_k e^{ζ_k \cdot x}$, where $ζ_k = (ζ_{k,1}, ..., ζ_{k,d})$ and $x = (x_1, ..., x_d)$. A $d$-dimensional Hankel operator can be viewed as a linear operator on the tensor product $C^{N_1} \otimes ... \otimes C^{N_d}$ whose entry corresponding to multi-indices $m = (m_1, ..., m_d)$ and $n = (n_1, ..., n_d)$ equals $f(m + n)$. If we identify $C^{N_1} \otimes ... \otimes C^{N_d}$ with $C^{N_1...N_d}$ using the reverse lexicographical order, i.e. by identifying entry $m$ in the former with

\[ m_1 + m_2 N_1 + m_3 N_1 N_2 + ... + m_d N_1 ... N_{d-1} \]  \hspace{1cm} (3.1)

in the latter, the corresponding operator can be realized as a block-Hankel matrix, which we denote by $H$. Analogously, given multiindex $m$ we will write $m$ for the number (3.1), and if $u$ is an element in $C^{N_1} \otimes ... \otimes C^{N_d}$ we write $u$ for its vectorized version. For the remainder, we assume for simplicity that $N_1 = ... = N_d$ and simply denote it $N$ (the general case being covered by Section 6 anyways).

Just as for the one-dimensional case it is easy to see that if $Λ$ is the vandermonde matrix generated by the vectors $ζ_k$, i.e.,

\[ Λ(m, k) = e^{ζ_k \cdot m}, \]

then

\[ H = \sum_{k=1}^{K} c_k Λ_k Λ_k^T = Λ\text{diag}(c) Λ^T, \]

and that, given the singular value decomposition $H = UΣV^T$, it holds that the singular vectors $U$ (or $V$) are linear combinations of the columns of $Λ$, i.e.,

\[ U = ΛB, \]  \hspace{1cm} (3.2)

4
where $B$ is an invertible $K \times K$ matrix like before. This step of course assumes that $\Lambda$ has full rank, i.e. that \{e^{\xi \cdot m}\}_{k=1}^{K}$ forms a linearly independent set on $\mathbb{C}^{N} \otimes \ldots \otimes \mathbb{C}^{N}$, which gives the natural restriction $K < N^d$.

We now generalize the previous operation of deleting the first and last row respectively to several dimensions. By $\Lambda^{p+}$ we denote the matrix that is obtained by deleting all entries of $\Lambda$ with indices $m$ corresponding to multiindices $m$ of the form $(m_1, \ldots, 1, \ldots, m_d)$, i.e., all the first elements with respect to dimension $p$. Similarly, we denote by $\Lambda^{p-}$ the matrix that is obtained by deleting from $\Lambda$ all the elements related to multiindices $(m_1, \ldots, N_p, \ldots, m_d)$, i.e., all the last elements with respect to dimension $p$. Let

$$D_p = \text{diag}(e^{\xi_{1,p}}, \ldots, e^{\xi_{K,p}}).$$

and note that in each dimension $p$ we have

$$\Lambda^{p+} = \Lambda^{p-} D_p.$$

Moreover, it holds that

$$U^{p+} = \Lambda^{p-} B = \Lambda^{p+} D_p B$$

$$U^{p-} = \Lambda^{p-} B$$

Let us now again consider

$$A_p = (U^{p-})^{\dagger} U^{p+} = (U^{p-})^{\dagger} (U^{p-})^{-1} (U^{p-})^{\dagger} U^{p+}$$

$$= (B^* (\Lambda^{p-})^{\dagger} \Lambda^{p-} B)^{-1} B^* (\Lambda^{p-})^{\dagger} \Lambda^{p-} D_p B$$

$$= B^{-1} ((\Lambda^{p-})^{\dagger} \Lambda^{p-})^{-1} (B^*)^{-1} B^* (\Lambda^{p-})^{\dagger} \Lambda^{p-} D_p B$$

$$= B^{-1} D_p B.$$  

For this step to work out we need of course that $(\Lambda^{p-})^{\dagger} \Lambda^{p-}$ is invertible for all $p$, which happens if each $\Lambda^{p-}$ has full rank. This leads to the restriction

$$K < N^d (N - 1)$$

5
which was mentioned already in the introduction, since $\Lambda^{p-}$ needs to have fewer columns than rows. In degenerate cases it may of course happen that $\Lambda^{p-}$ does not have full rank anyways. We assume from now on that this does not happen.

The remarkable observation here is that $B^{-1}$ simultaneously diagonalizes all the matrices $A_p$, and that the eigenvectors to $A_p$ are $(e^{\zeta_1,p}, \ldots, e^{\zeta_K,p})$ for each $p$. This implies that all the exponentials $e^{\zeta_k,p}$ can be recovered from $H_f$ by diagonalization of e.g. $A_1$, and then use the same eigenvectors for diagonalization of the remaining $A_p$'s, and in this way obtaining the $\zeta_k$ directly without any need for a pairing procedure.

**Algorithm 2** $d$-dimensional ESPRIT

1. Form the $d$-block Hankel matrix $H(m, n) = f(m + n)$ from samples of $f$.
2. Compute the singular value decomposition $H = U\Sigma V^*$.
3. for $p = 1, \ldots, d$ do
4. Form $U^p+$ and $U^p-$ by deleting first and last elements in dimension $p$, respectively.
5. Form $A_p = (U^p-)^*U^p-1(U^p-)^*U^p+$.
6. end for
7. Diagonalize $A_1 = B^{-1}D_1B$ by making an eigenvalue decomposition of $A_1$.
8. for $p = 1, \ldots, d$ do
9. Compute the diagonal matrices $D_p = \text{diag}(\lambda_1,p, \ldots, \lambda_K,p) = BA_pB^{-1}$.
10. Recover $\zeta_k,p = \log(\lambda_k,p)$ (which are automatically correctly paired).
11. end for

This completes the multidimensional version of ESPRIT in the case when data $f$ is sampled on a regular multi-cube. Next we address more general setting of data measured on regular grids with various shapes. For this we need to introduce so called general domain Hankel matrices, by viewing block Hankel matrices as multi-dimensional summing operators.

4. The summation operator formalism in one variable

Given a subset $\Upsilon \subset \mathbb{N}$, let $\ell^2(\Upsilon)$ be the “sequences” indexed by $\Upsilon$ and equipped with the standard $\ell^2$-norm. If $\Upsilon = \{1, 2, \ldots, N\}$, then $\ell^2(\Upsilon)$ reduces to the standard $\mathbb{C}^N$. Note that we may define a classical Hankel matrix $H$ given by the sequence $f = (f_2, f_3, \ldots, f_{2N})$ as the operator with domain and codomain equal to $\ell^2(\{1, \ldots, N\})$, given by the summation formula

$$ (H_f(a))_n = \sum_{m=1}^{N} f_{m+n}a_m, \quad n \in \{1, \ldots, N\} \quad (4.1) $$

where $a$ represents any element in $\ell^2(\{1, \ldots, N\})$. This suggests the following generalization: Let $M, N \in \mathbb{N}$ such that $M + N = 2P$ be given, and consider $H_{f,M,N} : \ell^2(\{1, \ldots, M\}) \to \ell^2(\{1, \ldots, N\})$ given by

$$ (H_{f,M,N}(a))_n = \sum_{m=1}^{M} f_{m+n}a_m, \quad n \in \{1, \ldots, N\}. \quad (4.2) $$

We remark that

$$ \{1, \ldots, M\} + \{1, \ldots, N\} = \{2, \ldots, 2P\}, \quad (4.3) $$
where the latter is the grid on which \( F \) is sampled. A moments thought reveals that \( H_{f,M,N} \) equals the \( N \times M \) Hankel matrix given by \( f \). For example, if \( f = (1, 2, 3, 4, 5) \), then \( P = 3 \) and we can pick e.g. \( M = 4 \) and \( N = 2 \). We then have

\[
H_{f,4,2} = \begin{pmatrix}
1 & 2 & 3 & 4 \\
2 & 3 & 4 & 5
\end{pmatrix}.
\] (4.4)

As long as \( M \) and \( N \) are larger than \( K \), it is possible to perform ESPRIT on these rectangular Hankel matrices as well, although square ones seems to be preferred.

5. General domain Hankel operators

General domain Hankel integral operators were introduced in [1] (albeit under the name truncated correlation operators) and their discretizations were studied in [2] where the term general domain Hankel matrix was coined. We briefly revisit their construction, which is explained in greater detail in Section 4 of [2].

Suppose we are working with samples of a function \( f \) on some domain in \( \mathbb{R}^d \). Let \( \Xi, \Upsilon \) be any bounded subsets of \( \mathbb{Z}^d \), and in analogy with (4.3) set

\[
\Upsilon + \Xi = \Omega.
\] (5.1)

We suppose that \( \Upsilon \) and \( \Xi \) are such that \( \{j_1 \Delta x_1, \ldots, j_d \Delta x_d : j \in \Omega\} \) cover the domain where \( f \) is defined, for some choice of sampling length \( \Delta x_1, \ldots, \Delta x_d \). The function \( f \) thus gives rise to a multidimensional sequence

\[
f_j = f(j_1 \Delta x_1, \ldots, j_d \Delta x_d), \quad j \in \Omega,
\] (5.2)

(or more formally a function in \( \ell^2(\Omega) \)). We will refer to such functions as \textit{md-sequences}, to distinguish them from ordinary sequences, (i.e. vectors in \( \mathbb{C}^N \)). In analogy with (4.2), the md-sequence \( f \) gives rise to a corresponding general domain Hankel operator

\[
(H_{f,\Upsilon,\Xi}(g))_n = \sum_{m \in \Upsilon} f_{m+n} g_m, \quad n \in \Xi,
\] (5.3)

where \( g \) is an arbitrary md-sequence on \( \Upsilon \). When \( \Upsilon, \Xi \) are clear from the context or irrelevant, we drop them from the notation.

We may of course represent \( g \) as a vector, by ordering the entries in some (non-unique) way. More precisely, by picking any bijection

\[
o_g : \{1, \ldots, |\Upsilon|\} \to \Upsilon
\] (5.4)

(where \( |\Upsilon| \) denotes the amount of elements in the set \( \Upsilon \), we can identify \( g \) with the vector \( \mathbf{g} \) given by

\[
(g_j)_{j=1}^{|\Upsilon|} = g(o_g(j)).
\]
Letting \( o_x \) be an analogous bijection for \( \Xi \), it is clear that \( H_{f,\Xi,\Upsilon} \) can be represented as a \( |\Xi| \times |\Upsilon| \)-matrix, where the \((n,m)\)th element is \( f(o_x(n) + o_y(m)) \). Such matrices will be called \textit{general domain Hankel matrices} and denoted \( H_{f,\Upsilon,\Xi} \), letting the bijections \( o_x \) and \( o_y \) be implicit in the notation (usually we will use the reverse lexicographical order as before). In particular, if we set

\[
\Upsilon = \Xi = \{1, \ldots, N_1\} \times \ldots \times \{1, \ldots, N_d\},
\]

we retrieve in this way the block Hankel matrices discussed in Section 3. Note that in the one dimensional case, \( H_{f,\Upsilon,\Xi} \) and \( H_{f,\Pi,\Xi} \) are virtually the same thing, whereas this is not the case in several variables. The former is an operator acting on md-sequences in \( \ell^2(\Upsilon) \) to md-sequences in \( \ell^2(\Xi) \), the latter is a matrix representation of the former that can be used e.g. for computer implementations.

An example where \( \Xi = \Upsilon \) (and hence also \( \Omega \)) are triangles is shown in Figures 3 and 4. The domains \( \Xi = \Upsilon \) are shown in Figures 3 with a particular ordering. \( \Omega \) is then a triangle of side-length 9, and a function \( f \) on \( \Omega \) is shown in Figure 4a). The corresponding general domain Hankel matrix, given the ordering from Figure 3 is shown in b). Note especially how columns 3, 4 and 5 are visible and show up as rectangular Hankel matrices of different sizes. Columns 3 and 5 give rise to a central square Hankel matrix, whereas number 4 does not.

### 6. Frequency estimation

As earlier, we assume that we have samples of a function \( f \) of the form

\[
f(x) = \sum_{k=1}^{K} c_k e^{\zeta_k \cdot x},
\]

where the axles have been scaled so that we sample at integer points. However, we now assume that samples are available only on \( \Omega \) which is a domain of the form \( \Xi + \Upsilon \) for some domains \( \Xi \) and \( \Upsilon \), where we only require that \( \Xi \) be convex (in the sense that it arise as the discretization of a
convex domain). Given \( \lambda \in \mathbb{C}^d \) we let \( \Lambda_\Omega(\lambda) \) denote the md-sequence \( \lambda^j \) for \( j \in \Omega \), where we use multi-index notation \( \lambda^j = \lambda_1^{j_1} \lambda_2^{j_2} \ldots \lambda_d^{j_d} \). Rephrased, we suppose that \( f \) can be written

\[
f_j = \sum_{k=1}^{K} c_k \Lambda_{k,\Omega}, \quad \lambda_k = (\epsilon_k, \ldots, \epsilon_k, d), \quad (6.1)
\]

It is then easy to see that

\[
H_f(g) = \sum_{k=1}^{K} c_k H_{\Lambda_k,\Omega}(g) = \sum_{k=1}^{K} c_k \Lambda_{k,\Xi}(g, \tilde{\Lambda}_{k,\Upsilon}), \quad (6.2)
\]

which in particular shows that \( H_f \) is a rank \( K \) operator, assuming of course that the \( c_k \)'s are non-zero, the \( \lambda_k \)'s are distinct and that \( \{ \Lambda_k,\Xi \}_{k=1}^{K} \) and \( \{ \Lambda_k,\Upsilon \}_{k=1}^{K} \) form linearly independent sets. The latter condition will be fulfilled if the real domains \( \Xi, \Upsilon \subset \mathbb{R}^d \) are connected and the sampling is dense enough, we refer to [2] for more information, and assume henceforth that the linear independence is fulfilled.

Letting \( \sigma_\Xi \) be the lexicographical ordering on \( \Xi \), we identify md-sequences \( u \) in \( \ell^2(\Xi) \) with sequences (vectors) \( u \) in \( \ell^2(\Xi) \), as explained in Section [3]. The SVD of \( H_f \) thus gives rise to singular vectors \( u_k, v_k \), whose md-sequence counterparts satisfy

\[
H_f(u_k) = \sigma_k u_k, \quad H_f^*(u_k) = \sigma_k v_k.
\]

Since \( \{ u_k \}_{k=1}^{K} \) span the range of \( H_f \), i.e. the span of \( \{ \Lambda_k,\Xi \}_{k=1}^{K} \) by (6.2), it follows that we can write

\[
u_j = \sum_{k=1}^{K} b_{j,k} \Lambda_{k,\Xi} \quad (6.3)
\]

where the numbers \( b_{j,k} \) form a square \( K \times K \) invertible matrix. Let \( U_K \) be the matrix with columns \( u_1, \ldots, u_K \) and let \( \Lambda \) be the matrix with the columns \( \Lambda_{1,\Xi}, \ldots, \Lambda_{K,\Xi} \). The relation (6.3) can then be expressed

\[
U_K = \Lambda \quad (6.4)
\]

(compare with (3.2)).

Since we have assumed that \( \Xi \) is convex, we have that each “fiber” \( \{ m_1 : (m_1, m_2, \ldots, m_d) \in \Xi \} \), is of the form \( \{ j \}_{j=M_1}^{M_2} \) for some integers \( M_1 \leq M_2 \). The grid that arises by removing the first (respectively last) element of each such fiber will be denoted \( \Xi^{+,-1} \) (respectively \( \Xi^{-,-1} \)). Analogous definitions of course apply to the other variables, yielding subdomains \( \Xi^{\pm,2}, \ldots, \Xi^{\pm,d} \). We assume that the sampling has been done so that no fiber consists of one sole element.

Now, given an md-sequence \( w \) in \( \ell^2(\Xi) \), we let \( w^{+,-p} \) be the md-sequence restricted to the grid \( \Xi^{+,-p} \). Moreover, given a matrix like \( U_K \), whose columns are given by vectorized elements of \( \ell^2(\Xi) \), we denote by \( U_K^{+,-p} \) (resp. \( U_K^{-,-p} \)) the matrix formed by the corresponding vectorized elements \( u_k^{+,-p} \in \ell^2(\Xi^{+,-p}) \) (resp. \( u_k^{-,-p} \in \ell^2(\Xi^{-,-p}) \)). Equation (6.4) implies that

\[
U_K^{+,-p} = \Lambda^{+,-p} B. \quad (6.5)
\]

Moreover, using the same notation as (3.3), we also have

\[
\Lambda^{+,-p} = \Lambda^{-,-p} \text{diag}(\lambda_{1,j}, \ldots, \lambda_{K,j}) = \Lambda^{-,-p} D_p \quad (6.6)
\]
Algorithm 3 General domain ESPRIT

1: Order the elements in $\Xi$ and $\Upsilon$.
2: Form the general domain Hankel matrix $H(m, n) = f(m + n)$ from samples of $f$, where e.g. $m$ is the order of the multi-index $m$ in $\Xi$.
3: Compute the singular value decomposition $H = U\Sigma V^*$.
4: for $p = 1, \ldots, d$ do
5: Form $U^{p+}$ and $U^{p-}$ by deleting first and last elements in each fiber in the $p$:th coordinate, respectively.
6: Form $A_p = ((U^{p-})^*U^{p-})^{-1}(U^{p-})^*U^{p+}$.
7: end for
8: Diagonalize $A_1 = B^{-1}D_1B$ by making an eigenvalue decomposition of $A_1$.
9: for $p = 1, \ldots, d$ do
10: Compute the diagonal matrices $D_p = \text{diag}(\lambda_{1,p}, \ldots, \lambda_{K,p}) = BA_pB^{-1}$.
11: Recover $\zeta_{k,p} = \log(\lambda_{k,p})$ (which are automatically correctly paired).
12: end for

Combined this implies

$$U^{p+}_K = \Lambda^{-p}D_p B.$$  \hspace{1cm} (6.7)

In analogy with the computations in Section 3 we have

$$A_p := ((U^{p-}_K)^*U^{p-}_K)^{-1}(U^{p-}_K)^*U^{p+}_K = (B^*(\Lambda^{-p})^*\Lambda^{-p}B)^{-1}B^*(\Lambda^{-p})^*\Lambda^{-p}D_p B = B^{-1}D_p B$$

so we can retrieve the desired frequencies $\lambda_{k,p} = e^{\zeta_{k,p}}$ by computing $A_p$ and diagonalizing it. Since all matrices $A_p$ are diagonalized by the same matrix $B$, the issue with grouping of the complex frequencies is easily solved just as in the previous case.

7. Numerical examples

To illustrate the methods discussed we will perform a number of numerical simulations. To begin with we study problems in the absence of noise to verify that we can accurately recover the exponentials for functions of the form (1.2) for different domains. We end by briefly discussing what happens in the presence of noise. Note that due to the non-linearity of the problem, this can behave quite differently in different situations.

Figure 5 shows results for a two-dimensional example using block-Hankel structure. The underlying function $f$ is in this case constructed as a linear combination of 300 purely oscillatory exponential functions with frequencies distributed along a spiral as depicted in panel (c). The coefficients are chosen randomly, and $f$ is sampled on a 61 by 61 square grid. The real part of $f$ is shown in panel (a). The block-Hankel matrix that is generated from $f$ is shown in panel (b).

We have followed Algorithm 2 to recover the underlying frequencies and they are recovered at machine precision error as shown in panel (d). We again point out that, upon using reduction to 1 dimensional techniques, the maximum amount of frequencies that could have been retrieved from this data set is 60. With the present method it is $30 \times 31 - 1 = 929$ (see (3.4)) compared to the total amount of measurements $61^2 = 3721$.

Next, we make a similar study for the case of three-dimensional block-Hankel matrices. The layout of Figure 5 is similar to that of Figure 6. In this case 900 purely oscillatory exponential
functions were used to generate a function $f$ sampled on a grid of size $21 \times 21 \times 21$. In contrast to the previous example, the frequencies are now randomly distributed. As before, the real part of $f$ is shown in panel (a), the corresponding block-Hankel matrix is shown in panel (b), the distribution of the exponentials in panel (c) and the frequency reconstruction error is shown in panel (d). According to (3.4), the maximal amount of frequencies we may retrieve in this situation is $11^2 \times 10 - 1 = 1209$. We now turn our focus to an example using the general domain Hankel structure. In this case, we have used 100 exponential functions to generate a function $f$, but the sampling of this function is no longer on a rectangular grid. Instead, it is sampled on a the circular-like domain $\Omega$ shown in

Figure 5: Two dimensional example with data being a linear combination of 300 exponential functions.
Figure 6: Three dimensional example with data being a linear combination of 900 exponential functions.

Figure 8(a). Figure 7(a) shows how the domain $\Omega$ is constructed from the two domains $\Upsilon$ and $\Xi$. Note that the set $\Xi$ is a rather small domain - a grid of size $11 \times 11$. The size of $\Xi$ will determine how many frequencies that can recovered, in this case $11 \times 10 = 109$. The general domain Hankel matrix that is constructed from $f$ is shown in Figure 7(b). Figure 8(b) shows the distribution of the underlying frequencies, and Figure 7(c) shows the reconstruction error in determining these frequencies. Again, the error is down at machine precision.

Finally, we briefly illustrate the impact of noise on the proposed algorithms. The scatter plot in Figure 9(a) shows the distribution of 40 frequencies nodes in a case of purely oscillatory exponential
functions. These are used to generate a function $f$ that is then sampled on a $41 \times 41$ grid. In addition, different levels of normally distributed noise is added to the original data. Six noise levels are chosen so that the ratio between the $\ell^2$ norm of the noise and the noise-free data is $10^0$, $10^{-0.5}$, $10^{-1}$, $10^{-2}$, $10^{-3}$, and $10^{-4}$, respectively. The impact that the noise have on the singular values of the corresponding block-Hankel matrix is shown in Figure 9(b). Note that the noise free block Hankel matrix has rank 40, i.e. 40 non-zero singular values. When low noise is present this shows up as a jump in the magnitude of the singular values, clearly visible in plot 9(b). The first 40 singular values for e.g. yellow and orange are covered by the brown ones, therefore not visible. We can see that for the four lower levels, the impact of noise does not affect the original distribution on the singular values much, whereas for the highest noise level it affects essentially all singular values, and for the second highest level, it is just starting to have an impact. From this, we would expect to see a high error for the highest noise level, and a relatively small error for the four smallest noise levels.

To illustrate the effect on the individual frequency nodes, each of the nodes in Figure 9(a) have
(a) Real part of a 2d general domain data set constructed as a linear combination of 100 exponential functions.

(b) True 2d frequencies in blue dots and estimated frequencies in red circles.

(c) Error between estimated and true frequencies.

Figure 8: Two dimensional general domain example.

6 circles around it. The color of each one of these circles show the error level in logarithmic scale according to the colorbar. Here we can see that the impact is pretty much as could be expected, with a low error for the lower noise levels and increasing for higher noise levels.

References

[1] Fredrik Andersson and Marcus Carlsson. On General Domain Truncated Correlation and Convolution Operators with Finite Rank. *Integr. Eq. Op. Th.*, 82(3), 2015.
Figure 9: Noise influence on frequency estimation. Random noise where added with six different noise levels. (a) Illustration of error in the estimation of frequencies. The color of the discs illustrate the estimation error at each node for the corresponding noise level in $\log_{10}$-scale. (b) The impact on the singular values for different noise levels.

[2] Fredrik Andersson and Marcus Carlsson. On the structure of positive semi-definite finite rank general domain Hankel and Toeplitz operators in several variables. *Complex Analysis and Operator Theory*, pages 1–30, 2016.

[3] Fredrik Andersson, Marcus Carlsson, and Maarten de Hoop. Nonlinear approximation of functions in two dimensions by sums of exponential functions. *Applied and Computational Harmonic Analysis*, 29(2):156–181, 2010.

[4] Annie Cuyt. Multivariate padé-approximants. *Journal of mathematical analysis and applications*, 96(1):283–293, 1983.

[5] Annie Cuyt and Wen-shin Lee. Sparse interpolation of multivariate rational functions. *Theoretical Computer Science*, 412(16):1445–1456, 2011.

[6] Nina Golyandina, Anton Korobeynikov, Alex Shlemov, and Konstantin Usevich. Multivariate and 2d extensions of singular spectrum analysis with the rssa package. *arXiv preprint arXiv:1309.5050*, 2013.

[7] Jouhayna Harmouch, Houssam Khalil, and Bernard Mourrain. Structured low rank decomposition of multivariate hankel matrices. *Linear Algebra and its Applications*, 2017.

[8] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[9] Sun-Yuan Kung, K Si Arun, and DV Bhaskar Rao. State-space and singular-value decomposition-based approximation methods for the harmonic retrieval problem. *JOSA*, 73(12):1799–1811, 1983.

[10] Stefan Kunis, Thomas Peter, Tim Römer, and Ulrich von der Ohe. A multivariate generalization of Prony’s method. *Linear Algebra and its Applications*, 490:31–47, 2016.
[11] Thomas Peter, Gerlind Plonka, and Robert Schaback. Reconstruction of multivariate signals via Prony’s method. *Proc. Appl. Math. Mech.*, to appear.

[12] Stephanie Rouquette and Mohamed Najim. Estimation of frequencies and damping factors by two-dimensional esprit type methods. *IEEE Transactions on signal processing*, 49(1):237–245, 2001.

[13] Richard Roy and Thomas Kailath. Esprit-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on acoustics, speech, and signal processing*, 37(7):984–995, 1989.

[14] Joseph J Sacchini, William M Steedly, and Randolph L Moses. Two-dimensional Prony modeling and parameter estimation. *IEEE Transactions on signal processing*, 41(11):3127–3137, 1993.

[15] Tomas Sauer. Prony’s method in several variables. *Numerische Mathematik*, pages 1–28, 2016.

[16] Filiep Vanpoucke, Marc Moonen, and Yannick Berthoumieu. An efficient subspace algorithm for 2-d harmonic retrieval. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume 4, pages IV–461. IEEE, 1994.

[17] Yi Zhou, Da-zheng Feng, and Jian-qiang Liu. A novel algorithm for two-dimensional frequency estimation. *Signal Processing*, 87(1):1–12, 2007.