2D-EXTENSION OF SINGULAR SPECTRUM ANALYSIS: ALGORITHM AND ELEMENTS OF THEORY

N.E. GOLYANDINA * AND K.D. USEVICH †

Abstract. Singular Spectrum Analysis is a nonparametric method, which allows one to solve problems like decomposition of a time series into a sum of interpretable components, extraction of periodic components, noise removal and others. In this paper, the algorithm and theory of the SSA method are extended to analyse two-dimensional arrays (e.g. images). The 2D-SSA algorithm based on the SVD of a Hankel-block-Hankel matrix is introduced. Another formulation of the algorithm by means of Kronecker-product SVD is presented. Basic SSA notions such as separability are considered. Results on ranks of Hankel-block-Hankel matrices generated by exponential, sine-wave and polynomial 2D-arrays are obtained. An example of 2D-SSA application is presented.

Key words. Singular Spectrum Analysis, image analysis, Hankel-block-Hankel matrix, separability, finite rank, Singular Value Decomposition, Kronecker-product SVD

AMS subject classifications. 62H35, 62H25, 62-07, 15A03, 15A18

1. Introduction. The purpose of this paper is to extend the SSA (Singular Spectrum Analysis) algorithm and theory developed in [7] to the case of *two-dimensional arrays* of data (i.e. real-valued functions of two variables defined on Cartesian grid). The monochrome digital images are a standard example here. Singular Spectrum Analysis is a well-known model-free technique for analysis of real-valued time series. Basically, SSA is an exploratory method intended to perform decomposition of a time series into a sum of interpretable components, such as trend, periodicities and noise (see [3, 4, 7] for more details). SSA has proved to be successful for such tasks. Moreover, there are several SSA extensions for time series forecasting, change-point detection, missing values imputation and so on. These are the reasons to believe that the two-dimensional extension of SSA (2D-SSA, first presented in [6]) has similar capabilities. However, its application was hampered by lack of theory, which this paper is intended to reduce.

Suppose we observe a 2D-array of data (a real matrix) being a sum of unknown components $\mathrm{F} = \mathrm{F}^{(1)} + \ldots + \mathrm{F}^{(m)}$. The general task of the 2D-SSA algorithm is to produce a decomposition

$$\mathrm{F} = \widetilde{\mathrm{F}}^{(1)} + \ldots + \widetilde{\mathrm{F}}^{(m)}, \tag{1.1}$$

where the terms approximate the initial components.

In §2 we present the algorithm of 2D-SSA. First of all, the algorithm is formulated basing on the SVD of the Hankel-block-Hankel (HbH for short) matrix generated by the input 2D-array. However, another equivalent representation of the algorithm fits better for examination and analysis. It is based on the decomposition of a matrix into a sum of Kronecker products.

The key step of the algorithm is grouping of terms of the SVD. This step governs the resulting decomposition (1.1). Main problems of grouping are: possibility of proper grouping and identification of terms in the SVD. These problems are discussed in §2.4 and investigated in §3 and §4.

*Mathematical Department, St.Petersburg State University, Universitetskij pr. 28, St.Petersburg Petrodvorets 198504, Russia, (nina@gistatgroup.com).

†Mathematical Department, St.Petersburg State University, Universitetskij pr. 28, St.Petersburg Petrodvorets 198504, Russia, (usevich.k.d@gmail.com).

In §3 we study the notion of *separability* inherited from the 1D case. Separability means possibility to extract constituents from their sum by 2D-SSA. We also provide a brief review of results on one-dimensional separability as the basis for results in the 2D case.

Section 4 deals with the so-called *2D-SSA rank* of a 2D-array defined as the number of SVD terms corresponding to the 2D-array and equal to the rank of a Hankel-block-Hankel matrix generated by the 2D-array. This number is important, as it should be taken into account when performing identification. We provide rank calculations for different 2D-arrays: exponents, polynomials and sine-waves.

In §5 we demonstrate 2D-SSA notions by an example of periodic noise removal.

General definitions. First of all, let us review definitions that will be used throughout this paper.

The following operator is widely used in the SSA theory and is quite helpful for the 2D-SSA algorithm formulation.

DEFINITION 1.1. *Let* $\mathrm{A} = \left(a_{ij}\right)_{i,j=1}^{m,n} \in M_{m,n}(Q)$ *be a matrix over Euclidean space* Q*. The* hankelization operator $\mathcal{H}^Q : M_{m,n}(Q) \mapsto M_{m,n}(Q)$ *is defined by*

$$\mathcal{H}^Q\mathrm{A} = \begin{pmatrix} \widetilde{a}_1 & \widetilde{a}_2 & \dots & \widetilde{a}_n \\ \widetilde{a}_2 & \widetilde{a}_3 & \dots & \widetilde{a}_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{a}_m & \widetilde{a}_{m+1} & \dots & \widetilde{a}_{m+n-1} \end{pmatrix}, \quad \widetilde{a}_k = \left(\sum_{(i,j)\in\mathcal{D}_k} a_{ij}\right) \Big/ \#\mathcal{D}_k,$$

where $\mathcal{D}_k = \{(i,j) : \ 1 \le i \le m, \ 1 \le j \le n, \ i+j = k+1\}$.

Further, we will denote by $\mathcal{M}_{m,n} \stackrel{\mathrm{def}}{=} M_{m,n}(\mathbb{R})$ the space of real matrices with Frobenius inner product:

$$\langle \mathrm{X}, \mathrm{Y} \rangle_{\mathcal{M}} = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}y_{ij}, \tag{1.2}$$

where $\mathrm{X} = (x_{ij})_{i,j=1}^{m,n}, \mathrm{Y} = (y_{ij})_{i,j=1}^{m,n} \in \mathcal{M}_{m,n}$.

Introduce an isomorphism between $\mathcal{M}_{m,n}$ and $\mathbb{R}^{mn}$.

DEFINITION 1.2. The vectorization *(see, for instance, [8]) of* $\mathrm{A} = (a_{ij})_{i,j=1}^{m,n} \in \mathcal{M}_{m,n}$ *is given by*

$$\operatorname{vec}\mathrm{A} \stackrel{\mathrm{def}}{=} (a_{11}, \dots, a_{m1}; a_{12}, \dots, a_{m2}; \dots; a_{1n}, \dots, a_{mn})^{\mathrm{T}}. \tag{1.3}$$

DEFINITION 1.3. *The* (m,n)-matricizing *of* $\mathrm{X} \in \mathbb{R}^{mn}$ *denoted by* $\operatorname{matr}_{m,n}(\mathrm{X})$ *is defined to be* $\mathrm{A} \in \mathcal{M}_{m,n}$ *satisfying* $\operatorname{vec}\mathrm{A} = \mathrm{X}$.

Then, recall the operation of Kronecker product [8, 9].

DEFINITION 1.4. *For* $\mathrm{A} = (a_{ij})_{i,j=1}^{m,n} \in \mathcal{M}_{m,n}$ *and* $\mathrm{B} = (b_{kl})_{k,l=1}^{p,q} \in \mathcal{M}_{p,q}$ *their* Kronecker product *is, by definition,*

$$\mathrm{A} \otimes \mathrm{B} \stackrel{\mathrm{def}}{=} \begin{pmatrix} a_{11}\mathrm{B} & \dots & a_{1n}\mathrm{B} \\ \vdots & & \vdots \\ a_{m1}\mathrm{B} & \dots & a_{mn}\mathrm{B} \end{pmatrix}. \tag{1.4}$$

Finally, we need an isomorphism between classes of block matrices.

DEFINITION 1.5. *The* rearrangement $\mathcal{R} : \mathcal{M}_{mp,nq} \mapsto \mathcal{M}_{pq,mn}$ *is defined as*

$$\begin{gathered} \mathcal{R}(\mathrm{C}) \stackrel{\mathrm{def}}{=} \mathrm{D} \in \mathcal{M}_{pq,mn}, \quad \textit{where} \\ (\mathrm{D})_{i+(j-1)p,k+(l-1)m} = (\mathrm{C})_{i+(k-1)p,j+(l-1)q} \end{gathered} \tag{1.5}$$

for $1 \le i \le p, 1 \le j \le q, 1 \le k \le m, 1 \le l \le n$.

Note that the introduced rearrangement of a matrix is the transpose of the rearrangement defined in [2]. The following properties of the rearrangement are quite useful, despite being easily checked.

- Let $\mathrm{A} = (a_{ij})_{i,j=1}^{m,n} \in \mathcal{M}_{m,n}$ and $\mathrm{B} = (b_{kl})_{k,l=1}^{p,q} \in \mathcal{M}_{p,q}$. Then

$$\mathcal{R}(\mathrm{A} \otimes \mathrm{B}) = \operatorname{vec} \mathrm{B} (\operatorname{vec} \mathrm{A})^{\mathrm{T}}. \tag{1.6}$$

- For any $\mathrm{C} \in \mathcal{M}_{mp,nq}$

$$\|\mathcal{R}(\mathrm{C})\|_{\mathcal{M}} = \|\mathrm{C}\|_{\mathcal{M}}. \tag{1.7}$$

2. 2D-SSA.

2.1. Basic algorithm. Consider a *2D-array* of data

$$\mathrm{F} = \begin{pmatrix} f(0,0) & f(0,1) & \dots & f(0,N_y-1) \\ f(1,0) & f(1,1) & \dots & f(1,N_y-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N_x-1,0) & f(N_x-1,1) & \dots & f(N_x-1,N_y-1) \end{pmatrix}.$$

The algorithm is based on the SVD of a Hankel-block-Hankel (HbH) matrix constructed from the 2D-array. The dimensions of the HbH matrix are defined by the *window sizes* (L_x, L_y), which are restricted by $1 \le L_x \le N_x$, $1 \le L_y \le N_y$ and $1 < L_x L_y < N_x N_y$. Let $K_x = N_x - L_x + 1$ and $K_y = N_y - L_y + 1$ for convenience of notation.

Embedding

At this step, the input 2D-array is arranged into a Hankel-block-Hankel matrix of size $L_x L_y \times K_x K_y$:

$$\mathbf{W} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 & \mathbf{H}_2 & \dots & \mathbf{H}_{K_y-1} \\ \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{H}_3 & \dots & \mathbf{H}_{K_y} \\ \mathbf{H}_2 & \mathbf{H}_3 & \iddots & \iddots & \vdots \\ \vdots & \vdots & \iddots & \iddots & \vdots \\ \mathbf{H}_{L_y-1} & \mathbf{H}_{L_y} & \dots & \dots & \mathbf{H}_{N_y-1} \end{pmatrix}, \tag{2.1}$$

where

$$\mathbf{H}_j = \begin{pmatrix} f(0,j) & f(1,j) & \dots & f(K_x-1,j) \\ f(1,j) & f(2,j) & \dots & f(K_x,j) \\ \vdots & \vdots & \ddots & \vdots \\ f(L_x-1,j) & f(L_x,j) & \dots & f(N_x-1,j) \end{pmatrix}.$$

Obviously, there is the one-to-one correspondence between 2D-arrays of size $N_x \times N_y$ and HbH matrices (2.1). Let us call the matrix $\mathbf{W}$ a *Hankel-block-Hankel matrix generated by* the 2D-array F.

SVD

Then, the SVD is applied to the Hankel-block-Hankel matrix (2.1):

$$\mathbf{W} = \sum_{i=1}^{d} \sqrt{\lambda_i} U_i V_i^{\mathrm{T}}. \tag{2.2}$$

Here λ_i $(1 \le i \le d)$ are the non-zero *eigenvalues* of the matrix $\mathbf{W}\mathbf{W}^{\mathrm{T}}$ arranged in decreasing order $\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_d > 0$; $\{U_1, \ldots, U_d\}$ is a system of orthonormal in $\mathbb{R}^{L_x L_y}$ *eigenvectors* of the matrix $\mathbf{W}\mathbf{W}^{\mathrm{T}}$; $\{V_1, \ldots, V_d\}$ is an orthonormal system of vectors in $\mathbb{R}^{K_x K_y}$, hereafter called *factor vectors.* The factor vectors can be expressed as follows: $V_i = \mathbf{W}^{\mathrm{T}} U_i / \sqrt{\lambda_i}$. The triple $(\sqrt{\lambda_i}, U_i, V_i)$ is said to be the *ith eigentriple.* Note that $\sqrt{\lambda_i}$ is called a *singular value* of the matrix $\mathbf{W}$.

Grouping

After specifying m disjoint subsets of indices I_k (groups of eigentriples),

$$I_1 \cup I_2 \cup \ldots \cup I_m = \{1, \ldots, d\}, \tag{2.3}$$

one obtains the decomposition of the HbH matrix

$$\mathbf{W} = \sum_{k=1}^{m} \mathbf{W}_{I_k}, \quad \text{where } \mathbf{W}_I = \sum_{i \in I} \sqrt{\lambda_i} U_i V_i^{\mathrm{T}}. \tag{2.4}$$

This is the most important step of the algorithm as it controls the resulting decomposition of the input 2D-array. The problem of proper grouping of the eigentriples will be discussed further (in §2.4).

Projection

Projection step is necessary in order to obtain a decomposition (1.1) of the input 2D-array from the decomposition (2.4) of the HbH matrix. Firstly, matrices $\mathbf{W}_{I_k}$ are reduced to Hankel-block-Hankel matrices $\widetilde{\mathbf{W}}_{I_k}$. Secondly, 2D-arrays $\widetilde{\mathrm{F}}_{I_k}$ are obtained from $\widetilde{\mathbf{W}}_{I_k}$ by the one-to-one correspondence.

The matrices $\widetilde{\mathbf{W}}_{I_k}$, in their turn, are obtained by orthogonal projection of matrices $\mathbf{W}_{I_k}$ in Frobenius norm (1.2) onto the linear space of block-Hankel $L_x L_y \times K_x K_y$ matrices with Hankel $L_x \times K_x$ blocks. The orthogonal projection of

$$\mathbf{Z} = \begin{pmatrix} \mathrm{Z}_{1,1} & \mathrm{Z}_{1,2} & \ldots & \mathrm{Z}_{1,K_y} \\ \mathrm{Z}_{2,1} & \mathrm{Z}_{2,2} & \ldots & \mathrm{Z}_{2,K_y} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Z}_{L_y,1} & \mathrm{Z}_{L_y,2} & \ldots & \mathrm{Z}_{L_y,K_y} \end{pmatrix}, \quad \mathrm{Z}_{i,j} \in \mathcal{M}_{L_x,K_x},$$

can be expressed as a two-step hankelization

$$\widetilde{\mathbf{Z}} = \mathcal{H}^{\mathcal{M}_{L_x,K_x}} \begin{pmatrix} \mathcal{H}^{\mathbb{R}}\, \mathrm{Z}_{1,1} & \mathcal{H}^{\mathbb{R}}\, \mathrm{Z}_{1,2} & \ldots & \mathcal{H}^{\mathbb{R}}\, \mathrm{Z}_{1,K_y} \\ \mathcal{H}^{\mathbb{R}}\, \mathrm{Z}_{2,1} & \mathcal{H}^{\mathbb{R}}\, \mathrm{Z}_{2,2} & \ldots & \mathcal{H}^{\mathbb{R}}\, \mathrm{Z}_{2,K_y} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{H}^{\mathbb{R}}\, \mathrm{Z}_{L_y,1} & \mathcal{H}^{\mathbb{R}}\, \mathrm{Z}_{L_y,2} & \ldots & \mathcal{H}^{\mathbb{R}}\, \mathrm{Z}_{L_y,K_y} \end{pmatrix}.$$

In other words, the hankelization is applied at first to the blocks (*within-block* hankelization) and then to the whole matrix, i.e. the blocks on secondary diagonals are averaged between themselves (*between-block* hankelization). Certainly, the hankelization operators can be applied in the reversed order.

Thus, the result of the algorithm is

$$\mathrm{F} = \sum_{k=1}^{m} \widetilde{\mathrm{F}}_{I_k}. \tag{2.5}$$

A component $\widetilde{\mathrm{F}}_{I_k}$ is said to be the *reconstructed by eigentriples with indices* I_k 2D-array.

2.2. Algorithm: Kronecker products. Let us examine the algorithm in terms of tensors and matrix Kronecker products.

Embedding

Columns of the Hankel-block-Hankel matrix $\mathbf{W}$ generated by the 2D-array F can be treated as vectorized $L_x \times L_y$ submatrices (moving 2D windows) of the input 2D-array F (see Fig. 2.1).

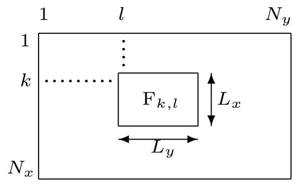

FIG. 2.1. *Moving 2D windows*

More precisely, if W_m stands for the mth column of the Hankel-block-Hankel matrix $\mathbf{W} = [W_1 : \ldots : W_{K_xK_y}]$, then

$$W_{k+(l-1)K_x} = \mathrm{vec}(\mathrm{F}_{k,l}) \quad \text{for} \quad 1 \le k \le K_x,\ 1 \le l \le K_y, \tag{2.6}$$

where $\mathrm{F}_{k,l}$ denotes the $L_x \times L_y$ submatrix beginning from the entry (k,l)

$$\mathrm{F}_{k,l} = \begin{pmatrix} f(k-1,l-1) & \ldots & f(k-1,l+L_y-2) \\ \vdots & \ddots & \vdots \\ f(k+L_x-2,l-1) & \ldots & f(k+L_x-2,l+L_y-2) \end{pmatrix}. \tag{2.7}$$

An analogous equality holds for the rows of the Hankel-block-Hankel matrix $\mathbf{W}$. Let W^n be the nth row of the matrix $\mathbf{W} = [W^1 : \ldots : W^{L_xL_y}]^{\mathrm{T}}$. Then

$$W^{i+(j-1)L_x} = \mathrm{vec}(\mathrm{F}^{i,j}) \quad \text{for} \quad 1 \le i \le L_x,\ 1 \le j \le L_y, \tag{2.8}$$

where $\mathrm{F}^{i,j}$ denotes the $K_x \times K_y$ submatrix beginning from the entry (i,j).

Basically, the HbH matrix is a 2D representation of the 4-order tensor $\mathcal{X}^{ij}_{kl}$

$$\mathcal{X}^{ij}_{kl} = (\mathrm{F}_{k,l})_{i,j} = (\mathrm{F}^{i,j})_{k,l} = f(i+k-2, j+l-2) \tag{2.9}$$

and the SVD of the matrix $\mathbf{W}$ is an orthogonal decomposition of this tensor. Another 2D representation of the tensor $\mathcal{X}^{ij}_{kl}$ can be obtained by the rearrangement (1.5) of $\mathbf{W}$:

$$\mathbf{X} = \mathcal{R}(\mathbf{W}) = \begin{pmatrix} \mathrm{F}_{1,1} & \mathrm{F}_{1,2} & \ldots & \mathrm{F}_{1,K_y} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{F}_{K_x,1} & \mathrm{F}_{K_x,2} & \ldots & \mathrm{F}_{K_x,K_y} \end{pmatrix}. \tag{2.10}$$

Let us call this block $L_xK_x \times L_yK_y$ matrix the *2D-trajectory matrix* and formulate the subsequent steps of the algorithm in terms of 2D-trajectory matrices.

SVD

First of all, recall that the eigenvectors $\{U_i\}_{i=1}^d$ form an orthonormal basis of $\mathrm{span}(W_1, \ldots, W_{K_xK_y})$ and the factor vectors $\{V_i\}_{i=1}^d$ form an orthonormal basis of $\mathrm{span}(W^1, \ldots, W^{L_xL_y})$. Consider matrices

$$\Psi_i = \mathrm{matr}_{L_x,L_y}(U_i) \in \mathcal{M}_{L_x,L_y},$$
$$\Phi_i = \mathrm{matr}_{K_x,K_y}(V_i) \in \mathcal{M}_{K_x,K_y},$$

and call Ψ_i and Φ_i *eigenarrays* and *factor arrays* respectively. It is easily seen that systems $\{\Psi_i\}_{i=1}^d$ and $\{\Phi_i\}_{i=1}^d$ form orthogonal bases of $\mathrm{span}(\{\mathrm{F}_{k,l}\}_{k,l=0}^{K_x,K_y})$ and $\mathrm{span}(\{\mathrm{F}^{i,j}\}_{i,j=0}^{L_x,L_y})$ (see (2.6) and (2.8)). Moreover, by (1.6) one can rewrite the SVD step of the algorithm as a decomposition of the 2D-trajectory matrix

$$\mathbf{X} = \sum_{i=1}^{d} \mathbf{X}_i = \sum_{i=1}^{d} \sqrt{\lambda_i} \Phi_i \otimes \Psi_i. \tag{2.11}$$

The decomposition is biorthogonal and has the same optimality properties as the SVD (see [2]). We will call it *Kronecker-product SVD* (*KP-SVD* for short).

Grouping

Grouping step in terms of Kronecker products has exactly the same form as (2.4). Choosing m disjoint subsets I_k (2.3) one obtains the grouped expansion

$$\mathbf{X} = \sum_{k=1}^{m} \mathbf{X}_{I_k}, \quad \text{where } \mathbf{X}_I = \sum_{i \in I} \sqrt{\lambda_i} \Phi_i \otimes \Psi_i. \tag{2.12}$$

Note that it is more convenient in practice to perform the grouping step on the base of Ψ_i and Φ_i (instead of U_i and V_i), since they are two-dimensional as well as the input 2D-array.

Projection

It follows from (2.11) and (1.6) that matrices $\mathbf{X}_{I_k}$ are rearrangements of corresponding matrices $\mathbf{W}_{I_k}$. Since the rearrangement $\mathcal{R}$ preserves Frobenius inner product, the resulting 2D-arrays $\widetilde{\mathrm{F}}_{I_k}$ in (2.5) can be expressed through orthogonal projections in Frobenius norm of the matrices $\mathbf{X}_{I_k}$ onto the linear subspace of 2D-trajectory matrices (2.10) and the one-to-one correspondence between 2D-arrays and matrices like (2.10).

2.3. Special cases. Here we will consider some special cases of 2D-SSA. It happens that these special cases describe most of well-known SSA-like algorithms.

2.3.1. 1D sequences: SSA for time series. The first special case occurs when the input array has only one dimension, namely it is a one-dimensional finite real-valued sequence (*1D-sequence* for short):

$$\mathrm{F} = (f(0,0), \ldots, f(N_x - 1, 0))^{\mathrm{T}}. \tag{2.13}$$

In this case, the 2D-SSA algorithm coincides with the original SSA algorithm [7] applied to the same data. Let us briefly describe the SSA algorithm in its standard notation denoting $f(i, 0)$ by f_i and N_x by N.

The only parameter $L = L_x$ is called the *window length*. Let $K = N - L + 1 = K_x$. Algorithm consists of four steps (the same as those of 2D-SSA). The result of Embedding step is the Hankel matrix

$$\mathbf{W} = \begin{pmatrix} f_0 & f_1 & f_2 & \ldots & f_{K-1} \\ f_1 & f_2 & f_3 & \ldots & f_K \\ f_2 & f_3 & f_4 & \ldots & f_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{L-1} & f_L & f_{L+1} & \ldots & f_{N-1} \end{pmatrix}. \tag{2.14}$$

This matrix is called the *trajectory matrix*[1]. SVD and Decomposition steps are exactly the same as in the 2D case. Projection in the 1D case is formulated as one-step hankelization $\mathcal{H}^{\mathbb{R}}$.

2.3.2. Extreme window sizes. Let us return to a general 2D-array case when $N_x, N_y > 1$. Consider extreme window sizes: (a) $L_x = 1$ or $L_x = N_x$; (b) $L_y = 1$ or $L_x = N_y$.

1. If conditions (a) and (b) are met both, then due to condition $1 < L_x L_y < N_x N_y$ we get $(L_x, L_y) = (N_x, 1)$ or $(L_x, L_y) = (1, N_y)$. In this case, the HbH matrix $\mathbf{W}$ coincides with the 2D-array F itself or with its transpose. Thus, the algorithm of 2D-SSA is reduced to a grouping of the SVD components of the 2D-array F. This technique is used in image processing and it works well for 2D-arrays that are products of 1D-sequences $(f(i,j) = p_i q_j)$.
2. Consider the case when either (a) or (b) is met. Let it be (b). Without loss of generality, we can assume that $L_y = 1$ and $1 < L_x < N_x$. Then the HbH matrix $\mathbf{W}$ generated by F consists of stacked Hankel matrices

$$\mathbf{W} = [\mathbf{H}_0 : \mathbf{H}_1 : \ldots : \mathbf{H}_{N_y}]$$

   and we come to the algorithm of MSSA [4, 6, 10] for simultaneous decomposition of multiple time series. More precisely, we treat the 2D-array as a set of time series arranged into columns and apply the MSSA algorithm with parameter L_x to this set of series.
   Practically, MSSA is more preferred than the general 2D-SSA if we expect only one dimension of the input 2D-array to be 'structured'.

2.3.3. Product of 1D sequences. In §2.3.1, we have shown that SSA for time series can be considered as a special case of the 2D-SSA. However, we can establish another relation between SSA and 2D-SSA. Consider the outer product of 1D-sequences as an important particular case of 2D-arrays: $f(i,j) = p_i q_j$. Products of 1D-sequences are of great importance for the general case of 2D-SSA as we can study properties (e.g. separability) of sums of products of 1D-sequences based on properties of the factors. The main fact here is that a 2D-SSA decomposition of the 2D-array $\mathrm{F} = \big(f(i,j)\big)_{i,j=0}^{N_x-1,N_y-1}$ can be expressed through SSA decompositions of the 1D-sequences $(p_i)_{i=0}^{N_x-1}$ and $(q_j)_{j=0}^{N_y-1}$.

In matrix notation, the product of two 1D-sequences $\mathrm{P} = (p_0, \ldots, p_{N_x-1})^{\mathrm{T}}$ and $\mathrm{Q} = (q_0, \ldots, q_{N_y-1})^{\mathrm{T}}$ is $\mathrm{F} = \mathrm{P}\mathrm{Q}^{\mathrm{T}}$. Let us fix window sizes (L_x, L_y) and denote by

[1] In the SSA literature, the trajectory matrix is usually denoted by $\mathbf{X}$

$\mathbf{W}^{(p)}$ and $\mathbf{W}^{(q)}$ the Hankel matrices generated by P and Q respectively:

$$\mathbf{W}^{(p)} = \begin{pmatrix} p_0 & p_1 & \dots & p_{K_x-1} \\ p_1 & p_2 & \dots & p_{K_x} \\ \vdots & \vdots & \ddots & \vdots \\ p_{L_x-1} & p_{L_x} & \dots & p_{N_x-1} \end{pmatrix}, \quad \mathbf{W}^{(q)} = \begin{pmatrix} q_0 & q_1 & \dots & q_{K_y-1} \\ q_1 & q_2 & \dots & q_{K_y} \\ \vdots & \vdots & \ddots & \vdots \\ q_{L_y-1} & q_{L_y} & \dots & q_{N_y-1} \end{pmatrix}.$$

Then the Hankel-block-Hankel matrix $\mathbf{W}$ generated by the 2D-array F is

$$\mathbf{W} = \mathbf{W}^{(q)} \otimes \mathbf{W}^{(p)}.$$

Thus, the following theorem holds.

THEOREM 2.1 ([9, Th. 13.10]). *Let $\mathbf{W}^{(p)}$ and $\mathbf{W}^{(q)}$ have singular value decompositions*

$$\mathbf{W}^{(p)} = \sum_{m=1}^{d_p} \sqrt{\lambda_m^{(p)}} U_m^{(p)} {V_m^{(p)}}^{\mathrm{T}}, \quad \mathbf{W}^{(q)} = \sum_{n=1}^{d_q} \sqrt{\lambda_n^{(q)}} U_n^{(q)} {V_n^{(q)}}^{\mathrm{T}}. \tag{2.15}$$

Then

$$\mathbf{W} = \sum_{m=1}^{d_p} \sum_{n=1}^{d_q} \sqrt{\lambda_m^{(p)} \lambda_n^{(q)}} \left(U_n^{(q)} \otimes U_m^{(p)}\right) \left(V_n^{(q)} \otimes V_m^{(p)}\right)^{\mathrm{T}} \tag{2.16}$$

yields a singular value decomposition of the matrix $\mathbf{W}$, after rearranging of its terms (in decreasing order of $\lambda_m^{(p)} \lambda_n^{(q)}$).

2.4. Comments on Grouping step. Let us now discuss perhaps the most sophisticated point of the algorithm: grouping of the eigentriples. Rules for grouping are not defined within the 2D-SSA algorithm and this step is supposed to be performed by hand, on the base of theoretical results. The way of grouping depends on the task one has to solve. The general task of 2D-SSA is to extract additive components from the observed 2D-array. Let us try to formalize this task.

Suppose we observe a sum of 2D-arrays: $\mathrm{F} = \mathrm{F}^{(1)} + \ldots + \mathrm{F}^{(m)}$. For example, F is a sum of a smooth surface, regular fluctuations and noise. When applying the 2D-SSA algorithm to F, we have to group somehow the eigentriples (i.e. to group the terms of (2.2) or (2.11)) at Grouping step. The problems arising here are:

- Is it possible to group the eigentriples providing the initial decomposition of F into $\mathrm{F}^{(k)}$?
- How to identify the eigentriples corresponding to a component $\mathrm{F}^{(k)}$?

In order to answer the first question, we introduce the notion of *separability of the 2D-arrays* $\mathrm{F}^{(1)}, \ldots, \mathrm{F}^{(m)}$ *by 2D-SSA* (following the 1D case [7]) as the possibility to extract them from their sum. In other words, we call the set of 2D-arrays separable if the answer to the first question is positive. In §3.1 we present the strict definition of separability and study its properties. In §3.2 we review some facts on separability of time series (the 1D-SSA case), establish a link between the 1D-SSA and 2D-SSA cases and deduce several important examples of 2D-SSA separability (§3.3). For practical reasons, we discuss approximate and asymptotic separability.

If components are separable, then we come to the second question: how to perform an appropriate grouping? The main idea is based on the following fact: the eigenarrays $\{\Psi_i\}_{i\in I_k}$ and factor arrays $\{\Phi_i\}_{i\in I_k}$ corresponding to a component $\mathrm{F}^{(k)}$ can be expressed as linear combinations of submatrices of the component. We can conclude that they repeat the form of the component $\mathrm{F}^{(k)}$. For example, smooth surfaces produce smooth eigenarrays (factor arrays), periodic components generate periodic eigenarrays, and so on. In §3.4 we also describe a tool of *weighted correlations* for checking separability a-posteriori. This tool can be an additional guess for grouping.

Another matter of concern is the number of eigentriples we have to gather to obtain a component $\mathrm{F}^{(k)}$. This number is called the *2D-SSA rank* of the 2D-array $\mathrm{F}^{(k)}$ and is equal to the rank of the HbH matrix generated by $\mathrm{F}^{(k)}$. Actually, we are interested in separable 2D-arrays. Clearly, they have rank-deficient HbH matrices in non-trivial case. This class of 2D-arrays has an important subclass: the 2D-arrays keeping their 2D-SSA rank constant within a range of window sizes. In the 1D case (see §2.3.1) the HbH matrices are Hankel and the subclass coincides with the whole class. For the general 2D case it is not so. However, 2D-arrays from the defined above subclass are of considerable interest since the number of eigentriples they produce does not depend on the choice of window sizes. §4 contains several examples of such 2D-arrays and rank calculations for them.

3. 2D separability. This section deals with the problem of separability stated in §2.4 as a possibility to extract terms from the observed sum. We consider the problem of separability for two 2D-arrays, $\mathrm{F}^{(1)}$ and $\mathrm{F}^{(2)}$. Let us fix window sizes (L_x, L_y) and consider the SVD of the HbH matrix $\mathbf{W}$ generated by $\mathrm{F} = \mathrm{F}^{(1)} + \mathrm{F}^{(2)}$:

$$\mathbf{W} = \sum_{i=1}^{d} \sqrt{\lambda_i} U_i V_i^{\mathrm{T}}.$$

If we denote $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ the Hankel-block-Hankel matrices generated by $\mathrm{F}^{(1)}$ and $\mathrm{F}^{(2)}$, then the problem of separability can be formulated as follows: does there exist such a grouping $\{I_1, I_2\}$ that

$$\mathbf{W}^{(1)} = \sum_{i\in I_1} \sqrt{\lambda_i} U_i V_i^{\mathrm{T}} \quad \text{and} \quad \mathbf{W}^{(2)} = \sum_{i\in I_2} \sqrt{\lambda_i} U_i V_i^{\mathrm{T}}. \tag{3.1}$$

The important point to note here is that if $\mathbf{W}$ has equal singular values, then the SVD of $\mathbf{W}$ is not unique. For this reason, we introduce two notions (in the same fashion as in [7]): *strong* and *weak separability.* Strong separability means that **any** SVD of the matrix $\mathbf{W}$ allows the desired grouping, while weak separability means that there **exists** such an SVD.

3.1. Basic definitions. Let $\mathcal{L}^{(m,n)} = \mathcal{L}^{(m,n)}(\mathrm{G})$ denote the linear space spanned by the $m\times n$ submatrices of a 2D-array G. Particulary, for fixed window sizes (L_x, L_y), we have

$$\mathcal{L}^{(L_x,L_y)}(\mathrm{F}) = \mathrm{span}(\{\mathrm{F}_{k,l}\}) \quad \text{and} \quad \mathcal{L}^{(K_x,K_y)}(\mathrm{F}) = \mathrm{span}(\{\mathrm{F}^{i,j}\}).$$

Definition 3.1. *Two 2D-arrays* $\mathrm{F}^{(1)}$ *and* $\mathrm{F}^{(2)}$ *with equal sizes are* weakly (L_x, L_y)-separable *if*

$$\mathcal{L}^{(L_x,L_y)}(\mathrm{F}^{(1)}) \perp \mathcal{L}^{(L_x,L_y)}(\mathrm{F}^{(2)}) \quad \textit{and} \quad \mathcal{L}^{(K_x,K_y)}(\mathrm{F}^{(1)}) \perp \mathcal{L}^{(K_x,K_y)}(\mathrm{F}^{(2)}).$$

Due to properties of SVDs, Definition 3.1 means that if $\mathrm{F}^{(1)}$ and $\mathrm{F}^{(2)}$ are weakly separable, then the sum of SVDs of $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ (3.1) is an SVD of the $\mathbf{W}$. We also introduce the definition of strong separability.

DEFINITION 3.2. *We call two 2D-arrays* $\mathrm{F}^{(1)}$ *and* $\mathrm{F}^{(2)}$ strongly separable *if they are weakly separable and the sets of singular values of their Hankel-block-Hankel matrices do not intersect.*

Hereafter we will speak mostly about the weak separability and will say 'separability' for short.

REMARK 3.3. *The set of 2D-arrays separable from a fixed 2D-array* F *is a linear space.*

Since the exact separability is not feasible, let us introduce the approximate separability as almost orthogonality of the corresponding subspaces. Consider 2D-arrays F and G and fix window sizes (L_x, L_y). As in (2.7), F_{k_1,l_1}, G_{k_2,l_2} stand for $L_x \times L_y$ submatrices of F and G and F^{i_1,j_1}, G^{i_2,j_2} do for $K_x \times K_y$ submatrices. Let us introduce a distance between two 2D-arrays in order to measure the approximate separability:

$$\rho^{(L_x,L_y)}(\mathrm{F},\mathrm{G}) \stackrel{\mathrm{def}}{=} \max(\rho_L, \rho_K), \tag{3.2}$$

where

$$\rho_K = \max_{(k_1,l_1),(k_2,l_2)\in J_K} \left| \frac{\langle \mathrm{F}_{k_1,l_1}, \mathrm{G}_{k_2,l_2}\rangle_{\mathcal{M}}}{\|\mathrm{F}_{k_1,l_1}\|_{\mathcal{M}}\|\mathrm{G}_{k_2,l_2}\|_{\mathcal{M}}} \right|, \quad J_K = \{1,\ldots,K_x\}\times\{1,\ldots,K_y\};$$

$$\rho_L = \max_{(i_1,j_1),(i_2,j_2)\in J_L} \left| \frac{\langle \mathrm{F}^{i_1,j_1}, \mathrm{G}^{i_2,j_2}\rangle_{\mathcal{M}}}{\|\mathrm{F}^{i_1,j_1}\|_{\mathcal{M}}\|\mathrm{G}^{i_2,j_2}\|_{\mathcal{M}}} \right|, \quad J_L = \{1,\ldots,L_x\}\times\{1,\ldots,L_y\}.$$

REMARK 3.4. *The 2D-arrays* F *and* G *are separable iff* $\rho^{(L_x,L_y)}(\mathrm{F},\mathrm{G}) = 0$.

A quite natural way to deal with approximate separability is studying asymptotic by array sizes separability of 2D-arrays, namely 'good' approximate separability for relatively big 2D-arrays. Consider two infinite 2D-arrays $\mathcal{F} = (f_{ij})_{i,j=0}^{\infty,\infty}$ and $\mathcal{G} = (g_{ij})_{i,j=0}^{\infty,\infty}$. Let $\mathcal{F}|_{m,n}$ and $\mathcal{G}|_{m,n}$ denote finite submatrices of infinite 2D-arrays $\mathcal{F}$ and $\mathcal{G}$: $\mathcal{F}|_{m,n} = (f_{ij})_{i,j=0}^{m-1,n-1}$, $\mathcal{G}|_{m,n} = (g_{ij})_{i,j=0}^{m-1,n-1}$.

DEFINITION 3.5. $\mathcal{F}$ *and* $\mathcal{G}$ *are said to be* asymptotically separable *if*

$$\lim_{N_x,N_y\to\infty} \rho^{(L_x,L_y)}(\mathcal{F}|_{N_x,N_y}, \mathcal{G}|_{N_x,N_y}) = 0 \tag{3.3}$$

for any $L_x = L_x(N_x,N_y)$ *and* $L_y = L_y(N_x,N_y)$ *such that* $L_x, K_x, L_y, K_y \to \infty$ *as* $N_x, N_y \to \infty$.

3.2. Separability of 1D sequences. As well as the original 1D-SSA algorithm can be treated as a special case of 2D-SSA, the notion of *L-separability of time series* (originally introduced in [7]) is a special case of (L_x, L_y)-separability.

REMARK 3.6. *Time series* $\mathrm{F}^{(1)} = (f_0^{(1)},\ldots,f_{N-1}^{(1)})^{\mathrm{T}}$ *and* $\mathrm{F}^{(2)} = (f_0^{(2)},\ldots,f_{N-1}^{(2)})^{\mathrm{T}}$ *are L-separable if they are $(L,1)$-separable as 2D-arrays.*

Let us now give several examples of the (weak) L-separability, which is thoroughly studied in [7].

EXAMPLE 3.7. The sequence $\mathrm{F} = (f_0,\ldots,f_{N-1})^{\mathrm{T}}$ with $f_n = \cos(2\pi\omega n + \varphi)$ is L-separable from a non-zero constant sequence $(c,\ldots,c)^{\mathrm{T}}$ if $L\omega$ and $K\omega$, where $K = N - L + 1$, are integers.

EXAMPLE 3.8. Two cosine sequences of length N given by

$$f_n^{(1)} = \cos(2\pi\omega_1 n + \varphi_1) \quad \text{and} \quad f_n^{(2)} = \cos(2\pi\omega_2 n + \varphi_2)$$

are L-separable if $\omega_1 \neq \omega_2$, $0 < \omega_1, \omega_2 \le 1/2$ and $L\omega_1, L\omega_2, K\omega_1, K\omega_2$ are integers.

In general, there are only a small number of exact separability examples. Hence, we come to consideration of approximate separability. It is studied with the help of *asymptotic separability of time series* first introduced in [7]. Asymptotic separability is defined in the same fashion as that in the 2D case (see Definition 3.5). The only difference is that we let just one dimension (and parameter) tend to infinity (because another dimension is fixed).

EXAMPLE 3.9. Two cosine sequences given by

$$f_n^{(l)} = \sum_{k=0}^{m} c_k^{(l)} \cos(2\pi\omega_k^{(l)} n + \varphi_k^{(l)}), \quad 0 < \omega_k^{(l)} \le 1/2,\ l = 1, 2, \tag{3.4}$$

with different frequencies are asymptotically separable.

In Table 3.1, one can see a short summary on asymptotic separability of time series.

TABLE 3.1
Asymptotic separability

	const	cos	exp	exp cos	poly
const	$-$	$+$	$+$	$+$	$-$
cos	$+$	$+$	$+$	$+$	$+$
exp	$+$	$+$	$+$	$+$	$+$
exp cos	$+$	$+$	$+$	$+$	$+$
poly	$-$	$+$	$+$	$+$	$-$

In this table, **const** stands for non-zero constant sequences, **cos** does for cosine sequences (3.4), **exp** denotes sequences $\exp(\alpha n)$, **exp cos** stands for $e^{\alpha n} \cos(2\pi\omega n + \phi)$ and **poly** does for polynomial sequences. Note that conditions of separability are omitted in the table. For more details, such as conditions, convergence rates, and other types of separability (e.g. stochastic separability of a deterministic signal from the white noise), see [7].

3.3. Products of 1D sequences. Let us study separability properties for products of 1D-sequences (introduced in §2.3.3). Consider four 1D-sequences

$$\mathrm{P}^{(1)} = (p_0^{(1)}, \ldots, p_{N_x-1}^{(1)})^{\mathrm{T}}, \quad \mathrm{Q}^{(1)} = (q_0^{(1)}, \ldots, q_{N_y-1}^{(1)})^{\mathrm{T}},$$
$$\mathrm{P}^{(2)} = (p_0^{(2)}, \ldots, p_{N_x-1}^{(2)})^{\mathrm{T}}, \quad \mathrm{Q}^{(2)} = (q_0^{(2)}, \ldots, q_{N_y-1}^{(2)})^{\mathrm{T}}.$$

PROPOSITION 3.10. *If* $\mathrm{P}^{(1)}$ *and* $\mathrm{P}^{(2)}$ *are* L_x*-separable or sequences* $\mathrm{Q}^{(1)}$ *and* $\mathrm{Q}^{(2)}$ *are* L_y*-separable, then their products* $\mathrm{F}^{(1)} = \mathrm{P}^{(1)}(\mathrm{Q}^{(1)})^{\mathrm{T}}$ *and* $\mathrm{F}^{(2)} = \mathrm{P}^{(2)}(\mathrm{Q}^{(2)})^{\mathrm{T}}$ *are* (L_x, L_y)*-separable.*

Proof. First of all, let us notice that submatrices of the 2D-arrays are products of subvectors of 1D-sequences

$$\begin{aligned} \mathrm{F}_{k_1,l_1}^{(1)} &= (p_{k_1-1}^{(1)}, \ldots, p_{k_1+L_x-2}^{(1)})^{\mathrm{T}} (q_{l_1-1}^{(1)}, \ldots, q_{l_1+L_y-2}^{(1)}), \\ \mathrm{F}_{k_2,l_2}^{(2)} &= (p_{k_2-1}^{(2)}, \ldots, p_{k_2+L_x-2}^{(2)})^{\mathrm{T}} (q_{l_2-1}^{(2)}, \ldots, q_{l_2+L_y-2}^{(2)}). \end{aligned} \tag{3.5}$$

Let us recall an important feature of Frobenius inner product:

$$\left\langle AB^{\mathrm{T}}, CD^{\mathrm{T}} \right\rangle_{\mathcal{M}} = \langle A, C \rangle_2 \langle B, D \rangle_2, \tag{3.6}$$

where A, B, C, and D are vectors.

Applying (3.6) to (3.5), we obtain the orthogonality of all $L_x \times L_y$ submatrices of 2D-arrays:

$$\left\langle \mathrm{F}^{(1)}_{k_1,l_1}, \mathrm{F}^{(2)}_{k_2,l_2} \right\rangle_{\mathcal{M}} = 0.$$

Likewise, all their $K_x \times K_y$ submatrices are orthogonal too. According to Remark 3.4, we conclude that the 2D-arrays $\mathrm{F}^{(1)}$ and $\mathrm{F}^{(2)}$ are separable, and the proof is complete. ∎

Furthermore, we can generalize Proposition 3.10 to approximate and asymptotic separability.

LEMMA 3.11. *Under the assumptions of Proposition 3.10,*

$$\rho^{(L_x,L_y)}(\mathrm{F}^{(1)}, \mathrm{F}^{(2)}) \leq \rho^{L_x}(\mathrm{P}^{(1)}, \mathrm{P}^{(2)}) \rho^{L_y}(\mathrm{Q}^{(1)}, \mathrm{Q}^{(2)}).$$

Proof. Equalities (3.5) and (3.6) make the proof obvious. ∎

PROPOSITION 3.12. *Let $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$ be products of infinite 1D-sequences:*

$$\mathcal{F}^{(1)} = \mathcal{P}^{(1)}(\mathcal{Q}^{(1)})^{\mathrm{T}}, \quad \mathcal{F}^{(2)} = \mathcal{P}^{(2)}(\mathcal{Q}^{(2)})^{\mathrm{T}},$$
$$\mathcal{P}^{(j)} = (p^{(j)}_0, \ldots, p^{(j)}_n, \ldots)^{\mathrm{T}} \quad \textit{and} \quad \mathcal{Q}^{(j)} = (q^{(j)}_0, \ldots, q^{(j)}_n, \ldots)^{\mathrm{T}}.$$

If $\mathcal{P}^{(1)}, \mathcal{P}^{(2)}$ or $\mathcal{Q}^{(1)}, \mathcal{Q}^{(2)}$ are asymptotically separable, then $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$ are asymptotically separable too.

Proof. The proposition follows immediately from Lemma 3.11. ∎

The following example of asymptotic separability can be shown using Proposition 3.12 and Remark 3.3.

EXAMPLE 3.13. The 2D-array given by $f^{(1)}(i,j) = \cos(2\pi\omega_1 i)\ln(j+1) + \ln(i+1)\cos(2\pi\omega_2 j)$ is asymptotically separable from a constant 2D-array $f^{(2)}(i,j) = \textit{const}$.

Example 3.13 demonstrates that separability in the 2D case is more varied than in the 1D case. For instance, nothing but periodic 1D-sequences are separable from a constant sequence.

The next example is an analogue of Example 3.9.

EXAMPLE 3.14. Two 2D sine-wave arrays given by

$$f^{(l)}(i,j) = \sum_{k=1}^{m} c^{(l)}_k \cos(2\pi\omega^{(l)}_{1k} i + \varphi^{(l)}_{1k}) \cos(2\pi\omega^{(l)}_{2k} j + \varphi^{(l)}_{2k}),\ l = 1, 2,$$

with different frequencies are asymptotically separable by 2D-SSA.

However, the problem of lack of strong separability in presence of weak separability appears more frequently in the 2D case. The wider is the range of eigenvalues of the HbH matrix corresponding to a 2D-array, the more likely is mixing of components produced by the 2D-array and other constituents. This becomes a problem at Grouping step. For example, if two 1D-sequences have eigenvalues from the range $[\lambda_2, \lambda_1]$, then the range of eigenvalues of their product, by Proposition 2.1, is wider: $[\lambda_2^2, \lambda_1^2]$.

3.4. Checking the separability: weighted correlations. Following the 1D case, we introduce a necessary condition of separability, which can be applied in practice.

DEFINITION 3.15. *A* weighted inner product *of 2D-arrays* $\mathrm{F}^{(1)}$ *and* $\mathrm{F}^{(2)}$ *is defined as follows:*

$$\left\langle \mathrm{F}^{(1)}, \mathrm{F}^{(2)} \right\rangle_w \stackrel{\text{def}}{=} \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} f^{(1)}(i,j) \cdot f^{(2)}(i,j) \cdot w_x(i) \cdot w_y(j),$$

where

$$w_x(i) = \min(i+1, L_x, K_x, N_x - i) \quad \textit{and} \quad w_y(j) = \min(j+1, L_y, K_y, N_y - j).$$

In fact, the functions $w_x(i)$ and $w_y(j)$ define the number of entries on secondary diagonals of Hankel $L_x \times K_x$ and $L_y \times K_y$ matrices respectively. More precisely,

$$w_x(i) = \#\big\{(k,l) :\ 1 \le k \le K_x, 1 \le l \le L_x,\ k+l = i+1\big\},$$
$$w_y(j) = \#\big\{(k,l) :\ 1 \le k \le K_y, 1 \le l \le L_y,\ k+l = j+1\big\}.$$

Hence, for a Hankel-block-Hankel matrix $\mathbf{W}$ generated by F, the product $w_x(i)w_y(j)$ is equal to the number of entries in $\mathbf{W}$ corresponding to the entry (i,j) of the 2D-array F. The same holds for the number of entries in a 2D-trajectory matrix $\mathbf{X}$. This observation implies the following proposition.

PROPOSITION 3.16.

$$\left\langle \mathrm{F}^{(1)}, \mathrm{F}^{(2)} \right\rangle_w = \left\langle \mathbf{X}^{(1)}, \mathbf{X}^{(2)} \right\rangle_{\mathcal{M}} = \left\langle \mathbf{W}^{(1)}, \mathbf{W}^{(2)} \right\rangle_{\mathcal{M}}.$$

With the help of the weighted inner product, we can formulate a necessary condition for separability.

PROPOSITION 3.17. *If* $\mathrm{F}^{(1)}$ *and* $\mathrm{F}^{(2)}$ *are separable, then* $\left\langle \mathrm{F}^{(1)}, \mathrm{F}^{(2)} \right\rangle_w = 0$.

Finally, we introduce weighted correlations to measure approximate separability and the matrix of weighted correlations to provide an additional information useful for grouping.

DEFINITION 3.18. *A* weighted correlation (w-correlation) ρ_w *between two 2D-arrays* $\mathrm{F}^{(1)}$ *and* $\mathrm{F}^{(2)}$ *is defined as*

$$\rho_w(\mathrm{F}^{(1)}, \mathrm{F}^{(2)}) = \frac{\left\langle \mathrm{F}^{(1)}, \mathrm{F}^{(2)} \right\rangle_w}{\|\mathrm{F}^{(1)}\|_w \|\mathrm{F}^{(2)}\|_w}.$$

Consider the 2D-array F and apply 2D-SSA with parameters (L_x, L_y). If we choose the maximal grouping (2.3), namely $m = d$ and $I_k = \{k\}, 1 \le k \le d$, then each $\widetilde{\mathrm{F}}_{I_k}$ is called the kth *elementary reconstructed component* and the *matrix of weighted correlations* $\mathrm{R} = (r_{ij})_{i,j=1}^d$ is given by

$$r_{ij} = |\rho_w(\widetilde{\mathrm{F}}_{I_i}, \widetilde{\mathrm{F}}_{I_j})|. \tag{3.7}$$

For an example of application see §5.

4. 2D-SSA ranks of 2D-arrays. Examples of calculation.

4.1. Basic properties. Let us first introduce a definition of the *2D-SSA rank.*

DEFINITION 4.1. *The* (L_x, L_y)-rank *(*2D-SSA rank *for window sizes* (L_x, L_y)*) of the 2D-array* F *is defined to be*

$$\operatorname{rank}_{L_x,L_y}(\mathrm{F}) \stackrel{\mathrm{def}}{=} \dim \mathcal{L}^{(L_x,L_y)} = \dim \mathcal{L}^{(K_x,K_y)} = \operatorname{rank} \mathbf{W}.$$

It is immediate that the (L_x, L_y)-rank is equal to the number of components in the SVD (2.2) of the Hankel-block-Hankel matrix generated by F. There is another way to express the rank through the 2D-trajectory matrix (2.10).

LEMMA 4.2. *If for fixed window sizes* (L_x, L_y) *there exists representation*

$$\mathbf{X} = \sum_{i=1}^{m} \mathrm{A}_i \otimes \mathrm{B}_i, \quad \mathrm{B}_i \in \mathcal{M}_{L_x,L_y}, \quad \mathrm{A}_i \in \mathcal{M}_{K_x,K_y}, \tag{4.1}$$

then $\operatorname{rank}_{L_x,L_y} \mathrm{F}$ *does not exceed* m*. Furthermore, if each system* $\{\mathrm{A}_i\}_{i=1}^m$, $\{\mathrm{B}_i\}_{i=1}^m$ *is linearly independent, then* $\operatorname{rank}_{L_x,L_y}(\mathrm{F}) = m$*.*

Proof. The proof is evident, since equality (4.1) can be rewritten as

$$\mathbf{W} = \sum_{i=1}^{m} \operatorname{vec} \mathrm{B}_i (\operatorname{vec} \mathrm{A}_i)^{\mathrm{T}}$$

by (1.6). □

By Theorem 2.1, the 2D-SSA rank of a product of 1D-sequences 2D-SSA rank is equal to the product of the ranks:

$$\operatorname{rank}_{L_x,L_y}(\mathrm{P}\mathrm{Q}^{\mathrm{T}}) = \operatorname{rank}_{L_x}(\mathrm{P}) \operatorname{rank}_{L_y}(\mathrm{Q}), \tag{4.2}$$

where $\operatorname{rank}_L(\cdot)$ stands for $\operatorname{rank}_{L,1}(\cdot)$.

For a sum of products of 1D-sequences $\mathrm{F} = \sum\limits_{i=1}^{n} \mathrm{P}^{(i)}(\mathrm{Q}^{(i)})^{\mathrm{T}}$, the 2D-SSA rank is not generally equal to the sum of products of ranks due to possible linear dependence of vectors. In order to calculate 2D-SSA ranks for this kind of 2D-arrays, the following lemma may be useful.

LEMMA 4.3. *If for fixed window sizes* (L_x, L_y) *there exist linearly independent systems* $\{\mathrm{A}_j\}_{j=1}^n$ *and* $\{\mathrm{B}_i\}_{i=1}^m$ *such that*

$$\mathbf{X} = \sum_{i,j=1}^{m,n} c_{ij} \mathrm{A}_j \otimes \mathrm{B}_i, \ \mathrm{B}_i \in \mathcal{M}_{L_x,L_y}, \ \mathrm{A}_j \in \mathcal{M}_{K_x,K_y}, \tag{4.3}$$

then $\operatorname{rank}_{L_x,L_y}(\mathrm{F}) = \operatorname{rank} \mathbf{C}$*, where* $\mathbf{C} = (c_{ij})_{i,j=1}^{m,n}$*.*

Proof. Let us rewrite the condition (4.3) in the same way as in the proof of Lemma 4.2:

$$\mathbf{W} = \sum_{i,j=1}^{m,n} c_{ij} \operatorname{vec} \mathrm{B}_i (\operatorname{vec} \mathrm{A}_j)^{\mathrm{T}}.$$

If we set $\mathbf{A} = [\operatorname{vec} \mathrm{A}_1 : \ldots : \operatorname{vec} \mathrm{A}_n]$ and $\mathbf{B} = [\operatorname{vec} \mathrm{B}_1 : \ldots : \operatorname{vec} \mathrm{B}_m]$, then $\mathbf{W} = \mathbf{B}\mathbf{C}\mathbf{A}^{\mathrm{T}}$. Since $\mathbf{A}$ and $\mathbf{B}$ have linearly independent columns, the ranks of $\mathbf{W}$ and $\mathbf{C}$ coincide. □

4.2. Ranks of time series. In the 1D case, class of series having constant rank within a range of window length is called *time series of finite rank* [7]. This class mostly consist of sums of products of polynomials, exponents and cosines:

$$f_n = \sum_{k=1}^{d'} P_{m_k}^{(k)}(n)\,\rho_k^n \cos(2\pi\omega_k n + \varphi_k) + \sum_{k=d'+1}^{d} P_{m_k}^{(k)}(n)\,\rho_k^n. \tag{4.4}$$

Here $0 < \omega_k < 0.5$, $\rho_k \neq 0$, and $P_l^{(k)}$ are polynomials of degree l. The time series (4.4) form the class of time series *governed by linear recurrent formulae* (see [3, 7]).

It happens that SSA ranks of time series like (4.4) can be explicitly calculated.

Proposition 4.4. *Let a time series* $\mathrm{F}_N = (f_0, ..., f_{N-1})$ *be defined in (4.4) with* $(\omega_k, \rho_k) \neq (\omega_l, \rho_l)$ *for* $1 \leq k, l \leq d'$ *and* $\rho_k \neq \rho_l$ *for* $d' < k, l \leq d$. *Then* $\mathrm{rank}_L(\mathrm{F}_N)$ *is equal to*

$$r = 2\sum_{k=1}^{d'}(m_k+1) + \sum_{k=d'+1}^{d}(m_k+1) \tag{4.5}$$

if $L \geq r$ *and* $K \geq r$.

Proof. Equality (4.4) can be rewritten as a sum of complex exponents:

$$f_n = \sum_{k=1}^{d'} P_{m_k}^{(k)}(n)\,(\alpha_k(\lambda_k)^n + \beta_k(\lambda_k')^n) + \sum_{k=d'+1}^{d} P_{m_k}^{(k)}(n)\,\rho_k^n,$$

where $\lambda_k = \rho_k e^{2\pi i\omega_k}$, $\lambda_k' = \rho_k e^{-2\pi i\omega_k}$ and $\alpha_k, \beta_k \neq 0$. The latter equality yields a canonical representation (see [1, §8]) of the Hankel matrix $\mathbf{W}$ with rank r. Under the stated conditions on L and K, $\mathrm{rank}\,\mathbf{W} = r$ by [1, Theorem 8.1]. ☐

4.3. Calculation of 2D-SSA ranks. Proposition 4.4 together with (4.2) gives possibility to calculate 2D-SSA ranks for 2D-arrays that are products of 1D-sequences. However, the general 2D case is much more complicated. In this section, we introduce results concerning 2D-SSA ranks for 2D exponential, polynomial and sine-wave arrays.

In the examples below, one can observe the effect that the 2D-SSA rank of a 2D-array given by $f(i,j) = p_{i+j}$ is equal to the SSA rank of the sequence (p_i). It is not surprising, since 2D-SSA is in general invariant to rotation (and to other linear maps) of arguments of a 2D-function $f(i,j)$.

4.3.1. Exponent. The result on rank of a sum of 2D exponents is quite simple.

Proposition 4.5. *For an exponential 2D-array* $\mathrm{F} = \big(f(i,j)\big)_{i,j=0}^{N_x-1,N_y-1}$ *defined by*

$$f(i,j) = \sum_{n=1}^{m} c_k \rho_n^i \mu_n^j, \qquad \rho_n, \mu_n \neq 0, \tag{4.6}$$

$\mathrm{rank}_{L_x,L_y}(\mathrm{F}) = m$ *if* $L_x, L_y, K_x, K_y \geq m$ *and* $(\rho_l, \mu_l) \neq (\rho_k, \mu_k)$ *for* $l \neq k$.

Proof. The proof is based on Lemma 4.2. Let us express entries of the matrix $\mathbf{X}$ using equality (2.9):

$$(\mathrm{F}_{k,l})_{i,j} = f(i+k-2, j+l-2) = \sum_{n=1}^{m} c_n \rho_n^{(i-1)} \mu_n^{(j-1)} \rho_n^{(k-1)} \mu_n^{(l-1)}. \tag{4.7}$$

It is easy to check that equality (4.7) defines decomposition

$$\begin{aligned} \mathbf{X} &= \sum_{n=1}^{m} \mathrm{A}_n \otimes \mathrm{B}_n, \quad \text{where} \\ \mathrm{A}_n &= (\rho_n^0, \ldots, \rho_n^{(K_x-1)})^{\mathrm{T}}(\mu_n^0, \ldots, \mu_n^{(K_y-1)}), \\ \mathrm{B}_n &= (\rho_n^0, \ldots, \rho_n^{(L_x-1)})^{\mathrm{T}}(\mu_n^0, \ldots, \mu_n^{(L_y-1)}). \end{aligned}$$

Obviously, each system $\{\mathrm{A}_i\}_{i=1}^m$, $\{\mathrm{B}_i\}_{i=1}^m$ is linearly independent. Applying Lemma 4.2 finishes the proof. ∎

4.3.2. Polynomials. Let P_m be a polynomial of degree m:

$$P_m(i,j) = \sum_{s=0}^{m} \sum_{t=0}^{m-s} g_{st} i^s j^t$$

and at least one of leading coefficients $g_{s,m-s}$ for $s = 0, \ldots, m$ is non-zero. Consider the 2D-array F of sizes $N_x, N_y > 2m+1$ with $f(i,j) = P_m(i,j)$.

PROPOSITION 4.6. *If $L_x, L_y, K_x, K_y \geq m+1$, then*

$$\mathrm{rank}_{L_x,L_y}(\mathrm{F}) = \mathrm{rank}_{m+1,m+1}(\mathrm{G}'),$$

where

$$\mathrm{G}' = \begin{pmatrix} \mathrm{G}'' & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{m\times m} \end{pmatrix}, \quad \mathrm{G}'' = \begin{pmatrix} g'_{00} & \cdots & g'_{0m} \\ \vdots & \iddots & \\ g'_{m0} & & \mathbf{0} \end{pmatrix}, \quad g'_{st} = g_{st} s!\, t!.$$

In addition, the following inequality hold:

$$m+1 \leq \mathrm{rank}_{L_x,L_y}(\mathrm{F}) \leq \begin{cases} (m/2+1)^2, & \text{for even } m, \\ ((m+1)/2+1)\,(m+1)/2, & \text{for odd } m. \end{cases} \tag{4.8}$$

Proof. The first part of the proposition is proved in the same way as Proposition 4.5 except for using Lemma 4.3 instead of Lemma 4.2. Let us apply Taylor formula

$$\begin{aligned} (\mathrm{F}_{k,l})_{i,j} &= P_m(i+k-2, j+l-2) = \\ &= \sum_{s=0}^{m}\sum_{t=0}^{m} (i-1)^s (j-1)^t \frac{1}{s!\,t!} \left(\frac{\partial^{s+t} P_m}{\partial i^s \partial j^t}\right)(k-1, l-1) = \\ &= \sum_{s=0}^{m}\sum_{t=0}^{m} \frac{(i-1)^s}{s!} \frac{(j-1)^t}{t!} \sum_{u=0}^{m-s}\sum_{v=0}^{m-t} g_{u+s,v+t} (u+s)!(v+t)! \frac{(k-1)^u}{u!} \frac{(l-1)^v}{v!}. \end{aligned} \tag{4.9}$$

If we set $g'_{st} = 0$ for $s+t > m+1$, then we can rewrite (4.9) as

$$\mathbf{X} = \sum_{s,t,u,v=0}^{m} g'_{u+s,v+t} \mathrm{A}_{u+(m+1)v} \otimes \mathrm{B}_{s+(m+1)t}, \quad \text{where} \tag{4.10}$$

$$\begin{aligned} \mathrm{A}_{u+(m+1)v} &= \frac{1}{u!\ v!} \left(0^u, \ldots, (K_x-1)^u\right)^{\mathrm{T}} \left(0^v, \ldots, (K_y-1)^v\right) \text{ for } 0 \leq u,v \leq m \\ \mathrm{B}_{s+(m+1)t} &= \frac{1}{s!\ t!} \left(0^s, \ldots, (L_x-1)^s\right)^{\mathrm{T}} \left(0^t, \ldots, (L_y-1)^t\right) \text{ for } 0 \leq s,t \leq m. \end{aligned}$$

Let $\mathbf{W}^{(g)}$ be the Hankel-block-Hankel matrix generated by G' with window sizes $(m+1, m+1)$. Then (4.10) can be rewritten as

$$\mathbf{X} = \sum_{i,j=0}^{(m+1)^2-1} (\mathbf{W}^{(g)})_{ji} \mathrm{A}_i \otimes \mathrm{B}_j.$$

The systems $\{\mathrm{A}_i\}_{i=0}^{(m+1)^2-1}$ and $\{\mathrm{B}_j\}_{j=0}^{(m+1)^2-1}$ are linearly independent due to restrictions on L_x, L_y. By Lemma 4.3, the first part of the proposition is proved.

The bounds in (4.8) can be proved using the fact that

$$\mathrm{rank}_{m+1,m+1}(\mathrm{G}') = \dim \mathcal{L}^{(m+1,m+1)}(\mathrm{G}') = \dim \mathrm{span}\left(\{\mathrm{G}'_{k,l}\}_{k,l=1}^{m+1,m+1}\right),$$

where $\mathrm{G}'_{k,l}$ is the $(m+1)\times(m+1)$ submatrix of G' beginning from the entry (k,l). Define by $\mathcal{T}_n$ the space of $(m+1)\times(m+1)$ matrices with zero entries below the nth secondary diagonal:

$$\mathcal{T}_n \stackrel{\mathrm{def}}{=} \{\mathrm{A} = (a_{ij})_{i,j=0}^{m,m} \in \mathcal{M}_{m+1,m+1} : a_{ij} = 0 \quad \text{for} \quad i+j > n\}.$$

Then $\mathrm{G}'_{k,l}$ belongs to $\mathcal{T}_n$ for $n \geq m-(k+l)+2$ and does not, in general, for smaller n. Let us introduce

$$\mathcal{C}_n \stackrel{\mathrm{def}}{=} \mathrm{span}\left(\{\mathrm{G}'_{k,l}\}_{k+l=m-n+2}\right) \subseteq \mathcal{T}_n,$$
$$\mathcal{S}_n \stackrel{\mathrm{def}}{=} \mathrm{span}(\mathcal{C}_0, \ldots, \mathcal{C}_n) = \mathrm{span}(\mathcal{S}_{n-1}, \mathcal{C}_n) \subseteq \mathcal{T}_n.$$

Then $\mathcal{L}^{(m+1,m+1)}(\mathrm{G}') = \mathcal{S}_m$.

By the theorem conditions, there exists i such that $g_{i,m-i} \neq 0$. Hence, there exist $\mathrm{C}_0, \ldots, \mathrm{C}_m \in \mathcal{M}_{m+1,m+1}$ such that $\mathrm{C}_n \in \mathcal{C}_n \subseteq \mathcal{T}_n, \mathrm{C}_n \notin \mathcal{T}_{n-1}$. Therefore, the system $\{\mathrm{C}_0, \ldots, \mathrm{C}_m\}$ is linearly independent and the lower bound is proved.

To prove the upper bound, note that

$$\dim \mathcal{S}_n \leq \min(\dim \mathcal{S}_{n-1} + \dim \mathcal{C}_n, \dim \mathcal{T}_n).$$

Since $\dim \mathcal{C}_n \leq m+1-n$ and $\dim \mathcal{T}_n = \sum_{k=1}^{n+1} k$, one can show that

$$\dim \mathcal{S}_m \leq \sum_{n=0}^{m} \min(n+1, m-n+1) = \begin{cases} (m/2+1)^2, & \text{for even } m, \\ ((m+1)/2+1)(m+1)/2, & \text{for odd } m. \end{cases}$$

□

Let us demonstrate two example that meet the bounds in inequality (4.8) exactly: the 2D-SSA rank of the 2D array given by $f(k,l) = (k+l)^2$ $(m=2)$ equal to 3, while the 2D-SSA rank for $f(k,l) = kl$ is equal to 4.

4.3.3. Sine-wave 2D-arrays. Consider a sum of sine-wave functions

$$h_d(k,l) = \sum_{m=1}^{d} A_m(k,l), \tag{4.11}$$

$$A_m(k,l) = \begin{pmatrix} \cos(2\pi\omega_m^{(X)}k) \\ \sin(2\pi\omega_m^{(X)}k) \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} a_m & b_m \\ c_m & d_m \end{pmatrix} \begin{pmatrix} \cos(2\pi\omega_m^{(Y)}l) \\ \sin(2\pi\omega_m^{(Y)}l) \end{pmatrix}, \tag{4.12}$$

where $1 \le k \le N_x$, $1 \le l \le N_y$, at least one coefficient in each group $\{a_m, b_m, c_m, d_m\}$ is non-zero and the frequencies meet the following conditions:

$$(\omega_n^{(X)}, \omega_n^{(Y)}) \neq (\omega_m^{(X)}, \omega_m^{(Y)}), \quad \text{for } n \neq m,\ \omega_m^{(X)}, \omega_m^{(Y)} \in (0, 1/2). \tag{4.13}$$

PROPOSITION 4.7. *For window sizes (L_x, L_y) such that $L_x, L_y, K_x, K_y \ge 4d$ the 2D-SSA rank of* $\mathrm{F} = \big(h_d(k,l)\big)_{k,l=0}^{N_x-1,N_y-1}$ *is equal to*

$$\operatorname{rank}_{L_x,L_y}(\mathrm{F}) = \sum_{m=1}^{d} \nu_m, \quad \textit{where} \quad \nu_m = 2 \textit{ or } 4;$$

and numbers ν_m can be expressed as

$$\nu_m = 2 \operatorname{rank} \begin{pmatrix} a_m & b_m & c_m & d_m \\ d_m & -c_m & -b_m & a_m \end{pmatrix}. \tag{4.14}$$

Proof. Summands A_m of (4.12) can be rewritten as a sum of complex exponents:

$$\begin{aligned} 4A_m(k,l) = {} & (a_m - d_m - i(c_m + b_m))\, e^{\ 2\pi i \omega_m^{(X)} k} e^{\ 2\pi i \omega_m^{(Y)} l} + \\ & + (a_m - d_m + i(c_m + b_m))\, e^{-2\pi i \omega_m^{(X)} k} e^{-2\pi i \omega_m^{(Y)} l} + \\ & + (a_m + d_m + i(c_m - b_m))\, e^{-2\pi i \omega_m^{(X)} k} e^{\ 2\pi i \omega_m^{(Y)} l} + \\ & + (a_m + d_m - i(c_m - b_m))\, e^{\ 2\pi i \omega_m^{(X)} k} e^{-2\pi i \omega_m^{(Y)} l}. \end{aligned}$$

Note that the coefficients of the first pair of complex exponents become zero at once if $a_m = d_m$ and $b_m = -c_m$. The second pair of complex exponents vanishes if $a_m = -d_m$ and $b_m = c_m$. Therefore, the number of non-zero coefficients of the complex exponents corresponding to each summand $A_m(k,l)$ is equal to ν_m defined in (4.14). Then the 2D-array can be represented as a sum of products:

$$h_d(k,l) = \sum_{n=1}^{r} x_n {y_n}^k {z_n}^l, \quad r = \sum_{m=1}^{d} \nu_m, \tag{4.15}$$

where all the coefficients $x_n \in \mathbb{C}$ are non-zero, while y_n and z_n have the form $y_n = e^{2\pi i \omega_n^{(X)}}$, $z_n = e^{2\pi i \omega_n^{(Y)}}$, and pairs (y_n, z_n) are distinct due to conditions (4.13), namely $(y_n, z_n) \neq (y_m, z_m)$ for $n \neq m$.

Due to [5], the rank of the Hankel-block-Hankel matrix $\mathbf{W}$ generated by the 2D-array (4.15) is equal to r at least for $L_x, L_y \ge 4d$. □

Note that the condition $L_x, L_y \ge 4d$ is just sufficient for the result of Proposition 4.7. The same result is valid for a larger range of L_x, L_y; this range depends on the input 2D array, see [5] for the case of complex exponents.

Let us apply the proposition to two examples. Let $f(k,l) = \cos(2\pi\omega^{(X)}k + 2\pi\omega^{(Y)}l)$. Then the 2D-SSA rank equals 2. If $f(k,l) = \cos(2\pi\omega^{(X)}k)\cos(2\pi\omega^{(Y)}l)$, then the 2D-SSA rank equals 4.

5. Example of analysis. Consider a real-life digital image of Mars (275×278) obtained by web-camera[2] (see Fig. 5.1). As one can see, the image is corrupted by a kind of periodic noise, probably sinusoidal due to possible electromagnetic nature of noise. Let us try to extract this noise by 2D-SSA. It is more suitable to use the 2D-trajectory matrix notation. After choosing window sizes $(25, 25)$ we obtain expansion

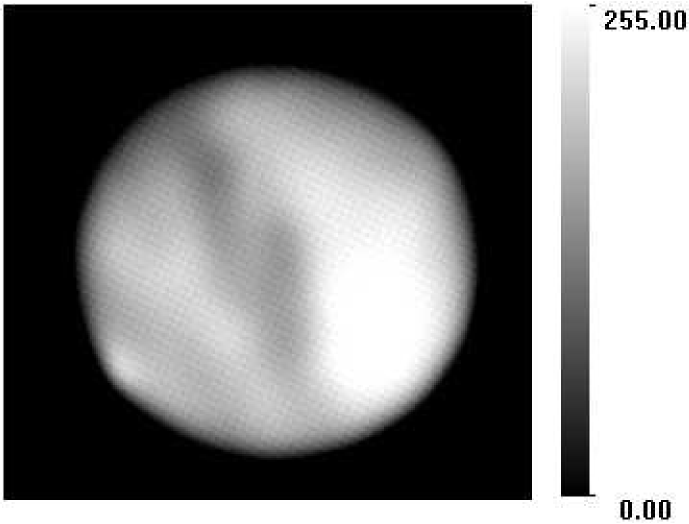

FIG. 5.1. *2D-array: Mars*

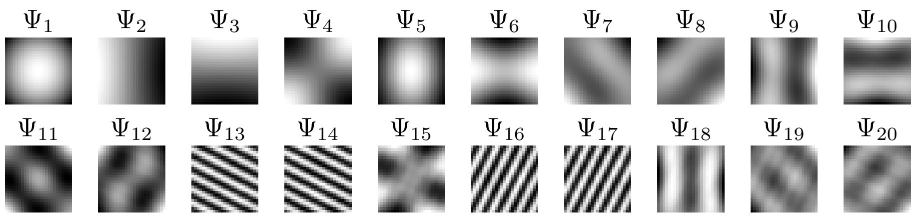

FIG. 5.2. *Eigenarrays*

(2.11). As we will show, these window sizes are enough for separation of periodic noise.

Let us look at the eigenarrays (Fig. 5.2). The eigenarrays from the eigentriples with indices $\mathcal{N} = \{13, 14, 16, 17\}$ have periodic structure similar to the noise. The factor arrays have the same periodicity too. This observation entitles us to believe that these eigentriples constitute the periodic noise. In addition, 4 is a likely rank for sine-wave 2D-arrays.

Let us validate our conjecture examining the plot of weighted correlations matrix (see Fig. 5.3). The plot depicts w-correlations r_{ij} (3.7) between elementary reconstructed components (the left-top corner represents the entry r_{11}). Values are plotted in grayscale, white stands for 0 and black does for 1.

The plot contains two blocks uncorrelated to the rest. This means that the sum of elementary reconstructed components corresponding to indices from $\mathcal{N}$ is separable from the rest. Reconstruction of a 2D-array by the set $\mathcal{N}$ gives us the periodic noise, while the residual produces a filtered image.

As the matter of fact, the noise is not pure periodic and is in a sense modulated. This happens due to clipping of the signal values range to $[0, 255]$.

[2]Source: Pierre Thierry

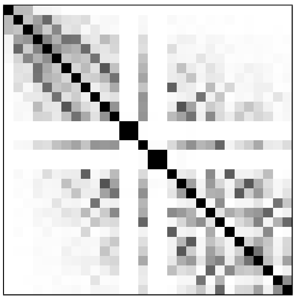

Fig. 5.3. *Weighted correlations for the leading 30 components*

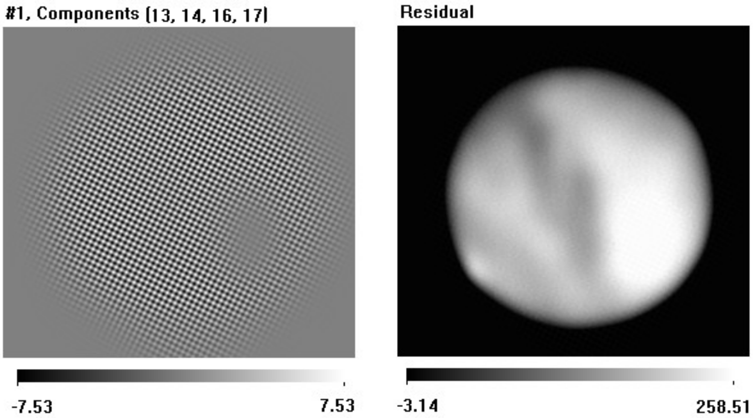

Fig. 5.4. *Reconstructed noise and residual (filtered image)*

REFERENCES

[1] G. Heinig and K. Rost *Algebraic methods for Toeplitz-like matrices and operators*, Akademie Verlag, Berlin, 1984.

[2] C.F. Van Loan and N.P. Pitsianis *Approximation with Kronecker products* in M.S.Moonen and G. H. Golub, eds., *Linear Algebra for Large Scale and Real Time Applications*, Kluwer Publications, pp. 293–314, 1993.

[3] V.M. Buchstaber *Time series analysis and grassmanians* in S. Gindikin, ed., *Applied Problems of Radon Transform*, AMS Transaction – Series 2, Vol. 162, Providence (RI), pp. 1–17, 1994.

[4] J. Elsner and A. Tsonis *Singular Spectrum Analysis. A New Tool in Time Series Analysis*, Plenum Press, New York, 1996.

[5] H. Hua Yang and Y. Hua *On Rank of Block Hankel Matrix for 2-D Frequency Detection and Estimation*, IEEE Transactions on Signal Processing, Vol. 44, Issue 4, pp. 1046–1048 1996.

[6] D. Danilov and A. Zhigljavsky, eds., *Principal Components of Time Series: the "Caterpillar" method*, St.Petersburg State University, St.Petersburg, 1997 (in Russian).

[7] N. Golyandina, V. Nekrutkin, and A. Zhigljavsky *Analysis of Time Series Structure: SSA and Related Techniques*, Chapman & Hall/CRC, Boca Raton, 2001.

[8] J.R. Magnus and H. Neudecker *Matrix Differential Calculus with Applications to Statistics and Econometrics*, John Wiley & Sons, 2004.

[9] A.J. Laub *Matrix Analysis for Scientists and Engineers*, SIAM, 2004.

[10] D. Stepanov and N.Golyandina *SSA-based approaches to analysis and forecast of multidimensional time series*, Proceedings of the 5th St.Petersburg Workshop on Simulation, St.Petersburg State University, St.Petersburg, pp. 293–298, 2005.