SIAM J. MATRIX ANAL. APPL.
Vol. 42, No. 2, pp. 883–912

SYSTEMS OF POLYNOMIAL EQUATIONS, HIGHER-ORDER TENSOR DECOMPOSITIONS, AND MULTIDIMENSIONAL HARMONIC RETRIEVAL: A UNIFYING FRAMEWORK. PART I: THE CANONICAL POLYADIC DECOMPOSITION*

JEROEN VANDERSTUKKEN† AND LIEVEN DE LATHAUWER†

Abstract. We propose a multilinear algebra framework to solve systems of polynomial equations with simple roots. We translate connections between univariate polynomial root-finding, eigenvalue decompositions, and harmonic retrieval to their higher-order counterparts: a canonical polyadic decomposition (CPD) that exploits shift invariance structures in the null space of the Macaulay matrix reveals the roots of the polynomial system. The new framework allows us to use numerical CPD algorithms for solving systems of polynomial equations. For the same degree of the Macaulay matrix as in numerical polynomial algebra/polynomial numerical linear algebra, the CPD is interpreted as the joint eigenvalue decomposition of the multiplication tables. In our approach the degree can also be lower. Affine roots and roots at infinity can be handled in the same way. With minor modifications, the technique can be used to estimate approximate roots of overconstrained systems.

Key words. system of polynomial equations, multilinear algebra, canonical polyadic decomposition, harmonic retrieval, Macaulay matrix, Vandermonde matrix

AMS subject classifications. 13P15, 15A69, 65H04

DOI. 10.1137/17M1150050

1. Introduction.

Systems of polynomial equations arise often in science and engineering (chemistry, mechanics, optimization, etc.). Solving such a system means finding all the common roots of the polynomials. Formally, the roots of a system of s polynomial equations in n complex variables $x_j \in \mathbb{C}$

$$\left\{ \begin{array}{c} f_1(x_1, \dots, x_n) = 0, \\ \vdots \\ f_s(x_1, \dots, x_n) = 0 \end{array} \right. \tag{1}$$

are all points $\mathbf{x} \in \mathbb{C}^n$ that satisfy (1). The problem has been studied extensively in algebraic geometry. Most of the algebraic geometry-based methods compute a Gröbner basis for the system, the common roots of which are easier to obtain. One seminal method to compute a Gröbner basis is due to Buchberger [3]. However, the implied symbolic manipulations are subject to numerical instabilities, and they are not very meaningful when the polynomial coefficients are derived from measured data [13, 14]. Arguably the most popular numerical method to solve a system of polynomial equations is numerical polynomial homotopy continuation (PHC) [37]. Continuation

*Received by the editors October 2, 2017; accepted for publication (in revised form) March 26, 2021; published electronically June 17, 2021.
https://doi.org/10.1137/17M1150050

Funding: The work of the authors was supported by the Research Council KU Leuven: C1 project c16/15/059-nD, the FWO under EOS project G0F6718N (SeLMA), the Flemish Government under the "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen," and the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC Advanced grant BIOTENSORS 339804). This paper reflects only the authors' views, and the Union is not liable for any use that may be made of the contained information.

†Group Science, Engineering and Technology, KU Leuven Kulak, E. Sabbelaan 53, 8500 Kortrijk, Belgium and Department of Electrical Engineering ESAT/STADIUS, KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium (vanderstukken.jeroen@gmail.com, lieven.delathauwer@kuleuven.be).

retrieves the roots of an easy, parametrized system that can be continuously transformed into the more difficult given system. Among the first to look at (1) from a linear algebra point of view were Sylvester (1853) and Macaulay (1902). Their work introduced a resultant (matrix)—itself a polynomial (matrix) that generalizes the characteristic polynomial in the univariate case. In his numerical polynomial algebra (NPA), Stetter (2004) linked the problem to eigenvalue computations of so-called "multiplication tables" and brought it to the field of numerical linear algebra [30]. Batselier and Dreesen (2013) developed polynomial numerical linear algebra (PNLA): applying a reasoning known as "estimation of signal parameters by rotational invariance techniques" (ESPRIT) in array processing to the multivariate monomials in the null space of the system's Macaulay (resultant) matrix yields an eigenvalue decomposition (EVD) that reveals the roots of the system [1, 12, 25].

A higher-order tensor is a multiway array indexed by three or more indices.[1] As such, a tensor naturally generalizes the concept of a one-way vector, which is indexed by one index, and a two-way matrix, which is indexed by two indices. Tensor decompositions like the canonical polyadic decomposition (CPD) [19] are then generalizations of matrix decompositions. Whereas the matrix singular value decomposition (SVD) is only unique due to the imposed orthogonality constraints, the CPD is unique under much milder conditions, making it a crucial tool for data analysis [4, 26].

An isomorphism between polynomials and higher-order tensors has been long known in algebraic geometry. Yet, this paper translates the well-known connections between univariate polynomial root-finding, linear algebra, and harmonic retrieval (HR) to their higher-order counterparts: systems of multivariate polynomial equations, multilinear algebra, and multidimensional harmonic retrieval (MHR). As does PNLA, we exploit the structure of the null space of a system's Macaulay matrix—to then build a third-order tensor of which the CPD reveals the roots of the system. Moreover, we explain that this CPD may be seen as the *joint* EVD of NPA's multiplication tables—opposed to only *one* EVD in PNLA. In our framework there is no need to handle affine and projective roots in a different manner. Numerical experiments confirm that the precision of our framework is as good as the precision of PHC. The roots may be found from a Macaulay matrix of lower degree. The framework also allows us to find the approximate roots of overconstrained systems.

The paper is organized as follows. Section 2 will review our notation and introduce some elementary definitions. In sections 3–4 we will derive a connection between the null space of the Macaulay matrix of a generic system of polynomial equations, i.e., a system that has only (i) simple and (ii) affine roots, the MHR problem and CPD. The material will be discussed in a discipline-specific manner in section 3 and combined in section 4. At the end of section 4 we will have expressed the problem as a so-called coupled CPD. This is the polynomial equations counterpart of a recently developed technique for MHR [28, 29]. In section 5 we will go further and reduce the polynomial problem to a single CPD. In subsection 5.1 we focus on the case of affine roots only, and in subsection 5.2 we will generalize to the projective case; i.e., in subsection 5.2 we will drop constraint (ii) above. In section 6 we will make the connection with the generalized eigenvalue decomposition (GEVD) of a matrix pencil and with NPA/PNLA. In section 7 we will extend our approach to Macaulay matrices of degree one less than the degree required in PNLA. Section 8 will present the overall multilinear algebra-based algorithm to find the roots of a system of polynomial

[1] An Nth-order tensor can be thought of as the outer product of N vector spaces. Mathematicians tend to prefer this coordinate-free definition [23].

equations that has only (i) simple roots. The companion paper [36] will drop constraint (i) as well and will relate the topics to a third-order tensor block-term decomposition. Section 9 will present the results of two numerical experiments. Section 10 will summarize our findings.

2. Notation.

2.1. Higher-order tensors. To infer the type of a quantity from its notation, scalars, vectors, matrices, and tensors are denoted by italic, boldface lowercase, boldface uppercase, and calligraphic letters, respectively: $a \in \mathbb{C}$, $\mathbf{a} \in \mathbb{C}^{I_1}$, $\mathbf{A} \in \mathbb{C}^{I_1 \times I_2}$, and the Nth-order tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$. In this paper we will mainly work with third-order tensors ($N = 3$). We will consistently write $a_{i_1} = \mathbf{a}(i_1) = (\mathbf{a})_{i_1}$ for the i_1th (scalar) entry of the vector $\mathbf{a}$ and $a_{i_1,i_2} = \mathbf{A}(i_1, i_2) = (\mathbf{A})_{i_1,i_2}$ for the entry of the matrix $\mathbf{A}$ with row index i_1 and column index i_2. Using MATLAB colon notation, $\mathbf{a}_{i_2} = \mathbf{A}(:, i_2) = (\mathbf{A})_{i_2}$ denotes the i_2th column of $\mathbf{A}$. Likewise for tensor entries and for fibers: a mode-n fiber of a tensor $\mathcal{A}$ is a vector obtained when all but the nth index of $\mathcal{A}$ are kept fixed. Mode-1 and mode-2 fibers correspond to column and row vectors, respectively. We denote the i_3th matrix slice of $\mathcal{A}$ as $\mathbf{A}_{i_3} = \mathcal{A}(:, :, i_3)$. We use $\cdot^*$, $\cdot^T$, $\cdot^H$, $\cdot^{-1}$, and $\cdot^\dagger$ for the complex conjugate, transpose, Hermitian transpose, inverse, and Moore–Penrose pseudoinverse, respectively.

$\mathbf{D} = \operatorname{diag}(\mathbf{d})$ represents a diagonal matrix with the vector $\mathbf{d}$ on its diagonal, and $\mathbf{D}_i(\mathbf{C}) = \operatorname{diag}(\mathbf{C}(i, :))$ holds the ith row of the matrix $\mathbf{C}$. $\mathbf{I}_I$ is the identity matrix of order $I \times I$. $\operatorname{span}(\{\mathbf{a}_1, \ldots, \mathbf{a}_I\})$ is the span of the vectors $\mathbf{a}_1$ through $\mathbf{a}_I$. $\operatorname{col}(\mathbf{A})$, $\operatorname{row}(\mathbf{A})$, and $\operatorname{null}(\mathbf{A})$ are used to denote the column, row, and right null space of $\mathbf{A}$, respectively. The dimension of a vector space is denoted by $\dim \cdot$. The rank of matrix $\mathbf{A}$ is denoted by $r_{\mathbf{A}} = \dim \operatorname{col}(\mathbf{A}) = \dim \operatorname{row}(\mathbf{A})$, while $k_{\mathbf{A}}$ is its Kruskal rank, i.e., the largest number k such that any subset of k columns of $\mathbf{A}$ is linearly independent. The Kronecker product of $\mathbf{A} \in \mathbb{C}^{I_1 \times J_1}$ and $\mathbf{B} \in \mathbb{C}^{I_2 \times J_2}$ is given by

$$\mathbf{A} \otimes \mathbf{B} \stackrel{\text{def}}{=} \begin{pmatrix} a_{1,1}\mathbf{B} & \cdots & a_{1,J_1}\mathbf{B} \\ \vdots & & \vdots \\ a_{I_1,1}\mathbf{B} & \cdots & a_{I_1,J_1}\mathbf{B} \end{pmatrix} \in \mathbb{C}^{I_1 I_2 \times J_1 J_2}.$$

The Khatri–Rao or columnwise Kronecker product of $\mathbf{A} \in \mathbb{C}^{I_1 \times R}$ and $\mathbf{B} \in \mathbb{C}^{I_2 \times R}$ is given by $\mathbf{A} \odot \mathbf{B} \stackrel{\text{def}}{=} (\mathbf{a}_1 \otimes \mathbf{b}_1 \cdots \mathbf{a}_R \otimes \mathbf{b}_R) \in \mathbb{C}^{I_1 I_2 \times R}$.

A third-order tensor $\mathcal{A}$ is vectorized into $\operatorname{vec}(\mathcal{A}) = \mathbf{a}_{[3,2,1]}$ by vertically stacking all entries such that i_3 varies slowest and i_1 varies fastest. In other words, the tensor entry a_{i_1,i_2,i_3} corresponds to the entry of $\operatorname{vec}(\mathcal{A})$ with index $(i_3 - 1)I_2 I_1 + (i_2 - 1)I_1 + i_1$. The mode-1 matrix representation denoted by $\mathbf{A}_{[1;3,2]}$ is obtained by horizontally stacking the columns of $\mathcal{A}$ in such a way that i_2 varies fastest along the second dimension. In other words, a_{i_1,i_2,i_3} corresponds to the entry of $\mathbf{A}_{[1;3,2]}$ with row index i_1 and column index $(i_3 - 1)I_2 + i_2$. Similarly, a_{i_1,i_2,i_3} corresponds to the entry of $\mathbf{A}_{[1,2;3]}$ with row index $(i_1 - 1)I_2 + i_2$ and column index i_3. Other mode-n matrix representations are defined analogously. The mode-1 product $\mathcal{C} = \mathcal{A} \cdot_1 \mathbf{B} \in \mathbb{C}^{J \times I_2 \times I_3}$ of a tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$, and a matrix $\mathbf{B} \in \mathbb{C}^{J \times I_1}$ has the matrix representation $\mathbf{C}_{[1;3,2]} = \mathbf{B} \cdot \mathbf{A}_{[1;3,2]}$, i.e., it is the result of multiplying all columns of $\mathcal{A}$ with $\mathbf{B}$. Likewise, the mode-2 product $\tilde{\mathcal{C}} = \mathcal{A} \cdot_2 \tilde{\mathbf{B}} \in \mathbb{C}^{I_1 \times \tilde{J} \times I_3}$ is obtained by multiplying all rows of $\mathcal{A}$ with $\tilde{\mathbf{B}} \in \mathbb{C}^{\tilde{J} \times I_2}$. The mode-$n$ rank $R_n = \operatorname{rank}_n(\mathcal{A})$ is the dimension of the mode-n fiber space, i.e., $R_n = r_{\mathbf{A}_{[n;\bullet]}}$, in which $\bullet$ indicates that the order of the indices different from n does not matter. In particular, R_1 and R_2 are known as the column rank and row rank of $\mathcal{A}$, respectively. The tuple $\text{rank}_{\boxplus}(\mathcal{A}) = (R_1, R_2, R_3)$ is called the multilinear rank of $\mathcal{A}$.

The outer product $\mathcal{T} = \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c}$ with nonzero $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ yields a rank-1 tensor with entries $t_{i_1,i_2,i_3} = a_{i_1} b_{i_2} c_{i_3}$. In matrix format we can write $\mathbf{T}_{[1,2;3]} = (\mathbf{a} \otimes \mathbf{b})\mathbf{c}^T$. Note that the larger symbol $\otimes$ denotes the Kronecker product, whereas the smaller symbol $\otimes$ denotes the outer product. We further define the inner product of two tensors as $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ as $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1,i_2,i_3} a_{i_1,i_2,i_3} b^*_{i_1,i_2,i_3}$ and the induced Frobenius norm as $\|\mathcal{A}\| = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$.

2.2. Polynomial equations. In the system of polynomial equations (1), the basic building blocks are monomials $\mathbf{x}^{\boldsymbol{\alpha}} = \prod_{j=1}^{n} x_j^{\alpha_j}$ with exponent vector $\boldsymbol{\alpha}$ and polynomials $f(x_1, \ldots, x_n) = \sum_{l=1}^{p} f_l \mathbf{x}_l^{\boldsymbol{\alpha}_l}$ with coefficient vector $\mathbf{f}$. The degree of a monomial is defined as $\deg(\mathbf{x}^{\boldsymbol{\alpha}}) = \sum_{j=1}^{n} \alpha_j$. There exist several schemes for ordering monomials by their exponent vector. In this paper, we will adopt the degree negative lexicographic order. The monomials $\mathbf{x}^{\boldsymbol{\alpha}} < \mathbf{x}^{\boldsymbol{\beta}}$ if one of the following two conditions is satisfied: (i) $\deg(\mathbf{x}^{\boldsymbol{\alpha}}) < \deg(\mathbf{x}^{\boldsymbol{\beta}})$ or (ii) $\deg(\mathbf{x}^{\boldsymbol{\alpha}}) = \deg(\mathbf{x}^{\boldsymbol{\beta}})$ and the leftmost nonzero entry of $\boldsymbol{\beta} - \boldsymbol{\alpha}$ is negative.

Example 2.1. Consider monomials in two variables. We have that (i) $x_2 < x_1^2$ because $\deg(x_2) = 1 < 2 = \deg(x_1^2)$ and (ii) $x_1^2 < x_1 x_2$ because $\deg(x_1^2) = \deg(x_1 x_2) = 2$ and $\boldsymbol{\beta} - \boldsymbol{\alpha} = \begin{pmatrix} -1 & 1 \end{pmatrix}^T$, the first entry of which is negative.

Each polynomial f_i has a degree d_i equal to the degree of the monomial with the highest degree in f_i. The ring of all polynomials in n variables is denoted by $\mathcal{C}^n$. The vector space $\mathcal{C}_d^n$ is the subset of the ring $\mathcal{C}^n$ that contains all polynomials up to degree d. Its dimension is given by

$$q(d) \overset{\text{def}}{=} \dim \mathcal{C}_d^n = \binom{n+d}{n}.$$

A polynomial is homogeneous if all its monomials have equal degree. One can homogenize a polynomial f of degree d to f^h by multiplying each monomial $\mathbf{x}_l^{\boldsymbol{\alpha}_l}$ in f with a power β_l of the variable x_0 such that $\deg(x_0^{\beta_l} \mathbf{x}_l^{\boldsymbol{\alpha}_l}) = d$ for all l. The ring (vector space) of all homogeneous polynomials in $n+1$ variables (up to degree d) is then denoted by $\mathcal{P}^n$ ($\mathcal{P}_d^n$). Having introduced the variable x_0, the projective space $\mathbb{P}^n$ arises as the set of equivalence classes on $\mathbb{C}^{n+1} \setminus \{\mathbf{0}\}$: we have that $(x'_0 \quad x'_1 \quad \ldots \quad x'_n)^T \sim (x_0 \quad x_1 \quad \ldots \quad x_n)^T$ if there exists a $\lambda \in \mathbb{C}$ such that $(x'_0 \quad x'_1 \quad \ldots \quad x'_n)^T = \lambda(x_0 \quad x_1 \quad \ldots \quad x_n)^T$. Points with $x_0 = 0$ cannot be normalized to their affine counterpart $(1 \quad \frac{x_1}{x_0} \quad \ldots \quad \frac{x_n}{x_0})^T$: they are points at infinity.

The degree of the system (1) is $d_0 = \max_{i=1}^{s} d_i$. The set of all roots of (1) is called the solution set. For square ($n = s$) systems with individual degrees d_i, $i = 1 : n$, and under the important assumption that the solution set is 0-dimensional, meaning that all roots are isolated and that their number is finite,[2] the number of roots in the projective space, counting multiplicities, is given by the Bézout number

$$m \overset{\text{def}}{=} \prod_{i=1}^{n} d_i.$$

[2]The solution set is called a variety in algebraic geometry. Its dimension equals the degree of the Hilbert polynomial. As long as the greatest common divisor of the multivariate polynomials f_i is a constant, the solution set is 0-dimensional.

The m roots of (1) will be represented by $(x_1^{(k)} \quad x_2^{(k)} \quad \cdots \quad x_n^{(k)})^T \in \mathbb{C}^n$, $k = 1 : m$. If there are roots of multiplicity greater than 1, $m_0 < m$ denotes the number of disjoint roots.[3]

2.3. Vandermonde matrices. A (univariate) Vandermonde matrix is of the following form:

$$\begin{aligned}\mathbf{V}^{(1)}(\{z_r\}_{r=1}^R) &\overset{\text{def}}{=} \begin{pmatrix}\mathbf{v}_1^{(1)} & \dots & \mathbf{v}_R^{(1)}\end{pmatrix} \in \mathbb{C}^{I \times R}, \\ \mathbf{v}_r^{(1)} &\overset{\text{def}}{=} \begin{pmatrix}1 & z_r & z_r^2 & \dots & z_r^{I-1}\end{pmatrix}^T, \quad r = 1 : R.\end{aligned} \tag{2}$$

The scalars $z_r \in \mathbb{C}$ are sometimes called the generators of $\mathbf{V}^{(1)}$. The $((d+1) \times m)$ univariate Vandermonde matrix generated by the jth coordinate of the m roots of (1), i.e., by $\{x_j^{(k)}\}_{k=1}^m$, will specifically be denoted as $\mathbf{V}^{(j)}(d)$, $j = 1 : n$.

A *multivariate* Vandermonde matrix is of the following form:

$$\begin{aligned}&\mathbf{V}(\{z_{j,r}, 1 \le j \le n, 1 \le r \le R\}) \overset{\text{def}}{=} \begin{pmatrix}\mathbf{v}_1 & \dots & \mathbf{v}_R\end{pmatrix} \in \mathbb{C}^{q(d) \times R}, \\ &\mathbf{v}_r \overset{\text{def}}{=} \begin{pmatrix}1 & z_{1,r} & z_{2,r} & \dots & z_{1,r}^2 & z_{1,r}z_{2,r} & \dots & z_{n-1,r}z_{n,r}^{d-1} & z_{n,r}^d\end{pmatrix}^T, \quad r = 1 : R.\end{aligned} \tag{3}$$

The entries of multivariate Vandermonde vectors are ordered by the degree negative lexicographic order. The $(q(d) \times m)$ multivariate Vandermonde matrix generated by the coordinates of the m roots of (1), i.e., by $\{x_j^{(k)}, 1 \le j \le n, 1 \le k \le m\}$, will specifically be denoted as $\mathbf{V}(d)$.

3. CPD, PNLA, and MHR. In this paper we combine insights from three disciplines: tensor methods, PNLA, and MHR. This section puts the ingredients that we will need on the table, presented in a way that will facilitate their combination.

3.1. Tensor CPD and matrix GEVD. An R-term polyadic decomposition (PD) expresses a tensor $\mathcal{T} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ as a sum of R rank-1 terms

$$\mathcal{T} = \sum_{r=1}^R \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r \overset{\text{def}}{=} [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!], \tag{4}$$

where $\mathbf{A} \in \mathbb{C}^{I_1 \times R}$, $\mathbf{B} \in \mathbb{C}^{I_2 \times R}$, and $\mathbf{C} \in \mathbb{C}^{I_3 \times R}$ are called factor matrices. If R is minimal, then the PD is called a CPD. The minimal number of rank-1 terms is called *the* rank of $\mathcal{T}$ and denoted as $r_{\mathcal{T}}$. The decomposition is visualized in Figure 1. In terms of matrix slices, (4) can be written as

$$\mathbf{T}_{i_3} = \mathbf{A} \cdot \mathbf{D}_{i_3}(\mathbf{C}) \cdot \mathbf{B}^T, \quad i_3 = 1 : I_3. \tag{5}$$

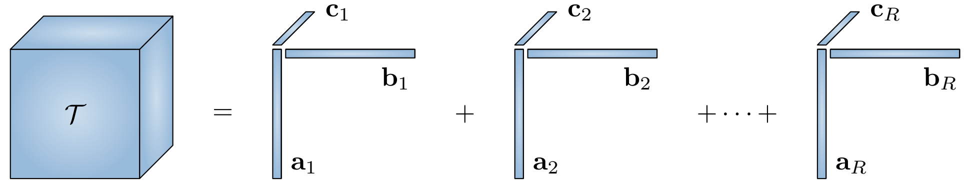


FIG. 1. *(C)PD of a third-order tensor is a decomposition in a (minimal) number of rank-1 terms.*

[3]This case is handled in the companion paper [36].

Working with matrix representations, (4) can also be written as

$$\mathbf{T}_{[1,2;3]} = \sum_{r=1}^{R} (\mathbf{a}_r \otimes \mathbf{b}_r)\mathbf{c}_r^T = (\mathbf{A} \odot \mathbf{B})\,\mathbf{C}^T. \tag{6}$$

Obviously, the rank-1 terms in a CPD can be arbitrarily permuted, and the corresponding columns of the different factor matrices can be scaled/counterscaled. Formally, the CPD of a rank-R tensor $\mathcal{T} \in \mathbb{C}^{I_1\times I_2\times I_3}$ is said to be essentially *unique* iff $\mathcal{T} = [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!] = [[\tilde{\mathbf{A}},\tilde{\mathbf{B}},\tilde{\mathbf{C}}]]$ implies that there exist a permutation matrix $\mathbf{\Pi} \in \mathbb{C}^{R\times R}$ and nonsingular diagonal matrices $\mathbf{\Lambda_A} \in \mathbb{C}^{R\times R}$, $\mathbf{\Lambda_B} \in \mathbb{C}^{R\times R}$, and $\mathbf{\Lambda_C} \in \mathbb{C}^{R\times R}$ such that

$$\tilde{\mathbf{A}} = \mathbf{A}\mathbf{\Pi}\mathbf{\Lambda_A}, \quad \tilde{\mathbf{B}} = \mathbf{B}\mathbf{\Pi}\mathbf{\Lambda_B}, \quad \tilde{\mathbf{C}} = \mathbf{C}\mathbf{\Pi}\mathbf{\Lambda_C}, \quad \text{and} \quad \mathbf{\Lambda_A}\mathbf{\Lambda_B}\mathbf{\Lambda_C} = \mathbf{I}_R.$$

For brevity, we will drop the term "essential" from now on. The following theorem presents a first sufficient uniqueness condition.

THEOREM 3.1 (see [24]). *Let $\mathcal{T} \in \mathbb{C}^{I_1\times I_2\times I_3}$ admit a PD $\mathcal{T} = [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]$ where $\mathbf{A} \in \mathbb{C}^{I_1\times R}$ and $\mathbf{B} \in \mathbb{C}^{I_2\times R}$ have full column rank; then*

$$r_{\mathcal{T}} = R,^{4} \text{ and the CPD of } \mathcal{T} \text{ is unique} \qquad \Leftrightarrow \qquad k_{\mathbf{C}} \geq 2.$$

Under the conditions in Theorem 3.1, the CPD is not only unique; it can directly be obtained from a matrix GEVD. To explain this, let us consider two matrices $\tilde{\mathbf{T}}_1, \tilde{\mathbf{T}}_2 \in \mathbb{C}^{I_1\times I_2}$, with $I_1 \geq I_2$ (without loss of generality), structured as $\tilde{\mathbf{T}}_1 = \mathbf{A}\mathbf{D}_1\mathbf{B}^T$ and $\tilde{\mathbf{T}}_2 = \mathbf{A}\mathbf{D}_2\mathbf{B}^T$. Here we assume that $\mathbf{A} \in \mathbb{C}^{I_1\times R}$ and $\mathbf{B} \in \mathbb{C}^{I_2\times R}$ have full column rank, that $\mathbf{D}_1, \mathbf{D}_2 \in \mathbb{C}^{R\times R}$ are diagonal, and that there are no collinear vectors in the set $\{((\mathbf{D}_1)_{r,r}, (\mathbf{D}_2)_{r,r})^T\}_{r=1}^{R}$. Clearly, the columns of $\mathbf{B}^{\dagger,T}$ are generalized eigenvectors of the pencil $(\tilde{\mathbf{T}}_1, \tilde{\mathbf{T}}_2)$ and the GEVD is unique since all the generalized eigenvalues are distinct. Condition $k_{\mathbf{C}} \geq 2$ in the theorem means that no two columns of $\mathbf{C}$ are collinear. This implies that it is possible to take $\tilde{\mathbf{T}}_1$, $\tilde{\mathbf{T}}_2$ equal to two of the tensor slices $\mathbf{T}_{i_3}$ or to suitable linear combinations of the slices if this is needed to ensure that all the generalized eigenvalues are distinct. Note that CPD may be seen as an extension of GEVD to more than two matrices.

One could say that, under the conditions in Theorem 3.1, the computation of a CPD is a task of linear algebra. However, this is a matter of perspective. Although the CPD has *algebraically* been reduced to a matrix GEVD, there are *numerical* differences. Formally, collapsing the structure of the full tensor into the structure of a matrix pencil may increase the condition number [2]. Moreover, in many applications the tensor $\mathcal{T}$ is only known with limited precision (e.g., it consists of noisy measurements) and the CPD structure does not hold exactly. In such cases, the factor matrices are most often estimated by a numerical optimization routine that fits the CPD model to the given tensor [27, 39, 26], and this is clearly a multilinear problem. In practice, one often initializes the optimization algorithm with estimates obtained by GEVD. In other words, the problem of linear algebra is solved to obtain a first estimate of the solution of the multilinear problem.

It may come as a surprise that the CPD of $\mathcal{T}$ can be obtained from a matrix GEVD, while CPD is known to be an NP-hard problem [18]. Again this is a matter of perspective. The qualification "NP-hard" concerns "CPD in general." However, in Theorem 3.1 we consider a specific class of CPDs, namely, the class for which

[4] In other words, R is the minimal number of rank-1 terms and the PD is canonical.

$r_{\mathbf{A}} = r_{\mathbf{B}} = R$ and $k_{\mathbf{C}} \geq 2$. Under these conditions it *is indeed* possible to obtain the factors from a GEVD. At least, there is an algebraic guarantee for CPDs that are exact. However, as mentioned above, there are numerical aspects and also data quality aspects. For instance, the CPD structure that we will discuss in subsection 3.3 is, under certain application-specific assumptions, known to hold exactly for a range of array processing problems in the absence of noise. In practice, data are noisy and the CPD model describes what happens with the "true" underlying signals. One assumes that the signal-to-noise ratio (SNR) is high enough to allow the factor matrices to be estimated with reasonable accuracy. Simulations may give an idea of the SNR that is required. The numerical experiment in subsection 9.2 will be an example of this approach.

Theorem 3.1 assumes that two factor matrices, $\mathbf{A}$ and $\mathbf{B}$, have full column rank. The next theorem relaxes this to a full column rank assumption on a single factor matrix; for notational convenience we take $\mathbf{C}$ for the latter. The theorem is formulated in terms of compound matrices. For $\mathbf{A} \in \mathbb{C}^{I \times R}$, the second compound matrix $\mathbf{M}_2(\mathbf{A}) \in \mathbb{C}^{\binom{I}{2} \times \binom{R}{2}}$ is the matrix that contains all (2×2) minors, ordered lexicographically [10, section 2].

THEOREM 3.2 (see [7, 20]). *Let $\mathcal{T} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ admit a PD $\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ where $\mathbf{C} \in \mathbb{C}^{I_3 \times R}$ has full column rank. If $\mathbf{M}_2(\mathbf{A}) \odot \mathbf{M}_2(\mathbf{B}) \in \mathbb{R}^{\binom{I}{2}\binom{J}{2} \times \binom{R}{2}}$ has full column rank, then $r_{\mathcal{T}} = R$ and the CPD of $\mathcal{T}$ is unique.*

Like Theorem 3.1, Theorem 3.2 admits a constructive interpretation [7]. Let $\mathbf{T}_{[1,2;3]} = \mathbf{E} \cdot \mathbf{F}^T$ denote a rank-revealing decomposition of $\mathbf{T}_{[1,2;3]}$. Comparing with (6), we want to find a nonsingular matrix $\mathbf{G} \in \mathbb{C}^{R \times R}$ such that $\mathbf{EG}$ takes the form of a Khatri–Rao product. If the matrix $\mathbf{G}$ is unique (up to trivial indeterminacies), then $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ follow immediately from the connection with (6). It turns out that, under the conditions in Theorem 3.2, an auxiliary tensor $\mathcal{U} \in \mathbb{C}^{R \times R \times R}$ can be derived from $\mathcal{T}$, with CPD given by $\mathcal{U} = [\![\mathbf{G}^{-1}, \mathbf{G}^{-1}, \mathbf{F}]\!]$, in which $\mathbf{F} \in \mathbb{C}^{R \times R}$ is also nonsingular. As the auxiliary CPD satisfies the conditions of Theorem 3.1, the desired $\mathbf{G}$ can be obtained from a GEVD. The auxiliary tensor $\mathcal{U}$ itself can be obtained from an overdetermined set of linear equations.

Summarizing, also under the conditions in Theorem 3.2, the computation of an exact CPD can be reduced to a matrix GEVD. If the tensor $\mathcal{T}$ is only known with limited precision, then we may proceed as follows. The GEVD derived from the auxiliary tensor $\mathcal{U}$ may be used to initialize a numerical optimization algorithm that fits a CPD model to $\mathcal{U}$. The resulting estimate of $\mathbf{G}$ yields first estimates of the factor matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ of the original tensor $\mathcal{T}$. The latter may in turn be used to initialize a numerical optimization algorithm that fits a CPD model to $\mathcal{T}$.

The conditions in Theorem 3.2 can be relaxed further; see [26, section IV] for a short tutorial on CPD uniqueness results.

3.2. The Macaulay matrix. To fully comprehend the construction of the Macaulay matrix, polynomial ideals and their quotient rings need to be introduced first. A polynomial is defined as a linear combination of p monomials. An extension is given by a polynomial combination $g = \sum_{i=1}^{s} c_i f_i$ where both f_i and c_i are polynomials in $\mathcal{C}^n, i = 1 : s$ [5]. The subset of the ring $\mathcal{C}^n$ that is reached by polynomial combinations of the elements of $\mathcal{F} = \{f_i\}_{i=1}^{s}$ is an ideal: it is closed under polynomial combination. On the other hand, given a fixed set of m points $\mathcal{Z} = \{\mathbf{z}_k\}_{k=1}^{m} \subset \mathbb{C}^n$, the subset $\mathcal{I} \subset \mathcal{C}^n$ of polynomials that attain zero in $\mathcal{Z}$ is also an ideal. Indeed, every polynomial combination of the polynomials in $\mathcal{I}$ is again zero in $\mathcal{Z}$. If $\mathcal{F}$ is now a (nonunique) basis for $\mathcal{I}$,[5] we write $\mathcal{I} = \langle \mathcal{F} \rangle$ and we know that $\mathcal{Z}$ is nothing but the solution set of the system defined by the basis $\mathcal{F}$.

If g is a polynomial such that $\mathbf{z} \in \mathcal{Z} : g(\mathbf{z}) = a \neq 0$, then $g \in \mathcal{I}$ is impossible. Instead, we can write $g = \sum_{i=1}^{s} c_i f_i + r$ with $r(\mathbf{z}) = a$ or, more generally, $g(\mathbf{z}_k) = r(\mathbf{z}_k)$ for all k. We say that $g \sim r \Leftrightarrow g - r \in \mathcal{I}$ and that the residue class of $g \mod \mathcal{I}$ is the set $[g] = \{r \in \mathcal{C}^n | \, g \sim r\}$ [32]. In particular, $[0] = \mathcal{I}$. If $g \in \mathcal{I}$, it follows that $g(\mathbf{z}_k) = r(\mathbf{z}_k) = 0$ for all k. One can show that, if all roots in $\mathcal{Z}$ defined by the elements of $\mathcal{F}$ are simple, then the converse is true, i.e., $g(\mathbf{z}_k) = 0$ for all k is sufficient for $g \in \mathcal{I}$. The set of all residue classes $[r]$ is a quotient ring $\mathcal{C}^n/\mathcal{I}$ of the polynomial ideal $\mathcal{I}$. From the above reasoning, any residue class is completely characterized by the values its members take on $\mathcal{Z}$ and $\dim \mathcal{C}^n/\mathcal{I} = m$.

Definition 3.3 defines the aforementioned Macaulay matrix. The definition is most easily understood by means of Example 3.4. For a given system (1) and a chosen degree $d \geq d_0$, the Macaulay matrix $\mathbf{M}(d)$ is a matrix constructed from the polynomial coefficients in such a way that its row space $\mathcal{M}_d$ is the set of polynomial combinations

$$\mathcal{M}_d = \left\{ \sum_{i=1}^{s} c_i f_i | \, c_i \in \mathcal{C}^n_{d-d_i} \right\}.$$

Definition 3.3 (see [14, page 263]). *Let $f_i \in \mathcal{C}^n_{d_i}$, $i = 1 : s$, be s polynomials of degree d_i in n variables $x_1, \ldots, x_n$; then the Macaulay matrix $\mathbf{M}(d)$ of degree d contains as its rows the coefficients of*

$$\mathbf{M}(d) = \begin{pmatrix} f_1 \\ x_1 f_1 \\ \vdots \\ x_n^{d-d_1} f_1 \\ f_2 \\ x_1 f_2 \\ \vdots \\ x_n^{d-d_s} f_s \end{pmatrix} \in \mathbb{C}^{p \times q(d)}$$

where each polynomial $f_i, i = 1 : s$, is multiplied with all possible monomials $\mathbf{x}^{\boldsymbol{\alpha}}$, $\deg(\boldsymbol{\alpha}) = 0 : d - d_i \in \mathbb{N}$*—eventually determining the number of rows p.*

Example 3.4 (see [12, page 17]). Consider the system of $s = 2$ polynomial equations in $n = 2$ variables x_1 and x_2

$$\begin{cases} f_1(x_1, x_2) = -x_1^2 + 2x_1x_2 + x_2^2 + 5x_1 - 3x_2 - 4 = 0, \\ f_2(x_1, x_2) = x_1^2 + 2x_1x_2 + x_2^2 - 1 = 0 \end{cases}$$

where $d_1 = d_2 = 2$. The system has $m = \prod_{i=1}^{n} d_i = 2 \cdot 2 = 4$ solutions $(x_1^{(k)} \quad x_2^{(k)})^T$, $k = 1 : 4$, namely, $(0 \quad -1)^T$, $(1 \quad 0)^T$, $(3 \quad -2)^T$, and $(4 \quad -5)^T$.

We start constructing the Macaulay matrix at $d = 2 \geq 2 = d_0$. The rows of $\mathbf{M}(2)$ are shifted versions of the polynomial coefficient vectors that are the result of multiplying each f_i with each $x_j^{2-2} = x_j^0 = 1$, $j = 1 : 2$. Simply stated, $\mathbf{M}(2)$ does not involve any shifts:

[5] One such basis is the Gröbner basis for the ideal.

$$\mathbf{M}(2) = \begin{array}{c} \\ f_1(x_1,x_2) \\ f_2(x_1,x_2) \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \end{array} \\ \left(\begin{array}{c|cc|ccc} -4 & 5 & -3 & -1 & 2 & 1 \\ -1 & 0 & 0 & 1 & 2 & 1 \end{array}\right) \end{array}.$$

Note that we have adopted the degree negative lexicographic order for the monomials in the columns.

It should be clear that the common roots of f_1 and f_2 generate bivariate Vandermonde vectors in the null space of $\mathbf{M}(2)$:

$$\begin{pmatrix} -4 & 5 & -3 & -1 & 2 & 1 \\ -1 & 0 & 0 & 1 & 2 & 1 \end{pmatrix} \left(\begin{array}{c} 1 \\ \hline x_1 \\ x_2 \\ \hline x_1^2 \\ x_1x_2 \\ x_2^2 \end{array}\right) = \mathbf{0}. \tag{7}$$

The rank $r_{\mathbf{M}(2)} = 2$; hence the nullity of $\mathbf{M}(2)$ is $m = 4$.

At $d = 3$, $\mathbf{M}(3)$ contains four additional rows, which are the result of multiplying both f_1 and f_2 with both x_1^1 and x_2^1:

$$\mathbf{M}(3) = \begin{array}{c} \\ f_1(x_1,x_2) \\ f_2(x_1,x_2) \\ x_1f_1(x_1,x_2) \\ x_2f_1(x_1,x_2) \\ x_1f_2(x_1,x_2) \\ x_2f_2(x_1,x_2) \end{array} \begin{array}{c} \begin{array}{cccccccccc} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 & x_1^3 & x_1^2x_2 & x_1x_2^2 & x_2^3 \end{array} \\ \left(\begin{array}{c|cc|ccc|cccc} -4 & 5 & -3 & -1 & 2 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & -4 & 0 & 5 & -3 & 0 & -1 & 2 & 1 & 0 \\ 0 & 0 & -4 & 0 & 5 & -3 & 0 & -1 & 2 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 2 & 1 \end{array}\right) \end{array}.$$

The bivariate Vandermonde vectors in the null space of $\mathbf{M}(3)$ reach the additional monomials x_1^3, $x_1^2x_2$, $x_1x_2^2$, and x_2^3, and the dimension of the embedding space $\mathbb{C}^{10}$ of $\mathcal{M}_3$ has grown to 10. It can be verified that also $r_{\mathbf{M}(3)}$ has increased to 6, so that the nullity $10 - 6 = 4$ has remained unchanged: it is still equal to the number of solutions m of our set of polynomial equations.

Say we flip the columns of $\mathbf{M}(d)$ from left to right and bring the flipped matrix into reduced row echelon form. The monomials that correspond to the linearly dependent columns of the result are known as the *standard monomials* or the *normal set* [1, page 97]. They constitute a basis for the orthogonal complement of $\mathcal{M}_d$. For d greater than or equal to the so-called *degree of regularity* d^*, the null space of $\mathbf{M}(d)$ is completely isomorphic with $\mathcal{C}_d^n/\mathcal{I}$, its dimension $r(d) = \dim \mathcal{C}_d^n/\mathcal{I} = m$, and, most important, it contains all the necessary information to determine whether the associated system has any common roots [14, page 275]. This implies that for $d \geq d^*$ the nullity of $\mathbf{M}(d)$ does not change. From the study of resultants, for the square homogeneous case, i.e., $s = n + 1$, d^* is bounded by [5, page 104]:

$$d^* \leq \sum_{i=1}^{s}(d_i - 1) + 1 = \sum_{i=1}^{n+1} d_i - n. \tag{8}$$

For the square affine case, i.e., $s = n$, where one is interested in solutions in the projective space, one can take $d_{n+1} = 0$ in the right-hand side in (8) [22]. Example 3.5

illustrates how PNLA finds the solutions of a square affine system from an EVD of a basis matrix for the null space of the Macaulay matrix constructed at degree $d^* + 1$ [12].

Example 3.5. Consider again the system in Example 3.4. For this system $d^* + 1$ is bounded by $2+2-2+1 = 3$. Let us collect the $m = 4$ bivariate Vandermonde vectors that constitute a basis for $\mathrm{null}\,(\mathbf{M}(3))$ in a bivariate Vandermonde matrix $\mathbf{V}(3)$:

$$
\left(\begin{array}{cccc}
1 & 1 & 1 & 1 \\
\hline
x_1^{(1)} & x_1^{(2)} & x_1^{(3)} & x_1^{(4)} \\
x_2^{(1)} & x_2^{(2)} & x_2^{(3)} & x_2^{(4)} \\
\hline
x_1^{(1)2} & x_1^{(2)2} & x_1^{(3)2} & x_1^{(4)2} \\
x_1^{(1)}x_2^{(1)} & x_1^{(2)}x_2^{(2)} & x_1^{(3)}x_2^{(3)} & x_1^{(4)}x_2^{(4)} \\
x_2^{(1)2} & x_2^{(2)2} & x_2^{(3)2} & x_2^{(4)2} \\
\hline
x_1^{(1)3} & x_1^{(2)3} & x_1^{(3)3} & x_1^{(4)3} \\
x_1^{(1)2}x_2^{(1)} & x_1^{(2)2}x_2^{(2)} & x_1^{(3)2}x_2^{(3)} & x_1^{(4)2}x_2^{(4)} \\
x_1^{(1)}x_2^{(1)2} & x_1^{(2)}x_2^{(2)2} & x_1^{(3)}x_2^{(3)2} & x_1^{(4)}x_2^{(4)2} \\
x_2^{(1)3} & x_2^{(2)3} & x_2^{(3)3} & x_2^{(4)3}
\end{array}\right)
=
\left(\begin{array}{cccc}
1 & 1 & 1 & 1 \\
\hline
0 & 1 & 3 & 4 \\
-1 & 0 & -2 & -5 \\
\hline
0 & 1 & 9 & 16 \\
0 & 0 & -6 & -20 \\
1 & 0 & 4 & 25 \\
\hline
0 & 1 & 27 & 64 \\
0 & 0 & -18 & -80 \\
0 & 0 & 12 & 100 \\
-1 & 0 & -8 & -125
\end{array}\right). \tag{9}
$$

Multiplication of the kth column of $\mathbf{V}(3)$ with $x_1^{(k)}$ yields

$$
\mathbf{v}_k(3) \cdot x_1^{(k)} =
\left(\begin{array}{c}
1 \\
\hline
x_1^{(k)} \\
x_2^{(k)} \\
\hline
x_1^{(k)2} \\
x_1^{(k)}x_2^{(k)} \\
x_2^{(k)2} \\
\hline
x_1^{(k)3} \\
x_1^{(k)2}x_2^{(k)} \\
x_1^{(k)}x_2^{(k)2} \\
x_2^{(k)3}
\end{array}\right) \cdot x_1^{(k)} =
\left(\begin{array}{c}
x_1^{(k)} \\
\hline
x_1^{(k)2} \\
x_1^{(k)}x_2^{(k)} \\
\hline
x_1^{(k)3} \\
x_1^{(k)2}x_2^{(k)} \\
x_1^{(k)}x_2^{(k)2} \\
\hline
x_1^{(k)4} \\
x_1^{(k)3}x_2^{(k)} \\
x_1^{(k)2}x_2^{(k)2} \\
x_1 x_2^{(k)3}
\end{array}\right) \tag{10}
$$

for every value of k. Multiplication of the first six entries in $\mathbf{v}_k(3)$ with $x_1^{(k)}$ has the effect of the selection of entries from $\mathbf{v}_k(3)$ that is visible in the right-hand side of (10). On the other hand, the last four monomials in the right-hand side do not occur in $\mathbf{v}_k(3)$. To formalize things, let $\mathbf{S}_0, \mathbf{S}_1 \in \mathbb{R}^{6\times 10}$ denote the row selection matrices that select the rows of $\mathbf{v}_k(3)$ from degree 0 up to $d - 1 = 2$ and the rows onto which they are mapped after multiplication with $x_1^{(k)}$, respectively:[6]

$$
\mathbf{S}_0\mathbf{V}(3)\mathbf{D}_1 = \mathbf{S}_1\mathbf{V}(3) \tag{11}
$$

where $\mathbf{D}_1 = \mathrm{diag}(x_1^{(1)}, \ldots, x_1^{(4)})$.

In practice, one cannot readily compute $\mathbf{V}(3)$, as this would require knowledge of the roots. *It is possible*, however, to compute a numerical basis for the null space of

[6]We could as well have considered the multiplication of all rows with $x_2^{(k)}$. In practice, PNLA suggests to use a linear combination of multiplications with $x_j, j = 1 : n$. Section 9 will show that there exist means to simultaneously take *all* variables into account.

$\mathbf{M}(3)$ by means of standard linear algebra tools (e.g., an orthonormal basis). Stacking the numerical basis vectors in $\mathbf{K}(3) \in \mathbb{C}^{10\times 4}$, writing $\mathbf{K}(3) = \mathbf{V}(3)\mathbf{C}(3)^T$ where $\mathbf{C}(3)$ is an invertible basis transformation matrix, and plugging into (11), we obtain the rectangular GEVD

$$\mathbf{S}_0\mathbf{K}(3)\mathbf{C}(3)^{-T}\mathbf{D}_1 = \mathbf{S}_1\mathbf{K}(3)\mathbf{C}(3)^{-T}. \tag{12}$$

Equation (12) can be converted into the square EVD:

$$\mathbf{T}\mathbf{D}_1\mathbf{T}^{-1} = (\mathbf{S}_0\mathbf{K}(3))^{\dagger}\,\mathbf{S}_1\mathbf{K}(3) \quad \text{and} \quad \mathbf{T} = \mathbf{C}(3)^{-T}. \tag{13}$$

The eigenvalues correspond to the x_1 components of the solutions. The matrix $\mathbf{V}(3) = \mathbf{K}(3)\mathbf{T}$ reveals all solution components. Note that this was not possible for the smaller Macaulay matrix $\mathbf{M}(2)$. Indeed, for $d = d^*$ the selection matrices $\mathbf{S}_0, \mathbf{S}_1 \in \mathbb{R}^{3\times 6}$ lead to a (3×3) EVD that does not reveal $m = 4 > 3$ solutions.

3.3. The MHR problem. In this section we introduce the (M)HR problem and some relevant properties. This will help us understand how the structure in null space of the Macaulay matrix can be exploited.

Given a matrix $\mathbf{W} \in \mathbb{C}^{I\times M}$, the (1D) HR problem[7] consists of finding the factorization

$$\mathbf{W} = \mathbf{V}^{(1)}\mathbf{C}^T = \sum_{r=1}^{R} \mathbf{v}_r^{(1)} \otimes \mathbf{c}_r \tag{14}$$

where $\mathbf{V}^{(1)}(\{z_r\}_{r=1}^R) \in \mathbb{C}^{I\times R}$ is (univariate) Vandermonde and $\mathbf{C} \in \mathbb{C}^{M\times R}$ is unconstrained, if $\mathbf{W}$ admits such a factorization.[8] Due to its multiplicative shift structure, a Vandermonde matrix exhibits an important property called *shift invariance* [28, page 531]: let $\overline{\mathbf{V}}^{(1)}$ and $\underline{\mathbf{V}}^{(1)}$ denote the matrix $\mathbf{V}^{(1)}$ with its first and last row removed, respectively; then

$$\begin{pmatrix} \underline{\mathbf{V}}^{(1)} \\ \overline{\mathbf{V}}^{(1)} \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{v}}_1^{(1)} & \dots & \underline{\mathbf{v}}_R^{(1)} \\ \underline{\mathbf{v}}_1^{(1)} z_1 & \dots & \underline{\mathbf{v}}_R^{(1)} z_R \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ z_1 & z_2 & \dots & z_R \end{pmatrix} \odot \underline{\mathbf{V}}^{(1)} \stackrel{\text{def}}{=} \mathbf{V}^{(2,1)} \odot \underline{\mathbf{V}}^{(1)}.$$

The rth column of $\mathbf{V}^{(2,1)} \odot \underline{\mathbf{V}}^{(1)}$ is the Kronecker product of two vectors. Each such a column corresponds to a vectorized $(2 \times (I-1))$ rank-1 Hankel matrix:

$$\begin{aligned} \left(\mathbf{V}^{(2,1)} \odot \underline{\mathbf{V}}^{(1)}\right)_r = \begin{pmatrix} 1 \\ z_r \end{pmatrix} \otimes \begin{pmatrix} 1 \\ z_r \\ z_r^2 \\ \vdots \\ z_r^{I-2} \end{pmatrix} &= \operatorname{vec}\left(\begin{pmatrix} 1 \\ z_r \end{pmatrix} \begin{pmatrix} 1 & z_r & z_r^2 & \dots & z_r^{I-2} \end{pmatrix}\right) \\ &= \operatorname{vec}\left(\begin{pmatrix} 1 & z_r & \dots & z_r^{I-2} \\ z_r & z_r^2 & \dots & z_r^{I-1} \end{pmatrix}\right). \end{aligned} \tag{15}$$

Applying the same process to factorization (14), we obtain

$$\begin{pmatrix} \underline{\mathbf{W}} \\ \overline{\mathbf{W}} \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{V}}^{(1)} \\ \overline{\mathbf{V}}^{(1)} \end{pmatrix} \mathbf{C}^T = \left(\mathbf{V}^{(2,1)} \odot \underline{\mathbf{V}}^{(1)}\right) \mathbf{C}^T = \mathbf{Y}_{[1,2;3]}, \tag{16}$$

[7]In array processing terminology, we will more specifically discuss the "multichannel 1D HR problem."

[8]For clarity, $\mathbf{W}$ is given, and both $\mathbf{V}^{(1)}$ and $\mathbf{C}$ are unknown.

which is a matrix representation of the (C)PD of a two-slice tensor:

$$\mathcal{Y} = \left[\!\left[\mathbf{V}^{(2,1)}, \underline{\mathbf{V}}^{(1)}, \mathbf{C}\right]\!\right] = \sum_{r=1}^{R} \begin{pmatrix} 1 \\ z_r \end{pmatrix} \otimes \mathbf{v}_r^{(1)} \otimes \mathbf{c}_r \in \mathbb{C}^{2\times(I-1)\times M}. \tag{17}$$

The process relying on shift invariance, outlined above, is called *spatial smoothing*; it has allowed us to go from the second-order matrix model (14) to the third-order tensor model (17).

Example 3.6. HR is one of the basic problems in signal and array processing. Assume $R = 2$ source signals $\{c_{m1}\}$ and $\{c_{m2}\}$, transmitted at the same discrete time instances $m = 1, 2, \ldots, M$ and at the same frequency, but from different locations. After propagation, the signals are captured by a so-called uniform linear array consisting of $I = 3$ antennas, one of the antennas positioned exactly in the middle between the other two. We assume that the sources are in the "far field" of the array, meaning that the distance from source to array is substantially larger than the array itself. If we assemble the observations in a matrix $\mathbf{W} \in \mathbb{C}^{I\times M}$ where w_{im} gives the observation at antenna i at time m, then the data model which allows one to estimate the original source signals is given by

$$\mathbf{W} = \mathbf{V}^{(1)}\mathbf{C}^T = \begin{pmatrix} \left(\frac{1}{2}\right)^0 & \left(\frac{1}{3}\right)^0 \\ \left(\frac{1}{2}\right)^1 & \left(\frac{1}{3}\right)^1 \\ \left(\frac{1}{2}\right)^2 & \left(\frac{1}{3}\right)^2 \end{pmatrix} \mathbf{C}^T = \begin{pmatrix} 1 & 1 \\ \frac{1}{2} & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{9} \end{pmatrix} \mathbf{C}^T \tag{18}$$

where $\mathbf{C} \in \mathbb{C}^{M\times 2}$ holds the source signal values and $\mathbf{V}^{(1)}$ is the antenna response matrix; the latter is a Vandermonde matrix, of which the generators, here chosen equal to $z_1 = \frac{1}{2}$ and $z_2 = \frac{1}{3}$, depend on the angles of arrival with which the $R = 2$ signals impinge on the $M = 2$ antennas [25]. Leveraging the shift invariance property of $\mathbf{V}^{(1)}$, we obtain

$$\begin{pmatrix} \underline{\mathbf{V}}^{(1)} \\ \overline{\mathbf{V}}^{(1)} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{9} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ \frac{1}{2} & \frac{1}{3} \end{pmatrix} \odot \begin{pmatrix} 1 & 1 \\ \frac{1}{2} & \frac{1}{3} \end{pmatrix}.$$

Applying the very same spatial smoothing to the observed matrix $\mathbf{W}$, we obtain

$$\begin{pmatrix} \underline{\mathbf{W}} \\ \overline{\mathbf{W}} \end{pmatrix} = \mathbf{Y}_{[1,2;3]}$$

which is the matrix representation of a tensor $\mathcal{Y} \in \mathbb{C}^{2\times 2\times 2}$. Reorganizing the observed samples w_{im} in such a tensor $\mathcal{Y}$, we obtain the CPD $\mathcal{Y} = [[\mathbf{V}^{(2,1)}, \underline{\mathbf{V}}^{(1)}, \mathbf{C}]]$ with here $\mathbf{V}^{(2,1)} = \underline{\mathbf{V}}^{(1)}$. This CPD is unique if $r_{\mathbf{C}} = 2$, i.e., if the two source signals are not the same up to scaling.

Given a tensor $\mathcal{W} \in \mathbb{C}^{I_1\times I_2\times\ldots\times I_N\times M}$, the ($N$-dimensional) MHR problem consists of finding the constrained CPD

$$\mathcal{W} = \sum_{r=1}^{R} \mathbf{v}_r^{(1)} \otimes \mathbf{v}_r^{(2)} \otimes \ldots \otimes \mathbf{v}_r^{(N)} \otimes \mathbf{c}_r = \left[\!\left[\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(N)}, \mathbf{C}\right]\!\right] \tag{19}$$

where $\mathbf{V}^{(n)}(\{z_{r,n}\}_{r=1}^{R}) \in \mathbb{C}^{I_n \times R}$ is univariate Vandermonde, $n = 1 : N$, and $\mathbf{C} \in \mathbb{C}^{R\times M}$ is unconstrained, if $\mathcal{W}$ admits such a CPD. Analogous to the third-order case (6), the CPD in (19) can be matricized as

$$\mathbf{W}_{[1,2,\ldots,N;N+1]} = \left(\mathbf{V}^{(1)} \odot \ldots \odot \mathbf{V}^{(N)}\right)\mathbf{C}^T \in \mathbb{C}^{\left(\prod_{n=1}^{N} I_n\right)\times M}. \tag{20}$$

Equation (20) is a multivariate generalization of the univariate HR problem (14). With all factor matrices $\mathbf{V}^{(n)}$ Vandermonde, spatial smoothing is possible in each mode. Let $\overline{\mathbf{S}}^{(n)}$ and $\underline{\mathbf{S}}^{(n)}$ denote the row selection matrices that delete all rows of $\mathbf{W}_{[1,2,\ldots,N;N+1]}$ in (20) associated with the top and bottom row of $\mathbf{V}^{(n)}$, respectively. Formally, $\overline{\mathbf{S}}^{(n)}$ and $\underline{\mathbf{S}}^{(n)}$ can be defined as follows. Let $\overline{\mathbf{I}}_{I_n} \in \mathbb{R}^{(I_n-1)\times I_n}$ and $\underline{\mathbf{I}}_{I_n} \in \mathbb{R}^{(I_n-1)\times I_n}$ be extracted from the identity matrix $\mathbf{I}_{I_n}$ by deleting the top and bottom row, respectively. Then $\overline{\mathbf{S}}^{(n)} = \otimes_{p=1}^{n-1}\mathbf{I}_{I_p} \otimes \overline{\mathbf{I}}_{I_n} \otimes_{p=n+1}^{N}\mathbf{I}_{I_p}$ and $\underline{\mathbf{S}}^{(n)} = \otimes_{p=1}^{n-1}\mathbf{I}_{I_p} \otimes \underline{\mathbf{I}}_{I_n} \otimes_{p=n+1}^{N}\mathbf{I}_{I_p}$. Like spatial smoothing turned the second-order model (14) into the $(2+1)$th-order model (16), exploiting the multiplicative shift structure in the Vandermonde matrix $\mathbf{V}^{(n)}$ turns the $(N+1)$th-order model (20) into the $(N+2)$th-order model

$$\mathbf{Y}^{(n)} = \begin{pmatrix} \underline{\mathbf{S}}^{(n)}\mathbf{W}_{[1,2,\ldots,N;N+1]} \\ \overline{\mathbf{S}}^{(n)}\mathbf{W}_{[1,2,\ldots,N;N+1]} \end{pmatrix} = \left(\mathbf{V}^{(2,n)} \odot \mathbf{B}^{(n)}\right)\mathbf{C}^T \tag{21}$$

where

$$\mathbf{V}^{(2,n)} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ z_{1,n} & z_{2,n} & \cdots & z_{R,n} \end{pmatrix}, \quad \mathbf{B}^{(n)} = \left(\odot_{p=1}^{n-1}\mathbf{V}^{(n)}\right) \odot \underline{\mathbf{V}}^{(n)} \odot \left(\odot_{p=n+1}^{N}\mathbf{V}^{(n)}\right).$$

This can be expressed in a third-order tensor format, analogous to (17):

$$\mathcal{Y}^{(n)} = \left[\!\left[\mathbf{V}^{(2,n)}, \mathbf{B}^{(n)}, \mathbf{C}\right]\!\right] = \sum_{r=1}^{R} \begin{pmatrix} 1 \\ z_{r,n} \end{pmatrix} \otimes \mathbf{b}_r^{(n)} \otimes \mathbf{c}_r \in \mathbb{C}^{2\times\left(\left(\prod_{p=1}^{n} I_p\right)(I_n-1)\left(\prod_{p=n+1}^{N} I_p\right)\right)\times M}, \tag{22}$$

$n = 1 : N$. Let us take a step back here. So far, the 1-dimensional and N-dimensional case are not too different. By exploiting the structure of the problem, spatial smoothing allowed us to increase the order of the factorization by 1. The true difference arises if we exploit the structure not just once, but N times. Considered together, the $\left\{\mathcal{Y}^{(n)}\right\}_{n=1}^{N}$ in (22) admit a *coupled* CPD where the coupling takes place through the third factor matrix $\mathbf{C}$. An algebraic method to reduce such a coupled CPD to a matrix GEVD is given in [29, Algorithm 1]. By exploiting the structure in all modes together, [28] has derived the most relaxed MHR uniqueness conditions to date.

4. From the Macaulay null space to coupled CPD.

In the previous section we have displayed the ingredients needed to establish a connection between the structure in the null space of the Macaulay matrix and the CPD structure in the MHR problem. In this section we will explain how the roots of (1) can be obtained from a coupled CPD that is derived from the Macaulay null space. In section 5 the coupled CPD will be reduced to a single CPD.

From Example 3.4, we know that the null space of $\mathbf{M}(d)$ (at least for $d \geq d^*$) is generated by m multivariate Vandermonde vectors. Consistent with subsection 2.3 we stack these vectors in the multivariate Vandermonde matrix

$$\mathbf{V}(d) = \begin{pmatrix} \mathbf{v}_1(d) & \ldots & \mathbf{v}_m(d) \end{pmatrix} \in \mathbb{C}^{q(d)\times m}. \tag{23}$$

FIG. 2. *Illustration of the difference between the products that appear in MHR (section 3.3) and the products that determine the Macaulay null space (section 3.2) for the case $N = n = 2$ and $I_1 = I_2 = 4$.* (a) *The (4×4) square represents all products that appear in the outer product of the univariate Vandermonde vectors generated by z_1 and z_2. The dark and light shaded entries correspond to the rows of $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$ in (21), respectively. Clearly, $\mathbf{B}^{(1)} \neq \mathbf{B}^{(2)}$.* (b) *The triangle as a whole represents the rows of the multivariate Vandermonde matrix $\mathbf{V}(3)$ in (9), which correspond to the 10 monomials of degree $d \leq 3$. The filled entries correspond to the rows of $\mathbf{B}(2) \stackrel{\text{def}}{=} \mathbf{V}(2) = \mathbf{B}^{(j)}(2), j = 1:2$, in (27).*

Further,

$$\mathbf{V}^{(j)}(d) = \begin{pmatrix} \mathbf{v}_1^{(j)}(d) & \dots & \mathbf{v}_m^{(j)}(d) \end{pmatrix} \in \mathbb{C}^{(d+1)\times m} \tag{24}$$

denotes the univariate Vandermonde matrix of which the kth column is generated by the jth coordinate of the kth root $x_j^{(k)}$, $k = 1:m$, $j = 1:n$.

To contrast the derivation in the present section with the discussion in section 3.3, and in particular with the structure in (20), note that $\mathbf{v}_k(d) \neq \mathbf{v}_k^{(1)}(d) \otimes \cdots \otimes \mathbf{v}_k^{(n)}(d)$. Indeed, the entries of $\mathbf{v}_k(d)$ correspond to all the monomials up to degree d, while $\mathbf{v}_k^{(1)}(d) \otimes \cdots \otimes \mathbf{v}_k^{(n)}(d)$ also involves monomials of higher degree, but not all of them. (Compare (2) and (3); the difference is also illustrated in Figure 2.) Similarly, the multivariate Vandermonde matrix $\mathbf{V}(d)$ holds only the rows of the Khatri–Rao product $\mathbf{V}^{(1)}(d) \odot \cdots \odot \mathbf{V}^{(n)}(d)$ that correspond to the monomials up to degree d (and in a different order). Formally, we have

$$\mathbf{V}(d) = \mathbf{S}_{(d+1)^n \to q(d)} \left(\mathbf{V}^{(1)}(d) \odot \cdots \odot \mathbf{V}^{(n)}(d) \right) \in \mathbb{C}^{q(d)\times m} \tag{25}$$

where $\mathbf{S}_{(d+1)^n \to q(d)} \in \mathbb{C}^{q(d)\times(d+1)^n}$ denotes the row selection and ordering matrix that (i) selects all rows of the Khatri–Rao product that correspond to the $q(d)$ monomials from degree 0 up to degree d and (ii) permutes these rows to the degree negative lexicographic order.

In practice, it is a numerical basis of $\operatorname{null}(\mathbf{M}(d))$ that will be computed. The matrix $\mathbf{K}(d) \in \mathbb{C}^{q(d)\times m}$ in which such a numerical basis is stacked is related to the matrix of multivariate Vandermonde vectors $\mathbf{V}(d)$ by an invertible transformation: $\mathbf{K}(d) = \mathbf{V}(d)\mathbf{C}(d)^T$.[9] Substitution of (25) yields the following variant of the MHR model (20):

$$\mathbf{K}(d) = \mathbf{S}_{(d+1)^n \to q(d)} \left(\mathbf{V}^{(1)}(d) \odot \cdots \odot \mathbf{V}^{(n)}(d) \right) \mathbf{C}(d)^T \in \mathbb{C}^{q(d)\times m}. \tag{26}$$

[9] Perhaps less obviously, $\mathbf{C}(d)$ depends on d as well. Indeed, the $q(d-1)$ top rows of $\mathbf{V}(d)$ equal the rows of $\mathbf{V}(d-1)$, but this does not hold for the computed $\mathbf{K}(d)$ and $\mathbf{K}(d-1)$. For instance, $\mathbf{K}(d-1)$ and $\mathbf{C}(d-1)^{-T}$ could be the orthogonal and triangular factor in a QR-factorization of $\mathbf{V}(d-1)$; it is clear that $\mathbf{C}(d-1)^{-T}$ does not necessarily orthogonalize the larger matrix $\mathbf{V}(d)$ as well.

Note that the matrix $\mathbf{C}(d)$ is square and that its size corresponds to the number of solutions to the polynomial system, i.e., in the notation of section 3.3 we have $M = R = m$.

Now let us investigate the counterpart of the coupled CPD in (22). As in section 3.3, we can apply spatial smoothing in each mode, i.e., for each variable x_j. Let $\overline{\mathbf{S}}^{(j)}(d-1) \in \mathbb{C}^{q(d-1)\times q(d)}$ and $\underline{\mathbf{S}}^{(j)}(d-1) \in \mathbb{C}^{q(d-1)\times q(d)}$ denote two additional row selection matrices (i.e., they implement a further selection on top of the selection by $\mathbf{S}_{(d+1)^n\to q(d)}$ in (26)). The matrix $\underline{\mathbf{S}}^{(j)}(d-1)$ selects all the rows that correspond to the $q(d-1)$ monomials from degree 0 up to degree $d-1$, so that globally $\underline{\mathbf{S}}^{(j)}(d-1)\cdot \mathbf{S}_{(d+1)^n\to q(d)} = \mathbf{S}_{(d+1)^n\to q(d-1)}$. Note that $\underline{\mathbf{S}}^{(j)}(d-1)$ is the same for all j. On the other hand, the matrix $\overline{\mathbf{S}}^{(j)}(d-1)$ does depend on j; it selects all the rows that correspond to the $q(d-1)$ monomials up to degree d that have at least degree 1 in x_j. In Figure 2(b), $\underline{\mathbf{S}}^{(1)}(2) = \underline{\mathbf{S}}^{(2)}(2)$ would select the filled entries, $\overline{\mathbf{S}}^{(1)}(2)$ the filled entries shifted one position down, and $\overline{\mathbf{S}}^{(2)}(2)$ the filled entries shifted one position to the right.

Exploiting the multiplicative shift structure in the corresponding univariate Vandermonde matrix $\mathbf{V}^{(j)}(d)$ yields

$$(27)\qquad \mathbf{Y}^{(j)} = \begin{pmatrix} \underline{\mathbf{S}}^{(j)}(d-1)\cdot \mathbf{K}(d) \\ \overline{\mathbf{S}}^{(j)}(d-1)\cdot \mathbf{K}(d) \end{pmatrix} = \left(\mathbf{V}^{(2,j)} \odot \mathbf{B}(d-1)\right)\mathbf{C}(d)^T \in \mathbb{C}^{(2\cdot q(d-1))\times m}$$

where

$$\mathbf{V}^{(2,j)} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_j^{(1)} & x_j^{(2)} & \dots & x_j^{(m)} \end{pmatrix} \in \mathbb{C}^{2\times m}$$

and $\mathbf{B}(d-1) = \mathbf{V}(d-1) \in \mathbb{C}^{q(d-1)\times m}$ contains the top rows of $\mathbf{V}(d)$ that correspond to the $q(d-1)$ monomials from degree 0 up to degree $d-1$. Expressing (27) in a third-order tensor format, similar to (22), yields

$$(28)\qquad \begin{aligned} \mathcal{Y}^{(j)} &= \left[\!\left[\mathbf{V}^{(2,j)}, \mathbf{B}(d-1), \mathbf{C}(d)\right]\!\right] \\ &= \sum_{k=1}^{m} \begin{pmatrix} 1 \\ x_j^{(k)} \end{pmatrix} \otimes \mathbf{b}_k(d-1) \otimes \mathbf{c}_k(d) \in \mathbb{C}^{2\times q(d-1)\times m}, \qquad j = 1:n. \end{aligned}$$

Equations (27)/(28) and (21)/(22) are similar, but there is important difference: the matrix $\mathbf{B}(d-1)$ in (27)/(28) is the same for all j. More precisely, we have $\mathbf{B}(d-1) \overset{\text{def}}{=} \mathbf{V}(d-1) = \mathbf{B}^{(j)}(d-1), j = 1:n$. Indeed, to ensure that the rows, onto which the rows of $\mathbf{B}(d-1)$ are mapped after multiplication with the second row of $\mathbf{V}^{(2,j)}$, occur in $\mathbf{K}(d)$, we need to remove *all* rows of degree d—rather than only the rows in which x_j has degree d, as was the case in section 3.3. Consequently, the matrices $\left\{\mathbf{Y}^{(j)}\right\}_{j=1}^{n}$ in (27) have their first $q(d-1)$ rows in common; these are the rows of $\mathbf{V}(d-1)\mathbf{C}(d)^T$. In Figure 2(b) the rows of $\mathbf{V}(d-1)$ correspond to the filled entries.

Example 4.1. Consider again $\mathbf{V}(3)$ in Example 3.5. We have

$$\mathbf{V}^{(2,1)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ x_1^{(1)} & x_1^{(2)} & x_1^{(3)} & x_1^{(4)} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 4 \end{pmatrix}$$

and, using MATLAB notation for indexing,

$$\mathbf{B}(2) \overset{\text{def}}{=} \mathbf{B}^{(1)}(2) = \mathbf{B}^{(2)}(2) = \mathbf{V}(2) = \begin{pmatrix} \mathbf{v}_1(2) & \mathbf{v}_2(2) & \mathbf{v}_3(2) & \mathbf{v}_4(2) \end{pmatrix} = \mathbf{V}(3)(1:6,:),$$

where the rows of the latter correspond to the black triangle in Figure 2(b). It is easy to verify that

$$\left(\mathbf{V}^{(2,1)} \odot \mathbf{B}(2)\right) \mathbf{C}(3)^T = \begin{pmatrix} 1\cdot\mathbf{v}_1(2) & 1\cdot\mathbf{v}_2(2) & 1\cdot\mathbf{v}_3(2) & 1\cdot\mathbf{v}_4(2) \\ \hline 0\cdot\mathbf{v}_1(2) & 1\cdot\mathbf{v}_2(2) & 3\cdot\mathbf{v}_3(2) & 4\cdot\mathbf{v}_4(2) \end{pmatrix} \mathbf{C}(3)^T \tag{29}$$

$$= \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 4 \\ -1 & 0 & -2 & -5 \\ 0 & 1 & 9 & 16 \\ 0 & 0 & -6 & -20 \\ 1 & 0 & 4 & 25 \\ \hline 0 & 1 & 3 & 4 \\ 0 & 1 & 9 & 16 \\ 0 & 0 & -6 & -20 \\ 0 & 1 & 27 & 64 \\ 0 & 0 & -18 & -80 \\ 0 & 0 & 12 & 100 \end{pmatrix} \mathbf{C}(3)^T = \begin{pmatrix} \underline{\mathbf{S}}^{(1)}(2)\cdot\mathbf{K}(3) \\ \hline \overline{\mathbf{S}}^{(1)}(2)\cdot\mathbf{K}(3) \end{pmatrix} = \mathbf{Y}^{(1)}.$$

Note that

$$\underline{\mathbf{S}}^{(1)}(2) = \begin{pmatrix} \mathbf{I}_{q(2)} & \mathbf{0}_{q(2)\times\binom{4}{3}} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_6 & \mathbf{0}_{6\times 4} \end{pmatrix}$$

deletes all rows from $\mathbf{K}(3)$ (and $\mathbf{V}(3)$; see (9)) associated with the entries that are white in Figure 2(b). On the other hand,

$$\overline{\mathbf{S}}^{(1)}(2) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

deletes all rows associated with the entries that are white after shifting the black triangle down over one position. (Likewise, $\overline{\mathbf{S}}^{(2)}(2)$ deletes all rows associated with the entries that are white after shifting the black triangle to the right over one position.)

5. From coupled CPD to CPD.

When considered together, the tensors $\left\{\mathcal{Y}^{(j)}\right\}_{j=1}^{n}$ in (28) admit a coupled CPD. Unlike the coupled CPD (22) that we obtained for MHR in section 4, the coupled CPD for polynomial equations in (28) can easily be reduced to a single CPD of a third-order tensor, which in turn can be computed by means of a matrix GEVD. In subsection 5.1 we first consider the case of only affine roots. In subsection 5.2 we also allow roots at infinity.

5.1. Simple affine case.

Because the $\mathbf{Y}^{(j)}$ do not only have the third factor matrix $\mathbf{C}(d)$ in common but also the second factor matrix $\mathbf{B}(d-1)$, simple stacking yields

$$\mathbf{Y}^{\text{stack}}_{[1,2;3]} \overset{\text{def}}{=} \begin{pmatrix} \mathbf{Y}^{(1)} \\ \mathbf{Y}^{(2)} \\ \vdots \\ \mathbf{Y}^{(n)} \end{pmatrix} = \left(\begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(m)} \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \dots & 1 \\ x_n^{(1)} & x_n^{(2)} & \dots & x_n^{(m)} \end{pmatrix} \odot \mathbf{B}(d-1) \right) \mathbf{C}(d)^T. \tag{30}$$

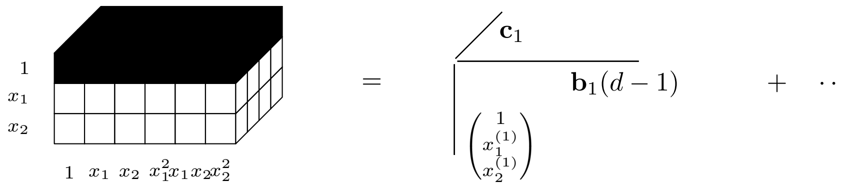


FIG. 3. *The horizontal slices of the third-order tensor $\mathcal{Y} \in \mathbb{C}^{(n+1)\times q(d-1)\times m}$ in* (32)*, for $n = 2$, $d = 3$, and $m = 4$, contain the rows that correspond to the filled entries in Figure* 2(b)*, the entries shifted one position downwards (x_1), and the entries shifted one position to the right (x_2), respectively.*

Dropping the redundant rows of the first factor matrix in (30) and the corresponding redundant rows of $\mathbf{Y}^{\text{stack}}_{[1,2;3]}$, we obtain

$$
\begin{aligned}
\mathbf{Y}_{[1,2;3]} &\stackrel{\text{def}}{=} \begin{pmatrix} \underline{\mathbf{S}}^{(1)}(d-1)\cdot \mathbf{K}(d) \\ \overline{\mathbf{S}}^{(1)}(d-1)\cdot \mathbf{K}(d) \\ \vdots \\ \overline{\mathbf{S}}^{(n)}(d-1)\cdot \mathbf{K}(d) \end{pmatrix} \\
&= \left(\begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(m)} \\ \vdots & \vdots & & \vdots \\ x_n^{(1)} & x_n^{(2)} & \dots & x_n^{(m)} \end{pmatrix} \odot \mathbf{B}(d-1) \right) \mathbf{C}(d)^T \\
&= (\mathbf{V}(1) \odot \mathbf{V}(d-1))\, \mathbf{C}(d)^T \in \mathbb{C}^{(n+1)\cdot q(d-1)\times m}.
\end{aligned} \tag{31}
$$

In the third-order tensor format we have

$$
\mathcal{Y} = [\![\mathbf{V}(1), \mathbf{V}(d-1), \mathbf{C}(d)]\!] = \sum_{k=1}^{m} \mathbf{v}_k(1) \otimes \mathbf{v}_k(d-1) \otimes \mathbf{c}_k(d) \in \mathbb{C}^{(n+1)\times q(d-1)\times m}; \tag{32}
$$

see Figure 3 for an illustration.

5.2. Simple projective case. Let us now drop the constraint that there are only affine roots. Equations (31) and (32) admit the projective interpretation

$$
\begin{aligned}
\mathbf{Y}_{[1,2;3]} &= \begin{pmatrix} \overline{\mathbf{S}}^{(0)}(d-1)\cdot \mathbf{K}(d) \\ \overline{\mathbf{S}}^{(1)}(d-1)\cdot \mathbf{K}(d) \\ \vdots \\ \overline{\mathbf{S}}^{(n)}(d-1)\cdot \mathbf{K}(d) \end{pmatrix} \\
&= \left(\begin{pmatrix} x_0^{(1)} & x_0^{(2)} & \dots & x_0^{(m)} \\ x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(m)} \\ \vdots & \vdots & & \vdots \\ x_n^{(1)} & x_n^{(2)} & \dots & x_n^{(m)} \end{pmatrix} \odot \mathbf{B}^h(d-1) \right) \mathbf{C}(d)^T \\
&= \left(\mathbf{V}^h(1) \odot \mathbf{B}^h(d-1)\right) \mathbf{C}(d)^T \in \mathbb{C}^{((n+1)\cdot q(d-1))\times m},
\end{aligned} \tag{33}
$$

(34)

$$\mathcal{Y} = [\![\mathbf{V}^h(1), \mathbf{V}^h(d-1), \mathbf{C}(d)]\!] = \sum_{k=1}^{m} \mathbf{v}_k^h(1) \otimes \mathbf{v}_k^h(d-1) \otimes \mathbf{c}_k(d) \in \mathbb{C}^{(n+1)\times q(d-1)\times m},$$

respectively, in which $\overline{\mathbf{S}}^{(0)}(d-1) \overset{\text{def}}{=} \underline{\mathbf{S}}^{(1)}(d-1)$, and $\mathbf{B}^h(d) \overset{\text{def}}{=} \mathbf{V}^h(d) = \begin{pmatrix}\mathbf{v}_1^h(d) & \dots & \mathbf{v}_m^h(d)\end{pmatrix} \in \mathbb{C}^{q(d)\times m}$ with

$$\text{(35)} \quad \mathbf{v}_k^h(d) = \left(\mathbf{V}^h(d)\right)_k \overset{\text{def}}{=} \begin{pmatrix} x_0^{(k)d} & x_0^{(k)d-1}x_1^{(k)} & \dots & x_0^{(k)d-2}x_1^{(k)2} & x_0^{(k)d-2}x_1^{(k)}x_2^{(k)} & \dots & x_n^{(k)d}\end{pmatrix}^T \in \mathbb{C}^{q(d)}.$$

Recall from subsection 3.1 that CPD is always subject to trivial scaling indeterminacies, i.e., the corresponding columns of the different factor matrices can be scaled/counterscaled as long as the overall rank-1 terms do not change. These indeterminacies can now be interpreted very naturally as scaling equivalences in the coordinates of a solution point in the projective space $\mathbb{P}^n$. In (33), (34) roots at infinity are handled in the same way as affine roots. The only difference is whether the value $x_0^{(k)} = 0$ or not.

5.3. Computing only affine roots. In practice, computing only the affine roots might be sufficient as the roots at infinity are typically of less interest. In [13, 12] strategies are proposed that restrict the computation to the affine roots only. Having computed a null space basis $\mathbf{K}(d)$ of $\mathbf{M}(d)$, one can separate the parts associated to the roots at infinity and the affine roots by a column compression of $\mathbf{K}(d)$. The number $m_a \leq m$ of affine roots corresponds to the cardinality of the set of affine standard monomials [1], a subset of the set of standard monomials associated to the linearly independent rows of $\mathbf{K}(d)$. As shown in [12, 1] a precise knowledge of those sets is not needed since m_a can be read off easily from $\mathbf{M}(d)$ by basic rank decisions. As a matter of fact, this detection can already be done during the construction of the numerical null space [1, Algorithm 5.1]. Define $p \overset{\text{def}}{=} q(\hat{d})$, where $\hat{d}$ is the highest degree within the affine standard monomials. Let $\mathbf{K}(d) = \begin{pmatrix}\mathbf{K}_1^T & \mathbf{K}_2^T\end{pmatrix}^T$ with $\mathbf{K}_1 \in \mathbb{C}^{p\times m}$ be a corresponding partition of $\mathbf{K}(d)$, and let $\mathbf{K}_1 = \mathbf{U\Sigma Q}^T$ denote the SVD of $\mathbf{K}_1$. Then

$$\hat{\mathbf{K}} \overset{\text{def}}{=} \mathbf{K}(d)\mathbf{Q} = \begin{pmatrix}\hat{\mathbf{K}}_{11} & \mathbf{0} \\ \hat{\mathbf{K}}_{21} & \hat{\mathbf{K}}_{22}\end{pmatrix}$$

yields $\hat{\mathbf{K}}_{11} \in \mathbb{C}^{p\times m_a}$, containing all the required information for the $m_a \leq m$ affine roots. PNLA then continues the GEVD-based root finding as illustrated in Example 3.5 using $\hat{\mathbf{K}}_{11}$ and appropriate selection matrices associated with the reduced degree $\hat{d}$; see [12, Theorem 6.10]. For our approach this entails using $\hat{\mathbf{K}}_{11}$ and $\overline{\mathbf{S}}^{(j)}(\hat{d})$, $\underline{\mathbf{S}}^{(j)}(\hat{d}-1)$ in (31).

Alternatively, since the roots at infinity correspond to the highest degree standard monomials, one can work with a reduced Macaulay matrix where the associated columns have been discarded.

The potential downside of both approaches is that they may still require the construction of a relatively large Macaulay matrix (and the computation of its null space) in order to extract a possibly small number m_a of affine roots. For computational efficiency it would be desirable to a priori deflate roots at infinity from the system. We leave this issue for future research.

6. CPD, GEVD, and NPA for $d \geq d^* + 1$.

6.1. CPD and GEVD. In the case $d \geq d^* + 1$, the CPD in (32)/(34) can directly be connected to Theorem 3.1.

THEOREM 6.1. *Let* $\mathbf{Y}_{[1,2;3]} \in \mathbb{C}^{((n+1)\cdot q(d-1))\times m}$ *be derived from* $\mathbf{M}(d)$ *with* $d \geq d^* + 1$ *as in subsection* 5.1/*subsection* 5.2. *Then* $r_{\mathcal{Y}} = m$, *and the CPD of* $\mathcal{Y}$ *in* (32)/(34) *is unique.*

Proof. It suffices to show that all the conditions in Theorem 3.1 are satisfied for decomposition (32) if $d \geq d^* + 1$. For (34) it suffices to add the superscript $\cdot^h$.

- If all roots are simple, then no columns in $\mathbf{V}(1)$ are collinear: $k_{\mathbf{V}(1)} \geq 2$.
- If all roots are simple and $d \geq d^*$, then $\mathbf{K}(d)$ is related to $\mathbf{V}(d)$ by $\mathbf{K}(d) = \mathbf{V}(d)\mathbf{C}(d)^T$ in which $\mathbf{C}(d) \in \mathbb{C}^{m\times m}$ is invertible and thus $\mathbf{C}(d)$ has full column rank m.
- The m standard monomials correspond to the linearly independent rows of $\mathbf{V}(d)$. At least one standard monomial has exactly degree d^*, meaning that $d \geq d^* + 1$ guarantees that $\dim \mathrm{row}\,(\mathbf{V}(d-1)) = \dim \mathrm{row}\,(\mathbf{V}(d^*)) = m$, such that also $\mathbf{V}(d-1) \in \mathbb{C}^{q(d-1)\times m}$ has full column rank m. ∎

Since for $d \geq d^* + 1$ the conditions in Theorem 3.1 are satisfied, the CPD of $\mathcal{Y}$ is not only unique; it can be computed by a matrix GEVD; cf. the discussion in subsection 3.1.

6.2. Connection with NPA. The ESPRIT-like reasoning in section 4 allows us to further interpret (32). As illustrated in Example 3.5, the exploitation of the multiplicative shift structure in x_1 in the null space of the Macaulay matrix derives from the system of polynomial equations a single rectangular GEVD or a single square EVD (for the example, given in (12) and (13), respectively). The exploitation of the multiplicative shift structure in *all* variables in the CPD of the $(n+1)$-slice third-order tensor $\mathcal{Y}$ in (32) can be interpreted as the *joint* EVD of n matrices. Corollary 6.3 below demonstrates that there is in fact a tight connection between (32) and the joint diagonalization of the n so-called "multiplication tables" $\left\{\mathbf{A}_{x_j}\right\}_{j=1}^{n}$ in NPA's Theorem 6.2 in the simple affine case.

THEOREM 6.2 (central theorem of NPA [30, Theorem 2.27]). *Let the system of polynomials* $\mathcal{F}$ *have* $m_0 \leq m$ *disjoint roots. Consider the family of multiplication tables* $\left\{\mathbf{A}_{x_j}\right\}_{j=1}^{n}$. *The matrix* $\mathbf{A}_h \in \mathbb{C}^{m\times m}$ *represents a multiplication with the residue class* $[h]$ *in the* m*-dimensional quotient ring* $\mathcal{C}^n/\mathcal{I} = \mathcal{C}^n/\langle\mathcal{F}\rangle$ *w.r.t. an arbitrary basis, e.g., the normal set denoted by* $\{[t_k]\}_{k=1}^{m}$:

$$\phi_h : \mathcal{C}^n/\mathcal{I} \to \mathcal{C}^n/\mathcal{I} : \begin{pmatrix} [t_1] \\ \vdots \\ [t_m] \end{pmatrix} \mapsto \begin{pmatrix} [h \cdot t_1] \\ \vdots \\ [h \cdot t_m] \end{pmatrix} = \mathbf{A}_h \begin{pmatrix} [t_1] \\ \vdots \\ [t_m] \end{pmatrix}.$$

For each μ_k*-fold root* $\mathbf{x}^{(k)}, k = 1 : m_0$, *the matrices* $\mathbf{A}_{x_j}$ *have* $x_j^{(k)}$ *as an eigenvalue of multiplicity* μ_k *and the associated joint eigenvector*

$$\left([t_1(\mathbf{x}^{(k)})] \quad \dots \quad [t_m(\mathbf{x}^{(k)})]\right)^T \in \mathrm{span}\,(\mathbf{X}_k)\,.$$

Here, $\mathrm{span}\,(\mathbf{X}_k)$ *denotes the associated joint invariant subspace of dimension* μ_k *such that*

$$\mathbf{A}_{x_j}\begin{pmatrix}\mathbf{X}_1 & \dots & \mathbf{X}_{m_0}\end{pmatrix} = \begin{pmatrix}\mathbf{X}_1 & \dots & \mathbf{X}_{m_0}\end{pmatrix}\begin{pmatrix}\mathbf{T}_{x_{j,1}} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{T}_{x_{j,m_0}}\end{pmatrix}, \tag{36}$$

where $\mathbf{T}_{x_{j,k}} \in \mathbb{C}^{\mu_k \times \mu_k}$ *is upper-triangular with diagonal entries* $x_j^{(k)}$.

If $m_0 = m$, i.e., if all roots are simple, then Theorem 6.2 implies that (36) is an EVD and that the set of matrices $\left\{\mathbf{A}_{x_j}\right\}_{j=1}^n$ is jointly diagonalizable.

COROLLARY 6.3. *Let the polynomial system $\mathcal{F}$ have m roots, and let the column echelon basis of* null$(\mathbf{M}(d))$ *be stacked in the matrix* $\mathbf{H}(d)$.[10] *Consider the third-order tensor $\mathcal{H}(d) \in \mathbb{C}^{n\times m\times m}$ with matrix representation*

$$\mathbf{H}_{[1,2;3]}(d) = \begin{pmatrix}\hat{\bar{\mathbf{S}}}^{(1)}(d-1)\cdot\mathbf{H}(d) \\ \vdots \\ \hat{\bar{\mathbf{S}}}^{(n)}(d-1)\cdot\mathbf{H}(d)\end{pmatrix} \in \mathbb{C}^{(n\cdot m)\times m},$$

where $\hat{\bar{\mathbf{S}}}^{(j)}(d-1)$ *denotes the row selection matrix that selects the rows of* $\mathbf{H}(d)$ *onto which the m standard monomials are mapped after multiplication with x_j. If*

1. *all roots are simple,*
2. *all roots are affine, and*
3. $d = d^* + 1$,

then the n slices $\{\mathbf{H}_j(d) \overset{\text{def}}{=} \mathcal{H}(j,:,:)(d)\}_{j=1}^n$ *are equal to the n multiplication tables w.r.t. the normal set basis for the quotient ring* $\mathcal{C}^n/\langle\mathcal{F}\rangle$.

Proof. The structure in (31) does not rely on the specific choice $\mathbf{K}(d) = \mathbf{V}(d)\mathbf{C}(d)^T$ that is made for the basis of null$(\mathbf{M}(d))$, so the CPD (32) holds for $\mathbf{K}(d) = \mathbf{H}(d)$ as well and

$$\mathbf{H}_j(d) = \hat{\mathbf{B}}(d-1)\cdot\mathbf{D}_j(\mathbf{V}(2:n+1,:))\cdot\mathbf{C}(d)^T.$$

The matrix $\hat{\mathbf{B}}(d-1) \in \mathbb{C}^{m\times m}$ contains the m rows of $\mathbf{B}(d-1) \in \mathbb{C}^{q(d-1)\times m}$ that correspond to the m standard monomials. At least one standard monomial has exactly degree d^*, meaning that one needs to choose $d = d^* + 1$ for $\mathbf{B}(d-1)$ to contain the rows corresponding to all standard monomials. Let $\mathbf{V}(d) = \mathbf{H}(d)\mathbf{T}$ where $\mathbf{T} = \begin{pmatrix}\mathbf{t}_1 & \dots & \mathbf{t}_m\end{pmatrix} \in \mathbb{C}^{m\times m}$ is an invertible transformation matrix and $\mathbf{C}(d)^T = \mathbf{T}^{-1}$. [15, Proposition 1] shows that $\mathbf{t}_k$ contains the m standard monomials evaluated at the solution $\mathbf{x}^{(k)}$. From this, $\hat{\mathbf{B}}(d-1) = \mathbf{T}$ and

$$\mathbf{H}_j(d) = \mathbf{T}\,\mathrm{diag}(x_j^{(1)},\dots,x_j^{(m)})\mathbf{T}^{-1} = \mathbf{A}_{x_j}, \quad j = 1:n, \tag{37}$$

where the last equality is implied by Theorem 6.2 for simple affine roots. ∎

We give an example that connects the insights that have emerged for multivariate polynomial equations to the basic univariate case.

Example 6.4. Consider the univariate polynomial equation of degree $d = 2$

$$f(x) = a_d x^2 + a_{d-1}x + a_{d-2} = x^2 + a_1 x + a_0 = x^2 - \frac{5}{6}x + \frac{1}{6} = 0. \tag{38}$$

[10] The matrix $\mathbf{H}(d) \in \mathbb{C}^{q(d)\times m}$ is such that its top m rows form $\mathbf{I}_m$; see Example 6.4 for an illustration.

Flipping the columns of $\mathbf{f}^T = (\,1 \quad -\frac{5}{6} \quad \frac{1}{6}\,)$ from left to right and reduction to the row echelon form

$$\begin{array}{ccc} x^2 & x & 1 \\ (1 & 0 & 0) \end{array}$$

yields the normal set $\{1, x\}$ as the monomials associated with the last two columns.

The Frobenius companion matrix of f (with $a_d = 1$)

$$\mathbf{A}_x \stackrel{\text{def}}{=} \left(\begin{array}{c|c} \mathbf{0}_{(d-1)\times 1} & \mathbf{I}_{d-1} \\ \hline -a_0 \ \ldots & -\ a_{d-1} \end{array}\right) = \begin{pmatrix} 0 & 1 \\ -\frac{1}{6} & \frac{5}{6} \end{pmatrix}$$

can be interpreted as the matrix that describes the effect of multiplying $\{1, x\}$ with $h = x$ in terms of $\{1, x\}$, i.e., as a multiplication table:

$$\begin{aligned} x \cdot (1 \cdot 1 + 0 \cdot x) &= 0 \cdot 1 + 1 \cdot x, \\ x \cdot (0 \cdot 1 + 1 \cdot x) &= 1 \cdot x^2 = -\frac{1}{6} \cdot 1 + \frac{5}{6} \cdot x. \end{aligned}$$

The $m = d = 2$ simple roots $x^{(1)} = \frac{1}{2}$ and $x^{(2)} = \frac{1}{3}$ of f are obtained as the isolated eigenvalues of the multiplication table $\mathbf{A}_x$.

Next, as mentioned in the proof of Corollary 6.3, $\mathbf{Y}_{[1,2;3]}$ in (31) may be constructed from $\mathbf{H}(2)$ as a special case of $\mathbf{K}(2) = \mathbf{V}(2)\mathbf{C}(2)^T$:

$$\mathbf{Y}_{[1,2;3]} = \left(\frac{\left(\mathbf{I}_2 \quad \mathbf{0}_{2\times 1}\right) \cdot \mathbf{H}(2)}{\left(\mathbf{0}_{2\times 1} \quad \mathbf{I}_2\right) \cdot \mathbf{H}(2)}\right) = \left(\frac{\underline{\mathbf{H}}(2)}{\overline{\mathbf{H}}(2)}\right) = \left(\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ \hline 0 & 1 \\ -\frac{1}{6} & \frac{5}{6} \end{array}\right). \tag{39}$$

It is easy to verify that $\mathcal{Y}$ can be written as $\mathcal{Y} = [\![\mathbf{V}(1), \mathbf{V}(1), \mathbf{C}(2)]\!]$ where

$$\mathbf{V}(1) = \begin{pmatrix} 1 & 1 \\ \frac{1}{2} & \frac{1}{3} \end{pmatrix} \quad \text{and} \quad \mathbf{C}(2) = \begin{pmatrix} -2 & 3 \\ 6 & -6 \end{pmatrix}.$$

The factor matrices in the two-slice CPD follow from the GEVD of the matrix pencil $(\mathcal{Y}(1,:,:), \mathcal{Y}(2,:,:)) = \left(\underline{\mathbf{H}}(2), \overline{\mathbf{H}}(2)\right)$. As $\underline{\mathbf{H}}(2) = \mathbf{I}_2$ and $\overline{\mathbf{H}}(2) = \mathbf{A}_x$, the GEVD matches the EVD of $\mathbf{A}_x$.

Note that, for the univariate polynomial equation (38), $\mathbf{H}(2) = \mathbf{V}(2)\mathbf{C}(2)^T$ is an instance of the 1D HR problem (14) in subsection 3.3 and that (39) corresponds to its spatially smoothed variant (16).

Let us also contrast the way the projective case is handled in (34) to PNLA. PNLA proposes some "artificial" solutions to cope with roots at infinity: either the affine roots are separated from the roots at infinity in $\mathbf{K}(d)$ at a degree $d \gg d^*$ or projective shift relations are introduced to make the EVD work [15].

7. CPD and GEVD for $d \geq d^*$. So far, we have obtained the insightful tensor CPD interpretation in (32)/(34), which comes with numerical tensor algorithms and a uniform way of handling affine roots and roots at infinity. However, section 6 hasn't really offered new or less restrictive *working conditions* than NPA. We will take this step in the present section. Recall that NPA works with the Macaulay matrix $\mathbf{M}(d)$ at $d = d^* + 1$. It turns out that in the CPD approach it is possible to work with the smaller Macaulay matrix $\mathbf{M}(d)$ at $d = d^*$.

Theorem 7.2 establishes the *generic* uniqueness of (34) at a degree $d \geq d^*$. A generic uniqueness condition is meaningful, as in this article our assumptions of a 0-dimensional solution set with m solutions in the projective space and of only simple roots were in fact already generic. First, Definition 7.1 draws from [11] to explain when we say that decomposition (33) is generically unique.

Definition 7.1. *Let $\Omega \subset \mathbb{C}^{m\cdot(n+1)}$ be the subset of vectors with $m(n+1)$ entries, where all m roots of a set of n homogeneous polynomials in $n+1$ variables are stacked vertically. Let $\mathbf{z} \in \Omega$ contain the roots of a system of n homogeneous polynomials in $n+1$ variables,*[11] *and let $\mu_{m\cdot(n+1)}$ be a measure that is absolutely continuous w.r.t. the Lebesgue measure on $\mathbb{C}^{m\cdot(n+1)}$. The CPD (34) is generically unique iff*

$$\mu_{m\cdot(n+1)}\{\mathbf{z} \in \Omega \mid \text{the CPD of the tensor} \quad \mathcal{Y} = [\![(\mathbf{V}^h(1))(\mathbf{z}), (\mathbf{B}^h(d-1))(\mathbf{z}), (\mathbf{C}(d))(\mathbf{z})]\!] \text{ in (34) is not unique}\} = 0. \tag{40}$$

Let us have a look at how the factor matrices depend on the parameter vector $\mathbf{z}$. First, as $\mathbf{V}^h(1)$ holds all the roots, we simply have $\mathbf{z} = \mathrm{vec}(\mathbf{V}^h(1))$. The dependence of $\mathbf{B}^h(d-1)$ on $(\mathbf{z})$ follows from (35). We do not make any assumptions on how $\mathbf{C}(d)$ depends on $\mathbf{z}$.

We now establish generic uniqueness of the CPD in (34) for d down to $d = d^*$. The theorem involves a bound on m that is little restrictive, as we will clarify in section 9.

Theorem 7.2. *Let $\mathcal{Y} \in \mathbb{C}^{(n+1)\times q(d-1)\times m}$ admit a PD of the form (34); then generically $r_{\mathcal{Y}} = m$ and the CPD unique if*

$$d \geq d^* \quad \text{and} \quad m \leq m_{\max}(d) \stackrel{\text{def}}{=} \binom{n+d}{n} - n - 1. \tag{41}$$

Proof. To show the sufficiency of (41), we resort to an algebraic geometry-based tool for checking generic uniqueness of structured matrix factorizations of the form $\mathbf{Y}(\mathbf{z}) = \mathbf{M}(\mathbf{z})\mathbf{C}(\mathbf{z})^T$, in which the entries of $\mathbf{M}(\mathbf{z})$ can be parametrized by rational functions of $\mathbf{z}$; see [11, Theorem 1].

From (34), the parameters are taken equal to $\mathbf{z} = (x_0^{(1)} \ \dots \ x_n^{(m)})^T$. On the other hand, the entries of $\mathbf{M}(\mathbf{z}) = \mathbf{V}^h(1) \odot \mathbf{B}^h(d-1)$ take the form $\prod_{j=0}^n x_j^{(k)\alpha_j}$. The latter are monomials and thus rational functions of $\mathbf{z}$. [11, Theorem 1] states that the structured matrix factorization is generically unique if the number of rank-1 terms m is bounded by $m \leq \hat{N} - \hat{l}$, where the meaning of $\hat{N}$ and $\hat{l}$ will be clarified below.

- $\hat{N}$ is a lower bound on the dimension of the vector space spanned by arbitrary column vectors of $\mathbf{M}(\mathbf{z})$, i.e., by arbitrary vectors of the form $\mathbf{v}^h(1) \otimes \mathbf{b}^h(d-1)$. The distinct entries in $\mathbf{v}^h(1) \otimes \mathbf{b}^h(d-1)$ are the same as the distinct entries in $\underbrace{\mathbf{v}^h(1) \otimes \ldots \otimes \mathbf{v}^h(1)}_{d \text{ times}}$, which in turn are the entries in $\mathbf{v}^h(d)$, so $\hat{N} \leq q(d)$.

  We will show that $\hat{N} = q(d)$. Let

$$x_0^{(k)} = 1 \quad \text{and} \quad x_j^{(k)} = e^{2\pi \cdot i \cdot \frac{k-1}{q(d)} \cdot \left(\sum_{l=0}^{j-1} d^l\right)}, \quad k = 1 : q(d). \tag{42}$$

[11]The restriction to Ω is necessary, since not every choice of m points in $\mathbb{C}^{n+1}$ devises the solution set of a system of n polynomial equations of degree d_0 if $d_0 < m$ [17].

Then $\underbrace{\mathbf{V}^h(1) \odot \ldots \odot \mathbf{V}^h(1)}_{d \text{ times}} \in \mathbb{C}^{(n+1)^d \times q(d)}$ and
$\mathbf{V}^h(1) \odot \mathbf{B}^h(d-1) \in \mathbb{C}^{((n+1)\cdot q(d-1)) \times q(d)}$ contain $q(d)$ distinct rows of a Vandermonde matrix with the $q(d)$ different generators $x_1^{(k)}$ in (42), which span the entire $q(d)$-dimensional space [21, Proposition 4]. Hence, $\hat{N} = q(d)$.

- $\hat{l}$ is an upper bound on the number of parameters needed to parametrize a vector $\mathbf{v}^h(1)_k \otimes \mathbf{b}_k^h(d-1)$, so $\hat{l} = n+1$ is equal to the number of components $\{x_j^{(k)}\}_{j=0}^n$. ∎

In the proof of Theorem 7.2 the use of [11, Theorem 1] leads only to (41) because (33) exploits the multiplicative shift structure contained in *all* modes of (25).[12] In other words, we owe the bound to the simultaneous exploitation of the shift structure in all modes. NPA does not allow such a result, as it essentially exploits only one shift structure.[13]

The conditions in Theorem 7.2 do not guarantee that two factor matrices have full column rank, i.e., the CPD of $\mathcal{Y}$ does not necessarily satisfy the conditions in Theorem 3.1. On the other hand, the conditions in Theorem 7.2 do guarantee that the conditions in Theorem 3.2 are generically satisfied. (In the discussion of CPD uniqueness in [26, section IV], this corresponds to the fact that [26, Theorem 5] implies [26, Theorem 6].) We conclude from subsection 3.1 that, under the generic conditions in Theorem 7.2, the CPD of $\mathcal{Y}$ is not only unique; via an overdetermined set of linear equations it can be reworked into an auxiliary CPD that does satisfy the conditions in Theorem 3.1, and the latter can be reduced to a matrix GEVD. In a particular (nongeneric) case, the conditions in Theorem 3.2 may be verified for $\mathbf{A} = \mathbf{V}^h(1)$ and $\mathbf{B} = \mathbf{B}^h(d-1)$.

8. Algorithm.

The goal of this section is to put the theoretical insights from the previous sections to the fore.

Algorithm 1 CPD for multivariate polynomial root-finding

Input: A system $f_i \in \mathcal{C}_{d_i}^n$, $i = 1:n$, in the $n+1$ projective unknowns $x_j \in \mathbb{C}$, $j = 0:n$, with $m_0 = m$ simple roots.
Output: $\{\mathbf{x}^{(k)}\}_{k=1}^m$

1: Choose $d \geq d_0 = \max_i d_i$.
2: Construct $\mathbf{M}(d)$.
3: $\mathbf{K}(d) \leftarrow \operatorname{null}(\mathbf{M}(d))$.
4: Build $\mathcal{Y}$ slicewise by row selection $\mathcal{Y}(j+1,:,:) \leftarrow \overline{\mathbf{S}}^{(j)}(d-1) \cdot \mathbf{K}(d)$, $j = 0:n$.
5: Compute the SVD $\mathbf{Y}_{[2;1,3]} = \mathbf{U}^{(2)} \cdot \mathbf{S}^{(2)} \cdot \mathbf{U}^{(1,3)H}$.
6: Orthogonal compression: $\mathcal{Y}_c \leftarrow \mathcal{Y} \cdot_2 \mathbf{U}^{(2)H}$.
7: Compute the CPD $\mathcal{Y}_c = [\![\mathbf{A}, \mathbf{B}_c(d-1), \mathbf{C}(d)]\!]$.
8: Columnwise scaling: $\mathbf{X} \leftarrow \sim \mathbf{A}$.
9: **return X**

[12] The same argument actually proves [28, equation (26)] for MHR. The bound for R there and the bound for m here are very similar. Only $q(d)$ needs to be replaced by $\prod_{j=1}^n I_j = (d+1)^n$: $q(d)$ is exactly the number of rows that is selected by $\mathbf{S}_{(d+1)^n \to q(d)} \in \mathbb{C}^{q(d)\times(d+1)^n}$ in (26) or the number of rows left when going from Figure 2(a) to 2(b).

[13] Similarly, MHR approaches that exploit the shift invariance in only one mode do not reach the bound in [28, equation (26)].

Algorithm 1 summarizes the polynomial root-finding procedure implied by the derivation in the previous sections. Although the sequence of steps matches the derivation closely, the comments below are in order.

Step 1. d_0 is the minimum value needed to construct $\mathbf{M}(d)$, according to Definition 3.3. If further one takes $d \geq d^*$, Algorithm 1 can determine the roots of a generic system (section 6), and if one takes $d \geq d^* + 1$, the roots will be found in all cases (section 7).

Steps 2–3. The Macaulay matrix $\mathbf{M}(d)$ quickly becomes large, while on the other hand it is sparse [12]. Instead of constructing $\mathbf{M}(d)$ explicitly and calculating $\mathbf{K}(d)$ using dense linear algebra tools, e.g., the SVD-based `null` command in MATLAB, one may resort to numerical algorithms for sparse matrices, such as the sparse QR algorithm in [6]. An alternative is to not construct $\mathbf{M}(d)$ explicitly: [1, Algorithm 4.2] is a recursive orthogonalization scheme that exploits the sparsity properties of $\mathbf{M}(d)$ to obtain $\mathbf{K}(d)$ via updating.

Steps 5–6. The matrix $\mathbf{B}(d-1)$ quickly becomes very tall: the number of columns m is fixed, while the number of rows grows as $q(d-1) \approx \frac{1}{n!}(d-1)^n \gg m$. To reduce the cost of the computation in step 7, we may replace $\mathcal{Y}$ by an orthogonally compressed variant $\mathcal{Y}_c = \mathcal{Y} \cdot_2 \mathbf{U}_c^{(2)H}$. This compression is lossless iff $\mathrm{col}(\mathbf{Y}_{[2;1,3]}) \subseteq \mathrm{col}(\mathbf{U}^{(2)})$. A numerical basis of minimal size m is given by the m dominant left singular vectors of $\mathbf{Y}_{[2;1,3]}$, i.e., we can take $\mathbf{U}^{(2)} \in \mathbb{C}^{q(d-1)\times m}$ equal to the matrix of left singular vectors in the "economic size" SVD of $\mathbf{Y}_{[2;1,3]}$. Such a dimensionality reduction is a common preprocessing step in tensor computations [8, 26, 4]. Note that $(\mathbf{Y}_c)_{[2;1,3]} = \mathbf{S}^{(2)} \cdot \mathbf{U}^{(1,3)H}$, i.e., $\mathcal{Y}_c \in \mathbb{C}^{(n+1)\times m\times m}$ can be obtained by tensorizing the matrix $\mathbf{S}^{(2)} \cdot \mathbf{U}^{(1,3)H}$.

Step 7. The core of Algorithm 1 is the computation of the CPD of $\mathcal{Y}_c$. If $d \geq d^* + 1$, the CPD of $\mathcal{Y}_c$ can directly be found from a matrix GEVD (section 6 and Theorem 3.1). If $d = d^*$ and the conditions in Theorem 3.2 are satisfied (which is generically the case for $d = d^*$), an auxiliary CPD is derived first. The factor matrices of the auxiliary CPD can then be found from a matrix GEVD (section 7). The procedure is detailed in [7].

Approximate roots of a noisy polynomial system may be estimated by means of numerical optimization-based CPD algorithms such as nonlinear least squares (NLS) [27]. GEVD may provide a starting value for the optimization. In optimization algorithms prior knowledge about the roots (e.g., nonnegativity) can be imposed as constraints on $\mathbf{A}$ and/or $\mathbf{B}(d-1)$ [26]. The compression in steps 5–6 allows a reduction of the computational cost of the numerical optimization, also in constrained cases [39]. For a further discussion of CPD algorithms we refer to [39, 26] and references therein.

Step 8. As is clear from both (31) and (33), the m simple roots of the polynomial system appear in the first factor matrix. To distinguish between affine roots and roots at infinity, we normalize each column $\mathbf{x}^{(k)}$ to its affine counterpart ($x_0^{(k)} = 1$) iff $x_0^{(k)} \geq \tau \|\mathbf{x}^{(k)}\|$, given some tolerance τ. Eventually we obtain $\mathbf{X} = \begin{pmatrix} \mathbf{x}^{(1)} & \dots & \mathbf{x}^{(m)} \end{pmatrix} \in \mathbb{C}^{(n+1)\times m}$.

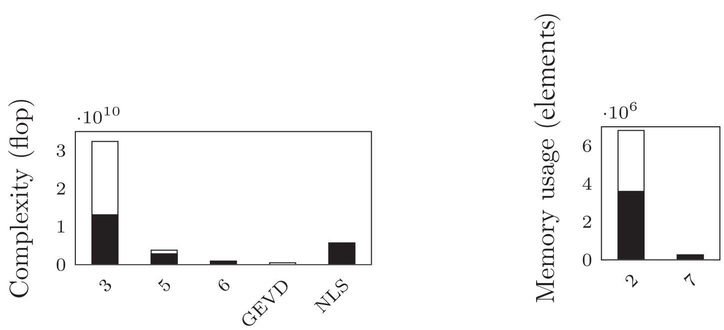
Table 1 gives an overview of the computational cost of the different steps of Algorithm 1. The derivation is given in Appendix A. Figure 4 shows a concrete example. We note the following:

- Although in general tensor problems suffer from the curse of dimensionality, this needs to be interpreted with some care. When solving sets of polynomial equations, the curse of dimensionality does not reside in the computation of the third-order CPD but in the size of $\mathbf{M}(d)$, which is the same for all Macaulay matrix based methods. Not step 7 but steps 3 and 2 are the bottleneck in Figure 4(a) and 4(b), respectively.

Table 1

Complexity and memory usage of Algorithm 1. *The Macaulay matrix* $\mathbf{M}(d) \in \mathbb{C}^{p\times q(d)}$ *and* $\overline{q} = q(d-1)$. *The cost in step* 3 *is given for the computation of* $\mathbf{K}(d)$ *by the SVD-based* `null` *command in MATLAB. In step* 7, *"it" denotes the number of iterations of the NLS algorithm.*

Step		Complexity (flop)	Memory usage (elements)
2			pq
3		$\mathcal{O}\left(2pq^2\right)$	
5		$\mathcal{O}\left(2\overline{q}n^2m^2\right)$	
6		$\mathcal{O}\left(2\overline{q}nm^2\right)$	
7	GEVD	$\mathcal{O}\left(30m^3\right)$	nm^2
	NLS	$\mathcal{O}\left(\text{it}\left(6n+\mathcal{O}\left(10^2\right)\right)m^3\right)$	



(a) (b)

Fig. 4. *Illustration of* (a) *computational complexity and* (b) *memory requirements of the different steps of Algorithm* 1, *detailed in Table* 1. *For the example we take* $n = 4$, $d_i = d_0 = 4, i = 1:4$. *We consider both* $d = d^* = 12$ *(filled) and* $d = d^*+1 = 13$ *(white). We set* it $= 10$, *as this is usually sufficient.*

- The possibility in our approach to take $d = d^* < d^*+1$, and hence to work with a smaller Macaulay matrix, conveys a far from marginal improvement of the bottleneck. The gain in steps 3 and 2 compensates the higher cost of the NLS algorithm that replaces the GEVD in step 7.

9. Experimental results. This section contains the results of some numerical experiments that illustrate the potential of our approach.

9.1. Uniqueness. Theorem 7.2 states that the CPD in step 7 in Algorithm 1 is generically unique if one takes $d \geq d^*$ and if $m \leq m_{\max}(d)$. Turned the other way around, Algorithm 1 will generically find the polynomial roots if $d \geq d^*$ and $m \leq m_{\max}(d)$. Table 2 shows the degree of regularity d^*, the Bézout number m, and $m_{\max}(d)$ for systems of n multivariate polynomial equations of degree d_0 in n affine variables, for various combinations of n and d_0. The table indicates that the condition $m \leq m_{\max}(d)$ is little restrictive at the minimally necessary degree $d = d^*$. Only for bivariate quadratic systems it is not satisfied ($n = d_0 = 2$). Moreover, the gap between m and $m_{\max}(d)$ increases with n and d_0.

These findings are confirmed by numerical experiments. By way of example, Figure 5 shows histograms over 200 Monte Carlo simulations of the relative forward error

$$\epsilon_{\hat{\mathbf{X}}} = \frac{\|\hat{\mathbf{X}} - \mathbf{X}\|}{\|\mathbf{X}\|} \tag{43}$$

Table 2

Values of $d^* = \sum_{i=1}^n d_i - n = n \cdot (d_0 - 1)$, $m = \prod_{i=1}^n d_i = d_0^n$, *and* $m_{\max}(d^*) = \binom{n+d^*}{n} - n - 1$ *for systems of polynomial equations in* n *affine variables with* $d_i = d_0, i = 1 : n$. *Only for* $n = d_0 = 2$ *we have* $m > m_{\max}(d^*)$ *(underlined).*

d_0	2			3			4		
n	d^*	m	$m_{\max}(d)$	d^*	m	$m_{\max}(d)$	d^*	m	$m_{\max}(d)$
2	$\underline{2}$	$\underline{4}$	$\underline{3}$	4	9	12	6	16	25
3	3	8	16	6	27	80	9	64	216
4	4	16	65	8	81	480	12	256	1815

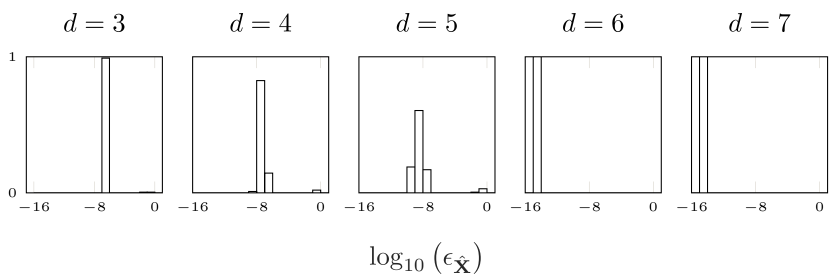


Fig. 5. *Histogram over* 200 *trials of the relative forward error* $\epsilon_{\hat{\mathbf{X}}}$ *on the estimated roots of a generic system of polynomial equations with* $n = 3$, $d_0 = d_i = 3$, $i = 1 : n$, *for which* $d^* = 6$. *The CPD in step* 7 *of Algorithm* 1 *was computed in all cases by the SD algorithm underlying Theorem* 3.2.

on the estimated solution $\hat{\mathbf{X}}$ of random polynomial systems with $n = 3$ and $d_0 = 3$. The systems are generic in the sense that all their coefficients have been drawn independently from the standard Gaussian distribution with mean 0 and standard deviation 1. The CPD in step 7 of Algorithm 1 is computed by the algorithm in [7], which we denote as "SD". For this we used the `cpd3_sd` function of Tensorlab [40]. For the CPD of the auxiliary tensor, we used the extended QZ iteration in [35, 9]. The reference solution $\mathbf{X}$ in (43) is obtained with the general purpose homotopy continuation-based solver from PHCpack [38]. In Figure 5, we let d vary between $d_0 = 3$ and $d^* + 1 = 7$. We observe the following:

- $d \geq d^*$ is indeed necessary and generically sufficient to retrieve the correct roots up to machine precision.
- Remarkably, even if $d < d^*$, i.e., if $r_{\mathbf{K}(d)} = \nu < m$, the SD algorithm retrieved most roots with a reasonable accuracy (about half the machine precision). The formal justification of this requires further study.
- Recall that GEVD and PNLA can only be used from $d \geq d^* + 1$ onward; under this condition they retrieved all roots correctly.

9.2. An overconstrained system of polynomial equations. We consider an overconstrained polynomial system, consisting of N noisy specifications (with limited precision) of the same underlying square ($s = n$) polynomial system [12, Chapter 8]. Such an overconstrained system may result from N measurements in the presence of noise. Applications may be found in, e.g., chemistry, kinematics, and computer vision. The overconstrained system has more equations ($s = Nn$) than unknowns (n). Typically there is no exact solution, which makes the problem unsuitable for the symbolic manipulations in computer algebra. However, Algorithm 1 can be used with slight modifications. First, note that for $s = Nn$, the Bézout number $m = \prod_{i=1}^n d_i$

and the degree of regularity $d^* = \sum_{i=1}^{n} d_i - n$ are the same as for $s = n$, since the degrees d_i have not changed. Step 3 in Algorithm 1 requires some attention, because the null space of $\mathbf{M}(d)$ of the overconstrained system has typically dimension 0. Instead, we could fill $\mathbf{K}(d)$ with the right singular vectors that correspond to the m smallest singular values of $\mathbf{M}(d)$. PNLA [12, Algorithm 6] proposes the same modification. The matrix $\mathbf{Y}_{[2;1,3]}$ in step 5 may not be exactly rank-m, so a best rank-m approximation is in order here. The CPD in step 7 is not exact either. It may still be estimated by a numerical optimization algorithm, and the latter may be initialized by GEVD or SD, as explained in subsection 3.1.

In an experiment, consider the underlying system [12, Example 8.3]

$$\left\{ \begin{array}{l} f_1(x_1, x_2) = x_1^3 + x_2^3 - 9x_1^2 x_2 + 20x_1x_2 - 3x_1 - 20 = 0, \\ f_2(x_1, x_2) = x_1^2 + 4x_2^2 - x_1x_2 - 80 = 0, \end{array} \right. \tag{44}$$

where $s = n = 2$ so that $m = 6$. Zero-mean Gaussian noise $\mathbf{e}_i^T$ is added to the $n = 2$ coefficient vectors $\mathbf{f}_i^T$ in (44), and the variance chosen such that

$$10 \log_{10} \left(\frac{\|\mathbf{f}_i\|^2}{\|\mathbf{e}_i\|^2} \right) \tag{45}$$

is equal to a preset SNR. We repeat this N times and collect the Nn noisy coefficient vectors in an overconstrained system. Figure 6 shows the median approximation error $\epsilon_{\hat{\mathbf{X}}}$ over 200 Monte Carlo trials for varying SNR and $N \in \{1, 2, 5, 10\}$. We make use of the compression in step 6 of Algorithm 1. PHCpack does not provide a solver for overconstrained systems; for reference we report the error that is obtained by PHCpack for a square noisy system. The figure indicates the following:

1. If $N = 1$, all algorithms "see" the square noisy system as if it was a different but exact system. They all return the same roots and show the same asymptotic performance as the SNR increases.[14]
2. As N increases, the overconstrained system provides more information than the square system, and the Macaulay matrix-based algorithms become more accurate than PHCpack.
3. At low SNR, the SD variant of Algorithm 1 is clearly the most accurate algorithm, because it takes the multiplicative shift structure in *all* variables into account.
4. The higher accuracy of the GEVD variant of Algorithm 1 compared to (GEVD-based) PNLA can be explained by the denoising effect of the orthogonal compression in steps 5 and 6. Indeed, recall that step 5 involves a truncation, i.e., the smallest "noise" singular values are discarded.

The standard deviations of the relative errors ϵ in the 200 trial runs were similar for all used methods: starting from about 0.09 for SNR=0 down to $6 \cdot 10^{-4}$ for SNR=60. Using an NLS-type algorithm, we obtained the same results as with SD if a good initial value was provided. Because of their expensive first step, the Macaulay resultant-based methods were roughly 10 times slower than PHCpack on a 16 GB RAM Intel Core i7-5500U CPU server. Recall from the discussion in section 8 that various speed-ups are possible.

[14]The asymptotic performance depends on the condition of the roots. The asymptotic performance shown in Figure 6 is representative for a large number of relatively well-conditioned polynomial root-finding problems with $n = 2$ and $n = 3$.

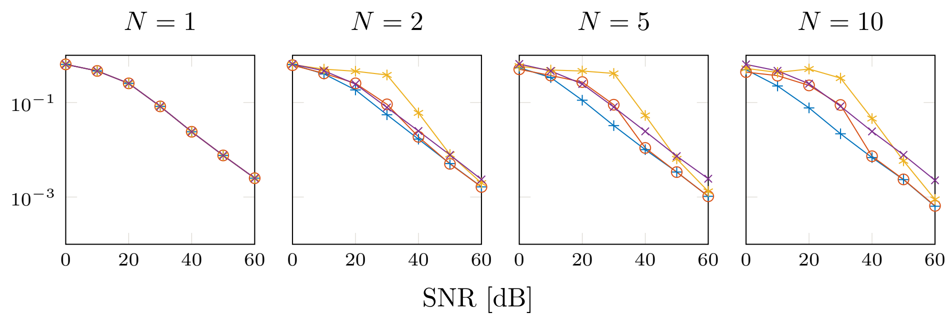


FIG. 6. *Relative forward error $\epsilon_{\hat{\mathbf{x}}}$ on the estimated roots of the overconstrained system of noisy polynomial equations derived from (44) for varying N. The median over 200 trials is plotted as a function of SNR. The results are shown for Algorithm 1, using a GEVD (—○—) or SD (—+—) in step 7, and PNLA (—∗—). We also show the PHCpack results for a square subsystem (i.e., $N = 1$) (—×—).*

10. Conclusions. As a thought-provoking implication of the central theorem of NPA, it has been stated that "The numerical solution of 0-dimensional systems of polynomial equations is a task of numerical linear algebra" [30, page 52]. From a particular point of view this statement is correct. Nevertheless, in this paper we have shown that, in line with what one would expect, the problem is rightfully qualified as a task of numerical multilinear algebra. Under certain working assumptions, linear algebra yields the exact solution of the exact equations. However, it exploits the available structure only partially. Technically, the CPD of a multislice tensor is collapsed in the GEVD of a pencil that captures the structure in only two of the slices. The significance of the multilinear perspective becomes clear when the working assumptions are relaxed and/or when the equations are inexact and only approximately satisfied.

Combining different higher-order tensor decompositions, each one exploiting the multiplicative structure in just one of the unknowns, we have eventually obtained the CPD in (34). This is arguably our central decomposition: it improves upon a "flat" matrix model, it can be linked to the joint (G)EVD of NPA's multiplication tables, and it does not distinguish between affine and projective roots. We have also illustrated some of the potential of Algorithm 1, which follows naturally from the derivation. The accuracy of the algorithm is as good as PHC, it allows the use of Macaulay matrices of degree $d = d^*$ instead of $d \geqslant d^* + 1$, and it can handle overconstrained systems. Like in "linear" Macaulay resultant based algorithms, the size of the Macaulay matrix is the computational bottleneck. Therefore, a clear need for fast, e.g., matrix-free, algorithms that fully exploit the sparsity of the Macaulay matrix, remains. The companion paper [36] will drop the constraint of only simple roots and will relate the topics of our study to a more general third-order block-term decomposition. The recent work in [33, 31, 32] opens an interesting perspective on a further extension to sparse sets of polynomial equations, the polyhedral structure of which results in smaller matrices.

Appendix A. Computational complexity of Algorithm 1. The memory usage in number of elements stored should be self-explanatory. Here, we derive the computational complexity in flop. The operation count of the SVD of an $I_1 \times I_2$ matrix with $I_1 > I_2$ is approximately $\mathcal{O}\left(2I_1I_2^2\right)$ [34, page 238]. To compute the CPD of an $I_1 \times I_2 \times I_3$ third-order tensor $\mathcal{T}$ by means of a GEVD, it is assumed that a QZ algorithm is used, which requires $\mathcal{O}\left(30I^3\right)$ flop for square $I \times I$ pencils [16]. The computation of an R-term CPD of $\mathcal{T}$ by means of the (inexact) Gauss–Newton algorithm with dogleg trust region costs $\mathcal{O}\left(2(3+\mathrm{it_{tr}})R\prod I_n + \mathrm{it_{cg}}(\frac{45}{2}R^2 + R^3 + 8R^2\sum_n I_n)\right)$ flop per iteration. Within each iteration the dogleg trust region step requires $\mathrm{it_{tr}}$ iterations, and $\mathrm{it_{cg}}$ conjugate gradient iterations are required to solve the linear system to a prescribed accuracy [27, page 708]. The cost of the steps in Algorithm 1 is then given by

$$\begin{aligned}
&\mathcal{O}\left(2pq^2\right) && \text{(SVD-based \texttt{null} of } \mathbf{M}(d)\text{)},\\
&\mathcal{O}\left(2\overline{q}(nm)^2\right) && \text{(SVD of } \mathbf{Y}_{[2;1,3]}\text{)},\\
&\mathcal{O}\left(nm^2 2\overline{q}\right) && \text{(matrix product)},\\
&\mathcal{O}\left(30m^3\right) && \text{(CPD by means of a GEVD)},\\
&\mathcal{O}\left(\mathrm{it}\left(2(3+\mathrm{it_{gn}})mnm^2 + \mathrm{it_{cg}}\left(\tfrac{45}{2}m^2 + m^3 + 8m^2 2m\right)\right)\right) && \\
&\quad = \mathcal{O}\left(\mathrm{it}\left(8nm^3 + \mathcal{O}\left(10^2\right)m^3\right)\right) && \text{(CPD by means of NLS)},
\end{aligned}$$

where $q = q(d)$, $\overline{q} = q(d-1)$, and "it" denotes the number of iterations of the Gauss–Newton algorithm. The final estimate is based on the experience that typically fewer than 10 conjugate gradient iterations and only one trust region iteration are needed.

REFERENCES

[1] K. Batselier, *A Numerical Linear Algebra Framework for Solving Problems with Multivariate Polynomials*, Ph.D thesis, KU Leuven — ESAT/STADIUS, 2013.

[2] C. Beltrán, P. Breiding, and N. Vannieuwenhoven, *Pencil-based algorithms for tensor rank decomposition are not stable*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 739–773.

[3] B. Buchberger, *Gröbner bases: A short introduction for systems theorists*, in Computer Aided Systems Theory, Lecture Notes in Comput. Sci. 2178, Springer, Berlin, 2001, pp. 1–19.

[4] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, *Tensor decompositions for signal processing applications: From two-way to multiway component analysis*, IEEE Signal Process. Mag., 32 (2015), pp. 145–163.

[5] D. Cox, J. Little, and D. O'shea, *Using Algebraic Geometry*, Springer-Verlag, Berlin, 1998, https://doi.org/10.1007/b138611.

[6] T. Davis, *Algorithm* 915*, SuiteSparseQR: Multifrontal multithreaded rank-revealing sparse QR factorization*, ACM Trans. Math. Software, 38 (2011), pp. 8:1–8:22, https://doi.org/10.1145/2049662.2049670.

[7] L. De Lathauwer, *A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 642–666, https://doi.org/10.1137/040608830.

[8] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278, https://doi.org/10.1137/S0895479896305696.

[9] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 295–327, https://doi.org/10.1137/S089547980139786X.

[10] I. Domanov and L. De Lathauwer, *On the uniqueness of the canonical polyadic decomposition of third-order tensors—Part* I*: Basic results and uniqueness of one factor matrix*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 855–875.

[11] I. Domanov and L. De Lathauwer, *Generic uniqueness of a structured matrix factorization and applications in blind source separation*, IEEE J. Sel. Top. Signal Process., 10 (2016), pp. 701–711, https://doi.org/10.1109/JSTSP.2016.2526971.

[12] P. Dreesen, *Back to the Roots: Polynomial System Solving Using Linear Algebra*, Ph.D thesis, KU Leuven — ESAT/STADIUS, 2013.

[13] P. Dreesen, K. Batselier, and B. De Moor, *Back to the roots: Polynomial system solving, linear algebra, systems theory*, IFAC Proc., 45 (2012), pp. 1203–1208, https://doi.org/10.3182/20120711-3-BE-2027.00217.

[14] P. Dreesen, K. Batselier, and B. De Moor, *On the null spaces of the Macaulay matrix*, Linear Algebra Appl., 460 (2014), pp. 259–289, https://doi.org/10.1016/j.laa.2014.07.035.

[15] P. Dreesen, K. Batselier, and B. De Moor, *On Polynomial System Solving and Multidimensional Realization Theory*, Tech. Report 15-145, KU Leuven — ESAT/STADIUS, 2015.

[16] G. Golub and C. Van Loan, *Matrix Computations*, 4th ed., John Hopkins University Press, Baltimore, MD, 2013.

[17] J. Harris, *Algebraic Geometry: A First Course*, Springer-Verlag, Berlin, 1998, https://doi.org/10.1007/978-1-4757-2189-8.

[18] C. J. Hillar and L.-H. Lim, *Most tensor problems are NP-hard*, J. ACM, 60 (2013), p. 45.

[19] F. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys., 6 (1927), pp. 164–189, https://doi.org/10.1002/sapm192761164.

[20] T. Jiang and N. Sidiropoulos, *Kruskal's permutation lemma and the identification of CANDECOMP/PARAFAC and bilinear models with constant modulus constraints*, IEEE Trans. Signal Process., 52 (2004), pp. 2625–2636.

[21] T. Jiang, N. Sidiropoulos, and J. ten Berge, *Almost-sure identifiability of multidimensional harmonic retrieval*, IEEE Trans. Signal Process., 49 (2001), pp. 1849–1859, https://doi.org/10.1109/78.942615.

[22] G. Jónsson and S. Vavasis, *Accurate solution of polynomial equations using Macaulay resultant matrices*, Math. Comp., 74 (2004), pp. 221–262, https://doi.org/10.1090/S0025-5718-04-01722-3.

[23] J. Landsberg, *Tensors: Geometry and Applications*, American Mathematical Society, Providence, RI, 2012, https://doi.org/10.1090/gsm/128.

[24] S. E. Leurgans, R. T. Ross, and R. B. Abel, *A decomposition for three-way arrays*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1064–1083, http://dx.doi.org/10.1137/0614071.

[25] R. Roy and T. Kailath, *ESPRIT — estimation of signal parameters via rotational invariance techniques*, IEEE Trans. Acoust. Speech Signal Process., 37 (1989), pp. 984–995, https://doi.org/10.1109/29.32276.

[26] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, *Tensor decomposition for signal processing and machine learning*, IEEE Trans. Signal Process., 65 (2017), pp. 3551–3582.

[27] L. Sorber, M. Van Barel, and L. De Lathauwer, *Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(L_r, L_r, 1)$ terms, and a new generalization*, SIAM J. Optim., 23 (2013), pp. 695–720, https://doi.org/10.1137/120868323.

[28] M. Sørensen and L. De Lathauwer, *Multidimensional harmonic retrieval via coupled canonical polyadic decomposition – Part* I*: Model and identifiability*, IEEE Trans. Signal Process., 65 (2017), pp. 517–527, https://doi.org/10.1109/TSP.2016.2614796.

[29] M. Sørensen and L. De Lathauwer, *Multidimensional harmonic retrieval via coupled canonical polyadic decomposition – Part* II*: Algorithm and multirate sampling*, IEEE Trans. Signal Process., 65 (2017), pp. 528–539, https://doi.org/10.1109/TSP.2017.2706183.

[30] H. Stetter, *Numerical Polynomial Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2004, https://doi.org/10.1137/1.9780898717976.

[31] S. Telen, *Numerical root finding via Cox rings*, J. Pure Appl. Algebra, 224 (2020), 106367, https://doi.org/10.1016/j.jpaa.2020.106367.

[32] S. Telen, *Solving Systems of Polynomial Equations*, PhD thesis, KU Leuven - Faculty of Engineering Science, 2020.

[33] S. Telen, B. Mourrain, and M. V. Barel, *Solving polynomial systems via truncated normal forms*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1421–1447, https://doi.org/10.1137/17M1162433.

[34] L. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.

[35] A.-J. Van Der Veen and A. Paulraj, *An analytical constant modulus algorithm*, IEEE Trans. Signal Process., 44 (1996), pp. 1136–1155.

[36] J. Vanderstukken, A. Stegeman, and L. De Lathauwer, *Systems of Polynomial Equations, Higher-Order Tensor Decompositions and Multidimensional Harmonic Retrieval: A Unifying Framework – Part* II: *The Block-Term Decomposition*, Tech. Report 17-134, KU Leuven — ESAT/STADIUS, 2017.

[37] J. Verschelde, *Homotopy Continuation Methods for Solving Polynomial Systems*, Ph.D thesis, KU Leuven — Faculty of Engineering Science, 1996.

[38] J. Verschelde, *Algorithm* 795*, PHCpack: A general-purpose solver for polynomial systems by homotopy continuation*, ACM Trans. Math. Software, 25 (1999), pp. 251–276, https://doi.org/10.1145/317275.317286.

[39] N. Vervliet and L. De Lathauwer, *Numerical optimization based algorithms for data fusion*, in Data Fusion Methodology and Applications, M. Cocchi, ed., Elsevier, New York, 2019, pp. 81–128.

[40] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, *Tensorlab* 3.0, Mar. 2016, http://www.tensorlab.net. Available online.