Citation/Reference	Authors: Mariya Ishteva, Philippe Dreesen Title: Solving Systems of Polynomial Equations - a Tensor Approach Conference: Proc. of the 13th International Conference on Large-Scale Scientific Computations (LSSC 2021)
Dates	**Conference paper** • Date of conference: June 7–11, 2021 • Date of presentation: June 8, 2021
Archived version	☐ Submitted version ■ Accepted (Author manuscript: the content is identical to the content of the published paper, but without the final typesetting by the publisher) ☐ Published ☐ Published (Open Access)
Acknowledgements	In this manuscript, the following funding is acknowledged: ■ EU H22020 ■ EU Horizon Europe ■ FWO ■ KULeuven Internal Funds ☐ VLAIO ■ AI ■ Other
Published version	NA
Author contact	philippe.dreesen@gmail.com

Solving Systems of Polynomial Equations - a Tensor Approach ⋆

Mariya Ishteva[1][0000−0002−7951−536X] and
Philippe Dreesen[2][0000−0002−6272−2004]

[1] KU Leuven, Dept. Computer Science, ADVISE-NUMA, Belgium
[2] KU Leuven, Dept. Electrical Engineering, ESAT-STADIUS, Belgium
{mariya.ishteva,philippe.dreesen}@kuleuven.be

Abstract. Polynomial relations are at the heart of mathematics. The fundamental problem of solving polynomial equations shows up in a wide variety of (applied) mathematics, science and engineering problems. Although different approaches have been considered in the literature, the problem remains difficult and requires further study.

We propose a solution based on tensor techniques. In particular, we build a partially symmetric tensor from the coefficients of the polynomials and compute its canonical polyadic decomposition. Due to the partial symmetry, a structured canonical polyadic decomposition is needed. The factors of the decomposition can then be used for building systems of linear equations, from which we find the solutions of the original system.

This paper introduces our approach and illustrates it with a detailed example. Although it cannot solve any system of polynomial equations, it is applicable to a large class of sub-problems. Future work includes comparisons with existing methods and extending the class of problems, for which the method can be applied.

Keywords: systems of polynomial equations · tensors · canonical polyadic decomposition · partial symmetry.

1 Introduction

Solving systems of multivariate polynomial equations is a fundamental problem in mathematics, having a multitude of scientific and engineering applications. This task typically involves square systems (as many equations as unknowns), which generically have a solution set consisting of isolated points. Computational methods for solving polynomial systems are largely dominated by the symbolic Groebner basis approach [6], although other approaches exist, such as homotopy continuation methods.

⋆ This research was supported by KU Leuven Research Fund; KU Leuven start-up-grant STG/19/036 ZKD7924; FWO (EOS Project 30468160, SBO project S005319N, Infrastructure project I013218N, TBM Project T001919N, Research projects G028015N, G090117N, PhD grants SB/1SA1319N, SB/1S93918, SB/151622); ERC AdG "Back to the Roots" (885682); Flemish Government (AI Research Program); PD is affiliated to Leuven.AI - KU Leuven institute for AI. Part of this work was done while the authors were with Vrije Universiteit Brussel (VUB-ELEC).

From the (numerical) linear algebra viewpoint, there is a less known connection between polynomial systems and eigenvalue problems. In the 1980s, the work of Stetter [16], among others, demonstrated that eigenvalue problems are at the core of polynomial systems. The eigenvalue problem formulation, returning all solutions of the system, can be obtained from a Groebner basis, or from resultant-based approaches [6]. Although the number of solutions grows quickly with system size and degree (it is equal to the product of the equations degrees), the computational bottleneck in these approaches is at the steps preceding the eigenvalue problem formulation, such as computing a Groebner basis, or manipulating large resultant-based matrices.

In this article, we look for the hidden eigenvalue problem in another way, by exploring the connection between polynomials and tensors (multiway arrays). In particular, we build a partially symmetric tensor from the coefficients of the polynomials and decompose this tensor with a canonical polyadic decomposition, which can often be reformulated as an eigenvalue problem [7]. Due to partial symmetry, a structured version of the canonical polyadic decomposition is needed. The factors of the decomposition can then be used for building systems of linear equations, from which we find the solutions of the polynomial system.

The remainder of this paper is organized as follows. Section 2 introduces the basic background material. Section 3 first presents the main idea in the case of bivariate polynomial equations of degree two and then discusses generalizations to more equations (more variables) and higher degree polynomials. Section 4 summarizes the main conclusions and discusses future work.

2 Background material

This section introduces our notation, the canonical polyadic decompositions of tensors, and the link between (symmetric) tensors and polynomials.

2.1 Notation

We use lowercase (a), boldface lowercase ($\mathbf{a}$), uppercase boldface ($\mathbf{A}$), and calligraphic font ($\mathcal{A}$) for scalars, vectors, matrices, and tensors (a dth-order tensor is a multiway array with d indices), respectively. The elements of the vectors, matrices, and tensors are accessed as a_i, A_{ij} and $\mathcal{A}_{i_1 \dots i_d}$, respectively.

$\mathbf{A}^\top$ denotes the transpose of the matrix $\mathbf{A}$ and the symbols $\circ$, $\otimes$ and $\odot$ stand for the outer, the Kronecker, and the Khatri-Rao product, respectively.

The elements of the n-mode product $\mathcal{A} \bullet_n \mathbf{x} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_d}$, where $1 \leq n \leq d$, of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_d}$ and a vector $\mathbf{x} \in \mathbb{R}^{I_n}$ are defined as

$$(\mathcal{A} \bullet_n \mathbf{x})_{i_1 \dots i_{n-1} i_{n+1} \dots i_d} = \sum_{i_n=1}^{I_n} \mathcal{A}_{i_1 \dots i_d} x_{i_n}.$$

Thus, if $\mathcal{A}$ is a third-order tensor, the products $\mathcal{A} \bullet_n \mathbf{x}$ are matrices whose elements are the scalar products of the vector $\mathbf{x}$ and the mode-n vectors of $\mathcal{A}$. The product $\mathcal{A} \bullet_1 \mathbf{x} \bullet_2 \mathbf{x}$ is a vector and the product $\mathcal{A} \bullet_1 \mathbf{x} \bullet_2 \mathbf{x} \bullet_3 \mathbf{x}$ is a scalar.

2.2 The canonical polyadic decomposition of tensors

A tensor $\mathcal{A}$ has rank equal to one, if it can be written as an outer product of vectors. For example, for a third-order tensor $\mathcal{A}$ of rank one, we have $\mathcal{A} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$, that is, $\mathcal{A}_{ijk} = a_i b_j c_k$. Every tensor can be expressed as a sum of rank-one tensors, i.e., for a third-order tensor $\mathcal{A}$ we have $\mathcal{A} = \sum_{i=1}^{r} \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i$. We denote this by $\mathcal{A} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, where $\mathbf{a}_i$, $\mathbf{b}_i$, and $\mathbf{c}_i$ are the r columns of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, respectively. If r is minimal, we call this the canonical polyadic decomposition (CPD) of $\mathcal{A}$. [10, 2] (Figure 1) and r is called the rank of tensor $\mathcal{A}$.

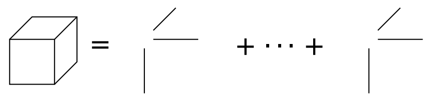

Fig. 1. The canonical polyadic decomposition decomposes a third-order tensor into a minimal sum of rank-one terms (outer products of vectors). The number of terms is called the rank of the tensor.

For symmetric tensors (tensors invariant under permutations of the indices), symmetric decompositions are considered, i.e., decompositions with identical factors ($\mathcal{A} = [\![\mathbf{A}, \mathbf{A}, \mathbf{A}]\!]$). We deal with partially symmetric tensors (symmetric with respect to some of the modes), so we seek a partially symmetric decomposition (where some of the factors are identical), for example $\mathcal{A} = [\![\mathbf{A}, \mathbf{A}, \mathbf{C}]\!]$.

Computing the CPD of a tensor is a difficult problem in general but has been studied extensively in the literature. This problem can often be reduced to an eigenvalue problem and our preference goes for this option [7]. In some cases, iterative algorithms are required, for which a number of implementations are available, for example Tensorlab [17]. For further details on tensor decompositions and their applications, we refer to the overview papers [3, 5, 12, 8, 11, 14], books [4, 9, 13, 15], and references therein.

2.3 A link between polynomials and symmetric tensors

We next discuss the link between polynomials and symmetric tensors, first in the case of polynomials of degree two and then for higher-degree polynomials.

Polynomials of degree two. Every polynomial of degree two can be associated with a symmetric coefficient matrix $\mathbf{C}$. A bivariate polynomial in x and y can be written as $\begin{bmatrix} x \; y \; 1 \end{bmatrix} \mathbf{C} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$. For example, consider the polynomials p and q,

$$\begin{aligned} p(x,y) &= -x^2 + 2xy + 8y^2 - 12x = \begin{bmatrix} x \; y \; 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & -6 \\ 1 & 8 & 0 \\ -6 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \\ q(x,y) &= 2x^2 + 8xy + \tfrac{7}{2}y^2 + 8x - 2y - 2 = \begin{bmatrix} x \; y \; 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 4 \\ 4 & \frac{7}{2} & -1 \\ 4 & -1 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \end{aligned} \tag{1}$$

We denote $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ by $\mathbf{u}$. In case of more variables, the length of $\mathbf{u}$ increases, but any polynomial can still be written as $\mathbf{u}^\top \mathbf{C}\mathbf{u}$, with some symmetric matrix $\mathbf{C}$.

Polynomials of higher degree. Every polynomial of higher degree can be associated with a higher-order tensor (instead of with a matrix). For example, a polynomial of degree three can be described by a third-order tensor $\mathcal{C}$:

$$\begin{aligned} &a_{111}x^3 + 3a_{112}x^2y + 3a_{122}xy^2 + a_{222}y^3 \\ &+ 3a_{110}x^2 + 6a_{120}xy + 3a_{220}y^2 + 3a_{100}x + 3a_{200}y + a_{000} \\ &= \mathcal{C} \bullet_1 \mathbf{u} \bullet_2 \mathbf{u} \bullet_3 \mathbf{u}, \end{aligned}$$

where $\mathcal{C} \in \mathbb{R}^{3\times3\times3}$ has the following frontal slices

$$\begin{bmatrix} a_{111} & a_{112} & a_{110} \\ a_{112} & a_{122} & a_{120} \\ a_{110} & a_{120} & a_{100} \end{bmatrix}, \quad \begin{bmatrix} a_{112} & a_{122} & a_{120} \\ a_{122} & a_{222} & a_{220} \\ a_{120} & a_{220} & a_{200} \end{bmatrix}, \quad \begin{bmatrix} a_{110} & a_{120} & a_{100} \\ a_{120} & a_{220} & a_{200} \\ a_{100} & a_{200} & a_{000} \end{bmatrix}.$$

In case of more variables, the length of $\mathbf{u}$ increases, but the polynomial can still be written as $\mathcal{C} \bullet_1 \mathbf{u} \bullet_2 \mathbf{u} \bullet_3 \mathbf{u}$, with some symmetric tensor $\mathcal{C}$.

3 Solving polynomial systems by tensor decompositions

This section first presents the main idea in the case of two polynomials (in two variables) of degree two and then discusses possible extensions to the cases of more equations (and variables) and higher degree polynomials.

3.1 Solving systems of two polynomial equations of degree two

For systems of two bivariate polynomial equations of degree two, our approach consist of four steps: building a partially symmetric tensor from the coefficients of the polynomials (Step 1), computing its partially symmetric CPD (Step 2) and using the factors of the decomposition to build systems of linear equations, from which we find the solutions (x, y) (Steps 3–4). We use the polynomials from (1) to illustrate the approach, see also Figure 2.

Step 1. We associate one matrix with each equation (as in (1)) and stack them behind each other in a partially symmetric third-order tensor $\mathcal{T} \in \mathbb{R}^{3\times3\times2}$. The system then becomes

$$\mathcal{T} \bullet_1 \mathbf{u} \bullet_2 \mathbf{u} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \text{with } \mathbf{u} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{2}$$

Step 2. We next decompose $\mathcal{T}$ in (partially symmetric) rank-one terms,

$$\mathcal{T} = [\![\mathbf{V}, \mathbf{V}, \mathbf{W}]\!],$$

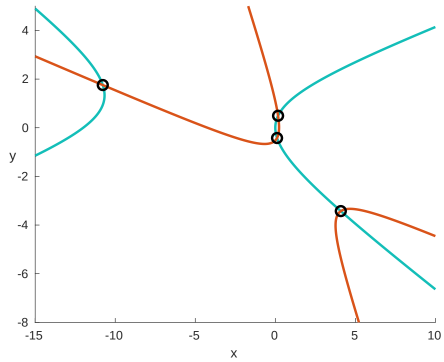

Fig. 2. The equations from the running example are visualized as the blue and red lines. The solutions of the system are the four points of intersection, circled in black.

with $\mathbf{V} \in \mathbb{R}^{3\times r}$ and $\mathbf{W} \in \mathbb{R}^{2\times r}$, where r is the rank of $\mathcal{T}$. The typical ranks of a $3 \times 3 \times 2$ tensor are three and four over $\mathbb{R}$. However, if we allow decompositions with complex numbers, the typical rank is only three. In the following, we thus consider the rank to be three and will work with complex numbers, if necessary.

For our example the rank over $\mathbb{R}$ is three. We obtain[1]

$$\mathbf{V} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & -1 & 2 \\ 2 & -2 & 0 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} -1 & 1 & 2 \\ \frac{1}{2} & -1 & 1 \end{bmatrix}.$$

The system of polynomial equations (2) can now be re-written as

$$\mathcal{T} \bullet_1 \mathbf{u} \bullet_2 \mathbf{u} = [\![\mathbf{V}, \mathbf{V}, \mathbf{W}]\!] \bullet_1 \mathbf{u} \bullet_2 \mathbf{u} = [\![\mathbf{u}^\top \mathbf{V}, \mathbf{u}^\top \mathbf{V}, \mathbf{W}]\!] = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{3}$$

In the following, we will first ignore the fact that the last element of $\mathbf{u}$ equals one. The system (3) then has an intrinsic scaling indeterminacy. We will resolve this issue by re-scaling $\mathbf{u}$ in the last step of the algorithm.

Step 3. Let $\mathbf{z} = \mathbf{V}^\top \mathbf{u}$. We re-write equation (3) and solve it for $\mathbf{z}$:

$$[\![\mathbf{u}^\top \mathbf{V}, \mathbf{u}^\top \mathbf{V}, \mathbf{W}]\!] = [\![\mathbf{z}^\top, \mathbf{z}^\top, \mathbf{W}]\!] = \mathbf{W}(\mathbf{z}^\top \odot \mathbf{z}^\top)^\top = \mathbf{W}\mathbf{z}.^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{4}$$

where the elements of $\mathbf{z}.^2$ are the squared elements of $\mathbf{z}$. This is a linear system of equations for $\mathbf{z}.^2$ (we disregard the scaling indeterminacy for now and will resolve it in Step 4.). $\mathbf{z}.^2$ leads to eight possible $\mathbf{z}$ but only four of them are essentially different because $\mathbf{z}$ and $-\mathbf{z}$ eventually produce the same (x, y), due to the scaling indeterminacy of $\mathbf{u}$.

[1] The CPD is invariant under some scaling and permutation of the columns of the factors. Here we have chosen to re-scale the solution obtained from the eigenvalue algorithm, in order to simplify the numbers in the example.

For our example,

$$\mathbf{z}.^2 = \begin{bmatrix} 0.8242 \\ 0.5494 \\ 0.1374 \end{bmatrix},$$

so the four (essentially different) solutions are

$$\mathbf{z}^{(1)} = \begin{bmatrix} 0.9078 \\ 0.7412 \\ 0.3706 \end{bmatrix}, \mathbf{z}^{(2)} = \begin{bmatrix} -0.9078 \\ 0.7412 \\ 0.3706 \end{bmatrix}, \mathbf{z}^{(3)} = \begin{bmatrix} 0.9078 \\ -0.7412 \\ 0.3706 \end{bmatrix}, \mathbf{z}^{(4)} = \begin{bmatrix} 0.9078 \\ 0.7412 \\ -0.3706 \end{bmatrix}. \quad (5)$$

Step 4. We find four solutions for $\mathbf{u}$ from $\mathbf{z}^\top = \mathbf{u}^\top \mathbf{V}$ by solving the four systems of linear equations (one for each $\mathbf{z}^{(i)}$ from (5))

$$\mathbf{V}^\top \mathbf{u} = \mathbf{z}.$$

Finally, we rescale each of the four solutions for $\mathbf{u}$, so that the last element becomes one. The first two elements are then x and y.

For our example we obtain

$$\mathbf{u}^{(1)} = \begin{bmatrix} 0.5497 \\ -0.0895 \\ -0.0510 \end{bmatrix} = -0.0510 \begin{bmatrix} -10.7766 \\ 1.7553 \\ 1 \end{bmatrix}, \mathbf{u}^{(2)} = \begin{bmatrix} -0.0555 \\ 0.2131 \\ -0.5049 \end{bmatrix} = -0.5049 \begin{bmatrix} 0.1100 \\ -0.4220 \\ 1 \end{bmatrix},$$

$$\mathbf{u}^{(3)} = \begin{bmatrix} 0.0555 \\ 0.1575 \\ 0.3196 \end{bmatrix} = 0.3196 \begin{bmatrix} 0.1737 \\ 0.4929 \\ 1 \end{bmatrix}, \mathbf{u}^{(4)} = \begin{bmatrix} 0.5497 \\ -0.4602 \\ 0.1343 \end{bmatrix} = 0.1343 \begin{bmatrix} 4.0929 \\ -3.4263 \\ 1 \end{bmatrix}$$

The solutions are

$$(x^{(1)}, y^{(1)}) = (-10.7766, 1.7553), \quad (x^{(2)}, y^{(2)}) = (0.1100, -0.4220),$$
$$(x^{(3)}, y^{(3)}) = (0.1737, 0.4929), \quad (x^{(4)}, y^{(4)}) = (4.0929, -3.4263).$$

The procedure is summarized as Algorithm 1.

Algorithm 1 Solving a polynomial system of equations via tensor decomposition

Input: A system of 2 polynomial equations of degree 2 (and 2 variables)
Output: The solutions $(x^{(i)}, y^{(i)}), i = 1, ..., 4$ of the system

1: Reformulate the problem as $\mathcal{T} \bullet_1 \mathbf{u} \bullet_2 \mathbf{u} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.
2: Decompose the tensor $\mathcal{T}$ in (partially symmetric) rank-one terms, $\mathcal{T} = [\![\mathbf{V}, \mathbf{V}, \mathbf{W}]\!]$.
3: Solve the linear system $\mathbf{W}(\mathbf{z}.^2) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ for $\mathbf{z}.^2$.
   Find the 4 (essentially different) solutions $\mathbf{z}^{(i)}$, $i = 1, \dots, 4$.
4: Solve the linear systems $\mathbf{V}^\top \mathbf{u}^{(i)} = \mathbf{z}^{(i)}$ for $\mathbf{u}^{(i)}$, $i = 1, \dots, 4$.
   Normalize $\mathbf{u}^{(i)}$, $i = 1, \dots, 4$ so that the last elements become 1.
   Extract the solutions $(x^{(i)}, y^{(i)})$, $i = 1, ..., 4$ by removing the last element of each $\mathbf{u}^{(i)}$.

Remark. In our example we have four real roots. It is also possible for two bivariate equations of degree two to have four complex roots or two real and two complex roots. The proposed algorithm can deal with these cases as well.

A polynomial system of equations can also have roots at infinity. These roots correspond to solutions of the homogenized version of the system, where a third variable is introduced that multiplies each monomial (zero, one or two times) to complete it to degree two. Solutions at infinity are the solutions, for which the additional variable is zero. Our algorithm can find such roots as well. In this case, the last entry of $\mathbf{u}$ becomes zero.

3.2 Systems with more variables or higher degree polynomials

We now briefly discuss two generalizations of the main problem (2): the cases of larger number of variables or higher degree polynomials.

More variables. In case of more variables, the length of $\mathbf{u}$ and the number of slices of the system's tensor will increase, but we could proceed in a similar way if the rank of the tensor is small enough. If the rank is very large (even if we allow complex factors), a modification of the main algorithm will be necessary. A possible direction to consider here would be to reformulate the problem as

$$\mathbf{W}(\mathbf{V} \odot \mathbf{V})^\top (\mathbf{u} \otimes \mathbf{u}) = \mathbf{0}$$

and solve for $\mathbf{u} \otimes \mathbf{u}$ as in [1]. This line of research is left for future work.

Polynomials of higher degree. In case of polynomials of higher degree, the associated tensors are of higher order. For example, if we have bivariate polynomials of degree three, we associate a third-order tensor (instead of a matrix) with each equation and stack these tensors in a fourth-order tensor $\mathcal{T} \in \mathbb{R}^{3\times3\times3\times2}$. The decomposition of $\mathcal{T}$ in rank-one terms contains then one additional factor $\mathbf{V}$,

$$\mathcal{T} = [\![\mathbf{V}, \mathbf{V}, \mathbf{V}, \mathbf{W}]\!].$$

We can proceed in a similar way as in Section 3.1, except that now in the first system we solve for $\mathbf{z}.^3$. Unfortunately, the the rank of $\mathcal{T}$ could increase as well.

4 Conclusions and Perspectives

We proposed a new procedure for solving bivariate polynomial equations, exclusively using tools from numerical (multi-)linear algebra, such as the eigenvalue decomposition and solving linear systems. Although our approach is currently not general enough for solving an arbitrary system of polynomials, it is applicable to a large class of sub-problems. The core computational steps of the procedure are i) a CPD of the coefficient tensor, and ii) solving linear systems involving the CPD factors. In many cases the CPD is known to be an eigenvalue problem in disguise. For this reason, our new approach is particularly interesting as said eigenvalue problem is phrased 'directly' in the equations' coefficients, as opposed to existing methods in which the eigenvalue formulation follows after several computationally intensive steps.

Future work will focus on comparing and establishing connections to existing eigenvalue-based approaches for polynomial system solving. We aim to generalize the method to deal with systems in more variables and of larger degrees, which involves higher-order coefficient tensors, instead of matrices. In this context it remains to be seen what is the (complex) rank of the resulting coefficient tensor, and how to generalize all the steps of the proposed algorithm.

References

1. Boussé, M., Vervliet, N., Domanov, I., Debals, O., De Lathauwer, L.: Linear systems with a canonical polyadic decomposition constrained solution: Algorithms and applications. Numerical Linear Algebra with Applications **25**(6), e2190 (2018)
2. Carroll, J., Chang, J.: Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. Psychometrika **35**(3), 283–319 (1970)
3. Cichocki, A., Mandic, D.P., Phan, A.H., Caiafa, C.F., Zhou, G., Zhao, Q., De Lathauwer, L.: Tensor decompositions for signal processing applications. From two-way to multiway component analysis. IEEE Signal Processing Magazine **32**(2), 145–163 (2015)
4. Cichocki, A., Zdunek, R., Phan, A., Amari, S.: Nonnegative Matrix and Tensor Factorizations. Wiley (2009)
5. Comon, P.: Tensors: a brief introduction. IEEE Signal Processing Magazine **31**(3), 44–53 (2014)
6. Cox, D.A., Little, J.B., O'Shea, D.: Ideals, Varieties and Algorithms. Springer-Verlag, 3rd ed. (2007)
7. Domanov, I., De Lathauwer, L.: Canonical polyadic decomposition of third-order tensors: Relaxed uniqueness conditions and algebraic algorithm. Linear Algebra Appl. **513**, 342–375 (2017)
8. Grasedyck, L., Kressner, D., Tobler, C.: A literature survey of low-rank tensor approximation techniques. GAMM-Mitteilungen **36**(1), 53–78 (2013)
9. Hackbusch, W.: Tensor spaces and numerical tensor calculus, Springer series in computational mathematics, vol. 42. Springer, Heidelberg (2012)
10. Harshman, R.A.: Foundations of the PARAFAC procedure: Model and conditions for an "explanatory" multi-mode factor analysis. UCLA Working Papers in Phonetics **16**(1), 1–84 (1970)
11. Khoromskij, B.N.: Tensors-structured numerical methods in scientific computing: Survey on recent advances. Chemometrics and Intelligent Laboratory Systems **110**(1), 1–19 (2012)
12. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
13. Kroonenberg, P.M.: Applied Multiway Data Analysis. Wiley (2008)
14. Sidiropoulos, N.D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E.E., Faloutsos, C.: Tensor decomposition for signal processing and machine learning. IEEE Transactions on Signal Processing **65**(13), 3551–3582 (2017)
15. Smilde, A., Bro, R., Geladi, P.: Multi-way Analysis. Applications in the Chemical Sciences. John Wiley and Sons, Chichester, U.K. (2004)
16. Stetter, H.J.: Numerical Polynomial Algebra. SIAM (2004)
17. Vervliet, N., Debals, O., Sorber, L., Van Barel, M., De Lathauwer, L.: Tensorlab 3.0, available online, Mar. 2016. URL: https://www.tensorlab.net/.