Results in Applied Mathematics 31 (2026) 100762

Contents lists available at ScienceDirect

Results in Applied Mathematics

journal homepage: www.elsevier.com/locate/results-in-applied-mathematics

Multilinearization of polynomial models by linear state transformation

Christoph Kaufmann [a,1], Sabine Le Borne [b,*], Gerwald Lichtenberg [c,1], Simon Wallgram [b,1]

[a] *Application Center for Integration of Local Energy Systems, Fraunhofer, Am Schleusengraben 22, Hamburg, 21029, Germany*
[b] *Institute of Mathematics, Hamburg University of Technology, Am Schwarzenberg-Campus 3, Hamburg, 21073, Germany*
[c] *Faculty of Life Sciences, Hamburg University of Applied Sciences, Ulmenliet 20, Hamburg, 21033, Germany*

ARTICLE INFO

MSC:
46G25
47H60

Keywords:
Polynomial models
Linear transformation
Multilinearization

ABSTRACT

Many problems in engineering involve the analysis and model-based control of nonlinear systems. Often a tradeoff exists between the fidelity of the model and the computational complexity of analysis and design algorithms. Multilinear models offer significant advantages in this regard since they can be represented as tensors which enables efficient representation and computation, even in high-dimensional settings. This paper deals with the multilinearization of polynomial models through linear transformation. We are able to provide a comprehensive analysis for the case of two variables. For the higher dimensional case, we propose partial solutions which exploit our results for the two-dimensional case. We perform theoretical runtime analyses of the resulting algorithms which we also illustrate through numerical results.

1. Introduction

Many problems in engineering involve the analysis and control of nonlinear systems. Often a tradeoff exists between the fidelity of the model and the computational complexity of analysis and design algorithms. Multilinear models offer significant advantages in this regard. They can be represented as tensors, which enables the use of tensor decomposition methods. This allows for efficient representation and computation, even in high-dimensional settings, while still maintaining the ability to capture essential nonlinear dynamics [1].

In this work, we focus on higher order polynomial models. The following example shows that certain polynomial models can be represented as multilinear ones after applying a linear state transformation.

Example 1.1. Consider a system of ordinary differential equations given by the polynomial model

$$\begin{aligned}\dot{x}_1 &= 2x_1^2 + x_1x_2 + x_1,\\ \dot{x}_2 &= -4x_1^2 - 2x_1x_2 + x_2.\end{aligned}$$

* Corresponding author.
E-mail addresses: christoph.kaufmann@iwes.fraunhofer.de (C. Kaufmann), leborne@tuhh.de (S. Le Borne), gerwald.lichtenberg@haw-hamburg.de (G. Lichtenberg), simon.wallgram@tuhh.de (S. Wallgram).

[1] Contributing authors.

https://doi.org/10.1016/j.rinam.2026.100762
Received 29 May 2026; Received in revised form 20 August 2026; Accepted 25 August 2026

Available online 5 September 2026
2590-0374/

Application of the linear transformation of variables

$$\begin{pmatrix}\tilde{x}_1\\ \tilde{x}_2\end{pmatrix} = \begin{pmatrix}1 & 0\\ 2 & 1\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} \iff \begin{pmatrix}x_1\\ x_2\end{pmatrix} = \begin{pmatrix}1 & 0\\ 2 & 1\end{pmatrix}^{-1}\begin{pmatrix}\tilde{x}_1\\ \tilde{x}_2\end{pmatrix} = \begin{pmatrix}1 & 0\\ -2 & 1\end{pmatrix}\begin{pmatrix}\tilde{x}_1\\ \tilde{x}_2\end{pmatrix}$$

to the model results in a multilinear model in the transformed variables,

$$\begin{aligned}\begin{pmatrix}\dot{\tilde{x}}_1\\ \dot{\tilde{x}}_2\end{pmatrix} &= \begin{pmatrix}1 & 0\\ 2 & 1\end{pmatrix}\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\end{pmatrix} = \begin{pmatrix}1 & 0\\ 2 & 1\end{pmatrix}\begin{pmatrix}2x_1^2 + x_1x_2 + x_1\\ -4x_1^2 - 2x_1x_2 + x_2\end{pmatrix}\\ &= \begin{pmatrix}1 & 0\\ 2 & 1\end{pmatrix}\begin{pmatrix}2\tilde{x}_1^2 + \tilde{x}_1(-2\tilde{x}_1 + \tilde{x}_2) + \tilde{x}_1\\ -4\tilde{x}_1^2 - 2\tilde{x}_1(-2\tilde{x}_1 + \tilde{x}_2) - 2\tilde{x}_1 + \tilde{x}_2\end{pmatrix}\\ &= \begin{pmatrix}\tilde{x}_1 + \tilde{x}_1\tilde{x}_2\\ \tilde{x}_2\end{pmatrix}.\end{aligned}$$

Such a transformation has the advantage that the transformed model is easier to simulate and analyse while the original dynamics of the model are preserved. The central problem that will be addressed in this work is the following:

Problem 1.2 (*Vector-valued Polynomial Multilinearization Problem*)**.** Given a (vector-valued) multivariate polynomial function $g : \mathbb{R}^n \to \mathbb{R}^n$ with $n \in \mathbb{N}$, decide whether there exists an invertible linear transformation $\mathbf{T} \in \mathbb{R}^{n\times n}$ and a (vector-valued) multilinear function $f : \mathbb{R}^n \to \mathbb{R}^n$ such that

$$f(\mathbf{x}) = \mathbf{T}g(\mathbf{T}^{-1}\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathbb{R}^n. \tag{1}$$

In particular, we will address the following questions:

(a) (How) can we characterize the set of polynomial models that can be transformed into multilinear models via linear state transformations?
(b) If there exist solutions to Problem 1.2, are they unique?
(c) If there exist solutions to Problem 1.2, how can they be computed (efficiently)?

The multilinearization of polynomial models through linear state transformations intersects several areas of existing research, yet we are not aware of any prior work that addresses our specific Problem 1.2 and the approach we present for its solution.

We will review some related work from the literature before summarizing our own contributions. The relevant literature can be categorized into two main groups: approximation-based multilinearization methods in the area of control theory, and exact algebraic methods for polynomial equivalence in the area of computational algebra and cryptography.

Multilinear Approximation of Nonlinear Models: Kruppa et al. [2] address the multilinearization problem through an approximation framework that fundamentally differs from our exact approach. They approximate a given nonlinear model by a multilinear model using projections which minimize the approximation error in a suitable norm. Their approach is broader in scope in the sense that they consider general nonlinear functions while we focus on polynomial functions. In contrast to our approach, their method is of approximate nature. Thus, the resulting multilinear function is not guaranteed to preserve the exact dynamics of the original model. While the technical details of their projection-based method differ significantly from our transformation approach, their work validates the importance of multilinear models. In particular, they expand the class of models that can be approximated in multilinear form which allows for the application of tensor decomposition methods.

The Polynomial Equivalence Problem: Another related area of research is the so-called Polynomial Equivalence Problem from the fields of computational algebra and cryptography. This problem was first introduced by Patarin [3] and its linear variant is defined by Faugère and Perret [4] as follows:

Problem 1.3 (*Polynomial Equivalence Problem*)**.** Given two polynomials $g, h : \mathbb{F}^n \to \mathbb{F}^n$ with $n \in \mathbb{N}$ and a finite field $\mathbb{F}$, decide whether there exist invertible linear transformations $\mathbf{T}, \mathbf{S} \in \mathbb{F}^{n\times n}$ such that

$$g(\mathbf{x}) = \mathbf{T}h(\mathbf{S}\mathbf{x}), \quad \text{for all } \mathbf{x} \in \mathbb{F}^n.$$

This problem differs in three key aspects from our Problem 1.2. The first and foremost difference is that in Problem 1.3, two polynomials g and h are given, and the unknowns are the linear transformations $\mathbf{T}$ and $\mathbf{S}$. In our Problem 1.2, only g is given, and the unknowns are the transformation $\mathbf{T}$ and the multilinear function f. Secondly, Problem 1.3 allows for two distinct transformations $\mathbf{T}$ and $\mathbf{S}$ whereas we require $\mathbf{S} = \mathbf{T}^{-1}$ and hence are looking only for a single transformation $\mathbf{T}$. Lastly, Problem 1.3 is defined over finite fields $\mathbb{F}$ while our work considers real-valued functions.

A different version of the polynomial equivalence problem has also been studied by Kayal [5]. They consider only homogeneous polynomials and target specific multilinear functions (so-called elementary symmetric polynomials) while our problem considers general polynomial functions and the class of all multilinear functions. Furthermore, in their work they only consider scalar-valued polynomials while we consider vector-valued polynomial functions. They provide a randomized algorithm that solves their version of the polynomial equivalence problem.

In this paper, after laying the groundwork by giving all necessary definitions and notations in Section 2, we will make the following contributions towards the solution of Problem 1.2:

(a) In Section 3, we provide a full analytical treatment of the bivariate case ($n = 2$) which includes results on existence, uniqueness and the computation of solutions.
(b) We extend those results in Section 4 to polynomial models arising from multilinear functions transformed by block diagonal matrices as well as briefly discuss the general n-variate case.
(c) In Section 5, we develop algorithms for bivariate polynomial systems as well as for polynomial systems arising from multilinear functions transformed by block diagonal matrices. We analyse and test the runtime of these algorithms for randomly generated polynomial functions.

2. Preliminaries

In order to rigorously describe and analyse the Problem 1.2 of multilinearizing polynomial models by linear state transformations, we now introduce the relevant definitions and terminology.

Definition 2.1 (*Monomial, Degree*)**.** For $n \in \mathbb{N}$ and a multi-index $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}_0^n$, i.e., an n-tuple of non-negative integers, an n-variate monomial is a term of the form

$$\mathbf{x}^\alpha = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \dots \cdot x_n^{\alpha_n}.$$

The total degree of a monomial is defined as $|\alpha| = \alpha_1 + \dots + \alpha_n$.

Definition 2.2 (*Polynomials and Multilinear Polynomials*)**.** Let $n \in \mathbb{N}$.

(i) An n-variate polynomial $g(\mathbf{x})$ over $\mathbb{R}$ is a linear combination of monomials with coefficients in $\mathbb{R}$. We write

$$g(\mathbf{x}) = \sum_{\alpha \in \mathbb{N}_0^n} \pi_\alpha \mathbf{x}^\alpha, \quad \pi_\alpha \in \mathbb{R}.$$

(ii) For a given polynomial g, we call π_α the coefficient of the monomial $\mathbf{x}^\alpha$ and $\{\pi_\alpha \in \mathbb{R} \mid \alpha \in \mathbb{N}_0^n\}$ the (set of) coefficients of g. The degree of g, denoted as $\deg(g)$, is defined as the maximum total degree of its monomials with nonzero coefficients. The degree of the zero polynomial, which has all coefficients equal to zero, is defined to be -1.
(iii) We say that a polynomial g is multilinear if for all $\alpha \in \mathbb{N}_0^n$ with $\pi_\alpha \neq 0$, it holds that $\max_{i=1,\dots,n} \alpha_i \leq 1$, i.e., in all the monomials $\mathbf{x}^\alpha$ of g, the exponents α_i are at most 1 for $i = 1, \dots, n$.
(iv) The set of all n-variate polynomials with coefficients in $\mathbb{R}$ is denoted by $\mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, \dots, x_n]$.

When writing polynomials, we order their monomials according to the *total degree ordering* (also called graded lexicographic order, or *grlex*). Further details about monomial orderings can be found, for example, in [6, Chapter 1].

Definition 2.3 (*Total Degree Ordering*)**.** Let $\mathbf{x}^\alpha$ and $\mathbf{x}^\beta$, where $\alpha, \beta \in \mathbb{N}_0^n$, be two monomials. We write $\mathbf{x}^\alpha > \mathbf{x}^\beta$ if either

1. $|\alpha| > |\beta|$, or
2. $|\alpha| = |\beta|$ and there exists $i \in \{1, \dots, n\}$ such that $\alpha_i > \beta_i$ and $\alpha_j = \beta_j$ for all $j < i$.

The following example illustrates Definition 2.3.

Example 2.4. For $\alpha = (2, 3, 1),\ \beta = (3, 1, 1)$, there holds

$$\mathbf{x}^\alpha = x_1^2 x_2^3 x_3 > x_1^3 x_2 x_3 = \mathbf{x}^\beta$$

since $|\alpha| = 6 > 5 = |\beta|$. Furthermore, for $\tilde{\alpha} = (2, 1, 2),\ \tilde{\beta} = (1, 2, 2)$ there holds

$$\mathbf{x}^{\tilde{\alpha}} = x_1^2 x_2 x_3^2 > x_1 x_2^2 x_3^2 = \mathbf{x}^{\tilde{\beta}}$$

since $|\tilde{\alpha}| = |\tilde{\beta}| = 5$ but $\tilde{\alpha}_1 > \tilde{\beta}_1$.

We write polynomials with monomials ordered from largest to smallest in this sense.

In the remainder of this work, we will primarily consider vector-valued polynomial and multilinear functions defined as follows.

Definition 2.5 (*Polynomial, Multilinear and Affine Functions*)**.** Let $n, m \in \mathbb{N}$.

(i) Let $g_1 \dots, g_m$ be n-variate polynomials. We call a function $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$ defined by

$$\mathbf{g}(\mathbf{x}) := \begin{pmatrix} g_1(x_1, x_2, \dots, x_n) \\ \vdots \\ g_m(x_1, x_2, \dots, x_n) \end{pmatrix}$$

a vector-valued polynomial function. We define the degree of $\mathbf{g}$ as $\deg(\mathbf{g}) := \max_{i=1,\dots,m} \deg(g_i)$.
(ii) If $g_1, g_2, \dots, g_m$ are in particular multilinear polynomials, then the function $\mathbf{g}$ defined as above is called a vector-valued multilinear function.

(iii) If $g_1, g_2, \ldots, g_m$ are in particular polynomials of degree at most 1, then the function $\mathbf{g}$ defined as above is called an affine function.

For brevity, we will refer to vector-valued polynomial and multilinear functions simply as "polynomial" and "multilinear functions" and use the same symbol as for polynomials, i.e., g instead of $\mathbf{g}$. We typically denote general polynomial functions by the symbol g and multilinear functions by f.

We next define polynomial and multilinear models as well as linear (state) transformations of such models.

Definition 2.6 (*Polynomial and Multilinear Models*)**.** Let $n \in \mathbb{N}$ and $\mathbf{x} : I \subseteq \mathbb{R} \to \mathbb{R}^n$, $t \mapsto \mathbf{x}(t) = (x_1(t), \ldots, x_n(t))$ be a differentiable function. Let $g : \mathbb{R}^n \to \mathbb{R}^n$ be a polynomial function. A system of autonomous ordinary differential equations (ODEs) of the form

$$\frac{d}{dt}\mathbf{x}(t) = g(\mathbf{x}(t))$$

is called a polynomial model. If g is in particular a multilinear function, then the system is called a multilinear model.

We usually write polynomial models as $\dot{\mathbf{x}} = g(\mathbf{x})$ and multilinear models as $\dot{\mathbf{x}} = f(\mathbf{x})$. The variable $\mathbf{x}$ is commonly referred to as a *state* of the model.

Remark 2.7. According to Kruppa et al. [2], one can express multilinear models as a contracted tensor product

$$\dot{\mathbf{x}} = \langle F | M(\mathbf{x}) \rangle,$$

with a the so-called state transition tensor F and the so-called monomial tensor $M(\mathbf{x})$.

Definition 2.8 (*Equivalence of Systems*)**.** Let $h, \tilde{h} : \mathbb{R}^n \to \mathbb{R}^n$, $n \in \mathbb{N}$, be continuous functions, and consider the two systems of ODEs

$$\dot{\mathbf{x}} = h(\mathbf{x}), \quad \dot{\mathbf{y}} = \tilde{h}(\mathbf{y}).$$

Then we define the following.

(i) A continuously differentiable function $\mathbf{x} : I \subseteq \mathbb{R} \to \mathbb{R}^n$ is called a *trajectory* of the system $\dot{\mathbf{x}} = h(\mathbf{x})$ if it is a solution of the ODE.

(ii) The systems $\dot{\mathbf{x}} = h(\mathbf{x})$ and $\dot{\mathbf{y}} = \tilde{h}(\mathbf{y})$ are *topologically equivalent* if there exists a homeomorphism $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ (i.e., a bijective, continuous map whose inverse is also continuous) that maps trajectories of one system onto trajectories of the other.

Even though it would be possible to consider any homeomorphism Φ to transform polynomial models into multilinear ones, in this work we focus on the class of homeomorphisms given by linear transformations.

Definition 2.9 (*Linear State Transformation*)**.** Let $\mathbf{T} \in \mathbb{R}^{n\times n}$, $n \in \mathbb{N}$, be an invertible matrix. The transformation

$$T : \mathbb{R}^n \to \mathbb{R}^n, \quad \mathbf{x} \mapsto \tilde{\mathbf{x}} := \mathbf{T}\mathbf{x},$$

is called a *linear state transformation* (or simply *linear transformation*). The matrix $\mathbf{T}$ is called the *transformation matrix*.

While we primarily consider polynomial models over $\mathbb{R}$, i.e., $f, g : \mathbb{R}^n \to \mathbb{R}^n$, all definitions in this section extend naturally to the complex setting $f, g : \mathbb{C}^n \to \mathbb{C}^n$.

Problem 1.2 now arises as follows: Given $n \in \mathbb{N}$ and a polynomial model with a polynomial function $g : \mathbb{R}^n \to \mathbb{R}^n$,

$$\dot{\mathbf{x}} = g(\mathbf{x}), \tag{2}$$

the goal is to find a linear state transformation $T : \mathbb{R}^n \to \mathbb{R}^n$ with an invertible transformation matrix $\mathbf{T} \in \mathbb{R}^{n\times n}$ such that, under

$$\tilde{\mathbf{x}} := T(\mathbf{x}) = \mathbf{T}\mathbf{x}, \tag{3}$$

the polynomial model (2) is transformed into a (topologically equivalent) multilinear model

$$\dot{\tilde{\mathbf{x}}} = f(\tilde{\mathbf{x}}) \tag{4}$$

with a multilinear function $f : \mathbb{R}^n \to \mathbb{R}^n$. Substituting (3) into (4), we obtain the equality

$$f(\tilde{\mathbf{x}}) = T \circ g \circ T^{-1}(\tilde{\mathbf{x}}) \tag{5}$$

which leads directly to Problem 1.2 and (1).

Remark 2.10. Since $\mathbf{T}$ is invertible, we can write (1) equivalently as

$$g(\mathbf{x}) = \mathbf{T}^{-1} f(\mathbf{T}\mathbf{x}). \tag{6}$$

3. The bivariate multilinearization problem

In this section, we derive necessary and sufficient conditions for the existence of transformations that solve Problem 1.2 in the bivariate case ($n = 2$) and discuss their uniqueness. In doing so, we provide an analytical approach to the problem of transforming polynomial models into multilinear ones via linear state transformations for models with two state variables. This low-dimensional setting allows for explicit computations and helps us gain insight into the structure of the problem. We will see that the problem can be reduced to finding the common roots of two sets of two quadratic polynomials. Solutions are found to be only unique up to a certain equivalence and it is possible to compute necessary and sufficient conditions for their existence.

3.1. Bivariate models

Let $n = 2$. Then a general multilinear model with multilinear function $f : \mathbb{R}^2 \to \mathbb{R}^2$ can be written as

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = f\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \varphi_{11}^{(1)} x_1 x_2 + \varphi_{10}^{(1)} x_1 + \varphi_{01}^{(1)} x_2 + \varphi_{00}^{(1)} \\ \varphi_{11}^{(2)} x_1 x_2 + \varphi_{10}^{(2)} x_1 + \varphi_{01}^{(2)} x_2 + \varphi_{00}^{(2)} \end{pmatrix}. \tag{7}$$

Here we have $(x_1, x_2)^T \in \mathbb{R}^2$ with coefficients $\varphi_\alpha^{(i)} \in \mathbb{R}$ where $\alpha \in \mathbb{N}_0^2$ is a multi-index and the superscript $i \in \{1, 2\}$ in $\varphi_\alpha^{(i)}$ indicates the equation in which the coefficient appears. In this two-dimensional setting, an invertible transformation matrix $\mathbf{T}$ is a 2×2 matrix

$$\mathbf{T} = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix} \in \mathbb{R}^{2\times 2}$$

where $\det(\mathbf{T}) = t_{11}t_{22} - t_{12}t_{21} \neq 0$.

Since (1) and (6) are equivalent, we can get an idea of which polynomial functions have a chance to be transformed into a multilinear function by plugging both the general two-dimensional multilinear function f (7) and the transformation matrix $\mathbf{T}$ into (6). The resulting g contains expressions that if expanded only contain monomials of degree at most two. This means that, in the two-dimensional case, it is not possible to linearly transform a polynomial function with polynomials of degree larger than two into a multilinear function. Hence, any bivariate polynomial function g for which Problem 1.2 is solvable has the form

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = g\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \pi_{20}^{(1)} x_1^2 + \pi_{11}^{(1)} x_1 x_2 + \pi_{02}^{(1)} x_2^2 + \pi_{10}^{(1)} x_1 + \pi_{01}^{(1)} x_2 + \pi_{00}^{(1)} \\ \pi_{20}^{(2)} x_1^2 + \pi_{11}^{(2)} x_1 x_2 + \pi_{02}^{(2)} x_2^2 + \pi_{10}^{(2)} x_1 + \pi_{01}^{(2)} x_2 + \pi_{00}^{(2)} \end{pmatrix}. \tag{8}$$

In this section, unless stated otherwise, we therefore assume that any polynomial function $g : \mathbb{R}^2 \to \mathbb{R}^2$ has degree less than or equal to two and can be written as in (8).

Remark 3.1. We define the polynomial and multilinear models, as well as the transformation matrix, only for real numbers, as this is our main focus. However, these definitions can be easily extended to the complex case which becomes relevant later on in this section.

With a bivariate polynomial g of the form (8), we can plug a transformation matrix $\mathbf{T}$ into (5) and explicitly compute the coefficients of f. Using the shorthand $D = \det(\mathbf{T}) \neq 0$, we obtain

$$f(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{pmatrix} := \mathbf{T} g\left(\mathbf{T}^{-1}\mathbf{x}\right) = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix} g\left(\frac{1}{D} \begin{pmatrix} t_{22} & -t_{12} \\ -t_{21} & t_{11} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \right)$$

with

$$\begin{aligned} f_1(\mathbf{x}) =& t_{11}\big[\pi_{00}^{(1)} + D^{-1}\pi_{10}^{(1)}(t_{22}x_1 - t_{12}x_2) + D^{-1}\pi_{01}^{(1)}(-t_{21}x_1 + t_{11}x_2) \\ &+ D^{-2}\big(\pi_{11}^{(1)}(t_{22}x_1 - t_{12}x_2)(-t_{21}x_1 + t_{11}x_2) + \pi_{20}^{(1)}(t_{22}x_1 - t_{12}x_2)^2 \\ &+ \pi_{02}^{(1)}(-t_{21}x_1 + t_{11}x_2)^2\big)\big] \\ &+ t_{12}\big[\pi_{00}^{(2)} + D^{-1}\pi_{10}^{(2)}(t_{22}x_1 - t_{12}x_2) + D^{-1}\pi_{01}^{(2)}(-t_{21}x_1 + t_{11}x_2) \\ &+ D^{-2}\big(\pi_{11}^{(2)}(t_{22}x_1 - t_{12}x_2)(-t_{21}x_1 + t_{11}x_2) + \pi_{20}^{(2)}(t_{22}x_1 - t_{12}x_2)^2 \\ &+ \pi_{02}^{(2)}(-t_{21}x_1 + t_{11}x_2)^2\big)\big] \end{aligned}$$

and

$$\begin{aligned} f_2(\mathbf{x}) =& t_{21}\big[\pi_{00}^{(1)} + D^{-1}\pi_{10}^{(1)}(t_{22}x_1 - t_{12}x_2) + D^{-1}\pi_{01}^{(1)}(-t_{21}x_1 + t_{11}x_2) \\ &+ D^{-2}\big(\pi_{11}^{(1)}(t_{22}x_1 - t_{12}x_2)(-t_{21}x_1 + t_{11}x_2) + \pi_{20}^{(1)}(t_{22}x_1 - t_{12}x_2)^2 \\ &+ \pi_{02}^{(1)}(-t_{21}x_1 + t_{11}x_2)^2\big)\big] \\ &+ t_{22}\big[\pi_{00}^{(2)} + D^{-1}\pi_{10}^{(2)}(t_{22}x_1 - t_{12}x_2) + D^{-1}\pi_{01}^{(2)}(-t_{21}x_1 + t_{11}x_2) \\ &+ D^{-2}\big(\pi_{11}^{(2)}(t_{22}x_1 - t_{12}x_2)(-t_{21}x_1 + t_{11}x_2) + \pi_{20}^{(2)}(t_{22}x_1 - t_{12}x_2)^2 \end{aligned}$$

$$+\pi_{02}^{(2)}(-t_{21}x_1+t_{11}x_2)^2)\big].$$

The function f will be multilinear if and only if the coefficients of the monomials x_1^2 and x_2^2 are equal to zero. This leads to the following Eqs. (9a), (9b) for the coefficients of x_1^2, x_2^2 in f_1, and Eqs. (9c), (9d) for the coefficients of x_1^2, x_2^2 in f_2, respectively,

$$t_{11}(\pi_{20}^{(1)}t_{22}^2+\pi_{02}^{(1)}t_{21}^2-\pi_{11}^{(1)}t_{21}t_{22})+t_{12}(\pi_{20}^{(2)}t_{22}^2+\pi_{02}^{(2)}t_{21}^2-\pi_{11}^{(2)}t_{21}t_{22})=0, \tag{9a}$$
$$t_{11}(\pi_{20}^{(1)}t_{12}^2+\pi_{02}^{(1)}t_{11}^2-\pi_{11}^{(1)}t_{11}t_{12})+t_{12}(\pi_{20}^{(2)}t_{12}^2+\pi_{02}^{(2)}t_{11}^2-\pi_{11}^{(2)}t_{11}t_{12})=0, \tag{9b}$$
$$t_{21}(\pi_{20}^{(1)}t_{22}^2+\pi_{02}^{(1)}t_{21}^2-\pi_{11}^{(1)}t_{21}t_{22})+t_{22}(\pi_{20}^{(2)}t_{22}^2+\pi_{02}^{(2)}t_{21}^2-\pi_{11}^{(2)}t_{21}t_{22})=0, \tag{9c}$$
$$t_{21}(\pi_{20}^{(1)}t_{12}^2+\pi_{02}^{(1)}t_{11}^2-\pi_{11}^{(1)}t_{11}t_{12})+t_{22}(\pi_{20}^{(2)}t_{12}^2+\pi_{02}^{(2)}t_{11}^2-\pi_{11}^{(2)}t_{11}t_{12})=0. \tag{9d}$$

The structure of these Eqs. (9) suggests the possibility of a further simplification, namely that Eqs. (9a) and (9c) can be written as a linear systems of equations,

$$\underbrace{\begin{pmatrix} t_{11} & t_{12}\\ t_{21} & t_{22}\end{pmatrix}}_{=\mathbf{T}}\begin{pmatrix}\pi_{20}^{(1)}t_{22}^2+\pi_{02}^{(1)}t_{21}^2-\pi_{11}^{(1)}t_{21}t_{22}\\ \pi_{20}^{(2)}t_{22}^2+\pi_{02}^{(2)}t_{21}^2-\pi_{11}^{(2)}t_{21}t_{22}\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix}, \tag{10}$$

and similarly for (9b) and (9d). This leads to the following proposition.

Proposition 3.2. *Let a polynomial function $g:\mathbb{R}^2\to\mathbb{R}^2$ as in (8) be given. Then a transformation $\mathbf{T}$ satisfies the Eqs. (9) if and only if $\mathbf{T}$ satisfies the Eqs. (11),*

$$\pi_{20}^{(1)}t_{22}^2+\pi_{02}^{(1)}t_{21}^2-\pi_{11}^{(1)}t_{21}t_{22}=0, \tag{11a}$$
$$\pi_{20}^{(2)}t_{22}^2+\pi_{02}^{(2)}t_{21}^2-\pi_{11}^{(2)}t_{21}t_{22}=0, \tag{11b}$$
$$\pi_{20}^{(1)}t_{12}^2+\pi_{02}^{(1)}t_{11}^2-\pi_{11}^{(1)}t_{11}t_{12}=0, \tag{11c}$$
$$\pi_{20}^{(2)}t_{12}^2+\pi_{02}^{(2)}t_{11}^2-\pi_{11}^{(2)}t_{11}t_{12}=0. \tag{11d}$$

In particular, a matrix $\mathbf{T}\in\mathbb{R}^{2\times 2}$ solves Problem 1.2 if and only if $\mathbf{T}$ satisfies (11) and $\det(\mathbf{T})\neq 0$.

Proof. The statement follows from Eqs. (10) together with the assumption that $\mathbf{T}$ is invertible. □

3.2. Uniqueness of solutions

Next, we discuss the uniqueness of solutions to Problem 1.2 for bivariate models. The following proposition immediately allows us to draw conclusions about the uniqueness of solutions. In addition, it provides a tool to further simplify (11), enabling a more detailed analysis.

Proposition 3.3. *Let $\lambda,\mu\in\mathbb{R}$ and let $\mathbf{D}=\operatorname{diag}(\lambda,\mu)\in\mathbb{R}^{2\times 2}$ be a diagonal matrix. Furthermore, let $\mathbf{P}\in\mathbb{R}^{2\times 2}$ be a permutation matrix and let $f:\mathbb{R}^2\to\mathbb{R}^2$ be a multilinear function. Then the functions $\hat f:\mathbb{R}^2\to\mathbb{R}^2$ and $\tilde f:\mathbb{R}^2\to\mathbb{R}^2$ defined by*

$$\hat f(\mathbf{x}) := f(\mathbf{D}\mathbf{x}) \text{ and } \quad \tilde f(\mathbf{x}) := f(\mathbf{P}\mathbf{x}) \quad \text{for all } \mathbf{x}\in\mathbb{R}^2$$

are multilinear. In other words, scaling and permuting the variables of a multilinear function preserves its multilinearity.

Proof. The claim follows by direct computation. □

Corollary 3.4. *Let $g:\mathbb{R}^2\to\mathbb{R}^2$ be a polynomial function and let $\mathbf{T}\in\mathbb{R}^{2\times 2}$ be a solution to Problem 1.2. Furthermore, let $\lambda,\mu\in\mathbb{R}\setminus\{0\}$ and let $\mathbf{D}=\operatorname{diag}(\lambda,\mu)\in\mathbb{R}^{2\times 2}$ be a diagonal matrix and $\mathbf{P}\in\mathbb{R}^{2\times 2}$ be a permutation matrix. Then the matrices $\mathbf{DT}$ and $\mathbf{PT}$ are also solutions to Problem 1.2.*

Proof. Let $f(\mathbf{x}) := \mathbf{T}g(\mathbf{T}^{-1}\mathbf{x})$ be multilinear and let $\mathbf{D}$ and $\mathbf{P}$ be as above. Since $\lambda,\mu\neq 0$, the matrix $\mathbf{D}$ is invertible and $\mathbf{P}$ is invertible as a permutation matrix. Thus also $\mathbf{DT}$ and $\mathbf{PT}$ are invertible. Using Proposition 3.3, we have that the functions $f(\mathbf{D}^{-1}\mathbf{x})$ and $f(\mathbf{P}^{-1}\mathbf{x})$ remain multilinear. Since multiplying a multilinear function by a matrix preserves multilinearity, the function

$$\tilde f(\mathbf{x}) := \mathbf{D}f(\mathbf{D}^{-1}\mathbf{x})$$

is multilinear. For this multilinear $\tilde f$, it holds that

$$\tilde f(\mathbf{x}) = \mathbf{D}f(\mathbf{D}^{-1}\mathbf{x}) = \mathbf{DT}g(\mathbf{T}^{-1}\mathbf{D}^{-1}\mathbf{x}) = \mathbf{DT}g((\mathbf{DT})^{-1}\mathbf{x}).$$

Hence, the matrix $\mathbf{DT}$ solves Problem 1.2. The same argument applies to $\mathbf{PT}$. □

Corollary 3.4 in particular shows that for a given solution $\mathbf{T}$, any matrix $\mathbf{M}$ that can be written as a product of an invertible diagonal matrix $\mathbf{D}$ and a permutation matrix $\mathbf{P}$ has the property that $\mathbf{MT}$ is again a solution to Problem 1.2. Hence, solutions to Problem 1.2 are not unique. We call such matrices *scaled permutation matrices*.

Definition 3.5 (*Scaled Permutation Matrix*)**.** A matrix $\mathbf{M} \in \mathbb{R}^{2\times 2}$ is called a *scaled permutation matrix* if it is the product $\mathbf{M} = \mathbf{DP}$ of a diagonal matrix $\mathbf{D} = \operatorname{diag}(\lambda, \mu) \in \mathbb{R}^{2\times 2}$ with $\lambda, \mu \in \mathbb{R} \setminus \{0\}$ and a permutation matrix $\mathbf{P} \in \mathbb{R}^{2\times 2}$.

We next establish equivalence classes of solutions to Problem 1.2 based on their relation via scaled permutation matrices $\mathbf{M}$.

Definition 3.6 (*Equivalence of Solutions*)**.** Let $\mathbf{T}, \tilde{\mathbf{T}} \in \mathbb{R}^{2\times 2}$ be solutions of Problem 1.2. We say $\mathbf{T}$ is equivalent to $\tilde{\mathbf{T}}$, denoted by $\mathbf{T} \sim \tilde{\mathbf{T}}$, if there exists a scaled permutation matrix $\mathbf{M}$ such that $\mathbf{T} = \mathbf{M}\tilde{\mathbf{T}}$. We denote the equivalence class of $\mathbf{T}$ by $[\mathbf{T}]$.

Proposition 3.7. *The relation $\sim$ in Definition 3.6 is an equivalence relation.*

Proof. The statement follows from the fact that scaled permutation matrices are invertible and that the product of two scaled permutation matrices is again a scaled permutation matrix. □

Remark 3.8. Definitions 3.5 and 3.6 as well as Corollary 3.4 and Proposition 3.7 can easily be extended to n-variate models. The proofs are analogous.

By Corollary 3.4 we may, without loss of generality, restrict our search for solutions to Problem 1.2 to a special type of matrix.

Proposition 3.9. *Let $\mathbf{T} \in \mathbb{R}^{2\times 2}$ be an invertible matrix. Then there exists a scaled permutation matrix $\mathbf{M} \in \mathbb{R}^{2\times 2}$ such that $\tilde{\mathbf{T}} := \mathbf{MT}$ is of the form*

$$\tilde{\mathbf{T}} = \begin{pmatrix} 1 & * \\ * & 1 \end{pmatrix}.$$

Proof. Let $\mathbf{T} \in \mathbb{R}^{2\times 2}$ be an invertible matrix. It holds that $\det(\mathbf{T}) = t_{11}t_{22} - t_{12}t_{21} \neq 0$. Hence, $t_{11}t_{22}$ and $t_{12}t_{21}$ cannot both be zero at the same time. If $t_{11}t_{22} \neq 0$, we define

$$\mathbf{M} := \begin{pmatrix} \frac{1}{t_{11}} & 0 \\ 0 & \frac{1}{t_{22}} \end{pmatrix}$$

which is a scaled permutation matrix that yields

$$\tilde{\mathbf{T}} := \mathbf{MT} = \begin{pmatrix} \frac{1}{t_{11}} & 0 \\ 0 & \frac{1}{t_{22}} \end{pmatrix} \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix} = \begin{pmatrix} 1 & \frac{t_{12}}{t_{11}} \\ \frac{t_{21}}{t_{22}} & 1 \end{pmatrix}.$$

Similarly, if $t_{12}t_{21} \neq 0$, we define $\mathbf{M}$ by

$$\mathbf{M} := \begin{pmatrix} 0 & \frac{1}{t_{21}} \\ \frac{1}{t_{12}} & 0 \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{1}{t_{21}} & 0 \\ 0 & \frac{1}{t_{12}} \end{pmatrix}}_{:=\mathbf{D}} \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}_{:=\mathbf{P}}. \quad \square$$

Combining Propositions 3.2 and 3.9 yields the following

Proposition 3.10. *The matrix $\mathbf{T} = \begin{pmatrix} 1 & t_{12} \\ t_{21} & 1 \end{pmatrix}$ solves Problem 1.2 for a given polynomial function $g : \mathbb{R}^2 \to \mathbb{R}^2$ of the form (8) if and only if the following three conditions are satisfied:*

(i) $\det(\tilde{\mathbf{T}}) \neq 0$.

(ii) The entry t_{21} of $\tilde{\mathbf{T}}$ is a common root of the polynomials p_1 and p_2 defined by

$$p_1(x) := \pi_{02}^{(1)} x^2 - \pi_{11}^{(1)} x + \pi_{20}^{(1)}, \tag{12a}$$
$$p_2(x) := \pi_{02}^{(2)} x^2 - \pi_{11}^{(2)} x + \pi_{20}^{(2)}. \tag{12b}$$

(iii) The entry t_{12} is a common root of the polynomials q_1 and q_2 defined by

$$q_1(y) := \pi_{20}^{(1)} y^2 - \pi_{11}^{(1)} y + \pi_{02}^{(1)}, \tag{13a}$$
$$q_2(y) := \pi_{20}^{(2)} y^2 - \pi_{11}^{(2)} y + \pi_{02}^{(2)}. \tag{13b}$$

Remark 3.11. In Proposition 3.10, we set the diagonal entries of $\mathbf{T}$ equal to one since it simplifies solving the nonlinear equations (11) to finding common roots of two quadratic polynomials in (12) and (13). However, if it turns out that a resulting solution $\mathbf{T}$ is poorly conditioned, Corollary 3.4 allows us to diagonally scale $\mathbf{T}$ to obtain a possibly better conditioned solution.

Remark 3.12. In Proposition 3.10 we allow the roots x^* and y^* to be complex numbers. Thus, the resulting transformation matrix $\mathbf{T}$ is complex-valued if the polynomials have complex common roots. In this case, the transformed function $f(\mathbf{x}) = \mathbf{T}g(\mathbf{T}^{-1}\mathbf{x})$ is a multilinear function from $\mathbb{C}^2$ to $\mathbb{C}^2$ even if g is a polynomial function from $\mathbb{R}^2$ to $\mathbb{R}^2$.

We return to Example 1.1 and illustrate the results of Proposition 3.10.

Example 3.13. Let $g : \mathbb{R}^2 \to \mathbb{R}^2$ be the polynomial function from Example 1.1. In order to find a linear transformation matrix $\mathbf{T}$ such that $f(\mathbf{x}) = \mathbf{T}g(\mathbf{T}^{-1}\mathbf{x})$ is multilinear, we need to find common roots of the pairs of polynomials p_1, p_2 and q_1, q_2, defined in (12) and (13), which in this example are:

$$p_1(x) := -x + 2, \quad p_2(x) := 2x - 4,$$
$$q_1(y) := 2y^2 - y, \quad q_2(y) := -4y^2 + 2y.$$

A quick calculation shows that p_1 and p_2 have one common root, $x^* = 2$, while q_1 and q_2 have two common roots, $y_1^* = 0$ and $y_2^* = \frac{1}{2}$. Thus, there are two potential solutions to Problem 1.2 in the form $\begin{pmatrix} 1 & * \\ * & 1 \end{pmatrix}$, namely

$$\mathbf{T}_1 = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}, \quad \text{and} \quad \mathbf{T}_2 = \begin{pmatrix} 1 & \frac{1}{2} \\ 2 & 1 \end{pmatrix}.$$

However, only $\mathbf{T}_1$ satisfies $\det(\mathbf{T}) \neq 0$, and thus only $\mathbf{T}_1$ is a solution (which we had already seen in Example 1.1).

Proposition 3.10 implies that in the bivariate case, solutions to Problem 1.2 can be divided into at most four different equivalence classes with representatives

$$\mathbf{T}_1 = \begin{pmatrix} 1 & y_1^* \\ x_1^* & 1 \end{pmatrix}, \quad \mathbf{T}_2 = \begin{pmatrix} 1 & y_1^* \\ x_2^* & 1 \end{pmatrix}, \quad \mathbf{T}_3 = \begin{pmatrix} 1 & y_2^* \\ x_1^* & 1 \end{pmatrix}, \quad \text{and } \mathbf{T}_4 = \begin{pmatrix} 1 & y_2^* \\ x_2^* & 1 \end{pmatrix}. \tag{14}$$

where x_1^*, x_2^* and y_1^*, y_2^* denote common roots of the polynomials (12) and (13), respectively.

Indeed, we will show that unless g is affine (and hence already multilinear), all solutions belong to the same equivalence class, i.e., they are unique up to this equivalence. To prove this, we first establish the following lemma.

Lemma 3.14. *Let p be a univariate polynomial of degree $n \geq 1$ with coefficients $\mu_0, \mu_1, \dots, \mu_{n-1}, \mu_n \in \mathbb{R}$,*

$$p(x) = \mu_n x^n + \mu_{n-1} x^{n-1} + \cdots + \mu_1 x + \mu_0.$$

Let q be the polynomial defined by $q(x) = x^n p(\frac{1}{x})$, i.e., the coefficients of q are the same as the ones of p but in reversed order,

$$q(x) = \mu_0 x^n + \mu_1 x^{n-1} + \cdots + \mu_{n-1} x + \mu_n.$$

Then $x^* \in \mathbb{R} \setminus \{0\}$ is a root of p if and only if $\frac{1}{x^*}$ is a root of q. Furthermore, $x^* = 0$ is a root of p if and only if q has a lower degree than p.

Proof. For $x^* \neq 0$, the statement follows directly by evaluating q at the point $\frac{1}{(x^*)^n}$. The point $x^* = 0$ is a root of p if and only if $\mu_0 = 0$, i.e., if and only if q has a lower degree than p. □

Theorem 3.15. *Let $g : \mathbb{R}^2 \to \mathbb{R}^2$ be a polynomial function that is not affine. Then solutions to Problem 1.2 are unique up to equivalence.*

Proof. Let $\mathbf{T} \in \mathbb{R}^{2\times 2}$ be a solution of Problem 1.2. By Corollary 3.4 and Proposition 3.9, we may assume that $\mathbf{T}$ has the form

$$\mathbf{T} = \begin{pmatrix} 1 & y^* \\ x^* & 1 \end{pmatrix}$$

with $x^* y^* \neq 1$. Consider the polynomials p_1, p_2 and q_1, q_2 defined in (12) and (13). Since $\mathbf{T}$ solves Problem 1.2, by Proposition 3.10, we have that x^* is a common root of p_1 and p_2 whereas y^* is a common root of q_1 and q_2. Furthermore, note that the polynomials have reversed coefficients. Thus Lemma 3.14 is applicable to the pair (p_i, q_i) of polynomials if there holds $\deg(p_i) \geq 1$.

We distinguish cases based on the coefficients $\pi_{02}^{(1)}, \pi_{11}^{(1)} \pi_{20}^{(1)}$ of $p_1(x) = \pi_{02}^{(1)} x^2 - \pi_{11}^{(1)} x + \pi_{20}^{(1)}$ being zero or nonzero.

Case 1: Let $\pi_{02}^{(1)}, \pi_{20}^{(1)} \neq 0$. This implies that $x^*, y^* \neq 0$. Lemma 3.14 now states that since x^* is a root of p_1 (and p_2), $\frac{1}{x^*}$ must be a root of q_1 (and q_2) different from y^*. Otherwise, $y^* = \frac{1}{x^*} \Leftrightarrow x^* y^* = 1$, contradicting our assumption that $\mathbf{T}$ solves the problem and hence is regular. Analogously, $\frac{1}{y^*}$ must be a root of p_1 (and p_2) different from x^*. Thus, all common roots are given by $x^*, \frac{1}{y^*}$ for p_1, p_2 and $y^*, \frac{1}{x^*}$ for q_1, q_2. The four representatives of the potential equivalence classes (15) are thus given by

$$\mathbf{T} = \mathbf{T}_1 = \begin{pmatrix} 1 & y^* \\ x^* & 1 \end{pmatrix}, \quad \mathbf{T}_2 = \begin{pmatrix} 1 & \frac{1}{x^*} \\ x^* & 1 \end{pmatrix}, \quad \mathbf{T}_3 = \begin{pmatrix} 1 & y^* \\ \frac{1}{y^*} & 1 \end{pmatrix}, \quad \text{and } \mathbf{T}_4 = \begin{pmatrix} 1 & \frac{1}{x^*} \\ \frac{1}{y^*} & 1 \end{pmatrix}.$$

$\mathbf{T}_2$ and $\mathbf{T}_3$ cannot be solutions since $\det(\mathbf{T}_2) = \det(\mathbf{T}_3) = 0$. Furthermore, it holds that

$$\mathbf{T}_1 = \begin{pmatrix} 1 & y^* \\ x^* & 1 \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & y^* \\ x^* & 0 \end{pmatrix}}_{:=\mathbf{M}} \begin{pmatrix} 1 & \frac{1}{x^*} \\ \frac{1}{y^*} & 1 \end{pmatrix} = \mathbf{M}\mathbf{T}_4,$$

with a scaled permutation matrix $\mathbf{M}$. Hence, $\mathbf{T}_1$ and $\mathbf{T}_4$ are equivalent. In conclusion, in the case $\pi_{20}^{(1)}, \pi_{02}^{(1)} \neq 0$, there exists at most one solution of Problem 1.2 up to equivalence.

Case 2a: Let $\pi_{02}^{(1)} = 0$ and $\pi_{20}^{(1)}, \pi_{11}^{(1)} \neq 0$. This implies that $p_1(x^*) = -\pi_{11}^{(1)} x^* + \pi_{20}^{(1)} = 0$ and thus $x^* \neq 0$. Furthermore, since y^* is a root of $q_1(y) = \pi_{20}^{(1)} y^2 - \pi_{11}^{(1)} y$, we have that either $y^* = 0$, or $y^* = \frac{1}{x^*}$, by Lemma 3.14. This reduces the possible classes of solutions to the representatives

$$\mathbf{T}_1 = \begin{pmatrix} 1 & 0 \\ x^* & 1 \end{pmatrix}, \quad \text{and} \quad \mathbf{T}_2 = \begin{pmatrix} 1 & \frac{1}{x^*} \\ x^* & 1 \end{pmatrix}.$$

Again, $\mathbf{T}_2$ cannot be a solution since $\det(\mathbf{T}_2) = 0$.

Case 2b: Let $\pi_{20}^{(1)} = 0$ and $\pi_{11}^{(1)}, \pi_{02}^{(1)} \neq 0$. This case is analogous to Case 2a with the roles of p_i and q_i reversed. Here, we have that

$$\mathbf{T}_1 = \begin{pmatrix} 1 & y^* \\ 0 & 1 \end{pmatrix}$$

represents the only possible solution class.

Case 3: Let $\pi_{20}^{(1)}, \pi_{02}^{(1)} = 0$ and $\pi_{11}^{(1)} \neq 0$. Then the only root of p_1 is $x^* = 0$ whereas the only root of q_1 is $y^* = 0$. This implies that the only class of solutions is represented by $\mathbf{T} = \mathbf{I}$, the identity matrix.

Case 4a: Let $\pi_{20}^{(1)}, \pi_{11}^{(1)} = 0$ and $\pi_{02}^{(1)} \neq 0$. This case cannot occur since the respective polynomial $q_1(x) = \pi_{02}^{(1)}$ has no roots, i.e., there are no solutions $\mathbf{T}$.

Case 4b: Let $\pi_{02}^{(1)}, \pi_{11}^{(1)} = 0$ and $\pi_{20}^{(1)} \neq 0$. This case cannot occur since the respective polynomial $p_1(x) = \pi_{20}^{(1)}$ has no roots, i.e., there are no solutions $\mathbf{T}$.

Case 5: Let $\pi_{20}^{(1)} = \pi_{11}^{(1)} = \pi_{02}^{(1)} = 0$. Since g is assumed not to be affine, at least one of the coefficients of p_2 must be nonzero. Switching the roles of p_1 and p_2, we can now apply one of the previous cases to p_2.
With that we have covered all possible cases. □

In addition to solutions of Problem 1.2 being unique up to equivalence (for a non-affine g), the proof of Theorem 3.15 also shows the following.

Corollary 3.16. *Let $g : \mathbb{R}^2 \to \mathbb{R}^2$ be a polynomial function that is not affine. Then any solution $\mathbf{T}$ to Problem 1.2 lies in one of four equivalence classes.*

$$\mathbf{T}_1 = \begin{pmatrix} 1 & t_{12} \\ t_{21} & 1 \end{pmatrix}, \quad \mathbf{T}_2 = \begin{pmatrix} 1 & 0 \\ t_{21} & 1 \end{pmatrix}, \quad \mathbf{T}_3 = \begin{pmatrix} 1 & t_{12} \\ 0 & 1 \end{pmatrix}, \quad \textit{or} \quad \mathbf{T}_4 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{15}$$

for some $t_{12}, t_{21} \in \mathbb{R} \setminus \{0\}$ with $t_{12} \cdot t_{21} \neq 1$.

If the function g is affine, then any invertible linear transformation matrix $\mathbf{T}$ is a solution to Problem 1.2. This is because any (invertible) linear transformation applied to an affine function again yields an affine function. Thus, all four equivalence classes yield solutions such that Theorem 3.15 does not apply.

3.3. Necessary and sufficient conditions for the existence of solutions

In this subsection we state necessary and sufficient conditions for the existence of solutions to Problem 1.2 in the bivariate (non-affine) case. By Corollary 3.16, we know that solutions, if they exist, lie in one of the four equivalence classes with a representative listed in (15). Hence we only need to determine necessary and sufficient conditions for the matrices in (15) to be solutions of Problem 1.2.

Theorem 3.17. *Let $g : \mathbb{R}^2 \to \mathbb{R}^2$ be a polynomial function that is not affine.*

(i) $\mathbf{T}_4 = \mathbf{I}$ *(and all equivalent matrices) solves Problem 1.2 if and only if* $\pi_{20}^{(1)}, \pi_{20}^{(2)}, \pi_{02}^{(1)}, \pi_{02}^{(2)} = 0$.

(ii) $\mathbf{T}_2 = \begin{pmatrix} 1 & 0 \\ t_{21} & 1 \end{pmatrix}$ *with* $t_{21} \neq 0$ *(and all equivalent matrices) solves Problem 1.2 if and only if it holds that* $\pi_{02}^{(1)}, \pi_{02}^{(2)} = 0$ *and exactly one of the following three conditions is satisfied:*

(a) $\frac{\pi_{20}^{(1)}}{\pi_{11}^{(1)}} = \frac{\pi_{20}^{(2)}}{\pi_{11}^{(2)}}$, *and* $\pi_{20}^{(1)}, \pi_{11}^{(1)}, \pi_{20}^{(2)}, \pi_{11}^{(2)} \neq 0$,

(b) $\pi_{20}^{(1)}, \pi_{11}^{(1)} \neq 0,$ *and* $\pi_{20}^{(2)}, \pi_{11}^{(2)} = 0,$

(c) $\pi_{20}^{(1)}, \pi_{11}^{(1)} = 0,$ *and* $\pi_{20}^{(2)}, \pi_{11}^{(2)} \neq 0.$

(iii) $\mathbf{T}_3 = \begin{pmatrix} 1 & t_{12} \\ 0 & 1 \end{pmatrix}$ *with* $t_{12} \neq 0$ *(and all equivalent matrices) solves Problem 1.2 if and only if it holds that* $\pi_{20}^{(1)}, \pi_{20}^{(2)} = 0$ *and exactly one of the following three conditions is satisfied:*

(a) $\frac{\pi_{02}^{(1)}}{\pi_{11}^{(1)}} = \frac{\pi_{02}^{(2)}}{\pi_{11}^{(2)}},$ *and* $\pi_{02}^{(1)}, \pi_{11}^{(1)}, \pi_{02}^{(2)}, \pi_{11}^{(2)} \neq 0,$

(b) $\pi_{02}^{(1)}, \pi_{11}^{(1)} \neq 0,$ *and* $\pi_{02}^{(2)}, \pi_{11}^{(2)} = 0,$

(c) $\pi_{02}^{(1)}, \pi_{11}^{(1)} = 0,$ *and* $\pi_{02}^{(2)}, \pi_{11}^{(2)} \neq 0.$

(iv) $\mathbf{T}_1 = \begin{pmatrix} 1 & t_{12} \\ t_{21} & 1 \end{pmatrix}$ *with* $t_{12} \cdot t_{21} \notin \{0, 1\}$ *(and all equivalent matrices) solves Problem 1.2 if and only if one of the following holds:*

(a) $\frac{\pi_{11}^{(1)}}{\pi_{02}^{(1)}} = \frac{\pi_{11}^{(2)}}{\pi_{02}^{(2)}}, \quad \frac{\pi_{20}^{(1)}}{\pi_{02}^{(1)}} = \frac{\pi_{20}^{(2)}}{\pi_{02}^{(2)}}, \quad \pi_{02}^{(1)}, \pi_{20}^{(1)}, \pi_{02}^{(2)}, \pi_{20}^{(2)} \neq 0,$ *and* $\left(\pi_{11}^{(1)}\right)^2 - 4\pi_{20}^{(1)}\pi_{02}^{(1)} \neq 0.$

(b) $\pi_{02}^{(1)}, \pi_{20}^{(1)} \neq 0, \quad \pi_{02}^{(2)}, \pi_{11}^{(2)}, \pi_{20}^{(2)} = 0,$ *and* $\left(\pi_{11}^{(1)}\right)^2 - 4\pi_{20}^{(1)}\pi_{02}^{(1)} \neq 0.$

(c) $\pi_{02}^{(1)}, \pi_{11}^{(1)}, \pi_{20}^{(1)} = 0, \quad \pi_{02}^{(2)}, \pi_{20}^{(2)} \neq 0,$ *and* $\left(\pi_{11}^{(2)}\right)^2 - 4\pi_{20}^{(2)}\pi_{02}^{(2)} \neq 0.$

Proof. Let $g : \mathbb{R}^2 \to \mathbb{R}^2$ be a polynomial function that is not affine, and let p_1, p_2 and q_1, q_2 denote the polynomials defined in (12) and (13), respectively. We will apply Proposition 3.10 repeatedly.

Ad (i): $\Longrightarrow$: Let $\mathbf{T} = \mathbf{I}$ be a solution to Problem 1.2. Then, by Proposition 3.10, $x^* := t_{21} = 0$ and $y^* := t_{12} = 0$ are roots of p_1, p_2 and q_1, q_2, respectively, which implies that $\pi_{20}^{(1)} = \pi_{20}^{(2)} = \pi_{02}^{(1)} = \pi_{02}^{(2)} = 0.$
$\Longleftarrow$: Let $\pi_{20}^{(1)} = \pi_{20}^{(2)} = \pi_{02}^{(1)} = \pi_{02}^{(2)} = 0$. Then $x^* := 0$ and $y^* := 0$ are common roots of p_1, p_2 and q_1, q_2, respectively. Thus $\mathbf{T} = \mathbf{I}$ is a solution to Problem 1.2, and by Theorem 3.15 it is the only one up to equivalence.

Ad (ii): $\Longrightarrow$: Let $\mathbf{T}_2 = \begin{pmatrix} 1 & 0 \\ t_{21} & 1 \end{pmatrix}$ with $t_{21} \neq 0$ be a solution to Problem 1.2. Then, by Proposition 3.10, $x^* := t_{21} \neq 0$ and $y^* := t_{12} = 0$ are common roots of p_1, p_2 and q_1, q_2, respectively. The point $y^* = 0$ being a root of q_1 and q_2 implies that $\pi_{02}^{(1)} = \pi_{02}^{(2)} = 0$. Hence $x^* \neq 0$ is a root of

$$p_1(x) = -\pi_{11}^{(1)} x + \pi_{20}^{(1)} \text{ and } \quad p_2(x) = -\pi_{11}^{(2)} x + \pi_{20}^{(2)}.$$

We distinguish the following three subcases.

(a) Assume that neither p_1 nor p_2 are the zero polynomial. Since $x^* \neq 0$ is a root of p_1 we have that $\pi_{20}^{(1)} \neq 0$. Furthermore it also holds that $\pi_{11}^{(1)} \neq 0$, since otherwise p_1 would be a nonzero constant polynomial, which cannot have a nonzero root. The same argument applies to p_2, and thus $\pi_{20}^{(2)}, \pi_{11}^{(2)} \neq 0$. Lastly we have that

$$-\pi_{11}^{(1)} x^* + \pi_{20}^{(1)} = 0 = -\pi_{11}^{(2)} x^* + \pi_{20}^{(2)} \quad \Longrightarrow \quad \frac{\pi_{20}^{(1)}}{\pi_{11}^{(1)}} = \frac{\pi_{20}^{(2)}}{\pi_{11}^{(2)}},$$

which corresponds to condition (ii), (a).

(b) Now assume that $p_1 \neq 0$ and $p_2 = 0$. Then $\pi_{20}^{(1)}, \pi_{11}^{(1)} \neq 0$, using the same arguments as in the previous subcase, and $\pi_{20}^{(2)}, \pi_{11}^{(2)} = 0$, since p_2 is the zero polynomial. This corresponds to condition (ii), (b).

(c) Lastly, if $p_1 = 0$ and $p_2 \neq 0$, then $\pi_{20}^{(1)}, \pi_{11}^{(1)} = 0$ and $\pi_{20}^{(2)}, \pi_{11}^{(2)} \neq 0$, analogously to the previous subcase. This corresponds to condition (ii), (c).

Note that the case that $p_1 = p_2 = 0$ cannot occur since in this case g would be affine, which we have excluded by assumption.
$\Longleftarrow$: Let $\pi_{02}^{(1)} = \pi_{02}^{(2)} \neq 0$. We distinguish the following three subcases.

(a) We first assume that the additional conditions in (ii) (a) hold and define

$$x^* := \frac{\pi_{20}^{(1)}}{\pi_{11}^{(1)}} = \frac{\pi_{20}^{(2)}}{\pi_{11}^{(2)}}, \quad y^* := 0.$$

Then clearly, x^* is a root of both p_1 and p_2 and $y^* = 0$ is a root of both q_1 and q_2. Hence, using Proposition 3.10, $\mathbf{T} := \begin{pmatrix} 1 & 0 \\ x^* & 1 \end{pmatrix}$, which satisfies $\det \mathbf{T} = 1 \neq 0$, solves Problem 1.2, and by Theorem 3.15 it is the only solution up to equivalence.

(b) Assuming that the additional conditions in (ii) (b) hold, we have that p_2 and q_2 are the zero polynomial. We define

$$x^* := \frac{\pi_{20}^{(1)}}{\pi_{11}^{(1)}} \neq 0, \quad y^* := 0.$$

Then x^* and y^* are roots of p_1 and q_1, respectively. Additionally, x^* is by default a root of the zero polynomial p_2 and y^* is a root of the zero polynomial q_2. Thus, the same $\mathbf{T}$ as in (a) solves Problem 1.2 uniquely up to equivalence.

(c) An analogous argument holds for case (ii) (c) where p_1 and q_1 are the zero polynomials and

$$x^* := \frac{\pi_{20}^{(2)}}{\pi_{11}^{(2)}} \neq 0, \quad y^* := 0.$$

Ad (iii): The proof works analogously to the one for (ii).

Ad (iv): Since g is not affine, at least one of p_1, p_2 is not the zero polynomial.

$\Longrightarrow$: Let $\mathbf{T}_1 = \begin{pmatrix} 1 & t_{12} \\ t_{21} & 1 \end{pmatrix}$ with $t_{12} \cdot t_{21} \notin \{0, 1\}$ be a solution to Problem 1.2. We again distinguish the following three subcases:

(a) Assume that $p_1, p_2 \neq 0$. $x^* := t_{21} \neq 0, y^* := t_{12} \neq 0$ are non-zero roots of p_1, p_2 and q_1, q_2, respectively, hence we have that $\pi_{02}^{(1)}, \pi_{20}^{(1)}, \pi_{02}^{(2)}, \pi_{20}^{(2)} \neq 0$. Furthermore, using Lemma 3.14 and the same arguments as in Case 1 of the proof of Theorem 3.15, we have that x^* and $\frac{1}{y^*}$ are two distinct common roots of p_1 and p_2. This implies that the discriminant of p_1 (and the one of p_2) is non zero, i.e., $\left(\pi_{11}^{(1)}\right)^2 - 4\pi_{20}^{(1)}\pi_{02}^{(1)} \neq 0$ (and $\left(\pi_{11}^{(2)}\right)^2 - 4\pi_{20}^{(2)}\pi_{02}^{(2)} \neq 0$). This allows us to factorize the polynomials p_1 and p_2 into

$$p_1(x) = \pi_{02}^{(1)} x^2 - \pi_{11}^{(1)} x + \pi_{20}^{(1)} = \pi_{02}^{(1)} \left(x - x^*\right)\left(x - \frac{1}{y^*}\right),$$
$$p_2(x) = \pi_{02}^{(2)} x^2 - \pi_{11}^{(2)} x + \pi_{20}^{(2)} = \pi_{02}^{(2)} \left(x - x^*\right)\left(x - \frac{1}{y^*}\right).$$

Equating the coefficients in p_1 and p_2, respectively, yields

$$\pi_{02}^{(1)}\left(x^* + \frac{1}{y^*}\right) = \pi_{11}^{(1)}, \quad \pi_{02}^{(1)}\frac{x^*}{y^*} = \pi_{20}^{(1)}, \text{ and } \quad \pi_{02}^{(2)}\left(x^* + \frac{1}{y^*}\right) = \pi_{11}^{(2)}, \quad \pi_{02}^{(2)}\frac{x^*}{y^*} = \pi_{20}^{(2)}.$$

We rearrange this and get:

$$x^* + \frac{1}{y^*} = \frac{\pi_{11}^{(1)}}{\pi_{02}^{(1)}} = \frac{\pi_{11}^{(2)}}{\pi_{02}^{(2)}}, \text{ and } \quad \frac{x^*}{y^*} = \frac{\pi_{20}^{(1)}}{\pi_{02}^{(1)}} = \frac{\pi_{20}^{(2)}}{\pi_{02}^{(2)}}.$$

All together, this corresponds to condition (iv), (a). Note that the same arguments can also be conducted for the polynomials q_1 and q_2. This results in the following equivalent equalities:

$$\frac{\pi_{11}^{(1)}}{\pi_{20}^{(1)}} = \frac{\pi_{11}^{(2)}}{\pi_{20}^{(2)}}, \text{ and } \quad \frac{\pi_{02}^{(1)}}{\pi_{20}^{(1)}} = \frac{\pi_{02}^{(2)}}{\pi_{20}^{(2)}}.$$

(b) We now assume that $p_1 \neq 0$ and $p_2 = 0$. This implies $q_1 \neq 0$ as well as $\pi_{02}^{(2)}, \pi_{11}^{(2)}, \pi_{20}^{(2)} = 0$. Since $x^* := t_{21}$ and $y^* := t_{12}$ are non-zero roots of $p_1 \neq 0$ and $q_1 \neq 0$, respectively, there holds $\deg(p_1), \deg(q_1) \geq 1$. Using Lemma 3.14, we have that $\frac{1}{y^*}$ is also a root of p_1. This root cannot be equal to x^* in view of the assumption that $\mathbf{T}_1$ is regular, i.e., $t_{12} \cdot t_{21} = y^* \cdot x^* \neq 1$. Hence p_1 has two distinct, non-zero roots which implies $\pi_{02}^{(1)}, \pi_{20}^{(1)} \neq 0$. Furthermore, the discriminant of p_1 is non-zero which corresponds to condition (iv), (b).

(c) The case $p_1 = 0$ and $p_2 \neq 0$ is analogous to the previous subcase and implies the condition (iv), (c).

$\Longleftarrow$: We distinguish the same three cases as before and show each time that $\mathbf{T}_1 = \begin{pmatrix} 1 & t_{12} \\ t_{21} & 1 \end{pmatrix}$ with $t_{12} \cdot t_{21} \notin \{0, 1\}$ will be a solution to Problem 1.2 under the given conditions.

(a) For the inverse direction, assume that condition (iv), (a) holds. Using the quadratic formula, we can compute the roots of p_1 and q_1:

$$x = \frac{\pi_{11}^{(1)} \pm \sqrt{\left(\pi_{11}^{(1)}\right)^2 - 4\pi_{20}^{(1)}\pi_{02}^{(1)}}}{2\pi_{02}^{(1)}}, \quad \text{and} \quad y = \frac{\pi_{11}^{(1)} \pm \sqrt{\left(\pi_{11}^{(1)}\right)^2 - 4\pi_{20}^{(1)}\pi_{02}^{(1)}}}{2\pi_{20}^{(1)}}.$$

Let x^+ be the root where the plus sign is used in the formula and x^- the one where the minus sign is used, and define y^+ and y^- analogously. Using the equalities in condition (iv), (a) we can rewrite these roots as

$$x^+ = \frac{\pi_{11}^{(1)} + \sqrt{\left(\pi_{11}^{(1)}\right)^2 - 4\pi_{20}^{(1)}\pi_{02}^{(1)}}}{2\pi_{02}^{(1)}} = \frac{1}{2}\left(\frac{\pi_{11}^{(1)}}{\pi_{02}^{(1)}} + \sqrt{\frac{\left(\pi_{11}^{(1)}\right)^2}{\left(\pi_{02}^{(1)}\right)^2} - \frac{4\pi_{20}^{(1)}}{\pi_{02}^{(1)}}}\right)$$
$$= \frac{1}{2}\left(\frac{\pi_{11}^{(2)}}{\pi_{02}^{(2)}} + \sqrt{\frac{\left(\pi_{11}^{(2)}\right)^2}{\left(\pi_{02}^{(2)}\right)^2} - \frac{4\pi_{20}^{(2)}}{\pi_{02}^{(2)}}}\right) = \frac{\pi_{11}^{(2)} + \sqrt{\left(\pi_{11}^{(2)}\right)^2 - 4\pi_{20}^{(2)}\pi_{02}^{(2)}}}{2\pi_{02}^{(2)}},$$

and analogously for x^-. The same holds for y^+ and y^- using the equivalence:

$$\frac{\pi_{11}^{(1)}}{\pi_{02}^{(1)}} = \frac{\pi_{11}^{(2)}}{\pi_{02}^{(2)}} \quad\wedge\quad \frac{\pi_{20}^{(1)}}{\pi_{02}^{(1)}} = \frac{\pi_{20}^{(2)}}{\pi_{02}^{(2)}} \quad\Longleftrightarrow\quad \frac{\pi_{11}^{(1)}}{\pi_{20}^{(1)}} = \frac{\pi_{11}^{(2)}}{\pi_{20}^{(2)}} \quad\wedge\quad \frac{\pi_{02}^{(1)}}{\pi_{20}^{(1)}} = \frac{\pi_{02}^{(2)}}{\pi_{20}^{(2)}}.$$

Thus, x^+ and x^- are common roots of p_1 and p_2, and y^+ and y^- are common roots of q_1 and q_2. Since the discriminant of p_1 is non zero by assumption, we have that x^+ and x^- are distinct, and analogously for y^+ and y^-. A quick calculation shows that $x^+y^- = 1$ and hence $x^+ = \frac{1}{y^-}$. By Lemma 3.14 we thus also have that $x^- = \frac{1}{y^+}$. Since $x^+ \neq x^-$, this implies that $x^+y^+ \neq 1$.

Thus, by Proposition 3.10 and Theorem 3.15,

$$\mathbf{T} = \begin{pmatrix} 1 & y^+ \\ x^+ & 1 \end{pmatrix}$$

solves Problem 1.2 uniquely up to equivalence. Note that also $x^-y^- \neq 1$, and thus we could have also used those in the definition of $\mathbf{T}$. This would have resulted in an equivalent solution.

(b) Now we assume $p_1 \neq 0$ and $p_2 = 0$. This case can be shown analogously to the previous subcase. The only difference is that p_2 and q_2 are zero polynomials. Thus x^+ and y^+ are common roots by default.

(c) The case $p_1 = 0$ and $p_2 \neq 0$ is analogous to the previous one. □

Remark 3.18. In Theorem 3.17, the conditions in (i) to (iii) guarantee that $x^*, y^* \in \mathbb{R}$ if g has real coefficients. However, this does not necessarily hold true in case (iv). Here, any solution $\mathbf{T}$ will be in $\mathbb{R}^{2\times 2}$ only if the discriminants are positive, i.e., if

$$\left(\pi_{11}^{(1)}\right)^2 - 4\pi_{20}^{(1)}\pi_{02}^{(1)} > 0, \quad\text{and}\quad \left(\pi_{11}^{(2)}\right)^2 - 4\pi_{20}^{(2)}\pi_{02}^{(2)} > 0. \tag{16}$$

We now have two results characterizing the existence and uniqueness of solutions of Problem 1.2 for models with two variables. Theorem 3.15 states that any solution is unique up to equivalence and lies in one of four equivalence classes. Theorem 3.17 gives necessary and sufficient conditions for the existence of a solution. Furthermore, its proof also states how such a solution can be found in case it exists. Note that all the results exclude the case where g is already affine. In this case the problem is trivial and any invertible matrix is a solution.

4. The n-variate multilinearization problem

In this section, we analyse models with n variables for $n \geq 3$. In Section 4.1, we discuss the limitations of extending the direct approach as developed in Section 3 to higher dimensions. In Section 4.2, we examine special classes of polynomial functions for which we can apply the results from the two-variable setting to all or only a subset of variables.

4.1. General n-variate models

Our analysis of Problem 1.2 in the case $n = 2$ exploited the fact that the problem can be reduced to finding (common) roots of systems of (quadratic) polynomial equations. In this section, we will try to explore the same approach for general $n \in \mathbb{N}$. We will see that this quickly becomes infeasible as the number of nonlinear equations grows rapidly with n.

Throughout this section, for $n \in \mathbb{N}$, let $g : \mathbb{R}^n \to \mathbb{R}^n$ be a polynomial and $f : \mathbb{R}^n \to \mathbb{R}^n$ a multilinear function. We begin by stating results similar to the derivation of (8) in the previous section, i.e., we try to gain insight into the structure of a polynomial function g that is given by $g(\mathbf{x}) = \mathbf{T}^{-1}f(\mathbf{T}\mathbf{x})$.

Lemma 4.1. *Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be a multilinear function and let $\mathbf{T} \in \mathbb{R}^{n\times n}$ be an invertible matrix. Then the polynomial function $g : \mathbb{R}^n \to \mathbb{R}^n$ defined as*

$$g(\mathbf{x}) := \mathbf{T}^{-1}f(\mathbf{T}\mathbf{x})$$

has degree of at most n.

Table 1
Comparison of the numbers of equations and unknowns for different values of n.

n	#{coeffs. in g_i} $\binom{2n}{n}$	#{coeffs. in f_i} 2^n	#{nonlin. equ.} $n\left(\binom{2n}{n}-2^n\right)$	#{coeffs. in $\mathbf{T}$} n^2
2	6	4	4	4
3	20	8	36	9
4	70	16	216	16
5	252	32	1100	25

Proof. Let $\mathbf{T} \in \mathbb{R}^{n\times n}$ be an invertible matrix and $f : \mathbb{R}^n \to \mathbb{R}^n$ be a multilinear function with component functions f_i, $i \in \{1,\dots,n\}$. A monomial $m(\mathbf{x})$ in f_i has the form $m(\mathbf{x}) = \varphi \prod_{j\in J} x_j$ where $\varphi \in \mathbb{R}$ is a coefficient and $J \subseteq \{1,\dots,n\}$ is a subset of indices. The transformed monomial in $f(\mathbf{Tx})$ is given by $m(\mathbf{Tx}) = \varphi \prod_{j\in J} (\mathbf{Tx})_j$. Since for polynomials p and q it holds that

$$\deg(p\cdot q) = \deg(p) + \deg(q),$$

the degree of $m(\mathbf{Tx})$ is the same as the degree of $m(\mathbf{x})$ which is $|J| \le n$. Thus, all monomials in $f(\mathbf{Tx})$ have degree at most n, hence also all monomials in $g(\mathbf{x}) = \mathbf{T}^{-1} f(\mathbf{Tx})$ have degree at most n. □

Lemma 4.1 is a generalization of the result we obtained in the case $n = 2$. In order to be able to transform a polynomial function g into a multilinear function f in the way described in Problem 1.2 for $n \in \mathbb{N}$, g cannot have a degree larger than n.

In Section 3, we reduced the problem of finding a solution to Problem 1.2 to solving a system of polynomial equations. Since for $n = 2$ the only non-multilinear monomials were x_1^2 and x_2^2, we obtained two equations for each of the two components of g, i.e., a total of four equations. A similar approach can be taken for general $n \in \mathbb{N}$. To this end, we first determine how many monomials, and thus coefficients, there are in f and g.

Proposition 4.2. *Let $n \in \mathbb{N}$, $f : \mathbb{R}^n \to \mathbb{R}^n$ be a multilinear function of degree n and let $g : \mathbb{R}^n \to \mathbb{R}^n$ be a polynomial function of degree n. Then the number of coefficients in f is equal to $n2^n$ and the number of coefficients in g is equal to $n\binom{2n}{n}$ (if zero coefficients are counted as well).*

Proof. Each component of f is a linear combination of 2^n possible monomials since each variable x_i, $i \in \{1,\dots,n\}$, can either be present in a monomial or not. Since f has n components, the total number of coefficients in f is $n2^n$ (also counting zero coefficients).

For an $i \in \{1,\dots,n\}$, let g_i be the ith component of g which can be written as

$$g_i(\mathbf{x}) = \sum_{|\alpha|\le n} \pi_\alpha^{(i)} \mathbf{x}^\alpha$$

for some coefficients $\pi_\alpha^{(i)} \in \mathbb{R}$. The number of monomials in n variables of degree less than or equal to k is the same as the number of monomials in $n+1$ variables of degree equal to k in view of the bijective mapping

$$\left\{ \begin{array}{l} (n+1)\text{-variate polynomials} \\ \text{of degree equal to } k \end{array} \right\} \to \left\{ \begin{array}{l} n\text{-variate polynomials} \\ \text{of degree at most } k \end{array} \right\},$$
$$x_1^{\alpha_1} \cdots x_n^{\alpha_n} x_{n+1}^{\alpha_{n+1}} \mapsto x_1^{\alpha_1} \cdots x_n^{\alpha_n}.$$

The number of monomials in $n+1$ variable of degree equal to k corresponds to the number of ways to choose k elements from the set $\{1,\dots,n+1\}$ with repetitions being allowed and is hence equal to $\binom{n+k}{k}$ (see e.g. [7, page 71]). Hence the total number of monomials in g_i follows from setting $k = n$ and is given by $\binom{2n}{n}$. Since g has n components, the total number of monomials (and thus also the number of coefficients) in g is $n\binom{2n}{n}$. □

This means that there are $n\left(\binom{2n}{n} - 2^n\right)$ coefficients in $T \circ g \circ T^{-1}$ that have to be set to zero, resulting in as many nonlinear equations that have to be solved simultaneously in order to obtain the n^2 unknown entries of $\mathbf{T}$ (which could be reduced to $n^2 - n$ unknown entries by using Remark 3.8 and setting $t_{ii} = 1$).

As Table 1 shows, the number of equations quickly outgrows the number of unknowns as n increases. In fact, the number of equations grows exponentially in n which follows from the so-called *Stirling's formula* (see [8, Appendix A]) This shows that our approach in Section 3, i.e., writing out the equations explicitly, quickly becomes infeasible for larger n.

4.2. N-variate multilinearization by block diagonal transformations

In this subsection we will apply our results for the bivariate case from Section 3 to certain n-variate polynomial functions $g : \mathbb{R}^n \to \mathbb{R}^n$ for $n > 2$. In particular, we consider polynomial functions g that arise from transforming multilinear functions $f : \mathbb{R}^n \to \mathbb{R}^n$ with matrices that are block diagonal with two-by-two blocks. The goal is now to recover a transformation as well as a multilinear function when only g is given, along with the knowledge that pairwise transformations involving only the two consecutive variables x_{2j-1}, x_{2j} will lead to success.

More general, we will then also allow for permuted block diagonal transformation matrices with two-by-two blocks. Here, only g is given, along with the knowledge that pairwise transformations will lead to success but suitable sets of pairs first have to be identified.

Finally, we also consider functions f that are multilinear in only a subset of the variables. We will see that for all of these cases, we can apply Theorem 3.17 to certain restrictions of g to reconstruct the blocks of a (permuted) block-diagonal transformation matrix. We will motivate our approach by the following example.

Example 4.3. Let $g : \mathbb{R}^3 \to \mathbb{R}^3$ be the polynomial function defined by

$$g(\mathbf{x}) = \begin{pmatrix} 2x_1^2x_2 + x_1x_2x_3 + x_1 \\ x_2 + x_1 \\ -4x_1^2x_2 - 2x_1x_2x_3 + x_3 \end{pmatrix} \quad \text{for } \mathbf{x} \in \mathbb{R}^3.$$

The function g is (multi-)linear with respect to the variables x_2, x_3 but not for x_1. We will restrict g to its two component functions g_1, g_3 and consider it as a function of only the variables x_1 and x_3, i.e., x_2 is considered to be a constant. Then we have

$$g_{1,3}(x_1, x_3) := \begin{pmatrix} g_1(x_1, x_3) \\ g_3(x_1, x_3) \end{pmatrix} = \begin{pmatrix} 2x_1^2x_2 + x_1x_2x_3 + x_1 \\ -4x_1^2x_2 - 2x_1x_2x_3 + x_3 \end{pmatrix}.$$

The polynomials in Proposition 3.10 that result from the coefficients of $g_{1,3}$ then have the form (with x_2 being a constant)

$$\begin{aligned} p_1(x) &= 0x^2 - x_2x + 2x_2 = (-x+2)\cdot x_2, \\ p_2(x) &= 0x^2 + 2x_2x - 4x_2 = (x-2)\cdot 2x_2, \\ q_1(x) &= 2x_2x^2 - 4x_2x + 0 = x(2x-1)\cdot x_2, \\ q_2(x) &= -4x_2x^2 + 2x_2x + 0 = x(-2x+1)\cdot 2x_2. \end{aligned}$$

For arbitrary x_2, p_1 and p_2 have the common root $x^* = 2$ whereas q_1 and q_2 have the common roots $y_1^* = 0$ and $y_2^* = \frac{1}{2}$. By Proposition 3.10, the regular matrix $\mathbf{T}_{1,3} := \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$ transforms $g_{1,3}$ into a function that is multilinear in its variables x_1, x_3:

$$\begin{aligned} \mathbf{T}_{1,3}g_{1,3}(\mathbf{T}_{1,3}^{-1}(x_1, x_3)) &= \mathbf{T}_{1,3}g_{1,3}(x_1, -2x_1 + x_3) \\ &= \mathbf{T}_{1,3}\begin{pmatrix} 2x_1^2x_2 + x_1x_2(-2x_1 + x_3) + x_1 \\ -4x_1^2x_2 - 2x_1x_2(-2x_1 + x_3) + (-2x_1 + x_3) \end{pmatrix} \\ &= \mathbf{T}_{1,3}\begin{pmatrix} x_1x_2x_3 + x_1 \\ -2x_1x_2x_3 - 2x_1 + x_3 \end{pmatrix} \\ &= \begin{pmatrix} x_1x_2x_3 + x_1 \\ x_3 \end{pmatrix}. \end{aligned}$$

By defining $\mathbf{T}$ as a permuted block diagonal matrix with the above $\mathbf{T}_{1,3}$ as a block,

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \left(\begin{array}{cc|c} 1 & 0 & 0 \\ 2 & 1 & 0 \\ \hline 0 & 0 & 1 \end{array}\right) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix},$$

the transformed function

$$f(\mathbf{x}) = \mathbf{T}g(\mathbf{T}^{-1}\mathbf{x}) = \begin{pmatrix} x_1x_2x_3 + x_1 \\ x_2 + x_1 \\ x_3 \end{pmatrix}$$

is multilinear with respect to all its variables.

To formalize the approach used in Example 4.3, we introduce the following notation and terminology.

Definition 4.4. For $n \in \mathbb{N}$, $i, j \in \{1, \dots, n\}$ with $i \neq j$ and a polynomial function $g : \mathbb{R}^n \to \mathbb{R}^n$, we denote by $g^{(i,j)}$ the function g considered as a function of only the variables x_i and x_j with all other variables being fixed parameters. Thus, $g^{(i,j)}$ is a polynomial whose coefficients are polynomials in the other variables, i.e.,

$$g^{(i,j)} : \mathbb{R}^2 \to (\mathbb{R}[x_1, \dots, \cancel{x_i}, \dots, \cancel{x_j}, \dots, x_n])^n$$

where the notation $\cancel{x_i}$ indicates that the variable x_i is omitted. Furthermore, for two distinct indices $k, \ell \in \{1, \dots, n\}$, we denote by $g_{k,\ell}^{(i,j)}$ the restriction of $g^{(i,j)}$ to its two components $g_k^{(i,j)}$ and $g_\ell^{(i,j)}$, i.e.,

$$g_{k,\ell}^{(i,j)} : \mathbb{R}^2 \to (\mathbb{R}[x_1, \dots, \cancel{x_i}, \dots, \cancel{x_j}, \dots, x_n])^2, \quad (x_i, x_j) \mapsto \begin{pmatrix} g_k^{(i,j)}(x_i, x_j) \\ g_\ell^{(i,j)}(x_i, x_j) \end{pmatrix}.$$

We will call such a function $g_{k,\ell}^{(i,j)}$ the *restriction* of g to the kth and ℓth components and ith and jth variables.

For the remainder of this subsection, unless stated otherwise, let $m \in \mathbb{N}$ and let $f : \mathbb{R}^{2m} \to \mathbb{R}^{2m}$ be a multilinear function. Let $\mathbf{T} \in \mathbb{R}^{2m\times 2m}$ be an invertible block diagonal matrix with 2×2-blocks,

$$\mathbf{T} = \operatorname{diag}(\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_m) \quad \text{where } \mathbf{T}_i \in \mathbb{R}^{2\times 2} \text{ for } i = 1, \dots, m. \tag{17}$$

Furthermore, let the polynomial function $g : \mathbb{R}^{2m} \to \mathbb{R}^{2m}$ be defined by

$$g(\mathbf{x}) := \mathbf{T}^{-1} f(\mathbf{T}\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^{2m}. \tag{18}$$

The goal is to recover $\mathbf{T}$ (or any equivalent matrix) and f with only g being given.

We begin with the following proposition. It states that for a function g as defined in (18), the block $\mathbf{T}_j$ of $\mathbf{T}$ is a solution to Problem 1.2 for all restrictions $g^{(2j-1,2j)}_{k,\ell}$ of g for all $j \in \{1, \dots, m\}$.

Proposition 4.5. *Let $m \in \mathbb{N}$, let $\mathbf{T} \in \mathbb{R}^{2m\times 2m}$ be block-diagonal as in (17) and let $f, g : \mathbb{R}^{2m} \to \mathbb{R}^{2m}$ be functions where f is multilinear and g is defined as in (18). Then for all $j = 1, \dots, m$ and for all $k, \ell \in \{1, \dots, 2m\}$, the matrix block $\mathbf{T}_j \in \mathbb{R}^{2\times 2}$ solves Problem 1.2 for the functions $g^{(2j-1,2j)}_{k,\ell}$.*

Proof. Let $j \in \{1, \dots, m\}$ and let the matrix $\widetilde{\mathbf{T}}$ be defined by $\widetilde{\mathbf{T}} = \operatorname{diag}(\mathbf{T}_j, \dots, \mathbf{T}_j)$. We first note that

$$g \circ \widetilde{\mathbf{T}}^{-1} = \mathbf{T}^{-1} \circ f \circ (\mathbf{T}\widetilde{\mathbf{T}}^{-1}) = \mathbf{T}^{-1} \circ f \circ \operatorname{diag}\big(\mathbf{T}_1\mathbf{T}_j^{-1}, \dots, \underbrace{\mathbf{T}_j\mathbf{T}_j^{-1}}_{\mathbf{I}}, \dots, \mathbf{T}_m\mathbf{T}_j^{-1}\big)$$

is multilinear with respect to the variables x_{2j-1} and x_{2j} which implies that

$$g^{(2j-1,2j)}_{k,\ell} \circ \mathbf{T}_j^{-1}$$

and hence also $\mathbf{T}_j \circ g^{(2j-1,2j)}_{k,\ell} \circ \mathbf{T}_j^{-1}$ is multilinear with respect to the variables x_{2j-1} and x_{2j} for all $k, \ell \in \{1, \dots, 2m\}$. Thus $\mathbf{T}_j$ solves Problem 1.2 for all functions $g^{(2j-1,2j)}_{k,\ell}$. ☐

The next Proposition is fundamental to this section. It states that if we can find blocks $\hat{\mathbf{T}}_j$ for each pair of variables x_{2j-1}, x_{2j}, $j = 1, \dots, m$, then the block diagonal matrix $\hat{\mathbf{T}} = \operatorname{diag}(\hat{\mathbf{T}}_1, \dots, \hat{\mathbf{T}}_m)$ solves Problem 1.2 for $g : \mathbb{R}^{2m} \to \mathbb{R}^{2m}$.

Proposition 4.6. *For $m \in \mathbb{N}$, let $\mathbf{T} \in \mathbb{R}^{2m\times 2m}$ be block-diagonal as in (17) and let $f, g : \mathbb{R}^{2m} \to \mathbb{R}^{2m}$ be functions with f being multilinear and g defined as in (18). For each $j \in \{1, \dots, m\}$, let $k_j, \ell_j \in \{1, \dots, 2m\}$ be indices such that $g^{(2j-1,2j)}_{k_j,\ell_j}$ is not affine, and let $\hat{\mathbf{T}}_j \in \mathbb{R}^{2\times 2}$ be a matrix that solves Problem 1.2 for this function. Then the block diagonal matrix $\hat{\mathbf{T}} = \operatorname{diag}(\hat{\mathbf{T}}_1, \dots, \hat{\mathbf{T}}_m)$ solves Problem 1.2 for g.*

Proof. For all $j \in \{1, \dots, m\}$, let $k_j, \ell_j \in \{1, \dots, 2m\}$ be such that $g^{(2j-1,2j)}_{k_j,\ell_j}$ is not affine and there exists a solution $\hat{\mathbf{T}}_j$ to Problem 1.2 for this function. Since both $\mathbf{T}_j$ and $\hat{\mathbf{T}}_j$ solve Problem 1.2 for the function $g^{(2j-1,2j)}_{k_j,\ell_j}$ which by assumption is not affine, Theorem 3.15 yields that $\mathbf{T}_j \sim \hat{\mathbf{T}}_j$. Hence for all $j \in \{1, \dots, m\}$, there exist scaled permutation matrices $\mathbf{M}_j \in \mathbb{R}^{2\times 2}$ such that $\mathbf{T}_j = \mathbf{M}_j\hat{\mathbf{T}}_j$. The block diagonal matrix $\mathbf{M} = \operatorname{diag}(\mathbf{M}_1, \dots, \mathbf{M}_m) \in \mathbb{R}^{2m\times 2m}$ is also a scaled permutation matrix, and for $\hat{\mathbf{T}} = \operatorname{diag}(\hat{\mathbf{T}}_1, \dots, \hat{\mathbf{T}}_m)$, there holds $\mathbf{T} = \mathbf{M}\hat{\mathbf{T}}$ and hence $\mathbf{T} \sim \hat{\mathbf{T}}$. Using again Remark 3.8 and the fact that $\mathbf{T}$ solves Problem 1.2 for g by construction, we conclude that $\hat{\mathbf{T}}$ also solves Problem 1.2 for g. ☐

Remark 4.7. In order to find solutions $\hat{\mathbf{T}}_j$ to Problem 1.2 for restrictions $g^{(2j-1,2j)}_{k_j,\ell_j} : \mathbb{R}^2 \to (\mathbb{R}[x_1, \dots, \cancel{x_i}, \dots, \cancel{x_j}, \dots, x_n])^2$, we can apply the results of Section 3 - in particular Proposition 3.10, Theorems 3.15 and 3.17. As in Example 4.3, the coefficients of the polynomials p_1, p_2 and q_1, q_2 will now be polynomials in $\mathbb{R}[x_1, \dots, \cancel{x_i}, \dots, \cancel{x_j}, \dots, x_n]$, and we need to be able to find common roots in $\mathbb{C}$ to obtain a solution $\hat{\mathbf{T}}_j$, i.e., roots that do not depend on polynomial terms in $\mathbb{R}[x_1, \dots, \cancel{x_i}, \dots, \cancel{x_j}, \dots, x_n]$. In particular, if no such common roots in $\mathbb{C}$ exist, then Problem 1.2 has no solution.

For a permutation matrix $\mathbf{P} \in \mathbb{R}^{2m\times 2m}$, we next consider the function

$$g := (\mathbf{P}\mathbf{T})^{-1} \circ f \circ (\mathbf{P}\mathbf{T}).$$

Then a matrix $\hat{\mathbf{T}} = \operatorname{diag}(\hat{\mathbf{T}}_1, \dots, \hat{\mathbf{T}}_m)$ as constructed in Proposition 4.6 once more solves Problem 1.2 for this g in view of $\mathbf{P}\mathbf{T} \sim \mathbf{T} \sim \hat{\mathbf{T}}$. In the following, we generalize this idea of permuting $\mathbf{T}$ further. Before we do so, we introduce the notion of a submatrix.

Definition 4.8 (*Submatrix*). Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a matrix for $m, n \in \mathbb{N}$. Let $I \subseteq \{1, \dots, m\}$ and $J \subseteq \{1, \dots, n\}$ be sets of indices. Then we define the submatrix $\mathbf{A}_{I,J} \in \mathbb{R}^{|I|\times|J|}$ of A by

$$\mathbf{A}_{I,J} = \left(a_{i,j}\right)_{i\in I, j\in J}.$$

We obtain the following result.

Theorem 4.9. *For $m \in \mathbb{N}$, let $\mathbf{T}$ be an invertible block-diagonal matrix as in (17), and let $f : \mathbb{R}^{2m} \to \mathbb{R}^{2m}$ be multilinear. Let $\mathbf{P} \in \mathbb{R}^{2m\times 2m}$ be a permutation matrix and let $g : \mathbb{R}^{2m} \to \mathbb{R}^{2m}$ be the polynomial function defined by*

$$g : \mathbb{R}^{2m} \to \mathbb{R}^{2m}, \quad g(\mathbf{x}) := \mathbf{P}\mathbf{T}^{-1} f(\mathbf{T}\mathbf{P}^T\mathbf{x}). \tag{19}$$

Assume that for all $j \in \{1, \dots, m\}$, there exist indices $k_j, \ell_j, \iota_j, \kappa_j \in \{1, \dots, 2m\}$ such that

- *the sets $\{\iota_j, \kappa_j\}$ form a partition of $\{1, \dots, 2m\}$,*
- *for all $j \in \{1, \dots, m\}$, there exists a matrix $\hat{\mathbf{T}}_j \in \mathbb{R}^{2\times 2}$ such that $\hat{\mathbf{T}}_j$ solves Problem 1.2 for $g^{(\iota_j,\kappa_j)}_{k_j,\ell_j}$, and*
- *the functions $g^{(\iota_j,\kappa_j)}_{k_j,\ell_j}$ are not affine.*

Let $\hat{\mathbf{T}} := \operatorname{diag}(\hat{\mathbf{T}}_1, \dots, \hat{\mathbf{T}}_m)$, and let the matrix $\widetilde{\mathbf{T}} \in \mathbb{R}^{2m\times 2m}$ have zero entries except for

$$\widetilde{\mathbf{T}}_{\{\iota_j,\kappa_j\},\{\iota_j,\kappa_j\}} := \hat{\mathbf{T}}_j, \qquad j = 1, \dots, m. \tag{20}$$

Then it holds that $\widetilde{\mathbf{T}} = \mathbf{P}\hat{\mathbf{T}}\mathbf{P}^T$, and $\widetilde{\mathbf{T}}$ solves Problem 1.2 for g. In particular, the matrix $\widetilde{\mathbf{T}}$ can be computed when only g is known, i.e., without explicit knowledge of $\mathbf{P}$.

Proof. Let $\sigma : \{1, \dots, 2m\} \to \{1, \dots, 2m\}$ be the permutation associated with the permutation matrix $\mathbf{P}$, i.e.,

$$\mathbf{P} = \left(e_{\sigma(1)}, e_{\sigma(2)}, \dots, e_{\sigma(2m)}\right)^T$$

where e_i is the ith standard basis vector in $\mathbb{R}^{2m}$. We define

$$\widetilde{g} : \mathbb{R}^{2m} \to \mathbb{R}^{2m}, \quad \widetilde{g}(\mathbf{x}) := \mathbf{P}^T g(\mathbf{P}\mathbf{x}).$$

In view of (19), there holds

$$\widetilde{g}(\mathbf{x}) = \mathbf{P}^T g(\mathbf{P}\mathbf{x}) = \mathbf{P}^T\mathbf{P}\mathbf{T}^{-1} f(\mathbf{T}\mathbf{P}^T\mathbf{P}\mathbf{x}) = \mathbf{T}^{-1} f(\mathbf{T}\mathbf{x}).$$

Hence, $\mathbf{T}$ solves Problem 1.2 for $\widetilde{g}$. By Proposition 4.5, for all $j = 1, \dots, m$, the matrix $\mathbf{T}_j$ solves Problem 1.2 for the restrictions $\widetilde{g}^{(2j-1,2j)}_{k,\ell}$ for all $k, \ell \in \{1, \dots, 2m\}$. Now let $\hat{\mathbf{T}}_j \in \mathbb{R}^{2\times 2}$ also be a solution of Problem 1.2 for the function $\widetilde{g}^{(2j-1,2j)}_{k,\ell}$. Since by assumption this function is not affine, Theorem 3.15 yields that $\mathbf{T}_j \sim \hat{\mathbf{T}}_j$, i.e., there exist scaled permutation matrices $\mathbf{M}_j \in \mathbb{R}^{2\times 2}$ such that $\mathbf{T}_j = \mathbf{M}_j\hat{\mathbf{T}}_j$. We define the block diagonal matrices $\mathbf{M} = \operatorname{diag}(\mathbf{M}_1, \dots, \mathbf{M}_m)$ and $\hat{\mathbf{T}} = \operatorname{diag}(\hat{\mathbf{T}}_1, \dots, \hat{\mathbf{T}}_m)$. Then $\mathbf{M} \in \mathbb{R}^{2m\times 2m}$ is also a scaled permutation matrix and $\mathbf{T} = \mathbf{M}\hat{\mathbf{T}}$, and thus $\mathbf{T} \sim \hat{\mathbf{T}}$. This implies that $\hat{\mathbf{T}}$ solves Problem 1.2 for $\widetilde{g}(\mathbf{x})$. In particular, we have that

$$\hat{\mathbf{T}} \circ \widetilde{g} \circ \hat{\mathbf{T}}^{-1} = (\hat{\mathbf{T}}\mathbf{P}^T) \circ g \circ (\mathbf{P}\hat{\mathbf{T}}^{-1})$$

is multilinear. Hence $\hat{\mathbf{T}}\mathbf{P}^T$, and by equivalence also $\mathbf{P}\hat{\mathbf{T}}\mathbf{P}^T$, solve Problem 1.2 for g.

The matrix $\widetilde{\mathbf{T}} := \mathbf{P}\hat{\mathbf{T}}\mathbf{P}^T$ can be computed without knowledge of the matrix $\mathbf{P}$ as follows. Since we have $\widetilde{g} = \mathbf{P}^T \circ g \circ \mathbf{P}$, there also holds $g = \mathbf{P} \circ \widetilde{g} \circ \mathbf{P}^T$. Thus $\hat{\mathbf{T}}_j$ solves Problem 1.2 for $\widetilde{g}^{(2j-1,2j)}_{k,\ell}$ if and only if $\hat{\mathbf{T}}_j$ solves Problem 1.2 for

$$g^{(\iota_j,\kappa_j)}_{\sigma(k),\sigma(\ell)} \quad \text{with } \iota_j := \sigma^{-1}(2j-1),\ \kappa_j := \sigma^{-1}(2j) \text{ for } j = 1, \dots, m.$$

Then, the ι_jth and the κ_jth columns of $\hat{\mathbf{T}}\mathbf{P}^T$ are given by the $(2j-1)$th and $(2j)$th columns of $\hat{\mathbf{T}}$. Similarly the ι_jth and the κ_jth rows of $\mathbf{P}\hat{\mathbf{T}}$ are given by the $(2j-1)$th and $(2j)$th rows of $\hat{\mathbf{T}}$. Hence, we have that

$$\big(\underbrace{\mathbf{P}\hat{\mathbf{T}}\mathbf{P}^T}_{\widetilde{\mathbf{T}}}\big)_{\{\iota_j,\kappa_j\},\{\iota_j,\kappa_j\}} = \hat{\mathbf{T}}_j, \qquad j = 1, \dots, m. \quad \square$$

Until now we have only considered the case where $n = 2$ m is even. Now let $f : \mathbb{R}^{2m+1} \to \mathbb{R}^{2m+1}$ be a multilinear function depending on an odd number of variables and let $\mathbf{T} \in \mathbb{R}^{(2m+1)\times(2m+1)}$ be an invertible block diagonal matrix with 2×2-blocks $\mathbf{T}_i \in \mathbb{R}^{2\times 2}$ for $i = 1, \dots, m$, and a single entry $t_{n,n} \in \mathbb{R} \setminus \{0\}$ for $n = 2m + 1$. We define the polynomial function

$$g : \mathbb{R}^{2m+1} \to \mathbb{R}^{2m+1}, \quad g(\mathbf{x}) := \mathbf{T}^{-1} f(\mathbf{T}\mathbf{x}).$$

Then we can apply our previous results by only considering the first $2m$ components of f and g. The last component of f and g is then transformed by a single non-zero scalar which does not affect multilinearity.

To conclude this section, we consider the case where f is multilinear not necessarily with respect to all but only to a subset of the variables. $\mathbf{T}$ is again assumed to be a block diagonal matrix. This relaxed problem is a generalization of Problem 1.2 and stated formally as follows.

Problem 4.10. Given a polynomial function $g : \mathbb{R}^n \to \mathbb{R}^n$, our goal is to find an invertible linear transformation $T : \mathbb{R}^n \to \mathbb{R}^n$ that is a permuted block-diagonal matrix with two-by-two and one-by-one blocks and a subset $I \subseteq \{1, \dots, n\}$ such that the function

$$f := T \circ g \circ T^{-1}$$

is multilinear with respect to all variables $\{x_j \mid j \in I\}$. If such a transformation T exists, it is called a solution to the relaxed multilinearization problem.

To analyse this problem, assume that there exists a subset of indices $I \subseteq \{1, \dots, n\}$ such that a polynomial function $f : \mathbb{R}^n \to \mathbb{R}^n$ is multilinear in the variables x_i, $i \in I$. Assume for simplicity that $I = \{1, \dots, m\}$ for $m \le n$, i.e., f is multilinear in the first m variables. Let $\mathbf{T}$ be an invertible block diagonal matrix of the form

$$\mathbf{T} = \operatorname{diag}(\mathbf{T}_1, \dots, \mathbf{T}_{\lfloor \frac{m}{2} \rfloor}, t_{\lfloor \frac{m}{2} \rfloor + 1}, \dots, t_n) \tag{21}$$

where $\mathbf{T}_i \in \mathbb{R}^{2\times 2}$ for $i = 1, \dots, \lfloor m/2 \rfloor$ and $t_j \in \mathbb{R} \setminus \{0\}$ for $j = \lfloor m/2 \rfloor + 1, \dots, n$. Moreover, let $\mathbf{P} \in \mathbb{R}^{n\times n}$ be a permutation matrix. Let $g : \mathbb{R}^n \to \mathbb{R}^n$ be defined by

$$g(\mathbf{x}) := \mathbf{P}\mathbf{T}^{-1} f(\mathbf{T}\mathbf{P}^T\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n. \tag{22}$$

Note that the assumption that f is multilinear in the first m variables is not a restriction since we can always permute the variables. We now generalize the results from Section 4.2 to this setting.

Theorem 4.11. *Let $m, n \in \mathbb{N}$, $m \le n$ and let f, $\mathbf{T}$, $\mathbf{P}$ as well as g be defined as above. Additionally, let $\sigma : \{1, \dots, n\} \to \{1, \dots, n\}$ be the permutation corresponding to $\mathbf{P}$. Assume that for all $j \in \{1, \dots, \left\lfloor \frac{m}{2} \right\rfloor\}$, there exist indices $k_j, \ell_j, \iota_j, \kappa_j \in \{1, \dots, n\}$ such that the sets $\{\iota_j, \kappa_j\}$ form a partition of $\sigma(\{1, \dots, m\})$ if m is even and of $\sigma(\{1, \dots, m-1\})$ if m is odd. Furthermore, assume that for all $j \in \{1, \dots, \left\lfloor \frac{m}{2} \right\rfloor\}$, there exist matrices $\hat{\mathbf{T}}_j \in \mathbb{R}^{2\times 2}$ such that $\hat{\mathbf{T}}_j$ solves Problem 1.2 for $g_{k_j,\ell_j}^{(\iota_j,\kappa_j)}$, and assume that these functions are not affine. Define the matrix $\widetilde{\mathbf{T}} \in \mathbb{R}^{n\times n}$ with zero entries except for*

$$\begin{aligned} \widetilde{\mathbf{T}}_{\{\iota_j,\kappa_j\},\{\iota_j,\kappa_j\}} &= \hat{\mathbf{T}}_j, \quad j = 1, \dots, \left\lfloor \frac{m}{2} \right\rfloor \\ t_{i,i} &= 1, \quad i \in \{1, \dots, n\} \setminus \{\iota_j, \kappa_j \mid j = 1, \dots, \left\lfloor \frac{m}{2} \right\rfloor\}. \end{aligned} \tag{23}$$

Then $\widetilde{\mathbf{T}}$ solves Problem 4.10 for g.

Proof. To simplify notation, we define $I := \{\iota_j, \kappa_j \mid j = 1, \dots, \lfloor m/2 \rfloor\}$ as well as the variables

$$\mathbf{y} = (y_1, \dots, y_{2\cdot\lfloor \frac{m}{2} \rfloor})^T := (x_{\iota_1}, x_{\kappa_1}, \dots, x_{\iota_{\lfloor \frac{m}{2} \rfloor}}, x_{\kappa_{\lfloor \frac{m}{2} \rfloor}})^T.$$

Consider the restrictions $g_{k_j,\ell_j}^{(\mathbf{y})} := g_{k_j,\ell_j}^{(\iota_1,\kappa_1,\dots,\iota_{\lfloor \frac{m}{2} \rfloor},\kappa_{\lfloor \frac{m}{2} \rfloor})}$ for $j = 1, \dots, \left\lfloor \frac{m}{2} \right\rfloor$ and define the function

$$\widetilde{g} : \mathbb{R}^{2\cdot\lfloor m/2 \rfloor} \to (\mathbb{R}[\mathbf{x}, \not{\mathbf{y}}])^{2\cdot\lfloor m/2 \rfloor}, \quad \widetilde{g}(\widetilde{\mathbf{y}}) := \left(g_{k_j,\ell_j}^{(\mathbf{y})} \right)_{j=1}^{\lfloor m/2 \rfloor} (\widetilde{\mathbf{y}})$$

where $(\mathbb{R}[\mathbf{x}, \not{\mathbf{y}}])$ denotes the set of polynomials in the variables x_i for $i \in \{1, \dots, n\} \setminus I$. Let $\widetilde{\mathbf{T}}$ be defined by (23). Since a restriction of a restriction is again a restriction, we can apply Theorem 4.9 to $\widetilde{g}$ and obtain that the matrix $\widetilde{\mathbf{T}}_{I,I} \in \mathbb{R}^{2\lfloor m/2 \rfloor \times 2\lfloor m/2 \rfloor}$ solves Problem 1.2 for $\widetilde{g}$. Thus, $\widetilde{g}$ is multilinear in the variables $\mathbf{y}$ which implies that

$$f(\mathbf{x}) = \widetilde{\mathbf{T}} g(\widetilde{\mathbf{T}}^{-1}\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n$$

is multilinear in the first m variables. Thus $\widetilde{\mathbf{T}}$ solves Problem 4.10 for g. $\square$

5. Algorithms for multilinearization of polynomial models

In this section, we present an algorithm for bivariate models in Section 5.1 and an algorithm that solves Problem 4.10 for the case where g is of the form (22) in Section 5.2. We furthermore analyse its computational complexity (Section 5.3) and present numerical experiments evaluating its runtime on randomly generated test cases (Section 5.4).

5.1. Algorithm for bivariate models

We first consider the case $n = 2$ where we can apply Theorem 3.17 to construct an algorithm that solves Problem 1.2. Let $g : \mathbb{R}^2 \to \mathbb{R}^2$ be a polynomial function. We know from the considerations in Section 3 that we can only find a transformation matrix $\mathbf{T}$ such that $f = \mathbf{T} \circ g \circ \mathbf{T}^{-1}$ is multilinear if g is of degree at most two. Hence, we assume that g can be written in the form (8). In order to decide whether there exists a transformation matrix $\mathbf{T}$ such that f is multilinear, we have to check whether the coefficients $\pi_\alpha^{(i)}$ for $i = 1, 2$ and $\alpha \in \{(2,0), (1,1), (0,2)\}$ satisfy any of the conditions given in Theorem 3.17. If this is the case, the Theorem tells us how to construct a respective transformation matrix $\mathbf{T}$. If the coefficients do not satisfy any of those conditions, Theorem 3.17 states that there does not exist a transformation matrix $\mathbf{T}$ such that $f(\mathbf{x}) = \mathbf{T}g(\mathbf{T}^{-1}\mathbf{x})$ is multilinear. We summarize this in Algorithm 1.

As pointed out in Remark 3.18, Algorithm 1 may return complex-valued matrices. If we require a real-valued solution, we also need to check whether the coefficients satisfy (16). Subsequently, we will use Algorithm 1 in an algorithm that solves Problem 4.10 for a polynomial function g of the form (22). As we discussed in Remark 4.7, we also have to ensure that the polynomials (12), (13) (with possibly polynomial coefficients) have real or complex roots. In practice, this can for example be done by symbolically evaluating the quadratic formula.

Algorithm 1 Algorithm for the Multilinearization of Bivariate Polynomial Models

Require: Polynomial function $g : \mathbb{R}^2 \to \mathbb{R}^2$ of the form (8)
Ensure: Transformation matrix $\mathbf{T} \in \mathbb{C}^{2\times 2}$ such that $f := \mathbf{T} \circ g \circ \mathbf{T}^{-1}$ is multilinear (if such a transformation exists)

```
if the coefficients π_α^(i), i = 1, 2, α ∈ {(2,0), (1,1), (0,2)}, satisfy any of the conditions (i)–(iv) from Theorem 3.17 then
    Calculate x* and y* according to the respective condition
    T := ( 1   x* )
         ( y*  1  )
else
    No such transformation matrix T exists
end if
```

5.2. Algorithm for n-variate models

We now turn our attention to Problem 4.10. Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be a polynomial function that is multilinear in a subset of variables x_i, $i \in I \subseteq \{1,\dots,n\}$. Let $m := |I|$ and let $\mathbf{T}$ be defined by (21) and let $\mathbf{P} \in \mathbb{R}^{n\times n}$ be a permutation matrix. Let $g : \mathbb{R}^n \to \mathbb{R}^n$ be defined by (22). Theorem 4.11 states that if we are able to find solutions $\hat{\mathbf{T}}_j$, $j = 1,\dots,\left\lfloor \frac{m}{2} \right\rfloor$, for the restrictions $g^{(\iota_j,\kappa_j)}_{k_j,\ell_j}$ of g, then the matrix $\tilde{\mathbf{T}}$ defined in (23) solves Problem 4.10. An algorithm could thus look like the following: For all pairs of variables $\{x_i, x_j\}$, $i, j = 1,\dots,n$ and $i \neq j$, check whether Algorithm 1 returns a solution $\hat{\mathbf{T}} \in \mathbb{R}^{2\times 2}$ for a restriction $g^{(i,j)}_{k,\ell}$ with some $k, \ell \in \{1,\dots,n\}$. If there does exist a solution, store it. If there does not exist a solution, move on to the next pair. Finally, construct the matrix $\tilde{\mathbf{T}} \in \mathbb{R}^{n\times n}$ using all previously stored $\hat{\mathbf{T}}$.

In practical problems, however, one does usually not know whether a given polynomial function g is of the form (22). Hence, we adapt the algorithm in the following way: For each pair of indices $\{i,j\} \subseteq \{1,\dots,n\}$, $i \neq j$, check whether for all pairs of indices $\{k,\ell\}$ in $\{1,\dots,n\}$, Algorithm 1 returns a solution $\hat{\mathbf{T}}$ for $g^{(i,j)}_{k,\ell}$. If this is the case, check whether, for the pair of variables x_i, x_j, all these matrices $\hat{\mathbf{T}}$ are equivalent or the identity matrix. Note that $\hat{\mathbf{T}}$ is the identity if $g^{(i,j)}_{k,l}$ only depends on x_i and x_j in an affine way. If they are, store $\hat{\mathbf{T}}$. If they are not, move on to the next pair of variables. Now, let $\hat{I}$ be the set consisting of all pairs of indices $\{i,j\}$ for which we found a matrix $\hat{\mathbf{T}}$. It is possible that the sets $\{i,j\} \in \hat{I}$ are not disjoint. In this case, we choose the largest subset $I \subseteq \hat{I}$ such that the sets $\{i,j\} \in I$ are disjoint. We summarize this in Algorithm 2.

5.3. Complexity analysis

Next, we analyse the computational complexities of the algorithms introduced in Sections 5.1 and 5.2. For each algorithm, we provide asymptotic bounds in terms of the model size $n \in \mathbb{N}$, and where applicable, we discuss the distinction between worst-case and average-case behaviour.

Polynomial representation and storage complexity

We first address a fundamental computational challenge that affects all subsequent analyses. By Proposition 4.2, a polynomial function $g : \mathbb{R}^n \to \mathbb{R}^n$ can have up to $\binom{2n}{n}$ monomials in each of its n components. This results in exponential storage complexity if we were to explicitly store all coefficients. Furthermore, operations that iterate through all monomials, such as coefficient extraction or polynomial evaluation, inherit this exponential complexity. This renders the algorithm impractical for even moderately sized problems. We hence restrict ourselves to polynomials with a "sparse" monomial structure, i.e., to those with only a small number of nonzero coefficients. In particular, our complexity analyses will assume that polynomial operations (e.g., coefficient access and extracting $g^{(i,j)}_{k,l}$ from g) can be performed efficiently. In the worst case, however, the exponential growth of polynomial terms in the polynomial degree dominates all other complexity considerations.

Complexity of Algorithm 1

Algorithm 1 determines whether a given bivariate polynomial function $g : \mathbb{R}^2 \to \mathbb{R}^2$ of the form (8) can be multilinearized by a linear transformation $\mathbf{T} \in \mathbb{R}^{2\times 2}$, i.e., whether Problem 1.2 is solvable. It does so by performing a fixed number of conditional checks and simple algebraic operations involving the coefficients of g. Thus, the algorithm has complexity $O(1)$.

Complexity of Algorithm 2

Algorithm 2 consists of two nested loops, each of length $\binom{n}{2}$, which may terminate early. However, the worst-case complexity is

$$\binom{n}{2} \cdot \binom{n}{2} = \left(\frac{n(n-1)}{2}\right)^2 = O(n^4).$$

In line 18, we choose the largest subset $I \subseteq \hat{I}$ such that the sets $\{i,j\} \in I$ are disjoint. This is related to the *maximum matching problem* in graph theory which can be solved in $O(|\hat{I}|^{2.5})$ [9] where $|\hat{I}|$ is the size of the set $\hat{I}$. The worst-case occurs when we have $|\hat{I}| = O(n^2)$, i.e., all pairs $\{i,j\}$ are valid, which results in complexity $O(n^5)$ for finding the maximum matching. The last step has complexity $O(n)$, hence the overall complexity of Algorithm 2 is, in the worst case, $O(n^5)$.

Algorithm 2 Partial Multilinearization of n-variate Polynomial Models

Require: $n \in \mathbb{N}$, Polynomial function $g : \mathbb{R}^n \to \mathbb{R}^n$
Ensure: Transformation matrix $\tilde{\mathbf{T}} \in \mathbb{C}^{n\times n}$ such that $f := \tilde{\mathbf{T}} \circ g \circ \tilde{\mathbf{T}}^{-1}$ is multilinear in a subset of variables x_I with $I \subseteq \{1,\dots,n\}$ (if such a matrix exists)

$\tilde{\mathbf{T}} := \mathbf{I}_n$; $\hat{I} := \emptyset$
for each subset $\{i,j\} \subseteq \{1,\dots,n\}$, $i \neq j$ **do**
  for each subset $\{k,\ell\} \subseteq \{1,\dots,n\}, k \neq \ell$ **do**
    if $\deg(g_{k,\ell}^{(i,j)}) \leq 2$ **then**
      Call Algorithm 1 for $g_{k,\ell}^{(i,j)}$
    else
      $\hat{I} \leftarrow \hat{I} \setminus \{\{i,j\}\}$
      break
    end if
    if Algorithm 1 returned a transformation matrix $\hat{\mathbf{T}}$ **then**
      if $\hat{\mathbf{T}} \neq \mathbf{I}_2$ **then**
        $\hat{\mathbf{T}}_{i,j} \leftarrow \hat{\mathbf{T}}$
        $\hat{I} \leftarrow \hat{I} \cup \{\{i,j\}\}$
      else
        continue
      end if
    else
      $\hat{I} \leftarrow \hat{I} \setminus \{\{i,j\}\}$
      break
    end if
  end for
end for
Choose the largest subset $I \subseteq \hat{I}$ such that the sets $\{i,j\} \in I$ are disjoint
for each $\{i,j\} \in I$ **do**
  Set $\tilde{\mathbf{T}}_{\{i,j\},\{i,j\}} \leftarrow \hat{\mathbf{T}}_{i,j}$
end for

However, the complexity of Algorithm 2 depends heavily on the structure of the input polynomial function g. Hence, making general statements about the average-case complexity is difficult. Under the two assumptions that

1. the average number $m \in \mathbb{N}$ of iterations of the inner loop before it is exited satisfies $m \ll n^2$, and
2. there holds $|\hat{I}| \leq n$,

the complexity of the two nested loops reduces to $O(n^2 m)$ and the complexity of the maximum matching reduces to $O(n^{2.5})$ leading to an overall complexity of $O(n^2(m + n^{0.5}))$. In the following section we evaluate the runtime of Algorithm 2 numerically. In the experiments we additionally try to work out the values of m and $|\hat{I}|$.

5.4. Numerical analysis

Runtime experiment setup

We consider the following two test scenarios to test Algorithm 2.

Scenario 1: We randomly generate polynomial functions $g : \mathbb{R}^n \to \mathbb{R}^n$ of degree less than or equal to n which we call the model size. It varies from 10 to 250 in increments of 20. For each n, we generate 10 polynomial functions as follows: For each component g_i of g, we cap the number of monomials with nonzero coefficients by $N_{\text{monom}} \in \{10, 50, 100, 500, 1000\}$. The coefficients of these monomials are chosen uniformly at random from $[-10, 10]$.

Scenario 2: We randomly generate polynomial functions in the form of (19). This is done by first generating random multilinear functions $f : \mathbb{R}^n \to \mathbb{R}^n$ in the same way as the polynomial functions in the first scenario but for $N_{\text{monom}} \in \{5, 10\}$ and a model size $n \in \{2, 4, 6, \dots, 20\}$. Subsequently, we transform these multilinear functions through (19) using randomly generated transformation matrices $\mathbf{T}$ of the form (17) and random permutation matrices. The individual blocks of $\mathbf{T}$ are chosen uniformly random from the set $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3, \mathbf{T}_4\}$ where the $\mathbf{T}_i$ are defined in (15). Their nonzero entries are chosen uniformly random from $[-5, 5]$.

The algorithm is implemented in Matlab R2025b, using the Symbolic Math Toolbox for symbolic computations which appear in Algorithm 1. The experiments were performed on a computer with an AMD Ryzen 7 PRO 4750U and 32 GB RAM. To speed up the experiments, the Parallel Computing Toolbox was used to parallelize the generation of test cases and the execution of the tests. For each of the experiments, we measured the runtime of Algorithm 2 as well as the sizes of the sets I and $\hat{I}$, i.e., the number of pairs of variables for which Algorithm 1 returned a solution. Furthermore, the average number of iterations m in the inner loop as well

Table 2
Comparison of the runtime, t_{restr}, t_{sym} and the number of iterations m performed in the inner loop of Algorithm 2 for selected values of n and N_{monom}. All values are averaged over the 10 experiments performed for each n and N_{monom} in Scenario 1.

N_{monom}	Runtime [s]	m	t_{restr}	t_{sym}
$n = 90$				
10	26.867	1.454	12.619	13.899
50	3.030	1.001	2.547	0.349
100	0.320	1.000	0.207	0.000
500	0.118	1.000	0.034	0.000
1000	0.136	1.000	0.045	0.000
$n = 210$				
10	372.374	2.365	153.955	214.602
50	44.021	1.026	35.676	7.500
100	12.182	1.001	11.166	0.424
500	0.965	1.000	0.306	0.000
1000	1.207	1.000	0.456	0.000

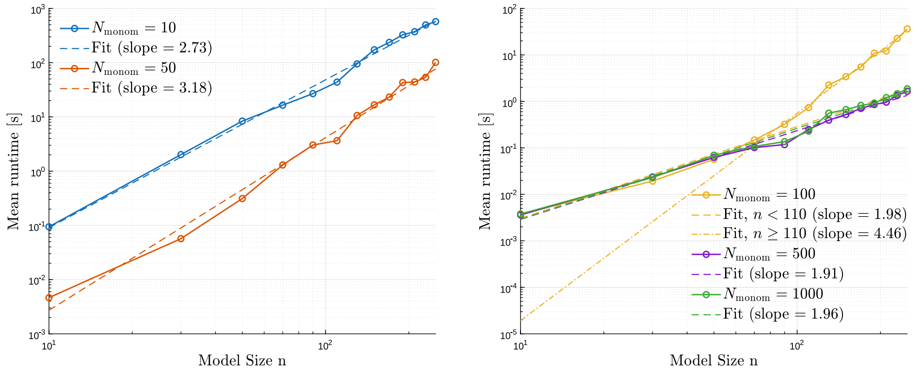

Fig. 1. Log–log scale runtime analysis of Algorithm 2 for randomly generated polynomial functions (Scenario 1) showing the relationship between the model size n and the execution time for $N_{\text{monom}} \in \{10, 50\}$ at the left and $N_{\text{monom}} \in \{100, 500, 1000\}$ at the right. For each N_{monom}, a linear model was fitted to the log–log transformed data indicating the exponent in the polynomial growth of the runtime. For $N_{\text{monom}} = 100$, the fit was split into two parts, one for the model sizes $n < 110$ the other for $n \geq 110$.

as the time spent on symbolic computations and the time spent on the construction of the restrictions $g_{i,j}^{k,\ell}$ of a given input g were measured. The latter two are denoted by t_{sym} and t_{restr}, respectively.

Runtime results

Fig. 1 displays the measured runtimes (averaged over 10 polynomials per model size) on a log–log scale for Scenario 1. In order to empirically assess the growth rate of the runtime with respect to the model size n, we fitted a linear model to the log–log transformed data. This linear model is depicted as a dashed line in the plots. Its slope indicates the exponent in the polynomial growth of the runtime. The results indicate that the empirical growth rate of the runtime is significantly lower than the worst-case bound of $O(n^5)$. In particular, for $N_{\text{monom}} \in \{10, 50\}$, the slope is approximately 3, indicating a cubic growth rate. For $N_{\text{monom}} \in \{500, 1000\}$, the runtime shows an almost quadratic growth rate. In the case $N_{\text{monom}} = 100$, we have a runtime of about $O(n^2)$ for $n < 110$ and $O(n^{4.5})$ for $n \geq 110$.

The observed runtimes of smaller values of N_{monom} are lower than those for larger values. This can be explained by the fact that as N_{monom} increases, the probability of a restriction $g_{k,l}^{(i,j)}$ containing terms of degree greater than two increases as well. This causes the inner loop to be exited early more frequently. Evidence for this can be found in Table 2. While m is small for all values of N_{monom}, it seems to tend towards 1 as N_{monom} grows. This leads to fewer calls of Algorithm 1 and thus a smaller overall runtime for larger N_{monom}. Additionally, line 4 of Algorithm 2 is implemented in such a way that if a restriction $g_{k,l}^{(i,j)}$ would contain a term of degree greater than two, the function $g_{k,l}^{(i,j)}$ itself will never be assembled, and the loop is exited early. This effect can also be seen in Table 2 where t_{restr} (the time spent on computing the restrictions) is smaller for larger N_{monom}.

Another observation is that the assumption that the construction of $g_{k,l}^{(i,j)}$ can be performed efficiently only seems to hold for small model sizes n. In fact, as Fig. 2 suggests, t_{restr} grows at least with $O(n^{2.5})$ and even faster for larger N_{monom}. Furthermore, the runtime of Algorithm 1 cannot be considered as $O(1)$ for such restriction functions. If that were the case, the time spend on symbolic computations t_{sym} would grow at most with $O(n^2 m) = O(n^2)$, since in this scenario we observe $m \approx 1$. However Fig. 2 shows t_{sym} to grow with at least $O(n^3)$ even up to $O(n^{4.6})$.

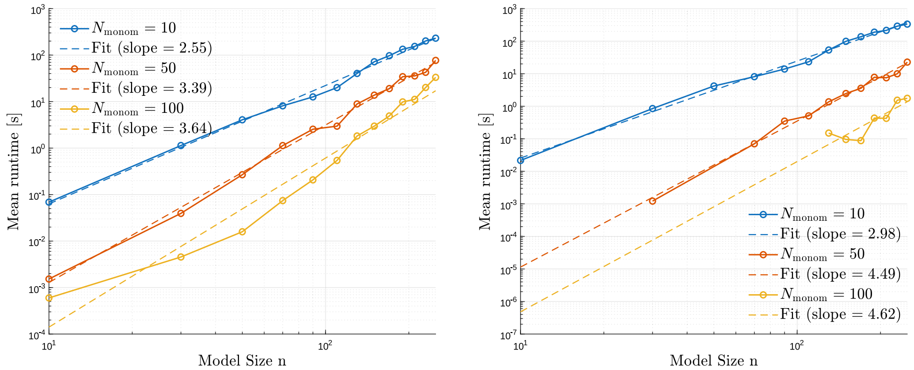

Fig. 2. Log–log scale runtime analysis of Algorithm 2 for randomly generated polynomial functions (Scenario 1) showing the relationship between the model size n and t_{restr} (left) as well as n and t_{sym} (right). For each $N_{\text{monom}} \in \{10, 50, 100\}$ a linear model was fitted to the log–log transformed data indicating the exponent in the polynomial growth of the runtime.

Table 3
Comparison of the number of iterations m performed in the inner loop, $|\hat{I}|$ and $|I|$ of Algorithm 2 for selected values of n and $N_{\text{monom}} \in \{5, 10\}$. All values are averaged over the 10 experiments performed for each n and N_{monom} in Scenario 2.

n	m	$\lvert\hat{I}\rvert$	$\lvert I\rvert$
$N_{\text{monom}} = 5$			
2	1.000	0.50	0.50
8	11.339	2.90	2.90
14	32.579	5.70	5.50
20	73.568	6.50	6.50
$N_{\text{monom}} = 10$			
2	1.000	0.60	0.60
8	13.400	2.70	2.70
14	34.064	4.90	4.90
20	52.753	7.30	7.30

This observation also explains why the runtimes grow with $O(n^{\alpha})$ with an $\alpha > 2.5$, for $N_{\text{monom}} \in \{10, 50, 100\}$. Even though our average case analysis suggests a runtime of $O(n^2 m) = O(n^2)$, we neglected the growing computational cost for symbolic computations and for computing $g_{k,l}^{(i,j)}$. For $N_{\text{monom}} = 100$ these additional costs come into play only for $n \geq 110$. It is at this point where the inner loop starts to not exit early. In this sense, $N_{\text{monom}} \in \{500, 1000\}$ are outliers as their runtime grows with $O(n^2)$. Here, the algorithm seems to exit the inner loop even before constructing $g_{k,l}^{(i,j)}$ most of the time, explaining the low values of t_{restr} and t_{sym} in Table 2. It is unclear whether this will change for even model sizes $n > 250$.

It is important to note that random polynomial functions as they were generated in Scenario 1 are highly unlikely to possess an algebraic structure such that they can be transformed into multilinear functions: Algorithm 2 was unsuccessful to find a solution to Problem 1.2 or Problem 4.10 for all tests in this first scenario.

The test cases in the Scenario 2 have been designed such that solutions exist for Problems 1.2 and 4.10 which Algorithm 2 should find. The average sizes of the sets I and $\hat{I}$ found by the algorithm can be seen in Table 3. Note that on average there holds $|I| < \frac{n}{2}$ since variables that already occur multilinearly in the input function are not added to $\hat{I}$. It can also be observed in Table 3 that the assumptions placed on m and $|\hat{I}|$, i.e., $m \ll n^2$ and $|\hat{I}| \leq n$, hold for all considered test cases.

Fig. 3 suggests that the runtimes in this scenario scale with $O(n^{3.2})$ whereas the number of inner iterations m seems to grow with $O(n^{1.8})$. This seems to be almost consistent with our average case analysis stating a runtime of $O(n^2 m)$ where we again neglected the cubic growth of t_{restr} and t_{sym} depicted in Fig. 4.

Lastly, note that neither for the first scenario nor the second one the maximum matching (line 23 of Algorithm 2) played a role in the runtime. This is because in the first scenario we have $|\hat{I}| = 0$ for all test cases and in the second scenario we have $|\hat{I}| \leq n$ and thus a complexity of $O(n^{2.5})$ for the maximum matching.

Numerical analysis of partial solutions

The general advantages of multilinear models have been studied in [1,2]. In the following, we analyse the numerical behaviour of solutions to Problem 4.10, i.e., models that can be partially multilinearized using Algorithm 2. In order to do this, we consider the following multilinear model that describes two Lorenz systems coupled by the term $\pm\alpha x_2 x_5$:

$$\dot{x}_1 = \sigma(x_2 - x_1), \qquad \dot{x}_4 = \sigma(x_5 - x_4),$$

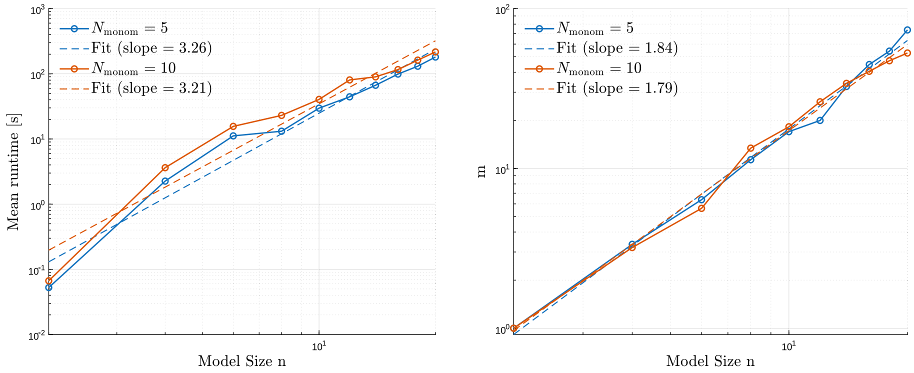

Fig. 3. Left: Log–log scale runtime analysis of Algorithm 2 for randomly generated polynomial functions of the form (19) (Scenario 2) showing the relationship between the model size n and the execution time for $N_{\text{monom}} \in \{5, 10\}$. For each N_{monom}, a linear model was fitted to the log–log transformed data indicating the exponent in the polynomial growth of the runtime. Right: Log–log scale analysis of the growth of m (inner iterations) with the model size.

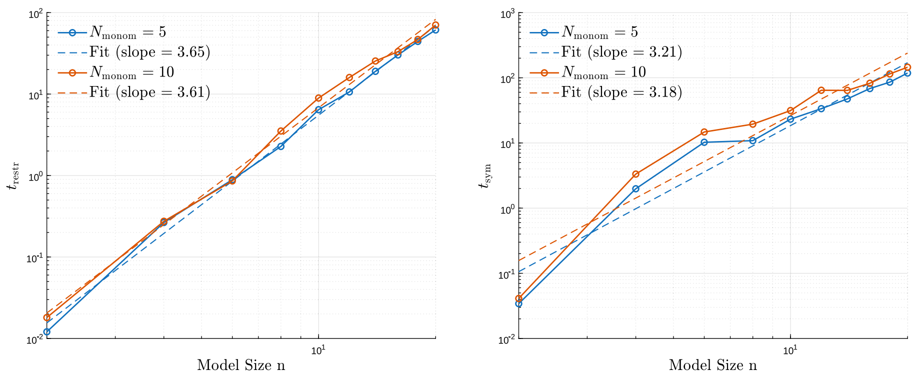

Fig. 4. Log–log scale runtime analysis of Algorithm 2 for randomly generated polynomial functions of the form (19) (Scenario 2) showing the relationship between the model size n and t_{restr} (left) as well as n and t_{sym} (right). For each $N_{\text{monom}} \in \{10, 50, 100\}$, a linear model was fitted to the log–log transformed data indicating the exponent in the polynomial growth of the runtime.

$$\begin{aligned} \dot{x}_2 &= x_1(\rho - x_3) - x_2 + \alpha x_2 x_5, & \dot{x}_5 &= x_4(\rho - x_6) - x_5 - \alpha x_2 x_5, \\ \dot{x}_3 &= x_1 x_2 - \beta x_3, & \dot{x}_6 &= x_4 x_5 - \beta x_6, \end{aligned}$$

with $\alpha = \frac{1}{2}$, $\beta = \frac{8}{3}$, $\rho = 14$, and $\sigma = 10$. This model is transformed into three different equivalent polynomial models. The first is obtained by linearly transforming the variables x_1 and x_3 using a matrix block $\mathbf{T}_{1,3}$ such that the model is polynomial (but no longer multilinear) in those variables, but still multilinear in the other variables. The second model is obtained by additionally transforming x_2 and x_5 by a matrix block $\mathbf{T}_{2,5}$ such that it is polynomial (but no longer multilinear) in all but the variables x_4 and x_6. The final model is obtained by also transforming the remaining x_4 and x_6 using a matrix block $\mathbf{T}_{4,6}$. The resulting model is then fully polynomial (i.e., not multilinear with respect to any variable). The matrix blocks are given by

$$\mathbf{T}_{1,3} = \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}, \quad \mathbf{T}_{2,5} = \begin{pmatrix} 1 & 3 \\ 1 & 1 \end{pmatrix}, \quad \text{and} \quad \mathbf{T}_{4,6} = \begin{pmatrix} 1 & 4 \\ 1 & 1 \end{pmatrix}.$$

These models are solved using the MATLAB MTI-Toolbox [10] on the interval $[0, 10]$, with the solutions of the (partial) polynomial models being transformed back using the inverse transformation matrices. In Fig. 5 we compare the error between the numerical solutions of each of the four models and a reference solution. The time taken to compute these numerical solutions can be seen in Table 4. The reference solution is computed using the purely multilinear representation of the Lorenz system and a relative tolerance of 10^{-12}.

We observe that, using a higher relative tolerance of 10^{-3}, the accuracy of the numerical solution decreases as the number of variables that appear non-multilinearly increases. However, this effect seems to vanish when using a lower relative tolerance of 10^{-5}. Nevertheless, Table 4 suggests that those models with fewer non-multilinear variables can be solved faster.

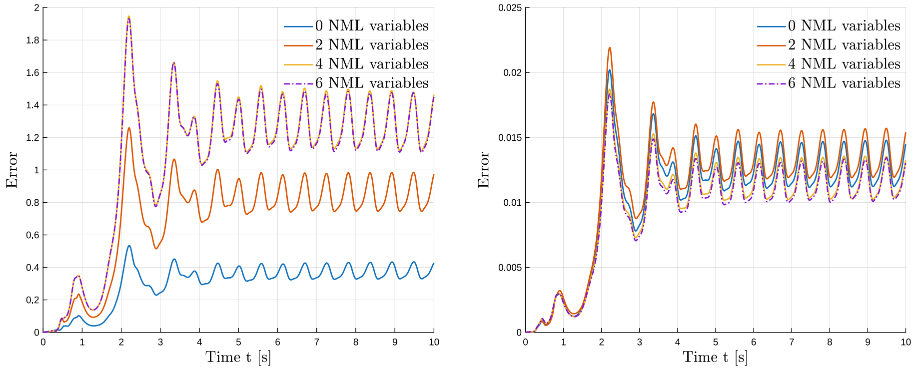

Fig. 5. The euclidean error at time $t \in [0, 10]$ between the trajectory of the reference solution and the trajectories of the four models with 0 to 6 non-multilinear (NML) variables and a relative tolerance of 10^{-3} (left) and 10^{-5} (right).

Table 4
Solve time [s] vs. number of NML variables.

RelTol	Number of NML variables			
	0	2	4	6
10^{-3}	0.214	0.256	0.339	0.422
10^{-5}	0.409	0.462	0.596	0.633

6. Conclusions

After formalizing the problem of multilinearizing polynomial models via linear state space transformations, we were able to fully classify the solutions to this problem for bivariate models. Moreover, we were able to give necessary and sufficient conditions for the existence of such solutions. For the general n-variate problem, we outlined some of the challenges involved. We were able to generalize the findings for bivariate models to models that arise from transforming multilinear functions with permuted block diagonal matrices with 2×2-blocks. Additionally, we looked at the relaxed problem where we only require multilinearity in a subset of variables. For this, we also derived an algorithm that solves the problem. Numerical experiments for both randomly generated polynomial functions and random block-transformed multilinear functions revealed that the worst-case runtime analysis of the algorithm of $O(n^5)$ seems too pessimistic. The results show a runtime of around $O(n^\alpha)$ with $\alpha \in [2, 4.5]$ for the different scenarios. However, we only considered randomly generated polynomials which – if not generated in such a way that solutions exist – are highly unlikely to possess the algebraic structure that allows them to be transformed into multilinear polynomials. Nevertheless, if such a transformation does exist, one can hope for a faster and more accurate numerical solution of the transformed system. Therefore, in order to increase the likelihood of finding such a transformation, it might be of interest to modify our Problem 1.2 and search for approximate instead of exact solutions. In view of Proposition 3.10, this could be achieved by constructing a regular transformation matrix $\mathbf{T} = \begin{pmatrix} 1 & t_{12} \\ t_{21} & 1 \end{pmatrix}$ such that its off-diagonal entries are not required to be equal but only close to roots of the respective polynomials p_i and q_i. This will be the subject of future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] Lichtenberg G, Pangalos G, Cateriano Yáñez C, Luxa A, Jöres N, Schnelle L, Kaufmann C. Implicit multilinear modeling. 2022, http://dx.doi.org/10.1515/auto-2021-0133.

[2] Kruppa K, Pangalos G, Lichtenberg G. Multilinear approximation of nonlinear state space models. IFAC Proc Vol 2014;47(3):9474–9. http://dx.doi.org/10.3182/20140824-6-ZA-1003.00455, 19th IFAC World Congress.

[3] Patarin J. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In: Advances in cryptology — EUROCRYPT '96. Berlin, Heidelberg: Springer Berlin Heidelberg; 1996, p. 33–48. http://dx.doi.org/10.1007/3-540-68339-9_4.
[4] Faugère J-C, Perret L. Polynomial equivalence problems: Algorithmic and theoretical aspects. In: Vaudenay S, editor. Advances in cryptology - EUROCRYPT 2006. Berlin, Heidelberg: Springer Berlin Heidelberg; 2006, p. 30–47. http://dx.doi.org/10.1007/11761679_3.
[5] Kayal N. Efficient algorithms for some special cases of the polynomial equivalence problem. In: Proceedings of the twenty-second annual ACM-SIAM symposium on discrete algorithms. SODA '11, USA: Society for Industrial and Applied Mathematics; 2011, http://dx.doi.org/10.1137/1.9781611973082.108.
[6] Cox DA, Little J, O'Shea D. Ideals, varieties, and algorithms. 5th ed. Cham: Springer; 2015, p. 71. http://dx.doi.org/10.1007/978-3-031-91841-4.
[7] Bóna M. A walk through combinatorics. 5th ed. Singapore: WORLD SCIENTIFIC; 2023, http://dx.doi.org/10.1142/13455.
[8] Flajolet P, Sedgewick R. Analytic combinatorics. Cambridge: Cambridge University Press; 2009, http://dx.doi.org/10.1017/CBO9780511801655.
[9] Galil Z. Efficient algorithms for finding maximum matching in graphs. ACM Comput Surv 1986;18(1). http://dx.doi.org/10.1145/6462.6502.
[10] Lichtenberg G, Uhlenberg E, et al. Multilinear time-invariant toolbox for MATLAB. 2025-10-18, http://mti.systems.