Canonical Polyadic Decomposition and Sets of Polynomial Equations

Lieven De Lathauwer
Joint work with Nithin Govindarajan and Raphaël Widdershoven

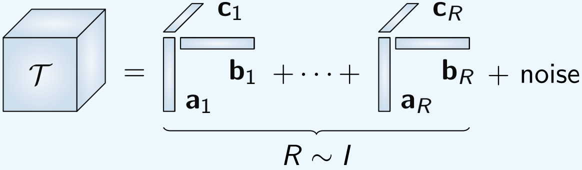
Back to the Roots seminar
Leuven, December 12th, 2023

KU LEUVEN
STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics
EOS THE EXCELLENCE OF SCIENCE
erc European Research Council Established by the European Commission

Overview

CPD in different communities

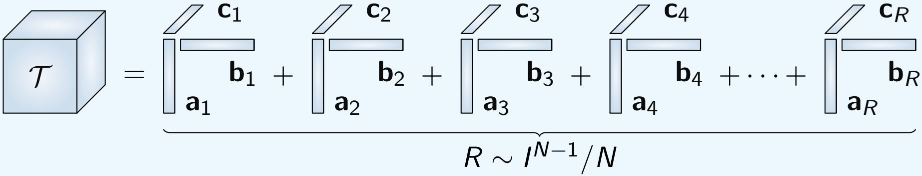
Signal processing/data analysis



- R small $\sim$ dimensions
- Noise: measurement/model error: often significant
- Approximation:
  - decomposition: pencil-based computation [Evert, Vandecappelle, et al. 2022]
  - then optimization-based refinement [Sorber, Van Barel, et al. 2013]
- Well-posed (within ball around exact solution) [Evert and De Lathauwer 2022]
- Use: uniqueness $\rightarrow$ finding data components/signal separation (basic tool!)

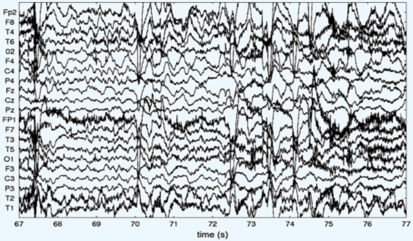
CPD in different communities

(Numerical) mathematics/computing

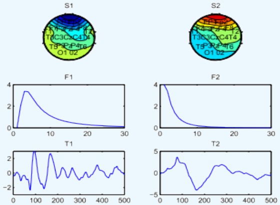


- Noise: numerical quantization error: typically small
- R possibly large $\sim$ generic rank
- Decomposition: NP-hard in general [Håstad 1990; Hillar and Lim 2013]
- Approximation: ill-posed in general [Kruskal, Harshman, et al. 1989; de Silva and Lim 2008]
- Use: numerical approximation (not orthogonal $\rightarrow$ MLSVD, TT/hT)

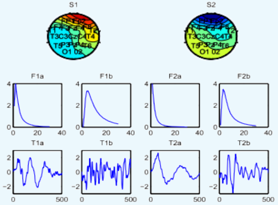
Application: detection epileptic seizure in EEG



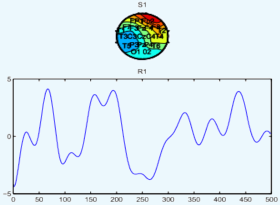
(a) Raw EEG



(b) CPD



(c) CWT-BTD



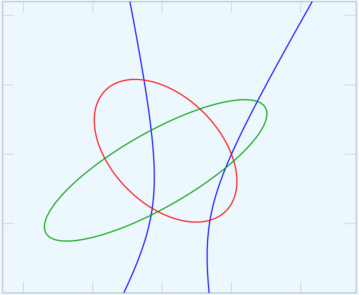
(d) H-BTD

[Hunyadi, Camps, et al. 2014; De Vos, Vergult, et al. 2007; Acar, Aykut-Bingol, et al. 2007]

Sets of polynomial equations in different communities

Signal processing/data analysis

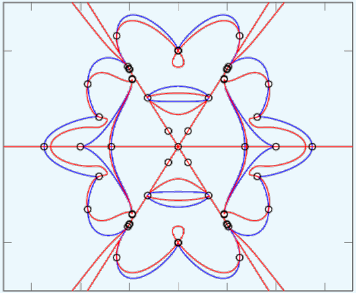
NEW!



- Noise: measurement/model error: often significant
- R small
- Possibly many equations (overdetermined system)
- Use: extension of linear least-squares estimation

Sets of polynomial equations in different communities

(Numerical) mathematics



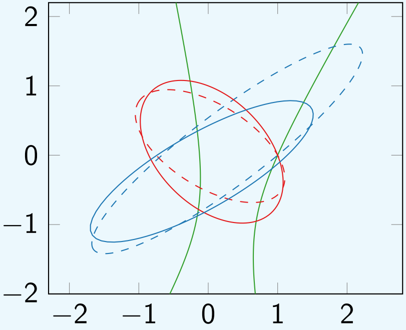
- Noise: numerical quantization error: typically small
- $R \sim$ Bézout number
- Typically number of equations = number of unknowns (square system)

Overview

Noisy overdetermined systems: looking for approximative roots

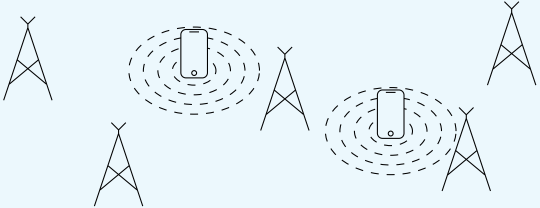
$$\begin{cases} -3 - x - 2y + 4x^2 + 6xy + 7y^2 = 0 \\ -2 - x + y + 3x^2 - 7xy + 5y^2 = 0 \\ 1 + 7x + y - 8x^2 + 3xy + y^2 = 0 \end{cases}$$

- $N = 2$ unknowns
- $S = 3$ equations $\rightarrow$ *overdetermined*
- Degree $d = 2$



Adding noise to the red and blue equations *destroys* the single exact root at $(1, 0)$

A practical application: "blind" multi-source localization



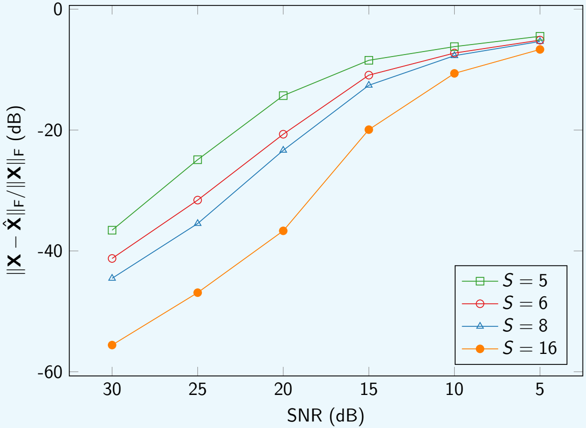
Friis transmission equation (before conversion into a polynomial expression):

$$P_i^r = \frac{A_i^r A_1^t}{\lambda^2} \frac{P_1^t}{(x_i^r - x_1^t)^2 + (y_i^r - y_1^t)^2} + \frac{A_i^r A_2^t}{\lambda^2} \frac{P_2^t}{(x_i^r - x_2^t)^2 + (y_i^r - y_2^t)^2}, \quad i = 1, \ldots, S.$$

Noisy measured quantities! Unknown Given

$\longrightarrow$ for $S \geq 5$, positions of transmitters can be *retrieved* up to permutation ambiguity!

Similar to least-squares: adding more equations (i.e., antennas) yield better estimates



Median relative error of estimated transmitter positions over 200 experiments
[Widdershoven, Govindarajan, et al. 2023]

-50 dB error ≈ 5 digits of precision

Overview

Algebraic methods: "classical" vs. recent numerical (multi)-linear algebra approaches

Find all (projective) roots of the system of the multivariate polynomials:

$$\Sigma : \quad \begin{cases} p_1 & = & p_1(x_1, x_2, \ldots, x_N) \\ & \vdots & \\ p_S & = & p_S(x_1, x_2, \ldots, x_N) \end{cases}, \qquad S \geq N, \qquad \deg(p_s) = d_s.$$

Auzinger, Stetter, Lazard, ...	Batselier, Dreesen, De Moor,...	Vanderstukken, De Lathauwer,...
Numerical Polynomial Algebra (NPA) →	Numerical Polynomial Linear Algebra (NPLA) →	Numerical Polynomial Multi-Linear Algebra (NPMLA)
Features: - Gröbner basis construction - (Generalized) eigenvalue problem	Features: - Macaulay null space construction - Generalized eigenvalue problem	Features: - Macaulay null space construction - tensor decomposition problem

DISCLAIMER: The above is a very selected overview and only shows "ancestors" of our own work. It is by no means a summary of all the contributions done on this topic.

The Macaulay-based method for polynomial root solving

The rows of the Macaulay matrix $\mathbf{M}(d)$ span the set

$$\mathcal{M}_d := \left\{ \sum_{s=1}^{S} g_s \cdot p_s : \quad \deg(g_s) = d - d_s \right\}.$$

For example, $M(3)$ for the system in slide 9:

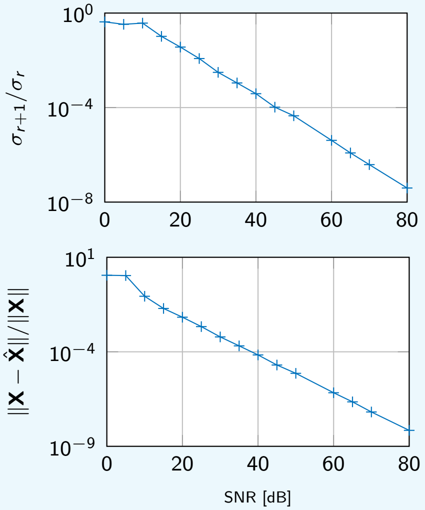
$$\begin{array}{r} p_1(x,y) \\ p_2(x,y) \\ p_3(x,y) \\ xp_1(x,y) \\ xp_2(x,y) \\ xp_3(x,y) \\ yp_1(x,y) \\ yp_2(x,y) \\ yp_3(x,y) \end{array}
\left[\begin{array}{c|cc|ccc|cccc}
-3 & -1 & -2 & 4 & 6 & 7 & 0 & 0 & 0 & 0 \\
-2 & -1 & 1 & 3 & -7 & 5 & 0 & 0 & 0 & 0 \\
1 & 7 & 1 & -8 & 3 & 1 & 0 & 0 & 0 & 0 \\
\hline
0 & -3 & 0 & -1 & -2 & 0 & 4 & 6 & 7 & 0 \\
0 & -2 & 0 & -1 & 1 & 0 & 3 & -7 & 5 & 0 \\
0 & 1 & 0 & 7 & 1 & 0 & -8 & 3 & 1 & 0 \\
\hline
0 & 0 & -3 & 0 & -1 & -2 & 0 & 4 & 6 & 7 \\
0 & 0 & -2 & 0 & -1 & 1 & 0 & 3 & -7 & 5 \\
0 & 0 & 1 & 0 & 7 & 1 & 0 & -8 & 3 & 1
\end{array}\right]
\left[\begin{array}{c} 1 \\ \hline x \\ y \\ \hline x^2 \\ xy \\ y^2 \\ \hline x^3 \\ x^2y \\ xy^2 \\ y^3 \end{array}\right]
=
\left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ 0 \end{array}\right]$$

If $d \geq d^*$ (degree of regularity), $\dim \operatorname{null} \mathbf{M}(d) =$ no. of projective roots of the system.

Our system from slide 9: Macaulay method is suitable in the noisy setting

Add noise to the nonzero coefficients:

$$\mathbf{M}(4) = \begin{bmatrix}
-3 & -1 & 4 & & & -2 & 6 & & & 7 & & & & & \\
-2 & -1 & 3 & & & 1 & 7 & & & 5 & & & & & \\
1 & 7 & -8 & & & 1 & 3 & & & 1 & & & & & \\
& -3 & -1 & 4 & & & -2 & 6 & & & 7 & & & & \\
& -2 & -1 & 3 & & & 1 & 7 & & & 5 & & & & \\
& 1 & 7 & -8 & & & 1 & 3 & & & 1 & & & & \\
& & -3 & -1 & 4 & & & -2 & 6 & & & 7 & & & \\
& & -2 & -1 & 3 & & & 1 & 7 & & & 5 & & & \\
& & 1 & 7 & -8 & & & 1 & 3 & & & 1 & & & \\
& & & & & -3 & -1 & 4 & & -2 & 6 & & 7 & & \\
& & & & & -2 & -1 & 3 & & 1 & 7 & & 5 & & \\
& & & & & 1 & 7 & -8 & & 1 & 3 & & 1 & & \\
& & & & & & -3 & -1 & 4 & & -2 & 6 & & 7 & \\
& & & & & & -2 & -1 & 3 & & 1 & 7 & & 5 & \\
& & & & & & 1 & 7 & -8 & & 1 & 3 & & 1 & \\
& & & & & & & & & -3 & -1 & 4 & -2 & 6 & 7 \\
& & & & & & & & & -2 & -1 & 3 & 1 & 7 & 5 \\
& & & & & & & & & 1 & 7 & -8 & 1 & 3 & 1
\end{bmatrix}$$



Note: $\mathbf{M}(4)$ is expressed in lex ordering this time!

The Macaulay-based method for polynomial root solving

The rows of the Macaulay matrix $\mathbf{M}(d)$ span the set

$$\mathcal{M}_d := \left\{ \sum_{s=1}^{S} g_s \cdot p_s : \quad \deg(g_s) = d - d_s \right\}.$$

For example, $M(3)$ for the system in slide 9:

$$\begin{array}{r} p_1(x,y) \\ p_2(x,y) \\ p_3(x,y) \\ xp_1(x,y) \\ xp_2(x,y) \\ xp_3(x,y) \\ yp_1(x,y) \\ yp_2(x,y) \\ yp_3(x,y) \end{array}
\left[\begin{array}{c|cc|ccc|cccc}
-3 & -1 & -2 & 4 & 6 & 7 & 0 & 0 & 0 & 0 \\
-2 & -1 & 1 & 3 & -7 & 5 & 0 & 0 & 0 & 0 \\
1 & 7 & 1 & -8 & 3 & 1 & 0 & 0 & 0 & 0 \\
\hline
0 & -3 & 0 & -1 & -2 & 0 & 4 & 6 & 7 & 0 \\
0 & -2 & 0 & -1 & 1 & 0 & 3 & -7 & 5 & 0 \\
0 & 1 & 0 & 7 & 1 & 0 & -8 & 3 & 1 & 0 \\
\hline
0 & 0 & -3 & 0 & -1 & -2 & 0 & 4 & 6 & 7 \\
0 & 0 & -2 & 0 & -1 & 1 & 0 & 3 & -7 & 5 \\
0 & 0 & 1 & 0 & 7 & 1 & 0 & -8 & 3 & 1
\end{array}\right]
\left[\begin{array}{c} 1 \\ \hline x \\ y \\ \hline x^2 \\ xy \\ y^2 \\ \hline x^3 \\ x^2y \\ xy^2 \\ y^3 \end{array}\right]
=
\left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ 0 \end{array}\right]$$

If $d \geq d^*$ (degree of regularity), $\dim \operatorname{null} \mathbf{M}(d) =$ no. of projective roots of the system.

Exponentials as rank-1 matrices and tensors

- Consider a univariate exponential signal $f(k) = ax^k$, sampled uniformly:

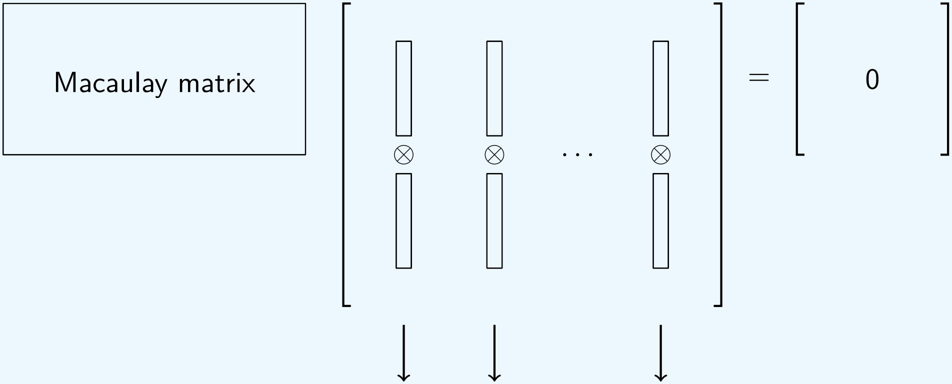
$$\begin{bmatrix} a & ax & ax^2 & ax^3 & \cdots \end{bmatrix}$$

- Let's arrange it in a Hankel matrix $\mathbf{H}$

$$\mathbf{H} = \begin{bmatrix} a & ax & ax^2 & \cdots \\ ax & ax^2 & ax^3 & \cdots \\ ax^2 & ax^3 & ax^4 & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix} = a \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \end{bmatrix} \begin{bmatrix} 1 & x & x^2 & \cdots \end{bmatrix}$$

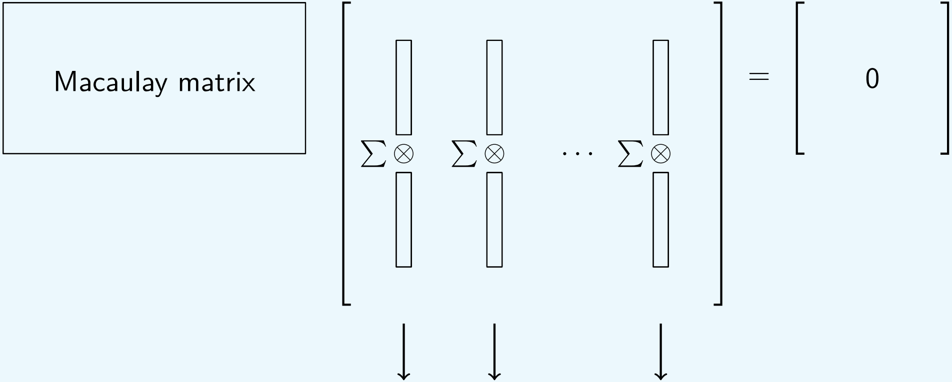
- $\mathbf{H}$ has rank 1 !
- $(2 \times 2 \times \ldots \times 2)$ tensor $\mathcal{H}$ of order k has rank 1: $a \begin{bmatrix} 1 \\ x \end{bmatrix} \otimes \begin{bmatrix} 1 \\ x \end{bmatrix} \otimes \ldots \otimes \begin{bmatrix} 1 \\ x \end{bmatrix}$
- Multivariate exponential: $a \underbrace{\begin{bmatrix} 1 \\ x \end{bmatrix} \otimes \ldots \otimes \begin{bmatrix} 1 \\ x \end{bmatrix}} \otimes \underbrace{\begin{bmatrix} 1 \\ y \end{bmatrix} \otimes \ldots \otimes \begin{bmatrix} 1 \\ y \end{bmatrix}} \otimes \underbrace{\begin{bmatrix} 1 \\ z \end{bmatrix} \otimes \ldots \otimes \begin{bmatrix} 1 \\ z \end{bmatrix}}$

Macaulay and CPD (1)



R multivariate Vandermonde vectors
(= one per root)

Macaulay and CPD (2)



R linear combinations of outer products

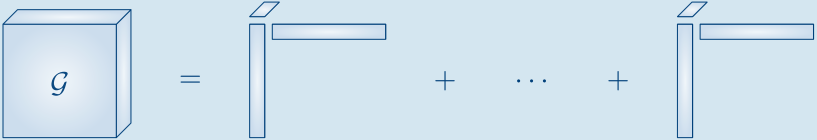
$\sim$ basis null space

Reformulation of the root recovery as a tensor decomposition problem

Theorem (Vanderstukken and De Lathauwer 2021)

Let $\mathcal{G}$ have frontal slices $\mathbf{G}_t, \mathbf{G}_{x_1}, \ldots, \mathbf{G}_{x_N}$, and assume Σ has only simple roots. If $\mathcal{G}$ is constructed from $\operatorname{null} \mathbf{M}(d)$ *with $d \geq d^* + 1$, then $\mathcal{G}$ has the essentially unique CPD*

$$\mathcal{G} = [\![\mathbf{V}, \mathbf{A}^{-1}, \mathbf{X}]\!], \quad \mathbf{X} = \begin{bmatrix} t^{(1)} & t^{(2)} & \cdots & t^{(R)} \\ x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(R)} \\ \vdots & \vdots & & \vdots \\ x_N^{(1)} & x_N^{(2)} & \cdots & x_N^{(R)} \end{bmatrix}.$$



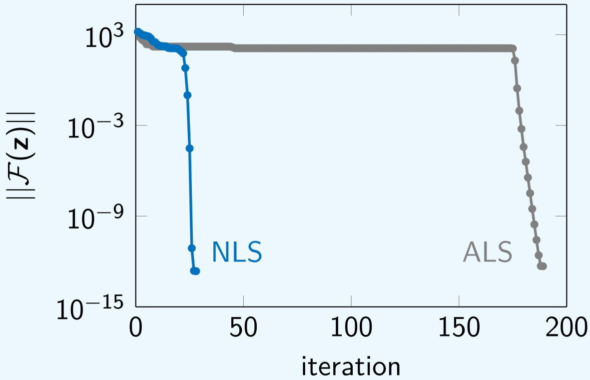
If the polynomial system has roots of multiplicity greater than one, the theorem can be generalized with the introduction of block-term decompositions [Vanderstukken, Kürschner, et al. 2021]

Benefits of tensor approach

- Exploit more of the structure for higher accuracy (noisy)
- Compute fewer null space vectors (speed up)

Algorithm basics: CPD

CPD of a $9 \times 9 \times 9 \times 9 \times 9$ tensor of rank 11



swamp

- init: EVD, random
- global $\leftrightarrow$ asymptotic
- asymptotic convergence: linear - superlinear - quadratic
- unconstrained decomposition $\leftrightarrow$ numerical challenges

Optimization-based framework to solve LS-CPDs when considering low-rank structure.

Algorithm 1: LS-CPD using Gauss–Newton with dogleg trust region

Input : $\mathbf{A}$, $\mathbf{b}$, and $\{\mathbf{U}^{(n)}\}_{n=1}^{N}$

Output: $\{\mathbf{U}^{(n)}\}_{n=1}^{N}$

while *not converged* **do**

  Compute gradient $\mathbf{g}$.

  Use PCG to solve $\mathbf{Hp} = -\bar{\mathbf{g}}$ for $\mathbf{p}$ using Gramian-vector products using a (block)-Jacobi preconditioner.

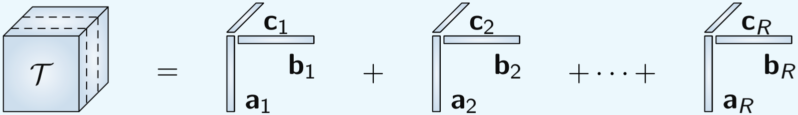
  Update $\mathbf{U}^{(n)}$, for $1 \leq n \leq N$, using dogleg trust region from $\mathbf{p}$, $\mathbf{g}$, and function evaluation.

end

[Boussé, Vervliet, et al. 2018]

Pencil-based computation: two factor matrices have f.c.r

CPD :



Slices :

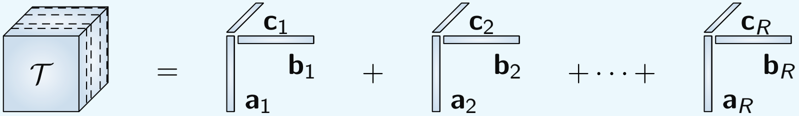
$$\mathbf{T}_{(:,:,1)} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 \cdots \mathbf{a}_R\end{bmatrix} \begin{bmatrix} c_{11} & & & \\ & c_{12} & & \\ & & \ddots & \\ & & & c_{1R} \end{bmatrix} \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 \cdots \mathbf{b}_R\end{bmatrix}^{\mathsf{T}}$$

$$\mathbf{T}_{(:,:,2)} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 \cdots \mathbf{a}_R\end{bmatrix} \begin{bmatrix} c_{21} & & & \\ & c_{22} & & \\ & & \ddots & \\ & & & c_{2R} \end{bmatrix} \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 \cdots \mathbf{b}_R\end{bmatrix}^{\mathsf{T}}$$

$$(G)EVD:\ \mathbf{T}_{(:,:,1)} \cdot \mathbf{T}_{(:,:,2)}^{-1} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 \cdots \mathbf{a}_R\end{bmatrix} \begin{bmatrix} c_{11}/c_{21} & & & \\ & c_{12}/c_{22} & & \\ & & \ddots & \\ & & & c_{1R}/c_{2R} \end{bmatrix} \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 \cdots \mathbf{a}_R\end{bmatrix}^{-1}$$

Multi-pencil-based computation: two factor matrices have f.c.r

CPD :



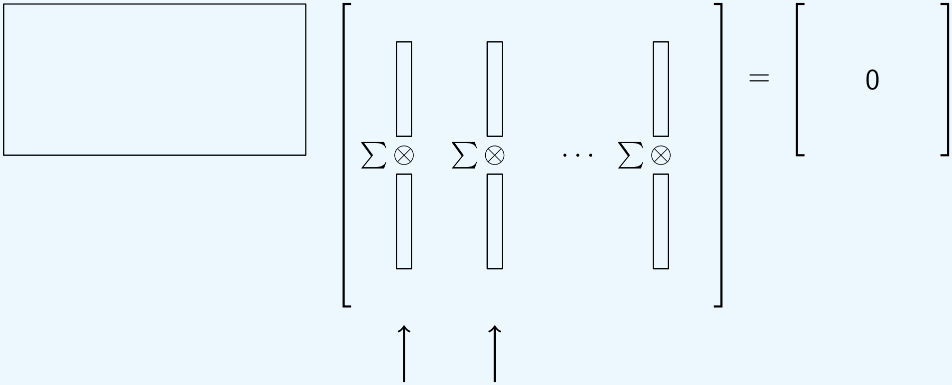
Slices :

$$\mathbf{T}_{(:,:,i)} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 \cdots \mathbf{a}_R\end{bmatrix} \begin{bmatrix} c_{i1} & & & \\ & c_{i2} & & \\ & & \ddots & \\ & & & c_{iR} \end{bmatrix} \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 \cdots \mathbf{b}_R\end{bmatrix}^{\mathrm{T}}$$

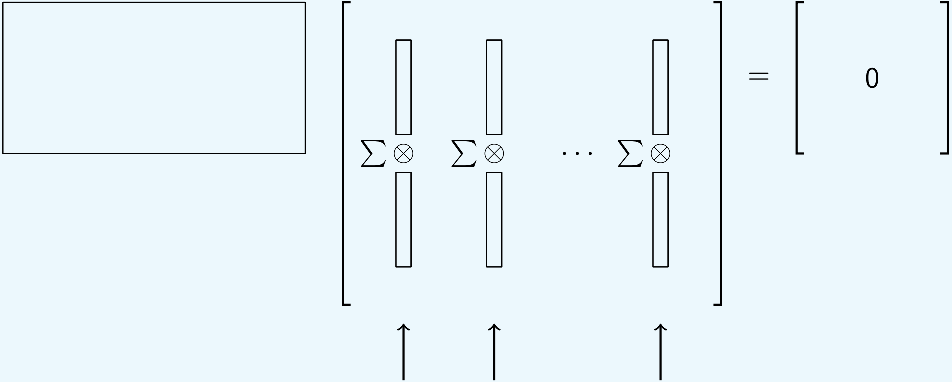
$$\mathbf{T}_{(:,:,j)} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 \cdots \mathbf{a}_R\end{bmatrix} \begin{bmatrix} c_{j1} & & & \\ & c_{j2} & & \\ & & \ddots & \\ & & & c_{jR} \end{bmatrix} \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 \cdots \mathbf{b}_R\end{bmatrix}^{\mathrm{T}}$$

$$(G)EVD: \ \mathbf{T}_{(:,:,i)} \cdot \mathbf{T}_{(:,:,j)}^{-1} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 \cdots \mathbf{a}_R\end{bmatrix} \begin{bmatrix} c_{i1}/c_{j1} & & & \\ & c_{i2}/c_{j2} & & \\ & & \ddots & \\ & & & c_{iR}/c_{jR} \end{bmatrix} \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 \cdots \mathbf{a}_R\end{bmatrix}^{-1}$$

Equivalent of two slices (1 pencil)



Equivalent of several slices (multi-pencil)



Small eigenvalue gaps lead to inaccuracy

Gen. eigenvalues of $(\mathbf{T}_k, \mathbf{T}_\ell)$ are interpreted as points on the unit circle. The pencil $(\mathbf{T}_k, \mathbf{T}_\ell)$ has R generalized eigenvalues.

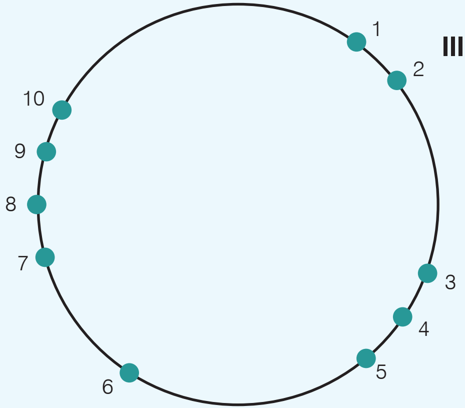


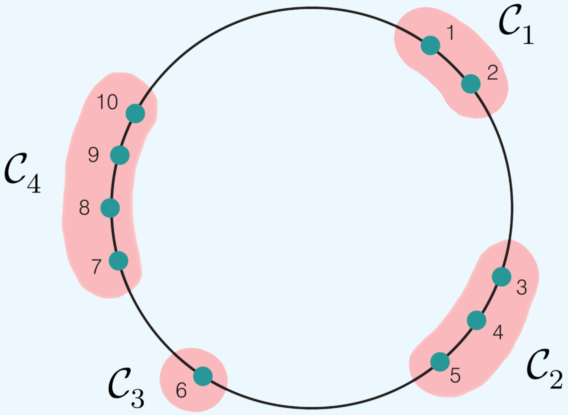
Illustration of generalized eigenvalues of $(\mathbf{T}_k, \mathbf{T}_\ell)$

● = generalized eigenvalue of $(\mathbf{T}_k, \mathbf{T}_\ell)$.

The small gap between generalized eigenvalues 1 and 2 leads to instability in computing the generalized eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$.

Similar issues occur in the other clusters of generalized eigenvalues.

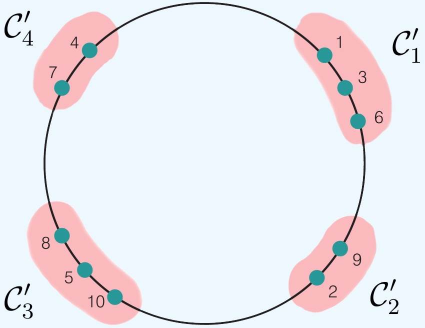
Generalized EigenSpace Decomp: Improve accuracy by computing eigenspaces corresponding to well separated eigenvalue clusters.



Clusters $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$ are well separated so can improve accuracy by only computing the corresponding eigenspaces $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$.

Use a new pencil to split eigenspaces!

Consider a new subpencil $(\mathbf{T}_m, \mathbf{T}_n)$. The eigenvectors of this pencil are the same as those of $(\mathbf{T}_k, \mathbf{T}_\ell)$, but the corresponding eigenvalues will lie in new positions on the unit circle.



The clusters $\mathcal{C}_1', \mathcal{C}_2', \mathcal{C}_3', \mathcal{C}_4'$ are well separated, so can compute the eigenspaces $\mathcal{E}_1', \mathcal{E}_2', \mathcal{E}_3', \mathcal{E}_4'$.

Observe $\mathcal{E}_1 = \text{span}\{\mathbf{v}_1, \mathbf{v}_2\}$ and $\mathcal{E}_1' = \text{span}\{\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_6\}$. Thus $\mathbf{v}_1 = \mathcal{E}_1 \cap \mathcal{E}_1'$.

GESD recursively deflates tensor rank.

In our implementation, GESD recursively writes $\mathcal{T}$ as a sum of tensors of reduced rank.

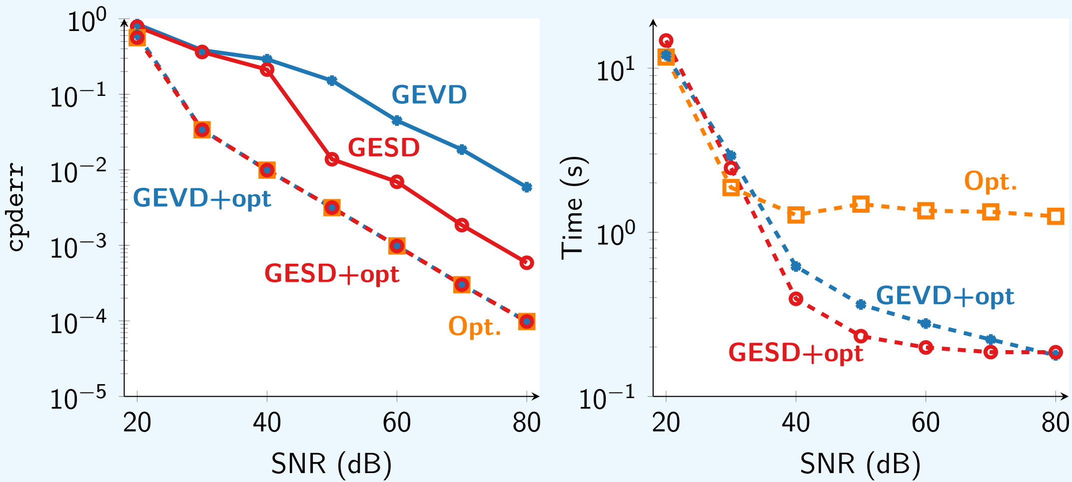
In the example, GESD would use $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$ to write the rank 10 tensor $\mathcal{T}$ as

$$\mathcal{T} = \mathcal{T}^1 + \mathcal{T}^2 + \mathcal{T}^3 + \mathcal{T}^4$$

where $\mathcal{T}^1, \mathcal{T}^2, \mathcal{T}^3$ and $\mathcal{T}^4$ have ranks $2, 3, 1$ and 4, respectively. $\mathcal{T}^1$ can then be decomposed into a sum of rank 1 tensors using the pencil $(\mathcal{T}^1_m, \mathcal{T}^1_n)$, etc.

Variations in GESD are possible. E.g. one could compute intersections of eigenspaces as described above rather than working recursively.

GESD vs synthetic data

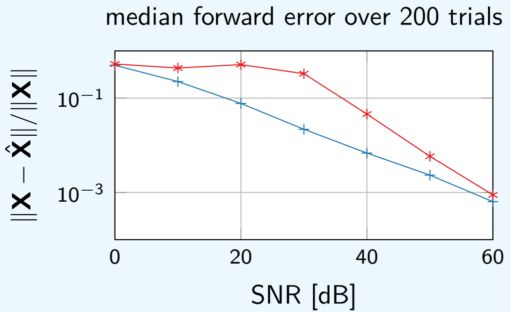


Accuracy and speed against Rank 10 tensors of size $100 \times 100 \times 100$ with highly correlated factor matrix columns.

Observed numerical benefits of the tensor approach for noisy overdetermined systems

Take $N = 10$ *noisy* copies of the square system:

$$\Sigma : \begin{cases} f_1(x_1, x_2) = x_1^3 + x_2^3 - 9x_1^2 x_2 + 20x_1x_2 - 3x_1 - 20 = 0 \\ f_2(x_1, x_2) = x_1^2 + 4x_2^2 - x_1x_2 - 80 = 0 \end{cases}$$



The tensor-based method that relies on simultaneous diagonalization is better capable of recovering roots in noisy conditions than a pure matrix-based method which relies solely on GEVD [Vanderstukken and De Lathauwer 2021].

Different tensorization approaches: compute roots from fewer null space vectors!

Problem: Square quadratic bivariate system (4 roots)

Approach 1: Retrieve roots from *full* 4-dimensional null space

$$\left[\!\left[\begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 \\ x_1 y_1 & x_2 y_2 & x_3 y_3 & x_4 y_4 \\ y_1^2 & y_2^2 & y_3^2 & y_4^2 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{bmatrix}, \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix} \right]\!\right]$$

Uniqueness properties:

First and third factor full column rank, second factor has no proportional columns

Different tensorization approaches: compute roots from fewer null space vectors!

Problem: Square quadratic bivariate system (4 roots)

Approach 2: Retrieve null space from *just two* null space vectors

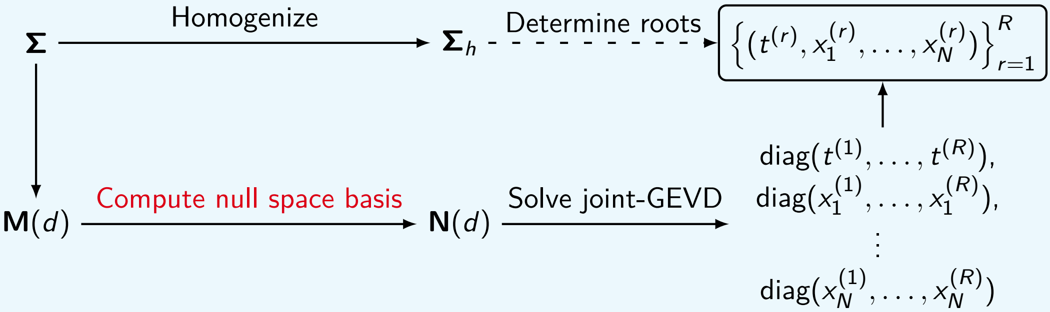
$$\left[\!\left[\begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 \\ x_1 y_1 & x_2 y_2 & x_3 y_3 & x_4 y_4 \\ y_1^2 & y_2^2 & y_3^2 & y_4^2 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 \end{bmatrix}, \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \end{bmatrix} \right]\!\right]$$

Uniqueness properties:

First and second factor full column rank, third factor has no proportional columns.

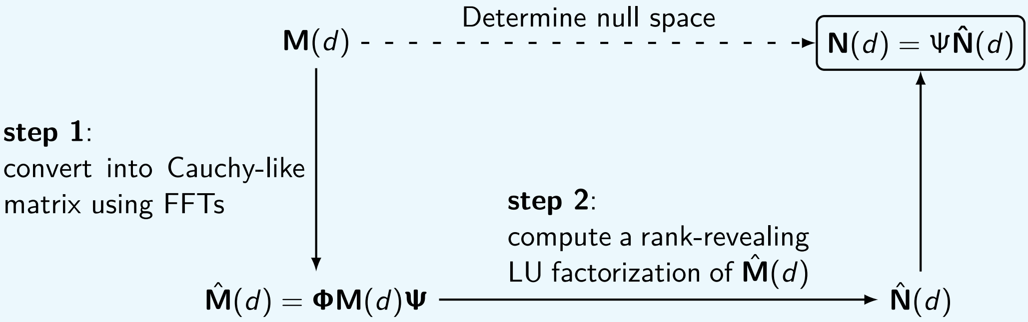
Overview

Null space computation is the major computational bottleneck in *many* algorithms!



Exploit the *Toeplitz* structures in Macaulay matrices?

Overview of the fast algorithm



Both steps can be done *fast*!
[Govindarajan, Widdershoven, et al. 2023]

Overview

What we have discussed in this talk

- Solving polynomial systems in the *noisy* overdetermined setting.
- Number of approximate roots $\sim$ number of small singular values
- Algebraic and optimization-based algorithms
- Benefits of taking on a "*tensor*" view towards polynomial root solving.
- Pencil and multi-pencil based CPD
- Progress and challenges towards (asymptotically) *faster* Macaulay null space algorithms.

Canonical Polyadic Decomposition and Sets of Polynomial Equations

Lieven De Lathauwer
Joint work with Nithin Govindarajan and Raphaël Widdershoven

Back to the Roots seminar
Leuven, December 12th, 2023

KU LEUVEN
STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics
EOS THE EXCELLENCE OF SCIENCE
erc European Research Council Established by the European Commission

References I

- Acar, E. et al. (2007). "Multiway analysis of epilepsy tensors". In: *Bioinformatics* 23.13, pp. i10–i18.
- Boussé, M. et al. (Aug. 2018). "Linear Systems with a Canonical Polyadic Decomposition Constrained Solution: Algorithms and applications". In: *Numerical Linear Algebra with Applications* 25.6, e2190.
- de Silva, V. and Lim, L.-H. (Sept. 2008). "Tensor Rank and the Ill-Posedness of the Best Low-Rank Approximation Problem". In: *SIAM Journal on Matrix Analysis and Applications* 30.3, pp. 1084–1127.
- De Vos, M. et al. (Sept. 2007). "Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone". In: *NeuroImage* 37.3, pp. 844–854.
- Evert, E. and De Lathauwer, L. (Mar. 2022). "Guarantees for existence of a best canonical polyadic approximation of a noisy low-rank tensor". In: *SIAM Journal On Matrix Analysis And Applications* 43.1, pp. 328–369.

References II

- Evert, E., Vandecappelle, M., and De Lathauwer, L. (Feb. 2022). "A recursive eigenspace computation for the canonical polyadic decomposition". In: *SIAM Journal On Matrix Analysis And Applications* 43.1, pp. 274–300.
- Govindarajan, N. et al. (2023). "A fast algorithm for computing Macaulay nullspaces of bivariate polynomial systems". In: *Technical Report 23-16, ESAT-STADIUS, KU Leuven (Leuven, Belgium)*.
- Håstad, J. (1990). "Tensor Rank is NP-complete". In: *Journal of Algorithms* 11.4, pp. 644–654.
- Hillar, C. J. and Lim, L.-H. (Nov. 2013). "Most Tensor Problems Are NP-Hard". In: *Journal of the ACM* 60.6, 45:1–45:39.
- Hunyadi, B. et al. (Sept. 2014). "Block term decomposition for modelling epileptic seizures". In: *EURASIP Journal on Advances in Signal Processing* 2014.139. Special Issue on Recent Advances in Tensor-Based Signal and Image Processing, pp. 1–19.

References III

Kruskal, J. B., Harshman, R. A., and Lundy, M. E. (1989). “How 3-MFA Data Can Cause Degenerate PARAFAC Solutions, Among Other Relationships”. In: *Multiway Data Analysis*. Ed. by R. Coppi and S. Bolasco. Amsterdam, The Netherlands: North-Holland Publishing Co., pp. 115–121.

Sorber, L., Van Barel, M., and De Lathauwer, L. (Apr. 2013). “Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(L_r, L_r, 1)$ terms, and a new generalization”. In: *SIAM Journal on Optimization* 23.2, pp. 695–720.

Vanderstukken, J. and De Lathauwer, L. (2021). “Systems of Polynomial Equations, Higher-Order Rensor Decompositions and Multidimensional Harmonic Retrieval: A Unifying Framework. Part I: The Canonical Polyadic Decomposition”. In: *SIAM Journal on Matrix Analysis and Applications* 42.2, pp. 883–912.

References IV

- Vanderstukken, J. et al. (2021). "Systems of Polynomial Equations, Higher-Order Tensor Decompositions, and Multidimensional Harmonic Retrieval: A Unifying Framework. Part II: The Block Term Decomposition". In: *SIAM Journal on Matrix Analysis and Applications* 42.2, pp. 913–953.
- Widdershoven, R., Govindarajan, N., and De Lathauwer, L. (Sept. 2023). "Overdetermined systems of polynomial equations: tensor-based solution and application". In: *2023 31st European Signal Processing Conference, (EUSIPCO), Helsinki, Finland, September 4-8, 2023*.