A tensor-based approach to solving systems of multivariate polynomials

Nithin Govindarajan

with Raphaël Widdershoven, Shiv Chandrasekaran (UCSB), and Lieven De Lathauwer

June 15th 2023

KU LEUVEN

Overview

Motivation: noisy overdetermined polynomial systems

Polynomial root solving: from an eigenvalue to a tensor decomposition problem

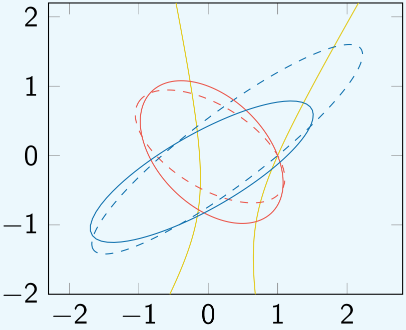
Faster Macaulay null space computations

Summary

Noisy overdetermined systems: looking for approximative roots

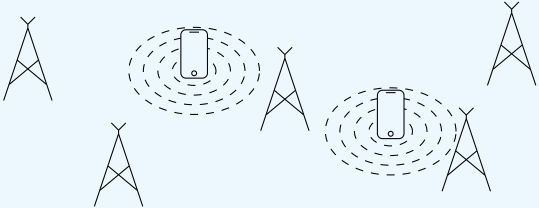
$$\begin{cases} -3 - x - 2y + 4x^2 + 6xy + 7y^2 = 0 \\ -2 - x + y + 3x^2 - 7xy + 5y^2 = 0 \\ 1 + 7x + y - 8x^2 + 3xy + y^2 = 0 \end{cases}$$

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



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

A practical application: "blind" multi-source localization



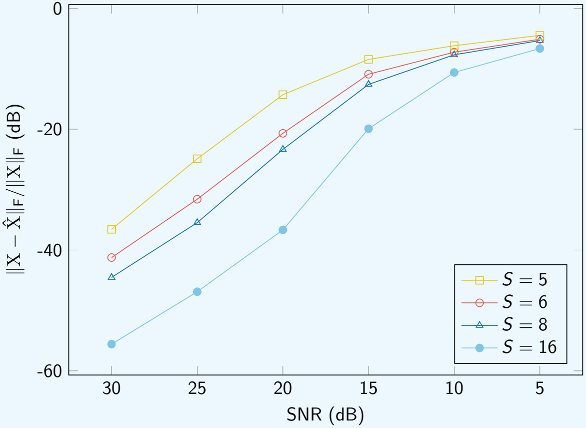
Friis transmission equation (before conversion into a polynomial expression):

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

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

Motivation: noisy overdetermined polynomial systems

Polynomial root solving: from an eigenvalue to a tensor decomposition problem

Faster Macaulay null space computations

Summary

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

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

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

Auzinger, Stetter, Lazard, ...		Batselier, Dreesen,...		Vanderstukken, De Lathauwer,...
Numerical Polynomial Algebra (NPA)	$\rightarrow$	Numerical Polynomial Linear Algebra (NPLA)	$\rightarrow$	Numerical Polynomial Multi-Linear Algebra (NPMLA)
Features: - Gröbner basis construction - 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 $\mathrm{M}(d)$ span the set

$$\mathcal{M}_d := \left\{ \sum_{s=1}^{S} h_s \cdot p_s : \quad \deg(h_s) = d - d_s \right\} = \left\langle p_1^h, \ldots, p_s^h \right\rangle_d .$$

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

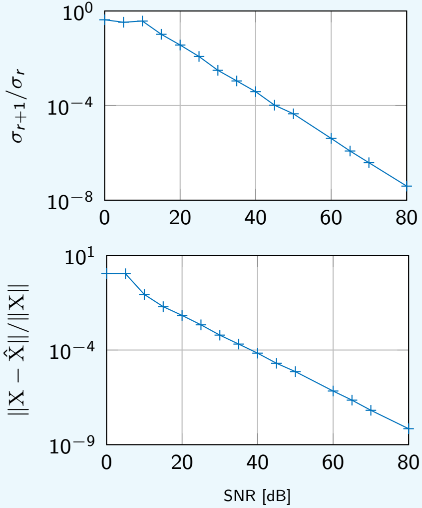
$$\begin{array}{c} p_1^h \\ p_2^h \\ p_3^h \\ xp_1^h \\ xp_2^h \\ xp_3^h \\ yp_1^h \\ yp_2^h \\ yp_3^h \end{array}
\left[\begin{array}{c|cc|ccc|cccc}
-3 & -1 & -2 & 4 & 6 & 7 & 0 & 0 & 0 & 0 \\
-2 & -1 & 1 & 3 & -7 & 5 & 0 & 0 & 0 & 0 \\
1 & 7 & 1 & -8 & 3 & 1 & 0 & 0 & 0 & 0 \\
\hline
0 & -3 & 0 & -1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

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

Our system from slide 3: Macaulay method is suitable in the noisy setting

Add noise to the nonzero coefficients:

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



Note: $M(4)$ is expressed in lex ordering this time!

The matrix view: recovering the roots from a generalized eigenvalue problem

- Let $\mathrm{S}_t, \mathrm{S}_{x_1}, \mathrm{S}_{x_2}, \ldots, \mathrm{S}_{x_N}$ denote appropriate “row selection” matrices.
- Construct $\mathrm{G}_t = \mathrm{S}_t\mathrm{N}$ and $\mathrm{G}_{x_i} = \mathrm{S}_{x_i}\mathrm{N}$ for $i = 1, \ldots, N$ with $\mathrm{N} = \mathrm{null}\,\mathrm{M}(d)$
- Solve the generalized eigenvalue decomposition (GEVD) problem:

$$\left(\alpha_t \mathrm{G}_t + \alpha_{x_1}\mathrm{G}_{x_1} + \ldots + \alpha_{x_N}\mathrm{G}_{x_N}\right)\boldsymbol{a} = \lambda\left(\beta_t \mathrm{G}_t + \beta_{x_1}\mathrm{G}_{x_1} + \ldots + \beta_{x_N}\mathrm{G}_{x_N}\right)\boldsymbol{a}$$

- For $i = 1, \ldots, R$ (number of roots), we have the eigenvalues:

$$\lambda_i = \frac{\alpha_t t^{(i)} + \alpha_{x_1} x_1^{(i)} + \ldots + \alpha_{x_N} x_N^{(i)}}{\beta_t t^{(i)} + \beta_{x_1} x_1^{(i)} + \ldots + \beta_{x_N} x_N^{(i)}}$$

- Eigenvectors *reveal* root location, since

$$\boldsymbol{v}_i = \left(\alpha_t \mathrm{G}_t + \alpha_{x_1}\mathrm{G}_{x_1} + \ldots + \alpha_{x_N}\mathrm{G}_{x_N}\right)\boldsymbol{a}_i$$

  are multivariate *Vandermonde* vectors evaluated at the system roots!

Typically $\alpha_{x_1}, \ldots, \alpha_{x_n}$ is set to zero.

Matrix pencils can be badly conditioned when eigenvalues coalesce...

Theorem (See e.g., Golub and Van Loan 2012))

Define $\mathsf{sep}(\mathrm{T}_{11}, \mathrm{T}_{22}) := \min \|\mathrm{T}_{11}\mathrm{X} - \mathrm{X}\mathrm{T}_{22}\|_F \,/\, \|\mathrm{X}\|_F$ *and let* $\mathrm{Q}^*\mathrm{A}\mathrm{Q} = \begin{bmatrix} \mathrm{T}_{11} & \mathrm{T}_{12} \\ 0 & \mathrm{T}_{22} \end{bmatrix}$ *with* $\mathrm{Q} = \begin{bmatrix} \mathrm{Q}_1 & \mathrm{Q}_2 \end{bmatrix}$. *Then for a "sufficiently small"* E, *there exists a* $\tilde{\mathrm{Q}}_1$ *with*

$$\mathsf{dist}(\mathsf{col}\ \mathrm{Q}_1, \mathsf{col}\ \tilde{\mathrm{Q}}_1) \lesssim \frac{1}{\mathsf{sep}(\mathrm{T}_{11}, \mathrm{T}_{22})}$$

such that $\tilde{\mathrm{Q}}_1$ *is an invariant subspace for* $\tilde{\mathrm{A}} = \mathrm{A} + \mathrm{E}$.

A bad choice of α's and β's can bring eigenvalues arbitrarily close!

The above is a compressed statement carrying the essence of Corollary 7.2.5 in (Golub and Van Loan 2012)

GESD principle: why limit to just one pencil, if you can exploit multiple?

GESD algorithm: using multiple pencils, recursively split eigenspaces corresponding to well-separated eigenvalue clusters (Evert, Vandecappelle, et al. 2022).

Simultaneous diagonalization:
There exists an invertible $\mathrm{A} \in \mathbb{C}^{R\times R}$ that simultaneously diagonalizes

$$\begin{aligned}
\mathrm{G}_t\mathrm{A} &= \mathrm{V}\,\mathrm{diag}(t^{(1)}, \ldots, t^{(R)}),\\
\mathrm{G}_{x_1}\mathrm{A} &= \mathrm{V}\,\mathrm{diag}(x_1^{(1)}, \ldots, x_1^{(R)}),\\
&\vdots\\
\mathrm{G}_{x_N}\mathrm{A} &= \mathrm{V}\,\mathrm{diag}(x_N^{(1)}, \ldots, x_N^{(R)}),
\end{aligned}$$

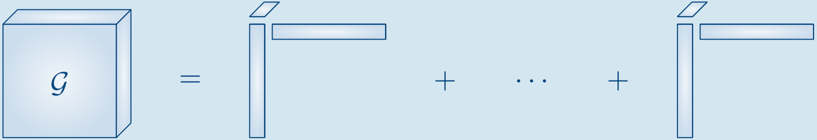
with V being multivariate Vandermonde matrix *evaluated* at the roots.

Reformulation of the root recovery as a tensor decomposition problem

Theorem (Vanderstukken and De Lathauwer 2021)

Let $\mathcal{G}$ have frontal slices $\mathrm{G}_t, \mathrm{G}_{x_1}, \ldots, \mathrm{G}_{x_2}$, and assume Σ has only simple roots. If $\mathcal{G}$ is constructed from $\operatorname{null} \mathrm{M}(d)$ *with $d \geq d^* + 1$, then $\mathcal{G}$ has the essentially unique CPD*

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

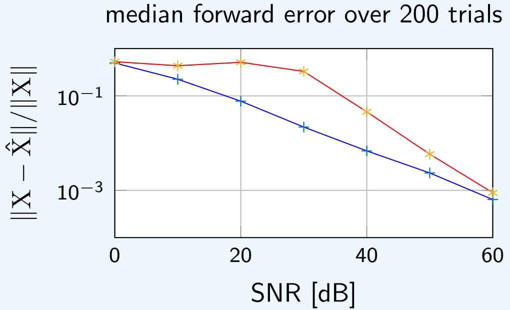


If the polynomial system has roots of multiplicity greater than one, the theorem can be generalized with the introduction of block-term decompositions (Vanderstukken, Kürschner, et al. 2021)

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).

Overview

Motivation: noisy overdetermined polynomial systems

Polynomial root solving: from an eigenvalue to a tensor decomposition problem

Faster Macaulay null space computations

Summary

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

$$
\begin{tikzcd}
\Sigma \arrow[r, "\text{Homogenize}"] \arrow[d] & \Sigma_h \arrow[r, dashed, "\text{Determine roots}"] & \boxed{\left\{\left(t^{(r)}, x_1^{(r)}, \ldots, x_N^{(r)}\right)\right\}_{r=1}^{R}} \\
\mathrm{M}(d) \arrow[r, "\text{Compute null space basis}"] & \mathrm{N}(d) \arrow[r, "\text{Solve joint-GEVD}"] & \begin{matrix} \operatorname{diag}(t^{(1)}, \ldots, t^{(R)}), \\ \operatorname{diag}(x_1^{(1)}, \ldots, x_1^{(R)}), \\ \vdots \\ \operatorname{diag}(x_N^{(1)}, \ldots, x_N^{(R)}) \end{matrix} \arrow[u]
\end{tikzcd}
$$

Exploit the *Toeplitz* structures in Macaulay matrices?

Bivariate systems: Macaulay matrix is almost Toeplitz block-(block-)Toeplitz

$$
M(4) = \begin{array}{c}
\\ p_1 \\ p_2 \\ xp_1 \\ xp_2 \\ x^2p_1 \\ x^2p_2 \\ yp_1 \\ yp_2 \\ xyp_1 \\ xyp_2 \\ y^2p_1 \\ y^2p_2
\end{array}
\begin{array}{c}
\begin{array}{ccccc|cccc|ccc|cc|c}
1 & x & x^2 & x^3 & x^4 & y & xy & x^2y & x^3y & y^2 & xy^2 & x^2y^2 & y^3 & xy^3 & y^4
\end{array} \\
\left[\begin{array}{ccccc|cccc|ccc|cc|c}
1 & 6 & 4 & & & 2 & 5 & & & 3 & & & & & \\
9 & 1 & 3 & & & 8 & 7 & & & 2 & & & & & \\
 & 1 & 6 & 4 & & & 2 & 5 & & & 3 & & & & \\
 & 9 & 1 & 3 & & & 8 & 7 & & & 2 & & & & \\
 & & 1 & 6 & 4 & & & 2 & 5 & & & 3 & & & \\
 & & 9 & 1 & 3 & & & 8 & 7 & & & 2 & & & \\
\hline
 & & & & & 1 & 6 & 4 & & 2 & 5 & & 3 & & \\
 & & & & & 9 & 1 & 3 & & 8 & 7 & & 2 & & \\
 & & & & & & 1 & 6 & 4 & & 2 & 5 & & 3 & \\
 & & & & & & 9 & 1 & 3 & & 8 & 7 & & 2 & \\
\hline
 & & & & & & & & & 1 & 6 & 4 & 2 & 5 & 3 \\
 & & & & & & & & & 9 & 1 & 3 & 8 & 7 & 2
\end{array}\right]
\end{array}
$$

The Macaulay matrix for the general bivariate case

Let $\Delta d := d - d_\Sigma$. Then,

$$\mathrm{M}(d) := \begin{bmatrix} \mathrm{M}_{0,0} & \mathrm{M}_{1,0} & \cdots & \mathrm{M}_{d_\Sigma,0} & & & \\ & \mathrm{M}_{0,1} & \mathrm{M}_{1,1} & \cdots & \mathrm{M}_{d_\Sigma,1} & & \\ & & \ddots & \ddots & & \ddots & \\ & & & \mathrm{M}_{0,\Delta d} & \mathrm{M}_{1,\Delta d} & \cdots & \mathrm{M}_{d_\Sigma,\Delta d} \end{bmatrix} \in \mathbb{C}^{\frac{S}{2}(\Delta d+1)(\Delta d+2)\times\frac{1}{2}(d+1)(d+2)},$$

with

$$\mathrm{M}_{i,j} := \begin{bmatrix} \boldsymbol{c}_{0i} & \boldsymbol{c}_{1i} & \cdots & \boldsymbol{c}_{(d_\Sigma-i)i} & & & \\ & \boldsymbol{c}_{0i} & \boldsymbol{c}_{1i} & \cdots & \boldsymbol{c}_{(d_\Sigma-i)i} & & \\ & & \ddots & \ddots & & \ddots & \\ & & & \boldsymbol{c}_{0i} & \boldsymbol{c}_{1i} & \cdots & \boldsymbol{c}_{(d_\Sigma-i)i} \end{bmatrix} \in \mathbb{C}^{S(\Delta d+1-j)\times(d+1-i-j)}.$$

The key observation that shall allow for a faster algorithm

Consider the displacement operator

$$\mathscr{D}\left\{\mathrm{M}(d)\right\} = \begin{bmatrix} \mathrm{Z}_{d+1,1} \otimes \mathrm{I}_S & & \\ & \ddots & \\ & & \mathrm{Z}_{1,1} \otimes \mathrm{I}_S \end{bmatrix} \mathrm{M}(d) - \mathrm{M}(d) \begin{bmatrix} \mathrm{Z}_{d+1,\varphi_{d+1}} & & \\ & \ddots & \\ & & \mathrm{Z}_{1,\varphi_1} \end{bmatrix}.$$

$\mathrm{M}(d)$ has relative “low” displacement rank

Dimensions of $\mathrm{M}(d) \in \mathbb{C}^{\frac{S}{2}(\Delta d+1)(\Delta d+2) \times \frac{1}{2}(d+1)(d+2)}$ grow *quadratically* w.r.t. d, but

$$\operatorname{rank} \mathscr{D}\left\{\mathrm{M}(d)\right\} \leq S(\Delta d + 1) = S\left(d + 1 - d_\Sigma\right).$$

grows only *linearly* with d.

Here $\mathrm{Z}_{p,\varphi} := \begin{bmatrix} & & \varphi \\ 1 & & \\ & \ddots & \\ & 1 & \end{bmatrix} \in \mathbb{C}^{p \times p}$.

Overview of the fast algorithm

$$\begin{tikzcd}
\mathrm{M}(d) \arrow[rr, dashed, "\text{Determine null space}"] \arrow[d, "\substack{\textbf{step 1}: \\ \text{convert into Cauchy-like} \\ \text{matrix using FFTs}}"'] & & \boxed{\mathrm{N}(d) = \Psi\hat{\mathrm{N}}(d)} \\
\hat{\mathrm{M}}(d) = \Phi\mathrm{M}(d)\Psi \arrow[rr, "\substack{\textbf{step 2}: \\ \text{compute a rank-revealing} \\ \text{LU factorization of } \hat{\mathrm{M}}(d)}"] & & \hat{\mathrm{N}}(d) \arrow[u]
\end{tikzcd}$$

Both steps can be done *fast*!
(Govindarajan, Widdershoven, et al. 2023)

Rank-revealing LU factorization of $\hat{\mathrm{M}}(d)$ (Miranian and Gu 2003)

Let $r(d) := \operatorname{rank} \mathrm{M}(d)$. Compute a rank-revealing LU (RRLU) factorization

$$\begin{aligned}
\Pi_1 \hat{\mathrm{M}}(d) \Pi_2 &= \begin{bmatrix} \mathrm{I}_{r(d)} & \\ \hat{\mathrm{M}}_{21}\hat{\mathrm{M}}_{11}^{-1} & \mathrm{I}_{d_\Sigma^2} \end{bmatrix} \begin{bmatrix} \hat{\mathrm{M}}_{11} & \\ & \hat{\mathrm{M}}_{22} - \hat{\mathrm{M}}_{21}\hat{\mathrm{M}}_{11}^{-1}\hat{\mathrm{M}}_{12} \end{bmatrix} \begin{bmatrix} \mathrm{I}_{r(d)} & \hat{\mathrm{M}}_{11}^{-1}\hat{\mathrm{M}}_{12} \\ & \mathrm{I}_{d_\Sigma^2} \end{bmatrix} \\
&\approx \begin{bmatrix} \hat{\mathrm{M}}_{11} \\ \hat{\mathrm{M}}_{21} \end{bmatrix} \begin{bmatrix} \mathrm{I}_{r(d)} & \hat{\mathrm{M}}_{11}^{-1}\hat{\mathrm{M}}_{12} \end{bmatrix}
\end{aligned}$$

Expression for the null space $\mathrm{N}(d)$

$$\mathrm{N}(d) = \Psi \Pi_2 \begin{bmatrix} \tilde{\mathrm{N}} \\ \mathrm{I}_{d_\Sigma^2} \end{bmatrix}, \qquad \tilde{\mathrm{N}} := -\hat{\mathrm{M}}_{11}^{-1}\hat{\mathrm{M}}_{12}.$$

The classical Schur algorithm: making things work for us

- Run Schur algorithm on Cauchy-like matrices, which satisfy the displacement relation $\mathrm{diag}(\boldsymbol{\mu})\mathrm{C} - \mathrm{C}\mathrm{diag}(\boldsymbol{\nu}) = \mathrm{R}\mathrm{S}^*$ (Heinig 1995)
- Replace total pivoting with approximate total pivoting
- Let $j_{\max}$ denote the column with largest 2-norm in $\mathrm{R}\mathrm{S}^*$. Then,

$$\max_{1\leq i\leq n} |c_{ij_{\max}}| \geq \frac{1}{K\sqrt{n}} \max_{1\leq i,j\leq n} |c_{ij}|, \quad K := \max_{1\leq i,j,\imath,\jmath\leq n} |\mu_i - \nu_j| / |\mu_\imath - \nu_\jmath|.$$

- Keeping R orthogonal during the Gaussian elimination process allows for fast pivot selection.
- Use clever updating strategies to keep cost low.

RESULT: from $\mathcal{O}(d^6)$ to $\mathcal{O}(d^5)$

Algorithm stability: error grows linearly with problem size

Median error $\epsilon := \|\mathrm{M}(d)\mathrm{Q}\|_2 \,/\, \|\mathrm{M}(d)\|_2 \geq \sigma_{r(d)+1}/\sigma_1$, with $\mathrm{Q} \in \mathbb{C}^{n(d)\times d_\Sigma^2}$ an orthonormal basis for col $\mathrm{N}(d)$, over 100 runs for randomly generated *square* systems.

	d_Σ				
	2	4	8	16	32
SVD on $\mathrm{M}(d)$	2.23e-16	3.75e-16	5.70e-16	7.94e-16	9.51e-16
GECP on $\mathrm{M}(d)$	1.40e-16	3.11e-16	8.33e-16	1.02e-14	1.40e-13
GECP on $\mathscr{C}$	4.35e-16	1.51e-15	1.35e-14	1.72e-13	2.81e-12
GEAP on $\mathscr{C}$	4.21e-16	3.63e-15	3.88e-14	3.19e-13	4.48e-12

Sources of error:

- switching to LU instead of an SVD
- working with the compact Cauchy representation $\mathscr{C}$
- switching to approximate pivoting $\leftarrow$ Surprisingly not so bad!

Extending the method

- Generalizations to Chebyshev systems possible; Toeplitz $\to$ Toeplitz-plus-Hankel ☺
- Displacement rank theory does not generalize nicely to higher dimensions with diminishing returns $\mathcal{O}(d^{3N})$ to $\mathcal{O}(d^{3N-1})$ ☹
- Assume additional (block) sparsity of Macaulay matrix to make breakthrough with n-dimensional systems?

$$p_i(x,y) = \sum_{k=0}^{d_\Sigma - r} a_{ikr} x^k y^r + \sum_{r \in \mathscr{S}} \sum_{k=0}^{d_\Sigma - r} a_{ikr} x^k y^r, \quad \mathscr{S} \subset \{1, \ldots, d_\Sigma\}, \quad |\mathscr{S}| = \rho \ll d_\Sigma$$

Overview

Motivation: noisy overdetermined polynomial systems

Polynomial root solving: from an eigenvalue to a tensor decomposition problem

Faster Macaulay null space computations

Summary

What we have discussed in this talk

- Solving polynomial systems in the *noisy* overdetermined setting.
- Benefits of taking on a "*tensor*" view towards polynomial root solving.
- Progress and challenges towards (asymptotically) *faster* Macaulay null space algorithms.

References I

Evert, Eric, Michiel Vandecappelle, and Lieven De Lathauwer (2022). “A Recursive Eigenspace Computation for the Canonical Polyadic Decomposition”. In: *SIAM J. Matrix Anal. Appl.* 43.1, pp. 274–300.

Golub, G.H. and C.F. Van Loan (2012). *Matrix Computations*. 4th ed. Johns Hopkins University Press.

Govindarajan, Nithin 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)*.

Heinig, Georg (1995). “Inversion of Generalized Cauchy Matrices and other Classes of Structured Matrices”. In: *Linear Algebra for Signal Processing*. Ed. by A. Bojanczyk and G. Cybenko. Vol. 69. The IMA Volumes in Mathematics and its Applications. New York, NY, USA: Springer.

References II

Miranian, L and Ming Gu (2003). “Strong rank revealing LU factorizations”. In: *Linear Algebra Appl.* 367, pp. 1–16.

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

Vanderstukken, Jeroen 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 J. Matrix Anal. Appl.* 42.2, pp. 913–953.

References III

Widdershoven, Raphaël, Nithin Govindarajan, and Lieven De Lathauwer (2023). "Overdetermined systems of polynomial equations: tensor-based solution and application". In: *Technical Report 23-37, ESAT-STADIUS, KU Leuven (Leuven, Belgium). Accepted for publication in European Signal Processing Conference (EUSIPCO) 2023.*