A fast algorithm for computing Macaulay nullspaces of bivariate polynomial systems

Nithin Govindarajan* Shivkumar Chandrasekaran† Lieven De Lathauwer*

* KU Leuven, † University of California Santa Barbara

Abstract

As a crucial first step towards finding the (approximate) common roots of an (overdetermined) bivariate polynomial system of equations Σ, the problem of determining an explicit numerical basis for the right nullspace of the system's Macaulay matrix is considered. If $d_\Sigma \in \mathbb{N}$ denotes the total degree of the $S \geqslant 2$ bivariate polynomials in the system, the cost of computing a nullspace basis containing all system roots involves $\mathcal{O}(d_\Sigma^6)$ floating point operations using standard numerical algebra techniques (e.g., a singular value decomposition, rank-revealing QR). We show that it is actually possible to design an algorithm that reduces the complexity to $\mathcal{O}(d_\Sigma^5)$.

The proposed algorithm exploits the almost Toeplitz-block-(block-)Toeplitz structure of the Macaulay matrix under a carefully chosen indexing of its entries and uses displacement rank theory to efficiently convert them into Cauchy-like matrices with the help of fast Fourier transforms. By modifying the classical total pivoting Schur algorithm for Cauchy matrices, a compact representation of the right nullspace is eventually found from a rank-revealing LU factorization.

I. The Macaulay matrix $\mathrm{M}(d)$

Overdetermined bivariate polynomial systems

We want to find the common roots of the system

$$\Sigma : \quad \begin{cases} p_1(x,y) := \displaystyle\sum_{i=0}^{d_\Sigma} \sum_{j=0}^{d_\Sigma - i} c_{1ij} x^i y^j \\ \qquad \vdots \\ p_S(x,y) := \displaystyle\sum_{i=0}^{d_\Sigma} \sum_{j=0}^{d_\Sigma - i} c_{Sij} x^i y^j \end{cases}, \tag{1}$$

where for all $s = 1, \dots, S$, $c_{si(d_\Sigma - i)} \neq 0$ for some $i = 0, 1, \dots, d_\Sigma$.

Description of $M(d)$

Let $\Delta d = d - d_\Sigma$. The Macaulay matrix

$$\mathrm{M}(d) \in \mathbb{C}^{\frac{S}{2}(\Delta d+1)(\Delta d+2) \times \frac{1}{2}(d+1)(d+2)},$$

with $d \geqslant d_\Sigma$, is the matrix constructed from the polynomial coefficients in (1) such that its rows span the set of polynomials

$$\mathscr{M}(d) := \left\{ \sum_{s=1}^{S} h_s \cdot p_s : \quad h_s \in \mathbb{C}[x,y],\ \deg(h_s) = \Delta d \right\}.$$

"Almost" Toeplitz-block-(block-)Toeplitzness of $\mathrm{M}(d)$

Let $\boldsymbol{c}_{kl} := \begin{bmatrix} c_{1kl} & \cdots & c_{Skl} \end{bmatrix}^\top$ for $k \leqslant d_\Sigma - l$ and $\boldsymbol{c}_{kl} = \mathbb{0}_S$, otherwise. By adopting a "tensorized" indexing for $\mathrm{M}(d)$ (see Figure 1), we may write

$$\mathrm{M}(d) = \begin{bmatrix} \mathrm{M}_{0,0} & \mathrm{M}_{1,0} & \cdots & \mathrm{M}_{d_\Sigma,0} & & & \\ & \mathrm{M}_{0,1} & \mathrm{M}_{1,1} & \cdots & \mathrm{M}_{d_\Sigma,1} & & \\ & & \ddots & \ddots & & \ddots & \\ & & & \mathrm{M}_{0,\Delta d} & \mathrm{M}_{1,\Delta d} & \cdots & \mathrm{M}_{d_\Sigma,\Delta d} \end{bmatrix}. \tag{2}$$

with

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

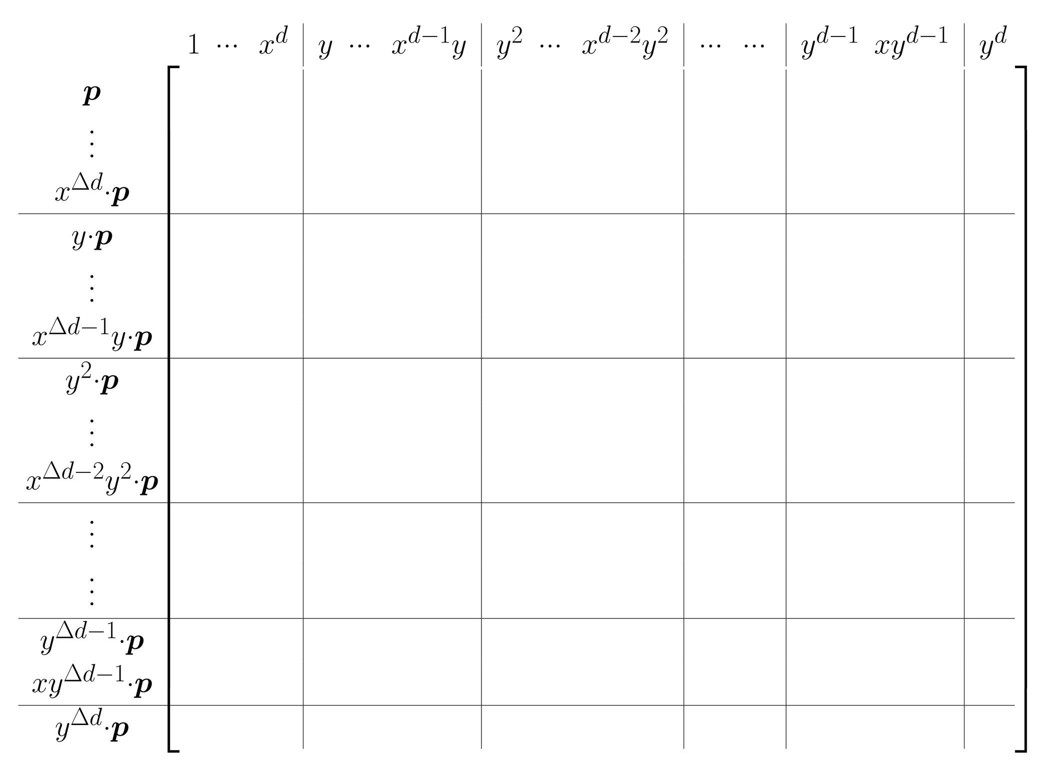

Figure 1. Significance of the rows and columns of the matrix defined in (2). Here, $x^i y^j \cdot \boldsymbol{p}$ is a shorthand for describing the polynomials $(x^i y^j \cdot p_1, x^i y^j \cdot p_2, \dots, x^i y^j \cdot p_S)$.

Reduction to a joint generalized eigenvalue problem

- If there exists two polynomials $p_i, p_j \in \Sigma$ such that $\mathcal{I}(p_i, p_j) = \mathcal{I}(\Sigma)$, and p_i, p_j share no common nontrivial components, *Bézout theorem* applies:

  The system (1) will have d_Σ^2 common roots
  (counting multiplicities and roots at infinity).

- The *degree of regularity* is the smallest $d^* \in \mathbb{N}$ for which $\dim \operatorname{null} \mathrm{M}(d) = d_\Sigma^2$ for all $d \geqslant d^*$. It can be shown that

$$d_\Sigma \leqslant d^* \leqslant 2d_\Sigma - 2. \tag{3}$$

- Starting with a basis for $\operatorname{null} M(d^*+1)$, the common roots of (1) can be retrieved from a joint generalized eigenvalue problem, or equivalently a *CPD computation* [2].

II. Displacement structure of $\mathrm{M}(d)$

Let $\varphi \in \mathbb{C}$ with $|\varphi| = 1$ and define

$$\mathrm{Z}_{m,\varphi} = \begin{bmatrix} & \cdots & \varphi \\ 1 & & \\ & \ddots & \vdots \\ & 1 & \end{bmatrix} \in \mathbb{C}^{m\times m},$$

Application of the Sylvester-type displacement operator

$$\mathscr{D}: \quad \mathrm{X} \mapsto \operatorname{diag}\{\mathrm{Z}_{i,1} \otimes \mathrm{I}_S\}_{i=\Delta d+1}^{1} \mathrm{X} - \mathrm{X} \operatorname{diag}\left\{\mathrm{Z}_{j,\varphi_j}\right\}_{j=d+1}^{1}, \tag{4}$$

onto $\mathrm{M}(d)$ transform the block entries $\mathrm{M}_{i,j}$ into $\bar{\mathrm{M}}_{i,j} = \mathrm{A}_j \mathrm{B}_{i,j}$ with

$$\mathrm{A}_j = \begin{bmatrix} \mathrm{I}_S \\ \mathbb{0}_{S(\Delta d - j)\times S} \end{bmatrix}$$

$$\mathrm{B}_{i,j} = \begin{bmatrix} \mathbb{0}_S & \cdots & \mathbb{0}_S & \boldsymbol{c}_{0i} & \cdots & \boldsymbol{c}_{(d_\Sigma-i)i} \end{bmatrix} - \begin{bmatrix} \boldsymbol{c}_{1i} & \cdots & \boldsymbol{c}_{(d_\Sigma-i)i} & \mathbb{0}_S & \cdots & \varphi_{i+j+1}\boldsymbol{c}_{0i} \end{bmatrix}.$$

$\mathrm{M}(d)$ has "low" displacement rank!

Consequently, the dimensions of $\mathrm{M}(d)$ grow *quadratically* w.r.t. d, but the rank of its displacement grows only *linearly* with d, since

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

III. Exploiting low displacement rank in nullspace computations

Outline of the fast algorithm

1. Apply unitary transformations Φ and Ψ such that $\Phi\mathrm{M}(d)\Psi =: \hat{\mathrm{M}}(d)$ is Cauchy-like, i.e., its entries are of the form

$$\left[\hat{\mathrm{M}}(d)\right]_{ij} := [\Phi\mathrm{M}(d)\Psi]_{ij} = \frac{\boldsymbol{u}_i^* \boldsymbol{v}_j}{\mu_i - \nu_j}, \qquad \boldsymbol{u}_i, \boldsymbol{v}_j \in \mathbb{C}^{S(\Delta d+1)}. \tag{5}$$

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

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

   to determine $\mathrm{N} := -\widehat{\mathrm{M}}_{11}^{-1}\widehat{\mathrm{M}}_{12}$, from where the nullspace of $\mathrm{M}(d)$ can be further derived.

Step 1: convert into Cauchy-like using FFTs

Substituting $\mathrm{Z}_{m,\varphi} = (\mathrm{D}_{m,\varphi}F_m)(\varphi^{1/m}\Omega_m)(\mathrm{D}_{m,\varphi}\mathrm{F}_m)^{-1}$ into (4), with $\mathrm{D}_{m,\varphi}$, Ω_m diagonal, and $[\mathrm{F}_m]_{ij} := \frac{1}{\sqrt{m}}\omega_m^{(i-1)(j-1)}$ as the DFT matrix, yields

$$\Phi := \operatorname{diag}\{\mathrm{F}_i^* \otimes \mathrm{I}_S\}_{i=\Delta d+1}^{1}, \qquad \Psi := \operatorname{diag}\left\{\mathrm{D}_{j,\varphi_j}\mathrm{F}_j\right\}_{j=d+1}^{1},$$

since $\hat{\mathrm{M}}(d)$ satisfies the displacement equation of a Cauchy-like matrix, i.e.,

$$\hat{\mathscr{D}}\left\{\hat{\mathrm{M}}(d)\right\} := \operatorname{diag}(\boldsymbol{\mu})\hat{\mathrm{M}}(d) - \hat{\mathrm{M}}(d)\operatorname{diag}(\boldsymbol{\nu}) = \Phi\mathscr{D}\{\mathrm{M}(d)\}\Psi,$$

with $\operatorname{diag}(\boldsymbol{\mu}) := \operatorname{diag}\{\Omega_i \otimes \mathrm{I}_S\}_{i=\Delta d+1}^{1}$, $\operatorname{diag}(\boldsymbol{\nu}) := \operatorname{diag}\{\varphi_j^{1/j}\ j\}_{j=d+1}^{1}$.

Step 2: use the Schur algorithm for RRLU

- Perform Gauss elimination on

$$\begin{bmatrix} \Pi_1 & \\ & -\mathrm{I}_{r(d)} \end{bmatrix} \begin{bmatrix} \widehat{\mathrm{M}}(d) \\ \Pi_{2,a}^\top \end{bmatrix} \begin{bmatrix} \Pi_{2,a} & \Pi_{2,b} \end{bmatrix} = \left[\begin{array}{cc} \widehat{\mathrm{M}}_{11} & \widehat{\mathrm{M}}_{12} \\ \widehat{\mathrm{M}}_{21} & \widehat{\mathrm{M}}_{22} \\ \hline -\mathrm{I}_{r(d)} & \end{array}\right] \sim \left[\begin{array}{cc} \widehat{\mathrm{M}}_{11} & \widehat{\mathrm{M}}_{12} \\ & \widehat{\mathrm{M}}_{22} + \widehat{\mathrm{M}}_{21}\mathrm{N} \\ \hline & \mathrm{N} \end{array}\right]$$

- Since Schur complements "preserve" displacement rank

$$\operatorname{diag}(\boldsymbol{\mu}) \begin{bmatrix} \mathrm{A} & \mathrm{G}^* \\ \mathrm{F} & \mathrm{B} \end{bmatrix} - \begin{bmatrix} \mathrm{A} & \mathrm{G}^* \\ \mathrm{F} & \mathrm{B} \end{bmatrix} \operatorname{diag}(\boldsymbol{\nu}) = \begin{bmatrix} \mathrm{R}_1 \\ \mathrm{R}_2 \end{bmatrix} \begin{bmatrix} \mathrm{S}_1 \\ \mathrm{S}_2 \end{bmatrix}^*$$

$$\downarrow$$

$$\operatorname{diag}(\boldsymbol{\mu}) \begin{bmatrix} \mathrm{A} & \\ & \mathrm{B} - \mathrm{F}\mathrm{A}^{-1}\mathrm{G}^* \end{bmatrix} - \begin{bmatrix} \mathrm{A} & \\ & \mathrm{B} - \mathrm{F}\mathrm{A}^{-1}\mathrm{G}^* \end{bmatrix} \operatorname{diag}(\boldsymbol{\nu}) = \begin{bmatrix} \mathrm{R}_1 \\ \tilde{\mathrm{R}}_2 \end{bmatrix} \begin{bmatrix} \mathrm{S}_1 \\ \tilde{\mathrm{S}}_2 \end{bmatrix}^*,$$

  where $\tilde{\mathrm{R}}_2 = \mathrm{R}_2 - \mathrm{F}\mathrm{A}^{-1}\mathrm{R}_1$ and $\tilde{\mathrm{S}}_2 = \mathrm{S}_2 - \mathrm{G}(\mathrm{A}^*)^{-1}\mathrm{S}_1$, one may efficiently perform row reductions with the *generators* of the Cauchy-like matrix, instead of directly performing operations on the dense matrix itself!

- To *avoid* the exhaustive global search in total pivoting, a suitable pivot may also be found from the generators using the technique proposed in [1]. This involves computing a QR factorization for one of the generators at every step.

The asymptotic complexity

The second step is most expensive. Assuming proper *QR updating techniques are utilized*, each step of Gauss elimination takes $\mathcal{O}(S^2 d^3)$ flops and $r(d)$ steps are required in total. Given (3), the complexity $\mathcal{O}(d_\Sigma^5)$ is attained.

References

[1] M. Gu, *Stable and efficient algorithms for structured systems of linear equations*, SIAM journal on matrix analysis and applications, 19 (1998), pp. 279–306.

[2] J. Vanderstukken and L. De Lathauwer, *Systems of polynomial equations, higher-order tensor decompositions and multidimensional harmonic retrieval: A unifying framework. part i: The canonical polyadic decomposition*, SIAM Journal On Matrix Analysis And Applications, 42 (2021), pp. 883–912.