Minimal determinantal representations of bivariate polynomials

Bor Plestenjak (University of Ljubljana)

Outline

- Two-parameter eigenvalue problem
- Main idea
- Determinantal representations
- Minimal representation
- Minimal uniform representation

Two-parameter eigenvalue problem

Two-parameter eigenvalue problem:

$$\begin{aligned}(A_1 + \lambda B_1 + \mu C_1)x &= 0\\ (A_2 + \lambda B_2 + \mu C_2)y &= 0,\end{aligned} \tag{2EP}$$

where A_i, B_i, C_i are $n \times n$ matrices, $\lambda, \mu \in \mathbb{C}$, $x, y \in \mathbb{C}^n$.

Eigenvalue: a pair (λ, μ) that satisfies (2EP) for nonzero x and y.

Eigenvector: the tensor product $x \otimes y$.

There are n^2 eigenvalues, which are solutions of

$$\begin{aligned}\det(A_1 + \lambda B_1 + \mu C_1) &= 0\\ \det(A_2 + \lambda B_2 + \mu C_2) &= 0.\end{aligned}$$

Tensor product approach

$$\begin{aligned}(A_1 + \lambda B_1 + \mu C_1)x &= 0 \\ (A_2 + \lambda B_2 + \mu C_2)y &= 0\end{aligned} \qquad \text{(2EP)}$$

On $\mathbb{C}^n \otimes \mathbb{C}^n$ we define $n^2 \times n^2$ matrices, so called operator determinants

$$\begin{aligned}\Delta_0 &= \begin{vmatrix} B_1 & C_1 \\ B_2 & C_2 \end{vmatrix}_\otimes = B_1 \otimes C_2 - C_1 \otimes B_2 \\ \Delta_1 &= \begin{vmatrix} C_1 & A_1 \\ C_2 & A_2 \end{vmatrix}_\otimes = C_1 \otimes A_2 - A_1 \otimes C_2 \\ \Delta_2 &= \begin{vmatrix} A_1 & B_1 \\ A_2 & B_2 \end{vmatrix}_\otimes = A_1 \otimes B_2 - B_1 \otimes A_2.\end{aligned}$$

Nonsingular 2EP ($\Longleftrightarrow \Delta_0$ is nonsingular) is equivalent to a coupled GEP

$$\begin{aligned}\Delta_1 z &= \lambda \Delta_0 z \\ \Delta_2 z &= \mu \Delta_0 z,\end{aligned} \qquad (\Delta)$$

where $\Delta_0^{-1}\Delta_1$ and $\Delta_0^{-1}\Delta_2$ commute and $z = x \otimes y$. (Atkinson 1970)

Using this relation we can compute all eigenpairs of a nonsingular (2EP).

Singular two-parameter eigenvalue problem

$$\begin{aligned} A_1x &= \lambda B_1x + \mu C_1x \\ A_2y &= \lambda B_2y + \mu C_2y \end{aligned}$$
(2EP)

$$\begin{aligned} \Delta_0 &= B_1 \otimes C_2 - C_1 \otimes B_2 \\ \Delta_1 &= A_1 \otimes C_2 - C_1 \otimes A_2 \\ \Delta_2 &= B_1 \otimes A_2 - A_1 \otimes B_2 \end{aligned}$$

$$\begin{aligned} \Delta_1 z &= \lambda \Delta_0 z \\ \Delta_2 z &= \mu \Delta_0 z \end{aligned}$$
(Δ)

Singular 2EP: All linear combinations of Δ_0, Δ_1, and Δ_2 are singular.

We can numerically extract the common finite regular part of matrix pencils (Δ) using the modified staircase algorithm. (Muhič, P. '09)

We get $\widetilde{\Delta}_0, \widetilde{\Delta}_1$, and $\widetilde{\Delta}_2$ such that $\widetilde{\Delta}_0$ is nonsingular and eigenvalues of

$$\begin{aligned} \widetilde{\Delta}_1 \widetilde{z} &= \lambda \widetilde{\Delta}_0 \widetilde{z} \\ \widetilde{\Delta}_2 \widetilde{z} &= \mu \widetilde{\Delta}_0 \widetilde{z} \end{aligned}$$

are exactly the finite regular eigenvalues of (Δ).

Motivation

For each monic polynomial $p(x) = p_0 + p_1 x + \cdots + p_{n-1}x^{n-1} + x^n$ we can construct a matrix $A \in \mathbb{C}^{n\times n}$, such that $\det(xI - A) = p(x)$.

One option is the companion matrix

$$A_p = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ 0 & 0 & & & 1 \\ -p_0 & -p_1 & \cdots & \cdots & -p_{n-1} \end{bmatrix}.$$

We can then compute the zeros of p as eigenvalues of A_p using standard eigenvalue solvers.

That is how function `roots` works in Matlab.

Can we do something similar for a system of bivariate polynomials?

The main idea (P., Hochstenbach, SISC 2016)

We want to compute the zeros of a system of two bivariate polynomials

$$p(x,y) := \sum_{i=0}^{n} \sum_{j=0}^{n-i} p_{ij}\, x^i\, y^j = 0,$$

$$q(x,y) := \sum_{i=0}^{n} \sum_{j=0}^{n-i} q_{ij}\, x^i\, y^j = 0.$$

An analogous approach: find matrices $A_1, B_1, C_1, A_2, B_2, C_2$ such that

$$\det(A_1 + xB_1 + yC_1) = p(x,y),$$
$$\det(A_2 + xB_2 + yC_2) = q(x,y),$$

we call these determinantal representations or linearizations of p and q.

This gives a two-parameter eigenvalue problem

$$(A_1 + xB_1 + yC_1)\, u_1 = 0,$$
$$(A_2 + xB_2 + yC_2)\, u_2 = 0$$

such that its finite regular eigenvalues are the zeros of $p(x,y) = 0$, $q(x,y) = 0$.

Theory

A pencil $A + xB + yC$ is a determinantal representation of $p(x, y)$ of size n if

$$p(x, y) = \det(A + xB + yC)$$

and A, B, C are $n \times n$ matrices.

Dixon (1902)

For each scalar bivariate polynomial of degree n there exists a determinantal representation of size n with symmetric matrices.

Dickson (1921)

The above result can not be generalized to generic polynomials in three or more variables except for polynomials in three variables of degrees 2 and 3 and polynomials in four variables of degree 2.

Overview of determinantal representations

Although it is known since 1902 that each bivariate polynomial of degree n,

$$p(x,y) = \sum_{i=0}^{n} \sum_{j=0}^{n-i} p_{ij} x^i y^j,$$

admits a determinantal representation of size n, there is no simple construction.

We can use larger matrices, but then the 2EP is singular. The idea is to find a linearization that can be constructed fast and is as small as possible.

Overview of determinantal representations

Although it is known since 1902 that each bivariate polynomial of degree n,

$$p(x,y) = \sum_{i=0}^{n} \sum_{j=0}^{n-i} p_{ij} x^i y^j,$$

admits a determinantal representation of size n, there is no simple construction.

We can use larger matrices, but then the 2EP is singular. The idea is to find a linearization that can be constructed fast and is as small as possible.

Uniform representations (no computation) for bivariate polynomials of degree n:

- Khazanov (2007): size n^2,
- Muhič, P. (2010): asymptotic size $n^2/2$,
- Quarez (2012): symmetric matrices of asymptotic size $n^2/4$.
- P., Hochstenbach (2014): asymptotic size $n^2/4$.
- Boralevi, van Doornmalen, Draisma, Hochstenbach, P. (2016): size $2n-1$.

Overview of determinantal representations

Although it is known since 1902 that each bivariate polynomial of degree n,

$$p(x,y)=\sum_{i=0}^{n}\sum_{j=0}^{n-i} p_{ij}x^i y^j,$$

admits a determinantal representation of size n, there is no simple construction.

We can use larger matrices, but then the 2EP is singular. The idea is to find a linearization that can be constructed fast and is as small as possible.

Uniform representations (no computation) for bivariate polynomials of degree n:

- Khazanov (2007): size n^2,
- Muhič, P. (2010): asymptotic size $n^2/2$,
- Quarez (2012): symmetric matrices of asymptotic size $n^2/4$.
- P., Hochstenbach (2014): asymptotic size $n^2/4$.
- Boralevi, van Doornmalen, Draisma, Hochstenbach, P. (2016): size $2n-1$.

Non-uniform representations (computation is required):

- P., Hochstenbach (2014): asymptotic size $n^2/6$.
- Buckley, P. (2016): size n for any polynomial of degree $n \leq 5$,
- P. (2017): size n for any polynomial.

Homogenization

We can homogenize $p(x, y)$ of degree n into

$$p_h(x, y, z) := z^n p\left(\frac{x}{z}, \frac{y}{z}\right) = \sum_{i=0}^{n} \sum_{j=0}^{n-i} p_{ij}\, x^i\, y^j z^{n-i-j}.$$

Clearly, if $\det(zA + xB + yC) = p_h(x, y, z)$, then $\det(A + xB + yC) = p(x, y)$.

The homogeneous form gives us are more freedom, as we can apply a linear change of variables

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = T \begin{bmatrix} \widetilde{x} \\ \widetilde{y} \\ \widetilde{z} \end{bmatrix}$$

to transform the polynomial into an appropriate initial form.

Initial transformation

We have a bivariate polynomial

$$p(x, y) := \sum_{i=0}^{n} \sum_{j=0}^{n-i} p_{ij}\, x^i\, y^j.$$

If p is a square-free polynomial, then after a linear transformation of variables we can assume that

a) $p_{n0} \neq 0$,

b) $p_{0n} = p_{0,n-1} = 0$,

c) all zeros $\xi_1, \ldots, \xi_{n-1}$ of the polynomial

$$v(\xi) := p_{n0}\xi^{n-1} + p_{n-1,1}\xi^{n-2} + \cdots + p_{1,n-1}$$

are simple and nonzero.

Linerization of size n for a square-free polynomial

We apply bivariate polynomials $q_0, \ldots, q_{n-1}$, defined recursively as

$$\begin{aligned} q_0(x,y) &= 1, \\ q_1(x,y) &= f_{11} q_0(x,y), \\ q_2(x,y) &= f_{21} q_1(x,y) + f_{22} q_0(x,y), \\ &\vdots \\ q_{n-1}(x,y) &= f_{n-1,1} q_{n-2}(x,y) + f_{n-1,2} q_{n-3}(x,y) + \cdots + f_{n-1,n-1} q_0(x,y), \end{aligned}$$

where $f_{ij} = \alpha_{ij} x + \beta_{ij} y$ and $\alpha_{ij} = 1$ for $i < n-1$.

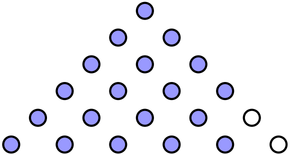
$$A + xB + yC = \begin{bmatrix} \gamma_{00} + \gamma_{10} x & \gamma_1 & \gamma_2 & \cdots & \gamma_{n-2} & p_{n0} x \\ -f_{11} & 1 & & & & \\ -f_{22} & -f_{21} & 1 & & & \\ -f_{33} & -f_{32} & -f_{31} & 1 & & \\ \vdots & & & \ddots & \ddots & \\ -f_{n-1,n-1} & -f_{n-1,n-2} & \cdots & \cdots & -f_{n-1,1} & 1 \end{bmatrix}.$$

$\det(A+xB+yC) = \gamma_{00} + \gamma_{10} x + \gamma_1 q_1(x,y) + \cdots + \gamma_{n-2} q_{n-2}(x,y) + p_{n0} x q_{n-1}(x,y)$.
We can find $\alpha_{ij}, \beta_{ij}, \gamma_{00}, \gamma_{10}, \gamma_1, \ldots, \gamma_{n-2}$ such that $\det(A+xB+yC) = p(x,y)$.

Sketch of the algorithm for $n = 5$

$p_5(x, y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

$p_5(x, y)$

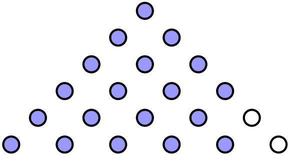


Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$

$p_5(x,y)$



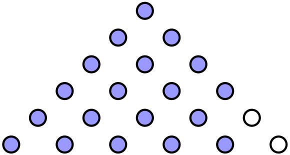
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$q_0 = 1,$$
$$q_1 = (x - \zeta_1 y)q_0,$$
$$q_2 = (x - \zeta_2 y)q_1$$
$$q_3 = (x - \zeta_3 y)q_2$$
$$q_4 = (x - \zeta_4 y)q_3$$

$p_5(x,y)$



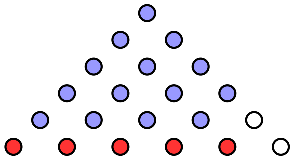
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$q_0 = 1,$$
$$q_1 = (x - \zeta_1 y)q_0,$$
$$q_2 = (x - \zeta_2 y)q_1$$
$$q_3 = (x - \zeta_3 y)q_2$$
$$q_4 = (x - \zeta_4 y)q_3$$

$p_5(x,y) - p_{50}xq_4(x,y)$



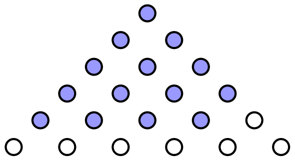
Sketch of the algorithm for $n = 5$

$p_5(x, y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$
\begin{aligned}
q_0 &= 1,\\
q_1 &= (x - \zeta_1 y)q_0,\\
q_2 &= (x - \zeta_2 y)q_1\\
q_3 &= (x - \zeta_3 y)q_2\\
q_4 &= (x - \zeta_4 y)q_3
\end{aligned}
$$

$p_5(x, y) - p_{50}xq_4(x, y)$



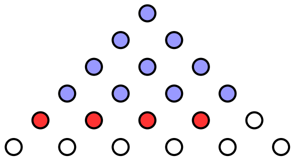
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$
\begin{aligned}
q_0 &= 1,\\
q_1 &= (x - \zeta_1 y)q_0,\\
q_2 &= (x - \zeta_2 y)q_1 + f_{22}q_0,\\
q_3 &= (x - \zeta_3 y)q_2 + f_{32}q_1\\
q_4 &= (x - \zeta_4 y)q_3 + f_{42}q_2
\end{aligned}
$$

$p_5(x,y) - p_{50}xq_4(x,y)$



Find:

- $f_{22} = x - \beta_{22}y$,
- $f_{32} = x - \beta_{32}y$,
- $f_{42} = \alpha_{42}x - \beta_{42}y$

to annihilate monomials of degree 4

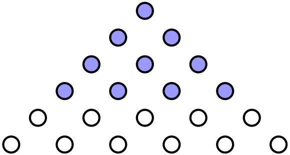
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$
\begin{aligned}
q_0 &= 1,\\
q_1 &= (x - \zeta_1 y)q_0,\\
q_2 &= (x - \zeta_2 y)q_1 + f_{22}q_0,\\
q_3 &= (x - \zeta_3 y)q_2 + f_{32}q_1\\
q_4 &= (x - \zeta_4 y)q_3 + f_{42}q_2
\end{aligned}
$$

$p_5(x,y) - p_{50}xq_4(x,y)$



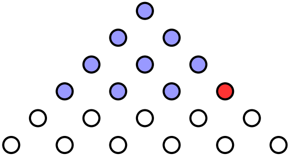
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$
\begin{aligned}
q_0 &= 1,\\
q_1 &= (x - \zeta_1 y)q_0,\\
q_2 &= (x - \zeta_2 y)q_1 + f_{22}q_0,\\
q_3 &= (x - \zeta_3 y)q_2 + f_{32}q_1\\
q_4 &= (x - \zeta_4 y)q_3 + f_{42}q_2
\end{aligned}
$$

$$p_5(x,y) - p_{50}xq_4(x,y) - \gamma_3 q_3(x,y)$$



Find γ_3 to zero the coefficient at y^3 using q_3.

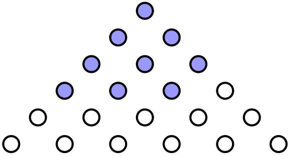
Sketch of the algorithm for $n = 5$

$p_5(x, y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y) \cdots (x - \zeta_4 y)$ and define

$$
\begin{aligned}
q_0 &= 1, \\
q_1 &= (x - \zeta_1 y)q_0, \\
q_2 &= (x - \zeta_2 y)q_1 + f_{22}q_0, \\
q_3 &= (x - \zeta_3 y)q_2 + f_{32}q_1 \\
q_4 &= (x - \zeta_4 y)q_3 + f_{42}q_2
\end{aligned}
$$

$p_5(x, y) - p_{50}xq_4(x, y) - \gamma_3 q_3(x, y)$



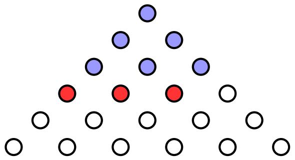
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x-\zeta_1 y)\cdots(x-\zeta_4 y)$ and define

$$
\begin{aligned}
q_0 &= 1,\\
q_1 &= (x-\zeta_1 y)q_0,\\
q_2 &= (x-\zeta_2 y)q_1 + f_{22}q_0,\\
q_3 &= (x-\zeta_3 y)q_2 + f_{32}q_1 + f_{33}q_0,\\
q_4 &= (x-\zeta_4 y)q_3 + f_{42}q_2 + f_{43}q_1
\end{aligned}
$$

$$p_5(x,y) - p_{50}xq_4(x,y) - \gamma_3 q_3(x,y)$$



Find:

- $f_{33} = x - \beta_{33}y$,
- $f_{43} = \alpha_{43}x - \beta_{43}y$

to annihilate monomials of degree 3

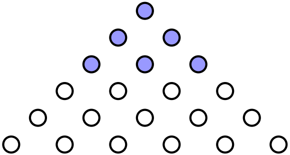
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$
\begin{aligned}
q_0 &= 1,\\
q_1 &= (x - \zeta_1 y)q_0,\\
q_2 &= (x - \zeta_2 y)q_1 + f_{22}q_0,\\
q_3 &= (x - \zeta_3 y)q_2 + f_{32}q_1 + f_{33}q_0,\\
q_4 &= (x - \zeta_4 y)q_3 + f_{42}q_2 + f_{43}q_1
\end{aligned}
$$

$p_5(x,y) - p_{50}xq_4(x,y) - \gamma_3 q_3(x,y)$



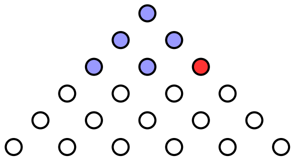
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$p_5(x,y) - p_{50}xq_4(x,y) - \gamma_3 q_3(x,y) - \gamma_2 q_2(x,y)$$

$$\begin{aligned}
q_0 &= 1,\\
q_1 &= (x - \zeta_1 y)q_0,\\
q_2 &= (x - \zeta_2 y)q_1 + f_{22}q_0,\\
q_3 &= (x - \zeta_3 y)q_2 + f_{32}q_1 + f_{33}q_0,\\
q_4 &= (x - \zeta_4 y)q_3 + f_{42}q_2 + f_{43}q_1
\end{aligned}$$



Find γ_2 to zero the coefficient at y^2 using q_2.

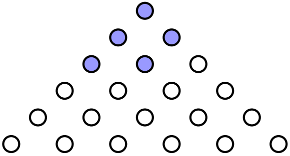
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$
\begin{aligned}
q_0 &= 1,\\
q_1 &= (x - \zeta_1 y)q_0,\\
q_2 &= (x - \zeta_2 y)q_1 + f_{22}q_0,\\
q_3 &= (x - \zeta_3 y)q_2 + f_{32}q_1 + f_{33}q_0,\\
q_4 &= (x - \zeta_4 y)q_3 + f_{42}q_2 + f_{43}q_1
\end{aligned}
$$

$p_5(x,y) - p_{50}xq_4(x,y) - \gamma_3 q_3(x,y) - \gamma_2 q_2(x,y)$



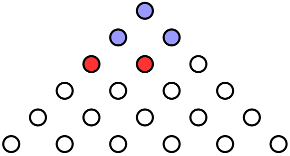
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$
\begin{aligned}
q_0 &= 1,\\
q_1 &= (x - \zeta_1 y)q_0,\\
q_2 &= (x - \zeta_2 y)q_1 + f_{22}q_0,\\
q_3 &= (x - \zeta_3 y)q_2 + f_{32}q_1 + f_{33}q_0,\\
q_4 &= (x - \zeta_4 y)q_3 + f_{42}q_2 + f_{43}q_1 + f_{44}q_0,
\end{aligned}
$$

$$p_5(x,y) - p_{50}xq_4(x,y) - \gamma_3 q_3(x,y) - \gamma_2 q_2(x,y)$$



Find:

- $f_{44} = \alpha_{44}x - \beta_{44}y$

to annihilate monomials of degree 2

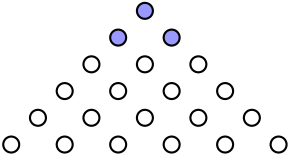
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$\begin{aligned}
q_0 &= 1,\\
q_1 &= (x - \zeta_1 y)q_0,\\
q_2 &= (x - \zeta_2 y)q_1 + f_{22}q_0,\\
q_3 &= (x - \zeta_3 y)q_2 + f_{32}q_1 + f_{33}q_0,\\
q_4 &= (x - \zeta_4 y)q_3 + f_{42}q_2 + f_{43}q_1 + f_{44}q_0,
\end{aligned}$$

$$p_5(x,y) - p_{50}xq_4(x,y) - \gamma_3 q_3(x,y) - \gamma_2 q_2(x,y)$$



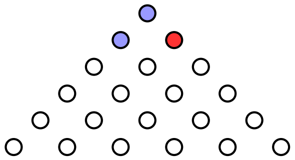
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x-\zeta_1 y)\cdots(x-\zeta_4 y)$ and define

$$\begin{aligned} q_0 &= 1, \\ q_1 &= (x-\zeta_1 y)q_0, \\ q_2 &= (x-\zeta_2 y)q_1 + f_{22}q_0, \\ q_3 &= (x-\zeta_3 y)q_2 + f_{32}q_1 + f_{33}q_0, \\ q_4 &= (x-\zeta_4 y)q_3 + f_{42}q_2 + f_{43}q_1 + f_{44}q_0, \end{aligned}$$

$$p_5(x,y) - p_{50}xq_4(x,y) - \gamma_3 q_3(x,y) - \gamma_2 q_2(x,y) - \gamma_1 q_1(x,y)$$



Find γ_1 to zero the coefficient at y using q_1.

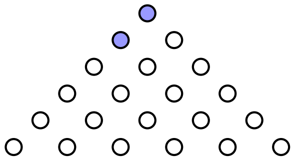
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$
\begin{aligned}
q_0 &= 1,\\
q_1 &= (x - \zeta_1 y)q_0,\\
q_2 &= (x - \zeta_2 y)q_1 + f_{22}q_0,\\
q_3 &= (x - \zeta_3 y)q_2 + f_{32}q_1 + f_{33}q_0,\\
q_4 &= (x - \zeta_4 y)q_3 + f_{42}q_2 + f_{43}q_1 + f_{44}q_0,
\end{aligned}
$$

$$p_5(x,y) - p_{50}xq_4(x,y) - \gamma_3 q_3(x,y) - \gamma_2 q_2(x,y) - \gamma_1 q_1(x,y)$$



The residual is $r(x,y) = \gamma_{00} + \gamma_{10}x$.

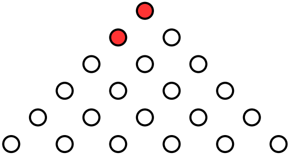
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$
\begin{aligned}
q_0 &= 1,\\
q_1 &= (x - \zeta_1 y)q_0,\\
q_2 &= (x - \zeta_2 y)q_1 + f_{22}q_0,\\
q_3 &= (x - \zeta_3 y)q_2 + f_{32}q_1 + f_{33}q_0,\\
q_4 &= (x - \zeta_4 y)q_3 + f_{42}q_2 + f_{43}q_1 + f_{44}q_0,
\end{aligned}
$$

$$
p_5(x,y) - p_{50}xq_4(x,y) - \gamma_3 q_3(x,y) - \gamma_2 q_2(x,y) - \gamma_1 q_1(x,y) - (\gamma_{00} + \gamma_{10}x)q_0(x,y)
$$



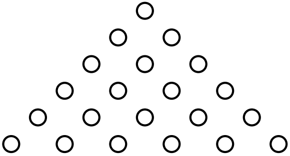
Sketch of the algorithm for $n = 5$

$p_5(x,y) = p_{00} + \cdots + p_{50}x^5 + p_{41}x^4y + \cdots + p_{14}xy^4$, where $p_{50} \neq 0$, $p_{05} = p_{04} = 0$.

Factor $p_{50}x^5 + \cdots + p_{14}xy^4 = p_{50}x(x - \zeta_1 y)\cdots(x - \zeta_4 y)$ and define

$$
\begin{aligned}
q_0 &= 1,\\
q_1 &= (x - \zeta_1 y)q_0,\\
q_2 &= (x - \zeta_2 y)q_1 + f_{22}q_0,\\
q_3 &= (x - \zeta_3 y)q_2 + f_{32}q_1 + f_{33}q_0,\\
q_4 &= (x - \zeta_4 y)q_3 + f_{42}q_2 + f_{43}q_1 + f_{44}q_0,
\end{aligned}
$$

$$
\begin{aligned}
&p_5(x,y) - p_{50}xq_4(x,y) - \gamma_3 q_3(x,y) - \gamma_2 q_2(x,y)\\
&\quad - \gamma_1 q_1(x,y) - (\gamma_{00} + \gamma_{10}x)q_0(x,y)
\end{aligned}
$$



$$
A + xB + yC = \begin{bmatrix} \gamma_{00} + \gamma_{10}x & \gamma_1 & \gamma_2 & \gamma_3 & p_{50}x \\ -x + \zeta_1 y & 1 & & & \\ -f_{22} & -x + \zeta_2 y & 1 & & \\ -f_{33} & -f_{32} & -x + \zeta_3 y & 1 & \\ -f_{44} & -f_{43} & -f_{42} & -x + \zeta_4 y & 1 \end{bmatrix}.
$$

Representation of non square-free polynomials

For such polynomials we have several options:

a) For $n \leq 5$ we can apply an algorithm (Buckley, P.) that works for non square-free polynomials as well.

b) We factorize p into a product

$$p(x,y) = p_1(x,y)p_2(x,y)\dots p_k(x,y),$$

where p_i is a square-free polynomial for $i = 1,\dots,k$. Now we apply algorithm to obtain matrices A_i, B_i, and C_i such that $p_i(x,y) = \det(A_i + xB_i + yC_i)$ for $i = 1,\dots,k$ and arrange them in block diagonal matrices A, B and C. Combined with a square-free factorization, one can thus find a determinantal representation of size n for each bivariate polynomial of degree n.

c) Square-free factorization is expensive. In our case, where we use representations to solve a system of two polynomials, it is more efficient to use larger representations that can be constructed faster.

Minimal uniform representations

(Boralevi, van Doornmalen, Draisma, Hochstenbach, P., to appear in SIAGA)

For a bivariate polynomial of max degree n there exists a uniform linearization of size $2n+1$. In case $n=4$,

$$p(x,y)=\sum_{i=0}^{4}\sum_{j=0}^{4}p_{ij}x^i y^j=\det(A+xB+yC)$$

for

$$A+xB+yC=\begin{bmatrix}-1 & x & & & & & & & \\ & -1 & x & & & & & & \\ & & -1 & x & & & & & \\ & & & -1 & x & & & & \\ p_{44} & p_{34} & p_{24} & p_{14} & p_{04} & -1 & & & \\ p_{43} & p_{33} & p_{23} & p_{13} & p_{03} & y & -1 & & \\ p_{42} & p_{32} & p_{23} & p_{12} & p_{02} & & y & -1 & \\ p_{41} & p_{31} & p_{21} & p_{11} & p_{01} & & & y & -1 \\ p_{40} & p_{30} & p_{20} & p_{10} & p_{00} & & & & y\end{bmatrix}.$$

Linerization of size $2n-1$ for any bivariate polynomial

Slight modification of previous results gives a uniform representation of size $2n-1$ for a bivariate polynomial of degree n.

In case $n=4$, $p_4(x,y) = \det(A + xB + yC)$ for

$$A+xB+yC = \begin{bmatrix} -1 & x & & & & & \\ & -1 & x & & & & \\ & & -1 & x & & & \\ & & & p_{13}x + p_{04}y & -1 & & \\ & & p_{12} + p_{22}x & p_{02} + p_{03}y & y & -1 & \\ & p_{21} + p_{31}x & p_{11} & p_{01} & & y & 1 \\ p_{30} + xp_{40} & p_{20} & p_{10} & p_{00} & & & -y \end{bmatrix}$$

Some numerical results

This is implemented in package BiRoots, available from Matlab Central File Exchange. We compared the code to NSolve in Mathematica and PHCLab for Matlab.

Table: Average computational times in milliseconds for `MinRep`, `MinUnif`, `NSolve`, and `PHCLab` for random bivariate polynomial systems of degree 3 to 10. For `MinUnif` and `NSolve` separate results are included for real ($\mathbb{R}$) and complex polynomials ($\mathbb{C}$).

d	`MinRep`	`MinUnif` ($\mathbb{R}$)	`MinUnif` ($\mathbb{C}$)	`NSolve` ($\mathbb{R}$)	NSolve ($\mathbb{C}$)	`PHCLab`
3	6	6	6	10	55	210
4	8	9	11	18	93	247
5	11	15	18	34	147	289
6	15	25	32	56	225	344
7	20	40	55	169	525	409
8	29	70	98	177	548	499
9	46	112	172	419	913	607
10	64	184	301	990	2219	739

`MinRep` is fast but not accurate enough for polynomials of degree $n \geq 11$.

Some numerical results

This is implemented in package BiRoots, available from Matlab Central File Exchange. We compared the code to NSolve in Mathematica and PHCLab for Matlab.

Table: Average computational times in milliseconds for `MinRep`, `MinUnif`, `NSolve`, and `PHCLab` for random bivariate polynomial systems of degree 3 to 10. For `MinUnif` and `NSolve` separate results are included for real ($\mathbb{R}$) and complex polynomials ($\mathbb{C}$).

d	`MinRep`	`MinUnif` ($\mathbb{R}$)	`MinUnif` ($\mathbb{C}$)	`NSolve` ($\mathbb{R}$)	NSolve ($\mathbb{C}$)	`PHCLab`
3	6	6	6	10	55	210
4	8	9	11	18	93	247
5	11	15	18	34	147	289
6	15	25	32	56	225	344
7	20	40	55	169	525	409
8	29	70	98	177	548	499
9	46	112	172	419	913	607
10	64	184	301	990	2219	739

`MinRep` is fast but not accurate enough for polynomials of degree $n \geq 11$.

Thank you for your attention!