Computing Popov and Hermite forms of rectangular polynomial matrices

ISSAC 2018 (New York, USA)

Vincent Neiger, Johan Rosenkilde, Grigory Solomatov

XLIM – University of Limoges, France
Technical University of Denmark

July 17, 2018

Outline

- Context and contribution
- Algorithmic tools and general approach
- Overview of new algorithms

Polynomial matrix computations

$\mathbb{K}$ a field

Matrices over $\mathbb{K}[X]$

$$\begin{bmatrix} 3X^3+X^2+5X+3 & 6X+5 & 2X+1 \\ 5 & 5X^2+3X+1 & 5X+3 \\ 3X+4 & X^3+4X+1 & 4X^2+3 \end{bmatrix}$$

Usual matrix operations

- matrix multiplication
- rank, determinant
- system solving, inversion

Transformations to normal forms

- triangularization $\rightsquigarrow$ Hermite form
- row reduction $\rightsquigarrow$ Popov form
- diagonalization $\rightsquigarrow$ Smith form

Hermite and Popov forms ($\mathbb{K} = \mathbb{Z}/7\mathbb{Z}$)

Input matrix: $\begin{bmatrix} 3X^3+X^2+5X+3 & 6X+5 & 2X+1 \\ 5 & 5X^2+3X+1 & 5X+3 \\ 3X+4 & X^3+4X+1 & 4X^2+3 \end{bmatrix}$

Transform, via elementary row operations,

$$\begin{cases} \text{row}_i \leftarrow \text{row}_i + p(X)\text{row}_j \\ \text{row}_i \leftrightarrow \text{row}_j \\ \text{row}_i \leftarrow \alpha\,\text{row}_i,\ \alpha \in \mathbb{K}\backslash\{0\} \end{cases}$$

$\rightsquigarrow$ into Popov form [Popov '72]

$$\begin{bmatrix} X^3+5X^2+4X+1 & 2X+4 & 3X+5 \\ 1 & X^2+2X+3 & X+2 \\ 3X+2 & 4X & X^2 \end{bmatrix}$$

$\rightsquigarrow$ into Hermite form [Hermite 1851]

$$\begin{bmatrix} X^6+6X^4+X^3+X+4 & 0 & 0 \\ 5X^5+5X^4+6X^3+2X^2+6X+3 & X & 0 \\ 3X^4+5X^3+4X^2+6X+1 & 5 & 1 \end{bmatrix}$$

Goal: fast algorithms

$\mathbb{K}$ a field

Matrices over $\mathbb{K}[X]$

$$\begin{bmatrix} 3X^3+X^2+5X+3 & 6X+5 & 2X+1 \\ 5 & 5X^2+3X+1 & 5X+3 \\ 3X+4 & X^3+4X+1 & 4X^2+3 \end{bmatrix}$$

Usual matrix operations

- matrix multiplication
- rank, determinant
- system solving, inversion

Transformations to normal forms

- triangularization $\rightsquigarrow$ Hermite form
  [Gupta-Storjohann '11] [Labahn-Neiger-Zhou '17]
- row reduction $\rightsquigarrow$ Popov form
  [Gupta-Sarkar-Storjohann-Valeriote '11 & '12]
- diagonalization $\rightsquigarrow$ Smith form

Goal: fast algorithms

$\mathbb{K}$ a field

Matrices over $\mathbb{K}[X]$

$$\begin{bmatrix} 3X^3+X^2+5X+3 & 6X+5 & 2X+1 \\ 5 & 5X^2+3X+1 & 5X+3 \\ 3X+4 & X^3+4X+1 & 4X^2+3 \end{bmatrix}$$

Usual matrix operations

- matrix multiplication
- rank, determinant
- system solving, inversion

Transformations to normal forms

- triangularization $\rightsquigarrow$ Hermite form
  [Gupta-Storjohann '11] [Labahn-Neiger-Zhou '17]
- row reduction $\rightsquigarrow$ Popov form
  [Gupta-Sarkar-Storjohann-Valeriote '11 & '12]
- diagonalization $\rightsquigarrow$ Smith form

cost $\tilde{\mathcal{O}}(m^\omega d)$

$m \times m$ matrix
with degree d

Contribution: rectangular normal forms

For a matrix in $\mathbb{K}[X]^{m \times n}$ with $m \leqslant n$:

Popov form

deterministic algorithm with cost $\tilde{\mathcal{O}}(m^{\omega-1}nd)$

d the degree of the matrix
size of Popov form is $\mathcal{O}(mnd)$

previous fastest: $\mathcal{O}(rmnd^2)$ [Mulders-Storjohann '03] and $\tilde{\mathcal{O}}(mn^{\omega}d)$ [$\star$]

Las Vegas: $\tilde{\mathcal{O}}(m^{\omega-1}nd)$ assuming full row rank [Sarkar-Storjohann '11]

[$\star$] = based on some kernel computation [Beckermann-Labahn-Villard '06]

Contribution: rectangular normal forms

For a matrix in $\mathbb{K}[X]^{m \times n}$ with $m \leqslant n$:

Hermite form

deterministic algorithm with cost $\mathcal{O}^{\sim}(m^{\omega-1} n \delta)$

$\delta \leqslant md$

$\delta :=$ min(sum of row/col degrees)

size of Hermite form can be $\Theta(mn\delta)$

previous fastest: $\mathcal{O}^{\sim}(n^{\omega+1}\delta)$ [$\star$] (speed-up factor $\geqslant n$)

[$\star$] = based on some kernel computation [Beckermann-Labahn-Villard '06]

Tool: kernel bases

Left kernel basis of a matrix $\mathbf{A} \in \mathbb{K}[X]^{m \times n}$

Matrix $\mathbf{K} \in \mathbb{K}[X]^{k \times m}$ such that
$$\begin{cases} \mathbf{K} \text{ has full row rank} \\ \mathbf{KA} = \mathbf{0} \\ \text{rows of } \mathbf{K} \text{ generate the kernel of } \mathbf{A} \end{cases}$$

- core algorithmic tool: rank, inversion, determinant, Hermite form, ...
- kernel bases can now be computed fast in Popov form
  $\rightsquigarrow$ combine [Zhou-Labahn-Storjohann '12] + this work
- shifted normal forms: $\mathbf{UA} = \mathbf{P}$
  (via shifted Popov approximant basis [Jeannerod-Neiger-Schost-Villard '16])
  $\rightsquigarrow$ cost of this approach: unsatisfactory

Reduction to the full row rank case

Row basis of a matrix $\mathbf{A} \in \mathbb{K}[X]^{m \times n}$

Matrix $\mathbf{B} \in \mathbb{K}[X]^{r \times n}$ such that $\begin{cases} \mathbf{B} \text{ has full row rank} \\ \mathbf{B} \text{ and } \mathbf{A} \text{ have the same row space} \end{cases}$

Fast algorithm: [Zhou-Labahn '13]

Consequence: deal with $m \geqslant n$ and rank-deficient matrices

Normal form of arbitrary $\mathbf{A}$

Step 1: $\mathbf{B} \leftarrow$ row basis of $\mathbf{A}$ `// use [Zhou-Labahn '13]`
Step 2: $\mathbf{P} \leftarrow$ the normal form of $\mathbf{B}$ `// full row rank case`
Step 3: Return $\mathbf{P}$

Step 1 costs $\tilde{\mathcal{O}}(m^{\omega-1}(m+n)d)$

What do pivots become in the full row rank case?

Degree structure and pivots

nonsingular, $\mathrm{m} \times \mathrm{m}$

pivot of a row: rightmost entry of largest degree

$$\begin{bmatrix} [6] & [1] & [2] & [4] \\ [2] & [2] & [1] & [1] \\ [3] & [1] & [3] & [2] \\ [5] & [1] & [2] & [5] \end{bmatrix}$$

Popov

size $\mathcal{O}(\mathrm{m}^2\mathrm{d})$

pivot of a row: rightmost nonzero entry

$$\begin{bmatrix} [12] & & & \\ [11] & [3] & & \\ [11] & [2] & [0] & \\ [11] & [2] & & [1] \end{bmatrix}$$

Hermite

size $\mathcal{O}(\mathrm{m}^2\mathrm{d})$

Degree structure and pivots

nonsingular, $m \times m$ | full row rank, $m \times n$

pivot of a row: rightmost entry of largest degree

$$\begin{bmatrix} [6] & [1] & [2] & [4] \\ [2] & [2] & [1] & [1] \\ [3] & [1] & [3] & [2] \\ [5] & [1] & [2] & [5] \end{bmatrix}$$

size $\mathcal{O}(m^2 d)$

Popov

$$\begin{bmatrix} [6] & [5] & [5] & [1] & [5] & [2] & [4] \\ [2] & [2] & [2] & [2] & [1] & [1] & [1] \\ [3] & [3] & [3] & [1] & [3] & [3] & [2] \\ [5] & [5] & [5] & [1] & [5] & [2] & [5] \end{bmatrix}$$

size $\mathcal{O}(mnd)$

pivot of a row: rightmost nonzero entry

$$\begin{bmatrix} [12] & & & \\ [11] & [3] & & \\ [11] & [2] & [0] & \\ [11] & [2] & & [1] \end{bmatrix}$$

size $\mathcal{O}(m^2 d)$

Hermite

Degree structure and pivots

nonsingular, $\mathrm{m} \times \mathrm{m}$ | full row rank, $\mathrm{m} \times \mathrm{n}$

pivot of a row: rightmost entry of largest degree

$$\begin{bmatrix} [6] & [1] & [2] & [4] \\ [2] & [2] & [1] & [1] \\ [3] & [1] & [3] & [2] \\ [5] & [1] & [2] & [5] \end{bmatrix}$$

size $\mathcal{O}(\mathrm{m}^2\mathrm{d})$

Popov

$$\begin{bmatrix} [6] & [5] & [5] & [1] & [5] & [2] & [4] \\ [2] & [2] & [2] & [2] & [1] & [1] & [1] \\ [3] & [3] & [3] & [1] & [3] & [3] & [2] \\ [5] & [5] & [5] & [1] & [5] & [2] & [5] \end{bmatrix}$$

size $\mathcal{O}(\mathrm{mnd})$

pivot of a row: rightmost nonzero entry

$$\begin{bmatrix} [12] & & & \\ [11] & [3] & & \\ [11] & [2] & [0] & \\ [11] & [2] & & [1] \end{bmatrix}$$

size $\mathcal{O}(\mathrm{m}^2\mathrm{d})$

Hermite

$$\begin{bmatrix} [12] & & & & & & \\ [11] & [21] & [3] & & & & \\ [11] & [37] & [2] & [193] & [0] & & \\ [11] & [32] & [2] & [5] & & [243] & [1] \end{bmatrix}$$

size $\mathcal{O}(\mathrm{mn}\delta)$

Pivot support

From now, focus on Popov form

- full row rank $m \times n$ matrix $\mathbf{A}$
- input/output size $\mathcal{O}(mnd)$
- target cost $\tilde{\mathcal{O}}(m^{\omega-1}nd)$

$$\begin{bmatrix} [4] & [3] & [5] & [2] & [1] & [4] & [5] \\ [7] & [9] & [4] & [4] & [2] & [9] & [3] \\ [5] & [8] & [5] & [5] & [0] & [5] & [9] \\ [7] & [4] & [9] & [3] & [5] & [5] & [9] \end{bmatrix} = \mathbf{A}$$

unimodular

$$\begin{bmatrix} [6] & [5] & [5] & [1] & [5] & [2] & [4] \\ [2] & [2] & [2] & [2] & [1] & [1] & [1] \\ [3] & [3] & [3] & [1] & [3] & [3] & [2] \\ [5] & [5] & [5] & [1] & [5] & [2] & [5] \end{bmatrix} = \mathbf{P}$$

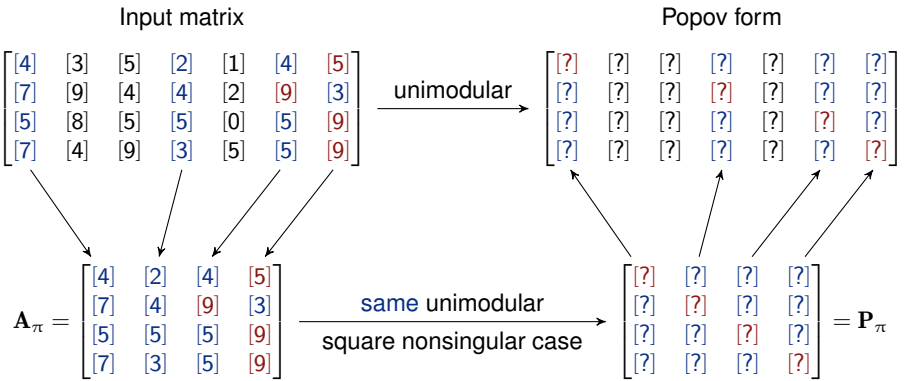
pivot support of $\mathbf{A}$: indices of the pivots in $\mathbf{P}$

$\rightsquigarrow$ any multiple $\mathbf{UA}$ has its pivots in the pivot support

Algorithmic approach

- Find the pivot support of $\mathbf{A}$
- Use the pivot support to compute $\mathbf{P}$

When the pivot support is known



Algorithm in $\tilde{\mathcal{O}}(m^{\omega-1}nd)$

Step 1: $\mathbf{A}_\pi \leftarrow$ submatrix of $\mathbf{A}$ with columns in pivot support

Step 2: $\mathbf{P}_\pi \leftarrow \mathbf{U}\mathbf{A}_\pi =$ Popov form of $\mathbf{A}_\pi$ `// nonsingular case`

Step 3: $\mathbf{P}_{\not\pi} \leftarrow \mathbf{U}\mathbf{A}_{\not\pi} = (\mathbf{P}_\pi \mathbf{A}_\pi^{-1})\mathbf{A}_{\not\pi} \bmod X^{d+1}$

Find the pivot support via saturation

$$\text{Saturation of } \mathbf{A} = \mathbb{K}[X]^{1\times n} \cap \mathbb{K}(X)^{1\times m}\mathbf{A}$$
$$= \text{left kernel of any right kernel basis of } \mathbf{A}$$

same rank as $\mathbf{A}$ + contains the row space of $\mathbf{A}$ $\Rightarrow$ same pivot support

Pivot Support in $\tilde{\mathcal{O}}(n^\omega d)$

Step 1: $\mathbf{K} \leftarrow$ right kernel basis of $\mathbf{A}$ `// use [Zhou-Labahn-Storjohann '12]`
Step 2: $\mathbf{B} \leftarrow$ pivot support revealing left kernel basis of $\mathbf{K}$
Step 3: return the indices of pivots in $\mathbf{B}$

Ensuring efficiency in Step 2

- $\mathbf{K}$ has large average row degree (not suitable for [Zhou-Labahn-Storjohann '12])
- $\mathbf{K}$ has low average column degree, and we have $\deg(\mathbf{B}) \leqslant d$ since $\mathbf{AK} = \mathbf{0}$

$\rightsquigarrow$ obtain $\mathbf{B}$ via fast Popov approximant basis [Jeannerod-Neiger-Schost-Villard '16]

$$\mathbf{BK} = \mathbf{0} \bmod \operatorname{diag}(X^{\delta_1}, \ldots, X^{\delta_{n-r}})$$

Find the pivot support of a wide matrix

Goal: when $n \gg m$, lower the cost from $\tilde{\mathcal{O}}(n^\omega d)$ to $\tilde{\mathcal{O}}(m^{\omega-1} n d)$

$$\left[\begin{array}{ccc:ccc:ccc:ccc} [4] & [3] & [5] & [2] & [1] & [4] & [5] & [4] & [2] & [9] & [0] & [3] \\ [7] & [9] & [4] & [4] & [2] & [9] & [3] & [6] & [3] & [6] & [7] & [4] \\ [5] & [8] & [5] & [5] & [0] & [5] & [9] & [6] & [4] & [6] & [2] & [1] \end{array}\right]$$

Strategy

apply the previous algorithm iteratively to

- n/m overlapping submatrices of $\mathbf{A}$
- each of dimensions $m \times 2m$

Find the pivot support of a wide matrix

Goal: when $n \gg m$, lower the cost from $\tilde{\mathcal{O}}(n^\omega d)$ to $\tilde{\mathcal{O}}(m^{\omega-1} n d)$

$$\left[\begin{array}{ccc:ccc:ccc:ccc} [4] & [3] & [5] & [2] & [1] & [4] & [5] & [4] & [2] & [9] & [0] & [3] \\ [7] & [9] & [4] & [4] & [2] & [9] & [3] & [6] & [3] & [6] & [7] & [4] \\ [5] & [8] & [5] & [5] & [0] & [5] & [9] & [6] & [4] & [6] & [2] & [1] \end{array}\right]$$

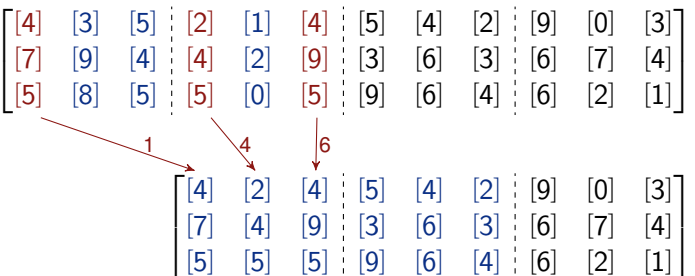
Strategy

apply the previous algorithm iteratively to

- n/m overlapping submatrices of $\mathbf{A}$
- each of dimensions $m \times 2m$

Find the pivot support of a wide matrix

Goal: when $n \gg m$, lower the cost from $\tilde{\mathcal{O}}(n^\omega d)$ to $\tilde{\mathcal{O}}(m^{\omega-1} n d)$



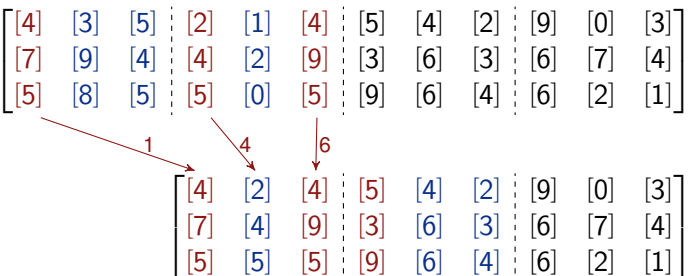
Strategy

apply the previous algorithm iteratively to

- n/m overlapping submatrices of $\mathbf{A}$
- each of dimensions $m \times 2m$

Find the pivot support of a wide matrix

Goal: when $n \gg m$, lower the cost from $\tilde{\mathcal{O}}(n^\omega d)$ to $\tilde{\mathcal{O}}(m^{\omega-1} n d)$



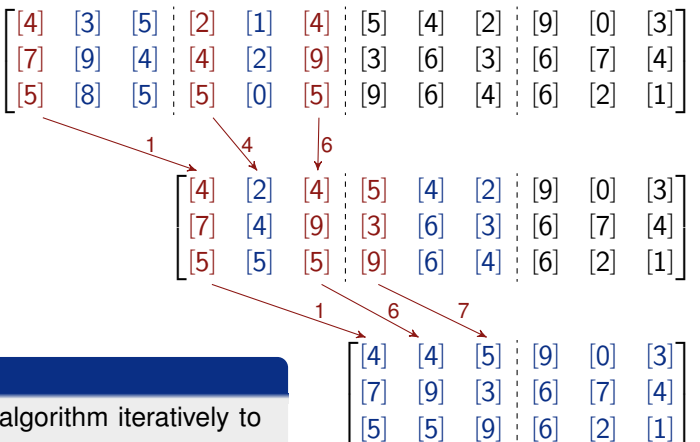
Strategy

apply the previous algorithm iteratively to

- n/m overlapping submatrices of $\mathbf{A}$
- each of dimensions $m \times 2m$

Find the pivot support of a wide matrix

Goal: when $n \gg m$, lower the cost from $\tilde{\mathcal{O}}(n^\omega d)$ to $\tilde{\mathcal{O}}(m^{\omega-1} n d)$



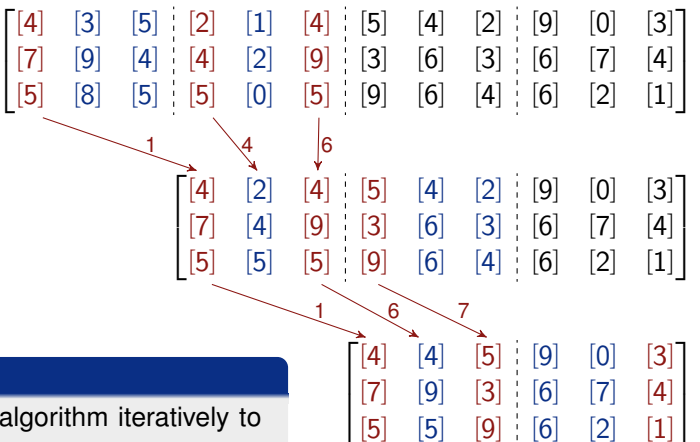
Strategy

apply the previous algorithm iteratively to

- n/m overlapping submatrices of $\mathbf{A}$
- each of dimensions $m \times 2m$

Find the pivot support of a wide matrix

Goal: when $n \gg m$, lower the cost from $\tilde{\mathcal{O}}(n^\omega d)$ to $\tilde{\mathcal{O}}(m^{\omega-1} n d)$



pivot support: $(1, 7, 12)$

Strategy

apply the previous algorithm iteratively to

- n/m overlapping submatrices of $\mathbf{A}$
- each of dimensions $m \times 2m$

Conclusion and perspectives

Conclusion:

- fast algorithm to find pivot support for Popov form
- fast Popov form when pivot support known

Also in the paper:

- algorithms based on completing the matrix into a square one
- second item works for all shifted normal forms (including Hermite)

Perspectives:

- normal forms for arbitrary shifts
- kernel bases for arbitrary shifts
- cost bound sensitive to average row/column degrees