Fast, Deterministic computation of the Hermite Form of Matrix Polynomials

George Labahn

Symbolic Computation Group
University of Waterloo

Outline

1. Preliminaries

2. Polynomial Matrices and Tools

3. Triangularization

4. Algorithm for Hermite Normal Form

Hermite Normal Form

Problem : Given nonsingular $\mathbf{A} \in \mathbb{K}[x]^{n \times n}$. Compute $\mathbf{U}$ and $\mathbf{H}$:

(i) $\mathbf{U}$ unimodular,

(ii) $\mathbf{H}$ in (column) Hermite form

(iii) $\mathbf{A} \cdot \mathbf{U} = \mathbf{H}$

Hermite Normal Form

Problem : Given nonsingular $\mathbf{A} \in \mathbb{K}[x]^{n \times n}$. Compute $\mathbf{U}$ and $\mathbf{H}$:

(i) $\mathbf{U}$ unimodular,

(ii) $\mathbf{H}$ in (column) Hermite form

(iii) $\mathbf{A} \cdot \mathbf{U} = \mathbf{H}$

Hermite Normal Form :

$$\mathbf{H} = \begin{bmatrix} h_{11} & 0 & \cdots & \cdots & 0 \\ h_{21} & h_{22} & 0 & & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & 0 \\ h_{n1} & \cdots & & \cdots & h_{nn} \end{bmatrix} \qquad \deg h_{ij} < \deg h_{ii}.$$

Naive Method : $\mathbf{A} \cdot \mathbf{U} = \mathbf{H}$

$$\begin{bmatrix} 25 & -16 & -38 & 57 & -32 \\ 94 & -9 & -18 & 27 & -74 \\ 12 & -50 & 87 & -93 & -4 \\ -2 & -22 & 33 & -76 & 27 \\ 50 & 45 & -98 & -72 & 8 \end{bmatrix}$$

Naive Method : $\mathbf{A} \cdot \mathbf{U} = \mathbf{H}$

Euclidean Algorithm: $1 = 25(-7) - 16(-11)$ and $0 = 25(16) - 16(25)$

$$\begin{bmatrix} 25 & -16 & -38 & 57 & -32 \\ 94 & -9 & -18 & 27 & -74 \\ 12 & -50 & 87 & -93 & -4 \\ -2 & -22 & 33 & -76 & 27 \\ 50 & 45 & -98 & -72 & 8 \end{bmatrix}$$

Naive Method : $\mathbf{A} \cdot \mathbf{U} = \mathbf{H}$

Euclidean Algorithm: $1 = 25(-7) - 16(-11)$ and $0 = 25(16) - 16(25)$

$$\begin{bmatrix} 25 & -16 & -38 & 57 & -32 \\ 94 & -9 & -18 & 27 & -74 \\ 12 & -50 & 87 & -93 & -4 \\ -2 & -22 & 33 & -76 & 27 \\ 50 & 45 & -98 & -72 & 8 \end{bmatrix} \cdot \begin{bmatrix} -7 & 16 & 0 & 0 & 0 \\ -11 & 25 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -38 & 57 & -32 \\ -559 & 1279 & -18 & 27 & -74 \\ 466 & -1058 & 87 & -93 & -4 \\ 256 & -582 & 33 & -76 & 27 \\ -845 & 1925 & -98 & -72 & 8 \end{bmatrix}$$

Naive Method : $\mathbf{A} \cdot \mathbf{U} = \mathbf{H}$

$$\begin{bmatrix} 25 & -16 & -38 & 57 & -32 \\ 94 & -9 & -18 & 27 & -74 \\ 12 & -50 & 87 & -93 & -4 \\ -2 & -22 & 33 & -76 & 27 \\ 50 & 45 & -98 & -72 & 8 \end{bmatrix} \cdot \begin{bmatrix} -7 & 16 & -266 & 399 & -224 \\ -11 & 25 & -418 & 627 & -352 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -559 & 1279 & -21260 & 31890 & -17962 \\ 466 & -1058 & 17795 & -26655 & 14908 \\ 256 & -582 & 9761 & -14668 & 8219 \\ -845 & 1925 & -32208 & 48093 & -27032 \end{bmatrix}$$

Naive Method : $\mathbf{A} \cdot \mathbf{U} = \mathbf{H}$

$$\begin{bmatrix} 25 & -16 & -38 & 57 & -32 \\ 94 & -9 & -18 & 27 & -74 \\ 12 & -50 & 87 & -93 & -4 \\ -2 & -22 & 33 & -76 & 27 \\ 50 & 45 & -98 & -72 & 8 \end{bmatrix} \cdot \begin{bmatrix} -7304387 & -3663017 & -2850298 & -7518539 & -8030879 \\ 663374 & 332670 & 258860 & 682823 & 729353 \\ -5523756 & -2770063 & -2155465 & -5685703 & -6073147 \\ -6027196 & -3022529 & -2351916 & -6203903 & -6626659 \\ -10214722 & -5122497 & -3985961 & -10514200 & -11230675 \end{bmatrix}$$

$\cdots$ continuing along $\cdots$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 2 & 3 & 0 \\ 558200904 & 279927586 & 217819634 & 574566395 & 613719389 \end{bmatrix}$$

Naive Method : $\mathbf{A} \cdot \mathbf{U} = \mathbf{H}$

$$\begin{bmatrix} 25 & -16 & -38 & 57 & -32 \\ 94 & -9 & -18 & 27 & -74 \\ 12 & -50 & 87 & -93 & -4 \\ -2 & -22 & 33 & -76 & 27 \\ 50 & 45 & -98 & -72 & 8 \end{bmatrix} \cdot \begin{bmatrix} -7304387 & -3663017 & -2850298 & -7518539 & -8030879 \\ 663374 & 332670 & 258860 & 682823 & 729353 \\ -5523756 & -2770063 & -2155465 & -5685703 & -6073147 \\ -6027196 & -3022529 & -2351916 & -6203903 & -6626659 \\ -10214722 & -5122497 & -3985961 & -10514200 & -11230675 \end{bmatrix}$$

$\cdots$ continuing along $\cdots$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 2 & 3 & 0 \\ 558200904 & 279927586 & 217819634 & 574566395 & 613719389 \end{bmatrix}$$

Notice how entries grow quickly.

This talk

Results:

- Fast, deterministic algorithm to compute $\mathbf{H}$
- Fast, deterministic algorithm for determinant of $(\mathbf{A})$
- Complexity : $\widetilde{\mathcal{O}}\left(n^{\omega}\lceil s\rceil\right)$ where s bounded by average
  : of row and column degrees of $\mathbf{A}$
  : here $\widetilde{\mathcal{O}}$ is big O without log factors.
  : here ω exponent of matrix multiplication.

This talk

Results:

- Fast, deterministic algorithm to compute $\mathbf{H}$
- Fast, deterministic algorithm for determinant of $(\mathbf{A})$
- Complexity : $\widetilde{\mathcal{O}}\left(\mathfrak{n}^{\omega}\lceil s\rceil\right)$ where s bounded by average
  : of row and column degrees of $\mathbf{A}$
  : here $\widetilde{\mathcal{O}}$ is big O without log factors.
  : here ω exponent of matrix multiplication.

Details :

- G. Labahn, V. Neiger and W. Zhou,
  Fast, deterministic computation of determinants and Hermite normal forms of polynomial matrices,
  Journal of Complexity 2018.

Previous work : Hermite Form

- Polynomial-time over $\mathbb{Q}[x]$: Kannan 1985.
- $\widetilde{\mathcal{O}}(n^4 d)$: Hafner-McCurley 1991 deterministic
- $\widetilde{\mathcal{O}}(n^{\omega+1} d)$: Hafner-McCurley (1991), Villard (1996) Storjohann and Labahn (1996) deterministic
- $\widetilde{\mathcal{O}}(n^3 d^2)$: Mulders and Storjohann (2003) deterministic
- $\widetilde{\mathcal{O}}(n^3 d)$: Gupta and Storjohann (2012) probabilistic
- $\widetilde{\mathcal{O}}(n^\omega d)$: Gupta and Storjohann (2012) probabilistic
- $\widetilde{\mathcal{O}}(n^\omega s)$: Labahn-Neiger-Zhou (This talk) deterministic

Where do polynomial matrices arise?

- Solving linear equations with polynomial coefficients
  - Often encountered in computer algebra computations

Where do polynomial matrices arise?

- Solving linear equations with polynomial coefficients
  - Often encountered in computer algebra computations
- Combinatorics
  - when is a generating function $y(z)$ algebraic? holonomic?

Where do polynomial matrices arise?

- Solving linear equations with polynomial coefficients
  - Often encountered in computer algebra computations
- Combinatorics
  - when is a generating function $y(z)$ algebraic? holonomic?
- Formalism used for linear control theory/linear systems theory

Where do polynomial matrices arise?

- Solving linear equations with polynomial coefficients
  - Often encountered in computer algebra computations
- Combinatorics
  - when is a generating function $y(z)$ algebraic? holonomic?
- Formalism used for linear control theory/linear systems theory
- Coding theory

Where do polynomial matrices arise?

- Solving linear equations with polynomial coefficients
  - Often encountered in computer algebra computations
- Combinatorics
  - when is a generating function $y(z)$ algebraic? holonomic?
- Formalism used for linear control theory/linear systems theory
- Coding theory
- Block-Wiedemann's algorithm;

Where do polynomial matrices arise?

- Solving linear equations with polynomial coefficients
  - Often encountered in computer algebra computations
- Combinatorics
  - when is a generating function $y(z)$ algebraic? holonomic?
- Formalism used for linear control theory/linear systems theory
- Coding theory
- Block-Wiedemann's algorithm; Order bases

Types of constructions for polynomial matrices

Given $\mathbf{F} \in \mathbb{K}[x]^{m \times n}$ constructions include:

- Kernel basis of $\mathbf{F}$

$$\{ \mathbf{p} \in \mathbb{K}[z]^n \mid \mathbf{F} \cdot \mathbf{p} = 0 \}$$

Types of constructions for polynomial matrices

Given $\mathbf{F} \in \mathbb{K}[x]^{m \times n}$ constructions include:

- Kernel basis of $\mathbf{F}$

$$\{ \mathbf{p} \in \mathbb{K}[z]^n \mid \mathbf{F} \cdot \mathbf{p} = 0 \}$$

- Column basis of $\mathbf{F}$

$$\{ \mathbf{q} \in \mathbb{K}[z]^m \mid \exists\, \mathbf{p} \in \mathbb{K}[z]^n \quad \text{with} \quad \mathbf{q} = \mathbf{F} \cdot \mathbf{p} \}$$

Types of constructions for polynomial matrices

Given $\mathbf{F} \in \mathbb{K}[x]^{m \times n}$ constructions include:

- Kernel basis of $\mathbf{F}$

$$\{ \mathbf{p} \in \mathbb{K}[z]^n \mid \mathbf{F} \cdot \mathbf{p} = 0 \}$$

- Column basis of $\mathbf{F}$

$$\{ \mathbf{q} \in \mathbb{K}[z]^m \mid \exists\, \mathbf{p} \in \mathbb{K}[z]^n \quad \text{with} \quad \mathbf{q} = \mathbf{F} \cdot \mathbf{p} \}$$

Note: Basis $\equiv$ Module Basis. Write as single polynomial matrix.

How to measure size of polynomial matrices?

Given $\mathbf{F} \in \mathbb{K}[x]^{m \times n}$: degree of $\mathbf{F}$

$$\deg(\mathbf{F}) = \max \quad \deg \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{mn} \end{bmatrix}$$

How to measure size of polynomial matrices?

Given $\mathbf{F} \in \mathbb{K}[x]^{m \times n}$: column degree of $\mathbf{F}$

$$\gamma_1, \quad \gamma_2 \quad \cdots \quad \gamma_n$$

$$\text{cdeg}(\mathbf{F}) = \max \ \deg \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{mn} \end{bmatrix}$$

How to measure size of polynomial matrices?

Given $\mathbf{F} \in \mathbb{K}[x]^{m \times n}$, $\vec{s} \in \mathbb{Z}^m$: shifted column degree of $\mathbf{F}$

$$\text{cdeg}_{\vec{s}}(\mathbf{F}) = \max \deg \begin{array}{c} \begin{matrix} \gamma_1 & \gamma_2 \cdots & \cdots & \gamma_n \end{matrix} \\ \begin{bmatrix} x^{s_1} f_{11} & x^{s_1} f_{12} & \cdots & x^{s_1} f_{1n} \\ x^{s_2} f_{21} & x^{s_2} f_{22} & \cdots & x^{s_2} f_{2n} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ x^{s_m} f_{m1} & x^{s_m} f_{m2} & \cdots & x^{s_m} f_{mn} \end{bmatrix} \end{array}$$

How to measure size of polynomial matrices?

Given $\mathbf{F} \in \mathbb{K}[x]^{m \times n}$, $\vec{s} \in \mathbb{Z}^m$: shifted column degree of $\mathbf{F}$

$$\text{cdeg}_{\vec{s}}(\mathbf{F}) = \max \deg \begin{array}{c} \begin{matrix} \gamma_1 & \gamma_2 \cdots & \cdots & \gamma_n \end{matrix} \\ \begin{bmatrix} x^{s_1} f_{11} & x^{s_1} f_{12} & \cdots & x^{s_1} f_{1n} \\ x^{s_2} f_{21} & x^{s_2} f_{22} & \cdots & x^{s_2} f_{2n} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ x^{s_m} f_{m1} & x^{s_m} f_{m2} & \cdots & x^{s_m} f_{mn} \end{bmatrix} \end{array}$$

Also corresponding notions of : leading coefficient, reduced, minimal

Cost of Tools

$\mathbf{F} \in \mathbb{K}[z]^{n \times n}$, $\vec{s} \in \mathbb{Z}^n$ bounds column degrees, $\sum \vec{s} \leq \xi$

Theorem
$\vec{s}$-Minimal kernel basis computation costs $\widetilde{\mathcal{O}}(n^\omega s)$.

(Zhou-Labahn-Storjohann, ISSAC 2012)

Theorem
Column basis computation costs $\widetilde{\mathcal{O}}(n^\omega s)$.

(Zhou-Labahn, ISSAC 2013)

Our Approach

- Triangularize $\mathbf{A}$
  - Gives diagonal entries of $\mathbf{H}$ which can be large
- Reduce remaining off-diagonal entries

Our Approach

- Triangularize $\mathbf{A}$
  - Gives diagonal entries of $\mathbf{H}$ which can be large
- Reduce remaining off-diagonal entries
- Need to avoid computing unimodular multiplier $\mathbf{U}$
  - Issue is that $\mathbf{U}$ can be too large

Finding Diagonal Elements

Partition $\mathbf{A}$ and $\mathbf{U}$ and reduce via

$$\mathbf{A} \cdot \mathbf{U} = \begin{bmatrix} \mathbf{A}_u \\ \mathbf{A}_d \end{bmatrix} \begin{bmatrix} \mathbf{U}_\ell & \mathbf{U}_r \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 & 0 \\ * & \mathbf{B}_2 \end{bmatrix}.$$

Finding Diagonal Elements

Partition $\mathbf{A}$ and $\mathbf{U}$ and reduce via

$$\mathbf{A} \cdot \mathbf{U} \;=\; \begin{bmatrix} \mathbf{A}_{\mathrm{u}} \\ \mathbf{A}_{\mathrm{d}} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{\ell} & \mathbf{U}_{\mathrm{r}} \\ & \end{bmatrix} \;=\; \begin{bmatrix} \mathbf{B}_1 & 0 \\ * & \mathbf{B}_2 \end{bmatrix}.$$

Here

(i) $\mathbf{U}_{\mathrm{r}}$ a right kernel basis of $\mathbf{A}_{\mathrm{u}}$.

Finding Diagonal Elements

Partition $\mathbf{A}$ and $\mathbf{U}$ and reduce via

$$\mathbf{A} \cdot \mathbf{U} \quad = \quad \begin{bmatrix} \mathbf{A}_{\mathrm{u}} \\ \mathbf{A}_{\mathrm{d}} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{\ell} & \mathbf{U}_{\mathrm{r}} \\ & \end{bmatrix} \quad = \quad \begin{bmatrix} \mathbf{B}_1 & 0 \\ * & \mathbf{B}_2 \end{bmatrix}.$$

Here

(i) $\mathbf{U}_{\mathrm{r}}$ a right kernel basis of $\mathbf{A}_{\mathrm{u}}$.

(ii) $\mathbf{B}_1$ is nonsingular and a column basis of $\mathbf{A}_{\mathrm{u}}$.

Finding Diagonal Elements

Partition $\mathbf{A}$ and $\mathbf{U}$ and reduce via

$$\mathbf{A} \cdot \mathbf{U} \;=\; \begin{bmatrix} \mathbf{A}_u \\ \mathbf{A}_d \end{bmatrix} \begin{bmatrix} \mathbf{U}_\ell & \mathbf{U}_r \\ & \end{bmatrix} \;=\; \begin{bmatrix} \mathbf{B}_1 & 0 \\ * & \mathbf{B}_2 \end{bmatrix}.$$

Here

(i) $\mathbf{U}_r$ a right kernel basis of $\mathbf{A}_u$.

(ii) $\mathbf{B}_1$ is nonsingular and a column basis of $\mathbf{A}_u$.

(iii) $\mathbf{B}_2 = \mathbf{A}_d \cdot \mathbf{U}_r$.

Finding Diagonal Elements

Partition $\mathbf{A}$ and $\mathbf{U}$ and reduce via

$$\mathbf{A} \cdot \mathbf{U} \quad = \quad \begin{bmatrix} \mathbf{A}_u \\ \mathbf{A}_d \end{bmatrix} \begin{bmatrix} \mathbf{U}_\ell & \mathbf{U}_r \\ & \end{bmatrix} \quad = \quad \begin{bmatrix} \mathbf{B}_1 & 0 \\ * & \mathbf{B}_2 \end{bmatrix}.$$

Here

(i) $\mathbf{U}_r$ a right kernel basis of $\mathbf{A}_u$.

(ii) $\mathbf{B}_1$ is nonsingular and a column basis of $\mathbf{A}_u$.

(iii) $\mathbf{B}_2 = \mathbf{A}_d \cdot \mathbf{U}_r$.

Recurse on $\mathbf{B}_1$ and $\mathbf{B}_2$ to get diagonal elements

Finding Diagonal Elements

Partition $\mathbf{A}$ and $\mathbf{U}$ and reduce via

$$\mathbf{A} \cdot \mathbf{U} \quad = \quad \begin{bmatrix} \mathbf{A}_u \\ \mathbf{A}_d \end{bmatrix} \begin{bmatrix} \mathbf{U}_\ell & \mathbf{U}_r \\ & \end{bmatrix} \quad = \quad \begin{bmatrix} \mathbf{B}_1 & 0 \\ * & \mathbf{B}_2 \end{bmatrix}.$$

Here

(i) $\mathbf{U}_r$ a right kernel basis of $\mathbf{A}_u$.

(ii) $\mathbf{B}_1$ is nonsingular and a column basis of $\mathbf{A}_u$.

(iii) $\mathbf{B}_2 = \mathbf{A}_d \cdot \mathbf{U}_r$.

Recurse on $\mathbf{B}_1$ and $\mathbf{B}_2$ to get diagonal elements

Important to control size (measured by column degrees).

Costs?

- Compute (shifted) Kernel Basis : $\mathbf{U}_r$
- Compute Column Basis : $\mathbf{B}_1$
- Multiply two polynomial matrices : $\mathbf{B}_2 = \mathbf{A}_d \cdot \mathbf{U}_r$

Important Properties (ZLS ISSAC 2012)

$\mathbf{F} \in \mathbb{K}[z]^{m \times n}$, $\quad \vec{s} \in \mathbb{Z}^n$ bounds column degrees, $\quad \sum \vec{s} \leq \xi$

Theorem

For $\mathbf{M}$ *a* $\vec{s}$*-minimal kernel basis of* $\mathbf{F}$*:* $\quad \sum \operatorname{cdeg}_{\vec{s}} \mathbf{M} \leq \sum \vec{s}$

Theorem

(i) $\mathbf{A} \in \mathbb{K}[z]^{m \times n}$, $m \leq n$, $\vec{s} \in \mathbb{Z}^n$ *bounding col. degrees of* $\mathbf{A}$

(ii) $\mathbf{B} \in \mathbb{K}[z]^{n \times k}$ *with* $k \in O(m)$, $\quad \sum \operatorname{cdeg}_{\vec{s}} \mathbf{B} \leq \sum \vec{s} \in O(\xi)$

Multiply $\mathbf{A}$ *and* $\mathbf{B}$ *:* $\quad \widetilde{\mathcal{O}}(n^2 m^{\omega-2} s) \subset \widetilde{\mathcal{O}}(n^\omega s), \quad s = \xi/n.$

Complexity for finding diagonals

Theorem

$\mathbf{A} \in \mathbb{K}[z]^{n \times n}$. *Diagonals costs* $\widetilde{\mathcal{O}}\left(n^{\omega} \lceil s \rceil\right)$ *where* $s = \frac{\sum \operatorname{cdeg} \mathbf{A}}{n}$.

Complexity for finding diagonals

Theorem

$\mathbf{A} \in \mathbb{K}[z]^{n \times n}$. *Diagonals costs* $\widetilde{\mathcal{O}}\left(n^{\omega}\lceil s \rceil\right)$ *where* $s = \frac{\sum \operatorname{cdeg} \mathbf{A}}{n}$.

Proof.

If cost : $g(n)$ then recurrence relation: (with $s = \frac{\xi}{n}$)

$$g(n) \quad \in \quad \widetilde{\mathcal{O}}(n^{\omega}\lceil s \rceil) + g(\lceil n/2 \rceil) + g(\lfloor n/2 \rfloor)$$

Complexity for finding diagonals

Theorem

$\mathbf{A} \in \mathbb{K}[z]^{n \times n}$. *Diagonals costs* $\widetilde{\mathcal{O}}(n^\omega \lceil s \rceil)$ *where* $s = \frac{\sum \operatorname{cdeg} \mathbf{A}}{n}$.

Proof.

If cost : $g(n)$ then recurrence relation: (with $s = \frac{\xi}{n}$)

$$\begin{aligned} g(n) &\in \widetilde{\mathcal{O}}(n^\omega \lceil s \rceil) + g(\lceil n/2 \rceil) + g(\lfloor n/2 \rfloor) \\ &\in \widetilde{\mathcal{O}}(n^{\omega-1}\xi + n^\omega) + g(\lceil n/2 \rceil) + g(\lfloor n/2 \rfloor) \end{aligned}$$

Complexity for finding diagonals

Theorem

$\mathbf{A} \in \mathbb{K}[z]^{n \times n}$. *Diagonals costs* $\widetilde{\mathcal{O}}(n^\omega \lceil s \rceil)$ *where* $s = \frac{\sum \operatorname{cdeg} \mathbf{A}}{n}$.

Proof.

If cost : $g(n)$ then recurrence relation: (with $s = \frac{\xi}{n}$)

$$\begin{aligned} g(n) &\in \widetilde{\mathcal{O}}(n^\omega \lceil s \rceil) + g(\lceil n/2 \rceil) + g(\lfloor n/2 \rfloor) \\ &\in \widetilde{\mathcal{O}}(n^{\omega-1}\xi + n^\omega) + g(\lceil n/2 \rceil) + g(\lfloor n/2 \rfloor) \\ &\in \widetilde{\mathcal{O}}(n^{\omega-1}\xi + n^\omega) + 2g(\lceil n/2 \rceil) \\ &\in \widetilde{\mathcal{O}}(n^{\omega-1}\xi + n^\omega) = \widetilde{\mathcal{O}}(n^\omega \lceil s \rceil). \end{aligned}$$

□

Hermite Normal Form

- Reducing off-diagonal elements
- Complexity
- Better complexity

Finding Rest of $\mathbf{H}$

Know : $\vec{\delta}$ diagonal degrees of $\mathbf{H}$. Set $\mu = \max(\vec{\delta})$

$$\mathbf{x}^{\vec{\mu}-\vec{\delta}}\,\mathbf{A} \xrightarrow{\text{reduce}} \mathbf{x}^{\vec{\mu}-\vec{\delta}}\,\mathbf{R} \xrightarrow{\text{normalize}} \mathbf{H} = \mathbf{R} \cdot \mathrm{lc}_{-\vec{\delta}}(\mathbf{R})^{-1}$$

where $\mathbf{R}$ is any $-\vec{\delta}$-column reduced form of $\mathbf{A}$.

Finding Rest of $\mathbf{H}$

Know : $\vec{\delta}$ diagonal degrees of $\mathbf{H}$. Set $\mu = \max(\vec{\delta})$

$$\mathbf{x}^{\vec{\mu}-\vec{\delta}}\,\mathbf{A} \xrightarrow{\text{reduce}} \mathbf{x}^{\vec{\mu}-\vec{\delta}}\,\mathbf{R} \xrightarrow{\text{normalize}} \mathbf{H} = \mathbf{R} \cdot \mathrm{lc}_{-\vec{\delta}}(\mathbf{R})^{-1}$$

where $\mathbf{R}$ is any $-\vec{\delta}$-column reduced form of $\mathbf{A}$.

Problem : Shift $\vec{\mu} - \vec{\delta}$ might be too large

Finding Rest of $\mathbf{H}$

Know : $\vec{\delta}$ diagonal degrees of $\mathbf{H}$. Set $\mu = \max(\vec{\delta})$

$$\mathbf{x}^{\vec{\mu}-\vec{\delta}}\,\mathbf{A} \xrightarrow{\text{reduce}} \mathbf{x}^{\vec{\mu}-\vec{\delta}}\,\mathbf{R} \xrightarrow{\text{normalize}} \mathbf{H} = \mathbf{R} \cdot \mathrm{lc}_{-\vec{\delta}}(\mathbf{R})^{-1}$$

where $\mathbf{R}$ is any $-\vec{\delta}$-column reduced form of $\mathbf{A}$.

Problem : Shift $\vec{\mu} - \vec{\delta}$ might be too large

Answer : Partial linearization of Storjohann (2007): $\mathbf{A} \rightarrow \mathcal{L}(\mathbf{A})$

Smooths shifts, keeps properties of $\mathbf{A}$ while enlarging a bit.

Partial Linearization

Consider $\mathbf{H}$ with diagonal degrees $(2, 37, 7, 18)$.

$$\mathbf{H} = \begin{bmatrix} (2) & & & \\ [36] & (37) & & \\ [6] & [6] & (7) & \\ [17] & [17] & [17] & (18) \end{bmatrix},$$

$[\mathrm{d}]$: degree at most d and (d) : monic , degree exactly d.

$\delta = 1 + \lfloor (2 + 37 + 7 + 18)/4 \rfloor = 17.$ Construct by "expanding rows":

$$\widetilde{\mathbf{H}} = \begin{bmatrix} (2) & & & \\ [16] & [16] & & \\ [16] & [16] & & \\ [2] & (3) & & \\ [6] & [6] & (7) & \\ [16] & [16] & [16] & [16] \\ [0] & [0] & [0] & (1) \end{bmatrix}.$$

$\mathbf{H}$ and $\widetilde{\mathbf{H}}$ are related by $\mathbf{H} = \boldsymbol{\mathcal{E}}_{\vec{\delta}} \cdot \widetilde{\mathbf{H}}$ where

$$\boldsymbol{\mathcal{E}}_{\vec{\delta}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & x^{17} & x^{34} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & x^{17} \end{bmatrix}.$$

Insert elementary columns in $\widetilde{\mathbf{H}}$ by

$$\mathcal{L}_{\vec{\delta}}(\mathbf{H}) = \begin{bmatrix} (2) & & & & & & \\ [16] & x^{17} & & [16] & & & \\ [16] & -1 & x^{17} & [16] & & & \\ [2] & & -1 & (3) & & & \\ [6] & & & [6] & (7) & & \\ [16] & & & [16] & [16] & x^{17} & [16] \\ [0] & & & [0] & [0] & -1 & (1) \end{bmatrix}$$

Row degrees $\vec{d} = (2, 17, 17, 3, 7, 17, 1)$ - maximum 17.

Main property kept : shifted column reduction.

$$\begin{array}{ccccc}
\mathbf{x}^{\vec{\mu}-\vec{\delta}}\,\mathbf{A} & \xrightarrow{\text{reduce}} & \mathbf{x}^{\vec{\mu}-\vec{\delta}}\,\mathbf{R} & \xrightarrow{\text{normalize}} & \mathbf{H} = \mathbf{R}\,\mathrm{lc}_{-\vec{\delta}}(\mathbf{R})^{-1} \\
\Big\downarrow \text{partial linearization} & & & & \Big\downarrow \text{partial linearization} \\
\mathbf{x}^{\vec{m}-\vec{d}}\,\mathcal{L}_{\vec{\delta}}(\mathbf{A}) & \xrightarrow{\text{reduce}} & \mathbf{x}^{\vec{m}-\vec{d}}\,\hat{\mathbf{R}} & \xrightarrow{\text{normalize}} & \mathcal{L}_{\vec{\delta}}(\mathbf{H}) = \hat{\mathbf{R}}\,\mathrm{lc}_{-\vec{d}}(\hat{\mathbf{R}})^{-1}
\end{array}$$

.

Theorem

Let $\mathbf{A} \in \mathbb{K}[x]^{n\times n}$ nonsingular with $\vec{\delta}$ the degrees of the diagonal entries of the Hermite form.

Then the Hermite form is computed using $\widetilde{\mathcal{O}}(n^{\omega} d)$ field operations.

Improving the Complexity

Repeat : partial linearization (this time with columns) :

(i) Enlarge : $\mathbf{A} \rightarrow \mathcal{L}^c(\mathbf{A})$

- size of $\mathcal{L}^c(\mathbf{A})$ at most twice size of $\mathbf{A}$
- degree $\mathcal{L}^c(\mathbf{A})$ at most average of $\mathbf{A}$

(ii) Compute Hermite form of $\mathcal{L}^c(\mathbf{A})$

(iii) $\mathbf{H}$ is found in lower right corner of Hermite form of $\mathcal{L}^c(\mathbf{A})$

Theorem
$\mathbf{A} \in \mathbb{K}[x]^{n \times n}$ *nonsingular. Hermite form computed:* $\widetilde{\mathcal{O}}(n^\omega \lceil s \rceil)$.

What about Determinant?

Diagonals not enough - need to worry about unimodular part.

What about Determinant?

Diagonals not enough - need to worry about unimodular part.

$$\det \mathbf{A} = \frac{\det \mathbf{B}_1 \cdot \det \mathbf{B}_2}{\det \mathbf{U}}$$

What about Determinant?

Diagonals not enough - need to worry about unimodular part.

$$\det \mathbf{A} = \frac{\det \mathbf{B}_1 \cdot \det \mathbf{B}_2}{\det \mathbf{U}}$$

For $\det \mathbf{U} = \det [\, \mathbf{U}_\ell \; \mathbf{U}_\mathrm{r}]$ we do:

1. $\det \mathbf{U} = \det \mathbf{U} \bmod z = \det \mathcal{U} = \det [\mathcal{U}_\ell, \mathcal{U}_\mathrm{r}]$
2. $\mathbf{V} = \mathbf{U}^{-1} = \begin{bmatrix} \mathbf{V}_\mathrm{u} \\ \mathbf{V}_\mathrm{d} \end{bmatrix}$
3. $\mathbf{U}_\mathrm{r}$ and $\mathbf{V}_\mathrm{u}$ determined in column bases computation
4. Find $\mathcal{U}_\ell{}^*$ such that $\mathcal{U}^* = [\mathcal{U}_\ell{}^*, \mathcal{U}_\mathrm{r}]$ is unimodular
5. Let $\mathcal{V}_\mathrm{u} = \mathbf{V}_\mathrm{u} \bmod z$. Then $\det \mathbf{U} = \frac{\det \mathcal{U}^*}{\det \mathcal{V}_\mathrm{u} \mathcal{U}_\ell{}^*}$

Previous work : Determinants

- Storjohann (2000) $\widetilde{\mathcal{O}}(n^{\omega+1}d)$, deterministic
- Mulders and Storjohann (2003) $O(n^3d^2)$, deterministic
- Eberly-Giesbrecht-Villard (2000) $\widetilde{\mathcal{O}}(n^{2+\omega/2}d)$, probabilistic
- Storjohan (2003) $\widetilde{\mathcal{O}}(n^{\omega}d)$; probabilistic
- Giorgi-Jeannerod-Villard (2003) $\widetilde{\mathcal{O}}(n^{\omega}d)$,
- Kaltofen (1992)
- Kaltofen and Villard (2004)

Future Work

We want to make progress with:

- Fast but with coefficient control (e.g. matrices over $Z[x]$)
  - Beckermann, Labahn, Villard (2006)

    $\widetilde{\mathcal{O}}((m+n)(m^2 d \min(m,n))^3 \log \|\mathbf{A}\|)$ bit operations

    $\widetilde{\mathcal{O}}(n^{10} d^3 \log \|\mathbf{A}\|)$ when $m = n$

- Fast Popov form. Fast shifted Popov form
- Fast Hermite form for integer matrices
- Fast Hermite and Popov for alternate domains
  (e.g. matrices over $\mathbb{K}(x)[D_x]$)

References

Other relevant papers:

- W. Zhou, G. Labahn and A. Storjohann, Computing Minimal Nullspace Bases, *ISSAC 2012*,
- W. Zhou and G. Labahn, Computing Column Bases for polynomial matrices, *ISSAC 2013*
- S. Gupta, S. Sarkar, A. Storjohann, J. Valeriote, Triangular x-basis decompositions ..., *ISSAC 2012*
- S. Gupta and A. Storjohann, Computing Hermite Forms of Polynomial Matrices, *ISSAC 2012*
- V. Neiger, Fast computation of shifted Popov forms, *ISSAC 2016*