Sigma Bases, Kernel Bases and applications of fast computation for polynomial matrices

George Labahn

Symbolic Computation Group
Cheriton School of Computer Science
University of Waterloo, Canada

October 2015

Outline

1. Sigma Bases
2. Minimal Kernel Bases
3. Applications for fast Polynomial Matrix arithmetic
   - Application 1 : Polynomial Matrix Inversion
   - Application 2 : Determinant and Matrix Triangulation
   - Application 3 : Hermite Normal Form

Outline

1. Sigma Bases
2. Minimal Kernel Bases
3. Applications for fast Polynomial Matrix arithmetic
   - Application 1 : Polynomial Matrix Inversion
   - Application 2 : Determinant and Matrix Triangulation
   - Application 3 : Hermite Normal Form

Padé Approximants

$$A(z) \cdot V(z) - U(z) = z^{m+n+1} W(z)$$

Solved via associated linear system

$$\begin{bmatrix} a_{m-n+1} & \cdots & \cdots & a_{n-1} & a_n \\ a_{m-n+2} & \cdots & \cdots & a_n & a_{n+1} \\ \vdots & & & & \vdots \\ \vdots & & & & \vdots \\ a_{n-1} & \cdots & \cdots & & a_{m+n-2} \\ a_n & \cdots & \cdots & a_{m+n-2} & a_{m+n-1} \end{bmatrix} \cdot \begin{bmatrix} v_n \\ v_{n-1} \\ \vdots \\ \vdots \\ v_2 \\ v_1 \end{bmatrix} = -v_0 \begin{bmatrix} a_{m+1} \\ a_{m+2} \\ \vdots \\ \vdots \\ a_{m+n-1} \\ a_{m+n} \end{bmatrix}$$

- Nice when coefficient matrix is nonsingular.
- Matrix Padé $\equiv a_i$ square matrices.

Additional Information

Special structure of Hankel matrices implies:

- All Padé approximants known in scalar case
  - Padé table in scalar case has a type of block structure
- Algorithms take advantage of structure of Hankel matrices
  - in $n \times n$ case cost $O(n^2)$ or $O(n \log^2 n)$

However:

- Nothing known about structure of matrix Padé case
- Same issue with Hermite-Padé :
  Given $\mathbf{A}(z) = [A_1(z), \ldots, A_m(z)]$ and deg bounds $\{n_i\}$:

  Find $\mathbf{P}(z) = [P_1(z), ..., P_m(z)]^T$ with deg $P_i(z) \leq n_i$ and

$$\mathbf{A}(z) \cdot \mathbf{P}(z) = O(z^{n_1+\cdots+n_m+m})$$

Additional Information

Special structure of Hankel matrices implies:

- All Padé approximants known in scalar case
  - Padé table in scalar case has a type of block structure
- Algorithms take advantage of structure of Hankel matrices
  - in $n \times n$ case cost $O(n^2)$ or $O(n \log^2 n)$

However:

- Nothing known about structure of matrix Padé case
- Same issue with Hermite-Padé :

  Given $\mathbf{A}(z) = [A_1(z), \ldots, A_m(z)]$ and deg bounds $\{n_i\}$:

  Find $\mathbf{P}(z) = [P_1(z), ..., P_m(z)]^T$ with deg $P_i(z) \leq n_i$ and

  $$\mathbf{A}(z) \cdot \mathbf{P}(z) = O(z^{n_1+\cdots+n_m+m})$$

Additional Information

Special structure of Hankel matrices implies:

- All Padé approximants known in scalar case
  - Padé table in scalar case has a type of block structure
- Algorithms take advantage of structure of Hankel matrices
  - in $n \times n$ case cost $O(n^2)$ or $O(n \log^2 n)$

However:

- Nothing known about structure of matrix Padé case
- Same issue with Hermite-Padé :

  Given $\mathbf{A}(z) = [A_1(z), \ldots, A_m(z)]$ and deg bounds $\{n_i\}$:

  Find $\mathbf{P}(z) = [P_1(z), ..., P_m(z)]^T$ with deg $P_i(z) \leq n_i$ and

$$\mathbf{A}(z) \cdot \mathbf{P}(z) = O(z^{n_1+\cdots+n_m+m})$$

Sigma (or Order) Bases

Idea : look at order condition independently of degree bounds,

$$R_\sigma = \{\mathbf{Q}(z) \in \mathbb{K}^{(m)}[z] \mid \mathbf{A}(z) \cdot \mathbf{Q}(z) = O(z^\sigma)\}$$

Find basis of R_σ as a *module* over $\mathbb{K}[z]$.

Basis always has m elements : $\mathbf{M}_1(z), \ldots, \mathbf{M}_m(z)$.

- write as columns of an $m \times m$ matrix polynomial $\mathbf{M}(z)$.

Matrix Padé, Hermite-Padé Approximants, etc

- can be represented as vector order problems

Sigma Bases

Degree bounds? Given $\vec{n} = (n_1, \ldots, n_m)$:

Then

$$\mathbf{Q}(z) \quad = \quad \alpha_1(z)\mathbf{M}_1(z) + \cdots + \alpha_m(z)\mathbf{M}_m(z)$$

with

$$\deg \alpha_i(z) \leq \text{ defect } \mathbf{Q}(z) - \text{ defect } \mathbf{M}_i(z)$$

Here : defect $(\mathbf{Q}(z)) = \min_i\{ n_i + 1 - \deg Q_i(z) \}$

$\mathbf{M}(z)$ gives all solutions of $\mathbf{A}(z)\mathbf{Q}(z) = O(z^\sigma)$ having deg bounds .

Sigma Basis Algorithm [SIMAX - BL 1994]

1. Start with Order basis $\mathbf{M}(z) = \mathbf{I}$ and order $= 0$.
2. Of columns $\mathbf{M}_1(z), \ldots, \mathbf{M}_m(z)$ needing order increased:
   - pick one with minimal defect.
   - use to eliminate other columns needing order increase.
3. Multiply pivot column by z. Continue.
4. Quadratic complexity.
5. Double order everytime : obtain superfast version.

Sigma Basis Algorithm [SIMAX - BL 1994]

1. Start with Order basis $\mathbf{M}(z) = \mathbf{I}$ and order $= 0$.
2. Of columns $\mathbf{M}_1(z), \ldots, \mathbf{M}_m(z)$ needing order increased:
   - pick one with minimal defect.
   - use to eliminate other columns needing order increase.
3. Multiply pivot column by z. Continue.
4. Quadratic complexity.
5. Double order everytime : obtain superfast version.

Other Algorithms

- Stable numeric algorithms
  - (1) Padé approximate : (C-M SIMAX 1992)
  - (2) Hermite-Padé, Simultaneous-Padé : (C-J-L SIMAX 1994)
- Fraction-free algorithms
  - (1) Sigma bases : (B-L SIMAX 2000)
  - (2) General case : linear functional with special element

    - e.g. interpolation bases, M-Padé, etc

Superfast Computation

If $\mathbf{F} \in \mathbb{K}[z]^{m \times n}$, $m \leq n$, order σ

- σ-Basis : $O^{\sim}(n^2 m\sigma + nm^2\sigma)$
- MBasis : $O(n^{\omega}\sigma^{1+\epsilon})$ - in case of matrix input (GJV - 2003)
- Generating set : $O^{\sim}(n^{\omega}\lceil m\sigma/n \rceil)$ (Storjohann. 2006)
- Order basis : $O^{\sim}(n^{\omega}\lceil m\sigma/n \rceil)$ Zhou-L (2009)
- Interpolation basis: Neiger-J-S-V (2015) (see Wed talk)

Outline

1. Sigma Bases
2. Minimal Kernel Bases
3. Applications for fast Polynomial Matrix arithmetic
   - Application 1 : Polynomial Matrix Inversion
   - Application 2 : Determinant and Matrix Triangulation
   - Application 3 : Hermite Normal Form

Minimal Kernel Bases

Given $\mathbf{F} \in \mathbb{K}[z]^{m \times n}, \quad m \leq n$:

A *Kernel Basis* for $\mathbf{F}$ is a $\mathbb{K}[z]$ module basis for

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

Can represent basis as matrix $\mathbf{M} \in \mathbb{K}[z]^{n \times *}$.

Minimal Kernel Basis if $\mathbf{M}$ is column reduced.

Shifted $\vec{s}$-Minimal Kernel Basis if $z^{\vec{s}} \cdot \mathbf{M}$ is column reduced.

Minimal Kernel Bases

Given $\mathbf{F} \in \mathbb{K}[z]^{m \times n}, \quad m \leq n$:

A *Kernel Basis* for $\mathbf{F}$ is a $\mathbb{K}[z]$ module basis for

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

Can represent basis as matrix $\mathbf{M} \in \mathbb{K}[z]^{n \times *}$.

Minimal Kernel Basis if $\mathbf{M}$ is column reduced.

Shifted $\vec{s}$-Minimal Kernel Basis if $z^{\vec{s}} \cdot \mathbf{M}$ is column reduced.

Minimal Kernel Bases

Given $\mathbf{F} \in \mathbb{K}[z]^{m \times n}, \quad m \leq n$:

A *Kernel Basis* for $\mathbf{F}$ is a $\mathbb{K}[z]$ module basis for

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

Can represent basis as matrix $\mathbf{M} \in \mathbb{K}[z]^{n \times *}$.

Minimal Kernel Basis if $\mathbf{M}$ is column reduced.

Shifted $\vec{s}$-Minimal Kernel Basis if $z^{\vec{s}} \cdot \mathbf{M}$ is column reduced.

Classical Uses of Kernel Bases

Main application is use of polynomial matrix formalism in systems theory

- reduced transfer functions
- fault diagnostics

Specific algebraic problems

- right and left co-prime transfer function representations
- linear solving
- column reduction
- matrix normal forms

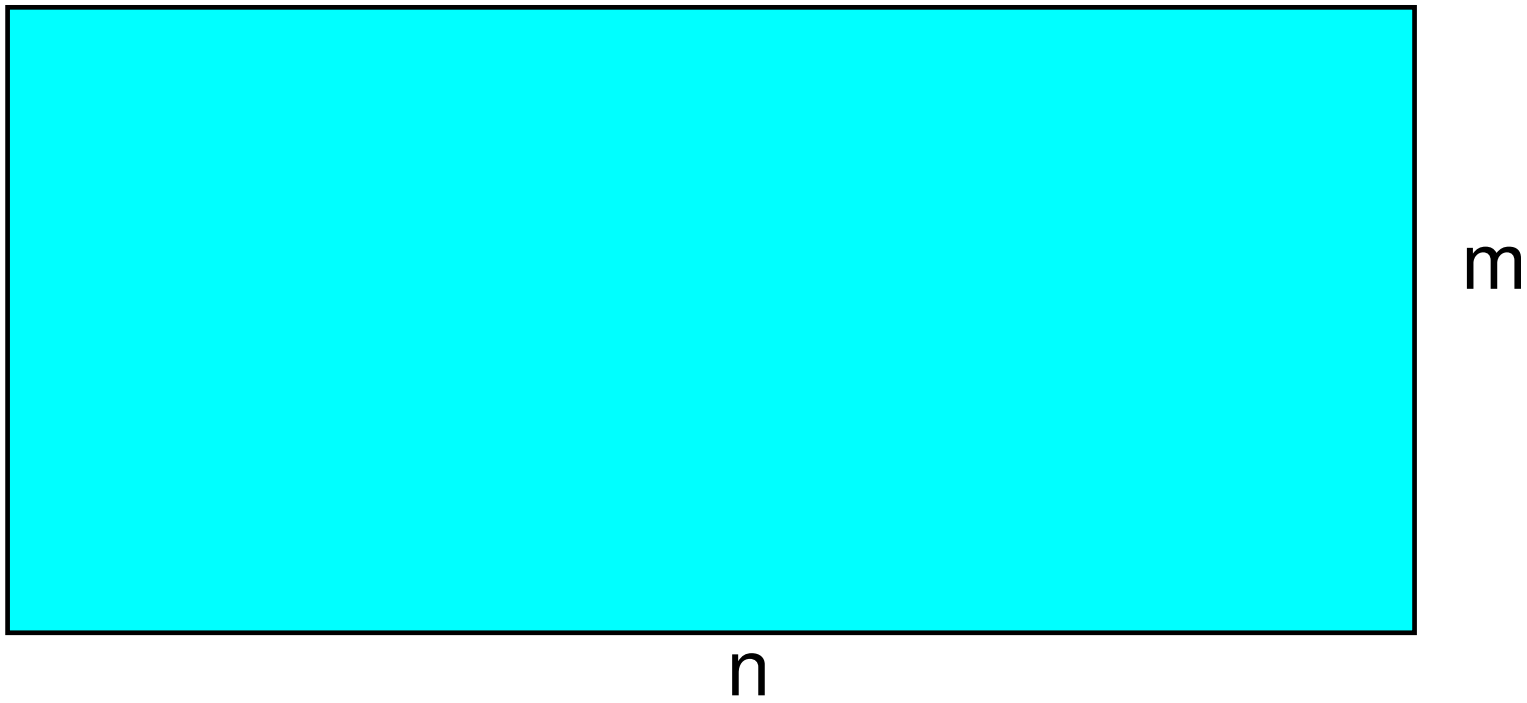
Algorithms

- Matrix pencil methods
  - convert problem to larger matrix size but degree 1.
  - Use Kronecker Normal Forms
  - Beelen, Dooren, Misra, etc.
  - Cost $O(m^2nd^3)$
- Resultant methods
  - convert to structured linear system (e.g. block Toeplitz)
  - use high dimension, get high complexity
- Elimination methods
  - Mulders and Storjohann (2003) Cost : $O(mnrd^2)$
  - Storjohann and Villard (2005) Cost : $O^{\sim}(mnr^{\omega-2}d)$

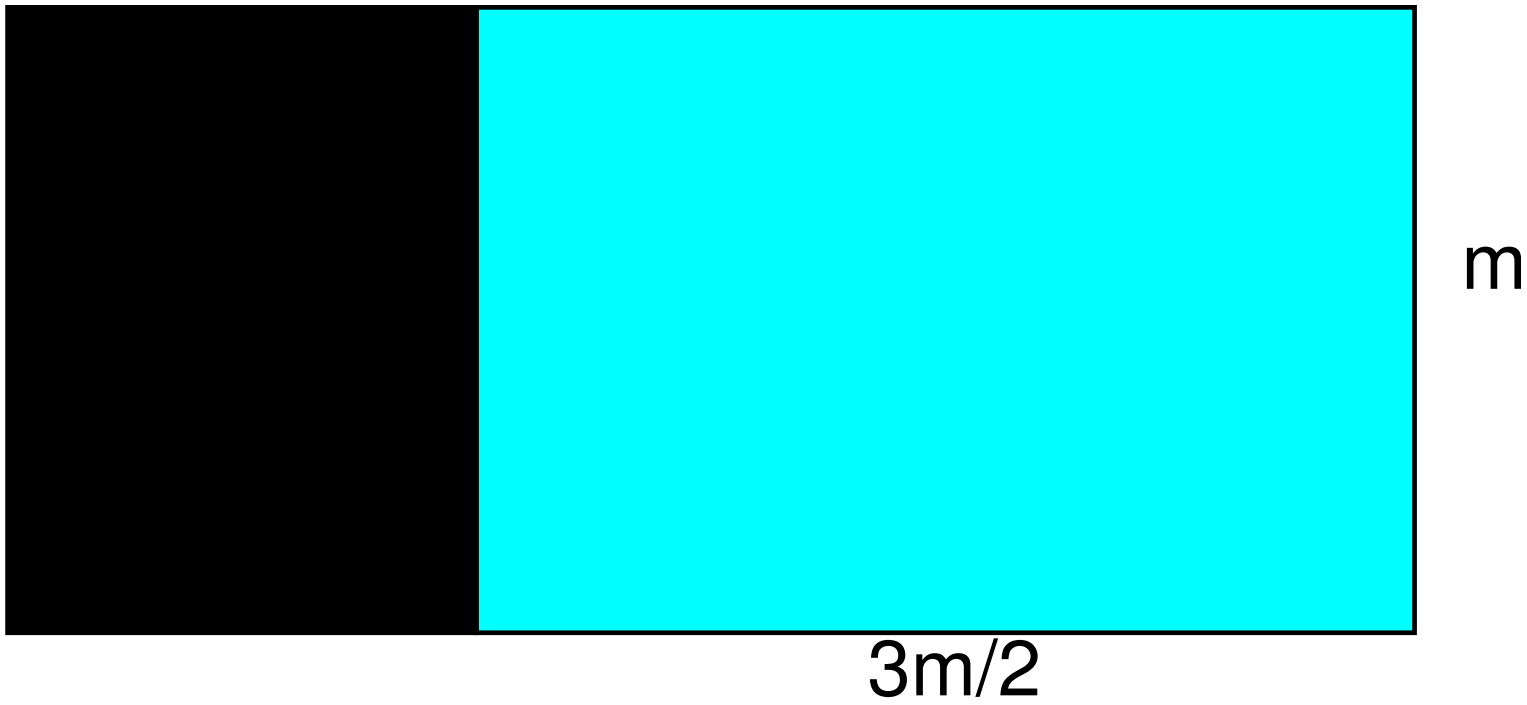
Key Ideas from Zhou, L, Storjohann [ISSAC 2012]

- Order basis computation results in a subset of kernel basis and reduces the column dimension and the degree.
- Row dimension can be reduced by computing the kernel basis of a subset of rows of the input matrix.
- We can use the initial column degrees (or shift) to produce a bound for the degrees of all intermediate subproblems.
- The more general problem involving degree shift can be easier to tackle than the problem without shift
- Can efficiently multiply matrices with unbalanced degrees

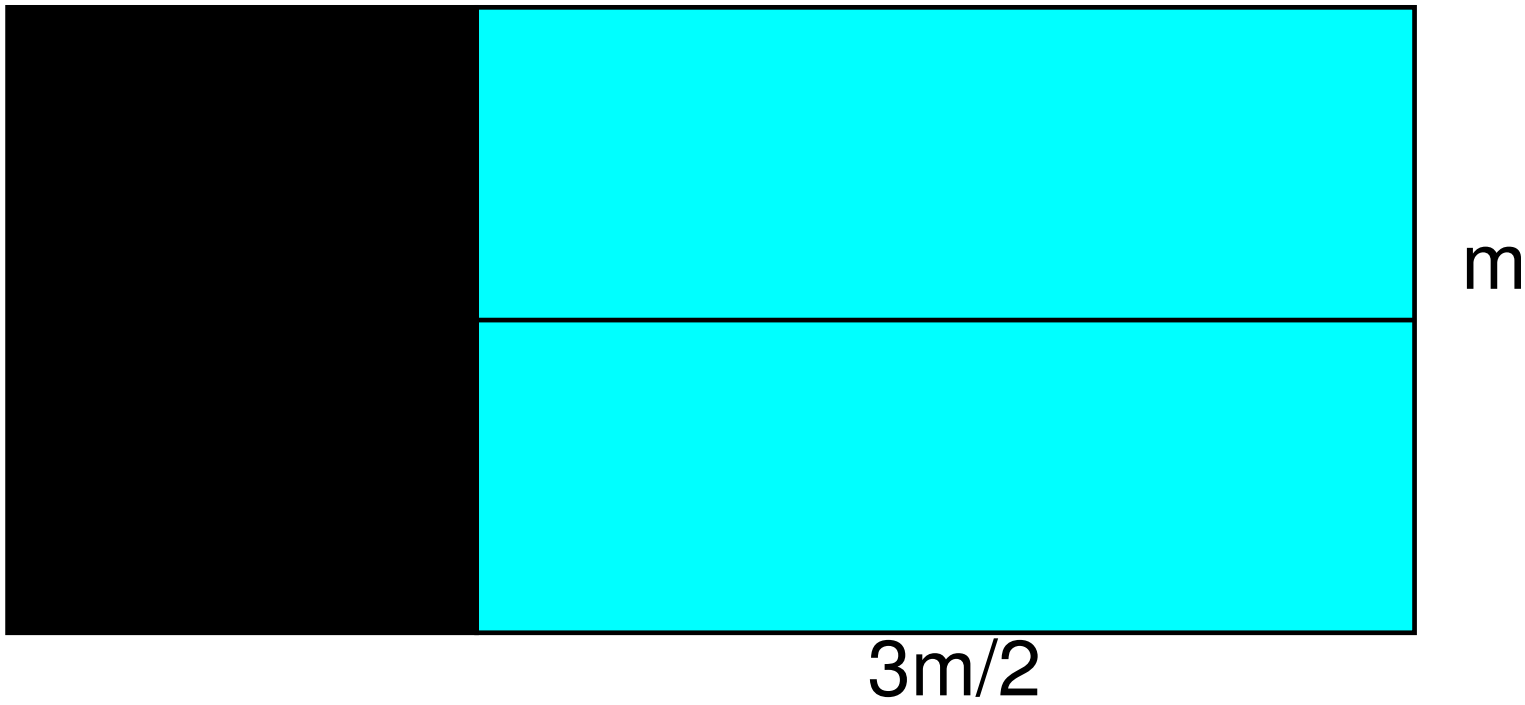
Dimension of Input



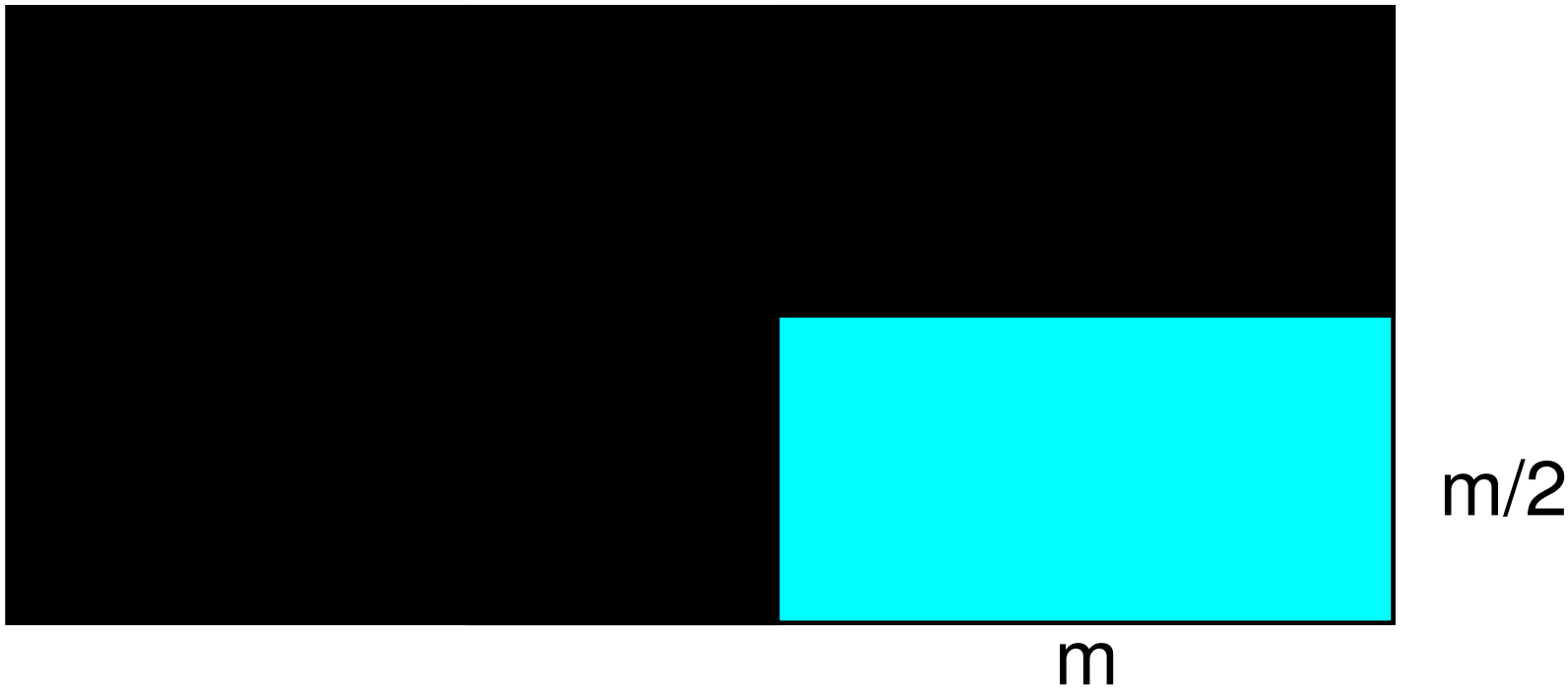
Dimension of Input



Dimension of Input



Dimension of Input



Partial Nullspace Basis using Order Bases

To compute a $\vec{s}$-minimal kernel basis of $\mathbf{F}$ we do:

(1) Compute an $(\mathbf{F}, \sigma, \vec{s})$-basis $\mathbf{P}$ of $\mathbf{F}$

(2) Let $[\mathbf{P}_1, \mathbf{P}_2] = \mathbf{P}$ such that $\mathbf{F}\mathbf{P}_1 = 0$

i.e. $\mathbf{P}_1$ is a part of a $\vec{s}$-minimal kernel basis

(3) Use residual $\mathbf{F}\mathbf{P}_2$ to compute the remaining kernel basis

- Let $\vec{b}_2$ be the $\vec{s}$-column degrees of $\mathbf{P}_2$
- Then $\mathbf{Q}$ is a $\vec{b}_2$-minimal kernel basis of $\mathbf{F}\mathbf{P}_2$, implies $[\mathbf{P}_1, \mathbf{P}_2\mathbf{Q}]$ is $\vec{s}$-minimal kernel basis of $\mathbf{F}$.

Partial Nullspace Basis using Order Bases

To compute a $\vec{s}$-minimal kernel basis of $\mathbf{F}$ we do:

(1) Compute an $(\mathbf{F}, \sigma, \vec{s})$-basis $\mathbf{P}$ of $\mathbf{F}$

(2) Let $[\mathbf{P}_1, \mathbf{P}_2] = \mathbf{P}$ such that $\mathbf{F}\mathbf{P}_1 = 0$

i.e. $\mathbf{P}_1$ is a part of a $\vec{s}$-minimal kernel basis

(3) Use residual $\mathbf{F}\mathbf{P}_2$ to compute the remaining kernel basis

- Let $\vec{b}_2$ be the $\vec{s}$-column degrees of $\mathbf{P}_2$
- Then $\mathbf{Q}$ is a $\vec{b}_2$-minimal kernel basis of $\mathbf{F}\mathbf{P}_2$, implies $[\mathbf{P}_1, \mathbf{P}_2\mathbf{Q}]$ is $\vec{s}$-minimal kernel basis of $\mathbf{F}$.

Partial Nullspace Basis using Order Bases

To compute a $\vec{s}$-minimal kernel basis of $\mathbf{F}$ we do:

(1) Compute an $(\mathbf{F}, \sigma, \vec{s})$-basis $\mathbf{P}$ of $\mathbf{F}$

(2) Let $[\mathbf{P}_1, \mathbf{P}_2] = \mathbf{P}$ such that $\mathbf{F}\mathbf{P}_1 = 0$

i.e. $\mathbf{P}_1$ is a part of a $\vec{s}$-minimal kernel basis

(3) Use residual $\mathbf{F}\mathbf{P}_2$ to compute the remaining kernel basis

- Let $\vec{b}_2$ be the $\vec{s}$-column degrees of $\mathbf{P}_2$
- Then $\mathbf{Q}$ is a $\vec{b}_2$-minimal kernel basis of $\mathbf{F}\mathbf{P}_2$, implies $[\mathbf{P}_1, \mathbf{P}_2\mathbf{Q}]$ is $\vec{s}$-minimal kernel basis of $\mathbf{F}$.

Some Important Properties (ZLS ISSAC 2012)

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

Theorem

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

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

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

Theorem

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

Algorithms

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

Theorem

(Z-L-S ISSAC 2012)
$\vec{s}$-Minimal kernel basis computation costs $O^{\sim}(n^{\omega} s)$.

Also : Column basis : $\mathbf{F} \cdot \mathbf{U} = \mathbf{C}$.

Combining Minimal Kernel and Order Bases computation gives:

Theorem

(Z-L ISSAC 2013)
Column basis computation costs $O^{\sim}(m^{\omega-1} ns)$.

Algorithms

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

Theorem

(Z-L-S ISSAC 2012)
$\vec{s}$-Minimal kernel basis computation costs $O^{\sim}(n^{\omega} s)$.

Also : Column basis : $\mathbf{F} \cdot \mathbf{U} = \mathbf{C}$.

Combining Minimal Kernel and Order Bases computation gives:

Theorem

(Z-L ISSAC 2013)
Column basis computation costs $O^{\sim}(m^{\omega-1} n s)$.

Outline

1. Sigma Bases
2. Minimal Kernel Bases
3. Applications for fast Polynomial Matrix arithmetic
   - Application 1 : Polynomial Matrix Inversion
   - Application 2 : Determinant and Matrix Triangulation
   - Application 3 : Hermite Normal Form

Outline

1. Sigma Bases
2. Minimal Kernel Bases
3. Applications for fast Polynomial Matrix arithmetic
   - Application 1 : Polynomial Matrix Inversion
   - Application 2 : Determinant and Matrix Triangulation
   - Application 3 : Hermite Normal Form

Inversion : Reduce $\mathbf{F}$ to Diagonal Form

Partition and reduce $\mathbf{F}$ via

$$\mathbf{F} \cdot \mathbf{N} = \left[\begin{array}{c} \mathbf{F}_u \\ \hline \mathbf{F}_d \end{array} \right] \cdot [\mathbf{N}_\ell, \mathbf{N}_r] = \left[\begin{array}{cc} \mathbf{F}_u\mathbf{N}_\ell & \mathbf{F}_u\mathbf{N}_r \\ \mathbf{F}_d\mathbf{N}_\ell & \mathbf{F}_d\mathbf{N}_r \end{array} \right] = \left[\begin{array}{cc} \mathbf{R}_u & 0 \\ 0 & \mathbf{R}_d \end{array} \right]$$

Notice:

- $\mathbf{N}_\ell$ a kernel basis for $\mathbf{F}_d$
- $\mathbf{N}_r$ a kernel basis for $\mathbf{F}_u$

Recurse on $\mathbf{R}_u$ and $\mathbf{R}_d$ to get diagonal $\mathbf{B}$. Cost is $O^\sim(n^\omega s)$

Modeled on approach of Jeannerod and Villard (2003). $O^\sim(n^\omega d)$

Previous to Jeannerod/Villard fastest algorithms $O^\sim\left(n^{\omega+1}d\right)$

Measuring Size

$\mathbf{F}$ and $\vec{s}$ bound on column degree.

(i) Partition

$$\mathbf{F} \cdot \mathbf{N} = \left[\begin{array}{c} \mathbf{F}_u \\ \hline \mathbf{F}_d \end{array}\right] \cdot [\mathbf{N}_\ell, \mathbf{N}_r] = \left[\begin{array}{cc} \mathbf{F}_u\mathbf{N}_\ell & \mathbf{F}_u\mathbf{N}_r \\ \mathbf{F}_d\mathbf{N}_\ell & \mathbf{F}_d\mathbf{N}_r \end{array}\right] = \left[\begin{array}{cc} \mathbf{R}_u & 0 \\ 0 & \mathbf{R}_d \end{array}\right]$$

(ii) Size control: $\mathbf{M}$ a $\vec{s}$-minimal kernel basis

$$\sum \operatorname{cdeg}_{\vec{s}} \mathbf{M} \leq \sum \vec{s}$$

Implies : $\sum \operatorname{cdeg}_{\vec{s}} \mathbf{N}_\ell \leq \sum \vec{s}$ and $\sum \operatorname{cdeg}_{\vec{s}} \mathbf{N}_r \leq \sum \vec{s}$

Complexity

Theorem

Inversion of $\mathbf{F} \in \mathbb{K}[x]^{n \times n}$ *costs* $O^{\sim}(n^{\omega} s)$ *field operations.*

Proof.

If cost : $g(n)$ then recurrence relation:

$$\begin{aligned} g(n) &\in O^{\sim}(n^{\omega} s) + g(\lceil n/2 \rceil) + g(\lfloor n/2 \rfloor) \\ &\in O^{\sim}(n^{\omega} s) + 2g(\lceil n/2 \rceil) \\ &\in O^{\sim}(n^{\omega} s). \end{aligned}$$

□

Also compute largest invariant factor : $\operatorname{lcm}(b_{11}(x), \ldots, b_{nn}(x))$.

Complexity

Theorem

Inversion of $\mathbf{F} \in \mathbb{K}[x]^{n \times n}$ *costs* $O^{\sim}(n^{\omega} s)$ *field operations.*

Proof.

If cost : $g(n)$ then recurrence relation:

$$\begin{aligned} g(n) &\in O^{\sim}(n^{\omega} s) + g(\lceil n/2 \rceil) + g(\lfloor n/2 \rfloor) \\ &\in O^{\sim}(n^{\omega} s) + 2g(\lceil n/2 \rceil) \\ &\in O^{\sim}(n^{\omega} s). \end{aligned}$$

□

Also compute largest invariant factor : $\operatorname{lcm}(b_{11}(x), \ldots, b_{nn}(x))$.

Complexity

Theorem

Inversion of $\mathbf{F} \in \mathbb{K}[x]^{n \times n}$ *costs* $O^{\sim}(n^{\omega} s)$ *field operations.*

Proof.

If cost : $g(n)$ then recurrence relation:

$$\begin{aligned} g(n) &\in O^{\sim}(n^{\omega} s) + g(\lceil n/2 \rceil) + g(\lfloor n/2 \rfloor) \\ &\in O^{\sim}(n^{\omega} s) + 2g(\lceil n/2 \rceil) \\ &\in O^{\sim}(n^{\omega} s). \end{aligned}$$

□

Also compute largest invariant factor : lcm($b_{11}(x), \ldots, b_{nn}(x)$).

Outline

1. Sigma Bases
2. Minimal Kernel Bases
3. Applications for fast Polynomial Matrix arithmetic
   - Application 1 : Polynomial Matrix Inversion
   - Application 2 : Determinant and Matrix Triangulation
   - Application 3 : Hermite Normal Form

Triangulating

Partition and reduce $\mathbf{F}$ via

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

(i) $\mathbf{G}_1$ is nonsingular and a column basis of $\mathbf{F}_u$.

(ii) $\mathbf{U}_r$ a right kernel basis of $\mathbf{F}_u$

(iii) $\mathbf{G}_2 = \mathbf{F}_d \cdot \mathbf{U}_r$,

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

Theorem

$\mathbf{F} \in \mathbb{K}[x]^{n \times n}$. Diagonals costs $O^{\sim}(n^\omega s)$ with $s = \frac{\sum \operatorname{cdeg} \mathbf{F}}{n}$.

Triangulating

Partition and reduce $\mathbf{F}$ via

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

(i) $\mathbf{G}_1$ is nonsingular and a column basis of $\mathbf{F}_u$.

(ii) $\mathbf{U}_r$ a right kernel basis of $\mathbf{F}_u$

(iii) $\mathbf{G}_2 = \mathbf{F}_d \cdot \mathbf{U}_r$,

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

Theorem

$\mathbf{F} \in \mathbb{K}[x]^{n \times n}$. Diagonals costs $O^{\sim}(n^\omega s)$ with $s = \frac{\sum \operatorname{cdeg} \mathbf{F}}{n}$.

Triangulating

Partition and reduce $\mathbf{F}$ via

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

(i) $\mathbf{G}_1$ is nonsingular and a column basis of $\mathbf{F}_u$.

(ii) $\mathbf{U}_r$ a right kernel basis of $\mathbf{F}_u$

(iii) $\mathbf{G}_2 = \mathbf{F}_d \cdot \mathbf{U}_r$,

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

Theorem

$\mathbf{F} \in \mathbb{K}[x]^{n \times n}$. Diagonals costs $O^{\sim}(n^\omega s)$ with $s = \frac{\sum \operatorname{cdeg} \mathbf{F}}{n}$.

Determinants

Previous algorithms by:

Storjohann $O^{\sim}(n^{\omega+1}d)$, Mulders and Storjohann $O(n^3d^2)$, E-G-V $O^{\sim}(n^{2+\omega/2}d)$; Storj; G-J-V $O^{\sim}(n^{\omega}d)$; Kaltofen; Kaltofen and Villard

$$\det \mathbf{F} = \frac{\det \mathbf{G}_1 \cdot \det \mathbf{G}_2}{\det \mathbf{U}}$$

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

1. $\det \mathbf{U} = \det \mathbf{U} \bmod z = \det U = \det [U_\ell, U_r]$
2. $\mathbf{V} = \mathbf{U}^{-1} = \begin{bmatrix} \mathbf{V}_u \\ \mathbf{V}_d \end{bmatrix}$
3. $\mathbf{U}_r$ and $\mathbf{V}_u$ determined in column bases computation
4. Find $U_\ell{}^*$ such that $U^* = [U_\ell{}^*, U_r]$ is unimodular
5. Let $V_u = \mathbf{V}_u \bmod z$. Then $\det \mathbf{U} = \frac{\det U^*}{\det V_u U_\ell{}^*}$

Determinants

Previous algorithms by:

Storjohann $O^{\sim}(n^{\omega+1}d)$, Mulders and Storjohann $O(n^3d^2)$, E-G-V $O^{\sim}(n^{2+\omega/2}d)$; Storj; G-J-V $O^{\sim}(n^{\omega}d)$; Kaltofen; Kaltofen and Villard

$$\det \mathbf{F} = \frac{\det \mathbf{G}_1 \cdot \det \mathbf{G}_2}{\det \mathbf{U}}$$

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

1. $\det \mathbf{U} = \det \mathbf{U} \bmod z = \det U = \det [U_\ell, U_r]$
2. $\mathbf{V} = \mathbf{U}^{-1} = \begin{bmatrix} \mathbf{V}_u \\ \mathbf{V}_d \end{bmatrix}$
3. $\mathbf{U}_r$ and $\mathbf{V}_u$ determined in column bases computation
4. Find $U_\ell{}^*$ such that $U^* = [U_\ell{}^*, U_r]$ is unimodular
5. Let $V_u = \mathbf{V}_u \bmod z$. Then $\det \mathbf{U} = \dfrac{\det U^*}{\det V_u U_\ell{}^*}$

Outline

1. Sigma Bases
2. Minimal Kernel Bases
3. Applications for fast Polynomial Matrix arithmetic
   - Application 1 : Polynomial Matrix Inversion
   - Application 2 : Determinant and Matrix Triangulation
   - Application 3 : Hermite Normal Form

Finding Hermite Normal Form

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

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

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

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

Results :

(i) Deterministic algorithm

(ii) Complexity : $O^{\sim}(n^{\omega} d)$ where $d = \text{degree } \mathbf{H}$

Finding Hermite Normal Form

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

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

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

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

Results :

(i) Deterministic algorithm

(ii) Complexity : $O^{\sim}(n^{\omega} d)$ where $d =$ degree $\mathbf{H}$

Hermite Form: Finding Rest of $\mathbf{H}$

Use method of Gupta and Storjohann (2012) to get rest of $\mathbf{H}$.

(i) Convert HNF to shifted $\vec{s}$-minimal kernel basis problem

$$\mathbf{FU} = \mathbf{H} \quad \text{same as} \quad [\mathbf{F} \quad -\mathbf{I}] \begin{bmatrix} \mathbf{U} \\ \mathbf{H} \end{bmatrix} = \mathbf{0}.$$

Hermite Form: Finding Rest of $\mathbf{H}$

Use method of Gupta and Storjohann (2012) to get rest of $\mathbf{H}$.

(ii) Adjust to alternate $\vec{s}'$-minimal kernel basis problem

$$[\mathbf{F} \quad -\mathbf{E}] \begin{bmatrix} \mathbf{U} \\ \mathbf{H}' \end{bmatrix} = \mathbf{0}.$$

Easy to construct $\mathbf{E}$. Easy to get $\mathbf{H}$ from $\mathbf{H}'$

Hermite Form: Finding Rest of $\mathbf{H}$

Use method of Gupta and Storjohann (2012) to get rest of $\mathbf{H}$.

(iii) Find $\mathbf{Q}$ and $\mathbf{R}$ such that $\mathbf{E} = \mathbf{FQ} + \mathbf{R}$. Solve via HOL.

Then repeat (ii) but with $\mathbf{E}$ replaced by $\mathbf{R}$.

Future Work

- Shifted kernel basis algorithm depending on rank
- Popov from Column basis
- Alternate domains (noncommutative)