Fast, deterministic computation of Hermite forms for polynomial matrices

George Labahn

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

Joint work with V. Neiger and Wei Zhou

July 16 2016

Outline

Background story : Hermite forms

Problem : Find a basis as a module for the set of rows $\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_n \end{bmatrix}$.

Each $\mathbf{A}_i$ a square matrix of polynomials.

Problem from finding integral basis needed for algebraic Risch

Method: Reduce

$$\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \\ \mathbf{A}_n \end{bmatrix}$$

Background story with Hermite forms

Problem : Find a basis as a module for the set of rows $\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_n \end{bmatrix}$.

Each $\mathbf{A}_i$ a square matrix of polynomials.

Problem from finding integral basis needed for algebraic Risch

Method: Reduce

$$\underbrace{\begin{bmatrix} \mathbf{U}_{11} & \cdots & \mathbf{U}_{1n} \\ \vdots & & \vdots \\ \mathbf{U}_{n1} & \cdots & \mathbf{U}_{nn} \end{bmatrix}}_{\text{unimodular}} \cdot \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_n \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{B}_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\text{reduced}}$$

Background Story : Matrix Gcds

$$\begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ \mathbf{U}_{21} & \mathbf{U}_{22} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}$$

$$\mathbf{U}_{11} \cdot \mathbf{A}_1 + \mathbf{U}_{12} \cdot \mathbf{A}_2 = \mathbf{B}$$

Background Story : Matrix Gcds

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}$$

$$\mathbf{U}_{11} \cdot \mathbf{A}_1 + \mathbf{U}_{12} \cdot \mathbf{A}_2 = \mathbf{B}$$

$$\mathbf{A}_1 = \mathbf{V}_{11} \cdot \mathbf{B} \quad \text{and} \quad \mathbf{A}_2 = \mathbf{V}_{21} \cdot \mathbf{B}$$

Background Story : Matrix Gcds

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}$$

$$\mathbf{U}_{11} \cdot \mathbf{A}_1 + \mathbf{U}_{12} \cdot \mathbf{A}_2 = \mathbf{B}$$

$$\mathbf{A}_1 = \mathbf{V}_{11} \cdot \mathbf{B} \quad \text{and} \quad \mathbf{A}_2 = \mathbf{V}_{21} \cdot \mathbf{B}$$

$$\mathbf{U} \cdot \mathbf{V} = I_n \quad \Longrightarrow \quad \mathbf{U}_{11} \cdot \mathbf{V}_{11} + \mathbf{U}_{12} \cdot \mathbf{V}_{21} = \mathbf{I}$$

Arne does a masters

Arne's project : see if gcdheu could be extended to HNF

- Seemed like a good idea at the time.

Arne does a masters

Arne's project : see if gcdheu could be extended to HNF

- Seemed like a good idea at the time. Didn't work

Arne does a masters

Arne's project : see if gcdheu could be extended to HNF

- Seemed like a good idea at the time. Didn't work
- Instead Arne discovered probabilistic methods

  (reading HNF paper of Erich, Krishnamoorthy and Dave)

  (reading SNF paper of Erich, Krishnamoorthy and Dave)

Arne does a masters

Arne's project : see if gcdheu could be extended to HNF

- Seemed like a good idea at the time. Didn't work
- Instead Arne discovered probabilistic methods

  (reading HNF paper of Erich, Krishnamoorthy and Dave)

  (reading SNF paper of Erich, Krishnamoorthy and Dave)
- Ultimately Arne had nice Las Vegas result with SNF
- However I was not so comfortable with prob. methods

Arne meets Erich

Invited Erich to Maple Retreat to meet Arne (1994)

- I wanted Erich's comments (e.g. is Arne really correct?)

Arne meets Erich

Invited Erich to Maple Retreat to meet Arne (1994)

- I wanted Erich's comments (e.g. is Arne really correct?)
- Erich said result was correct and okay

  I thought it was better than just okay
- Erich said result was okay

  I said it was really good, etc etc

Arne meets Erich

Invited Erich to Maple Retreat to meet Arne (1994)

- I wanted Erich's comments (e.g. is Arne really correct?)
- Erich said result was correct and okay

  I thought it was better than just okay
- Erich said result was okay

  I said it was really good, etc etc

Erich's point :

Don't tell grad student a result is good. Can always be better

Technical Part I (HNF)

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

Technical Part I (HNF)

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

Our Results :

- Fast, deterministic algorithms
- Complexity : First Try : $O^{\sim}(n^{\omega} d)$ where $d = \deg \mathbf{A}$
- Complexity : Second Try : $O^{\sim}(n^{\omega} \lceil s \rceil)$ where s bounded
  : by average row and column degrees of $\mathbf{A}$

References

Report on paper

- G. Labahn, V. Neiger and W. Zhou,
  Fast, deterministic computation of determinants and Hermite normal forms of polynomial matrices, Arxiv 2016.

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*
- V. Neiger, Fast computation of shifted Popov forms, *ISSAC 2016*

Previous work : HNF

- Polynomial-time over $\mathbb{Q}[x]$: Kannan 1985.
- $\widetilde{\mathcal{O}}(n^4 d)$: Hafner-McCurley 1991,
- $\widetilde{\mathcal{O}}(n^{\omega+1} d)$: Hafner-McCurley (1991), Storjohann and Labahn (1996), Villard (1996)
- $\mathcal{O}(n^3 d^2)$: Mulders and Storjohann (2003)
- $\widetilde{\mathcal{O}}(n^3 d)$: Gupta and Storjohann (2011)
- $\widetilde{\mathcal{O}}(n^\omega d)$: Gupta and Storjohann (2011)
- $\widetilde{\mathcal{O}}(n^\omega s)$: Labahn, Neiger, Zhou (2016)

Algorithm Part I : Finding Diagonal Elements

Partition and reduce $\mathbf{A}$ via

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

Algorithm Part I : Finding Diagonal Elements

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

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

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

Algorithm Part I : Finding Diagonal Elements

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

(ii) $\mathbf{U}_r$ a right kernel 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.

Theorem

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

Example

$$\mathbf{A} = \begin{bmatrix} x & -x^3 & -2x^4 & 2x & -x^2 \\ 1 & -1 & -2x & 2 & -x \\ -3 & 3x^2+x & 2x^2 & -x^4+1 & 3x \\ 0 & 1 & x^2+2x-2 & x^3+2x-2 & 0 \\ 1 & -x^2+2 & -2x^3-3x+3 & 2x+2 & 0 \end{bmatrix} \in \mathbb{Z}_7[x]^{5\times 5}.$$

$$\begin{bmatrix} \mathbf{A}_u \\ \mathbf{A}_d \end{bmatrix} \cdot [\mathbf{U}_\ell, \mathbf{U}_r] = \begin{bmatrix} x & -x^3 & -2x^4 & & \\ 1 & -1 & -2x & & \\ -3 & 3x^2+x & 2x^2 & & \\ * & * & * & x^3-1 & 0 \\ * & * & * & -x & x \end{bmatrix}$$

Example

$$\mathbf{A} = \begin{bmatrix} x & -x^3 & -2x^4 & 2x & -x^2 \\ 1 & -1 & -2x & 2 & -x \\ -3 & 3x^2+x & 2x^2 & -x^4+1 & 3x \\ 0 & 1 & x^2+2x-2 & x^3+2x-2 & 0 \\ 1 & -x^2+2 & -2x^3-3x+3 & 2x+2 & 0 \end{bmatrix} \in \mathbb{Z}_7[x]^{5\times 5}.$$

$$\begin{bmatrix} \mathbf{A}_u \\ \mathbf{A}_d \end{bmatrix} \cdot [\mathbf{U}_\ell^{(2)}, \mathbf{U}_r^{(2)}] = \begin{bmatrix} x & 0 & & & \\ 1 & x^2-1 & & & \\ * & * & x^3 & & \\ * & * & * & x^3-1 & 0 \\ * & * & * & -x & x \end{bmatrix}$$

Example

$$\mathbf{A} = \begin{bmatrix} x & -x^3 & -2x^4 & 2x & -x^2 \\ 1 & -1 & -2x & 2 & -x \\ -3 & 3x^2+x & 2x^2 & -x^4+1 & 3x \\ 0 & 1 & x^2+2x-2 & x^3+2x-2 & 0 \\ 1 & -x^2+2 & -2x^3-3x+3 & 2x+2 & 0 \end{bmatrix} \in \mathbb{Z}_7[x]^{5\times 5}.$$

$$\begin{bmatrix} \mathbf{A}_u \\ \mathbf{A}_d \end{bmatrix} \cdot [\mathbf{U}_\ell^{(2)}, \mathbf{U}_r^{(2)}] = \begin{bmatrix} x & & & & \\ * & x^2-1 & & & \\ * & * & x^3 & & \\ * & * & * & x^3-1 & \\ * & * & * & * & x \end{bmatrix}$$

Algorithm Part II : 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 $\mathbf{A}$.

Algorithm Part II : 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

Algorithm Part II : 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

Use partial linearization of Storjohann (2007): $\mathbf{A} \to \mathcal{L}(\mathbf{A})$

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

Partial Linearization

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

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

$[d]$: 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

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

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.

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

Theorem

$\mathbf{A} \in \mathbb{K}[x]^{n\times n}$ nonsingular. $\vec{\delta}$ degrees of the diag. entries of the Hermite form. Then Hermite form computed using $\widetilde{O}(n^\omega d)$ 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 HNF of $\mathcal{L}^c(\mathbf{A})$

(iii) $\mathbf{H}$ 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: $\widetilde{O}(n^\omega \lceil s \rceil)$.

Future Work

We want to make progress with:

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

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

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

Algorithm Part II : Finding Rest of $\mathbf{H}$ (First Try)

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

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

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

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

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

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

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

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

(iv) Complexity is $O^{\sim}(n^{\omega} d)$