Computing the invariant structure of integer matrices

Colton Pauderis   Arne Storjohann

David R. Cheriton School of Computer Science
University of Waterloo

June 28, 2013

Matrix normal forms: Hermite form

Triangular basis H for row lattice of input matrix $A \in \mathbb{Z}^{n \times n}$

- Obtained using unimodular row operations: $H = UA$
- Non-negative diagonal entries.
- Reduced off-diagonal entries: $0 \leq H_{ij} < H_{jj}$ for $i < j$.

$$\overset{A}{\begin{bmatrix} -13 & 10 & -20 & 27 \\ 27 & 30 & 15 & 30 \\ 0 & 15 & 15 & 6 \\ -21 & 0 & -15 & 9 \end{bmatrix}} \longrightarrow \overset{H}{\begin{bmatrix} 1 & 5 & 5 & 0 \\ & 15 & 0 & 15 \\ & & 15 & 12 \\ & & & 21 \end{bmatrix}}$$

- Fact: $\prod_{j=1}^{i} H_{jj} =$ gcd of $i \times i$ minors in first i columns of A

Computing the Hermite form: Some previous results

Asymptotically fast algorithms

Given a nonsingular $n \times n$ input matrix. Counting bit operations.

- Kannan and Bachem (1979)
  - polynomial
- Hafner and McCurley (1991)
  - $\tilde{O}(n^4)$
- Storjohann and Labahn (1996)
  - $\tilde{O}(n^{\omega+1})$

Our goal:

- $\tilde{O}(n^3)$ bit operations using standard integer arithmetic

Matrix normal forms: Smith form

Diagonal form $S \in \mathbb{Z}^{n \times n}$ Smith Normal Form: $S \in \mathbb{Z}^{n \times n}$

- Obtained using unimodular row and column operations: $S = UAV$
- $S = \text{diag}(s_1, s_2, \cdots, s_n)$
- $\{s_i\}$ are *invariant factors* of A: $s_{i-1} \mid s_i$

$$\overset{A}{\begin{bmatrix} -13 & 10 & -20 & 27 \\ 27 & 30 & 15 & 30 \\ 0 & 15 & 15 & 6 \\ -21 & 0 & -15 & 9 \end{bmatrix}} \longrightarrow \overset{S}{\begin{bmatrix} 1 & & & \\ & 3 & & \\ & & 15 & \\ & & & 105 \end{bmatrix}}$$

- Fact: $\prod_{j=1}^{i} s_i =$ gcd of all $i \times i$ minors of A

Invariant factor through system solving

Idea: use nonsingular system solving to find s_n.

- Pick random vector $v \in \mathbb{Z}^{n \times 1}$.
- Find $x = A^{-1}v \in \mathbb{Q}^{n \times 1}$.
- $\operatorname{lcm}(\operatorname{denom}(x))$ is likely a large factor of s_n.

Previous appearances of this idea:

- Pan (1988)
- Abbott, Bronstein, Mulders (1999)
- Eberly, Giesbrecht, Villard (2000)
- Saunders, Wan (2004)

Triangular lattice decomposition

Write Hermite form H as product of triangular matrices:

$$H = U\,(T_1\,T_2\,T_3\ldots T_n)$$

Each T_i corresponds to a projection $A^{-1}v$ (and roughly to s_i).

For $H = \begin{bmatrix} 1 & 5 & 5 & 0 \\ & 15 & 0 & 15 \\ & & 15 & 12 \\ & & & 21 \end{bmatrix}$ and $S = \mathrm{diag}(1, 3, 15, 105)$

$$H = U \left(\underbrace{\begin{bmatrix} 1 & \mathbf{1} & & \\ & \mathbf{3} & & \\ & & 1 & \\ & & & 1 \end{bmatrix}}_{T_2} \underbrace{\begin{bmatrix} 1 & \mathbf{0} & & \mathbf{2} \\ & \mathbf{5} & & \mathbf{1} \\ & & 1 & \mathbf{1} \\ & & & \mathbf{3} \end{bmatrix}}_{T_3} \underbrace{\begin{bmatrix} 1 & & \mathbf{10} & \mathbf{6} \\ & 1 & \mathbf{8} & \mathbf{6} \\ & & \mathbf{15} & \mathbf{5} \\ & & & \mathbf{7} \end{bmatrix}}_{T_4} \right)$$

Minimal triangular denominator

Definition:

- Given $x = A^{-1}v \in \mathbb{Q}^{n\times 1}$, find triangular $T \in \mathbb{Z}^{n\times n}$ of minimal determinant with Tx integral.
- T is a *minimal triangular denominator*.

Idea:

- Let $w := dx$, with $d \in \mathbb{Z}_{>0}$ such that w is integral.
- Hermite form of $\left[\begin{array}{c|c} d & \\ \hline w & I_n \end{array}\right] \in \mathbb{Z}^{n+1\times n+1}$ is $\left[\begin{array}{c|c} * & * \\ \hline & T \end{array}\right]$.

Minimal triangular denominator

Combine two approaches to find Hermite form of $\left[\begin{array}{c|c} d & \\ \hline w & I_n \end{array}\right]$.

1. Use unimodular row operations to find diagonal entries of T.
   - Computing all of T this way is prohibitively costly.
2. Appeal to definition of T as minimal denominator for off-diagonal entries.

Total cost: $O(n(\log d)^2)$ bit operations

Minimal triangular denominator

Off diagonal-entries:

- Fill one column at a time (i.e., no row operations)
- Total size of diagonal entries bounded by d

Total cost: $O(n(\log d)^2)$ bit operations

$$\underbrace{\begin{bmatrix} 210604204176 \\ -28570205200 \\ -85901387498 \\ -25110292487 \\ -163096297770 \end{bmatrix}}_{dx} \longrightarrow \underbrace{\begin{bmatrix} 1 & & & & \\ & 373 & & & \\ & & 7 & & \\ & & & 352910 & \\ & & & & 541 \end{bmatrix}}_{T}$$

Minimal triangular denominator

Off diagonal-entries:

- Fill one column at a time (i.e., no row operations)
- Total size of diagonal entries bounded by d

Total cost: $O(n(\log d)^2)$ bit operations

$$\underbrace{\begin{bmatrix} 1174883912 \\ -28570205200 \\ -230298626 \\ -67319819 \\ -437255490 \end{bmatrix}}_{dx} \longrightarrow \underbrace{\begin{bmatrix} 1 & 341 & & & \\ & 373 & & & \\ & & 7 & & \\ & & & 352910 & \\ & & & & 541 \end{bmatrix}}_{T}$$

Minimal triangular denominator

Off diagonal-entries:

- Fill one column at a time (i.e., no row operations)
- Total size of diagonal entries bounded by d

Total cost: $O(n(\log d)^2)$ bit operations

$$\underbrace{\begin{bmatrix} 36241344 \\ 112402422 \\ -230298626 \\ -9617117 \\ -62465070 \end{bmatrix}}_{dx} \longrightarrow \underbrace{\begin{bmatrix} 1 & 341 & 4 & & \\ & 373 & 6 & & \\ & & 7 & & \\ & & & 352910 & \\ & & & & 541 \end{bmatrix}}_{T}$$

Minimal triangular denominator

Off diagonal-entries:

- Fill one column at a time (i.e., no row operations)
- Total size of diagonal entries bounded by d

Total cost: $O(n(\log d)^2)$ bit operations

$$\underbrace{\begin{bmatrix} 65 \\ 32 \\ 527 \\ -9617117 \\ -177 \end{bmatrix}}_{dx} \longrightarrow \underbrace{\begin{bmatrix} 1 & 341 & 4 & 118402 & \\ & 373 & 6 & 252396 & \\ & & 7 & 135232 & \\ & & & 352910 & \\ & & & & 541 \end{bmatrix}}_{T}$$

Minimal triangular denominator

Off diagonal-entries:

- Fill one column at a time (i.e., no row operations)
- Total size of diagonal entries bounded by d

Total cost: $O(n(\log d)^2)$ bit operations

$$\underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -177 \end{bmatrix}}_{dx} \longrightarrow \underbrace{\begin{bmatrix} 1 & 341 & 4 & 118402 & 251 \\ & 373 & 6 & 252396 & 315 \\ & & 7 & 135232 & 437 \\ & & & 352910 & 469 \\ & & & & 541 \end{bmatrix}}_{T}$$

Extracting triangular denominators

First projection captures T_n, the portion of H corresponding to s_n.

After i projections...

- Hermite form captured this far $\overline{H} \cong T_{n-i+1} \cdots T_{n-1} T_n$.
- "Pull out" $\overline{H}$ from initial matrix:
  $B := A(T_{n-i+1} \cdots T_{n-1} T_n)^{-1}$.
- Subsequent projections operate on B.
- Non-trivial invariant factors of B: $s_1, s_2, \ldots, s_{n-i}$.

Repeat process (project, find T_i, pull out) for rest of Hermite form.

p-adic lifting

Efficient nonsingular system solving is based on p-adic lifting.

Given $A \in \mathbb{Z}^{n\times n}$ and $v \in \mathbb{Z}^{n\times m}$, find $x = A^{-1}v \in \mathbb{Q}^{n\times m}$.
Compute:

- Low precision inverse: $\tilde{O}(n^3)$.
  - $A^{-1} \bmod p$
- Truncated p-adic expansion of solution: $\tilde{O}(n^2 m\ell)$
  - $A^{-1}v = c_0 + c_1 p + \cdots + c_{i-1}p^{\ell-1} \mod p^\ell$

Cost depends on size m, precision ℓ; want $m\ell \in O(n)$.

Problems with repeated system solving

Consider a matrix with $k \in \Omega(n)$ nontrivial invariant factors.

- Requires solving $\Omega(n)$ systems at full precision.
- As costly as computing exact inverse.
- $A^{-1}v$ may have numerator much larger than its denominator.

$$A^{-1}v = \begin{bmatrix} \frac{-2826334476994}{15} \\ \frac{-5485776224414}{15} \\ \vdots \\ \frac{-9437737474004}{15} \end{bmatrix}$$

Ideally, leverage decreasing size of remaining invariant factor.

High-order residue

Use *high-order residue* $R \in \mathbb{Z}^{n \times n}$ to compress further projections.

$$A^{-1} = (A^{-1} \bmod p^{\ell}) + A^{-1}\mathbf{R}p^{\ell}$$

- $A^{-1}v$ may have numerator much larger than its denominator.
- $A^{-1}Rv$ is a nearly proper matrix fraction.

E.g., for $A \in \mathbb{Z}^{10\times10}$, $v \in \mathbb{Z}^{10\times1}$,

$$A^{-1}v = \begin{bmatrix} \frac{-2826334476994}{15} \\ \frac{-5485776224414}{15} \\ \vdots \\ \frac{-9437737474004}{15} \end{bmatrix} \quad A^{-1}Rv = \begin{bmatrix} \frac{46}{15} \\ \frac{11}{15} \\ \vdots \\ \frac{26}{15} \end{bmatrix}$$

p-adic lifting

As largest remaining invariant factor s_n decreases...

- Required solve precision ℓ decreases proportionally.
- Projection size m can be increased.

$$\ell = 4 \quad m = 1 \quad s_i = 6545$$

$$\underbrace{\begin{bmatrix} \frac{4307}{6545} \\ \frac{5815}{6545} \\ \frac{3360}{6545} \\ \frac{2768}{6545} \\ \frac{5928}{6545} \end{bmatrix}}_{A^{-1}Rv} \equiv \begin{bmatrix} 95 \\ 8 \\ 12 \\ 96 \\ 37 \end{bmatrix} 97^0 + \begin{bmatrix} 66 \\ 25 \\ 58 \\ 76 \\ 44 \end{bmatrix} 97^1 + \begin{bmatrix} 66 \\ 76 \\ 57 \\ 72 \\ 58 \end{bmatrix} 97^2 + \begin{bmatrix} 88 \\ 42 \\ 65 \\ 39 \\ 96 \end{bmatrix} 97^3 \bmod 97^4$$

p-adic lifting

As largest remaining invariant factor s_n decreases...

- Required solve precision ℓ decreases proportionally.
- Projection size m can be increased.

$$\ell = 2 \quad m = 2 \quad s_i = 55$$

$$\underbrace{\begin{bmatrix} \frac{49}{55} & \frac{6}{11} \\ \frac{46}{55} & \frac{3}{11} \\ \frac{2}{55} & 0 \\ \frac{2}{5} & \frac{41}{55} \\ \frac{15}{11} & \frac{4}{55} \end{bmatrix}}_{A^{-1}Rv} \equiv \begin{bmatrix} 39 & 46 \\ 96 & 23 \\ 59 & 0 \\ 33 & 94 \\ 18 & 21 \end{bmatrix} 97^0 + \begin{bmatrix} 89 & 68 \\ 77 & 34 \\ 7 & 0 \\ 32 & 58 \\ 74 & 15 \end{bmatrix} 97^1 \bmod 97^2$$

p-adic lifting

As largest remaining invariant factor s_n decreases...

- Required solve precision ℓ decreases proportionally.
- Projection size m can be increased.

$$\ell = 1 \quad m = 4 \quad s_i = 5$$

$$\underbrace{\begin{bmatrix} \frac{3}{5} & \frac{4}{5} & \frac{2}{5} & 1 \\ 0 & \frac{3}{5} & \frac{2}{5} & \frac{4}{5} \\ 0 & \frac{3}{5} & 0 & \frac{1}{5} \\ \frac{1}{5} & \frac{3}{5} & 1 & \frac{4}{5} \\ \frac{1}{5} & \frac{1}{5} & 1 & \frac{4}{5} \end{bmatrix}}_{A^{-1}Rv} \equiv \begin{bmatrix} 20 & 59 & 78 & 1 \\ 0 & 20 & 78 & 59 \\ 0 & 20 & 0 & 39 \\ 39 & 20 & 1 & 59 \\ 39 & 39 & 1 & 59 \end{bmatrix} 97^0 \bmod 97$$

Verification

How many iterations of the process are required?

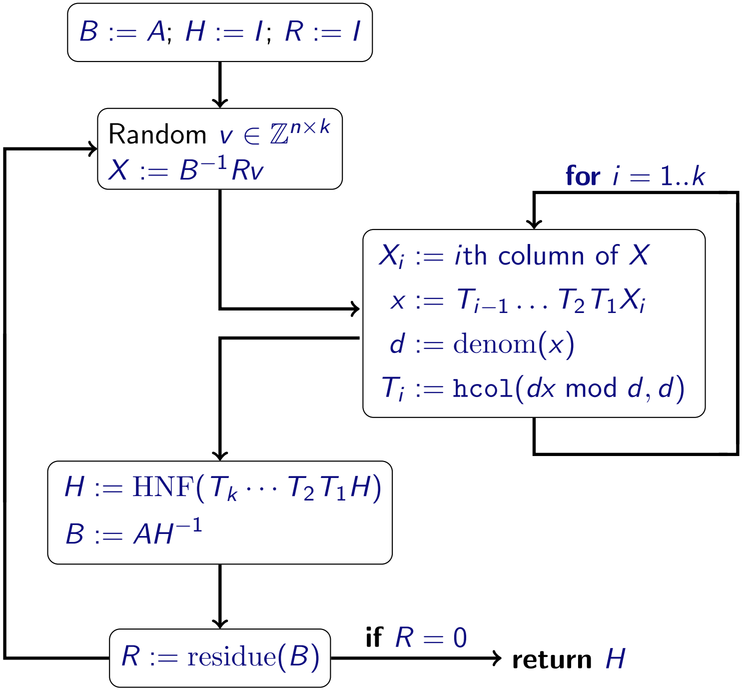
How can we know when we are done?

- If $B = AH^{-1}$ is unimodular, H is the entire Hermite form of A.
- High-order residue R of B can detect this:

$$B^{-1} = (B^{-1} \bmod p^{\ell}) + B^{-1}\mathbf{R}p^{\ell}$$

$$R = 0 \iff \det B = \pm 1$$

- Algorithm is Las Vegas randomized.



$B := A;\ H := I;\ R := I$
Random $v \in \mathbb{Z}^{n \times k}$
$X := B^{-1}Rv$
for $i = 1..k$
$X_i :=$ ith column of X
$x := T_{i-1} \dots T_2 T_1 X_i$
$d := \mathrm{denom}(x)$
$T_i := \mathtt{hcol}(dx \bmod d, d)$
$H := \mathrm{HNF}(T_k \cdots T_2 T_1 H)$
$B := AH^{-1}$
$R := \mathrm{residue}(B)$
if $R = 0$
return H

Random matrices are well-suited to this method.

- Matrices with i.i.d. entries of a specified size.
- HNF has few non-trivial diagonal entries, one large entry:
  - e.g. $n = 20$: $1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,$ $229197391045696758685556941390636601552950456 22$

		time[1] (s)		
entry size	n	this	Magma 2.19	Sage 5.2[2]
8 bits	500	7.57	**6.00**	21.19
	1000	51.73	**48.23**	139.98
	2000	398.40	**370.73**	1013.93
32 bits	500	**21.71**	28.68	33.02
	1000	**148.72**	238.39	226.57
	2000	**1144.75**	1739.44	

[1] AMD Opteron 8356 @ 1.15 GHz

[2] Pernet and Stein (2010)

Matrices with highly non-trivial Hermite forms are challenging.

- Build from diagonal matrix via random row/column ops.
  - as per Allan Steel's "Hermite Normal Form Timings Page"[3]
- HNF has about $n/2$ non-trivial diagonal entries:
  - $n = 20$: $1, \ldots, 1, 2, 6, 2, 12, 18, 12, 252, 33264,$
    $395134740, 8084487861597125114136\,0$

n	this	Magma 2.19	Sage 5.2
100	**0.150**	0.330	2.01
200	3.67	**2.12**	31.39
400	19.05	**14.03**	480.9
800	124.77	**97.69**	
1000	229.93	**196.72**	

[3] http://magma.maths.usyd.edu.au/users/allan/mat/hermite.html

A still more difficult class of matrices:

- $A_{ij} = (i-1)^{(j-1)} \bmod n$, for prime n
  - as per Jaeger, Wagner (2009)[4]
- HNF has about $n/2$ non-trivial, non-smooth diagonal entries:
  - $n = 29$: $1, \ldots, 1, 2, 4, 4, 4, 4, 540, 4, 16, 4333140, 1008,$
    $472312260, 12907349441280, 11772$

n	this	Magma 2.19	Sage 5.2
101	**0.52**	1.98	2.29
211	**2.98**	44.17	38.06
401	**20.54**	1528	912.9
809	**123.6**		
1009	**232.1**		

[4] "Efficient parallelizations of Hermite and Smith normal form algorithms", J. of Parallel Comp.

Comparision against determinant

n	this	Magma 2.19	Sage 5.2
100	(0.30)	0.070	0.23
200	(1.82)	0.480	0.90
400	(11.42)	4.150	5.45
800	(78.56)	38.960	35.55
1000	(148.72)	73.330	67.77

Random, 32 bit entries

n	this	Magma 2.19	Sage 5.2
100	(0.150)	0.180	0.66
200	(3.67)	0.960	2.67
400	(19.05)	6.590	17.15
800	(124.77)	40.350	137.78

Nontrivial Hermite diagonal (Steel)