Computing the Hermite form of an integer matrix

Arne Storjohann

David R. Cheriton School of Computer Science
University of Waterloo

June 13, 2018

Integer matrix triangularization

- Interested in the lattice generated by $\mathbb{Z}$-linear combinations of rows of an integer matrix A
- Lattice is invariant under unimodular row operations
  1. Add an integer multiple of one row to a different row
  2. Multiply a row by -1
  3. Swap two rows
- Obtainable using "Gaussian elimination" with integer row operations

$$A = \begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix}$$

Integer matrix triangularization: Example

- Triangularize A using integer row operations
- Record row operations in U, initially set to I_n

$$\begin{array}{ccc} U & A & UA \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix} = & \begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix} \end{array}$$

- Next row operation: Swap rows 1 and 3

Integer matrix triangularization: Example

- Triangularize A using integer row operations
- Record row operations in U, initially set to I_n

$$\overset{U}{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}} \overset{A}{\begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix}} = \overset{UA}{\begin{bmatrix} 2 & -7 & 6 \\ -6 & 3 & 3 \\ -4 & -4 & -1 \end{bmatrix}}$$

- Next row operation: Add 3 times row 1 to row 2

Integer matrix triangularization: Example

- Triangularize A using integer row operations
- Record row operations in U, initially set to I_n

$$\overset{U}{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 3 \\ 1 & 0 & 0 \end{bmatrix}} \overset{A}{\begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix}} = \overset{UA}{\begin{bmatrix} 2 & -7 & 6 \\ & -18 & 21 \\ -4 & -4 & -1 \end{bmatrix}}$$

- Next row operation: Add 2 times row 1 to row 3

Integer matrix triangularization: Example

- Triangularize A using integer row operations
- Record row operations in U, initially set to I_n

$$\overset{U}{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 3 \\ 1 & 0 & 2 \end{bmatrix}} \overset{A}{\begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix}} = \overset{UA}{\begin{bmatrix} 2 & -7 & 6 \\ & -18 & 21 \\ & -18 & 11 \end{bmatrix}}$$

- Next row operation: Subract row 2 from row 3

Integer matrix triangularization: Example

- Triangularize A using integer row operations
- Record row operations in U, initially set to I_n

$$\overset{U}{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 3 \\ 1 & -1 & -1 \end{bmatrix}} \overset{A}{\begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix}} = \overset{UA}{\begin{bmatrix} 2 & -7 & 6 \\ & -18 & 21 \\ & & -10 \end{bmatrix}}$$

- Next row operation: Multiply row 2 by -1

Integer matrix triangularization: Example

- Triangularize A using integer row operations
- Record row operations in U, initially set to I_n

$$\begin{array}{ccccc} U & & A & & UA \\ \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & -3 \\ 1 & -1 & -1 \end{bmatrix} & & \begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix} & = & \begin{bmatrix} 2 & -7 & 6 \\ & 18 & -21 \\ & & -10 \end{bmatrix} \end{array}$$

- Next row operation: Add row 2 to row 1

Integer matrix triangularization: Example

- Triangularize A using integer row operations
- Record row operations in U, initially set to I_n

$$\begin{array}{ccccc} U & & A & & UA \\ \begin{bmatrix} 0 & -1 & -2 \\ 0 & -1 & -3 \\ 1 & -1 & -1 \end{bmatrix} & & \begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix} & = & \begin{bmatrix} 2 & 11 & -15 \\ & 18 & -21 \\ & & -10 \end{bmatrix} \end{array}$$

- Next row operations: Put last column in correct form

Integer matrix triangularization: Example

- Triangularize A using integer row operations
- Record row operations in U, initially set to I_n

$$\begin{array}{ccc} U & A & UA \\ \begin{bmatrix} -2 & 1 & 0 \\ -3 & 2 & 0 \\ -1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix} = & \begin{bmatrix} 2 & 11 & 5 \\ & 18 & 9 \\ & & 10 \end{bmatrix} \end{array}$$

- U is unimodular: $\det U = -1$
- $H := UA$ is the *Hermite form* for A

The Hermite form

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

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

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

- Fact: $\prod_{j=1}^{i} H_{jj} =$ gcd of $i \times i$ minors in first i columns of A
- Fact: $\det H = |\det A|$ has bitlength $O(n \times \log n||A||)$

The Hermite form: arbitrary shape and rank input

- Example: 6×9 input matrix with rank 3

$$A \xrightarrow{hnf} \begin{bmatrix} h_1 & * & * & \bar{*} & \bar{*} & * & * & * & * \\ & & & h_2 & \bar{*} & * & * & * & * \\ & & & & h_3 & * & * & * & * \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{bmatrix}$$

- Column rank profile of A is $[1, 4, 5]$
- Define Hermite basis $\bar{H} = \begin{bmatrix} h_1 & \bar{*} & \bar{*} \\ & h_2 & \bar{*} \\ & & h_3 \end{bmatrix}$

The Hermite form: full row rank

- Example: 3×9 input matrix with rank 3

$$A \xrightarrow{hnf} \begin{bmatrix} h_1 & * & * & \bar{*} & \bar{*} & * & * & * & * \\ & & & h_2 & \bar{*} & * & * & * & * \\ & & & & h_3 & * & * & * & * \end{bmatrix}$$

- Let $\bar{A} = \left[\ A_1 \ \middle\|\ A_2\ \right] = AC$, perm. C based on rank profile

$$\bar{A} \xrightarrow{hnf} \left[\begin{array}{ccc||cccccc} h_1 & \bar{*} & \bar{*} & * & * & * & * & * & * \\ & h_2 & \bar{*} & * & * & * & * & * & * \\ & & h_3 & * & * & * & * & * & * \end{array}\right]$$

The Hermite form: full row rank

- Example: 3×9 input matrix with rank 3

$$A \longrightarrow \begin{bmatrix} h_1 & * & * & \bar{*} & \bar{*} & * & * & * & * \\ & & & h_2 & \bar{*} & * & * & * & * \\ & & & & h_3 & * & * & * & * \end{bmatrix}$$

- Let $\bar{A} = \left[\ A_1 \ \middle\| \ A_2\ \right] = AC$, perm. C based on rank profile

$$\bar{A} \xrightarrow{hnf} \left[\quad \bar{H} \quad \middle\| \quad \bar{H} A_1^{-1} A_2 \quad \right]$$

The Hermite form: arbitrary rank

- Input $A \in \mathbb{Z}^{n\times m}$ of arbitrary rank r

Let

$$\left[\begin{array}{c||c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] = RAC$$

where

- C a permutation based on column rank profile
- R *any* row permutation such that $A_{11} \in \mathbb{Z}^{r\times r}$ is nonsingular

Then

$$\left[\begin{array}{c||c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \xrightarrow{hnf} \left[\begin{array}{c||c} \bar{H} & \bar{H}A_{11}^{-1}A_{12} \\ \hline & \end{array}\right]$$

What about the transforming matrix? $UA = H$

$A \in \mathbb{Z}^{n\times n}$ nonsingular

- U is unique and can be found using linear solving: $U = HA^{-1}$
- also note that $\left[\begin{array}{c|c} A & I_n \end{array}\right] \xrightarrow{hnf} \left[\begin{array}{c|c} H & U \end{array}\right]$

$A \in \mathbb{Z}^{n\times m}$ of full column rank

- use rank computation to reduce to nonsingular case

$$A \xrightarrow{(1)} \left[\begin{array}{c} A_1 \\ \hline A_2 \end{array}\right] \xrightarrow{(2)} \left[\begin{array}{c|c} A_1 & \\ \hline A_2 & I_{n-m} \end{array}\right] \xrightarrow{hnf} \left[\begin{array}{c|c} \bar{H} & U_1 \\ \hline & U_2 \end{array}\right]$$

Then

$$U = \left[\begin{array}{c|c} \bar{H} - U_1 A_2 & U_1 \\ \hline -U_2 A_2 & U_2 \end{array}\right] \left[\begin{array}{c|c} A_1^{-1} & \\ \hline & I_{n-m} \end{array}\right]$$

The transforming matrix: a closer view

- Let $A \in \mathbb{Z}^{n \times m}$ have full column rank
- Let $U \in \mathbb{Z}^{n \times n}$ be unimodular such that $UA = H$

$$\begin{array}{c} U \\ \left[\begin{array}{c} E \\ \hline \\ K \\ \\ \end{array}\right] \end{array} \begin{array}{c} A \\ \left[\begin{array}{c} * \\ \hline \\ * \\ \\ \end{array}\right] \end{array} = \left[\begin{array}{c} \bar{H} \\ \hline \\ \\ \\ \end{array}\right]$$

$E \in \mathbb{Z}^{m \times n}$ and $K \in \mathbb{Z}^{(n-m) \times n}$ arbitrary but satisfy

1. $EA = \bar{H}$
   $\rightarrow$ E is a solution to the *extended gcd problem* for A
2. $KA = 0_{(n-m) \times m}$ and columns of K span I_{n-m}
   $\rightarrow$ K is a *(left) kernel basis* for A

Computing the Hermite form: Pre-21'st century

Hermite form of full column rank $A \in \mathbb{Z}^{n \times m}$.
Counting bit operations. Assuming standard matrix multiplication.

- Kannan & Bachem (79)
  - polynomial time
- Chou and Collins (82)
  - $\tilde{O}(n^6 \log ||A||)$
- Domich *et al. (87), Iliopoulos (89), Hafner & McCurley (91)*
  - $\tilde{O}(nm^3 \log ||A||)$
- *Storjohann & Labahn (96), Storjohann (00)*
  - $\tilde{O}(nm^3 \log ||A||)$ *with transformation* U
- *Sims (84), Havas, Majewski & Matthews (98)*
  - *application of lattice reduction (LLL)*

Elimination based methods

- Extended gcd: given $a, b \in \mathbb{Z}$ compute $s, t \in \mathbb{Z}$ such that

$$sa + tb = g = \gcd(a, b) \iff \begin{bmatrix} s & t \\ -b/g & a/g \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} g \\ \end{bmatrix}$$

- "Gaussian elimination" via unimodular transformations

$$\begin{bmatrix} s & & & t & \\ & 1 & & & \\ & & 1 & & \\ -b/g & & & a/g & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} a & * \\ & * \\ & * \\ b & * \\ * & * \end{bmatrix} = \begin{bmatrix} g & * \\ & * \\ & * \\ & * \\ * & * \end{bmatrix}$$

Elimination based methods with potential expression swell

1. Triangularize column by column

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \xrightarrow{1,2,3} \left[\begin{array}{ccc|cc} h_1 & * & * & * & * \\ & h_2 & * & * & * \\ & & h_3 & * & * \\ & & & * & * \\ & & & * & * \end{array}\right] \xrightarrow{4,5} \left[\begin{array}{ccc|cc} h_1 & * & * & * & * \\ & h_2 & * & * & * \\ & & h_3 & * & * \\ & & & h_4 & * \\ & & & & h_5 \end{array}\right]$$

2. Reduce off diagonals column by column

$$\begin{bmatrix} h_1 & * & * & * & * \\ & h_2 & * & * & * \\ & & h_3 & * & * \\ & & & h_4 & * \\ & & & & h_5 \end{bmatrix} \xrightarrow{1,2,3} \left[\begin{array}{ccc|cc} h_1 & \bar{*} & \bar{*} & * & * \\ & h_2 & \bar{*} & * & * \\ & & h_3 & * & * \\ & & & h_4 & * \\ & & & & h_5 \end{array}\right] \xrightarrow{4,5} \left[\begin{array}{ccc|cc} h_1 & \bar{*} & \bar{*} & \bar{*} & \bar{*} \\ & h_2 & \bar{*} & \bar{*} & \bar{*} \\ & & h_3 & \bar{*} & \bar{*} \\ & & & h_4 & \bar{*} \\ & & & & h_5 \end{array}\right]$$

- Fang & Havas (ISSAC 97): Exponential growth in bit length

Elimation based methods with controlled growth

1. Triangularize row by row

$$\begin{bmatrix} \ast & \ast & \ast & \ast & \ast \\ \ast & \ast & \ast & \ast & \ast \\ \ast & \ast & \ast & \ast & \ast \\ \ast & \ast & \ast & \ast & \ast \\ \ast & \ast & \ast & \ast & \ast \end{bmatrix} \xrightarrow{1,2,3} \left[\begin{array}{ccc|cc} \ast & \bar{\ast} & \bar{\ast} & \ast & \ast \\ & \ast & \bar{\ast} & \ast & \ast \\ & & \ast & \ast & \ast \\ \hline\hline \ast & \ast & \ast & \ast & \ast \\ \ast & \ast & \ast & \ast & \ast \end{array}\right] \xrightarrow{4} \left[\begin{array}{cccc|c} \ast & \bar{\ast} & \bar{\ast} & \bar{\ast} & \ast \\ & \ast & \bar{\ast} & \bar{\ast} & \ast \\ & & \ast & \bar{\ast} & \ast \\ & & & \ast & \ast \\ \hline\hline \ast & \ast & \ast & \ast & \ast \end{array}\right]$$

2. During triangularization reduce off diagonals row by row

$$\left[\begin{array}{ccc|cc} \ast & \ast & \ast & \ast & \ast \\ & \ast & \ast & \ast & \ast \\ & & \ast & \ast & \ast \\ \hline\hline \ast & \ast & \ast & \ast & \ast \\ \ast & \ast & \ast & \ast & \ast \end{array}\right] \xrightarrow{2} \left[\begin{array}{ccc|cc} \ast & \ast & \ast & \ast & \ast \\ & \ast & \bar{\ast} & \ast & \ast \\ & & \ast & \ast & \ast \\ \hline\hline \ast & \ast & \ast & \ast & \ast \\ \ast & \ast & \ast & \ast & \ast \end{array}\right] \xrightarrow{1} \left[\begin{array}{ccc|cc} \ast & \bar{\ast} & \bar{\ast} & \ast & \ast \\ & \ast & \bar{\ast} & \ast & \ast \\ & & \ast & \ast & \ast \\ \hline\hline \ast & \ast & \ast & \ast & \ast \\ \ast & \ast & \ast & \ast & \ast \end{array}\right]$$

- Chou & Collins (86): $O\tilde{}(n^6 \log ||A||)$ bit operations

Modulo determinant methods

Let $A \in \mathbb{Z}^{n \times n}$ be nonsingular.

1. Compute $d = \det A$.
2. Triangularize working modulo d
3. "Fix up" diagonal entries in using extended gcd computations
4. Reduce off diagonal entries working modulo d

Cost is $O(n^3)$ operations modulo d: $\tilde{O}(n^4 \log ||A||)$ bit operations

Computing a transforming matrix

- recursive kernel basis computation of Hafner & McCurley (91)
- example of $8m \times m$ input matrix $(n = 8m)$
- kernel produced has at most $(n/m)\log_2(n/m)$ nonzero blocks

$$\begin{bmatrix} I &&&&&&& \\ & I &&&&&& \\ && I &&&&& \\ &&& I &&&& \\ &&&& I &&& \\ &&&&& I && \\ &&&&&& I & \\ &&&&&&& I \end{bmatrix} \begin{bmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{bmatrix}$$

Computing a transforming matrix

- recursive kernel basis computation of Hafner & McCurley (91)
- example of $8m \times m$ input matrix ($n = 8m$)
- kernel produced has at most $(n/m)\log_2(n/m)$ nonzero blocks

$$\begin{bmatrix} * & * & & & & & & \\ * & * & & & & & & \\ & & * & * & & & & \\ & & * & * & & & & \\ & & & & * & * & & \\ & & & & * & * & & \\ & & & & & & * & * \\ & & & & & & * & * \end{bmatrix} \begin{bmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{bmatrix} = \begin{bmatrix} b_1 \\ \\ b_2 \\ \\ b_3 \\ \\ b_4 \\ \\ \end{bmatrix}$$

Computing a transforming matrix

- recursive kernel basis computation of Hafner & McCurley (91)
- example of $8m \times m$ input matrix ($n = 8m$)
- kernel produced has at most $(n/m)\log_2(n/m)$ nonzero blocks

$$\begin{bmatrix} * & * & * & * & & & & \\ * & * & & & & & & \\ * & * & * & * & & & & \\ & & * & * & & & & \\ & & & & * & * & * & * \\ & & & & * & * & & \\ & & & & * & * & * & * \\ & & & & & & * & * \end{bmatrix} \begin{bmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{bmatrix} = \begin{bmatrix} c_1 \\ \\ \\ \\ c_2 \\ \\ \\ \\ \end{bmatrix}$$

Computing a transforming matrix

- recursive kernel basis computation of Hafner & McCurley (91)
- example of $8m \times m$ input matrix $(n = 8m)$
- kernel produced has at most $(n/m)\log_2(n/m)$ nonzero blocks

$$\left[\begin{array}{cccccccc} * & * & * & * & * & * & * & * \\ \hline * & * & & & & & & \\ * & * & * & * & & & & \\ & & * & * & & & & \\ * & * & * & * & * & * & * & * \\ & & & & * & * & & \\ & & & & * & * & * & * \\ & & & & & & * & * \end{array}\right] \left[\begin{array}{c} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{array}\right] = \left[\begin{array}{c} d_1 \\ \hline \\ \\ \\ \\ \\ \\ \\ \end{array}\right]$$

Summary: Elimination based computation of Hermite form

- Input $A \in \mathbb{Z}^{n \times m}$ of rank r
- Let $\beta = (\sqrt{r}||A||)^r$

$$\begin{array}{ccccc} U & & A & & H \\ \left[\begin{array}{c} E \\ \hline \\ K \\ \\ \end{array}\right] & & \left[\begin{array}{c} * \\ \hline \\ * \\ \\ \end{array}\right] & = & \left[\begin{array}{c} * \\ \hline \\ \\ \\ \end{array}\right] \end{array}$$

In $O\tilde{}(nmr^2 \log ||A||)$ bit operations we can compute U, H with

- $||H|| \leq \beta$
- $||E|| \leq \beta$
- $||K|| \leq r\beta^2$

Using lattice reduction

- Sims (84): Use LLL after the fact to reduce size of U
- Example for random 8×4 input

```
>RandomMatrix(8,4);
```

$$A = \begin{bmatrix} 50 & -38 & -93 & 8 \\ 10 & -18 & -76 & 69 \\ -16 & 87 & -72 & 99 \\ -9 & 33 & -2 & 29 \\ -50 & -98 & -32 & 44 \\ -22 & -77 & -74 & 92 \\ 45 & 57 & -4 & -31 \\ -81 & 27 & 27 & 67 \end{bmatrix}$$

Using lattice reduction

- Sims (84): Use LLL after the fact to reduce size of U
- Example for random 8×4 input

```
>HermiteForm(A,output=['U']);
```

$$U = \left[\begin{array}{c} E \\ \hline K \end{array}\right] = \left[\begin{array}{rrrrr} 9086206 & -8629726 & -5060749 & 29822997 & 3 \\ 8363956 & -7943761 & -4658477 & 27452408 & 3 \\ 2467635 & -2343664 & -1374400 & 8099340 & 9 \\ 3307464 & -3141301 & -1842160 & 10855850 & 1 \\ \hline 9507196 & -9029566 & -5295228 & 31204782 & 3 \\ 9199746 & -8737563 & -5123987 & 30195660 & 3 \\ 6367234 & -6047352 & -3546362 & 20898713 & 2 \\ 3264133 & -3100146 & -1818026 & 10713625 & 1 \end{array}\right.$$

Using lattice reduction

- Sims (84): Use LLL after the fact to reduce size of U
- Example for random 8×4 input

```
>HermiteForm(A,output=['U']);
```

$$U = \left[\begin{array}{c} E \\ \hline \mathrm{LLL}(K) \end{array}\right] = \left[\begin{array}{ccccc} 9086206 & -8629726 & -5060749 & 29822997 & 3 \\ 8363956 & -7943761 & -4658477 & 27452408 & 3 \\ 2467635 & -2343664 & -1374400 & 8099340 & 9 \\ 3307464 & -3141301 & -1842160 & 10855850 & 1 \\ \hline 11 & -80 & -2 & 50 & \\ 2 & 26 & -26 & 69 & \\ -41 & 36 & 16 & 22 & \\ 12 & 1 & 15 & 47 & \end{array}\right.$$

Using lattice reduction

- Sims (84): Use LLL after the fact to reduce size of U
- Example for random 8×4 input

```
>HermiteForm(A,output=['U'],method='integer[reduced]')
```

$$U = \left[\frac{E + M\,\mathrm{LLL}(K)}{\mathrm{LLL}(K)} \right] = \left[\begin{array}{cccccccc} 0 & 4 & -3 & 59 & 21 & -18 & -8 & -18 \\ -18 & 24 & 12 & 1 & 8 & -25 & -31 & -26 \\ 3 & 5 & -3 & -16 & -7 & 1 & 2 & 10 \\ -21 & -57 & 24 & -18 & 9 & 48 & 33 & -23 \\ \hline 11 & -80 & -2 & 50 & 18 & 58 & 65 & 1 \\ 2 & 26 & -26 & 69 & 43 & -24 & 47 & 8 \\ -41 & 36 & 16 & 22 & 41 & -36 & -9 & -47 \\ 12 & 1 & 15 & 47 & -20 & -31 & -101 & -36 \end{array} \right]$$

Using lattice reduction

- Havas, Majewski & Matthews (98)
- Scale columns of A with large $L \in \mathbb{Z}$ and apply LLL
- Produces Hermite form with rows reversed

$$\left[\; L^m a_1 \;\middle|\; L^{m-1} a_2 \;\middle|\; \cdots \;\middle|\; L a_m \;\right] \xrightarrow{LLL} \left[\begin{array}{c|c|c|c} & & & Lh_m \\ & & & \vdots \\ & L^{m-1}h_2 & \cdots & L\bar{*} \\ L^m h_1 & L^{m-1}\bar{*} & \cdots & L\bar{*} \end{array}\right]$$

Example with $L = 100$

$$\begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix} \xrightarrow{scale} \begin{bmatrix} -40000 & -400 & -1 \\ -60000 & 300 & 3 \\ 20000 & -700 & 6 \end{bmatrix} \xrightarrow{LLL} \begin{bmatrix} & & 10 \\ & 1800 & 9 \\ 20000 & 1100 & 5 \end{bmatrix}$$

Hermite form and auxilliary problems: Complexity

Hermite form

- Curent worst case: $O\tilde{}(n^4 \log ||A||)$
- Goal: $O\tilde{}(n^3 \log ||A||)$

Auxilliary problems

1. Determinant
   - Classical: $O\tilde{}(n^4 \log ||A||)$
   - Eberly, Giesbrecht & Villard (00): $O\tilde{}(n^{3.5} (\log ||A||)^{1.5})$
   - Kaltofen (92): $O\tilde{}(n^{3.5} \log ||A||)$
   - Kaltofen & Villard (04): $O\tilde{}(n^{3.2} \log ||A||)$ (Las Vegas)
   - Storjohann (05): $O\tilde{}(n^3 \log ||A||)$ (Las Vegas)
2. Rank
   - Classical: $O\tilde{}(n^2 \log ||A|| + n^3 \log\log ||A||)$ (Monte Carlo)
   - Storjohann (09): $O\tilde{}(n^3 \log ||A||)$ (Las Vegas)
3. Kernel basis computation
   - Chen, Stehle & Villard (ISSAC 18)

Alternative definition of Hermite form

$$A = \begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix}$$

- Let A be square nonsingular with $UA = H$ in Hermite form
- Note that $U = HA^{-1}$ is integral

Definition A nonsingular H is the Hermite form of A if

- H is in Hermite form
- $\det H$ is minimal
- HA^{-1} is integral

$$\begin{bmatrix} h_1 & h_{12} & h_{13} \\ & h_2 & h_{23} \\ & & h_3 \end{bmatrix} \overbrace{\begin{bmatrix} -\frac{13}{120} & -\frac{31}{360} & \frac{1}{40} \\ -\frac{7}{60} & \frac{11}{180} & -\frac{1}{20} \\ -\frac{1}{10} & \frac{1}{10} & \frac{1}{10} \end{bmatrix}}^{A^{-1}}$$

Alternative definition of Hermite form

$$A = \begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix}$$

- Let A be square nonsingular with $UA = H$ in Hermite form
- Note that $U = HA^{-1}$ is integral

Definition A nonsingular H is the Hermite form of A if

- H is in Hermite form
- $\det H$ is minimal
- HA^{-1} is integral

$$\begin{bmatrix} h_1 & h_{12} & h_{13} \\ & h_2 & h_{23} \\ & & 10 \end{bmatrix} \overbrace{\begin{bmatrix} -\frac{13}{120} & -\frac{31}{360} & \frac{1}{40} \\ -\frac{7}{60} & \frac{11}{180} & -\frac{1}{20} \\ -\frac{1}{10} & \frac{1}{10} & \frac{1}{10} \end{bmatrix}}^{A^{-1}}$$

Alternative definition of Hermite form

$$A = \begin{bmatrix} -4 & -4 & -1 \\ -6 & 3 & 3 \\ 2 & -7 & 6 \end{bmatrix}$$

- Let A be square nonsingular with $UA = H$ in Hermite form
- Note that $U = HA^{-1}$ is integral

<u>Definition</u> A nonsingular H is the Hermite form of A if

- H is in Hermite form
- $\det H$ is minimal
- HA^{-1} is integral

$$\begin{bmatrix} h_1 & h_{12} & h_{13} \\ & 18 & 9 \\ & & 10 \end{bmatrix} \overset{A^{-1}}{\begin{bmatrix} -\frac{13}{120} & -\frac{31}{360} & \frac{1}{40} \\ -\frac{7}{60} & \frac{11}{180} & -\frac{1}{20} \\ -\frac{1}{10} & \frac{1}{10} & \frac{1}{10} \end{bmatrix}}$$

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: $S = UAV$
- $S = \mathrm{diag}(s_1, s_2, \cdots, s_n)$
- $\{s_i\}$ 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.

Some previous appearances of this idea:

- Pan (88)
- Abbott, Bronstein, Mulders (99)
- Eberly, Giesbrecht, Villard (00)
- Saunders, Wan (04)

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 $S = \mathrm{diag}(1, 3, 15, 105)$

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

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 $d \in \mathbb{Z}_{>0}$ such that $dx \in \mathbb{Z}^{n\times 1}$ is integral.
- Tx is integral if and only if $T(dx)$ is zero modulo d.

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

$$\underbrace{\begin{bmatrix} 65 \\ 32 \\ 527 \\ 831 \\ -177 \end{bmatrix}}_{dx} \longrightarrow \underbrace{\begin{bmatrix} 1 & 341 & 4 & 118402 & \\ & 373 & 6 & 252396 & \\ & & 7 & 135232 & \\ & & & 352910 & \\ & & & & 541 \end{bmatrix}}_{T}$$

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

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

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

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 $A \in \mathbb{Z}^{10\times 10}$, $v \in \mathbb{Z}^{10\times 1}$,

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

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?

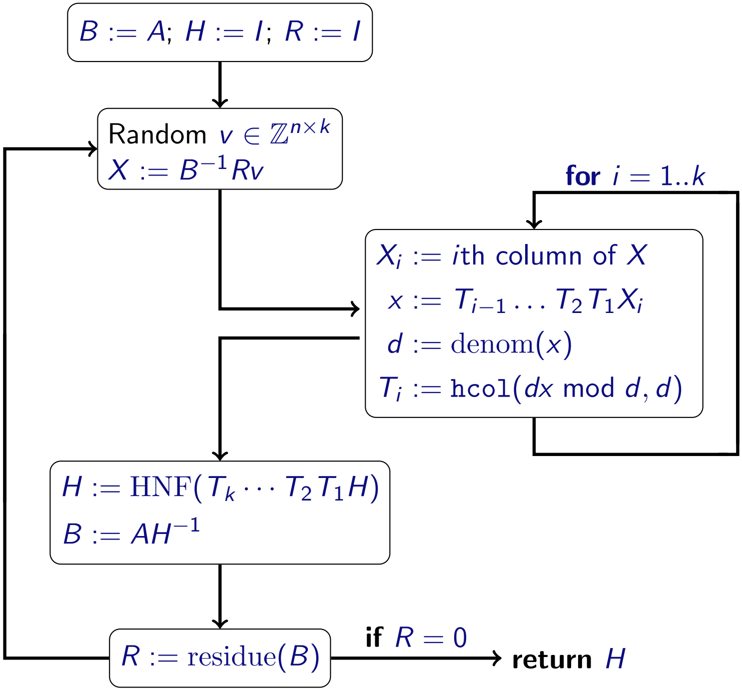
How can we know when we are done?

- 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
$x := T_{i-1}\dots T_2T_1X_i$
$d := \mathrm{denom}(x)$
$T_i := \texttt{hcol}(dx \bmod d, d)$
$H := \mathrm{HNF}(T_k\cdots T_2T_1H)$
$B := AH^{-1}$
$R := \mathrm{residue}(B)$
if $R = 0$
return H

Experimental results

Random matrices

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. $n = 20$: $1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,$ 22919739104569675868555694139063660155295045622

entry size	n	time[1] (s) 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.

Compact representations of inverses

Trivial Hermite form

$$A = \begin{bmatrix} -1 & -7 & 8 & 7 & -9 & -5 \\ -8 & -6 & 7 & -7 & 7 & -3 \\ -8 & 2 & 7 & -5 & -7 & 5 \\ -6 & -2 & 4 & 4 & -1 & 7 \\ -8 & -1 & -7 & -8 & -5 & -7 \\ -1 & -5 & 4 & 6 & 9 & -7 \end{bmatrix} \xrightarrow{hnf} \begin{bmatrix} 1 & & & & & 752896 \\ & 1 & & & & 77390 \\ & & 1 & & & 1236066 \\ & & & 1 & & 1512582 \\ & & & & 1 & 664447 \\ & & & & & 1885994 \end{bmatrix}$$

- Setting $d = |\det(A)| = 1885994$ we have

$$dA^{-1} = \begin{bmatrix} 61738 & 72090 & -104952 & -109262 & -110006 & -149216 \\ -136854 & -222242 & 314760 & -148880 & 3006 & 265942 \\ 27366 & 19552 & 150638 & -113194 & -96600 & 63078 \\ -6482 & -144120 & 35580 & 88148 & 23708 & 156250 \\ -84990 & 29215 & -11531 & 12884 & -26660 & 79494 \\ -10258 & 73649 & -108085 & 149390 & -55588 & -165890 \end{bmatrix}$$

Compact representations of inverses

Trivial Hermite form: Xiaoyu Liu (17)

$$A = \begin{bmatrix} -1 & -7 & 8 & 7 & -9 & -5 \\ -8 & -6 & 7 & -7 & 7 & -3 \\ -8 & 2 & 7 & -5 & -7 & 5 \\ -6 & -2 & 4 & 4 & -1 & 7 \\ -8 & -1 & -7 & -8 & -5 & -7 \\ -1 & -5 & 4 & 6 & 9 & -7 \end{bmatrix} \xrightarrow{hnf} \begin{bmatrix} 1 & & & & & 752896 \\ & 1 & & & & 77390 \\ & & 1 & & & 1236066 \\ & & & 1 & & 1512582 \\ & & & & 1 & 664447 \\ & & & & & 1885994 \end{bmatrix}$$

- Setting $d = |\det(A)| = 1885994$ we have

$$dA^{-1} = \begin{bmatrix} -752896 \\ -77390 \\ -1236066 \\ -1512582 \\ -664447 \\ 1 \end{bmatrix} \begin{bmatrix} -10258 \\ 73649 \\ -108085 \\ 149390 \\ -55588 \\ -165890 \end{bmatrix}^T \mod d$$

Compact representations of inverses

Unimodular matrix

$$A = \begin{bmatrix} 5 & 3 & -4 & -8 & 2 & 2 \\ -4 & -5 & 7 & -2 & -5 & 0 \\ 8 & 1 & 9 & 4 & 8 & -18 \\ -2 & 0 & 4 & 2 & -1 & -3 \\ 6 & -9 & -4 & 3 & 4 & 1 \\ -3 & -7 & 4 & 5 & 5 & -6 \end{bmatrix}$$

- A is unimodular with

$$A^{-1} = \begin{bmatrix} 7598 & -4014 & -3346 & 26309 & 4289 & 131 \\ 7825 & -4134 & -3446 & 27095 & 4417 & 135 \\ 26947 & -14236 & -11867 & 93307 & 15211 & 465 \\ 6551 & -3461 & -2885 & 22684 & 3698 & 113 \\ 21202 & -11201 & -9337 & 73414 & 11968 & 366 \\ 28164 & -14879 & -12403 & 97521 & 15898 & 486 \end{bmatrix}$$

Compact representations of inverses

Unimodular matrix

$$A = \begin{bmatrix} 5 & 3 & -4 & -8 & 2 & 2 \\ -4 & -5 & 7 & -2 & -5 & 0 \\ 8 & 1 & 9 & 4 & 8 & -18 \\ -2 & 0 & 4 & 2 & -1 & -3 \\ 6 & -9 & -4 & 3 & 4 & 1 \\ -3 & -7 & 4 & 5 & 5 & -6 \end{bmatrix}$$

- A is unimodular with

$$A^{-1} = \begin{bmatrix} -167461 \\ -172465 \\ -593913 \\ -144388 \\ -467291 \\ -620736 \end{bmatrix} \begin{bmatrix} -452358 \\ 238980 \\ 199211 \\ -1566341 \\ -255347 \\ -7805 \end{bmatrix}^T \ \mathrm{quo}\ 9970006$$