Linear Algebra 1: Computing canonical forms in exact linear algebra

Clément PERNET,

LIG/INRIA-MOAIS, Grenoble Université, France

ECRYPT II: Summer School on Tools,
Mykonos, Grece,
June 1st, 2012

Introduction : matrix normal forms

Given a transformation $B = f(A)$,

- Identifiy invariants
- Unique representant of an equivalence class
- Simplify computations (structured form)

Different types:

Equivalence over a field: $B = UA$, where U is invertible

- Reduced echelon form:

$$E = \begin{bmatrix} 1 & * & 0 & * & * & 0 & * \\ & & 1 & * & * & 0 & * \\ & & & & & 1 & * \\ & & & & & & \end{bmatrix}$$

- Gauss-Jordan elimination

Introduction : matrix normal forms

Given a transformation $B = f(A)$,

- Identifiy invariants
- Unique representant of an equivalence class
- Simplify computations (structured form)

Different types:

Equivalence over a ring: $B = UA$, where $\det(U) = \pm 1$

- Hermite normal form: $$H = \begin{bmatrix} p_1 & * & x_{1,2} & * & * & x_{1,3} & * \\ & & p_2 & * & * & x_{2,3} & * \\ & & & & & p_3 & * \\ & & & & & & \end{bmatrix}$$, with

$0 \le x_{*,j} < p_j$

Introduction : matrix normal forms

Given a transformation $B = f(A)$,

- Identifiy invariants
- Unique representant of an equivalence class
- Simplify computations (structured form)

Different types:

Similarity over a field: $B = U^{-1}AU$

- Frobenius normal form (or canonical rational form):

$$F = \begin{bmatrix} C_{P_0} & & & \\ & C_{P_1} & & \\ & & \ddots & \\ & & & C_{P_k} \end{bmatrix}, \text{ with } p_{i+1} | p_i \text{ and } P_0 = \text{MinPoly}(A).$$

- Krylov method or ZigZag elimination

Motivation

Equivalence over a field: Gaussian elimination

- Reduced echelon form, rank profile: PolSys-Gröbner basis.
- Linear system solving: sieves, index calculus, ...

Equivalence over a ring: lattice reduction

- Hermite normal form: $\mathbb{Z}$-modules and their saturation
- short vector problem:
  - hard problem
  - help improve computational complexities

Complexities

Matrix multiplication: door to fast linear algebra

- over a field: $\mathcal{O}(n^\omega)$. $\omega \in]2.3727, 3]$ (exponent of linear algebra)
- over $\mathbb{Z}$: $\mathcal{O}(n^\omega M(\log \|A\|)) = \mathcal{O}\tilde{}(n^\omega \log \|A\|)$

Equivalence over a field: Reduced echelon form

- Gauss-Jordan: $\mathcal{O}(n^\omega)$

Complexities

Matrix multiplication: door to fast linear algebra

- over a field: $\mathcal{O}(n^\omega)$. $\omega \in]2.3727, 3]$ (exponent of linear algebra)
- over $\mathbb{Z}$: $\mathcal{O}(n^\omega M(\log \|A\|)) = \mathcal{O}\tilde{}(n^\omega \log \|A\|)$

Equivalence over a field: Reduced echelon form

- Gauss-Jordan: $\mathcal{O}(n^\omega)$

Equivalence over $\mathbb{Z}$: Hermite normal form

- [Kannan & Bachem 79]: $\in P$
- [Chou & Collins 82]: $\mathcal{O}\tilde{}\left(n^6 \log \|A\|\right)$
- [Domich & Al. 87], [Illiopoulos 89]: $\mathcal{O}\tilde{}\left(n^4 \log \|A\|\right)$
- [Micciancio & Warinschi01]: $\mathcal{O}\tilde{}\left(n^5 \log \|A\|^2\right)$, $\mathcal{O}\tilde{}\left(n^3 \log \|A\|\right)$ heur.
- [Storjohann & Labahn 96]: $\mathcal{O}\tilde{}\left(n^{\omega+1} \log \|A\|\right)$

Complexities

Matrix multiplication: door to fast linear algebra

- over a field: $\mathcal{O}(n^\omega)$. $\omega \in]2.3727, 3]$ (exponent of linear algebra)
- over $\mathbb{Z}$: $\mathcal{O}(n^\omega M(\log \|A\|)) = \mathcal{O}\tilde{~}(n^\omega \log \|A\|)$

Equivalence over a field: Reduced echelon form

- Gauss-Jordan: $\mathcal{O}(n^\omega)$

Equivalence over $\mathbb{Z}$: Hermite normal form

- [Kannan & Bachem 79]: $\in P$
- [Chou & Collins 82]: $\mathcal{O}\tilde{~}\left(n^6 \log \|A\|\right)$
- [Domich & Al. 87], [Illiopoulos 89]: $\mathcal{O}\tilde{~}\left(n^4 \log \|A\|\right)$
- [Micciancio & Warinschi01]: $\mathcal{O}\tilde{~}\left(n^5 \log \|A\|^2\right)$, $\mathcal{O}\tilde{~}\left(n^3 \log \|A\|\right)$ heur.
- [Storjohann & Labahn 96]: $\mathcal{O}\tilde{~}\left(n^{\omega+1} \log \|A\|\right)$

Similarity over a field: Frobenius normal form

- [Storjohann00]: $\mathcal{O}\tilde{~}(n^\omega)$
- [P. & Storjohann07]: Las Vegas without U $\mathcal{O}(n^\omega)$

Complexities

Matrix multiplication: door to fast linear algebra

- over a field: $\mathcal{O}(n^\omega)$. $\omega \in]2.3727, 3]$ (exponent of linear algebra)
- over $\mathbb{Z}$: $\mathcal{O}(n^\omega M(\log \|A\|)) = \mathcal{O}^\sim(n^\omega \log \|A\|)$

Equivalence over a field: Reduced echelon form

- Gauss-Jordan: $\mathcal{O}(n^\omega)$

Equivalence over $\mathbb{Z}$: Hermite normal form

- [Kannan & Bachem 79]: $\in P$
- [Chou & Collins 82]: $\mathcal{O}^\sim\left(n^6 \log \|A\|\right)$
- [Domich & Al. 87], [Illiopoulos 89]: $\mathcal{O}^\sim\left(n^4 \log \|A\|\right)$
- [Micciancio & Warinschi01]: $\mathcal{O}^\sim\left(n^5 \log \|A\|^2\right)$, $\mathcal{O}^\sim\left(n^3 \log \|A\|\right)$ heur.
- [Storjohann & Labahn 96]: $\mathcal{O}^\sim\left(n^{\omega+1} \log \|A\|\right)$

Similarity over a field: Frobenius normal form

- [Storjohann00]: $\mathcal{O}^\sim(n^\omega)$
- [P. & Storjohann07]: Las Vegas without U $\mathcal{O}(n^\omega)$

Motivation

Algorithmic building blocks in the design of efficient computation of exact linear algebra normal forms.

In brief

Reductions to a building block

Matrix Mult: block rec. $\sum_{i=1}^{\log n} n \left(\frac{n}{2^i}\right)^{\omega-1} = \mathcal{O}\left(n^\omega\right)$ (Gauss, REF)

Matrix Mult: Iterative $\sum_{k=1}^{n} n \left(\frac{n}{k}\right)^{\omega-1} = \mathcal{O}\left(n^\omega\right)$ (Frobenius)

Linear Sys: over $\mathbb{Z}$ (Hermite Normal Form)

Size/dimension compromises

- Hermite normal form : rank 1 updates reducing the determinant
- Frobenius normal form : degree k, dimension n/k for $k = 1 \ldots n$

Motivation

Algorithmic building blocks in the design of efficient computation of exact linear algebra normal forms.

In brief

Reductions to a building block

Matrix Mult: block rec. $\sum_{i=1}^{\log n} n \left(\frac{n}{2^i}\right)^{\omega-1} = \mathcal{O}(n^\omega)$ (Gauss, REF)

Matrix Mult: Iterative $\sum_{k=1}^{n} n \left(\frac{n}{k}\right)^{\omega-1} = \mathcal{O}(n^\omega)$ (Frobenius)

Linear Sys: over $\mathbb{Z}$ (Hermite Normal Form)

Size/dimension compromises

- Hermite normal form : rank 1 updates reducing the determinant
- Frobenius normal form : degree k, dimension n/k for $k = 1 \ldots n$

Study originating from, the design of the libraries

- `FFLAS-FFPACK`: dense over word size $\mathbb{Z}_p$.
- `M4RI`: dense over GF(2) (see Martin Albrecht's Talk)
- `LinBox`: dense/sparse/black-box over $\mathbb{Z}$, $\mathbb{Z}_p$

Outline

Outline

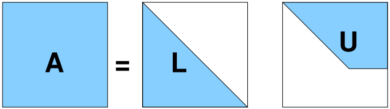
- Reduced Echelon forms and Gaussian elimination
  - Gaussian elimination based matrix decompositions
  - Relations between decompositions
  - Algorithms
- Hermite normal form
  - Micciancio & Warinschi algorithm
  - Double Determinant
  - AddCol
- Frobenius normal form
  - Krylov method
  - Algorithm
  - Reduction to matrix multiplication

Reduced echelon form and Gaussian elimination

Gaussian elimination $<$ Reduction to red. Echelon form

- Extensively studied for numerical computations
- Specificities of exact computations:
  - No partial/full pivoting
  - Rank profile matters
- size of coefficients (e.g. compressed in GF(2))
  $\Rightarrow$asymmetry

LU decomposition

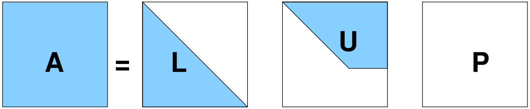


- L unit lower triangular,
- U non-sing upper triangular

Exists for

- matrices having the generic rank profile (every leading principal minor is non zero)

LUP, PLU decomposition

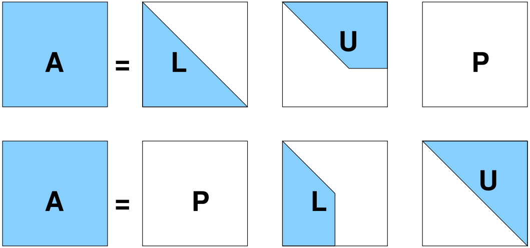


- P a permutation matrix

Exists for

- Any non-singular matrix
- Or any matrix with generic row rank profile

LUP, PLU decomposition

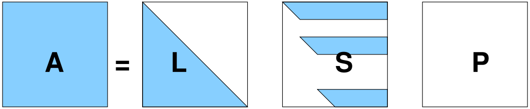


- P a permutation matrix

Exists for

- Any non-singular matrix
- Or any matrix with generic row rank profile

LSP, LQUP, PLUQ decompositions

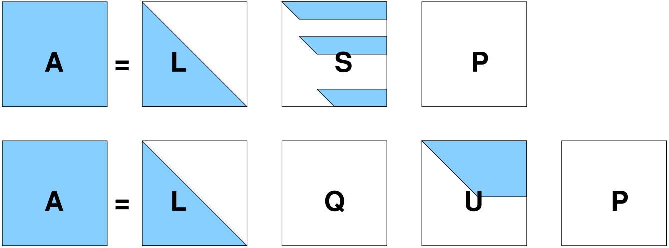


- S: semi-upper triangular,
- Q permutation matrix

Exists for

- any $m \times n$ matrix

LSP, LQUP, PLUQ decompositions

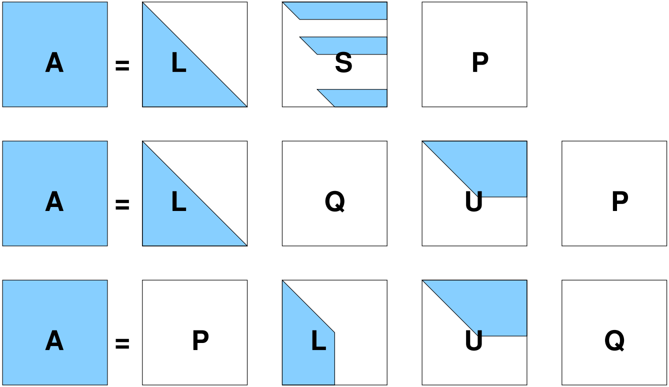


- S: semi-upper triangular,
- Q permutation matrix

Exists for

- any $m \times n$ matrix

LSP, LQUP, PLUQ decompositions



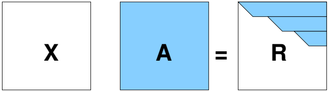
- S: semi-upper triangular,
- Q permutation matrix

Exists for

- any $m \times n$ matrix

Echelon form decomposition

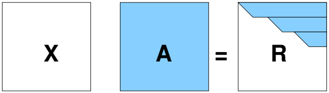
Row Echelon Form $XA = R$



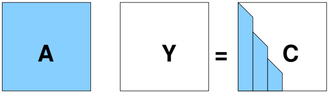
- X, Y: non-singular transformation matrices
- R, C: matrices in row/col echelon form

Echelon form decomposition

Row Echelon Form $XA = R$



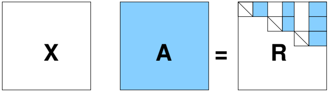
Column Echelon Form $AY = C$



- X, Y: non-singular transformation matrices
- R, C: matrices in row/col echelon form

Reduced echelon form decomposition

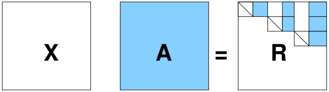
Row Reduced Echelon Form $XA = R$



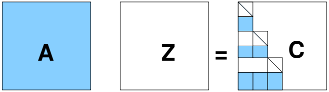
- X, Y: non-singular transformation matrices
- R, C: matrices in reduced row/col echelon form

Reduced echelon form decomposition

Row Reduced Echelon Form $XA = R$

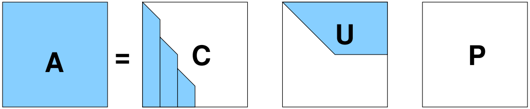


Column Reduced Echelon Form $AY = C$



- X, Y: non-singular transformation matrices
- R, C: matrices in reduced row/col echelon form

CUP and PLE decompositions

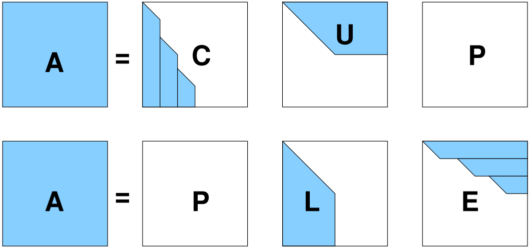


- C: column echelon form
- E: row echelon form

Exists for

- any $m \times n$ matrix

CUP and PLE decompositions



- C: column echelon form
- E: row echelon form

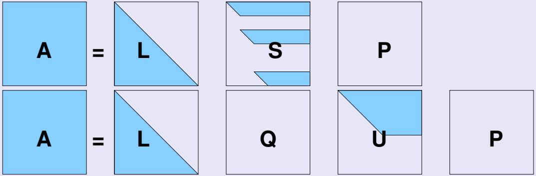
Exists for

- any $m \times n$ matrix

Relations: up to permutations

From LSP to LQUP

$S = QU$



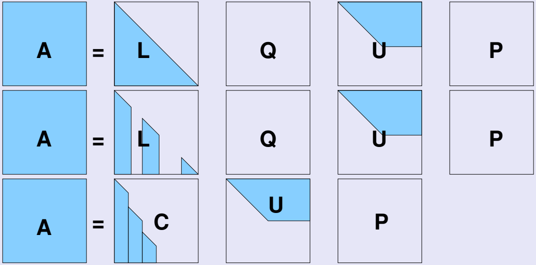
Fact

The first $r = \text{rank}(A)$ values of the permutation Q are monotonically increasing.

Relations: up to permutations

From LQUP to CUP

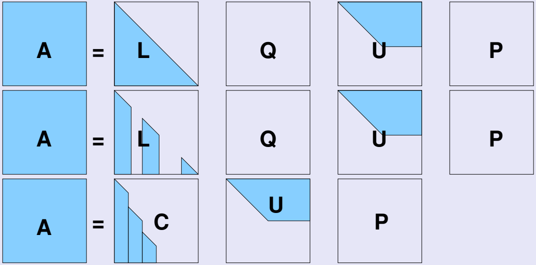
$C = LQ$



Relations: up to permutations

From LQUP to CUP

$$C = LQ$$



From LQUP to PLE

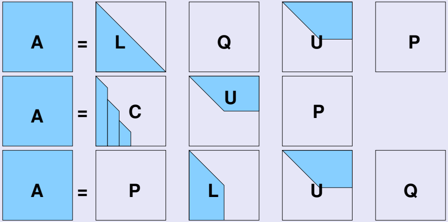
Using transposition:

$$PLE(A^T) = CUP(A)^T$$

Relations:

From LQUP to PLUQ

$$P \leftrightarrow Q, L \leftarrow = Q^T L Q$$



Algorithms: main types

Three ways to group operations:

1. simple iterative
   - Apply the standard Gaussian elimination in dimension n
   - Main loop **for** i=1 to n
2. block algorithms

   2.1 block iterative (Tile)
   - Apply Gaussian elimination in dimension n/k over blocks of size k
   - Main loop: **for** i=1 to n/k

   2.2 block recursive
   - Apply Gaussian elimination in dimension 2 recursively on blocks of size $n/2^i$
   - Main loop: **for** i=1 to 2

Type of algorithms

Data locality: prefer block algorithms

- cache aware: block iterative
- cache oblivious: block recursive

Base case efficieny: simple iterative

Asymptotic time complexity: block recursive

Parallelization: block iterative

Block recursive gaussian elimination

Author	Year	Computation	Requirement
Strassen	69	Inverse	gen. rank prof.
Bunch, Hopcroft	74	LUP	gen. row rank p
Ibarra, Moran, Hui	82	LSP, LQUP	none
Schönage, Keller-Gerig	85	StepForm	none
Storjohann	00	Echelon, RedEch	none
here	11	CUP,PLE,PLUQ	none

Block recursive gaussian elimination

Author	Year	Computation	Requirement
Strassen	69	Inverse	gen. rank prof.
Bunch, Hopcroft	74	LUP	gen. row rank p
Ibarra, Moran, Hui	82	LSP, LQUP	none
Schönage, Keller-Gerig	85	StepForm	none
Storjohann	00	Echelon, RedEch	none
here	11	CUP,PLE,PLUQ	none

Comparison according to:

- No requirement on the input matrix

Block recursive gaussian elimination

Author	Year	Computation	Requirement
Strassen	69	Inverse	gen. rank prof.
Bunch, Hopcroft	74	LUP	gen. row rank p
Ibarra, Moran, Hui	82	LSP, LQUP	none
Schönage, Keller-Gerig	85	StepForm	none
Storjohann	00	Echelon, RedEch	none
here	11	CUP,PLE,PLUQ	none

Comparison according to:

- No requirement on the input matrix
- Rank sensitive complexity

Block recursive gaussian elimination

Author	Year	Computation	Requirement
Strassen	69	Inverse	gen. rank prof.
Bunch, Hopcroft	74	LUP	gen. row rank p
Ibarra, Moran, Hui	82	LSP, LQUP	none
Schönage, Keller-Gerig	85	StepForm	none
Storjohann	00	Echelon, RedEch	none
here	11	CUP,PLE,PLUQ	none

Comparison according to:

- No requirement on the input matrix
- Rank sensitive complexity
- Memory allocations
- Constant factor in the time complexity

Memory requirements:

Definition

In place = *output overrides the input and computation does not need extra memory (considering Matrix multiplication $C \leftarrow C + AB$ as a black box)*

Remark: a unit lower triangular and an upper triangular matrix can be stored on the same $m \times n$ storage!

Preliminaries

TRSM: TRiangular Solve with Matrix

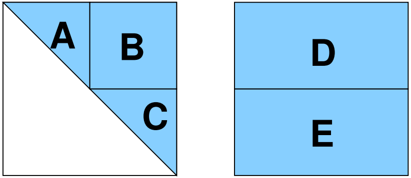
$$\begin{bmatrix} A & B \\ & C \end{bmatrix}^{-1} \begin{bmatrix} D \\ E \end{bmatrix} = \begin{bmatrix} A^{-1} & \\ & I \end{bmatrix} \begin{bmatrix} I & -B \\ & I \end{bmatrix} \begin{bmatrix} I & \\ & C^{-}1 \end{bmatrix} \begin{bmatrix} D \\ E \end{bmatrix}$$

Preliminaries

TRSM: TRiangular Solve with Matrix

$$\begin{bmatrix} A & B \\ & C \end{bmatrix}^{-1} \begin{bmatrix} D \\ E \end{bmatrix} = \begin{bmatrix} A^{-1} & \\ & I \end{bmatrix} \begin{bmatrix} I & -B \\ & I \end{bmatrix} \begin{bmatrix} I & \\ & C^{-1} \end{bmatrix} \begin{bmatrix} D \\ E \end{bmatrix}$$

Compute $F = C^{-1}E$ (Recursive call)
Compute $G = D - BF$ (MM)
Compute $H = A^{-1}G$ (Recursive call)
Return $\begin{bmatrix} H \\ F \end{bmatrix}$



Preliminaries

TRSM: TRiangular Solve with Matrix

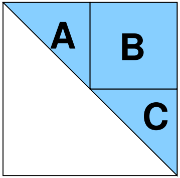
$$\begin{bmatrix} A & B \\ & C \end{bmatrix}^{-1} \begin{bmatrix} D \\ E \end{bmatrix} = \begin{bmatrix} A^{-1} & \\ & I \end{bmatrix} \begin{bmatrix} I & -B \\ & I \end{bmatrix} \begin{bmatrix} I & \\ & C^{-1} \end{bmatrix} \begin{bmatrix} D \\ E \end{bmatrix}$$

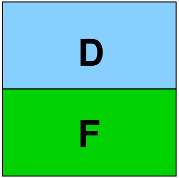
Compute $F = C^{-1}E$ (Recursive call)

Compute $G = D - BF$ (MM)

Compute $H = A^{-1}G$ (Recursive call)

Return $\begin{bmatrix} H \\ F \end{bmatrix}$



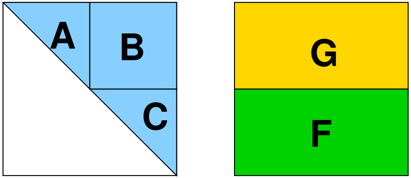


Preliminaries

TRSM: TRiangular Solve with Matrix

$$\begin{bmatrix} A & B \\ & C \end{bmatrix}^{-1} \begin{bmatrix} D \\ E \end{bmatrix} = \begin{bmatrix} A^{-1} & \\ & I \end{bmatrix} \begin{bmatrix} I & -B \\ & I \end{bmatrix} \begin{bmatrix} I & \\ & C^{-1} \end{bmatrix} \begin{bmatrix} D \\ E \end{bmatrix}$$

Compute $F = C^{-1}E$ (Recursive call)
Compute $G = D - BF$ (MM)
Compute $H = A^{-1}G$ (Recursive call)
Return $\begin{bmatrix} H \\ F \end{bmatrix}$



Preliminaries

TRSM: TRiangular Solve with Matrix

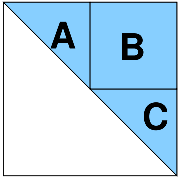
$$\begin{bmatrix} A & B \\ & C \end{bmatrix}^{-1} \begin{bmatrix} D \\ E \end{bmatrix} = \begin{bmatrix} A^{-1} & \\ & I \end{bmatrix} \begin{bmatrix} I & -B \\ & I \end{bmatrix} \begin{bmatrix} I & \\ & C^{-}1 \end{bmatrix} \begin{bmatrix} D \\ E \end{bmatrix}$$

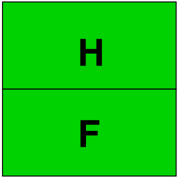
Compute $F = C^{-1}E$ (Recursive call)

Compute $G = D - BF$ (MM)

Compute $H = A^{-1}G$ (Recursive call)

Return $\begin{bmatrix} H \\ F \end{bmatrix}$





Preliminaries

TRSM: TRiangular Solve with Matrix

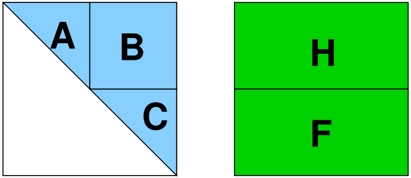
$$\begin{bmatrix} A & B \\ & C \end{bmatrix}^{-1} \begin{bmatrix} D \\ E \end{bmatrix} = \begin{bmatrix} A^{-1} & \\ & I \end{bmatrix} \begin{bmatrix} I & -B \\ & I \end{bmatrix} \begin{bmatrix} I & \\ & C^{-1} \end{bmatrix} \begin{bmatrix} D \\ E \end{bmatrix}$$

Compute $F = C^{-1}E$ (Recursive call)

Compute $G = D - BF$ (MM)

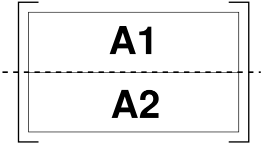
Compute $H = A^{-1}G$ (Recursive call)

Return $\begin{bmatrix} H \\ F \end{bmatrix}$



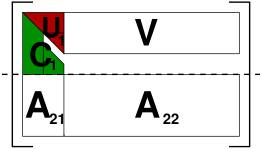
- $\mathcal{O}\left(n^{\omega}\right)$
- In place

The CUP decomposition



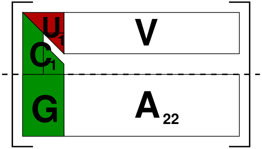
1. Split A Row-wise

The CUP decomposition



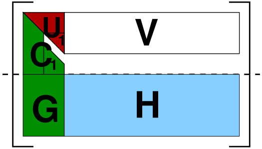
1. Split A Row-wise
2. Recursive call on A_1

The CUP decomposition



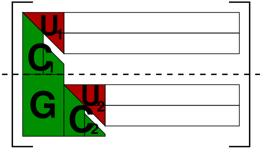
1. Split A Row-wise
2. Recursive call on A_1
3. $G \leftarrow A_{21}U_1^{-1}$ (`trsm`)

The CUP decomposition



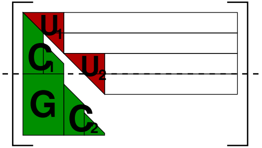
1. Split A Row-wise
2. Recursive call on A_1
3. $G \leftarrow A_{21} U_1^{-1}$ (`trsm`)
4. $H \leftarrow A_{22} - G \times V$ (`MM`)

The CUP decomposition



1. Split A Row-wise
2. Recursive call on A_1
3. $G \leftarrow A_{21} U_1^{-1}$ (`trsm`)
4. $H \leftarrow A_{22} - G \times V$ (`MM`)
5. Recursive call on H

The CUP decomposition



1. Split A Row-wise
2. Recursive call on A_1
3. $G \leftarrow A_{21}U_1^{-1}$ (`trsm`)
4. $H \leftarrow A_{22} - G \times V$ (`MM`)
5. Recursive call on H
6. Row permutations

Memory: LSP vs LQUP vs PLUQ vs CUP

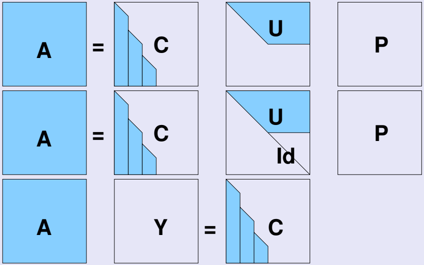
Decomposition	In place
LSP	N
LQUP	N
PLUQ	Y
CUP	Y

Echelon forms

From CUP to ColumnEchelon form

$$Y = P^T \begin{bmatrix} U_1 & U_2 \\ & I_{n-r} \end{bmatrix}^{-1}$$

$$= P^T \begin{bmatrix} U_1^{-1} & -U_1^{-1}U_2 \\ & I_{n-r} \end{bmatrix}$$



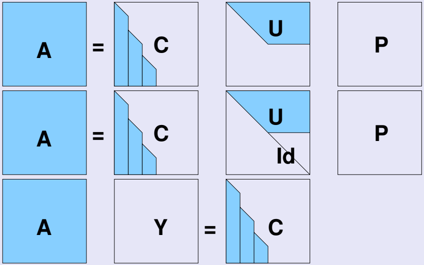
Additional operations:

$-U^{-1}U_2$ `trsm` (triangular system solve) in-place

Echelon forms

From CUP to ColumnEchelon form

$$
\begin{aligned}
Y &= P^T \begin{bmatrix} U_1 & U_2 \\ & I_{n-r} \end{bmatrix}^{-1} \\
&= P^T \begin{bmatrix} U_1^{-1} & -U_1^{-1}U_2 \\ & I_{n-r} \end{bmatrix}
\end{aligned}
$$



Additional operations:

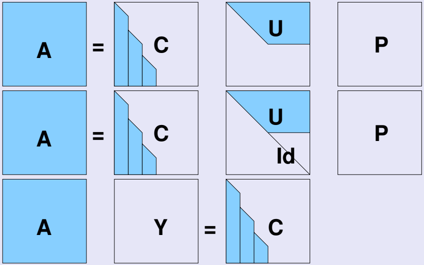
$-U^{-1}U_2$ `trsm` (triangular system solve) in-place

U_1^{-1}: `trtri` (triangular inverse)

Echelon forms

From CUP to ColumnEchelon form

$$
\begin{aligned}
Y &= P^T \begin{bmatrix} U_1 & U_2 \\ & I_{n-r} \end{bmatrix}^{-1} \\
&= P^T \begin{bmatrix} U_1^{-1} & -U_1^{-1}U_2 \\ & I_{n-r} \end{bmatrix}
\end{aligned}
$$



Additional operations:

- $-U^{-1}U_2$ `trsm` (triangular system solve) in-place
- U_1^{-1}: `trtri` (triangular inverse) in-place

From CUP to Column Echelon form

TRTRI: triangular inverse

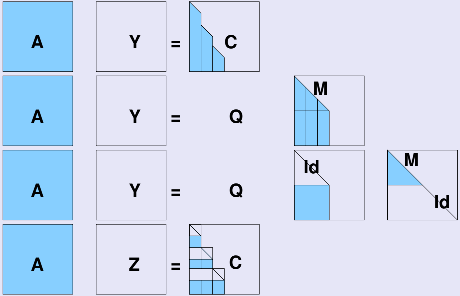
$$\begin{bmatrix} U_1 & U_2 \\ & U_3 \end{bmatrix}^{-1} = \begin{bmatrix} U_1^{-1} & -U_1^{-1} U_2 U_3^{-1} \\ & U_3^{-1} \end{bmatrix}$$

if $n = 1$ **then**
    $U \leftarrow U^{-1}$
else
    $U_2 \leftarrow U_3^{-1} U_2$ TRSM
    $U_2 \leftarrow -U_2 U_3^{-1}$ TRSM
    $U_1 \leftarrow U_1^{-1}$ TRTRI
    $U_3 \leftarrow U_3^{-1}$ TRTRI
end if

Reduced Echelon forms

From Col. Echelon form to Reduced Col. Echelon form

$$Z = Y \begin{bmatrix} M & \\ & I_{n-r} \end{bmatrix}^{-1}$$



Similarly, from PLE to RowEchelon form

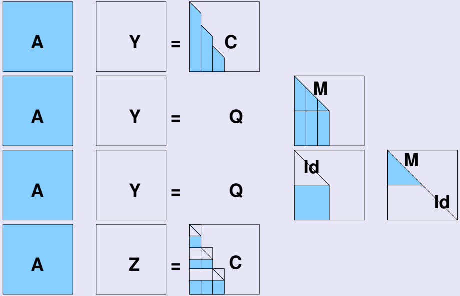
Again reduces to:

- $U^{-1}X$: `TRSM`, in-place
- U^{-1}: `TRTRI`, in-place

Reduced Echelon forms

From Col. Echelon form to Reduced Col. Echelon form

$$Z = Y \begin{bmatrix} M & \\ & I_{n-r} \end{bmatrix}^{-1}$$



Similarly, from PLE to RowEchelon form

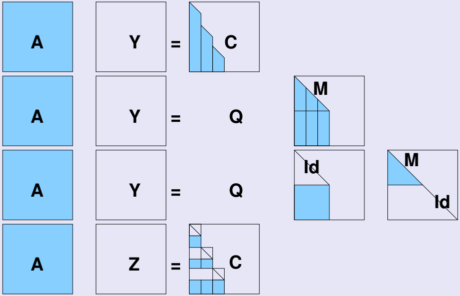
Again reduces to:

- $U^{-1}X$: TRSM, in-place
- U^{-1}: TRTRI, in-place
- UL: TRTRM,

Reduced Echelon forms

From Col. Echelon form to Reduced Col. Echelon form

$$Z = Y \begin{bmatrix} M & \\ & I_{n-r} \end{bmatrix}^{-1}$$



Similarly, from PLE to RowEchelon form

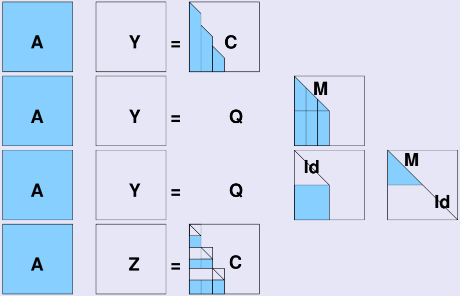
Again reduces to:

- $U^{-1}X$: TRSM, in-place
- U^{-1}: TRTRI, in-place
- UL: TRTRM,

Reduced Echelon forms

From Col. Echelon form to Reduced Col. Echelon form

$$Z = Y \begin{bmatrix} M & \\ & I_{n-r} \end{bmatrix}^{-1}$$



Similarly, from PLE to RowEchelon form

Again reduces to:

- $U^{-1}X$: TRSM, in-place
- U^{-1}: TRTRI, in-place
- UL: TRTRM, in-place

From Echelon to Reduced Echelon

TRTRM: triangular triangular multiplication

$$\begin{bmatrix} U_1 & U_2 \\ & U_3 \end{bmatrix} \begin{bmatrix} L_1 & \\ L_2 & L_3 \end{bmatrix} = \begin{bmatrix} U_1L_1 + U_2L_2 & U_2L_3 \\ U_3L_2 & U_3L_3 \end{bmatrix}$$

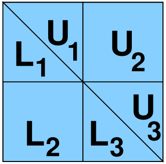
1: $X_1 \leftarrow U_1L_1$ TRTRM
2: $X_1 \leftarrow X_1 + U_2L_2$ MM
3: $X_2 \leftarrow U_2L_3$ TRMM
4: $X_3 \leftarrow U_3L_2$ TRMM
5: $X_4 \leftarrow U_3L_3$ TRTRM

From Echelon to Reduced Echelon

TRTRM: triangular triangular multiplication

$$\begin{bmatrix} U_1 & U_2 \\ & U_3 \end{bmatrix} \begin{bmatrix} L_1 & \\ L_2 & L_3 \end{bmatrix} = \begin{bmatrix} U_1L_1 + U_2L_2 & U_2L_3 \\ U_3L_2 & U_3L_3 \end{bmatrix}$$

1: $X_1 \leftarrow U_1L_1$ TRTRM
2: $X_1 \leftarrow X_1 + U_2L_2$ MM
3: $X_2 \leftarrow U_2L_3$ TRMM
4: $X_3 \leftarrow U_3L_2$ TRMM
5: $X_4 \leftarrow U_3L_3$ TRTRM

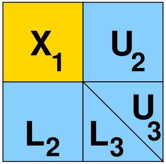


From Echelon to Reduced Echelon

TRTRM: triangular triangular multiplication

$$\begin{bmatrix} U_1 & U_2 \\ & U_3 \end{bmatrix} \begin{bmatrix} L_1 & \\ L_2 & L_3 \end{bmatrix} = \begin{bmatrix} U_1L_1 + U_2L_2 & U_2L_3 \\ U_3L_2 & U_3L_3 \end{bmatrix}$$

1: $X_1 \leftarrow U_1L_1$ TRTRM
2: $X_1 \leftarrow X_1 + U_2L_2$ MM
3: $X_2 \leftarrow U_2L_3$ TRMM
4: $X_3 \leftarrow U_3L_2$ TRMM
5: $X_4 \leftarrow U_3L_3$ TRTRM

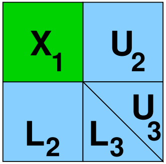


From Echelon to Reduced Echelon

TRTRM: triangular triangular multiplication

$$\begin{bmatrix} U_1 & U_2 \\ & U_3 \end{bmatrix} \begin{bmatrix} L_1 & \\ L_2 & L_3 \end{bmatrix} = \begin{bmatrix} U_1L_1 + U_2L_2 & U_2L_3 \\ U_3L_2 & U_3L_3 \end{bmatrix}$$

1: $X_1 \leftarrow U_1L_1$ `TRTRM`
2: $X_1 \leftarrow X_1 + U_2L_2$ `MM`
3: $X_2 \leftarrow U_2L_3$ `TRMM`
4: $X_3 \leftarrow U_3L_2$ `TRMM`
5: $X_4 \leftarrow U_3L_3$ `TRTRM`



From Echelon to Reduced Echelon

`TRTRM`: triangular triangular multiplication

$$\begin{bmatrix} U_1 & U_2 \\ & U_3 \end{bmatrix} \begin{bmatrix} L_1 & \\ L_2 & L_3 \end{bmatrix} = \begin{bmatrix} U_1L_1 + U_2L_2 & U_2L_3 \\ U_3L_2 & U_3L_3 \end{bmatrix}$$

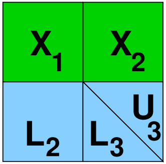
1: $X_1 \leftarrow U_1L_1$ `TRTRM`

2: $X_1 \leftarrow X_1 + U_2L_2$ `MM`

3: $X_2 \leftarrow U_2L_3$ `TRMM`

4: $X_3 \leftarrow U_3L_2$ `TRMM`

5: $X_4 \leftarrow U_3L_3$ `TRTRM`

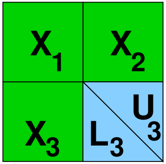


From Echelon to Reduced Echelon

TRTRM: triangular triangular multiplication

$$\begin{bmatrix} U_1 & U_2 \\ & U_3 \end{bmatrix} \begin{bmatrix} L_1 & \\ L_2 & L_3 \end{bmatrix} = \begin{bmatrix} U_1 L_1 + U_2 L_2 & U_2 L_3 \\ U_3 L_2 & U_3 L_3 \end{bmatrix}$$

1: $X_1 \leftarrow U_1 L_1$ `TRTRM`
2: $X_1 \leftarrow X_1 + U_2 L_2$ `MM`
3: $X_2 \leftarrow U_2 L_3$ `TRMM`
4: $X_3 \leftarrow U_3 L_2$ `TRMM`
5: $X_4 \leftarrow U_3 L_3$ `TRTRM`



From Echelon to Reduced Echelon

TRTRM: triangular triangular multiplication

$$\begin{bmatrix} U_1 & U_2 \\ & U_3 \end{bmatrix} \begin{bmatrix} L_1 & \\ L_2 & L_3 \end{bmatrix} = \begin{bmatrix} U_1 L_1 + U_2 L_2 & U_2 L_3 \\ U_3 L_2 & U_3 L_3 \end{bmatrix}$$

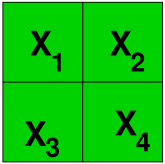
1: $X_1 \leftarrow U_1 L_1$ TRTRM

2: $X_1 \leftarrow X_1 + U_2 L_2$ MM

3: $X_2 \leftarrow U_2 L_3$ TRMM

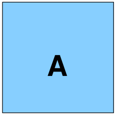
4: $X_3 \leftarrow U_3 L_2$ TRMM

5: $X_4 \leftarrow U_3 L_3$ TRTRM

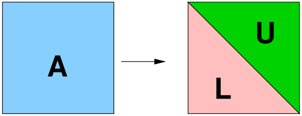


- $\mathcal{O}\left(n^\omega\right)$
- In place

Example: in place matrix inversion



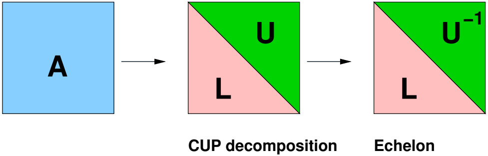
Example: in place matrix inversion



CUP decomposition

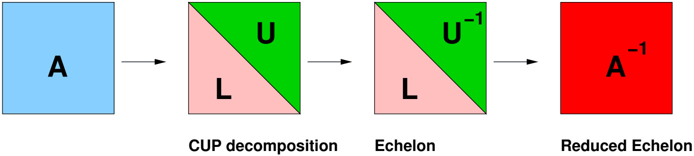
$$A = LU$$

Example: in place matrix inversion



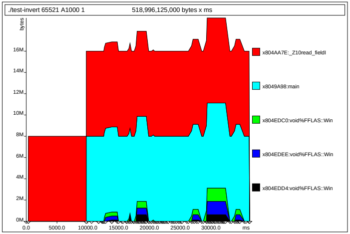
$$AU^{-1} = L$$

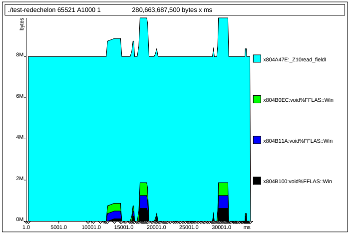
Example: in place matrix inversion



$$A(U^{-1}L^{-1}) = I$$

Experiments





Direct computation of the Reduced Echelon form

- Strassen 69: inverse of generic matrices
- Storjohann 00: `Gauss-Jordan` generalization for any rank profile

Matrix Inversion [Strassen 69]

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & \\ & I \end{bmatrix} \begin{bmatrix} I & -B \\ & I \end{bmatrix} \begin{bmatrix} I & \\ & (D - CA^{-1}B)^{-1} \end{bmatrix} \begin{bmatrix} I & \\ CA^{-1} & I \end{bmatrix}$$

Direct computation of the Reduced Echelon form

- Strassen 69: inverse of generic matrices
- Storjohann 00: `Gauss-Jordan` generalization for any rank profile

Matrix Inversion [Strassen 69]

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & \\ & I \end{bmatrix} \begin{bmatrix} I & -B \\ & I \end{bmatrix} \begin{bmatrix} I & \\ & (D - CA^{-1}B)^{-1} \end{bmatrix} \begin{bmatrix} I & \\ CA^{-1} & I \end{bmatrix}$$

1: Compute $E = A^{-1}$ (Recursive call)
2: Compute $F = D - CEB$ (MM)
3: Compute $G = F^{-1}$ (Recursive call)
4: Compute $H = -EB$ (MM)
5: Compute $J = HG$ (MM)
6: Compute $K = CE$ (MM)
7: Compute $L = E + JK$ (MM)
8: Compute $M = GK$ (MM)
9: Return $\begin{bmatrix} E & J \\ M & G \end{bmatrix}$

Strassen-Storjohann's `Gauss-Jordan` elimination

Problem

Needs to perform operations of the form $A \leftarrow AB$

$\Rightarrow$ *not doable in place by a usual matrix multiplication algorithm*

Strassen-Storjohann's `Gauss-Jordan` elimination

Problem

Needs to perform operations of the form $A \leftarrow AB$
$\Rightarrow$ *not doable in place by a usual matrix multiplication algorithm*

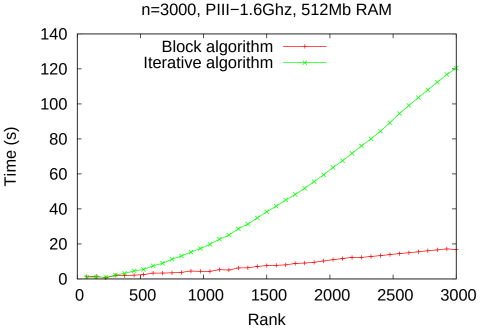
Workaround [Storjohann]:

1. Decompose $B = LU$ `LU`
2. $A \leftarrow AL$ `trmm`
3. $A \leftarrow AU$ `trmm`

Rank sensitive time complexity

Fact

Algorithms LSP, CUP, LQUP, PLUQ, ... have a rank sensitive computation time: $\mathcal{O}\left(mnr^{\omega-2}\right)$



Time complexity: comparing constants

$$\mathcal{O}(n^\omega) = C_\omega n^3$$

Algorithm	Constant C_ω	C_3	$C_{\log_2 7}$	in-place
`MM`	C_ω	2	6	$\times$
`TRSM`	$\frac{C_\omega}{2^{\omega-1}-2}$	1	4	$\vee$
`TRTRI`	$\frac{C_\omega}{(2^{\omega-1}-2)(2^{\omega-1}-1)}$	$\frac{1}{3} \approx 0.33$	$\frac{8}{5} = 1.6$	$\vee$
`TRTRM`, `CUP` `PLUQ` `LQUP`,	$\frac{C_\omega}{2^{\omega-1}-2} - \frac{C_\omega}{2^\omega-2}$	$\frac{2}{3} \approx 0.66$	$\frac{14}{5} = 2.8$	$\vee$
`Echelon`	$\frac{C_\omega}{2^{\omega-2}-1} - \frac{3C_\omega}{2^\omega-2}$	1	$\frac{22}{5} \approx 4.4$	$\vee$
`RedEchelon`	$\frac{C_\omega(2^{\omega-1}+2)}{(2^{\omega-1}-2)(2^{\omega-1}-1)}$	2	$\frac{44}{5} = 8.8$	$\vee$
`StepForm`	$\frac{5C_\omega}{2^{\omega-1}-1} + \frac{C_\omega}{(2^{\omega-1}-1)(2^{\omega-2}-1)}$	4	$\frac{76}{5} = 15.2$	$\times$
`GJ`*	$\frac{C_\omega}{2^{\omega-2}-1}$	2	8	$\times$

*: `GJ`: `GaussJordan` alg of [Storjohann00] computing the reduced echelon form

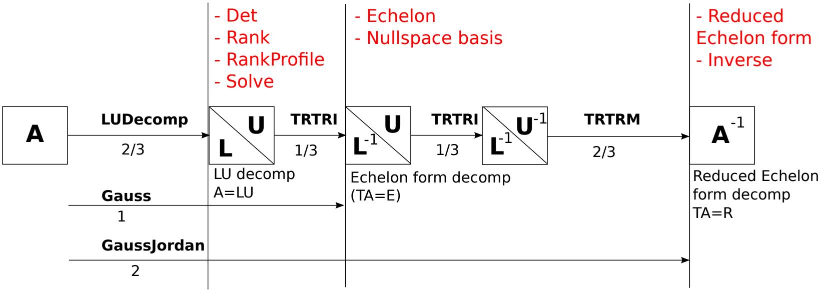
Applications to standard linalg problems

Problem	Using	C_ω	C_3	$C_{\log_2 7}$	In place

`Rank`
`RankProfile`
`IsSingular`
`Det`
`Solve`
	`GJ`
	`CUP`
		$\frac{C_\omega}{2^{\omega-2}-1}$
		$\frac{C_\omega}{2^{\omega-1}-2} - \frac{C_\omega}{2^{\omega}-2}$
			2
			0.66
				8
				2.8
					×
					∨

`Inverse`	`GJ` `CUP`	$\frac{C_\omega}{2^{\omega-2}-1}$ $\frac{C_\omega(2^{\omega-1}+2)}{(2^{\omega-1}-2)(2^{\omega-1}-1)}$	2 2	8 8.8	× ∨

Summary


- Det
- Rank
- RankProfile
- Solve
- Echelon
- Nullspace basis
- Reduced
Echelon form
- Inverse
A
LUDecomp
2/3
L U
TRTRI
1/3
L-1 U
TRTRI
1/3
L-1 U-1
TRTRM
2/3
A-1
LU decomp
A=LU
Echelon form decomp
(TA=E)
Reduced Echelon
form decomp
TA=R
Gauss
1
GaussJordan
2

Outline

Computing Hermite Normal form

Equivalence over a ring: $H = UA$, where $\det(U) = \pm 1$

Hermite normal form: $H = \begin{bmatrix} p_1 & * & x_{1,2} & * & * & x_{1,3} & * \\ & & p_2 & * & * & x_{2,3} & * \\ & & & & & p_3 & * \\ & & & & & & \end{bmatrix}$, with $0 \le x_{*,j} < p_j$

Reduced Echelon form over a Ring

Improving Micciancio-Warinshi algorihm

- $\mathcal{O}\tilde{}\left(n^5 \log \|A\|\right)$ (heuristically: $\mathcal{O}\tilde{}\left(n^3 \log \|A\|\right)$)
- space: $\mathcal{O}\left(n^2 \log \|A\|\right)$

$\Rightarrow$Good on random matrices, common in num. theory, crypto.

Computing Hermite Normal form

Equivalence over a ring: $H = UA$, where $\det(U) = \pm 1$

Hermite normal form: $H = \begin{bmatrix} p_1 & * & x_{1,2} & * & * & x_{1,3} & * \\ & & p_2 & * & * & x_{2,3} & * \\ & & & & & p_3 & * \\ & & & & & & \end{bmatrix}$, with $0 \leq x_{*,j} < p_j$

Reduced Echelon form over a Ring

Improving Micciancio-Warinshi algorihm

- $\mathcal{O}\tilde{}\left(n^5 \log \|A\|\right)$ (heuristically: $\mathcal{O}\tilde{}\left(n^3 \log \|A\|\right)$)
- space: $\mathcal{O}\left(n^2 \log \|A\|\right)$

$\Rightarrow$Good on random matrices, common in num. theory, crypto.
Implementation, reduction to building blocks:

- LinSys over $\mathbb{Z}$,
- CUP and MatMul over $\mathbb{Z}_p$

Naive algorithm

begin

  foreach i **do**

    $(g, t_i, \ldots, t_n) = \text{xgcd}(A_{i,i}, A_{i+1,i}, \ldots, A_{n,i})$;

    $L_i \leftarrow \sum_{j=i+1}^{n} t_j L_j$;

    for $j = i + 1 \ldots n$ **do**

      $L_j \leftarrow L_j - \frac{A_{j,i}}{g} L_i$; /* eliminate */

    for $j = 1 \ldots i - 1$ **do**

      $L_j \leftarrow L_j - \lfloor \frac{A_{j,i}}{g} \rfloor L_i$; /* reduce */

end

$$\begin{bmatrix} p_1 & * & x_{1,2} & * & * & x_{1,3} & * \\ & & p_2 & * & * & x_{2,3} & * \\ & & & & & p_3 & * \\ & & & & & & \end{bmatrix}$$

Computing modulo the determinant [Domich & Al. 87]

Property

For A *non-singular:* $\max_i \sum_j H_{ij} \leq \det H$

Example

$$A = \begin{bmatrix} -5 & 8 & -3 & -9 & 5 & 5 \\ -2 & 8 & -2 & -2 & 8 & 5 \\ 7 & -5 & -8 & 4 & 3 & -4 \\ 1 & -1 & 6 & 0 & 8 & -3 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 3 & 237 & -299 & 90 \\ 0 & 1 & 1 & 103 & -130 & 40 \\ 0 & 0 & 4 & 352 & -450 & 135 \\ 0 & 0 & 0 & 486 & -627 & 188 \end{bmatrix}$$

$$\det A = 1944$$

Computing modulo the determinant [Domich & Al. 87]

Property

For A non-singular: $\max_i \sum_j H_{ij} \leq \det H$

Example

$$A = \begin{bmatrix} -5 & 8 & -3 & -9 & 5 & 5 \\ -2 & 8 & -2 & -2 & 8 & 5 \\ 7 & -5 & -8 & 4 & 3 & -4 \\ 1 & -1 & 6 & 0 & 8 & -3 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 3 & 237 & -299 & 90 \\ 0 & 1 & 1 & 103 & -130 & 40 \\ 0 & 0 & 4 & 352 & -450 & 135 \\ 0 & 0 & 0 & 486 & -627 & 188 \end{bmatrix}$$

$$\det A = 1944$$

Moreover, every computation can be done modulo $d = \det A$:

$$U' \begin{bmatrix} A & \\ dI_n & I_n \end{bmatrix} = \begin{bmatrix} H & \\ & I_n \end{bmatrix}$$

$$\Rightarrow \mathcal{O}\left(n^3\right) \times M(n(\log n + \log \|A\|)) = \mathcal{O}\tilde{}\left(n^4 \log \|A\|\right)$$

Computing modulo the determinant

- Pessimistic estimate on the arithmetic size
- d large but most coefficients of H are small
- *On the average* : only the last few columns are *large*

$\Rightarrow$ Compute H' *close* to H but with small determinant

Computing modulo the determinant

- Pessimistic estimate on the arithmetic size
- d large but most coefficients of H are small
- *On the average* : only the last few columns are *large*

⇒Compute H' *close* to H but with small determinant

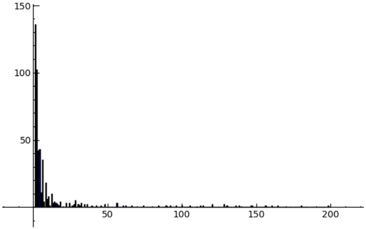
[Micciancio & Warinschi 01]

$$A = \begin{bmatrix} B & b \\ c^T & a_{n-1,n} \\ d^T & a_{n,n} \end{bmatrix}$$

$$d_1 = \det\left(\begin{bmatrix} B \\ c^T \end{bmatrix}\right), d_2 = \det\left(\begin{bmatrix} B \\ d^T \end{bmatrix}\right)$$

$g = \text{gcd}(d_1, d_2) = sd_1 + td_2$ Then

$$\det\left(\begin{bmatrix} B \\ sc^T + td^T \end{bmatrix}\right) = g$$



Micciancio & Warinschi algorithm

begin

Compute $d_1 = \det\left(\begin{bmatrix} B \\ c^T \end{bmatrix}\right), d_2 = \det\left(\begin{bmatrix} B \\ d^T \end{bmatrix}\right)$; /* Double Det */

$(g, s, t) =$ xgcd(d_1, d_2);

Compute H_1 the HNF of $\begin{bmatrix} B \\ sc^T + td^T \end{bmatrix} \mod g$; /* Modular HNF */

Recover H_2 the HNF of $\begin{bmatrix} B & b \\ sc^T + td^T & sa_{n-1,n} + ta_{n,n} \end{bmatrix}$; /* AddCol */

Recover H_3 the HNF of $\begin{bmatrix} B & b \\ c^T & a_{n-1,n} \\ d^T & a_{n,n} \end{bmatrix}$; /* AddRow */

end

Micciancio & Warinschi algorithm

begin

Compute $d_1 = \det\left(\begin{bmatrix} B \\ c^T \end{bmatrix}\right), d_2 = \det\left(\begin{bmatrix} B \\ d^T \end{bmatrix}\right)$; /* Double Det */

$(g, s, t) =$ xgcd(d_1, d_2);

Compute H_1 the HNF of $\begin{bmatrix} B \\ sc^T + td^T \end{bmatrix} \mod g$; /* Modular HNF */

Recover H_2 the HNF of $\begin{bmatrix} B & b \\ sc^T + td^T & sa_{n-1,n} + ta_{n,n} \end{bmatrix}$; /* AddCol */

Recover H_3 the HNF of $\begin{bmatrix} B & b \\ c^T & a_{n-1,n} \\ d^T & a_{n,n} \end{bmatrix}$; /* AddRow */

end

Double Determinant

First approach: LU mod $p_1, \ldots, p_k$ + CRT

- Only one elimination for the $n-2$ first rows
- 2 updates for the last rows (triangular back substitution)
- k *large* such that $\prod_{i=1}^{k} p_i > n^n \log \|A\|^{n/2}$

Double Determinant

First approach: LU mod $p_1, \dots, p_k$ + CRT

- Only one elimination for the $n-2$ first rows
- 2 updates for the last rows (triangular back substitution)
- k *large* such that $\prod_{i=1}^{k} p_i > n^n \log \|A\|^{n/2}$

Second approach: [Abbott Bronstein Mulders 99]

- Solve $Ax = b$.
- $\delta = \text{lcm}(q_1, \dots, q_n)$ s.t. $x_i = p_i/q_i$

Then δ is a *large* divisor of $D = \det A$.

- Compute D/δ by LU mod $p_1, \dots, p_k$ + CRT
- k *small*, such that $\prod_{i=1}^{k} p_i > n^n \log \|A\|^{n/2} / \delta$

Double Determinant : improved

Property

Let $x = [x_1, \ldots, x_n]$ *be the solution of* $\left[\begin{array}{c|c} A & c \end{array}\right] x = d$*. Then* $y = [-\frac{x_1}{x_n}, \ldots, -\frac{x_{n-1}}{x_n}, \frac{1}{x_n}]$ *is the solution of* $\left[\begin{array}{c|c} A & d \end{array}\right] y = c$*.*

- 1 system solve
- 1 LU for each p_i

$\Rightarrow d_1, d_2$ computed at about the cost of 1 déterminant

AddCol

Problem

Find a vector e such that

$$\left[\begin{array}{c|c} H_1 & e \end{array}\right] = U \begin{bmatrix} B & b \\ sc^T + td^T & sa_{n-1,n} + ta_{n,n} \end{bmatrix}$$

$$\begin{aligned} e &= U \begin{bmatrix} b \\ sa_{n-1,n} + ta_{n,n} \end{bmatrix} \\ &= H_1 \begin{bmatrix} B \\ sc^T + td^T \end{bmatrix}^{-1} \begin{bmatrix} b \\ sa_{n-1,n} + ta_{n,n} \end{bmatrix} \end{aligned}$$

⇒Solve a system.

- $n-1$ first rows are *small*
- last row is *large*

AddCol

Idea:

replace the last row by a random *small* one w^T.

$$\begin{bmatrix} B \\ w^T \end{bmatrix} y = \begin{bmatrix} b \\ a_{n-1,n-1} \end{bmatrix}$$

Let k be a basis of the kernel of B. Then

$$x = y + \alpha k.$$

where

$$\alpha = \frac{a_{n-1,n-1} - (sc^T + td^T) \cdot y}{(sc^T + td^T) \cdot k}$$

$\Rightarrow$limits the *expensive* arithmetic to a few dot products

Outline

- Reduced Echelon forms and Gaussian elimination
  - Gaussian elimination based matrix decompositions
  - Relations between decompositions
  - Algorithms
- Hermite normal form
  - Micciancio & Warinschi algorithm
  - Double Determinant
  - AddCol
- Frobenius normal form
  - Krylov method
  - Algorithm
  - Reduction to matrix multiplication

Krylov Method

Definition (degree d Krylov matrix of one vector v)

$$K = \begin{bmatrix} v & Av & \dots & A^{d-1}v \end{bmatrix}$$

Property

$$A \times K = K \times \begin{bmatrix} 0 & & & * \\ 1 & & & * \\ & \ddots & & * \\ & & 1 & * \end{bmatrix}$$

Krylov Method

Definition (degree d Krylov matrix of one vector v)

$$K = \begin{bmatrix} v & Av & \dots & A^{d-1}v \end{bmatrix}$$

Property

$$A \times K = K \times \begin{bmatrix} 0 & & & * \\ 1 & & & * \\ & \ddots & & * \\ & & 1 & * \end{bmatrix}$$

$\Rightarrow$if $d = n$,

$$K^{-1}AK = C_{P^A_{car}}$$

Krylov Method

Definition (degree d Krylov matrix of one vector v)

$$K = \begin{bmatrix} v & Av & \dots & A^{d-1}v \end{bmatrix}$$

Property

$$A \times K = K \times \begin{bmatrix} 0 & & & * \\ 1 & & & * \\ & \ddots & & * \\ & & 1 & * \end{bmatrix}$$

⇒if $d = n$,

$$K^{-1}AK = C_{P^A_{car}}$$

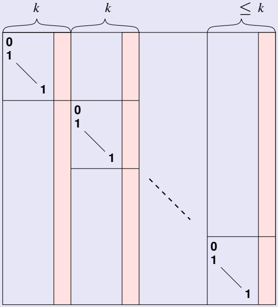
⇒[Keller-Gehrig, alg. 2] computes K in $\mathcal{O}\left(n^\omega \log n\right)$

Definition (degree k Krylov matrix of several vectors v_i)

$$K = \left[\begin{array}{ccc|ccc|c|ccc} v_1 & \dots & A^{k-1}v_1 & v_2 & \dots & A^{k-1}v_2 & \dots & v_l & \dots & A^{k-1}v_l \end{array}\right]$$

Property

$A \times K = K \times$



Hessenberg poly-cyclic form

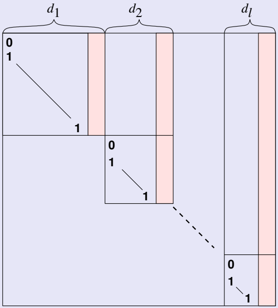
Fact

If $(d_1, \ldots d_l)$ *is lexicographically maximal such that*

$$K = \left[\begin{array}{ccc|c|ccc} v_1 & \ldots & A^{d_1-1}v_1 & \ldots & v_l & \ldots & A^{d_l-1}v_l \end{array}\right]$$

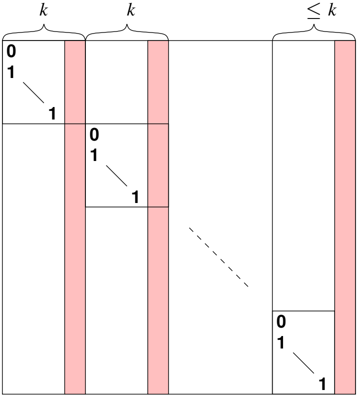
is non-singular, then

$A \times K = K \times$



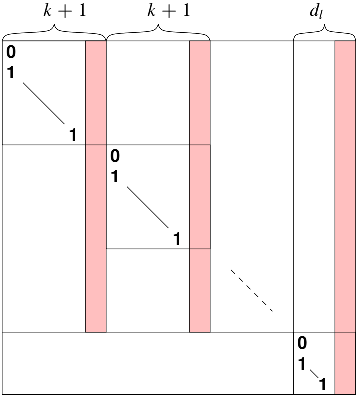
Principle

k-shifted form:



Principle

$k+1$-shifted form:



Principle

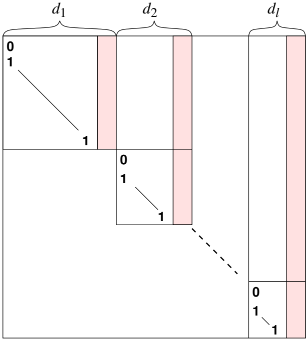
- Compute iteratively from 1-shifted form to d_1-shifted form

Principle

- Compute iteratively from 1-shifted form to d_1-shifted form
- each diagonal block appears in the increasing degree

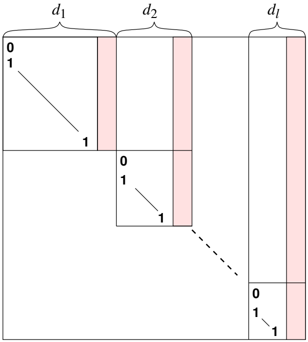
Principle

- Compute iteratively from 1-shifted form to d_1-shifted form
- each diagonal block appears in the increasing degree
- until the shifted Hessenberg form is obtained:



Principle

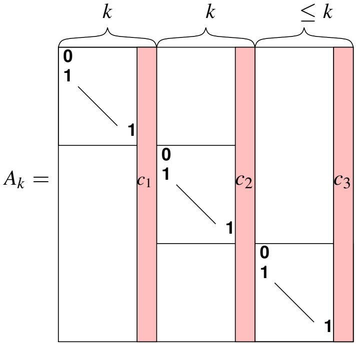
- Compute iteratively from 1-shifted form to d_1-shifted form
- each diagonal block appears in the increasing degree
- until the shifted Hessenberg form is obtained:



How to transform from k to $k+1$-shifted form ?

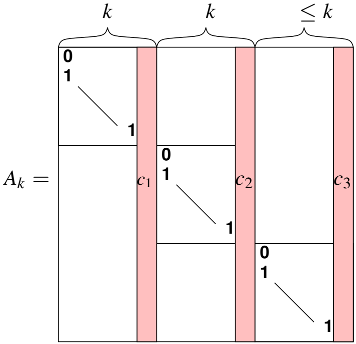
Krylov normal extension

for any k-shifted form

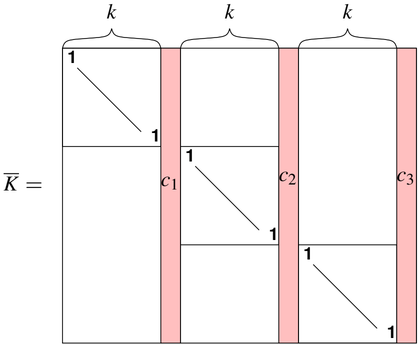


Krylov normal extension

for any k-shifted form

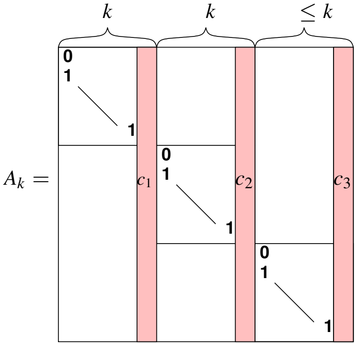


compute the $n \times (n+k)$ matrix

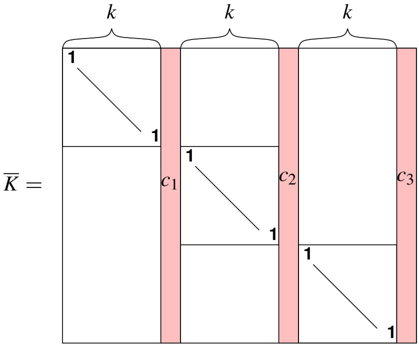


Krylov normal extension

for any k-shifted form



compute the $n \times (n+k)$ matrix



and form K by picking its first linearly independent columns.

The algorithm

- Form $\overline{K}$: just copy the columns of A_k

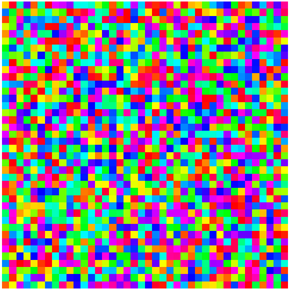
The algorithm

- Form $\overline{K}$: just copy the columns of A_k
- Compute K: rank profile of $\overline{K}$

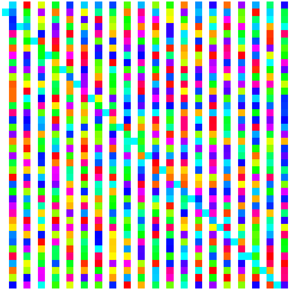
The algorithm

- Form $\overline{K}$: just copy the columns of A_k
- Compute K: rank profile of $\overline{K}$
- Apply the similarity transformation $K^{-1}A_kK$

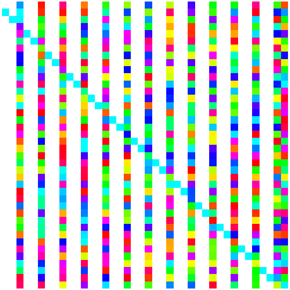
Example



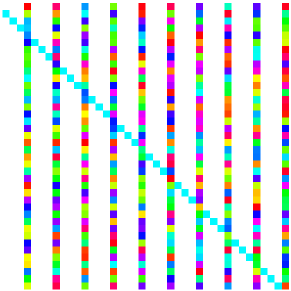
Example



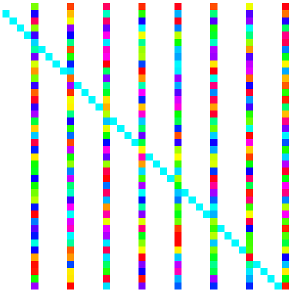
Example



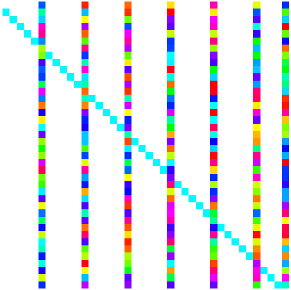
Example



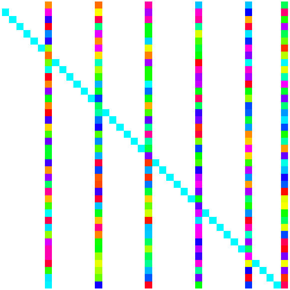
Example



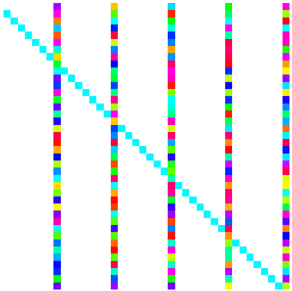
Example



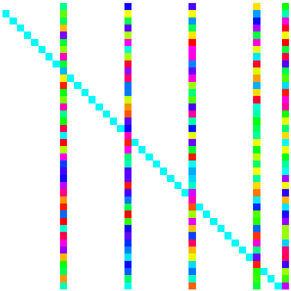
Example



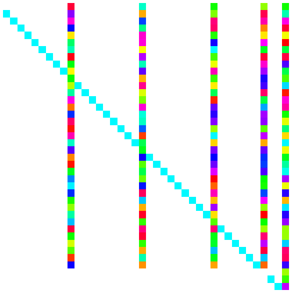
Example



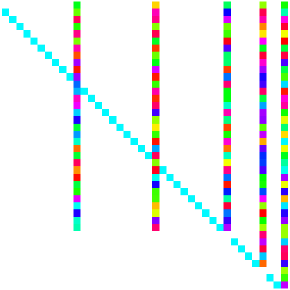
Example



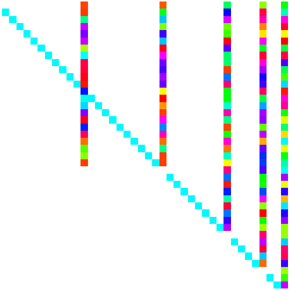
Example



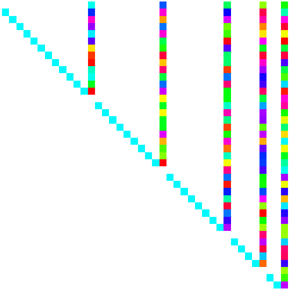
Example



Example



Example



Lemma

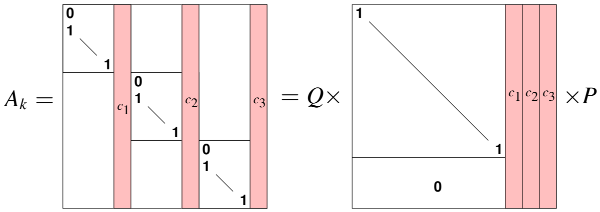
If $\#F > 2n^2$, the transformation will succeed with high probability. Failure is detected.

Lemma

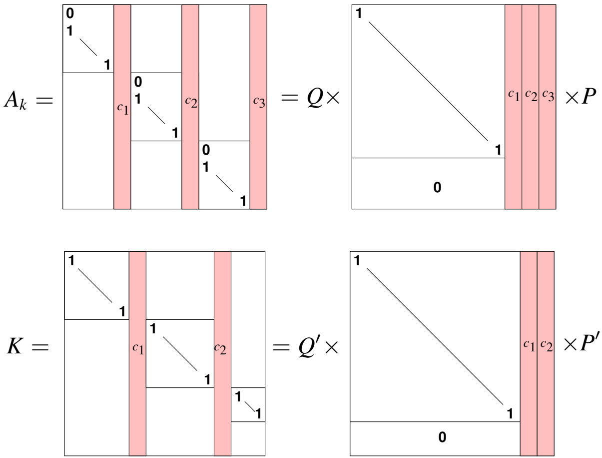
If $\#F > 2n^2$, the transformation will succeed with high probability. Failure is detected.

How to use fast matrix arithmetic ?

Permutations: compressing the dense columns



Permutations: compressing the dense columns



Reduction to Matrix multiplication

Similarity transformation: parenthesing

$$K^{-1}AK = Q'^{-1} \begin{bmatrix} I & * \\ 0 & * \end{bmatrix} P'^{-1} Q \begin{bmatrix} I & * \\ 0 & * \end{bmatrix} PQ' \begin{bmatrix} I & * \\ 0 & * \end{bmatrix} P'$$

Reduction to Matrix multiplication

Similarity transformation: parenthesing

$$K^{-1}AK = Q'^{-1}\left(\begin{bmatrix} I & * \\ 0 & * \end{bmatrix}\left(P'^{-1}Q\left(\begin{bmatrix} I & * \\ 0 & * \end{bmatrix}\left(PQ'\begin{bmatrix} I & * \\ 0 & * \end{bmatrix}\right)\right)\right)\right)P'$$

Reduction to Matrix multiplication

Similarity transformation: parenthesing

$$K^{-1}AK = Q'^{-1}\left(\begin{bmatrix} I & * \\ 0 & * \end{bmatrix}\left(P'^{-1}Q\left(\begin{bmatrix} I & * \\ 0 & * \end{bmatrix}\left(PQ'\begin{bmatrix} I & * \\ 0 & * \end{bmatrix}\right)\right)\right)\right)P'$$

$$\Rightarrow \mathcal{O}\left(k\left(\frac{n}{k}\right)^{\omega}\right)$$

Reduction to Matrix multiplication

Similarity transformation: parenthesing

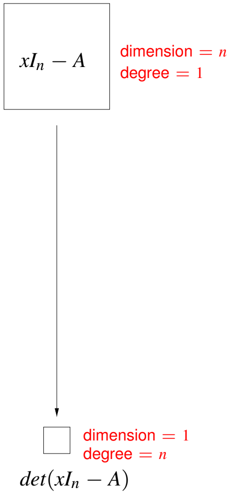
$$K^{-1}AK = Q'^{-1}\left(\begin{bmatrix} I & * \\ 0 & * \end{bmatrix}\left(P'^{-1}Q\left(\begin{bmatrix} I & * \\ 0 & * \end{bmatrix}\left(PQ'\begin{bmatrix} I & * \\ 0 & * \end{bmatrix}\right)\right)\right)\right)P'$$

$$\Rightarrow \mathcal{O}\left(k\left(\frac{n}{k}\right)^{\omega}\right)$$

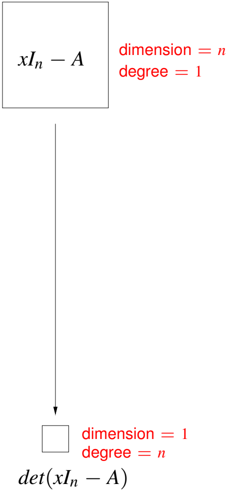
Overall complexity: summing for each iteration:

$$\sum_{k=1}^{n} k\left(\frac{n}{k}\right)^{\omega} = n^{\omega}\sum_{k=1}^{n}\left(\frac{1}{k}\right)^{\omega-1} = \zeta(\omega-1)n^{\omega} = \mathcal{O}\left(n^{\omega}\right)$$

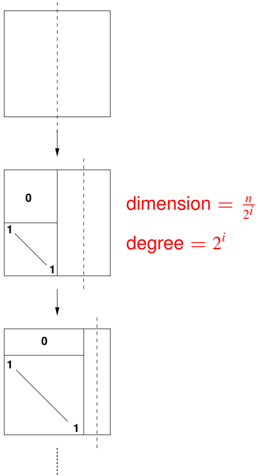
A new type of reduction



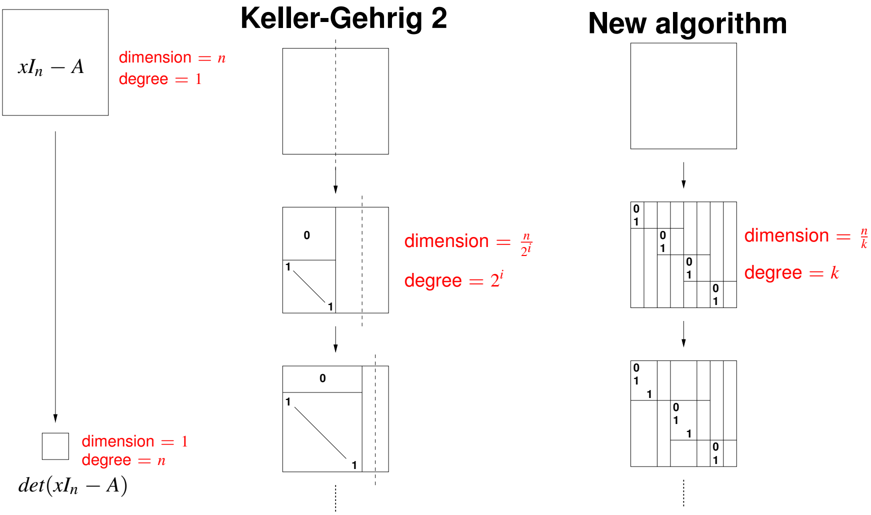
A new type of reduction



Keller-Gehrig 2



A new type of reduction



Conclusion

Reductions to a building block

Matrix Mult: block rec. $\sum_{i=1}^{\log n} n \left(\frac{n}{2^i}\right)^{\omega-1} = \mathcal{O}\left(n^\omega\right)$ (Gauss, REF)

Matrix Mult: Iterative $\sum_{k=1}^{n} n \left(\frac{n}{k}\right)^{\omega-1} = \mathcal{O}\left(n^\omega\right)$ (Frobenius)

Linear Sys: over $\mathbb{Z}$ (Hermite Normal Form)

Size/dimension compromises

- Hermite normal form : rank 1 updates reducing the determinant
- Frobenius normal form : degree k, dimension n/k for $k = 1 \ldots n$