How much does exactness cost?*

On polynomial and integer matrix computations

Gilles Villard

CNRS/ LIP, École Normale Supérieure de Lyon

http://www.ens-lyon.fr/~gvillard

*SIAM Conference on Applied Linear Algebra, July 16, 2003, Williamsburg.

Problem:

Study of the **complexity** of fundamental problems in **exact** linear algebra over $\mathsf{K}[x]$ and $\mathbb{Z}$.

- ▷ Worst case complexity;
- ▷ Time complexity *i.e.* fastest algorithms;
- ▷ Up to logarithmic factors, soft "O" notation: $O\tilde{~}(f) = f^{1+o(1)}$;
- ▷ Deterministic or randomized algorithms.

Models of computation/matrix domains.

Algebraic complexity,

matrices in $\mathsf{K}^{n\times n}$ with K a commutative field,

arithmetic operations $+, \times, /$ in K.

versus

$\hookrightarrow$ $\mathsf{K}[x]^{n\times n}$, arithmetic operations $+, \times, /$ in K.

$\hookrightarrow$ Bit complexity,

$\mathbb{Z}^{n\times n}$, bit operations.

Motivations.

- Complexity estimates with “concrete” entry domains,
- Better understanding of linear algebra under bit complexity models,
- Improved algorithms for exact (or accurate) results.

Organization of the talk

I - Algebraic *versus* bit complexity.

II - Reductions between problems and target complexity.

III - Polynomial matrix computations.

IV - Integer matrix computations.

Conclusion

Organization of the talk

I - Algebraic *versus* bit complexity.

II - Reductions between problems and target complexity.

III - Polynomial matrix computations.

IV - Integer matrix computations.

Conclusion

Algebraic complexity over K

Equivalence to **matrix multiplication** (*straight-line*)

Matrix multiplication $n \times n$

$A \times B$

n^{ω}, n^3 ou $n^{2.376}$

Determinant, **inversion**,

rank, characteristic polynomial,

Frobenius form, QR decomposition...

RAM algorithms in $O\tilde{}(n^{\omega})$

- [Strassen 69, Bunch & Hopcroft 74] Det $\preceq$ MM
- [Strassen 73, Baur & Strassen 83] MM $\preceq$ Det

$\hookrightarrow$ MM $\preceq$ Det $\preceq$ MM

$\mathsf{K}[x]$ or **Bit complexity**

Inputs and outputs have a **size** or a **precision**.

$\rightsquigarrow$ **Impact** on the problem's complexity?

$\mathsf{K}[x]$ or **Bit complexity**

Inputs and outputs have a **size** or a **precision**.

$\rightsquigarrow$ **Impact** on the problem's complexity?

- $A \in \mathsf{K}[x]^{n\times n}$: $\deg\det A = O(nd)$.

$\mathsf{K}[x]$ or **Bit complexity**

Inputs and outputs have a **size** or a **precision**.

$\rightsquigarrow$ **Impact** on the problem's complexity?

- $A \in \mathsf{K}[x]^{n\times n}$: $\deg \det A = O(nd)$.

$\|A\| = \max_{i,j} |a_{i,j}|$ (or d).

- $A \in \mathbb{Z}^{n\times n}$: size$(\det A) = \tilde{O}(n \log \|A\|)$.

$\mathsf{K}[x]$ or **Bit complexity**

Inputs and outputs have a **size** or a **precision**.

$\rightsquigarrow$ **Impact** on the problem's complexity?

- $A \in \mathsf{K}[x]^{n\times n}$: $\deg \det A = O(nd)$.

$\|A\| = \max_{i,j} |a_{i,j}|$ (or d).

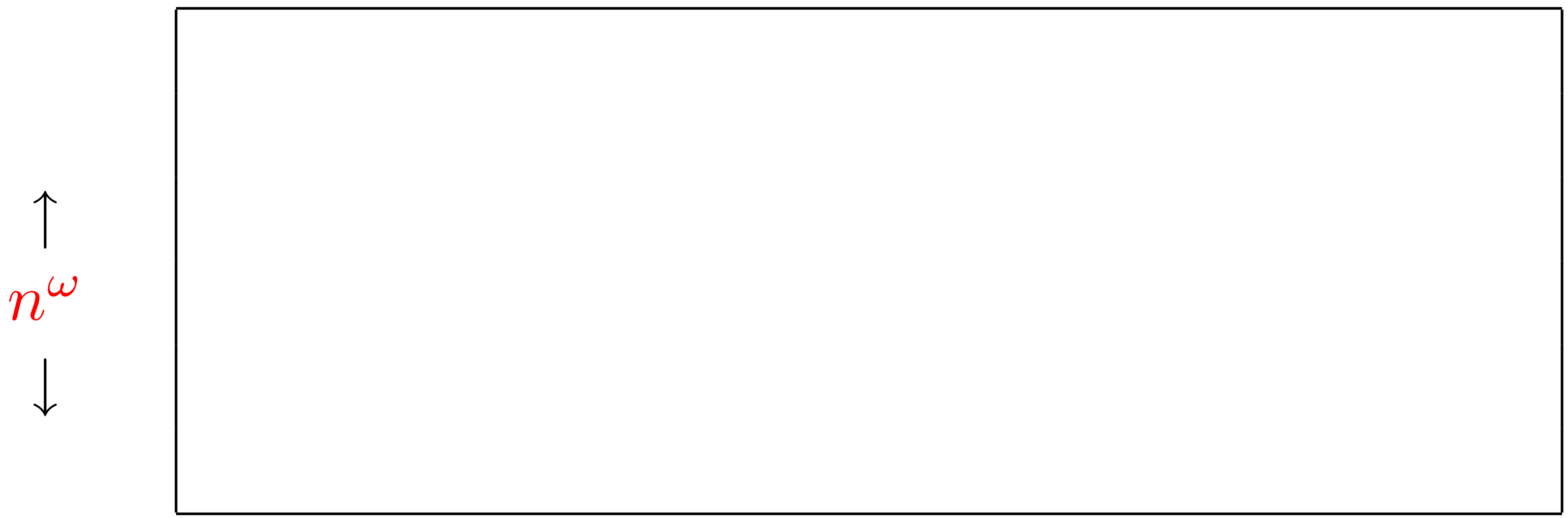
- $A \in \mathbb{Z}^{n\times n}$: size$(\det A) = \tilde{O}(n \log \|A\|)$.
- $A \in \mathbb{Z}^{n\times n}$: $O(\log \mathsf{cond}(A)) = \tilde{O}(n \log \|A\|)$ bits for accuracy.

Impact of data size?

Ex. Determinant computation/Output size: nd or $O\tilde{}(n \log \|A\|)$,

Evaluation/interpolation scheme or Chinese remaindering

or $O\tilde{}(n \log \|A\|)$ bits *a priori*:



Impact of data size?

Ex. Determinant computation/Output size: $\color{red}{n}d$ or $O\tilde{}(\color{red}{n}\log\|A\|)$,

Evaluation/interpolation scheme or Chinese remaindering

or $O\tilde{}(\color{red}{n}\log\|A\|)$ bits *a priori*:

$\longleftarrow$ $\color{red}{n}d$ points or $O\tilde{}(\color{red}{n}\log\|A\|)$ bits $\longrightarrow$

$\uparrow$

$\color{red}{n^\omega}$

$\downarrow$

Impact of data size?

Ex. Determinant computation/Output size: nd or $O\tilde{}(n \log \|A\|)$,

Evaluation/interpolation scheme or Chinese remaindering

or $O\tilde{}(n \log \|A\|)$ bits *a priori*:

$\longleftarrow$ nd points or $O\tilde{}(n \log \|A\|)$ bits $\longrightarrow$

$\uparrow$

n^ω

$\downarrow$

Complexity estimates:

$$O\tilde{}(n^\omega \times nd) = O\tilde{}(n^{\omega+1} d),$$
$$O\tilde{}(n^{\omega+1} \log \|A\|).$$

Fundamentals of symbolic dense linear algebra over $\mathsf{K}[x]$ or $\mathbb{Z}$:

System solution [Moenck & Carter 79, Dixon 82] Hensel lifting	$\tilde{O}(n^\omega \log \|A\|)$ Las Vegas
Determinant, **inversion**, nullspace. . . [Edmonds 67, Bareiss 69, Moenck & Carter 79] Fraction-free, Chinese remaindering, Newton-Hensel lifting	$\tilde{O}(n \cdot n^\omega \log \|A\|)$ Deterministic
Frobenius form (minimum, characteristic polynomial) [Giesbrecht 93, Giesbrecht & Storjohann 02] Danilevsky elimination, Keller-Gehrig, Chinese remaindering	$\tilde{O}(n \cdot n^\omega \log \|A\|)$ Las Vegas
Hermite and **Smith forms**, (diophantine systems) [Kannan & Bachem 79, Domich 85, Giesbrecht 95, Storjohann 96-00] Unimodular eliminations	$\tilde{O}(n \cdot n^\omega \log \|A\|)$ Deterministic

Bit complexity $\preceq$ algebraic complexity $\times$ output size

$$\boxed{\textbf{Bit complexity} \preceq \textbf{algebraic complexity} \times \textbf{output size}}$$

Is this bound pessimistic?

$$\textbf{Bit complexity} \preceq \textbf{algebraic complexity} \times \textbf{output size}$$

Is this bound pessimistic?

Clue. The output length may not be necessary *a priori*, i.e. at the beginning of the computation, but only at its very end.

Change of the situation: **reduced overhead or no overhead**

Theorem. The determinant and the Smith normal form of $A \in \mathbb{Z}^{n\times n}$ can be computed by a Monte Carlo algorithm in $O\tilde{}(\sqrt{n} \cdot n^3 \log^{1.5} \|A\|)$ bit operations.

- Search and structured rank-k perturbations for the characteristic polynomial of a sparse matrix;

- Search and dense integer rank-k perturbations for the Smith form of an integer matrix.

[Eberly, Giesbrecht & Villard 00, Kaltofen 92/00, Villard 00]

Theorem. The determinant and the Hermite normal form of $A \in \mathsf{K}[x]^{n\times n}$ can be computed in $O(n^3 d^2)$ operations in K.

- Column reduction [Mulders & Storjohann 00].

Organization of the talk

I - Algebraic *versus* bit complexity.

II - Reductions between problems and target complexity.

III - Polynomial matrix computations.

IV - Integer matrix computations.

Conclusion

$A \in \mathsf{K}[x]^{n\times n}$ or $A \in \mathbb{Z}^{n\times n}$

Target problems: determinant, characteristic polynomial, nullspace, rank, inversion, Frobenius, Hermite, Smith normal form and associated transform, minimal bases, matrix gcd . . .

$A \in \mathsf{K}[x]^{n \times n}$ or $A \in \mathbb{Z}^{n \times n}$

Target problems: determinant, characteristic polynomial, nullspace, rank, inversion, Frobenius, Hermite, Smith normal form and associated transform, minimal bases, matrix gcd . . .

Target complexity estimate?

$A \in \mathsf{K}[x]^{n\times n}$ or $A \in \mathbb{Z}^{n\times n}$

Target problems: determinant, characteristic polynomial, nullspace, rank, inversion, Frobenius, Hermite, Smith normal form and associated transform, minimal bases, matrix gcd . . .

Target complexity estimate?

Nota. Known algebraic complexity **reduction techniques** between problems may not be preserved in bit complexity.

Example.

▷ Over K, Determinant in n^ω $\Longrightarrow$ Inversion in n^ω

Derivative inequality [Linnainmaa 76, Baur et Strassen 83, Morgenstern 85].

$$a^*_{j,i} = \frac{\partial \det A}{\partial\ a_{i,j}}.$$

Example.

▷ Over K, Determinant in n^ω $\Longrightarrow$ Inversion in n^ω

Derivative inequality [Linnainmaa 76, Baur et Strassen 83, Morgenstern 85].

$$a^*_{j,i} = \frac{\partial \det A}{\partial\ a_{i,j}}.$$

▷ Over $\mathbb{Z}$, x and y vectors with constant entries, c a large constant,

$\phi = c \cdot x^t \cdot y$ takes $O(n + \log |c|)$ bit operations,

$[\partial\phi/\partial x_i] = c \cdot y$ takes $O(n \log |c|)$ bit operations.

$\rightsquigarrow$ Link with polynomial or integer matrix multiplication?

Theorem. If there is a straight-line program of length $D(n,d)$ over K which computes the $(d+1)$**st coefficient of the determinant** of an $n \times n$ matrix of degree d, then there is a straight-line program of length no more than $8D(n,d)$ which **multiplies two $n \times n$ matrices of degree d** [Giorgi, Jeannerod & Villard 03].

$\rightsquigarrow$ Link with polynomial or integer matrix multiplication?

Theorem. If there is a straight-line program of length $D(n, d)$ over K which computes the $(d+1)$**st coefficient of the determinant** of an $n \times n$ matrix of degree d, then there is a straight-line program of length no more than $8D(n, d)$ which **multiplies two** $n \times n$ **matrices of degree** d [Giorgi, Jeannerod & Villard 03].

C.f. the relation between estimating error bounds (condition estimation) and testing matrix multiplication

[Demmel, Diament & Malajovich 01].

Candidate target complexity estimate:

$$\boxed{\mathsf{MM}(n, \log \|A\|) + \text{input/output size}}$$

Which integer (resp. polynomial) exact matrix problems can be solved with roughly the same number of arithmetic operations than integer (resp. polynomial) matrix multiplication plus the input/output size?

Organization of the talk

I - Algebraic *versus* bit complexity.

II - Reductions between problems and target complexity.

III - Polynomial matrix computations.

IV - Integer matrix computations.

Conclusion

$A \in \mathsf{K}[x]^{n \times n}$ of degree d, $\mathsf{MM}(n,d) = O\tilde{~}(n^{\omega+1}d)$ or $O(n^3 d^2)$.

▷ **Inversion**

(generic inputs) [Jeannerod & Villard 02]

▷ **Determinant**

[Storjohann 02]

▷ **Column reduction**

[Giorgi, Jeannerod & Villard 03]

III-1/ Matrix inversion

Size of the output: $n^2 \times n(d+1) = O(n^3 d)$ elements in K.

III-1/ Matrix inversion

Size of the output: $n^2 \times n(d+1) = O(n^3 d)$ elements in K.

Rich literature on the subject,

[Gauss, Hensel, Hermite/Lagrange, Le Verrier . . .]

$\rightsquigarrow$ Algorithms in $\tilde{O}(nd \times n^3)$ or $\tilde{O}(nd \times n^\omega)$.

Essentially optimal computation of the inverse

[Jeannerod & Villard 02]

Theorem. Except on a subvariety, the inverse of $A \in \mathsf{K}[x]^{n\times n}$ of degree d can be computed in $O\tilde{}(n^3 d)$ operations in K.

Diagonalization in $\log_2(n)$ steps

$$A = \begin{bmatrix} * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \end{bmatrix}$$

Diagonalization in $\log_2(n)$ steps

$$BA = \begin{bmatrix} * & * & * & * & & & & \\ * & * & * & * & & & & \\ * & * & * & * & & & & \\ * & * & * & * & & & & \\ & & & & * & * & * & * \\ & & & & * & * & * & * \\ & & & & * & * & * & * \\ & & & & * & * & * & * \end{bmatrix}$$

Diagonalization in $\log_2(n)$ steps

$$
B'BA = \begin{bmatrix}
* & * & & & & & & \\
* & * & & & & & & \\
& & * & * & & & & \\
& & * & * & & & & \\
& & & & * & * & & \\
& & & & * & * & & \\
& & & & & & * & * \\
& & & & & & * & *
\end{bmatrix}
$$

Diagonalization in $\log_2(n)$ steps

$$B''B'BA = \begin{bmatrix} * & & & & & & & \\ & * & & & & & & \\ & & * & & & & & \\ & & & * & & & & \\ & & & & * & & & \\ & & & & & * & & \\ & & & & & & * & \\ & & & & & & & * \end{bmatrix} = D$$

$$\begin{bmatrix} \times & \times \\ ? & ? \end{bmatrix} \begin{bmatrix} A_1 & \times \\ A_2 & \times \end{bmatrix} = \begin{bmatrix} \times & \times \\ 0 & \times \end{bmatrix}$$

$$\begin{bmatrix} \times & \times \\ -A_2 A_1^{-1} & I_n \end{bmatrix} \begin{bmatrix} A_1 & \times \\ A_2 & \times \end{bmatrix} = \begin{bmatrix} \times & \times \\ 0 & \times \end{bmatrix}$$

Schur complement: too fast increase of the degrees,

the first step already uses $O(n^\omega \times nd)$ operations in K,

$$\Longrightarrow \quad O(n^{\omega+1} \times d).$$

Minimal kernel bases over $\mathsf{K}[x]$ [Forney 75]

$$A = \begin{bmatrix} * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \end{bmatrix} \quad \longrightarrow \quad BA = \begin{bmatrix} * & * & * & * & & & & \\ * & * & * & * & & & & \\ * & * & * & * & & & & \\ * & * & * & * & & & & \\ & & & & * & * & * & * \\ & & & & * & * & * & * \\ & & & & * & * & * & * \\ & & & & * & * & * & * \end{bmatrix}$$

$$\begin{bmatrix} \overline{B} \\ \underline{B} \end{bmatrix} \begin{bmatrix} A_L & A_R \end{bmatrix} = \begin{bmatrix} A'_L & 0 \\ 0 & A'_R \end{bmatrix}$$

with $\underline{B}$ (and $\overline{B}$) **minimal basis** of $\ker A_L$ as a $\mathsf{K}[x]$-submodule.

The row degrees of $\underline{B}$ and of $\overline{B}$ are the smallest possible ones.

Example.

$$\underline{B}A_L = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ -1 & x & -x^2 & x^3 & 0 \end{bmatrix} \begin{bmatrix} x & 0 & 0 \\ 1 & x & 0 \\ 0 & 1 & x \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = 0.$$

The degree may be as large as $nd/2$ for $\ker A_L$ in the worst case.

The row degrees of $\underline{B}$ and of $\overline{B}$ are the smallest possible ones.

Example.

$$\underline{B}A_L = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ -1 & x & -x^2 & x^3 & 0 \end{bmatrix} \begin{bmatrix} x & 0 & 0 \\ 1 & x & 0 \\ 0 & 1 & x \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = 0.$$

The degree may be as large as $nd/2$ for $\ker A_L$ in the worst case.

Lemma. For a generic A_L the degree is d exactly.

Inversion: operation count

Generic minimal basis computation: $O\tilde{}(n^3 d)$ or $O\tilde{}(n^\omega d)$

[e.g. matrix Padé approximation, Beckermann & Labahn 94]

[or Knuth/Schönhage/Moenck Euclidean algorithm for matrix polynomials]

Determinant - Computation of the minimal kernels and of D:

$$\sum_{i=0}^{\log n - 1} 2^{i+1} \times O\tilde{}\left(\left(\frac{n}{2^i}\right)^\omega \times 2^i d\right) = O\tilde{}(n^\omega d).$$

Inversion - Product of the $\log n$ transformations:

$$\sum_{i=0}^{\log n - 1} 2^i \times 2^i \times O\tilde{}\left(\left(\frac{n}{2^i}\right)^\omega \times 2^i d\right) = O\tilde{}(n^3 d).$$

III-2/ Matrix determinant

$O\tilde{}(n^{\omega+1}d) \leq \quad n^{3.38}d^{1+\epsilon}$ [Classical approaches]

III-2/ Matrix determinant

$$O\tilde{\ }(n^{\omega+1}d) \leq \quad n^{3.38}d^{1+\epsilon} \text{ [Classical approaches]}$$

$$n^{3.19}d^{1+\epsilon} \text{ [Eberly, Giesbrecht, Villard 2000]}$$

III-2/ Matrix determinant

$$O\tilde{}(n^{\omega+1}d) \leq \quad n^{3.38}d^{1+\epsilon} \text{ [Classical approaches]}$$

$$n^{3.19}d^{1+\epsilon} \text{ [Eberly, Giesbrecht, Villard 2000]}$$

$$n^{3.03}d^{1+\epsilon} \text{ [Kaltofen 1992/2000]}$$

III-2/ Matrix determinant

$$O\tilde{}(n^{\omega+1}d) \leq \quad n^{3.38}d^{1+\epsilon} \text{ [Classical approaches]}$$
$$n^{3.19}d^{1+\epsilon} \text{ [Eberly, Giesbrecht, Villard 2000]}$$
$$n^{3.03}d^{1+\epsilon} \text{ [Kaltofen 1992/2000]}$$
$$n^{2.7}d^{1+\epsilon} \text{ [Kaltofen, Villard 2001]}$$

III-2/ Matrix determinant

$$
\begin{aligned}
\tilde{O}(n^{\omega+1}d) \leq \quad & n^{3.38}d^{1+\epsilon} \text{ [Classical approaches]} \\
& n^{3.19}d^{1+\epsilon} \text{ [Eberly, Giesbrecht, Villard 2000]} \\
& n^{3.03}d^{1+\epsilon} \text{ [Kaltofen 1992/2000]} \\
& n^{2.7}d^{1+\epsilon} \text{ [Kaltofen, Villard 2001]} \\
\tilde{O}(n^{\omega}d) \leq \quad & n^{2.38}d^{1+\epsilon} \text{ [Storjohann 2002]}
\end{aligned}
$$

One of the ingredients: **high-order lifting**

(quadratic iterative refinement for computing the error)

III-3/ Column reduction

Matrix pencils and Kronecker indices,

[Van Dooren 79-81, Beelen, Van den Hurk & Praagman 88, Praagman *et al.* 88-98].

Basis reduction,

[Wolovitch 78, Kailath 80, Paulus 98, Mulders & Storjohann 00].

$\rightsquigarrow$ Algorithms in $O(n^3d^2)$.

Definition

$$A(x) = \begin{bmatrix} x+1 & x^2 \\ x^2 & x^3 + x^2 + 1 \end{bmatrix}$$

Definition

$$A(x) = \begin{bmatrix} x+1 & x^2 \\ x^2 & x^3+x^2+1 \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} x+1 & x \\ x^2 & x^2+1 \end{bmatrix}$$

Definition

$$A(x) = \begin{bmatrix} x+1 & x^2 \\ x^2 & x^3+x^2+1 \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} x+1 & x \\ x^2 & x^2+1 \end{bmatrix} \rightarrow \quad C(x) = \begin{bmatrix} x+1 & 1 \\ x^2 & 1 \end{bmatrix}$$

The column leading matrix $[C]_l$ of $C = AU$ has maximal rank.

Consequence. The columns of C provide a minimal degree basis of the corresponding $\mathsf{K}[x]$-module.

$\rightsquigarrow$ Link with polynomial matrix multiplication?

“Easier” than polynomial matrix multiplication

[Giorgi, Jeannerod & Villard 03]

Theorem. A column reduced form of a non singular matrix A of degree d in $\mathsf{K}[x]^{n\times n}$ can be computed by a Las Vegas (certified) algorithms in $\mathsf{MM}'(n,d) + \tilde{O}(n^2 d)$ or $\tilde{O}(n^\omega d)$ operations in K.

NB. $\mathsf{MM}'(n,d) = O(\mathsf{MM}(n,d) + \sum_{i=0}^{\log d} 2^i \mathsf{MM}(n, 2^{-i}d) + \sum_{i=0}^{\log d} 4^i \mathsf{MM}(2^{-i}n, d)$.

Difficulty

Small degrees in the matrix but large degrees in the transformation
(possibly long chain of cancellations)

Example.

$$\begin{bmatrix} 1 & x & 0 & 0 \\ 0 & 1 & x & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & x & 0 & 0 \\ 0 & 1 & x & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Difficulty

Small degrees in the matrix but large degrees in the transformation
(possibly long chain of cancellations)

Example.

$$\begin{bmatrix} 1 & x & 0 & 0 \\ 0 & 1 & x & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & x & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Difficulty

Small degrees in the matrix but large degrees in the transformation
(possibly long chain of cancellations)

Example.

$$\begin{bmatrix} 1 & x & 0 & 0 \\ 0 & 1 & x & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -x & x^2 & 0 \\ 0 & 1 & -x & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Difficulty

Small degrees in the matrix but large degrees in the transformation

(possibly long chain of cancellations)

Example.

$$\begin{bmatrix} 1 & x & 0 & 0 \\ 0 & 1 & x & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -x & x^2 & -x^3 \\ 0 & 1 & -x & x^2 \\ 0 & 0 & 1 & -x \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Column reduction: the approach

Minimal basis or matrix approximation,

[Beelen, van den Hurk & Praagman 88]

[Villard 96] [Beckermann, Labahn & Villard 99]

$$A(x)U(x) = C(x) \iff \left[\begin{array}{cc} A^{-1}(x) & I \end{array}\right] \cdot \left[\begin{array}{c} C(x) \\ U(x) \end{array}\right] = 0$$

Column reduction: the approach

Minimal basis or matrix approximation,

[Beelen, van den Hurk & Praagman 88]

[Villard 96] [Beckermann, Labahn & Villard 99]

$$A(x)U(x) = C(x) \iff \begin{bmatrix} A^{-1}(x) & I \end{bmatrix} \cdot \begin{bmatrix} C(x) \\ U(x) \end{bmatrix} = 0$$

Not enough: **too big degrees** (nd) in the transformation U.

High-order lifting and fraction reconstruction

$$A^{-1} = (A^{-1} \text{ mod } x^h) + x^h R A^{-1}.$$

$$\begin{array}{ccc} \text{Left fraction} & \longleftrightarrow & \text{Right fraction} \\ \text{Non proper} & \longleftrightarrow & \text{Proper} \end{array}$$

High-order lifting and fraction reconstruction

$$A^{-1} = (A^{-1} \text{ mod } x^h) + x^h RA^{-1}.$$

Left fraction	$\longleftrightarrow$	Right fraction
Non proper	$\longleftrightarrow$	Proper

$$A(x)U(x) = C(x) \iff \begin{bmatrix} R(x)A^{-1}(x) & I \end{bmatrix} \cdot \begin{bmatrix} C(x) \\ U'(x) \end{bmatrix} = 0$$

High-order lifting and fraction reconstruction

$$A^{-1} = (A^{-1} \bmod x^h) + x^h R A^{-1}.$$

Left fraction	$\longleftrightarrow$	Right fraction
Non proper	$\longleftrightarrow$	Proper

$$A(x)U(x) = C(x) \iff \begin{bmatrix} R(x)A^{-1}(x) & I \end{bmatrix} \cdot \begin{bmatrix} C(x) \\ U'(x) \end{bmatrix} = 0$$

Degree d everywhere.

Column reduction

Input : $A \in \mathsf{K}[x]^{n\times n}$ of degree d
Output : $C = AU$ column reduced

Column reduction

Input : $A \in \mathsf{K}[x]^{n \times n}$ of degree d
Output : $C = AU$ column reduced

> $2d$ terms of the expansion of A^{-1} of order higher than $(n-1)d$

Column reduction

Input : $A \in \mathsf{K}[x]^{n \times n}$ of degree d
Output : $C = AU$ column reduced

> $2d$ terms of the expansion of A^{-1} of order higher than $(n-1)d$
>
> Reconstruction of the fraction description $U'C^{-1}$

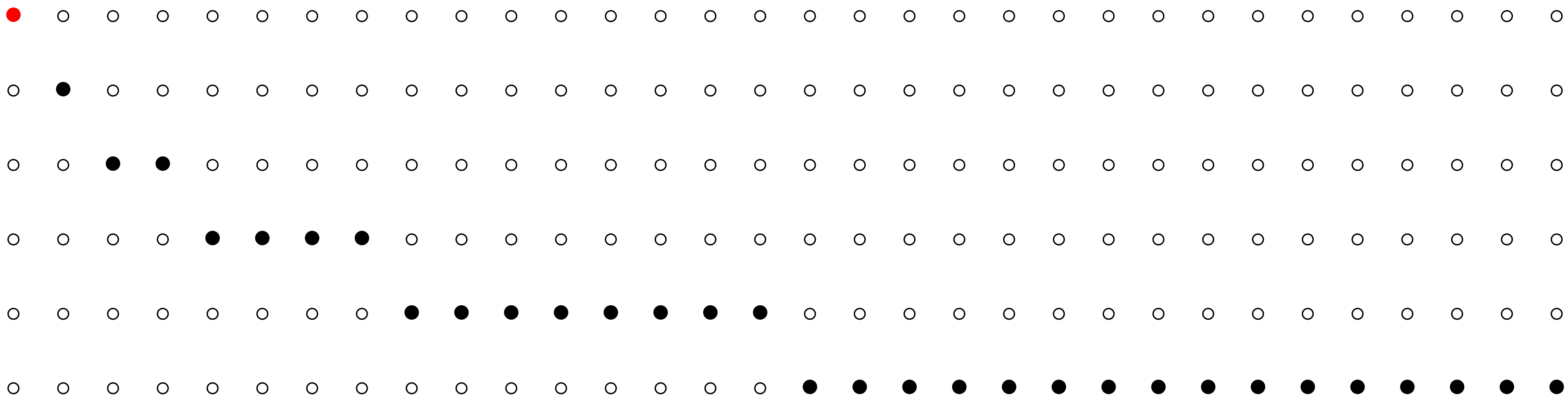
Second step: Knuth/Schönhage/Moenck fast recursive algorithm extended to matrix polynomials.

First step: **High-order lifting** [Storjohann 02]

(quadratic iterative refinement for computing the error)

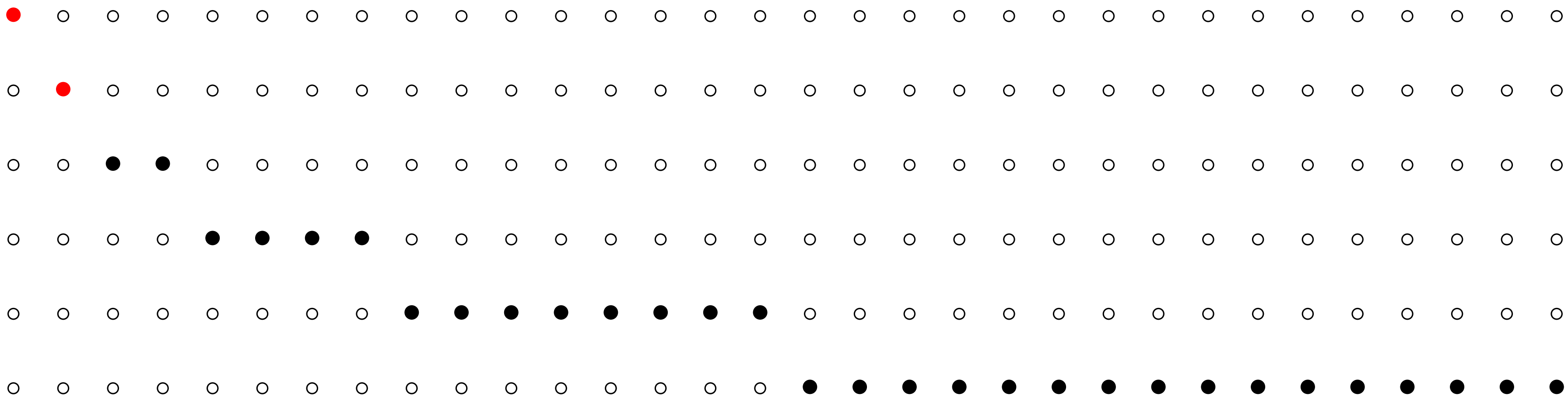
Computing the expansion of A^{-1}

Quadratic approximation



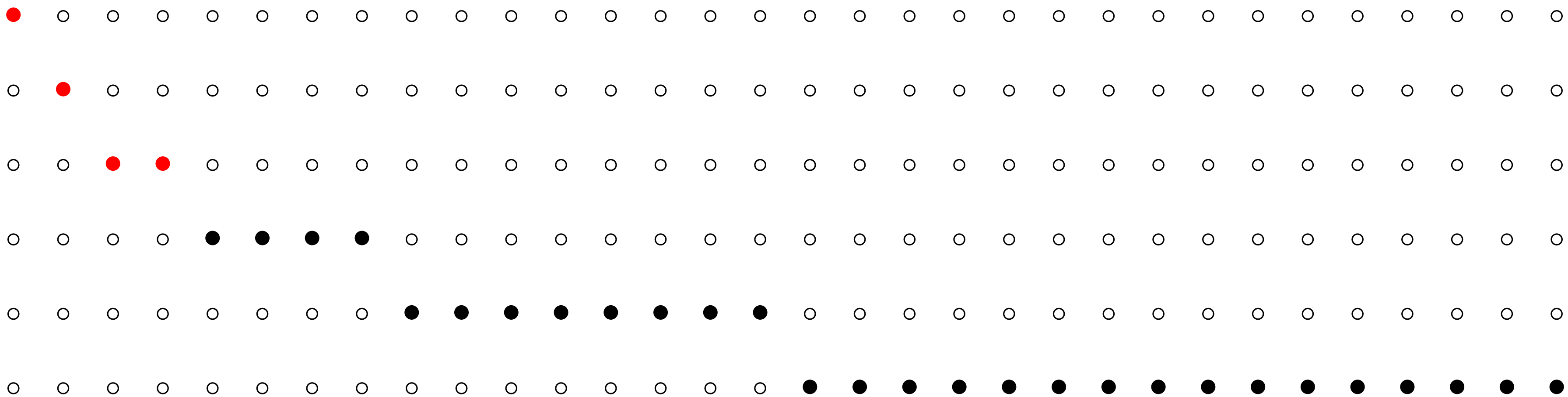
Computing the expansion of A^{-1}

Quadratic approximation



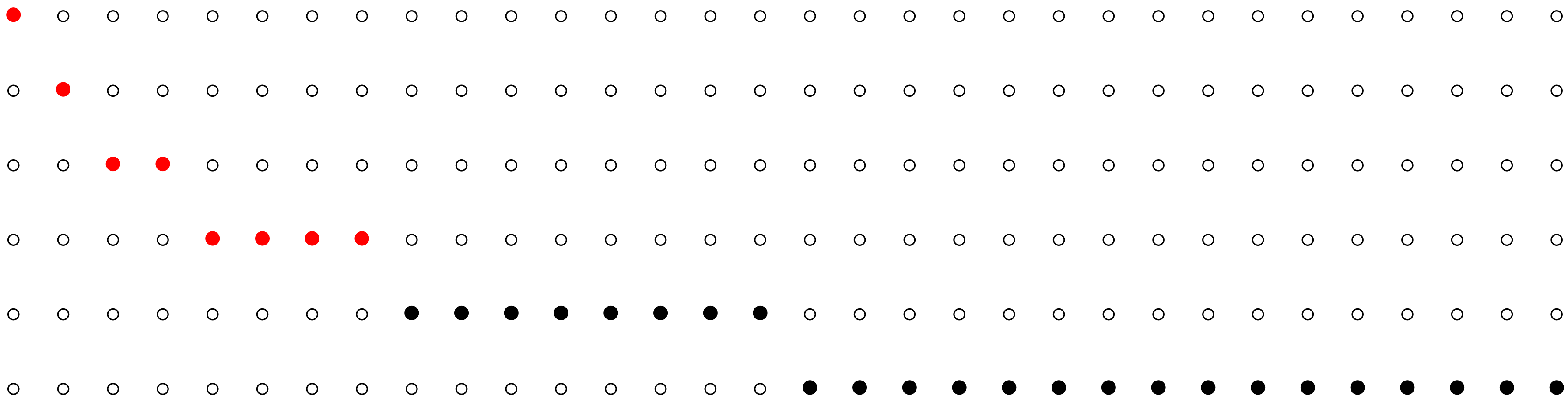
Computing the expansion of A^{-1}

Quadratic approximation



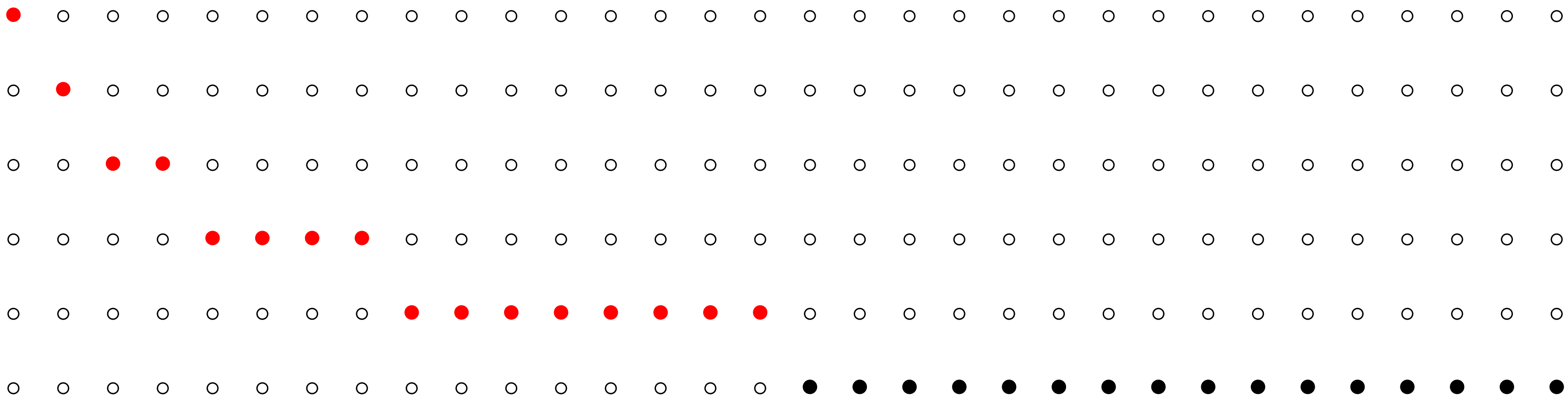
Computing the expansion of A^{-1}

Quadratic approximation



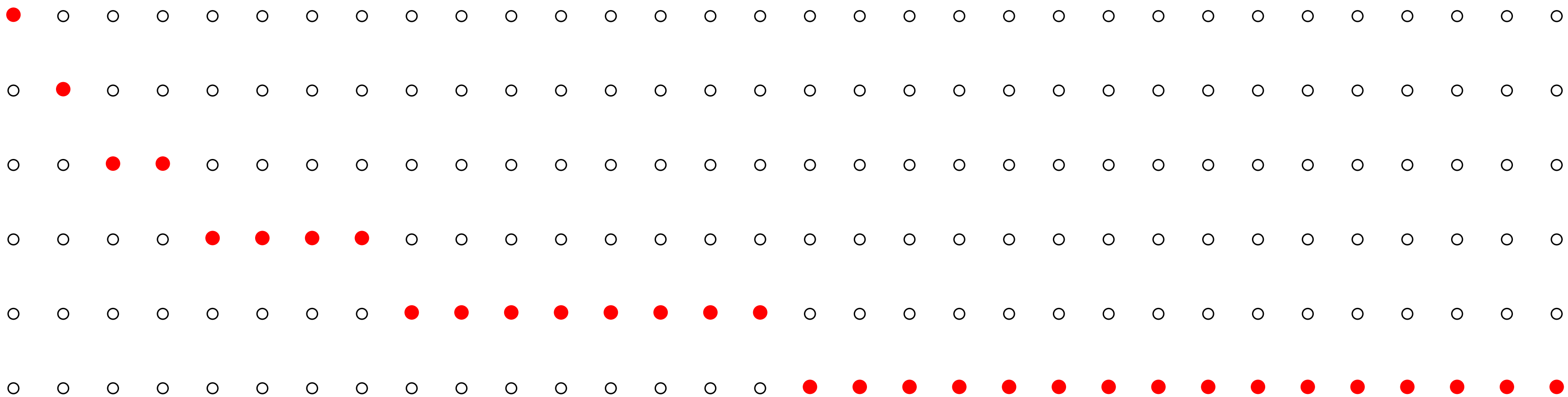
Computing the expansion of A^{-1}

Quadratic approximation



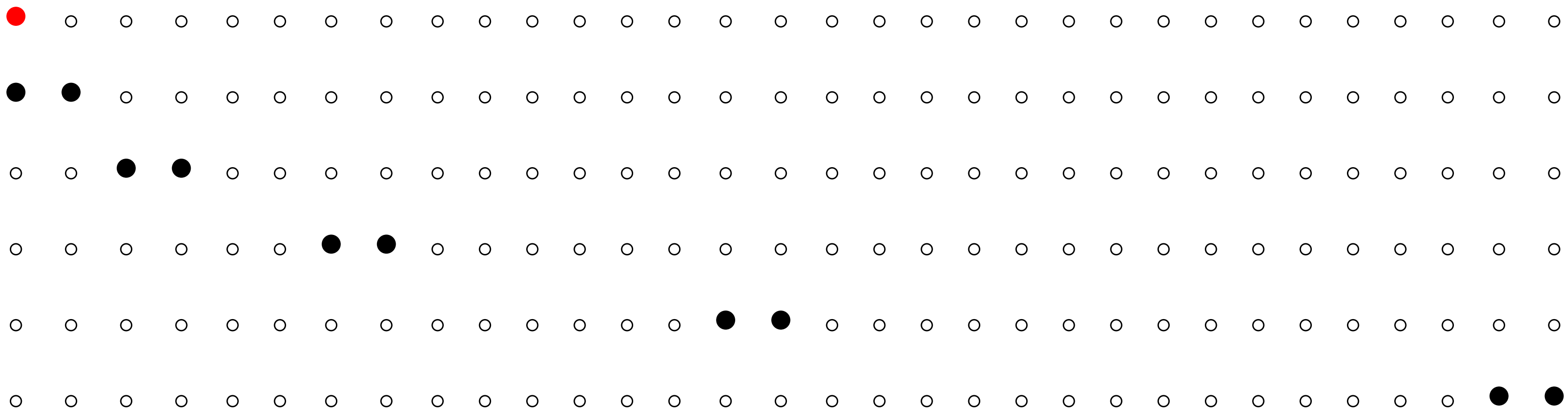
Computing the expansion of A^{-1}

Quadratic approximation



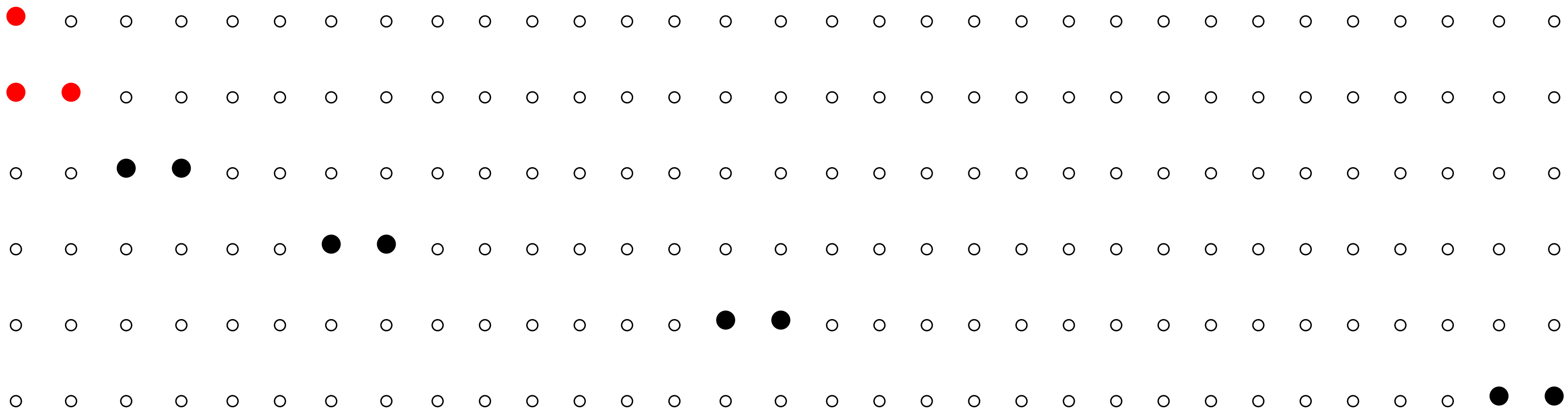
Computing the expansion of A^{-1}

High-order component lifting [Storjohann 02]



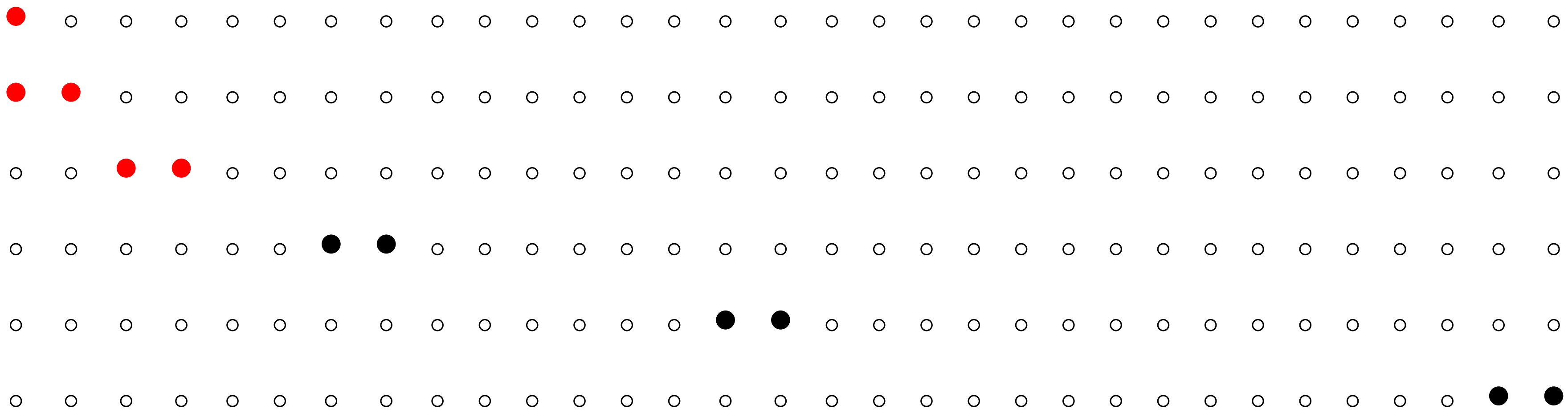
Computing the expansion of A^{-1}

High-order component lifting [Storjohann 02]



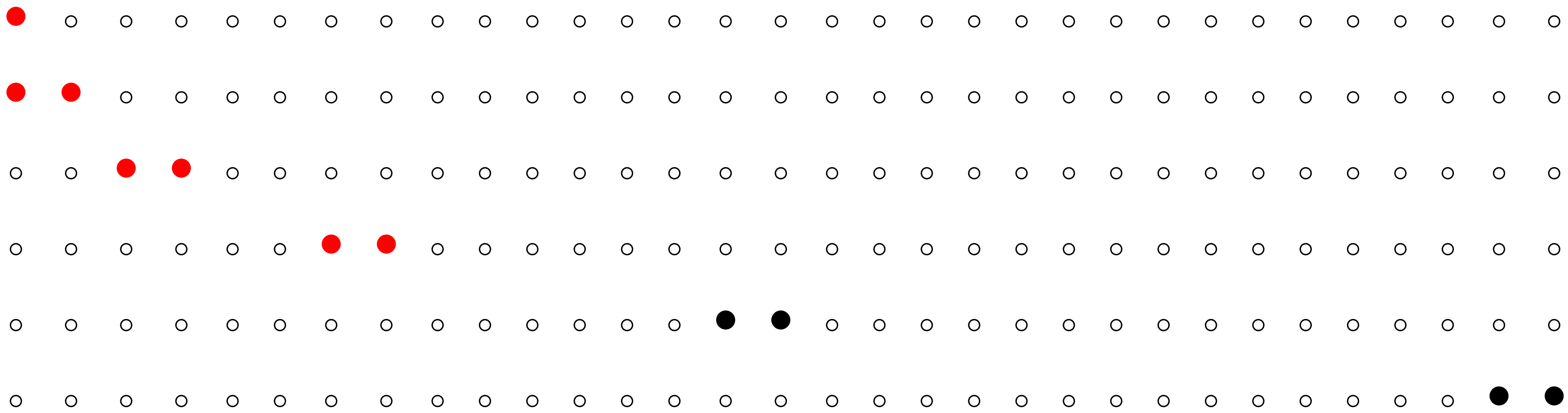
Computing the expansion of A^{-1}

High-order component lifting [Storjohann 02]



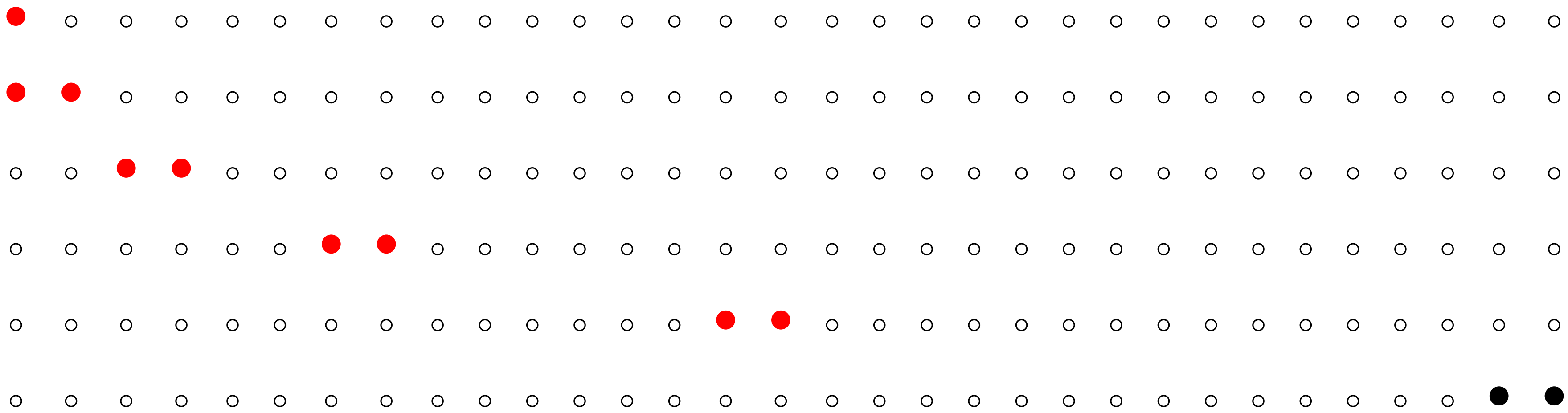
Computing the expansion of A^{-1}

High-order component lifting [Storjohann 02]



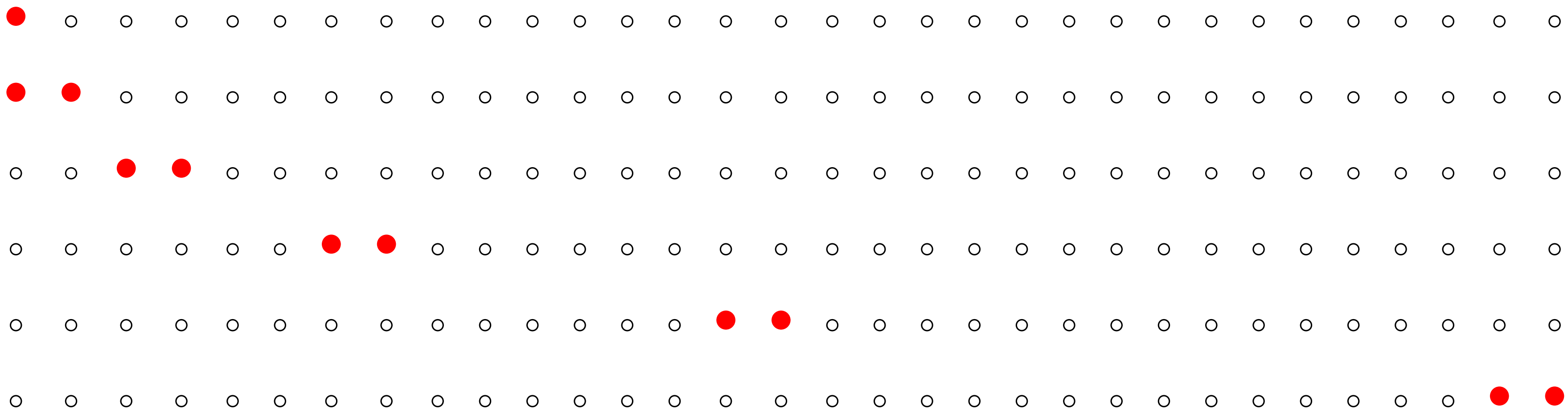
Computing the expansion of A^{-1}

High-order component lifting [Storjohann 02]



Computing the expansion of A^{-1}

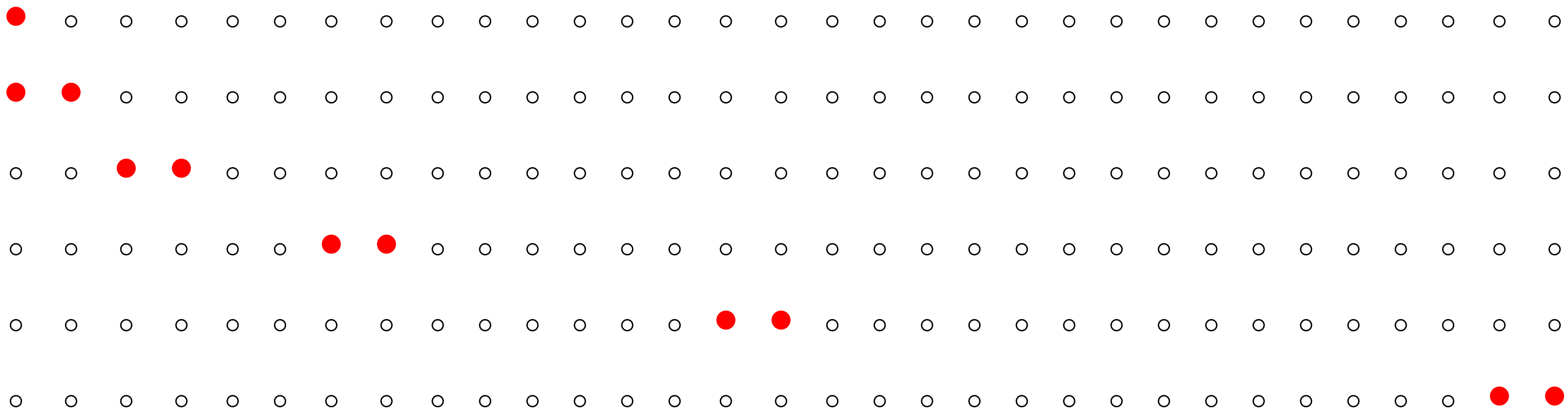
High-order component lifting [Storjohann 02]



$$A^{-1} = (A^{-1}(x) \bmod x^k) + \quad x^{k+1} R A^{-1}$$

Computing the expansion of A^{-1}

High-order component lifting [Storjohann 02]



$$
\begin{aligned}
A^{-1} &= (A^{-1}(x) \bmod x^k) + \quad x^{k+1} R A^{-1} \\
&= (A^{-1}(x) \bmod x^k) + \quad x^{k+1} R_L (A^{-1})_L + x^{2k-1} {\color{red} R_H (A^{-1})_H}
\end{aligned}
$$

Organization of the talk

I - Algebraic *versus* bit complexity.

II - Reductions between problems and target complexity.

III - Polynomial matrix computations.

IV - Integer matrix computations.

Conclusion

$A \in \mathbb{Z}^{n \times n}$,
MM$(n, \log \|A\|) = O\tilde{~}(n^{\omega+1} \log \|A\|)$ or $O(n^3 \log^2 \|A\|)$.

Nota. More difficult than the polynomial case.

▷ **Integer determinant**

[Storjohann 03]

▷ **Integer characteristic polynomial**

[Kaltofen & Villard 03]

IV-1/ Matrix determinant

$b = \log^{\alpha} \|A\|$

$O\tilde{}(n^{\omega+1} \log \|A\|) \leq \quad n^{3.38} b$ [Classical approaches]

IV-1/ Matrix determinant

$$b = \log^{\alpha} \|A\|$$

$$O\tilde{}(n^{\omega+1} \log \|A\|) \leq \quad n^{3.38}b \text{ [Classical approaches]}$$
$$n^{3.19}b \text{ [Eberly, Giesbrecht, Villard 2000]}$$

IV-1/ Matrix determinant

$b = \log^{\alpha} \|A\|$

$$O\tilde{}(n^{\omega+1} \log \|A\|) \leq \quad n^{3.38}b \text{ [Classical approaches]}$$

$$n^{3.19}b \text{ [Eberly, Giesbrecht, Villard 2000]}$$

$$n^{3.03}b \text{ [Kaltofen 1992/2000]}$$

IV-1/ Matrix determinant

$b = \log^{\alpha} \|A\|$

$$O\tilde{}(n^{\omega+1} \log \|A\|) \leq \begin{array}{l} n^{3.38} b \text{ [Classical approaches]} \\ n^{3.19} b \text{ [Eberly, Giesbrecht, Villard 2000]} \\ n^{3.03} b \text{ [Kaltofen 1992/2000]} \\ n^{2.7} b \text{ [Kaltofen, Villard 2001]} \end{array}$$

IV-1/ Matrix determinant

$$b = \log^{\alpha} \|A\|$$

$$\begin{aligned}
O\tilde{}(n^{\omega+1} \log \|A\|) \leq \quad & n^{3.38} b \text{ [Classical approaches]} \\
& n^{3.19} b \text{ [Eberly, Giesbrecht, Villard 2000]} \\
& n^{3.03} b \text{ [Kaltofen 1992/2000]} \\
& n^{2.7} b \text{ [Kaltofen, Villard 2001]} \\
O\tilde{}(n^{\omega} \log \|A\|) \leq \quad & n^{2.38} b \text{ [Storjohann 2002]}
\end{aligned}$$

IV-1/ Matrix determinant

$b = \log^{\alpha} \|A\|$

$$
\begin{aligned}
O\tilde{}(n^{\omega+1} \log \|A\|) \leq \quad & n^{3.38} b \text{ [Classical approaches]} \\
& n^{3.19} b \text{ [Eberly, Giesbrecht, Villard 2000]} \\
& n^{3.03} b \text{ [Kaltofen 1992/2000]} \\
& n^{2.7} b \text{ [Kaltofen, Villard 2001]} \\
O\tilde{}(n^{\omega} \log \|A\|) \leq \quad & n^{2.38} b \text{ [Storjohann 2002]}
\end{aligned}
$$

Nota. Apparently no progress on this side of $O\tilde{}(n^{\omega+1} \log \|A\|)$ bit operations for matrix inversion.

IV-2/ Integer characteristic polynomial

Iterated powers or Krylov approach and Chinese remaindering,

$\rightsquigarrow O\tilde{}(n^{\omega+1} \log \|A\|)$ bit operations
Las Vegas randomized

[Giesbrecht & Storjohann 02]

Via algebraic complexity without divisions

[Kaltofen & Villard 01-03]

Theorem. The determinant of any matrix A in $\mathsf{R}^{n\times n}$ can be computed with $O\tilde{}(n^{3+1/5})$ or $n^{2.7}$ ring operations.

Theorem. The characteristic polynomial of any matrix A in $\mathsf{Z}^{n\times n}$ can be computed by a randomized Monte Carlo algorithm with $O\tilde{}(n^{3+1/5}\log\|A\|)$ or $O\tilde{}(n^{2.7\log\|A\|})$ bit operations.

To postpone the size increase?

Exemple:

x and y two vectors in $\mathbb{Z}^n$ with constant entries,

c a large constant,

Compute $c \cdot x^t \cdot y$?

$\longrightarrow$ Solution 1. $\quad c \cdot x^t$ then $(c \cdot x^t) \cdot y \quad$ Cost: $O(\log |c|^2)$.

To postpone the size increase?

Exemple:

x and y two vectors in $\mathbb{Z}^n$ with constant entries,

c a large constant,

Compute $c \cdot x^t \cdot y$?

$\longrightarrow$ Solution 1. $c \cdot x^t$ then $(c \cdot x^t) \cdot y$ Cost: $O(\log^2 |c|)$.

$\longrightarrow$ Solution 2. $x^t \cdot y$ then $c \cdot (x^t \cdot y)$ Cost: $O(\log |c|)$.

Integer determinant and characteristic polynomial,

Ingredients:

- Elimination of divisions [Strassen 73]
- Baby step/giant step [Kaltofen 92]
- Krylov/Lanczos [Wiedemann 86]
- Block Krylov/Lanczos [Coppersmith 86, Villard 97]
- Multifactor Hensel lifting [Sorjohann 00]

1. Computation of $u^t v, u^t A v, u^t A^2 v, \ldots, u^t A^{2n} v$
2. Computation of the **minimum polynomial**

From scalar polynomials in $\mathsf{K}[x]$ to **matrix polynomials** in

$$(\mathsf{K}[x])^{m\times m} = (\mathsf{K}^{m\times m})[x].$$

Matrix minimum polynomial,

$U, V \in \mathsf{K}^{n\times m}, \quad U^tV,\ U^tAV,\ U^tA^2V, \ldots,\ U^tA^{2n/m}V$

$$\downarrow$$

$$F(x) = x^d I + x^{d-1}F_{d-1} + \ldots + F_0 \in \mathsf{K}[x]^{m\times m},$$

$$A, n\times n \quad \rightarrow \quad F(x), m\times m \quad \rightarrow \quad \det F(x)$$

▷ **Postpone the size increase:** less powers of A.

Nota. First gain by using baby steps/giant steps, additional gain with the minimum matrix polynomial.

Conclusion.

des resultats de l'expose: dire ce qu'il reste a faire pour les finir

inversion cas general ?

reunifier les complexites

pgcd recursif

without divisions

inverse cas general

MMprime

all problems had the "same" complexity, and now lien + numerique avec pepier Demmel. Eg complexity calcul du conditionnement?

recapituler le meilleurs exposants connus

certified rank? char poly? transfor matrices ?

revenir au titre (how does cost?)

matrices creuses, structurees

Linbox

floating points / accuracy

Suppression du facteur n / Modle binaire sur $\mathbb{Z}$ (en $O\tilde{}$)

	Facteur	Total
Dterminant (LV)	$n^{1/5}$	$n^{2.7} \log \|A\|$
Storjohann 2003	$\log^{\alpha} n$	$n^{\omega} \log \|A\|^*$
Forme de Smith (LV)	$\log^{\alpha} n$	$n^{\omega} \log \|A\|^*$
Polynme caractristique (MC) + Forme normale de Frobenius	$n^{1/5}$	$n^{2.7} \log \|A\|$

Suppression du facteur n / Modle algbrique sur K$[x]$ (en $\tilde{O}$)

	Facteur	Total
Dterminant, Smith (LV)[1]	$\log^{\alpha} n$	$n^{\omega} d^{*}$
Déterminant sans division[2]	$n^{1/5}$	$n^{2.7}$
Rduction en colonnes, pgcd (LV)[3]	$\log^{\beta} n$	$n^{\omega} d^{*}$
Inversion[4] (SLP)	$\log^{\gamma} n$	$n^{3} d^{*}$

1 [Sto2002], 2 [KaVi2001-03], 3 [GiJeVi2003], 4 [JeVi2002].