On the complexity of computing characteristic polynomials

joint work with P. Karpman, V. Neiger, H. Signargout, A. Storjohann and G. Villard

Clément Pernet
Grenoble INP – UGA, France

Recent Trends in Computer Algebra: Fundamental Algorithms and Algorithmic complexity,
Insitut Henri Poincarré, Paris. September 28, 2023

Introduction

Introduction

Context

- Exact linear algebra: over a field $\mathbb{K}$, (sometimes a ring R or $\mathbb{Z}$
- Mostly algebraic complexity, counting field operations), (somtimes bitcomplexity)

Problem

Given $\mathbf{M} \in \mathbb{K}^{m \times m}$, *compute* $\chi_{\mathbf{M}} = \det(x\mathbf{I}_m - \mathbf{M}) \in \mathbb{K}[x]$.

Applications

- Matrix invariants (eigenvalues, invariant factors), test for similarity
- Invariant subspace decomposition
- Gröbner basis (change of ordering)
- Modular forms (action of the Hecke Operator)

A challenging complexity problem: dense linear algebra over a field

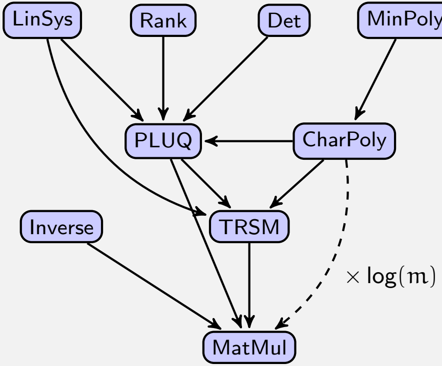
Dense linear algebra: reductions of most problems to matrix multiplication

ω: a feasible exponent of MatMul over $\mathbb{K}$: $m \times m$ by $m \times m$ in $O(m^\omega)$

A challenging complexity problem: dense linear algebra over a field

Dense linear algebra: reductions of most problems to matrix multiplication

ω: a feasible exponent of MatMul over $\mathbb{K}$: $m \times m$ by $m \times m$ in $O(m^\omega)$

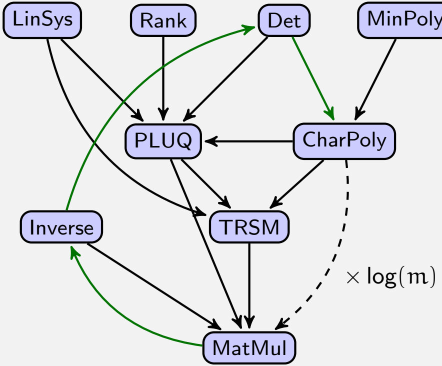


$$\left.\begin{array}{c}\text{LinSys}\\ \text{Det}\\ \text{Rank}\\ \text{PLUQ}\\ \text{TRSM}\\ \text{Inverse}\end{array}\right\} = O(\text{MatMul})$$

A challenging complexity problem: dense linear algebra over a field

Dense linear algebra: reductions of most problems to matrix multiplication

ω: a feasible exponent of MatMul over $\mathbb{K}$: $m \times m$ by $m \times m$ in $O(m^\omega)$



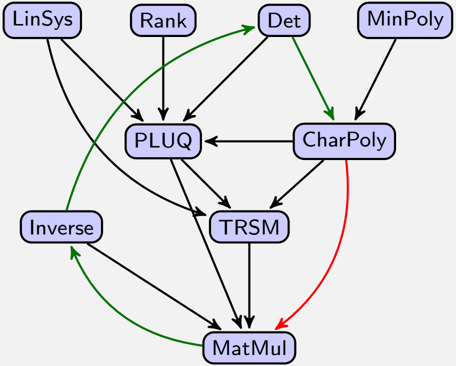
$$\left.\begin{array}{c}\text{LinSys}\\ \text{Det}\\ \text{Rank}\\ \text{PLUQ}\\ \text{TRSM}\\ \text{Inverse}\end{array}\right\} = O(\text{MatMul})$$

$$\text{MatMul} = O\begin{pmatrix}\text{Det},\\ \text{PLUQ},\\ \text{CharPoly},\\ \text{Inverse}\end{pmatrix}$$

A challenging complexity problem: dense linear algebra over a field

Dense linear algebra: reductions of most problems to matrix multiplication

ω: a feasible exponent of MatMul over $\mathbb{K}$: $m \times m$ by $m \times m$ in $O(m^\omega)$



$$\left.\begin{array}{c}\text{LinSys}\\ \text{Det}\\ \text{Rank}\\ \text{PLUQ}\\ \text{TRSM}\\ \text{Inverse}\end{array}\right\} = O(\text{MatMul})$$

$$\text{MatMul} = O\left(\begin{array}{c}\text{Det,}\\ \text{PLUQ,}\\ \text{CharPoly,}\\ \text{Inverse}\end{array}\right)$$

Challenge: CharPoly $= O(\text{MatMul})$?

A challenging complexity problem: sparse/structured linear algebra over a field

Sparse matrices via black-box methods

- Only operation: Mat $\times$ Vect → Cost: E

A challenging complexity problem: sparse/structured linear algebra over a field

Sparse matrices via black-box methods

- Only operation: Mat $\times$ Vect → Cost: E
- [Wiedeman 86]: Minpoly$(m) = O(mE + m^2)$

A challenging complexity problem: sparse/structured linear algebra over a field

Sparse matrices via black-box methods

- Only operation: Mat $\times$ Vect → Cost: E
- [Wiedeman 86]: $\text{Minpoly}(m) = O(mE + m^2)$
- [Villard 01]: $\text{Charpoly}(m) = O(m^{1.5}E + m^{2.5})$

A challenging complexity problem: sparse/structured linear algebra over a field

Sparse matrices via black-box methods

- Only operation: Mat $\times$ Vect → Cost: E
- [Wiedeman 86]: $\text{Minpoly}(m) = O(mE + m^2)$
- [Villard 01]: $\text{Charpoly}(m) = O(m^{1.5}E + m^{2.5})$
- [Villard 03]: $\text{Charpoly}(m) = O(m^{2.36})$ when $E = O(m)$

Challenge: Blackbox methods $\ll$ Dense methods

A challenging complexity problem: sparse/structured linear algebra over a field

Sparse matrices via black-box methods

- Only operation: Mat $\times$ Vect → Cost: E
- [Wiedeman 86]: $\mathrm{Minpoly}(m) = O(mE + m^2)$
- [Villard 01]: $\mathrm{Charpoly}(m) = O(m^{1.5}E + m^{2.5})$
- [Villard 03]: $\mathrm{Charpoly}(m) = O(m^{2.36})$ when $E = O(m)$

Challenge: Blackbox methods $\ll$ Dense methods

Matrices with rank displacement structure

- Toeplitz, Hankel, Cauchy, Vandermonde, etc
- Generalization: $\mathrm{rank}(\Delta(\mathbf{A})) = \alpha \ll m$
- [Bostan-Jeannerod-Mouilleron-Schost 17] LinSys, Det, Inverse $= O^{(\sim)}(m\alpha^{\omega-1})$
- until recently: $\mathrm{Charpoly} = O(m^2\alpha^{\omega-1})$

Challenge: Charpoly in sub-quadratic time in m ?

Outline

Via Krylov methods

Krylov Methods

Iterates of one vector

For a vector $\mathbf{v} \in \mathbb{K}^m$, let

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

If d is maximal s.t. $\mathbf{K}$ full-rank , then

$$\mathbf{A}\mathbf{K} = \mathbf{K} \underbrace{\begin{bmatrix} 0 & & & p_0 \\ 1 & & & p_1 \\ & \ddots & & \vdots \\ & & 1 & p_{m-1} \end{bmatrix}}_{C_P}$$

and $P = X^m - p_{m-1}X^{m-1} - \cdots - p_0$ is the minpoly of $\mathbf{v}$ wrt. $\mathbf{A}$.

Krylov Methods

Iterates of one vector

For a vector $\mathbf{v} \in \mathbb{K}^m$, let

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

If d is maximal s.t. $\mathbf{K}$ full-rank and $d = m$, then

$$\mathbf{K}^{-1}\mathbf{A}\mathbf{K} = \underbrace{\begin{bmatrix} 0 & & & p_0 \\ 1 & & & p_1 \\ & \ddots & & \vdots \\ & & 1 & p_{m-1} \end{bmatrix}}_{C_P}$$

and $P = X^m - p_{m-1}X^{m-1} - \dots - p_0$ is the minpoly of $\mathbf{v}$ wrt. $\mathbf{A}$.

then $\chi_A = P$

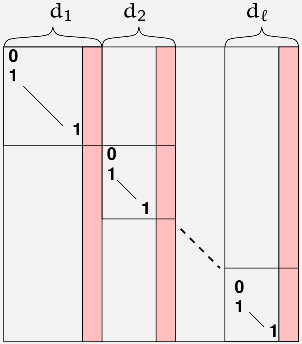
Krylov Methods

Iterates of multiple vectors

For a family of vectors $\mathbf{v}_1, \dots, \mathbf{v}_\ell \in \mathbb{K}^m$, let

$$\mathbf{K} = \left[\begin{array}{cccc|ccc|c|ccc} \mathbf{v}_1 & \mathbf{A}\mathbf{v}_1 & \dots & \mathbf{A}^{\mathrm{d}_1 - 1}\mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{A}^{\mathrm{d}_2 - 1}\mathbf{v}_2 & \dots & \mathbf{v}_\ell & \dots & \mathbf{A}^{\mathrm{d}_\ell - 1}\mathbf{v}_\ell \end{array}\right].$$

If $\mathbf{K}$ is invertible then $\mathbf{K}^{-1}\mathbf{A}\mathbf{K} =$

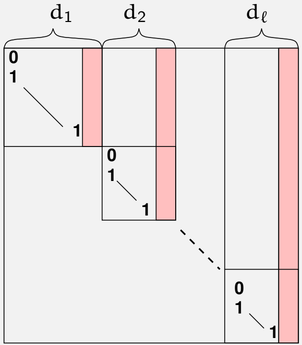


Iterates of multiple vectors

For a family of vectors $\mathbf{v}_1, \dots, \mathbf{v}_\ell \in \mathbb{K}^m$, let

$$\mathbf{K} = \left[\begin{array}{cccc|ccc|c|ccc} \mathbf{v}_1 & \mathbf{A}\mathbf{v}_1 & \dots & \mathbf{A}^{d_1-1}\mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{A}^{d_2-1}\mathbf{v}_2 & \dots & \mathbf{v}_\ell & \dots & \mathbf{A}^{d_\ell-1}\mathbf{v}_\ell \end{array} \right].$$

If $\mathbf{K}$ is invertible and $(d_1, \dots, d_\ell)$ is lexico. maximal then $\mathbf{K}^{-1}\mathbf{A}\mathbf{K} =$



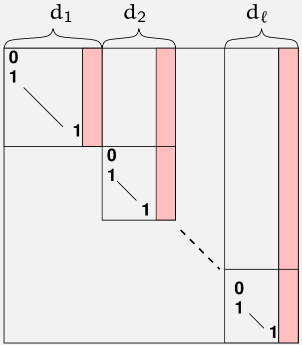
Krylov Methods

Iterates of multiple vectors

For a family of vectors $\mathbf{v}_1, \dots, \mathbf{v}_\ell \in \mathbb{K}^m$, let

$$\mathbf{K} = \left[\begin{array}{cccc|ccc|c|ccc} \mathbf{v}_1 & \mathbf{A}\mathbf{v}_1 & \dots & \mathbf{A}^{d_1-1}\mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{A}^{d_2-1}\mathbf{v}_2 & \dots & \mathbf{v}_\ell & \dots & \mathbf{A}^{d_\ell-1}\mathbf{v}_\ell \end{array}\right].$$

If $\mathbf{K}$ is invertible and $(d_1, \dots, d_\ell)$ is lexico. maximal then $\mathbf{K}^{-1}\mathbf{A}\mathbf{K} =$



Then

$$\chi_{\mathbf{A}} = P_1 \times \cdots \times P_\ell$$

where C_{P_i} is the i-th diagonal block

Approach 1: computing the Krylov matrix

1. Compute $\mathbf{K} = \left[\begin{array}{cccc|ccc|c|ccc} \mathbf{v}_1 & \mathbf{A}\mathbf{v}_1 & \dots & \mathbf{A}^{d_1-1}\mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{A}^{d_2-1}\mathbf{v}_2 & \dots & \mathbf{v}_\ell & \dots & \mathbf{A}^{d_\ell-1}\mathbf{v}_\ell \end{array}\right]$.
2. Compute $H = \mathbf{K}^{-1}\mathbf{A}\mathbf{K}$ ➔ $O(m^\omega)$

Approach 1: computing the Krylov matrix

1. Compute $\mathbf{K} = \left[\begin{array}{cccc|ccc|c|ccc} \mathbf{v}_1 & \mathbf{A}\mathbf{v}_1 & \dots & \mathbf{A}^{d_1-1}\mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{A}^{d_2-1}\mathbf{v}_2 & \dots & \mathbf{v}_\ell & \dots & \mathbf{A}^{d_\ell-1}\mathbf{v}_\ell \end{array} \right]$.
2. Compute $H = \mathbf{K}^{-1}\mathbf{A}\mathbf{K}$ ➔ $O(m^\omega)$

Iteratively

Using m Matrix-Vector products ($+$ Gaussian elimination) ➔ $O(m^3)$

Approach 1: computing the Krylov matrix

1. Compute $\mathbf{K} = \left[\mathbf{v}_1 \quad \mathbf{A}\mathbf{v}_1 \quad \dots \quad \mathbf{A}^{d_1-1}\mathbf{v}_1 \;\middle|\; \mathbf{v}_2 \quad \dots \quad \mathbf{A}^{d_2-1}\mathbf{v}_2 \;\middle|\; \dots \;\middle|\; \mathbf{v}_\ell \quad \dots \quad \mathbf{A}^{d_\ell-1}\mathbf{v}_\ell \right]$.
2. Compute $\mathrm{H} = \mathbf{K}^{-1}\mathbf{A}\mathbf{K}$ → $O(m^\omega)$

Iteratively

Using m Matrix-Vector products (+ Gaussian elimination) → $O(m^3)$

[Keller-Gehrig 85]'s iteration (adaptation of square & multiply)

- Iteratively compute ($\log_2 m$ iterations)
  - $\mathbf{K}_0 \leftarrow \begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_\ell \end{bmatrix}$
  - $\mathbf{K}_1 \leftarrow \begin{bmatrix} \mathbf{K}_0 & \mathbf{A}\mathbf{K}_0 \end{bmatrix}$
  - $\dots$
  - $\mathbf{K}_i \leftarrow \begin{bmatrix} \mathbf{K}_{i-1} & \mathbf{A}^{2^i}\mathbf{K}_{i-1} \end{bmatrix}$
- Interleave Gaussian elimination to discard linearly dependent columns
  → each $\mathbf{K}_i$ has no more than m columns

Approach 1: computing the Krylov matrix

1. Compute $\mathbf{K} = \left[\begin{array}{cccc|ccc|c|ccc} \mathbf{v}_1 & \mathbf{A}\mathbf{v}_1 & \dots & \mathbf{A}^{d_1-1}\mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{A}^{d_2-1}\mathbf{v}_2 & \dots & \mathbf{v}_\ell & \dots & \mathbf{A}^{d_\ell-1}\mathbf{v}_\ell \end{array}\right]$.
2. Compute $H = \mathbf{K}^{-1}\mathbf{A}\mathbf{K}$ → $O(m^\omega)$

Iteratively

Using m Matrix-Vector products ($+$ Gaussian elimination) → $O(m^3)$

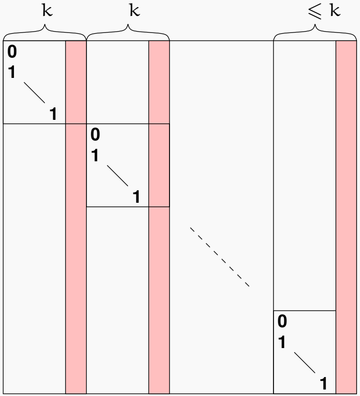
[Keller-Gehrig 85]'s iteration (adaptation of square & multiply)

- Iteratively compute ($\log_2 m$ iterations)
  - $\mathbf{K}_0 \leftarrow \begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_\ell \end{bmatrix}$
  - $\mathbf{K}_1 \leftarrow \begin{bmatrix} \mathbf{K}_0 & \mathbf{A}\mathbf{K}_0 \end{bmatrix}$
  - $\dots$
  - $\mathbf{K}_i \leftarrow \begin{bmatrix} \mathbf{K}_{i-1} & \mathbf{A}^{2^i}\mathbf{K}_{i-1} \end{bmatrix}$
- Interleave Gaussian elimination to discard linearly dependent columns
  → each $\mathbf{K}_i$ has no more than m columns

→ $O(m^\omega \log m)$

Approach 2: avoid computing the Krylov matrix [P.-Storjohann 07]

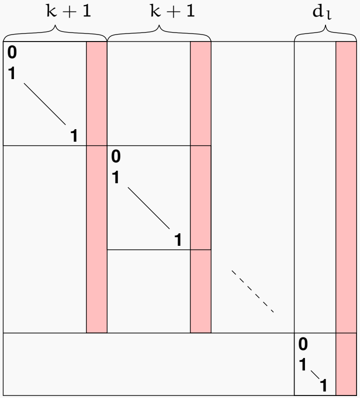
k-shifted form:



- Any matrix is in 1-shifted form

Approach 2: avoid computing the Krylov matrix [P.-Storjohann 07]

$k + 1$-shifted form:



- Any matrix is in 1-shifted form

Approach 2: avoid computing the Krylov matrix

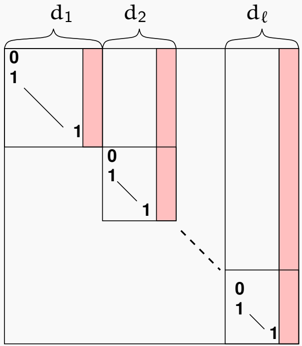
- Compute iteratively from 1-shifted form to d_1-shifted form

Approach 2: avoid computing the Krylov matrix

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

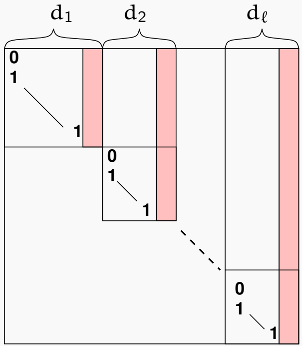
Approach 2: avoid computing the Krylov matrix

- 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:



Approach 2: avoid computing the Krylov matrix

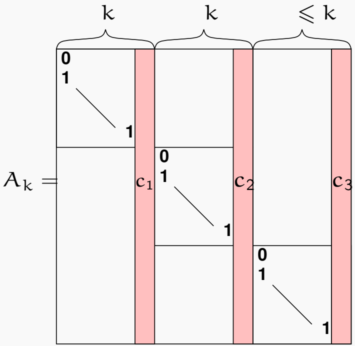
- Compute iteratively from 1-shifted form to d_1-shifted form
- each diagonal block appears in the increasing degree
- until the shifted Hessenberg form is obtained:



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

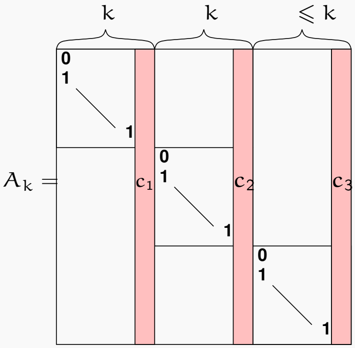
Approach 2: avoid computing the Krylov matrix

for any k-shifted form

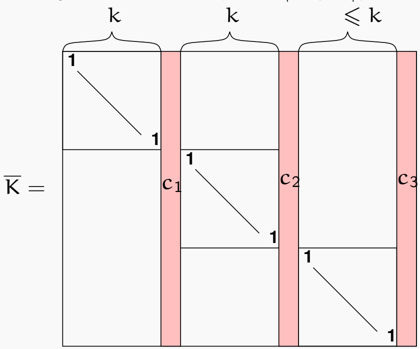


Approach 2: avoid computing the Krylov matrix

for any k-shifted form

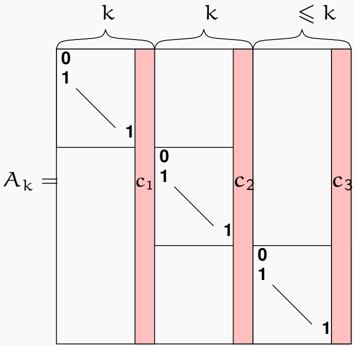


compute the $m \times (m + \lceil m/k \rceil)$ matrix

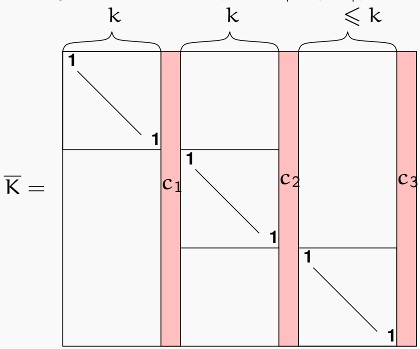


Approach 2: avoid computing the Krylov matrix

for any k-shifted form



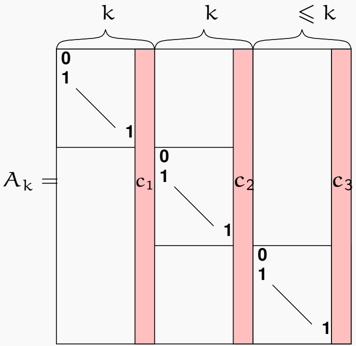
compute the $m \times (m + \lceil m/k \rceil)$ matrix



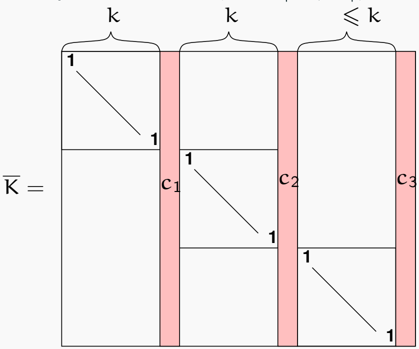
- $\mathbf{K} =$ first linearly indep. cols of $\overline{\mathbf{K}}$

Approach 2: avoid computing the Krylov matrix

for any k-shifted form



compute the $m \times (m + \lceil m/k \rceil)$ matrix

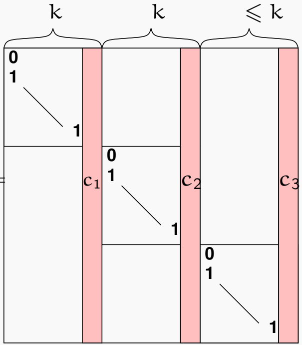


- $\mathbf{K} =$ first linearly indep. cols of $\overline{\mathbf{K}}$
- $\mathbf{A}_{k+1} = \mathbf{K}^{-1}\mathbf{A}_k\mathbf{K}$ in $O(m(\frac{m}{k})^{\omega-1})$

Approach 2: avoid computing the Krylov matrix

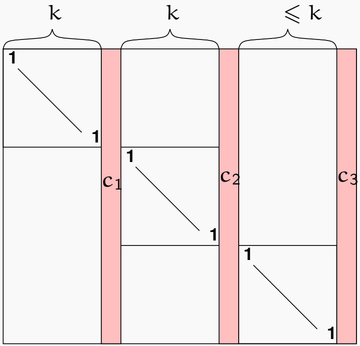
for any k-shifted form

$A_k =$



compute the $m \times (m + \lceil m/k \rceil)$ matrix

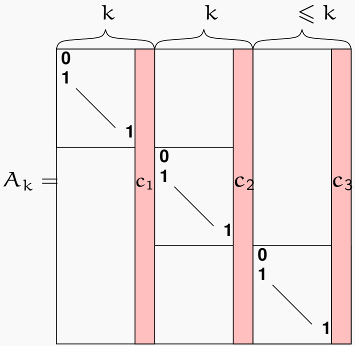
$\overline{K} =$



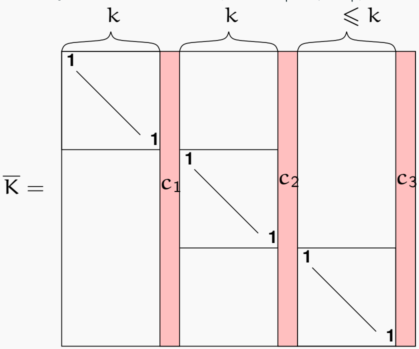
- $\mathbf{K} =$ first linearly indep. cols of $\overline{\mathbf{K}}$
- $\mathbf{A}_{k+1} = \mathbf{K}^{-1}\mathbf{A}_k\mathbf{K}$ in $O(m(\frac{m}{k})^{\omega-1})$
- Overall cost $T(m) = O(m^\omega \sum_{k=1}^{m} \frac{1}{k^{\omega-1}}) = O(m^\omega)$

Approach 2: avoid computing the Krylov matrix

for any k-shifted form

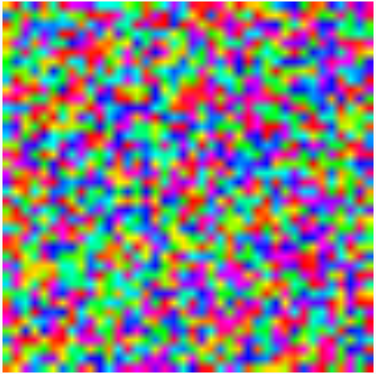


compute the $m \times (m + \lceil m/k \rceil)$ matrix

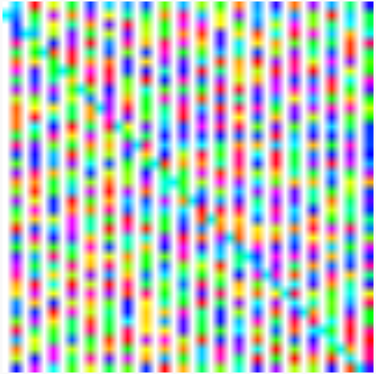


- $\mathbf{K} =$ first linearly indep. cols of $\overline{\mathbf{K}}$ — w.h.p. $\mathbf{K} =$ first m cols of $\overline{\mathbf{K}}$
- $\mathbf{A}_{k+1} = \mathbf{K}^{-1}\mathbf{A}_k\mathbf{K}$ in $O(m(\frac{m}{k})^{\omega-1})$ — w.h.p $\mathbf{A}_{k+1}$ is $(k+1)$-shifted
- Overall cost $T(m) = O(m^\omega \sum_{k=1}^{m} \frac{1}{k^{\omega-1}}) = O(m^\omega)$ — Las-Vegas probabilistic

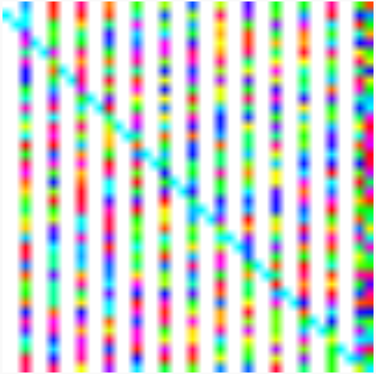
Example



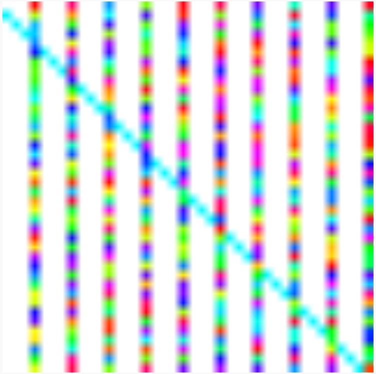
Example



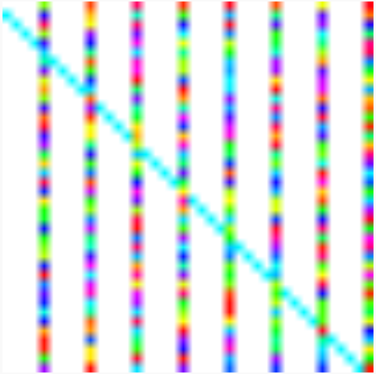
Example



Example

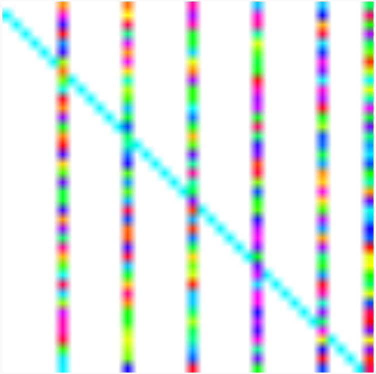


Example

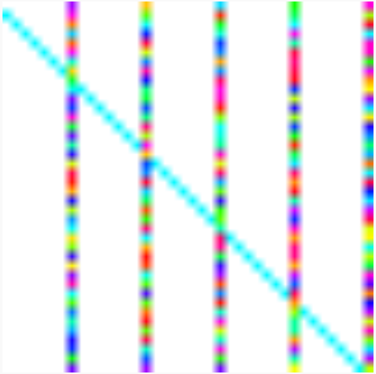


Example

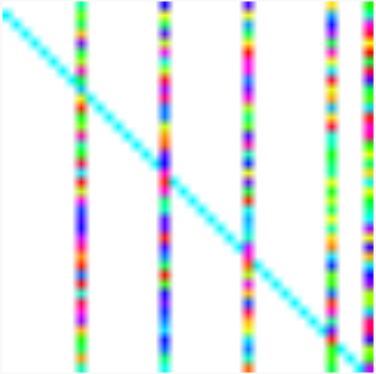
Example



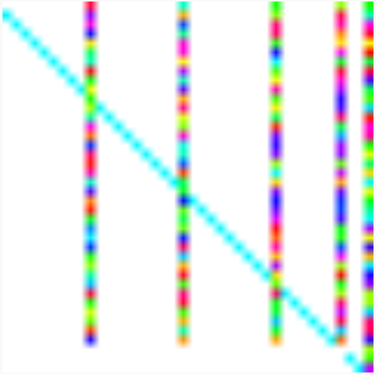
Example



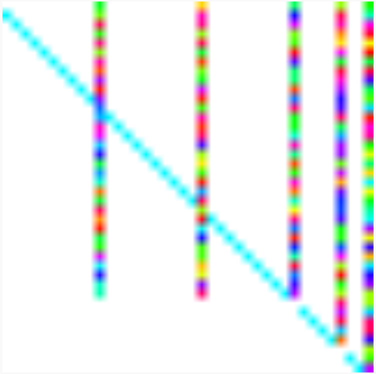
Example



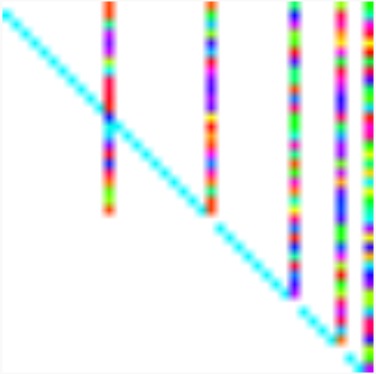
Example



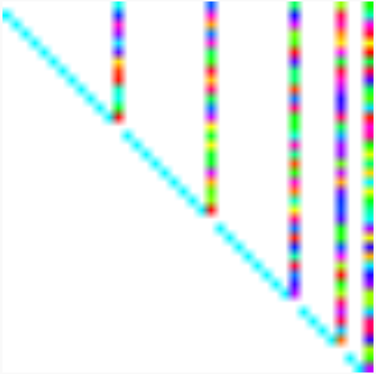
Example



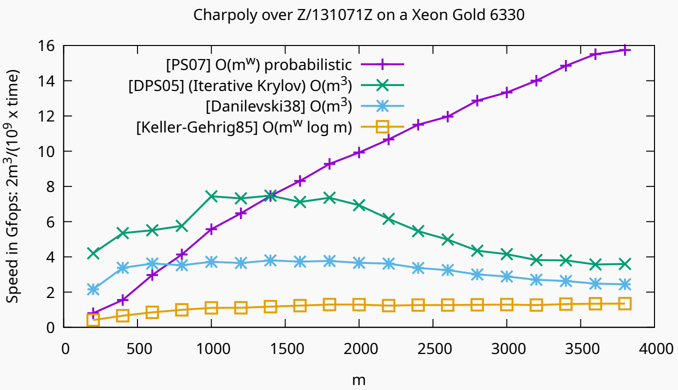
Example



Example



Algorithms in practice



- implementations in the `fflas-ffpack` library[1]: finite field dense linear algebra

[1] https://github.com/linbox-team/fflas-ffpack

Via polynomial matrix arithmetic

Charpoly via $\mathbb{K}[x]$-linear algebra

Determinant of a matrix $\mathbf{A} \in \mathbb{K}[x]^{m \times m}$ of degree d $\qquad d = 1$

Evaluation-Interpolation: [folklore] $\qquad O(m^{\omega+1})$

at $\sim md$ points: requires large enough field

Diagonalization (Smith form)**:** [Storjohann 2003] $\qquad O(m^{\omega} \log(m)^2)$

Las Vegas randomized + additional logs for small fields

Partial triangularization:

- Iterative [Mulders-Storjohann 2003] $\qquad O(m^3)$
  via weak Popov form computations
- Divide and conquer, generic [Giorgi-Jeannerod-Villard 2003] $\qquad O(m^{\omega})$
  diagonal of Hermite form must be $1, \ldots, 1, \det(\mathbf{A})$
- Divide and conquer [Neiger-Labahn-Zhou 2017] $\qquad \tilde{O}(m^{\omega})$
  logarithmic factors in m and d

Partial block triangularization

[Mulders-Storjohann 2003, Giorgi-Jeannerod-Villard 2003, Zhou 2012, Neiger-Labahn-Zhou 2017]

Triangularization of $m \times m$ matrix $\mathbf{A}$ using $m/2 \times m/2$ blocks

not computed

$$\begin{bmatrix} * & * \\ \mathbf{K}_1 & \mathbf{K}_2 \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & * \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$$

row basis of $\left[\begin{smallmatrix} \mathbf{A}_1 \\ \mathbf{A}_3 \end{smallmatrix}\right]$

kernel basis of $\left[\begin{smallmatrix} \mathbf{A}_1 \\ \mathbf{A}_3 \end{smallmatrix}\right]$

$\mathbf{K}_1\mathbf{A}_2 + \mathbf{K}_2\mathbf{A}_4$

Property: $\det(\mathbf{A}) = \det(\mathbf{R})\det(\mathbf{B})$

Generic case without log factor

[Mulders-Storjohann 2003, Giorgi-Jeannerod-Villard 2003, Zhou 2012, Neiger-Labahn-Zhou 2017]

Triangularization of $m \times m$ matrix $\mathbf{A}$ using $m/2 \times m/2$ blocks

not computed

$$\begin{bmatrix} * & * \\ \mathbf{K}_1 & \mathbf{K}_2 \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & * \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$$

row basis of $\left[\begin{smallmatrix} \mathbf{A}_1 \\ \mathbf{A}_3 \end{smallmatrix}\right]$

kernel basis of $\left[\begin{smallmatrix} \mathbf{A}_1 \\ \mathbf{A}_3 \end{smallmatrix}\right]$

$\mathbf{K}_1\mathbf{A}_2 + \mathbf{K}_2\mathbf{A}_4$

Property: $\det(\mathbf{A}) = \det(\mathbf{R}) \det(\mathbf{B})$

Generic input $\Rightarrow \det(\mathbf{A})$ **without** $\log(m)$ **[Giorgi-Jeannerod-Villard 2003]**

$\mathbf{A}_1$ and $\mathbf{A}_3$ are coprime $\Rightarrow \mathbf{R} = \mathbf{I}_{m/2} \Rightarrow \det(\mathbf{A}) = \det(\mathbf{B})$

- Compute kernel $[\mathbf{K}_1 \;\; \mathbf{K}_2]$; deduce $\mathbf{B}$ by MatMul $O(m^\omega M'(d))$
- Recursively, compute $\det(\mathbf{B})$, return it

$\mathbf{A}$ and $[\mathbf{K}_1 \;\; \mathbf{K}_2]$ have degree $d \Rightarrow \mathbf{B}$ has degree $2d$: controlled total degree

General case with log factor

[Mulders-Storjohann 2003, Giorgi-Jeannerod-Villard 2003, Zhou 2012, Neiger-Labahn-Zhou 2017]

Triangularization of $m \times m$ matrix $\mathbf{A}$ using $m/2 \times m/2$ blocks

not computed

$$\begin{bmatrix} * & * \\ \mathbf{K}_1 & \mathbf{K}_2 \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & * \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$$

row basis of $\left[\begin{smallmatrix} \mathbf{A}_1 \\ \mathbf{A}_3 \end{smallmatrix}\right]$

kernel basis of $\left[\begin{smallmatrix} \mathbf{A}_1 \\ \mathbf{A}_3 \end{smallmatrix}\right]$

$\mathbf{K}_1\mathbf{A}_2 + \mathbf{K}_2\mathbf{A}_4$

Property: $\det(\mathbf{A}) = \det(\mathbf{R})\det(\mathbf{B})$

Matrix degree not controlled: degree of $\mathbf{B}$ up to $D = \sum \mathrm{rdeg}(\mathbf{A}) \leqslant md$

but controlled average row degree: at most $\frac{D}{m}$

General input $\Rightarrow$ $\det(\mathbf{A})$ **in** $O\tilde{~}(m^{\omega}\frac{D}{m})$ **[Labahn-Neiger-Zhou 2017]**

- Compute kernel $[\mathbf{K}_1 \;\; \mathbf{K}_2]$; deduce $\mathbf{B}$ by MatMul $\quad O(m^{\omega}M'(\frac{D}{m}))$
- Compute row basis $\mathbf{R}$ $\quad O\tilde{~}(m^{\omega}\frac{D}{m})$ with $\log(m)$
- Recursively, compute $\det(\mathbf{R})$ and $\det(\mathbf{B})$, return $\det(\mathbf{R})\det(\mathbf{B})$

Be lazy: if hard to compute, don't compute

[Mulders-Storjohann 2003, Giorgi-Jeannerod-Villard 2003, Zhou 2012, Neiger-Labahn-Zhou 2017]

Triangularization of $m \times m$ matrix $\mathbf{A}$ using $m/2 \times m/2$ blocks

not computed

$$\begin{bmatrix} * & * \\ \mathbf{K}_1 & \mathbf{K}_2 \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & * \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$$

row basis of $\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_3 \end{bmatrix}$

kernel basis of $\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_3 \end{bmatrix}$

$\mathbf{K}_1\mathbf{A}_2 + \mathbf{K}_2\mathbf{A}_4$

Property: $\det(\mathbf{A}) = \det(\mathbf{R})\det(\mathbf{B})$

Obstacle: removing log factors in row basis computation

$\Rightarrow$ solution: remove row basis computation

$$\begin{bmatrix} \mathbf{I}_{m/2} & \mathbf{0} \\ \mathbf{K}_1 & \mathbf{K}_2 \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$$

Property: $\det(\mathbf{A}) = \det(\mathbf{A}_1)\det(\mathbf{B}) / \det(\mathbf{K}_2)$

Further obstacles (brought by laziness)

$$\begin{bmatrix} \mathbf{I}_{m/2} & \mathbf{0} \\ \mathbf{K}_1 & \mathbf{K}_2 \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$$

Property: $\det(\mathbf{A}) = \det(\mathbf{A}_1)\det(\mathbf{B}) / \det(\mathbf{K}_2)$

- no $\log(m)$ in the computation of $\mathbf{A}_1$, $\mathbf{B}$, $\mathbf{K}_2$
- requires nonsingular $\mathbf{A}_1$, otherwise $\det(\mathbf{K}_2) = 0$
- 3 recursive calls in matrix size $m/2$ is 👍, but requires $\sum \mathrm{rdeg}(\mathbf{A}_1) \leqslant D/2$ otherwise degree control is too weak. (this implies $\sum \mathrm{rdeg}(\mathbf{K}_2) \leqslant D/2$)

Further obstacles (brought by laziness)

$$\begin{bmatrix} \mathbf{I}_{m/2} & \mathbf{0} \\ \mathbf{K}_1 & \mathbf{K}_2 \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$$

Property: $\det(\mathbf{A}) = \det(\mathbf{A}_1)\det(\mathbf{B}) / \det(\mathbf{K}_2)$

- no $\log(m)$ in the computation of $\mathbf{A}_1$, $\mathbf{B}$, $\mathbf{K}_2$
- requires nonsingular $\mathbf{A}_1$, otherwise $\det(\mathbf{K}_2) = 0$
- 3 recursive calls in matrix size $m/2$ is 👍, but requires $\sum \mathrm{rdeg}(\mathbf{A}_1) \leqslant D/2$ otherwise degree control is too weak. (this implies $\sum \mathrm{rdeg}(\mathbf{K}_2) \leqslant D/2$)

Solution: require $\mathbf{A}$ in weak Popov form (the characteristic matrix $\mathbf{A} = x\mathbf{I}_m - \mathbf{M}$ is in Popov form)

- implies $\mathbf{A}_1$ nonsingular and $\sum \mathrm{rdeg}(\mathbf{A}_1) \leqslant D/2$ up to easy transformations
- both $\mathbf{A}_1$ and $\mathbf{B}$ are also in weak Popov form $\Rightarrow$ suitable for recursive calls
- $\mathbf{K}_2$ is in "shifted reduced" form... find weak Popov $\mathbf{P}$ with same determinant

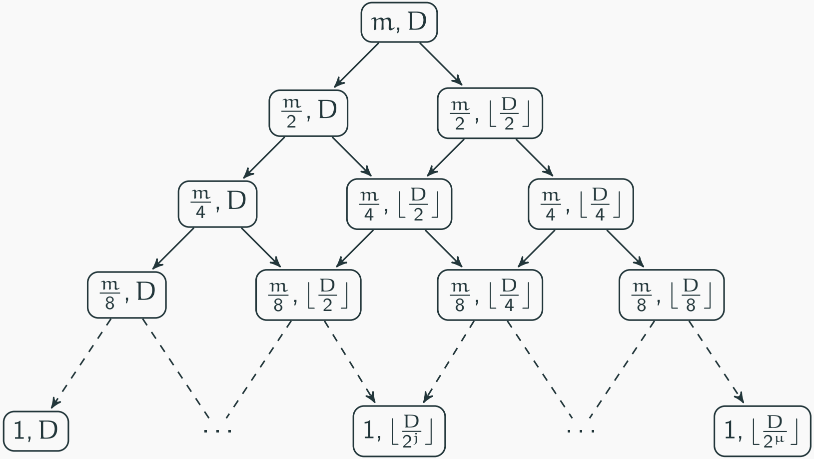
Complexity

$$\mathcal{C}(m, D) \leqslant 2\mathcal{C}\left(\frac{m}{2}, \left\lfloor \frac{D}{2} \right\rfloor\right) + \mathcal{C}\left(\frac{m}{2}, D\right) + O(m^{\omega} M'\left(\frac{D}{m}\right))$$

where: $\mathsf{M}(d) = \mathsf{PolMul}(d) = O(d^{\omega-1-\varepsilon})$ $\quad \mathsf{M}'(d) = \mathsf{GCD}(d) \in O(\mathsf{M}(d)\log(d))$ $\quad \frac{D}{m} = \frac{\mathsf{degdet}}{m} =$ avg row degree

$$\mathcal{C}(m, D) \leqslant 2\mathcal{C}\left(\frac{m}{2}, \left\lfloor \frac{D}{2} \right\rfloor\right) + \mathcal{C}\left(\frac{m}{2}, D\right) + O(m^{\omega} M'\left(\frac{D}{m}\right))$$

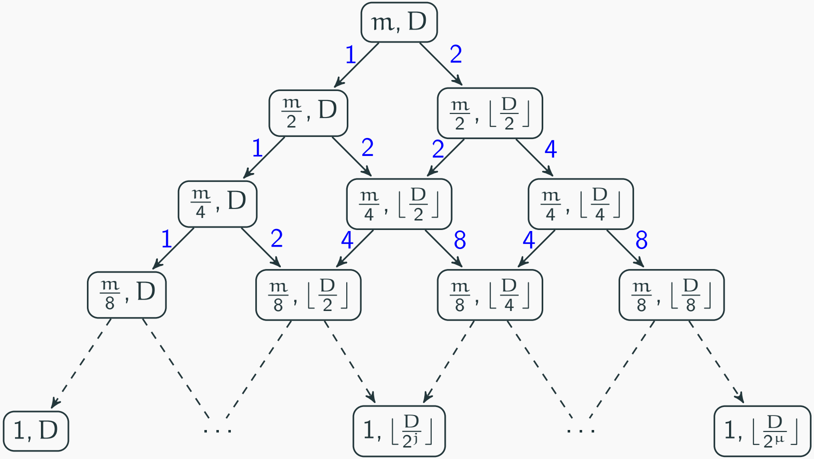
where: $M(d) = \mathsf{PolMul}(d) = O(d^{\omega-1-\varepsilon})$ $\quad M'(d) = \mathsf{GCD}(d) \in O(M(d)\log(d))$ $\quad \frac{D}{m} = \frac{\mathsf{degdet}}{m} = $ avg row degree



Complexity

$$\mathcal{C}(m, D) \leqslant 2\mathcal{C}\left(\frac{m}{2}, \left\lfloor \frac{D}{2} \right\rfloor\right) + \mathcal{C}\left(\frac{m}{2}, D\right) + O(m^{\omega} M'\left(\frac{D}{m}\right))$$

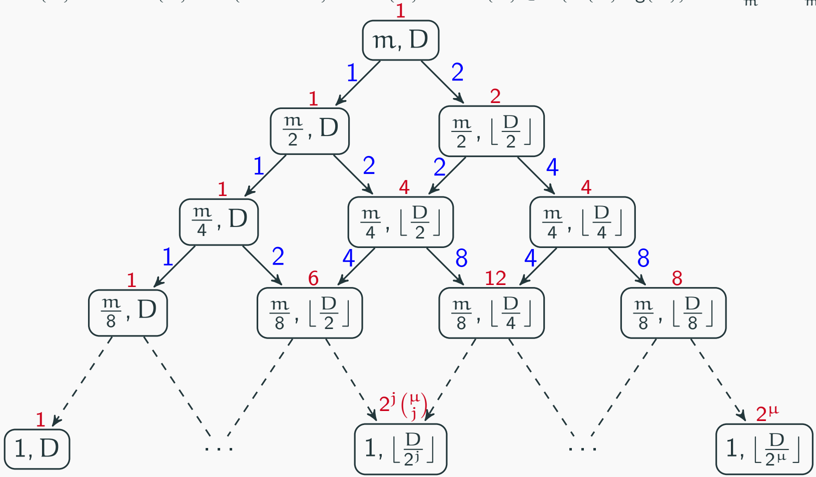
where: $M(d) = \mathsf{PolMul}(d) = O(d^{\omega-1-\varepsilon})$ $\quad M'(d) = \mathsf{GCD}(d) \in O(M(d)\log(d))$ $\quad \frac{D}{m} = \frac{\mathsf{degdet}}{m} = \text{avg row degree}$



Complexity

$$\mathcal{C}(m, D) \leqslant 2\mathcal{C}\left(\frac{m}{2}, \left\lfloor \frac{D}{2} \right\rfloor\right) + \mathcal{C}\left(\frac{m}{2}, D\right) + O\left(m^{\omega} M'\left(\frac{D}{m}\right)\right)$$

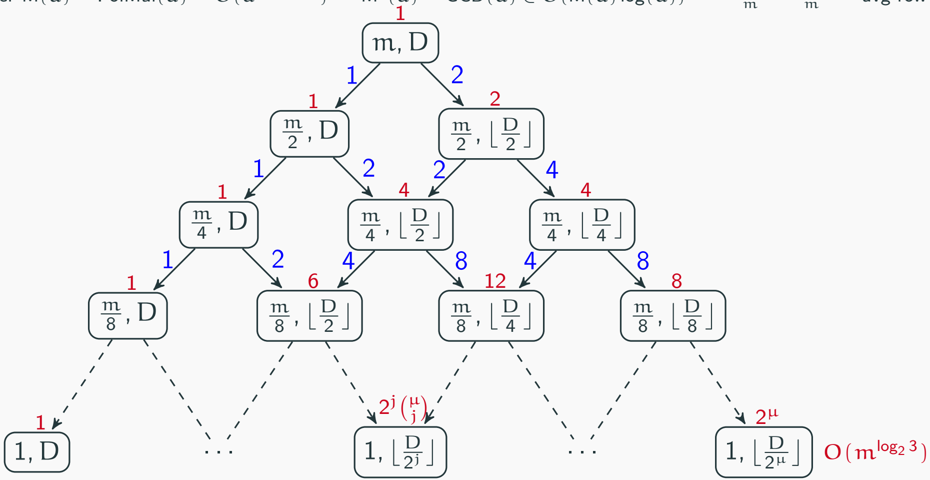
where: $M(d) = \mathrm{PolMul}(d) = O(d^{\omega-1-\varepsilon})$ $\quad M'(d) = \mathrm{GCD}(d) \in O(M(d)\log(d))$ $\quad \frac{D}{m} = \frac{\mathrm{degdet}}{m} =$ avg row degree



Complexity

$$\mathcal{C}(m, D) \leqslant 2\mathcal{C}\left(\frac{m}{2}, \left\lfloor \frac{D}{2} \right\rfloor\right) + \mathcal{C}\left(\frac{m}{2}, D\right) + O\left(m^{\omega} M'\left(\frac{D}{m}\right)\right)$$

where: $M(d) = \mathrm{PolMul}(d) = O(d^{\omega-1-\varepsilon})$ $\quad M'(d) = \mathrm{GCD}(d) \in O(M(d)\log(d))$ $\quad \frac{D}{m} = \frac{\text{degdet}}{m} =$ avg row degree

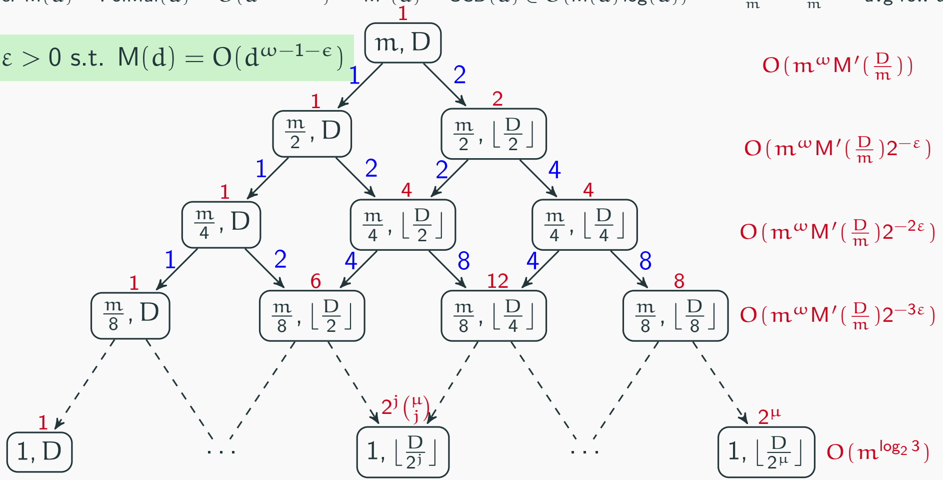


Complexity

$$\mathcal{C}(m, D) \leqslant 2\mathcal{C}\left(\frac{m}{2}, \left\lfloor \frac{D}{2} \right\rfloor\right) + \mathcal{C}\left(\frac{m}{2}, D\right) + O\left(m^{\omega} M'\left(\frac{D}{m}\right)\right)$$

where: $M(d) = \mathsf{PolMul}(d) = O(d^{\omega-1-\varepsilon})$ $\quad M'(d) = \mathsf{GCD}(d) \in O(M(d)\log(d))$ $\quad \frac{D}{m} = \frac{\mathsf{degdet}}{m} = \text{avg row degree}$

for $\varepsilon > 0$ s.t. $M(d) = O(d^{\omega-1-\epsilon})$

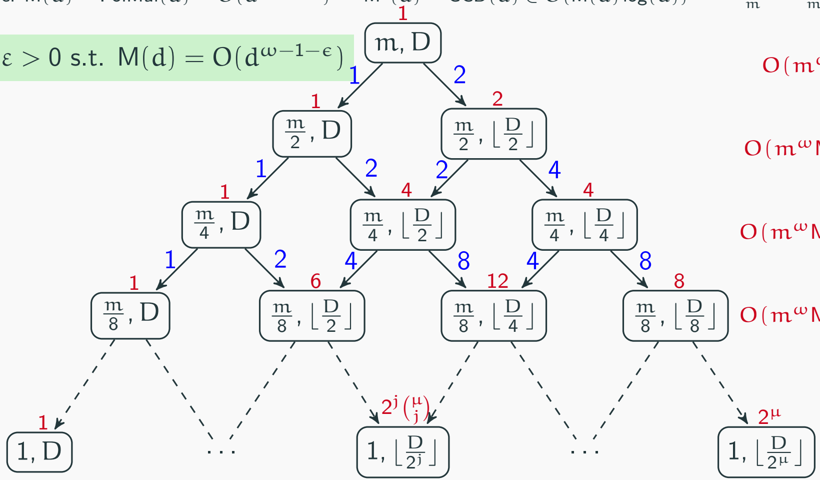


Complexity

$$\mathcal{C}(m, D) \leqslant 2\mathcal{C}\left(\frac{m}{2}, \left\lfloor \frac{D}{2} \right\rfloor\right) + \mathcal{C}\left(\frac{m}{2}, D\right) + O(m^{\omega} M'\left(\frac{D}{m}\right)) \leqslant O(m^{\omega} M'\left(\frac{D}{m}\right))$$

where: $M(d) = \mathsf{PolMul}(d) = O(d^{\omega-1-\varepsilon})$ $\quad M'(d) = \mathsf{GCD}(d) \in O(M(d)\log(d))$ $\quad \frac{D}{m} = \frac{\mathsf{degdet}}{m} =$ avg row degree

for $\varepsilon > 0$ s.t. $M(d) = O(d^{\omega-1-\epsilon})$



$O(m^{\omega} M'(\frac{D}{m}))$

$O(m^{\omega} M'(\frac{D}{m}) 2^{-\varepsilon})$

$O(m^{\omega} M'(\frac{D}{m}) 2^{-2\varepsilon})$

$O(m^{\omega} M'(\frac{D}{m}) 2^{-3\varepsilon})$

$O(m^{\log_2 3})$

A deterministic reduction to Matrix multiplication

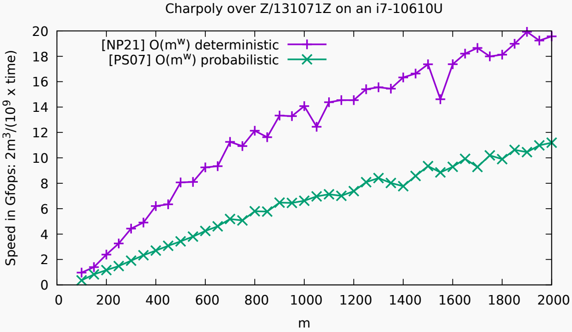
Results [Neiger-P. 21]

- CharPoly $= \Theta(\text{MatMul}) = \Theta(n^{\omega})$ deterministically
- Determinant of reduced polynomial matrices in $O(m^{\omega} M'(\frac{D}{m}))$

A deterministic reduction to Matrix multiplication

Results [Neiger-P. 21]

- CharPoly $= \Theta(\text{MatMul}) = \Theta(n^{\omega})$ deterministically
- Determinant of reduced polynomial matrices in $O(m^{\omega} M'(\frac{D}{m}))$



- Prototype incomplete implementation (does not deal with the non-generic cases)

Spin-off result

Lemma

A right kernel basis of $\mathbf{A} \in \mathbb{K}[x]^{m \times O(m)}$ *with constant degree can be computed in reduced form in* $O(m^{\omega})$ *field operations.*

Spin-off result

Lemma

A right kernel basis of $\mathbf{A} \in \mathbb{K}[x]^{m \times O(m)}$ *with constant degree can be computed in reduced form in* $O(m^\omega)$ *field operations.*

Corollary

The Krylov matrix $\mathbf{K}_{\mathbf{A},\mathbf{v}} = \begin{bmatrix} \mathbf{v} & \mathbf{A}\mathbf{v} & \dots & \mathbf{A}^{m-1}\mathbf{v} \end{bmatrix}$ with $\mathbf{A} \in \mathbb{K}^{m \times m}$ can be computed in $O(m^\omega)$.

Spin-off result

Lemma

A right kernel basis of $\mathbf{A} \in \mathbb{K}[x]^{m \times O(m)}$ *with constant degree can be computed in reduced form in* $O(m^\omega)$ *field operations.*

Corollary

The Krylov matrix $\mathbf{K}_{\mathbf{A},\mathbf{v}} = \begin{bmatrix} \mathbf{v} & \mathbf{A}\mathbf{v} & \dots & \mathbf{A}^{m-1}\mathbf{v} \end{bmatrix}$ with $\mathbf{A} \in \mathbb{K}^{m \times m}$ can be computed in $O(m^\omega)$.

Sketch of proof.

$$\left[\begin{array}{c|c} \mathbf{I}_m - x\mathbf{A} & -\mathbf{v} \end{array}\right] \begin{bmatrix} s \\ t \end{bmatrix} = 0$$

Hence

$$s/t = (\mathbf{I}_m - x\mathbf{A})^{-1}\mathbf{v} = \sum_{i=0}^{\infty} x^k \mathbf{A}^k \mathbf{v}.$$

A truncated series expansion of s/t at order m produces the Krylov iterates. □

Via Block-Wiedemann's algorithm

Block-Wiedemann approach

$$\det(\lambda \mathbf{I}_m - \mathbf{A}) = 1/X^m \det(\mathbf{I}_m - X\mathbf{A}) \text{ for } X = 1/\lambda$$

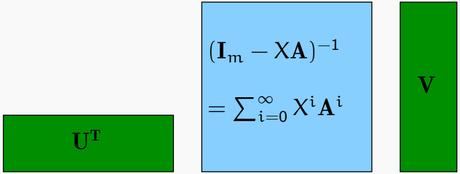
$(\mathbf{I}_m - X\mathbf{A})^{-1}$

$= \sum_{i=0}^{\infty} X^i \mathbf{A}^i$

Block-Wiedemann approach

$$\det(\lambda \mathbf{I}_m - \mathbf{A}) = 1/X^m \det(\mathbf{I}_m - X\mathbf{A}) \text{ for } X = 1/\lambda$$

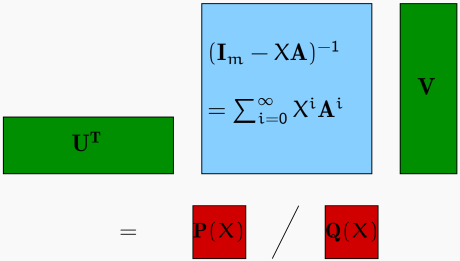
1. Sample unif. $\mathbf{U}, \mathbf{V} \in \mathbb{K}^{m \times k}$
2. For all $i \in \{0, \ldots, 2m/k\}$ Compute $\mathbf{U}^\top \mathbf{A}^i \mathbf{V}$



Block-Wiedemann approach

$$\det(\lambda \mathbf{I}_m - \mathbf{A}) = 1/X^m \det(\mathbf{I}_m - X\mathbf{A}) \text{ for } X = 1/\lambda$$

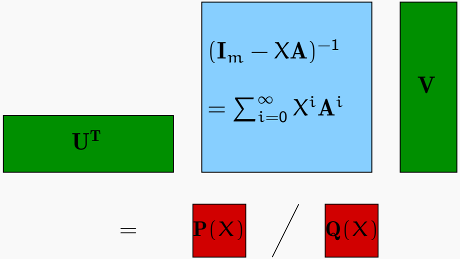
1. Sample unif. $\mathbf{U}, \mathbf{V} \in \mathbb{K}^{m \times k}$
2. For all $i \in \{0, \dots, 2m/k\}$ Compute $\mathbf{U}^\intercal \mathbf{A}^i \mathbf{V}$
3. Reconstruct a matrix fraction
   $\mathbf{P}(X)/\mathbf{Q}(X) = \mathbf{U}^\intercal (\mathbf{I}_m - X\mathbf{A})^{-1} \mathbf{V}$



Block-Wiedemann approach

$$\det(\lambda \mathbf{I}_m - \mathbf{A}) = 1/X^m \det(\mathbf{I}_m - X\mathbf{A}) \text{ for } X = 1/\lambda$$

1. Sample unif. $\mathbf{U}, \mathbf{V} \in \mathbb{K}^{m \times k}$
2. For all $i \in \{0, \ldots, 2m/k\}$ Compute $\mathbf{U}^\top \mathbf{A}^i \mathbf{V}$
3. Reconstruct a matrix fraction $\mathbf{P}(X)/\mathbf{Q}(X) = \mathbf{U}^\top (\mathbf{I}_m - X\mathbf{A})^{-1}\mathbf{V}$
4. Return $\det \mathbf{Q}(X)$ ($= \det(\mathbf{I}_m - X\mathbf{A})$ w.h.p.)



Block-Wiedemann with dense matrices and without divisions

A Baby Step Giant Step approach: [Preparata-Sarwate 78] [Kaltofen 92] [Kaltofen-Villard 05]

$$\begin{array}{c|ccccc}
\times & \mathbf{V} & \mathbf{BV} & \dots & \mathbf{B}^{s-1}\mathbf{V} \\
\hline
\mathbf{U}^\top & \mathbf{U}^\top\mathbf{V} & \mathbf{U}^\top\mathbf{A}^r\mathbf{V} & \dots & \mathbf{U}^\top\mathbf{A}^{rs-r}\mathbf{V} \\
\mathbf{U}^\top\mathbf{A} & \mathbf{U}^\top\mathbf{A}\mathbf{V} & \mathbf{U}^\top\mathbf{A}^{r+1}\mathbf{V} & \dots & \mathbf{U}^\top\mathbf{A}^{rs-r+1}\mathbf{V} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{U}^\top\mathbf{A}^{r-1} & \mathbf{U}^\top\mathbf{A}^{r-1}\mathbf{V} & \mathbf{U}^\top\mathbf{A}^{2r-1}\mathbf{V} & \dots & \mathbf{U}^\top\mathbf{A}^{m-1}\mathbf{V}
\end{array}$$

with $rs = m$ and $\mathbf{B} = \mathbf{A}^r$

Block-Wiedemann with dense matrices and without divisions

A Baby Step Giant Step approach: [Preparata-Sarwate 78] [Kaltofen 92] [Kaltofen-Villard 05]

$$\begin{array}{c|ccccc}
\times & \mathbf{V} & \mathbf{B}\mathbf{V} & \dots & \mathbf{B}^{s-1}\mathbf{V} \\
\hline
\mathbf{U}^T & \mathbf{U}^T\mathbf{V} & \mathbf{U}^T\mathbf{A}^r\mathbf{V} & \dots & \mathbf{U}^T\mathbf{A}^{rs-r}\mathbf{V} \\
\mathbf{U}^T\mathbf{A} & \mathbf{U}^T\mathbf{A}\mathbf{V} & \mathbf{U}^T\mathbf{A}^{r+1}\mathbf{V} & \dots & \mathbf{U}^T\mathbf{A}^{rs-r+1}\mathbf{V} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{U}^T\mathbf{A}^{r-1} & \mathbf{U}^T\mathbf{A}^{r-1}\mathbf{V} & \mathbf{U}^T\mathbf{A}^{2r-1}\mathbf{V} & \dots & \mathbf{U}^T\mathbf{A}^{m-1}\mathbf{V}
\end{array}$$

with $rs = m$ and $\mathbf{B} = \mathbf{A}^r$

Combined with avoidance of divisions ([Strassen 73], [Kaltofen 92]) yields

- Division free algorithms for the characteristic polynomial [Kaltofen-Villard 05]
  - over $\mathbb{Z}$ in $O\tilde{~}(m^{2.6973} \log \|A\|)$ bit operations probabilistic
  - over any commutative ring in $O(m^{2.6973})$ ring operations deterministic

Block-Wiedemann with dense matrices and without divisions

A Baby Step Giant Step approach: [Preparata-Sarwate 78] [Kaltofen 92] [Kaltofen-Villard 05]

$$\begin{array}{c|cccc} \times & \mathbf{V} & \mathbf{BV} & \dots & \mathbf{B}^{s-1}\mathbf{V} \\ \hline \mathbf{U}^T & \mathbf{U}^T\mathbf{V} & \mathbf{U}^T\mathbf{A}^r\mathbf{V} & \dots & \mathbf{U}^T\mathbf{A}^{rs-r}\mathbf{V} \\ \mathbf{U}^T\mathbf{A} & \mathbf{U}^T\mathbf{AV} & \mathbf{U}^T\mathbf{A}^{r+1}\mathbf{V} & \dots & \mathbf{U}^T\mathbf{A}^{rs-r+1}\mathbf{V} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{U}^T\mathbf{A}^{r-1} & \mathbf{U}^T\mathbf{A}^{r-1}\mathbf{V} & \mathbf{U}^T\mathbf{A}^{2r-1}\mathbf{V} & \dots & \mathbf{U}^T\mathbf{A}^{m-1}\mathbf{V} \end{array}$$

with $rs = m$ and $\mathbf{B} = \mathbf{A}^r$

Combined with avoidance of divisions ([Strassen 73], [Kaltofen 92]) yields

- Division free algorithms for the characteristic polynomial [Kaltofen-Villard 05]
  - over $\mathbb{Z}$ in $O\tilde{}(m^{2.6973} \log \|A\|)$ bit operations probabilistic
  - over any commutative ring in $O(m^{2.6973})$ ring operations deterministic

Open Problems:

- fill the gap with $O\tilde{}(m^{\omega} \log \|A\|)$ bit complexity over $\mathbb{Z}$ (reached for `Det`, `LinSys`, `Smith`)
- fill the gap with $O(m^{\omega})$ division free

Matrices with rank displacement structures

Toeplitz matrix

$$\underset{\mathbf{T}}{\begin{bmatrix} d & e & f & g \\ c & d & e & f \\ b & c & d & e \\ a & b & c & d \end{bmatrix}}$$

Matrices with rank displacement structures

Toeplitz matrix

$$\begin{bmatrix} d & e & f & g \\ c & d & e & f \\ b & c & d & e \\ a & b & c & d \end{bmatrix} - \begin{bmatrix} & & & \\ & d & e & f \\ & c & d & e \\ & b & c & d \end{bmatrix} = \begin{bmatrix} d & e & f & g \\ c & & & \\ b & & & \\ a & & & \end{bmatrix}$$

$$\mathbf{T} - \mathbf{ZTZ} = \Delta_{\mathsf{Z},\mathsf{Z}}(\mathbf{T})$$

Matrices with rank displacement structures

Toeplitz matrix

$$\begin{bmatrix} d & e & f & g \\ c & d & e & f \\ b & c & d & e \\ a & b & c & d \end{bmatrix} - \begin{bmatrix} & & & \\ & d & e & f \\ & c & d & e \\ & b & c & d \end{bmatrix} = \begin{bmatrix} d & e & f & g \\ c & & & \\ b & & & \\ a & & & \end{bmatrix} = \begin{bmatrix} d & 1 \\ c & 0 \\ b & 0 \\ a & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e & f & g \end{bmatrix}$$

$$\mathbf{T} - \mathbf{ZTZ} = \Delta_{Z,Z}(\mathbf{T}) = \mathbf{G} \quad \mathbf{H}^\top$$

Matrices with rank displacement structures

Toeplitz matrix

$$
\begin{bmatrix} d & e & f & g \\ c & d & e & f \\ b & c & d & e \\ a & b & c & d \end{bmatrix}
-
\begin{bmatrix} & & & \\ & d & e & f \\ & c & d & e \\ & b & c & d \end{bmatrix}
=
\begin{bmatrix} d & e & f & g \\ c & & & \\ b & & & \\ a & & & \end{bmatrix}
=
\begin{bmatrix} d & 1 \\ c & 0 \\ b & 0 \\ a & 0 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e & f & g \end{bmatrix}
$$

$$
\mathbf{T} \quad - \quad \mathbf{ZTZ} \quad = \quad \Delta_{Z,Z}(\mathbf{T}) \quad = \quad \mathbf{G} \quad \mathbf{H}^{T}
$$

Generalizations

- Toeplitz-like : s.t. $\Delta_{Z,Z}(\mathbf{T}) = \mathbf{T} - \mathbf{ZTZ}$ has rank α ($= \mathbf{GH}^{T}$ with $\mathbf{G}, \mathbf{H} \in \mathbb{K}^{m \times \alpha}$).
- Hankel-like, Vandemonde-like, Cauchy-like, etc : similarly with other displ. operators $\Delta_{X,Y}, \nabla_{X,Y}$.

RDP$_{\alpha}$: matrices with a rank displacement structure of order α.

Matrices with rank displacement structures

Toeplitz matrix

$$\begin{bmatrix} d & e & f & g \\ c & d & e & f \\ b & c & d & e \\ a & b & c & d \end{bmatrix} - \begin{bmatrix} & & & \\ & d & e & f \\ & c & d & e \\ & b & c & d \end{bmatrix} = \begin{bmatrix} d & e & f & g \\ c & & & \\ b & & & \\ a & & & \end{bmatrix} = \begin{bmatrix} d & 1 \\ c & 0 \\ b & 0 \\ a & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e & f & g \end{bmatrix}$$

$$\mathbf{T} \quad - \quad \mathbf{ZTZ} \quad = \quad \Delta_{Z,Z}(\mathbf{T}) \quad = \quad \mathbf{G} \quad \mathbf{H}^{\mathsf{T}}$$

Generalizations

- Toeplitz-like : s.t. $\Delta_{Z,Z}(\mathbf{T}) = \mathbf{T} - \mathbf{ZTZ}$ has rank α $(= \mathbf{GH}^{\mathsf{T}}$ with $\mathbf{G}, \mathbf{H} \in \mathbb{K}^{m \times \alpha})$.
- Hankel-like, Vandemonde-like, Cauchy-like, etc : similarly with other displ. operators $\Delta_{X,Y}, \nabla_{X,Y}$.

RDP$_{\alpha}$: matrices with a rank displacement structure of order α.

Property

A Toeplitz like matrix decomposes as $\mathbf{T} = \sum_{i=1}^{\alpha} \mathbf{L}_i \mathbf{U}_i$ *where* $\mathbf{L}_i$ *are lower triangular and* $\mathbf{U}_i$ *upper triangular Toeplitz matrices.*

Computing with matrices with rank displacement structure

Multiplication

- Toeplitz $\times$ Vector: (via polynomial multiplication) $O\tilde{}(m)$
- RDP_α $\times (m \times \alpha)$ block-vector: [Bostan-Jeannerod-Mouilleron-Schost 07,17] $O\tilde{}(m\alpha^{\omega-1})$

Computing with matrices with rank displacement structure

Multiplication

- Toeplitz $\times$ Vector: (via polynomial multiplication) $\tilde{O}(m)$
- $\mathrm{RDP}_\alpha \times (m \times \alpha)$ block-vector: [Bostan-Jeannerod-Mouilleron-Schost 07,17] $\tilde{O}(m\alpha^{\omega-1})$

Linear system

- Toeplitz $^{-1}$ $\times$ Vector: (via polynomial multiplication) [Pan01] $\tilde{O}(m)$
- $\mathrm{RDP}_\alpha{}^{-1} \times (m \times \alpha)$ block-vector: [Bostan Et al. 17] $\tilde{O}(m\alpha^{\omega-1})$
- Same costs for `Det`, `Inverse` [Pan 01, Bostan-Jeannerod-Mouilleron-Schost 17]

Computing with matrices with rank displacement structure

Multiplication

- Toeplitz $\times$ Vector: (via polynomial multiplication) $O\tilde{~}(m)$
- RDP_α $\times(m \times \alpha)$ block-vector: [Bostan-Jeannerod-Mouilleron-Schost 07,17] $O\tilde{~}(m\alpha^{\omega-1})$

Linear system

- Toeplitz $^{-1}$ $\times$ Vector: (via polynomial multiplication) [Pan01] $O\tilde{~}(m)$
- RDP_α $^{-1}$ $\times(m \times \alpha)$ block-vector: [Bostan Et al. 17] $O\tilde{~}(m\alpha^{\omega-1})$
- Same costs for `Det`, `Inverse` [Pan 01, Bostan-Jeannerod-Mouilleron-Schost 17]

Characteristic polynomial

- `Charpoly`(RDP_α) by Evalutation-Interpolation: $m\times$ `Det` $O\tilde{~}(m^2\alpha^{\omega-1})$

Computing with matrices with rank displacement structure

Multiplication

- Toeplitz $\times$ Vector: (via polynomial multiplication) $\tilde{O}(m)$
- $\text{RDP}_\alpha \times (m \times \alpha)$ block-vector: [Bostan-Jeannerod-Mouilleron-Schost 07,17] $\tilde{O}(m\alpha^{\omega-1})$

Linear system

- Toeplitz $^{-1} \times$ Vector: (via polynomial multiplication) [Pan01] $\tilde{O}(m)$
- $\text{RDP}_\alpha{}^{-1} \times (m \times \alpha)$ block-vector: [Bostan Et al. 17] $\tilde{O}(m\alpha^{\omega-1})$
- Same costs for `Det`, `Inverse` [Pan 01, Bostan-Jeannerod-Mouilleron-Schost 17]

Characteristic polynomial

- `Charpoly`(RDP_α) by Evalutation-Interpolation: $m\times$ `Det` $\tilde{O}(m^2\alpha^{\omega-1})$
- `Minpoly`(Toeplitz-/Hankel-like) probabilistic [Karpman-P.-Signargout-Villard 21] $\tilde{O}(m^{1.86}\alpha^{0.53})$
- `Charpoly`(Toeplitz-/Hankel-like) generic [Karpman-P.-Signargout-Villard 21] $\tilde{O}(m^{1.58}\alpha^{0.53})$

Computing with matrices with rank displacement structure

Multiplication

- Toeplitz $\times$ Vector: (via polynomial multiplication) $O\tilde{}(m)$
- $\text{RDP}_\alpha \times (m \times \alpha)$ block-vector: [Bostan-Jeannerod-Mouilleron-Schost 07,17] $O\tilde{}(m\alpha^{\omega-1})$

Linear system

- Toeplitz $^{-1}$ $\times$ Vector: (via polynomial multiplication) [Pan01] $O\tilde{}(m)$
- $\text{RDP}_\alpha{}^{-1} \times (m \times \alpha)$ block-vector: [Bostan Et al. 17] $O\tilde{}(m\alpha^{\omega-1})$
- Same costs for `Det`, `Inverse` [Pan 01, Bostan-Jeannerod-Mouilleron-Schost 17]

Characteristic polynomial

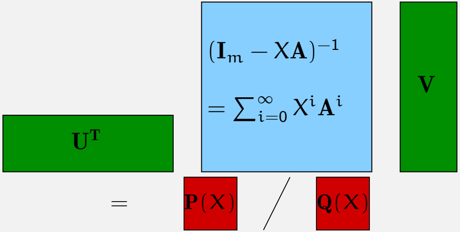
- `Charpoly`(RDP_α) by Evalutation-Interpolation: $m\times$ `Det` $O\tilde{}(m^2\alpha^{\omega-1})$
- `Minpoly`(Toeplitz-/Hankel-like) probabilistic [Karpman-P.-Signargout-Villard 21] $O\tilde{}(m^{1.86}\alpha^{0.53})$
- `Charpoly`(Toeplitz-/Hankel-like) generic [Karpman-P.-Signargout-Villard 21] $O\tilde{}(m^{1.58}\alpha^{0.53})$

Block Wiedemann algorithm with rank displacement structure

Explicit iteration

[Karpman-P.-Signargout-Villard 21]

- Dense projections: $\mathbf{U}, \mathbf{V} \in \mathbb{K}^{m \times k}$



$\times$	$\mathbf{V}$	$\mathbf{BV}$	$\dots$	$\mathbf{B}^{s-1}\mathbf{V}$
$\mathbf{U}^T$	$\mathbf{U}^T\mathbf{V}$	$\mathbf{U}^T\mathbf{A}^r\mathbf{V}$	$\dots$	$\mathbf{U}^T\mathbf{A}^{rs-r}\mathbf{V}$
$\mathbf{U}^T\mathbf{A}$	$\mathbf{U}^T\mathbf{AV}$	$\mathbf{U}^T\mathbf{A}^{r+1}\mathbf{V}$	$\dots$	$\mathbf{U}^T\mathbf{A}^{rs-r+1}\mathbf{V}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{U}^T\mathbf{A}^{r-1}$	$\mathbf{U}^T\mathbf{A}^{r-1}\mathbf{V}$	$\mathbf{U}^T\mathbf{A}^{2r-1}\mathbf{V}$	$\dots$	$\mathbf{U}^T\mathbf{A}^{m-1}\mathbf{V}$

Block Wiedemann algorithm with rank displacement structure

Explicit iteration

[Karpman-P.-Signargout-Villard 21]

- Dense projections: $\mathbf{U}, \mathbf{V} \in \mathbb{K}^{m \times k}$

$$\mathbf{U}^T \quad (\mathbf{I}_m - X\mathbf{A})^{-1} = \sum_{i=0}^{\infty} X^i \mathbf{A}^i \quad \mathbf{V}$$

$$= \mathbf{P}(X) \, / \, \mathbf{Q}(X)$$

$\times$	$\mathbf{V}$	$\mathbf{BV}$	$\dots$	$\mathbf{B}^{s-1}\mathbf{V}$
$\mathbf{U}^T$	$\mathbf{U}^T\mathbf{V}$	$\mathbf{U}^T\mathbf{A}^r\mathbf{V}$	$\dots$	$\mathbf{U}^T\mathbf{A}^{rs-r}\mathbf{V}$
$\mathbf{U}^T\mathbf{A}$	$\mathbf{U}^T\mathbf{AV}$	$\mathbf{U}^T\mathbf{A}^{r+1}\mathbf{V}$	$\dots$	$\mathbf{U}^T\mathbf{A}^{rs-r+1}\mathbf{V}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{U}^T\mathbf{A}^{r-1}$	$\mathbf{U}^T\mathbf{A}^{r-1}\mathbf{V}$	$\mathbf{U}^T\mathbf{A}^{2r-1}\mathbf{V}$	$\dots$	$\mathbf{U}^T\mathbf{A}^{m-1}\mathbf{V}$

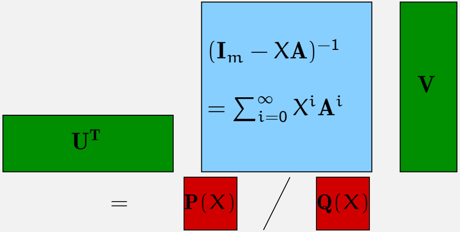
- Toeplitz-/Hankel-like: inflation of the disp. rank. $\alpha(\mathbf{A}^r) = r \times \alpha(\mathbf{A})$ → $\tilde{O}(m^{1.86}\alpha^{0.53})$

Block Wiedemann algorithm with rank displacement structure

Explicit iteration

[Karpman-P.-Signargout-Villard 21]

- Dense projections: $\mathbf{U}, \mathbf{V} \in \mathbb{K}^{m\times k}$



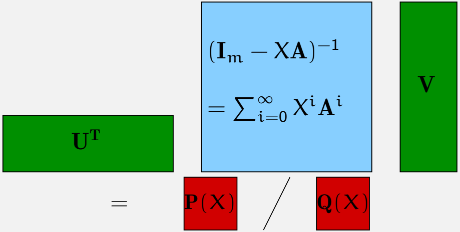
$\times$	$\mathbf{V}$	$\mathbf{BV}$	$\dots$	$\mathbf{B}^{s-1}\mathbf{V}$
$\mathbf{U}^T$	$\mathbf{U}^T\mathbf{V}$	$\mathbf{U}^T\mathbf{A}^r\mathbf{V}$	$\dots$	$\mathbf{U}^T\mathbf{A}^{rs-r}\mathbf{V}$
$\mathbf{U}^T\mathbf{A}$	$\mathbf{U}^T\mathbf{AV}$	$\mathbf{U}^T\mathbf{A}^{r+1}\mathbf{V}$	$\dots$	$\mathbf{U}^T\mathbf{A}^{rs-r+1}\mathbf{V}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{U}^T\mathbf{A}^{r-1}$	$\mathbf{U}^T\mathbf{A}^{r-1}\mathbf{V}$	$\mathbf{U}^T\mathbf{A}^{2r-1}\mathbf{V}$	$\dots$	$\mathbf{U}^T\mathbf{A}^{m-1}\mathbf{V}$

- Toeplitz-/Hankel-like: inflation of the disp. rank. $\alpha(\mathbf{A}^r) = r \times \alpha(\mathbf{A})$ → $\tilde{O}(m^{1.86}\alpha^{0.53})$
- Considering structured $\mathbf{U}, \mathbf{V}$ does not help

Explicit iteration

[Karpman-P.-Signargout-Villard 21]

- Dense projections: $\mathbf{U}, \mathbf{V} \in \mathbb{K}^{m \times k}$



$\times$	$\mathbf{V}$	$\mathbf{BV}$	$\ldots$	$\mathbf{B}^{s-1}\mathbf{V}$
$\mathbf{U}^T$	$\mathbf{U}^T\mathbf{V}$	$\mathbf{U}^T\mathbf{A}^r\mathbf{V}$	$\ldots$	$\mathbf{U}^T\mathbf{A}^{rs-r}\mathbf{V}$
$\mathbf{U}^T\mathbf{A}$	$\mathbf{U}^T\mathbf{AV}$	$\mathbf{U}^T\mathbf{A}^{r+1}\mathbf{V}$	$\ldots$	$\mathbf{U}^T\mathbf{A}^{rs-r+1}\mathbf{V}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{U}^T\mathbf{A}^{r-1}$	$\mathbf{U}^T\mathbf{A}^{r-1}\mathbf{V}$	$\mathbf{U}^T\mathbf{A}^{2r-1}\mathbf{V}$	$\ldots$	$\mathbf{U}^T\mathbf{A}^{m-1}\mathbf{V}$

- Toeplitz-/Hankel-like: inflation of the disp. rank. $\alpha(\mathbf{A}^r) = r \times \alpha(\mathbf{A})$ $\rightarrow$ $O\tilde{}(m^{1.86}\alpha^{0.53})$
- Considering structured $\mathbf{U}, \mathbf{V}$ does not help
- Rk: in [Neiger-Salvy-Schost-Villard 23] Modular composition uses Charpoly(ModPolyMult) disp. rank remains stable. $\alpha(\mathbf{A}^r) = \alpha(\mathbf{A})$

$\rightarrow$ $O\tilde{}(m^{1.43})$

Block Wiedemann algorithm with rank displacement structure

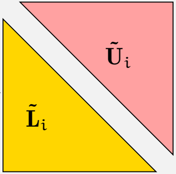
Implicit iteration using structured inverse [Karpman-P.-Signargout-Villard 21] **based on** [Villard 18]

$$\boxed{\begin{array}{l}(\mathbf{I}_m - X\mathbf{A})^{-1} \\ = \sum_{i=1}^{\alpha} \tilde{\mathbf{L}}_i \tilde{\mathbf{U}}_i \\ \quad \bmod X^{2m/k}\end{array}}$$

1. Structured inversion modulo $X^{2m/k}$:
   $(\mathbf{I}_m - X\mathbf{A})^{-1} = \sum_{i=1}^{\alpha} \tilde{\mathbf{L}}_i \tilde{\mathbf{U}}_i \mod X^{2m/k}$

Block Wiedemann algorithm with rank displacement structure

Implicit iteration using structured inverse [Karpman-P.-Signargout-Villard 21] based on [Villard 18]

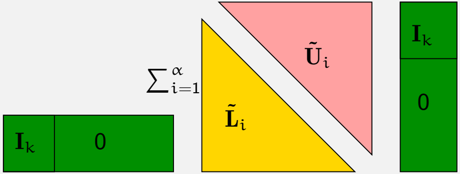
$\sum_{i=1}^{\alpha}$ 

1. Structured inversion modulo $X^{2m/k}$:
   $(\mathbf{I}_m - X\mathbf{A})^{-1} = \sum_{i=1}^{\alpha} \tilde{\mathbf{L}}_i \tilde{\mathbf{U}}_i \mod X^{2m/k}$

Block Wiedemann algorithm with rank displacement structure

Implicit iteration using structured inverse [Karpman-P.-Signargout-Villard 21] based on [Villard 18]

- Structured projections: $\mathbf{U} = \mathbf{V} = \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0} \end{bmatrix} \in \mathbb{K}^{m \times k}$

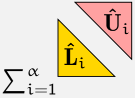


1. Structured inversion modulo $X^{2m/k}$:
   $(\mathbf{I}_m - X\mathbf{A})^{-1} = \sum_{i=1}^{\alpha} \tilde{\mathbf{L}}_i \tilde{\mathbf{U}}_i \mod X^{2m/k}$
2. Crop : $\begin{bmatrix} \mathbf{I}_k & \mathbf{0} \end{bmatrix} \sum \tilde{\mathbf{L}}_i \tilde{\mathbf{U}}_i \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0} \end{bmatrix} =$
   $\sum_{i=1}^{\alpha} (\tilde{\mathbf{L}}_i)_{1..k,1..k} (\tilde{\mathbf{U}}_i)_{1..k,1..k}$

Block Wiedemann algorithm with rank displacement structure

Implicit iteration using structured inverse [Karpman-P.-Signargout-Villard 21] **based on** [Villard 18]

- Structured projections: $\mathbf{U} = \mathbf{V} = \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0} \end{bmatrix} \in \mathbb{K}^{m \times k}$



1. Structured inversion modulo $X^{2m/k}$:
   $(\mathbf{I}_m - X\mathbf{A})^{-1} = \sum_{i=1}^{\alpha} \tilde{\mathbf{L}}_i \tilde{\mathbf{U}}_i \mod X^{2m/k}$
2. Crop : $\begin{bmatrix} \mathbf{I}_k & \mathbf{0} \end{bmatrix} \sum \tilde{\mathbf{L}}_i \tilde{\mathbf{U}}_i \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0} \end{bmatrix} =$
   $\sum_{i=1}^{\alpha} (\tilde{\mathbf{L}}_i)_{1..k,1..k} (\tilde{\mathbf{U}}_i)_{1..k,1..k}$

Block Wiedemann algorithm with rank displacement structure

Implicit iteration using structured inverse [Karpman-P.-Signargout-Villard 21] **based on** [Villard 18]

- Structured projections: $\mathbf{U} = \mathbf{V} = \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0} \end{bmatrix} \in \mathbb{K}^{m \times k}$

1. Structured inversion modulo $X^{2m/k}$:
   $(\mathbf{I}_m - X\mathbf{A})^{-1} = \sum_{i=1}^{\alpha} \tilde{\mathbf{L}}_i \tilde{\mathbf{U}}_i \mod X^{2m/k}$
2. Crop : $\begin{bmatrix} \mathbf{I}_k & \mathbf{0} \end{bmatrix} \sum \tilde{\mathbf{L}}_i \tilde{\mathbf{U}}_i \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0} \end{bmatrix} =$
   $\sum_{i=1}^{\alpha} (\tilde{\mathbf{L}}_i)_{1..k,1..k} (\tilde{\mathbf{U}}_i)_{1..k,1..k}$
3. Expand to dense

Block Wiedemann algorithm with rank displacement structure

Implicit iteration using structured inverse [Karpman-P.-Signargout-Villard 21] **based on** [Villard 18]

- Structured projections: $\mathbf{U} = \mathbf{V} = \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0} \end{bmatrix} \in \mathbb{K}^{m \times k}$

$$\mathbf{S}$$

$$= \quad \mathbf{P}(X) \; / \; \mathbf{Q}(X)$$

1. Structured inversion modulo $X^{2m/k}$: $(\mathbf{I}_m - X\mathbf{A})^{-1} = \sum_{i=1}^{\alpha} \tilde{\mathbf{L}}_i \tilde{\mathbf{U}}_i \mod X^{2m/k}$
2. Crop : $\begin{bmatrix} \mathbf{I}_k & \mathbf{0} \end{bmatrix} \sum \tilde{\mathbf{L}}_i \tilde{\mathbf{U}}_i \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0} \end{bmatrix} = \sum_{i=1}^{\alpha} (\tilde{\mathbf{L}}_i)_{1..k,1..k} (\tilde{\mathbf{U}}_i)_{1..k,1..k}$
3. Expand to dense
4. Reconstruct a matrix fraction: $\mathbf{S} = \mathbf{P}/\mathbf{Q}$

Block Wiedemann algorithm with rank displacement structure

Implicit iteration using structured inverse [Karpman-P.-Signargout-Villard 21] based on [Villard 18]

- Structured projections: $\mathbf{U} = \mathbf{V} = \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0} \end{bmatrix} \in \mathbb{K}^{m \times k}$

$$\mathbf{S}$$

$$= \mathbf{P}(X) \;/\; \mathbf{Q}(X)$$

1. Structured inversion modulo $X^{2m/k}$: $(\mathbf{I}_m - X\mathbf{A})^{-1} = \sum_{i=1}^{\alpha} \tilde{\mathbf{L}}_i \tilde{\mathbf{U}}_i \mod X^{2m/k}$
2. Crop : $\begin{bmatrix} \mathbf{I}_k & \mathbf{0} \end{bmatrix} \sum \tilde{\mathbf{L}}_i \tilde{\mathbf{U}}_i \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0} \end{bmatrix} = \sum_{i=1}^{\alpha} (\tilde{\mathbf{L}}_i)_{1..k,1..k} (\tilde{\mathbf{U}}_i)_{1..k,1..k}$
3. Expand to dense
4. Reconstruct a matrix fraction: $\mathbf{S} = \mathbf{P}/\mathbf{Q}$
5. Return $\mathrm{Det}(\mathbf{Q})$

- Generic algorithm
- Applies to Toeplitz-like, Hankel-like and Toeplitz-like+Hankel-like
- $O\tilde{}(m^{c(\omega)} \alpha^{\omega - c(\omega)})$ with $c(\omega) = 2 - 1/\omega$ → $O\tilde{}(m^{1.58} \alpha^{0.53})$

Open problems

Open Problems

Dense matrices over a field

- Is fast polynomial arithmetic ($M(d) = O(d^{\omega-1-\varepsilon})$) required for charpoly in $O(m^{\omega})$?

Open Problems

Dense matrices over a field

- Is fast polynomial arithmetic ($M(d) = O(d^{\omega-1-\varepsilon})$) required for charpoly in $O(m^\omega)$?
- MinPoly vs CharPoly:
  - Earlier algorithms: MinPoly often a step towards CharPoly
  - [P.-Storjohann 07]: MinPoly deduced from the probabilistic $O(m^\omega)$ CharPoly
  - [Neiger-P. 21]: MinPoly no longer deducible from the deterministic $O(m^\omega)$ CharPoly

Open Problems

Dense matrices over a field

- Is fast polynomial arithmetic ($M(d) = O(d^{\omega-1-\varepsilon})$) required for charpoly in $O(m^\omega)$?
- MinPoly vs CharPoly:
  - Earlier algorithms: MinPoly often a step towards CharPoly
  - [P.-Storjohann 07]: MinPoly deduced from the probabilistic $O(m^\omega)$ CharPoly
  - [Neiger-P. 21]: MinPoly no longer deducible from the deterministic $O(m^\omega)$ CharPoly
- Frobenius normal form (and the transformation matrix ?) in $O(m^\omega)$

Open Problems

Dense matrices over a field

- Is fast polynomial arithmetic ($M(d) = O(d^{\omega-1-\varepsilon})$) required for charpoly in $O(m^\omega)$?
- MinPoly vs CharPoly:
  - Earlier algorithms: MinPoly often a step towards CharPoly
  - [P.-Storjohann 07]: MinPoly deduced from the probabilistic $O(m^\omega)$ CharPoly
  - [Neiger-P. 21]: MinPoly no longer deducible from the deterministic $O(m^\omega)$ CharPoly
- Frobenius normal form (and the transformation matrix ?) in $O(m^\omega)$

Structured matrices

- Make the blackbox approach significantly competitive (improve over [Villard 03] $\tilde{O}(m^{2.36})$ estimate)

Open Problems

Dense matrices over a field

- Is fast polynomial arithmetic ($M(d) = O(d^{\omega-1-\varepsilon})$) required for charpoly in $O(m^\omega)$?
- MinPoly vs CharPoly:
  - Earlier algorithms: MinPoly often a step towards CharPoly
  - [P.-Storjohann 07]: MinPoly deduced from the probabilistic $O(m^\omega)$ CharPoly
  - [Neiger-P. 21]: MinPoly no longer deducible from the deterministic $O(m^\omega)$ CharPoly
- Frobenius normal form (and the transformation matrix ?) in $O(m^\omega)$

Structured matrices

- Make the blackbox approach significantly competitive (improve over [Villard 03] $O\tilde{}(m^{2.36})$ estimate)
- Randomize the $O\tilde{}(m^{1.58}\alpha^{0.53})$ generic algorithm for Toeplitz-like+Hankel-like matrices

Open Problems

Dense matrices over a field

- Is fast polynomial arithmetic ($M(d) = O(d^{\omega-1-\varepsilon})$) required for charpoly in $O(m^\omega)$?
- MinPoly vs CharPoly:
  - Earlier algorithms: MinPoly often a step towards CharPoly
  - [P.-Storjohann 07]: MinPoly deduced from the probabilistic $O(m^\omega)$ CharPoly
  - [Neiger-P. 21]: MinPoly no longer deducible from the deterministic $O(m^\omega)$ CharPoly
- Frobenius normal form (and the transformation matrix ?) in $O(m^\omega)$

Structured matrices

- Make the blackbox approach significantly competitive (improve over [Villard 03] $\tilde{O}(m^{2.36})$ estimate)
- Randomize the $\tilde{O}(m^{1.58}\alpha^{0.53})$ generic algorithm for Toeplitz-like+Hankel-like matrices

CharPoly over $\mathbb{K}[y]$

No improvement since [Kaltofen Villard 05] $O(m^{2.6973})$ division free algorithm

- Better understanding of the bivariate matrix structure required