Faster algorithms for the characteristic polynomial

Clément PERNET and Arne STORJOHANN

Symbolic Computation Group
University of Waterloo, Canada.

ISSAC 2007, Waterloo,
July 30

Problem

*Compute the **characteristic polynomial** of a **dense** matrix over a **field***

Problem

*Compute the **characteristic polynomial** of a **dense** matrix over a **field***

Result

Randomized Las-Vegas algorithm in $\mathcal{O}(n^\omega)$ field operations for large fields ($\#F > 2n^2$).

Problem

*Compute the **characteristic polynomial** of a **dense** matrix over a **field***

Result

Randomized Las-Vegas algorithm in $\mathcal{O}(n^\omega)$ field operations for large fields ($\#F > 2n^2$).

- Improves previous complexity by a $\log n$ factor,
- Optimal reduction to Matrix multiplication.

Problem

*Compute the **characteristic polynomial** of a **dense** matrix over a **field***

Result

Randomized Las-Vegas algorithm in $\mathcal{O}(n^\omega)$ field operations for large fields ($\#F > 2n^2$).

- Improves previous complexity by a $\log n$ factor,
- Optimal reduction to Matrix multiplication.
- Practical efficiency. E.g. over $\mathbb{Z}_{547\,909}$:

n	500	5000	15 000
`LinBox`	0.91s	4m44s	2h20m
`magma-2.13`	1.27s	15m32s	7h28m

Outline

Outline

Pre-Strassen age

Leverrier 1840: trace of powers of A, and Newton's formula

- improved/rediscovered by Souriau, Faddeev, Frame and Csanky
- $\mathcal{O}\left(n^4\right)$, based on Matrix multiplication
- Suited for parallel computation model

Pre-Strassen age

Leverrier 1840: trace of powers of A, and Newton's formula

- improved/rediscovered by Souriau, Faddeev, Frame and Csanky
- $\mathcal{O}\left(n^4\right)$, based on Matrix multiplication
- Suited for parallel computation model

Danilevskii 1937: elementary row/column operations

$\Rightarrow \mathcal{O}\left(n^3\right)$

Pre-Strassen age

Leverrier 1840: trace of powers of A, and Newton's formula

- improved/rediscovered by Souriau, Faddeev, Frame and Csanky
- $\mathcal{O}\left(n^4\right)$, based on Matrix multiplication
- Suited for parallel computation model

Danilevskii 1937: elementary row/column operations
$\Rightarrow \mathcal{O}\left(n^3\right)$

Hessenberg 1942: transformation to quasi-upper triangular and determinant expansion formula.
$\Rightarrow \mathcal{O}\left(n^3\right)$

Post-Strassen age

Preparata & Sarwate 1978: Update Csanky with fast matrix multiplication
$\Rightarrow \mathcal{O}\left(n^{\omega+1}\right)$

Post-Strassen age

Preparata & Sarwate 1978: Update Csanky with fast matrix multiplication
$\Rightarrow \mathcal{O}\left(n^{\omega+1}\right)$

Keller-Gehrig 1985, alg.1: computes $(A^{2^i})_{i=1\ldots\log_2 n}$ to form a Krylov basis.

- $\mathcal{O}\left(n^{\omega}\log n\right)$
- the best complexity up to now

Post-Strassen age

Preparata & Sarwate 1978: Update Csanky with fast matrix multiplication
$\Rightarrow \mathcal{O}\left(n^{\omega+1}\right)$

Keller-Gehrig 1985, alg.1: computes $(A^{2^i})_{i=1\ldots\log_2 n}$ to form a Krylov basis.

- $\mathcal{O}\left(n^{\omega} \log n\right)$
- the best complexity up to now

Keller-Gehrig 1985, alg.2: inspired by Danilevskii, block operations

- $\mathcal{O}\left(n^{\omega}\right)$
- but only valid with generic matrices

Outline

Definition (degree d Krylov matrix of one vector v)

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

Property

$$A \times K = K \times \underbrace{\begin{bmatrix} 0 & & & * \\ 1 & & & * \\ & \ddots & & * \\ & & 1 & * \end{bmatrix}}_{C_{P_{min}^{A,v}}}$$

Definition (degree d Krylov matrix of one vector v)

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

Property

$$A \times K = K \times \underbrace{\begin{bmatrix} 0 & & & * \\ 1 & & & * \\ & \ddots & & * \\ & & 1 & * \end{bmatrix}}_{C_{P_{min}^{A,v}}}$$

$\Rightarrow$if $d = n$,

$$K^{-1}AK = C_{P_{car}^A}$$

Definition (degree d Krylov matrix of one vector v)

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

Property

$$A \times K = K \times \underbrace{\begin{bmatrix} 0 & & & * \\ 1 & & & * \\ & \ddots & & * \\ & & 1 & * \end{bmatrix}}_{C_{P^{A,v}_{min}}}$$

$\Rightarrow$if $d = n$,

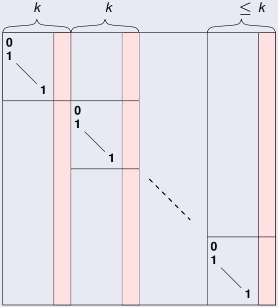
$$K^{-1}AK = C_{P^A_{car}}$$

[Keller-Gehrig, alg. 2] : $K^{-1}AK$ in $\mathcal{O}(n^\omega)$ for A generic

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

$A \times K = K \times$ 

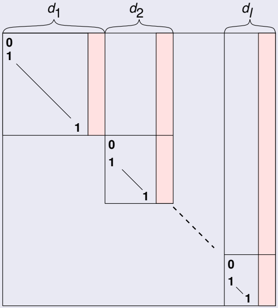
Fact (Shift Hessenberg form)

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

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

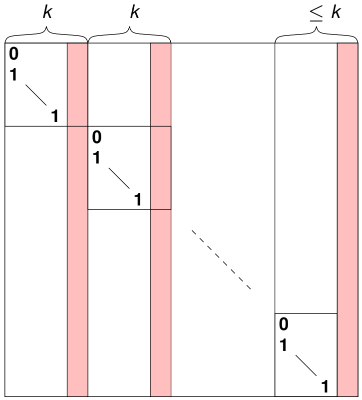
is non-singular, then

$A \times K = K \times$



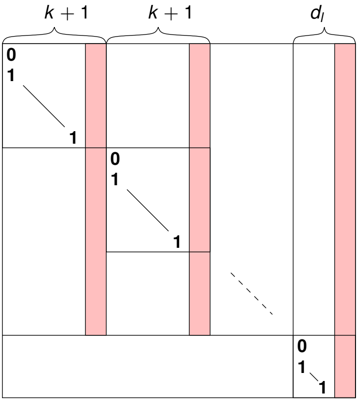
Principle

k-shifted form:



Principle

$k+1$-shifted form:



Principle

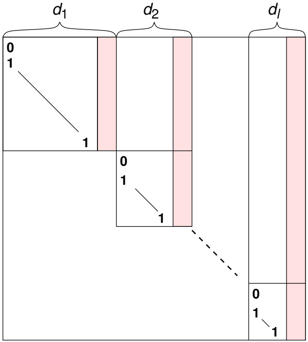
- Compute iteratively from 1-shifted form to d_1-shifted form

Principle

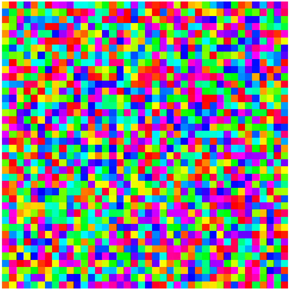
- Compute iteratively from 1-shifted form to d_1-shifted form
- each completed block appears in the increasing degree order

Principle

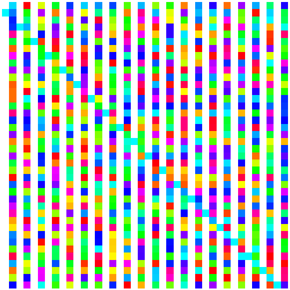
- Compute iteratively from 1-shifted form to d_1-shifted form
- each completed block appears in the increasing degree order
- until the shifted Hessenberg form is obtained:



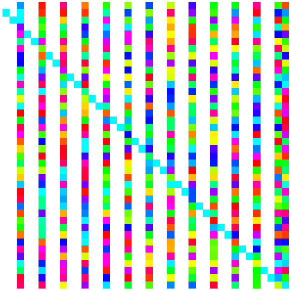
Example



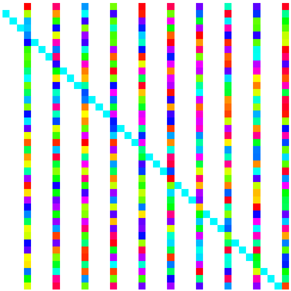
Example



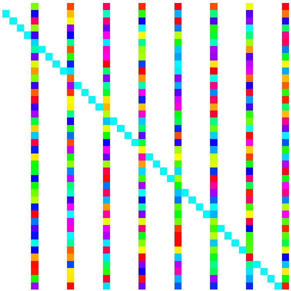
Example



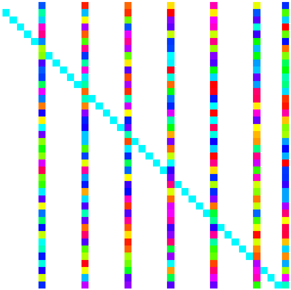
Example



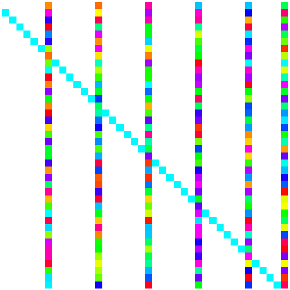
Example



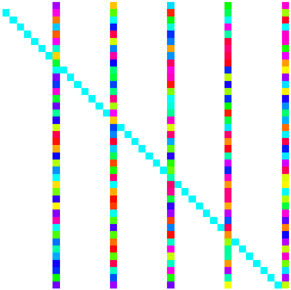
Example



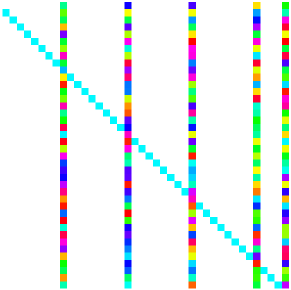
Example



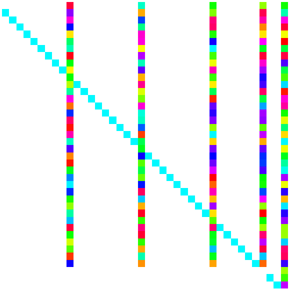
Example



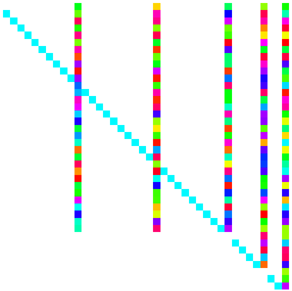
Example



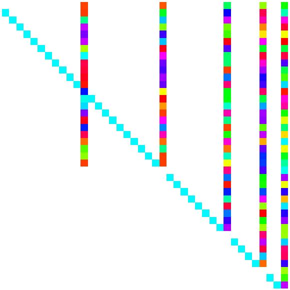
Example



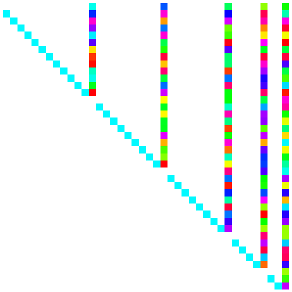
Example



Example



Example



Lemma

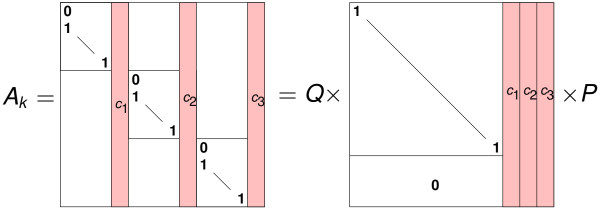
If $\#F > 2n^2$, the transformation will succeed with high probability. Failure is detected.

Lemma

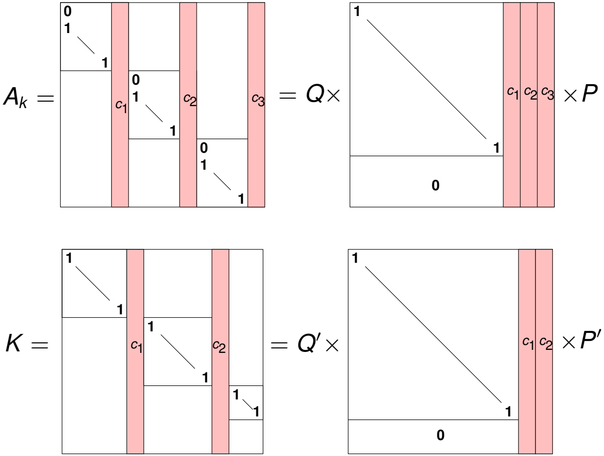
If $\#F > 2n^2$, the transformation will succeed with high probability. Failure is detected.

How to use fast matrix arithmetic ?

Permutations: compressing the dense columns



Permutations: compressing the dense columns



Reduction to Matrix multiplication

Similarity transformation:

$$K^{-1}AK = Q'^T \begin{bmatrix} I & * \\ 0 & * \end{bmatrix} P'^T Q \begin{bmatrix} I & * \\ 0 & * \end{bmatrix} PQ' \begin{bmatrix} I & * \\ 0 & * \end{bmatrix} P'$$

Reduction to Matrix multiplication

Similarity transformation:

$$K^{-1}AK = Q'^{T}\left(\begin{bmatrix} I & * \\ 0 & * \end{bmatrix}\left(P'^{T}Q\left(\begin{bmatrix} I & * \\ 0 & * \end{bmatrix}\left(PQ'\begin{bmatrix} I & * \\ 0 & * \end{bmatrix}\right)\right)\right)\right)P'$$

Reduction to Matrix multiplication

Similarity transformation:

$$K^{-1}AK = Q'^T \left(\begin{bmatrix} I & * \\ 0 & * \end{bmatrix} \left(P'^T Q \left(\begin{bmatrix} I & * \\ 0 & * \end{bmatrix} \left(PQ' \begin{bmatrix} I & * \\ 0 & * \end{bmatrix} \right) \right) \right) \right) P'$$

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

Reduction to Matrix multiplication

Similarity transformation:

$$K^{-1}AK = Q'^T \left(\begin{bmatrix} I & * \\ 0 & * \end{bmatrix} \left(P'^T Q \left(\begin{bmatrix} I & * \\ 0 & * \end{bmatrix} \left(PQ' \begin{bmatrix} I & * \\ 0 & * \end{bmatrix} \right) \right) \right) \right) P'$$

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

Rank profile: derived from LQUP

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

Reduction to Matrix multiplication

Similarity transformation:

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

Rank profile: derived from LQUP

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

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

A new type of reduction

$xI_n - A$

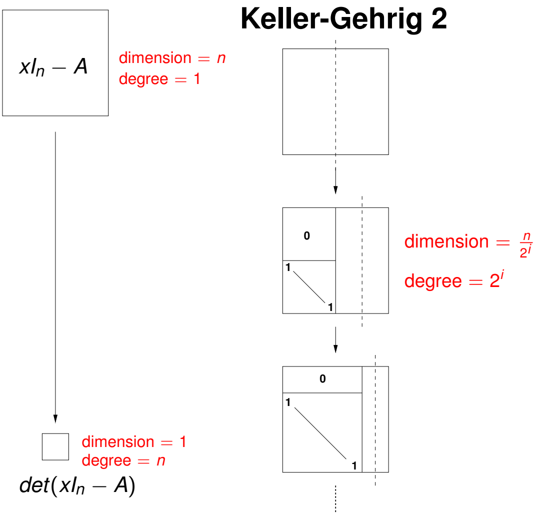
dimension $= n$
degree $= 1$

$\downarrow$

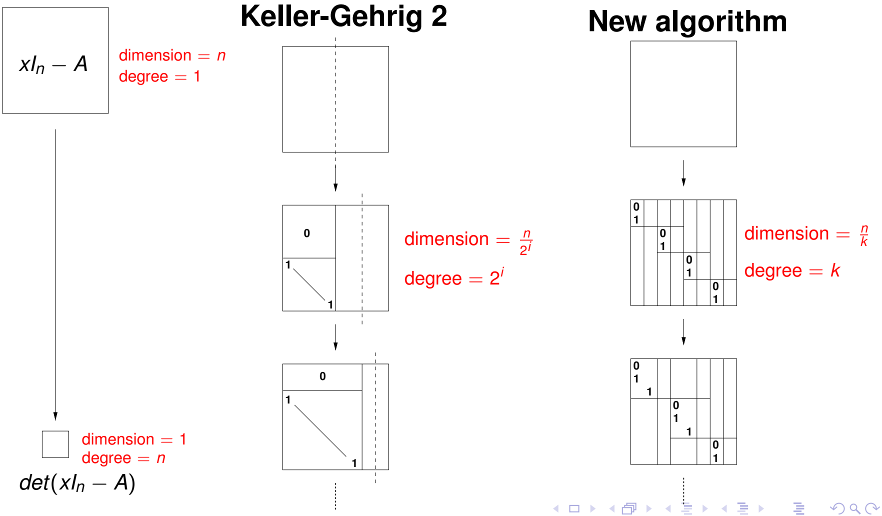
dimension $= 1$
degree $= n$

$det(xI_n - A)$

A new type of reduction



A new type of reduction



Outline

Improving the preconditioning

The preconditioning phase:

$$A \leftarrow U^{-1}AU$$

for a random matrix U.

(reminds [Kaltofen, Krishnamoorthy, Saunders 87])

Improving the preconditioning

The preconditioning phase:

$$A \leftarrow U^{-1}AU$$

for a random matrix U.

(reminds [Kaltofen, Krishnamoorthy, Saunders 87])

Instead, use a block Krylov preconditioning:

$$A \leftarrow V^{-1}AV,$$

$$V = \begin{bmatrix} W & AW & \dots & A^{c-1}W \end{bmatrix}$$

for a random $n \times n/c$ matrix W.

Improving the preconditioning

The preconditioning phase:

$$A \leftarrow U^{-1}AU$$

for a random matrix U.

(reminds [Kaltofen, Krishnamoorthy, Saunders 87])

Instead, use a block Krylov preconditioning:

$$A \leftarrow V^{-1}AV,$$

$$V = \begin{bmatrix} W & AW & \dots & A^{c-1}W \end{bmatrix}$$

for a random $n \times n/c$ matrix W.

Property

$V^{-1}AV$ is in c shifted form.

Efficiency balancing parameter

c small: full square matrix multiplications, but more ops
c large: tends to matrix-vector products, but less ops

Efficiency balancing parameter

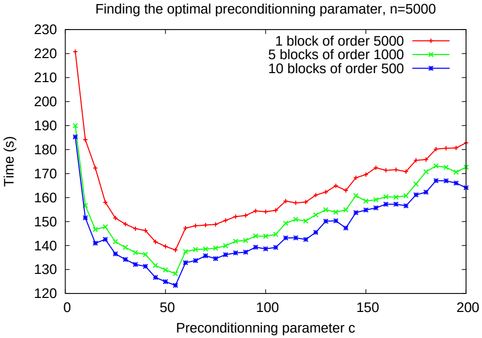
c small: full square matrix multiplications, but more ops
c large: tends to matrix-vector products, but less ops

$\Rightarrow$ parameter c balances efficiency

Efficiency balancing parameter

c small: full square matrix multiplications, but more ops
c large: tends to matrix-vector products, but less ops

$\Rightarrow$parameter c balances efficiency

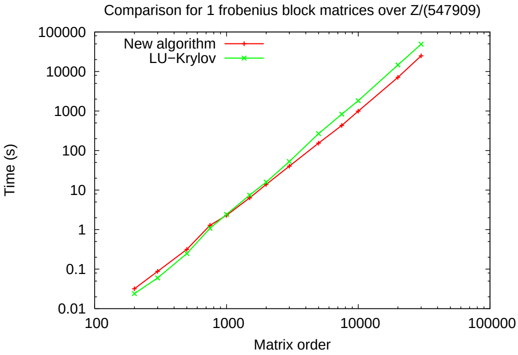


Experiments

n	LU-Krylov	New algorithm
200	0.024	0.032
300	0.06s	0.088s
500	0.248s	0.316s
750	1.084s	1.288s
1000	2.42s	2.296s
5000	267.6s	153.9s
10 000	1827s	991s
20 000	14 652s	7097s
30 000	48 887s	24 928s

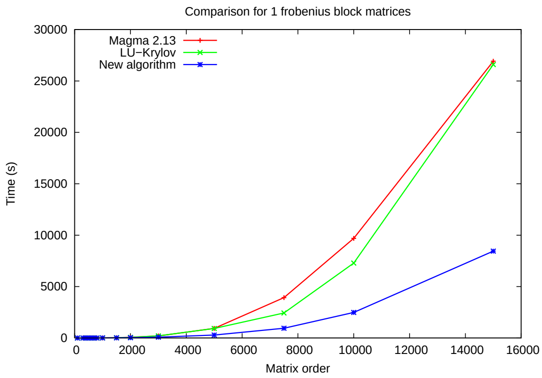
Computation time for 1 Frobenius block matrices, Itanium2-64 1.3Ghz, 192Gb

Experiments



Timing comparison between the new algorithm and LU-Krylov, logarithmic

Comparison to Magma and previous LinBox



Conclusion and perspectives

Results:

- Las Vegas reduction to matrix multiplication,
- The Frobenius normal form is easily derivable in $\mathcal{O}(n^\omega)$...
- ...but no transformation matrix
- Adaptive combination with block Krylov in practice.

Conclusion and perspectives

Results:

- Las Vegas reduction to matrix multiplication,
- The Frobenius normal form is easily derivable in $\mathcal{O}(n^{\omega})$...
- ...but no transformation matrix
- Adaptive combination with block Krylov in practice.

Still to be done:

- Condition on the size of the field is a limitation. Eberly's algorithm ?
- Ideally: derandomization ? (deterministic)
- Unification with matrix polynomial algorithms