State of the art
A new algorithm
The new algorithm into practice

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 $O(n^\omega)$ 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** $O(n^\omega)$ **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 $\mathbb{Z}_{547,909}$:

<table>
<thead>
<tr>
<th>$n$</th>
<th>500</th>
<th>5000</th>
<th>15000</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinBox</td>
<td>0.91s</td>
<td>4m44s</td>
<td>2h20m</td>
</tr>
<tr>
<td>magma-2.13</td>
<td>1.27s</td>
<td>15m32s</td>
<td>7h28m</td>
</tr>
</tbody>
</table>
State of the art

A new algorithm

Outline

1. State of the art

2. A new algorithm
   - Shifted forms
   - Principle of the new algorithm
   - Complexity

3. The new algorithm into practice
Outline

1. State of the art

2. A new algorithm
   - Shifted forms
   - Principle of the new algorithm
   - Complexity

3. The new algorithm into practice
Pre-Strassen age

**Leverrier 1840:** trace of powers of $A$, and Newton’s formula
- improved/rediscovered by Souriau, Faddeev, Frame and Csanky
- $O(n^4)$, 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
- $O(n^4)$, based on Matrix multiplication
- Suited for parallel computation model

**Danilevskii 1937:** elementary row/column operations

$\Rightarrow O(n^3)$
Pre-Strassen age

**Leverrier 1840:** trace of powers of $A$, and Newton’s formula
- improved/rediscovered by Souriau, Faddeev, Frame and Csanky
- $O(n^4)$, based on Matrix multiplication
- Suited for parallel computation model

**Danilevskii 1937:** elementary row/column operations
⇒ $O(n^3)$

**Hessenberg 1942:** transformation to quasi-upper triangular and determinant expansion formula.
⇒ $O(n^3)$
Preparata & Sarwate 1978: Update Csanky with fast matrix multiplication

\[ \mathcal{O}(n^{\omega+1}) \]
Post-Strassen age

**Preparata & Sarwate 1978:** Update Csanky with fast matrix multiplication

\[ \Rightarrow \mathcal{O}(n^{\omega+1}) \]

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

- \(\mathcal{O}(n^\omega \log n)\)
- the best complexity up to now
Post-Strassen age

Preparata & Sarwate 1978: Update Csanky with fast matrix multiplication
\[ \mathcal{O}(n^{\omega+1}) \]

Keller-Gehrig 1985, alg.1: computes \((A^2)^i\) for \(i = 1 \ldots \log_2 n\) to form a Krylov basis.
- \(\mathcal{O}(n\omega \log n)\)
- the best complexity up to now

Keller-Gehrig 1985, alg.2: inspired by Danilevskii, block operations
- \(\mathcal{O}(n^\omega)\)
- but only valid with generic matrices
Outline

1. State of the art

2. A new algorithm
   - Shifted forms
   - Principle of the new algorithm
   - Complexity

3. The new algorithm into practice
Definition (degree $d$ Krylov matrix of one vector $v$)

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

Property

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

$$C_{PA,v}^{min} \Rightarrow \text{if } d = n, K^{d-1}A = C_{PA}^{car}$$

[Reference: Keller-Gehrig, alg. 2]

Faster algorithms for the characteristic polynomial
State of the art
A new algorithm
The new algorithm into practice
Shifted forms
Principle of the new algorithm
Complexity

**Definition (degree \(d\) Krylov matrix of one vector \(v\))**

\[
K = \begin{bmatrix} v & Av & \ldots & A^{d-1}v \end{bmatrix}
\]

**Property**

\[
A \times K = K \times \begin{bmatrix} 0 & \ast \\ 1 & \ast \\ \vdots & \ast \\ 1 & \ast \end{bmatrix}_{\text{\(C_{PA,v}\) min}}
\]

\(\Rightarrow\) if \(d = n\),

\[
K^{-1}AK = C_{PA_{\text{car}}}
\]
Definition (degree $d$ Krylov matrix of one vector $v$)

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

Property

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

$$C_{P_{\min}}^{A,v}$$

⇒ if $d = n$,

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

[Keller-Gehrig, alg. 2] : $K^{-1}AK$ in $\mathcal{O}(n^\omega)$ for $A$ generic
Definition (degree $k$ Krylov matrix of several vectors $v_i$)

$$K = \begin{bmatrix} v_1 & \ldots & A^{k-1}v_1 & v_2 & \ldots & A^{k-1}v_2 & \ldots & v_l & \ldots & A^{k-1}v_l \end{bmatrix}$$

Property

$$A \times K = K \times$$

C. PERNET and A. STORJOHANN  Faster algorithms for the characteristic polynomial
Fact (Shift Hessenberg form)

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

\[
K = \begin{bmatrix}
v_1 & \ldots & A^{d_1-1}v_1 \\
& \ddots & \ddots \\
& & v_l & \ldots & A^{d_l-1}v_l
\end{bmatrix}
\]

is non-singular, then

\[
A \times K = K \times
\]
Principle

\(k\)-shifted form:

\[
\begin{array}{ccc}
0 & 1 & 0 \\
1 & 1 & 1 \\
\end{array}
\]

C. PERNET and A. STORJOHANN  Faster algorithms for the characteristic polynomial
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:
<table>
<thead>
<tr>
<th>State of the art</th>
<th>Shifted forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>A new algorithm</td>
<td>Principle of the new algorithm</td>
</tr>
<tr>
<td>The new algorithm into practice</td>
<td>Complexity</td>
</tr>
</tbody>
</table>

Example

C. PERNET and A. STORJOHANN
Faster algorithms for the characteristic polynomial
### Example

<table>
<thead>
<tr>
<th>State of the art</th>
<th>Shifted forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>A new algorithm</td>
<td>Principle of the new algorithm</td>
</tr>
<tr>
<td>The new algorithm into practice</td>
<td>Complexity</td>
</tr>
</tbody>
</table>

C. PERNET and A. STORJOHANN

Faster algorithms for the characteristic polynomial
Example
## Example

The following image illustrates the principle of the new algorithm:

![Illustration of the new algorithm](image)

C. Pernet and A. Storjohann

Faster algorithms for the characteristic polynomial
Example
Example

State of the art
A new algorithm
The new algorithm into practice

Shifted forms
Principle of the new algorithm
Complexity

C. PERNET and A. STORJOHANN
Faster algorithms for the characteristic polynomial
Example
Example
Example
Example

State of the art
A new algorithm
The new algorithm into practice

Shifted forms
Principle of the new algorithm
Complexity

C. PERNET and A. STORJOHANN
Faster algorithms for the characteristic polynomial
### Example

<table>
<thead>
<tr>
<th>State of the art</th>
<th>Shifted forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>A new algorithm</td>
<td>Principle of the new algorithm</td>
</tr>
<tr>
<td>The new algorithm into practice</td>
<td>Complexity</td>
</tr>
</tbody>
</table>

C. PERNET and A. STORJOHANN

Faster algorithms for the characteristic polynomial
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

\[ A_k = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times P \]
Permutations: compressing the dense columns

\[ A_k = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = Q \times \begin{bmatrix} c_1 & c_2 & c_3 \\ 0 & 1 & 1 \end{bmatrix} \times P \]

\[ K = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = Q' \times \begin{bmatrix} c_1 & c_2 & 1 \\ 0 & 1 & 1 \end{bmatrix} \times P' \]
Reduction to Matrix multiplication

Similarity transformation:

\[ K^{-1}AK = Q^T \begin{bmatrix} I & \ast \\ 0 & \ast \end{bmatrix} P'^T Q \begin{bmatrix} I & \ast \\ 0 & \ast \end{bmatrix} PQ' \begin{bmatrix} I & \ast \\ 0 & \ast \end{bmatrix} P' \]
Reduction to Matrix multiplication

Similarity transformation:

\[ K^{-1} AK = Q'^T \left( \begin{bmatrix} I & \ast \\ 0 & \ast \end{bmatrix} \left( P'^T Q \left( \begin{bmatrix} I & \ast \\ 0 & \ast \end{bmatrix} \left( PQ' \begin{bmatrix} I & \ast \\ 0 & \ast \end{bmatrix} \right) \right) \right) \right) P' \]
Reduction to Matrix multiplication

Similarity transformation:

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

\[ \Rightarrow O(k \left( \frac{n}{k} \right) \omega) \]
Reduction to Matrix multiplication

Similarity transformation:

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

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

Rank profile: derived from LQUP

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

Similarity transformation:

\[ \Rightarrow O \left( k \left( \frac{n}{k} \right)^\omega \right) \]

Rank profile: derived from LQUP

\[ \Rightarrow O \left( k \left( \frac{n}{k} \right)^\omega \right) \]

\[ \sum_{k=1}^{n} k \left( \frac{n}{k} \right)^\omega = n^\omega \sum_{k=1}^{n} \frac{1}{k^\omega - 1} = O \left( n^\omega \right) \]
A new type of reduction

\[ xI_n - A \]

dimension = \( n \)
degree = 1

\[ \text{det}(xI_n - A) \]

dimension = 1
degree = \( n \)

C. PERNET and A. STORJOHANN

Faster algorithms for the characteristic polynomial
A new type of reduction

\[ x l_n - A \]

- Dimension = \( n \)
- Degree = 1

\[ \text{det}(x l_n - A) \]

- Dimension = 1
- Degree = \( n \)

Keller-Gehrig 2

- Dimension = \( \frac{n}{2^i} \)
- Degree = \( 2^i \)

C. Pernet and A. Storjohann

Faster algorithms for the characteristic polynomial
A new type of reduction

\[ xI_n - A \]

Dimension = \( n \)
Degree = 1

\[ \det(xI_n - A) \]

Degree = \( n \)

Keller-Gehrig 2

Dimension = \( \frac{n}{2^i} \)
Degree = \( 2^i \)

New algorithm

Dimension = \( \frac{n}{k} \)
Degree = \( k \)

C. Pernet and A. Storjohann

Faster algorithms for the characteristic polynomial
Outline

1. State of the art
2. A new algorithm
   - Shifted forms
   - Principle of the new algorithm
   - Complexity
3. The new algorithm into practice
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:

\[ A \leftarrow V^{-1} AV, \]

\[ V = [W \ AW \ \ldots \ \ Ac^{c-1} W] \]

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:

\[ A \leftarrow V^{-1} AV, \]

\[ V = \begin{bmatrix} W & AW & \ldots & A^{c-1}W \end{bmatrix} \]

for a random \( n \times n/c \) matrix \( W \).

Property

\( V^{-1} AV \) is in \( c \) shifted form.
Efficiency balancing parameter

\( c \text{ small}: \) full square matrix multiplications, but more ops
\( c \text{ large}: \) 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

⇒ parameter \(c\) balances efficiency
## Experiments

<table>
<thead>
<tr>
<th>n</th>
<th>LU-Krylov</th>
<th>New algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>0.024</td>
<td>0.032</td>
</tr>
<tr>
<td>300</td>
<td>0.06s</td>
<td>0.088s</td>
</tr>
<tr>
<td>500</td>
<td>0.248s</td>
<td>0.316s</td>
</tr>
<tr>
<td>750</td>
<td>1.084s</td>
<td>1.288s</td>
</tr>
<tr>
<td>1000</td>
<td>2.42s</td>
<td>2.296s</td>
</tr>
<tr>
<td>5000</td>
<td>267.6s</td>
<td>153.9s</td>
</tr>
<tr>
<td>10000</td>
<td>1827s</td>
<td>991s</td>
</tr>
<tr>
<td>20000</td>
<td>14652s</td>
<td>7097s</td>
</tr>
<tr>
<td>30000</td>
<td>48887s</td>
<td>24928s</td>
</tr>
</tbody>
</table>

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

Comparison for 1 Frobenius block matrices over \(\mathbb{Z}/(547909)\)

Timing comparison between the new algorithm and LU-Krylov, logarithmic scales, Itanium2-64 1.3Ghz, 192Gb

Faster algorithms for the characteristic polynomial

C. PERNET and A. STORJOHANN
Comparison to Magma and previous LinBox

Comparison for 1 frobenius block matrices

- Magma 2.13
- LU–Krylov
- New algorithm

C. Pernet and A. Storjohann
Faster algorithms for the characteristic polynomial
Conclusion and perspectives

Results:

- Las Vegas reduction to matrix multiplication,
- The Frobenius normal form is easily derivable in $O(n^\omega)$...
- ...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 $O(n^\omega)$ ...
- ...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