On the complexity of the determinant

Erich Kaltofen
North Carolina State University
www.kaltofen.us

Joint work with Gilles Villard
ENS Lyon, France

Overview

1. Faster bit complexity without Strassen matrix multiplication

2. New speed-ups: the use of block vectors
   With Gilles Villard (middle)

3. Determinant computation without division

4. Matrices over polynomials; practicality issues

The LinBox project `www.linalg.org`

Objective a generic library for exact linear algebra
("Symbolic MatLab")

New abstraction mechanism black box matrix

Programming languages C++, Maple, GAP, C (Saclib)

Design principle
genericity through template parameter types (matrix entries)
and black box matrix model (sparseness and structuredness)

Participants 24 current researchers and students in USA, Canada and France

Fast matrix multiplication

Strassen's [1969] $O(n^{2.81})$ matrix multiplication algorithm

$$
\begin{array}{l|l}
m_1 \leftarrow (a_{1,2} - a_{2,2})(b_{2,1} - b_{2,2}) & \\
m_2 \leftarrow (a_{1,1} + a_{2,2})(b_{1,1} + b_{2,2}) & \\
m_3 \leftarrow (a_{1,1} - a_{2,1})(b_{1,1} + b_{1,2}) & \\
m_4 \leftarrow (a_{1,1} + a_{1,2})b_{2,2}) & a_{1,1}b_{1,1} + a_{1,2}b_{2,1} = m_1 + m_2 - m_4 + m_6 \\
m_5 \leftarrow a_{1,1}(b_{1,2} - b_{2,2}) & a_{1,1}b_{1,2} + a_{1,2}b_{2,2} = m_4 + m_5 \\
m_6 \leftarrow a_{2,2}(b_{2,1} - b_{1,1}) & a_{2,1}b_{1,1} + a_{2,2}b_{2,1} = m_6 + m_7 \\
m_7 \leftarrow (a_{2,1} + a_{2,2})b_{1,1}) & a_{2,1}b_{1,2} + a_{2,2}b_{2,2} = m_2 - m_3 + m_5 - m_7
\end{array}
$$

Problems reducible to matrix multiplication:

linear system solving, determinant [Bunch and Hopcroft 1974], characteristic polynomial [Keller-Gehrig 1985],...

Coppersmith and Winograd [1990]: $O(n^{2.38})$

Life after Strassen matrix multiplication: bit complexity

Linear system solving $x = A^{-1}b$ where $A \in \mathbb{Z}^{n\times n}$ and $b \in \mathbb{Z}^n$:

With Strassen and Chinese remaindering [McClellan 1973]:

Step 1: For prime numbers $p_1, \ldots, p_k$ Do
Solve $Ax^{[j]} \equiv b \pmod{p_j}$ where $x^{[j]} \in \mathbb{Z}/(p_j)$

Step 2: Chinese remainder $x^{[1]}, \ldots, x^{[k]}$ to $A\bar{x} \equiv b \pmod{p_1 \cdots p_k}$

Step 3: Recover denominators of x_i by continued fractions of $\dfrac{\bar{x}_i}{p_1 \cdots p_k}$.

Life after Strassen matrix multiplication: bit complexity

Linear system solving $x = A^{-1}b$ where $A \in \mathbb{Z}^{n\times n}$ and $b \in \mathbb{Z}^n$:

With Strassen and Chinese remaindering [McClellan 1973]:

Step 1: For prime numbers $p_1, \ldots, p_k$ Do
Solve $Ax^{[j]} \equiv b \pmod{p_j}$ where $x^{[j]} \in \mathbb{Z}/(p_j)$

Step 2: Chinese remainder $x^{[1]}, \ldots, x^{[k]}$ to $A\overline{x} \equiv b \pmod{p_1 \cdots p_k}$

Step 3: Recover denominators of x_i by continued fractions of $\dfrac{\overline{x}_i}{p_1 \cdots p_k}$.

Length of integers: $k = (n \max\{\log \|A\|, \log \|b\|\})^{1+o(1)}$

Bit complexity: $n^{3.38} \max\{\log \|A\|, \log \|b\|\}^{1+o(1)}$

With Hensel lifting [Moenck and Carter 1979, Dixon 1982]:

Step 1: For $j = 0, 1, \ldots, k$ and a prime p Do

Compute $\overline{x}^{[j]} = x^{[0]} + px^{[1]} + \cdots + p^j x^{[j]} \equiv x \pmod{p^{j+1}}$

1.a. $\widehat{b}^{[j]} = \dfrac{b - A\overline{x}^{[j-1]}}{p^j} = \dfrac{\widehat{b}^{[j-1]} - Ax^{[j-1]}}{p}$

1.b. $x^{[j]} \equiv A^{-1}\widehat{b}^{[j]} \pmod{p}$ re-using $A^{-1} \bmod p$

Step 2: Recover denominators of x_i by continued fractions of $\dfrac{\overline{x}_i^{[k]}}{p^k}$.

With Hensel lifting [Moenck and Carter 1979, Dixon 1982]:

Step 1: For $j = 0, 1, \ldots, k$ and a prime p Do

Compute $\overline{x}^{[j]} = x^{[0]} + px^{[1]} + \cdots + p^j x^{[j]} \equiv x \pmod{p^{j+1}}$

1.a. $\widehat{b}^{[j]} = \dfrac{b - A\overline{x}^{[j-1]}}{p^j} = \dfrac{\widehat{b}^{[j-1]} - Ax^{[j-1]}}{p}$

1.b. $x^{[j]} \equiv A^{-1}\widehat{b}^{[j]} \pmod{p}$ re-using $A^{-1} \bmod p$

Step 2: Recover denominators of x_i by continued fractions of $\dfrac{\overline{x}_i^{[k]}}{p^k}$.

With classical matrix arithmetic:

Bit complexity of 1.a: $(n \cdot \max\{\log \|A\|, \log \|b\|\} + n^2 \cdot \log \|A\|)^{1+o(1)}$

Total bit complexity: $(n^3 \cdot \log \|A\| + n^2 \cdot \log \|b\|\ )^{1+o(1)}$

Diophantine solutions
by Giesbrecht, Mulders&Storjohann:
Find several rational solutions.

$$A(\tfrac{1}{2}x^{[1]}) = b, \quad x^{[1]} \in \mathbb{Z}^n$$
$$A(\tfrac{1}{3}x^{[2]}) = b, \quad x^{[2]} \in \mathbb{Z}^n$$
$$\gcd(2,3) = 1 = 2 \cdot 2 - 1 \cdot 3$$
$$A(2x^{[1]} - x^{[2]}) = 4b - 3b = b$$

$\Longrightarrow$Can compute **integral** solutions of **sparse** linear systems.

Matrix determinant definition

$$\det(Y) = \det\left(\begin{bmatrix} y_{1,1} & \cdots & y_{1,n} \\ y_{2,1} & \cdots & y_{2,n} \\ \vdots & & \vdots \\ y_{n,1} & \cdots & y_{n,n} \end{bmatrix}\right) = \sum_{\sigma \in S_n} \left(\operatorname{sign}(\sigma) \prod_{i=1}^{n} y_{i,\sigma(i)} \right),$$

where $y_{i,j}$ are from an *arbitrary commutative ring*, and S_n is the set of all permutations on $\{1, 2, \ldots, n\}$.

Interesting rings: $\mathbb{Z}$, $\mathbb{K}[x_1, \ldots, x_n]$, $\mathbb{K}[x]/(x^n)$

An important algebraic reduction and de-randomization

Theorem [Strassen 1973, Baur and Strassen 1983]
Suppose you have a Monte Carlo randomized algorithm on a ***random access machine*** *that can compute the determinant of an* $n \times n$ *matrix in* $D(n)$ *arithmetic operations.*

Then for any $\varepsilon > 0$ *you have a* ***deterministic*** *algorithm on a random access machine that can multiply two* $n \times n$ *matrices in* $O(D(n)^{1+\varepsilon})$ *arithmetic operations.*

An important algebraic reduction and de-randomization

Theorem [Strassen 1973, Baur and Strassen 1983]
Suppose you have a Monte Carlo randomized algorithm on a ***random access machine*** *that can compute the determinant of an* $n \times n$ *matrix in* $D(n)$ *arithmetic operations.*

Then for any $\varepsilon > 0$ *you have a* ***deterministic*** *algorithm on a random access machine that can multiply two* $n \times n$ *matrices in* $O(D(n)^{1+\varepsilon})$ *arithmetic operations.*

By fast LU-factorization we thus obtain a **deterministic** algorithm that can compute the determinant of an $n \times n$ matrix in $O(D(n)^{1+\varepsilon})$ arithmetic operations.

Bit complexity of the determinant

With Chinese remaindering: $(n \cdot \log ||A||)^{1+o(1)}$ times matrix multiplication complexity.

Sign of the determinant [Clarkson 92]: $n^{4+o(1)}$ if matrix has large orthogonal defect.

Using denominators of linear system solutions [Pan 1989; Abbott, Bronstein, Mulders 1999]: fast when large first invariant factor.

Using fast Smith form method $n^{3.5+o(1)}(\log ||A||)^{1.5+o(1)}$ [Eberly, Giesbrecht, Villard 2000]

Wiedemann's 1986 determinant algorithm

For $u, v \in \mathbb{F}^n$ and $A \in \mathbb{F}^{n \times n}$ and consider the sequence of field elements

$$a_0 = u^T v,\ a_1 = u^T A v,\ a_2 = u^T A^2 v,\ a_3 = u^T A^3 v,\ \ldots$$

Let $f^{(A)}(\lambda) = c_0 + c_1\lambda + \cdots + c_k\lambda^k \in \mathbb{F}[\lambda]$ with $f^{(A)}(A) = 0$. Since $u^T A^j f^{(A)}(A) v = 0$, we have

$$\forall\, j \geq 0 \colon\ c_0 a_{0+j} + c_1 a_{1+j} + \cdots + c_k a_{k+j} = 0,$$

that is, $\{a_i\}_{i=0,1,\ldots}$ satisfies a linear recurrence.

Wiedemann's 1986 determinant algorithm

For $u, v \in \mathbb{F}^n$ and $A \in \mathbb{F}^{n \times n}$ and consider the sequence of field elements

$$a_0 = u^T v,\ a_1 = u^T A v,\ a_2 = u^T A^2 v,\ a_3 = u^T A^3 v,\ \ldots$$

Let $f^{(A)}(\lambda) = c_0 + c_1\lambda + \cdots + c_k\lambda^k \in \mathbb{F}[\lambda]$ with $f^{(A)}(A) = 0$. Since $u^T A^j f^{(A)}(A) v = 0$, we have

$$\forall\, j \geq 0 \colon\ c_0 a_{0+j} + c_1 a_{1+j} + \cdots + c_k a_{k+j} = 0,$$

that is, $\{a_i\}_{i=0,1,\ldots}$ satisfies a linear recurrence.

By the Berlekamp/Massey (1969) we can compute in $n^{1+o(1)}$ operations a minimal linear generator for $\{a_i\}_{i=0,1,\ldots}$

Wiedemann's 1986 determinant algorithm

For $u, v \in \mathbb{F}^n$ and $A \in \mathbb{F}^{n \times n}$ and consider the sequence of field elements

$$a_0 = u^T v,\ a_1 = u^T A v,\ a_2 = u^T A^2 v,\ a_3 = u^T A^3 v,\ \ldots$$

Let $f^{(A)}(\lambda) = c_0 + c_1\lambda + \cdots + c_k\lambda^k \in \mathbb{F}[\lambda]$ with $f^{(A)}(A) = 0$. Since $u^T A^j f^{(A)}(A) v = 0$, we have

$$\forall\, j \geq 0 \colon\ c_0 a_{0+j} + c_1 a_{1+j} + \cdots + c_k a_{k+j} = 0,$$

that is, $\{a_i\}_{i=0,1,\ldots}$ satisfies a linear recurrence.

By the Berlekamp/Massey (1969) we can compute in $n^{1+o(1)}$ operations a minimal linear generator for $\{a_i\}_{i=0,1,\ldots}$

Wiedemann randomly perturbs A and chooses random u and v; then $\det(\lambda I - A) =$ the minimal recurrence polynomial of $\{a_i\}_{i=0,1,\ldots 2n-1}$.

Baby steps/giant steps algorithm [Kaltofen 1992/2000]

Detail of sequence $a_i = u^T A^i v$ computation

Let $r = \lceil \sqrt{2n} \rceil$ and $s = \lceil 2n/r \rceil$.

Substep 1. For $j = 1, 2, \ldots, r-1$ Do $v^{[j]} \leftarrow A^j v$;

Substep 2. $Z \leftarrow A^r$;
[$O(n^3)$ operations; integer length $(\sqrt{n}\ \log \|A\|)^{1+o(1)}$]

Substep 3. For $k = 1, 2, \ldots, s$ Do ${u^{[k]}}^T \leftarrow u^T Z^k$;
[$O(n^{2.5})$ operations; integer length $(n\ \log \|A\|)^{1+o(1)}$]

Substep 4. For $j = 0, 1, \ldots, r-1$ Do
For $k = 0, 1, \ldots, s$ Do $a_{kr+j} \leftarrow \langle u^{[k]}, v^{[j]} \rangle$.

Overall bit complexity $(n^{3+1/2} \log \|A\|)^{1+o(1)}$.

Speed-up with fast matrix multiplication

Suppose $k \times k$ matrices can be multiplied in $O(k^{2.3755})$ ring operations.

Suppose $k \times k^{0.29462}$ can be multiplied in $k^{2+o(1)}$ ring operations.

Overall bit complexity reduces to $n^{3.0281}(\log \|A\|)^{1+o(1)}$ bit operations.

Speed-up with fast matrix multiplication

Suppose $k \times k$ matrices can be multiplied in $O(k^{2.3755})$ ring operations.

Suppose $k \times k^{0.29462}$ can be multiplied in $k^{2+o(1)}$ ring operations.

Overall bit complexity reduces to $n^{3.0281}(\log \|A\|)^{1+o(1)}$ bit operations.

Speed-up with early termination

If the determinant is small, it is possible to terminate early via Chinese remaindering [Kaltofen 2002].

Coppersmith's 1992 blocking

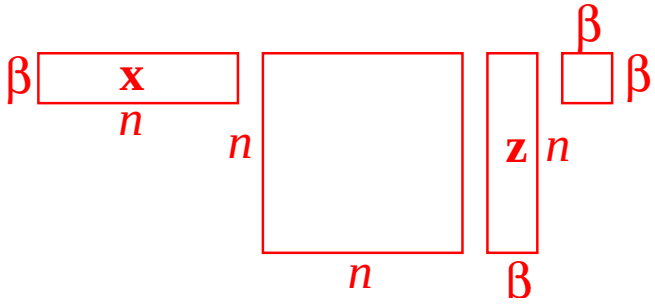
Use of the block vectors $\mathbf{x} \in \mathbb{F}^{n\times\beta}$ in place of u

$\mathbf{y} \in \mathbb{F}^{n\times\beta}$ in place of v

$$\mathbf{a}_i = \mathbf{x}^T A^i \mathbf{y} \in \mathbb{F}^{\beta\times\beta}, \quad 0 \le i < 2n/\beta + 2.$$

Find a minimal **matrix** polynomial generator
$\mathbf{c}_0\lambda^0 + \cdots + \mathbf{c}_d\lambda^d \in \mathbb{F}^{\beta\times\beta}[\lambda]$, $d = \lceil n/\beta \rceil$:

$$\forall\, j \ge 0\colon \sum_{i=0}^{d} \mathbf{a}_{j+i}\mathbf{c}_i = \sum_{i=0}^{d} \mathbf{x}^T A^{i+j}\mathbf{y}\mathbf{c}_i = \mathbf{0} \in \mathbb{F}^{\beta\times\beta}$$



Note: A must be in general position, otherwise $d > \lceil n/\beta \rceil$ and more sequence elements are needed.

Advantages of blocking

1. Sequence is shorter, therefore intermediate integers are shorter.

2. Parallel computation of each column of the sequence is possible

Disadvantages of blocking

1. Block Berlekamp/Massey step more intricate
   and more expensive: $\beta^{1.3755} n^{1+o(1)}$

   [Beckermann and Labahn 1994]

2. Must compute $\det(\mathbf{c}_0 + \cdots + \mathbf{c}_d \lambda^d)$, which costs extra.
   After preconditioning A, with high probability

$$\det(\lambda I - A) = \det(\mathbf{c}_0 + \cdots + \mathbf{c}_d \lambda^d).$$

Sketch of multivariable control theory

From $(\lambda I - A)^{-1} = I\lambda^{-1} + A\lambda^{-2} + A^2\lambda^{-3} + \cdots$

$$\mathbf{x}^T(\lambda I - A)^{-1}\mathbf{y}(\mathbf{c}_0 + \cdots + \mathbf{c}_d\lambda^d) = R(\lambda) \in \mathbb{F}[\lambda]^{\beta\times\beta}$$

we obtain a matrix Padé approximation ("realization")

$$\mathbf{x}^T(\lambda I - A)^{-1}\mathbf{y} = \sum_i \mathbf{a}_i\lambda^i = R(\lambda)(\mathbf{c}_0 + \cdots + \mathbf{c}_d\lambda^d)^{-1}$$

Denominator on left side: $\det(\lambda I - A)$.
Denominator on right side: $\det(\mathbf{c}_0 + \cdots + \mathbf{c}_d\lambda^d)$.

Sketch of multivariable control theory

From $(\lambda I - A)^{-1} = I\lambda^{-1} + A\lambda^{-2} + A^2\lambda^{-3} + \cdots$

$$\mathbf{x}^T(\lambda I - A)^{-1}\mathbf{y}(\mathbf{c}_0 + \cdots + \mathbf{c}_d\lambda^d) = R(\lambda) \in \mathbb{F}[\lambda]^{\beta\times\beta}$$

we obtain a matrix Padé approximation ("realization")

$$\mathbf{x}^T(\lambda I - A)^{-1}\mathbf{y} = \sum_i \mathbf{a}_i\lambda^i = R(\lambda)(\mathbf{c}_0 + \cdots + \mathbf{c}_d\lambda^d)^{-1}$$

Denominator on left side: $\det(\lambda I - A)$.
Denominator on right side: $\det(\mathbf{c}_0 + \cdots + \mathbf{c}_d\lambda^d)$.

Theorem 1

The determinant of an integer matrix can be computed in $n^{2.6973}(\log\|A\|)^{1+o(1)}$ *bit operations (at* $\beta = n^{0.507}$ *and giant stepping* $s = n^{0.172}$*).*

Division-free determinant complexity

Special sequence for Berlekamp/Massey

```
In[2]:= S = {1,1,2,3,6,10,20,35,70,126,252,462,924,1716}
In[3]:= BM[S, x]
Discrepancy for r = 1 is 1
L updated to 1, Lambda = 1
Discrepancy for r = 2 is 1
Lambda updated to 1 - x
Discrepancy for r = 3 is 1
                                    2
L updated to 2, Lambda = 1 - x - x
Discrepancy for r = 4 is 0
Discrepancy for r = 5 is 1
                                      2    3
L updated to 3, Lambda = 1 - x - 2 x  + x
Discrepancy for r = 6 is 0
Discrepancy for r = 7 is 1
                                      2      3    4
L updated to 4, Lambda = 1 - x - 3 x  + 2 x  + x
Discrepancy for r = 8 is 0
Discrepancy for r = 9 is 1
                                      2      3      4    5
L updated to 5, Lambda = 1 - x - 4 x  + 3 x  + 3 x  - x
```

```
Discrepancy for r = 10 is 0
Discrepancy for r = 11 is 1
                                      2      3      4      5
L updated to 6, Lambda = 1 - x - 5 x  + 4 x  + 6 x  - 3 x
   6
- x
Discrepancy for r = 12 is 0
Discrepancy for r = 13 is 1
                                      2      3       4      5
L updated to 7, Lambda = 1 - x - 6 x  + 5 x  + 10 x  - 6 x
     6    7
- 4 x  + x
Discrepancy for r = 14 is 0
```

Special case for Wiedemann's determinant algorithm: for

$$C = \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ c_0 & c_1 & \dots & c_{n-2} & c_{n-1} \end{bmatrix} \qquad c_i = (-1)^{\lfloor (n-i-1)/2 \rfloor} \binom{\lfloor (n+i)/2 \rfloor}{i}$$

and

$$a_i = \underbrace{\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix}}_{u^T = e_1^T} \times C^i \times v, \qquad v = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}, \qquad a_i = \binom{i}{\lfloor i/2 \rfloor}$$

the algorithm needs no divisions/decisions.

Block algorithm: $\mathbf{x} = \begin{bmatrix} u & & \\ & \ddots & \\ & & u \end{bmatrix}$, $\begin{bmatrix} C & & \\ & \ddots & \\ & & C \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} v & & \\ & \ddots & \\ & & v \end{bmatrix}$.

Strassen's homotopy:

Compute $\det(C + z(A - C))$ by truncated power series operations in $\mathbb{Z}[z]/(z^{n+1})$.

Polynomials in z are like integers: length $\leftrightarrow$ degree.

Theorem 2

The determinant and adjoint of a matrix over a commutative ring can be computed with $O(n^{2.6973})$ ring additions, subtractions and multiplications. The characteristic polynomial with $O(n^{2.8066})$ ring additions, subtractions and multiplications.

More recent results

Storjohann 2002: determinant of matrix with polynomial entries in $n^{2.3755} \times (\text{input degree})^{1+o(1)}$ field operations;
possibly generalizable to integers.

Jeannerod and Villard 2003: inverse of matrix with polynomial entries in $(n^3 \times (\text{input degree}))^{1+o(1)}$ straight-line steps.

Practicability

Without Strassen matrix multiplication and Knuth/Schönhage GCD we have $(n^{3+1/3}\log\|A\|)^{1+o(1)}$ bit operations.

$+o(1)$ from Chinese remaindering, but $\log_2\left(\prod_{\substack{p \text{ prime}\\ p\le 2^{32}}} p\right) > 0.5\cdot 10^{10}$.

Poly-logarithmic factors in further speedups are killers:

for $n_1 = 10000$: $\dfrac{\log_2 n_1}{n_1^{1/3}} > ???$

Practicability

Without Strassen matrix multiplication and Knuth/Schönhage GCD we have $(n^{3+1/3}\log\|A\|)^{1+o(1)}$ bit operations.

$+o(1)$ from Chinese remaindering, but $\log_2\left(\prod_{\substack{p \text{ prime}\\ p\le 2^{32}}} p\right) > 0.5\cdot 10^{10}$.

Poly-logarithmic factors in further speedups are killers:

for $n_1 = 10000$: $\dfrac{\log_2 n_1}{n_1^{1/3}} > 0.616$

Practicability

Without Strassen matrix multiplication and Knuth/Schönhage GCD we have $(n^{3+1/3} \log \|A\|)^{1+o(1)}$ bit operations.

$+o(1)$ from Chinese remaindering, but $\log_2 \left(\prod_{\substack{p \text{ prime} \\ p \leq 2^{32}}} p \right) > 0.5 \cdot 10^{10}$.

Poly-logarithmic factors in further speedups are killers:

for $n_1 = 10000$: $\dfrac{\log_2 n_1}{n_1^{1/3}} > 0.616$

We have run the algorithm against Gaussian elimination and it barely wins. However, the block version is **highly parallelizable**.