On the complexity of the determinant

Erich Kaltofen
North Carolina State University
www.kaltofen.us

Joint work with Gilles Villard
ENS Lyon, France

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

Theorem [Giesbrecht 1992] *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 you have a Monte Carlo randomized algorithm on a random access machine that can multiply two $n \times n$ *matrices in* $O(D(n))$ *arithmetic operations.*

No proof is known for Las Vegas or deterministic algorithms. (At the conference, Peter Bürgisser pointed out to me that $D(n)^{1+o(1)}$ is achievable deterministically by designing a divide-and-conquer matrix multiplication algorithm from a sufficiently large fixed dimension.)

Bit complexity of the determinant

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

Sign of the determinant [Clarkson 92]: $n^{4+o(1)}$ if matrix is ill-conditioned.

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

Using fast Smith form method $n^{3.5+o(1)}(\log \|A\|)^{2.5+o(1)}$ [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.

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.

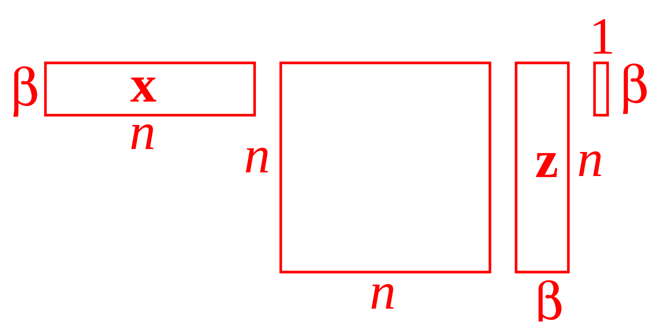
Coppersmith's 1992 blocking

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
$\mathbf{c}_0\lambda^0 + \cdots + \mathbf{c}_d\lambda^d \in \mathbb{F}^{\beta}[\lambda]$, $d = \lceil n/\beta \rceil$:

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



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

Advantages of blocking

Sequence is shorter, therefore intermediate integers are shorter.

Disadvantages of blocking

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

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

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

Sketch of multivariable control theory

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

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

we obtain a matrix Padé approximation (“realization”)

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

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

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

L updated to 2, Lambda = $1 - x - x^2$

Discrepancy for r = 4 is 0

Discrepancy for r = 5 is 1

L updated to 3, Lambda = $1 - x - 2x^2 + x^3$

Discrepancy for r = 6 is 0

Discrepancy for r = 7 is 1

L updated to 4, Lambda = $1 - x - 3x^2 + 2x^3 + x^4$

Discrepancy for r = 8 is 0

Discrepancy for r = 9 is 1

L updated to 5, Lambda = $1 - x - 4x^2 + 3x^3 + 3x^4 - x^5$

```
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

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

and

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

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, 2003: determinant of matrix with polynomials/integers in $n^{2.3755} \times (\text{input degree/length})^{1+o(1)}$ field/bit operations.

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

Note: automatic differentiation does not preserve bit complexity:

$x^T yc$ where x, y are vectors with constant entries,
c a large constant

takes $O(n + \log|c|)$ bit operations,
yc takes $O(n \log|c|)$ bit operations [Villard 2003].