On the complexity of computing determinants

Erich Kaltofen
North Carolina State University
www.kaltofen.net

Overview

1. Second beginning:
   Faster bit complexity without Strassen matrix multiplication

2. First beginning:
   Determinant computation without division

3. New speed-ups: the use of blocking
   With Gilles Villard (middle)

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

1. Bit complexity of linear algebra problems

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 [Bunch and Hopcroft 1974],...

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

Life after Strassen: 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 [McClellan 1973]:

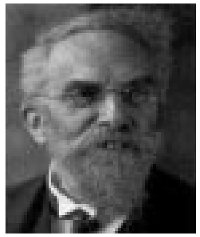
Step 1: For prime numbers $p_1, \ldots p_k$ Do
Solve $Ax^{[j]} \equiv b \pmod{p_j}$ where $x^{[j]} \in \mathbb{Z}/(p_j)$

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]:

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

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

1.a. $b^{[j]} = \dfrac{b - A\bar{x}^{[j-1]}}{p^j} = \dfrac{b - (A\bar{x}^{[j-2]} + A p^{j-1} x^{[j-1]})}{p^j}$

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

Step 3: Recover denominators of x_i by continued fractions of $\dfrac{\bar{x}_i^{[k]}}{p^k}$.

With classical matrix arithmetic:

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

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

Bit complexity of the determinant

Wiedemann's [1986] determinant algorithm
For $u, v \in \mathbb{K}^n$ and $A \in \mathbb{K}^{n\times n}$ consider the sequence of field elements

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

The minimal polynomial of A linearly generates $\{a_i\}_{i=0,1,\ldots}$.

By the Berlekamp/Massey [1967] algorithm we can compute in $n^{1+o(1)}$ arithmetic operations a minimal linear generator for $\{a_i\}_{i=0,1,\ldots}$.

Wiedemann randomly perturbs A and chooses random u and v; then

$$\det(\lambda I - A) = \text{minimal recurrence polynomial of } \{a_i\}_{i=0,1,\ldots}.$$

Detail of algorithm
[exactly like my division-free determinant algorithm ISSAC 92]

For $i = 0, 1, \ldots, 2n-1$ Do Compute the $a_i = u^T A^i v$;

Done by baby steps/giant steps: 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 $a_{kr+j} \leftarrow \langle u^{[k]}, v^{[j]} \rangle$.

Using fast rectangular matrix multiplication: $O(n^{3.064} \log \|A\|)$

Problem 1 (from my 3ECM 2000 talk)
Improve the bit complexity of algorithms for the determinant, resultant, linear system solution, over the integers.

2. Determinant computation without division

Gauss's elimination (1826)

Let $y_{j,k}^{(0)} = y_{j,k}$.

For $i \leftarrow 1,\ldots,n-1$ Do

For $j \leftarrow i+1,\ldots,n$ Do

For $k \leftarrow i+1,\ldots,n$ Do

$$y_{j,k}^{(i)} \leftarrow y_{j,k}^{(i-1)} - \frac{y_{j,i}^{(i-1)} y_{i,k}^{(i-1)}}{y_{i,i}^{(i-1)}}$$

$\det(Y) \leftarrow y_{1,1}^{(0)} y_{2,2}^{(1)} \cdots y_{n,n}^{(n-1)}$

Exact division elimination

Observe that $y^{(i)}_{j,k} = \frac{\overbrace{\det(Y^{(0)}_{1,\dots,i,j;1,\dots,i,k})}^{b^{(i)}_{j,k}}}{\det(Y^{(0)}_{1,\dots,i;1,\dots,i})}$.

Let $B^{(0)} = Y$ and $b^{(-1)}_{0,0} = 1$.

For $i \leftarrow 1,\dots,n-1$ Do

For $j \leftarrow i+1,\dots,n$ Do

For $k \leftarrow i+1,\dots,n$ Do

$$b^{(i)}_{j,k} \leftarrow \left(b^{(i-1)}_{j,k} b^{(i-1)}_{i,i} - b^{(i-1)}_{j,i} b^{(i-1)}_{i,k}\right) \Big/ b^{(i-2)}_{i-1,i-1}$$

this division is exact;

$\det(Y) \leftarrow b^{(n-1)}_{n,n}$.

Division-free elimination (Sasaski and Murao 1982)

Use exact divsion algorithm on $B^{(0)} = \lambda I_n + Y$,
where I_n is the $n \times n$ identity matrix.
Then all exact divisions are by the monic polynomials

$$b_{i,i}^{(i-1)} = \det(\lambda I_i + Y_{1,\dots,i;1,\dots,i}) = \lambda^i + \text{lower order terms.}$$

Polynomial arithmetic over a commutative ring costs $O(n \log n \log\log n)$ ring operations (Schönhage and Strassen 1971, Sieveking 1972, Kung 1974, Cantor and Kaltofen 1991).

Algorithm computes $\det(\lambda I - Y)$ with

$$O(n^4 \log n \log\log n) \quad +, -, \text{ and } \times \text{ operations.}$$

Table of division-free determinant algorithms

Strassen 1973	General method for eliminating divisions; same complexity for determinant computation as Sasaki and Murao.
Preparata and Sarwate 1978	$O(n^3\sqrt{n})$ method with divisions by $2, 3, 5, 7, \ldots, n$.
Berkowitz 1984, Chistov 1985	$O(n^4 \log n)$ method.
Kaltofen 1992	$O(n^3\sqrt{n} \log n \operatorname{loglog} n)$ method.
Kaltofen & Villard 2000	$O(n^3\sqrt[3]{n} \log n \operatorname{loglog} n)$ method.

All methods run faster with fast matrix multiplication. E.g., our 2000 method yields $O(n^{2.809})$ ring operations.

All results extend to computing the adjoint of an $n \times n$ matrix by computing all partial derivatives (Baur and Strassen 1983).

Ingredients in my division-free determinant algorithm

— Use of Strassen's general approach to eliminating divisions

— Based on Wiedemann's determinant algorithm

— Construction of special instance for Wiedemann's algorithm

— Baby steps/giant steps method in a substep

— NEW: Coppersmith's 1992 blocking

Show Maple B/M run here

```
> with(combinat):
Warning, the protected name Chi has been redefined and unprotected
> for i from 0 to 20 do a[i]:=fibonacci(i); od;
```

$$a_0 := 0$$
$$a_1 := 1$$
$$a_2 := 1$$
$$a_3 := 2$$
$$a_4 := 3$$
$$a_5 := 5$$
$$a_6 := 8$$
$$a_7 := 13$$
$$a_8 := 21$$
$$a_9 := 34$$
$$a_{10} := 55$$
$$a_{11} := 89$$
$$a_{12} := 144$$
$$a_{13} := 233$$
$$a_{14} := 377$$
$$a_{15} := 610$$
$$a_{16} := 987$$
$$a_{17} := 1597$$
$$a_{18} := 2584$$
$$a_{19} := 4181$$

$$a_{20} := 6765$$

```
> read("BM.mpl");
> bermass(a,20);
```

"Delta[", 1, "]=", 0

$$1$$

"Delta[", 2, "]=", 1

$$1$$

"Delta[", 3, "]=", 1

$$1 - z$$

"Delta[", 4, "]=", 1

$$1 - z - z^2$$

"Delta[", 5, "]=", 0

$$1 - z - z^2$$

"Delta[", 6, "]=", 0

$$1 - z - z^2$$

"Delta[", 7, "]=", 0

$$1 - z - z^2$$

"Delta[", 8, "]=", 0

$$1 - z - z^2$$

"Delta[", 9, "]=", 0

$$1 - z - z^2$$

"Delta[", 10, "]=", 0

$$1 - z - z^2$$

"Delta[", 11, "]=", 0

$$1 - z - z^2$$

"Delta[", 12, "]=", 0

$$1 - z - z^2$$

"Delta[", 13, "]=", 0

$$1 - z - z^2$$

"Delta[", 14, "]=", 0

$$1 - z - z^2$$

"Delta[", 15, "]=", 0

$$1 - z - z^2$$

"Delta[", 16, "]=", 0

$$1 - z - z^2$$

"Delta[", 17, "]=", 0

$$1 - z - z^2$$

"Delta[", 18, "]=", 0

$$1 - z - z^2$$

"Delta[", 19, "]=", 0

$$1 - z - z^2$$

"Delta[", 20, "]=", 0

$$1 - z - z^2$$

$$z^2 - z - 1$$

```
> a[6]:=9;
```

$$a_6 := 9$$

```
> bermass(a,20);
```

"Delta[", 1, "]=", 0

$$1$$
“Delta[”, 2, “]=”, 1
$$1$$
“Delta[”, 3, “]=”, 1
$$1-z$$
“Delta[”, 4, “]=”, 1
$$1-z-z^2$$
“Delta[”, 5, “]=”, 0
$$1-z-z^2$$
“Delta[”, 6, “]=”, 0
$$1-z-z^2$$
“Delta[”, 7, “]=”, 1
$$1-z-z^2-z^5$$
“Delta[”, 8, “]=”, -2
$$1+z-3z^2-z^5-2z^3$$
“Delta[”, 9, “]=”, -5
$$1+z+2z^2-z^5-7z^3-5z^4$$
“Delta[”, 10, “]=”, -10
$$1+z+2z^2-11z^5+3z^3-15z^4$$
“Delta[”, 11, “]=”, -20
$$1+z+2z^2-31z^5+3z^3+5z^4-20z^6$$
“Delta[”, 12, “]=”, -39
$$1-\frac{19}{20}z+\frac{1}{20}z^2-\frac{7}{4}z^5-\frac{9}{10}z^3-\frac{17}{20}z^4+\frac{29}{20}z^6$$

$$\text{“Delta[”}, 13, \text{“]=”}, \frac{81}{20}$$

$$1-\frac{19}{20}z+\frac{101}{400}z^2-\frac{457}{400}z^5-\frac{279}{400}z^3-\frac{89}{200}z^4-\frac{127}{80}z^6-\frac{891}{400}z^7$$

$$\text{“Delta[”}, 14, \text{“]=”}, \frac{201}{400}$$

$$1-\frac{29}{27}z+\frac{10}{27}z^2-\frac{28}{27}z^5-\frac{19}{27}z^3-\frac{1}{3}z^4-\frac{37}{27}z^6-\frac{65}{27}z^7$$

$$\text{“Delta[”}, 15, \text{“]=”}, \frac{34}{9}$$

$$1-\frac{29}{27}z-\frac{410}{729}z^2-\frac{16}{81}z^5+\frac{133}{729}z^3-\frac{277}{729}z^4-\frac{421}{729}z^6-\frac{565}{729}z^7-\frac{986}{729}z^8$$

$$\text{“Delta[”}, 16, \text{“]=”}, \frac{517}{243}$$

$$1-\frac{167}{102}z+\frac{13}{306}z^2-\frac{1}{102}z^5-\frac{4}{153}z^3+\frac{5}{306}z^4+\frac{1}{153}z^6-\frac{1}{306}z^7+\frac{1}{306}z^8$$

$$\text{“Delta[”}, 17, \text{“]=”}, \frac{-1}{306}$$

$$1-\frac{167}{102}z+\frac{451}{10404}z^2-\frac{325}{31212}z^5-\frac{845}{31212}z^3+\frac{130}{7803}z^4+\frac{65}{10404}z^6-\frac{65}{15606}z^7+\frac{65}{31212}z^8-\frac{65}{31212}z^9$$

$$\text{“Delta[”}, 18, \text{“]=”}, \frac{65}{31212}$$

$$1-z-z^2$$

$$\text{“Delta[”}, 19, \text{“]=”}, 0$$

$$1 - z - z^2$$

"Delta[", 20, "]=", 0

$$1 - z - z^2$$

$$z^9 - z^8 - z^7$$

```
> for i from 0 to 20 do b[i] := binomial(i, floor(i/2)) od;
```

$$b_0 := 1$$

$$b_1 := 1$$

$$b_2 := 2$$

$$b_3 := 3$$

$$b_4 := 6$$

$$b_5 := 10$$

$$b_6 := 20$$

$$b_7 := 35$$

$$b_8 := 70$$

$$b_9 := 126$$

$$b_{10} := 252$$

$$b_{11} := 462$$

$$b_{12} := 924$$

$$b_{13} := 1716$$

$$b_{14} := 3432$$

$$b_{15} := 6435$$

$$b_{16} := 12870$$

$$b_{17} := 24310$$

$$b_{18} := 48620$$

$$b_{19} := 92378$$

$$b_{20} := 184756$$

```
> bermass(b, 20);
```

"Delta[", 1, "]=", 1

$$1$$

"Delta[", 2, "]=", 1

$$1 - z$$

"Delta[", 3, "]=", 1

$$1 - z - z^2$$

"Delta[", 4, "]=", 0

$$1 - z - z^2$$

"Delta[", 5, "]=", 1

$$1 - z - 2z^2 + z^3$$

"Delta[", 6, "]=", 0

$$1 - z - 2z^2 + z^3$$

"Delta[", 7, "]=", 1

$$1 - z - 3z^2 + 2z^3 + z^4$$

"Delta[", 8, "]=", 0

$$1 - z - 3z^2 + 2z^3 + z^4$$

"Delta[", 9, "]=", 1

$$1 - z - 4z^2 + 3z^3 + 3z^4 - z^5$$

"Delta[", 10, "]=", 0

$$1 - z - 4z^2 + 3z^3 + 3z^4 - z^5$$

"Delta[", 11, "]=", 1

$$1 - z - 5z^2 + 4z^3 + 6z^4 - 3z^5 - z^6$$

"Delta[", 12, "]=", 0

$$1 - z - 5z^2 + 4z^3 + 6z^4 - 3z^5 - z^6$$

"Delta[", 13, "]=", 1

$$1 - z - 6z^2 + 5z^3 + 10z^4 - 6z^5 - 4z^6 + z^7$$

"Delta[", 14, "]=", 0

$$1 - z - 6z^2 + 5z^3 + 10z^4 - 6z^5 - 4z^6 + z^7$$

"Delta[", 15, "]=", 1

$$1 - z - 7z^2 + 6z^3 + 15z^4 - 10z^5 - 10z^6 + 4z^7 + z^8$$

"Delta[", 16, "]=", 0

$$1 - z - 7z^2 + 6z^3 + 15z^4 - 10z^5 - 10z^6 + 4z^7 + z^8$$

"Delta[", 17, "]=", 1

$$1 - z - 8z^2 + 7z^3 + 21z^4 - 15z^5 - 20z^6 + 10z^7 + 5z^8 - z^9$$

"Delta[", 18, "]=", 0

$$1 - z - 8z^2 + 7z^3 + 21z^4 - 15z^5 - 20z^6 + 10z^7 + 5z^8 - z^9$$

"Delta[", 19, "]=", 1

$$1 - z - 9z^2 + 8z^3 + 28z^4 - 21z^5 - 35z^6 + 20z^7 + 15z^8 - 5z^9 - z^{10}$$

"Delta[", 20, "]=", 0

$$1 - z - 9z^2 + 8z^3 + 28z^4 - 21z^5 - 35z^6 + 20z^7 + 15z^8 - 5z^9 - z^{10}$$

$$z^{10} - z^9 - 9z^8 + 8z^7 + 28z^6 - 21z^5 - 35z^4 + 20z^3 + 15z^2 - 5z - 1$$

Computer generated binomial identities

$$\binom{2m}{m} = 1 + \sum_{i=0}^{m-1} (-1)^{\left\lfloor \frac{m-i-1}{2} \right\rfloor} \binom{\left\lfloor \frac{m+i}{2} \right\rfloor}{i} \binom{m+i}{\left\lfloor \frac{m+i}{2} \right\rfloor}$$

$$\binom{2m+1}{m} = \sum_{i=0}^{m} (-1)^{\left\lfloor \frac{m-i}{2} \right\rfloor} \binom{\left\lfloor \frac{m+i+1}{2} \right\rfloor}{i} \binom{m+i}{\left\lfloor \frac{m+i}{2} \right\rfloor}$$

$$\binom{2m}{m} = 2^m - \sum_{i=0}^{\left\lfloor \frac{m}{2} \right\rfloor - 1} (-1)^{\left\lfloor \frac{m-2i}{2} \right\rfloor} \binom{\left\lceil \frac{m+2i}{2} \right\rceil}{2i + \left\lceil \frac{m}{2} \right\rceil - \left\lfloor \frac{m}{2} \right\rfloor} \binom{2\left\lceil \frac{m}{2} \right\rceil + 2i}{\left\lceil \frac{m}{2} \right\rceil + i}$$

J. Riordan

Combinatorial Identities (1968, p. 37)

Special case for Wiedemann's algorithm

For

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

and

$$a_i = \underbrace{[1 \;\; 0 \;\; 0 \;\; \ldots \;\; 0]}_{u^{\mathrm{Tr}} = e_1^{\mathrm{Tr}}} \times C^i \times v, \quad v = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}, \quad a_i = \binom{i}{\lfloor i/2 \rfloor}$$

the algorithm needs no divisions/decisions.

Detail of "symbolic homotopy" algorithm

For $i = 0, 1, \ldots, 2n-1$ Do

$\{$Compute the coefficients of z^j, $0 \le j \le i$, of

$$\alpha_i(z) = e_1^{\mathrm{Tr}}(C + z(Y - C))^i v_0, \quad v_0 = \begin{bmatrix} a_0 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

where C is the previous companion matrix and $a_j = \binom{j}{\lfloor j/2 \rfloor}$.$\}$

Done by baby steps/giant steps: let $r = \lceil \sqrt{2n} \rceil$ and $s = \lceil 2n/r \rceil$.

Substep 1. For $j = 1, 2, \ldots, r-1$ Do

$$v_j(z) = (C + z(Y - C))^j v_0;$$

Substep 2. $Z(z) = (C + z(Y - C))^r$;

Substep 3. For $k = 1, 2, \ldots, s$ Do $u_k^{\mathrm{Tr}}(z) = e_1^{\mathrm{Tr}} Z(z)^k$;

Substep 4. For $j = 0, 1, \ldots, r-1$ Do

For $k = 0, 1, \ldots, s$ Do

$$\alpha_{kr+j}(z) = u_k^{\mathrm{Tr}}(z) v_j(z).$$

Analysis with fast matrix multiplication $O(n^{\omega})$ where $\omega \lessapprox 2.3755$

Split into $r \cdot s \geq 2n$, $r = \lceil (2n)^{1-\beta} \rceil$, $s = \lceil (2n)^{\beta} \rceil$:

Asymptotic time for	My choice $\beta = \frac{1}{2}$	T. Spencer's choice $\beta = \frac{\omega-2}{\omega-1} = 0.273$
Substep 1: $O(n^{\omega+1-\beta})$	$O(n^{2.875})$	$O(n^{3.103})$
Substep 2: same as Substep 1		
Substep 3: $s \cdot O(r^2 s^{\omega}) \cdot \widetilde{O}(r) = O(n^{3+(\omega-2)\beta})$	$O(n^{3.188})$	$O(n^{3.103})$
Substep 4: $\frac{r^2}{s} \cdot O(s^{\omega}) \cdot \widetilde{O}(n) = O(n^{3-(3-\omega)\beta})$	$O(n^{2.688})$	$O(n^{2.830})$

Using fast rectangular matrix multiplication, one can even get $O(n^{3.064})$ arithmetic arithmetic operations.

3. Coppersmith's blocking

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

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

$$\mathbf{a}_i = \mathbf{x}^{\mathrm{Tr}} B^i \mathbf{y} \in \mathbb{K}^{\beta\times\beta}, \quad 0 \le i < \frac{2n}{\beta} + 2.$$

Find a matrix polynomial $\mathbf{c}_0 + \mathbf{c}_1\lambda + \cdots + \mathbf{c}_d\lambda^d \in \mathbb{K}^{\beta\times\beta}[\lambda]$, $d = \lceil n/\beta \rceil$, such that

$$\forall\, j \ge 0\colon \ \sum_{i=0}^{d} \mathbf{a}_{j+i}\mathbf{c}_i = \sum_{i=0}^{d} \mathbf{x}^{\mathrm{Tr}} B^{i+j} \mathbf{y}\mathbf{c}_i = \mathbf{0} \in \mathbb{K}^{\beta\times\beta}$$

Probabilistic analysis

Theorem [K&V 2000]: *If B is nonsingular with distinct eigenvalues then we have for the* ***minimal*** *generating polynomial*

$$\det(\mathbf{c}_0 + \mathbf{c}_1\lambda + \cdots + \mathbf{c}_d\lambda^d) = \det(\lambda I - B)$$

for ***random*** $\mathbf{x}$, $\mathbf{z}$ *with probability*

$$\geq 1 - \frac{2n-1}{|\mathbb{K}|}.$$

Distinct eigenvalues can be obtained by preconditioning B à la [Wiedemann, 1986], for instance

$\widetilde{B} \leftarrow V \cdot B \cdot W \cdot G$ where V is randomized butterfly network
W is randomized butterfly network
G is random diagonal

Proof idea for probabilistic analysis

$$(I - \lambda B)^{-1} = I + B\lambda + B^2\lambda^2 + \cdots$$

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

$$\mathbf{x}^{\mathrm{Tr}}(I - \lambda B)^{-1}\mathbf{y} = R(\lambda)(\mathbf{c}_d + \cdots + \mathbf{c}_0\lambda^d)^{-1}$$

Use theorems from multivariable control theory (irreducible realizations) to show that polynomial denominators are the same.

Show run-time estimates in Maple worksheet

```
> beta := n^sigma; # blocking factor
```

$$\beta := n^{\sigma}$$

```
> s := n^tau; # number of giant steps
```

$$s := n^{\tau}$$

```
> r := simplify( (n/beta) / s); # number of baby steps
```

$$r := n^{(1-\sigma-\tau)}$$

Standard matrix arithmetic, quadratic B/M, Chinese remainder integer arithmetic

Step 1.1: Compute $B^j y$, $j = 0,\ldots,r$

```
> substep1 := simplify( r * beta * n^2 * r );
```

$$substep1 := n^{(4-\sigma-2\tau)}$$

Step 1.2: Compute $Z = B^r$ by repeated squaring

```
> substep2 := simplify( n^3 * r);
```

$$substep2 := n^{(4-\sigma-\tau)}$$

Step 1.3: Compute $x^{Tr} Z^k$, $k = 0,\ldots,s$

```
> substep3 := simplify( s * beta * n^2 * n/beta);
```

$$substep3 := n^{(\tau+3)}$$

Step 1.4: Compute $(x^{Tr} Z^k)(B^j y)$

```
> substep4 := simplify( r * s * beta^2 * n * n/beta );
```

$$substep4 := n^3$$

Step 2: Blocked Berlekamp/Massey for n moduli

```
> step2 := simplify( (n/beta)^2 * beta^3 * n );
```

$$step2 := n^{(3+\sigma)}$$

Step 3: Determinant of generator matrix polynomial for n moduli

```
> step3 := simplify( beta^3 * n * n );
```

$$step3 := n^{(3\sigma+2)}$$

Overall bit complexity

```
> eval([substep1, substep2, substep3, substep4, step2, step3],
>{sigma=1/3, tau=1/3});
```

$$[n^3, n^{(10/3)}, n^{(10/3)}, n^3, n^{(10/3)}, n^3]$$

"The asymptotically best algorithms frequently turn out

to be worst on all problems for which they are used."

— D. G. CANTOR and H. ZASSENHAUS (1981)

Fast matrix multiplication O(n^ω), linear B/M, linear integer arithmetic

Step 1.1: Compute $B^j y$, $j = 0,...,r$ by $B^{(2^i)}$ [$B^0 y$ — ... — $B^{(2^i-1)}$ y]

```
> fastsubstep1 := simplify( n^omega * r );
```

$$fastsubstep1 := n^{(\omega+1-\sigma-\tau)}$$

Step 1.2: Compute $Z = B^r$ by repeated squaring

```
> fastsubstep2 := simplify( n^omega * r);
```

$$fastsubstep2 := n^{(\omega+1-\sigma-\tau)}$$

Step 1.3: Compute $x^{Tr} Z^k$, $k = 0,...,s$ as *fat* vectors times a *thin* matrix

```
> fastsubstep3 := simplify( s * (n/(beta*s))^2 * (beta*s)^omega * r,
> 'power', 'symbolic' );
```

$$fastsubstep3 := n^{(-2\tau+3-3\sigma+(\sigma+\tau)\omega)}$$

Step 1.4: Compute $(x^{Tr} Z^k) (B^j y)$

```
> fastsubstep4 := simplify( r^2/s * (beta*s)^omega * n/beta, 'power',
> 'symbolic' );
```

$$fastsubstep4 := n^{(3-3\sigma-3\tau+(\sigma+\tau)\,\omega)}$$

Step 2: Blocked Berlekamp/Massey for n moduli

```
> faststep2 := simplify( beta^2 * n * n);
```

$$faststep2 := n^{(2\sigma+2)}$$

Step 3: Determinant of generator matrix polynomial for n moduli

```
> faststep3 := simplify( beta^omega * n, 'power', 'symbolic' );
```

$$faststep3 := n^{(\sigma\omega+1)}$$

Overall bit complexity

```
> total := eval([fastsubstep1, fastsubstep2, fastsubstep3,
> fastsubstep4, faststep2, faststep3]);
```

$$total := [\, n^{(\omega+1-\sigma-\tau)}, n^{(\omega+1-\sigma-\tau)}, n^{(-2\tau+3-3\sigma+(\sigma+\tau)\,\omega)}, n^{(3-3\sigma-3\tau+(\sigma+\tau)\,\omega)}, n^{(2\sigma+2)}, n^{(\sigma\omega+1)} \,]$$

```
> expos:=simplify(map(x -> log[n](x), total), 'symbolic');
```

$$expos := [\omega+1-\sigma-\tau, \omega+1-\sigma-\tau, -2\tau+3-3\sigma+\sigma\omega+\omega\tau, 3-3\sigma-3\tau+\sigma\omega+\omega\tau, 2\sigma+2, \sigma\omega+1]$$

```
> numexpos:=eval(expos, omega=2.3755);
```

$$numexpos := [3.3755-\sigma-\tau, 3.3755-\sigma-\tau, .3755\tau+3.-.6245\sigma, 3.-.6245\sigma-.6245\tau, 2\sigma+2, 2.3755\sigma+1]$$

```
> minexpo := solve({numexpos[2]=numexpos[3],
> numexpos[2]=numexpos[5]}, {sigma,tau});
```

$$minexpo := \{\sigma = .4042922554, \tau = .1626232338\}$$

```
> eval(numexpos, minexpo);
```

$$[2.808584511, 2.808584511, 2.808584510, 2.645961276, 2.808584511, 1.960396253]$$