The nature ("art") of symbolic computation

Erich Kaltofen
North Carolina State University
www.kaltofen.net

Where it began

1960s-early 70s: MIT project MAC [Moses]

$$\int 1 + (x+1)^n \mathrm{d}x = x + (x+1)^{n+1}/(n+1)$$

S. C. Johnson, "Tricks for Improving Kronecker's Method," Bell Laboratories Report 1966.

Berlekamp/Zassenhaus's, Risch's algorithms

$$\int \frac{x+1}{x^4} e^{1/x} \mathrm{d}x = -\frac{x^2-x+1}{x^2} e^{1/x}$$

B. G. Claybrook, "A new approach to the symbolic factorization of multivariate polynomials," *Artificial Intelligence*, vol. 7, (1976), pp. 203–241.

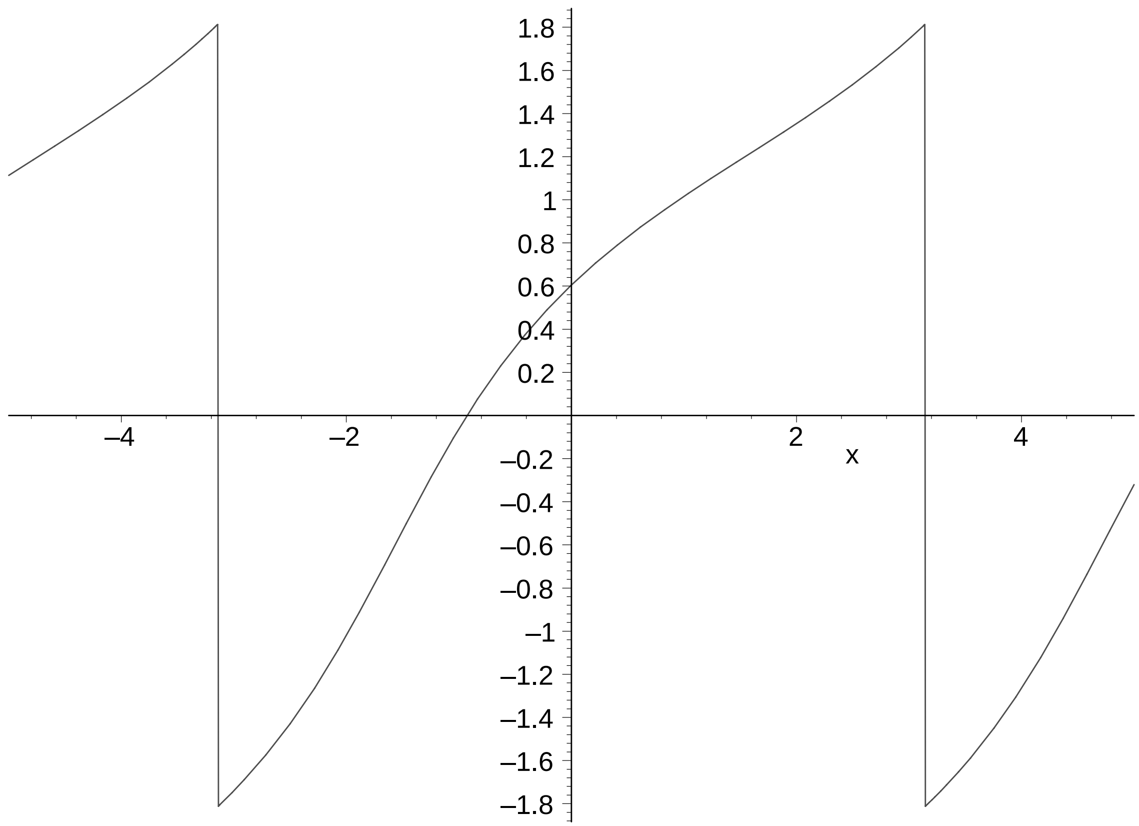
```
> # Example by Corless and Jeffrey
> f := 1/(sin(x) + 2);
```

$$f := \frac{1}{\sin(x) + 2}$$

```
> g := int(f, x);
```

$$g := \frac{2}{3}\sqrt{3}\arctan(\frac{1}{3}(2\tan(\frac{1}{2}x) + 1)\sqrt{3})$$

```
> plot(g, x=-5..5);
```



What is an algorithm?

– **finite** unambiguous list of steps ("control, program")

– computes a function from $D \longrightarrow E$ where D is **infinite** ("infinite Turing tape")

Ambiguity through randomization

– Monte Carlo: "always fast, probably correct". Examples: `isprime`

Lemma *[DeMillo&Lipton'78, Schwartz/Zippel'79]*
Let $f, g \in \mathbb{F}[x_1, \ldots, x_n], f \neq g, S \subseteq \mathbb{F}$.

$$\text{Probability}(f(a_1, \ldots, a_n) \neq g(a_1, \ldots, a_n) \mid a_i \in S) \\ \geq 1 - \max\{\deg(f), \deg(g)\}/\text{cardinality}(S)$$

sparse polynomial interpolation, factorization, minimal polynomial of a sparse matrix

Do we exactly know what the algorithm computes? E.g., in the presence of floating point arithmetic?

– Las Vegas: "always correct, probably fast".
Examples: polynomial factorization in $\mathbb{Z}_p[x]$, where $p \gg 2$.
Determinant of a sparse matrix

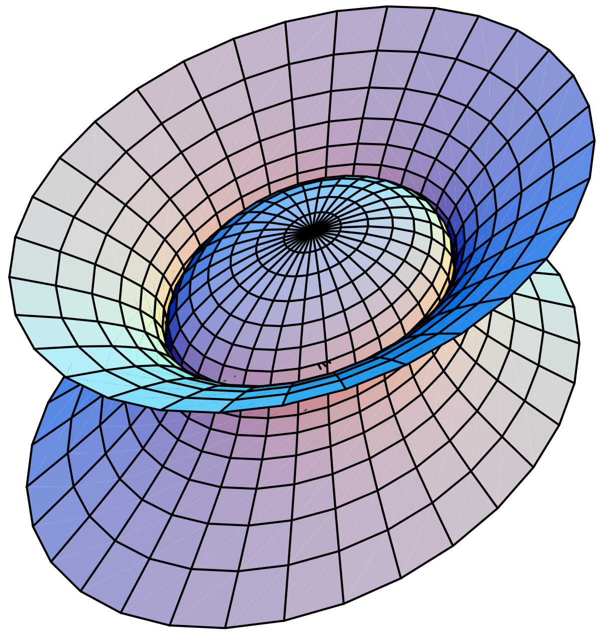
De-randomization: conjectured slow-down is within polynomial complexity.

Shuhong Gao, E. Kaltofen, and Lauder, A., "Deterministic distinct degree factorization for polynomials over finite fields," 2001.

Efficiency dilemma: the higher the confidence in the result, the more time does it takes to compute it.

Factorization of nearby polynomials over the complex numbers

$$81x^4 + 16y^4 - 648z^4 + 72x^2y^2 - 648x^2 - 288y^2 + 1296 = 0$$



$$(9x^2 + 4y^2 + 18\sqrt{2}\,z^2 - 36)(9x^2 + 4y^2 - 18\sqrt{2}\,z^2 - 36) = 0$$

$$81x^4 + 16y^4 - 648.003z^4 + 72x^2y^2 + .002x^2z^2 + .001y^2z^2 - 648x^2 - 288y^2 - .007z^2 + 1296 = 0$$

Open Problem *[Kaltofen LATIN'92]*
Given is a polynomial $f(x, y) \in \mathbb{Q}[x, y]$ *and* $\epsilon \in \mathbb{Q}$.

Decide in polynomial time in the degree and coefficient size if there is a factorizable $\tilde{f}(x, y) \in \mathbb{C}[x, y]$ *with*

$$\|f - \tilde{f}\| \leq \epsilon \text{ and } \deg(\tilde{f}) \leq \deg(f),$$

for a reasonable coefficient vector norm $\| \cdot \|$.

Theorem *[Hitz, Kaltofen, Lakshman ISSAC'99]*
We can compute in polynomial time in the degree and coefficient size if there is an $\tilde{f}(x, y) \in \mathbb{C}[x, y]$ *with a factor of a* ***constant degree*** *and* $\|f - \tilde{f}\|_2 \leq \epsilon$.

Numerical algorithms

Conclusion on my exact algorithm [JSC 1985]:

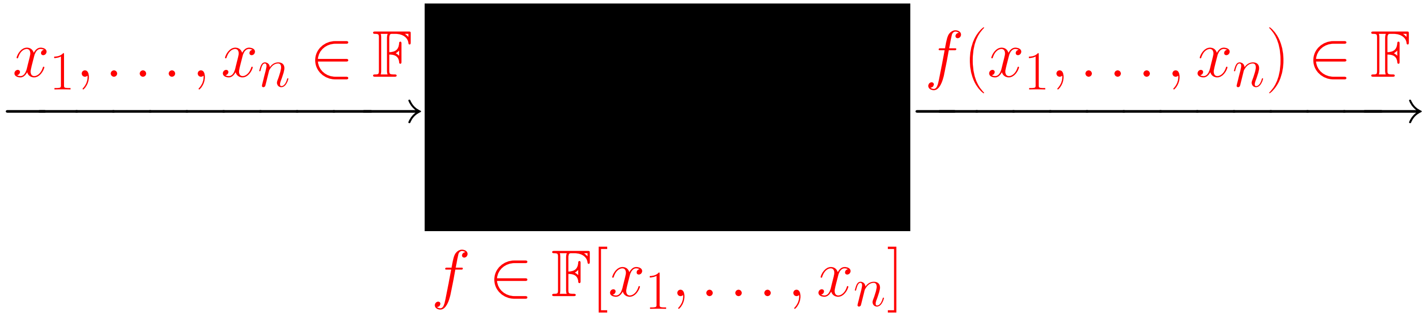
"D. Izraelevitz at Massachusetts Institute of Technology has already implemented a version of algorithm 1 using complex floating point arithmetic. Early experiments indicate that the linear systems computed in step (L) tend to be ***numerically ill-conditioned.*** *How to overcome this numerical problem is an important question which we will investigate."*

Sasaki *et al.* [Japan J. Indust. Applied Math, 1991, ISSAC'01]: Combine sums of powers of roots to low degree polys

Stetter, Huang, Wu and Zhi [ISSAC'2K]: Hensel lift factor combinations numerically and eliminate extraneous factors early

Corless, Giesbrecht, Kotsireas, van Hoeij, Watt [ISSAC'01]: sample curve by points and interpolate

Black box polynomials

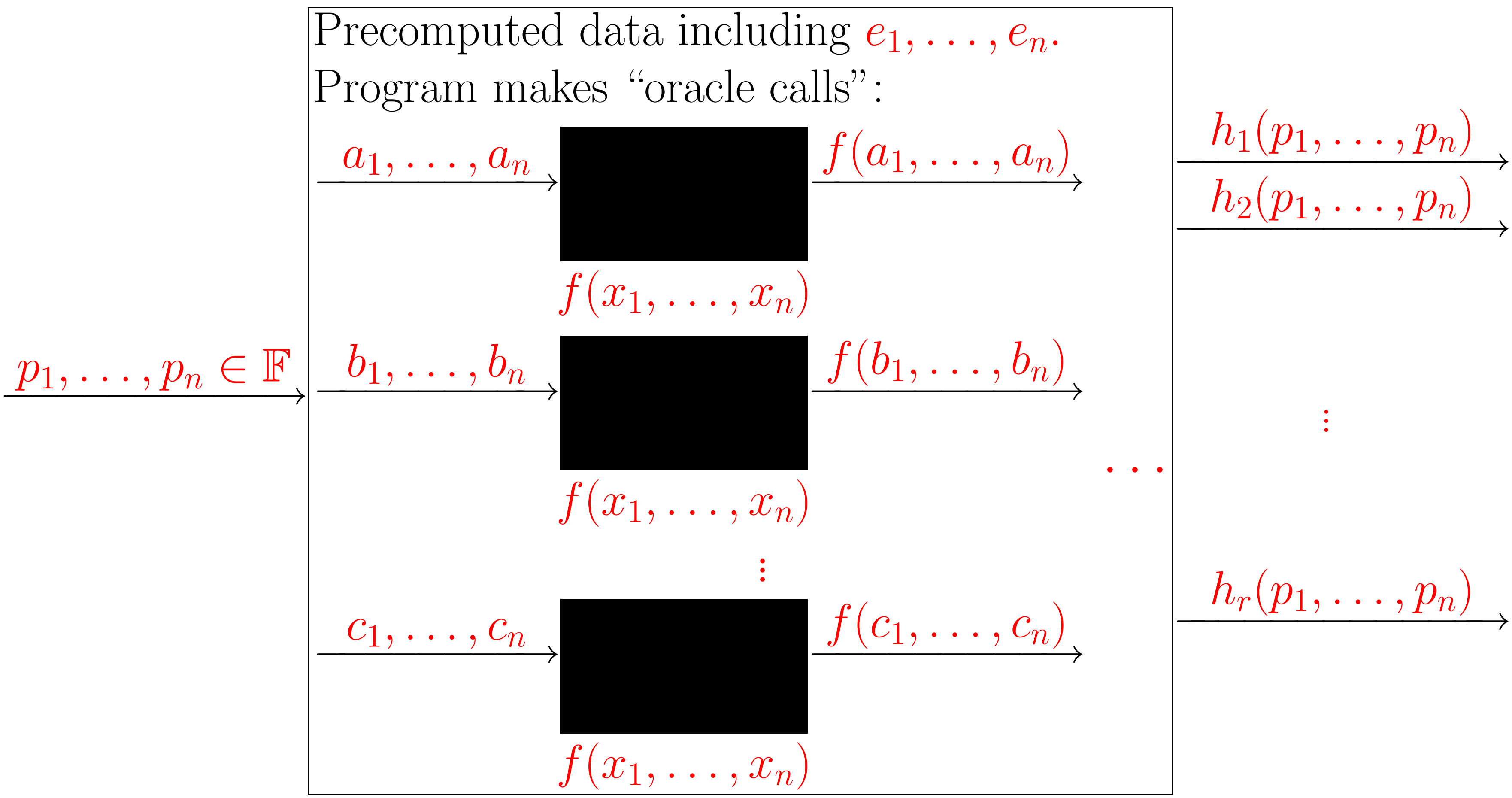


$\mathbb{F}$ an arbitrary field, e.g., rationals, reals, complexes

Perform polynomial algebra operations, e.g., factorization with

$n^{O(1)}$ black box calls,
$n^{O(1)}$ arithmetic operations in $\mathbb{F}$ and
$n^{O(1)}$ randomly selected elements in $\mathbb{F}$

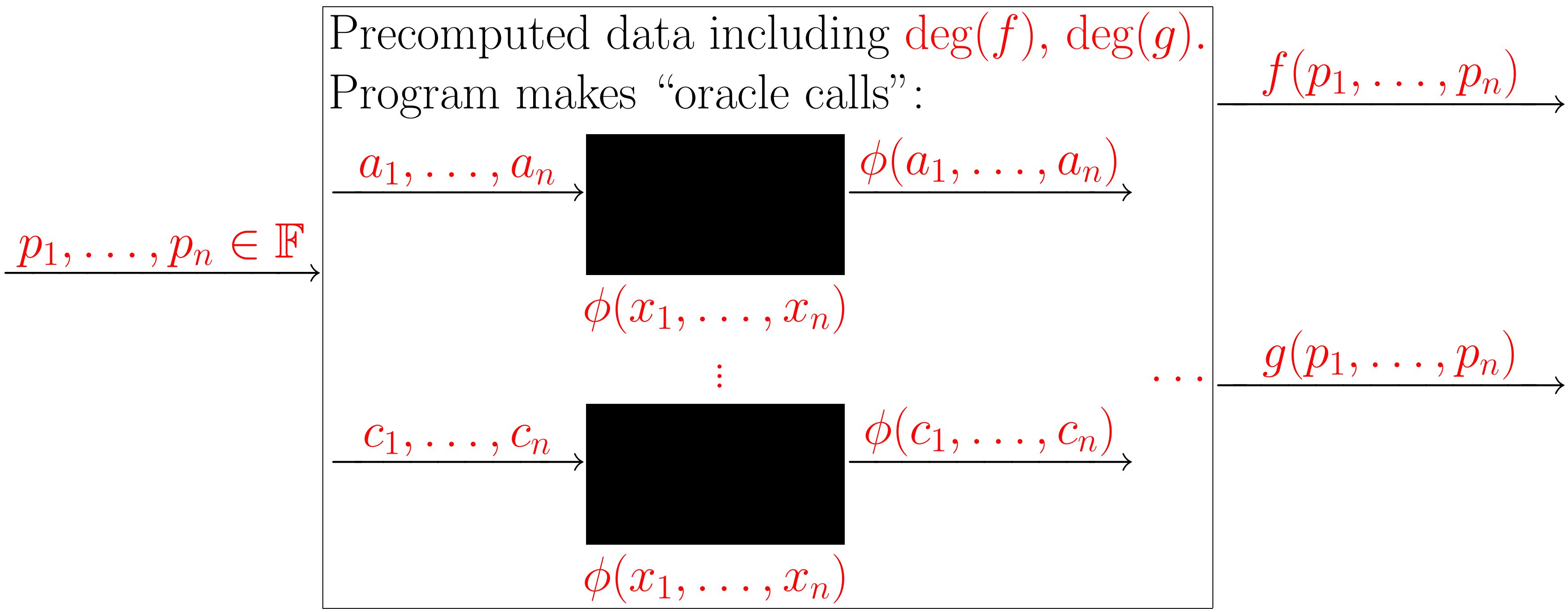
Kaltofen and Trager (1988) efficiently construct the following efficient program:



$$f(x_1, \ldots, x_n) = h_1(x_1, \ldots, x_n)^{e_1} \cdots h_r(x_1, \ldots, x_n)^{e_r}$$

$h_i \in \mathbb{F}[x_1, \ldots, x_n]$ irreducible.

Numerator and denominator separation [Kaltofen and Trager 1988]



$$\phi(x_1, \ldots, x_n) = \frac{f(x_1, \ldots, x_n)}{g(x_1, \ldots, x_n)}, f, g \in \mathbb{F}[x_1, \ldots, x_n], \gcd(f, g) = 1.$$

Given a black box

$f(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$

$\mathbb{F}$ a field

compute by multiple evaluation of this black box the sparse representation of f

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{t} a_i x_1^{e_{i,1}} \cdots x_n^{e_{i,n}}, \quad a_i \neq 0$$

Several solutions that are polynomial-time in n and t:

Zippel (1979, 1988), Ben-Or, Tiwari (1988)
Kaltofen, Lakshman (1988)
Grigoriev, Karpinski, Singer (1988)
Mansour (1992)
Kaltofen, Lee, Lobo (2000)

Sparsity with non-standard basis

In place of x^e use

$(x-a)^e$	shifted basis
$x(x+1)\cdots(x+e-1)$	Pochhammer basis
$T_e(x)$	Chebyshev basis

Solutions (not all polynomial-time):

Lakshman, Saunders (1992, 1994): Chebyshev, Pochh., shifted
Grigoriev, Karpinski (1993): shifted
Grigoriev, Lakshman (1995): shifted
Lee (2001): Chebyshev, Pochhammer, shifted
Giesbrecht, Kaltofen and Lee (2002): shifted

Example: determinant of Cauchy matrix

$$\det\left(\begin{bmatrix} \frac{1}{x_1+y_1} & \frac{1}{x_1+y_2} & \cdots & \frac{1}{x_1+y_n} \\ \frac{1}{x_2+y_1} & \frac{1}{x_2+y_2} & \cdots & \frac{1}{x_2+y_n} \\ \vdots & \vdots & & \vdots \\ \frac{1}{x_n+y_1} & \frac{1}{x_n+y_2} & \cdots & \frac{1}{x_n+y_n} \end{bmatrix}\right) = \frac{\prod\limits_{1\le i<j\le n}(x_j - x_i)(y_j - y_i)}{\prod\limits_{1\le i,j\le n}(x_i + y_j)}.$$

Compute sparse factors of numerators and denominators

FoxBox [Díaz and K 1998] example: determinant of symmetric Toeplitz matrix

$$\det\left(\begin{bmatrix} a_0 & a_1 & \dots & a_{n-2} & a_{n-1} \\ a_1 & a_0 & \dots & a_{n-3} & a_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-2} & a_{n-3} & \dots & a_0 & a_1 \\ a_{n-1} & a_{n-2} & \dots & a_1 & a_0 \end{bmatrix}\right)$$

$$= F_1(a_0, \dots, a_{n-1}) \cdot F_2(a_0, \dots, a_{n-1}).$$

over the integers.

```
> readlib(showtime):
> showtime():
O1 := T := linalg[toeplitz]([a,b,c,d,e,f]):
time   0.03   words   7701
O2 := factor(linalg[det](T));
```

$$-(2\,d\,c\,a - 2\,b\,c\,e + 2\,c^2\,a - a^3 - d\,a^2 + 2\,d^2\,c + d^2\,a + b^3 + 2\,a\,b\,c - 2\,c^2\,b + d^3 + 2\,a\,b^2 - 2\,d\,c\,b - 2\,c\,b^2 - 2\,e\,c^2 + 2\,e\,b^2 + 2\,f\,c\,b + 2\,b\,a\,e + b^2\,f + c^2\,f + b\,e^2 - b\,a^2 - f\,d\,b - f\,d\,a - f\,a^2 - f\,b\,a + e^2\,a - 2\,d\,b^2 + d\,c^2 - 2\,d\,e\,b - 2\,d\,e\,c - d\,b\,a)(2\,d\,c\,a - 2\,b\,c\,e - 2\,c^2\,a + a^3 - d\,a^2 - 2\,d^2\,c - d^2\,a + b^3 + 2\,a\,b\,c - 2\,c^2\,b + d^3 - 2\,a\,b^2 + 2\,d\,c\,b + 2\,c\,b^2 + 2\,e\,c^2 - 2\,e\,b^2 - 2\,f\,c\,b + 2\,b\,a\,e + b^2\,f + c^2\,f + b\,e^2 - b\,a^2 - f\,d\,b + f\,d\,a - f\,a^2 + f\,b\,a - e^2\,a - 2\,d\,b^2 + d\,c^2 + 2\,d\,e\,b - 2\,d\,e\,c + d\,b\,a)$$

```
time   27.30   words   857700
```

FoxBox timings for symmtric Toeplitz deteriminant challenge

N	CPU Time	Degree	# Terms
10	$1^h 20'$	5	931
11	$1^h 34'$	5	847
12	$10^h 14'$	6	5577
13	$15^h 24'$	6	4982

CPU times (hours^{h}minutes$'$) to retrieve the distributed representation of a factor from the factors black box of a symmetric Toeplitz determinant black box. Construction is over $\mathbb{Q}$, evaluation is over $\mathbb{Z}_{10^8+7}$ for $N = 10, 11,$ and 12 (Pentium 133, Linux 2.0) and $\mathbb{Z}_{2^{30}-35}$ for $N = 13$ (Sun Ultra 2 168MHz, Solaris 2.4).

Serialization of **factors box** of 8 by 8 symmetric Toeplitz matrix modulo 65521

15,8,-1,1,2,2,-1,8,1,7,1,1,20752,-1,1,39448,33225,984,17332,53283,35730,
23945,13948,22252,52005,13703,8621,27776,33318,2740,4472,36959,
17038,55127,16460,26669,39430,1,0,1,4,20769,16570,58474,30131,770,
4,25421,22569,51508,59396,10568,4,20769,16570,58474,30131,770,8,
531,55309,40895,38056,34677,30870,397,59131,12756,3,13601,54878,
13783,39334,3,41605,59081,10842,15125,3,45764,5312,9992,25318,3,
59301,18015,3739,13650,3,23540,44673,45053,33398,3,4675,39636,45179,
40604,3,49815,29818,2643,16065,3,46787,46548,12505,53510,3,10439,
37666,18998,32189,3,38967,14338,31161,12779,3,27030,21461,12907,
22939,3,24657,32725,47756,22305,3,44226,9911,59256,54610,3,56240,
51924,26856,52915,3,16133,61189,17015,39397,3,24483,12048,40057,
21323

Serialization of **checkpoint** during sparse interpolation

28, 14, 9, 64017, 31343, 5117, 64185, 47755, 27377, 25604, 6323, 41969, 14, 3, 4, 0, 0, 3, 4, 0, 1, 3, 4, 0, 2, 3, 4, 0, 3, 3, 4, 0, 4, 3, 4, 1, 0, 3, 4, 1, 1, 3, 4, 1, 2, 3, 4, 1, 3, 3, 4, 2, 0, 3, 4, 2, 1, 3, 4, 2, 2, 3, 4, 3, 0, 3, 4, 3, 1, 14, 59877, 1764, 59012, 44468, 1, 19485, 25871, 3356, 2, 58834, 49014, 65518, 15714, 65520, 1, 2, 4, 4, 1, 1

Early termination strategies

Early termination in Newton interpolation [Kaltofen 1986]

For $i \leftarrow 1, 2, \ldots$ *Do*

Pick distinct p_i *and from* $f(p_i)$

compute

$$\begin{aligned} f^{[i]}(x) \leftarrow c_0 + c_1(x-p_1) + c_2(x-p_1)(x-p_2) + \cdots \\ \equiv f(x) \pmod{(x-p_1)\cdots(x-p_i)} \end{aligned}$$

If $f^{[i]}(a) = f(a)$ *for a* ***random*** a *stop.*

End For

Threshold η: In order to obtain a better probability, we require $f^{[i]}(a_j) = f(a_j)$ for several random a_j.

Early termination in the Chinese remainder algorithm

Theorem *[Kaltofen 2002]*
Input: $A \in \mathbb{Z}^{n\times n}$, $b = \log \| A \|$, *threshold* η. *Output:* $\det A$
Method: baby steps/giant steps [KV 2001] with early termination (Monte Carlo)
Bit complexity: $(\sqrt{b(b+\eta+\log|\det A|)} \cdot n^3)^{1+o(1)}$
Example $\det(A) = O(n^{1-\alpha} b), \eta = O(1):\ \ (bn^{3+1/2-\alpha/2})^{1+o(1)}$

– [Pan 1988, Abbott, Bronstein, and Mulders 1999] use denominator of linear system solution

– [Emiris 1998] $(bn^{4-\alpha})^{1+o(1)}$

– [Eberly, Giesbrecht, and Villard 2001] compute invariant factors

– [Kaltofen and Villard 2000] $n^{2.697263} b^{1+o(1)}$

– [Storjohann 2002] $n^{2.375477} b^{1+o(1)}$ for polynomial entries

The early termination of Ben-Or/Tiwari's interpolation algorithm [Kaltofen, Lobo and Lee 2000].

If $p_1, \ldots, p_n$ are chosen randomly and uniformly from a subset S of the domain of values then for the linearly recurrent sequence

$$a_i = f(p_1^i, \ldots, p_n^i), i = 1, 2, \ldots$$

the Berlekamp/Massey algorithm encounters $\Delta = 0$ (when $2L < r$) the first time for $r = 2t + 1$ with probability no less than

$$1 - \frac{t(t+1)(2t+1)\deg(f)}{6 \cdot \mathrm{cardinality}(S)},$$

where t is the number of terms of f.

Threshold ζ: In order to obtain a better probability, we require $\Delta = 0$ (when $2L < r$) more than once before terminating.

Success and failure: different moduli and thresholds [Lee 2001]

$$f_1 = x_1^2x_3^3x_4x_6x_8x_9^2 + x_1x_2x_3x_4^2x_5^2x_8x_9 + x_2x_3x_4x_5^2x_8x_9 + x_1x_3^3x_4^2x_5^2x_6^2x_7x_8^2 + x_2x_3x_4x_5^2x_6x_7x_8^2$$
$$f_2 = x_1x_2^2x_4^2x_8x_9^2x_{10}^2 + x_2^2x_4x_5^2x_6x_7x_9x_{10}^2 + x_1^2x_2x_3x_5^2x_7^2x_9^2 + x_1x_3^2x_4^2x_7^2x_9^2 + x_1^2x_3x_4x_7^2x_8^2$$
$$f_3 = 9x_2^3x_3^3x_5^2x_6^2x_8^3x_9^3 + 9x_1^3x_2^2x_3^3x_5^2x_7^2x_8^2x_9^3 + x_1^4x_3^4x_4^2x_5^4x_6^4x_7x_8^5x_9 + 10x_1^4x_2x_3^4x_4^4x_5^4x_7x_8^3x_9 + 12x_2^3x_4^3x_6^3x_7^2x_8^3$$
$$f_4 = 9x_1^2x_3x_4x_6^3x_7^2x_8x_{10}^4 + 17x_1^3x_2x_5^2x_6^2x_7x_8^3x_9^4x_{10}^3 + 17x_2^2x_3^4x_4^2x_7^4x_8^3x_9x_{10}^3 + 3x_1^3x_2^2x_6^3x_{10}^2 + 10x_1x_3x_5^2x_6^2x_7^4x_8^4$$

	Thresholds			mod 31			mod 37			mod 41			mod 43			mod 47			mod 53		
	η, ζ	τ	κ, γ	=	≠	!	=	≠	!	=	≠	!	=	≠	!	=	≠	!	=	≠	!
f_1	1	0	0	8	2	90	7	1	92	15	3	82	11	5	84	25	3	72	20	2	78
	2	1	2	30	0	70	38	1	61	44	0	56	55	0	45	71	0	29	52	0	48
	3	2	4	38	0	62	36	0	64	50	0	50	60	0	40	79	0	21	70	0	30
f_2	1	0	0	4	3	93	4	3	93	5	3	92	7	5	88	22	4	74	23	1	76
	2	1	2	22	0	78	36	0	64	38	0	62	48	1	51	61	0	39	66	0	34
	3	2	4	41	0	59	45	0	55	51	0	49	57	0	43	83	0	17	81	0	19
f_3	1	0	0	0	2	98	0	6	94	3	3	94	4	0	96	6	5	89	9	1	90
	2	1	2	3	1	96	8	0	92	16	0	84	10	0	90	37	0	63	27	0	73
	3	2	4	9	0	91	8	0	92	**26**	**0**	**74**	15	0	85	52	0	48	54	0	46
f_4	1	0	0	1	4	95	0	2	98	4	2	94	8	3	89	18	2	80	5	3	92
	2	1	2	8	0	92	5	0	95	20	0	80	22	0	78	63	0	37	44	0	56
	3	2	4	10	0	90	10	0	90	33	0	67	32	0	68	80	0	20	47	0	53

Numerical	*Randomized (Monte Carlo)*
more efficiency, but approximate result	more efficiency, but uncertain result
ill-conditionedness near singular inputs	unfavorable inputs: pseudo-primes, $\sum_i \prod_j (x_i - j)$, Coppersmith's "pathological" matrices
convergence analysis	probabilistic analysis
try algorithms on unproven inputs	try algorithms with limited randomness

Numerical + randomized, e.g., LinBox's matrix preconditioners:
all of the above(?)

Hallmarks of a good heuristic

– Is algorithmic in nature, i.e., always terminates with a result of possibly unknown validity

– Is a proven complete solution in a more stringent setting, for example, by restricting the inputs or by slowing the algorithm

– Has an experimental track record, for example, works on 50% of cases

Example: van Hoeij's lattice-based factorization algorithm