The Art of Symbolic Computation

Erich Kaltofen

`google->kaltofen`

Caviness's foreword to the Computer Algebra Handbook

Two ideas lie gleaming on the jeweler's velvet. The first is the calculus, the second, the algorithm. The calculus and the rich body of mathematical analysis to which it gave rise made modern science possible; but it has been the algorithm that has made possible the modern world.

—David Berlinski, *The Advent of the Algorithm*

Caviness's foreword to the Computer Algebra Handbook

Two ideas lie gleaming on the jeweler's velvet. The first is the calculus, the second, the algorithm. The calculus and the rich body of mathematical analysis to which it gave rise made modern science possible; but it has been the algorithm that has made possible the modern world.

—David Berlinski, *The Advent of the Algorithm*

So, gentle reader, I recommend this volume and all its concepts, symbols, and algorithms to you.

—Bob Caviness, *Computer Algebra Handbook*

Where it began

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

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

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.

Important algorithms: “classical” computer algebra

Euclid, Chinese remainder

Sturm chains, Seidenberg’s algorithm

Gauss’s distinct degree factorization, Berlekamp/Zassenhaus

Berlekamp/Massey

Gröbner, Macaulay resultants, Wu triangular sets

Risch integration and transcendence theory of special functions

FFT-based polynomial arithmetic

Gosper and Karr

Collins cylindrical algebraic decomposition

. . .

KUNGL. VETENSKAPSAKADEMIEN
THE ROYAL SWEDISH ACADEMY OF SCIENCES

Information Department, PO Box 50005, SE-104 05 Stockholm, Sweden, webbsite: www.kva.se
Tel: +46-8-673 95 95, Fax +46-8-15 56 70, e-mail: info@kva.se

THE NOBEL PRIZE IN PHYSICS 1999

PRESS RELEASE 12 OCTOBER 1999
The Prize I Further reading I The laureates

The Royal Swedish Academy of Sciences has awarded
the 1999 Nobel Prize in Physics
jointly to

Professor **Gerardus ’t Hooft**, University of Utrecht, Utrecht, the Netherlands, and
Professor Emeritus **Martinus J.G. Veltman**, University of Michigan, USA, resident in Bilthoven, the Netherlands.

The two researchers are being awarded the Nobel Prize for having placed particle physics theory on a firmer mathematical foundation. **...**

The Academy’s citation:
"for elucidating the quantum structure of electroweak interactions in physics."

...
One person who had not given up hope of being able to renormalize non-abelian gauge theories was **Martinus J.G.Veltman**. At the end of the 1960s he was a newly appointed professor at the University of Utrecht. Veltman had developed the *Schoonschip* computer program which, using symbols, performed algebraic simplifications of the complicated expressions that all quantum field theories result in when quantitative calculations are performed. Twenty years earlier, Feynman had indeed systematised the problem of calculation and introduced *Feynman diagrams* that were rapidly accepted by researchers. But at that time there were no computers. Veltman believed firmly in the possibility of finding a way of renormalizing the theory and his computer program was the cornerstone of the comprehensive work of testing different ideas.

Important algorithms: “middle earth”

Zippel and Ben-Or-Tiwari sparse interpolation

Singer and Kovacic differential equation solvers

Lattice basis reduction [LLL]

Zeilenberger

Wiedemann, block Wiedemann/Lanczos, matrix Padé

Straight-line and black box polynomial factorization

Baby steps/giant steps algorithms for linear and polynomial algebra

Tellegen’s principle

Real roots of polynomial systems

Noda-Sasaki approximate GCD, Sasaki approx. factorization

Corless et al. SVD methods

. . .

Important algorithms: “modern” symbolic computation

Sparse resultants, A- and J-resultants

Giesbrecht/Mulders-Storjohann diophantine linear solvers

Fast bit complexity in linear algebra over the integers

Black box matrix preconditioners, early termination

Sasaki/van Hoeij power sums, Bostan et al. logarithmic derivatives

Sparsest shift of polynomials

Villard-Jeannerod optimal polynomial matrix inverse

Skew, Ore and differential polynomial factorization

Approximate polynomial factorization via PDEs

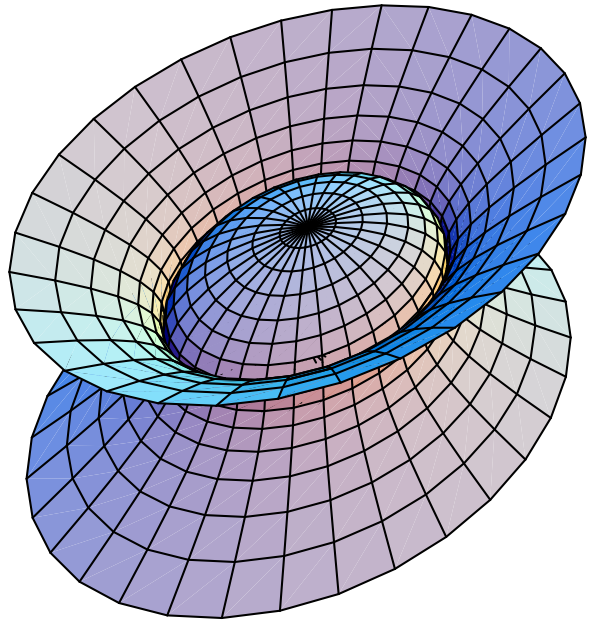
Barvinok-Woods and De Loera et al. short rational functions

Lenstra/Kaltofen-Koiran lacunary polynomial factorization

. . .

Factorization of “noisy” polynomials over the complex numbers [my 1998 Challenge Problem 1]

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



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

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

Conclusion on my exact algorithm [JSC 1(1)'85]

"*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.*"

The Approximate Factorization Problem [Kaltofen '89; Sasaki '89]

Given $f \in \mathbb{C}[x_1, \ldots, x_r]$ irreducible, find $\tilde{f} \in \mathbb{C}[x_1, \ldots, x_r]$ s.t. $\deg \tilde{f} \leq \deg f$, $\tilde{f}$ factors, and $\| f - \tilde{f} \|$ is minimal.

The Approximate Factorization Problem [Kaltofen '89; Sasaki '89]

Given $f \in \mathbb{C}[x_1, \ldots, x_r]$ irreducible, find $\tilde{f} \in \mathbb{C}[x_1, \ldots, x_r]$ s.t. $\deg \tilde{f} \leq \deg f$, $\tilde{f}$ factors, and $\| f - \tilde{f} \|$ is minimal.

Problem depends on choice of norm $\| \cdot \|$, and notion of degree.

We use 2-norm, and multi-degree:
$\mathrm{mdeg}\ f = (\deg_{x_1} f, \ldots, \deg_{x_r} f)$

The Approximate Factorization Problem [Kaltofen '89; Sasaki '89]

Given $f \in \mathbb{C}[x_1, \ldots, x_r]$ irreducible, find $\tilde{f} \in \mathbb{C}[x_1, \ldots, x_r]$ s.t. $\deg \tilde{f} \leq \deg f$, $\tilde{f}$ factors, and $\| f - \tilde{f} \|$ is minimal.

Degree bound is important:
$(1 + \delta x) f$ is reducible but for $\delta < \varepsilon / \| f \|$,

$$\| (1 + \delta x) f - f \| = \| \delta x f \| = \delta \| f \| < \varepsilon$$

Previous Work on Approximate Factorization

- No polynomial time algorithm (except for constant degree factors [Hitz, Kaltofen, Lakshman ’99])

Previous Work on Approximate Factorization

- No polynomial time algorithm (except for constant degree factors [Hitz, Kaltofen, Lakshman '99])

- Several algorithms and heuristics to find a nearby factorizable $\overline{f}$ if f is "nearly factorizable" [Corless et al. '01 & '02, Galligo and Rupprecht '01, Galligo and Watt '97, Huang et al. '00, Sasaki '01,...]

Previous Work on Approximate Factorization

- No polynomial time algorithm (except for constant degree factors [Hitz, Kaltofen, Lakshman '99])

- Several algorithms and heuristics to find a nearby factorizable $\overline{f}$ if f is "nearly factorizable"
  [Corless et al. '01 & '02, Galligo and Rupprecht '01, Galligo and Watt '97, Huang et al. '00, Sasaki '01,...]

- There are lower bounds for $\min \| f - \tilde{f} \|$ ("irreducibility radius")
  [Kaltofen and May ISSAC '03; Nagasaka CASC '04, '05]

Our ISSAC'04, ASCM'05, 2005, 2006 Results [joint with John May, Zhengfeng Yang, Lihong Zhi (and Shuhong Gao ISSAC'04)]

- Several practical algorithms to compute approximate multivariate GCDs

Our ISSAC'04, ASCM'05, 2005, 2006 Results [joint with John May, Zhengfeng Yang, Lihong Zhi (and Shuhong Gao ISSAC'04)]

- Several practical algorithms to compute approximate multivariate GCDs

- Practical algorithms to find the factorization of a nearby factorizable polynomial given any f

  especially "noisy" f:
  Given $f = f_1 \cdots f_s + f_{\text{noise}}$,
  we find $\bar{f}_1, \ldots \bar{f}_s$ s.t. $\| f_1 \cdots f_s - \bar{f}_1 \cdots \bar{f}_s \| \approx \| f_{\text{noise}} \|$

  even for large noise: $\| f_{\text{noise}} \| / \| f \| \geq 10^{-3}$

Maple Demonstration

Ruppert's Theorem (Bivariate Case)

$f \in \mathbb{K}[x,y]$, mdeg $f = (m,n)$

$\mathbb{K}$ is a field, algebraically closed, and characteristic 0

Theorem. f is reducible $\iff \exists g,h \in \mathbb{K}[x,y]$, non-zero,

$$\frac{\partial}{\partial y}\frac{g}{f} - \frac{\partial}{\partial x}\frac{h}{f} = 0$$

$$\text{mdeg}\, g \leq (m-2,n)\,, \text{mdeg}\, h \leq (m,n-1)$$

Ruppert's Theorem (Bivariate Case)

$f \in \mathbb{K}[x,y]$, mdeg $f = (m,n)$

$\mathbb{K}$ is a field, algebraically closed, and characteristic 0

Theorem. f is reducible $\iff \exists g,h \in \mathbb{K}[x,y]$, non-zero,

$$\frac{\partial}{\partial y}\frac{g}{f} - \frac{\partial}{\partial x}\frac{h}{f} = 0$$

$$\text{mdeg}\, g \leq (m-2,n)\,, \text{mdeg}\, h \leq (m,n-1)$$

PDE $\rightsquigarrow$ linear system in the coefficients of g and h

Ruppert's Theorem (Bivariate Case)

$f \in \mathbb{K}[x,y]$, mdeg $f = (m,n)$

$\mathbb{K}$ is a field, algebraically closed, and characteristic 0

Theorem. f is reducible $\iff \exists g,h \in \mathbb{K}[x,y]$, non-zero,

$$f\frac{\partial g}{\partial y} - g\frac{\partial f}{\partial y} + h\frac{\partial f}{\partial x} - f\frac{\partial h}{\partial x} = 0$$

$$\text{mdeg}\, g \leq (m-2,n)\,, \text{mdeg}\, h \leq (m,n-1)$$

PDE $\rightsquigarrow$ linear system in the coefficients of g and h

Gao's PDE based Factorizer

Change degree bound: $\text{mdeg}\, g \leq (m-1, n), \text{mdeg}\, h \leq (m, n-1)$

so that: # linearly indep. solutions to the PDE = # factors of f

Require square-freeness: $\text{GCD}(f, \frac{\partial f}{\partial x}) = 1$

Gao's PDE based Factorizer

Change degree bound: $\operatorname{mdeg} g \leq (m-1,n), \operatorname{mdeg} h \leq (m,n-1)$

so that: # linearly indep. solutions to the PDE = # factors of f

Require square-freeness: $\mathrm{GCD}(f, \frac{\partial f}{\partial x}) = 1$

Let

$$G = \mathrm{Span}_{\mathbb{C}}\{g \mid [g,h] \text{ is a solution to the PDE}\}.$$

Any solution $g \in G$ satisfies $g = \sum_{i=1}^{r} \lambda_i \frac{\partial f_i}{\partial x} \frac{f}{f_i}$ with $\lambda_i \in \mathbb{C}$, so

$$f = f_1 \cdots f_s = \prod_{\lambda \in \mathbb{C}} \gcd(f, g - \lambda \frac{\partial f}{\partial x})$$

(f_i the distinct irreducible factors of f)

With high probability $\exists$ distinct λ_i s.t. $f_i = \gcd(f, g - \lambda_i \frac{\partial f}{\partial x})$

Gao's PDE based Factorizer

Algorithm

Input: $f \in \mathbb{K}[x,y]$, $\mathbb{K} \subseteq \mathbb{C}$
Output: $f_1, \ldots, f_s \in \mathbb{C}[x,y]$

1. Find a basis for the linear space G, and choose a random element $g \in G$.

2. Compute the polynomial $E_g = \prod_i (z - \lambda_i)$ via an eigenvalue formulation
   If E_g not squarefree, choose a new g

3. Compute the factors $f_i = \gcd(f, g - \lambda_i \frac{\partial f}{\partial x})$ in $\mathbb{K}(\lambda_i)$.

In exact arithmetic the extension field $\mathbb{K}(\lambda_i)$ is found via univariate factorization.

Adapting to the Approximate Bivariate Case

The following must be solved to create an approximate factorizer from Gao's algorithm:

1. Computing approximate GCDs of bivariate polynomials;
2. Determining the numerical dimension of G, and computing an approximate solution g;
3. Randomize s.t. the polynomial E_g has no clusters of roots;
4. Compute approximate squarefree factorization.

Approximate Factorization

Input: $f \in \mathbb{C}[x,y]$ abs. irreducible, approx. square-free
Output: $f_1, \ldots, f_s$ approx. factors of f.

1. Compute the SVD of $\mathrm{Rup}(f)$, determine s, its approximate nullity, and choose $g = \sum a_i g_i$, a random linear combination of the last s right singular vectors

2. Compute E_g and its roots via an eigenvalue computation

3. For each λ_i compute the approximate GCD $f_i = \gcd(f, g - \lambda_i f)$

4. Optimize $\| f - f_1 \cdots f_s \|_2$ via Gauss-Newton iterative refinement.

Approximate Polynomial GCD via STLN [joint with Z. Yang and L. Zhi ISSAC 2006]

For polynomials $f_1, \ldots f_s \in \mathbb{C}[x_1, x_2, \ldots, x_r]$ with total degree $\deg(f_i) = m_i$ and a positive integer k with $k \leq \min(m_i)$, we compute $\Delta f_i \in \mathbb{C}[x_1, x_2, \ldots, x_r]$ such that $\deg(\Delta f_i) \leq m_i$, and

- $\deg(\mathrm{GCD}_i(f_i + \Delta f_i)) \geq k$,
- $\sum_i \|\Delta f_i\|_2^2$ is minimized.

Approximate Polynomial GCD via STLN [joint with Z. Yang and L. Zhi ISSAC 2006]

For polynomials $f_1, \ldots f_s \in \mathbb{C}[x_1, x_2, \ldots, x_r]$ with total degree $\deg(f_i) = m_i$ and a positive integer k with $k \leq \min(m_i)$, we compute $\Delta f_i \in \mathbb{C}[x_1, x_2, \ldots, x_r]$ such that $\deg(\Delta f_i) \leq m_i$, and

- $\deg(\mathrm{GCD}_i(f_i + \Delta f_i)) \geq k$,
- $\sum_i \|\Delta f_i\|_2^2$ is minimized.

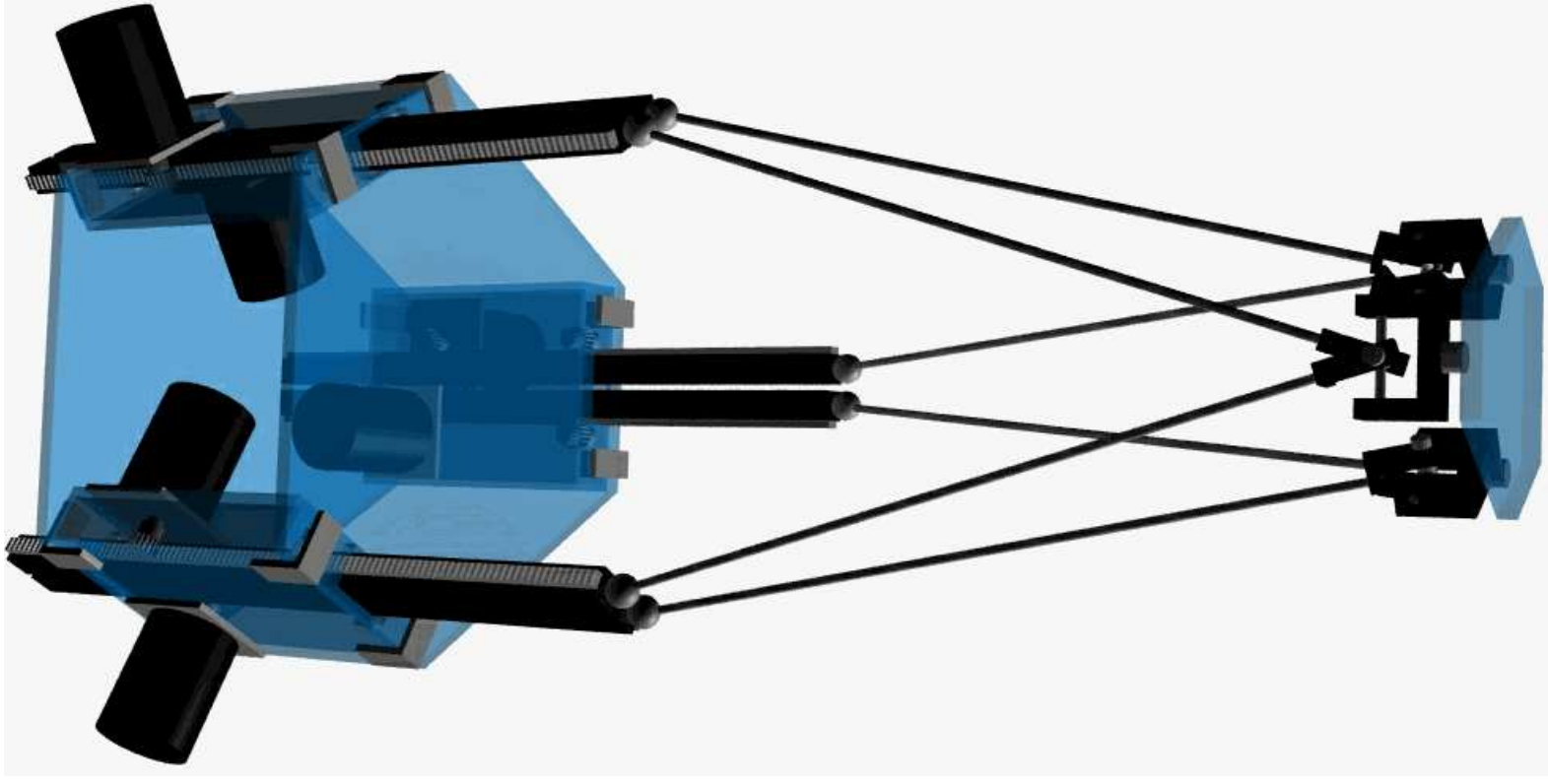
Based on structure preserving total least squares algorithms. Can be used to computed an approximate squarefree factorization.

More than two variables by sparse interpolation

- Our multivariate implementation together with Wen-shin Lee's numerical **sparse interpolation** code quickly factors polynomials arising in engineering Stewart-Gough platforms

  Polynomials were 3 variables, factor multiplicities up to 5, coefficient error 10^{-16}, are from [Sommese, Verschelde, Wampler 2004]

Stewart Platform Example



Josh Targownik’s bypass surgery motorized manipulator

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”)

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 (BPP): "always fast, probably correct". Examples: `isprime`

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

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

E.g., sparse polynomial interpolation, factorization, minimal polynomial and rank of a sparse matrix

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

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

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

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

– Las Vegas (RP): “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.

M. Agrawal, N. Kayal, N. Saxena, “PRIMES is in P,” 2002.

Kabanets and Impagliazzo [STOC 2003]
If Schwartz/Zippel **can be** de-randomized (subexponentially), then there **do not** exist polynomial-size circuits for NEXP or the permanent.

Zeev Dvir and Amir Shpilka, “Quasi-polynomial polynomial identity testing for depth-3 circuits with bounded top fan-in,” 2005.

Kabanets and Impagliazzo [STOC 2003]
If Schwartz/Zippel **can be** de-randomized (subexponentially), then there **do not** exist polynomial-size circuits for NEXP or the permanent.

Zeev Dvir and Amir Shpilka, “Quasi-polynomial polynomial identity testing for depth-3 circuits with bounded top fan-in,” 2005.

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

Black box polynomials

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

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

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

$$(n \cdot \deg(f))^{O(1)} \begin{cases} \text{black box calls,} \\ \text{arithmetic operations in } \mathbb{F} \text{ and} \\ \text{randomly selected elements in } \mathbb{F} \end{cases}$$

Black box matrices

$A \in \mathbb{F}^{n \times n}$ singular

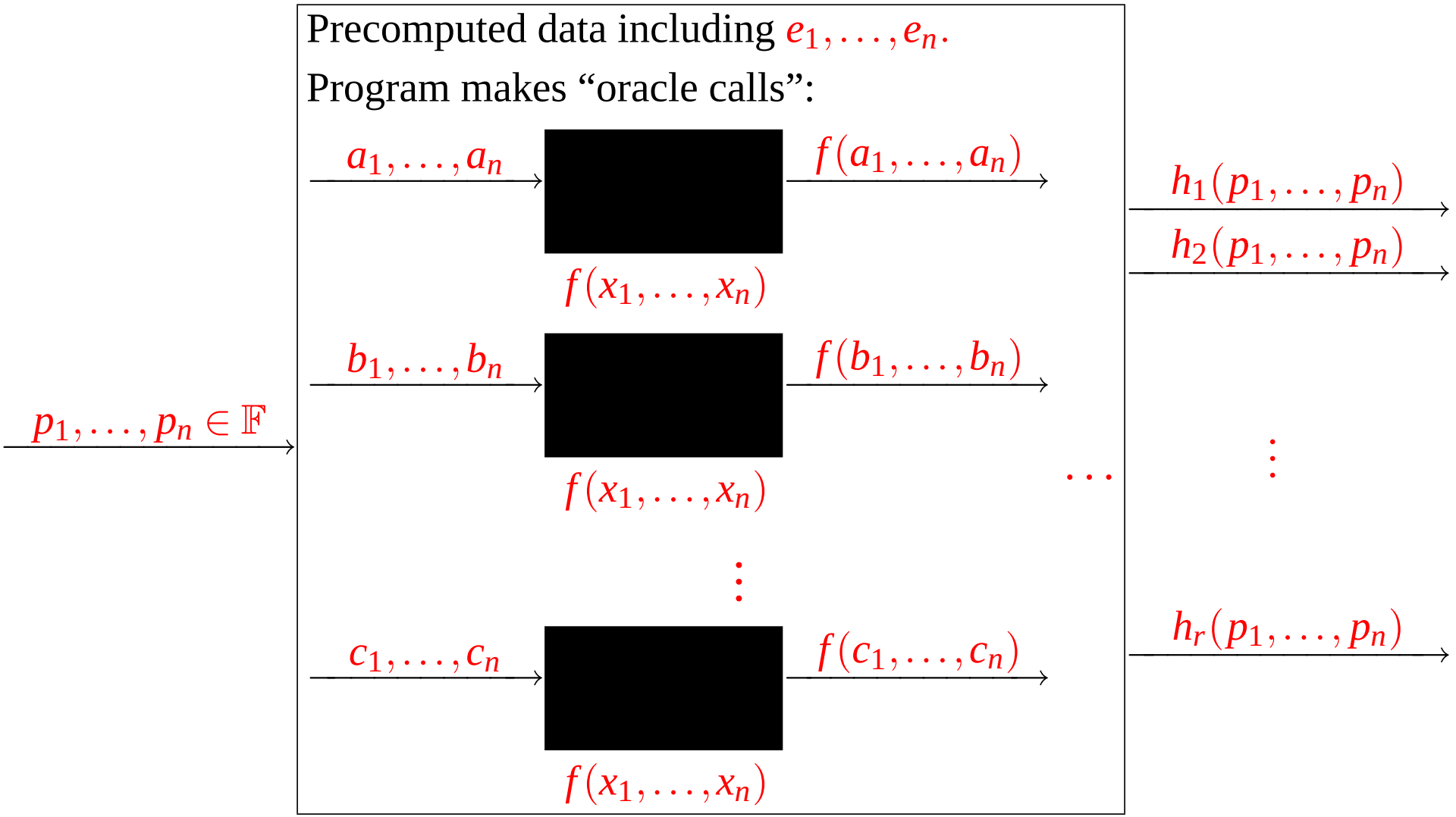
$\mathbb{F}$ an arbitrary, e.g., finite field

Perform linear algebra operations, e.g., $A^{-1}b$ [Wiedemann 86] with

$O(n)$	black box calls and
$n^2(\log n)^{O(1)}$	arithmetic operations in $\mathbb{F}$ and
$O(n)$	intermediate storage for field elements

LinBox Release 1.0 [`www.linalg.org`]: an exact Matlab

Black box manipulation (“functional programming”): Factorization [Kaltofen and Trager 1988]



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

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

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

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

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

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

Many algorithms that are polynomial-time in $\deg(f), n, t$:

Zippel 1979, 1988; Ben-Or, Tiwari 1988
Kaltofen, Lakshman, Wiley 1988, 1990
Grigoriev, Karpinski, Singer 1988
Kaltofen, Lee, Lobo 2000, 2003
Mansour 1992; Giesbrecht, Lee, Labahn 2006: numerical method

Show Wen-shin Lee's Maple worksheet

FoxBox [Díaz, Kaltofen 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.

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

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

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

Numerical + *randomized*, e.g., approximate factorizer:
all of the above (?)

Hallmarks of a good heuristic

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

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

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