The Art of Symbolic Computation

Erich Kaltofen
North Carolina State University
Department of Mathematics
www.kaltofen.us

Macdonald's Maple worksheet

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.

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.

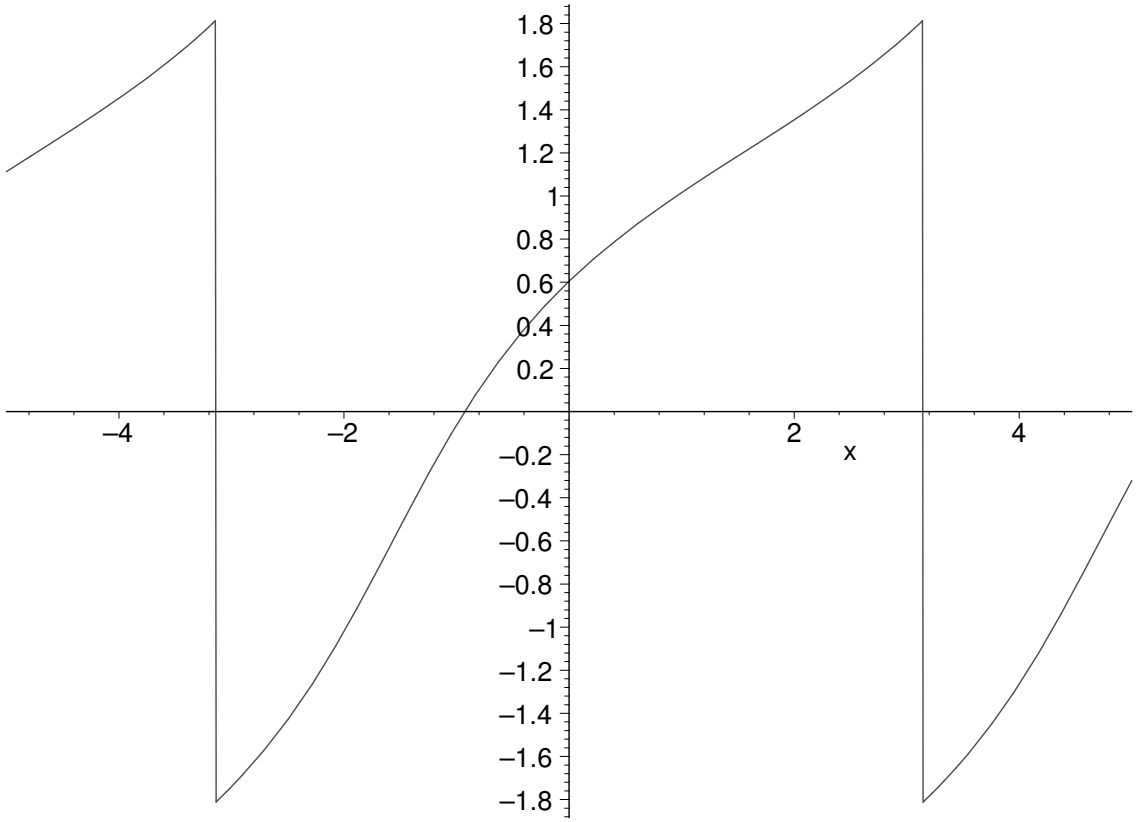
```
> # Example by Corless and Jeffrey
> f := 1/(sin(x) + 2);
```

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

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

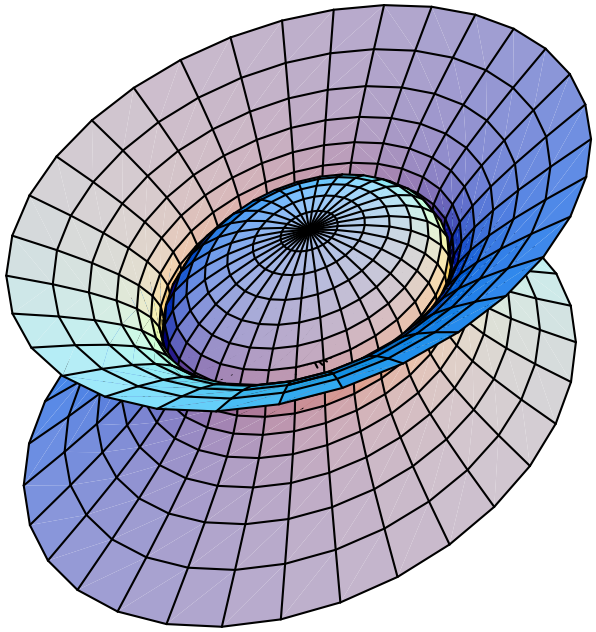
$$g := \frac{2}{3}\sqrt{3}\arctan\left(\frac{1}{3}\left(2\tan\left(\frac{1}{2}x\right)+1\right)\sqrt{3}\right)$$

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



Factorization of "noisy" polynomials over the complex numbers

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



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

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

The Approximate Factorization Problem [LATIN '94]

Given $f \in \mathbb{C}[x,y]$ irreducible, find $\tilde{f} \in \mathbb{C}[x,y]$ such that $\deg \tilde{f} \leq \deg f$, $\tilde{f}$ factors, and $\|f - \tilde{f}\|$ is minimal.

The Approximate Factorization Problem [LATIN '94]

Given $f \in \mathbb{C}[x,y]$ irreducible, find $\tilde{f} \in \mathbb{C}[x,y]$ such that $\deg \tilde{f} \leq \deg f$, $\tilde{f}$ factors, and $\|f - \tilde{f}\|$ is minimal.

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

We use Euclidean-norm, and multi-degree: $\mathrm{mdeg}\ f = (\deg_x f, \deg_y f)$

The Approximate Factorization Problem [LATIN '94]

Given $f \in \mathbb{C}[x,y]$ irreducible, find $\tilde{f} \in \mathbb{C}[x,y]$ such that $\deg \tilde{f} \leq \deg f$, $\tilde{f}$ factors, and $\|f - \tilde{f}\|$ is minimal.

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

We use Euclidean-norm, and multi-degree: $\operatorname{mdeg} f = (\deg_x f, \deg_y f)$

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

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

Status of the Approximate Factorization Problem

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

Status of the Approximate Factorization Problem

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

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

Status of the Approximate Factorization Problem

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

- Several algorithms and heuristics to find a nearby factorizable $\hat{f}$ 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 2003]

Our ISSAC'04 Results [Gao, Kaltofen, May, Yang, Zhi]

An algorithmically-engineered practical algorithm to find the factorization of a nearby factorizable polynomial given any f.

especially "noisy" f:
Given $f = f_1 f_2 + f_{\text{noise}}$,
we find $\bar{f}_1, \bar{f}_2$ such that $\|f_1 f_2 - \bar{f}_1 \bar{f}_2\| \approx \|f_{\text{noise}}\|$
even for large noise: $\|f_{\text{noise}}\| / \|f\| \geq 10^{-3}$

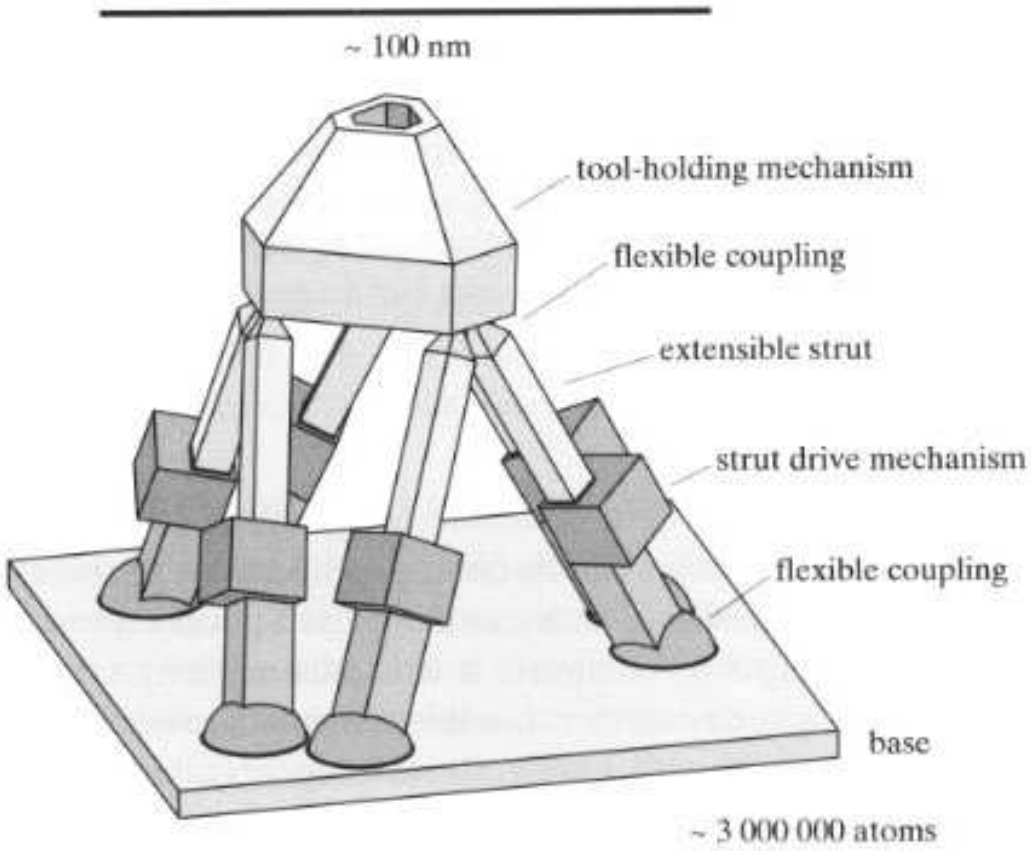
Our ISSAC'04 Results [Gao, Kaltofen, May, Yang, Zhi]

An algorithmically-engineered practical algorithm to find the factorization of a nearby factorizable polynomial given any f.

especially "noisy" f:
Given $f = f_1 f_2 + f_{\text{noise}}$,
we find $\bar{f}_1, \bar{f}_2$ such that $\|f_1 f_2 - \bar{f}_1 \bar{f}_2\| \approx \|f_{\text{noise}}\|$
even for large noise: $\|f_{\text{noise}}\| / \|f\| \geq 10^{-3}$

Show challenge problem Maple worksheet.

Verschelde's Stewart-Gouch Platform Benchmarks



Drexler's 1992 nano Stewart platform

Another of my ECCAD'98 Challenge Problems: #7

Problem 7: *Plug-and-play and generic programming methodology for symbolic computation*

Status: Open

Surprises from LinBox project using C++ allocators

```
myAllocator a;
myAllocator::pointer p = a.allocate(1);
a.construct(p,0); // effect: new((void*)p) T(0)
a.destroy(p);     // effect: ((T*)p)->~T()
a.deallocate(p,1);
```

Another of my ECCAD'98 Challenge Problems: #7

Problem 7: *Plug-and-play and generic programming methodology for symbolic computation*

Status: Open

Surprises from LinBox project using C++ allocators

```
myAllocator a;
myAllocator::pointer p = a.allocate(1);
a.construct(p,0); // effect: new((void*)p) T(0)
a.destroy(p);     // effect: ((T*)p)->~T()
a.deallocate(p,1);
```

ANSI/ISO 14882 Section 20.1.5.4

"Implementations of containers ... are permitted to assume that their Allocator template parameter meets the following two additional requirements ...

— the typedef members `pointer`, ... **are required to be** `T*` ..."

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 $f, g \in \mathbb{F}[x_1, \ldots, x_n], f \neq g, S \subseteq \mathbb{F}$.

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

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?

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.

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.

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

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

Black box matrices

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

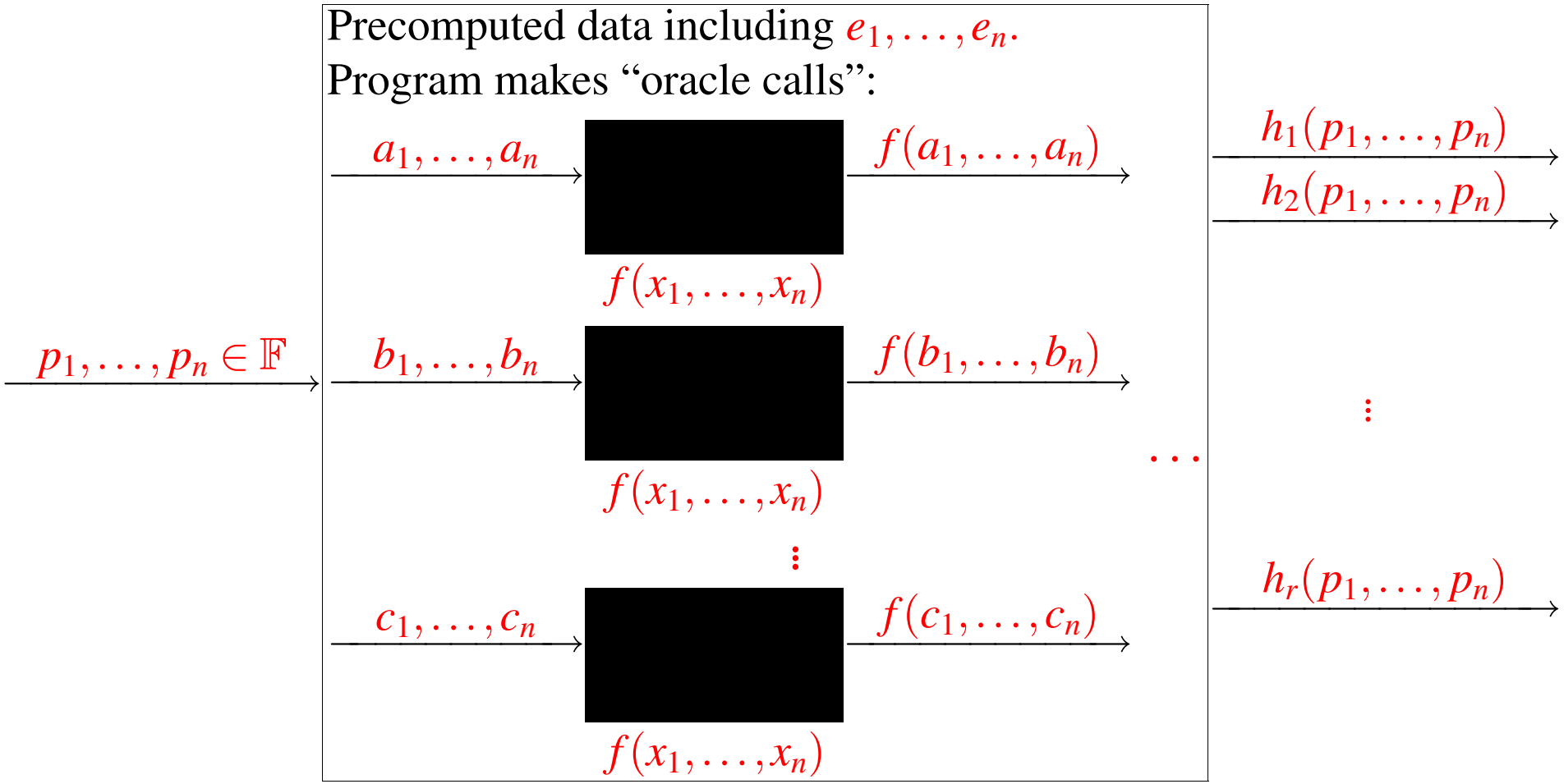
$\mathbb{F}$ an arbitrary, e.g., finite field

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

Project LinBox [`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

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

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 2003: numerical method

FoxBox [Díaz, Kaltofen 1998] example: determinant of symmetric Toeplitz matrix

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

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

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, $\sum_i \prod_j (x_i - j)$, Coppersmith's "pathological" matrices
convergence analysis	probabilistic analysis
try algorithms on unproven inputs	try algorithms with limited randomness

Numerical + *randomized*, e.g., 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

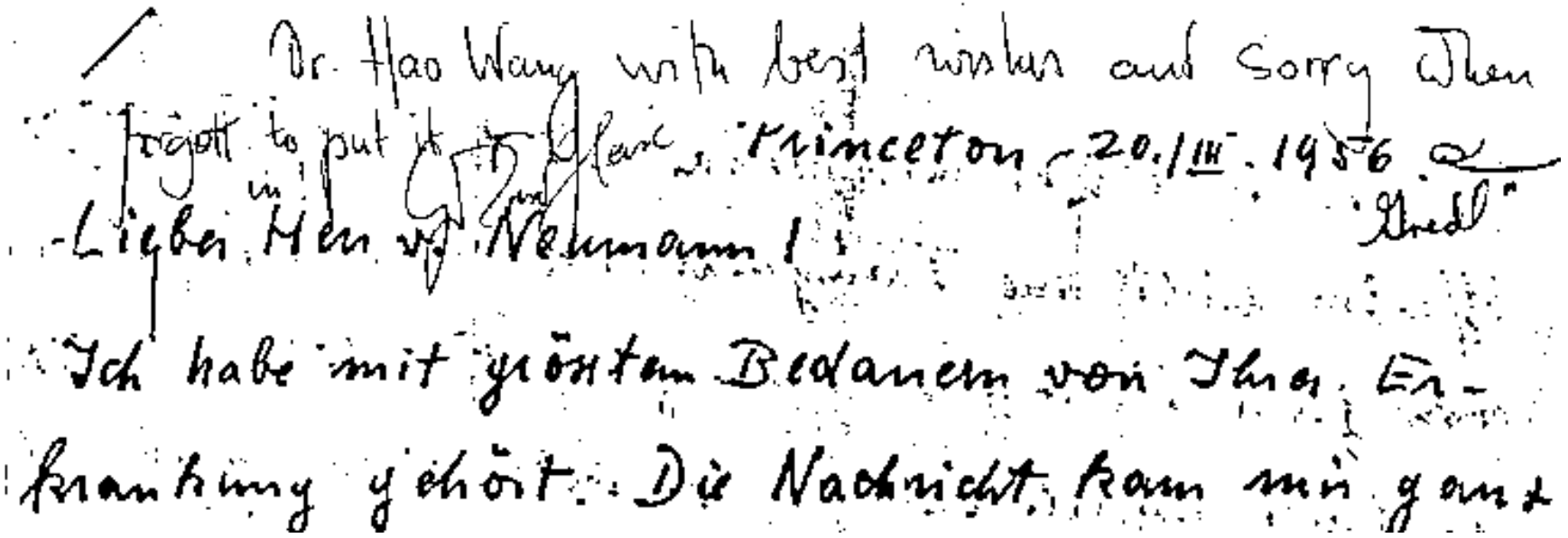
Hallmarks of a good heuristic

- 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

Letter by Gödel to John von Neumann 1956

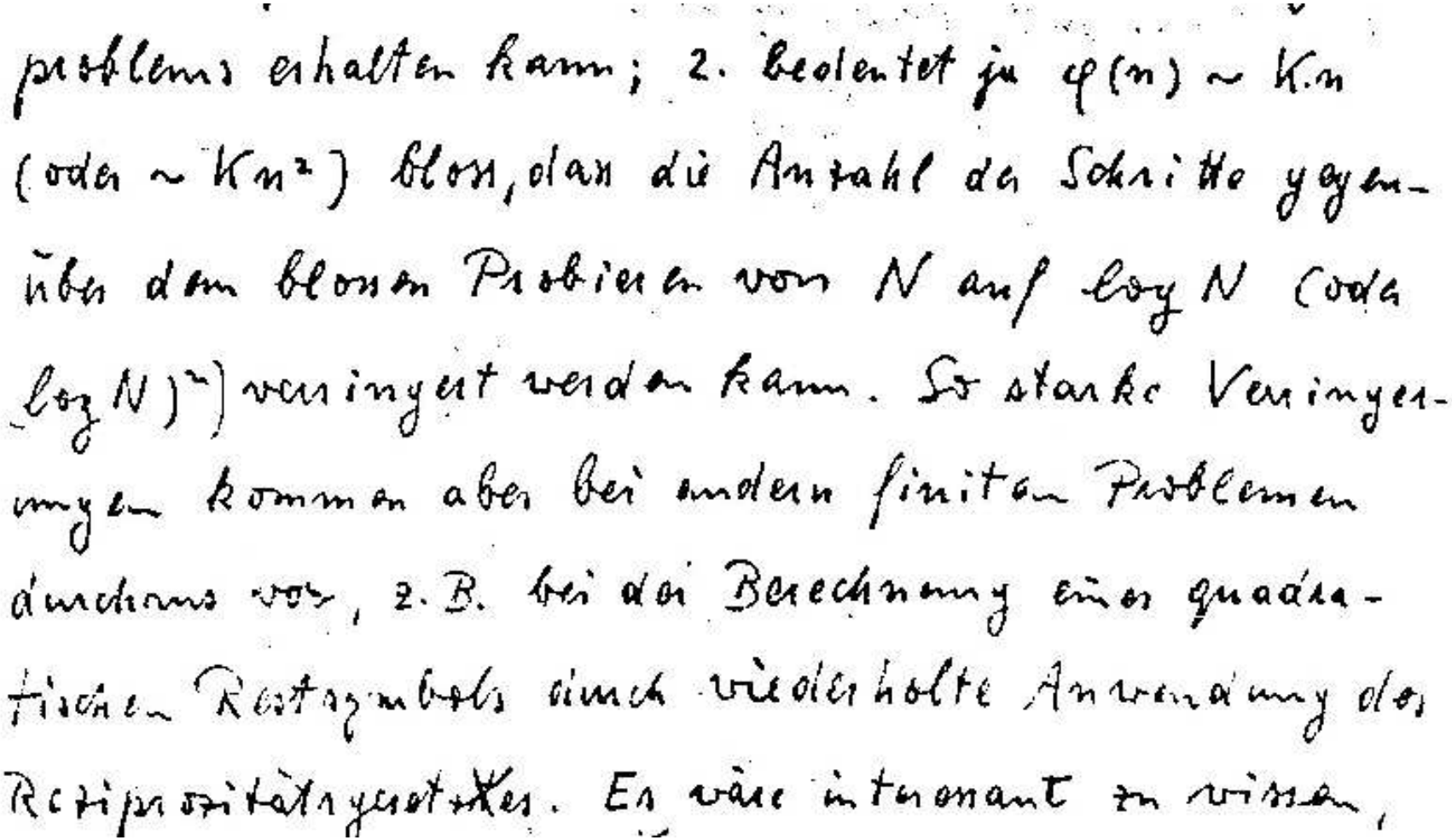

Dr. Hao Wang with best wishes and Sorry I forgot to put it in
Princeton 20./III. 1956
Goedl
Lieber Herr v. Neumann!
Ich habe mit größtem Bedauern von Ihrer Erkrankung gehört. Die Nachricht kam mir ganz

Princeton 20./III. 1956

Goedl

Lieber Herr v. Neumann!

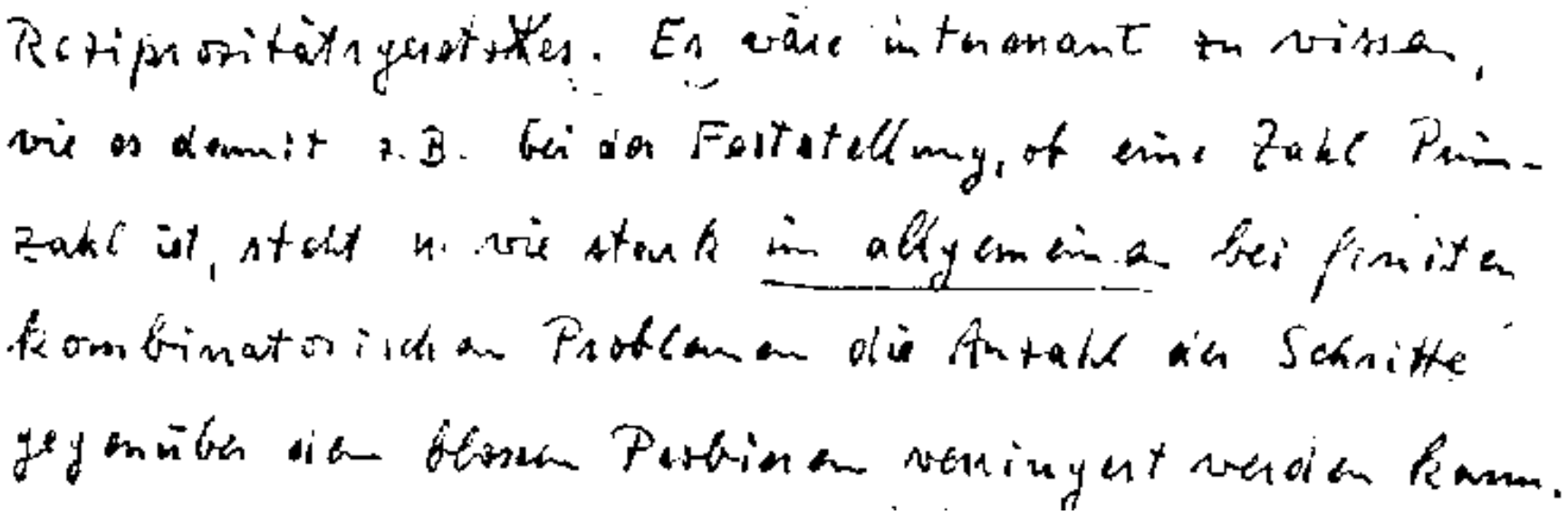
Letter by Gödel to John von Neumann 1956



problems erhalten kann; 2. bedeutet ja $\varphi(n) \sim Kn$ (oder $\sim Kn^2$) bloss, dass die Anzahl der Schritte gegenüber dem blossen Probieren von N auf $\log N$ (oder $(\log N)^2$) vermindert werden kann. So starke Verminderungen kommen aber bei anderen finiten Problemen durchaus vor, z.B. bei der Berechnung eines quadratischen Restsymbols durch wiederholte Anwendung des Reziprozitätsgesetzes. Es wäre interessant zu wissen,

... Such strong speedups [N to $(\log N)^2$] can occur for other finite problems, e.g. when computing the quadratic residuosity by repeated application of the reciprocity law.

Letter by Gödel to John von Neumann 1956



... It would be interesting to know, how it is with that, e.g. about the decision if a number is a prime number, a. how much in general for finite combinatorial problems the number of steps can be reduced versus trying all possibilities.

RSA with exponent 3

Private key: two prime numbers $P \equiv Q \equiv 2 \pmod 3$
Public key: $K = P \cdot Q$

Encryption of clear text $M \in \mathbb{Z}_K$

$$N = E_K(M) = (M^3 \bmod K)$$

Running time: 2 multiplications modulo K

Decryption of cipher text $N \in \mathbb{Z}_K$

$$M = D_{P,Q}(N) = (N^X \bmod K) \quad \text{where} \quad X = \frac{(P-1)(Q-1)+1}{3}$$

Running time: $\approx 1.5 \log_2 K$ multiplications modulo K
For $\log_2 K = 512$: ≈ 380-times slower than encryption

A Protocol for Spam Prevention [M. Naor et al., CRYPTO 2003]

From: "Dr. Cecilia Samarachi (Mrs)" <C.Samara91Dr@netscape.net>
Date: Sun, 25 May 2003 13:15:39
To: kaltofen@math.ncsu.edu
Dear Friend, VERY URGENT BUSINESS RELATIONSHIP.
...
My Ministry wants to award some major contracts and this contracts have been approved, implementation is on the pipeline and this contract is on supply of AGRICULTURAL CHEMICAL AND DRUGS/INJECTIONS FOR COW TREATMENT.
...
1. I want to use this last opportunity while still in the office to extract some money by inflating this contract to be awarded, and the over-invoiced amount I will use to establish my own hospital in U.K. or Germany after the transaction.

2. The inflated money (over-invoiced) from this contract will be immediately paid (Transfered) to my account in U.K. on confirmation of payment to your Bank.

3. I sincerely promise to approve your quotations on submission at all cost, provided my additional amount in your quotation will be 100% safe, immediately payment is made to your company. We would sign an agreement for the security and safety for my secret commission from the (over-invoiced) contarct.
...
Yours Faithfully,
Dr.(Mrs) Cecilia Samarachi.

Main idea:

1. take the unique message header as numeric data
2. spammer must perform “hard” computation and submit result with message
3. recipient “easily” checks result before accepting message

Main idea:
1. take the unique message header as numeric data
2. spammer must perform "hard" computation and submit result with message
3. recipient "easily" checks result before accepting message

Example: for message data N, compute digital signature M and "small" δ such that

$$M = D_{P,Q}(N+\delta) \quad \text{and} \quad 10^5 \text{ divides } M.$$

Note: 10^5 $D_{P,Q}$'s are much slower than verification that $|N - E_K(M)|$ is small.

Dwork, Goldberg, Naor design random table-lookup scheme that causes cache faults

NEEDED: non-localizable algorithmic problems whose results are easy to check

My suggestion: let spammer contribute to common good by spinning on a useful symbolic computation like a factorization, Gröbner basis,... problem