The art of symbolic computation

Erich Kaltofen
North Carolina State University
www.kaltofen.us
Where it began

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

\[ \int 1 + (x + 1)^n \, dx = x + (x + 1)^{n+1} / (n + 1) \]


Berlekamp/Zassenhaus’s, Risch’s algorithms

\[ \int \frac{x + 1}{x^4} e^{1/x} \, dx = - \frac{x^2 - x + 1}{x^2} e^{1/x} \]

> # 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);
What is an algorithm?

– **finite** unambiguous list of steps (“control, program”)
– computes a function from $D \rightarrow E$ 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?


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


Valentine Kabanets and Russell Impagliazo [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 does it take 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 $\varepsilon \in \mathbb{Q}$.

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

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

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 $\|f - \tilde{f}\|_2 \leq \varepsilon$. 
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
**Theorem** [Kaltofen and May ISSAC’03]

Let \( f, \tilde{f} \in \mathbb{C}[x, y] \),

\[ \deg_x \tilde{f} \leq m = \deg_x f, \quad \deg_y \tilde{f} \leq n = \deg_y f : \]

\( f \) irreducible and

\[ \| f - \tilde{f} \|_2 < \frac{\text{sm.pos.sing.val.}(\text{Ruppert mat.}(f))}{\max\{m, n\} \sqrt{2mn - m}} \]

\[ \implies \tilde{f} \text{ irreducible.} \]

**Example** \( f = x^2 + y^2 - 1 \).

\[ \| f - \tilde{f} \|_2 < 0.15843 \implies \tilde{f} \text{ irreducible.} \]

\[ \| f - (x + 1)(x - 1) \|_2 = 1 \]

\[ \| f - (0.49068y^2 + 0.84915x - 0.90735)(x + 1.21478) \|_2 = 0.67272 \]
Black box polynomials

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

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

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

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

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

\[ n^{O(1)} \] 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:

\[ p_1, \ldots, p_n \in \mathbb{F} \]

Precomputed data including \( e_1, \ldots, e_n \).

Program makes “oracle calls”:

- \( a_1, \ldots, a_n \) \( \rightarrow f(a_1, \ldots, a_n) \) \( \rightarrow h_1(p_1, \ldots, p_n) \)
- \( b_1, \ldots, b_n \) \( \rightarrow f(b_1, \ldots, b_n) \) \( \rightarrow h_2(p_1, \ldots, p_n) \)
- \( c_1, \ldots, c_n \) \( \rightarrow f(c_1, \ldots, c_n) \) \( \rightarrow h_r(p_1, \ldots, p_n) \)

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

\( h_i \in \mathbb{F}[x_1, \ldots, x_n] \) irreducible.
Given a black box

\[ p_1, \ldots, p_n \in \mathbb{F} \quad \longrightarrow \quad f(p_1, \ldots, p_n) \in \mathbb{F} \]

\[ f(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n] \]
\( \mathbb{F} \) 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)
Kaltofen, Lee, Lobo (2000)
Giesbrecht, Lee, Labahn (2003) numerical method
Sparsity with non-standard basis

In place of $x^e$ use

$$(x - a)^e \quad \text{shifted basis}$$

$$x(x + 1) \cdots (x + e - 1) \quad \text{Pochhammer basis}$$

$$T_e(x) \quad \text{Chebyshev basis}$$

Solutions (not all polynomial-time):

Grigoriev, Karpinski (1993): shifted
Lee (2001): Chebyshev, Pochhammer, shifted
Giesbrecht, Kaltofen and Lee (2002): shifted
FoxBox [Díaz and K 1998] example: determinant of symmetric Toeplitz matrix

\[
\text{det}( \begin{bmatrix}
a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \\
a_1 & a_0 & \cdots & a_{n-3} & a_{n-2} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
a_{n-2} & a_{n-3} & \cdots & a_0 & a_1 \\
a_{n-1} & a_{n-2} & \cdots & a_1 & a_0
\end{bmatrix} ) = F_1(a_0, \ldots, a_{n-1}) \cdot F_2(a_0, \ldots, a_{n-1}).
\]

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));

\[-(2dca - 2bce + 2c^2a - a^3 - da^2 + 2d^2c + d^2a + b^3 + 2abc - 2c^2b \\
+ d^3 + 2ab^2 - 2dcb - 2cb^2 - 2ec^2 + 2eb^2 + 2fcb + 2bae \\
+ b^2f + c^2f + be^2 - ba^2 - fdb - fda - f^2a - fba + e^2a - 2db^2 \\
+ dc^2 - 2deb - 2dec - dba)(2dca - 2bce - 2c^2a + a^3 \\
- da^2 - 2d^2c - d^2a + b^3 + 2abc - 2c^2b + d^3 - 2ab^2 + 2dcb \\
+ 2cb^2 + 2ec^2 - 2eb^2 - 2fcb + 2bae + b^2f + c^2f + be^2 - ba^2 \\
- fdb + fda - f^2a + fba - e^2a - 2db^2 + dc^2 + 2deb - 2dec \\
+ dba)\]
time 27.30 words 857700
FoxBox timings for symmetric Toeplitz determinant challenge

<table>
<thead>
<tr>
<th>$N$</th>
<th>CPU Time</th>
<th>Degree</th>
<th># Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$1^{h}20'$</td>
<td>5</td>
<td>931</td>
</tr>
<tr>
<td>11</td>
<td>$1^{h}34'$</td>
<td>5</td>
<td>847</td>
</tr>
<tr>
<td>12</td>
<td>$10^{h}14'$</td>
<td>6</td>
<td>5577</td>
</tr>
<tr>
<td>13</td>
<td>$15^{h}24'$</td>
<td>6</td>
<td>4982</td>
</tr>
</tbody>
</table>

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 $N = 10, 11, \text{ and } 12$ (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

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

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

End For

Threshold \( \eta \): 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 $\eta$.

*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* $\det(A) = O(n^{1-\alpha}b), \eta = O(1): (bn^{3+1/2-\alpha/2})^{1+o(1)}$

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

– [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].

Show Maple worksheet.
### Success and failure: different moduli and thresholds [Lee 2001]

\[ f_1 = x_1^2 x_3^3 x_4 x_6 x_8 x_9^2 + x_1 x_2 x_3 x_4^2 x_5^2 x_8 x_9 + x_2 x_3 x_4 x_5^2 x_6 x_7 x_8 + x_2 x_3 x_4 x_5^2 x_6 x_7 x_8 \]

\[ f_2 = x_1 x_2^2 x_4^2 x_6^2 x_9^2 x_{10} + x_2 x_4 x_5^2 x_6 x_7 x_9 x_{10} + x_1 x_2 x_3 x_5^2 x_7 x_8 + x_1 x_3 x_4^2 x_7 x_9 + x_1 x_3 x_4 x_7 x_8^2 \]

\[ f_3 = 9x_2 x_3^3 x_5^2 x_6^2 x_8 x_9 + 9x_2 x_3^3 x_4^2 x_5^2 x_7 x_8 + x_1^4 x_3 x_4^2 x_5^2 x_7 x_8 x_9 + 10x_1 x_2 x_3 x_4^2 x_5^2 x_7 x_8 x_9 + 12x_2 x_3 x_4 x_6 x_7 x_8^3 \]

\[ f_4 = 9x_1 x_3 x_4 x_6 x_7 x_8 x_{10} + 17x_1 x_2 x_3^2 x_5^2 x_6 x_7 x_8 x_9^3 x_{10} + 17x_2 x_3^2 x_4^2 x_5^2 x_6 x_7 x_8 x_9 x_{10} + 3x_1 x_2^2 x_3^2 x_6 x_{10} + 10x_1 x_3 x_5^2 x_6 x_7 x_8^4 \]

<table>
<thead>
<tr>
<th>Thresholds</th>
<th>mod 31</th>
<th>mod 37</th>
<th>mod 41</th>
<th>mod 43</th>
<th>mod 47</th>
<th>mod 53</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \eta, \zeta, \tau, \kappa, \gamma )</td>
<td>( = \neq ! )</td>
<td>( = \neq ! )</td>
<td>( = \neq ! )</td>
<td>( = \neq ! )</td>
<td>( = \neq ! )</td>
<td></td>
</tr>
<tr>
<td>( f_1 )</td>
<td>1 0 0</td>
<td>8 2 90</td>
<td>7 1 92</td>
<td>15 3 82</td>
<td>11 5 84</td>
<td>25 3 72</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>2 1 2</td>
<td>30 0 70</td>
<td>38 1 61</td>
<td>44 0 56</td>
<td>55 0 45</td>
<td>71 0 29</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>1 0 0</td>
<td>4 3 93</td>
<td>4 3 93</td>
<td>5 3 92</td>
<td>7 5 88</td>
<td>22 4 74</td>
</tr>
<tr>
<td>( f_4 )</td>
<td>1 0 0</td>
<td>0 2 98</td>
<td>0 6 94</td>
<td>3 3 94</td>
<td>4 0 96</td>
<td>6 5 89</td>
</tr>
<tr>
<td>Numerical</td>
<td>Randomized (Monte Carlo)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>more efficiency, but</td>
<td>more efficiency, but</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>approximate result</td>
<td>uncertain result</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ill-conditionedness</td>
<td>unfavorable inputs:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>near singular inputs</td>
<td>pseudo-primes,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\Sigma_i \Pi_j (x_i - j)$,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Coppersmith’s “pathological” matrices</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>convergence analysis</td>
<td>probabilistic analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>try algorithms on</td>
<td>try algorithms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unproven inputs</td>
<td>with limited randomness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 lattice-based factorization algorithm