Encounters in Symbolic Computation: Ideas for the Ages

Invited Talk

Erich L. Kaltofen
NCSU and Duke University
Raleigh, Durham, North Carolina, USA
kaltofen@ncsu.edu

ABSTRACT

Throughout my career, I had the good fortune to meet people and see ideas which have greatly influenced my scientific thinking. In the following I would like to tell my story of some.

CCS CONCEPTS

• **Theory of computation → Design and analysis of algorithms**.

KEYWORDS

History of polynomial factorization; computation of digits of pi; randomized linear algebra; Freivalds's algorithm; Wiedemann's algorithm;

ACM Reference Format:
Erich L. Kaltofen. 2024. Encounters in Symbolic Computation: Ideas for the Ages : Invited Talk. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '24), July 16–19, 2024, Raleigh, NC, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3666000.3672619

1. UNIVERSITY EDUCATION

Almost 50 years ago as a freshman in 1974–75 at the Johannes Kepler University in Linz, Austria, I wrote my first serious computer program. The program in Basic FORTRAN computed about 100 decimal digits of π on an IBM 1130 main frame computer. I used John Machin's 1706 formula

$$\pi = 16\arctan(\frac{1}{5}) - 4\arctan(\frac{1}{239})$$

with the Taylor series expansion $\arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} \mp \cdots$ and my own long integer decimal arithmetic procedures.

Remark 1.1. In 1974 I did not have Jonathan and Peter Borwein's 1984 AGM iterative algorithm, which I saw first in [5]:

$$\pi_0 = 2 + \sqrt{2};\ (\pi_0 \approx 3.414)$$
$$x_n = 0.5 * \left(\sqrt{x_{n-1}} + \frac{1}{\sqrt{x_{n-1}}}\right);$$
$$y_{n+1} = \left(y_n\sqrt{x_n} + \frac{1}{\sqrt{x_n}}\right) * \frac{1}{y_n + 1};$$
$$\pi_n = \pi_{n-1} * (x_n + 1) * \frac{1}{y_n + 1};$$

which can be implemented using only big-float addition, subtraction and multiplication and quadratically converging Newton iteration for big-float squareroots and inverses:

$$\mathrm{inv}_0 = \frac{1}{x};\ \text{(in short float)}$$
$$\mathrm{inv}_n = 2 * \mathrm{inv}_{n-1} - x * \mathrm{inv}_{n-1} * \mathrm{inv}_{n-1};$$
$$\mathrm{sqrt}_0 = 1;\ \mathrm{sqrt}_n = 0.5 * \left(\mathrm{sqrt}_{n-1} + x * \frac{1}{\mathrm{sqrt}_{n-1}}\right).$$

The AGM algorithm has quadratic convergence $0 < \pi_n - \pi < 10^{-2^{n+1}}$ for $n \geq 2$, and I would have had to do far less programming and could have computed many more digits. □

In the Summer 1977 I went to Rensselaer Polytechnic Institute (RPI) as an MS student in computer science on a Fulbright travel grant and an RPI graduate teaching assistantship. I had selected the offer by RPI after finding Bobby Caviness's and Robert McNaughton's names at RPI in Bruno Buchberger's list of addresses in Linz. I had searched Buchberger's list on his suggestion after I had asked him for advice. My Fall 1977 graduate courses included Artificial Intelligence taught by Joseph L. Mundy, Semantics of Programming Languages taught by S. Kamal Abdali and a Computer Algebra reading course with Bobby Caviness of Donald Knuth's Chapter 4 in Volume 2 of The Art of Computer Programming. As my MS project, in December 1977 I proposed to write a PASCAL program that compiled PASCAL programs into Abdali's λ-Calculus model for imperative programming language constructs. The λ-Calculus is a basic functional programming language with basic LISP-like evaluation. The compiler used an LL(1) context-free grammar for PASCAL, which was a technique I learned in P. M. Lewis II's Theory of Compiler Design class in Spring 1978. At that time I was fascinated by compilation: attaching meaning to a string of ASCII characters. It took me a year to write the compiler [23].

I stayed at RPI for a PhD in computer science, but I changed my research area to computer algebra and theoretical computer science. My decision in 1979 would fundamentally affect my career. In Fall 1978 Bobby Caviness had taught the graduate course Analysis of Algorithms using the book by Aho, Hopcroft and Ullman [1] and I had learned of the $\mathcal{P} \neq \mathcal{NP}$ problem.

1.1. My PhD Thesis

During my studies for my PhD Thesis, at the AMS Short Course *Computer Algebra—Symbolic Mathematical Computation* at the University of Michigan in August 1980 through my PhD thesis adviser Bobby Caviness, I made Hans Zassenhaus's acquaintance. Caviness had suggested the subject of polynomial factorization to me and in 1980 I was already familiar with Zassenhaus's introduction of Hensel lifting to integer polynomial factorization [55]. The "EZ" in the Hensel-lifting-based EZGCD algorithm stood for "Extended Zassenhaus." In late

ISSAC '24, July 16–19, 2024, Raleigh, NC, USA

ACM ISBN 979-8-4007-0696-7/24/07.
https://doi.org/10.1145/3666000.3672619

April 1981 Hans Zassenhaus sent me the manuscript for his Ann Arbor talk "A new polynomial factorization algorithm." The algorithm is similar to A. K. Lenstra's, H. W. Lenstra Jr.'s and L. Lovász's LLL algorithm [48, 49] and also computes a p-adic root. In the manuscript Zassenhaus points to H. W. Lenstra Jr.'s integer linear programming algorithm for solving the arising diophantine linear system with a size restriction on the integer solution. In his cover letter to me Zassenhaus suggests that the arising integer linear programming problem "finds its practical solution much more rapidly than available estimates would show." At that time I did not know the work by Helaman Ferguson and R. W. Forcade [10, 11].

I was also considering multivariate polynomial factorization, and my first result on the subject is a deterministic polynomial-time reduction to bivariate factorization via an effective Hilbert irreducibility theorem for function field coefficients, which in November 1981 I submitted to the STOC 1982 conference [24]. It took me until the Spring 1982 to realize that Hans Zassenhaus's method could be applied to reduce bivariate integer polynomial factorization to univariate by computing a Taylor series expansion for the roots in the algebraic function field of the second variable. The necessary polynomial bounds on the bit-lengths of the integer scalars were already my STOC paper. On April 14, 1982 I presented my algorithm at the University of Toronto and submitted it in May to FOCS 1982 [25]. Shortly before I had received the Math. Inst. Amsterdam Technical Report 82-05 (March 1982) on LLL [48]. Arjen Lenstra had brought a copy of the report to EUROCAM 1982 (April 5–7). Earlier, on October 7, 1981 Arjen Lenstra in a reply letter to me had written that the lattice algorithm was not polynomial time in the degree. In December 1981 Hendrik Lenstra Jr. in a postcard to László Lovász had mentioned the implications of Lovász's lattice basis reduction algorithm to polynomial factorization over $\mathbb{Q}$.[1] In that interview talk in Toronto in April 1982 I already stated the finished result: polynomial-time factoring of multivariate polynomials with integer coefficients. What a final year of my PhD research it had been!

1.2. Retrospective in 2024 of My PhD Thesis

As already stated in my FOCS 1982 paper, my polynomial-time reduction from multivariate to univariate polynomial factorization works over abstract fields, in particular, algebraic number fields and finite fields [14, 15, 36]. Lovász's lattice basis reduction algorithm has found many applications outside polynomial factorization. Hans Zassenhaus's "rapid practical solution" took another 20 years [21]. Multivariate polynomial factorization with floating point coefficients a little longer [13, 37].

One may speculate where the complexity of integer polynomial factorization would stand without Lovász's polynomial-time running-time analysis. At Toronto in 1982 Stephen A. Cook introduced me to the complexity class $\mathcal{NC} \subseteq \mathcal{P}$ of polynomial-time solvable problems that are also solvable on uniform polylog-depth Boolean circuits [7]. Whether integer GCD and more generally whether lattice basis reduction is $\in \mathcal{NC}$ is open. But whether a multivariate polynomial with integer coefficients is (absolutely) irreducible as a polynomial over the complex numbers is $\in \mathcal{NC}$, and with a suitable representation of algebraic numbers [50], factorization over the complex numbers is also $\in \mathcal{NC}$. The result follows by delaying the necessary univariate factorization in my reduction algorithm. I presented the algorithm for polynomial factors with complex coefficients in my talk at Harvard University on October 24, 1983 to an audience which included the algebraic geometer David Mumford, and published a paper in the inaugural issue of the Journal of Symbolic Computation [28].

In January 2022, I learned in an online virtual talk at NCSU by Theresa C. Anderson that Manjul Bhargava had solved the van der Waerden conjecture of 1936: let n be a constant degree. Then the probability

$$\text{Prob}\Big(\text{ Galois group}_{\mathbb{Q}}(f) \neq S_n \text{ (full symmetric group)} \;\Big|\; f(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_0, c_i \in \mathbb{Z},\ |c_i| \le H\Big) = O\big(1/H\big)\ [4].$$

The probability is asymptotically sharp: $1/(2H+1)$ of the polynomials have $c_0 = 0$ and a smaller Galois group. From the probabilistic version of my effective Hilbert irreducibility theorem [26, 27] one gets the following function field analogue: again, let n be a constant degree and let $\mathbb{F}_q$ be a finite field of q elements. Then the probability

$$\text{Prob}\Big(\text{ Galois group}_{\mathbb{F}_q(y)}(f) \neq S_n \text{ (full symmetric group)} \;\Big|\; f(x)=x^n+(b_{n-1}y+c_{n-1})x^{n-1}+\cdots+(b_0y+c_0),\ b_i, c_i \in \mathbb{F}_q\Big) = O\big(1/q\big)\ [35].$$

Again, the probability is asymptotically sharp.

2. RANDOMIZED ALGORITHMS

In June 1979 I attended the EUROSAM conference in Marseille, France, where both Jack Schwartz and Richard Zippel spoke about randomized polynomial identity testing. Because my MS project was in the discipline of program verification, I was already aware of the 1978 result by Richard DeMillo and Richard Lipton (see Remark 2.2 below).

First, let me give the analysis by Schwartz. The lemma would become a part in the analysis of many of my randomized algorithms, for example, the probabilistic effective Hilbert irreducibility theorem [27] mentioned before in connection with the van der Waerden conjecture and my algorithm for factoring multivariate polynomials represented by arithmetic circuits [30, 43, 53]. In Lemma 2.1 we use total degrees in subsets of the variables in f,

$$\deg_{x_1,\ldots,x_n}(f) = \max\Big\{ e_1 + \cdots + e_n \;\Big|\; x_1^{e_1}\cdots x_n^{e_1} \text{ is a factor of a term in } f \Big\}.$$

2.1. Jacob "Jack" Schwartz's 1979 Lemma

Lemma 2.1 ([51, 52]). *Let* K *be a field,* $f(x_1, \ldots, x_n, y_1, \ldots, y_m) \in \mathsf{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$, $f \neq 0$, *let* $d = \deg_{x_1,\ldots,x_n}(f)$, *let* $c_{\delta_1^*,\ldots,\delta_n^*}(y_1,\ldots,y_m)\, x_1^{\delta_1^*}\cdots x_n^{\delta_n^*}$ *be a monomial in* $x_1, \ldots, x_n$ *in* f, $c_{\delta_1^*,\ldots,\delta_n^*} \in \mathsf{K}[y_1,\ldots,y_m]$, $c_{\delta_1^*,\ldots,\delta_n^*} \neq 0$, *and let* $e^* =$

[1] LLL use $\mathbb{Q}$ as the coefficient domain to avoid the exception of factoring a large integer GCD of the coefficients. In my papers I state that an integer content requires integer factorization.

$\deg_{y_1,\ldots,y_m}(c_{\delta_1^*,\ldots,\delta_n^*})$; *note that* $e^* \le e = \deg_{y_1,\ldots,y_m}(f)$. *Furthermore, let* $S \subseteq \mathsf{K}$ *be a finite set with cardinality* $|S| \ge \max\{d, e^*\}$. *Then the probability*

$$\begin{aligned} Prob\big(& f(r_1,\ldots,r_{n+m}) \neq 0 \\ & \big| \; r_i \in S \text{ randomly uniformly selected} \big) \\ & \ge \Big(1-\frac{d}{|S|}\Big)\Big(1-\frac{e^*}{|S|}\Big) \\ & \ge \max\Big\{\Big(1-\frac{d}{|S|}\Big)\Big(1-\frac{e}{|S|}\Big), \\ & \qquad 1-\frac{\deg_{x_1,\ldots,x_n,y_1,\ldots,y_m}(f)}{|S|}\Big\}. \end{aligned} \tag{1}$$

We assume that $n \ge 1$ *but that the set of* y *variables may be empty: if* $m = 0$ *we can set* $e = e^* = 0$ *and the estimate* (1) *becomes* $\ge 1 - \deg_{x_1,\ldots,x_n}(f)/|S|$.

Proof. We prove by induction on $n+m$ for

$$W = \{(r_1,\ldots,r_{n+m}) \mid r_i \in S \text{ and } f(r_1,\ldots,r_{n+m}) \neq 0\}$$

that the cardinality $|W| \ge (|S|^n - d|S|^{n-1})(|S|^m - e^*|S|^{m-1})$. For $n+m=1$ we must have $n=1$ and $m = e = e^* = 0$. Then $f(x_1)$ has at most d roots in S and therefore $|W| \ge |S| - d$, which yields (1). For $n+m \ge 2$ we shall assume that $m \ge 1$. If $m = 0$ then we can set $\bar{f} = f(x_1,\ldots,x_{n-1},y_1)$ and obtain the probability estimate for f from the last estimate in (1) for $\bar{f}$ by $\deg_{x_1,\ldots,x_n}(f) = \deg_{x_1,\ldots,x_{n-1},y_1}(\bar{f})$. For $n \ge 1, m \ge 1$ let

$$V = \{(b_1,\ldots,b_m) \in S^m \mid c_{\delta_1^*,\ldots,\delta_n^*}(b_1,\ldots,b_m) \neq 0\},$$

and for $(b_1,\ldots,b_m) \in S^m$ let

$$\begin{aligned} U_{b_1,\ldots,b_m} = \{&(a_1,\ldots,a_n,b_1,\ldots,b_m) \\ &\mid (a_1,\ldots,a_n) \in S^n \text{ and } f(a_1,\ldots,a_n,b_1,\ldots,b_m) \neq 0\}. \end{aligned}$$

By induction hypothesis and $m < n+m$ and $n < n+m$, we have $|V| \ge |S|^m - e^*|S|^{m-1}$ and

$$\forall (b_1,\ldots,b_m) \in V\colon |U_{b_1,\ldots,b_m}| \ge |S|^n - d|S|^{n-1}.$$

Here we use (1) for $m = 0$ which is equivalent to having $\ge (1 - \deg_{x_1,\ldots,x_n}(f)/|S|) \times |S|^n$ non-zeros in S^n. Therefore,

$$|U| \ge \big(|S|^n - d|S|^{n-1}\big)\,\big(|S|^m - e^*|S|^{m-1}\big) \qquad \text{for } U = \bigcup\nolimits_{(b_1,\ldots,b_m)\in V} U_{b_1,\ldots,b_m}.$$

Furthermore $f(a_1,\ldots,a_n,b_1,\ldots,b_m) \neq 0$ for all $(a_1,\ldots,a_n,b_1,\ldots,b_m) \in U$, that is, $U \subseteq W$. The probability (1) is $= |W|/|S|^{n+m} \ge |U|/|S|^{n+m}$. The degree inequalities $d + e^* \le \deg_{x_1,\ldots,x_n,y_1,\ldots,y_m}(f)$ and $e \ge e^*$ and $(1-\epsilon_1)(1-\epsilon_2) \ge 1 - (\epsilon_1+\epsilon_2)$ for $\epsilon_1, \epsilon_2 \ge 0$ yield the rest of (1). □

Remark 2.2. DeMillo and Lipton in [8] for $m = 0$ prove the probability for $f(r_1,\ldots,r_n) \neq 0$ of $\ge (1 - \delta/|S|)^n$ as above by induction on n, where δ is the maximum of the degrees in the individual variables, $\delta = \max_i \deg_{x_i}(f)$. The authors then choose $|S| = n\delta$ and observe that $\lim_{n\to\infty}(1-1/n)^n = 1/\mathrm{e}$, with $\mathrm{e} = \exp(1)$ Euler's number. In fact, $1/\mathrm{e} > (1-1/n)^n \ge 1/4$ for all $n \ge 2$, and they suggest to perform the test k times, for a probability of $f(r_1,\ldots,r_n) \neq 0$ of $\ge 1 - (3/4)^k$. In [56, Theorem 1] the same probability estimate $\ge (1-\delta/|S|)^n$ of [8] is also proved and $(1-\delta/|S|)^n \ge 1 - n\delta/|S|$ deduced from the binomial expansion [57, p. 45]. In [51, 52] the refined estimates (1) are done. Schwartz's probability estimate $\ge 1 - \deg_{x_1,\ldots,x_n}(f)/|S|$ for $m = 0$ in Lemma 2.1 is often used. Schwartz also suggests that the estimate (1) can be iterated on several subsets of the variables including the sets of individual variables. □

2.2. Rūsiņš Freivalds's 1979 Quadratic-Time Matrix Multiplication Test

In September 1979 at the MFCS Conference in Olomouc, Czechoslovakia Freivalds presented what may be the first algorithm in randomized linear algebra [12]. The algorithm checks in $O(n^2)$ real arithmetic operations if $C = AB$ for matrices $A, B, C \in \mathbb{R}^{n\times n}$. Freivalds chooses a random vector $X \in \{-1,1\}^n$ and states that if $c_{i,j} \neq d_{i,j}$ with $d_{i,j}$ the (i,j)-th entry in AB and if $CX = A(BX)$, then the vectors $0^n \neq F_i = [c_{i,1} - d_{i,1}, \ldots, c_{i,n} - d_{i,n}]^T$ and X are orthogonal. Continuing in his own words, "it is easy to prove that no more than 2^{n-1} of all 2^n a priori possible vectors X are in the hyperplane orthogonal to F_i." Repeating the check $CX = A(BX)$ k-times exposes $C \neq AB$ with probability $\ge 1 - 1/2^k$. Note that $X \in \{0,1\}^n$ would also have worked for Freivalds.

Tracy Kimbrel and Rakesh Sinha in 1993 reduce then number of computed random bits in Freivalds's test to $\log(n) + O(1)$.

Theorem 2.3 ([47]). *Let* $\widetilde{A} \in \mathbb{C}^{n\times n}$, $\widetilde{A} \neq 0^{n\times n}$:

$$Prob(\widetilde{A}\underbrace{\begin{bmatrix} 1 \\ \xi \\ \xi^2 \\ \vdots \\ \xi^{n-1} \end{bmatrix}}_{X} \neq 0^n \mid \xi \in \mathbb{Z},\ 0 \le \xi \le 2n-3) \ge \frac{1}{2}.$$

Proof. Let $\widetilde{a}_{i,j} \neq 0$ and let ξ be a variable x. Then for the i-th row polynomial $f(x) = \cdots + \widetilde{a}_{i,j}x^{j-1} + \cdots$ in the variable x we have $f(x) \neq 0$, $\deg(f(x)) \le n-1$ and $f(x)$ has $\le n-1$ roots. □

For $\widetilde{A} = AB - C \in \mathbb{Z}^{n\times n}$, in order to have a bit complexity of $n^2(\log n)^{O(1)}$, Kimbrel and Sinha suggest reducing the entries in X modulo a prime p with $p \ge 2n-2$ and $p \ge |\widetilde{a}_{i,j}| + 1$ before performing the matrix times vector products BX, $A(BX)$ and CX. If the entries in $\widetilde{A}$ are large, p can be randomly selected to satisfy $\widetilde{a}_{i,j} \not\equiv 0 \pmod p$ with high probability.

In my talk "Hybrid Symbolic-Numeric Computation: A Marriage Made in Heaven" at the 8th International Congress on Industrial and Applied Mathematics (ICIAM) on August 10, 2015 in Beijing, China I presented an application of our hybrid multivariate sparse polynomial interpolation algorithms [44, 45]: see Figure 1.

The algorithm in Figure 1 reconstructs a matrix in sparse representation from a black box algorithm for the matrix times vector product by sparse polynomial interpolation. The problem was inspired by Jianlin Xia's talk "Randomized and matrix-free structured sparse direct solvers" at the ILAS Meeting in June 2013 in Providence, Rhode Island. In my ICIAM talk, I chose 2 variables so that the degrees in each in the sparse interpolants would be smaller, which results in a higher numerical stability [18–20].

Several variables also can be used in Kimbrel's and Sinha's 1993 matrix product checker.

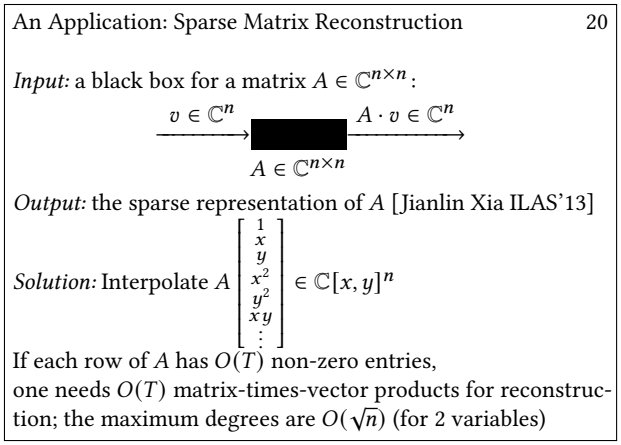

Figure 1: Page 20 of my ICIAM 2015 talk

Theorem 2.4. *Let* $\widetilde{A} \in \mathbb{C}^{n\times n}$, $\widetilde{A} \neq 0^{n\times n}$:

$$Prob\Big(\widetilde{A} \underbrace{\begin{bmatrix} 1 \\ \xi \\ \eta \\ \xi^2 \\ \eta^2 \\ \xi\eta \\ \vdots \end{bmatrix}}_{X} \neq 0^n \mid \xi,\eta \in \mathbb{Z},\ 0 \le \xi,\eta \le 4\lfloor\sqrt{n}\rfloor - 1\Big) \ge \left(1-\frac{1}{4}\right)^2 = \frac{9}{16}$$

Proof. Let $\widetilde{a}_{i,j} \neq 0$. We represent the column indices $j = \mu+(\lfloor\sqrt{n}\rfloor+1)\nu+1$ with $0 \le \mu,\nu \le \lfloor\sqrt{n}\rfloor$ and place the symbolic term $x^\mu y^\nu$ in the j-th row of X. There are $(\lfloor\sqrt{n}\rfloor+1)^2 > n$ such terms. The i-th row bivariate polynomial $f(x,y) = \cdots + \widetilde{a}_{i,j}x^\mu y^\nu + \cdots$ is $\neq 0$ and $d = \deg_x(f), e = \deg_y(f) \le \lfloor\sqrt{n}\rfloor$. The probability bound follows by Schwartz's Lemma 2.1: $|S| \ge 4d, 4e$. □

Selecting X again requires $\log(n) + O(1)$ random bits. For $\widetilde{A} \in \mathbb{Z}^{n\times n}$ one reduces the entries in X modulo a prime $p \ge 4\lfloor\sqrt{n}\rfloor, |\widetilde{a}_{i,j}|+1 \Longrightarrow$ there are $\max\left(\frac{\log(n)}{2}+O(1), \log(|\widetilde{a}_{i,j}|+2)\right)$ bits in the entries in X.

The entry size in X shrinks the more variables one uses in the vector X until one chooses $n-1$ variables, in which case one returns to Freivalds's check. We have the following general theorem:

Theorem 2.5. *Let* $\widetilde{A} \in \mathsf{R}^{n\times n}$, $\widetilde{A} \neq 0^{n\times n}$, R *an arbitrary ring.*

$$Prob(\widetilde{A} \begin{bmatrix} 1 \\ \xi_2 \\ \\ \vdots \\ \xi_n \end{bmatrix} \neq 0^n \mid \xi_i \in \{0,1\}) \ge \frac{1}{2}$$

Proof. *Case* $\widetilde{A} = \begin{bmatrix} * & 0 & \dots & 0 \\ * & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ * & 0 & \dots & 0 \end{bmatrix} \neq 0^{n\times n}$
$\Longrightarrow \exists i\colon \sum_j \widetilde{a}_{i,j}\xi_j = \widetilde{a}_{i,1} \neq 0.$
Case $\exists j \ge 2\colon \widetilde{a}_{i,j} \neq 0$: Let $\xi_1 = 1$. Then
not both $\widetilde{a}_{i,j} + \sum_{k\neq j} \widetilde{a}_{i,k}\xi_k \quad (\xi_j = 1)$
$\sum_{k\neq j} \widetilde{a}_{i,k}\xi_k \quad (\xi_j = 0) \quad$ can be $= 0$,
because otherwise their difference $\widetilde{a}_{i,j}$ would be $= 0$. Therefore at least half the vectors certify that $\widetilde{A}$ is not the zero matrix. □

Note that Theorem 2.5 saves 1 random bit in Freivalds's original algorithm and is valid, for instance, over the integers modulo p^ℓ, which occur in lifting algorithms. Schwartz's Lemma 2.1 and its variants, of course, is not: The non-zero polynomial $\sum_i 2^{\ell-1}x_i(x_i+1) \in \mathbb{Z}_{2^\ell}[x_1,\dots,x_n]$ evaluates to zero for all evaluations of the x_i's at residues $\in \mathbb{Z}_{2^\ell}$.

We have used Freivalds's quadratic-time matrix multiplication check to construct a certificate for the rank of an integer matrix which can be Monte Carlo ("always fast, probably correct") checked in $n^2(\log n)^{O(1)}$ bit complexity [38].

2.3. Douglas H. Wiedemann's 1985 Algorithms

In Fall 1985 when we were at MSRI in Berkeley for the Complexity Theory program, Gary Miller handed me a manuscript "Solving Sparse Linear Equations Over Finite Fields" by Douglas Wiedemann. I remembered the paper a few years later when working on algorithms for polynomials given by black box algorithms that evaluate them [42]. In the introduction of our 1990 JSC paper [43] we introduce the term "black box matrix," which is given by an algorithm for the matrix times vector product, and point to Wiedemann's work [54]. From 1990 on, black box matrix algorithms were on my mind, teaching Wiedemann's algorithms for the first time in a graduate computer science course during my sabbatical semester at the University of Toronto in Spring 1991. An outcome are our 1995 subquadratic-time polynomial factorization algorithms in $\mathbb{Z}_p[x]$ with Victor Shoup [39–41].

Let me briefly describe Wiedemann's basic idea, for a non-singular coefficient matrix $A \in \mathsf{K}^{n\times n}$ and a right-side vector $b \in \mathsf{K}^n$, where K is an abstract field. First, one has the sequence of Krylov vectors $A^i b \in \mathsf{K}^n$ for $i \ge 0$ with $A^0 b = b$. The coefficients of the minimum polynomial of the matrix A, denoted by $f^A(z) \in \mathsf{K}[z]$, constitute a linear relation with scalar coefficients and therefore the Krylov vector sequence is linearly generated by a polynomial divisor $f^{A,b}(z)$ of $f^A(z)$ of minimal degree: for $f^{A,b}(z) = z^m + c_{m-1}z^{m-1} + \cdots + c_0 \in \mathsf{K}[z]$ we have $A^{m+j}b = -c_{m-1}A^{m-1+j}b - \cdots - c_0A^jb$ for all $j \ge 0$. The Wiedemann sequence is a projection of the Krylov sequence: for a vector $u \in \mathsf{K}^n$ one computes the sequence of scalar vector products $u^TA^ib \in \mathsf{K}$ for $i \ge 0$. Again, the sequence is linearly generated by a minimal degree polynomial divisor $f_u^{A,b}(z) = z^{\widetilde{m}} + \widetilde{c}_{\widetilde{m}-1}z^{\widetilde{m}-1} + \cdots + \widetilde{c}_0 \in \mathsf{K}[z]$ of $f^{A,b}(z)$.

The basic Wiedemann linear system solver proceeds as follows:

1. Select a random vector $u \in S^n$, where $S \subseteq \mathsf{K}$ is a finite set.
2. Compute $f_u^{A,b}(z)$ from the Wiedemann sequence u^TA^ib by the Berlekamp-Massey algorithm. One needs at most $n + \deg(f_u^{A,b})$ sequence elements [46, Remark 4.20] and $O(n^2)$ field operations.
3. If $f_u^{A,b}(z) = f^{A,b}(z)$ then $b = A(\underbrace{(-1/c_0)\sum_{i\ge1} c_iA^{i-1}b}_{A^{-1}b})$.

Overall, the basic algorithm performs $\le 3n-1$ black box $A\times$ a vector products, $O(n^2)$ arithmetic operations, and requires $O(n)$ auxiliary storage for elements in K. A major contribution by Wiedemann is the probabilistic analysis for the condition

$f_u^{A,b}(z) = f^{A,b}(z)$, especially for small finite fields such as $\mathbb{Z}_2$.

In [33] we prove that the probability for $f_u^{A,b} = f^{A,b}$ is $\geq 1 - \deg(f^{A,b})/|S| \geq 1 - n/|S|$; $|S|$ is the number of elements in S. Our 1995 proof can also be extended to vectors u of the forms X in Theorems 2.3–2.5. We explain the Theorem 2.4 case.

Theorem 2.6.

$$Prob\Big(f_u^{A,b} = f^{A,b} \mid \xi,\eta \in S, u = \begin{bmatrix} 1 \\ \xi \\ \eta \\ \xi^2 \\ \eta^2 \\ \xi\eta \\ \vdots \end{bmatrix}\Big) \geq \Big(1 - \frac{n\lfloor\sqrt{n}\rfloor}{|S|}\Big)^2 \geq 1 - \frac{2n\lfloor\sqrt{n}\rfloor}{|S|}.$$

Proof. Let $u(x,y) = \begin{bmatrix} 1 \\ x \\ y \\ \vdots \end{bmatrix}$ and let the elements in the Wiedemann sequence be in the function field $\mathsf{K}(x,y)$. Because $f_{u(x,y)}^{A,b}(z) \in \mathsf{K}(x,y)[z]$ divides $f^{A,b}(z) \in \mathsf{K}[z]$, whose coefficients do not contain the variables x, y, we must have

$$f_{u(x,y)}^{A,b}(z) = z^{\widetilde{m}} + \widetilde{c}_{\widetilde{m}-1} z^{\widetilde{m}-1} + \cdots + \widetilde{c}_0 \in \mathsf{K}[z].$$

Because $f_{u(x,y)}^{A,b}$ generates the Wiedemann sequence with the symbolic $u(x,y)$, we have for all $j \geq 0$:

$$\sum_{\mu,\nu} x^\mu y^\nu (A^{\widetilde{m}+j} b)_{k(\mu,\nu)} = -\widetilde{c}_{m-1}\Big(\sum_{\mu,\nu} x^\mu y^\nu (A^{\widetilde{m}-1+j} b)_{k(\mu,\nu)}\Big) - \cdots - \widetilde{c}_0\Big(\sum_{\mu,\nu} x^\mu y^\nu (A^j b)_{k(\mu,\nu)}\Big),$$

where the range of (μ,ν) pairs is $0 \leq \mu,\nu \leq \lfloor\sqrt{n}\rfloor$ $1 \leq k(\mu,\nu) = \mu + (\lfloor\sqrt{n}\rfloor + 1)\nu + 1 \leq n$. Note again that $(\lfloor\sqrt{n}\rfloor + 1)^2 > n$. The coefficient of $x^\mu y^\nu$ is the $k(\mu,\nu)$-th row in the Krylov sequence, and $f_{u(x,y)}^{A,b}(z)$ simultaneously linearly generates all the rows. Therefore, $f_{u(x,y)}^{A,b}(z) = f^{A,b}(z)$ and $\widetilde{m} = m$.

Now consider the Hankel matrix $H(x,y) \in \mathsf{K}(x,y)^{m\times m}$ with $h_{i,j}(x,y) = u(x,y)^T A^{m+i-j-1} b \in \mathsf{K}(x,y)$. Because $f_{u(x,y)}^{A,b}$ is the minimum degree linear generator, $\Delta(x,y) = \det(H(x,y)) \neq 0$. For any $\xi,\eta \in \mathsf{K}$ with $\Delta(\xi,\eta) \neq 0$ we have by properties of minimum linear generators $\deg(f_{u(\xi,\eta)}^{A,b}) = m$ and therefore $f_{u(\xi,\eta)}^{A,b} = f^{A,b}$. Finally, both degrees $\deg_x(h_{i,j}(x,y))$, $\deg_y(h_{i,j}(x,y)) \leq \lfloor\sqrt{n}\rfloor$ and therefore

$$\deg_x(\Delta(x,y)), \deg_y(\Delta(x,y)) \leq m\lfloor\sqrt{n}\rfloor \leq n\lfloor\sqrt{n}\rfloor.$$

The probability estimates follow by Lemma 2.1. □

Several of Wiedemann's algorithms can now be converted to use only $O(\log n)$ or $(\log n)^{O(1)}$ random bits.

1. For non-singular matrices A we have a Las Vegas ("probably fast, always correct") algorithm for $A^{-1}b$ as described above.
2. We have a Monte Carlo algorithm for the minimum polynomial $f^A(z)$ of A. The minimum polynomial is computed, as suggested by Wiedemann, by also selecting a random vector b and computing $f_u^{A,b}$. For example, we have the following corollary:

Corollary 2.7.

$$Prob\Big(f^A = f_u^{A,b} \mid \beta,\gamma,\xi,\eta \in S, b = \begin{bmatrix} 1 \\ \beta \\ \gamma \\ \beta^2 \\ \gamma^2 \\ \beta\gamma \\ \vdots \end{bmatrix}, u = \begin{bmatrix} 1 \\ \xi \\ \eta \\ \xi^2 \\ \eta^2 \\ \xi\eta \\ \vdots \end{bmatrix}\Big) \geq \Big(1 - \frac{n\lfloor\sqrt{n}\rfloor}{|S|}\Big)^4 \geq 1 - \frac{4n\lfloor\sqrt{n}\rfloor}{|S|}.$$

Proof. As in the proof of Theorem 2.6, we can prove that if β,γ in b are left variables $\bar{x},\bar{y}$, one has $f^A = f^{A,b(\bar{x},\bar{y})}$ over $\mathsf{K}(\bar{x},\bar{y})$. As before, we then have $f^A = f_{u(x,y)}^{A,b(\bar{x},\bar{y})}$ and the determinant of the corresponding Hankel matrix yields the probability of the corollary. □

3. From the algorithm for f^A in 2. we get for singular matrices A, as suggested by Wiedemann, a Las Vegas algorithm for a non-zero nullspace vector w: $Aw = 0, w \neq 0^n$. For the random vectors u, b that yield $f^A(z) = f_u^{A,b}(z)$ we have $w = \Big(\frac{f^A(z)}{z}\Big|_{z=A}\Big) b \neq 0$ because $f^{A,b}(z) = f^A(z) \neq \frac{f^A(z)}{z}$.
4. We can compute $\det(A)$ by a Wiedemann-class Las Vegas randomized algorithm which uses $O(\log n)$ random bits [9, Section 7]. By Lemma 1 in [9], for a non-singular matrix $A \in \mathsf{K}^{n\times n}$ the characteristic polynomial $\det(I_n z - A\,\Gamma(x,y))$ of $A\,\Gamma(x,y)$ is irreducible in $\mathsf{K}(x,y)[z]$, where

$$\Gamma(x,y) = \begin{bmatrix} y & -1 & 0 & \dots & 0 \\ 0 & y & -1 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & & \ddots & y & -1 \\ x & 0 & \dots & 0 & y \end{bmatrix} \in \mathsf{K}[x,y]^{n\times n},$$

and $\det(\Gamma) = y^n + x$. We therefore must have

$$f_{e_1}^{A\Gamma(x,y),e_1}(z) = \det(I_n z - A\Gamma(x,y)).$$

projecting with the unit vector $e_1^T = [1,0,\dots,0]$. For randomly selected $\sigma,\tau \in S$ that are not zeros of the corresponding Hankel determinant [9, Eq. (9)] and $\det(\Gamma)$ we get $\deg\big(f_{e_1}^{A\Gamma(\sigma,\tau),e_1}\big) = n$ and $\det(A) = (-1)^n f_{e_1}^{A\Gamma(\sigma,\tau),e_1}(0) \;/\; (\sigma^n + \tau)$. The probability of success is bounded, for non-singular matrices, as $\geq 1 - n^2/|S|$ [9, Eq. (10) and $\sigma^n + \tau \neq 0$]. By 3. the algorithm becomes Las Vegas for all matrices.
5. From the algorithm for w in 3. we get for singular matrices A of co-rank $n - r = (\log n)^{O(1)}$ a Monte Carlo algorithm for the rank r of A. Wiedemann suggests to successively remove a dependent row of A as determined by the entries of w.

However, we have not been able to reduce to $(\log n)^{O(1)}$ the needed number of random bits in Wiedemann-class algorithms for computing the rank of all black box matrices A and solutions to highly singular inhomogeneous linear systems [6].

3. MORE COMPUTATIONS OF π

I would encounter the problem of computing digits of π for a second time. There is an algorithm for computing the decimal digits of certain transcendental numbers, like $\log(10/9)$ or $\arctan(1/10)$, starting at the d-th decimal place without

computing intermediate decimal digits. For a real number $\xi > 0$, let dec.part$(\xi) = \xi - \lfloor \xi \rfloor$. We explain the algorithm for $\log(10/9) = \sum_{n=1}^{\infty} 1/(n10^n)$. We have

$$\text{dec.part}(10^d \log(\frac{10}{9})) = \underbrace{\text{dec.part}\Big(\sum_{n=1}^{d} \frac{10^{d-n}}{n}\Big)}_{\text{dec.part}\Big(\sum_{n=1}^{d} \frac{10^{d-n} \bmod n}{n}\Big)} + \underbrace{\sum_{n=d+1}^{\infty} \frac{1}{n\,10^{n-d}}}_{< \frac{1}{9(d+1)}}$$

One computes all terms $\frac{10^{d-n} \bmod n}{n}$ by computing the numerators by repeated squaring and the fractions within floating point precision $\epsilon = \frac{1}{9d(d+1)}$ in a truly hybrid symbolic-numeric fashion. One then can obtain the infinite sum within precision $\frac{1}{9(d+1)} + \frac{1}{9(d+1)}$. Finally the fractional part is converted to decimal representation. In summary, with $O(\log d)$ integer length and floating point precision one can compute a floating point number ξ such that $|\xi - \text{dec.part}(10^d \log(\frac{10}{9}))| < \frac{2}{9(d+1)}$. The algorithm has $d(\log d)^{O(1)}$ bit complexity and requires $O(\log d)$ intermediate storage. It is easily parallelized.

David Bailey, Peter Borwein and Simon Plouffe in 1995 published the following formula for π [2, 3], [34, Section 4]:

$$\pi = \sum_{n=0}^{\infty} \frac{1}{16^n} \left(\frac{4}{8n+1} - \frac{2}{8n+4} - \frac{1}{8n+5} - \frac{1}{8n+6} \right). \quad (2)$$

With (2) one can compute an approximation of the binary expansion of π starting at the d-th place like above. The BBP algorithm was used to check the June 2022 Google cloud computation of 100 trillion decimal digits of π [22].

The problem remains to do the same for the decimal expansion of π. In my email correspondence with Peter Borwein about the formula (2), he wrote back to me on October 9, 1997: "We searched pretty hard over different bases. It would be finding one based on 3 or even better 10 that would make our day." When I visited Simon Fraser University in July 1998 I told Peter Borwein that by Boolean circuit complexity theory their AGM algorithm of Remark 1.1 would yield a $(\log d)^{O(1)}$-space algorithm for decimal digits, but conceded that the time complexity would not be $d(\log d)^{O(1)}$. That still seems to be the case at the present time in June 2024.

ACKNOWLEDGMENTS

I thank my PhD student Dr. Wen-shin Lee for inviting me to give this lecture and paper. Her 2001 idea of keeping both the arguments to the Prony algorithm and the shift symbolic variables for computing the sparsest shift of a polynomial is unforgettable to me [16, 17].

Note added on July 27, 2024: At the end of the proof of Lemma 2.1: "The probability (1) is = $|W|/|S|^{n+m} \ldots$" replaces "The probability (1) is $\geq |W|/|S|^{n+m} \ldots$"

In Section 2.3, beginning of second paragraph: "for a nonsingular coefficient matrix $A \in \mathsf{K}^{n \times n}$" replaces "for a nonsingular matrix $A \in \mathsf{K}^{n \times n}$"

Note added on July 30, 2024: Replaced 4. on p. 5 in Section 2.3 by [9, Section 7].

REFERENCES

[1] A. Aho, J. Hopcroft, and J. Ullman. 1974. *The Design and Analysis of Algorithms.* Addison and Wesley, Reading, MA.

[2] D. Bailey, P. Borwein, and S. Plouffe. 1995. *On The Rapid Computation of Various Polylogarithmic Constants.* Preprint 95:056. Centre for Experimental and Constructive Mathematics, Simon Fraser University, http://mosaic.cecm.sfu.ca/preprints/1995pp.html.

[3] D. Bailey, P. Borwein, and S. Plouffe. 1997. On The Rapid Computation of Various Polylogarithmic Constants. *Math. Comput.* 66, 218 (1997), 903–913.

[4] Manjul Bhargava. 2021. Galois groups of random integer polynomials and van der Waerden's Conjecture. URL: https://arxiv.org/abs/2111.06507.

[5] Jonathan M. Borwein and Peter B. Borwein. 1987. *Pi and the AGM—A Study in Analytic Number Theory and Computational Complexity.* John Wiley & Sons, New York.

[6] L. Chen, W. Eberly, E. Kaltofen, B. D. Saunders, W. J. Turner, and G. Villard. 2002. Efficient Matrix Preconditioners for Black Box Linear Algebra. *Linear Algebra and Applications* 343–344 (2002), 119–146. Special issue on *Structured and Infinite Systems of Linear Equations*, edited by P. Dewilde, V. Olshevsky and A. H. Sayed. URL: http://users.cs.duke.edu/~elk27/bibliography/02/CEKSTV02.pdf.

[7] S. A. Cook. 1985. A taxonomy of problems with fast parallel algorithms. *Inf. Control* 64 (1985), 2–22.

[8] R. A. DeMillo and R. J. Lipton. 1978. A probabilistic remark on algebraic program testing. *Information Process. Letters* 7, 4 (1978), 193–195.

[9] Jean-Guillaume Dumas, Erich Kaltofen, Emmanuel Thomé, and Gilles Villard. 2016. Linear Time Interactive Certificates for the Minimal Polynomial and the Determinant of a Sparse Matrix. In *ISSAC'16 Proc. 2016 ACM Internat. Symp. Symbolic Algebraic Comput.*, Markus Rosenkranz (Ed.). Association for Computing Machinery, New York, N. Y., 199–206. URL: http://users.cs.duke.edu/~elk27/bibliography/16/DKTV16.pdf.

[10] H. R. P. Ferguson and R. W. Forcade. 1979. Generalization of the Euclidean Algorithm for Real Numbers to All Dimensions Higher than Two. *Bull. Amer. Math. Soc.* 1 (1979), 912–914.

[11] H. R. P. Ferguson and R. W. Forcade. 1982. Multidimensional Euclidean algorithms. *J. reine angew. Math.* 334 (1982), 171–181.

[12] Rūsiņš Freivalds. 1979. Fast Probabilistic Algorithms. In *Mathematical Foundations of Computer Science 1979, Proceedings, 8th Symposium, Olomouc, Czechoslovakia, September 3-7, 1979 (Lecture Notes in Computer Science, Vol. 74)*, Jirí Becvár (Ed.). Springer, 57–69. https://doi.org/10.1007/3-540-09526-8_5

[13] Shuhong Gao, Erich Kaltofen, John P. May, Zhengfeng Yang, and Lihong Zhi. 2004. Approximate factorization of multivariate polynomials via differential equations. In *ISSAC 2004 Proc. 2004 Internat. Symp. Symbolic Algebraic Comput.*, Jaime Gutierrez (Ed.). ACM Press, New York, N. Y., 167–174. ACM SIGSAM's ISSAC 2004 Distinguished Student Author Award (May and Yang). URL: http://users.cs.duke.edu/~elk27/bibliography/04/GKMYZ04.pdf.

[14] Joachim von zur Gathen and E. Kaltofen. 1983. Factoring multivariate polynomials over finite fields. In *Proc. 1983 ICALP (Lect. Notes Comput. Sci., Vol. 154).* Springer Verlag, Heidelberg, Germany, 250–263. Journal version in [15].

[15] Joachim von zur Gathen and E. Kaltofen. 1985. Factoring multivariate polynomials over finite fields. *Math. Comput.* 45 (1985), 251–261. URL: http://users.cs.duke.edu/~elk27/bibliography/85/GaKa85_mathcomp.pdf; AMS URL: https://www.ams.org/journals/mcom/1985-45-171/home.html.

[16] Mark Giesbrecht, Erich Kaltofen, and Wen-shin Lee. 2003. Algorithms for Computing Sparsest Shifts of Polynomials in Power, Chebychev, and Pochhammer Bases. *J. Symbolic Comput.* 36, 3–4 (2003), 401–424. Special issue Internat. Symp. Symbolic Algebraic Comput. (ISSAC 2002). Guest editors: M. Giusti & L. M. Pardo. URL: http://users.cs.duke.edu/~elk27/bibliography/03/GKL03.pdf.

[17] Mark Giesbrecht, Erich Kaltofen, and Wen-shin Lee. 2002. Algorithms for Computing the Sparsest Shifts for Polynomials via the Berlekamp/Massey Algorithm. In *Proc. 2002 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'02)*, T. Mora (Ed.). ACM Press, New York, N. Y., 101–108. Journal version in [16]. URL: http://users.cs.duke.edu/~elk27/bibliography/02/GKL02.pdf.

[18] Mark Giesbrecht, George Labahn, and Wen-shin Lee. 2004. Symbolic-Numeric Sparse Interpolation of Multivariate Polynomials (Extended Abstract). In *Proc. Ninth Rhine Workshop on Computer Algebra (RWCA'04), University of Nijmegen, the Netherlands*. 127–139.

[19] Mark Giesbrecht, George Labahn, and Wen-shin Lee. 2006. Symbolic-numeric sparse interpolation of multivariate polynomials. In *ISSAC MMVI Proc. 2006 Internat. Symp. Symbolic Algebraic Comput.*, Jean-Guillaume Dumas (Ed.). ACM Press, New York, N. Y., 116–123. https://doi.org/10.1145/1145768.1145792

[20] Mark Giesbrecht, George Labahn, and Wen-shin Lee. 2009. Symbolic-numeric sparse interpolation of multivariate polynomials. *J. Symbolic Comput.* 44 (2009), 943–959.

[21] Mark van Hoeij. 2002. Factoring polynomials and the knapsack problem. *J. Number Theory* 95 (2002), 167–189. Implementation available at http://web.math.fsu.edu/~hoeij/.

[22] Emma Haruka Iwao. 2022. Even more pi in the sky: Calculating 100 trillion digits of pi on Google Cloud. https://cloud.google.com/blog/products/compute/calculating-100-trillion-digits-of-pi-on-google-cloud.

[23] E. Kaltofen. 1979. An Attributed LL(1) Compilation of Pascal into Lambda-Calculus. Master's Project, Rensselaer Polytechnic Instit., Math. Sci. Dept., Troy, N. Y.. Also Tech. Report CS-8103, RPI, Math. Sci. Dept., by E. Kaltofen and S. K. Abdali, June 1981. URL: http://users.cs.duke.edu/~elk27/bibliography/79/Ka79_msproject_greek_lambda_corr.pdf, URL: http://users.cs.duke.edu/~elk27/bibliography/79/Ka79_msproject_appendix_A.pdf, URL: http://users.cs.duke.edu/~elk27/bibliography/79/Ka79_msproject_appendix_B.pdf.

[24] E. Kaltofen. 1982. A polynomial reduction from multivariate to bivariate integral polynomial factorization. In *Proc. 14th Annual ACM Symp. Theory Comput.* ACM, 261–266. Journal version in [29]. URL: http://users.cs.duke.edu/~elk27/bibliography/82/Ka82_stoc.pdf.

[25] E. Kaltofen. 1982. A polynomial-time reduction from bivariate to univariate integral polynomial factorization. In *Proc. 23rd Annual Symp. Foundations of Comp. Sci.* IEEE, 57–64. Journal version in [29]. URL: http://users.cs.duke.edu/~elk27/bibliography/82/Ka82_focs.pdf.

[26] E. Kaltofen. 1984. Effective Hilbert Irreducibility. In *Proc. EUROSAM '84 (Lect. Notes Comput. Sci.)*, J. Fitch (Ed.). Springer Verlag, Heidelberg, Germany, 275–284. Journal version in [27].

[27] E. Kaltofen. 1985. Effective Hilbert irreducibility. *Information and Control* 66 (1985), 123–137. URL: http://users.cs.duke.edu/~elk27/bibliography/85/Ka85_infcontr.pdf.

[28] E. Kaltofen. 1985. Fast parallel absolute irreducibility testing. *J. Symbolic Comput.* 1, 1 (1985), 57–67. Misprint corrections: *J. Symbolic Comput.* vol. 9, p. 320 (1989). URL: http://users.cs.duke.edu/~elk27/bibliography/85/Ka85_jsc.pdf.

[29] E. Kaltofen. 1985. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM J. Comput.* 14, 2 (1985), 469–489. URL: http://users.cs.duke.edu/~elk27/bibliography/85/Ka85_sicomp.pdf.

[30] E. Kaltofen. 1986. Uniform closure properties of p-computable functions. In *Proc. 18th Annual ACM Symp. Theory Comput.* ACM, 330–337. Also published as part of [31] and [32]. URL: http://users.cs.duke.edu/~elk27/bibliography/86/Ka86_stoc.pdf.

[31] E. Kaltofen. 1988. Greatest common divisors of polynomials given by straight-line programs. *J. ACM* 35, 1 (1988), 231–264. URL: http://users.cs.duke.edu/~elk27/bibliography/88/Ka88_jacm.pdf.

[32] E. Kaltofen. 1989. Factorization of polynomials given by straight-line programs. In *Randomness and Computation*, S. Micali (Ed.). Advances in Computing Research, Vol. 5. JAI Press Inc., Greenwhich, Connecticut, 375–412. URL: http://users.cs.duke.edu/~elk27/bibliography/89/Ka89_slpfac.pdf.

[33] E. Kaltofen. 1995. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Math. Comput.* 64, 210 (1995), 777–806. URL: http://users.cs.duke.edu/~elk27/bibliography/95/Ka95_mathcomp.pdf.

[34] E. Kaltofen. 2000. Challenges of Symbolic Computation My Favorite Open Problems. *J. Symbolic Comput.* 29, 6 (2000), 891–919. With an additional open problem by R. M. Corless and D. J. Jeffrey. URL: http://users.cs.duke.edu/~elk27/bibliography/2K/Ka2K.pdf.

[35] Erich L. Kaltofen. 2022. A note on the van der Waerden conjecture on random polynomials with symmetric Galois group for function fields. https://arxiv.org/abs/2204.02836. 6 pages. URL: http://users.cs.duke.edu/~elk27/bibliography/22/Ka22vdw.pdf.

[36] Erich Kaltofen and Grégoire Lecerf. 2013. Section 11.5. Factorization of multivariate polynomials. In *Handbook of Finite Fields*, Gary L. Mullen and Daniel Panario (Eds.). CRC Press, Taylor & Francis Group, Boca Raton, Florida, 382–392. URL: http://users.cs.duke.edu/~elk27/bibliography/11/KL11.pdf.

[37] Erich Kaltofen, John May, Zhengfeng Yang, and Lihong Zhi. 2008. Approximate Factorization of Multivariate Polynomials Using Singular Value Decomposition. *J. Symbolic Comput.* 43, 5 (2008), 359–376. URL: http://users.cs.duke.edu/~elk27/bibliography/07/KMYZ07.pdf.

[38] Erich L. Kaltofen, Michael Nehring, and B. David Saunders. 2011. Quadratic-Time Certificates in Linear Algebra. In *Proc. 2011 Internat. Symp. Symbolic Algebraic Comput. ISSAC 2011*, Anton Leykin (Ed.). Association for Computing Machinery, New York, N. Y., 171–176. URL: http://users.cs.duke.edu/~elk27/bibliography/11/KNS11.pdf.

[39] E. Kaltofen and V. Shoup. 1995. Subquadratic-time factoring of polynomials over finite fields. In *Proc. 27th Annual ACM Symp. Theory Comput.* ACM Press, New York, N.Y., 398–406. Journal version in [41]. URL: http://users.cs.duke.edu/~elk27/bibliography/95/KaSh95.pdf.

[40] E. Kaltofen and V. Shoup. 1997. Fast polynomial factorization over high algebraic extensions of finite fields. In *Proc. 1997 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'97)*, W. Küchlin (Ed.). ACM Press, New York, N. Y., 184–188. URL: http://users.cs.duke.edu/~elk27/bibliography/97/KaSh97.pdf.

[41] E. Kaltofen and V. Shoup. 1998. Subquadratic-time factoring of polynomials over finite fields. *Math. Comput.* 67, 223 (July 1998), 1179–1197. URL: http://users.cs.duke.edu/~elk27/bibliography/98/KaSh98.pdf.

[42] E. Kaltofen and B. Trager. 1988. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. In *Proc. 29th Annual Symp. Foundations of Comp. Sci.* IEEE, 296–305. Journal version in [43].

[43] Erich Kaltofen and Barry M. Trager. 1990. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *J. Symbolic Comput.* 9, 3 (1990), 301–320. URL: http://users.cs.duke.edu/~elk27/bibliography/90/KaTr90.pdf.

[44] Erich L. Kaltofen and Zhengfeng Yang. 2014. Sparse Multivariate Function Recovery With a High Error Rate in Evaluations. In *ISSAC 2014 Proc. 39th Internat. Symp. Symbolic Algebraic Comput.*, Katsusuke Nabeshima (Ed.). Association for Computing Machinery, New York, N. Y., 280–287. URL: http://users.cs.duke.edu/~elk27/bibliography/14/KaYa14.pdf.

[45] Erich L. Kaltofen and Zhengfeng Yang. 2016. Sparse Multivariate Function Recovery With a Small Number of Evaluations. *J. Symbolic Comput.* 75 (July/Aug. 2016), 209–218. Special Issue on ISSAC 2014, URL: http://users.cs.duke.edu/~elk27/bibliography/15/KaYa15_jsc.pdf.

[46] Erich Kaltofen and George Yuhasz. 2013. On The Matrix Berlekamp-Massey Algorithm. *ACM Trans. Algorithms* 9, 4 (Sept. 2013), 33:1–33:24. URL: http://users.cs.duke.edu/~elk27/bibliography/06/KaYu06.pdf.

[47] Tracy Kimbrel and Rakesh K. Sinha. 1993. A Probabilistic Algorithm for Verifying Matrix Products Using $O(n^2)$ Time and $\log_2 n + O(1)$ Random Bits. *Inf. Process. Lett.* 45, 2 (1993), 107–110.

[48] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász. 1982. *Factoring polynomials with rational coefficients.* Technical Report 82-05. Mathematisch Instituut, Roetersstraat 15 Amsterdam. Received March, 1982.

[49] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. 1982. Factoring polynomials with rational coefficients. *Math. Ann.* 261 (1982), 515–534.

[50] C. Andrew Neff. 1994. Specified Precision Polynomial Root Isolation Is in NC. *J. Comput. Syst. Sci.* 48, 3 (1994), 429–463. https://doi.org/10.1016/S0022-0000(05)80061-3

[51] Jacob T. Schwartz. 1979. Probabilistic algorithms for verification of polynomial identities (invited). In *Symbolic and Algebraic Computation (Lect. Notes Comput. Sci., Vol. 72)*. Springer Verlag, Heidelberg, Germany, 200–215. Proc. EUROSAM '79.

[52] J. T. Schwartz. 1980. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM* 27 (1980), 701–717.

[53] Avi Widgerson. 2019. *Mathematics and Computation.* Princeton University Press. https://www.math.ias.edu/avi/book.

[54] D. Wiedemann. 1986. Solving sparse linear equations over finite fields. *IEEE Trans. Inf. Theory* IT-32 (1986), 54–62.

[55] H. Zassenhaus. 1969. On Hensel factorization I. *J. Number Theory* 1 (1969), 291–311.

[56] Richard Zippel. 1979. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation (Lect. Notes Comput. Sci., Vol. 72)*. Springer Verlag, Heidelberg, Germany, 216–226. Proc. EUROSAM '79.

[57] R. E. Zippel. 1979. *Probabilistic algorithms for sparse polynomials.* Ph. D. Dissertation. Massachusetts Inst. of Technology, Cambridge, USA.