JOACHIM VON ZUR GATHEN (2001). Factorization of polynomials. In *Encyclopaedia of Mathematics, Supplement III*, M. HAZEWINKEL, editor, 146–147. Kluwer Academic Publishers. ISBN 1-4020-0198-3. URL http://dx.doi.org/10.1007/978-0-306-48373-8_6.

Factorization of polynomials, *factoring polynomials* – Since C.F. Gauss it is known that an arbitrary **polynomial** over a **field** or over the integers can be factored into irreducible factors, essentially uniquely (cf. also **Factorial ring**). For an efficient version of Gauss' theorem, one asks to find these factors algorithmically, and to devise such algorithms with low cost.

Based on a precocious uncomputability result in [13], one can construct sufficiently bizarre (but still 'computable') fields over which even square-freeness of polynomials is undecidable in the sense of A.M. Turing (cf. also **Undecidability**; **Turing machine**). But for the fields of practical interest, there are algorithms that perform extremely well, both in theory and practice. Of course, factorization of integers and thus also in $\mathbf{Z}[x]$ remains difficult; much of cryptography (cf. **Cryptology**) is based on the belief that it will remain so. The base case concerns factoring univariate polynomials over a **finite field** $\mathbf{F}_q$ with q elements, where q is a prime power. A first step is to make the input polynomial $f \in \mathbf{F}_q[x]$, of degree n, square-free. This is easy to do by computing $\gcd(f, \partial f/\partial x)$ and possibly extracting pth roots, where $p = \operatorname{char} \mathbf{F}_q$. The main tool of all algorithms is the **Frobenius automorphism** $\sigma\colon R \to R$ on the $\mathbf{F}_q$-algebra $R = \mathbf{F}_q[x]/(f)$. The pioneering algorithms are due to E.R. Berlekamp [1], [2], who represents σ by its matrix on the basis $1, x, x^2, \ldots, x^{n-1} \bmod f$ of R. A second approach, due to D.G. Cantor and H. Zassenhaus [3], is to compute σ by repeated squaring. A third method (J. von zur Gathen and V. Shoup [7]) uses the so-called polynomial representation of σ as its basic tool. The last two algorithms are based on Gauss' theorem that $x^{q^d} - x$ is the product of all monic irreducible polynomials in $\mathbf{F}_q[x]$ whose degree divides d. Thus, $f_1 = \gcd(x^q - x, f)$ is the product of all linear factors of f; next, $f_2 = \gcd(x^{q^2} - x, f/f_1)$ consists of all quadratic factors, and so on. This yields the *distinct-degree factorization* $(f_1, f_2, \ldots)$ of f.

The *equal-degree factorization* step splits any of the resulting factors f_i. This is only necessary if $\deg f_i > i$. Since all irreducible factors of f_i have degree i, the algebra $R_i = \mathbf{F}_q[x]/(f_i)$ is a direct product of (at least two) copies of $\mathbf{F}_{q^i}$. A random element a of R_i is likely to have **Legendre symbol** $+1$ in some and -1 in other copies; then $\gcd\left(a^{(q^i-1)/2} - 1, f_i\right)$ is a non-trivial factor of f_i. To describe the cost of these methods, one uses fast arithmetic, so that polynomials over $\mathbf{F}_q$ of degree up to n can be multiplied with $O(n \log n \log\log n)$ operations in $\mathbf{F}_q$, or $O^\sim(n)$ for short, where the so-called 'soft O' hides factors that are logarithmic in n. Furthermore, ω is an exponent for matrix multiplication, with the current (in 2000) world record $\omega < 2.376$, from [4].

All algorithms first compute x^q modulo f, with $O^\sim(n \log q)$ operations in $\mathbf{F}_q$. One can show that in an appropriate model, $\Omega(\log q)$ operations are necessary, even for $n = 2$. The further cost is as follows:

	$\mathbf{F}_q[x]/(f)$ as	Cost in $O^\sim$
Berlekamp	$\mathbf{F}_q$-vector space	n^ω
Cantor–Zassenhaus	multiplicative semi-group	$n^2 \log q$
von zur Gathen–Shoup	$\mathbf{F}_q$-algebra	n^2

Table 1

For small fields, even better algorithms exist.

The next problem is factorization of some $f \in \mathbf{Q}[x]$. The central tool is Hensel lifting, which lifts a factorization of f modulo an appropriate prime number p to

C.F. Gauss
polynomial
field
Factorial ring
Gauss' theorem→ Gauss factorization theorem
undecidable→ Turing undecidability
A.M. Turing
Undecidability
Turing machine
cryptography
Cryptology
finite field
Frobenius automorphism
E.R. Berlekamp
D.G. Cantor
H. Zassenhaus
J. von zur Gathen
V. Shoup
polynomial representation→ polynomial representation of the Frobenius automorphism
Gauss' theorem→ Gauss theorem on products of monic irreducible polynomials
distinct-degree factorization
equal-degree factorization
Legendre symbol
'soft O'→ soft O
Hensel lifting

one modulo a large power p^k of p. Irreducible factors of f will usually factor modulo p, according to the **Chebotarev density theorem**. One can then try various factor combinations of the irreducible factors modulo p^k to recover a true factor in $\mathbf{Q}[x]$. This works quite well in practice, at least for polynomials of moderate degree, but uses exponential time on some inputs (for example, on the Swinnerton-Dyer polynomials). In a celebrated paper, A.K. Lenstra, H.W. Lenstra, Jr. and L. Lovász [11] introduced basis reduction of integer lattices (cf. **LLL basis reduction method**), and applied this to obtain a polynomial-time algorithm. Their reduction method has since found many applications, notably in cryptanalysis (cf. also **Cryptology**). A method in [9] promises an even faster factorizing method.

The next tasks are bivariate polynomials. It can be solved in a similar fashion, with Hensel lifting, say, modulo one variable, and an appropriate version of basis reduction, which is easy in this case. Algebraic extensions of the ground field are handled similarly.

Multivariate polynomials pose a new type of problem: how to represent them? The *dense representation*, where each term up to the degree is written out, is often too long. One would like to work with the *sparse representation*, using only the non-zero coefficients. The methods discussed above can be adapted and work reasonably well on many examples, but no guarantees of polynomial time are given. Two new ingredients are required. The first are efficient versions (due to E. Kaltofen and von zur Gathen) of Hilbert's irreducibility theorem (cf. also **Hilbert theorem**). These versions say that if one reduces many to two variables with a certain type of random linear substitution, then each irreducible factor is very likely to remain irreducible. The second ingredient is an even more concise representation, namely by a *black box* which returns the polynomial's value at any requested point. A highlight of this theory is the random polynomial-time factorization method in [10].

Each major computer algebra system has some variant of these methods implemented. Special-purpose software can factor huge polynomials, for example of degree more than one million over $\mathbf{F}_2$. Several textbooks describe the details of some of these methods, e.g. [8], [12], [5], [14].

Factorization of polynomials modulo a composite number presents some surprises, such as the possibility of exponentially many irreducible factors, which can nevertheless be determined in polynomial time, in an appropriate data structure,; see [6].

For a historical perspective, note that the basic idea of equal-degree factorization was known to A.M. Legendre, while Gauss had found, around 1798, the distinct-degree factorization algorithm and Hensel lifting. They were to form part of the eighth chapter of his 'Disquisitiones Arithmeticae', but only seven got published, due to lack of funding.

References

[1] Berlekamp, E.R.: 'Factoring polynomials over finite fields', *Bell System Techn. J.* **46** (1967), 1853–1859.

[2] Berlekamp, E.R.: 'Factoring polynomials over large finite fields', *Math. Comput.* **24**, no. 11 (1970), 713–735.

[3] Cantor, D.G., and Zassenhaus, H.: 'A new algorithm for factoring polynomials over finite fields', *Math. Comput.* **36**, no. 154 (1981), 587–592.

[4] Coppersmith, D., and Winograd, S.: 'Matrix multiplication via arithmetic progressions', *J. Symbolic Comput.* **9** (1990), 251–280.

[5] Gathen, J. von zur, and Gerhard, J.: *Modern computer algebra*, Cambridge Univ. Press, 1999.

[6] Gathen, J. von zur, and Hartlieb, S.: 'Factoring modular polynomials', *J. Symbolic Comput.* **26**, no. 5 (1998), 583–606.

[7] Gathen, J. von zur, and Shoup, V.: 'Computing Frobenius maps and factoring polynomials', *Comput. Complexity* **2** (1992), 187–224.

[8] Geddes, K.O., Czapor, S.R., and Labahn, G.: *Algorithms for computer algebra*, Kluwer Acad. Publ., 1992.

[9] Hoeij, M. van: 'Factoring polynomials and the knapsack problem', *www.math.fsu.edu/~hoeij/knapsack/paper/knapsack.ps* (2000).

[10] Kaltofen, E., and Trager, B.M.: 'Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators', *J. Symbolic Comput.* **9** (1990), 301–320.

Chebotarev density theorem
Swinnerton-Dyer polynomials→ Swinnerton-Dyer polynomial
A.K. Lenstra
H.W. Lenstra, Jr.
L. Lovász
LLL basis reduction method
Cryptology
Hensel lifting→ Hensel lifting
dense representation→ dense representation of a multivariate polynomial
sparse representation→ sparse representation of a multivariate polynomial
E. Kaltofen
Hilbert's irreducibility theorem→ Hilbert irreducibility theorem
Hilbert theorem
black box→ black box representation of a multivariate polynomial
random polynomial-time factorization method→ Kaltofen–Trager random polynomial-time factorization method
A.M. Legendre

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

[12] Shparlinski, I.E.: *Finite fields: theory and computation*, Kluwer Acad. Publ., 1999.

[13] Waerden, B.L. van der: 'Eine Bemerkung über die Unzerlegbarkeit von Polynomen', *Math. Ann.* **102** (1930), 738–739.

[14] Yap, Chee Keng: *Fundamental problems of algorithmic algebra*, Oxford Univ. Press, 2000.

Joachim von zur Gathen
E-mail address: gathen@upb.de

MSC1991: 12D05