





WIKIPEDIA
The Free Encyclopedia

[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikipedia store](#)

Interaction
[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

Tools
[What links here](#)
[Related changes](#)
[Upload file](#)
[Special pages](#)
[Permanent link](#)
[Page information](#)
[Wikidata item](#)
[Cite this page](#)


Print/export
[Create a book](#)
[Download as PDF](#)
[Printable version](#)

Languages 
[فارسی](#)
[Français](#)
[Polski](#)
 [Edit links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)



Cantor–Zassenhaus algorithm

From Wikipedia, the free encyclopedia

In [computational algebra](#), the **Cantor–Zassenhaus algorithm** is a well known method for factorising [polynomials](#) over [finite fields](#) (also called Galois fields).

The algorithm consists mainly of exponentiation and polynomial [GCD](#) computations. It was invented by [David G. Cantor](#) and [Hans Zassenhaus](#) in 1981.

It is arguably the dominant algorithm for solving the problem, having replaced the earlier [Berlekamp's algorithm](#) of 1967. It is currently implemented in many well-known [computer algebra systems](#).

Contents [hide]

- 1 Overview
 - 1.1 Background
 - 1.2 Core result
 - 1.3 Algorithm
- 2 Applications
- 3 Implementation in computer algebra systems
- 4 See also
- 5 References

Overview [\[edit\]](#)

Background [\[edit\]](#)

The Cantor–Zassenhaus algorithm takes as input a squarefree polynomial $f(x)$ (i.e. one with no repeated factors) of degree n with coefficients in a finite field \mathbb{F}_q whose [irreducible polynomial](#) factors are all of equal degree (algorithms exist for efficiently factorising arbitrary polynomials into a product of polynomials satisfying these conditions, so that the Cantor–Zassenhaus algorithm can be used to factorise arbitrary polynomials). It gives as output a polynomial $g(x)$ with coefficients in the same field such that $g(x)$ divides $f(x)$. The algorithm may then be applied recursively to these and subsequent divisors, until we find the decomposition of $f(x)$ into powers of irreducible polynomials (recalling that the [ring](#) of polynomials over any field is a [unique factorisation domain](#)).

All possible factors of $f(x)$ are contained within the [factor ring](#) $R = \frac{\mathbb{F}_q[x]}{\langle f(x) \rangle}$. If we suppose that $f(x)$ has irreducible factors $p_1(x), p_2(x), \dots, p_s(x)$, all of degree d , then this factor ring is isomorphic to the [direct product](#) of factor rings $S = \prod_{i=1}^s \frac{\mathbb{F}_q[x]}{\langle p_i(x) \rangle}$. The isomorphism from R to S , say ϕ , maps a polynomial

$g(x) \in R$ to the s -tuple of its reductions modulo each of the $p_i(x)$, i.e. if:

$$g(x) \equiv g_1(x) \pmod{p_1(x)},$$

$$g(x) \equiv g_2(x) \pmod{p_2(x)},$$

\vdots

$$g(x) \equiv g_s(x) \pmod{p_s(x)},$$

then $\phi(g(x) + \langle f(x) \rangle) = (g_1(x) + \langle p_1(x) \rangle, \dots, g_s(x) + \langle p_s(x) \rangle)$. It is important to note the following at this point, as it shall be of critical importance later in the algorithm: Since the $p_i(x)$ are each irreducible, each of the factor rings in this direct sum is in fact a field. These fields each have degree q^d .

Core result [\[edit\]](#)

The core result underlying the Cantor–Zassenhaus algorithm is the following: If $a(x) \in R$ is a polynomial satisfying:

$$a(x) \neq 0, \pm 1$$

$$a_i(x) \in \{0, -1, 1\} \text{ for } i = 1, 2, \dots, s,$$

where $a_i(x)$ is the reduction of $a(x)$ modulo $p_i(x)$ as before, and if any two of the following three sets is non-empty:

$$A = \{i | a_i(x) = 0\},$$

$$B = \{i | a_i(x) = -1\},$$

$$C = \{i | a_i(x) = 1\},$$

then there exist the following non-trivial factors of $f(x)$:

$$\gcd(f(x), a(x)) = \prod_{i \in A} p_i(x),$$

$$\gcd(f(x), a(x) + 1) = \prod_{i \in B} p_i(x),$$

$$\gcd(f(x), a(x) - 1) = \prod_{i \in C} p_i(x).$$

Algorithm [\[edit\]](#)

The Cantor–Zassenhaus algorithm computes polynomials of the same type as $a(x)$ above using the isomorphism discussed in the Background section. It proceeds as follows, in the case where the field \mathbb{F}_q is of odd-characteristic. The process can be generalised to characteristic 2 fields in a fairly straightforward way: Select a random polynomial $b(x) \in R$ such that $b(x) \neq 0, \pm 1$. Set $m = (q^d - 1)/2$ and compute $b(x)^m$. Since ϕ is an isomorphism, we have (using our now-established notation):

$$\phi(b(x)^m) = (b_1^m(x) + \langle p_1(x) \rangle, \dots, b_s^m(x) + \langle p_s(x) \rangle).$$

Now, each $b_i(x) + \langle p_i(x) \rangle$ is an element of a field of order q^d , as noted earlier. The multiplicative subgroup of this field has order $q^d - 1$ and so, unless $b_i(x) = 0$, we have $b_i(x)^{q^d-1} = 1$ for each i and hence $b_i(x)^m = \pm 1$ for each i . If $b_i(x) = 0$, then of course $b_i(x)^m = 0$. Hence $b(x)^m$ is a polynomial of the same type as $a(x)$ above. Further, since $b(x) \neq 0, \pm 1$, at least two of the sets A, B and C are non-empty and by computing the above GCDs we may obtain non-trivial factors. Since the ring of polynomials over a field is an [Euclidean domain](#), we may compute these GCDs using the [Euclidean algorithm](#).

Applications [\[edit\]](#)

One important application of the Cantor–Zassenhaus algorithm is in computing [discrete logarithms](#) over finite fields of prime-power order. Computing discrete logarithms is an important problem in [public key cryptography](#). For a field of prime-power order, the fastest known method is the [index calculus method](#), which involves the factorisation of field elements. If we represent the prime-power order field in the usual way – that is, as polynomials over the prime order base field, reduced modulo an irreducible polynomial of appropriate degree – then this is simply polynomial factorisation, as provided by the Cantor–Zassenhaus algorithm.

Implementation in computer algebra systems [\[edit\]](#)

The Cantor–Zassenhaus algorithm may be accessed in the PARI/GP package using the [factorcantor](#) [↗](#) command.

See also [\[edit\]](#)

- [Polynomial factorisation](#)
- [Factorization of polynomials over a finite field and irreducibility tests](#)

References [\[edit\]](#)

- [Cantor, David G.; Zassenhaus, Hans](#) (April 1981), "A new algorithm for factoring polynomials over finite fields", *Mathematics of Computation* **36** (154): 587–592, doi:10.1090/S0025-5718-1981-0606517-5, JSTOR 2007663 [↗](#), MR 606517 [↗](#)

This page was last modified on 19 December 2014, at 03:30.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

