# Number fields in C

B. Allombert

Institut de Mathématiques de Bordeaux
Univ. Bordeaux, CNRS, INRIA

24/06/2025

# Accessors for nf

- `GEN nf_get_pol(GEN nf)`: `nf.pol`.
- `long nf_get_varn(GEN nf)`: variable number of `nf.pol`.
- `long nf_get_degree(GEN nf)`: degree of `nf.pol`.
- `GEN nf_get_roots(GEN nf)` return `nf.roots`.
- `GEN nf_get_allroots(GEN nf)` also return the conjugates.
- `void nf_get_sign(GEN nf, long *r1, long *r2)`: set `r1`, `r2` to the signature.
- `long nf_get_r1(GEN nf)`: return `nf.r1`.
- `long nf_get_r2(GEN nf)`: return `nf.r2`.
- `GEN nf_get_disc(GEN nf)` return `nf.disc`.
- `GEN nf_get_index(GEN nf)` return `nf.index`.
- `GEN nf_get_zk(GEN nf)` return `nf.zk`.
- `GEN nf_get_invzk(GEN nf)` return $\texttt{nf.zk}^{-1}$.

# Operations on nf elements

The GP function `nfeltadd`, `nfeltmul`, etc. are named `nfadd`, `nfmul` etc. in C.

- `nf_to_scalar_or_alg`: convert a `nfelt` to either a `t_INT` or a `t_POL`.
- `nf_to_scalar_or_basis`: convert a `nfelt` to either a `t_INT` or a `t_COL`. `nfnewprec_shallow(GEN nf, long prec)` return a shallow copy of `nf` updated to precision `prec`.

# Linear algebra over number fields

- `GEN nfM_det(GEN nf, GEN M)`: $\det(M)$
- `GEN nfM_inv(GEN nf, GEN M)`: $M^{-1}$
- `GEN nfM_ker(GEN nf, GEN M)`: $\ker(M)$

# idealprimedec

- `GEN idealprimedec(GEN nf, GEN p)`: as in GP
- `GEN idealprimedec_degrees(GEN nf, GEN p)`: return the list of residual degree as a `t_VECSMALL`.
- `GEN idealprimedec_galois(GEN nf, GEN p)`: return a single prime ideal.
- `GEN idealprimedec_limit_f(GEN nf, GEN p, long f)`: only return the prime ideals of residual degree at most $f$.
- `GEN idealprimedec_limit_norm(GEN nf, GEN p, GEN B)`: only return the prime ideals of norm at most $B$.

# idealfactor

- `GEN idealfactor (GEN nf, GEN x)` as in GP
- `GEN idealfactor_limit(GEN nf, GEN x, ulong lim)` only return the prime ideals above rational primes $<$ `lim`.
- `GEN idealfactor_partial(GEN nf, GEN x, GEN L)` only return the prime ideals above rational primes in $L$.

# Format of prime ideals

- `GEN pr_get_p(GEN pr)` return `pr.p`
- `GEN pr_get_gen(GEN pr)` return `pr.gen`
- `long pr_get_e(GEN pr)` return `pr.e`
- `long pr_get_f(GEN pr)` return `pr.f`
- `int pr_is_inert(GEN pr)` test whether `pr` is inert
- `GEN pr_norm(GEN pr)` return the norm of `pr`
- `ulong upr_norm(GEN pr)` return the norm of `pr` assuming it fit in a `ulong`.

# Accessors for bnf

- `GEN bnf_get_nf(GEN bnf)`: `bnf.nf`
- `GEN bnf_get_disc(GEN bnf)`: `bnf.disc`
- `GEN bnf_get_clgp(GEN bnf)`: `bnf.clgp`
- `GEN bnf_get_cyc(GEN bnf)`: `bnf.cyc`
- `GEN bnf_get_no(GEN bnf)`: `bnf.no`
- `GEN bnf_get_gen(GEN bnf)`: `bnf.gen`
- `GEN bnf_get_reg(GEN bnf)`: `bnf.reg`
- `GEN bnf_get_fu(GEN bnf)`: `bnf.fu`
- `GEN bnf_has_fu(GEN bnf)`: either `NULL` or `bnf.fu`.
- `GEN bnf_get_logfu(GEN bnf)`: return the logarithmic embedding of the fundamental units.
- `GEN bnf_compactfu(GEN bnf)`: return compact units if available.