

Algebraic Number Theory

(PARI-GP version 2.15.2)

Binary Quadratic Forms

create $ax^2 + bxy + cy^2$ **Qfb**(a, b, c) or **Qfb**($[a, b, c]$)
reduce x ($s = \sqrt{D}$, $l = \lfloor s \rfloor$) **qfbred**($x, \{flag\}, \{D\}, \{l\}, \{s\}$)
return $[y, g]$, $g \in \text{SL}_2(\mathbf{Z})$, $y = g \cdot x$ reduced **qfbreds12**(x)
composition of forms $x*y$ or **qfbnucomp**(x, y, l)
 n -th power of form x^n or **qfbnpow**(x, n)
composition **qfbcomp**(x, y)
... without reduction **qfbcomppraw**(x, y)
 n -th power **qfbpow**(x, n)
... without reduction **qfbpowraw**(x, n)
prime form of disc. x above prime p **qfbprimeform**(x, p)
class number of disc. x **qfbclassno**(x)
Hurwitz class number of disc. x **qfbhclassno**(x)
solve $Q(x, y) = n$ in integers **qfbsolve**(Q, n)
solve $x^2 + Dy^2 = p$, p prime **qfbcornacchia**(D, p)
... $x^2 + Dy^2 = 4p$, p prime **qfbcornacchia**($D, 4 * p$)

Quadratic Fields

quadratic number $\omega = \sqrt{x}$ or $(1 + \sqrt{x})/2$ **quadgen**(x)
minimal polynomial of ω **quadpoly**(x)
discriminant of **Q**(\sqrt{x}) **quaddisc**(x)
regulator of real quadratic field **quadregulator**(x)
fundamental unit in O_D , $D > 0$ **quadunit**($D, \{ 'w \}$)
norm of fundamental unit in O_D **quadunitnorm**(D)
index of $O_{Df_2}^\times$ in O_D^\times **quadunitindex**(D, f)
class group of **Q**(\sqrt{D}) **quadclassunit**($D, \{flag\}, \{t\}$)
Hilbert class field of **Q**(\sqrt{D}) **quadhilbert**($D, \{flag\}$)
... using specific class invariant ($D < 0$) **polclass**($D, \{inv\}$)
ray class field modulo f of **Q**(\sqrt{D}) **quadray**($D, f, \{flag\}$)

General Number Fields: Initializations

The number field $K = \mathbf{Q}[X]/(f)$ is given by irreducible $f \in \mathbf{Q}[X]$. We denote $\theta = \bar{X}$ the canonical root of f in K . A *nf* structure contains a maximal order and allows operations on elements and ideals. A *bnf* adds class group and units. A *bnr* is attached to ray class groups and class field theory. A *rnf* is attached to relative extensions L/K .

init number field structure *nf* **nfinit**($f, \{flag\}$)
 known integer basis B **nfinit**($[f, B]$)
 order maximal at $vp = [p_1, \dots, p_k]$ **nfinit**($[f, vp]$)
 order maximal at all $p \leq P$ **nfinit**($[f, P]$)
 certify maximal order **nfcertify**(*nf*)

nf members:

a monic $F \in \mathbf{Z}[X]$ defining K **nf.pol**
number of real/complex places **nf.r1/r2/sign**
discriminant of *nf* **nf.disc**
primes ramified in *nf* **nf.p**
 T_2 matrix **nf.t2**
complex roots of F **nf.roots**
integral basis of \mathbf{Z}_K as powers of θ **nf.zk**
different/codifferent **nf.diff**, **nf.codiff**
index $[\mathbf{Z}_K : \mathbf{Z}[X]/(F)]$ **nf.index**
recompute *nf* using current precision **nfnewprec**(*nf*)
init relative *rnf* $L = K[Y]/(g)$ **rnfinit**(*nf*, g)
init *bnf* structure **bnfinit**($f, 1$)

bnf members: same as *nf*, plus
 underlying *nf* **bnf.nf**
 class group, regulator **bnf.clgp**, **bnf.reg**
 fundamental/torsion units **bnf.fu**, **bnf.tu**
 add S -class group and units, yield *bnfS* **bnfsunit**(*bnf*, S)
 init class field structure *bnr* **bnrinit**(*bnf*, $m, \{flag\}$)
bnr members: same as *bnf*, plus
 underlying *bnf* **bnr.bnf**
 big ideal structure **bnr.bid**
 modulus m **bnr.mod**
 structure of $(\mathbf{Z}_K/m)^*$ **bnr.zkst**

Fields, subfields, embeddings

Defining polynomials, embeddings
(some) number fields with Galois group G **nflist**(G)
... and $|\text{disc}(K)| = N$ and s complex places **nflist**($G, N, \{s\}$)
... and $a \leq |\text{disc}(K)| \leq b$ **nflist**($G, [a, b], \{s\}$)
smallest poly defining $f = 0$ (slow) **polredabs**($f, \{flag\}$)
small poly defining $f = 0$ (fast) **polredbest**($f, \{flag\}$)
monic integral $g = Cf(x/L)$ **poltomonic**($f, \{\&L\}$)
random Tschirnhausen transform of f **poltschirnhaus**(f)
Q[t]/(f) \subset **Q**[t]/(g) ? Isomorphic? **nfisincl**(f, g), **nfisisom**
reverse polmod $a = A(t) \bmod T(t)$ **modreverse**(a)
compositum of **Q**[t]/(f), **Q**[t]/(g) **polcompositum**($f, g, \{flag\}$)
compositum of $K[t]/(f)$, $K[t]/(g)$ **nfcompositum**(*nf*, $f, g, \{flag\}$)
splitting field of K (degree divides d) **nfsplitting**(*nf*, $\{d\}$)
signs of real embeddings of x **nfeltsign**(*nf*, $x, \{pl\}$)
complex embeddings of x **nfeltembed**(*nf*, $x, \{pl\}$)
 $T \in K[t]$, # of real roots of $\sigma(T) \in R[t]$ **nfpolsturm**(*nf*, $T, \{pl\}$)

Subfields, polynomial factorization

subfields (of degree d) of *nf* **nfsubfields**(*nf*, $\{d\}$)
maximal subfields of *nf* **nfsubfieldsmax**(*nf*)
maximal CM subfield of *nf* **nfsubfieldscm**(*nf*)
 $K_d \subset \mathbf{Q}(\zeta_n)$, using Gaussian periods **polsubcyclo**($n, d, \{v\}$)
... using class field theory **polsubcyclofast**(n, d)
roots of unity in *nf* **nfrootsof1**(*nf*)
roots of g belonging to *nf* **nfroots**(*nf*, g)
factor g in *nf* **nffactor**(*nf*, g)

Linear and algebraic relations

poly of degree $\leq k$ with root $x \in \mathbf{C}$ or \mathbf{Q}_p **algdep**(x, k)
alg. dep. with pol. coeffs for series s **seralgdep**(s, x, y)
diff. dep. with pol. coeffs for series s **serdiffdep**(s, x, y)
small linear rel. on coords of vector x **lindep**(x)

Basic Number Field Arithmetic (nf)

Number field elements are **t_INT**, **t_FRAC**, **t_POL**, **t_POLMOD**, or **t_COL** (on integral basis *nf.zk*).

Basic operations

$x + y$ **nfeltadd**(*nf*, x, y)
 $x \times y$ **nfeltmul**(*nf*, x, y)
 x^n , $n \in \mathbf{Z}$ **nfeltpow**(*nf*, x, n)
 x/y **nfeltdiv**(*nf*, x, y)
 $q = x \backslash y := \text{round}(x/y)$ **nfeltdiveuc**(*nf*, x, y)
 $r = x/y := x - (x \backslash y)y$ **nfeltmod**(*nf*, x, y)
... $[q, r]$ as above **nfeltdivrem**(*nf*, x, y)
reduce x modulo ideal A **nfeltreduce**(*nf*, x, A)
absolute trace $\text{Tr}_{K/\mathbf{Q}}(x)$ **nfelttrace**(*nf*, x)
absolute norm $N_{K/\mathbf{Q}}(x)$ **nfeltnorm**(*nf*, x)

is x a square? **nfeltissquare**(*nf*, $x, \{\&y\}$)
... an n -th power? **nfeltispower**(*nf*, $x, n, \{\&y\}$)

Multiplicative structure of K^* ; $K^*/(K^*)^n$
valuation $v_{\mathfrak{p}}(x)$ **nfeltval**(*nf*, x, \mathfrak{p})
... write $x = \pi^{v_{\mathfrak{p}}(x)}y$ **nfeltval**(*nf*, $x, \mathfrak{p}, \&y$)
quadratic Hilbert symbol (at \mathfrak{p}) **nfhilbert**(*nf*, $a, b, \{\mathfrak{p}\}$)
 b such that $xb^n = v$ is small **idealredmodpower**(*nf*, x, n)

Maximal order and discriminant

integral basis of field **Q**[x]/(f) **nfbasis**(f)
field discriminant of **Q**[x]/(f) **nfdisc**(f)
... and factorization **nfdiscfactors**(f)
express x on integer basis **nfalgtobasis**(*nf*, x)
express element x as a polmod **nfbasistoalg**(*nf*, x)

Hecke Grossencharacters

Let K be a number field and m a modulus. A *gchar* structure describes the group of Hecke Grossencharacters of K of modulus m and allows computations with these characters. A character χ is described by its components modulo *gc.cyc*.

init *gchar* structure *gc* for modulus m **gcharinit**(*bnf*, $m, \{cm\}$)
gc members:

 underlying *bnf* **gc.bnf**
 modulus **gc.mod**
 elementary divisors (including 0s) **gc.cyc**
recompute *gc* using current precision **gcharnewprec**(*gc*)
evaluate Hecke character *chi* at ideal *id* **gchareval**(*gc*, *chi*, *id*)
exponent column of *id* in \mathbf{R}^n **gcharideallog**(*gc*, *id*)
log representation of ideal *id* **gcharlog**(*gc*, *id*)
... of character χ **gcharduallog**(*gc*, *chi*)
exponent vector of χ in \mathbf{R}^n **gcharparameters**(*gc*, *chi*)
conductor of χ **gcharconductor**(*gc*, *chi*)
L-function of χ **lfuncreate**(*gc*, *chi*)
local component χ_v of χ **gcharlocal**(*gc*, *chi*, v)
 χ s.t. $\chi_v \approx L_{chiv}[i]$ for $v = Lv[L_{chiv}]$ **gcharidentify**(*gc*, Lv, L_{chiv})
basis of group of algebraic characters **gcharalgebraic**(*gc*)
is χ algebraic? **gcharisalgebraic**(*gc*, *chi*)

Dedekind Zeta Function ζ_K , Hecke L series

$R = [c, w, h]$ in initialization means we restrict $s \in \mathbf{C}$ to domain $|\Re(s) - c| < w$, $|\Im(s)| < h$; $R = [w, h]$ encodes $[1/2, w, h]$ and $[h]$ encodes $R = [1/2, 0, h]$ (critical line up to height h).

ζ_K as Dirichlet series, $N(I) \leq b$ **dirzetak**(*nf*, b)
init $\zeta_K^{(k)}(s)$ for $k \leq n$ **L = lfunitinit**(*bnf*, $R, \{n = 0\}$)
compute $\zeta_K(s)$ (n -th derivative) **lfun**($L, s, \{n = 0\}$)
compute $\Lambda_K(s)$ (n -th derivative) **lfunlambda**($L, s, \{n = 0\}$)

init $L_K^{(k)}(s, \chi)$ for $k \leq n$ **L = lfunitinit**($[bnr, chi], R, \{n = 0\}$)
compute $L_K(s, \chi)$ (n -th derivative) **lfun**($L, s, \{n\}$)
Artin root number of K **bnrrootnumber**(*bnr*, *chi*, $\{flag\}$)
 $L(1, \chi)$, for all χ trivial on H **bnrL1**(*bnr*, $\{H\}, \{flag\}$)

Class Groups & Units (bnf, bnr)

Class field theory data $a_1, \{a_2\}$ is usually *bnr* (ray class field), *bnr*, H (congruence subgroup) or *bnr*, χ (character on **bnr.clgp**). Any of these define a unique abelian extension of K .
units / S -units **bnfunits**(*bnf*, $\{S\}$)
remove GRH assumption from *bnf* **bnfcertify**(*bnf*)

expo. of ideal x on class gp	<code>bnfisprincipal(<i>bnf</i>, x, {<i>flag</i>})</code>
... on ray class gp	<code>bnrisprincipal(<i>bnr</i>, x, {<i>flag</i>})</code>
expo. of x on fund. units	<code>bnfisunit(<i>bnf</i>, x)</code>
... on S -units, U is <code>bnfunits(<i>bnf</i>, S)</code>	<code>bnfisunit(<i>bnfs</i>, x, U)</code>
signs of real embeddings of <i>bnf</i> .fu	<code>bnfsignunit(<i>bnf</i>)</code>
narrow class group	<code>bnfnarrow(<i>bnf</i>)</code>

Class Field Theory

ray class number for modulus m	<code>bnrclassno(<i>bnf</i>, m)</code>
discriminant of class field	<code>bnrdisc(a_1, {a_2})</code>
ray class numbers, l list of moduli	<code>bnrclassnolist(<i>bnf</i>, l)</code>
discriminants of class fields	<code>bnrdisclist(<i>bnf</i>, l, {<i>arch</i>}, {<i>flag</i>})</code>
decode output from <code>bnrdisclist</code>	<code>bnfdecodemodule(<i>nf</i>, fa)</code>
is modulus the conductor?	<code>bnrisconductor(a_1, {a_2})</code>
is class field (<i>bnr</i> , H) Galois over K^G	<code>bnrisgalois(<i>bnr</i>, G, H)</code>
action of automorphism on <code>bnr.gen</code>	<code>bnrgaloismatrix(<i>bnr</i>, aut)</code>
apply <code>bnrgaloismatrix</code> M to H	<code>bnrgaloisapply(<i>bnr</i>, M, H)</code>
characters on <code>bnr.clgp</code> s.t. $\chi(g_i) = e(v_i)$	<code>bnrchar(<i>bnr</i>, g, {v})</code>
conductor of character χ	<code>bnrconductor(<i>bnr</i>, chi)</code>
conductor of extension	<code>bnrconductor(a_1, {a_2}, {<i>flag</i>})</code>
conductor of extension $K[Y]/(g)$	<code>rnfconductor(<i>bnf</i>, g)</code>
canonical projection $\text{Cl}_F \rightarrow \text{Cl}_f$, $f \mid F$	<code>bnrmap</code>
Artin group of extension $K[Y]/(g)$	<code>rnfnormgroup(<i>bnr</i>, g)</code>
subgroups of <i>bnr</i> , index $\leq b$	<code>subgrouplist(<i>bnr</i>, b, {<i>flag</i>})</code>
compositum as [<code>bnr</code> , <code>H</code>]	<code>bnrcompositum([<i>bnr</i>1, H1], [<i>bnr</i>2, H2])</code>
class field defined by $H \subset \text{Cl}_f$	<code>bnrclassfield(<i>bnr</i>, H)</code>
... low level equivalent, prime degree	<code>rnfkummer(<i>bnr</i>, H)</code>
same, using Stark units (real field)	<code>bnrstark(<i>bnr</i>, sub, {<i>flag</i>})</code>
is a an n -th power in K_v ?	<code>nfislocalpower(<i>nf</i>, v, a, n)</code>
cyclic L/K satisf. local conditions	<code>nfgrunwaldwang(<i>nf</i>, P, D, pl)</code>

Cyclotomic and Abelian fields theory

An Abelian field F given by a subgroup $H \subset (Z/fZ)^*$ is described by an argument F , e.g. f (for $H = 1$, i.e. $Q(\zeta_f)$) or $[G, H]$, where G is `idealstar(f , 1)`, or a minimal polynomial.

minus class number $h^-(F)$	<code>subcyclohminus(F)</code>
... p -part	<code>subcyclohminus(F, p)</code>
minus part of Iwasawa polynomials	<code>subcycloiwasawa(F, p)</code>
p -Sylow of $\text{Cl}(F)$	<code>subcyclopclgp(F, p)</code>

Logarithmic class group

logarithmic ℓ -class group	<code>bnflog(<i>bnf</i>, ℓ)</code>
$[\tilde{e}(F_v/Q_p), \tilde{f}(F_v/Q_p)]$	<code>bnflogef(<i>bnf</i>, pr)</code>
$\exp \deg_F(A)$	<code>bnflogdegree(<i>bnf</i>, A, ℓ)</code>
is ℓ -extension L/K locally cyclotomic	<code>rnfislocalcyclo(<i>rnf</i>)</code>

Ideals: elements, primes, or matrix of generators in HNF

is id an ideal in nf ?	<code>nfisideal(<i>nf</i>, id)</code>
is x principal in bnf ?	<code>bnfisprincipal(<i>bnf</i>, x)</code>
give $[a, b]$, s.t. $a\mathbf{Z}_K + b\mathbf{Z}_K = x$	<code>idealtwoelt(<i>nf</i>, x, {a})</code>
put ideal a ($a\mathbf{Z}_K + b\mathbf{Z}_K$) in HNF form	<code>idealhnf(<i>nf</i>, a, {b})</code>
norm of ideal x	<code>idealnrm(<i>nf</i>, x)</code>
minimum of ideal x (direction v)	<code>idealmin(<i>nf</i>, x, v)</code>
LLL-reduce the ideal x (direction v)	<code>idealred(<i>nf</i>, x, {v})</code>

Ideal Operations

add ideals x and y	<code>idealadd(<i>nf</i>, x, y)</code>
multiply ideals x and y	<code>idealmul(<i>nf</i>, x, y, {<i>flag</i>})</code>
intersection of ideal x with Q	<code>idealdown(<i>nf</i>, x)</code>
intersection of ideals x and y	<code>idealintersect(<i>nf</i>, x, y, {<i>flag</i>})</code>
n -th power of ideal x	<code>idealpow(<i>nf</i>, x, n, {<i>flag</i>})</code>
inverse of ideal x	<code>idealinv(<i>nf</i>, x)</code>
divide ideal x by y	<code>idealdiv(<i>nf</i>, x, y, {<i>flag</i>})</code>

Algebraic Number Theory

(PARI-GP version 2.15.2)

Find $(a, b) \in x \times y$, $a + b = 1$	<code>idealaddtoone(<i>nf</i>, x, {y})</code>
coprime integral A, B such that $x = A/B$	<code>idealnumden(<i>nf</i>, x)</code>

Primes and Multiplicative Structure

check whether x is a maximal ideal	<code>idealismaximal(<i>nf</i>, x)</code>
factor ideal x in \mathbf{Z}_K	<code>idealfactor(<i>nf</i>, x)</code>
expand ideal factorization in K	<code>idealfactorback(<i>nf</i>, f, {e})</code>
is ideal A an n -th power ?	<code>idealispower(<i>nf</i>, A, n)</code>
expand elt factorization in K	<code>nffactorback(<i>nf</i>, f, {e})</code>
decomposition of prime p in \mathbf{Z}_K	<code>idealprimedec(<i>nf</i>, p)</code>
valuation of x at prime ideal pr	<code>idealval(<i>nf</i>, x, pr)</code>
weak approximation theorem in nf	<code>idealchinese(<i>nf</i>, x, y)</code>
$a \in K$, s.t. $v_{\mathfrak{p}}(a) = v_{\mathfrak{p}}(x)$ if $v_{\mathfrak{p}}(x) \neq 0$	<code>idealappr(<i>nf</i>, x)</code>
$a \in K$ such that $(a \cdot x, y) = 1$	<code>idealcoprime(<i>nf</i>, x, y)</code>
give bid =structure of $(\mathbf{Z}_K/id)^*$	<code>idealstar(<i>nf</i>, id, {<i>flag</i>})</code>
structure of $(1 + \mathfrak{p})/(1 + \mathfrak{p}^k)$	<code>idealprincipalunits(<i>nf</i>, pr, k)</code>
discrete log of x in $(\mathbf{Z}_K/bid)^*$	<code>ideallog(<i>nf</i>, x, bid)</code>
<code>idealstar</code> of all ideals of norm $\leq b$	<code>ideallist(<i>nf</i>, b, {<i>ar</i>}, {<i>flag</i>})</code>
add Archimedean places	<code>ideallistarch(<i>nf</i>, b, {<i>ar</i>}, {<i>flag</i>})</code>
init <code>modpr</code> structure	<code>nfmodprinit(<i>nf</i>, pr, {v})</code>
project t to \mathbf{Z}_K/pr	<code>nfmodpr(<i>nf</i>, t, <i>modpr</i>)</code>
lift from \mathbf{Z}_K/pr	<code>nfmodprlift(<i>nf</i>, t, <i>modpr</i>)</code>

Galois theory over Q

conjugates of a root θ of nf	<code>nfgaloisconj(<i>nf</i>, {<i>flag</i>})</code>
apply Galois automorphism s to x	<code>nfgaloisapply(<i>nf</i>, s, x)</code>
Galois group of field $\mathbf{Q}[x]/(f)$	<code>polgalois(f)</code>
resolvent field of $\mathbf{Q}[x]/(f)$	<code>nfresolvent(f)</code>
initializes a Galois group structure G	<code>galoisinit(<i>pol</i>, {<i>den</i>})</code>
... for the splitting field of pol	<code>galoisplittinginit(<i>pol</i>, {d})</code>
character table of G	<code>galoischartable(G)</code>
conjugacy classes of G	<code>galoisconjclasses(G)</code>
$\det(1 - \rho(g)T)$, χ character of ρ	<code>galoischarpoly(G, χ, {o})</code>
$\det(\rho(g))$, χ character of ρ	<code>galoischarDET(G, χ, {o})</code>
action of p in <code>nfgaloisconj</code> form	<code>galoispermtpol(G, {p})</code>
identify as abstract group	<code>galoisidentify(G)</code>
export a group for GAP/MAGMA	<code>galoisexport(G, {<i>flag</i>})</code>
subgroups of the Galois group G	<code>galoissubgroups(G)</code>
is subgroup H normal?	<code>galoisisnormal(G, H)</code>
subfields from subgroups	<code>galoissubfields(G, {<i>flag</i>}, {v})</code>
fixed field	<code>galoisfixedfield(G, <i>perm</i>, {<i>flag</i>}, {v})</code>
Frobenius at maximal ideal P	<code>idealfrobenius(<i>nf</i>, G, P)</code>
ramification groups at P	<code>idealramgroups(<i>nf</i>, G, P)</code>
is G abelian?	<code>galoisisabelian(G, {<i>flag</i>})</code>
abelian number fields/ \mathbf{Q}	<code>galoissubcyclo(N,H, {<i>flag</i>}, {v})</code>

The galpol package

query the package: polynomial	<code>galoisgetpol(a,b,{s})</code>
...: permutation group	<code>galoisgetgroup(a,b)</code>
...: group description	<code>galoisgetname(a,b)</code>

Relative Number Fields (rnf)

Extension L/K is defined by $T \in K[x]$.

absolute equation of L	<code>rnfequation(<i>nf</i>, T, {<i>flag</i>})</code>
is L/K abelian?	<code>rnfisabelian(<i>nf</i>, T)</code>
relative <code>nfalgtobasis</code>	<code>rnfalgtobasis(<i>rnf</i>, x)</code>
relative <code>nfbasistoalg</code>	<code>rnfbasistoalg(<i>rnf</i>, x)</code>
relative <code>idealhnf</code>	<code>rnfidealhnf(<i>rnf</i>, x)</code>
relative <code>idealmul</code>	<code>rnfidealmul(<i>rnf</i>, x, y)</code>
relative <code>idealtwoelt</code>	<code>rnfidealtwoelt(<i>rnf</i>, x)</code>

Lifts and Push-downs

absolute \rightarrow relative representation for x	<code>rnfeltabstorel(<i>rnf</i>, x)</code>
relative \rightarrow absolute representation for x	<code>rnfeltrretoabs(<i>rnf</i>, x)</code>
lift x to the relative field	<code>rnfeltup(<i>rnf</i>, x)</code>
push x down to the base field	<code>rnfeltdown(<i>rnf</i>, x)</code>
idem for x ideal: (<code>rnfideal</code>) <code>reltoabs</code> , <code>abstorel</code> , <code>up</code> , <code>down</code>	

Norms and Trace

relative norm of element $x \in L$	<code>rnfeltnrm(<i>rnf</i>, x)</code>
relative trace of element $x \in L$	<code>rnfelttrace(<i>rnf</i>, x)</code>
absolute norm of ideal x	<code>rnfidealnrmabs(<i>rnf</i>, x)</code>
relative norm of ideal x	<code>rnfidealnrmrel(<i>rnf</i>, x)</code>
solutions of $N_{K/\mathbf{Q}}(y) = x \in \mathbf{Z}$	<code>bnfisintnrm(<i>bnf</i>, x)</code>
is $x \in \mathbf{Q}$ a norm from K ?	<code>bnfisnrm(<i>bnf</i>, x, {<i>flag</i>})</code>
initialize T for norm eq. solver	<code>rnfisnorminit(K,<i>pol</i>, {<i>flag</i>})</code>
is $a \in K$ a norm from L ?	<code>rnfisnrm(T, a, {<i>flag</i>})</code>
initialize t for Thue equation solver	<code>thueinit(f)</code>
solve Thue equation $f(x, y) = a$	<code>thue(t, a, {<i>sol</i>})</code>
characteristic poly. of a mod T	<code>rnfcharpoly(<i>nf</i>, T, a, {v})</code>

Factorization

factor ideal x in L	<code>rnfidealfactor(<i>rnf</i>, x)</code>
$[S, T]: T_{i,j} \mid S_i$; S primes of K above p	<code>rnfidealprimedec(<i>rnf</i>, p)</code>

Maximal order \mathbf{Z}_L as a \mathbf{Z}_K -module

relative <code>polredbest</code>	<code>rnfpolredbest(<i>nf</i>, T)</code>
relative <code>polredabs</code>	<code>rnfpolredabs(<i>nf</i>, T)</code>
relative Dedekind criterion, prime pr	<code>rnfdedekind(<i>nf</i>, T, pr)</code>
discriminant of relative extension	<code>rnfdisc(<i>nf</i>, T)</code>
pseudo-basis of \mathbf{Z}_L	<code>rnfpseudobasis(<i>nf</i>, T)</code>

General \mathbf{Z}_K -modules: $M = [\text{matrix, vec. of ideals}] \subset L$

relative HNF / SNF	<code>nfhnf(<i>nf</i>, M), nfsnf</code>
multiple of $\det M$	<code>nfDETint(<i>nf</i>, M)</code>
HNF of M where $d = nfDETint(M)$	<code>nfhnfmod(x, d)</code>
reduced basis for M	<code>rnfilllgram(<i>nf</i>, T, M)</code>
determinant of pseudo-matrix M	<code>rnfdet(<i>nf</i>, M)</code>
Steinitz class of M	<code>rnfstEinitz(<i>nf</i>, M)</code>
\mathbf{Z}_K -basis of M if \mathbf{Z}_K -free, or 0	<code>rnfhnfBasis(<i>bnf</i>, M)</code>
n -basis of M , or $(n + 1)$ -generating set	<code>rnfbasis(<i>bnf</i>, M)</code>
is M a free \mathbf{Z}_K -module?	<code>rnfisfree(<i>bnf</i>, M)</code>

Associative Algebras

A is a general associative algebra given by a multiplication table *mt* (over **Q** or **F_p**); represented by *al* from `algtableinit`.
create *al* from *mt* (over **F_p**) `algtableinit(mt, {p = 0})`
group algebra **Q**[*G*] (or **F_p**[*G*]) `alggroup(G, {p = 0})`
center of group algebra `alggrouppcenter(G, {p = 0})`
Properties
is (*mt*, *p*) OK for `algtableinit`? `algisassociative(mt, {p = 0})`
multiplication table *mt* `algmultable(al)`
dimension of *A* over prime subfield `algdim(al)`
characteristic of *A* `algchar(al)`
is *A* commutative? `algiscommutative(al)`
is *A* simple? `algissimple(al)`
is *A* semi-simple? `algissemisimple(al)`
center of *A* `algcenter(al)`
Jacobson radical of *A* `algradical(al)`
radical *J* and simple factors of *A*/*J* `algsimpledec(al)`
Operations on algebras
create *A*/*I*, *I* two-sided ideal `algquotient(al, I)`
create *A*₁ ⊗ *A*₂ `algtensor(al1, al2)`
create subalgebra from basis *B* `algsubalg(al, B)`
quotients by ortho. central idempotents *e* `algcentralproj(al, e)`
isomorphic alg. with integral mult. table `algmakeintegral(mt)`
prime subalgebra of semi-simple *A* over **F_p** `algprimesubalg(al)`
find isomorphism *A* ≅ *M_d*(**F_q**) `algsplit(al)`
Operations on lattices in algebras
lattice generated by cols. of *M* `alglathnf(al, M)`
... by the products *xy*, *x* ∈ *lat1*, *y* ∈ *lat2* `alglatmul(al, lat1, lat2)`
sum *lat1* + *lat2* of the lattices `alglatadd(al, lat1, lat2)`
intersection *lat1* ∩ *lat2* `alglatinter(al, lat1, lat2)`
test *lat1* ⊂ *lat2* `alglatsubset(al, lat1, lat2)`
generalized index (*lat2* : *lat1*) `alglatindex(al, lat1, lat2)`
{*x* ∈ *al* | *x* · *lat1* ⊂ *lat2*} `alglatlefttransporter(al, lat1, lat2)`
{*x* ∈ *al* | *lat1* · *x* ⊂ *lat2*} `alglatrighttransporter(al, lat1, lat2)`
test *x* ∈ *lat* (set *c* = coord. of *x*) `alglatcontains(al, lat, x, {&c})`
element of *lat* with coordinates *c* `alglatelement(al, lat, c)`
Operations on elements
a + *b*, *a* − *b*, −*a* `algadd(al, a, b), algsub, algneg`
a × *b*, *a*² `algmul(al, a, b), algsqr`
aⁿ, *a*^{−1} `algpow(al, a, n), alginv`
is *x* invertible ? (then set *z* = *x*^{−1}) `alginv(al, x, {&z})`
find *z* such that *x* × *z* = *y* `algdivl(al, x, y)`
find *z* such that *z* × *x* = *y* `algdivr(al, x, y)`
does *z* s.t. *x* × *z* = *y* exist? (set it) `algsdivl(al, x, y, {&z})`
matrix of *v* ↦ *x* · *v* `algtomatrix(al, x)`
absolute norm `algnorm(al, x)`
absolute trace `algtrace(al, x)`
absolute char. polynomial `algcharpoly(al, x)`
given *a* ∈ *A* and polynomial *T*, return *T*(*a*) `algpoleval(al, T, a)`
random element in a box `algrandom(al, b)`

Central Simple Algebras

A is a central simple algebra over a number field *K*; represented by *al* from `algininit`; *K* is given by a *nf* structure.
create CSA from data `algininit(B, C, {v}, {maxord = 1})`
multiplication table over *K* *B* = *K*, *C* = *mt*
cyclic algebra (*L*/*K*, *σ*, *b*) *B* = *rnf*, *C* = [*sigma*, *b*]
quaternion algebra (*a*, *b*)_{*K*} *B* = *K*, *C* = [*a*, *b*]
matrix algebra *M_d*(*K*) *B* = *K*, *C* = *d*
local Hasse invariants over *K* *B* = *K*, *C* = [*d*, [*PR*, *HF*], *HI*]

Properties

type of *al* (*mt*, CSA) `algtype(al)`
dimension of *A* over **Q** `algdim(al, 1)`
dimension of *al* over its center *K* `algdim(al)`
degree of *A* (= √dim_{*K*} *A*) `algdegree(al)`
al a cyclic algebra (*L*/*K*, *σ*, *b*); return *σ* `algaut(al)`
... return *b* `algb(al)`
... return *L*/*K*, as an *rnf* `algsplittingfield(al)`
split *A* over an extension of *K* `algsplittingdata(al)`
splitting field of *A* as an *rnf* over center `algsplittingfield(al)`
multiplication table over center `algrelmultable(al)`
places of *K* at which *A* ramifies `algramifiedplaces(al)`
Hasse invariants at finite places of *K* `alghassef(al)`
Hasse invariants at infinite places of *K* `alghassei(al)`
Hasse invariant at place *v* `alghasse(al, v)`
index of *A* over *K* (at place *v*) `algindex(al, {v})`
is *al* a division algebra? (at place *v*) `algsdivision(al, {v})`
is *A* ramified? (at place *v*) `algsramified(al, {v})`
is *A* split? (at place *v*) `algsisplit(al, {v})`

Operations on elements

reduced norm `algnorm(al, x)`
reduced trace `algtrace(al, x)`
reduced char. polynomial `algcharpoly(al, x)`
express *x* on integral basis `algalgtobasis(al, x)`
convert *x* to algebraic form `algbasistoalg(al, x)`
map *x* ∈ *A* to *M_d*(*L*), *L* split. field `algtomatrix(al, x)`

Orders

Z-basis of order *O*₀ `algbasis(al)`
discriminant of order *O*₀ `algdisc(al)`
Z-basis of natural order in terms *O*₀'s basis `alginvbasis(al)`