PARI-GP Reference Card
(PARI-GP version 2.1.0)

Note: optional arguments are surrounded by braces {}.

Starting & Stopping GP
to enter GP, just type its name:
`gp`
to exit GP, type:
`\q` or `quit`

Help
describe function
`?function`
extended description
`??function`
list of relevant help topics
`???pattern`

Input/Output & Defaults
output previous line, the lines before output from line n
separate multiple statements on line
extend statements on additional lines
`\extend`
`\n` resume line
set default d to `val`
default `val` to `d`
liche behaviour of GP 1.39

Metacommands
toggle timer on/off
`#`
print time for last result
`##`
print %n in raw format
`\` n
print %n in pretty format
`\n` n
print defaults
`\d`
set debug level to n
`\gm` n
set memory debug level to n
set log on/off
`\l` (filename)
enable/disable logfile
print %n in pretty matrix format
`\m` n
set output mode (raw, default, prettyprint)
n
set n significant digits
`\p` n
set n terms in series
`\ns` n
quit GP
`\q`
print the list of PARI types
`\t`
print the list of user-defined functions
read file into GP
`\u`
write %n to file
`\w` n filename

GP Within Emacs
to enter GP from within Emacs:
word completion
`M-x gp`, `C-u M-x gp`
help menu window
`<TAB>`
describe function
`M-?`
display TEx'd PARI manual
`M-x gpman`
set prompt string
`M-` p
break line at column 100, insert \n
PARI metacommands Vletter
`M-x letter`

Reserved Variable Names
π = 3.14159... Pi
Euler's constant = .57721... Euler
square root of -1 i
big-oh notation O

PARI Types & Input Formats

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_INT</td>
<td>Integers</td>
<td>±n</td>
</tr>
<tr>
<td>t_REAL</td>
<td>Real Numbers</td>
<td>±n,0.5</td>
</tr>
<tr>
<td>t_NAVMOD</td>
<td>Integers modulo n</td>
<td>Mod(n,m)</td>
</tr>
<tr>
<td>t_FRAC</td>
<td>Rational Numbers</td>
<td>x / y</td>
</tr>
<tr>
<td>t_COMPLEX</td>
<td>Complex Numbers</td>
<td>x + i y</td>
</tr>
<tr>
<td>t_PADIC</td>
<td>p-adic Numbers</td>
<td>x + O(p^k)</td>
</tr>
<tr>
<td>t_QUAD</td>
<td>Quadratic Numbers</td>
<td>x + y* quad(Q)</td>
</tr>
<tr>
<td>t_POLMOD</td>
<td>Polynomials modulo g</td>
<td>Mod(f,g)</td>
</tr>
<tr>
<td>t_POL</td>
<td>Polynomials</td>
<td>a + x*n + ... + b</td>
</tr>
<tr>
<td>t_SER</td>
<td>Power Series</td>
<td>f + O(x^n)</td>
</tr>
<tr>
<td>t_QPQ</td>
<td>Imag/Real bin. quad. forms</td>
<td>Qb(a,b,c,d)</td>
</tr>
<tr>
<td>t_REALF</td>
<td>Rational Functions</td>
<td>f/g</td>
</tr>
<tr>
<td>t_VEC</td>
<td>Row/Column Vectors</td>
<td>[x,y,z]</td>
</tr>
<tr>
<td>t_MAT</td>
<td>Matrices</td>
<td>x,y,z,t</td>
</tr>
<tr>
<td>t_LIST</td>
<td>List</td>
<td>x,y,z</td>
</tr>
<tr>
<td>t_STR</td>
<td>Strings</td>
<td>&quot;aaa&quot;</td>
</tr>
</tbody>
</table>

Standard Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Add</td>
<td>x+y</td>
</tr>
<tr>
<td>-</td>
<td>Subtract</td>
<td>x-y</td>
</tr>
<tr>
<td>*</td>
<td>Multiply</td>
<td>x*y</td>
</tr>
<tr>
<td>/</td>
<td>Divide</td>
<td>x/y</td>
</tr>
<tr>
<td>%</td>
<td>Modulo</td>
<td>x%y</td>
</tr>
<tr>
<td>!</td>
<td>Sign</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>Maximum/Minimum</td>
<td>max(x,y,z)</td>
</tr>
<tr>
<td>=</td>
<td>Equal</td>
<td>x=y</td>
</tr>
<tr>
<td>!=</td>
<td>Not equal</td>
<td>x≠y</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
<td>x&lt;=y</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
<td>x&gt;=y</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
<td>x&lt;y</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
<td>x&gt;y</td>
</tr>
</tbody>
</table>

Conversions

<table>
<thead>
<tr>
<th>Conversion</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>vec</td>
<td>Convert to vector</td>
<td>Vec(x)</td>
</tr>
<tr>
<td>mat</td>
<td>Convert to matrix</td>
<td>Mat(x)</td>
</tr>
<tr>
<td>str</td>
<td>Convert to string</td>
<td>Str(x)</td>
</tr>
<tr>
<td>int</td>
<td>Convert to integer</td>
<td>Int(x)</td>
</tr>
<tr>
<td>float</td>
<td>Convert to floating point</td>
<td>Float(x)</td>
</tr>
<tr>
<td>pol</td>
<td>Convert to polynomial</td>
<td>Pol(x)</td>
</tr>
<tr>
<td>series</td>
<td>Convert to series</td>
<td>Series(x)</td>
</tr>
<tr>
<td>frac</td>
<td>Convert to fraction</td>
<td>Frac(x)</td>
</tr>
<tr>
<td>con</td>
<td>Convert to complex</td>
<td>Con(x)</td>
</tr>
<tr>
<td>conj</td>
<td>Conjugate</td>
<td>Conj(x)</td>
</tr>
<tr>
<td>lift</td>
<td>Lift</td>
<td>Lift(x)</td>
</tr>
</tbody>
</table>

Lists, Sorts & Sorting

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Create empty list</td>
<td>list</td>
</tr>
<tr>
<td>listcreate</td>
<td>Create list</td>
<td>listcreate(n)</td>
</tr>
<tr>
<td>listprint</td>
<td>Print list</td>
<td>listprint(l)</td>
</tr>
<tr>
<td>listprint1</td>
<td>Print list 1</td>
<td>listprint1(l)</td>
</tr>
<tr>
<td>listsort</td>
<td>Sort list</td>
<td>listsort(l)</td>
</tr>
<tr>
<td>listsort1</td>
<td>Sort list 1</td>
<td>listsort1(l)</td>
</tr>
<tr>
<td>listsearch</td>
<td>Search list</td>
<td>listsearch(l, x)</td>
</tr>
<tr>
<td>listunion</td>
<td>Union list</td>
<td>listunion(l1, l2)</td>
</tr>
<tr>
<td>listintersect</td>
<td>Intersection of lists</td>
<td>listintersect(l1, l2)</td>
</tr>
<tr>
<td>listinsert</td>
<td>Insert into list</td>
<td>listinsert(l, i, x)</td>
</tr>
<tr>
<td>listappend</td>
<td>Append to list</td>
<td>listappend(l, x)</td>
</tr>
</tbody>
</table>

Programming & User Functions

Control Statements

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>for</td>
<td>Execute statements n times</td>
<td>for(n=1,2,3)</td>
</tr>
<tr>
<td>forstep</td>
<td>Execute statements n times</td>
<td>forstep(N=1,2,3)</td>
</tr>
<tr>
<td>forprimes</td>
<td>Execute statements for all</td>
<td>forprimes(x=1,2,3)</td>
</tr>
<tr>
<td>forstepprimes</td>
<td>Execute statements for all</td>
<td>forstepprimes(x=1,2,3)</td>
</tr>
<tr>
<td>if</td>
<td>Execute only if true</td>
<td>if(x)</td>
</tr>
<tr>
<td>unless</td>
<td>Execute only if false</td>
<td>unless(x)</td>
</tr>
<tr>
<td>unlessif</td>
<td>Execute only if (x) false</td>
<td>unlessif(x)</td>
</tr>
<tr>
<td>until</td>
<td>Execute statements until</td>
<td>until(n)</td>
</tr>
<tr>
<td>exit</td>
<td>Exit GP</td>
<td>exit()</td>
</tr>
<tr>
<td>break</td>
<td>Exit loop</td>
<td>break()</td>
</tr>
</tbody>
</table>

Input/Output

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>print</td>
<td>Print to standard output</td>
<td>print(x)</td>
</tr>
<tr>
<td>printl</td>
<td>Print to standard output</td>
<td>printl(x)</td>
</tr>
<tr>
<td>printf</td>
<td>Print format</td>
<td>printf(x, y)</td>
</tr>
<tr>
<td>fprintf</td>
<td>Print format</td>
<td>fprintf(x, y)</td>
</tr>
<tr>
<td>input</td>
<td>Read input from standard</td>
<td>input()</td>
</tr>
<tr>
<td>output</td>
<td>Output to standard output</td>
<td>output(x)</td>
</tr>
<tr>
<td>write</td>
<td>Write to standard output</td>
<td>write(x)</td>
</tr>
<tr>
<td>writeln</td>
<td>Write to standard output</td>
<td>writeln(x)</td>
</tr>
<tr>
<td>writetex</td>
<td>Write to standard output</td>
<td>writetex(x)</td>
</tr>
<tr>
<td>writexy</td>
<td>Write to standard output</td>
<td>writexy(x)</td>
</tr>
</tbody>
</table>

Interface with User and System

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>allocatemem</td>
<td>Allocate memory</td>
<td>allocatemem()</td>
</tr>
<tr>
<td>extern</td>
<td>Execute system command</td>
<td>extern()</td>
</tr>
<tr>
<td>install</td>
<td>Install function</td>
<td>install(f, code, {gfp}, {lib})</td>
</tr>
<tr>
<td>alias</td>
<td>Define alias</td>
<td>alias(f, g)</td>
</tr>
<tr>
<td>whatnow</td>
<td>Query what function</td>
<td>whatnow(f)</td>
</tr>
</tbody>
</table>

User Defined Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>vec</td>
<td>Vector of elements</td>
<td>vec(x)</td>
</tr>
<tr>
<td>sum</td>
<td>Sum of elements</td>
<td>sum(x)</td>
</tr>
<tr>
<td>product</td>
<td>Product of elements</td>
<td>prod(x)</td>
</tr>
<tr>
<td>max</td>
<td>Maximum of elements</td>
<td>max(x)</td>
</tr>
<tr>
<td>min</td>
<td>Minimum of elements</td>
<td>min(x)</td>
</tr>
<tr>
<td>gcd</td>
<td>Greatest common divisor</td>
<td>gcd(x, y)</td>
</tr>
<tr>
<td>lcm</td>
<td>Least common multiple</td>
<td>lcm(x, y)</td>
</tr>
<tr>
<td>factorial</td>
<td>Factorial</td>
<td>factorial(x)</td>
</tr>
<tr>
<td>binomial</td>
<td>Binomial coefficient</td>
<td>binomial(n, k)</td>
</tr>
<tr>
<td>prime</td>
<td>Prime number</td>
<td>prime(n)</td>
</tr>
</tbody>
</table>

Iterations, Sums & Products

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>intnum</td>
<td>Integer of expression</td>
<td>intnum(x, y)</td>
</tr>
<tr>
<td>sum</td>
<td>Sum of expressions</td>
<td>sum(x)</td>
</tr>
<tr>
<td>product</td>
<td>Product of expressions</td>
<td>product(x)</td>
</tr>
<tr>
<td>gcd</td>
<td>Greatest common divisor</td>
<td>gcd(x, y)</td>
</tr>
<tr>
<td>lcm</td>
<td>Least common multiple</td>
<td>lcm(x, y)</td>
</tr>
<tr>
<td>factorial</td>
<td>Factorial</td>
<td>factorial(x)</td>
</tr>
<tr>
<td>binomial</td>
<td>Binomial coefficient</td>
<td>binomial(n, k)</td>
</tr>
<tr>
<td>prime</td>
<td>Prime number</td>
<td>prime(n)</td>
</tr>
<tr>
<td>product</td>
<td>Product of expressions</td>
<td>product(x)</td>
</tr>
<tr>
<td>intnum</td>
<td>Integer of expression</td>
<td>intnum(x, y)</td>
</tr>
<tr>
<td>sum</td>
<td>Sum of expressions</td>
<td>sum(x)</td>
</tr>
<tr>
<td>product</td>
<td>Product of expressions</td>
<td>product(x)</td>
</tr>
<tr>
<td>gcd</td>
<td>Greatest common divisor</td>
<td>gcd(x, y)</td>
</tr>
<tr>
<td>lcm</td>
<td>Least common multiple</td>
<td>lcm(x, y)</td>
</tr>
<tr>
<td>factorial</td>
<td>Factorial</td>
<td>factorial(x)</td>
</tr>
<tr>
<td>binomial</td>
<td>Binomial coefficient</td>
<td>binomial(n, k)</td>
</tr>
<tr>
<td>prime</td>
<td>Prime number</td>
<td>prime(n)</td>
</tr>
<tr>
<td>product</td>
<td>Product of expressions</td>
<td>product(x)</td>
</tr>
<tr>
<td>intnum</td>
<td>Integer of expression</td>
<td>intnum(x, y)</td>
</tr>
<tr>
<td>sum</td>
<td>Sum of expressions</td>
<td>sum(x)</td>
</tr>
<tr>
<td>product</td>
<td>Product of expressions</td>
<td>product(x)</td>
</tr>
<tr>
<td>gcd</td>
<td>Greatest common divisor</td>
<td>gcd(x, y)</td>
</tr>
<tr>
<td>lcm</td>
<td>Least common multiple</td>
<td>lcm(x, y)</td>
</tr>
<tr>
<td>factorial</td>
<td>Factorial</td>
<td>factorial(x)</td>
</tr>
<tr>
<td>binomial</td>
<td>Binomial coefficient</td>
<td>binomial(n, k)</td>
</tr>
<tr>
<td>prime</td>
<td>Prime number</td>
<td>prime(n)</td>
</tr>
<tr>
<td>product</td>
<td>Product of expressions</td>
<td>product(x)</td>
</tr>
</tbody>
</table>
Vectors & Matrices

dimensions of matrix x, concatenation of x and y, extract components of x, transpose of vector or matrix x, adjoint of the matrix x, eigenvectors of matrix x, characteristic polynomial of x, trace of matrix x

Constructors & Special Matrices

ew row vec. of expr eval’d at 1 ≤ x ≤ n vector(n, {X}, {expr})
col. vec. of expr eval’d at 1 ≤ x ≤ n vector(n, {X}, {expr})
matrix of m rows, 1 ≤ Y ≤ n matrix(m, n, {X}, {Y}, {expr})
diagonal matrix whose diag. is x, x

Hessenberg form of square matrix x

n x n Hilbert matrix H_{ij} = (i + j - 1)^{-1}
n x n Pascal triangle \binom{i}{j}

Gaussian elimination

determinant of matrix x

intersection of column spaces of x and y

solve M x = S (B invertible)
as solve, modulo D (col. vector)
one sol. of M x = S

basis for image of matrix x

supplement column of x to get basis rows, cols to extract invertible matrix rank of the matrix x

Lattices & Quadratic Forms

upper triangular Hermite Normal Form

HNF of x where d is a multiple of det(x)

vector of elementary divisors of x

LLL-algorithm applied to columns of x, like qflll, x is Gram matrix of lattice LLL-reduced basis for kernel of x

Z-lattice ---- Q-vector space

Signature of quad form \langle y \mid x \rangle

decomposed into squares of y \parallel x \parallel y

find up to m solns of y \parallel x \parallel y \leq b

eigenvalues/eigenvalues for real symmetric x

Formal & p-adic Series

truncate power series or p-adic number valuation of x at p

Taylor expansion of f at 0 of w.r.t. x

\sum a_k x^k from \sum a_k x^k and \sum b_k x^k

f = \sum a_k x^k from \sum a_k x^k and \sum b_k x^k

reverse power series F so F(x) = x

Dirichlet series multiplication / division

Dirichlet Euler product (b terms)

p-adic Functions

square of x, good for 2-adics

Teichmuller character of x

Newton polygon of f for prime p

matsize(x)

concat(x, {y})

veceextract(x, y, {z})

matt transpose(x, y, {z})

matadj(x)

matelgen(x)

charpoly(x, {v}, {f})

trace(x)

PARI-GP Reference Card

(PARI-GP version 2.1.0)

Polynomials & Rational Functions

degree of f

coefficient of degree n of f

round coeff of f to nearest integer

gcd of coefficients of f

replace x by y in f

discriminant of polynomial f

resultant of f and y

as above, give [v, n, d], xu + yv = d

derivative of f w.r.t. x

formal integral of f w.r.t. x

reciprocal poly \deg f_{x, t}

interpolating poly evaluated at \{X,Y\}

initialize t for Thue equation solver

solve Thue equation f(x,y) = a

Roots and Factorization

number of real roots of f, a < x \leq b

complex roots of f

symmetric powers of roots of f up to n roots of f mod p

factor f

factorization of f over \mathbb{F}_p

p-adic fact. of f to prec. r

p-adic root of f to prec. r

p-adic root of f congr. to a mod p

Newton polygon of f for prime p

Special Polynomials

nth cyclotomic polynomial in var. v

d-th degree subfield of Q_x

nth Legendre polynomial

nth Tchebicheff polynomial

Zagier’s polynomial of index n,m

Transcendental Functions

real, imaginary part of x

absolute value of x

square/nth root of x

trig functions

inverse trig functions

hyperbolic functions

inverse hyperbolic functions

exponential of x

natural log of x

gamma function \Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt

logarithm of gamma function \psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}

incomplete gamma function (y = \psi(x))

exponential integral \int_y^\infty e^{-t} t dt

error function 2/\sqrt{\pi} \int_0^x e^{-t^2} dt

dilogarithm of x

m-th polylogarithm of x

\mathbb{U}-confluent hypergeometric function

\text{J-Bessel function J}_{\nu/2}(x)

K-Bessel function of index \nu

poldegree(f)

polcoeff(f, n)

round(f, {k}, {e})

content(f)

subst(f, x, y)

poldisc(f)

polresultant(f, y, {R})

bezoutres(f, x, y)

deriv(f, x)

informal(f, x)

polrecip(f)

polfftfact(f, {x, y})

factorpol(f, {x, y})

factornf(f, {n, x})

Newton polygon of \mathbb{F}_p

number of distinct prime divisors

number of prime divisors with mult number of divisors of x

row vectors of divisor x

sum of (k-th powers of) divisors of x

Special Functions & Number Theory

binomial coefficient \binom{x}{y}

Bernoulli number B_n, as real

Bernoulli vector B_1, B_2, ..., B_{2n}

nth Fibonacci number

Euler \phi-function

Möbius \mu-function

Harit symbol of x and y (at p)

Kronecker-Legendre symbol \langle x \mid y \rangle

Miscellaneous

integer or real factorial of x

integer square root of x

solve \mathbb{Z} \times \mathbb{Z}

minimal u, so xu + yv = gcd(x,y)

multiplicative order of x (inmod x)

primitive root mod prime power x

structure of \mathbb{Z}/n\mathbb{Z}^*

continued fraction of x

best rational approximation to x

binary(x)

bittest(x, n)

ceil(x)

floor(x)

fractional part of x

round(x, {k}, {e})

truncate x

gcd of x and y

LCM of x and y

gcd of entries of a vector/matrix

Primes and Factorization

add primes in x to the prime table

the nth prime

smallest prime \geq x

largest prime \leq x

factorization of x

reconstruct x from its factorization

Divisors

number of prime divisors

number of prime divisors with mult number of divisors of x

row vectors of divisor x

sum of (k-th powers of) divisors of x

Special Functions & Number Theory

binomial coefficient \binom{x}{y}

Bernoulli number B_n, as real

Bernoulli vector B_1, B_2, ..., B_{2n}

nth Fibonacci number

Euler \phi-function

Möbius \mu-function

Harit symbol of x and y (at p)

Kronecker-Legendre symbol \langle x \mid y \rangle

Test-False Tests

is x the disc. of a quadratic field?

is a prime?

is a strong pseudo-prime?

is square-free?

is a square?

is a strong pseudo-prime?

is pol irreducible?

Based on an earlier version by Joseph H. Silverman

November 2000 v2.11. Copyright © 2000 K. Belabas

GP copyright by The PARI Group

Permission is granted to make and distribute copies of this card provided the copyright and this permission notice are preserved on all copies.

Send comments and corrections to (Karim.BELABAS@math.u-psud.fr)
Elliptic Curves

Elliptic curve initially given by 5-tuple \( E = \{a_1, a_2, a_3, a_4, a_6\} \). Points are \([x, y]\), the origin is \([0, 0]\).

Initialize elliptic structure. \texttt{ellinit}(\( E, \{f_1\} \))

\( a_1, a_2, a_3, a_4, a_6, b_2, b_4, b_6, c_4, c_6, \text{disc.} j \). This data can be recovered by typing \texttt{elltinit}.

\[ E \text{ defined over } \mathbf{R} \]
\[ \text{axes: points of order 2} \]
\[ \text{real and complex periods} \]
\[ \text{associated quasi-periods} \]
\[ \text{volume of complex lattice} \]
\[ E \text{ defined over } \mathbf{Q}_p, |p| > 1 \]
\[ \text{coordinate of } pt \text{ torsion point} \]
\[ \text{Tate's } [u^2, v, q] \]
\[ Mestre's w \]
\[ \text{change curve } E \text{ using } v = [u, r, s, t] \]
\[ \text{change point } z \text{ using } w = [u, r, s, t] \]
\[ \text{cond, min mod, Tamgawa nmbr } [N, v, c] \]

Kodaira type of p fiber of \( E \)

\[ \text{add points } z_1 + z_2 \]
\[ \text{subtract points } z_1 - z_2 \]
\[ \text{compute } n z \]
\[ \text{check if } z \text{ is on } E \]
\[ \text{order of torsion point } z \]
\[ \text{torsion subgroup with generators } \{ \text{coordinates of } (s) \text{ for } x \} \]
\[ \text{canonical bilinear form taken at } z_1, z_2 \]
\[ \text{canonical height of } z \]
\[ \text{height regulator matrix for pts in } x \]
\[ \text{p-th coeff } a_p \text{ of } L\text{-function, } \text{prime} \]
\[ \text{k-th coeff } a_k \text{ of } L\text{-function} \]

\[ \text{vector of first } n a_q \text{ in } L\text{-function} \]

\( L(E, s), s \approx 1 \)
\[ \text{root number for } L(E, s) \text{ at } p \]
\[ \text{modular parametrization of } E \]

\[ \text{point } [p(z), q(z)] \text{ corresp. to } z \]
\[ \text{complex } z \text{ such that } p = [p(z), q(z)] \]

Elliptic & Modular Functions

\[ \text{arithmetic-geometric mean} \]
\[ \text{elliptic j-function } 1 + \frac{1}{\pi^2} + 4 \cdots \]
\[ \text{Weierstrass } \sigma \text{ function} \]
\[ \text{Weierstrass } \delta \text{ function} \]
\[ \text{Weierstrass } \zeta \text{ function} \]
\[ \text{Weierstrass } \eta \text{ function} \]
\[ \text{Jacobi sine theta function} \]
\[ \text{k-th derivative at } z=0 \text{ of } \theta(q, z) \]
\[ \text{Weber's } f \text{ functions} \]
\[ \text{Riemann's } \zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} \]

Graphic Functions

\[ \text{crude graph of } \text{expr between } a \text{ and } b \]
\[ \text{High-resolution plot (immediate plot)} \]
\[ \text{plot expr between } a \text{ and } b \]
\[ \text{plot } X \text{ at } (x, y) \]
\[ \text{plot } \text{copy} (w, x, y) \]
\[ \text{plot } \text{copy} (w, x, y) \]
\[ \text{plot } \text{plot} (w, x, y) \]
\[ \text{plot } \text{plot} (w, x, y) \]
\[ \text{plot } \text{plot} (w, x, y) \]

Low-level Rectangular Functions

\[ \text{set current drawing color in } c \]
\[ \text{current position of } \text{cursor in } w \]
\[ \text{move cursor to } (x, y) \]
\[ \text{move cursor to } (x + dx, y + dy) \]
\[ \text{draw a box to } (x+1, y+1) \]
\[ \text{draw polygon} \]
\[ \text{draw line to } (x + dx, y + dy) \]
\[ \text{draw point } (x + dx, y + dy) \]

Postscript Functions

\[ \text{as plot} \]
\[ \text{as plot} \]
\[ \text{as plot} \]

Binary Quadratic Forms

\[ x^2 + kxy + cy^2 \]
\[ \text{reduce } x \text{ (} x = \sqrt{\Delta}, l = \{a\} \]
\[ \text{composition of } f_1 + f_2 \text{ in } (x, y) \]
\[ n \text{-th power of form} \]
\[ n \text{-th power without reduction} \]
\[ \text{prime form of } \text{disc.} x \text{ above prime } p \]
\[ \text{class number of disc. } x \]
\[ \text{Hurwitz class number of disc. } x \]

Quadratic Fields

\[ \text{quadratic number } \omega = \sqrt{x} \text{ or } \sqrt{1 + \sqrt{x}} \]
\[ \text{minimal polynomial of } \omega \]
\[ \text{discriminant of } Q(\sqrt{\omega}) \]
\[ \text{regulator of real quadratic field} \]
\[ \text{fundamental unit in real } Q(x) \]
\[ \text{class group of } \text{Q}(\sqrt{\omega}) \]
\[ \text{Hilbert class field of } Q(\sqrt{\omega}) \]
\[ \text{ray class field modulo } \text{Q}(\sqrt{\omega}) \]

General Number Fields: Initializations

A number field \( K \) given by a monic irreducible \( f \in \mathbf{Z}[X] \).

\[ \text{init number field structure } nft \]
\[ \text{init finite field } \{ f \} \]

nf members:

\[ \text{polynomial defining } f(\theta) = 0 \]
\[ \text{number of } [\text{real,complex} ] \text{ places} \]
\[ \text{discriminant of } nf \]
\[ \text{T2 matrix} \]
\[ \text{vector of roots of } f \]
\[ \text{integral basis of } \mathbf{Z}_K \text{ as powers of } \theta \]
\[ \text{different} \]
\[ \text{coefficient} \]
\[ \text{recompute } nf \]using current precision
\[ \text{init relative } nf \text{ given by } g = 0 \text{ over } K \]
\[ \text{init big number field structure } \text{bnf} \]

bnf members:

\[ \text{same as } nf \text{ plus} \]
\[ \text{underlying } nf \]
\[ \text{classgroup} \]
\[ \text{regulator} \]
\[ \text{fundamental units} \]
\[ \text{torsion units} \]
\[ \text{compute a } \text{bnf from small } \text{bnf} \]
\[ \text{add } S \text{-class group and units, yield } \text{bnfs} \]

\[ \text{init class field structure } \text{bnr} \]

bnr members:

\[ \text{same as } bnf \text{ plus} \]
\[ \text{underlying } bnf \]
\[ \text{structure of } \mathbf{Z}_K/m^* \]
Simple Arithmetic Invariants (nf)

Elements are rational numbers, polynomials, polmods, or column vectors (on integral basis nf.zk).

integral basis of field def. by $f = 0$

field discriminant of field $f = 0$

reverse polynomial $a = A(X) \mod T(X)$

Galois group of field $f = 0$, deg $\leq 11$

smallest poly defining $f = 0$

small polys defining subfields of $f = 0$

small polys defining subfields of $f = 0$

principal ideal generated by $x$

principal idele generated by $x$

$
\begin{array}{l}
\text{give (a, b), s.t. } aZ_K + bZ_K = x \\
\text{idealvoetf(nf, x, {u})}
\end{array}$

idealn(nf, x, {b})

idealmin(nf, x, v)

(LL-reduce the ideal x (direction c))

Ideal Operations

add ideals x and y

ideadd(nf, x, y, {f})

multiply ideals x and y

idealmul(nf, x, y, {f})

intersection of ideals x and y

idealintersec(nf, x, y, {f})

$n$-th power of ideal $x$

idealpow(nf, x, n, {f})

inverse of ideal $x$

idealinv(nf, x, {f})

divide ideal $x$ by $y$

idealdiv(nf, x, y, {f})

Find $[a, b]$ in $x \times y$, $a + b = 1$

idealaddtoone(nf, x, y, {f})

Primes and Multiplicative Structure

factor ideal $x$ in $nf$

idealfactor(nf, x, {f})

recover $x$ from its factorization in $nf$

factorback(x, {f})

decomposition of prime $p$ in $nf$

idealprimdec(nf, p, {f})

valuation of $x$ at prime ideal $p$

idealval(nf, x, p, {f})

valuation of $x$ at prime ideal $p$

idealval(nf, x, p, {f})

weak approximation theorem in $nf$

idealchinese(nf, x, y, {f})

give $bid = \text{structure of (Z}_K/\text{id})^*$

idealtot(nf, x, {f})

$\text{discrete log of } x \in (Z}_K/\text{bid})^*$

ideallog(nf, x, {f})

$\text{idealstar of all ideals of norm } \leq b$

idealstar(nf, x, {f})

add archimedean places

idealstarstech(nf, x, y, {f})

init $\text{prmod structure}$

nfsolveprid(nf, x, {f})

kernel of matrix $M$ in $(Z}_K/\text{pr})^*$

nfsolnmod(nf, M, prmod)

solve $Mx = B \in (Z}_K/\text{pr})^*$

nfsolvegprid(nf, M, B, prmod)

Relative Number Fields (rnf)

Extension $L/K$ is defined by $g \in K[x]$. We have order $\subset L$

absolute equation of $L$

nfequation(nf, g, {f})

Lifts and Push-downs

absolute $\rightarrow$ relative repres. for $x$

rnfeltabstore(nf, x, {f})

relative $\rightarrow$ absolute repres. for $x$

rnfeltretoabs(nf, x, {f})

lift $x$ to the relative field

rnfeltup(nf, x, {f})

push $x$ down to the base field

rnfeltdown(nf, x, {f})

iden for $x$ ideal: (rnfidealretoabs, abstore, up, down)

relative $\text{nfgaloisbasis}$

rnfidealbasis(nf, x, {f})

relative $\text{nfisbasisoalg}$

rnfisbasisoalg(nf, x, {f})

relative $\text{idealhnf}$

rnfidealh(nf, x, y, {f})

relative $\text{idealmul}$

rnfimulti(nf, x, {f})

relative $\text{idealvoet}$

rnfidealvoet(nf, x, y, {f})

PARI-GP Reference Card (2)

(PARI-GP version 2.1.0)

Projective $Z_K$-modules, maximal order

relative polred

relative polredabs

characteristic poly. of a mod $g$

rnfcharpoly(nf, g, a, {v})

relative Dedekind criterion, prime $p$

rnfisdedekind(nf, p, {f})

discriminant of relative extension

rnfisdisc(nf, x, {f})

pseudo-basis of $Z_L$

rnfisbasis(nf, {f})

relative HNF basis of order

rnfisbasisabs(nf, order)

reduced basis for order

rnfisbasisred(nf, order)

determinant of pseudo-matrix $A$

rnfdet(nf, A)

Steinitz class of order

rnfissteinitz(nf, order)

is order a free $Z_K$-module?

rnfisfree(nf, {f})

true basis of order, if it is free

rnfisbasis(nf, order)

Norms

absolute norm of ideal $x$

rnfidealnorm(nf, x, {f})

relative norm of ideal $x$

rnfidealnormreel(nf, x, {f})

solutions of $N_{K/Q}(y) = x \in Z$

rnfisnorm(nf, x, {f})

$rnfisnorm(nf, x, {f})$

is $x \in Q$ a norm from $K$?

$rnfisnorm(nf, x, {f})$

is $x \in K$ a norm from $L$?

$rnfisnorm(nf, x, {f})$

Class Groups & Units (bnf, bnr)

$a1, \{a2\}, \{a3\}$ usually bnr, subgp or bnf.module, \{subgp\}

remove GRH assumption from bnf

bnf certify(bnf)

expo. of ideal x on class gp

bnfisprincipal(bnf, x, {f})

expo. of ideal x on ray class gp

bnfisprincipal(bnf, x, {f})

expo. of x on fund. units

bnfisunit(bnf, x)

as above for S-units

bnfissunit(bnf, x)

fundamental units of bnf

bnfunit(bnf)

signs of real embeddings of bnf.fu

bnfsignunit(bnf)

Class Field Theory

ray class group structure for mod. $m$

bnrclass(bnf, m, {f})

ray class number for mod. $m$

bnrclassno(bnf, m, {f})

discriminant of class field ext

discriminants of class fields

bnrclass(bnf, l, {arch}, {f})

decode output from bnrclass(bnf, l, {arch}, {f})

is modulus the conductor?

bnrismax(bnf, l, {arch}, {f})

character conductor

bnrclassadd(bnf, l, {arch}, {f})

conductor of character chi

bnrclassadd(bnf, l, {arch}, {f})

add character conductor to bnr, chi

bnrclassadd(bnf, l, {arch}, {f})

bnrclassadd(bnf, l, {arch}, {f})

conductor of extension def. by

bnfisconductor(bnf, g)

Artin group of ext. def’d by $g$

bnrnormgroup(bnr, g)

subgroups of bnr, index <= $b$

bnrclass(bnr, b, {f})

rel. eq. for class field def’d by sub

bnrnf(kumer(bnr, sub, {d})

same, using Stark units (real field)

bnrstark(bnr, sub, {f})