

Introduction to PARI/GP

B. Allombert

IMB
CNRS/Université de Bordeaux

15/06/2022



Introduction

- ▶ PARI is a C library, allowing fast computations.
- ▶ GP is an easy-to-use interactive shell giving access to the PARI functions.
- ▶ GP is the name of gp's scripting language.
- ▶ GP2C, the GP \rightarrow PARI compiler allows to convert GP scripts to C.

Basic objects

? 57!

%1 = 40526919504877216755680601905432...

? 2 / 6

%2 = 1/3

? (1+I)^2

%3 = 2*I

? (x+1)^(-2)

%4 = 1/(x^2+2*x+1)

? Mod(2,5)^3

%5 = Mod(3,5)

? Mod(x, x^2+x+1)^3 \\ in Q[x]/(x^2+x+1)

%6 = Mod(1, x^2+x+1)

? w = ffgen([3,5], 'w); w^12 \\ in F_3^5

%7 = 2*w^4+2*w^3+2

Functions

? ?

- 1: PROGRAMMING under GP
- 2: Standard monadic or dyadic OPERATORS
- 3: CONVERSIONS and similar elementary functions
- 4: functions related to COMBINATORICS
- 5: NUMBER THEORETICAL functions
- 6: POLYNOMIALS and power series
- 7: Vectors, matrices, LINEAR ALGEBRA and sets
- 8: TRANSCENDENTAL functions
- 9: SUMS, products, integrals and similar functions
- 10: General NUMBER FIELDS
- 11: Associative and central simple ALGEBRAS
- 12: ELLIPTIC CURVES
- 13: L-FUNCTIONS
- 15: MODULAR FORMS

Help

? ?4

? ?atan

atan(x): arc tangent of x.

? ??atan

atan(x):

Principal branch of

$$\tan^{-1}(x) = \log \left(\frac{1+ix}{1-ix} \right) / 2i.$$

? ??

? ??refcard

? ??refcard-nf

? ??tutorial

? ???determinant

algdisc

bnfsunit

charker

ellpadicregulator

forsubgroup

matdet

mathnfmod

matrixqz

mspolygon

polresultant

rnfdet

See also:

Finite abelian groups

Pseudo-bases, determinant

Polymorphism

```
? \o0
```

```
? factor(91)
```

```
%61 = [7, 1; 13, 1]
```

```
? factor(x^4+4)
```

```
%62 = [x^2-2*x+2, 1; x^2+2*x+2, 1]
```

```
? factor((x^4+1)*Mod(1, a^2-2))
```

```
%63 = [x^2+Mod(-a, a^2-2)*x+1, 1; x^2+Mod(a, a^2-2)*x+1]
```

```
? factor((x^4+4)*Mod(1, 13))
```

```
%64 = [Mod(1, 13)*x+Mod(4, 13), 1; Mod(1, 13)*x+Mod(6, 13)]
```

```
? factor(x^4+1, Mod(1, a^2-2))
```

```
%65 = [x^2+Mod(-a, a^2-2)*x+1, 1; x^2+Mod(a, a^2-2)*x+1]
```

```
? factor(x^4+1, Mod(1, 13))
```

```
%66 = [Mod(1, 13)*x^2+Mod(5, 13), 1; Mod(1, 13)*x^2+Mod(
```


Comprehension

```
? [n^2|n<-[1..10]]
%72 = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
? [n^2|n<-[1..10], isprime(n)]
%73 = [4, 9, 25, 49]
? [n^2|n<-primes([1, 10])]
%74 = [4, 9, 25, 49]
? [a,b] = [1,2];
? print("a=", a, " b=", b)
% a=1 b=2
```

Control structures

- ▶ `if(cond,expr_true{,expr_false})`
- ▶ `while(cond, expr)`
- ▶ `for(var=start,end,expr(var))`
- ▶ `forstep(var=start,end,step,expr(var))`
- ▶ `forprime(var=start,end,expr(var))`
- ▶ `fordiv(N,var,expr(var))`

To configure the memory used by PARI, In the file `.gprc` (or `gprc.txt` under windows) add

```
parisizemax=1G
```

or do

```
default (parisizemax, "1G");
```

if the message 'the PARI stack overflows !' appears.