

# Introduction à PARI/GP

B. Allombert

IMB  
CNRS/Université de Bordeaux

22/11/2017



# Introduction

- ▶ PARI est une bibliothèque C, permettant des calculs rapides.
- ▶ GP est un interpréteur, donnant accès aux routines de PARI, mais bien plus simple à utiliser.
- ▶ GP est le nom du langage compris par gp.
- ▶ GP2C , le compilateur GP  $\rightarrow$  PARI permet de convertir les scripts GP en C.

## Objets de base

? 1 + 1

%1 = 2

? 57!

%2 = 40526919504877216755680601905432...

? 2 / 6

%3 = 1/3

? (1+I)^2

%4 = 2\*I

? (x+1)^(-2)

%5 = 1/(x^2+2\*x+1)

? Mod(2,5)^3

%6 = Mod(3,5)

? Mod(x, x^2+x+1)^3

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



# fonctions

? ?

- 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
- 14: MODULAR FORMS

## Aide

? ?4

? ?atan

atan(x): arc tangent of x.

? ??atan

? ?4

atan(x):

Principal branch of  $\tan^{-1}(x) = \log \left( \frac{1+ix}{1-ix} \right)$

The library syntax is GEN gatan(GEN x, long prec)

? ??

## vecteurs et matrices

```
? V = [1, 2, 3];  
? W = [4, 5, 6]~;  
? M = [1, 2, 3; 4, 5, 6];  
? V*W  
%4 = 32  
? M*W  
%5 = [32, 77]~  
? U = [1..10]  
%6 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

## composantes

```
? V[2]
```

```
%7 = 2
```

```
? W[1..2]
```

```
%8 = [4, 5]~
```

```
? M[2, 2]
```

```
%9 = 5
```

```
? M[1, ]
```

```
%10 = [1, 2, 3]
```

```
? M[, 2]
```

```
%11 = [2, 5]~
```

```
? M[1..2, 1..2]
```

```
%12 = [1, 2; 4, 5]
```



## polymorphisme

```
? factor(91)
```

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

```
? factor(91+I)
```

```
%14 = [-1, 1; 1+I, 1; 4+5*I, 1; 1+10*I, 1]
```

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

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

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

```
%16 = [x+(-1-I), 1; x+(1-I), 1; x+(-1+I), 1; x+(1+I), 1]
```

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

```
%17 = [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))
```

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

# intégration numérique

```
? intnum(x=0,1,1/(1+x^2))/Pi
%1 = 0.2500000000000000000000000000000000000000000000
? sumnum(n=1,1/n^2)/Pi^2
%2 = 0.16666666666666666666666666666666666666666667
? sumalt(n=1,(-1)^n*log(n))
%3 = 0.22579135264472743236309761494744107198
? exp(2*%)
%4 = 1.5707963267948966192313216916397514427
```

# Compréhension

```
? [n^2 | n <- [1..10]]
%5 = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
? [n^2 | n <- [1..10], isprime(n)]
%6 = [4, 9, 25, 49]
? [a, b] = [1, 2];
? print("a=", a, " b=", b)
% a=1 b=2
```

## Structure de contrôle

- ▶ `if(cond, expr_vrai{, expr_false})`
- ▶ `while(cond, expr)`
- ▶ `for(var=debut, fin, expr(var))`
- ▶ `forstep(var=debut, fin, pas, expr(var))`
- ▶ `forprime(var=debut, fin, expr(var))`
- ▶ `fordiv(N, var, expr(var))`

Pour configurer la mémoire alloué pari PARI, Dans le fichier `.gprc` (ou `gprc.txt` sous windows) ajouter

```
parisizemax=1G
```

ou faire

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

si le message 'the PARI stack overflows !' apparait.