Introduction à PARI/GP

B. Allombert

IMB
CNRS/Université de Bordeaux

23/11/2022

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 \\ Polynôme en x.
%5 = x^2+2*x+1
? Mod(2,5)^3  \\ Z/5Z
%6 = Mod(3,5)
? Mod(x, x^2+x+1)^3 \\ Q[x]/(x^2+x+1)
%7 = Mod(1,x^2+x+1)
```

Objets de base

```
? Pi
%8 = 3.1415926535897932384626433832795028842
? log(2)
%9 = 0.69314718055994530941723212145817656807
? \p100
? log(2)
%10 = 0.693147180559945309417232121458176568075500l
? exp(%)
%11 = 2.000000000000000000000000000000000000000000
? log(1+x)
%12 = x-1/2*x^2+1/3*x^3-1/4*x^4+1/5*x^5-...
? exp(%12)
%13 = 1+x+O(x^16)
```

Aide

```
? ?

   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
? ?eulerphi
%eulerphi(x): Euler's totient function of x.
? ??eulerphi
%eulerphi(x):
%
%   Euler's phi (totient) function of the integer |
%
%   ? eulerphi(40)
%   %1 = 16
? ??
? ??refcard
? ??tutorial
```

Éléments de syntaxe

```
? A = 5 + 3; \\ affectation
? A
%15 = 8
? A == 2*4 \\ égalité, 1 vrai,  0 faux
%16 = 1
? A != 2^3 \\ différent, 1 vrai,  0 faux
%17 = 0
? if (A==9-1,print("OUI"),print("NON"));
% OUI
```

Vecteurs et matrices

```
? V = [1,2,3]~; \\ vecteur colonne
? L = [4,5,6];  \\ vecteur ligne
? M = [1,2,3;4,5,6]; \\ matrice avec 2 lignes et 3
? L*V
%22 = 32
? M*V
%23 = [14,32]~
? U = [1..10]
%24 = [1,2,3,4,5,6,7,8,9,10]
```

Composantes

```
? V[2]
%25 = 2
? L[1..2]
%26 = [4,5]
? M[2,2]
%27 = 5
? M[1,]
%28 = [1,2,3]
? M[,2]
%29 = [2,5]~
? M[1..2,1..2]
%30 = [1,2;4,5]
```

Polymorphisme

Les fonctions s'adaptent au type d'entrée

```
? factor(91) \\ Z
%31 = [7,1;13,1]
? factor(91+I) \\ Z[i]
%32 = [-1,1;1+I,1;4+5*I,1;1+10*I,1]
? factor(x^4+4)  \\ Z[x]
%33 = [x^2-2*x+2,1;x^2+2*x+2,1]
? factor((x^4+4)*I)  \\ Z[i][x]
%34 = [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)) \\ Z[sqrt(2)][x]
%35 = [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)) \\ F_13[x]
%36 = [Mod(1,13)*x+Mod(4,13),1;Mod(1,13)*x+Mod(6,13
```

integration numérique

```
? \p38
   realprecision = 38 significant digits
? intnum(x=0,1,1/(1+x^2))/Pi
%37 = 0.25000000000000000000000000000000000000
? sumnum(n=1,1/n^2)/Pi^2
%38 = 0.16666666666666666666666666666666666667
? sumalt(n=1,(-1)^n*log(n))
%39 = 0.22579135264472743236309761494744107198
? exp(2*%)
%40 = 1.5707963267948966192313216916397514427
```

Compréhension

```
? [a,b] = [1,2];
? print("a=",a," b=",b)
% a=1 b=2
? [n^2|n<-[1..10]]
%43 = [1,4,9,16,25,36,49,64,81,100]
? [n^2|n<-[1..10],isprime(n)]
%44 = [4,9,25,49]
```

Structure de contrôle

- `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 allouée par 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.