Introduction to PARI/GP

B. Allombert

IMB
CNRS/Université de Bordeaux

16/04/2018

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 676541

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
%6 = Mod(1,x^2+x+1)
? a = ffgen([3,5],'a); a^12 \\ in F_3^5
%7 = 2*a^4+2*a^3+2
```

Basic objects

```
? Pi
%8 = 3.1415926535897932384626433832795028842
? log(2)
%9 = 0.69314718055994530941723212145817656807
? \p100
? log(2)
%10 = 0.693147180559945309417232121458176568075500134360255254120680009
? exp(%)
%11 = 2.000000000000000000000000000000000000000000000000000000000000000
? 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)
```

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
14: MODULAR FORMS
```

Help

```
? ?4
? ?atan

atan(x): arc tangent of x.


? ??atan

atan(x):

   Principal branch of tan^{-1}(x)  = log ((1+ix)/(

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

```
? ??
? ??refcard
? ??tutorial
? ???determinant

algdisc                bnfsunit              charker
ellpadicregulator      forsubgroup           matdet
mathnfmod              matrixqz              mspolygon
polresultant           rnfdet

See also:
  Finite abelian groups
  Pseudo-bases, determinant
```

Vectors and 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]
```

Components

```
? 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]
```

Polymorphism

```
? 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
```

Numerical integration

```
? intnum(x=0,1,1/(1+x^2))/Pi
%1 = 0.250000000000000000000000000000000000000
? sumnum(n=1,1/n^2)/Pi^2
%2 = 0.166666666666666666666666666666666666667
? sumalt(n=1,(-1)^n*log(n))
%3 = 0.225791352644727432363097614947441071 98
? exp(2*%)
%4 = 1.570796326794896619231321691639751 4427
```

Comprehension

```
? [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
```

Control structure

- `if(cond,expr_true{,expr_false})`
- `while(cond, expr)`
- `for(var=start,end,expr(var))`
- `forstep(var=start,end,pas,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) ajouter

```
parisizemax=1G
```

or do

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

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