Pari L functions

Pascal Molin

January 18, 2013

Contents

1 L functions package **1**
- 1.1 global interface functions . . . 1
- 1.2 Dirichlet coefficients or Euler factor . . . 2
- 1.3 Values . . . 2
- 1.4 Utilities . . . 3
  - 1.4.1 Guess missing conductor . . . 3
  - 1.4.2 Guess finite number of Euler factors . . . 4
  - 1.4.3 Build entirely from type and conductor . . . 4
- 1.5 I would like to update the structure (modify argument) . . . 6

2 For PARI/gp **7**
- 2.1 Series . . . 7
- 2.2 Operations on polynomials . . . 7
  - 2.2.1 Symmetric powers of a polynomial . . . 7
  - 2.2.2 `forpart` . . . 8
  - 2.2.3 reverse of `polsym` (Shönhage-Pan algorithm) . . . 8
  - 2.2.4 some kind of `dirfactor` ? . . . 8
- 2.3 Generalized exponentials and incomplete gamma . . . 8

1 L functions package

1.1 global interface functions

```
lj
L39 = Lelliptic("39a1");
Linfo(L39)
Lcheck(L39)
Lvalue(L39,1+I)
```

```
lj
L function of degree 2, level 39 and motivic weight 1
  spectral parameters: real=[], complex=[0]
  sign of functional equation = 1
  first Dirichlet coefficients = [1, 1, -1, -1, 2, -1, -4, -3, 1, 2, 4, 1, 1, -4, -2, -1, 2, 1, 0, -2]
0.99999999999999999999999999999999999999
[176ms]
1.1744992292714448165406729653628108502 + 0.37329534438807762816672745796130952932*I
```

1.2 Dirichlet coefficients or Euler factor

```
ell = ellinit("39a1");
Lfunction([[],[0]],39,1,0,vector(1000,k,ellak(ell,k)))
Lfunction([[],[0]],39,1,0,p->1-ellak(ell,p)*x+if(p%39,p*x^2,0))
\\ not yet:
Lfunction([0,1],39,1,0,k->ellak(ell,k))
```

1.3 Values

```
L1400j1 = Lelliptic([0, 1, 0, 167, 19963],5000);
Lvalue(L1400j1,[1+2*I,1+I,1+5*I])
```

```
[-0.0025728685055355290161001624886173002204 + 0.0041356335517991752692900383222210447783*I, 2.592805533
```

Any precision

```
\p 57
Lvalue(Lzeta,1+10^5*I)
##
%/zeta(1+10^5*I)
##
```

```
   realprecision = 57 significant digits
[480ms]
1.61812212284693679656775905790087726316024614070728099617 + 1.07044104147062368662603474543967113451753
  ***   last result computed in 519 ms.
1.00000000000000000000000000000000000000000000000000000000 + 0.E-57*I
  ***   last result computed in 3,352 ms.
```

Need to precompute if many values

```
L39 = Lelliptic("39a1",2000);
comp = L_int_precompute(L39,1+40*I);
plot(t=0,40,abs(L_int_fast(L39,comp,1+t*I)))
```

```
5.0709299 |''''''''''''''''''''''''''''''''''''''''''''''''''''''"''''''''|
          |           _"                  x                      :        |
          |           ::                  :                      :        |
          |           ::                  :                      :        |
          |           ::                  :                      :        |
          |           ::                  :                      :        |
          |           ::   _              :           _          :        |
          |           ::   :             ::           :          :   " _  |
          |          :  :  :             ::           :          ::  : :  |
          |          :  :  ::      x     ::           :     x    ::  : :  |
          |          :  :  ::      :     ::         "::    _:   : : :: :  |
          |          :  :  ::      :    x: :     "  :: :   ::   : : : ::  |
          |          :  : : x     ::  x :_ :     :  :: : x ::   : : x ::  |
          |          :  : : :     ::  : :  :     :  :: : : : :  : : : : : |
          |  ""      x  : : :     " : : :  :     ::: _ : : : :  : : : x : |
          | _ :      :  " :  :   "  ::: :  :    : ::   _ ::  :  : : :   : |
          |    : x   :  : :  :      ::::   :"   : ":   : ::  :  : ":    : x
          |x   " :" :    :_  :  _   :: :   ::   :  _   :: :  x  : ::    : :
          "    :: : :    x   x_     :_ :   :  : "       : :  :  :  :    x :
          |     :  ::          x    _  :   :  :"        : :   : x  :    ::|
          |     _  :"                  :   "  _         : _   :_   _     :|
        0 ,,,,,,,,,x,,,,,,,,,,,,,,,,,,,x,,,,,,,,,,,,,,,,",,,,,",,,,,,,,,,",
          0                                                              40
```

1.4 Utilities

1.4.1 Guess missing conductor

```
L5077 = Lelliptic([0, 0, 1, -7, 6]); Linfo(L5077)
Lcheck(L5077)
L5077[ilevel]=1; Linfo(L5077)
L_check_funceq(L5077)
{
  solve(level=500,10^5,
      L5077[ilevel]=level;
      [a,b]=L_theta_eq(L5077,1.1,,1);
      1-abs(a/b)
      )}
```

```
L function of degree 2, level 5077 and motivic weight 1
  spectral parameters: real=[], complex=[0]
  sign of functional equation = -1
  first Dirichlet coefficients = [1, -2, -3, 2, -4, 6, -4, 0, 6, 8, -6, -6, -4, 8, 12, -4, -4, -12, -7,
-1.000000000000000000000000000000000000001
L function of degree 2, level 1 and motivic weight 1
  spectral parameters: real=[], complex=[0]
  sign of functional equation = -1
  first Dirichlet coefficients = [1, -2, -3, 2, -4, 6, -4, 0, 6, 8, -6, -6, -4, 8, 12, -4, -4, -12, -7,
0
5076.999999999999999999999999999999999994
```

1.4.2 Guess finite number of Euler factors

```
L11a = Lelliptic("11a1",2000); L_check_funceq(L11a)
L = L_setEuler(L11a,[[2,1],[3,1]]); L_check_funceq(L)
L_guess_Euler(L,2^2*3^2)
round(%[2])
L=L_setEuler(L,%); L_check_funceq(L)
```

```
1.000000000000000000000000000000000000
0
[0.166666666666666546658031446922066537388, 0.333333333333333203614764510794903872037, 0.16666666666666587
[1.000000000000000000000000000000000000, 2.0000000000000066179220970170939003036, 1.000000000000000024716
[1, [[2, 1.99999999999999995101375987541141906908*x^2 + 2.0000000000000066179220970170939003036*x + 1.000
[[2, 2*x^2 + 2*x + 1], [3, 3*x^2 + x + 1]]
1.000000000000000000000000000000000000
```

1.4.3 Build entirely from type and conductor

Bruteforce on Dirichlet coefficients

```
L_build2_elliptic(11,1,,350,30);
```

```
[1] primes [2] coeffs [2] -> 0
[2] primes [2] coeffs [1] -> 0
[3] primes [2] coeffs [0] -> 0
[1] primes [3] coeffs [1] -> 0
[1] primes [5] coeffs [-1] -> 0.99990576123066954565147529007660997314
[1] primes [7] coeffs [2] -> 1.0000000586738188567057776265225506116
[1] primes [11] coeffs [-1] -> 1.0000000067107805351295972381582862258
[1] primes [13] coeffs [-4] -> 0.99999999999654992143599014846349647918
[1] primes [17] coeffs [2] -> 0.999999999999999994375832664989262374947
[1] primes [19] coeffs [0] -> 0.999999999999999994375832664989262374947
[1] primes [23] coeffs [1] -> 1.000000000000000000004089318930444495
[1] primes [29] coeffs [0] -> 1.000000000000000000004089318930444495
[1] primes [31] coeffs [-7] -> 1.000000000000000000000000000056850945073
```

For conductor 10, inequalities cannot be satisfied, there is no L function.

```
L_build2_elliptic(10,1,,350,30); \\ no curve
```

```
[1] primes [2] coeffs [1] -> 0
```

For conductor 26, one finds two valid solutions

```
L_build2_elliptic(26,1,,400,40); \\ two isogeny classes
```

```
[1] primes [2] coeffs [1] -> 0
[2] primes [2] coeffs [0] -> 0
[3] primes [2] coeffs [-1] -> 0
[1] primes [3] coeffs [1] -> 0.99934435141288979151697236870314659711
[2] primes [3] coeffs [3] -> 0
[3] primes [3] coeffs [0] -> 0
[4] primes [3] coeffs [-1] -> 0
[1] primes [5] coeffs [0] -> 0.99934435142464281237943254908393800242
[2] primes [5] coeffs [1] -> 1.0008803490281599751402377090894904147
[3] primes [5] coeffs [3] -> 0
[4] primes [5] coeffs [-3] -> 0
[1] primes [7] coeffs [-1] -> 0.99997433919017281514468268470664087011
[2] primes [7] coeffs [1] -> 1.0000976707575610769307109285706796336
[3] primes [7] coeffs [3] -> 0.99987393845487631958084172634232143638
[1] primes [11] coeffs [2] -> 0.99999862225660544897423066801500947002
[2] primes [11] coeffs [-6] -> 1.0000017079644139988998576805741023480
[3] primes [11] coeffs [3] -> 1.0000107642763574484915576125205316965
[1] primes [13] coeffs [1] -> 0.99999996284209074380713825029212734704
[2] primes [13] coeffs [-1] -> 0.99999994226216940571998515304843438585
[1] primes [17] coeffs [3] -> 1.0000000444942701625944852345825767704
[2] primes [17] coeffs [2] -> 0.99999999431871691015786842522101906474
[3] primes [17] coeffs [3] -> 1.0000000100570307532726500698358406839
[4] primes [17] coeffs [2] -> 0.99999998372034026719545126516903505887
[1] primes [19] coeffs [-2] -> 1.000000000001713285059298792654538795
[2] primes [19] coeffs [-6] -> 0.99999999992289675424659295443377018248
[1] primes [23] coeffs [4] -> 1.0000000000005662853194636961652525226
[2] primes [23] coeffs [0] -> 1.0000000000017132850592987926545387955
[1] primes [29] coeffs [-2] -> 1.0000000000000099486131872967841602796
[2] primes [29] coeffs [-6] -> 0.99999999999999868807712530760116178764 9
[3] primes [29] coeffs [-3] -> 0.99999999999999866086538077620043960589 3
[4] primes [29] coeffs [-5] -> 1.0000000000000176220603658718225526216
[1] primes [31] coeffs [-4] -> 1.0000000000000000089710208621587240629
[2] primes [31] coeffs [4] -> 0.99999999999999997237130890216510586562
[1] primes [37] coeffs [7] -> 0.99999999999999999999540556293103263667
[2] primes [37] coeffs [-3] -> 0.99999999999999999998247745128617104773
[1] primes [41] coeffs [0] -> 0.99999999999999999999540556293103263667
[2] primes [41] coeffs [1] -> 1.0000000000000000000164096796631841419
[3] primes [41] coeffs [0] -> 0.99999999999999999999824774512861710477 3
[4] primes [41] coeffs [1] -> 1.0000000000000000000040098033291277689 5
```

Check the number of isogeny classes for first conductors

```
{for(level=1,30,
  print("N=",level," at most ",#L_build2_elliptic(level,1,,350,30,,0))
  )}
```

```
N=1 at most 0
N=2 at most 0
N=3 at most 0
N=4 at most 0
N=5 at most 0
N=6 at most 0
N=7 at most 0
N=8 at most 0
N=9 at most 0
N=10 at most 0
N=11 at most 1
N=12 at most 0
N=13 at most 0
N=14 at most 1
N=15 at most 1
N=16 at most 0
N=17 at most 1
N=18 at most 0
N=19 at most 1
N=20 at most 1
N=21 at most 1
N=22 at most 0
N=23 at most 0
N=24 at most 1
N=25 at most 0
N=26 at most 2
N=27 at most 1
N=28 at most 0
N=29 at most 0
N=30 at most 1
```

1.5 I would like to update the structure (modify argument)

```
L = L_setEuler(L,[2,1-x])
L_setEuler(L,[2,1-x])
```

```
Lvalue(Lzeta,1+10^7*I)
Lvalue(Lzeta,.9+10^7*I) \\ same time

comp = L_int_precompute(Lzeta,1+10^7*I);
L_int_fast(Lzeta,comp,.9+10^7*I) \\ faster
```

2 For PARI/gp

2.1 Series

```
read("gpfunctions.gp.tmp");
\p 19
gammaseries(-2,5)
gammaseries(.5,7)
```

```
   realprecision = 19 significant digits
0.5000000000000000000*x^-1 + 0.4613921675492335697 + 0.9366162489878366322*x + 0.7204887516666950190*x^2
1.772453850905516027 - 3.480230906913262027*x + 7.790088721203126391*x^2 - 15.79476705153579721*x^3 + 31
```

2.2 Operations on polynomials

2.2.1 Symmetric powers of a polynomial

Let $P(T) = \alpha \prod_{i=1}^{n}(T - x_i)$. We define the symmetric r-th power of $P(T)$ to be the polynomial

$$P^{\otimes r}(T) = \alpha^d \prod_{1 \leqslant i_1 \leqslant \cdots i_r \leqslant n} (T - x_{i_1} \cdots x_{i_r}). \tag{1}$$

with

$$d = \frac{r}{n}\binom{n-1+r}{r}. \tag{2}$$

That is, the roots are all degree r homogeneous monomials in $x_1, \ldots x_n$, so that $P^{\otimes r}(T)$ is a polynomial of degree $\binom{n-1+r}{r}$ in T.

Theorem 2.1. *Let $N_k(P)$ be the Newton sums of P, and $N_k(P^{\otimes r})$ be those of $P^{\otimes r}$. Then we have*

$$N_k(P^{\otimes r}) = \sum_{r=\sum_i m_i a_i} \prod_i \frac{N_{a_i k}(P)^{m_i}}{m_i! a_i^{m_i}} \tag{3}$$

where the left sum runs over the integer partitions of r.

```
read("poloperations.gp");
polsympow(x^3-2*x+7,4)
```

```
x^15 - 4*x^14 - 228*x^13 - 13837*x^12 + 21032*x^11 + 849616*x^10 + 55805610*x^9 + 97009024*x^8 - 4222552
```

2.2.2 forpart

```
forpart(X=5,print(Vec(X)))
```

```
[1, 1, 1, 1, 1]
[1, 1, 1, 2]
[1, 2, 2]
[1, 1, 3]
[2, 3]
[1, 4]
[5]
```

```
forpart(X=7,print(Vec(X)),4)
```

```
[1, 2, 2, 2]
[1, 1, 2, 3]
[1, 1, 1, 4]
[2, 2, 3]
[1, 3, 3]
[1, 2, 4]
[1, 1, 5]
[3, 4]
[2, 5]
[1, 6]
[7]
```

2.2.3 reverse of `polsym` (Shönhage-Pan algorithm)

2.2.4 some kind of `dirfactor` ?

2.3 Generalized exponentials and incomplete gamma

Let $\gamma(s) = \prod \Gamma(\frac{s+\lambda_k}{2})$. We compute the γ-exponential

$$\exp_{(\gamma)}(-t) = \mathscr{M}^{-1}[\gamma(z);t] \tag{4}$$

and the incomplete γ function

$$t^{-s}\gamma(s,t) = \mathscr{M}^{-1}\left[\frac{\gamma(z)}{z-s};t\right] \tag{5}$$

$$= t^{-s}\int_t^\infty \exp_{(\gamma)}(-u)u^s \frac{\mathrm{d}u}{u} \tag{6}$$

Three methods:

- Taylor
- integration
- divergent series at infinity

useful outside ? interface for gp ?

```
L39 = Lelliptic("39a1");
data = L39[iinvgam];
invMellin(/*data,*/7)
2*sqrt(Pi)*exp(-2*7)
```

```
2.9476925606273000543051214593720226792 E-6
2.9476925606273000543051214593720226792 E-6
```

```
a = .5; s = 3+I; t = Euler;
initinvMellin([0,1,a,a+1]);
invMellin(t)
8*Pi*sqrt(t)^a*besselk(a,4*sqrt(t))
```

```
0.75414333925559310656602809014730500105
0.75414333925559310656602809014730500105
```