PARI/GP Atelier (13/01/2022)

[Tutorial] Hypergeometric Motives

The HGM package

Karim Belabas

Hypergeometric Motives ?

A good introduction

- David Roberts & Fernando Rodriguez Villegas, *Hypergeometric Motives*, `https://arxiv.org/abs/2109.00027`
- Frits Beukers, Henri Cohen & Anton Mellit, *Finite Hypergeometric Functions*, `https://arxiv.org/abs/1505.02900`

For the purpose of this tutorial, HGMs are a nice source of motivic L-functions (*sometimes conjecturally!*), related to point counting on families of algebraic varieties of the form

$$\prod_{i=1}^{n} x_i^{\gamma_i} = t, \quad \sum_{i=1}^{n} x_i = 0, \quad \prod_{i=1}^{n} x_i \neq 0 \ ,$$

where $(\gamma_i) \in \mathbb{Z}^n$ and $t \in \mathbb{Q}^*$ specifies a variety in the family. One can write periods in terms of classical hypergeometric functions ${}_nF_{n-1}(\alpha, \beta; t)$ and count points in terms of Jacobi sums.

One recovers in this way L-functions attached to Artin representations, curves over number fields, Siegel modular forms, etc. Example: the Legendre family of elliptic curves, $E_t : y^2 = x(x-1)(x-t)$; note that $t = 0, 1$ correspond to singular points.

Hypergeometric template (1/4): (α, β) format

A *hypergeometric template* is a pair of multisets of rational numbers $\alpha = (\alpha_1, \ldots, \alpha_d)$ and $\beta = (\beta_1, \ldots, \beta_d)$ having the same number of elements. We set

$$A(x) = \prod_{1 \leqslant j \leqslant d} (x - e^{2\pi i \alpha_j}), \quad B(x) = \prod_{1 \leqslant k \leqslant d} (x - e^{2\pi i \beta_k}) .$$

We make the following assumptions:

- $\alpha_j \not\equiv \beta_k \pmod 1$ for all j and k; or equivalently $\gcd(A, B) = 1$.
- $\alpha_j \notin \mathbb{Z}$ for all j; or equivalently $A(1) \neq 0$.
- our template is *defined over* $\mathbb{Q}$, in other words $A, B \in \mathbb{Z}[x]$; or equivalently if some a/D with $\gcd(a, D) = 1$ occurs in the α_j or β_k, then all the b/D modulo 1 with $\gcd(b, D) = 1$ also occur.

Hypergeometric templates (2/4): cyclotomic and γ formats

The *defined over* $\mathbb{Q}$ assumption allows to abbreviate each occurence of $[a_1/D, \ldots, a_{\varphi(D)}/D]$ (where the a_i range in $(\mathbb{Z}/D\mathbb{Z})^*$) to $[D]$. We have three possible ways of giving a hypergeometric template:

- by the two GP vectors $[\alpha_1, \ldots, \alpha_d]$ and $[\beta_1, \ldots, \beta_d]$ (*α, β parameters*),
- or by their denominators $[D_1, \ldots, D_m]$ and $[E_1, \ldots, E_n]$ (*cyclotomic parameters*); note that $\sum_j \varphi(D_j) = \sum_k \varphi(E_k) = d$.
- a third and final way is to give the *gamma vector* (γ_n) defined by $A(X)/B(X) = \prod_n (X^n - 1)^{\gamma_n}$, which satisfies $\sum_n n\gamma_n = 0$.

To any such data we associate a *hypergeometric template* using the function `hgminit`; then the α_j and β_k are obtained using `hgmalpha`, cyclotomic parameters using `hgmcyclo` and the gamma vectors using `hgmgamma`.

N.B. $\beta = (0, \ldots, 0)$ or $E = (1, \ldots, 1)$ can be omitted in (α, β) and cyclotomic formats, respectively.

Hypergeometric templates (3/4) : example

To such a hypergeometric template is associated a number of additional parameters: the degree d, the (motivic) *weight* w, a *Hodge polynomial* P, a *Tate twist* T, and a normalizing M-factor $M = \prod_n n^{n\gamma_n}$. The `hgmparams` function returns

$$[d, w, [P, T], M] \ .$$

Example with cyclotomic parameters $[5], [1, 1, 1, 1]$:

```
?   H = hgminit([5]);
?   hgmparams(H)
%2 = [4, 3, [x^3+x^2+x+1,0], 3125]
?   hgmalpha(H)
%3 = [[1/5, 2/5, 3/5, 4/5], [0, 0, 0, 0]]
?   hgmcyclo(H)
%4 = [Vecsmall([5]), Vecsmall([1, 1, 1, 1])]
?   hgmgamma(H)
%5 = Vecsmall([-5,0,0,0,1])   \\  A/B = (x^5-1) / (x-1)^5
```

Hypergeometric templates (4/4) : example

```
?  H2 = hgminit([2,3,4],[1,5]);
?  hgmparams(H2)
%7 = [5, 2, [x^2 + 3*x + 1, 1], 6912/3125]
?  hgmalpha(H2)
%8 = [[1/4, 1/3, 1/2, 2/3, 3/4], [0, 1/5, 2/5, 3/5, 4/5]]
?  hgmgamma(H2)
%9 = Vecsmall([-2, 0, 1, 1, -1])
```

Motives (1/3)

A *hypergeometric motive* (HGM) is a pair (H, t), where H is a hypergeometric template and $t \in \mathbb{Q}^*$. For $t \neq 1$, this data is (conjecturally) attached to a pure motive M of weight w, essentially the middle cohomology group of some algebraic variety. Traces of Frobenius on M_ℓ, are given by an explicit formula (Katz) involving Jacobi sums, equivalently by a finite hypergeometric sum evaluated at t: for each finite field $\mathbb{F}_q$, we can compute an *integer* $N_q(H, t) = \mathrm{Tr}(\mathsf{Fr}_q \mid (H, t))$.

This formula only makes sense for *good primes* p; there are two kinds of *bad primes*:

- p is *wild* if it divides a denominator of the α_j or β_i (equivalently, one of the cyclotomic parameters)
- else it is *tame* if $v_p(t) \neq 0$ or $v_p(t-1) \neq 0$.

Motives (2/3)

The *local Euler factor* at a good prime p is then given by the (inverse of the) usual formula

$$P_p(T) = \exp\left(-\sum_{f\geq 1}\frac{N_{p^f}(H,t)}{f}T^f\right)\ ,$$

always a polynomial of degree d. N.B. the Euler factor L_p used in the global L-function is $1/P_p(p^{-s})$. The formulas is modified for tame primes or for $t=1$ (Roberts, Rodriguez Villegas, Watkins,...) and *usually* $\deg P_p < d$ in this case.

Various recipes are *conjectured* for wild primes (often $L_p \equiv 1$) but we did not implement them. On the other hand L_p can be *guessed* via the global functional equation: once a global L-function is computed, we can obtain Euler factor at *any* prime, using `lfuneuler`.

Motives (3/3)

```
?  hgmeulerfactor(H, t = -1, p = 3)   \\  good prime
%10 = 729*x^4 + 135*x^3 + 45*x^2 + 5*x + 1
?  hgmeulerfactor(H, -1, 2)   \\  tame prime
%11 = 16*x^3 + 6*x^2 + x + 1
?  hgmeulerfactor(H, -1, 5)   \\  wild primes not implemented
%12 = 0
?  hgmeulerfactor(H, 1/3, 3)   \\  tame prime
%13 = -x + 1
?  hgmeulerfactor(H, 1/3, 2)   \\  tame prime
%14 = 16*x^3 + 6*x^2 + x + 1

\\For H2 now:  2,3,5 are wild
?  hgmeulerfactor(H2, 2, 7)   \\  good prime
%15 = 16807*x^5 - 2401*x^4 + 294*x^3 + 42*x^2 - 7*x + 1
?  hgmeulerfactor(H2, 1/8, 7)   \\  tame prime
%16 = -2401*x^4 - 343*x^3 + 7*x + 1
```

The global L-function (1/2)

If one suitably defines $P_p(T)$ for *all* primes p including the wild ones, then the L-function defined by

$$L(H,s)=\prod_p P_p(p^{-s})^{-1}$$

is motivic (Katz), with analytic continuation and functional equation, as used in the L-function package of `Pari/GP`. If the motivic weight w is even, there is a possible (multiple) pole at $w/2+1$.

The command `L = lfunhgm(H, t)` creates such an L-function. In particular it must guess the local Euler factors at wild primes, which can be very expensive when the conductor `lfunparams`$(L)[1]$ or the degree d is large. This L-function can then be used with all the functions of the `lfun` package. For instance we can now obtain the global conductor and check the Euler factors at all bad primes.

In our example, `lfunhgm(H, 1/2)` is very fast (only 5 is wild and the conductor is 5000). More complicated, `L = lfunhgm(H, 1/64)` finishes in about 20 seconds (the conductor is 525000).

The global L-function (2/2)

```
?  [N] = lfunparams(L); N \\the conductor
%17 = 525000
?  print(factor(N))
%18 = [2, 3; 3, 1; 5, 5; 7, 1]
?  lfuneuler(L,2)
%19 = 1/(-x + 1)
?  lfuneuler(L,3)
%20 = 1/(81*x^3 + 6*x^2 - 4*x + 1)
?  lfuneuler(L,5)
%21 = 1
?  lfuneuler(L,7)
%22 = 1/(2401*x^3 + 301*x^2 + x + 1)
```

Coefficients of the L-function

Two additional functions related to the global L-function are available which do *not* require its full initialization: `hgmcoefs(H,t,n)` computes the first n coefficients of the L-function by setting all wild Euler factors to 1; this is identical to `lfunan(L,n)` when this hypothesis is correct (in particular if there are no wild primes) otherwise all coefficients divisible by a wild prime will be wrong. In the above example, only 5 is wild and L_5 is indeed trivial, so all is fine.

The second is the function `hgmcoef(H,t,n)` which only computes the nth coefficient of the global L-function. It gives an error if n is divisible by a wild prime.

```
?  hgmcoefs(H, 1/64, 7^6)[7^6]   \\  slow:  7^6 > 10^6
time = 1min, 1,564 ms.
%26 = -25290600
?  hgmcoef(H, 1/64, 7^6)
%27 = -25290600
?  hgmcoef(H, 1/64, 10)
```