[Tutorial] Dirichlet characters

Karim Belabas

Institut de
Mathématiques de
Bordeaux

Generic abelian characters

In PARI/GP, given a *finite* abelian group

$$G = (\mathbb{Z}/o_1\mathbb{Z})g_1 \oplus \cdots \oplus (\mathbb{Z}/o_d\mathbb{Z})g_d,$$

with fixed generators g_i of respective order o_i, then

- the *column* vector $[x_1, \dots, x_d]$~ represents the element $g \cdot x := \sum_{i \leqslant d} x_i g_i$;
- the *row* vector $[c_1, \dots, c_d]$, represents the character mapping $g_i \mapsto e(c_i/o_i)$ for each i. The trivial character is $[0, \dots, 0]$.

The group G is given by a GP structure, e.g. `bid`, `bnf`, `bnr`. We can choose $(g_i) := G$`.gen` (SNF generators), hence $(o_i) = G$`.cyc` and $o_d \mid \cdots \mid o_1$ (elementary divisors).

Generic functions (1/3)

`charorder(G, chi)` \\ order of χ in $\hat{G}$

`charmul(G, chi, psi)` \\ $\chi \cdot \psi$

`chardiv(G, chi, psi)` \\ $\chi \cdot \psi^{-1}$

`charconj(G, chi)` \\ $\chi^{-1} = \overline{\chi}$

Try it for instance on

```
G = idealstar(,100)
G.cyc
chi = [1, 0]
psi = [1, 1]
```

Generic functions (2/3)

`charker(G, chi)` \\ the subgroup $H = \operatorname{Ker}\chi$

This returns a matrix whose columns give generators h_j of H (in terms of the fixed g_i)

`chareval(G, chi, x)` \\ $c/n \in \mathbb{Q}$ such that $\chi(x) = e(c/n)$,

This first maps x to G using a discrete logarithm: $x = \sum x_i \cdot g_i$ (`znlog(x, G)`). Return the sentinel value -1 if x is not in G, e.g.

```
G = idealstar(,100); chi = [1, 0];
chareval(G, chi, 0)
```
\\ $0 \notin (\mathbb{Z}/100\mathbb{Z})^*$

Generic functions (3/3)

Characters with values in arbitrary fields:

`chareval(G, chi, x, [z,o])` Assume that the integer o is a multiple of the order of χ and that z is an element in the multiplicative group of some field. Returns $\chi(x) = z^{o \cdot c/n}$. If $z = e(1/o) \in \mathbb{C}$, this is $e(c/n)$ as before. This time the sentinel value for $x \notin G$ is 0. As in the extension of Dirichlet characters from $(\mathbb{Z}/N\mathbb{Z})^*$ to $\mathbb{Z}$.

It is also possible to replace z with a vector containing its precomputed successive powers

```
[ z^i | i <- [0..o-1] ]
```

Functions specific to Dirichlet characters

We must have $G = \texttt{idealstar}(,\texttt{N})$ for some positive integer modulus N.

`znchar isodd(G, chi)`: returns 1 if $\chi(-1) = -1$ and 0 otherwise.

`znchartokronecker(G, chi)`: returns D if χ is real and equal to $(D/.)$; D is fundamental if and only if χ is primitive. ($D < 0$ if and only if χ is odd.)

`zncharinduce(G, chi, Q)`: assume that $N \mid Q$; returns the induced character on $(\mathbb{Z}/Q\mathbb{Z})^*$ in terms of *canonical* generators of that group. Which is not initialized!

Canonical generators

We started from SNF generators

$$G = (\mathbb{Z}/o_1\mathbb{Z})g_1 \oplus \cdots \oplus (\mathbb{Z}/o_d\mathbb{Z})g_d,$$

with $o_d \mid \cdots \mid o_1$. But it is possible to choose other generators !

If $G = (\mathbb{Z}/N\mathbb{Z})^*$, $N = \prod_p p^{e_p}$, we can choose canonical generators of the $(\mathbb{Z}/p^{e_p}\mathbb{Z})^*$ (smallest generator of $\mathbb{Z}_p^*$ for p odd; -1 and 5 for $p = 2$) and build from there via CRT. We obtain Conrey generators for G: $\tilde{g}_1, \ldots, \tilde{g}_d$ of order $\tilde{o}_i$. We no longer have $\tilde{o}_d \mid \cdots \mid \tilde{o}_1$.

A character given in terms of the $\tilde{g}_i$ is denoted by $[c_1, \ldots, c_d]^{\sim}$, which maps $\tilde{g}_i$ to $e(c_i/\tilde{o}_i)$ for all i. We call it a Conrey character.

The discrete log of $x \in (\mathbb{Z}/N\mathbb{Z})^*$ in terms of the Conrey generators is `znconreylog(G, x)`.

Conrey characters (1/2)

The map $x \in G = (\mathbb{Z}/N\mathbb{Z}) \mapsto$ `znconreylog(G,x)` is an isomorphism from G to $\hat{G}$.

```
G = idealstar(,100);
chi = znconreylog(G, 3)
znconreyexp(G, chi)
znconreychar(G, chi)
```

To sum up, we can represent a Dirichlet character χ mod N in the following formats:

- generic character: a `t_VEC` $[c_1, \dots, c_d]$ such that $\chi(g_i) = e(c_i/o_i)$;
- Conrey character: a `t_COL` $[\tilde{c}_1, \dots, \tilde{c}_d]^\sim$ $\chi(\tilde{g}_i) = e(c_i/\tilde{o}_i)$;
- Conrey label: a `t_INT` m whose Conrey log is $[\tilde{c}_1, \dots, \tilde{c}_d]^\sim$.

Given a character in any form, `znconreychar` gives the `t_VEC`, `znconreylog` gives the `t_COL`, and `znconreyexp` gives the `t_INT`.

Conrey characters (2/2)

Writing $\chi = \prod_p \chi_p$ or decomposing $\chi = \chi_Q \cdot \chi_{N/Q}$ for $Q \,||\, N$ is trivial for Conrey characters (`kb-mf` branch). One can induce characters, or compute a character conductor and the attached primitive character without initializing the `idealstar` corresponding to the new modulus !

```
N  = 100; G  = idealstar(, N); chi = [2, 0];
N2 = 900; G2 = idealstar(,N2);
chi2 = zncharinduce(G, chi, N2) \\ or G2
[chareval(G,chi,x)   | x <- [1..25], gcd(x,N2) == 1]
[chareval(G2,chi2,x) | x <- [1..25], gcd(x,N2) == 1]

znconreyconductor(G2, chi2)
znconreyconductor(G2, chi2, &chi0)
chi0
znconreyconductor(G, chi, &chi0)
chi0
```

A fun general alternative

```
N = 100;
bnr = bnrinit(bnfinit(x), [N,[1]]);
g = [3, 7]
znorder(Mod(g[1], N))
znorder(Mod(g[2], N))
```

`bnrchar(bnr, g)`: finds all characters that are trivial on the given g_i;

```
v = [1/10, 1/2]
```

`bnrchar(bnr, g, v)`: finds all characters s.t. $\chi(g_i) = e(v_i)$, assuming that the order of g_i divides the denominator of v_i for all i.