HAL
open science

The development version of the CHEVIE package of GAP3

Jean Michel

▶ To cite this version:

Jean Michel. The development version of the CHEVIE package of GAP3. 2013. hal-00877067

HAL Id: hal-00877067
https://hal.science/hal-00877067v1

Preprint submitted on 25 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THE DEVELOPMENT VERSION OF THE CHEVIE PACKAGE OF GAP3

JEAN MICHEL

1. Introduction

The published version 3 of the CHEVIE package was released with the last version of GAP3 in 1997. The paper [GHLMP1996] documents the state of the package in 1994; at the time the main authors of the package were M. Geck, G. Hiss, F. Lübeck, G. Malle, and G. Pfeiffer. The author started working on CHEVIE in 1995, motivated by collaboration with M. Geck on determining characters of Iwahori-Hecke algebras. For that he developed programs to compute in Artin-Tits braid groups, and improved the way reflection subgroups of Coxeter groups were handled. Prompted by the author's interest in complex reflection groups, he also implemented basic routines dealing with them and the associated cyclotomic Hecke algebras. This is the extent of his work on CHEVIE at the publication time in 1997.

Around that time, the author also started a collaboration with F. Lübeck to implement arbitrary reductive groups and their unipotent characters. This collaborative work lasted until roughly 2000, at which time the author became the main developer of the CHEVIE package, motivated by his research themes and collaborations with various people, including among others D. Bessis, C. Bonnafé, M. Broué, M. Chlouveraki, F. Digne, J. Gonzalez-Meneses, B. Howlett, G. Malle, I. Marin and R. Rouquier.

The latest CHEVIE version, described here, is available on the author's webpage [Michel], together with convenient facilities to install it bundled with GAP3.

Here is a list of some of the additional facilities provided by the CHEVIE package since 1997:

- Affine Weyl groups and general Coxeter groups, and the corresponding Hecke algebras with their Kazhdan-Lusztig bases and polynomials. The author implemented this via "generic" support for Coxeter groups, writing code in terms of a limited set of primitives `FirstLeftDescending`, `LeftDescentSet`, etc..., definable for arbitrary Coxeter groups whatever the representation (for instance, Affine Weyl group elements are represented as matrices, instead of the permutations used for finite Coxeter groups).
- Kazhdan-Lusztig polynomials and bases for unequal parameter Hecke algebras. Hecke modules on Hecke algebras for general Coxeter groups, including the bases defined by Deodhar and Soergel for these modules.
- Reflection cosets for arbitrary complex reflection groups, together with their automatic classification. This includes the possibility of defining Coxeter Cosets corresponding to the "very twisted" Ree and Suzuki groups of Lie type. Quite a few methods work now for arbitrary finite complex reflection groups and cosets, such as type recognition (decomposition into a product of

recognized irreducible groups), so routines for character tables, for instance, have become fast and easy to read using such decompositions.

- Complete lists of representations for Hecke algebras of all finite Coxeter groups, using Howlett's work for the big exceptional groups. An almost complete list of representations for cyclotomic Hecke algebras for finite complex reflection groups; most of this work was done jointly with G. Malle, see [Malle-Michel2010]. Currently are missing a few representations for G_{29} and most representations for groups in the range G_{31} to G_{34}. Similarly, there are partial character tables for these cyclotomic Hecke algebras. The character table for the algebra of G_{29} is complete; the list of representations of the reflection groups themselves are complete except for G_{34}.
- Complete lists of polynomial invariants for all finite complex reflection groups.
- General Garside and locally Garside monoids. This includes braid monoids for general Coxeter groups, dual braid monoids for finite Coxeter groups and well-generated complex reflection groups. There are algorithms to determine conjugacy sets and compute centralizers, implementing the work of [Gonzalez-Gebhardt2010].
- Semisimple elements of reductive groups, including the computation of centralizers, and determining the list of isolated and quasi-isolated classes.
- Unipotent characters for reductive groups, their Lusztig induction and Lusztig's Fourier transform have been implemented, as well as $\mathcal{L}$ functions attached to Deligne-Lusztig varieties. The above has also been implemented for "Spetses" attached to complex reflection groups.
- Unipotent classes of reductive groups (including the bad characteristic case) and the generalized Springer correspondence and Green functions. The Maple part of the `CHEVIE` package dealing with Green functions has become obsolete, since the corresponding computations can now be handled more conveniently within `GAP`.
- Systematic methods for formatting objects, in order to display them nicely or export them in TeX form or in Maple form.
- Some support for posets.

One may ask why this package was developed in `GAP3`, and not `GAP4`, which is the released version of `GAP` since 1999; the reason is that the authors of the package made considerable use of generic programming facilities (the "type system") in `GAP3`, which is incompatible with `GAP4`; the package represents a considerable investment of programming time, and the author is not yet willing to stop his research for one year, which is the minimum time which would be needed to port the package to `GAP4`. The main limitation that `GAP3` imposes is the limitation of memory to 2 gigabytes, which may some day be motivation enough for a port; but this port might as well be to another system like `sage` (see [Sage]), which can already use `CHEVIE` through its `GAP3` interface. We mention [Geck2012] where Geck has ported to Python (and substantially improved, using new mathematics) the `CHEVIE` facilities for Kazhdan-Lustig cells.

The rest of this paper introduces the package by giving some examples of its use in braid groups and algebraic groups. This covers only a small amount of the available facilities in `CHEVIE`. However the coverage is detailed in the sense that we give complete `CHEVIE` code for each of the considered problems. The code in

CHEVIE has already been used intensively in proving several important results, for instance in [Bonnafe-Michel2011], [Kessar-Malle2013], ...

2. Braid groups

We begin by looking at a conjecture that Lusztig formulated in [Lusztig2011] about the (possibly twisted) centralizer of some elements in the braid group.

Let (W, S) be a finite Coxeter system and let V be its reflection representation, a real vector space on which the elements of S act by reflections. We consider the 'twisted" situation where we are given in addition an automorphism $\sigma \in \mathrm{GL}(V)$ normalizing W (this situation is motivated by the study of non-split reductive groups). Such an automorphism is called a *diagram automorphism* since we may choose σ up to an inner automorphism such that it stabilizes S. This situation defines a *Coxeter coset* $W\sigma$. A *conjugacy class* of $W\sigma$ is a W-orbit for the conjugation action, and the *centralizer* of $x\sigma \in W\sigma$ is the set $\{w \in W \mid wx\sigma = x\sigma w\}$. All these notions generalize straightforwardly the case where $\sigma = \mathrm{Id}$.

A *standard parabolic subgroup* is a subgroup of W generated by a subset $J \subset S$. A conjugacy class of W (resp. of $W\sigma$) is called *elliptic* if it does not meet any proper subgroup W_J (resp. any proper subcoset $W_J\sigma$ where $\sigma(J) = J$).

If the presentation of W as a Coxeter group is

$$W = \langle S \mid s^2 = 1, \underbrace{sts\dots}_{m_{s,t}} = \underbrace{tst\dots}_{m_{s,t}} \text{ for } s, t \in S\rangle$$

then the *Artin-Tits braid group* attached to W is defined by the presentation $B^+ = \langle \mathbf{S} \mid \underbrace{\mathbf{sts}\dots}_{m_{s,t}} = \underbrace{\mathbf{tst}\dots}_{m_{s,t}} \text{ for } \mathbf{s}, \mathbf{t} \in \mathbf{S}\rangle$, where $\mathbf{S}$ is a copy of S, whose relations are called the *braid relations*. There is an obvious quotient map $B \to W$ since the relations of B are relations in W; Matsumoto's lemma, stating that two reduced expressions for an element of W can be related by using only braid relations, implies that there is a well-defined section of the quotient which maps a reduced expression $w = s_1 \dots s_n$ to the product $\mathbf{s}_1 \dots \mathbf{s}_n \in B$.

Since B is generated by a copy $\mathbf{S}$ of S, the automorphism σ extends naturally to B. The following result was proved for the Weyl groups of classical groups in [Lusztig2011], and later given a general proof in [He-Nie2012]. In the meanwhile the author could check it for exceptional finite Coxeter groups using CHEVIE.

Theorem 1. *[Lusztig, He-Nie] Let w be an element of minimal length of an elliptic conjugacy class of W (resp. of $W\sigma$). Let $\mathbf{w}$ be the lift of w to B. Then the map $C_B(\mathbf{w}) \to C_W(w)$ (resp. $C_B(\mathbf{w}\sigma) \to C_W(w\sigma)$) is surjective.*

Actually, one may conjecture that there is a strong structural relationship between the groups $C_B(\mathbf{w}\sigma)$ and $C_W(w\sigma)$: in most cases, the second is a complex reflection group and the first should be the corresponding braid group.

We will show the code in CHEVIE to check theorem 1 for a Coxeter coset of type ${}^2E_6$. We first construct the Coxeter group:

```
gap> W:=CoxeterGroup("E",6);;
gap> PrintDiagram(W);
E6      2
        |
1 - 3 - 4 - 5 - 6
```

In GAP, the result of a command which ends with a double semicolon is not printed. The command PrintDiagram shows the numbering the elements of S. We now specify the coset by giving the permutation that the automorphism σ does on the elements of S.

```
gap> WF:=CoxeterCoset(W,(1,6)(3,5));
2E6
```

We now need information on the conjugacy classes of the coset. There is a Chevie function which does that for arbitrary complex reflection groups or associated reflection cosets.

> ChevieClassInfo(W)
>
> This function returns a record containing information about the conjugacy classes of the finite reflection group or coset W. If the argument is a coset WF, where W denotes the reflection group and F the automorphism, the classes are defined as the W-orbits on WF for the conjugation action.
>
> The result is a record which contains among others the following fields:
>
> .classtext: words in the generators of W which define representatives of the conjugacy classes. These representatives are of minimal length and 'very good" in the sense of [Geck-Michel1997].
>
> .classnames: names for the conjugacy classes.
>
> .classes: sizes of the conjugacy classes

```
gap> ChevieClassInfo(CoxeterGroup("A",3));
rec(
  classparams := [ [ [ 1, 1, 1, 1 ] ], [ [ 2, 1, 1 ] ],
      [ [ 2, 2 ] ], [ [ 3, 1 ] ], [ [ 4 ] ] ],
  classnames := [ "1111", "211", "22", "31", "4" ],
  classtext := [ [  ], [ 1 ], [ 1, 3 ], [ 1, 2 ], [ 1, 3, 2 ] ],
  classes := [ 1, 6, 3, 8, 6 ],
  orders := [ 1, 2, 2, 3, 4 ] )
```

It has been proven by Howlett that a minimal length representative of a non-elliptic conjugacy class lies actually in a proper standard parabolic subgroup. Thus we can find the indices of the elliptic classes by testing if the minimal word for a representative contains an element of each F-orbit on S.

```
gap> cl:=ChevieClassInfo(WF).classtext;;
gap> elliptic:=Filtered([1..Length(cl)],i->
>   ForAll([[1,6],[3,5],[2],[4]],I->Intersection(cl[i],I)<>[]));
[ 1, 4, 5, 6, 7, 9, 10, 14, 15 ]
```

We now construct the lifts in the braid group of theses representatives by using the function Braid(W) which takes as argument a sequence of indices in S and returns the corresponding element of the braid group:

```
gap> B:=Braid(W);
function ( arg ) ... end
gap> cl:=List(cl{elliptic},B);
[ w0, 124315436543, 1231431543165431, 23423465423456, 142314354231465431,
  45423145, 4254234565423456, 1254, 123143 ]
```

The lift of the longest element of W is printed in a particular way as `w0` since it is the *Garside element* of the braid monoid. The braid monoid is a *Garside monoid*, that is a cancellative monoid which has a special element called the Garside element such that its set of left and righ-divisors coincide, generate the monoid, and form a lattice for divisibility. We now use the function

> `CentralizerGenerators(b[,F])`
>
> The element `b` should be an element of a Garside group. The function returns a list of generators of the centralizers of `b`, using the algorithm of Gebhardt and Gonzalez-Meneses.
>
> If an argument `F` is given it should be the automorphism of a reflection coset attached to the same group to which the Garside monoid is attached. Then the `F`-centralizer is computed, defined as the elements `x` such that $xb = bF(x)$.

In the above, the "automorphism of a reflection coset" can be obtained for a coset `WF` by the call `Frobenius(WF)`, which returns a `GAP` function which knows how to apply the automorphism F to various objects attached to W: words, elements, braids. . ..

```
gap> F:=Frobenius(WF);
function ( arg ) ... end
gap> cc:=List(cl,x->CentralizerGenerators(x,F));
[ [ 6, 5, 4, 2, 3, 1 ], [ 65, 124315436543, (3)^-1.24315436543,
    (4)^-1.1243654, (43)^-1.243654, 13, (5)^-1.12431543654,
    (35)^-1.2431543654, (45)^-1.124354, (435)^-1.24354 ],
  [ 6, 5, 2431543654, 3, 1 ],
  [ 4, 2, 342542345, 34265423145, (4354265431)^-1.314354265431,
      (542345)^-1.65423145 ],
  [ (5)^-1.1435, (456)^-1.154234565, 142314356, (4356)^-1.5423456 ],
  [ 4, 13454231435426 ],
  [ 42, 45426542314354265431, 5423, (2)^-1.5426542314354265431,
      (3)^-1.56542314354265431, (3)^-1.543, (43)^-1.6542314354265431 ],
  [ 12542346 ], [ 123465, 123142354654 ] ]
```

In the above, elements of the braid group are printed as "reduced fractions" `(a)^-1.b` where `a` and `b` are elements of the braid monoid which have no common left divisor. Some of the generating sets can be simplified:

```
gap> cc{[2,4,5,6]}:=List(cc{[2,4,5,6]},ShrinkGarsideGeneratingSet);
[ [ 65, 13, (4)^-1.1243654 ], [ 4, 2, 342542345, 34265423145 ],
  [ (5)^-1.1435, 142314356 ], [ 42, 5423, (3)^-1.56542314354265431 ]]
```

It is now straightforward to finish the computation. To check that $C_B(\mathbf{w}\sigma) \to C_W(w\sigma)$ is surjective, we compute the size of the image, using the function `EltBraid` which computes the quotient $B \to W$:

```
gap> List(cc,x->Size(Subgroup(W,List(x,EltBraid))));
[ 51840, 648, 216, 108, 96, 10, 72, 9, 12 ]
```

And we compare with the size of $C_W(w\sigma)$ that we can compute two ways: using the field `.classes` of `ChevieClassInfo`

```
gap> List(ChevieClassInfo(WF).classes{elliptic},x->Size(W)/x);
[ 51840, 648, 216, 108, 96, 10, 72, 9, 12 ]
```

or asking directly for the size of the centralizer:

```
gap> List(ChevieClassInfo(WF).classtext{elliptic},
>   x->Size(Centralizer(W,EltWord(WF,x))));
[ 51840, 648, 216, 108, 96, 10, 72, 9, 12 ]
```

3. Representing reductive groups

We now describe how to work with reductive groups in CHEVIE. We first look at the case of connected groups; a connected reductive group $\mathbf{G}$ over an algebraically closed field $\mathbb{F}$ is determined up to isomorphism by the *root datum* $(X(\mathbf{T}), \Phi, Y(\mathbf{T}), \Phi^\vee)$ where $\Phi \subset X(\mathbf{T})$ are the roots with respect to the maximal torus $\mathbf{T}$ and $\Phi^\vee \subset Y(\mathbf{T})$ are the corresponding coroots. This determines the Weyl group, a finite reflection group $W \subset \mathrm{GL}(Y(\mathbf{T}))$.

In CHEVIE, to specify $\mathbf{G}$, we give an integral matrix R whose lines represent the simple roots in terms of a basis of $X(\mathbf{T})$, and an integral matrix $R^\vee$ whose lines represent the simple coroots in terms of a basis of $Y(\mathbf{T})$. It is assumed that the bases of $X(\mathbf{T})$ and $Y(\mathbf{T})$ are chosen such that the canonical pairing is given by $\langle x, y\rangle_{\mathbf{T}} = \sum_i x_i y_i$.

For convenience, two particular cases are implemented in CHEVIE where the user just has to specify the Coxeter type of the Weyl group. If $\mathbf{G}$ is adjoint then R is the identity matrix and $R^\vee$ is the Cartan matrix of the root system given by $\{\alpha^\vee(\beta)\}_{\alpha,\beta}$ where $\alpha^\vee$ (resp. β) runs over the simple coroots (resp. simple roots). If $\mathbf{G}$ is semisimple simply connected, then the dual group $\mathbf{G}^*$ is adjoint thus the situation is reversed: $R^\vee$ is the identity matrix and R the transposed of the Cartan matrix. In all cases, the function we use constructs a particular integral representation of a Coxeter group, so it is called CoxeterGroup.

By default, the adjoint group is returned. For instance, the group PGL_3 is obtained by

```
gap> PGL:=CoxeterGroup("A",2);
CoxeterGroup("A",2)
gap> PGL.simpleRoots;
[ [ 1, 0 ], [ 0, 1 ] ]
gap> PGL.simpleCoroots;
[ [ 2, -1 ], [ -1, 2 ] ]
```

To get the semisimple simply connected group SL_3, the additional parameter "sc" has to be given:

```
gap> SL:=CoxeterGroup("A",2,"sc");
CoxeterGroup("A",2,"sc")
gap> SL.simpleRoots;
[ [ 2, -1 ], [ -1, 2 ] ]
gap> SL.simpleCoroots;
[ [ 1, 0 ], [ 0, 1 ] ]
```

To get GL_3 we must use the general form by giving R and $R^\vee$:

```
gap> GL := CoxeterGroup( [ [ -1, 1, 0], [ 0, -1, 1 ] ],
> [ [ -1, 1, 0], [ 0, -1, 1 ] ] );;
[ [ -1, 1, 0 ], [ 0, -1, 1 ] ]
gap> GL.simpleCoroots;
[ [ -1, 1, 0 ], [ 0, -1, 1 ] ]
```

`RootDatum(type[,index])`
For convenience, there is also a function `RootDatum` which understands some familiar names for algebraic groups (like `"halfspin"`) and does the appropriate call to `CoxeterGroup`.

For instance, the above call is equivalent to

```
gap> GL := RootDatum("gl",3);
```

4. Computations with semisimple elements using CHEVIE

We present briefly some of the CHEVIE facilities for computing with semisimple elements of finite order in reductive groups, with the programs for checking some lemmas which were used in [Bonnafe-Michel2011].

Let $\mathbf{S}$ be a torus defined over $\mathbb{F}$. The map $\mathbb{F}^\times \otimes_{\mathbb{Z}} Y(\mathbf{S}) \to \mathbf{S}$ given by $x \otimes \lambda \mapsto \lambda(x)$ is an isomorphism, where we identify $\mathbf{S}$ to the group of its points over $\mathbb{F}$. If $\mathbb{F} = \mathbb{C}$ we may choose an isomorphism between the elements of finite order in $\mathbb{F}^\times$ (the roots of unity) and $\mathbb{Q}/\mathbb{Z}$. If $\mathbb{F}$ is an algebraic closure of the finite field $\mathbb{F}_p$, we may choose an isomorphism $\mathbb{F}^\times \simeq (\mathbb{Q}/\mathbb{Z})_{p'}$. In these cases we get thus an isomorphism between $(\mathbb{Q}/\mathbb{Z}) \otimes_{\mathbb{Z}} Y(\mathbf{S})$ and the points of finite order of $\mathbf{S}$ (resp. $(\mathbb{Q}/\mathbb{Z})_{p'} \otimes_{\mathbb{Z}} Y(\mathbf{S}) \xrightarrow{\sim} \mathbf{S}$). Thus, if $\dim \mathbf{S} = r$, an element of $\mathbf{S}$ can be represented by an element of $(\mathbb{Q}/\mathbb{Z})^r$ as soon as we choose a basis of $Y(\mathbf{S})$. If $\mathbf{S}$ is a subtorus of $\mathbf{T}$, then the inclusion $\mathbf{S} \subset \mathbf{T}$ is determined by giving a basis of the sublattice $Y(\mathbf{S})$ inside $Y(\mathbf{T})$.

These are the basic ideas used to represent semisimple elements in CHEVIE.

We recall that a semisimple element s of $\mathbf{G}$ is *isolated* (resp. *quasi-isolated*) if $C_{\mathbf{G}}(s)^0$ (resp. $C_{\mathbf{G}}(s)$) does not lie in a proper Levi subgroup of $\mathbf{G}$. We now show how to use the CHEVIE package to check the following lemma:

Lemma 2. *Let $\mathbf{G}$ be an adjoint group of type E_6 and $\mathbf{M}$ a Levi subgroup of type $A_2 \times A_2$. If s is a semisimple element of $\mathbf{M}$ which is quasi-isolated in $\mathbf{M}$ and in $\mathbf{G}$, there exists $z \in \mathbf{Z}(\mathbf{M})$ of order 3 such that s and sz are not conjugate in $\mathbf{G}$.*

Proof. We first compute the list of elements of order 3 of $\mathbf{Z}(\mathbf{M})$. The first thing is to specify the Levi subgroup $\mathbf{M}$.

```
gap> G:=CoxeterGroup("E",6);;PrintDiagram(G);
E6      2
        |
1 - 3 - 4 - 5 - 6
gap>M:=ReflectionSubgroup(G,[1,3,5,6]);
ReflectionSubgroup(CoxeterGroup("E",6), [ 1, 3, 5, 6 ])
```

We now compute the torus $Z(\mathbf{M})^\circ = Z(\mathbf{M})$.

`AlgebraicCentre(G)`
This function returns a description of the centre Z of the algebraic group `G` (it may be a non-connected group, represented as an *extended Coxeter group*, see below) as a record with the following fields
`.Z0`: A basis of $Y(Z^0)$ (with respect to the canonical basis of $Y(\mathbf{T})$)

`.complement`: A basis of $Y(\mathbf{S})$, a complement lattice to `.Z0` in $Y(\mathbf{T})$ where $\mathbf{S}$ is a complement torus to Z^0 in $\mathbf{T}$.
`.AZ`: representatives of $A(Z) = Z/Z^0$ given as a subgroup of $\mathbf{S}$ (that is, elements of $\mathbb{Q}/\mathbb{Z} \otimes Y(\mathbf{S})$).

```
gap> ZM:=AlgebraicCentre(M).Z0;
[ [ 0, 1, 0, -1, 0, 0 ], [ 0, 0, 0, 1, 0, 0 ] ]
```

We now ask for the subgroup of elements of order 3 of $Z(\mathbf{M})^\circ$.

`SemisimpleSubgroup( G, V, n)`
Assuming that the characteristic of $\mathbb{F}$ does not divide n, this function returns the subgroup of elements of order dividing `n` in the subtorus $\mathbf{S}$ of the maximal torus $\mathbf{T}$ of the algebraic group $\mathbf{G}$, where $\mathbf{S}$ is represented by `V`, an integral basis of the sublattice $Y(\mathbf{S})$ of $Y(\mathbf{T})$.

```
gap> Z3:=SemisimpleSubgroup(G,ZM,3);
Group( <0,1/3,0,2/3,0,0>, <0,0,0,1/3,0,0> )
```

The above illustrates how semisimple elements are printed. The group `Z3` is represented as a subgroup of $\mathbf{T}$; elements of $\mathbf{T}$, which is of dimension 6, are represented as lists of 6 elements of $\mathbb{Q}/\mathbb{Z}$ in angle brackets; elements of $\mathbb{Q}/\mathbb{Z}$ are themselves represented as fractions r such that $0 \leq r < 1$. The subgroup of elements of order 3 of $Z(\mathbf{M})^\circ$ is generated by 2 elements which are given above. We may ask for the list of all elements of this group.

```
gap> Z3:=Elements(Z3);
[ <0,0,0,0,0,0>, <0,0,0,1/3,0,0>, <0,0,0,2/3,0,0>,
  <0,1/3,0,2/3,0,0>, <0,1/3,0,0,0,0>, <0,1/3,0,1/3,0,0>,
  <0,2/3,0,1/3,0,0>, <0,2/3,0,2/3,0,0>, <0,2/3,0,0,0,0> ]
```

We now compute the list of elements quasi-isolated in both $\mathbf{G}$ and $\mathbf{M}$.

```
gap> reps:=QuasiIsolatedRepresentatives(G);
[ <0,0,0,0,0,0>, <0,0,0,0,1/2,0>, <0,0,0,1/3,0,0>,
  <0,1/6,1/6,0,1/6,0>, <1/3,0,0,0,0,1/3> ]
```

The list `reps` now contains representatives of $\mathbf{G}$-orbits of quasi-isolated elements. The algorithm to get these was described in [Bonnafe2005]. To get all the quasi-isolated elements in $\mathbf{T}$, we need to take the orbits under the Weyl group:

```
gap> qi:=List(reps,s->Orbit(G,s));;
gap> List(qi,Length);
[ 1, 36, 80, 1080, 90 ]
```

We have not displayed the orbits since they are quite large: the first orbit is that of the identity element, which is trivial, but the fourth contains 1080 elements. We now filter each orbit by the condition to be quasi-isolated also in $\mathbf{M}$.

```
gap> qi:=List(qi,x->Filtered(x,y->IsQuasiIsolated(M,y)));;
gap> List(qi,Length);
[ 1, 3, 26, 36, 12 ]
gap> qi[2];
[ <0,0,0,1/2,0,0>, <0,1/2,0,1/2,0,0>, <0,1/2,0,0,0,0> ]
```

There is a way to do the same computation which does not need to compute the large intermediate orbits under the Weyl group of $\mathbf{G}$. The idea is to compute first the orbit of a semisimple quasi-isolated representative s under representatives of the double cosets $C_{\mathbf{G}}(s)\backslash\mathbf{G}/\mathbf{M}$, which are not too many, then test for being quasi-isolated in $\mathbf{M}$, and finally take the orbits under the Weyl group of $\mathbf{M}$.

We use the following CHEVIE function:

> `SemisimpleCentralizer( G, s)`
>
> This function returns the stabilizer in the Weyl group of the semisimple element `s` of the algebraic group `G`. The result describes also $C_{\mathbf{G}}(s)$ since it is returned as an extended reflection group, with the reflection group part equal to the Weyl group of $C^0_{\mathbf{G}}(s)$, and the diagram automorphism part being that induced by $C_{\mathbf{G}}(s)/C^0_{\mathbf{G}}(s)$ on $C^0_{\mathbf{G}}(s)$. The extended reflection groups represent non-connected reductive groups which are semi-direct product of their connected component by a group of diagram automorphisms.

So starting with `reps` as above, we first compute:

```
ce:=List(reps,s->SemisimpleCentralizer(G,s));;ce[5];
Extended(ReflectionSubgroup(CoxeterGroup("E",6),
[ 2, 3, 4, 5 ]),<(2,5,3)>)
gap> ce[5].group;
ReflectionSubgroup(CoxeterGroup("E",6), [ 2, 3, 4, 5 ])
gap> ce[5].permauts;
Group( ( 1,72, 6)( 2, 5, 3)( 7,71,11)( 8,10, 9)(12,70,16)
(13,14,15)(17,68,21)(18,69,20)(22,66,25)(23,67,65)(26,63,28)
(27,64,62)(29,59,31)(30,61,58)(32,57,53)(33,56,54)(34,52,48)
(35,47,43)(36,42,37)(38,41,39)(44,46,45)(49,50,51) )
```

We show for the 5th element of `reps` how an extended reflection group is represented: it contains a reflection subgroup of the Weyl group of $\mathbf{G}$, the Weyl group of $C^\circ_{\mathbf{G}}(s)$, obtained above as `ce[5].group`, extended by the group of diagram automorphisms induced on it by $C_{\mathbf{G}}(s)$, obtained above as `ce[5].permauts`; these automorphisms are denoted by the permutation of the simple roots of $C^\circ_{\mathbf{G}}(s)$ they induce.

To get the whole Weyl group of $C_{\mathbf{G}}(s)$ we need to combine these two pieces. For this we define a GAP function:

```
TotalGroup:=g->Subgroup(G,Concatenation(g.group.generators,
g.permauts.generators));
```

We then compute representatives of the double cosets $C_{\mathbf{G}}(s)\backslash\mathbf{G}/\mathbf{M}$, we apply them to `reps`, keep the ones still quasi-simple in $\mathbf{M}$:

```
dreps:=List(ce,g->List(DoubleCosets(G,TotalGroup(g),M),
Representative));;
qi:=List([1..Length(reps)],i->List(dreps[i],w->reps[i]^w));;
qi:=List(qi,x->Filtered(x,y->IsQuasiIsolated(M,y)));
[ [ <0,0,0,0,0,0> ], [ <0,1/2,0,0,0,0>, <0,0,0,1/2,0,0>,
  <0,1/2,0,1/2,0,0> ], [ <0,0,0,1/3,0,0>, <0,0,0,2/3,0,0>,
  <1/3,2/3,1/3,0,2/3,2/3>, <1/3,1/3,1/3,0,2/3,2/3> ],
  [ <1/3,1/2,1/3,1/2,2/3,2/3>, <1/3,1/2,1/3,0,2/3,2/3>,
  <2/3,0,2/3,5/6,2/3,2/3> ], [ <1/3,0,1/3,0,1/3,1/3> ] ]
```

We get a list such that the $\mathbf{M}$-orbits of the sublists give the same list as before. We will need this previous list of all $\mathbf{G}$-conjugates which are $\mathbf{M}$-quasi-isolated, so if we did not keep it we recompute this list containing the $\mathbf{M}$-orbits of the sublists by

```
qim:=List(qi,l->Union(List(l,s->Orbit(M,s))));;
```

We now ask, for each element s of each of our orbits, how many elements z of `Z3` are such that s and sz are not $\mathbf{G}$-conjugate. The test for being conjugate is that sz is in the same $\mathbf{G}$-orbit. We need to make the test only for our representatives of the $\mathbf{M}$-orbits, since if s is $\mathbf{G}$-conjugate to sz with $z \in \mathbf{Z}(\mathbf{M})$, then msm^{-1} is $\mathbf{G}$-conjugate to $msm^{-1}z = mszm^{-1}$.

```
gap> List([1..Length(qi)],i->List(qi[i],s->Number(Z3,
  z->PositionProperty(qim,o->s*z in o)<>i)));
[ [ 8 ], [ 8, 8, 8 ], [ 7, 7, 3, 3 ], [ 6, 6, 6 ], [ 6 ] ]
```

and we find indeed that there is always more than 0 elements z which work. Note that the function `PositionProperty` returns `false` when no element is found satisfying the given property, thus the number counted is the z such that s and sz are in a different orbit, as well as the cases when sz is not quasi-isolated in $\mathbf{G}$. □

5. Rational structures

We now assume that $\mathbb{F}$ is an algebraic closure of $\mathbb{F}_p$ and that F is the Frobenius on $\mathbf{G}$ corresponding to an $\mathbb{F}_q$-structure where q is a power of p. We assume that $\mathbf{G}$ is an adjoint group of type E_7 which in `CHEVIE` has the following labelling of the simple roots:

```
gap> G:=CoxeterGroup("E",7);;PrintDiagram(G);
E7       2
         |
1 - 3 - 4 - 5 - 6 - 7
```

We are going to show the `CHEVIE` code for the following lemma.

Lemma 3. *Assume* $\mathbf{M}$ *is an* F*-stable Levi of type* $A_1 \times A_1 \times A_1$ *corresponding to the roots 2, 3, 5 in the above diagram. If* $q \in \{3, 5\}$ *and if* $|\mathbf{Z}(\mathbf{M})^F| = \Phi_1(q)^a \Phi_2(q)^b$ *with* a, $b \geq 1$, *then* $\mathbf{Z}(\mathbf{M})^F$ *contains an element of order* 8.

In `CHEVIE`, to specify an $\mathbb{F}_q$-structure on a reductive group, we give an element $\phi \in \mathrm{GL}(Y(\mathbf{T}))$ such that $F = q\phi$. We may choose ϕ such that it stabilizes the set of simple roots. Such an element ϕ is determined by the coset $W\phi \subset \mathrm{GL}(Y(\mathbf{T}))$, so the structure which represents it in `CHEVIE` is a Coxeter coset.

Further, if $\mathbf{M}'$ is an F-stable $\mathbf{G}$-conjugate of the Levi subgroup $\mathbf{M}$, the pair $(\mathbf{M}', F)$ is isomorphic to $(\mathbf{M}, wF)$ for some $w \in W$ (determined by $\mathbf{M}'$ up to F-conjugacy). So, given a Coxeter coset $W\phi$, an F-stable conjugate of a Levi subgroup whose Weyl group is a standard parabolic subgroup W_I is represented by a subcoset of the form $W_I w\phi$, where $w\phi$ normalizes W_I.

To check the lemma, we first compute the list of elements of order 8 of $\mathbf{Z}(\mathbf{M})$, using the same commands as shown before.

```
gap> M:=ReflectionSubgroup(G,[2,5,7]);;
gap> ZM:=AlgebraicCentre(M);;
gap> Z8:=SemisimpleSubgroup(G,ZM.Z0,8);
Group( <1/8,0,0,0,0,0,0>, <0,0,1/8,7/8,0,1/8,0>,
<0,0,0,1/8,0,7/8,0>, <0,0,0,0,0,1/8,0> )
gap> Z8:=Elements(Z8);;Length(Z8);
4096
```

We now ask for representatives of the $\mathbf{G}^F$-classes of F-stable $\mathbf{G}$-conjugates of $\mathbf{M}$. The group $\mathbf{G}$ is split, so ϕ is trivial. Thus an F-stable-conjugate of $\mathbf{M}$ is represented by a coset of the form $W_I w$. We first ask for the list of all possible such twistings of $\mathbf{M}$:

```
gap> Mtwists:=Twistings(G,M);
[ A1<2>xA1<5>xA1<7>.(q-1)^4,
  (A1xA1xA1)<2,5,7>.(q-1)^2*(q^2+q+1),
  A1<2>xA1<5>xA1<7>.(q-1)^2*(q^2+q+1),
  (A1xA1xA1)<2,5,7>.(q^2+q+1)^2,
  (A1xA1xA1)<2,7,5>.(q-1)*(q+1)*(q^2+q+1),
  (A1xA1xA1)<2,7,5>.(q-1)*(q+1)*(q^2-q+1),
  ...
```

In the above list (of 24 entries of which the first 6 are listed), brackets around pairs or triples of A_1 denote an orbit of the Frobenius on the components. The element w is not displayed, but the order $|\mathbf{Z}(\mathbf{M})^{wF}|$ is displayed. We want to keep the sublist where that order is a product of $\Phi_1(q)$ and $\Phi_2(q)$. For this we use the function

> `PhiFactors(WF)`
>
> Let `WF` be a reflection coset of the form $W\phi$, and let V be the vector space on which W acts as a reflection group. Let $f_1, \ldots, f_n$ be the basic invariants of W on the symmetric algebra of V, chosen so that ϕ has the f_i as eigenvectors. The corresponding eigenvalues, listed in the same order as `ReflectionDegrees(W)` (the degrees of the f_i) are called the *factors* of ϕ acting on V.

```
gap> Mtwists:=Filtered(Mtwists,MF->Set(PhiFactors(MF))=[-1,1]);
[ A1<2>xA1<5>xA1<7>.(q+1)^4,
  A1<2>xA1<5>xA1<7>.(q-1)^2*(q+1)^2,
  (A1xA1)<2,7>xA1<5>.(q-1)^3*(q+1),
  A1<2>xA1<5>xA1<7>.(q-1)*(q+1)^3,
  A1<2>xA1<5>xA1<7>.(q-1)^3*(q+1),
  (A1xA1)<2,7>xA1<5>.(q-1)*(q+1)^3,
  (A1xA1)<2,7>xA1<5>.(q-1)^2*(q+1)^2 ]
```

Here `PhiFactors` gives the eigenvalues of w on the invariants of the Weyl group of $\mathbf{M}$ acting on the symmetric algebra of $X(\mathbf{T})\otimes\mathbb{C}$. The cases we want is when these eigenvalues are all equal to 1 or -1 (actually this gives us one extra case, where $|\mathbf{Z}(\mathbf{M})^{wF}| = (q+1)^4$ since the eigenvalues on the complement of $\mathbf{Z}(\mathbf{M})$ are always 1; we will just have to disregard the first entry of `Mtwists`).

Now for each of the remaining `Mtwists` we compute the fixed points of wF on `Z8`, and look at the maximal order of an element in there. We first illustrate the necessary commands one by one on an example before showing a line of code which combines them.

```
gap> Z8F:=Filtered(Z8,s->Frobenius(Mtwists[3])(s)^3=s);
[ <0,0,0,0,0,0,0>, <0,0,0,1/4,0,1/4,0>,
  <0,0,0,1/2,0,1/2,0>, <0,0,0,3/4,0,3/4,0>,
  ...
  <3/4,0,3/4,1/8,0,3/8,0>, <3/4,0,3/4,3/8,0,5/8,0>,
  <3/4,0,3/4,5/8,0,7/8,0>, <3/4,0,3/4,7/8,0,1/8,0> ]
```

The expression `Frobenius(Mtwists[3])` returns a function which applies to its argument the $w\phi$ associated to the third twisting `(A1xA1)<2,7>xA1<5>.(q-1)^3*(q+1)` of $\mathbf{M}$. To compute wF we still have to raise to the third power since $q=3$. We give above 8 of the 32 entries obtained; we can see from the denominators that some elements in the resulting list of wF-stable elements of `Z8` are of order 8. We can make this easier to see by writing a small function:

```
gap> OrderSemisimple:=s->Lcm(List(s.v,Denominator));
gap> List(Z8F,OrderSemisimple);
[ 1, 4, 2, 4, 2, 4, 2, 4, 8, 8, 8, 8, 8, 8, 8, 8,
  2, 4, 2, 4, 2, 4, 2, 4, 8, 8, 8, 8, 8, 8, 8, 8 ]
gap> Set(last);
[ 1, 2, 4, 8 ]
```

We now do the computation for all cosets in one command:

```
gap> List(Mtwists,MF->Set(List(Filtered(Z8,s->Frobenius(MF)(s)^3=s),
> OrderSemisimple)));
[ [ 1, 2, 4 ], [ 1, 2, 4, 8 ], [ 1, 2, 4, 8 ],
  [ 1, 2, 4, 8 ], [ 1, 2, 4, 8 ], [ 1, 2, 4, 8 ],
  [ 1, 2, 4, 8 ] ]
```

and we see that indeed, apart from the first twist which should be disregarded, for all twists the fixed points of `Z8` still contain elements of order 8.

6. Lusztig's map from conjugacy classes in the Weyl group to conjugacy classes in the reductive group

Let $\mathbf{G}$ be a connected reductive group over an algebraically closed field $\mathbb{F}$ of characteristic $p\geq 0$, and let $\mathcal{B}$ be the variety of its Borel subgroups. The $\mathbf{G}$-orbits (for the diagonal action) on $\mathcal{B}\times\mathcal{B}$ are naturally indexed by the Weyl group W of $\mathbf{G}$; we denote $\mathcal{O}_w$ the orbit indexed by $w\in W$.

For γ a conjugacy class of $\mathbf{G}$ and $w\in W$, we set

$$\mathcal{B}_w^\gamma := \{(g,\mathbf{B})\in\gamma\times\mathcal{B} \mid (\mathbf{B},{}^g\mathbf{B})\in\mathcal{O}_w\}.$$

For C a conjugacy class of W and γ a unipotent class of $\mathbf{G}$, Lusztig denotes $\gamma \vdash C$ if $\mathcal{B}_w^\gamma \neq \emptyset$ for any $w \in C_{\min}$ where $C_{\min}$ is the set of elements of C of minimal length.

In [Lusztig2011b] Lusztig shows:

Theorem 4. *If p is good for $\mathbf{G}$ then*

- *For C a class of W, among the classes such that $\gamma \vdash C$, there exists a unique class γ_C minimal for the partial order defined by:*

  $$\gamma \leq \gamma' \text{ if and only if } \gamma \text{ lies in the Zariski closure of } \gamma'.$$

- *The map $C \mapsto \gamma_C$ is surjective.*

Lusztig uses `CHEVIE` to show Theorem 4 for exceptional groups. He was not using the development version; see also [Geck2011] describing the same computation. The problem is very easy to solve using the development version of `CHEVIE`, as we will show by giving the complete code to solve it; we will see along the way:

Theorem 5. *Theorem 4 still holds in bad characteristic for exceptional groups.*

The idea of the computation is as follows. Assume that $\mathbb{F}$ is an algebraic closure of a finite prime field $\mathbb{F}_p$. Let F be the Frobenius corresponding to a split rational structure of $\mathbf{G}$ over $\mathbb{F}_q$ where q is a power of p. Lusztig gives a formula for $|(\mathcal{B}_w^\gamma)^F|$ when γ is unipotent, which shows that $|(\mathcal{B}_w^\gamma)^F|$ is a polynomial in q. It is thus equivalent that $\mathcal{B}_w^\gamma \neq \emptyset$ or $|(\mathcal{B}_w^\gamma)^F| \neq 0$ as a polynomial, which enables us to do the computation since this polynomial can be readily computed in `CHEVIE`.

To give the formula, we need some more notation related to the permutation module $\overline{\mathbb{Q}}_\ell \mathcal{B}^F$ for $\mathbf{G}^F$ (where ℓ is a prime different from p; we use $\overline{\mathbb{Q}}_\ell$ instead of $\mathbb{C}$ as coefficients to make a connection with ℓ-adic cohomology, see below). The Hecke algebra, defined as $\mathcal{H} := \mathrm{End}_{\mathbf{G}^F} \overline{\mathbb{Q}}_\ell \mathcal{B}^F$ has basis $\{T_w\}_{w \in W}$ defined by $T_w(\mathbf{B}) = \sum_{\{\mathbf{B}' | (\mathbf{B}, \mathbf{B}') \in \mathcal{O}_w^F\}} \mathbf{B}'$. The algebra $\mathcal{H}$ specializes for $q \mapsto 1$ to $\overline{\mathbb{Q}}_\ell W$, inducing a bijection $E_q \mapsto E : \mathrm{Irr}(\mathcal{H}) \to \mathrm{Irr}(W)$ and as $\mathcal{H} \times \mathbf{G}^F$-module we have the decomposition $\overline{\mathbb{Q}}_\ell \mathcal{B}^F = \sum_{E \in \mathrm{Irr}\, W} E_q \otimes \rho_E$ where ρ_E is an irreducible *unipotent representation of the principal series* of $\mathbf{G}^F$.

Finally, for E a representation of W, let us define the *almost character* $R_E^{\mathbf{G}} := |W|^{-1} \sum_{w \in W} \mathrm{Trace}(w \mid E) R_{\mathbf{T}_w}^{\mathbf{G}}(\mathrm{Id})$ where $\mathbf{T}_w$ is an F-stable maximal torus of $\mathbf{G}$ of type w and $R_{\mathbf{T}_w}^{\mathbf{G}}$ is Deligne-Lusztig induction; here $R_{\mathbf{T}_w}^{\mathbf{G}}(\mathrm{Id})$ is a virtual $\overline{\mathbb{Q}}_\ell \mathbf{G}^F$-module. Lusztig's formula is

$$|(\mathcal{B}_w^\gamma)^F| = \sum_{E' \in \mathrm{Irr}(W)} \mathrm{Trace}(T_w \mid \sum_{E \in \mathrm{Irr}\, W} \langle \rho_E, R_{E'}^{\mathbf{G}} \rangle_{\mathbf{G}^F} E_q) \sum_{u \in \gamma^F} \mathrm{Trace}(u \mid R_{E'}^{\mathbf{G}}). \quad (a)$$

By [Geck-Pfeiffer1993, Theorem 1.1 (b)] the conjugacy class of T_w in $\mathcal{H}$ is the same for all $w \in C_{\min}$ thus by (a) the condition $\mathcal{B}_w^\gamma \neq \emptyset$ does not depend on the choice of $w \in C_{\min}$; this makes the computation doable in a group like $W(E_8)$ which has 696729600 elements but only 112 conjugacy classes.

Now, we introduce some `CHEVIE` functions which can be used to compute the right-hand side of (a). We decompose the computation in two steps, computing the functions

$$f(C, E') := \mathrm{Trace}(T_w \mid \sum_{E \in \mathrm{Irr}\, W} \langle \rho_E, R_{E'}^{\mathbf{G}} \rangle_{\mathbf{G}^F} E_q) \text{ for } w \in C_{\min},$$

and $g(E', \gamma) := \sum_{u \in \gamma^F} \mathrm{Trace}(u \mid R_{E'}^{\mathbf{G}})$.

To compute f we need:

`UnipotentCharacters(G)`

This function returns a record containing information about the unipotent characters of the reductive group `G`. Some important fields are:

`charNames`: the list of names of the unipotent characters.

`harishChandra`: information about Harish-Chandra series of unipotent characters. This is itself a list of records, one for each pair $(\mathbf{L}, \lambda)$ of a Levi of an F-stable parabolic subgroup and a cuspidal unipotent character of $\mathbf{L}^F$.

`families`: information about Lusztig families of unipotent characters. These families correspond to blocks of the matrix of scalar products $\langle \rho, R_E^{\mathbf{G}} \rangle_{\mathbf{G}^F}$ where ρ runs over the unipotent characters. This matrix can also be obtained from the `UnipotentCharacters` record.

As an example we now show the result of TeXing the output of
`FormatTeX(UnipotentCharacters(CoxeterGroup("G",2)));`

Unipotent characters for G_2

γ	Deg(γ)	FakeDegree	Fr(γ)	Label
$\phi_{1,0}$	1	1	1	
$\phi_{1,6}$	q^6	q^6	1	
$\phi'_{1,3}$	$\frac{1}{3}q\Phi_3\Phi_6$	q^3	1	$(1,\rho)$
$\phi''_{1,3}$	$\frac{1}{3}q\Phi_3\Phi_6$	q^3	1	$(g_3,1)$
$\phi_{2,1}$	$\frac{1}{6}q\Phi_2^2\Phi_3$	$q\Phi_8$	1	$(1,1)$
$\phi_{2,2}$	$\frac{1}{2}q\Phi_2^2\Phi_6$	$q^2\Phi_4$	1	$(g_2,1)$
$G_2[-1]$	$\frac{1}{2}q\Phi_1^2\Phi_3$	0	-1	(g_2,ε)
$G_2[1]$	$\frac{1}{6}q\Phi_1^2\Phi_6$	0	1	$(1,\varepsilon)$
$G_2[\zeta_3]$	$\frac{1}{3}q\Phi_1^2\Phi_2^2$	0	ζ_3	(g_3,ζ_3)
$G_2[\zeta_3^2]$	$\frac{1}{3}q\Phi_1^2\Phi_2^2$	0	ζ_3^2	(g_3,ζ_3^2)

The `FakeDegree` is defined for a unipotent character ρ_E as $R_E^{\mathbf{G}}(1)$ and is 0 outside of the principal series. The column Fr(γ) contains the root of unity part of the Frobenius eigenvalue attached to the character γ when it appears in the cohomology of a Deligne-Lusztig variety. Finally the "Label" refers to the labelling by Lusztig families.

To compute f we use also:

`DeligneLusztigLefschetz(h)`

By [Digne-Michel1985, III, 1.3 and 2.3], the class function on $\mathbf{G}^F$ which associates to $g \in \mathbf{G}^F$ the number of fixed points of gF^m on the Deligne-Lusztig variety associated to the element $w \in W$ has, for m sufficiently divisible, the form $g \mapsto \sum_{E\in\mathrm{Irr}(W)} \mathrm{Trace}(T_w \mid E_{q^m}) R_E^{\mathbf{G}}(g)$. This expression is called the *Lefschetz character* associated to the Deligne-Lusztig variety; since $\mathcal{H}$ splits over $\mathbb{Q}[\sqrt{q}]$, it is a sum of unipotent characters with coefficients in $\mathbb{Q}[\sqrt{q}]$.

The function `DeligneLusztigLefschetz` takes as argument a Hecke element `h` and returns the corresponding Lefschetz character (the definition is extended from T_w to any `h`$\in \mathcal{H}$ by linearity).

Note that, since $\langle \rho_E, R_{E'}^{\mathbf{G}} \rangle_{\mathbf{G}^F} = \langle \rho_{E'}, R_E^{\mathbf{G}} \rangle_{\mathbf{G}^F}$ (see [Digne-Michel1985, III, 3.5(iii)]), the function $f(C, E')$ is the coefficient of `DeligneLusztigLefschetz`(T_w) on $\rho_{E'}$.

Here is a function which takes as arguments W and q, and returns the matrix of $f(C, E')$, with rows indexed by C and columns indexed by E'.

```
f:=function(W,q)
  return List(ChevieClassInfo(W).classtext,
    function(w)local Tw,psuc;
      Tw:=Basis(Hecke(W,q),"T")(w);
      psuc:=UnipotentCharacters(W).harishChandra[1].charNumbers;
      return DeligneLusztigLefschetz(Tw).v{psuc};
    end);
end;
```

We use here that `ChevieClassInfo(W).classtext` contains a representative of C in $C_{\min}$. The variable `psuc` which stands for "principal series unipotent characters" holds the indices of the unipotent characters of the principal series ρ_E amongst all the unipotent characters. The principal series is the first of the Harish-Chandra series described by the list `UnipotentCharacters(W).harishChandra`. The field `v` of the `DeligneLusztigLefschetz(Tw)` record is a list of coefficients on each unipotent character.

The function `f` takes on a 2Ghz computer 3 seconds for E_7, and 23 seconds for E_8.

To write the code for the function g, we will need to describe the `CHEVIE` functions dealing with unipotent elements, the Springer correspondence, and the Green functions.

Let (γ, φ) run over the pairs where γ is a unipotent class of $\mathbf{G}$, and φ is a character of the group of components $A(u) := C_{\mathbf{G}}(u)/C^0_{\mathbf{G}}(u)$ of $C_{\mathbf{G}}(u)$ for $u \in \gamma$. Such pairs describe $\mathbf{G}$-equivariant local systems on γ. The *Springer correspondence* is an injective map from $\mathrm{Irr}(W)$ to the set of such pairs. It is not surjective in general, but all the pairs (γ, Id) are in the image. The *generalized Springer Correspondence* extends this to a bijection. This time the source is the union of $\mathrm{Irr}(W_{\mathbf{G}}(\mathbf{L}))$, where $\mathbf{L}$ runs over Levi subgroups (taken up to conjugacy) which admit a *cuspidal* local system (λ, ϕ); the group $W_{\mathbf{G}}(\mathbf{L}) := N_{\mathbf{G}}(\mathbf{L})/\mathbf{L} = N_{\mathbf{G}}(\mathbf{L}, \phi)/\mathbf{L}$ is a Coxeter group for such Levis. The image of each of these sets is called a *Springer series*. We will just need the ordinary Springer correspondence in our computation (whose image is called the *principal Springer series*), but the `CHEVIE` functions describe the generalized correspondence.

`UnipotentClasses(G[,p])`

This function returns a record containing information about the unipotent classes of the algebraic group `G` in characteristic `p` (if omitted, `p` is assumed to be any good characteristic for $\mathbf{G}$). Some important fields of the record are:

`.orderClasses`: a list describing the partial order on unipotent classes. The poset is described by its Hasse diagram, that is the `i`-th element of the list is the list of the indices `j` of the classes immediately above the `i`-th class. That is `.orderclasses[i]` contains `j` if $\overline{\gamma}_j \supsetneq \gamma_i$ and there is no class γ_k such that $\overline{\gamma}_j \supsetneq \overline{\gamma}_k \supsetneq \gamma_i$.

`.classes`: a list of records holding information for each unipotent class. In particular the field `.Au` holds the group $A(u)$.

`.springerSeries`: a list of records, each of which describes a Springer series of $\mathbf{G}$. The main field in such a record is `.locsys`, a list of length `NrConjugacyClasses(WGL)`, where `WGL` is the group $W_{\mathbf{G}}(\mathbf{L})$ associated to the series, holding in `i`-th position a pair describing which local system corresponds to the `i`-th character of `WGL`. The first element of the pair is the index of the concerned unipotent class u, and the second is the index of the corresponding character of $A(u)$.

The record returned by `UnipotentClasses` depends on the isogeny type of $\mathbf{G}$ and on the characteristic. We give some examples.

First, here is what gives TeXing the output of
`FormatTeX(UnipotentClasses(CoxeterGroup("G",2)))`:
$1<A_1<\tilde{A}_1<G_2(a_1)<G_2$

u	diagram	$\dim \mathcal{B}_u$	$A(u)$		$G_2()$	$.(G_2)$
G_2	22	0	.		$\phi_{1,0}$	
$G_2(a_1)$	20	1	A_2	$21:\phi'_{1,3}$	$3:\phi_{2,1}$	$111:$
$\tilde{A}_1$	01	2	.		$\phi_{2,2}$	
A_1	10	3	.		$\phi''_{1,3}$	
1	00	6	.		$\phi_{1,6}$	

The first line describes the partial order of the unipotent classes. In that line and the first column the traditional name of the unipotent classes are used. The columns "diagram" shows the Dynkin-Richardson diagram which we do not explain here. The next column shows the dimension of the variety $\mathcal{B}_u := \{\mathbf{B} \in \mathcal{B} \mid \mathbf{B} \ni u\}$. The next columns describe the Springer correspondence. Only the second class has a non-trivial $A(u)$, equal to the Coxeter group of type A_2 (the symmetric group on 3 elements); this group has three irreducible characters, indexed by the partitions 3, 21 and 111. The first two characters are in the ordinary Springer correspondence, corresponding to the irreducible characters $\phi'_{1,3}$ and $\phi_{2,1}$ of G_2. The third corresponds to a cuspidal local system, so is a Springer series by itself. The head of a column describing a Springer series attached to the cuspidal pair $(\mathbf{L},(\lambda,\phi))$ is of the form `A(B)` where `A` describes $W_{\mathbf{G}}(\mathbf{L})$ and `B` describes $\mathbf{L}$.

The result is different in characteristic three
`FormatTeX(UnipotentClasses(CoxeterGroup("G",2),3))`:
$1<A_1,(\tilde{A}_1)_3<\tilde{A}_1<G_2(a_1)<G_2$

u	$\dim \mathcal{B}_u$	$A(u)$	$G_2()$	$.(G_2)$	$.(G_2)$	$.(G_2)$
G_2	0	Z_3	$1:\phi_{1,0}$		$\zeta_3:$	$\zeta_3^2:$
$G_2(a_1)$	1	A_1	$2:\phi_{2,1}$	$11:$		
$\tilde{A}_1$	2	.	$\phi_{2,2}$			
A_1	3	.	$\phi''_{1,3}$			
$(\tilde{A}_1)_3$	3	.	$\phi'_{1,3}$			
1	6	.	$\phi_{1,6}$			

As another example we show the difference between PGL_n and SL_n:
`FormatTeX(UnipotentClasses(RootDatum("pgl",4))):`
1111<211<22<31<4

u	diagram	$\dim \mathcal{B}_u$	$A(u)$	$A_3()$
4	222	0	.	4
31	202	1	.	31
22	020	2	.	22
211	101	3	.	211
1111	000	6	.	1111

`FormatTeX(UnipotentClasses(RootDatum("sl",4))):`
1111<211<22<31<4

u	diagram	$\dim \mathcal{B}_u$	$A(u)$	$A_3()$	$A_1(A_1^2)/-1$	$.(A_3)/i$	$.(A_3)/-i$
4	222	0	Z_4	1 : 4	$-1:2$	$i:$	$-i:$
31	202	1	.	31			
22	020	2	A_1	2 : 22	11 : 11		
211	101	3	.	211			
1111	000	6	.	1111			

In SL_4 there are more local systems on the same classes. Some are cuspidal, and some are in a Springer series corresponding to a Levi subgroup of type $A_1 \times A_1$. In the column heads for Springer series, we see a third parameter after a `/` describing a character of $Z\mathbf{G}$ attached to the series.

We need one more function to write the code for g:

`ICCTable(uc[,seriesNo[,q]])`

The first argument is a record describing the unipotent classes of a reductive group in some characteristic. `ICCTable` gives the table of decompositions of the functions $X_{\gamma,\varphi}$ in terms of the functions $Y_{\gamma,\varphi}$, where $Y_{\gamma,\varphi}$ is the characteristic function of the local system (γ,φ) and $X_{\gamma,\varphi}$ is the characteristic function of the corresponding intersection cohomology complex.

Since the coefficient of $X_{\gamma,\varphi}$ on $Y_{\gamma',\varphi'}$ is 0 if (γ,φ) and (γ',φ') are not in the same Springer series, the table given is for a single Springer series, the one whose number is given by the argument `seriesNo` (if omitted this defaults to `seriesNo=1` which is the principal series). The decomposition multiplicities are graded, and are given as polynomials in one variable (specified by the argument `q`; if not given `Indeterminate(Rationals)` is assumed).

The function `ICCTable` returns a record with various pieces of information which can help further computations. Some important fields are:

`.scalar`: the main result, the table of multiplicities of the X_ψ on the Y_χ, where both ψ and χ run over $\mathrm{Irr}(W_{\mathbf{G}}(\mathbf{L}))$.

`.dimBu`: The list of $\dim \mathcal{B}_u$ for each local system in the chosen Springer series.

`.L`: the matrix of unnormalized scalar products of the functions Y_ψ with themselves, that is, if F is a Frobenius corresponding to a split $\mathbb{F}_q$-structure

on $\mathbf{G}$, the (ϕ,ψ) entry is equal to $\sum_{g\in\mathbf{G}^F} Y_\phi(g)\overline{Y_\psi(g)}$. This is thus a symmetric, block-diagonal matrix where the diagonal blocks correspond to geometric unipotent conjugacy classes.

The relationship of `ICCTable(uc)` with characters of $\mathbf{G}^F$ is as follows: let $X_{(\gamma,\varphi)}$ be a characteristic function as above where (γ,φ) is in the ordinary Springer correspondence, image of $E'\in\mathrm{Irr}(W)$; we also write $X_{E'}$ for $X_{(\gamma,\varphi)}$ and write $b_{E'}$ for $\dim\mathcal{B}_u$. Then the restriction of the almost character $R^{\mathbf{G}}_{E'}$ to the unipotent elements is equal to $q^{b_{E'}}X_{E'}$. Thus $g(E',\gamma)=q^{b_{E'}}\sum_{g\in\mathbf{G}^F}X_{E'}(g)Y_{(\gamma,\mathrm{Id})}(g)$, thus if (γ,Id) is parameterized by $\psi\in\mathrm{Irr}(W)$, we have $g(E',\gamma)=q^{d_{E'}}P_{E',\psi}L_\psi$ where $P_{E',\psi}$ is the coefficient of $X_{E'}$ on Y_ψ and $L_\psi=\sum_{g\in\mathbf{G}^F}Y_\psi(g)\overline{Y_\psi(g)}$.

Here is a `CHEVIE` function which takes as arguments W, the indeterminate q and the characteristic p and computes the matrix of values $g(E',\gamma)$, with rows indexed by E' and columns indexed by unipotent classes γ.

```
g:=function(W,q,p)local uc,t,triv,d;
  uc:=UnipotentClasses(W,p);
  t:=ICCTable(uc,1,q);
  triv:=List([1..Size(uc)],i->
    Position(uc.springerSeries[1].locsys,
      [i,PositionId(uc.classes[i].Au)]));
  d:=List(t.dimBu,i->q^i);
  t:=Zip(t.scalar{triv},DiagonalOfMat(t.L{triv}{triv}),
    function(a,b)return a*b;end);
  return Zip(TransposedMat(t),d,function(a,b)return a*b;end);
end;
```

The list `triv` records, for each unipotent conjugacy class, the position in the ordinary Springer correspondence (that is, in the list of irreducible characters of the Weyl group) of the trivial local system in that class. As is traditional in functional languages, `Zip(l,v,f)` takes as arguments two lists `l` and `v` and a function `f` and returns a list `z` such that `z[i]=f(l[i],v[i])`.

We put what we have done so far together and we get:

```
vdash:=function(W,p)local q;q:=X(Rationals)^2;
  return List(f(W,q)*g(W,q,p),x->Filtered([1..Length(x)],
    i->x[i]<>0*X(Rationals)));
end;
```

The function `vdash` returns a list indexed by the classes of `W`, which contains for the `i`-th class C_i the list of indices j of unipotent classes γ_j such that $\gamma_j\vdash C_i$. Note that we use `X(Rationals)^2` as a variable since to compute the character table of the Hecke algebras of types E_7 and E_8 (which is used by `DeligneLusztigLefschetz`) `CHEVIE` must be able to extract square roots of the indeterminate.

It remains to check Lusztig's theorem 4, and to compute the map $C\mapsto\gamma_C$. The following function returns a list indexed by the conjugacy classes of `W` whose element indexed by C is the index of γ_C.

```
gamma:=function(W,p)local lt;
  lt:=Incidence(Poset(UnipotentClasses(W,p)));
  return List(vdash(W,p), function(l)local classes;
```

```
    classes:=Filtered(l,x->ForAll(l,y->lt[x][y]));
    if Length(classes)<>1 then Error("no minimal class \gamma_C");fi;
    return classes[1];
  end);
end;
```

We have used `CHEVIE` functions for posets: `Poset(UnipotentClasses(W,p))` returns the poset defined by Zariski closure of unipotent classes as a `CHEVIE` object, and `Incidence` returns the incidence matrix of this poset, that is, `lt[i][j]` is `true` if and only if $\gamma_i \subset \overline{\gamma}_j$.

The following function finally checks that $C \mapsto \gamma_C$ is surjective and displays this map.

```
LusztigMap:=function(W,p)local g,uc,i;
  g:=gamma(W,p);
  uc:=UnipotentClasses(W,p);
  if Set(g)<>[1..Size(uc)] then Error("not surjective");fi;
  for i in [1..Size(uc)] do
    Print(Join(List(Positions(g,i),j->ClassName(W,j)),", "),
          " -> ",ClassName(uc,i),"\n");
  od;
end;
```

Here is the result for G_2 in characteristics 0 and 3:

```
gap> LusztigMap(CoxeterGroup("G",2),0);
A0 -> 1
A1 -> A1
~A1, A1+~A1 -> ~A1
A2 -> G2(a1)
G2 -> G2
gap> LusztigMap(CoxeterGroup("G",2),3);
A0 -> 1
A1 -> A1
A1+~A1 -> ~A1
A2 -> G2(a1)
G2 -> G2
~A1 -> (~A1)3
```

The analogous computation takes 2 minutes on a 2 Ghz computer for a given characteristic in type E_8.

7. A generalization

In [Lusztig2013], Lusztig generalizes the above results to the case where γ is not necessarily unipotent. We now assume $\mathbf{G}$ simply connected to ensure that the centralizers of semisimple elements are connected.

We will see in formula (b) below that the condition $\mathcal{B}_w^\gamma \neq \emptyset$ is still independent of the choice of $w \in C_{\min}$. For a given class $C \subset W$, Lusztig denotes δ_C the minimum dimension of a class γ such that $\mathcal{B}_w^\gamma \neq \emptyset$ for $w \in C_{\min}$. It is clear from theorem 4 that $\delta_C \leq \dim \gamma_C$. Finally Lusztig denotes

$$\boxed{\mathbf{G}_C} = \bigcup_{\{\gamma \mid \mathcal{B}_w^\gamma \neq \emptyset \text{ and } \dim \gamma = \delta_C\}} \gamma$$

The main result of [Lusztig2013] is

Theorem 6.
- *Given two classes C and C' of W, the sets $\boxed{\mathbf{G}_C}$ and $\boxed{\mathbf{G}_{C'}}$ are equal or disjoint.*
- $\boxed{\mathbf{G}_C}$ *are the pieces of a stratification of* $\mathbf{G}$.

Assume again that $\mathbb{F}$ is an algebraic closure of $\mathbb{F}_p$, and that F is the Frobenius associated to an $\mathbb{F}_q$-structure for q a power of p. Up to replacing F by some power we may assume that γ is F-stable and that there exists s, the semisimple part of an F-stable element of γ, such that the restriction of F to $\mathbf{H} := C_{\mathbf{G}}(s)$ is split.

Then, with the notation $f(C, E')$ introduced after formula (a) we have for $w \in C_{\min}$

$$|(\mathcal{B}_w^\gamma)^F| = \frac{|\mathbf{G}^F|}{|\mathbf{H}^F|} \sum_{E' \in \mathrm{Irr}(W)} f(C, E') \sum_{u \in \mathbf{H}^F_{\mathrm{uni}} | su \in \gamma} \mathrm{Trace}(u \mid R^{\mathbf{H}}_{\mathrm{Res}^W_{W_{\mathbf{H}}} E'}). \qquad (b)$$

It is clear from formula (b) that $|(\mathcal{B}_w^\gamma)^F|$ depends on γ only through $\mathbf{H}$ and the unipotent class $\gamma_u \subset \mathbf{H}$ defined as the unipotent part of γ. Also, using the same argument as for (a), the right-hand side of (b) does not depend on the choice of $w \in C_{\min}$.

To compute (b), we proceed as for (a). The main difference is that we compute the function g in the group $\mathbf{H}$, and multiply on the left by the matrix describing the restriction of characters from W to $W_{\mathbf{H}}$. Here is the `CHEVIE` code replacing the `vdash` function:

```
List(f(W,q)*InductionTable(H,W).scalar*g(H,q,p),
  x->Filtered([1..Length(x)],i->x[i]<>0*X(Rationals)));
```

To check theorem 6, we need a list of possible groups $\mathbf{H}$. According to results of [Carter1978] and [Deriziotis1984], to describe their Weyl groups we must consider up to W-conjugacy the reflection subgroups of W generated by subsets of the *extended Dynkin diagram*, formed, for each irreducible component of W, by the simple roots and the negative of the highest root; for a given characteristic p we must exclude such subgroups which have a coefficient divisible by p when expressing their simple roots in an adapted basis of the initial root system.

The following function does the job for an irreducible W:

```
SemisimpleCentralizerRepresentatives:=function(W,p)
  local cent,E,J,R,indices;
  if Length(ReflectionType(W))<>1 then
    Error("only implemented for irreducible groups");
  fi;
  indices:=W->W.rootInclusion{W.generatingReflections};
  cent:=[];
  E:=Concatenation(indices(W),[W.rootInclusion[2*W.N]]);
  for J in Combinations(E) do
    R:=ReflectionSubgroup(W,J);
    if ForAll(cent,G->IsomorphismType(R)<>IsomorphismType(G) or
      RepresentativeOperation(W,indices(R),indices(G),OnSets)=false)
    then Add(cent,R);
    fi;
  od;
```

```
  if p=0 then return cent;fi;
  return Filtered(cent,
    G->ForAll(Concatenation(SmithNormalFormMat(W.roots{indices(G)})),
      x->x=0 or x mod p<>0));
end;
```

The function `IsomorphismType` describes the isomorphism type of a reflection group via, in the Weyl group case, a character string describing the isomorphism type of the root system; for instance `"A2+~A2"` describes the union of two systems of type A_2 where the second one consists of short roots.

Such a simple program is enough to detect some errors in the tables of [Deriziotis1984]: for $\mathbf{G}$ of type F_4 the type $A_2 + \tilde{A}_2$ for $\mathbf{H}$ is excluded in characteristic 3, not 2; the same is true for type $\mathbf{H} = 3A_2$ in $\mathbf{G} = E_7$; finally the type $\mathbf{H} = A_3 + 3A_1$ (which is excluded in characteristic 2) is altogether forgotten in $\mathbf{G} = E_8$.

We now have all that we need to compute and display the strata.

```
LusztigMapb:=function(W,p)local l,i,s,cent,m,q,fmat,strata,map,cover;
  q:=X(Rationals)^2; fmat:=f(W,q);
  cent:=SemisimpleCentralizerRepresentatives(W,p);
  l:=List([1..Length(cent)],function(i)local H,t,uc;
    H:=cent[i];uc:=UnipotentClasses(H,p);
    t:=List(fmat*InductionTable(H,W).scalar*g(H,q,p),
       x->Filtered([1..Length(x)],i->x[i]<>0*x[i]));
    return List(t, x->List(x,cl->rec(H:=i,
                        class:=ClassName(uc,cl),
       dim:=2*(W.N-uc.classes[cl].dimBu))));
    end);
  strata:=List(TransposedMat(l),function(s)local mindim;
    s:=Concatenation(s); mindim:=Minimum(List(s,x->x.dim));
    return Filtered(s,x->x.dim=mindim);end);
  map:=CollectBy([1..NrConjugacyClasses(W)],strata);
  strata:=Set(strata);
  if ForAny([1..Length(strata)],i->ForAny([i+1..Length(strata)],j->
    Length(Intersection(strata[i],strata[j]))<>0)) then
    Error("strata are not disjoint\n");
  fi;
  cover:=CollectBy(Concatenation(strata),i->i.H);
  if ForAny([1..Length(cover)],
    i->Length(cover[i])<>Size(UnipotentClasses(cent[i],p))) then
    Error("strata do not cover");
  fi;
  for i in [1..Length(map)] do
    Cut(SPrint("class(es) ",Join(List(map[i],j->ClassName(W,j)),", "),
       " => stratum(",strata[i][1].dim,") ",
       Join(List(strata[i],
         x->SPrint(IsomorphismType(cent[x.H]),":",x.class))," ")),
      rec(places:=" "));
  od;
end;
```

The function `CollectBy(l,f)` takes as argument a list `l` and a function `f` and returns a list of lists, each of them collecting all elements `l[i]` such that `f(l[i])` takes a given value. The second argument `f` may also be a list of same length as `l` and this time are collected the `l[i]` such that `f[l[i]]` takes a given value. The result is sorted by the value taken of `f`.

Here are the results of `LusztigMapb` for G_2 in characteristics 0 and 3:

```
gap> LusztigMapb(CoxeterGroup("G",2),0);
class(es) G2 => stratum(12) : A1:2 G2:G2 A2:3 ~A1:2 ~A1+A1:2,2
class(es) A2 => stratum(10) A1:11 G2:G2(a1) A2:21 ~A1:11
                            ~A1+A1:11,2 ~A1+A1:2,11
class(es) A0 => stratum(0) G2:1
class(es) A1 => stratum(6) G2:A1
class(es) A1+~A1 => stratum(8) G2:~A1 ~A1+A1:11,11
class(es) ~A1 => stratum(6) A2:111

gap> LusztigMapb(CoxeterGroup("G",2),3);
class(es) G2 => stratum(12) : A1:2 G2:G2 ~A1:2 ~A1+A1:2,2
class(es) A2 => stratum(10) A1:11 G2:G2(a1) ~A1:11 ~A1+A1:11,2
                            ~A1+A1:2,11
class(es) ~A1 => stratum(6) G2:(~A1)3
class(es) A0 => stratum(0) G2:1
class(es) A1 => stratum(6) G2:A1
class(es) A1+~A1 => stratum(8) G2:~A1 ~A1+A1:11,11
```

The classes γ composing a stratum are written in the form `A:B` where `A` describes the isomorphism type of the Weyl group of $\mathbf{H}$ and `b` describes γ_u.

We notice that no group $\mathbf{H}$ is of type A_2 in characteristic 3; in both cases the class `~A1` of W corresponds to a stratum composed of conjugacy classes of dimension 6, but the classes involved are quite different.

In the listing obtained for E_8 in characteristic 0, one can notice the same strata as pointed out by Lusztig: in all characteristics there is a unipotent stratum

```
class(es) A1 => stratum(58) E8:A1
```

And there is a stratum

```
class(es) 8A1, 6A1, 4A1'', 7A1, 5A1 => stratum(128) E8:4A1
                                          D8:1111111111111111
```

which loses the semisimple class `D8:1111111111111111` in characteristic 2.

8. Spetses

In [Malle1995] and [Broue-Malle-Michel2013], quite a few features of algebraic groups are generalized to a large class of finite complex reflection groups called "Spetsial". In particular, there is a definition of a set of "unipotent characters" of the "associated reductive group" (which does not exist but is a set of combinatorial data called a "Spets"). Here is an example of our computations, to be compared with the table given above for the unipotent characters in a reductive group of type G_2. We consider the group G_4 in the Shephard-Todd classification, which is a reflection subgroup of $\mathrm{GL}(\mathbb{C}^2)$ of size 24, isomorphic to $\mathrm{SL}_2(\mathbb{F}_3)$.

Here is the result of TeXing the output of

```
FormatTeX(UnipotentCharacters(ComplexReflectionGroup(4)));
```

Unipotent characters for G_4

γ	$\mathrm{Deg}(\gamma)$	FakeDegree	$\mathrm{Fr}(\gamma)$	Label
$\phi_{1,0}$	1	1	1	
$\phi_{1,4}$	$\frac{-\sqrt{-3}}{6}q^4\Phi_3''\Phi_4\Phi_6''$	q^4	1	$1\wedge-\zeta_3^2$
$\phi_{1,8}$	$\frac{\sqrt{-3}}{6}q^4\Phi_3'\Phi_4\Phi_6'$	q^8	1	$-1\wedge\zeta_3^2$
$\phi_{2,5}$	$\frac{1}{2}q^4\Phi_2^2\Phi_6$	$q^5\Phi_4$	1	$1\wedge\zeta_3^2$
$\phi_{2,3}$	$\frac{3+\sqrt{-3}}{6}q\Phi_3''\Phi_4\Phi_6'$	$q^3\Phi_4$	1	$1\wedge\zeta_3^2$
$\phi_{2,1}$	$\frac{3-\sqrt{-3}}{6}q\Phi_3'\Phi_4\Phi_6''$	$q\Phi_4$	1	$1\wedge\zeta_3$
$\phi_{3,2}$	$q^2\Phi_3\Phi_6$	$q^2\Phi_3\Phi_6$	1	
$Z_3:2$	$\frac{-\sqrt{-3}}{3}q\Phi_1\Phi_2\Phi_4$	0	ζ_3^2	$\zeta_3\wedge\zeta_3^2$
$Z_3:11$	$\frac{-\sqrt{-3}}{3}q^4\Phi_1\Phi_2\Phi_4$	0	ζ_3^2	$\zeta_3\wedge-\zeta_3$
G_4	$\frac{-1}{2}q^4\Phi_1^2\Phi_3$	0	-1	$-\zeta_3^2\wedge-1$

As in the Weyl group case, unipotent characters have a degree, but this time it is not given by a polynomial with rational coefficients, but with coefficients in $\mathbb{Q}[\sqrt{-3}]$, which is the smallest subfield of $\mathbb{C}$ over which G_4 can be realized.

References

[Bonnafe2005] C. Bonnafé, "Quasi-isolated elements in reductive groups", *Comm. in Algebra* **33** (2005), 2315–2337.

[Bonnafe-Michel2011] C. Bonnafé and J. Michel "Computational proof of the Mackey formula for $q > 2$", *Journal of Algebra* **327** (2011), 506–526.

[Broue-Malle-Michel2013] M. Broué, G. Malle and J. Michel, "Split Spetses for primitive reflection groups." `arXiv:1204.5846v1 [math.GR]`, to appear in *Astérisque*.

[Carter1978] R. Carter, "Centralizers of semisimple elements in finite groups of Lie type", *Proc. LMS* **37** (1978), 491–507.

[Digne-Michel1985] F. Digne and J. Michel "Fonctions $\mathcal{L}$ des variétés de Deligne-Lusztig et descente de Shintani", *Mémoires de la SMF* **20** (1985).

[Deriziotis1984] D. I. Deriziotis, "Conjugacy classes and centralizers of semisimple elements in finite groups of Lie type", *Vorl. Fach. Math. Universität Essen* **11** (1984).

[Geck2011] M. Geck,"Some applications of CHEVIE to the theory of algebraic groups. *Carpath. J. Math.* **27** (2011), 64–94.

[Geck2012] M. Geck,"PyCox: Computing with (finite) Coxeter groups and Iwahori–Hecke algebras.", *LMS J. of Comput. and Math.* **15** (2012), 231–256.

[GHLMP1996] M. Geck, G. Hiss, F. Lübeck, G. Malle, G. Pfeiffer, "CHEVIE — a system for computing and processing generic character tables.", Computational methods in Lie theory (Essen, 1994). *Appl. Algebra Engrg. Comm. Comput.* **7** (1996), 175–210.

[Geck-Michel1997] M. Geck and J. Michel, " "Good" elements of finite Coxeter groups and representations of Iwahori–Hecke algebras", *Proc. London Math. Soc.* **74 No 3** (1997), 275–305.

[Geck-Pfeiffer1993] M. Geck and G. Pfeiffer, "On the irreducible characters of Hecke algebras", *Adv. Math.* **102** (1993), 79–94.

[Gonzalez-Gebhardt2010] "Solving the conjugacy problem in Garside groups by cyclic sliding", *J. Symb. Computation* **45** (2010), 629–656.

[He-Nie2012] X. He and S. Nie, "Minimal length elements of finite Coxeter groups", *Duke math. J.* **161** (2012), 2945–2967.

[Kessar-Malle2013] R. Kessar and G. Malle, "Quasi-isolated blocks and Brauer's height zero conjecture." *Ann. of Math.* **178** (2013), 321–384.

[Lusztig2011] G. Lusztig "On certain varieties attached to a Weyl group element", *Bull. Inst. Math. Acad. Sinica* **6** (2011), 377–414.

[Lusztig2011b] G. Lusztig, "From conjugacy classes in the Weyl group to unipotent classes", *Representation Theory* **15** (2011), 494–530.

[Lusztig2013] G. Lusztig, "On conjugacy classes in a reductive group", `Arxiv:1305.7168`.

[Malle1995] G. Malle, "Unipotente Grade imprimitiver komplexer Spiegelungsgruppen", *J. Algebra* **177** (1995), 768–826.

[Malle-Michel2010] G. Malle and J. Michel "Constructing representations of Hecke algebras for complex reflection groups", *London mathematical society journal of computation and mathematics* **13** (2010), 426–450.

[Michel] J. Michel, `www.math.jussieu.fr/~jmichel/chevie`

[Sage] William A. Stein et al., "Sage Mathematics Software, The Sage Development Team", `http://www.sagemath.org`.

UFR de Mathématiques, Université Denis Diderot - Paris 7, Bat. Sophie Germain, case 7012 – 75013 Paris Cedex 13, France.

E-mail address: `jmichel@math.jussieu.fr`