MATHEMATICS OF COMPUTATION
Volume 83, Number 289, September 2014, Pages 2509–2526
S 0025-5718(2014)02812-3
Article electronically published on February 19, 2014

CLASSIFYING SEMISIMPLE ORBITS OF θ-GROUPS

WILLEM A. DE GRAAF AND FRANCESCO ORIENTE

Abstract. θ-groups are a class of reductive algebraic groups arising from $\mathbb{Z}/m\mathbb{Z}$-gradings of simple Lie algebras. They were introduced by Vinberg in the 70s, who developed the theory of their orbits. In this paper we describe algorithms to compute certain objects arising in this theory, namely a Cartan subspace, and the little Weyl group. We have implemented the algorithms in the language of the computer algebra system Magma. Practical experiences with the implementations are discussed.

1. Introduction

Let G be a linear algebraic group, acting on a vector space V. An orbit of G is said to be closed if it is closed in the Zariski topology. In general it is very hard to classify the closed orbits of G. In this paper we consider this problem for the case where G is a connected reductive algebraic group over $\mathbb{C}$ and (G, V) is a θ-representation (see below for the definitions). These form a wide class of linear reductive algebraic groups, where there is a well developed theory of closed orbits, which enables us to develop algorithms for classifying them.

Example 1.1. Maybe the most well-known example of the kind of group action that we deal with is provided by $G = \mathrm{GL}_n(\mathbb{C})$ acting by conjugation on $V = M_n(\mathbb{C})$ (the space of $n \times n$-matrices). In this case an orbit is closed if and only if it consists of semisimple elements. Therefore they are also called semisimple orbits. Every semisimple orbit has a point in the space $D_n(\mathbb{C})$ of diagonal matrices. Moreover, two elements of $D_n(\mathbb{C})$ lie in the same G-orbit if and only if they have the same eigenvalues, or, in other words, if they are conjugate under the symmetric group S_n, which acts on $D_n(\mathbb{C})$ by permuting the diagonal entries.

This example can be generalised by taking G to be a reductive algebraic group over $\mathbb{C}$, and $V = \mathfrak{g}$, its Lie algebra. Then G acts on $\mathfrak{g}$ by the adjoint representation. An $x \in \mathfrak{g}$ is called semisimple if the adjoint map $\mathrm{ad}_{\mathfrak{g}} x$ is semisimple (where $\mathrm{ad}_{\mathfrak{g}} x : \mathfrak{g} \to \mathfrak{g}$ is defined by $\mathrm{ad}_{\mathfrak{g}} x(y) = [x, y]$). Also in this case the closed orbits are the ones that consist of semisimple elements. Let $\mathfrak{h}$ be a fixed Cartan subalgebra of $\mathfrak{g}$; then, since all Cartan subalgebras are G-conjugate (cf. [9], Corollary 16.4), every semisimple orbit has a point in $\mathfrak{h}$. Moreover, two elements of $\mathfrak{h}$ are G-conjugate if and only if they are conjugate under the Weyl group $W = N_G(\mathfrak{h})/Z_G(\mathfrak{h})$, which is a finite group.

Received by the editor April 8, 2011 and, in revised form, October 14, 2012 and January 6, 2013.
2010 *Mathematics Subject Classification.* Primary 17B45, 20G05.

A further generalisation was introduced by Vinberg ([24]). We let G, $\mathfrak{g}$ be as above. Vinberg considered automorphisms θ of $\mathfrak{g}$ of order m. They yield a $\mathbb{Z}/m\mathbb{Z}$-grading of $\mathfrak{g}$,

$$\mathfrak{g} = \mathfrak{g}_0 \oplus \mathfrak{g}_1 \oplus \cdots \oplus \mathfrak{g}_{m-1},$$

where $\mathfrak{g}_i$ is the eigenspace of θ with eigenvalue ω^i, where ω is a primitive m-th root of unity. Let G_0 be the connected subgroup of G with Lie algebra $\mathfrak{g}_0$. Then G_0 is reductive and acts on $\mathfrak{g}_1$. The group G_0, together with its action on $\mathfrak{g}_1$, is called a θ-group. The representation of G_0 on $\mathfrak{g}_1$ is also called a θ-representation. Also in this situation an orbit is closed if and only if it consists of elements of $\mathfrak{g}_1$ that are semisimple (as elements of $\mathfrak{g}$).

Example 1.2. Let $\mathfrak{g} = \mathfrak{sl}_5(\mathbb{C})$; then $G = \mathrm{SL}_5(\mathbb{C})$ acts on $\mathfrak{g}$ by conjugation. Let $\omega \in \mathbb{C}$ be a primitive third root of unity. Let θ be the automorphism of order 3 of $\mathfrak{g}$, given by the following matrix:

$$\begin{pmatrix} 1 & \omega & \omega^2 & \omega^2 & \omega^2 \\ \omega^2 & 1 & \omega & \omega & \omega \\ \omega & \omega^2 & 1 & 1 & 1 \\ \omega & \omega^2 & 1 & 1 & 1 \\ \omega & \omega^2 & 1 & 1 & 1 \end{pmatrix}.$$

Here, if on position (i,j) there is ω^k, then $\theta(e_{i,j}) = \omega^k e_{i,j}$, where $e_{i,j}$ is the matrix with a 1 on position (i,j) and zeroes elsewhere. Let $h_i = e_{i,i} - e_{i+1,i+1}$. Then we see that $\mathfrak{g}_0$ is spanned by h_1, h_2, h_3, h_4, $e_{i,j}$ for $3 \leq i \neq j \leq 5$; which means that $\mathfrak{g}_0 \cong \mathfrak{sl}_3(\mathbb{C}) \oplus T_2$ (where T_2 denotes the subalgebra spanned by h_1, h_2). Furthermore, $\mathfrak{g}_1$ is spanned by $e_{1,2}$, $e_{2,3}$, $e_{2,4}$, $e_{2,5}$, $e_{3,1}$, $e_{4,1}$ and $e_{5,1}$. Finally, $\mathfrak{g}_2$ is spanned by $e_{1,3}$, $e_{1,4}$, $e_{1,5}$, $e_{2,1}$, $e_{3,2}$, $e_{4,2}$ and $e_{5,2}$.

Following [24], a Cartan subspace of $\mathfrak{g}_1$ is defined to be a maximal subspace consisting of commuting semisimple elements. Vinberg ([24], Theorem 1) showed that all Cartan subspaces are conjugate under G_0, and their dimension is called the rank of $\mathfrak{g}_1$ (or rather of the pair $(G_0, \mathfrak{g}_1)$, or of the θ-representation afforded by G_0 and $\mathfrak{g}_1$). In particular, it follows that every semisimple orbit has a point in a fixed Cartan subspace $\mathfrak{c} \subset \mathfrak{g}_1$. Moreover, two elements of $\mathfrak{c}$ lie in the same G_0-orbit if and only if they are conjugate under the little Weyl group $W_{\mathfrak{c}} = N_{G_0}(\mathfrak{c})/Z_{G_0}(\mathfrak{c})$.

Vinberg also showed that $W_{\mathfrak{c}}$ is a complex reflection group. By the Chevalley-Shephard-Todd theorem this implies that the ring of invariants $\mathbb{C}[\mathfrak{c}]^{W_{\mathfrak{c}}}$ is freely generated by r fundamental invariants, where r is the rank of $\mathfrak{g}_1$. Moreover, by [24], §4, different semisimple orbits are separated by the polynomial invariants of $W_{\mathfrak{c}}$ on $\mathfrak{c}$. This means that two elements of $\mathfrak{c}$ lie in different $W_{\mathfrak{c}}$-orbits (and hence in different G_0-orbits) if and only if there is at least one fundamental invariant taking different values on these elements.

For this reason we consider the classification of the semisimple orbits to be complete if we have a Cartan subspace, generators of the corresponding little Weyl group, and its fundamantal invariants. Fortunately there are algorithms to compute the fundamental invariants of $W_{\mathfrak{c}}$, given generators of this group (cf. [3], [22]). (We remark that it is much harder to obtain the invariants of G_0 acting on $\mathfrak{g}_1$; see [2] for an example in type E_6.) Therefore, the problems that we consider in this paper

are given a semisimple Lie algebra $\mathfrak{g}$ over $\mathbb{C}$ and a finite order automorphism θ of it, to

- find a Cartan subspace $\mathfrak{c}$ of $\mathfrak{g}_1$,
- find generators for the little Weyl group $W_{\mathfrak{c}} = N_{G_0}(\mathfrak{c})/Z_{G_0}(\mathfrak{c})$.

For a given θ-group it is possible to try and solve these problems using ad-hoc arguments; we refer to [4], [17], [26] for examples of this. In this paper we develop a more systematic approach using computational techniques. We remark that for all θ-groups, the isomorphism type of $W_{\mathfrak{c}}$ is known (combine [24], [14], [15], [19]). Our algorithms go beyond finding the isomorphism type, in the sense that they construct a basis of a Cartan subspace $\mathfrak{c}$, and generators of the little Weyl group $W_{\mathfrak{c}}$, given as matrices with respect to the given basis of $\mathfrak{c}$.

For this we first need to say a bit about our computational setup. We assume that $\mathfrak{g}$ is given by a multiplication table relative to a Chevalley basis. Then all structure constants in the multiplication table are integers. For different Chevalley bases these structure constants have the same absolute value, so can differ only by their signs. Hence constructing a multiplication table is equivalent to obtaining a valid choice for the signs. This way of constructing a semisimple Lie algebra was first proposed by Tits ([23]). A simpler method is given in the book by Kac ([12], see also [5]). The conjugacy classes (in $\mathrm{Aut}(\mathfrak{g})$) of the finite order automorphisms of $\mathfrak{g}$ have been classified by Kac ([11], see also [7]) in terms of so-called Kac diagrams ([11], see also [25], [7], [18]). Since a Kac diagram is an affine Dynkin diagram where the nodes are labeled by non-negative integers satisfying precise conditions, it is quite straightforward to enumerate all Kac diagrams that yield an automorphism of $\mathfrak{g}$ of a fixed order m. From the Kac diagram of an automorphism θ of order m it is straightforward to compute the matrix of θ relative to the given basis of $\mathfrak{g}$ (this follows directly from the description of the finite order automorphisms as, for example, given in [7], Chapter X, Theorem 5.15, see also the next section). For this we need to work over the field $\mathbb{Q}(\omega)$, where ω is a primitive m-th root of unity. Then by linear algebra we can construct bases of $\mathfrak{g}_0$ and $\mathfrak{g}_1$. If θ is an inner automorphism then $\mathfrak{g}_0$ and $\mathfrak{g}_1$ are spanned by root spaces of $\mathfrak{g}$. So, although the matrix of θ has coefficients in $\mathbb{Q}(\omega)$, the spaces $\mathfrak{g}_0$, $\mathfrak{g}_1$ are defined over $\mathbb{Q}$. In fact, this also happens for outer automorphisms, except for some cases in type D_4.

In Section 3 we give a method for constructing a Cartan subspace in $\mathfrak{g}_1$. It is based on the following observations: If $\mathfrak{g}_1$ contains semisimple elements, then the set of non-nilpotent elements is dense in $\mathfrak{g}_1$; so by trying random elements we quickly find a non-nilpotent element, and after computing its Jordan decomposition, a semisimple element. We compute the centraliser of this semisimple element, and reduce the problem to finding a Cartan subspace in there. On the other hand, if $\mathfrak{g}_1$ only contains nilpotent elements, then it has a dense G_0-orbit. We can check this by also trying random elements, and computing the dimension of their orbits. So the algorithm uses random choices; however, this does not affect the correctness of the output. It only influences the running time. At the end of Section 3 we give some running times of the algorithm with some sample inputs, obtained with an implementation of the algorithm in the computer algebra system Magma ([1]).

In Section 4 we describe methods to compute generators of the little Weyl group $W_{\mathfrak{c}}$. First we construct an approximation $\mathcal{C}_{\mathfrak{c}}$, i.e., a finite group containing $W_{\mathfrak{c}}$ as a subgroup. Then we work with explicit elements of G in order to check whether for a given $a \in \mathcal{C}_{\mathfrak{c}}$ there is a $g \in N_{G_0}(\mathfrak{c})$ that projects to a. In order to check whether

a given $g \in G$ lies in G_0 we need G to be simply connected. So if $\mathfrak{g}$ is of type E_8 we can take G to be the automorphism group of $\mathfrak{g}$. For other types we consider a suitable representation ρ of $\mathfrak{g}$, and let G be the connected algebraic group with Lie algebra $\rho(\mathfrak{g})$.

Finally, in the last section we comment on our implementation of the algorithms of Section 4 in Magma. Also we discuss how the algorithm behaves in practice, by giving some timings on sample inputs, and other data relevant to the performance of the algorithm. At the end of the section we describe the output of our algorithms for one θ-group arising from the Lie algebra of type E_6.

We do not give estimates for the complexity of our algorithms. It is likely that the algorithm for computing a Cartan subspace runs in polynomial time. However, this algorithm uses random choices, so in order to prove polynomiality we need some estimate on the expected number of random choices that are carried out, but we do not have such an estimate; although in practice very few choices are needed and the algorithm appears to run in time polynomial in $\dim \mathfrak{g}$. Regarding the algorithm for obtaining generators of the little Weyl group the problems are even greater. In that algorithm it is sometimes necessary to work over extensions of $\mathbb{Q}$, but we have no bounds on the degree of the extension needed.

The Magma code implementing the algorithms described here is available on the website: `http://www.science.unitn.it/ degraaf/w0.html`

2. Preliminaries

Here we briefly recall some notions regarding semisimple Lie algebras, and their automorphisms. For more detailed accounts we refer to standard works in the literature, such as [9], [10], [25].

Let $\mathfrak{g}$ be a semisimple Lie algebra over $\mathbb{C}$, with fixed Cartan subalgebra $\mathfrak{h}$. Let $\Phi \subset \mathfrak{h}^*$ denote the root system of $\mathfrak{g}$, and let $\Delta = \{\alpha_1, \ldots, \alpha_l\}$ be a fixed basis of it. For $\alpha \in \Phi$,

$$\mathfrak{g}_\alpha = \{x \in \mathfrak{g} \mid [h, x] = \alpha(h)x \text{ for } h \in \mathfrak{h}\}$$

denotes the corresponding root space.

The Killing form κ of $\mathfrak{g}$ is non-degenerate. Hence we get a bijection $\mathfrak{h}^* \to \mathfrak{h}$, $\mu \mapsto \hat{\mu}$, where $\kappa(h, \hat{\mu}) = \mu(h)$ for all $h \in \mathfrak{h}$. Using this bijection, κ induces a non-degenerate bilinear form $(\ ,\ )$ on $\mathfrak{h}^*$. For $\mu \in \mathfrak{h}^*$ we set

$$\mu^\vee = \frac{2\hat{\mu}}{(\mu, \mu)} \in \mathfrak{h}.$$

Let G be a connected algebraic group with Lie algebra $\mathfrak{g}$. Then the Weyl group of $\mathfrak{g}$ is $N_G(\mathfrak{h})/Z_G(\mathfrak{h})$. This group is isomorphic to the Weyl group W of the root system Φ; a reflection s_α acts on $\mathfrak{h}$ by $s_\alpha(h) = h - \alpha(h)\alpha^\vee$.

We use canonical sets of generators of $\mathfrak{g}$ (cf. [10]), which correspond to bases of Φ. A canonical set of generators corresponding to the basis Δ consists of elements $x_1, \ldots, x_l$, $y_1, \ldots, y_l$, $h_1, \ldots, h_l$, where $x_i \in \mathfrak{g}_{\alpha_i}$, $y_i \in \mathfrak{g}_{-\alpha_i}$, the $h_i \in \mathfrak{h}$ form a basis of $\mathfrak{h}$ and

$$\begin{aligned}
[h_i, h_j] &= 0,\\
[x_i, y_j] &= \delta_{ij} h_i,\\
[h_i, x_j] &= \langle \alpha_j, \alpha_i^\vee \rangle x_j,\\
[h_i, y_j] &= -\langle \alpha_j, \alpha_i^\vee \rangle y_j.
\end{aligned}$$

In particular, this means that $h_i = \alpha_i^\vee$. For the proof of the following lemma we refer to [5], Section 5.11.

Lemma 2.1. *Let $\{\tilde{\alpha}_1, \ldots, \tilde{\alpha}_l\}$ be a basis of Φ. Let $1 \leq i \leq l$ and let $\tilde{x}_i$ be a non-zero element of $\mathfrak{g}_{\tilde{\alpha}_i}$. Then there exists a non-zero $\tilde{y}_i \in \mathfrak{g}_{-\tilde{\alpha}_i}$ such that, by setting $\tilde{h}_i = [\tilde{x}_i, \tilde{y}_i]$, we have $[\tilde{h}_i, \tilde{x}_i] = 2\tilde{x}_i$. Moreover, the $\tilde{x}_i$, $\tilde{y}_i$, $\tilde{h}_i$ for $1 \leq i \leq l$ form a canonical generating set of $\mathfrak{g}$.*

We note that we can find the $\tilde{y}_i$ as in the lemma by solving a set of linear equations. Hence the lemma provides an algorithm for finding a canonical generating set, given a basis of Φ.

An automorphism of $\mathfrak{g}$ can be described by giving the images of the elements of a canonical generating set. Conversely, given any canonical generating set, x_i', y_i' and h_i' with the same commutation relations as above, then the map sending x_i to x_i', y_i to y_i', h_i to h_i' is an automorphism of $\mathfrak{g}$.

The conjugacy classes of the finite order automorphisms of $\mathfrak{g}$ are parametrised by Kac diagrams. Here we briefly indicate what a Kac diagram is, and how to get a representative of the corresponding conjugacy class, focussing on inner automorphisms. For a much more detailed description we refer to [25], Chapter 3, [7], Section X.5, or [18]. Let α_0 denote the lowest root of Φ. Then the Dynkin diagram of the roots $\alpha_0, \ldots, \alpha_l$ is called the extended Dynkin diagram of Φ (or of $\mathfrak{g}$). Let $n_i \in \mathbb{Z}$ be such that $\alpha_0 = -\sum_{i=0}^{l} n_i \alpha_i$ and set $n_0 = 1$. Let $s_0, \ldots, s_l$ be non-negative integers with $\gcd(s_0, \ldots, s_l) = 1$, and set $m = \sum_{i=0}^{l} n_i s_i$. Let $\omega \in \mathbb{C}$ be a primitive m-th root of unity. Let x_0 be a non-zero element of $\mathfrak{g}_{\alpha_0}$. Then $x_0, \ldots, x_l$ generate $\mathfrak{g}$. Moreover, $x_i \mapsto \omega^{s_i} x_i$ $(0 \leq i \leq l)$ defines an automorphism of $\mathfrak{g}$, of order m. The Kac diagram of this automorphism is the extended Dynkin diagram with labels $s_0, \ldots, s_l$. In terms of the canonical generating set the automorphism is given by $x_i \mapsto \omega^{s_i} x_i$, $y_i \mapsto \omega^{-s_i} y_i$, $h_i \mapsto h_i$ $(1 \leq i \leq l)$.

Now we describe how an element of the Weyl group can be lifted to an automorphism of $\mathfrak{g}$. Let $w \in W$; then

$$w(h_i) = w(\alpha_i^\vee) = w(\alpha_i)^\vee.$$

Let x_i' be a non-zero element of $\mathfrak{g}_{w(\alpha_i)}$, and let $y_i' \in \mathfrak{g}_{-w(\alpha_i)}$ be such that $[[x_i', y_i'], x_i'] = 2x_i'$. Set $h_i' = [x_i', y_i']$. Then by Lemma 2.1 the x_i', y_i', h_i' form a canonical generating set of $\mathfrak{g}$. Hence the map sending $x_i \to x_i'$, $y_i \to y_i'$, $h_i \to h_i'$ extends to an automorphism σ_w of $\mathfrak{g}$. Moreover, $h_i' = w(\alpha_i)^\vee = w(\alpha_i^\vee) = w(h_i)$. Hence σ_w is an automorphism of $\mathfrak{g}$, stabilising $\mathfrak{h}$, and such that its restriction to $\mathfrak{h}$ coincides with w.

Now let σ be an automorphism of $\mathfrak{g}$, stabilising $\mathfrak{h}$, and such that its restriction to $\mathfrak{h}$ is the identity. Then $\sigma(h_i) = h_i$. Moreover, the $\sigma(x_i)$, $\sigma(y_i)$ lie in $\mathfrak{g}_{\alpha_i}$, $\mathfrak{g}_{-\alpha_i}$, respectively. So $\sigma(x_i) = \lambda_i x_i$, $\sigma(y_i) = \mu_i y_i$. Since $[x_i, y_i] = h_i$ we get that $\lambda_i \mu_i = 1$. Conversely, if we take arbitrary $\lambda_i \in \mathbb{C}$, $\lambda_i \neq 0$, and set $\mu_i = \frac{1}{\lambda_i}$, then the h_i, $\lambda_i x_i$, $\mu_i y_i$ form a canonical generating set of $\mathfrak{g}$. So we get a corresponding automorphism of $\mathfrak{g}$, that is the identity on $\mathfrak{h}$. This gives us an explicit description of those automorphisms that are the identity on $\mathfrak{h}$. They depend on l non-zero parameters.

3. Constructing a Cartan subspace

In this section we let $\mathfrak{g}$ be a reductive Lie algebra with a $\mathbb{Z}/m\mathbb{Z}$-grading, $\mathfrak{g} = \bigoplus_{i=0}^{m-1} \mathfrak{g}_i$. A Cartan subspace of $\mathfrak{g}_1$ is a maximal subspace consisting of semisimple commuting elements. Here we describe an algorithm for computing such a Cartan subspace.

In the algorithm we need to compute the Jordan decomposition of elements of semisimple Lie algebras. This can be done as follows. Let $x \in \mathfrak{s}$, with $\mathfrak{s}$ a semisimple Lie algebra. Compute the Jordan decomposition of the adjoint map $\mathrm{ad}_{\mathfrak{s}} x = S + N$, with S semisimple and N nilpotent. Then there are unique $x_s, x_n \in \mathfrak{s}$ such that $S = \mathrm{ad}_{\mathfrak{s}} x_s$ and $N = \mathrm{ad}_{\mathfrak{s}} x_n$. Moreover, we can find these elements by solving systems of linear equations. Furthermore, the Jordan decomposition of x is $x = x_s + x_n$.

Next we give the algorithm for computing a Cartan subspace. In the algorithm a parameter R is used. It is a positive integer, fixed throughout. The behaviour of the algorithm depends on it. For the explanation of the algorithm we refer to the proof of Proposition 3.1.

Algorithm 1. *Input: a reductive $\mathbb{Z}/m\mathbb{Z}$-graded Lie algebra $\mathfrak{g}$.*
Output: a Cartan subspace of $\mathfrak{g}_1$.

(1) *Let $\mathfrak{s} = [\mathfrak{g}, \mathfrak{g}]$ and let $\mathfrak{r}$ be the centre of $\mathfrak{g}$. Let $\mathfrak{s}_1 = \mathfrak{s} \cap \mathfrak{g}_1$.*
(2) *If $\mathfrak{s}_1 = 0$ then return $\mathfrak{g}_1$.*
(3) *Let $x_1, \ldots, x_m$ be a basis of $\mathfrak{s}_1$, and set $x = \sum_{i=1}^{m} c_i x_i$, where the c_i are drawn randomly uniformly and independently from the set $\Omega = \{0, \ldots, R\}$.*
(4) *If x is not nilpotent, then execute Step 4(a). Otherwise execute Step 4(b).*
   (a) *(x not nilpotent) Compute the Jordan decomposition $x = x_s + x_n$. Let $\tilde{\mathfrak{g}}$ be the centraliser of x_s in $\mathfrak{g}$. Return the output of the algorithm when applied recursively to $\tilde{\mathfrak{g}}$, with the grading induced by the grading of $\mathfrak{g}$.*
   (b) *(x is nilpotent) Set $\mathfrak{s}_0 = \mathfrak{s} \cap \mathfrak{g}_0$. Compute $[\mathfrak{s}_0, x]$; if the dimension of this space is equal to the dimension of $\mathfrak{s}_1$, then return $\mathfrak{g}_1 \cap \mathfrak{r}$. Otherwise return to Step 3.*

Proposition 3.1. *If Algorithm 1 terminates, then it returns a Cartan subspace of $\mathfrak{g}_1$. Moreover, the probability that the algorithm terminates tends to 1 as R tends to ∞.*

Proof. The first statement is shown by induction on the dimension of $\mathfrak{g}$, the induction hypothesis being that the statement is true for all reductive graded Lie algebras of dimension less than $\dim \mathfrak{g}$.

We note that since $\mathfrak{g}$ is reductive, $\mathfrak{g} = \mathfrak{s} \oplus \mathfrak{r}$, with $\mathfrak{s}$ semisimple. Let θ be the automorphism corresponding to the grading of $\mathfrak{g}$, i.e., $\theta(x) = \omega^i x$ for $x \in \mathfrak{g}_i$, where ω is a primitive m-th root of unity. Then θ stabilises $\mathfrak{s}$ and $\mathfrak{r}$. Hence $\mathfrak{s}$ is also graded, where the grading is induced by the grading of $\mathfrak{g}$. Furthermore, $\mathfrak{g}_1 = (\mathfrak{g}_1 \cap \mathfrak{s}) \oplus (\mathfrak{g}_1 \cap \mathfrak{r})$. Note that $\mathfrak{g}_1 \cap \mathfrak{r}$ is always contained in a Cartan subspace. So if $\mathfrak{g}_1 \cap \mathfrak{s} = 0$, then $\mathfrak{g}_1 \subset \mathfrak{r}$ and hence $\mathfrak{g}_1$ is a Cartan subspace of $\mathfrak{g}_1$.

If the x found in Step 3 is not nilpotent, then its semisimple part, x_s, also lies in $\mathfrak{s}$. Therefore, its centraliser, $\tilde{\mathfrak{g}}$, is strictly contained in $\mathfrak{g}$. Furthermore, it is reductive as x_s is semisimple. Hence by induction the algorithm, with input $\tilde{\mathfrak{g}}$, computes a Cartan subspace, $\mathfrak{c}$ of $\tilde{\mathfrak{g}}_1$. Then $\mathfrak{c}$ is a Cartan subspace of $\mathfrak{g}$. Indeed, x_s is contained in a Cartan subspace $\mathfrak{c}'$ of $\mathfrak{g}_1$. But $\mathfrak{c}'$ is also contained in $\tilde{\mathfrak{g}}$. Now x_s is an eigenvector of θ, so it is also an eigenvector of θ^{-1}. This implies that $\tilde{\mathfrak{g}}$

is θ-stable. Therefore, $\tilde{\mathfrak{g}}_1 = \mathfrak{g}_1 \cap \tilde{\mathfrak{g}}$. Now $y \in \tilde{\mathfrak{g}}$ is semisimple as an element of $\tilde{\mathfrak{g}}$ if and only if it is semisimple as an element of $\mathfrak{g}$. It follows that $\mathfrak{c}'$ is a maximal commutative subspace of $\tilde{\mathfrak{g}}_1$ consisting of semisimple elements. In other words, it is a Cartan subspace of $\tilde{\mathfrak{g}}_1$. So $\mathfrak{c}'$ is $\widetilde{G}_0$-conjugate to $\mathfrak{c}$ (where $\widetilde{G}_0$ is the subgroup of G corresponding to $\tilde{\mathfrak{g}}_0$). Hence $\mathfrak{c}$ is a Cartan subspace of $\mathfrak{g}_1$.

If, on the other hand, x is nilpotent, then $[\mathfrak{s}_0, x]$ is the tangent space to the orbit $S_0 \cdot x$, where $S_0 \subset G_0$ is the group corresponding to $\mathfrak{s}_0$. If the dimension of this orbit equals the dimension of $\mathfrak{s}_1$, then $\mathfrak{s}_1$ contains an open nilpotent orbit (namely $S_0 \cdot x$). This implies that $\mathfrak{s}_1$ consists entirely of nilpotent elements. Indeed, the set of non-nilpotent elements in $\mathfrak{s}_1$ is open. If it is non-empty, then it has to have a non-empty intersection with the open nilpotent orbit, which is not possible. Hence in this case $\mathfrak{g}_1 \cap \mathfrak{r}$ is a Cartan subspace of $\mathfrak{g}_1$.

Now we turn to the question of termination. If the algorithm does not terminate, then the algorithm enters an infinite loop in steps 3 to 5. This means that the element found in Step 3 is always nilpotent. But for large R that means that $\mathfrak{s}_1$ consists of nilpotent elements, as the set of non-nilpotent elements is open (and therefore dense, when non-empty). By [24], §2.6, this implies that $\mathfrak{s}_1$ has an open nilpotent orbit. Again for large R that means that at some point a random x is found that lies in the open nilpotent orbit. But that contradicts the assumption that we have an infinite loop. □

Remark 3.2. Proposition 3.1 ensures that the algorithm works when R is large enough. In practice however, it turns out that it is enough to take a small R. In our implementation we have used $R = 2$, and hence $\Omega = \{0, 1\}$. On some occasions this makes it necessary to perform more iterations; but the upside of this approach is that the coefficients of the elements involved are much nicer. Of course, with such a small R it is not guaranteed that the algorithm terminates. This can be remedied by increasing R if, after a few rounds of the iteration, no "good" element has been found (i.e., a semisimple element, or a nilpotent element lying in a dense orbit).

Remark 3.3. Algorithm 1 may not work for reductive Lie algebras in characteristic $p > 0$. However, using some results of Levy ([14]) it is possible to formulate a variant of the algorithm that also works in characteristic $p > 2$. Levy proved ([14], Lemma 1.12) that the following are equivalent:

- $\mathfrak{g}_1$ contains no non-central semisimple elements.
- $\mathfrak{g}_1 = \mathfrak{s} \oplus \mathfrak{n}$, where $\mathfrak{s}$ and $\mathfrak{n}$ are the sets of semisimple and nilpotent elements of $\mathfrak{g}_1$ respectively, and $\mathfrak{s} \subset \mathfrak{z}(\mathfrak{g})$ (the centre of $\mathfrak{g}$).

Moreover, suppose that $\mathfrak{g}_1$ has non-central semisimple elements. Let $x \in \mathfrak{g}_1$ be a randomly chosen element, with Jordan decomposition $x = x_s + x_n$. Then from [14], Lemma 2.16, it follows that x_s is non-central with high probability.

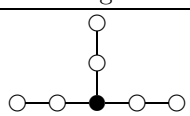
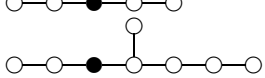
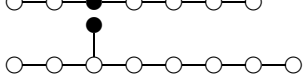
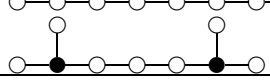
We proceed as follows. Let x be a randomly chosen element of $\mathfrak{g}_1$, with Jordan decomposition $x = x_s + x_n$. If x_s is non-central, then we construct $\tilde{\mathfrak{g}}$ as in Algorithm 1, and apply the algorithm to $\tilde{\mathfrak{g}}$. If x_s is central, then we compute the Jordan decomposition of all basis elements of $\mathfrak{g}_1$. Let $\mathfrak{s}$, $\mathfrak{n}$ be the subspaces spanned by all semisimple and nilpotent parts, respectively. If $\mathfrak{g}_1 = \mathfrak{s} \oplus \mathfrak{n}$ and $\mathfrak{s}$ is contained in the centre of $\mathfrak{g}$, and all elements of $\mathfrak{n}$ are nilpotent, then the output is $\mathfrak{s}$. If one of these conditions is not satisfied, then we choose another random element, and repeat the process.

If the characteristic is small, then the set Ω used in the algorithm may be too small. In that case one can replace Ω by a finite field of characteristic p, of at least R elements.

Example 3.4. Let $\mathfrak{g}$ and θ be as in Example 1.2. Consider the element $u = e_{1,2} + e_{2,3} + e_{3,1} \in \mathfrak{g}_1$. Its minimum polynomial is $x^4 - x$, and therefore u is semisimple. The centraliser of u in $\mathfrak{g}$, $\mathfrak{g}_u$, is spanned by $e_{4,5}$, $e_{5,4}$, h_4, $h_1+2h_2+3h_3$, u, $e_{1,3}+e_{2,1}+e_{3,2}$. Here the first four elements lie in $\mathfrak{g}_0$, and the last two in $\mathfrak{g}_1$ and $\mathfrak{g}_2$, respectively. We have that $\mathfrak{g}_u = \mathfrak{s} \oplus \mathfrak{r}$, where $\mathfrak{s}$ is spanned by $e_{4,5}$, $e_{5,4}$, h_4. We see that $\mathfrak{s}$ is contained in $\mathfrak{g}_0$. So a Cartan subspace of $\mathfrak{g}_u$, and hence of $\mathfrak{g}$, is $\mathfrak{g}_1 \cap \mathfrak{r}$, which is spanned by u. In particular, the rank of $\mathfrak{g}_1$ is 1.

We have implemented the algorithm in the computer algebra system Magma ([1]). In Table 1 we give the running times on some sample inputs. The timings have been obtained on a 3.16 GHz processor.

Table 1. Running times (in seconds) of the algorithm for computing a Cartan subspace, with input automorphisms of the Lie algebras of types E_6, E_7, E_8, D_8. The second column has the Kac diagram of the automorphism. The labels are 0 and 1; a black node has label 1, the other nodes have label 0. The third column lists the rank of $\mathfrak{g}_1$. The last column contains the running time.

type	Kac diagram	rank	time (s)
E_6		3	1.1
E_7		3	5.8
E_8		4	56.3
D_8		4	3.9

We see that the running times increase rather sharply with the dimension of the Lie algebra. The data given here suggests that the running time behaves like $c(\dim \mathfrak{g})^{3.4}$, where c is a constant. This is no surprise, as the algorithm mainly deals with matrices of adjoint maps $\mathrm{ad}_{\mathfrak{g}} x$, so $\dim \mathfrak{g}$ should be the main parameter influencing the running time.

4. The little Weyl group

Here we let $\mathfrak{g}$ be a simple Lie algebra over $\mathbb{C}$, with a finite order automorphism θ. We let G be a simply connected algebraic group with Lie algebra $\mathfrak{g}$. Then there exists an automorphism $\hat{\theta}$ of G, with differential equal to θ. Also, G_0 (which is defined to be the connected algebraic subgroup of G with Lie algebra $\mathfrak{g}_0$) is equal to $G^\theta = \{g \in G \mid \hat{\theta}(g) = g\}$ ([24], §2).

Let $\mathfrak{c}$ be a Cartan subspace of $\mathfrak{g}_1$. In this section we consider the problem of computing the little Weyl group, $W_{\mathfrak{c}} = N_{G_0}(\mathfrak{c})/Z_{G_0}(\mathfrak{c})$. According to a theorem of Vinberg this group is generated by complex reflections that act on the space $\mathfrak{c}$. We

want to find a set of complex reflections, given by their matrices with respect to a fixed basis of $\mathfrak{c}$, that generate $W_{\mathfrak{c}}$. We divide this section into several subsections.

4.1. **Approximating $W_{\mathfrak{c}}$ from above.** In this subsection we describe an algorithm to construct a finite subgroup of $\mathrm{GL}(\mathfrak{c})$ containing $W_{\mathfrak{c}}$. This is based on the construction of a θ-stable Cartan subalgebra, due to Levy ([14]).

Let $\mathfrak{t}$ be the unique minimal algebraic torus in $\mathfrak{g}$ containing $\mathfrak{c}$. Throughout we set $\mathfrak{s} = \mathfrak{z}_{\mathfrak{g}}(\mathfrak{c})$ (the centraliser of $\mathfrak{c}$ in $\mathfrak{g}$). Then $\mathfrak{s}$ is a reductive subalgebra. Moreover, it is θ-stable and hence graded, $\mathfrak{s}_k = \mathfrak{s} \cap \mathfrak{g}_k$.

Lemma 4.1. *Let $\mathfrak{z}(\mathfrak{s})$ denote the centre of $\mathfrak{s}$. Then*

$$\mathfrak{t} = \bigoplus_{\substack{\gcd(k,m)=1 \\ 0 \le k < m}} \mathfrak{g}_k \cap \mathfrak{z}(\mathfrak{s}).$$

Proof. By [24], §3.1 we have $\mathfrak{t} = \bigoplus_{\gcd(k,m)=1} \mathfrak{t} \cap \mathfrak{g}_k$. Since $\mathfrak{s}$ is reductive, its centre consists of semisimple elements. Hence $\mathfrak{z}(\mathfrak{s})$ is an algebraic torus containing $\mathfrak{c}$. Therefore it contains $\mathfrak{t}$. So the left-hand side is contained in the right-hand side.

For the converse, let k be such that $\gcd(k,m) = 1$. Then $\mathfrak{g}_1$ and $\mathfrak{g}_k$ have the same rank (which is the dimension of a maximal commutative subspace consisting of semisimple elements). Indeed: in [24], §3.1 it is shown that $\mathrm{rank}(\mathfrak{g}_1) \le \mathrm{rank}(\mathfrak{g}_k)$, but the same arguments, starting with $\mathfrak{g}_k$ instead of $\mathfrak{g}_1$, show the reverse inequality. This implies that the right-hand side cannot be strictly bigger than the left-hand side (as, by [24], §3.1, all $\mathfrak{t} \cap \mathfrak{g}_k$ have the same dimension). □

The next lemma, as well as Proposition 4.3 are taken from [14].

Lemma 4.2. *Let $\mathfrak{t}_0$ be a Cartan subalgebra of $\mathfrak{s}_0$. Then $\mathfrak{h} = \mathfrak{z}_{\mathfrak{g}}(\mathfrak{t}_0 + \mathfrak{t})$ is a θ-stable Cartan subalgebra of $\mathfrak{g}$ containing $\mathfrak{c}$.*

Proof. We have that $\mathfrak{t}_0$, $\mathfrak{t}$ are θ-stable; hence this holds for $\mathfrak{h}$ as well. Let $\mathfrak{h}'$ be a Cartan subalgebra of $\mathfrak{g}$ containing $\mathfrak{c}$. Then it contains $\mathfrak{t}$ as well. Also, $\mathfrak{h}' \subset \mathfrak{s}$, and hence $\mathfrak{h}'$ is a Cartan subalgebra of $\mathfrak{s}$. But by [7], Chapter X, Lemma 5.3, $\mathfrak{s}_0$ contains a regular element of $\mathfrak{s}$ (i.e., an element of which the centraliser is a Cartan subalgebra of $\mathfrak{s}$). So $\mathfrak{h}'$ is S-conjugate to $\mathfrak{z}_{\mathfrak{s}}(\mathfrak{t}_0)$ (here S is the connected subgroup of G with Lie algebra $\mathfrak{s}$). In particular, also $\mathfrak{z}_{\mathfrak{s}}(\mathfrak{t}_0)$ is a Cartan subalgebra of $\mathfrak{g}$. But $\mathfrak{z}_{\mathfrak{s}}(\mathfrak{t}_0)$ is equal to $\mathfrak{z}_{\mathfrak{g}}(\mathfrak{t}_0 + \mathfrak{t})$. (Note that a semisimple element of $\mathfrak{g}$ that centralises $\mathfrak{c}$ has to centralise $\mathfrak{t}$ as well.) It is clear that $\mathfrak{h}$ contains $\mathfrak{c}$. □

Let $\mathfrak{h}$ be as in the previous lemma. Let $W = N_G(\mathfrak{h})/Z_G(\mathfrak{h})$ be the corresponding Weyl group. Since $\mathfrak{h}$ is θ-stable we can consider the centraliser W^θ of θ in W. Let $w \in W^\theta$ and $x \in \mathfrak{c}$. Then $w(x) \in \mathfrak{h} \cap \mathfrak{g}_1 = \mathfrak{c}$. So w stabilises $\mathfrak{c}$, and we get a map

$$\pi : W^\theta \to \mathrm{GL}(\mathfrak{c})$$

by restriction, $\pi(w) = w|_{\mathfrak{c}}$. Throughout we set $\mathcal{C}_{\mathfrak{c}} = \pi(W^\theta)$.

Proposition 4.3. $W_{\mathfrak{c}} \subset \mathcal{C}_{\mathfrak{c}}$.

Proof. Let $a \in W_{\mathfrak{c}}$, and let $g \in N_{G_0}(\mathfrak{c})$ be a representative of it. Set $L = Z_G(\mathfrak{t})$, which is a subgroup of G with Lie algebra $\mathfrak{l} = \mathfrak{z}_{\mathfrak{g}}(\mathfrak{t})$. Now $g(\mathfrak{t})$ is an algebraic torus of $\mathfrak{g}$ containing $\mathfrak{c}$; hence $g(\mathfrak{t}) = \mathfrak{t}$. This implies that $g^{-1}Lg = L$, and $g(\mathfrak{l}) = \mathfrak{l}$.

Furthermore, $\mathfrak{l}$ is θ-stable, hence it is graded, and $\mathfrak{l}_0$ contains $\mathfrak{t}_0$. Hence $\mathfrak{t}_0$ is a Cartan subalgebra of $\mathfrak{l}_0$. Since g lies in $G_0 = G^\theta$, it leaves the grading invariant,

and hence $g(\mathfrak{t}_0)$ is a Cartan subalgebra of $\mathfrak{l}_0$. So there is an $h \in L_0 \subset G_0$ with $hg(\mathfrak{t}_0) = \mathfrak{t}_0$ (where L_0 denotes the connected subgroup of G with Lie algebra $\mathfrak{l}_0$).

Since $h \in L$ also $hg(\mathfrak{t}) = \mathfrak{t}$. But $\mathfrak{h}$ is the centraliser of $\mathfrak{t}_0 + \mathfrak{t}$. It follows that $hg(\mathfrak{h}) = \mathfrak{h}$. Also, $hg \in G_0 \subset G^\theta$, hence $(hg)|_\mathfrak{h} \in W^\theta$. Moreover, $(hg)|_\mathfrak{c} = g|_\mathfrak{c}$ as $h \in L$. So if we write $w = (hg)|_\mathfrak{h}$, then $\pi(w) = a$. □

Let Φ be the root system of $\mathfrak{g}$ with respect to $\mathfrak{h}$. Let $\Delta = \{\alpha_1, \ldots, \alpha_l\}$ be a fixed basis of Φ. Then W is isomorphic to the Weyl group of Φ, which is generated by the simple reflections $s_i = s_{\alpha_i}$. Such a reflection acts on $\mathfrak{h}$ by $s_i(h) = h - \alpha_i(h)\alpha_i^\vee$. Now an element of W permutes the set $\{\alpha^\vee \mid \alpha \in \Phi\}$. This yields a faithful permutation representation of W. Also, θ permutes the $\alpha^\vee$, and hence we can compute a generating set of W^θ by using permutation group algorithms (see [8], [20]).

We summarise our findings in the following algorithm.

Algorithm 2. *Input: a simple $\mathbb{Z}/m\mathbb{Z}$-graded Lie algebra $\mathfrak{g}$, and a Cartan subspace $\mathfrak{c}$ of $\mathfrak{g}_1$.*
Output: generators of $\mathcal{C}_\mathfrak{c}$.

(1) *Set $\mathfrak{s} = \mathfrak{z}_\mathfrak{g}(\mathfrak{c})$ and $\mathfrak{t} = \bigoplus_{\gcd(k,m)=1} \mathfrak{g}_k \cap \mathfrak{z}(\mathfrak{s})$.*
(2) *Compute a Cartan subalgebra $\mathfrak{t}_0$ of $\mathfrak{s}_0$ and set $\mathfrak{h} = \mathfrak{z}_\mathfrak{g}(\mathfrak{t}_0 + \mathfrak{t})$.*
(3) *Compute the root system Φ of $\mathfrak{g}$ with respect to $\mathfrak{h}$. Set $X = \{\alpha^\vee \mid \alpha \in \Phi\}$.*
(4) *Compute the permutation action of the generators of W, and of θ, on X.*
(5) *Using the permutations computed in the previous step, compute generators of W^θ.*
(6) *For each generator of W^θ compute its restriction to $\mathfrak{c}$. Return the subset of $\mathrm{GL}(\mathfrak{c})$ so obtained.*

Remark 4.4. This algorithm works verbatim when $\mathfrak{g}$ is defined over a field of characteristic $p > 2$. In order to prove Lemma 4.1 for that situation one can use [14], Lemma 1.10. Lemma 4.2 and Proposition 4.3 were proved for characteristic $p > 2$ in [14].

4.2. **Generators of $W_\mathfrak{c}$.** Since we can compute a finite group $\mathcal{C}_\mathfrak{c}$ containing $W_\mathfrak{c}$, the remaining problem is to decide, for a given $a \in \mathcal{C}_\mathfrak{c}$, whether $a \in W_\mathfrak{c}$.

We have the following scheme:

$$N_G(\mathfrak{h}) \xrightarrow{\psi} W \supset W^\theta \xrightarrow{\pi} \mathcal{C}_\mathfrak{c} \supset W_\mathfrak{c}.$$

(Here $\psi : N_G(\mathfrak{h}) \to N_G(\mathfrak{h})/Z_G(\mathfrak{h}) = W$ is the projection.) Let $a \in \mathcal{C}_\mathfrak{c}$. Then $a \in W_\mathfrak{c}$ if and only if there is a $g \in N_G(\mathfrak{h})$ such that

- $\psi(g) \in W^\theta$ and $\pi(\psi(g)) = a$,
- $g \in G_0$ or, equivalently, $\hat{\theta}(g) = g$.

So in order to construct the little Weyl group $W_\mathfrak{c}$ we need to decide whether, for a given $a \in \mathcal{C}_\mathfrak{c}$ there exists a $g \in N_G(\mathfrak{h})$ with the above properties. For this we need an explicit realisation of G. How we obtain it depends on $\mathfrak{g}$. We consider two cases, to which we devote two separate subsections.

4.2.1. *Simply connected automorphsim group.* The first case is when the inner automorphism group of $\mathfrak{g}$ is simply connected. This happens when $\mathfrak{g}$ is of type E_8, F_4, G_2. In these cases we take $G = \mathrm{Aut}(\mathfrak{g})$ and identify $\mathfrak{g}$ and the Lie algebra of

G (which is $\mathrm{ad}\mathfrak{g}$). The reason for taking this particular realisation is that we know well how to work with automorphisms of $\mathfrak{g}$ (see Section 2).

In these cases $\theta \in G = \mathrm{Aut}(\mathfrak{g})$. Hence we can define $\hat{\theta}$ by $\hat{\theta}(g) = \theta g \theta^{-1}$. So we can decide whether a given $g \in G$ lies in G_0 by checking the condition $\hat{\theta}(g) = g$.

Let $a \in \mathcal{C}_{\mathfrak{c}}$, and $K_a = \pi^{-1}(a) = \{w \in W^{\theta} \mid \pi(w) = a\}$. (Note that this is a finite set.) Let $w \in K_a$, and let $\sigma_w \in G$ be an automorphism of $\mathfrak{g}$ stabilising $\mathfrak{h}$, and such that its restriction to $\mathfrak{h}$ is equal to w. (As seen in Section 2, we can effectively compute σ_w.) Then the set of $g \in N_G(\mathfrak{h})$ such that $\psi(g) = w$ is equal to $\sigma_w Z_G(\mathfrak{h})$. Now in Section 2 we have given a description of $Z_G(\mathfrak{h})$: we can represent a generic element g_0 of $Z_G(\mathfrak{h})$ by a matrix (which is its matrix relative to a fixed basis of $\mathfrak{g}$) depending on l independent parameters, $\lambda_1, \ldots, \lambda_l$, $\mu_1, \ldots, \mu_l$, with $\lambda_i \mu_i = 1$. Set $g = \sigma_w g_0$, then $g \in G_0$ if and only if $\theta g = g\theta$. This is equivalent to a system of polynomial equations in λ_i, μ_i. Computing a Gröbner basis we can check whether this system has a solution over $\mathbb{C}$. Here it is important to note that we do not need to solve the equations. If the reduced Gröbner basis is not $\{1\}$, then the system has a solution, and hence $a \in W_{\mathfrak{c}}$. We perform this for all $w \in K_a$; if the resulting reduced Gröbner basis is always equal to $\{1\}$, then $a \notin W_{\mathfrak{c}}$.

In the next algorithm we assume that $G = \mathrm{Aut}(\mathfrak{g})$ is simply connected.

Algorithm 3. *Input: a simple $\mathbb{Z}/m\mathbb{Z}$-graded Lie algebra $\mathfrak{g}$, and a Cartan subspace $\mathfrak{c}$ of $\mathfrak{g}_1$.*
Output: generators of $W_{\mathfrak{c}}$.

(1) *Compute generators of $\mathcal{C}_{\mathfrak{c}}$ (Algorithm 2).*
(2) *Denote by g_0 the generic element of $Z_G(\mathfrak{h})$, depending on l independent parameters $\lambda_1, \ldots, \lambda_l$.*
(3) *Set $\mathcal{G} = \emptyset$.*
(4) *For each $a \in \mathcal{C}_{\mathfrak{c}}$ do*
  (a) *Compute K_a.*
  (b) *For each $w \in K_a$ do*
    (i) *Compute $\sigma_w \in G$ such that $\sigma_w|_{\mathfrak{h}} = w$.*
    (ii) *Construct the set P of polynomial equations equivalent to $\theta\sigma_w g_0 = \sigma_w g_0 \theta$.*
    (iii) *Compute a reduced Gröbner basis of P. If this is not $\{1\}$ then add a to $\mathcal{G}$, and jump to the next element of $\mathcal{C}_{\mathfrak{c}}$.*

4.2.2. *The automorphism group is not simply connected.* Now suppose that $\mathrm{Aut}(\mathfrak{g})^{\circ}$ is not simply connected, which happens when $\mathfrak{g}$ is of type E_6, E_7, or of classical type. We assume that $\mathfrak{g}$ is of one of these types, but not of type D_{2n}. In Remark 4.6 we indicate what has to be changed for $\mathfrak{g}$ of type D_{2n}.

We consider a finite dimensional irreducible representation $\rho : \mathfrak{g} \to \mathfrak{gl}(V)$, and we let G be the connected subgroup of $\mathrm{GL}(V)$ with Lie algebra $\rho(\mathfrak{g})$. We choose ρ such that G is simply connected. Theorem 2.18 of [25] exactly describes the representations that satisfy this condition. For example, for types E_6, E_7 we can take the smallest dimensional representation, where V is of dimension 27 and 56 respectively.

Note that G acts on $\rho(\mathfrak{g})$ by automorphisms $\mathrm{Ad}(g)(\rho(x)) = g\rho(x)g^{-1}$. This yields a surjective map $\varphi : G \to \mathrm{Aut}(\mathfrak{g})^{\circ}$, with kernel equal to the centre of G. (In type E_6, E_7, with ρ as before, this is the group of scalar matrices, with a third,

respectively second, root of unity on the diagonal.) We want to compute generators of $W_{\mathfrak{c}}$ along the lines of Algorithm 3, but with the group G instead of $\mathrm{Aut}(\mathfrak{g})^\circ$. For this we need to solve three problems:

(1) Compute an automorphism $\hat{\theta}$ of G with differential θ.
(2) Compute a generic element g_0 of $Z_G(\mathfrak{h})$, depending on l independent parameters.
(3) For $w \in W$ compute $g_w \in G$ such that $g_w|_{\mathfrak{h}} = w$.

One problem is that we have no way of directly working with elements of G. Therefore we consider the larger group

$$\widetilde{G} = N_{\mathrm{GL}(V)}(\rho(\mathfrak{g})) = \{g \in \mathrm{GL}(V) \mid g\rho(\mathfrak{g})g^{-1} = \rho(\mathfrak{g})\}.$$

We solve the problems outlined above for $\widetilde{G}$, and then show that this yields an algorithm for computing generators of $W_{\mathfrak{c}}$.

Note that $\widetilde{G}$ acts on $\mathfrak{g}$ by automorphisms; we let $\varphi : \widetilde{G} \to \mathrm{Aut}(\mathfrak{g})$ be the corresponding map. (Since $G \subset \widetilde{G}$ it is justified to use the same symbol φ.) For $\sigma \in \mathrm{Aut}(\mathfrak{g})^\circ$ set

$$U_\sigma = \{u \in \mathrm{End}(V) \mid u\rho(x) = \rho(\sigma(x))u \text{ for all } x \in \mathfrak{g}\}.$$

We have that $u \in U_\sigma$ if and only if $u\rho(x) = \rho(\sigma(x))u$, for x in a basis of $\mathfrak{g}$. Therefore, we can compute a basis of U_σ by solving a set of linear equations.

Lemma 4.5. *Let the notation be as above. Set $R = \{\lambda \cdot I_V \mid \lambda \in \mathbb{C}^*\}$, where $I_V \in \mathrm{End}(V)$ is the identity transformation. Then:*

(1) $\dim U_\sigma = 1$.
(2) $\widetilde{G} = GR$.
(3) *Let $u \in U_\sigma$, $u \neq 0$; then $u \in \widetilde{G}$.*

Proof. We have that $U_\sigma \cap \mathrm{GL}(V)$ is equal to the fibre $\varphi^{-1}(\sigma)$. If we let σ be the identity, then this fibre is an algebraic subgroup whose Lie algebra is $\mathfrak{z}_{\mathfrak{gl}(V)}(\rho(\mathfrak{g}))$. Since ρ is irreducible, this is 1-dimensional. Hence, since all fibres are of the same dimension, we get that U_σ is 1-dimensional.

It is obvious that $GR \subset \widetilde{G}$. For the other inclusion, let $g \in \widetilde{G}$. Write $\sigma = \varphi(g)$. Set $\overline{R} = \{\lambda \cdot I_V \mid \lambda \in \mathbb{C}\}$. Then, since $\dim U_\sigma = 1$, we have $U_\sigma = g\overline{R}$. So $\varphi^{-1}(\sigma) = gR$. However, $\varphi^{-1}(\sigma)$ also contains elements of G. So there is $h \in G$, $\lambda \in \mathbb{C}^*$ such that $\lambda g = h$. This shows the second point.

For the third statement, let $g \in \varphi^{-1}(\sigma)$. Then $g \neq 0$ and $g \in U_\sigma$, hence $g = \lambda u$ for some non-zero $\lambda \in \mathbb{C}$. This implies $u \in \widetilde{G}$. □

Now consider the first problem from above. For this we suppose that θ is an inner automorphism, i.e., $\theta \in \mathrm{Aut}(\mathfrak{g})^\circ$. Let $\tilde{\theta} \in U_\theta$, $\tilde{\theta} \neq 0$. By Lemma 4.5 we have $\lambda\tilde{\theta} \in G$, for some non-zero scalar λ, and $\tilde{\theta}\rho(x)\tilde{\theta}^{-1} = \rho(\theta(x))$ for all $x \in \mathfrak{g}$. So we can define an automorphism $\hat{\theta}$ of G by $\hat{\theta}(g) = \tilde{\theta}g\tilde{\theta}^{-1}$. The differential of $\hat{\theta}$ is equal to θ (after identifying $\mathfrak{g}$ and $\rho(\mathfrak{g})$). So $G_0 = \{g \in G \mid \tilde{\theta}g = g\tilde{\theta}\}$.

Now we deal with the second problem. Let σ_0 be a generic automorphism of $\mathfrak{g}$, stabilising $\mathfrak{h}$, such that its restriction to $\mathfrak{h}$ is the identity. As seen before, σ_0 can be represented (with respect to a fixed basis of $\mathfrak{g}$) by a matrix whose entries are polynomials in l independent parameters, $\lambda_1, \ldots, \lambda_l, \mu_1, \ldots, \mu_l$, with $\lambda_i\mu_i = 1$. Now we take λ_i to be the generators of a rational function field F in l indeterminates, and $\mu_i = \lambda_i^{-1}$. Then we compute a basis vector of the space U_{σ_0}, by solving a

set of linear equations over F. Taking a basis vector of U_{σ_0}, we get (up to scalar multiples) a generic element of $Z_G(\mathfrak{h})$ (Lemma 4.5). We denote this element by g_0.

For the third problem, let $w \in W$ and let $\sigma_w \in \mathrm{Aut}(\mathfrak{g})^\circ$ be such that it normalises $\mathfrak{h}$ and its restriction to $\mathfrak{h}$ coincides with w. Let g_w be a non-zero element of U_{σ_w}. Then $\lambda g_w \in G$ for some non-zero $\lambda \in \mathbb{C}$ (Lemma 4.5). Furthermore, $g_w \rho(x) g_w^{-1} = \rho(\sigma_w(x))$ for all $x \in \mathfrak{g}$.

Now we describe how the above considerations yield an algorithm to compute generators of $W_{\mathfrak{c}}$. Consider $a \in \mathcal{C}_{\mathfrak{c}}$, $K_a = \pi^{-1}(a) = \{w \in W^\theta \mid \pi(w) = a\}$, and $w \in K_a$. Let $g_w \in \widetilde{G}$ be as above. Let $\lambda \in \mathbb{C}$ be such that $\tilde{g}_w = \lambda g_w$ lies in G. Then $\tilde{g}_w Z_G(\mathfrak{h})$ is the set of elements of G that map to $w \in W$. We have $a \in W_{\mathfrak{c}}$ if and only if $\tilde{g}_w Z_G(\mathfrak{h}) \cap G_0$ is not empty for at least one $w \in K_a$. But $\tilde{g}_w Z_G(\mathfrak{h}) \cap G_0$ is not empty if and only if the polynomial equations that correspond to $\tilde{\theta} g_w g_0 = g_w g_0 \tilde{\theta}$ have a solution over $\mathbb{C}$. Again by Gröbner basis calculations we can check whether theses equations have a solution. Performing this for all $w \in K_a$, we can check whether a given $a \in \mathcal{C}_{\mathfrak{c}}$ lies in $W_{\mathfrak{c}}$. We leave the precise formulation of the algorithm, along the lines of Algorithm 3, to the reader.

Remark 4.6. When $\mathfrak{g}$ is of type D_{2n} then we have to act in a slightly different way, because the corresponding simply connected group Spin_{4n} has no faithful irreducible representations (cf. [25], §2.10). In this case one can take ρ to be the direct sum of two irreducible representations (for example, with highest weights λ_1 and λ_{2n}, where $\lambda_1, \ldots, \lambda_{2n}$ are the fundamental weights). Let V_1, V_2 denote the corresponding $\mathfrak{g}$-modules, and $V = V_1 \oplus V_2$. Let R_1 (respectively R_2) be the set of scalar matrices that are the identity on V_2 (respectively, V_1). Then $\widetilde{G} = G R_1 R_2$. Also, the spaces U_σ are 2-dimensional. So it is no longer true that a basis element of U_σ necessarily lies in $\widetilde{G}$. However, by taking different linear combinations of the two basis elements one easily finds an element of U_σ that lies in $\widetilde{G}$. With this change, one can proceed as for the other types.

4.2.3. *Remarks on the algorithm.*

Remark 4.7. In the last step of Algorithm 3, of course, one may restrict to a that are complex reflections. Also, once we find a set of $a \in \mathcal{C}_{\mathfrak{c}}$ that do lift to G_0, and generate a subgroup of $\mathcal{C}_{\mathfrak{c}}$ of prime index, and at the same time we have found a $b \in \mathcal{C}_{\mathfrak{c}}$ that does not lift to G_0, then we can stop.

Remark 4.8. We remark that for the constructions described in this section it is necessary to compute the root system of $\mathfrak{g}$ with respect to $\mathfrak{h}$. However, this Cartan subalgebra may not be split over $\mathbb{Q}$. In fact, that happens quite often. In these cases we have to work over algebraic extensions of $\mathbb{Q}$. Fortunately, Magma can handle these well. However, if the degree of the extension increases, then the computations tend to become much more difficult.

Remark 4.9. The automorphism θ is said to be N-regular if $\mathfrak{g}_1$ contains a regular nilpotent element. It is known that, up to conjugacy, there is exactly one N-regular inner automorphism of any given order ([16]). Also, Panyushev proved ([16]) that for N-regular θ we have that $W_{\mathfrak{c}}$ coincides with $\mathcal{C}_{\mathfrak{c}} = \pi(W^\theta)$. Finally, we remark that for the exceptional types the N-regular automorphisms have been determined in [6].

Remark 4.10. We note that also Algorithm 3 works in the same way when $\mathfrak{g}$ is defined over a field of characteristic $p > 2$.

Remark 4.11. Algorithm 3 runs also if $\mathrm{Aut}(\mathfrak{g})^\circ$ is not simply connected. Moreover, it may very well give the correct output. Indeed, that happens, for example, if the centraliser of θ in $\mathrm{Aut}(\mathfrak{g})^\circ$ is connected, or if $W_\mathfrak{c} = \mathcal{C}_\mathfrak{c}$. In fact, it gave the correct output on all examples that we tried.

Remark 4.12. We can compute the degrees of the invariants that generate $\mathbb{C}[\mathfrak{c}]^{W_\mathfrak{c}}$ directly. Indeed, this last ring is isomorphic to $\mathbb{C}[\mathfrak{g}_1]^{G_0}$ ([24]). The space of homogeneous elements of degree k of $\mathbb{C}[\mathfrak{g}_1]^{G_0}$ is isomorphic, as G_0-module, to $\mathrm{Sym}^k(\mathfrak{g}_1)$. Now in Magma there are algorithms implemented for computing the decomposition of this last module into irreducibles, given such a decomposition of $\mathfrak{g}_1$. Write $\mathfrak{g}_0 = \mathfrak{s} \oplus \mathfrak{t}$, where $\mathfrak{s}$ is semisimple and $\mathfrak{t}$ a central torus. Then there is an invariant of degree k if and only if there is a 1-dimensional submodule of $\mathrm{Sym}^k(\mathfrak{g}_1)$, on which $\mathfrak{t}$ acts trivially.

Moreover, the product of the degrees of the generating invariants is equal to the order of $W_\mathfrak{c}$ ([13], Theorem 4.19). So if this product equals the order of $\mathcal{C}_\mathfrak{c}$, then we can conclude $\mathcal{C}_\mathfrak{c} = W_\mathfrak{c}$.

This approach works well if the degrees of the generating invariants are relatively small. For higher degrees it becomes difficult to execute, due to the complexity of the algorithm for finding the decomposition of the module $\mathrm{Sym}^k(\mathfrak{g}_1)$.

5. Implementation and practical experiences

We have implemented the algorithms described in the previous section in the language of the computer algebra system Magma. This section has three subsections. The first describes some details of this implementation. The second reports on practical experiences with the implementation. In the third we briefly give an example in type E_6.

5.1. **Implementation.** Here we describe the two main problems that we encountered when implementing the algorithm, and how we deal with them.

Let $\mathfrak{c}$ be a Cartan subspace of $\mathfrak{g}_1$, and let $\mathfrak{h}$ be the Cartan subalgebra computed in Algorithm 2. In Step 3 of that algorithm the adjoint action of $\mathfrak{h}$ on $\mathfrak{g}$ needs to be diagonalised. However, in general, the elements of $\mathrm{ad}_\mathfrak{g}\mathfrak{h}$ are not split over $\mathbb{Q}$, and for that reason we need to work over an extension of $\mathbb{Q}$; the larger the degree of that field, the more difficult all subsequent computations become. Furthermore, the Cartan subspace $\mathfrak{c}$ is found by the randomised Algorithm 1, and different runs produce different Cartan subspaces, which lead to different field extensions. For these reasons we start our program by computing a number (in our implementation it is 10) of Cartan subspaces, and pick the one which leads to the smallest degree field extension of $\mathbb{Q}$.

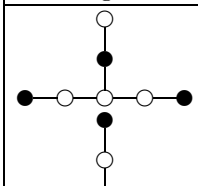
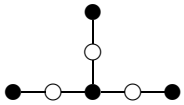
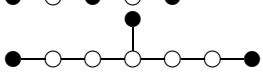
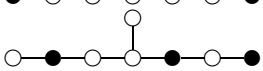
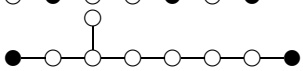
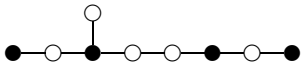
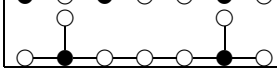
The second problem occurs when we are in the situation considered in Section 4.2.2, i.e., $\mathrm{Aut}(\mathfrak{g})^\circ$ is not simply connected, then one of the main computational problems is computing the lift of an element of $\mathrm{Aut}(\mathfrak{g})^\circ$ to the group $\widetilde{G}$. Let n be the dimension of the representation $\rho : \mathfrak{g} \to \mathfrak{gl}(V)$ and let $\sigma \in \mathrm{Aut}(\mathfrak{g})^\circ$. Then computing a basis of U_σ by linear equations involves equations in n^2 unknowns. For larger n (e.g, $n = 56$, as is the case for $\mathfrak{g}$ of type E_7) this becomes very cumbersome indeed. However, in a special case we can do much better. Let $\tilde{\mathfrak{h}}$ be a Cartan subalgebra of $\mathfrak{g}$ such that $\rho(\tilde{\mathfrak{h}})$ is diagonal, and suppose that $\sigma(h) = h$ for all $h \in \tilde{\mathfrak{h}}$. Then a $\tilde{\sigma} \in U_\sigma$ lies in the subgroup $\widetilde{H}$ of $\widetilde{G}$ with Lie algebra $\rho(\mathfrak{h}) \oplus \mathfrak{t}$, where $\mathfrak{t}$ is an abelian Lie algebra consisting of scalar matrices. In particular, $\tilde{\sigma}$ is a diagonal

matrix. So in this case we can find a basis of U_σ by solving a set of linear equations in n unknowns.

Now we construct ρ such that $\rho(\mathfrak{h})$ is diagonal, where $\mathfrak{h}$ is the Cartan subalgebra constructed in Algorithm 2. This implies that for computing the element g_0 (see Section 4.2.2) we can use the trick just described. However, we cannot do the same to compute $\tilde{\theta}$, as θ stabilises a different Cartan subalgebra $\tilde{\mathfrak{h}}$. We get around this by computing a base change matrix M of V such that, when setting $\tilde{\rho}(x) = M\rho(x)M^{-1}$, we have that $\tilde{\rho}(\tilde{\mathfrak{h}})$ is diagonal. Then we compute a lift $\tilde{\theta}'$ of θ using $\tilde{\rho}$, and solving equations in n unknowns. Finally, we set $\tilde{\theta} = M^{-1}\tilde{\theta}'M$.

Let $w \in W$, and let σ_w and g_w be as in Section 4.2.2. To compute g_w using equations in n unknowns we need one more step. First we compute a Cartan subalgebra $\tilde{\mathfrak{h}}$ of $\mathfrak{g}_{0,w} = \{x \in \mathfrak{g} \mid \sigma_w(x) = x\}$. Since σ_w is an inner automorphism, this is also a Cartan subalgebra of $\mathfrak{g}$. We compute the root system of $\mathfrak{g}$ with respect to $\tilde{\mathfrak{h}}$, and proceed as in the computation of $\tilde{\theta}$.

TABLE 2. Running times (in seconds) of the algorithm for computing generators of the little Weyl group. The first and second columns have, respectively, the Kac diagram and the order of the automorphism θ. The third column displays the time used to compute lifts to the group $\widetilde{G}$. And the fourth column lists the dimension of the representation. (Both of these are not applicable for $\mathfrak{g}$ of type E_8.) The fifth column displays the total time used in Gröbner basis computations. The sixth column has the size of the kernel of the homomorphism $\pi : W^\theta \to \mathcal{C}_\mathfrak{c}$. The seventh column lists the degree of the field extension used in the algorithm. The eight column has the isomorphism type of $W_\mathfrak{c}$. The last column has the total time.

Kac diagram	$\lvert\theta\rvert$	lift	$\dim V$	GB	$\lvert K\rvert$	dF	$W_\mathfrak{c}$	time
	4	85	27	0.03	1	2	$G(4,1,2)$	91
	6	104	27	0.05	1	6	G_5	150
	4	379	56	0.03	8	2	$G(4,1,2)$	436
	6	495	56	12	2	6	G_5	671
	3	$\times$	$\times$	0.0	36	2	$G(6,1,2)$	94
	12	$\times$	$\times$	0.2	1	12	G_{10}	2126
	4	6114	144	0.02	1	2	$G(4,2,4)$	6288

5.2. **Practical experiences.** Table 2 contains the running time of our implementation on various inputs, consisting of a simple Lie algebra $\mathfrak{g}$ together with an inner automorphism θ. These timings were obtained on a 3.16 GHz processor.

As in Table 1, we give the Kac diagram of an automorphism by giving the extended Dynkin diagram, where we make some nodes black. A black node means that the corresponding label is 1, otherwise the label is 0.

We describe the isomorphism type of a complex reflection group using the classification of Shephard and Todd ([21]). This means that $W_{\mathfrak{c}}$ is either given by a number, denoting its position in the list, or by a symbol of the form $G(m,p,n)$, meaning that it is isomorphic to the intransitive complex reflection group of that type. We refer to the book [13] for details about the Shephard-Todd classification.

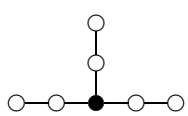
On Table 2 we make the following comments:

- When $\mathrm{Aut}(\mathfrak{g})^\circ$ is not simply connected, most time is spent on computing lifts of automorphisms of $\mathfrak{g}$ to $\widetilde{G}$. The main parameter influencing this is $\dim V$. This can be seen when comparing the running time for the last automorphism with the first four.
- However, with the dimensions of the representations being equal, the degree of the field extension is an important parameter influencing the running time, as using a field of higher degree makes all subsequent computations more time consuming. This comes most dramatically to light when comparing the running times in the two examples for $\mathfrak{g}$ of type E_8.
- Despite their theoretical complexity, the Gröbner basis computations do not bear heavily on the running time. For this reason, also the size of K, the kernel of π, does not greatly influence the running time. (Note that, if the size of K is larger, we potentially need to compute more Gröbner bases.)

Remark 5.1. With the help of these implementations we have computed the little Weyl groups for θ-groups corresponding to automorphisms of the Lie algebras of exceptional type. We have dealt with all automorphisms of rank ≥ 2, and with most automorphisms of rank 1. For the outer automorphisms of rank ≥ 2 of the Lie algebra of type E_6, the little Weyl group was in all cases proved to be equal to $\mathcal{C}_{\mathfrak{c}}$, using the method of Remark 4.12.

In all cases our computation confirmed the isomorphism type of $W_{\mathfrak{c}}$ known in the literature (see [24], [16], [15], [19]).

5.3. **An example.** Here we briefly describe an example of a little Weyl group, computed with our programs. Let $\mathfrak{g}$ be the Lie algebra of type E_6, and θ the automorphism of order 3, with Kac diagram



The invariants of this particular θ-representation have been studied in [2]. Here $\mathfrak{g}_0$ is isomorphic to $\mathfrak{sl}_3(\mathbb{C}) \oplus \mathfrak{sl}_3(\mathbb{C}) \oplus \mathfrak{sl}_3(\mathbb{C})$. As $\mathfrak{g}_0$-module, $\mathfrak{g}_1$ is isomorphic to $U \otimes V \otimes W$, where U, V, W are copies of the standard 3-dimensional $\mathfrak{sl}_3$-module. We denote the basis vectors of these modules by u_i, v_i, w_i $(1 \leq i \leq 3)$ respectively. Setting $y_1 = e_{2,1}$, $y_2 = e_{3,2}$ we have $y_1 u_1 = u_2$, $y_2 u_2 = u_3$, and the same for the v's and w's. A Cartan subspace $\mathfrak{c}$ and little Weyl group $W_{\mathfrak{c}}$ are computed by our

programs in 44 seconds. After identifying $\mathfrak{g}_1$ and $U \otimes V \otimes W$ we have that $\mathfrak{c}$ is spanned by

$$\begin{aligned} A &= u_2 \otimes v_1 \otimes w_1 + u_3 \otimes v_2 \otimes w_2 - u_1 \otimes v_3 \otimes w_3, \\ B &= u_1 \otimes v_2 \otimes w_1 + u_2 \otimes v_3 \otimes w_2 + u_3 \otimes v_1 \otimes w_3, \\ C &= u_1 \otimes v_1 \otimes w_2 - u_2 \otimes v_2 \otimes w_3 + u_3 \otimes v_3 \otimes w_1, \end{aligned}$$

and $W_{\mathfrak{c}}$ is generated by

$$\begin{pmatrix} \omega & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & \omega & 0 \\ 0 & 0 & 1 \end{pmatrix}, \frac{1}{3}\begin{pmatrix} -\omega+1 & \omega+2 & -\omega+1 \\ \omega+2 & -\omega+1 & \omega-1 \\ 2\omega+1 & -2\omega-1 & -\omega+1 \end{pmatrix},$$

where ω denotes a primitive third root of unity.

The fundamental invariants are comuted by Magma in 0.3 seconds. They are

$$\begin{aligned} f_1 &= x_1^6 + 10x_1^3x_2^3 - 10x_1^3x_3^3 + x_2^6 + 10x_2^3x_3^3 + x_3^6, \\ f_2 &= x_1^6x_2^3 + x_1^6x_3^3 + x_1^3x_2^6 - x_1^3x_3^6 - x_2^6x_3^3 - x_2^3x_3^6, \\ f_3 &= x_1^{12} - 58x_1^9x_2^3 + 58x_1^9x_3^3 - 210x_1^6x_2^6 - 336x_1^6x_2^3x_3^3 - 210x_1^6x_3^6 - 58x_1^3x_2^9 + \\ &\quad 336x_1^3x_2^6x_3^3 - 336x_1^3x_2^3x_3^6 + 58x_1^3x_3^9 + x_2^{12} - 58x_2^9x_3^3 - 210x_2^6x_3^6 - 58x_2^3x_3^9 + x_3^{12}. \end{aligned}$$

So two elements $a_1A + b_1B + c_1C$, $a_2A + b_2B + c_2C$ of $\mathfrak{c}$ are in the same G_0 orbit if and only if $f_i(a_1, b_1, c_1) = f_i(a_2, b_2, c_2)$, for $1 \le i \le 3$.

Acknowledgements

The authors are grateful to Oksana Yakimova for enlightening discussions, and for her remarks on a previous version of this paper. The authors also thank the anonymous referees for their careful reading of, and helpful comments on a previous version of this paper.

References

[1] Wieb Bosma, John Cannon, and Catherine Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24** (1997), no. 3-4, 235–265, DOI 10.1006/jsco.1996.0125. Computational algebra and number theory (London, 1993). MR1484478

[2] M. R. Bremner and J. Hu, Fundamental invariants for the action of $\mathrm{SL}_3(\mathbb{C}) \times \mathrm{SL}_3(\mathbb{C}) \times \mathrm{SL}_3(\mathbb{C})$ on $3 \times 3 \times 3$ arrays. *Math. Comp.*, **82** (2013), 2421–2438. MR3073208

[3] Harm Derksen and Gregor Kemper, *Computational invariant theory*, Invariant Theory and Algebraic Transformation Groups, I, Springer-Verlag, Berlin, 2002. Encyclopaedia of Mathematical Sciences, 130. MR1918599 (2003g:13004)

[4] L. Yu. Galitski and D. A. Timashev, *On classification of metabelian Lie algebras*, J. Lie Theory **9** (1999), no. 1, 125–156. MR1680007 (2000f:17015)

[5] Willem A. de Graaf, *Lie algebras: theory and algorithms*, North-Holland Mathematical Library, vol. 56, North-Holland Publishing Co., Amsterdam, 2000. MR1743970 (2001j:17011)

[6] Willem A. de Graaf, *Computing representatives of nilpotent orbits of θ-groups*, J. Symbolic Comput. **46** (2011), no. 4, 438–458, DOI 10.1016/j.jsc.2010.10.015. MR2765379 (2012f:17008)

[7] Sigurdur Helgason, *Differential geometry, Lie groups, and symmetric spaces*, Pure and Applied Mathematics, vol. 80, Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1978. MR514561 (80k:53081)

[8] Derek F. Holt, Bettina Eick, and Eamonn A. O'Brien, *Handbook of computational group theory*, Discrete Mathematics and its Applications (Boca Raton), Chapman & Hall/CRC, Boca Raton, FL, 2005. MR2129747 (2006f:20001)

[9] James E. Humphreys, *Introduction to Lie algebras and representation theory*, Springer-Verlag, New York, 1972. Graduate Texts in Mathematics, Vol. 9. MR0323842 (48 #2197)

[10] Nathan Jacobson, *Lie algebras*, Dover Publications Inc., New York, 1979. Republication of the 1962 original. MR559927 (80k:17001)

[11] V. G. Kac, *Automorphisms of finite order of semisimple Lie algebras* (Russian), Funkcional. Anal. i Priložen. **3** (1969), no. 3, 94–96. MR0251091 (40 #4322)

[12] Victor G. Kac, *Infinite-dimensional Lie algebras*, 3rd ed., Cambridge University Press, Cambridge, 1990. MR1104219 (92k:17038)

[13] Gustav I. Lehrer and Donald E. Taylor, *Unitary reflection groups*, Australian Mathematical Society Lecture Series, vol. 20, Cambridge University Press, Cambridge, 2009. MR2542964 (2010j:20056)

[14] Paul Levy, *Vinberg's θ-groups in positive characteristic and Kostant-Weierstrass slices*, Transform. Groups **14** (2009), no. 2, 417–461, DOI 10.1007/s00031-009-9056-y. MR2504929 (2010g:17022)

[15] Paul Levy. KW-sections for exceptional type Vinberg's θ-groups. `arXiv:0805.2064v2 [math.RA]`, 2010.

[16] Dmitri I. Panyushev, *On invariant theory of θ-groups*, J. Algebra **283** (2005), no. 2, 655–670, DOI 10.1016/j.jalgebra.2004.03.032. MR2111215 (2006g:17022)

[17] D. D. Pervushin, *Invariants and orbits of the standard* $(\mathrm{SL}_4(\mathbf{C}) \times \mathrm{SL}_4(\mathbf{C}) \times \mathrm{SL}_2(\mathbf{C}))$*-module* (Russian, with Russian summary), Izv. Ross. Akad. Nauk Ser. Mat. **64** (2000), no. 5, 133–146, DOI 10.1070/IM2000v064n05ABEH000306; English transl., Izv. Math. **64** (2000), no. 5, 1003–1015. MR1789188 (2002b:20065)

[18] Mark Reeder, *Torsion automorphisms of simple Lie algebras*, Enseign. Math. (2) **56** (2010), no. 1-2, 3–47. MR2674853 (2012b:17040)

[19] Mark Reeder, Paul Levy, Jiu-Kang Yu, and Benedict H. Gross, *Gradings of positive rank on simple Lie algebras*, Transform. Groups **17** (2012), no. 4, 1123–1190, DOI 10.1007/s00031-012-9196-3. MR3000483

[20] Ákos Seress, *Permutation group algorithms*, Cambridge Tracts in Mathematics, vol. 152, Cambridge University Press, Cambridge, 2003. MR1970241 (2004c:20008)

[21] G. C. Shephard and J. A. Todd, *Finite unitary reflection groups*, Canadian J. Math. **6** (1954), 274–304. MR0059914 (15,600b)

[22] Bernd Sturmfels, *Algorithms in invariant theory*, 2nd ed., Texts and Monographs in Symbolic Computation, SpringerWienNewYork, Vienna, 2008. MR2667486 (2011j:13009)

[23] J. Tits, *Sur les constantes de structure et le théorème d'existence des algèbres de Lie semi-simples* (French), Inst. Hautes Études Sci. Publ. Math. **31** (1966), 21–58. MR0214638 (35 #5487)

[24] È. B. Vinberg, *The Weyl group of a graded Lie algebra* (Russian), Izv. Akad. Nauk SSSR Ser. Mat. **40** (1976), no. 3, 488–526, 709. MR0430168 (55 #3175)

[25] È. B. Vinberg, V. V. Gorbatsevich, and A. L. Onishchik, *Structure of Lie groups and Lie algebras* (Russian), Current problems in mathematics. Fundamental directions, Vol. 41 (Russian), Itogi Nauki i Tekhniki, Akad. Nauk SSSR Vsesoyuz. Inst. Nauchn. i Tekhn. Inform., Moscow, 1990, pp. 5–259. MR1056486 (91j:17004)

[26] È. B. Vinberg and A. G. Èlašvili, *A classification of the three-vectors of nine-dimensional space* (Russian), Trudy Sem. Vektor. Tenzor. Anal. **18** (1978), 197–233. MR504529 (80b:15039)

Dipartimento di Matematica, Università di Trento, Italy
E-mail address: `degraaf@science.unitn.it`

Dipartimento di Matematica, Università di Trento, Italy
E-mail address: `francesco.oriente@tin.it`