Constructive recognition of classical groups in even characteristic

Heiko Dietrich, C. R. Leedham-Green, Frank Lübeck, and E. A. O'Brien

ABSTRACT. Let $G = \langle X \rangle \leq \mathrm{GL}(d, \mathbb{F})$ be a classical group in its natural representation defined over a finite field $\mathbb{F}$ of even characteristic. We present Las Vegas algorithms to construct standard generators for G which permit us to write an element of G as a straight-line program in X, and to construct an involution as a straight-line program in X. If $|\mathbb{F}| > 4$, then the algorithms run in time polynomial in the size of the input, subject to the existence of a discrete logarithm oracle for $\mathbb{F}$.

In memory of our friend, Ákos Seress

1. Introduction

Let $C \leq \mathrm{GL}(d, q)$ be a classical group in its natural representation, and let $G = \langle X \rangle$ be any group isomorphic to C. Informally, a *constructive recognition* algorithm for G constructs an isomorphism between G and C and exploits this isomorphism to write an arbitrary element of G as a word in its generators X. For a more formal definition, see [41, p. 192].

We can realise such an algorithm as follows. For each classical group C, we define a specific ordered set of *standard generators* $\mathcal{S}$. The first task is to construct, as words in X, an ordered subset $\mathcal{S}'$ of G that is the image of $\mathcal{S}$ under an isomorphism between C and G. The second task is to solve the *constructive membership problem* for G with respect to $\mathcal{S}'$: namely, express $g \in G$ as a word in $\mathcal{S}'$, and so as a word in X. Now the isomorphism $\varphi\colon G \to C$ that maps $\mathcal{S}'$ to $\mathcal{S}$ is *constructive*: every element in G is first written as a word $g = w(\mathcal{S}')$ in $\mathcal{S}'$, and the image $\varphi(g) = w(\mathcal{S})$ is immediately determined as the corresponding word in $\mathcal{S}$. In a similar way, the inverse of φ is constructive.

In this paper, as an important special case, we suppose that G is given in its natural representation, so G and C are conjugate in $\mathrm{GL}(d, q)$. Since a conjugating element that maps G to C can be found readily, we may assume that $G = C$. The constructive membership problem for C with respect to $\mathcal{S}$ is solved by work of Costi [20]. It remains to construct $\mathcal{S}' \subseteq G$ as a set of words in the given defining generators X; by construction, $\mathcal{S}$ and $\mathcal{S}'$ are conjugate in $\mathrm{GL}(d, q)$.

Leedham-Green & O'Brien [28] developed a Las Vegas algorithm which solves this problem for odd q. Subject to the existence of a discrete logarithm oracle, the algorithm runs in time polynomial in the size of the input. Efficient implementations are publicly available in the computational algebra system MAGMA [6]. The algorithm uses a recursion to classical groups of smaller degree. The first step is to find, by a random search in G, an involution with large ± 1-eigenspaces. The derived group of the centraliser in G of this involution acts on these eigenspaces

Key words and phrases. classical groups, constructive recognition, even characteristic.

We thank Peter Brooksbank and Robert Wilson for helpful discussions; Bill Kantor for comments on a draft; and Cheryl Praeger for making the results of the preprint [40] available to us. We also thank the referee for many helpful suggestions. Dietrich was funded by a University of Auckland Postdoctoral Fellowship. Lübeck acknowledges the generous support of the Alexander von Humboldt Stiftung for a visit by him to the University of Auckland. All authors were partially supported by the Marsden Fund of New Zealand via grant UOA 1015.

as the direct product of classical groups in smaller degrees, and these factors are used for the recursion.

For even q, the situation is more complex. Since the proportion of elements in G of even order is at most $5/q$ (see [22]), it is not practical, for large q, to find an involution by a random search. Another obstacle is that the structure of involution centralisers is more complicated than in odd characteristic, and the two groups for a recursion cannot be found in such a centraliser.

In this paper, we present a constructive recognition algorithm for classical groups in their natural representation defined over finite fields of even characteristic. Subject to the existence of a discrete logarithm oracle, we prove that the algorithm runs in time polynomial in the size of the input (provided that $q > 4$). Our implementation is publicly available in MAGMA.

This work contributes to the *Matrix Group Recognition Project*; its goal is to provide efficient algorithms to investigate matrix groups defined over finite fields. For an overview of this project, see the survey articles [37, 38].

1.1. The groups and their standard copies. The groups of interest are the special linear group, the symplectic group, the special unitary group, and the orthogonal groups, all over a finite field of even characteristic. The definition of all of these groups, except for the first, depends on the choice of a bilinear or quadratic form. However, the groups defined by two different forms of the same type are conjugate in the corresponding general linear group. The *standard copy* of a classical group is its unique conjugate which preserves a chosen *standard form*.

We now describe these groups and their standard forms; we refer to [43] for more details. The form is given as a matrix with respect to some chosen basis. In all cases, d is an integer greater than 1, q is an even prime-power, and V is the underlying row vector space on which the group acts by right multiplication. Let $\mathrm{GL}(d,q)$ be the group of all invertible $d \times d$ matrices over the field $\mathrm{GF}(q)$ with q elements. We denote by $\mathrm{diag}(M_1, \ldots, M_n)$ the block diagonal matrix with blocks $M_1, \ldots, M_n$.

- $\mathrm{SL}(d,q)$: the subgroup of elements of $\mathrm{GL}(d,q)$ with determinant 1.
- $\mathrm{Sp}(d,q)$: the subgroup of elements of $\mathrm{SL}(d,q)$ that preserve a given non-degenerate alternating bilinear form on V. The existence of such a form implies that d is even. The standard copy is the group that preserves the form $F = \mathrm{diag}(\left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right), \ldots, \left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right))$.
- $\mathrm{SU}(d,q)$: the subgroup of elements of $\mathrm{SL}(d,q^2)$ that preserve a given non-degenerate hermitian form on V. The standard hermitian forms for even and odd degree are F and $\mathrm{diag}(F,1)$, respectively, with F as defined for Sp.
- $\Omega^+(d,q)$: the derived subgroup of $\mathrm{O}^+(d,q)$, the subgroup of elements of $\mathrm{SL}(d,q)$ that preserve a given non-degenerate quadratic form on V of $+$ type. This implies that d is even, and we assume $d \geq 4$. The standard quadratic form of $+$ type is $Q = \mathrm{diag}(\left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right), \ldots, \left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right))$, which is preserved by $g \in \mathrm{GL}(d,q)$ if and only if $vgQg^{\mathsf{T}}v^{\mathsf{T}} = vQv^{\mathsf{T}}$ for all $v \in V$. The supported bilinear form is $Q + Q^{\mathsf{T}} = F$.
- $\Omega^-(d,q)$: defined as for $\Omega^+(d,q)$, except that the form is of $-$ type. Again, d is even, and we assume $d \geq 4$. If γ is a fixed primitive element of $\mathrm{GF}(q^2)$, then the standard quadratic form of $-$ type is $\mathrm{diag}(\left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right), \ldots, \left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}1&a\\0&b\end{smallmatrix}\right))$ where $a = \gamma + \gamma^q$ and $b = \gamma^{q+1}$. The supported bilinear form is $\mathrm{diag}(\left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right), \ldots, \left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&a\\a&0\end{smallmatrix}\right))$.

We write $\mathrm{SX}(d,q)$ for a conjugate of one of the above groups; this implicitly means that q is even and $d \geq 4$ if the group is orthogonal. We call SL, SU, Sp, Ω^+, and Ω^- the *type* of the group. A basis of the underlying vector space is *hyperbolic* if $\mathrm{SX}(d,q)$ is the standard copy with respect to it.

For each standard copy, a specific set of *standard generators* is defined in Section 2. This generating set has cardinality at most 7. Costi [20] developed an algorithm to write an arbitrary element in the standard copy as a word in these generators; it is deterministic and runs in time polynomial in the input size, cf. [38, §9.1].

Remark 1.1. We consider only classical groups over finite fields of even characteristic. With the exceptions of $\mathrm{Sp}(2,2)$ and $\mathrm{Sp}(4,2)$, all symplectic groups are simple. With the exception of $\mathrm{SL}(2,2)$, all special linear groups are perfect, and simple if and only if $\gcd(d, q-1) = 1$. With the exception of $\mathrm{SU}(2,2)$, all special unitary groups are perfect, and simple if and only if $\gcd(d, q+1) = 1$. With the exception of $\Omega^+(4,2)$, all groups of type $\Omega^\pm$ are perfect; with the exception of $\Omega^+(4,q)$, all groups of type $\Omega^\pm$ are simple.

1.2. Main results. Let $G = \mathrm{SX}(d,q)$ with q even. We present and analyse a Las Vegas algorithm that takes as input the type of G and a generating set X, and outputs the standard generators of G as words in X. Usually, these generators are defined with respect to a basis different to that for which X was defined, and a matrix relating these bases is also returned. All words are given as *straight-line programs* (SLPs). Intuitively, SLPs are efficiently stored group words in X; for a formal definition and discussion of their significance, see [41, p. 10].

Unless otherwise stated, all complexities are measured in field operations. Let ξ denote an upper bound to the number of field operations needed to construct an independent (nearly) uniformly distributed random element of a subgroup of $\mathrm{SX}(d,q)$. Our algorithms assume the existence of a discrete log oracle: if $G = \Omega^-(4,q) \cong \mathrm{PSL}(2,q^2)$, then the oracle is required for $\mathrm{GF}(q^2)$, otherwise for $\mathrm{GF}(q)$. To simplify statements, we ignore all $\log\log d$ and $\log\log q$ factors; and we use χ to denote an upper bound to the number of field operations equivalent to a call to a discrete logarithm oracle for the appropriate field.

Our main result is the following theorem.

Theorem 1.2. *Let X be a generating set of bounded cardinality for $G = \mathrm{SX}(d,q)$. There is a Las Vegas algorithm that constructs the standard generators for G as* SLPs *in X. If $q > 4$, then the complexity is $O(d((\log^2 q/\log d)\xi + d^3 + d^2 \log d \log^3 q + \log^4 q + \chi \log q))$.*

Guralnick & Lübeck [22] proved that the proportion of elements of even order in $\mathrm{SX}(d,q)$ is at most $5/q$, so a random search for an involution is not feasible for large fields. While the algorithm of Theorem 1.2 can be used to construct an involution, we describe an alternative which is much more efficient.

Theorem 1.3. *Let X be a generating set of bounded cardinality for $G = \mathrm{SX}(d,q)$. There is a Las Vegas algorithm which constructs an involution of G as an* SLP *in X. If $q > 4$, then the complexity is $O(\xi + d^3 \log d \log^3 q + \log^4 q + \chi \log q)$.*

The *corank* of a matrix involution i is the rank of $i - 1$. A modification of the algorithm of Theorem 1.2 yields an algorithm to construct involutions of large corank. While the theoretical complexity is as in Theorem 1.2, this algorithm is more efficient in practice.

Theorem 1.4. *Let X be a generating set of bounded cardinality for $G = \mathrm{SX}(d,q)$. There is a Las Vegas algorithm with the same complexity as in Theorem* 1.2 *that constructs an involution in G with corank r as an SLP in X, where r is as follows. If G is linear or unitary, then $r = \lfloor d/2 \rfloor$. If G has type* Sp *or Ω^+, then $r = 2\lfloor d/4 \rfloor$. If G has type Ω^-, then $\lfloor d/4 \rfloor - 1 \leq r \leq d/2$.*

Remark 1.5. The restriction to $q > 4$ arises from Theorem 5.1, proved by Praeger, Seress & Yalçınkaya [40] under this assumption. However, in practice our algorithms also work with comparable efficiency for $q \in \{2, 4\}$.

1.3. Related work. Kantor & Seress [26] developed *black-box* constructive recognition algorithms (see [41, p. 17]) for classical groups. The complexity of these algorithms involves a factor of q. Using a discrete logarithm oracle and [18, 19], Brooksbank and Kantor [10–13] demonstrate that the complexity of these algorithms can be made polynomial in $\log q$.

Brooksbank [9] devised Las Vegas algorithms to construct standard generators for $\mathrm{Sp}(d,q)$, $\mathrm{SU}(d,q)$, and $\Omega^{\pm}(d,q)$ in their natural representations; subject to the existence of a discrete logarithm oracle, the complexity is $O(d(\xi+d^2\log q(d+\log d\log^3 q+d^2\log q))+\chi\log q)$. The algorithm of Celler & Leedham-Green [17] for $\mathrm{SL}(d,q)$ has complexity $O(d^4q)$. In all cases, the algorithms construct Steinberg generators for the group.

1.4. Other directions. We have generalised our algorithms to classical groups in arbitrary representations [14]. Costi [20] developed an efficient algorithm to write an element of a classical group, given in an arbitrary absolutely irreducible representation in defining characteristic, as an SLP in the standard generators. A black-box algorithm for this task was developed by Ambrose *et al.* [1].

2. Standard generators for classical groups

We now define the standard generators for $G=\mathrm{SX}(d,q)$, where $d=2n$ or $d=2n+1$. Let $\{e_1,f_1,\ldots,e_n,f_n\}$, or $\{e_1,f_1,\ldots,e_n,f_n,w\}$, be a hyperbolic basis $\mathcal{B}$ of the natural G-module V, according as d is even or odd. All matrices of degree d are given with respect to $\mathcal{B}$. A matrix of degree $2k$ is given with respect to $\{e_1,f_1,\ldots,e_k,f_k\}$; it represents an automorphism of V which acts on $\{e_1,f_1,\ldots,e_k,f_k\}$ as the given matrix, and trivially on the remaining $d-2k$ basis elements. Permutation matrices are described by the corresponding permutation. To facilitate uniform exposition, we introduce trivial generators. If the dimension required to define a generator is greater than the dimension of the group, then the generator is assumed to be the identity. For an integer $k\geq 0$ let 1_k be the $k\times k$ identity matrix; if the degree is clear from the context, then we also write $1=1_k$. Analogously, we denote the zero matrix by 0_k or 0.

Definition 2.1. The standard generators of $\mathrm{SX}(d,q)$ are $\mathcal{S}(d,q,\mathrm{SX})=\{s,t,\delta,u,v,x,y\}$ as defined in Table 1, where ω is a specified primitive element of the underlying field $\mathbb{F}$; if the type is SU then $\mathbb{F}=\mathrm{GF}(q^2)$, otherwise $\mathbb{F}=\mathrm{GF}(q)$. For $\Omega^-(d,q)$, we choose a primitive element γ of $\mathrm{GF}(q^2)$ so that $\omega=\gamma^{(q+1)}$.

The group generated by $\mathcal{S}(3,2,\mathrm{SU})$ as given in Table 1 has index 2 in $\mathrm{SU}(3,2)$, so we choose a different element for t.

The generator v is the *cycle* of $\mathrm{SX}(d,q)$; all other standard generators of $\mathrm{SX}(d,q)$ lie in $\mathrm{SX}(4,q)$. This observation is significant since we construct the standard generators by a recursion to classical groups of smaller degree.

Lemma 2.2. *The standard copy of* $\mathrm{SX}(d,q)$ *is generated by* $\mathcal{S}(d,q,\mathrm{SX})$.

PROOF. The standard generators for SL, Sp, Ω^+, and $\mathrm{SU}(2n,q)$ are independent of the characteristic, cf. [28, §3]. For Ω^- and $\mathrm{SU}(2n+1,q)$, we use a slight modification of the generating sets for odd characteristic; the proof is similar. □

Group	s	t	δ	u	v	x	y
$\mathrm{SL}(2n,q)$	(e_1,f_1)	$\left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}\omega&0\\0&\omega^{-1}\end{smallmatrix}\right)$	1_d	$(e_1,\dots,e_n)(f_1,\dots,f_n)$	(e_1,f_1,e_2,f_2)	1_d
$\mathrm{SL}(2n+1,q)$	(e_1,f_1)	$\left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}\omega&0\\0&\omega^{-1}\end{smallmatrix}\right)$	1_d	$(e_1,\dots,e_n)(f_1,\dots,f_n,w)$	(e_1,f_1,e_2,f_2)	1_d
$\mathrm{Sp}(2n,q)$	(e_1,f_1)	$\left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}\omega&0\\0&\omega^{-1}\end{smallmatrix}\right)$	$(e_1,e_2)(f_1,f_2)$	$(e_1,\dots,e_n)(f_1,\dots,f_n)$	$\left(\begin{smallmatrix}1&0&0&0\\0&1&1&0\\0&0&1&0\\1&0&0&1\end{smallmatrix}\right)$	1_d
$\mathrm{SU}(2n,q)$	(e_1,f_1)	$\left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}\omega&0\\0&\omega^{-1}\end{smallmatrix}\right)^{q+1}$	$(e_1,e_2)(f_1,f_2)$	$(e_1,\dots,e_n)(f_1,\dots,f_n)$	$\left(\begin{smallmatrix}1&0&1&0\\0&1&0&0\\0&0&1&0\\0&1&0&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}\omega&0&0&0\\0&\omega^{-q}&0&0\\0&0&\omega^{-1}&0\\0&0&0&\omega^{q}\end{smallmatrix}\right)$
$\mathrm{SU}(2n+1,q)$	(e_1,f_1)	$\left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right)$ $(d,q)\neq(3,2)$	$\left(\begin{smallmatrix}\omega&0\\0&\omega^{-1}\end{smallmatrix}\right)^{q+1}$	$(e_1,e_2)(f_1,f_2)$	$(e_1,\dots,e_n)(f_1,\dots,f_n)$	$\left(\begin{smallmatrix}1_{d-3}&&\\&1&\eta&1\\&0&1&0\\&0&1&1\end{smallmatrix}\right)$ $\eta=\mathrm{Tr}(\omega,\mathrm{GF}(q))^{-1}\omega$	$\left(\begin{smallmatrix}1_{d-3}&&&\\&\omega&0&0\\&0&\omega^{-1}&0\\&0&0&\omega^{q-1}\end{smallmatrix}\right)$
$\Omega^+(2n,q)$	$(e_1,f_2)(e_2,f_1)$	$\left(\begin{smallmatrix}1&0&0&1\\0&1&0&0\\0&1&1&0\\0&0&0&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}\omega&0&0&0\\0&\omega^{-1}&0&0\\0&0&\omega&0\\0&0&0&\omega^{-1}\end{smallmatrix}\right)$	$(e_1,e_2)(f_1,f_2)$	$(e_1,\dots,e_n)(f_1,\dots,f_n)$	$\left(\begin{smallmatrix}1&0&1&0\\0&1&0&0\\0&0&1&0\\0&1&0&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}\omega&0&0&0\\0&\omega^{-1}&0&0\\0&0&\omega^{-1}&0\\0&0&0&\omega\end{smallmatrix}\right)$
$\Omega^-(2n,q)$	$\left(\begin{smallmatrix}1_{d-4}&&&&\\&0&1&0&0\\&1&0&0&0\\&0&0&1&0\\&0&0&\eta&1\end{smallmatrix}\right)$ $\gamma\in\mathrm{GF}(q^2)$ primitive $\omega=\gamma^{q+1}$ $\eta=\gamma+\gamma^q$	$\left(\begin{smallmatrix}1_{d-4}&&&&\\&1&1&1&0\\&0&1&0&0\\&0&0&1&0\\&0&\eta&0&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1_{d-4}&&&&\\&\omega&0&0&0\\&0&\omega^{-1}&0&0\\&0&0&1&a\\&0&0&b&c\end{smallmatrix}\right)$ $a=\gamma^{-1}+\gamma^{-q}$ $b=\gamma+\gamma^q$ $c=\gamma^{-q+1}+\gamma^{q-1}+1$	$(e_1,e_2)(f_1,f_2)$ if $n>2$	$(e_1,\dots,e_{n-1})(f_1,\dots,f_{n-1})$	1_d	1_d

TABLE 1. Standard generators for classical groups in even characteristic

3. General approach and structure of this paper

We outline our strategy to construct the standard generators $\mathcal{S}$ for $G = \mathrm{SX}(d,q) = \langle X \rangle$. If d is "small", then G is a *base case* and we use specialised algorithms to solve the problem. These define a single algorithm, `BaseCase`, described in Section 10. Here and later a "$\star$" within a matrix denotes a submatrix that is not further specified, and whose dimensions are implied by the context.

Definition 3.1. $\mathrm{SX}(d,q)$ is a base case if either $d \leq 6$, or it is one of the following individual groups: $\mathrm{SL}(8,2)$, $\mathrm{SU}(7,2)$, $\mathrm{SU}(9,2)$, $\Omega^-(8,4)$, $\Omega^-(10,4)$, or $\mathrm{Sp}(d,2)$ with $d \in \{8,10,12\}$, or $\Omega^\pm(d,2)$ with $d \in \{4,6,8,10,12,14\}$.

If G is not a base case, then we proceed as follows. The first step is to find a naturally embedded subgroup $H \cong \mathrm{SX}(m,q)$ of G where m lies in a prescribed range, for example, $m \in [d/3, 2d/3]$. If G has type SL or SU, then m is even and $\mathrm{SX}(m,q)$ has the same type as G; otherwise $\mathrm{SX}(m,q)$ has type Ω^+ and m is a multiple of 4. We describe `FirstSX`, the algorithm to construct H, in Section 5. Via a base change, we may assume that

$$H = \begin{pmatrix} \mathrm{SX}(m,q) & 0 \\ 0 & 1_{d-m} \end{pmatrix} \leq G.$$

By recursion, we construct a base change matrix $b = \mathrm{diag}(\star, 1_{d-m})$, the standard generators $\mathcal{S}_H$ of $\mathrm{SX}(m,q)$ in H^b, and a certain involution $i_H \in H^b$ of corank $m/2$; all elements are described by SLPs in X. For simplicity, let $b = 1_d$ in the following. In the centraliser $C_G(i_H)$ we find

$$K = \begin{pmatrix} 1_m & 0 \\ 0 & \mathrm{SX}(d-m,q) \end{pmatrix} \leq G$$

where $\mathrm{SX}(d-m,q)$ has the same type as G. We describe `SecondSX`, the algorithm to construct K, in Section 8. By another recursion, we construct a base change matrix $c = \mathrm{diag}(1_m, \star)$ and the standard generators $\mathcal{S}_K$ of $\mathrm{SX}(d-m,q)$ in K^c. Again, let $c = 1_d$ for simplicity.

With the exception of the cycle v of G, all standard generators of G lie in $\mathcal{S}_H \cup \mathcal{S}_K$. The missing generator is constructed as follows. First, the elements of $\mathcal{S}_H \cup \mathcal{S}_K$ are used to write down a specific involution $i \in G$. Second, in $C_G(i)$ a certain subgroup T of degree 4 (degree 8 if G is symplectic or orthogonal) is constructed. Finally, a *glue element* g is found in T: if $v_K \in \mathcal{S}_K$ and $v_H \in \mathcal{S}_H$ are the cycles in K and H, respectively, then $v = v_K g v_H$ is the cycle of G. To perform this task, we introduce the algorithm `GlueCycles` in Section 9.

We now summarise the main algorithm, `StandardGenerators`, and discuss it in detail in Section 11.

1) If G is a base case, then apply `BaseCase`, otherwise:
2) Construct the subgroup H with `FirstSX`.
3) Recursively apply `StandardGenerators` to H.
4) Construct the subgroup K with `SecondSX`.
5) Recursively apply `StandardGenerators` to K.
6) Find the glue element and combine recursive solutions with `GlueCycles`.

The main difference between this algorithm and that for odd characteristic [28] is the construction of the two subgroups for the recursion. In odd characteristic, these subgroups can be constructed simultaneously in the centraliser of an involution, and this involution is found by a random search. In even characteristic, we construct H using a different technique, and then construct K as a subgroup of the centraliser in G of an involution of large corank in H.

In Section 12 we present two algorithms to construct involutions. The algorithm to construct an involution of large corank is similar to `StandardGenerators`, but avoids the gluing of the cycles. Our algorithm to construct an involution of small corank uses recursion to construct $H \cong \mathrm{SX}(m, q)$ for some small m, usually $m \leq 6$, as a naturally embedded subgroup of G. We then apply specialised algorithms to construct an involution in H.

4. Algorithmic preliminaries

If f and g are real valued functions, defined on all sufficiently large integers, then $f = O(g)$ means $|f(n)| < c|g(n)|$ for some positive constant c and all sufficiently large n.

A *Monte Carlo* algorithm is a randomised algorithm that always terminates, but may return a wrong answer with probability less than any specified value. A *Las Vegas* algorithm is a randomised algorithm that never returns an incorrect answer, but may report failure with probability less than any specified value.

Babai [3] presented a Monte Carlo algorithm to construct, in polynomial time, independent nearly uniformly distributed random elements of a finite group. An alternative is the *product replacement algorithm* of Celler *et al.* [16], which runs in polynomial time by a result of [39]. For a discussion of both algorithms we refer to [41, pp. 26–30].

Our algorithms usually search for elements of G having a specified property. If $1/k$ is a lower bound for the proportion of such elements in G, then we can readily prescribe the probability of failure of the corresponding algorithm. Namely, to find such an element by random search with a probability of failure less than a given $\epsilon \in (0, 1)$ it suffices to choose (with replacement) a sample of uniformly distributed random elements in G of size at least $\lceil -\log_e(\epsilon)k \rceil$. We do not include such factors as part of each theorem.

Often it is necessary to investigate the order of $g \in \mathrm{GL}(d, q)$, which, due to problems with integer factorisation, cannot be determined in polynomial time. We can, however, determine its *pseudo-order*, a good multiplicative upper bound for $|g|$, and the exact power of any specified prime that divides $|g|$, using a Las Vegas algorithm with complexity $O(d^3 \log d + d^2 \log d \log q)$. A Las Vegas algorithm with the same complexity allows us to compute large powers g^n where $0 \leq n < q^d$. Multiplication and division operations for polynomials of degree d over $\mathrm{GF}(q)$ can be performed deterministically with complexity $O(d \log d)$. Using a Las Vegas algorithm, such a polynomial can be factored into its irreducible factors with complexity $O(d^2 \log d \log q)$. The characteristic and minimum polynomials of $g \in \mathrm{GL}(d, q)$ can be computed by a Las Vegas algorithm with complexity $O(d^3 \log d)$. We refer to [28, §2 & 10] for more details and references.

If a matrix group acts absolutely irreducibly on its natural module, then the form it preserves (up to scalar multiples) can be determined with complexity $O(d^3)$, see [23, §7.5.4]. Conjugating $\mathrm{SX}(d, q)$ to its standard copy amounts to finding a hyperbolic basis with respect to the given form; this can be done with complexity $O(d^3 + d^2 \log^2 q)$, see [33, Theorem 1.1].

The following theorem, proved in [5, Corollary 4.2], implies that two random non-scalar $g, h \in \mathrm{SX}(d, q)$ satisfy $g^{|h|} \neq 1$ with probability at least $1/2d$. Recall that an element is *p-regular* if its order is not divisible by p.

Theorem 4.1. *Let G be a finite simple classical group acting naturally on a projective space of dimension $d-1$, and let p be a prime. The proportion of p-regular elements in G is at least $1/2d$.*

We also use the following result on random generation proved in [25].

Lemma 4.2. *Let $G = \mathrm{SX}(d, q)$ be perfect. An $O(1)$ random search in G yields a bounded generating set X for G such that $\{x^2 \mid x \in X\}$ generates G.*

5. Constructing the first subgroup

In this section, we assume that $G = \mathrm{SX}(d, q) = \langle X \rangle$ is not a base case. The standard generators of G are constructed via a recursion to two smaller subgroups of classical type. Modulo a base change with matrix b, these subgroups are

$$H = \begin{pmatrix} \mathrm{SX}(m,q) & 0 \\ 0 & 1_{d-m} \end{pmatrix} \leq G^b \quad \text{and} \quad K = \begin{pmatrix} 1_m & 0 \\ 0 & \mathrm{SX}(d-m,q) \end{pmatrix} \leq G^b,$$

where $\mathrm{SX}(d-m, q)$ has the same type as G. If G is linear or unitary, then so is $\mathrm{SX}(m, q)$ and m is even; otherwise, $\mathrm{SX}(m, q)$ has type Ω^+ and m is divisible by 4. In each case, m is usually required to lie between $d/3$ and $2d/3$.

The construction of K is considered in Section 8. Here we describe the algorithm `FirstSX` used to construct H. We first outline the steps and then discuss them in the subsequent subsections. In the remainder of this section, unless explicitly stated, the characteristic p of the underlying field $\mathbb{F}$ can be either *even* or *odd*. Recall that if $g \in G$ has order prime to q, then g is semisimple.

(1) Find a semisimple $g \in G$ with 1-eigenspace E_g of dimension $d - l \in [2d/3, 5d/6]$ (with some variation for small d) such that g acts irreducibly on a complement to E_g in V, the natural G-module. If G is orthogonal or symplectic, then require that l is even.

(2) Construct a random conjugate g^h, for $h \in G$, such that the intersection E of the 1-eigenspaces of g and g^h has smallest possible dimension (that is, $d - 2l$) and the images of $g - 1$ and $g^h - 1$ span a complement I to E. Then $H = \langle g, g^h \rangle$ leaves E and I invariant, and (in the non-linear case) E and I are non-degenerate subspaces.

Note that the dimension of I is $m = 2l$; if G is orthogonal or symplectic, then m is divisible by 4. Let $\tilde{H} = H|_I$ be the group generated by the restrictions of g and g^h to I. Lemmas 5.9 and 5.10 show the following: $\tilde{H} \leq \mathrm{SX}(m, q)$ (acting on I); if G is orthogonal, then $\mathrm{SX}(m, q)$ is orthogonal (of possibly different type); if $G = \mathrm{Sp}(d, q)$ with q even, then $\mathrm{SX}(m, q)$ is orthogonal; otherwise $\mathrm{SX}(m, q)$ has the same type as G.

In Section 5.1 we describe the construction of the elements g in Step (1). Observe that $g|_I \in \mathrm{SX}(m, q)$ has a 1-eigenspace of dimension $l = m/2$ and acts irreducibly on a complement in I. Our construction ensures that the order of $g|_I$ is divisible by a certain Zsigmondy prime (see Definition 5.3).

As we establish in Lemma 5.9, the conjugacy class of $g|_I$ in $\mathrm{SX}(m, q)$ is determined by its eigenvalues in its action on I; therefore $g|_I$ and $g^h|_I$ are conjugate in $\mathrm{SX}(m, q)$. Thus $g^h|_I$ is random among all conjugates $(g|_I)^c$ of $g|_I$, for $c \in \mathrm{SX}(m, q)$, such that the 1-eigenspaces of $(g|_I)^c$ and $g|_I$ intersect trivially. In this situation, we can apply the following result of Praeger, Seress & Yalçınkaya [40].

Theorem 5.1. *Let $q > 4$. There is an absolute constant $\kappa > 0$ such that the following holds: if $\tilde{H} \leq \mathrm{SX}(m, q)$ is as defined above, then $\tilde{H} = \mathrm{SX}(m, q)$ with probability at least κ.*

Our investigations suggest that this theorem also holds for $q \leq 4$.

If G is orthogonal or symplectic with q even, then $\tilde{H} = \mathrm{SX}(m, q)$ is orthogonal, see Lemma 5.10. If $\tilde{H}$ has $+$ type, then we can readily construct the standard generators of G from the standard generators constructed for H and for the second subgroup $K \leq G$. To simplify exposition, we consider only the case that $\tilde{H}$ is of $+$ type. Theoretically, this is justified by the following third step of `FirstSX`; it does not change the complexity of our main algorithm.

(3) If q is even and $H \cong \mathrm{SX}(m, q)$ is orthogonal of $-$ type, then we constructively recognise $\Omega^-(m, q)$, and so obtain $\Omega^+(m-4, q)$ as a naturally embedded subgroup of H (and G); we

return $\Omega^+(m-4,q)$. If $m \leq 8$, then we apply the algorithm of [9] to find $\Omega^+(4,q)$ as a naturally embedded subgroup.

If q is even and $\tilde{H}$ is orthogonal, then our investigations suggest that the probability that $\tilde{H}$ is of $+$ type is approximately $1/2$ (or $1/3$ if $q=2$). In practice, we realise Step (3) by repeatedly constructing subgroups H until $\tilde{H}$ is of $+$ type.

5.1. Finding elements with large 1-eigenspace. We prove that $O(1)$ random elements in G suffice, in Step (1), to find one that powers up to a desired element. For $g \in G$, denote by E_g the 1-eigenspace of g, and by I_g the image of $g-1$. Recall that a (q,l)-Zsigmondy prime r is one that divides q^l-1 but no q^j-1 for $j<l$. If so, then q has order l modulo r, so $r \geq l+1$. Zsigmondy primes exist, except for $(q,l)=(2,6)$ and $(q,2)$ with q a Mersenne prime, see [35].

Let $\bar{\mathbb{F}}$ be an algebraic closure of the underlying field $\mathbb{F}$ of G, and define the map

$$\Phi\colon \bar{\mathbb{F}} \to \bar{\mathbb{F}}, \quad a \mapsto a^{\varepsilon q},$$

where $\varepsilon=-1$ in case SU and $\varepsilon=1$ in all other cases. The multiset of eigenvalues of $t \in G$ in $\bar{\mathbb{F}}$ is invariant under Φ, since Φ preserves the characteristic polynomial of t. Let $\lambda(t) \vdash d$ be the partition of d describing the cycle lengths of Φ acting on this multiset. If G is of type SU, and l is even, then $l'':=l$ and $l':=l/2$; if G is of type SU, and l is odd, then $l'':=2l$ and $l':=l$; in all other cases $l''=l':=l$.

Definition 5.2. For $l \in \{2,\dots,d/2-1\}$

$$P_l(G)=\{g \in G \mid g \text{ is semisimple, } g \text{ acts irreducibly on } I_g, \text{ and } \dim(E_g)=d-l\}.$$

Definition 5.3. Let $\tilde{P}_l(G)$ be the set of all $t \in G$ with the following properties: l appears exactly once in $\lambda(t)$; l' does not divide any other entry of $\lambda(t)$; and there is a (q,l'')-Zsigmondy prime r dividing $|t|$.

Lemma 5.4. *Elements of $\tilde{P}_l(G)$ power up to elements of $P_l(G)$.*

PROOF. If $a \in \bar{\mathbb{F}}$ is an eigenvalue of $t \in G$ corresponding to a cycle length e in $\lambda(t)$, then the order of a divides $(\varepsilon q)^e-1$. It is easy to see that a (q,l'')-Zsigmondy prime does not divide $|a|$ if $l' \nmid e$. Let $t \in \tilde{P}_l(G)$ with (q,l'')-Zsigmondy prime r dividing $|t|$. Let $e_1,\dots,e_k$ be the entries of $\lambda(t)$ not equal to l, and let $b=|((\epsilon q)^{e_1}-1)\cdots((\epsilon q)^{e_k}-1)|$. Then t^b has $d-l$ eigenvalues equal to 1, and l eigenvalues of order divisible by r. To construct a semisimple element with the same properties, we power t^b by p^j, where p^j is the largest power of p that divides $|t|$; in particular, $j=0$ if t has pairwise different eigenvalues. The non-trivial eigenvalues of this power g of t lie in a field extension of $\mathbb{F}$ of degree precisely l'', so g acts irreducibly on I_g. □

Lemma 5.5. a) *If l is odd and G has type* Sp *or $\Omega^\pm$, then $\tilde{P}_l(G)$ is empty. In all other cases the following holds: For every constant $\alpha \in (0,1/2)$ there exists a constant $c>0$ such that for every $d>1$, every prime power q, and every integer $l \in [\alpha d, d/2)$ for which (q,l'')-Zsigmondy primes exist, the proportion $|\tilde{P}_l(G)|/|G|$ is greater than c/l.*

b) *There exists a constant $c'>0$ such that, for every $d>5$ and every prime power q, if $G=\mathrm{SX}(d,q)$ and $P=\bigcup_l \tilde{P}_l(G)$, where l runs over all integers in $[d/6,d/3]$, then $|P|/|G|>c'$.*

PROOF. a) The proof is based on [31]. First, $\tilde{P}_l(G)$ is obviously closed under conjugation, and it contains an element of G if and only if it contains its semisimple part. Therefore the proportion $|\tilde{P}_l(G)|/|G|$ can be determined as in [31, Lemma 2.3]. This reduces the estimate to considering the proportion of elements in $\tilde{P}_l(G)$ in maximal tori of G, and to counting elements in the Weyl

group of G corresponding to tori with many elements in $\tilde{P}_l(G)$. The conjugacy classes of maximal tori in G are parametrised either by partitions of d in cases SL and SU, or by signed partitions of $\lfloor d/2 \rfloor$. A maximal torus is in all cases a subgroup of a direct product of cyclic groups of order $q^j - 1$ or $q^j + 1$, where j corresponds to an entry in the partition. For a detailed description see [31, §3].

Consider first the types SL and SU. For $2 \leq l < d/2$, we consider partitions with one entry equal to l, and all other entries not divisible by l'. From the description of the structure of the corresponding maximal tori, it is clear that these contain elements of $\tilde{P}_l(G)$, and the proportion of such elements is at least $1 - 1/r \geq 2/3$ for every (q, l'')-Zsigmondy prime r. Let $W \cong S_d$ be the Weyl group, the symmetric group of degree d. For the proportion of elements in W whose cycle type is one of these partitions, a lower bound is given by [31, Lemma 4.2 a), b)]. Using this, and the estimates $l \geq \alpha d$ and $d^{1/d} < 3/2$, part a) follows for SL and SU.

The remaining types are dealt with similarly. The explicit description of maximal tori shows that cycles induced by Φ on the eigenvalues of $t \in G$ come either in pairs or they have even length. This establishes the last statement of a), and we now assume that l is even. Here we consider maximal tori corresponding to elements of W with a negative cycle of length $l/2$ such that $l/2$ does not divide any other cycle length. It follows, from the description of the structure of these tori, that they contain elements in $\tilde{P}_l(G)$, and the proportion of such elements is at least $2/3$. The estimate for the proportion of elements in the Weyl group that are considered is reduced to the case $S_{\lfloor d/2 \rfloor}$ using [31, Lemma 4.2 c), d)].

b) This follows from a): for large d there are at least $d/12 - 1$ different l to consider and for $d > 5$ there is always at least one appropriate l, and every element of G can lie in at most 5 different $\tilde{P}_l(G)$: there can be at most 5 cycles of different lengths at least $d/6$. □

Let P and c be as defined in Lemma 5.5 b). For small rank, say $d < 60$, one can easily compute quite accurate values for the constant c, using [31, Lemma 2.3]. If G has type SL or SU, then for $d \leq 20$ the proportion of elements in P (if not empty) lies in $[0.18, 0.4]$, and for larger d in $[0.4, 0.5]$. For the other types, the proportion is about half as large, which is as expected from the proof above.

Remark 5.6. If $t \in G$, where G has type other than SU, then the cycle lengths induced by Φ on the multiset of eigenvalues of t are the degrees of the irreducible factors over $\mathbb{F}$ of the characteristic polynomial of t. Now let G have type SU. If f is the minimum polynomial over $\mathbb{F}$ of some $a \in \bar{\mathbb{F}}^\times$, then we denote by $\tilde{f}$ the minimum polynomial of a^{-q}. To compute $\tilde{f}$ from f: raise the coefficients to the q-th power, reverse coefficients, and normalise. If Φ induces a cycle of odd length l on the eigenvalues of t, then the characteristic polynomial of t contains an irreducible factor $f = \tilde{f}$ of degree l over $\mathbb{F}$ whose zeroes are the elements of the cycle. If Φ induces a cycle of even length l, its elements are the zeroes of two irreducible factors f and $\tilde{f} \neq f$ of degree $l/2$ over $\mathbb{F}$. Therefore, in all cases, elements in P are easy to detect by computing and factorising characteristic polynomials.

Remark 5.7. If d is small, then P may be empty. We usually solve this problem by extending the range for l in the definition of P to $[2, d/2)$. For $d \in \{3, 4\}$ we search for elements g that have one eigenvalue with multiplicity 1, and another with multiplicity $d - 1$. We take the derived subgroup of $\langle g, g^h \rangle$ as H. Such elements g are easy to find. For example, in case SL, if $t \in G$ satisfies $\lambda(t) = (1, d - 1)$, then $g = t^b$ with $b = (q^{d-1} - 1)/(q - 1)$ is a desired element, with probability $1 - 1/q$.

We summarise the resulting algorithm `FindElement`: it takes as input the generating set X of G and returns $g \in \tilde{P}_l(G)$ for some $l \in [d/6, d/3]$ (with some variation for small d).

(i) Construct random $t \in G$, factorise its characteristic polynomial, and compute $\lambda(t)$ as in Remark 5.6, so determining l.

(ii) If $t \in \tilde{P}_l(G)$, then power t up to $g \in P_l(G)$ and return g, otherwise go back to (i).

Lemma 5.5 shows that it suffices to select $O(1)$ random elements t in order to construct g. This, and the results cited in Section 4, shows the algorithm is Las Vegas and has complexity $O(\xi + d^3 \log d + d^2 \log d \log q)$.

5.2. Constructing the subgroup H. We now show that it is easy to find a conjugate of $g \in \tilde{P}_l(G)$ in sufficiently general position. Recall that V is the natural G-module, $E_g = \ker(g-1)$, and $I_g = \operatorname{im}(g-1)$.

Lemma 5.8. *Let $g \in P_l(G)$ and let T be the set of $h \in G$ such that $\dim(E_g \cap E_g h) = d - 2l$ (the smallest possible value) and $V = (E_g \cap E_g h) \oplus I_g \oplus I_g h$. There exists an absolute constant $c > 0$, independent of l and the type of G, such that $|T|/|G| > c$.*

PROOF. We give a detailed proof only for G of type SL; the proportion $|T|/|G|$ is larger if G is not of this type.

Since g is semisimple, $V = E_g \oplus I_g$. We choose an ordered basis of V such that the first $d-l$ vectors generate E_g, and the last l vectors generate I_g. We estimate the cardinality of T by counting images of these basis vectors under a suitable linear transformation $h \in T$. We start by mapping the first l basis vectors such that their images, together with E_g, span the whole space. This ensures that $E_g \cap E_g h$ has minimal dimension.

For $1 \le j \le l$, we map the j-th basis vector to a vector outside the span of the union of E_g with the set of previously chosen images; there are $q^d - q^{d-l+j-1}$ possible choices. Then we map the remaining $d-2l$ basis vectors of E_g to arbitrary vectors outside the span of the already chosen images. If $l+1 \le j \le d-l$, then there are $q^d - q^{j-1}$ choices for the j-th basis vector. Finally, we map the basis vectors of I_g so that their images span a complement to $(E_g \cap E_g h) \oplus I_g$. Thus the image of the j-th basis vector for $d-l+1 \le j \le d$ must be outside the span of the previously chosen $j-1$ images, and outside the span of the union of $(E_g \cap E_g h) \oplus I_g$ with the set of images of the basis vectors indexed by $d-l+1$ to $j-1$. These two subspaces of dimension $j-1 > d/2$ have an intersection of dimension at least $2j-2-d$. This yields at least $q^d - 2q^{j-1} + q^{2j-2-d}$ possible images (the last one divided by $q-1$ to get an element with determinant 1).

Comparing with $|G| = (\prod_{j=1}^{d}(q^d - q^{j-1}))/(q-1)$ we get a lower bound

$$|T|/|G| \ge \prod_{j=1}^{l} \frac{q^d - q^{d-l+j-1}}{q^d - q^{j-1}} \cdot \prod_{j=d-l+1}^{d} \frac{q^d - 2q^{j-1} + q^{2j-2-d}}{q^d - q^{j-1}}.$$

For the first factor of the product, observe that

$$\prod_{j=1}^{l} \left(1 - \frac{q^{d-l+j-1} - q^{j-1}}{q^d - q^{j-1}}\right) > \prod_{j=1}^{l} \left(1 - \frac{q^d}{q^d - q^{j-1}} \cdot \left(\frac{1}{q}\right)^{l-j+1}\right) > \prod_{k=1}^{l} \left(1 - \frac{4}{3} \cdot \left(\frac{1}{q}\right)^k\right).$$

For fixed q, this last expression converges to some positive constant as $l \to \infty$ (because the geometric series $\sum_j 1/q^j$ converges). For the second factor, we find a positive lower bound with a similar estimate; the critical term is the last, but it is easily checked that it is at least $1/2$. □

Evaluations of the formulae show that c, for $q = 2$ or 4, is bounded below by 0.08 and 0.47 respectively. Our investigations suggest that for $q = 2$ the proportion $|T|/|G|$ is about 0.25.

Lemma 5.9. *Let $g \in P_l(G)$, and let $h \in T$ as in Lemma 5.8.*

a) *$E = E_g \cap E_g h$ and $I = I_g \oplus I_g h$ are invariant under $\langle g, g^h \rangle$.*

b) *If G preserves some form, then E and I are mutually orthogonal, and non-degenerate spaces.*

c) *If G is orthogonal, then E and I are non-degenerate quadratic spaces, possibly of type different to that of G.*

d) *If G is not an orthogonal group in even dimension, then the conjugacy class of a semisimple element in G is determined by its eigenvalues in $\bar{\mathbb{F}}$ (with multiplicities). In the remaining case this is true if and only if the element has an eigenvalue 1, otherwise there are two semisimple classes whose elements have the same eigenvalues.*

e) *The restrictions $g|_I$ and $g^h|_I$ are conjugate within the group preserving the form specified in b) and c).*

PROOF. a) Clearly, $E_g \cap E_g h$ and $I_g \oplus I_g h$ are fixed by each of g and g^h; for example, if $v \in I_g$, then $vg^h = v + v(g^h - 1) \in I_g \oplus I_g h$.

b) Suppose G preserves a form $\beta(\cdot, \cdot)$. Let $v = w(g-1) \in I_g$, and $w' \in E_g$. Then $\beta(v, w') = \beta(wg, w') - \beta(w, w') = \beta(w, w'g^{-1}) - \beta(w, w') = 0$, thus I_g and E_g are orthogonal. Hence $E_g \cap E_g h$ is orthogonal to $I_g \oplus I_g h$. Since V is the direct sum of these spaces, $I_g \oplus I_g h$ is the orthogonal complement of $E_g \cap E_g h$. Thus the form restricted to each of $E_g \cap E_g h$ and $I_g \oplus I_g h$ is non-degenerate.

c) Let Q be the quadratic form preserved by G with associated bilinear form $F = Q + Q^\intercal$. By part a), the natural G-module decomposes into $V = E \perp I$, and with respect to a suitable basis, the matrix F has block diagonal form; we may also assume that Q is an upper triangular block matrix. Note that $vkQk^\intercal v^\intercal = vQv^\intercal$ for every $v \in V$ and $k \in \langle g, g^h \rangle$. Since the 1-eigenspaces of $g|_I$ and $g^h|_I$ intersect trivially, Q is a block diagonal matrix.

d) The semisimple conjugacy classes of G are parametrised by orbits of the Weyl group on elements of a maximal torus, see [15, Propositions 3.7.2 & 3.7.3]; in characteristic 2 the centralisers of semisimple elements are connected, but the proof of [15, Proposition 3.7.3] remains correct even if the group is not of simply-connected type. An explicit description of maximal tori in the natural representation and the Weyl group action is given in [31, Section 3]; our claim follows easily from that description.

e) Since both restrictions have 1 as an eigenvalue, the result follows from d). □

Let $g \in P_l(G)$, and $m = 2l$. By Lemma 5.8 and Theorem 5.1, the construction of $O(1)$ random elements is sufficient to find $h \in G$ such that $H = \langle g, g^h \rangle$ is isomorphic to $\mathrm{SX}(m, q)$. We can verify the latter using the one-sided Monte Carlo recognition algorithm of [35]; this has complexity $O(\xi + d^3 \log d \log^3 q)$.

We now suppose that $G = \mathrm{Sp}(d, q)$ and q is even. The next lemma shows that in this case H preserves a quadratic form, hence $H \cong \mathrm{SX}(m, q)$ is orthogonal by Theorem 5.1. Recall that g acts irreducibly on the orthogonal complement I_g of its 1-eigenspace E_g, and $I_g \cap I_g h = \{0\}$. Since g is semisimple, it has odd order.

Lemma 5.10. *There is a quadratic form on $I_g \oplus I_g h$ preserved by g and g^h.*

PROOF. Write $U = I_g$. Let β be a non-degenerate bilinear form left invariant by $\mathrm{Sp}(d, q)$; so β is unique up to multiplication by a non-zero scalar. The space U, together with the restriction $\gamma = \beta|_{U \times U}$, is a symplectic space, and every semisimple element of $\mathrm{Sp}(U)$ lies in a maximal torus of an orthogonal group on U. Thus, there exists a g-invariant quadratic form B_1 on U which supports γ: namely, B_1 is a g-invariant quadratic form with $B_1(u+v) = B_1(u) + B_1(v) + \gamma(u, v)$

for all $u, v \in U$. Conjugating B_1 by h defines a g^h-invariant quadratic form B_2 on Uh. Now define a quadratic form B on $U \oplus Uh$ by $B(v_1 + v_2) = B_1(v_1) + B_2(v_2) + \beta(v_1, v_2)$ for $v_1 \in U$ and $v_2 \in Uh$. We prove that this form is invariant under the action of g^2; since g has odd order, this shows that B is g-invariant. Since U is g-invariant, it suffices to prove that $B(vg^2) = B(v)$ for all $v \in Uh$. For $v \in V$ define $f(v) = vg - v$. Since g centralises V/U it follows that f takes values in U, and hence, by restriction, defines a linear map from Uh to U. Let $v \in Uh$. It follows from $vg^2 = f(v)(g+1) + v$ and the g-invariance of B_1 that

$$\begin{aligned} B(vg^2) &= B_1(f(v)(g+1)) + B_2(v) + \beta(f(v)(g+1), v) \\ &= \beta(f(v)g, f(v)) + B_2(v) + \beta(f(v)(g+1), v), \end{aligned}$$

so it suffices to prove that $\beta(f(v)g, f(v)) = \beta(f(v)(g+1), v)$. But this follows from

$$\begin{aligned} \beta(v, f(v)) &= \beta(v, vg) = \beta(vg, vg^2) \\ &= \beta(vg, v(g^2 - 1) + v) \\ &= \beta(vg, f(v)(g+1) + v) \\ &= \beta(v + f(v), f(v)(g+1) + v). \end{aligned}$$

Similarly B is preserved by g^h. □

If q is even and H is orthogonal, then we apply Step (3) to ensure that H is of $+$ type. Algorithm `FirstSX` returns $H \cong \mathrm{SX}(m, q)$, a base change matrix reflecting the decomposition $V = I_g \oplus I_g h \oplus (E_g \cap E_g h)$, and m. Observe that $m \in [d/3, 2d/3]$ with variations for small d. Hence, our previous discussion proves the following.

Lemma 5.11. *Algorithm* `FirstSX` *is correct and Las Vegas; if $q > 4$, then it has complexity $O(\xi + d^3 \log d \log^3 q + \log^4 q)$.*

6. Centralisers of involutions

We consider $G = \mathrm{SX}(d, q)$ with $d \geq 4$, and $d \geq 8$ if G is orthogonal. An involution i in G is *good in G* if either G is linear or unitary, or i has even corank and $vFi^{\intercal}v^{\intercal} = 0$ for all v in the natural G-module, where F is the alternating form preserved by G. (Recall that $\Omega^{\pm}(d, q)$ preserves the alternating form supported by the quadratic form.) Aschbacher & Seitz [2] describe the centraliser of an involution in Chevalley groups over fields of even size. (Our good involutions are those of type c_r as defined in [2].) The next theorem follows from [2, (4.2), (4.3), (6.2), (7.6), (7.7), (7.9), (8.5), (8.6), (8.10), (8.12)].

Theorem 6.1. *Let $i \in G = \mathrm{SX}(d, q)$ be a good involution of corank $r \leq d/2$ and let $\mathbb{F}$ be the underlying field of G. There exists a base change matrix $c \in \mathrm{GL}(d, \mathbb{F})$ such that $i^c = \begin{pmatrix} 1_r & 0 & 1_r \\ 0 & 1_{d-2r} & 0 \\ 0 & 0 & 1_r \end{pmatrix}$ and the elements of $C_{G^c}(i^c)$ have upper block triangular form with diagonal blocks a, b, a, of degree r, $d - 2r$, and r, respectively. Consider the homomorphism*

$$\psi \colon C_{G^c}(i^c) \to \mathrm{GL}(r, \mathbb{F}) \times \mathrm{GL}(d - 2r, \mathbb{F}), \quad \begin{pmatrix} a & \star & \star \\ 0 & b & \star \\ 0 & 0 & a \end{pmatrix} \mapsto (a, b).$$

If G is linear, unitary, or symplectic, then the image of ψ contains $\mathrm{SX}(r, q) \times \mathrm{SX}(d - 2r, q)$ with both factors of the same type as G. Otherwise, in the orthogonal case, the image contains $\mathrm{Sp}(r, q)' \times \mathrm{SX}(d - 2r, q)$, where $\mathrm{SX}(d - 2r, q)$ has the same type as G.

We call $i^* := i^c$ the *standard form* of i. The centraliser of a good involution $i \in G$ in standard form has the structure given in Theorem 6.1. The following easy observation is used in Sections 11 and 12.

Lemma 6.2. *Let* $G = \mathrm{SX}(d,q)$ *and*

$$H = \begin{pmatrix} \mathrm{SX}(m,q) & 0 \\ 0 & 1_{d-m} \end{pmatrix} \leq G \quad \textit{and} \quad K = \begin{pmatrix} 1_m & 0 \\ 0 & \mathrm{SX}(d-m,q) \end{pmatrix} \leq G.$$

If $i_H \in \mathrm{SX}(m,q)$ *and* $i_K \in \mathrm{SX}(d-m,q)$ *are good involutions, then* $\mathrm{diag}(i_H, i_K)$ *is a good involution in* G*.*

The centraliser $C_G(i)$ of an involution $i \in G$ can be constructed using an algorithm of Bray [7]. If g is an arbitrary element of G, then $[i,g]$ either has odd order $2k+1$, in which case $g[i,g]^k$ commutes with i, or has even order $2k$, in which case both $[i,g]^k$ and $[i,g^{-1}]^k$ commute with i. If g is random among the elements of G for which $[i,g]$ has odd order, then $g[i,g]^k$ is random in $C_G(i)$, see [36, Theorem 11]. Such an element $g[i,g]^k$ is a *Bray generator* of $C_G(i)$. Bray & Wilson [8] prove the following.

Theorem 6.3. *Let* $G = \mathrm{SX}(d,q)$ *and let* $i \in G$ *be an involution. There is a constant* $c > 0$ *such that the proportion of* $g \in G$ *with* $[i,g]$ *of odd order is bounded below by* $c/\log d$*.*

The equivalent theorem for odd characteristic is proved in [36]. Our investigations suggest that the proportion for even characteristic is greater than some absolute positive constant independent of the rank.

Let $i \in G$ be a good involution in standard form. A subgroup C of $C_G(i)$ is *sufficient* if its image $\psi(C)$ under the projection in Theorem 6.1 is the same as $\psi(C_G(i))$.

Theorem 6.4. *Let* i *be a good involution in* $G = \mathrm{SX}(d,q)$ *in standard form. A bounded generating set for a sufficient subgroup of* $C_G(i)$ *can be constructed using a Monte Carlo algorithm with complexity* $O(\log d(\xi + d^3 \log d + d^2 \log d \log q))$*.*

PROOF. Theorem 6.3 shows that it suffices to consider $O(\log d)$ random elements to construct a random element of $C_G(i)$. The results cited in Section 4 imply that the test for each element – to decide if it has even order and to compute a power – requires $O(d^3 \log d + d^2 \log d \log q)$ field operations.

Let K be the image of the projection of $\psi(C_G(i))$ into one of the direct factors of the range of ψ, that is, into $\mathrm{GL}(r,\mathbb{F})$ or $\mathrm{GL}(d-2r,\mathbb{F})$. The probability that two random elements of a cyclic group C generate C is $\prod(1-\frac{1}{p^2}) > \frac{6}{\pi^2}$, where the product is over all primes p dividing $|C|$. Hence we obtain elements whose image in the cyclic quotient $Q := K/K'$ generates Q, so we can construct a generator of Q. By multiplying a random element of $C_G(i)$ by an appropriate power of the preimage of this generator, we obtain random elements of K'. Kantor & Lubotzky [25] prove that a bounded number of random elements generate K'. □

6.1. An involution that is not good. We now consider a certain involution of corank 4 in a group of type Sp and $\Omega^{\pm}$. In contrast to our previous discussion, this involution is *not* good. Its centraliser is, up to conjugacy, described in [2]. However, in Section 9 we need explicit knowledge of the centraliser structure with respect to a particular hyperbolic basis.

Let F_1 be the matrix of a non-degenerate alternating form of rank $d-8$, and let

$$F = \begin{pmatrix} 0_4 & 0 & F_2 \\ 0 & F_1 & 0 \\ F_2 & 0 & 0_4 \end{pmatrix} \quad \text{with} \quad F_2 = \begin{pmatrix} 0&0&0&1 \\ 0&0&1&0 \\ 0&1&0&0 \\ 1&0&0&1 \end{pmatrix}.$$

Then F is the matrix of a non-degenerate alternating form. Let Q_1 be the matrix of a quadratic form of $+$ or $-$ type supporting F_1, that is, $Q_1 + Q_1^{\mathsf{T}} = F_1$, and define

$$Q = \begin{pmatrix} Q_2 & 0 & F_2 \\ 0 & Q_1 & 0 \\ 0 & 0 & Q_3 \end{pmatrix}$$

where $Q_2 = \mathrm{diag}(0,1,1,1)$ and $Q_3 = \mathrm{diag}(0,0,0,1)$. Now Q is a matrix of a quadratic form supporting F, of the same type as Q_1. Let $G_1 = \mathrm{Sp}(d,q)$ and $G_2 = \Omega^{\pm}(d,q)$ be the groups preserving F and Q, respectively. In the remainder of this section we determine $C_{G_1}(i)$ and $C_{G_2}(i)$, where

$$i = \begin{pmatrix} 1_4 & 0 & 1_4 \\ 0 & 1_{d-8} & 0 \\ 0 & 0 & 1_4 \end{pmatrix} \tag{6.1}$$

is a non-good involution of corank 4 contained in G_1 and G_2.

Lemma 6.5. *Let $\Delta \leq \mathrm{SL}(4,q)$ be the subgroup of elements*

$$\delta = \begin{pmatrix} 1 & 0 & 0 & 0 \\ u_1 & a_1 & a_2 & 0 \\ u_2 & a_3 & a_4 & 0 \\ v & w_1 & w_2 & 1 \end{pmatrix} \in \mathrm{SL}(4,q) \quad \textit{where} \quad \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} a_2 & a_1 \\ a_4 & a_3 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}.$$

Then

$$C_{G_1}(i) = \left\{ g = \begin{pmatrix} \delta & \star & \star \\ 0 & x & \star \\ 0 & 0 & \delta \end{pmatrix} \mid \delta \in \Delta,\ x \in \mathrm{Sp}(d-8,q) \right\},$$

where $\mathrm{Sp}(d-8,q)$ preserves the form F_1, and the entries $\star$ are subject to the sole constraint that g lies in G_1. (Such entries may be found for every choice of δ and x.) The same holds when G_1 is replaced by G_2; here $x \in \Omega^{\pm}(d-8,q)$ is required to preserve the associated quadratic form Q_1.

PROOF. The centraliser of i in $\mathrm{GL}(d,q)$ is the set of matrices of the same shape as the matrices g in the lemma, except that Δ is replaced by $\mathrm{GL}(d,4)$, and x may be any element of $\mathrm{GL}(d-8,q)$, and there is no restriction on the entries marked $\star$. Thus we need only consider the condition that a matrix of this shape should lie in G_1 or G_2.

Taking G_1 first, and considering the copy of δ in the bottom right corner, it is easy to see that a necessary condition for g to lie in G_1 is for δ to lie in Δ. Conversely, if $\delta \in \Delta$ and $x \in \mathrm{Sp}(d-8,q)$ then $\mathrm{diag}(\delta, x, \delta) \in G_1$, as is straightforward to check.

Now consider $C_{G_2}(i)$ and let $\delta \in \Delta$ and $x \in \Omega^{\pm}(d-8,q)$ as in the statement of the lemma. Then

$$\begin{pmatrix} \delta & 0 & \delta' \\ 0 & x & 0 \\ 0 & 0 & \delta \end{pmatrix} \in C_{G_2}(i) \quad \text{if} \quad \delta' = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & a_2{+}a_3 & a_1{+}a_4 & 0 \\ 0 & a_1{+}a_4 & a_2{+}a_3 & 0 \\ 0 & w_2 & w_1 & 0 \end{pmatrix}. \qquad \square$$

A routine calculation proves the following.

Lemma 6.6.

$$g = \begin{pmatrix} 1_4 & 0 & y \\ 0 & 1_{d-8} & 0 \\ 0 & 0 & I_4 \end{pmatrix} \in G_1 \quad \textit{if and only if } y \textit{ has the form} \quad \begin{pmatrix} y_{11} & y_{12} & y_{13} & y_{14} \\ y_{21} & y_{22} & y_{23} & y_{13} \\ y_{31} & y_{32} & y_{22} & y_{12} \\ y_{41} & y_{31}{+}y_{12} & y_{21}{+}y_{13} & y_{11}{+}y_{14} \end{pmatrix},$$

and $g \in G_2$ if, in addition, $y_{14} = 0$ and $y_{13}^2 = y_{23}$ and $y_{12}^2 = y_{32}$ and $y_{11}^2 + y_{41} + y_{11} = 0$.

For $j \in \{1,2\}$ let A_j be the subgroup of G_j consisting of matrices g as in Lemma 6.6. Note that the set of elements

$$\begin{pmatrix} \delta & 0 & \delta' \\ 0 & x & 0 \\ 0 & 0 & \delta \end{pmatrix} \in C_{G_j}(i) \quad \text{with} \quad \delta = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & a_1 & a_2 & 0 \\ 0 & a_3 & a_4 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \in \Delta$$

and $\delta' = 0$ if $j = 1$, forms a group, $S_j \leq C_{G_j}(i)$, isomorphic to $\mathrm{SL}(2,q)$. Also S_j acts by conjugation on A_j; or, equivalently, on the corresponding additive group of matrices of the form y, also by conjugation.

Remark 6.7. As S_1-module, A_1 is isomorphic to the direct sum of three copies of $\mathrm{GF}(q)$ with trivial S_1-action, two $\mathrm{GF}(q)$-modules each of dimension 2 with natural $\mathrm{SL}(2,q)$-action, and one copy of $\mathrm{sl}(2,q)$, the group of 2×2 matrices over $\mathrm{GF}(q)$ of trace zero. As S_2-module, A_2 is

isomorphic to the direct sum of one copy of $\mathrm{GF}(q)$, one copy of the natural module, and one copy of $\mathrm{sl}(2,q)$.

It follows from the above analysis that $C_{G_j}(i)$ is a split extension of $O_2(C_{G_j}(i))$ by $\mathrm{SL}(2,q)\times \mathrm{SX}(d-8,q)$. For $j=1$ a stronger statement holds: $C_{G_1}(i)$ is a split extension of a normal 2-subgroup by $\Delta \times \mathrm{Sp}(d-8,q)$.

7. Extracting sections from involution centralisers

Let $G=\mathrm{SX}(d,q)$ and let $i\in G$ be a good involution of corank r in standard form. It follows from Theorem 6.1 that a sufficient subgroup C of $C_G(i)$ has sections $\mathrm{SX}(r,q)$ and $\mathrm{SX}(d-2r,q)$. We now describe how to construct $\mathrm{SX}(d-2r,q)$ as a subgroup of C. For this task, two Monte Carlo algorithms are introduced. The first constructs

$$\hat{B}=\begin{pmatrix} 1_r & \star & \star \\ 0 & \mathrm{SX}(d-2r,q) & \star \\ 0 & 0 & 1_r \end{pmatrix} \leq C,$$

and the second constructs

$$B=\begin{pmatrix} 1_r & 0 & 0 \\ 0 & \mathrm{SX}(d-2r,q) & 0 \\ 0 & 0 & 1_r \end{pmatrix} \leq \hat{B}.$$

7.1. Constructing direct factors. Let $D=\mathrm{SX}(n,q)\times\mathrm{SX}(m,q)$ be described by a bounded generating set X. We want to find generators for $\mathrm{SX}(n,q)$ and $\mathrm{SX}(m,q)$ as SLPs in X. This problem is considered in [28, §11] for odd characteristic; the same strategy works for even q. The general approach is the following: repeatedly construct random $(g_1,h_1),(g_2,h_2)\in D$ with $g_1,g_2\in\mathrm{SX}(n,q)$ and $h_1,h_2\in\mathrm{SX}(m,q)$ until these power up to $(g_1',1)$ and $(g_2',1)$ with orders divisible by certain Zsigmondy primes. For sufficiently large degree n, in general $n\geq 10$, a result of [35] is applied to estimate the probability that g_1' and g_2' generate $\mathrm{SX}(n,q)$. It is proved in [28, Lemma 11.5] that this algorithm to construct $\mathrm{SX}(n,q)$ is Monte Carlo with complexity

$$O(\tfrac{d}{\log d}(\xi\log^2 q + d^3\log d + d^2\log d\log q)) \tag{7.1}$$

where $d=n+m$.

For $n\leq 9$, which includes the *non-generic cases* of [35], we follow the approach of Babai & Beals [4]. The first step is to find a random $(g,h)\in D$ with non-scalar g. If $|g|$ has a prime divisor coprime to $q-1$, then $(g',1)=(g,h)^{|h|}$ is non-scalar with probability $1/2m$, see Theorem 4.1. If $\mathrm{SX}(n,q)$ is quasisimple, then it can be constructed as the normal closure of $\langle(g',1)\rangle$ in $\mathrm{SX}(d,q)$, which essentially amounts to constructing the normal closure of $\langle g'\rangle$ in $\mathrm{SX}(n,q)$. Since n is bounded, the normal closure algorithm described in [41, Theorem 2.3.9] has complexity $O(\xi\log^2 q)$, see [42]. The generating set returned by this algorithm has length $O(\log q)$; Lemma 4.2 is used to find a bounded one. The group $\Omega^+(4,q)$ is not quasisimple, but a direct product of two copies of $\mathrm{SL}(2,q)$ if $q>2$, see [43, Corollary 12.39]. Thus we can use a similar normal closure construction if $\mathrm{SX}(n,q)=\Omega^+(4,q)$.

We use these results to design a Monte Carlo algorithm `KillFactor`. Let $G=\mathrm{SX}(d,q)$ and let $i\in G$ be a good involution of corank r in standard form. Let Y be a bounded generating set for a sufficient subgroup of $C_G(i)$, and let *pos* be either "middle" or "top". The input to `KillFactor` is Y and *pos*. If *pos* is "middle" then `KillFactor` returns a bounded generating set for

$$\hat{A}=\left\{\begin{pmatrix} a & \star & \star \\ 0 & 1_{d-2r} & \star \\ 0 & 0 & a \end{pmatrix} \mid a\in\mathrm{SX}(r,q)\right\}\leq C_G(i);$$

otherwise it returns one for

$$\hat{B}=\left\{\begin{pmatrix} 1_r & \star & \star \\ 0 & b & \star \\ 0 & 0 & 1_r \end{pmatrix} \mid b\in\mathrm{SX}(d-2r,q)\right\}\leq C_G(i).$$

In our application of the theoretical results cited above, it suffices to determine the pseudo-order of a diagonal block only, which can be computed in polynomial time [28, §2.2]. Thus `KillFactor` has complexity stated in Equation (7.1).

7.2. Extracting the middle section. We now describe the algorithm `ExtractMiddleBlock` which constructs

$$B = \begin{pmatrix} 1_r & 0 & 0 \\ 0 & \mathrm{SX}(d-2r,q) & 0 \\ 0 & 0 & 1_r \end{pmatrix} \leq C.$$

Variations of the following lemma have been employed by Conway, Parker, Kleidman and Wilson; see [29, §4.10].

Lemma 7.1. *Let $R = Q \ltimes M$ where M has exponent 2. Let $f \in Q$ have odd order and assume it acts fixed-point freely on M. If $r = qm \in R$ where $q \in C_Q(f)$ and $m \in M$, then $q = fr(ff^r)^{(|f|-1)/2}$.*

PROOF. Write $o = |f|$ and note that $f^r = fm^f m$. A straightforward computation shows that $fr(ff^r)^{(o-1)/2} = qf^o m^{f^{o-1}} m^{f^{o-2}} \dots m^f m$. Since $f-1$ is invertible, the lemma follows from $0 = (f^o - 1)(f-1)^{-1} = f^{o-1} + f^{o-2} + \dots + f + 1$. □

It suffices to use the pseudo-order of f.

Algorithm 1: `ExtractMiddleBlock`(Y, f)

/* Y is a bounded generating set for a sufficient subgroup C of $C_G(i)$, where $i \in G = \mathrm{SX}(d,q)$ is a good involution of corank r in standard form; assume the middle block of C is not $\Omega^+(4,2)$ and $f = \begin{pmatrix} c & 0 & * \\ 0 & 1 & 0 \\ 0 & 0 & c \end{pmatrix} \in C$ with c fixed-point free of odd order. Return a bounded generating set for

$$B = \begin{pmatrix} 1_r & 0 & 0 \\ 0 & \mathrm{SX}(d-2r,q) & 0 \\ 0 & 0 & 1_r \end{pmatrix} \leq C$$

where $\mathrm{SX}(d-2r,q)$ has the same type as G. */

1 **begin**
2 $\hat{B}$:= `KillFactor`(Y, "top");
3 let $\varphi\colon \hat{B} \to \mathrm{SX}(d-2r,q)$ be the projection onto the middle diagonal block;
4 by a random search in $\hat{B}$ find a bounded subset *gen* such that $\operatorname{im}\varphi = \langle \{\varphi(x^2) \mid x \in \mathit{gen}\} \rangle$;
5 Return $\{(hg(hh^g)^{(|h|-1)/2})^2 \mid g \in \mathit{gen}\}$ where $h := f^2$;
6 **end**

Lemma 7.2. *Algorithm* `ExtractMiddleBlock` *is correct, Monte Carlo, and has complexity stated in Equation* (7.1).

PROOF. An $O(1)$ random search in $\hat{B}$ is sufficient to find the subset *gen* in Line 4, see Lemma 4.2. The element h in Line 5 has odd order. It follows from Lemma 7.1, and can also be verified directly, that if g in *gen* has diagonal blocks $1, v, 1$, then

$$\left(hg(hh^g)^{(|h|-1)/2}\right)^2 = \begin{pmatrix} 1 & 0 & * \\ 0 & v & 0 \\ 0 & 0 & 1 \end{pmatrix}^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & v^2 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \qquad \square$$

Where `ExtractMiddleBlock` is applied, the element f is constructed simultaneously with the involution i.

8. Constructing the second subgroup

Recall that usually the standard generators of $G = \mathrm{SX}(d,q)$ are constructed via a recursion to two smaller subgroups of classical type. Modulo a base change with matrix b, these subgroups are

$$H = \begin{pmatrix} \mathrm{SX}(m,q) & 0 \\ 0 & 1_{d-m} \end{pmatrix} \leq G^b \quad \text{and} \quad K = \begin{pmatrix} 1_m & 0 \\ 0 & \mathrm{SX}(d-m,q) \end{pmatrix} \leq G^b,$$

where $\mathrm{SX}(d-m,q)$ has the same type as G. If G is linear or unitary, then so is $\mathrm{SX}(m,q)$ and m is even. Otherwise, $\mathrm{SX}(m,q)$ has type Ω^+ and m is divisible by 4. In both cases, m usually lies between $d/3$ and $2d/3$. In Section 5 we described the construction of H. We now describe the construction of K. Again, X is a bounded generating set for G, and we assume that G is not a base case.

Let H, b, and m be the output of `FirstSX` and we assume that, via a base change, $H \leq G^b$ is the standard copy. We also assume that, by recursion, we have found the standard generators of $\mathrm{SX}(m,q)$ as elements in H and, in addition, a good involution $i \in H$ of corank $m/2$. Algorithm `SecondSX` accepts H, b, and i as input and returns

$$K = \begin{pmatrix} 1_m & 0 \\ 0 & \mathrm{SX}(d-m,q) \end{pmatrix} \leq G^b,$$

where $\mathrm{SX}(d-m,q)$ has the same type as G.

The first step is to construct a base change matrix $c = \mathrm{diag}(\star, 1_{d-m})$ such that $cic^{-1} = \mathrm{diag}(\left(\begin{smallmatrix} 1_r & 1_r \\ 0 & 1_r \end{smallmatrix}\right), 1_{d-m})$. Using Theorem 6.4, we find a bounded generating set Y of a sufficient subgroup of the centraliser of cic^{-1} in $G^{bc^{-1}}$. Now let $u = \mathrm{diag}(\left(\begin{smallmatrix} f & \star \\ 0 & f \end{smallmatrix}\right), 1_{d-m}) \in H^{c^{-1}}$ with f fixed-point free of odd order; u can be found by an $O(1)$ random search in the centraliser of cic^{-1} in cHc^{-1}, see [34]. Let $w = \mathrm{diag}(1_r, \left(\begin{smallmatrix} 0 & 1_r \\ 1_{d-m} & 0 \end{smallmatrix}\right))$ be a base change matrix. Now $\hat{K}$ is constructed as the output of `ExtractMiddleBlock`(wYw^{-1}, wu^2w^{-1}). Note that

$$wcic^{-1}w^{-1} = \begin{pmatrix} 1_r & 0 & 1_r \\ 0 & 1_{d-m} & 0 \\ 0 & 0 & 1_r \end{pmatrix} \quad \text{and} \quad wu^2w^{-1} = \begin{pmatrix} f^2 & 0 & \star \\ 0 & 1_{d-m} & 0 \\ 0 & 0 & f^2 \end{pmatrix}.$$

We verify that $\hat{K} \cong \mathrm{SX}(d-m,q)$ using the one-sided Monte Carlo algorithm of [35], and return $K = \hat{K}^w$.

Lemma 8.1. *Algorithm* `SecondSX` *is correct, Las Vegas, and has complexity*

$$O((d \log^2 q / \log d)\xi + d^4 + d^3 \log d \log^3 q).$$

PROOF. The complexity follows from that stated in Equation (7.1) for `ExtractMiddleBlock`, Theorem 6.4, and [35]. □

9. Gluing the cycles

Let $G = \mathrm{SX}(d,q)$ be a non-base case, so $d > 6$. Using the algorithms of the previous section, modulo a base change, we have constructed

$$H = \begin{pmatrix} \mathrm{SX}(m,q) & 0 \\ 0 & 1_{d-m} \end{pmatrix} \leq G \quad \text{and} \quad K = \begin{pmatrix} 1_m & 0 \\ 0 & \mathrm{SX}(d-m,q) \end{pmatrix} \leq G,$$

where $m = 2r$ is even and $d - m > 2$. Via a recursion and another base change, the standard generators $\mathcal{S}_H$ and $\mathcal{S}_K$ of $\mathrm{SX}(m,q)$ and $\mathrm{SX}(d-m,q)$ are obtained in H and K, respectively. We assume that $\mathrm{SX}(m,q)$, $\mathrm{SX}(d-m,q)$, and G are standard copies. Write d as $2n$ or $2n+1$, and let $\{e_1, f_1, \ldots, e_n, f_n\}$, or $\{e_1, f_1, \ldots, e_n, f_n, w\}$, be the corresponding hyperbolic basis $\mathcal{B}$ of the G-module V.

All standard generators of G, except the cycle v, are in $\mathcal{S}_H \cup \mathcal{S}_K$. If v_H and v_K are the cycles of $\mathrm{SX}(m,q)$ and $\mathrm{SX}(d-m,q)$ in H and K, respectively, then $v = v_K g v_H$, where $g = (e_r, e_{r+1})(f_r, f_{r+1}) \in G$ is the glue element. We now describe `FindGlueElement`, the algorithm that constructs g. We find g in the centraliser of a specific involution $i \in G$ constructed from the elements in $\mathcal{S}_H$ and $\mathcal{S}_K$. We first provide more details for the different types of G, and then describe the algorithm.

9.1. The cases SL **and** SU**.** The groups $\mathrm{SX}(m,q)$ and $\mathrm{SX}(d-m,q)$ have the same type as G, and all standard generators of G, except the cycle v, are contained in $\mathcal{S}_H$. First, let $d-m > 3$. The elements of $\mathcal{S}_H$ and $\mathcal{S}_K$ are used to construct $i = (e_r, f_r)(e_{r+1}, f_{r+1})$ and $f = \mathrm{diag}(x_1, 1_4, x_2)$, where x_1 and x_2 have degrees $m-2$ and $d-m-2$ respectively, and both are fixed-point free of odd order. We use i and f as input to `FindGlueElement`. If $d-m = 3$, which only occurs for odd $d \leq 9$, then $x_2 = 1$ is not fixed-point free, but a similar construction can be used to find the glue element. The base case $d = 6$ and $m = 2$ can be processed in the same way.

9.2. The cases Sp **and** Ω^+**.** The degree m is divisible by 4, and $d - m = 2$ if and only if $d = 6$, which is a base case. Hence, $d - m \geq 4$. Via a base change, we swap $\mathrm{SX}(m,q)$ and $\mathrm{SX}(d-m,q)$: namely, we assume that $\mathrm{SX}(m,q)$ with $m \geq 4$ has the same type as G, and $\mathrm{SX}(d-m,q)$ has type Ω^+ with $d-m$ divisible by 4. The elements of $\mathcal{S}_H$ and $\mathcal{S}_K$ are used to construct $i = (e_{r-1}, f_{r-1})(e_r, f_r)(e_{r+1}, f_{r+1})(e_{r+2}, f_{r+2})$ and $f = \mathrm{diag}(x_1, 1_8, x_2)$, where x_1 and x_2 have degrees $m-4$ and $d-m-4$ respectively, and both are fixed-point free of odd order. We use i and f as input to `FindGlueElement`.

9.3. The case Ω^-**.** The group $\mathrm{SX}(m,q)$ has type Ω^+ with m divisible by 4, and $\mathrm{SX}(d-m,q)$ has type Ω^- with $d-m \geq 4$. With the exception of the cycle v, all standard generators of G are contained in $\mathcal{S}_K$. If $d-m \geq 6$, then we construct i and f as in the case Sp. If $d-m = 4$, then we construct f as in the case Sp, and an involution

$$i = \mathrm{diag}(1_{m-4}, \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 0 & \omega^{q/2} \\ \omega^{q/2-1} & 0 \end{smallmatrix}\right)),$$

where ω is as specified in Definition 2.1.

9.4. Algorithm `FindGlueElement`**.** We use the notation of the previous sections and consider i and f as constructed in Sections 9.1–9.3. Let G be of type Sp or $\Omega^{\pm}$. We choose a new ordered basis for V, namely

$$\{e_{r-1}, e_r, e_{r+1}, e_{r+2}\} \cup \mathcal{B}' \cup \{e_{r-1} + f_{r-1}, e_r + f_r, e_{r+1} + f_{r+1}, e_{r+2} + f_{r+2}\},$$

where $\mathcal{B}'$ is the basis $\mathcal{B}$ with e_s and f_s deleted for $s \in \{r-1, r, r+1, r+2\}$. With respect to this basis, the matrix of f is

$$\begin{pmatrix} 1_4 & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1_4 \end{pmatrix}$$

with c of odd order acting fixed-point freely on its underlying space of dimension $d-8$; the matrix of i is given in Equation (6.1), and its centraliser is described in Section 6.1.

By [2, (7.7), (8.5), (8.12)], two non-good involutions in $\mathrm{SX}(d,q)$ of the same even corank are conjugate in $\mathrm{SX}(d,q)$. Thus, there exists a base change matrix $b \in C_{\mathrm{SL}(d,q)}(i)$ such that $(C_G(i))^b = C_{G^b}(i)$ is the centraliser described in detail in Section 6.1. Note that $\mathrm{diag}(\delta, 1_{d-8}, \delta) \in C_G(i)$ for δ a 4×4 matrix if and only if $\delta\delta^{\mathsf{T}} = 1_4$.

Lemma 9.1. *The subset of* $\mathrm{SL}(4,q)$ *consisting of matrices of the form*

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & a_2{+}1 & a_1{+}1 & a_1{+}a_2{+}1 \\ 0 & a_4{+}1 & a_3{+}1 & a_3{+}a_4{+}1 \\ 0 & a_2{+}a_4{+}1 & a_1{+}a_3{+}1 & a_1{+}a_2{+}a_3{+}a_4{+}1 \end{pmatrix}$$

is a subgroup S isomorphic to $\mathrm{SL}(2,q)$, and $T = \{\mathrm{diag}(s, 1_{d-8}, s) \mid s \in S\}$ is a subgroup of $\Omega^{\pm}(d,q) \leq \mathrm{Sp}(d,q)$ when referred to the above basis for V.

PROOF. It is routine to check that these matrices form a group isomorphic to $\mathrm{SL}(2,q)$, the element of S displayed in the statement of the lemma mapping to the matrix $\left(\begin{smallmatrix} a_1 & a_2 \\ a_3 & a_4 \end{smallmatrix}\right)$, and that the bilinear form defining $\mathrm{Sp}(d,q)$ is preserved. One readily checks that the quadratic form defining $\Omega^{\pm}(d,q)$ is preserved by T. □

The group T of the lemma contains the required glue element $g = (e_r, e_{r+1})(f_r, f_{r+1})$. We now describe how to construct a generating set for T, and thus find g.

Let $G = \mathrm{Sp}(d,q)$ and define $\Delta = \{\delta \in \mathrm{SL}(4,q) \mid \mathrm{diag}(\delta, 1_{d-8}, \delta) \in C_G(i)\}$; note that Δ is conjugate to the subgroup Δ in Section 6.1. We outline the construction of elements of

$$A_1 = \left\{ \begin{pmatrix} \delta & \star & \star \\ 0 & 1 & \star \\ 0 & 0 & \delta \end{pmatrix} \mid \delta \in \Delta \right\} \leq C_G(i) \quad \text{and} \quad A_2 = \left\{ \begin{pmatrix} \delta & 0 & \star \\ 0 & 1 & 0 \\ 0 & 0 & \delta \end{pmatrix} \mid \delta \in \Delta \right\} \leq A_1,$$

so that we can find T as a subgroup of A_2. First, `KillFactor` is used to obtain elements of A_1. To construct an element of A_2 from $s \in A_1$ we use Lemma 7.1 with $h = f$: namely, s is replaced by $fs(ff^s)^{(|f|-1)/2} \in A_2$. We show that this construction yields elements that are sufficiently random in A_2, as described below.

Note that $A_2 \cong T \ltimes U$ where $T \cong \mathrm{SL}(2,q)$ and $U \trianglelefteq C_G(i)$ is the unipotent kernel of the natural map of A_2 onto S. Lemmas 6.5 and 6.6 show that U has a normal series $U = U_0 > U_1 > \cdots > U_9 = 0$ such that five of the sections U_j/U_{j+1} are isomorphic to $\mathrm{GF}(q)$ and are centralised by T, and four are isomorphic to the natural $\mathrm{SL}(2,q)$-module. Let $tu \in A_2$ with $t \in T \setminus \{1\}$ and $u \in U_j \setminus U_{j+1}$ be obtained as described above: namely, $tu = fs(ff^s)^{(|f|-1)/2}$ where $s = h^k \in A_1$ is constructed in `KillFactor` for some random $h \in C_G(i)$ and some integer k. Suppose U_j/U_{j+1} is isomorphic to the natural $\mathrm{SL}(2,q)$-module. By construction, $s = h^k$ and tu both act as t on U_j/U_{j+1} and, by [34], we can assume that this action is fixed-point free. Since h is random, we could, with equal probability, have chosen $h' = hv$ for some random $v \in U_j \setminus U_{j+1}$; then the element obtained from `KillFactor` would be $s' = (h')^k = sv^{h^{k-1}} v^{h^{k-2}} \ldots v^h v$. If $v^{1+h+\ldots+h^{k-1}} = 0$ in U_j/U_{j+1}, then

$$0 = v^{(1+h+\ldots+h^{k-1})(h-1)} + U_{j+1} = v^{h^k-1} + U_{j+1} = v^{t-1} + U_{j+1} = 0 + U_{j+1}.$$

By our assumption, t acts fixed-point freely; therefore, $1 + h + \ldots + h^{k-1}$ is an automorphism of U_j/U_{j+1}, which proves that $s' = sv'$ where $v' \in U_j$ is such that $v' + U_{j+1}$ is random in U_j/U_{j+1}. Replacing s by s', one deduces that our initial $u \in U_j$ is random in the sense that $u + U_{j+1} \in U_j/U_{j+1}$ is random. We call $u_j = u$ a *helper* in U_j.

We now construct a generating set for T as follows. Let $\mathcal{T}$ be a bounded subset of A_2 mapping onto a generating set for T; using `KillFactor`, the complexity for this task is given in Equation (7.1). Write $h \in \mathcal{T}$ as $h = tu$, where t is the image in T of h, and define M as the normal T-closure of the group generated by the elements u that arise in this way. If $M = \{0\}$, then $\mathcal{T}$ is the generating set we seek. Otherwise, we can easily find j such that $U_j \geq M + U_{j+1} > U_{j+1}$ as defined above. Now the object is to replace $\mathcal{T}$ by a different subset that maps onto a generating set of T, but where the corresponding group M is smaller. Iterating this procedure will terminate in a generating set for T.

Suppose that M is non-trivial. Write $N = U_{j+1}$. If M lies in the kernel of the natural homomorphism of A_2 onto Δ, then we may regard M as lying in an additive group of 4×4 matrices over $\mathrm{GF}(q)$. Otherwise we need only consider the image of M modulo this kernel, and again reduce the problem to linear algebra. There are two cases to consider. Suppose first that T centralises $(M+N)/N$. In this case we replace $\mathcal{T}$ by a bounded set of commutators of elements

of $\mathcal{T}$ that maps onto a generating set for T. This new generating set will give rise to a new M that lies in the old N; in fact, every abelian quotient of M that is centralised by T will be destroyed in this way. Now suppose that $(M+N)/N$ is a copy of the natural $\mathrm{SL}(2,q)$-module. If we have already found a helper $u_j \in U_j$, then every element tu of $\mathcal{T}$ may be pre-multiplied by a product of T-conjugates of u_j that is congruent to u modulo U_{j+1}, and this defines the new generating set $\mathcal{T}$ whose corresponding M lies in the old N. If we have not found a helper $u_j \in U_j$ previously, then we have found one now and we restart the whole construction with a new $\mathcal{T}$; this happens at most 5 times. Once M is trivial, so $\mathcal{T}$ generates T, the required glue involution is obtained using the algorithm of [19].

The same algorithm applies to $G = \Omega^{\pm}(d,q)$; but in this case only three sections of U are isomorphic to the natural S-module. In summary, we have shown how to construct the glue element. We call the resulting Las Vegas algorithm `FindGlueElement`; it takes as input the involution i, the group G, and the element f constructed in Sections 9.1–9.3, and returns the glue element g.

Remark 9.2. If G is linear or unitary, then the involution $i \in G$ is good and has corank 2. If $d - m \neq 3$, then the same approach as above can be used to construct, modulo base change, a subgroup $T = \mathrm{diag}(\mathrm{SL}(2,q), 1_{d-4}, \mathrm{SL}(2,q))$ of the centraliser $C_G(i)$ such that T contains the glue element. If $d-m=3$, that is, $d \in \{7,9\}$, then the element f in Section 9.1 is $\mathrm{diag}(1_2, c, 1_2)$ with $c = \mathrm{diag}(1, c')$ where c' is fixed-point free of odd order. In this case,

$$fy(ff^y)^{(|f|-1)/2} = \left(\begin{smallmatrix} a & \tilde{u} & \star \\ 0 & 1 & \tilde{w} \\ 0 & 0 & a \end{smallmatrix}\right) \quad \text{for all} \quad y = \left(\begin{smallmatrix} a & u & v \\ 0 & 1 & w \\ 0 & 0 & a \end{smallmatrix}\right),$$

and $\tilde{u}$ and $\tilde{w}$ have only one non-zero column and row, respectively. We proceed as before.

Lemma 9.3. *Algorithm* `FindGlueElement` *is Las Vegas and has complexity stated in Equation* (7.1).

PROOF. We only consider the more complicated cases Sp and $\Omega^{\pm}$. The complexity for constructing all helpers in A_2 is determined by `KillFactor`, see Equation (7.1), and Lemma 7.1. The remaining calculations are carried out in 4×4 matrices over $\mathrm{GF}(q)$ and are thus independent of d. We use the algorithm of [19] to find an element in T mapping two given elements of a copy of the natural $\mathrm{SL}(2,q)$-module onto each other. □

9.5. The gluing algorithm. Algorithm `GlueCycles` has input X, H, K, b, m, $\mathcal{S}_H$, $\mathcal{S}_K$ where X generates $G = \mathrm{SX}(d,q)$,

$$H = \left(\begin{smallmatrix} \mathrm{SX}(m,q) & 0 \\ 0 & 1_{d-m} \end{smallmatrix}\right) \leq G^b \quad \text{and} \quad K = \left(\begin{smallmatrix} 1_m & 0 \\ 0 & \mathrm{SX}(d-m,q) \end{smallmatrix}\right) \leq G^b,$$

as described in Section 9. The sets $\mathcal{S}_H$ and $\mathcal{S}_K$ are the standard generators of $\mathrm{SX}(m,q)$ and $\mathrm{SX}(d-m,q)$ in H and K, respectively. The output is the standard generators for G. The algorithm is Las Vegas with complexity as in Equation (7.1).

10. Base cases

For small degree, in general $d \leq 6$, the standard generators of $\mathrm{SX}(d,q)$ cannot be constructed recursively, so we use different methods.

- The Las Vegas algorithms of [19] and [30] are used to construct an arbitrary element of $\mathrm{SL}(d,q)$ with $d \in \{2,3\}$ as an SLP in its defining generators; these algorithms have complexity $O(\xi + \log q + \chi)$.

- The Las Vegas algorithm of [9] is used to construct an arbitrary element of $\mathrm{Sp}(4,q)$ or $\mathrm{SU}(d,q)$ with $d \in \{3,4\}$ as an SLP in its defining generators. This algorithm has complexity $O(\xi + \log^4 q + \chi \log q)$.

More generally, for bounded d, we could use Brooksbank's algorithm [9], while still achieving the complexity of Theorem 1.2. We present alternatives that seem more efficient in practice.

10.1. Special linear and unitary groups. The individual base cases are $\mathrm{SL}(d,2)$ with $d \in \{4,6,8\}$, and $\mathrm{SU}(d,2)$ with $d \in \{5,6,7,9\}$. Groups of degree 6 with $q \geq 4$ are solved recursively using the standard algorithm. We now discuss briefly the outstanding base cases: namely, $\mathrm{SL}(4,q)$ with $q \geq 4$, and $\mathrm{SL}(5,q)$ and $\mathrm{SU}(5,q)$ with $q \geq 2$.

10.1.1. *Degree 4.* Let $G = \mathrm{SL}(4,q)$ with $q \geq 4$. The first step is to construct an involution $i_1 \in G$ of corank 2 in standard form; an algorithm to do this is described in Section 12. In $C_G(i_1)$ we find a second involution $i_2 = \left(\begin{smallmatrix} 1_2 & k \\ 0 & 1_2 \end{smallmatrix}\right)$ with non-scalar $k \in \mathrm{GL}(2,q)$: to do this, we use the algorithm of [19] to construct an element of the form $\left(\begin{smallmatrix} j_2 & k \\ 0 & j_2 \end{smallmatrix}\right)$, where j_2 is an involution, and square this element. Let K be the group generated by sufficient subgroups of $C_G(i_1)$ and $C_G(i_2)$, so K is a parabolic subgroup of $\mathrm{SL}(4,q)$, fixing a 2-dimensional subspace. Using a modification of `KillFactor` and a random search, we obtain

$$\hat{A} = \left(\begin{smallmatrix} \mathrm{SL}(2,q) & \star \\ 0 & 1_2 \end{smallmatrix}\right) \leq K, \quad \hat{B} = \left(\begin{smallmatrix} 1_2 & \star \\ 0 & \mathrm{SL}(2,q) \end{smallmatrix}\right) \leq K, \quad \text{and} \quad f = \left(\begin{smallmatrix} 1_2 & u \\ 0 & f' \end{smallmatrix}\right) \in \hat{B},$$

where f' is fixed-point free, and hence of odd order. Via a base change we arrange $u = 0$. (This requires that f and f' have the same order.) Now Lemma 7.1 is applied to construct

$$A = \left(\begin{smallmatrix} \mathrm{SL}(2,q) & 0 \\ 0 & 1_2 \end{smallmatrix}\right) \leq \hat{A} \quad \text{and} \quad B = \left(\begin{smallmatrix} 1_2 & 0 \\ 0 & \mathrm{SL}(2,q) \end{smallmatrix}\right) \leq \hat{B}.$$

Using [19], all standard generators of G, except v and x, are found in A. It suffices to construct $m = \mathrm{diag}(1, \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right), 1) \in G$, since $x = s'ms$, where $s' = s^v$ is found in B using [19], and $v = x^2$. To find m, we construct

$$i = \mathrm{diag}(\left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)) \in A \times B$$

and a permutation matrix c such that $i^* = cic^{-1}$ is in standard form and $cmc^{-1} = \mathrm{diag}(s, 1_2)$. With the same construction as for A, we start with i^* (instead of i_1) to construct $\mathrm{diag}(\mathrm{SL}(2,q), 1_2)$. This group contains cmc^{-1} and, using [19], we find m.

10.1.2. *Degree 5.* Let $G = \mathrm{SU}(5,q)$ with $q \geq 4$. In summary, we construct subgroups $H = \mathrm{diag}(\mathrm{SU}(4,q), 1)$ and $K = \mathrm{diag}(1_2, \mathrm{SU}(3,q))$ of G^b for some base change matrix b. The lists of standard generators of $\mathrm{SU}(4,q)$ in H and of $\mathrm{SU}(3,q)$ in K include all of the standard generators of G.

In more detail, we use `FirstSX` and [9] to construct H, b, and the standard generators of $\mathrm{SU}(4,q)$ in H. Let C be the centraliser in G^b of the standard generator $t \in H$. We now obtain K as a subgroup of C. Modulo a base change, this amounts to constructing the subgroup $B = \mathrm{diag}(1, \mathrm{SU}(3,q), 1)$ of

$$\hat{B} = \left(\begin{smallmatrix} 1 & \star & \star \\ 0 & \mathrm{SU}(3,q) & \star \\ 0 & 0 & 1 \end{smallmatrix}\right).$$

Via the same base change, the standard generator $\delta \in H$ is $\mathrm{diag}(\omega, 1_3, \omega^{-1})^{q+1}$. Now we can construct $B \leq \hat{B}$ using `ExtractMiddleBlock`, since we can choose the required fixed-point free element f to be δ.

The approach for $G = \mathrm{SL}(5,q)$ with $q \geq 2$ is the same. Observe that the cycle of G is the product of the cycles constructed in K and H, and no gluing is required.

10.2. Orthogonal groups. The groups $\Omega^+(d,2)$ with $d \leq 14$ are individual base cases, so let both $d, q \geq 4$. Recall that $\Omega^+(4,q)$ is the direct product of two copies of $\mathrm{SL}(2,q)$ arising from a tensor decomposition of the underlying space, see [43, Corollary 12.39]. This tensor decomposition is readily made explicit: by random selection, we construct an element of $\Omega^+(4,q)$ which acts as a scalar on one of the tensor factors and, using the algorithm of [27, §4], construct the tensor factors. We now use [19] to recognise constructively the copies of $\mathrm{SL}(2,q)$. The complexity for solving $\Omega^+(4,q)$ is the same as for $\mathrm{SL}(2,q)$. Since $\Omega^+(6,q)$ is an exterior square representation of $\mathrm{SL}(4,q)$, see [43, Corollary 12.21], it is constructively recognised by the algorithm described in [32]; the complexity is $O(\xi \log q + \log^2 q)$.

For Ω^- the individual base cases are $\Omega^-(d,2)$ with $d \in \{8,10,12,14\}$, and $\Omega^-(d,4)$ with $d \in \{8,10\}$. Recall that $\Omega^-(4,q) \cong \mathrm{SL}(2,q^2)$, see [43, Corollary 12.43], and an isomorphism can be defined by mapping the standard generators s, t, and δ of $\Omega^-(4,q)$ to $\left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right)$, $\left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right)$, and $\left(\begin{smallmatrix}\gamma&0\\0&\gamma^{-1}\end{smallmatrix}\right)$, respectively, with $\gamma \in \mathrm{GF}(q^2)$ primitive. The group $\Omega^-(6,q)$ is an exterior square representation of $\mathrm{SU}(4,q)$, see [43, Corollary 12.36]. The algorithm of [21] is used to find the preimage in $\mathrm{SU}(4,q)$ of every standard generator of $\Omega^-(6,q)$, and $\mathrm{SU}(4,q)$ is solved using [9]. The complexity is that for $\mathrm{SU}(4,q)$.

10.3. Symplectic groups. The individual base cases are $\mathrm{Sp}(d,2)$ with $d \leq 12$. The case $\mathrm{Sp}(2,q) = \mathrm{SL}(2,q)$ is solved using [19], and [9] is used for groups of degree 4.

The standard generators $\{s,t,\delta,u,v,x\}$ of $\mathrm{Sp}(6,q)$ with $q \geq 4$ are found as follows. First, via a base change, we construct subgroups $H = \mathrm{diag}(\mathrm{SL}(2,q), 1_4)$ and $K = \mathrm{diag}(1_2, \Omega^+(4,q))$ of G using `FirstSX` and `SecondSX`. The standard generators of $\mathrm{SL}(2,q)$ in H already contain s, t, and δ, and it remains to find u; observe that v and x can be constructed from u and the standard generators of $\Omega^+(4,q)$ in K. We obtain u in the centraliser C of the involution $i = \mathrm{diag}(\left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right))$, which is constructed using the standard generators of H and K. Note that i is not a good involution since its corank is odd. By [2, (7.6) & (7.10)], the natural C-module has sections of degrees $2,2,1,1$, and C acts as $\mathrm{SL}(2,q)$ on the factors of degree 2. Hence, u can be found in C by applying [24] and [19]. The complexity is $O(\log^2 q(\xi + \log^2 q))$.

10.4. The base case algorithm. We summarise these algorithms as a single Las Vegas algorithm `BaseCase`. It takes as input a bounded generating set X for a base case $G = \mathrm{SX}(d,q)$ of type *type*. It returns a base change matrix b and standard generators $\mathcal{S} \subseteq G^b$ of G. If d is even and *type* is SL or SU, or d is divisible by 4 and *type* is not Ω^-, then it also returns a good involution $i \in G^b$ of corank $d/2$, otherwise *false*. Following [9], `BaseCase` has complexity $O(\xi + \log^4 q + \chi \log q)$.

11. Constructing standard generators

Let $G = \mathrm{SX}(d,q) = \langle X \rangle$. We now describe `StandardGenerators` which constructs the standard generators of G as SLPs in X.

Lemma 11.1. *Algorithm* `StandardGenerators` *is correct, Las Vegas, and, if $q > 4$, then it has complexity*

$$O(d((\log^2 q/\log d)\xi + d^3 + d^2 \log d \log^3 q + \log^4 q + \chi \log q)).$$

PROOF. The correctness follows from the definition of the functions used in this algorithm. Although most of these functions use Monte Carlo algorithms, `StandardGenerators` is Las Vegas. We now discuss some details.

Algorithm 2: `StandardGenerators`(X, type)

/* X is a generating set for $G = \mathrm{SX}(d, q)$ of type *type*. Return base change matrix b and standard generators $\mathcal{S} \subseteq G^b$ of G. If d is even and G is linear or unitary, or d is divisible by 4 and G is of type Sp or Ω^+, then also return a good involution $i \in G^b$ of corank $d/2$, otherwise *false*. */

1 **begin**
2 **if** G is a base case **then return** `BaseCase`(X, type);
construct first subgroup and make first recursive call:
3 $H, b, m :=$ `FirstSX`(X, type);
4 let A be the group of type type_A generated by all upper left $m \times m$ blocks in H;
5 $\mathcal{S}_A, b_A, i_A :=$ `StandardGenerators`(X_A, type_A) where X_A is a generating set of A;
6 $\mathcal{S}_H := \{\mathrm{diag}(u, 1_{d-m}) \mid u \in \mathcal{S}_A\}$ and $i_H := \mathrm{diag}(i_A, 1_{d-m})$;
7 $H := H^s$ and $b := bs$ where $s := \mathrm{diag}(b_A, 1_{d-m})$;
construct second subgroup and make second recursive call:
8 $K :=$ `SecondSX`(X, H, b, m, i_H);
9 $K := K^e$ where $e := \mathrm{diag}(1_m, \star)$ such that G^{be} and lower block of K^e are standard copies;
10 $b := be$;
11 let B be the group generated by all lower right $(d-m) \times (d-m)$ blocks in K;
12 $\mathcal{S}_B, b_B, i_B :=$ `StandardGenerators`(X_B, type) where X_B is a generating set of B;
13 $\mathcal{S}_K := \{\mathrm{diag}(1_m, u) \mid u \in \mathcal{S}_B\}$, and $K := K^t$ and $b := bt$ where $t := \mathrm{diag}(1_m, b_B)$;
construct involution and swap groups:
14 **if** $i_B \neq$ *false* **then** $i_K := \mathrm{diag}(1_m, i_B)$ and $i := i_H i_K$ **else** $i :=$ *false*;
15 **if** *type* is Sp **then** swap H and K, and, accordingly, all other elements, $m := d - m$;
glue cycles:
16 $\mathcal{S} :=$ `GlueCycles`$(X, H, K, b, m, \mathcal{S}_H, \mathcal{S}_K)$;
17 **return** $\mathcal{S}, i, b$;
18 **end**

Define $z = 2$ if G is linear or unitary, and $z = 4$ otherwise. In Line 7, $\mathcal{S}_H \subseteq H$ and

$$H = \begin{pmatrix} \mathrm{SX}(m,q) & 0 \\ 0 & 1_{d-m} \end{pmatrix} \leq G^b$$

where m is divisible by z. If G is linear or unitary, then so is $\mathrm{SX}(m, q)$, otherwise its type is Ω^+. Observe that $i_A \in A$ is a good involution of corank $m/2$ since m is divisible by z.

In Line 10,

$$H = \begin{pmatrix} \mathrm{SX}(m,q) & 0 \\ 0 & 1_{d-m} \end{pmatrix} \leq G^b \quad \text{and} \quad K = \begin{pmatrix} 1_m & 0 \\ 0 & \mathrm{SX}(d-m,q) \end{pmatrix} \leq G^b$$

where $\mathrm{SX}(d - m, q)$ has the same type as G, and all of $\mathrm{SX}(m, q)$, $\mathrm{SX}(d - m, q)$, and G^b are standard copies. In Line 15, we ensure that $\mathrm{SX}(m, q)$ has the same type as G, unless G has type Ω^- in which case $\mathrm{SX}(m, q)$ has type Ω^+ and $\mathrm{SX}(d - m, q)$ has type Ω^-. In Line 16, `GlueCycles` completes the construction of the standard generators.

We now show that `StandardGenerators` returns a good involution if d is divisible by z and G is not of type Ω^-. This is true for the base cases and, by induction, we can assume that $i_A \in A$ is a good involution of corank $m/2$. By construction, m is divisible by z, and so is $d-m$. Again, by induction, $i_B \in B$ is a good involution of corank $(d - m)/2$. Lemma 6.2 shows that $i = i_H i_K$ is a good involution of corank $d/2$.

The cost of the base cases for the algorithm is $O(d(\log^4 q + \chi \log q))$. As shown in [28, Lemma 2.4], the cost of the recursive calls does not affect the complexity of the overall algorithm. The claim follows. □

12. Constructing involutions

Let $G = \mathrm{SX}(d, q) = \langle X \rangle$. For large even q, we cannot find an involution by a random search in G. The combination of `StandardGenerators` and Costi's algorithm [20] allows us to write every $g \in G$ as an SLP in X. Since the algorithm of [20] has complexity $O(d^3 \log q)$, this proves Theorems 1.3 and 1.4. We provide an alternative approach which is more efficient in practice.

In the following we describe the construction of two involutions in G; one of unspecified (small) corank and one of large corank. Again, we use recursion to classical groups of smaller degree.

12.1. Base cases. First, we consider the base cases for the recursion. Again, we could use [9], but we present alternatives that seem more efficient in practice. Observe that for all individual base cases (and small q in general) we could use a random search to find an involution. However, if we want to construct one of large corank, then this may not be efficient.

• Types SL and SU. The base cases are $d \in \{2, 3, 4, 5, 7\}$ with $q \geq 4$ (and some individual groups); we use [19], [30], and [9] for $d \in \{2, 3\}$. If $d \in \{5, 7\}$, then `FirstSX` is used to construct a group of degree $d - 1$. Degree 6 is handled by recursion to groups of degree 4 and 2. Degree 4 is handled as follows. First, we recurse to a group of degree 2 to find an involution of corank 1 in standard form. In its centraliser we construct elements

$$j_f = \begin{pmatrix} 1 & \star & \star & \star \\ 0 & 1 & f & \star \\ 0 & 0 & 1 & \star \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{with} \quad f \in \mathrm{GF}(q)$$

using the algorithm of [19]. Note that the order of j_f divides 4, and $[j_1, j_f]$ is an involution, usually with corank 2.

• Types Sp and Ω^+. In degree 6, we recurse to $\Omega^+(4, q)$. For degree 4 we use [9] and the methods described in Section 10.2.

• Type Ω^-. There is no good involution in $\Omega^-(4, q)$, and $\Omega^-(6, q)$ has only good involutions of corank 2. An involution of corank at least $\lfloor d/4 \rfloor - 1$ in $\Omega^-(d, q)$ is found either by a random search (for small fields), or by recursion to $\Omega^+(m, q)$ for some $m \geq d/2$ divisible by 4.

In summary, the resulting algorithm `InvolutionBaseCase` is Las Vegas. For our theoretical analysis, we may assume that the complexity of the base case algorithm is $O(\xi + \log^4 q + \chi \log q)$; see the comment at the beginning of Section 10.

12.2. Small corank. We construct involutions of small corank by recursion to subgroups of smaller degree (using `FirstSX`) until we can apply a base case method. We do not require a good involution, so for small q we could randomly search. The resulting algorithm is Las Vegas and, if $q > 4$, then its complexity is $O(\xi + d^3 \log d \log^3 q + \log^4 q + \chi \log q)$, see Lemma 11.1.

12.3. Large corank. To construct involutions of large corank, `StandardGenerators` is modified as follows: we replace `BaseCase` by `InvolutionBaseCase` and omit the calls to `GlueCycles`. While the theoretical complexity remains unchanged, in practice the algorithm is more efficient.

If d is even and G is of type SL and SU, or d is divisible by 4 and G is of type Sp or Ω^+, then Lemma 11.1 shows that the algorithm returns a good involution of corank $d/2$. For all other groups, the involution returned has corank precisely $\lfloor d/2 \rfloor$, or at least $\lfloor d/4 \rfloor - 1$ for Ω^-.

13. An implementation

Our implementation of these algorithms is available in MAGMA [6]. We use Brooksbank's implementations for $d = 3$ and 4 of the algorithms in [10, 13], and O'Brien's implementations of

the algorithms in [19, 30]. The current implementation of [10] for $d = 4$ is not optimal, requiring a search through the defining field.

We apply the MAGMA function COMPOSITIONTREE [38] to all individual base cases. All computations were carried out using MAGMA V2.18-8 on a computer with 28GB RAM and 3.07GHz processor. In Table 2, we list the CPU time in rounded seconds taken to construct the standard generators, involutions, and large corank involutions respectively. The time is averaged over three runs.

group / d	12	20	40	100
$\mathrm{SL}(d, 2^4)$	1	2	5	18
$\mathrm{SL}(d, 2^8)$	2	4	11	61
$\mathrm{SL}(d, 2^{12})$	2	4	13	64
$\mathrm{Sp}(d, 2^4)$	2	4	11	35
$\mathrm{Sp}(d, 2^8)$	4	8	19	93
$\mathrm{Sp}(d, 2^{12})$	13	19	34	105
$\Omega^+(d, 2^4)$	2	4	10	34
$\Omega^+(d, 2^8)$	3	6	17	85
$\Omega^+(d, 2^{12})$	3	8	21	90
$\Omega^-(d, 2^4)$	2	4	10	34
$\Omega^-(d, 2^8)$	3	6	17	85
$\Omega^-(d, 2^{12})$	3	7	21	90
$\mathrm{SU}(d, 2^2)$	2	2	5	21
$\mathrm{SU}(d, 2^4)$	6	12	30	122
$\mathrm{SU}(d, 2^6)$	13	26	57	185

group / d	12	20	40	100
$\mathrm{SL}(d, 2^4)$	0	0	0	0
$\mathrm{SL}(d, 2^8)$	0	0	0	2
$\mathrm{SL}(d, 2^{12})$	0	0	1	3
$\mathrm{Sp}(d, 2^4)$	0	0	0	0
$\mathrm{Sp}(d, 2^8)$	0	1	1	4
$\mathrm{Sp}(d, 2^{12})$	0	1	1	5
$\Omega^+(d, 2^4)$	0	0	0	0
$\Omega^+(d, 2^8)$	2	1	3	4
$\Omega^+(d, 2^{12})$	1	1	1	6
$\Omega^-(d, 2^4)$	0	0	0	0
$\Omega^-(d, 2^8)$	0	0	1	4
$\Omega^-(d, 2^{12})$	0	1	1	6
$\mathrm{SU}(d, 2^2)$	0	0	0	0
$\mathrm{SU}(d, 2^4)$	0	0	0	3
$\mathrm{SU}(d, 2^6)$	0	0	1	4

group / d	12	20	40	100
$\mathrm{SL}(d, 2^4)$	0	0	1	5
$\mathrm{SL}(d, 2^8)$	0	1	4	29
$\mathrm{SL}(d, 2^{12})$	0	1	4	25
$\mathrm{Sp}(d, 2^4)$	1	2	5	17
$\mathrm{Sp}(d, 2^8)$	2	4	8	41
$\mathrm{Sp}(d, 2^{12})$	2	3	9	38
$\Omega^+(d, 2^4)$	1	2	5	17
$\Omega^+(d, 2^8)$	1	3	8	40
$\Omega^+(d, 2^{12})$	2	3	9	39
$\Omega^-(d, 2^4)$	1	1	4	13
$\Omega^-(d, 2^8)$	1	2	6	27
$\Omega^-(d, 2^{12})$	1	2	7	32
$\mathrm{SU}(d, 2^2)$	1	1	1	6
$\mathrm{SU}(d, 2^4)$	1	1	4	37
$\mathrm{SU}(d, 2^6)$	1	2	4	25

TABLE 2. Times for standard generators, involutions, and involutions of large corank

References

[1] S. Ambrose, S. H. Murray, C. E. Praeger, and C. Schneider. Constructive membership testing in black-box classical groups, Proceedings of The Third International Congress on Mathematical Software. *Lecture Notes in Computer Science*, **6327** (2010), 54–57.

[2] M. Aschbacher and G. M. Seitz. Involutions in Chevalley groups over fields of even order, *Nagoya Math. J.* **63** (1976), 1–91.

[3] L. Babai. Local expansion of vertex-transitive graphs and random generation in finite groups. *Theory of Computing*, (Los Angeles, 1991), pp. 164–174. Association for Computing Machinery, New York, 1991.

[4] L. Babai and R. Beals. A polynomial-time theory of black box groups. Groups St Andrews 1997 in Bath, I. London Math. Soc. Lecture Note Series **260** (1999), 30–64.

[5] L. Babai, P. Pálfy, and J. Saxl. On the number of p-regular elements in finite simple groups. LMS J. Comput. Math. **12** (2009), 82–119.

[6] W. Bosma, J. Cannon, and C. Playoust. The MAGMA algebra system I: The user language, *J. Symbolic Comput.* **24** (1997), 235–265.

[7] J. N. Bray. An improved method of finding the centralizer of an involution. *Arch. Math. (Basel)* **74** (2000), 241–245.

[8] J. N. Bray and R. A. Wilson. Constructing the centraliser of an involution in even characteristic. In preparation, 2013.

[9] P. A. Brooksbank. Constructive recognition of classical groups in their natural representation. *J. Symbolic Comput.* **35** (2003), 195–239.

[10] P. A. Brooksbank. Fast constructive recognition of black-box unitary groups. *LMS J. Comput. Math.* **6** (2003), 162–197.

[11] P. A. Brooksbank and W. M. Kantor. On constructive recognition of a black box $\mathrm{PSL}(d, q)$. In *Groups and Computation, III (Columbus, OH, 1999)*, pp. 95–111. Volume 8 of *Ohio State Univ. Math. Res. Inst. Publ.*, de Gruyter, Berlin, 2001.

[12] P. A. Brooksbank and W. M. Kantor. Fast constructive recognition of black box orthogonal groups. *J. Algebra* **300** (2006), 256–288.
[13] P. A. Brooksbank. Fast constructive recognition of black box symplectic groups. *J. Algebra* **320** (2008), 885–909.
[14] Damien Burns, Heiko Dietrich, C.R. Leedham-Green, and E.A. O'Brien. Black-box constructive recognition for classical groups. *In preparation*, 2013.
[15] R. W. Carter. Finite groups of Lie type. Conjugacy classes and complex characters. Reprint of the 1985 original. John Wiley & Sons, Ltd., Chichester, 1993. xii+544 pp.
[16] F. Celler, C. R. Leedham-Green, S. H. Murray, A. C. Niemeyer, and E. A. O'Brien. Generating random elements of a finite group. *Comm. Algebra* **23** (1995), 4931–4948.
[17] F. Celler and C. R. Leedham-Green. A constructive recognition algorithm for the special linear group. In *The atlas of finite groups: ten years on (Birmingham, 1995)*, volume 249 of *London Math. Soc. Lecture Note Ser.*, pp. 11–26, Cambridge, 1998. Cambridge Univ. Press.
[18] M. Conder and C. R. Leedham-Green. Fast recognition of classical groups over large fields. In *Groups and Computation, III (Columbus, OH, 1999)*, volume 8 of *Ohio State Univ. Math. Res. Inst. Publ.*, pp. 113–121, de Gruyter, Berlin, 2001.
[19] M. D. E. Conder, C. R. Leedham-Green, and E. A. O'Brien. Constructive recognition of $\mathrm{PSL}(2, q)$. *Trans. Amer. Math. Soc.* **358** (2006), 1203–1221.
[20] E. M. Costi. Constructive membership testing in classical groups. PhD thesis, Queen Mary, University of London, 2009.
[21] C. Greenhill. An algorithm for recognising the exterior square of a matrix. *Linear and Multilinear Algebra* **46** (1999), 213–244.
[22] R. Guralnick and F. Lübeck. On p-singular elements in Chevalley groups in characteristic p. Groups and computation, III (Columbus, OH, 1999), 169–182, Ohio State Univ. Math. Res. Inst. Publ., 8, de Gruyter, Berlin, 2001.
[23] D. F. Holt, B. Eick, and E. A. O'Brien. Handbook of computational group theory. Chapman and Hall/CRC, London, 2005.
[24] D. F. Holt and S. Rees. Testing modules for irreducibility. *J. Aust. Math. Soc.* **57** (1994), 1–16.
[25] W. M. Kantor and A. Lubotzky. The probability of generating a finite classical group. *Geom. Dedicata* **36** (1990), 67–87.
[26] W. M. Kantor and Á. Seress. Black box classical groups. *Mem. Amer. Math. Soc.* **149**, 2001.
[27] C. R. Leedham-Green and E. A. O'Brien. Tensor Products are Projective Geometries. *J. Algebra*, **189** (1997), 514–528.
[28] C. R. Leedham-Green and E. A. O'Brien. Constructive recognition of classical groups in odd characteristic. *J. Algebra*, **322** (2009), 833–881.
[29] S. A. Linton. The Art and Science of Computing in Large Groups. *Computational algebra and number theory* (Sydney, 1992), 91–109, Math. Appl. 325, Dordrecht 1995.
[30] F. Lübeck, K. Magaard, and E. A. O'Brien. Constructive recognition of $\mathrm{SL}_3(q)$. *J. Algebra* **316** (2007), 619–633.
[31] F. Lübeck, A. C. Niemeyer, and C. E. Praeger. Finding involutions in finite Lie type groups of odd characteristic. *J. Algebra* **321** (2009), 3397–3417.
[32] K. Magaard, E. A. O'Brien, and Á. Seress. Recognition of small dimensional representations of general linear groups. *J. Aust. Math. Soc.* **85** (2008), 229–250.
[33] S. H. Murray and C. M. Roney-Dougal. Constructive homomorphisms for classical groups. *J. Symbolic Comp.* **46** (2011), 371–384.
[34] P. M. Neumann and C. E. Praeger. Derangements and eigenvalue-free elements in finite classical groups. *J. London Math. Soc.* (2) **58** (1998), 564–584.
[35] A. C. Niemeyer and C. E. Praeger. A recognition algorithm for classical groups over finite fields, *Proc. London Math. Soc.* **77** (1998), 117–169.
[36] Christopher W. Parker and Robert A. Wilson. Recognising simplicity of black-box groups. *J. Algebra* **324** (2010), 885–915.
[37] E. A. O'Brien. Towards effective algorithms for linear groups. *Finite Geometries, Groups and Computation*, (Colorado), pp. 163–190, de Gruyter, Berlin, 2006.
[38] E. A. O'Brien. Algorithms for matrix groups. Groups St Andrews 2009 in Bath II, London Math. Soc. Lecture Note Series **388** (2011), 297–323.
[39] I. Pak. The product replacement algorithm is polynomial. In *41st Annual Symposium on Foundations of Computer Science (Redondo Beach, CA, 2000)*, pp. 476–485, IEEE Comput. Soc. Press, Los Alamitos, CA, 2000.

[40] C. E. Praeger, Á. Seress and S. Yalçınkaya. Generation of finite classical groups by pairs of elements with large fixed point spaces. *In preparation*, 2013.
[41] Á. Seress. Permutation group algorithms. Volume 152 of Cambridge Tracts in Mathematics, Cambridge University Press, Cambridge, 2003.
[42] R. Solomon and A. Turull. Chains of subgroups in groups of Lie type I. *J. Algebra* **132** (1990), 174–184.
[43] D. E. Taylor. The geometry of the classical groups. Sigma Series in Pure Mathematics, **9**. Heldermann Verlag, Berlin, 1992.

SCHOOL OF MATHEMATICAL SCIENCES, MONASH UNIVERSITY, VIC 3800, AUSTRALIA
E-mail address: heiko.dietrich@monash.edu

SCHOOL OF MATHEMATICAL SCIENCES, QUEEN MARY, UNIVERSITY OF LONDON, LONDON E1 4NS, UNITED KINGDOM
E-mail address: c.r.leedham-green@qmul.ac.uk

LEHRSTUHL D FÜR MATHEMATIK, RWTH AACHEN, TEMPLERGRABEN 64, 52062 AACHEN, GERMANY
E-mail address: frank.luebeck@math.rwth-aachen.de

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF AUCKLAND, PRIVATE BAG 92019, AUCKLAND, NEW ZEALAND
E-mail address: e.obrien@auckland.ac.nz