

Constructive recognition of classical groups in their natural representation*

Peter A. Brooksbank[†]

June 6, 2004

Abstract

Let $\mathcal{S} \subset \mathrm{GL}(V)$ be a given set of generators for a group G , where V is a finite dimensional vector space over a finite field \mathbb{F} . We present an algorithm which recognises, constructively, when G is $\mathrm{Sp}(V)$, $\mathrm{SU}(V)$ or $\Omega^\varepsilon(V)$. Our algorithm handles all of those classical groups uniformly and runs in time which is polynomial in the input length, assuming a Discrete Logarithm Oracle for \mathbb{F} .

1 Introduction and results

Let V be a finite dimensional vector space over the finite field \mathbb{F} . In this paper we will use the term *classical form on V* to mean a nondegenerate alternating, hermitian or quadratic form on V , and a *classical group on V* will be a perfect subgroup of $\mathrm{GL}(V)$ preserving such a form. That is to say, the classical groups on V are just $\mathrm{Sp}(V)$, $\mathrm{SU}(V)$ or $\Omega^\varepsilon(V)$ ($\varepsilon = +, -$ or 0): we will refer to these possibilities as cases **S**, **U** and **O**, respectively.

Suppose that $G = \langle \mathcal{S} \rangle \leq \mathrm{SL}(V)$ is given. Then one can efficiently test whether or not G preserves a classical form on V and, if it does, one can also obtain a matrix representing such a G -invariant form. Hence we may assume that G is a subgroup of a particular classical group on V . A *nonconstructive recognition algorithm*, such as the algorithm of Niemeyer and Praeger [NiP], can then be applied to decide whether or not G is that classical group. If it is then one can decide via an elementary test whether or not any given element of $\mathrm{GL}(V)$ is in G . The additional feature of a *constructive recognition algorithm* is a routine which, for any given element $g \in G$, constructs a *straightline program (SLP)* from \mathcal{S} to g . Constructive recognition algorithms are a key ingredient in the ambitious project to construct a composition series for

*This paper is based on part of the author's Ph. D. thesis [Br2], completed under the direction of W. Kantor.

[†]This research was supported, in part, by the National Science Foundation.

any given matrix group (this has become known as the “computational matrix group project” [LG]).

Because of the size of the groups concerned, all known constructive recognition algorithms for finite simple groups employ randomized algorithms rather than the more traditional deterministic ones. Randomized algorithms for groups fundamentally require a method of selecting nearly uniformly distributed random elements; such a method is provided by Babai [Ba]. A randomized algorithm is called *Monte Carlo* if the output of the algorithm is correct with probability $> 1/2$ (higher reliability can be achieved by repetition and majority vote). *Las Vegas* algorithms form a subclass of Monte Carlo algorithms. Here a positive output is guaranteed to be correct, but failure may be reported (with probability $< 1/2$) if a suitable output has not been determined after a prescribed time. The randomized algorithms presented here will always be Las Vegas.

The principal reason for treating the natural representation separately from other representations is to make use of algorithmic techniques which are unique to this setting. For example, we will frequently be able simply to write down matrices lying in certain subgroups and then use linear algebra to construct them from known generators for that subgroup. This is a luxury which is not available in other settings and it gives rise to much faster algorithms (cf. 4.5.1).

Celler and Leedham-Green [CeLG] provided the first example of this type of algorithm for the case when G contains the special linear group $SL(V)$ and Celler [Ce] later dealt with the case $G = Sp(V)$. In [Br1], a simplified version of our algorithm is given which deals only with the case $G = \Omega(d, q)$ for d odd. Our goal here is a general algorithm for all classical groups with a proven asymptotic running time. Hence we both develop more theory and include more detailed proofs than in [Ce, CeLG, Br1]. We obtain an alternative algorithm to [Ce] for $G = Sp(V)$ and we will use the timing in [Ce] as a measure of the efficiency of our algorithm (cf. 1.1 and also section 7).

A drawback to the algorithms in [Ce] and [CeLG] is that their running times are not polynomial in the length of the input. In particular, they both require the construction of a multiple of q random group elements in order to guarantee success with high probability. However, the recent advances of Conder, Leedham-Green and O’Brien [CoLG, LGO2] show that explicit occurrences of q in the running time of an algorithm for any irreducible representation of $SL(2, q)$ (over a field having the natural characteristic) can be avoided at the expense of using an “oracle” which computes discrete logarithms in $GF(q)^*$.

In this paper we also assume the availability of a discrete log oracle and use this, together with the algorithm in [CoLG], to devise an algorithm which recognises constructively any classical group in its natural representation. We will see that $SL(2, q)$ -subgroups are the heart of the matter; indeed, we will show that $SL(2, q)$ is the polynomial time bottleneck.

The main advantage of our algorithm is that, while certain subroutines need to be slightly

modified for the different classical groups, the architecture of the main algorithm remains the same. That is, we are able to deal with all classical groups simultaneously. The algorithm has been implemented by the author in the computer algebra system GAP [GAP4] and the results of some preliminary performance tests are summarised in section 7.

Our main result can be stated as follows:

Theorem 1.1 *Let $G = \langle \mathcal{S} \rangle \leq \text{GL}(V)$ be given, where V is a vector space of dimension $d \geq 2$ over the finite field $\text{GF}(q)$. Then there is a Las Vegas algorithm which, if G is a classical group on V , finds a new generating set \mathcal{T} for G (whose elements are constructed using SLPs from \mathcal{S}) having the property that an SLP of length $O(d^2 \log q)$ can be found from \mathcal{T} to any given $g \in G$.*

Assuming the availability of an oracle to compute discrete logarithms in $\text{GF}(q)^$, in $\text{GF}(\sqrt{q})^*$ if G is unitary, or in $\text{GF}(q^2)^*$ in the single case $G = \Omega^-(4, q)$, the algorithm to construct \mathcal{T} runs in polynomial time*

$$O(d^3 \log q (d + \log d \log^3 q) + \xi \{d + \log \log q\} + \mu \{|\mathcal{S}| + d^2 \log^2 q\} + \chi \log q),$$

while each application of the deterministic routine to write an SLP from \mathcal{T} takes $O(d^3 \log q + \log^2 q)$ time.

Here ξ is an upper bound on the time requirement per element for the construction of independent, (nearly) uniformly distributed random elements of G , μ is an upper bound on the time required to perform each group operation in G , and χ represents the cost, per call, to the appropriate discrete logarithm oracle.

Remarks:

- (i) The groups $\text{SL}(V)$ could also have been included in Theorem 1.1, but were omitted to facilitate a more uniform treatment. An algorithm for $\text{SL}(V)$ is given in [CeLG], and suggestions for an improved algorithm, in the same spirit as Theorem 1.1, are discussed in [CoLG].
- (ii) The $\text{GF}(q^2)^*$ oracle for $G = \Omega^-(4, q)$ arises from the isomorphism $\Omega^-(4, q) \cong \text{PSL}(2, q^2)$ (cf. 6.1.4).
- (iii) As suggested by the statement of the Theorem, the basic idea of the algorithm (as in [Ce, CeLG]) is to construct a set \mathcal{T} from which it is easy to write a straightline program of modest length to any given element $g \in G$. In [CeLG], \mathcal{T} consists of sufficiently many transvections to perform Gaussian elimination in $\text{SL}(V)$. In 5.2 we give a more involved analogue of Gaussian elimination which works inside any classical group, and the set \mathcal{T} will consist of the elements necessary to execute this procedure.

To obtain a comparison of our running time with that of existing algorithms, we state the following consequence of Theorem 1.1.

Corollary 1.2 *For $d \geq 5$, there is an alternative version of the constructive recognition algorithm in Theorem 1.1 which does not assume a discrete logarithm oracle and constructs \mathcal{T} in time*

$$O(d^3 \log q(d + \log d \log^4 q) + \xi\{d + \log \log q\} + \mu\{|\mathcal{S}| + d^2 \log^2 q\} + q^\varepsilon \log q),$$

where $\varepsilon = 1/2$ if G is unitary, $\varepsilon = 2$ if $G = \Omega^-(4, q)$, and $\varepsilon = 1$ otherwise.

Remark: The term $q^\varepsilon \log q$ that appears above arises from the preprocessing required to form a list of the necessary field elements. Discrete logarithms are then found using a binary search in that list.

1.1 Timing comparisons

Perhaps the most important comparison is with the black box classical group algorithm in [KS] since, if we do not obtain a significantly improved running time over that algorithm, one might question the value of treating the natural representation separately. As in [KS], Theorem 1.1 (vii) (ignoring the cost of verifying a presentation), the black box algorithm runs in time

$$O(\xi d^2 q \log d \log q + \mu(d^4 q^{3/2} \log^2 q \log^3 d + d^5 \log^{5/2} q)).$$

Comparing with the running time in Corollary 1.2 it is clear that our algorithm runs much more quickly (noting, of course, that our “ q^2 ” is their “ q ” when $G = \Omega^-(4, q) \cong \text{PSL}(2, q^2)$).

An examination of the subroutines in [Ce] for $G = \text{Sp}(V)$ reveals that, in our notation, the main algorithm in [Ce] runs in time $O(d^2 q(\xi + \mu)) = O(\xi d^2 q)$. The coefficient of ξ in Theorem 1.1 is $d + \log \log q$, so our algorithm uses almost a factor of dq fewer random elements.

2 Classical Group Preliminaries

We assume a basic familiarity with the classical groups and the geometries associated with them. We will introduce the terminology, notation and theory necessary for our algorithm, but refer to [KS, Kil, Ta] both for more complete treatments of classical groups and for proofs of some elementary results stated here.

Throughout this section, let $\mathbb{F} = \text{GF}(p^l) = \text{GF}(\tilde{q})$ be a finite field, and let V , an \mathbb{F} -space of dimension d , be the natural module of a classical group G of Witt index m . We may assume that $d > 2$ since otherwise G is solvable or is isomorphic to $\text{SL}(2, q)$ (the latter case is discussed in 6.1).

Let $k = l/2$ in case **U** and $k = l$ otherwise. Set $q := p^k$ so that $\tilde{q} = q$ in cases **O** and **S** and $\tilde{q} = q^2$ in case **U**. In case **O**, let φ denote a nondegenerate G -invariant quadratic form on V . Let $(\ , \)$ denote a nondegenerate G -invariant alternating or hermitian form in cases **S** or **U** respectively, or the symmetric form associated with φ in case **O**. In case **U**, let “overbar” denote the involutory automorphism $\lambda \mapsto \lambda^q$ of \mathbb{F} .

2.1 Standard bases

The module V has a *standard basis* \mathcal{B} of one of the following types:

$$\begin{aligned} \mathcal{B} &= e_1, \dots, e_m, f_1, \dots, f_m && \text{cases } \mathbf{S}, \mathbf{U}^e, \mathbf{O}^+ (d = 2m); \\ \mathcal{B} &= e_1, \dots, e_m, v, f_1, \dots, f_m && \text{cases } \mathbf{U}^o, \mathbf{O}^o (d = 2m + 1); \text{ or} \\ \mathcal{B} &= e_1, \dots, e_m, v_1, v_2, f_1, \dots, f_m && \text{case } \mathbf{O}^- (d = 2m + 2). \end{aligned} \tag{1}$$

Here $(e_i, e_j) = (f_i, f_j) = 0$, $(e_i, f_j) = \delta_{ij}$, $v, v_1, v_2 \in \langle e_1, \dots, e_m, f_1, \dots, f_m \rangle^\perp$ and $(v, v) = 1$ (cf. [KIL], Propositions 2.3.2, 2.4.1 and 2.5.3). In case **O**⁻, v_1 and v_2 behave as in the following lemma.

Lemma 2.1 *Let U be a definite line of the orthogonal space V (that is, one which contains no singular vectors) and let ρ be a generator of \mathbb{F}^* . Then, replacing φ with $\rho\varphi$ if necessary, there exists a basis v_1, v_2 of U such that*

- (i) $(v_1, v_2) = 1$ and $\varphi(v_i) = \alpha_i$ for some $\alpha_i \in \mathbb{F}^*$ ($i = 1, 2$) when q is even; or
- (ii) $(v_1, v_2) = 0$ and $(v_i, v_i) = 1$ ($i = 1, 2$) when $q \equiv 3 \pmod{4}$; or
- (iii) $(v_1, v_2) = 0$, $(v_1, v_1) = 1$ and $(v_2, v_2) = \rho$, when $q \equiv 1 \pmod{4}$.

Proof. (i) is trivial. For (ii) and (iii), replacing φ with $\rho\varphi$ (and hence $(\ , \)$ with $\rho(\ , \)$) if necessary, choose v_1 such that $(v_1, v_1) = 1$ and choose $v_2 \in v_1^\perp$. Note that the equation $(xv_1 + v_2, xv_1 + v_2) = x^2 + (v_2, v_2) = 0$ has a root in \mathbb{F} if $q \equiv 1 \pmod{4}$ and (v_2, v_2) is a square, or if $q \equiv 3 \pmod{4}$ and (v_2, v_2) is a nonsquare. The result now follows easily. \square

We will say that classical groups G and H are *of the same type* if they are both in one of the cases **S**, **U**^e, **U**^o, **O**⁺, **O**^o, **O**⁻; i.e. there are six “types” of classical group. Let \mathbb{F}_p denote the prime subfield $\text{GF}(p)$ of \mathbb{F} and let \mathcal{B} be a standard basis of V . For a generator ρ of \mathbb{F}^* , let \mathcal{B}_ρ be the ordered \mathbb{F}_p -basis obtained from \mathcal{B} by replacing each vector $u \in \mathcal{B}$ by the l vectors $u, \rho u, \dots, \rho^{l-1}u$. We call \mathcal{B}_ρ *the standard \mathbb{F}_p -basis obtained from \mathcal{B} and ρ* .

Table 1: Matrix constraints for cases \mathbf{S} , \mathbf{U}^e , and \mathbf{O}^+

Case \mathbf{S}	Case \mathbf{U}^e	Case \mathbf{O}^+
$AB^{\text{tr}} - BA^{\text{tr}} = 0$	$A\bar{B}^{\text{tr}} + B\bar{A}^{\text{tr}} = 0$	$AB^{\text{tr}} + BA^{\text{tr}} = 0$
$CD^{\text{tr}} - DC^{\text{tr}} = 0$	$C\bar{D}^{\text{tr}} + D\bar{C}^{\text{tr}} = 0$	$CD^{\text{tr}} + DC^{\text{tr}} = 0$
$AD^{\text{tr}} - BC^{\text{tr}} = I$	$A\bar{D}^{\text{tr}} + B\bar{C}^{\text{tr}} = I$	$AD^{\text{tr}} + BC^{\text{tr}} = I$

2.2 Matrices

Let G be a classical group with natural module V and let \mathcal{B} be a standard basis for V . We now consider the matrix representing an arbitrary element of G relative to \mathcal{B} .

Cases \mathbf{S} , \mathbf{U}^e , \mathbf{O}^+ : $\mathcal{B} = e_1, \dots, e_m, f_1, \dots, f_m$ and elements of G have matrix

$$g = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \quad (2)$$

where A, B, C, D are $m \times m$ matrices satisfying the constraints in Table 1.

The constraints arise from the fact that G preserves $(\ , \)$ in each case. There are some additional constraints in case \mathbf{O}^+ when q is even arising from the fact that G also preserves a quadratic form and we will state these as we need them.

Cases \mathbf{U}^o , \mathbf{O}^o : Elements of G have matrix

$$g = \begin{pmatrix} A & \xi^{\text{tr}} & B \\ \eta & \lambda & \omega \\ C & \zeta^{\text{tr}} & D \end{pmatrix}, \quad (3)$$

where A, B, C, D are $m \times m$, ξ, η, ω, ζ are $1 \times m$ and $\lambda \in \mathbb{F}$ satisfying the constraints in Table 2.

Case \mathbf{O}^- : Elements of G have matrix

$$g = \begin{pmatrix} A & \xi_1^{\text{tr}} & \xi_2^{\text{tr}} & B \\ \eta_1 & \lambda_{11} & \lambda_{12} & \omega_1 \\ \eta_2 & \lambda_{21} & \lambda_{22} & \omega_2 \\ C & \zeta_1^{\text{tr}} & \zeta_2^{\text{tr}} & D \end{pmatrix}. \quad (4)$$

There are several more constraints of a similar type to those in case \mathbf{O}^o above. However, there are three different sets of constraints corresponding to the three different possibilities in Lemma 2.1. Furthermore, when q is even, there are additional constraints arising from the fact that G preserves a quadratic form. We therefore opt to introduce them only as we need them.

Table 2: Matrix constraints for cases \mathbf{U}° and \mathbf{O}°

Case \mathbf{U}°	Case \mathbf{O}°
$A\bar{B}^{\text{tr}} + B\bar{A}^{\text{tr}} + \xi^{\text{tr}}\bar{\xi} = 0$	$AB^{\text{tr}} + BA^{\text{tr}} + \xi^{\text{tr}}\xi = 0$
$C\bar{D}^{\text{tr}} + D\bar{C}^{\text{tr}} + \zeta^{\text{tr}}\bar{\zeta} = 0$	$CD^{\text{tr}} + DC^{\text{tr}} + \zeta^{\text{tr}}\zeta = 0$
$A\bar{D}^{\text{tr}} + B\bar{C}^{\text{tr}} + \xi^{\text{tr}}\bar{\zeta} = I$	$AD^{\text{tr}} + BC^{\text{tr}} + \xi^{\text{tr}}\zeta = I$
$A\bar{\omega}^{\text{tr}} + B\bar{\eta}^{\text{tr}} + \bar{\lambda}\xi^{\text{tr}} = 0$	$A\omega^{\text{tr}} + B\eta^{\text{tr}} + \lambda\xi^{\text{tr}} = 0$
$C\bar{\omega}^{\text{tr}} + D\bar{\eta}^{\text{tr}} + \bar{\lambda}\zeta^{\text{tr}} = 0$	$C\omega^{\text{tr}} + D\eta^{\text{tr}} + \lambda\zeta^{\text{tr}} = 0$
$\eta\bar{\omega}^{\text{tr}} + \omega\bar{\eta}^{\text{tr}} + \lambda\bar{\lambda} = 1$	$\eta\omega^{\text{tr}} + \omega\eta^{\text{tr}} + \lambda^2 = 1$

2.3 Transvection groups

A vector $v \in V$ is *singular* if $(v, v) = 0$ in cases \mathbf{S} and \mathbf{U} , or if $\varphi(v) = 0$ in case \mathbf{O} . Let x be a singular point of V . In cases \mathbf{S} and \mathbf{U} , G contains a subgroup $T(x)$ of order q which is the identity on x^\perp and on V/x . The group $T(x)$ is called the *group of (x, x^\perp) -transvections*; G -conjugates of $T(x)$ are also called *long root groups* of G .

Lemma 2.2 ([KS], 5.1.2 and 6.1.2) *For singular points $x \neq y$ of V :*

- (i) *if $x \in y^\perp$, then $\langle T(x), T(y) \rangle \cong T(x) \times T(y)$ has order q^2 ; or*
- (ii) *if $x \notin y^\perp$, then $V = \langle x, y \rangle \perp \langle x, y \rangle^\perp$ and $\langle T(x), T(y) \rangle \cong \text{SL}(2, q)$ inducing this group on the first summand and the identity on the second.*

Remark: We note that $\text{Sp}(2, q) \cong \text{SU}(2, q) \cong \text{SL}(2, q)$ ([KLL], Proposition 2.9.1(i)).

2.4 Stabilisers of singular points

Let x and y be the singular points $\langle e_1 \rangle$ and $\langle f_1 \rangle$ respectively. The point-stabiliser G_x splits as the semidirect product

$$G_x = Q(x) \rtimes G_{x,y}, \quad (5)$$

where $Q(x) = O_p(G_x)$, the largest normal p -subgroup of G_x . For $1 \leq i \leq m$ and $w \in \langle e_i, f_i \rangle^\perp$, let $r_i(w, \lambda)$ be the linear transformation defined in Table 3. Then $Q(x)$ consists of all possible transformations $r_1(w, \lambda)$.

Let $r'_i(w, \lambda)$ be the linear transformation $r_i(w, \lambda)$ with all occurrences of e_i in Table 3 replaced with f_i . Then $Q(y)$ consists of all possible linear transformations $r'_1(w, \lambda)$. The next result summarises some elementary properties of $Q(x)$, each of which either is an easy calculation or is in [KS], 4.1.3, 5.1.3 or 6.1.3.

Table 3: The linear transformations $r_i(w, \lambda)$

Case	$r_i(w, \lambda)$	λ	$ Q $
S	$u \mapsto u - (u, w - \lambda e_i)e_i - (u, e_i)w$	$\lambda \in \mathbb{F}$	$q^{d-2} \cdot q$
U	$u \mapsto u + (u, w - \lambda e_i)e_i - (u, e_i)w$	$\lambda + \bar{\lambda} = (w, w)$	$(q^2)^{d-2} \cdot q$
O	$u \mapsto u + (u, w - \lambda e_i)e_i - (u, e_i)w$	$\lambda = \varphi(w)$	q^{d-2}

Theorem 2.3 *Let $Q = Q(x)$, let $r(w, \lambda) = r_1(w, \lambda)$ and let $T = T(x)$ (in cases **S** and **U**). Then the following hold:*

- (i) *T is a normal subgroup of Q and consists of all elements $r(0, \lambda)$ for $\lambda \in \mathbb{F}$, where $\lambda + \bar{\lambda} = 0$ in case **U**.*
- (ii) *$r(w, \lambda)^g = r(wg, \lambda)$ for all $g \in (G_{x,y})' = G_{e_1, f_1}$.*
- (iii) *$r(w, \lambda) \cdot r(w', \lambda') = r(w + w', \lambda + \lambda' + (w, w'))$.*
- (iv) *In case **S**, $[r(w, \lambda), r(w', \lambda')] = r(0, 2(w, w'))$; when q is even, Q is elementary abelian; when q is odd, $Z(Q) = T = \Phi(Q)$ (Frattini subgroup) and Q/T is elementary abelian.*
- (v) *In case **U**, $[r(w, \lambda), r(w', \lambda')] = r(0, (w, w') - (w', w))$; $Z(Q) = \Phi(Q) = T$ and Q/T is elementary abelian.*
- (vi) *In case **O**, Q is elementary abelian.*
- (vii) *Q acts regularly on the set of singular points not perpendicular to x .*

Observe that the group Q (in case **O**) or Q/T (in cases **S** and **U**) has order $|\mathbb{F}^{d-2}|$. In fact, it is the natural module of the subgroup $(G_{x,y})'$ of G .

Corollary 2.4 *Let $\tilde{Q} = Q$ in case **O**, and let $\tilde{Q} = Q/T$ in cases **S** and **U**. Then $(G_{x,y})'$ is a classical group of the same type as G , and the $(G_{x,y})'$ -modules \tilde{Q} and $\langle x, y \rangle^\perp$ are isomorphic (natural) modules for $(G_{x,y})'$.*

Proof. The map $w \mapsto r_1(w, \varphi(w))$ in case **O**, and $w \mapsto r_1(w, \lambda)T$ (where $\lambda = 0$ in case **S** and $\lambda + \bar{\lambda} = -(w, w)$ in case **U**), defines a $(G_{x,y})'$ -module isomorphism $\langle x, y \rangle^\perp \rightarrow \tilde{Q}$ (this follows from Theorem 2.3(ii) and (iii)). \square

Table 4: Structure of the subgroup $L = G_{E,F}$

case	typical element	constraints	$ L $
S	$\text{diag}(A, A^{-\text{tr}})$		N
U^e	$\text{diag}(A, \bar{A}^{-\text{tr}})$	$\det(A) = \det(\bar{A})$	$N/(q+1)$
U^o	$\text{diag}(A, \lambda, \bar{A}^{-\text{tr}})$	$\lambda \det(A) = \det(\bar{A})$	N
O⁺	$\text{diag}(A, A^{-\text{tr}})$	$\det(A) = \square$	$N/(2, q-1)$
O^o	$\text{diag}(A, 1, A^{-\text{tr}})$	$\det(A) = \square$	$N/2$
O⁻	$\text{diag}(A, \Lambda, A^{-\text{tr}})$	Equation (7)	$(q+1)N/(2, q+1)$

2.5 Stabilisers of maximal t.s. subspaces

A subspace $U \leq V$ is *totally singular (t.s.)* if $(U, U) = 0$ in cases **S** and **U**, or if $\varphi(U) = 0$ in case **O**. Let $E = \langle e_1, \dots, e_m \rangle$ and $F = \langle f_1, \dots, f_m \rangle$, both maximal t.s. subspaces of V . Then the subspace stabilisers G_E and G_F split as semidirect products

$$G_E = U(E) \rtimes L \quad \text{and} \quad G_F = U(F) \rtimes L, \quad (6)$$

where $U(E) = O_p(G_E)$, $U(F) = O_p(G_F)$ and $L = G_{E,F}$. We now use the matrix constraints in Tables 1 and 2 to study the structure of the subgroups L , $U(E)$ and $U(F)$.

2.5.1 The subgroup L

Let $\tilde{q} = q^2$ in case **U** and $\tilde{q} = q$ otherwise, and let $N = |\text{GL}(m, \tilde{q})|$. For $\lambda \in \mathbb{F}^*$ write $\lambda = \square$ if λ is a square and $\lambda = \boxtimes$ otherwise. The structure of L is summarised in Table 4, where $A \in \text{GL}(m, \tilde{q})$ and $\Lambda \in \text{GL}(2, q)$ and the constraints on the matrix entries in case **O⁻** are as follows

$$\text{if } \det(A) = \begin{Bmatrix} \square \\ \boxtimes \end{Bmatrix} \text{ then } \Lambda \in \left\{ \begin{array}{l} \Omega^-(2, q) \\ \text{SO}^-(2, q) \setminus \Omega^-(2, q) \end{array} \right\}. \quad (7)$$

2.5.2 The subgroups $U(E)$ and $U(F)$

As matrices relative to \mathcal{B} we have $U(F) = U(E)^{\text{tr}}$ so we consider only the subgroup $U = U(E)$. For a row vector $z \in \mathbb{F}^m$, define \tilde{z} to be \bar{z} in case **U** and z otherwise. Define matrices $u(M)$, $u(z, M)$ and $u(z_1, z_2, M)$ to be

$$\begin{pmatrix} I & 0 \\ M & I \end{pmatrix}, \quad \begin{pmatrix} I & 0 & 0 \\ -\tilde{z} & 1 & 0 \\ M & z^{\text{tr}} & I \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} I & 0 & 0 & 0 \\ z_{\sigma(1)} & 1 & 0 & 0 \\ z_{\sigma(2)} & 0 & 1 & 0 \\ M & -z_1^{\text{tr}} & -\alpha z_2^{\text{tr}} & I \end{pmatrix} \quad (8)$$

Table 5: Structure of the subgroup U

case	typical element	constraints	$ U $
S	$u(M)$	$M - M^{\text{tr}} = 0$	$q^{m(m+1)/2}$
U^e	$u(M)$	$M + \overline{M}^{\text{tr}} = 0$	q^{m^2}
U^o	$u(z, M)$	$M + \overline{M}^{\text{tr}} + z^{\text{tr}}\overline{z} = 0$	$q^{m(m+2)}$
O⁺	$u(M)$	$M + M^{\text{tr}} = 0^{(\dagger)}$	$q^{m(m-1)/2}$
O^o	$u(z, M)$	$M + M^{\text{tr}} + z^{\text{tr}}z = 0$	$q^{m(m+1)/2}$
O⁻	$u(z_1, z_2, M)$	$M + M^{\text{tr}} + \alpha z_1^{\text{tr}}z_{\sigma(1)} + z_2^{\text{tr}}z_{\sigma(2)} = 0^{(\dagger)}$	$q^{m(m+3)/2}$

(\dagger) Additional constraints exist when $p = 2$ since elements of U also preserve φ . In case **O⁺** we have $M_{ii} = 0$ while in case **O⁻** we have $(M + z_2^{\text{tr}}z_1)_{ii} = (\alpha_1 z_1^{\text{tr}}z_1 + \alpha_2 z_2^{\text{tr}}z_2)_{ii}$ for $1 \leq i \leq m$, where α_1, α_2 are as in Lemma 2.1(i).

respectively, where $M \in \mathbb{M}_m(\mathbb{F})$, $z, z_1, z_2 \in \mathbb{F}^m$ and $\sigma \in \text{Sym}(2)$. The structure of U is summarised in Table 5 where, in case **O⁻**, $\sigma = (1, 2)$ if $p = 2$ and $\sigma = 1$ otherwise, and $\alpha = \rho$ if $q \equiv 1 \pmod{4}$ and $\alpha = 1$ otherwise.

In particular, notice that U is abelian only in cases **S**, **U^e** and **O⁺**. In the other cases U is a class 2 nilpotent group with $Z(U) = \{u(0, M)\}$ in cases **U^o** and **O^o** and $Z(U) = \{u(0, 0, M)\}$ in case **O⁻**.

2.6 Root elements and commutator relations

In cases **S** and **U** we have already seen that a long root subgroup of G is simply a transvection group $T(x)$ corresponding to a singular point x . In case **O**, a long root group corresponds to a t.s. line Σ in V , its elements inducing the identity on the $(d-2)$ -space Σ^\perp . Long root groups in case **O** have analogues in cases **S** and **U**, where they are called short root groups.

Let $0 \neq w \in \langle e_1, f_1 \rangle^\perp$ be a singular vector. A *long root group* of G in case **O**, or a *short root group* in cases **S** and **U**, is a G -conjugate of the group

$$R(e_1, w) = \{ r_1(\lambda w, 0) \mid \lambda \in \mathbb{F} \} \cong \mathbb{F}^+. \quad (9)$$

The following result, which is easily checked by direct computation, gives some useful commutator relations between elements from certain pairs of root groups.

Lemma 2.5 *Let G be a classical group and let $\alpha, \beta \in \mathbb{F}$. Then*

$$(i) \ [r_1(\alpha e_i, 0), r'_1(\beta e_j, 0)] = r_i(\alpha \beta e_j, 0) \in U(E) \text{ for } 1 \leq i < j \leq m.$$

Table 6: j is a $\text{ppd}^\#(p; n)$

$n = 1$	if p is a Fermat prime then $4 \mid j$; else j is not a power of 2.
$n \geq 2$	if $n = 6$ and $p = 2$ then $21 \mid j$; else if $n = 2$ and p is a Mersenne prime, then $4 \mid j$; else j is divisible by a ppd of $p^n - 1$.

(ii) $[r_1(\alpha f_i, 0), r'_1(\beta f_j, 0)] = r'_i(\alpha \beta f_j, 0) \in U(F)$ for $1 \leq i < j \leq m$.

(iii) $[r_1(\alpha e_i, 0), r'_1(\beta f_j, 0)] = r_i(\alpha \beta f_j, 0) \in L$ for $1 \leq i \neq j \leq m$.

2.7 Primitive prime divisors

By a fundamental theorem of Zsigmondy [Zs], if p is a prime and $n \geq 2$, then there is a prime dividing $p^n - 1$ but not $p^i - 1$ for $1 \leq i < n$, except when either $p = 2$, $n = 6$, or $n = 2$ and p is a Mersenne prime. Such a prime is called a *primitive prime divisor (ppd)* of $p^n - 1$. We define a $\text{ppd}^\#(p; n)$ to be an integer $j > 1$ behaving as in Table 6. We call an element g of a group G a $\text{ppd}^\#(p; n)$ -*element* if $|g|$ is a $\text{ppd}^\#(p; n)$. We also say that g is a $\text{ppd}^\#(p; n_1) \cdot \text{ppd}^\#(p; n_2)$ -*element* if $|g|$ is both a $\text{ppd}^\#(p; n_1)$ and a $\text{ppd}^\#(p; n_2)$.

Certain primitive prime divisor elements are highly abundant in classical groups and are useful because of their action on the natural module. The next result gives statistical information about such elements (see [KS], 4.1.5, 5.1.5 and 6.1.5 for cases **O**, **S** and **U** respectively).

Theorem 2.6 *Let $q = p^k \geq 16$, let $n \geq 3$ and let G be one of $\text{Sp}(n, q)$, $\text{SU}(n, q)$ (n odd), or $\Omega^-(n, q)$. Let V be the natural module of G over $\mathbb{F} = \text{GF}(p^l)$. Then G contains at least $\frac{|G|}{2n}$ (in cases **S** and **O**) and $\frac{|G|}{4n}$ (in case **U**) elements of $\text{ppd}^\#(p; ln)$ -order and each acts irreducibly on V .*

2.8 Probability estimates

We conclude section 2 by summarising some results that we will need for the correctness and reliability proofs of certain subroutines of our main algorithm. The first result concerns the natural module of a classical group.

Lemma 2.7 *Let V be the natural module of a classical group G of dimension d and let N denote the number of singular points of V .*

- (i) There are $\tilde{q}^{d-2}N/(q^\delta + 1)$ hyperbolic lines in V , where δ is 0 in case **O** and is 1 otherwise.
- (ii) With probability $> 1/4$, a randomly selected line in V is hyperbolic.
- (iii) If $G = \Omega^+(2m, q)$ then V has $q^{2m-2}(q^m - 1)(q^{m-1} - 1)/\{2(q + 1)\}$ definite lines.

Proof. (i) Fix a singular point x of V . Each of the \tilde{q}^{d-1} points not perpendicular to x determines a hyperbolic line containing x , each containing \tilde{q} points other than x . On the other hand, each hyperbolic line contains $q^\delta + 1$ singular points.

We prove (ii) only in the case where the lower bound is weakest, namely when $G = \Omega^-(2m + 2, q)$. By (i), since $N = (q^m - 1)(q^{m-1} + 1)$ in this case, the proportion of hyperbolic lines is

$$\begin{aligned} \frac{q^{d-2}(q^m-1)(q^{m+1}+1)(q^2-1)}{2(q^d-1)(q^{d-1}-1)} &> \frac{1}{2} \cdot \frac{q^{d-2}(q^{d+1}-q^{d-1}-q^{m+3}+q^{m+1})}{q^{2d-1}-q^d-q^{d-1}+1} \\ &> \frac{1}{2} \left\{ 1 - \frac{q^{2d-3}+q^{d+m+1}+q^{d+m-1}-2q^d}{q^{2d-1}-2q^d} \right\} > \frac{1}{4}. \end{aligned}$$

Finally, for (iii), subtract the number of lines of the form $\langle e, u \rangle$, where e is singular and $u \in e^\perp$ is nonsingular, and the number of hyperbolic and totally singular lines from the total number of lines in V . Those which remain are the definite lines. \square

The following result will enable us to package the most time consuming computations of our main algorithm inside a low dimensional subgroup (cf. 4.3.2).

Theorem 2.8 *Let G be a classical group with natural module V of dimension $d \geq 5$ with $q = p^k \geq 16$. Let $a \in G$ be an element of $\text{ppd}^\#(p; k)$ -order having 2-dimensional nonsingular support $[V, a]$. Let b be a random G -conjugate of a and let $W = [V, \langle a, b \rangle]$.*

- (i) With probability $\geq 1/640$, in case **O** or if q is even in case **S**, $\langle a, b \rangle$ induces $\Omega^+(4, q)$ on the nonsingular 4-space W and 1 on W^\perp .
- (ii) In case **S** (q odd), assuming also that a, b are $\text{ppd}^\#(p; k/2)$ -elements when k is even then, with probability $\geq 1/32$, $\langle a, b \rangle$ induces $\text{Sp}(4, q)$ on the nonsingular 4-space W and 1 on W^\perp .
- (iii) In case **U**, assuming also that a, b are $\text{ppd}^\#(p; k/2)$ -elements when k is even then, with probability $\geq 1/32$, $\langle a, b \rangle$ induces $\text{SU}(4, q)$ on the nonsingular 4-space W and 1 on W^\perp .

Proof. For (i) see [KS], Lemmas 4.12(i) and 5.10(i), for cases **O** and **S** respectively. For (ii) and (iii), as in [KS] Lemma 5.10(v), $\langle a, b \rangle$ is an irreducible subgroup of the stated group with probability $\geq 1/32$. We now refer to [KLi], Theorems 5.6 and 5.7, for a catalogue of subgroups of $\text{Sp}(4, q)$ and $\text{SU}(4, q)$ respectively. In each case, there are no proper irreducible subgroups generated by two elements of the stated order having 2-dimensional nonsingular support. \square

Finally, we state additional results that we will use in our treatment of low dimensional unitary groups. The first follows from [KS], Lemma 3.8(ii), in view of the isomorphism $SU(2, q) \cong SL(2, q)$.

Lemma 2.9 *Let $G = SU(2, p^k)$ with $p^k > 16$. Then two elements of the same $\text{ppd}^\#(p; 2k)$ -order generate G with probability > 0.55 .*

Lemma 2.10 *Let $G = SU(d, q)$ with $d \geq 4$ and $q \geq 8$, and let T be transvection subgroup of G . With probability $> 1/2$, the group T together with two G -conjugates T^{g_1} , T^{g_2} , generate a subgroup J of G inducing $SU(3, q)$ on the nonsingular 3-space $[V, J]$ and 1 on $[V, J]^\perp$.*

Proof. As in the proof of [KS], Lemma 3.7, J acts irreducibly on the nonsingular 3-space $[V, J]$ and is the identity on $[V, J]^\perp$ with probability at least $(1 - 1/q)^4 \geq (7/8)^4 > 1/2$. The result now follows from [KS], 6.1.4, by noting that there are no proper, irreducible, subgroups of $SU(3, q)$ generated by transvection groups. \square

3 Algorithmic Preliminaries

In this section we outline some elementary procedures which we will use as subroutines in our main algorithm.

3.1 Matrix groups

We first discuss some issues which arise when computing with matrix groups.

3.1.1 Straightline programs

It is important, when computing in a group $\langle \mathcal{S} \rangle$, to have an efficient means of recording how an element $g \in \langle \mathcal{S} \rangle$ is constructed from \mathcal{S} (a word in \mathcal{S} representing g could have exponential length). A *straightline program (SLP) of length m from \mathcal{S} to g* is a sequence (w_1, \dots, w_m) such that, for each i , either w_i is a symbol representing some element of \mathcal{S} , or $w_i = (j, -1)$ for $j < i$ (representing the inverse of w_j), or $w_i = (j, k)$ for $j, k < i$ (representing the product of w_j and w_k), such that if each expression w_i is evaluated sequentially in the obvious way, then the value of w_m is g .

One should think of each symbol w_i in such a sequence as being an actual group element. However, the rather abstract definition of SLP given above emphasises two things: that we do not store each element of the sequence as a matrix (we evaluate an SLP from a suitable set \mathcal{S} only when necessary); and also that SLPs can be written from one set and then evaluated from another (thus effecting an isomorphism, for example).

3.1.2 Random elements

We will assume that we can generate uniformly distributed random elements of a given matrix group G . A fundamental result of Babai [Ba] gives an algorithm, which runs in time bounded by a polynomial in $|\mathcal{S}|$, d , $\log q$ and μ , for computing sufficiently random elements of $G = \langle \mathcal{S} \rangle \leq \text{GL}(d, q)$ (see [KS], 2.2.2, for a statement and discussion of this result). A more practical, heuristic algorithm is given in [Ce+].

We introduce a parameter ξ in our complexity statements to ensure that the running time estimates can easily be adapted to different constructions of random elements in G . However, we **assume that** $\xi \geq \mu|\mathcal{S}|$ since it is presumed that each generator will be involved in the construction of a random element.

Reliability claims concerning subroutines which use random elements of a given group should take into account the fact that such elements are not taken from a perfectly uniform distribution. However, our probability estimates will be sufficiently crude so as not to be affected by small deviations from uniformity. Furthermore, since our algorithm is Las Vegas, it will return a correct output no matter what method of random generation is used, including less reliable methods such as [Ce+].

3.1.3 Element orders and ppds

Computing the exact order of an element $g \in G$ is unnecessary for our purposes, but we will need to detect properties of $|g|$ so that we can make deductions about the action of g on the natural module. In Theorem 2.6 we saw that certain primitive prime divisor elements occur with high frequency in classical groups. The next result shows that an efficient deterministic test can be applied to any given element of G to decide whether or not it has a specified ppd order.

Lemma 3.1 *For any prime p and positive integer n , following a one time integer computation taking time $O(n^3 \log n \log^4 p)$, one can test whether or not any given $g \in G$ is a $\text{ppd}^\#(p; n)$ -element in time $O(\mu n \log p)$.*

Proof. In [NeP] (p. 578), a deterministic $O(n^3 \log n \log^4 p)$ algorithm is presented which factors $p^n - 1 = P \cdot P'$, where P is the product of all primitive prime divisors of $p^n - 1$, including multiplicities. For a given integer i , we can compute g^i in time $O(\mu \log i)$ by repeated squaring. The result now follows by noting that $g \in G$ is a $\text{ppd}^\#(p; n)$ -element if $g^{p^n - 1} = 1$ but $g^{P'} \neq 1$. The test is easily modified to accommodate the more stringent definition given in 2.7. \square

3.1.4 Normal closures and derived subgroups

We will need a method for computing derived subgroups inside low dimensional classical groups. The following is a special case of [Se], Theorem 2.4.8.

Proposition 3.2 *There is a Monte Carlo $O(\mu\{d^2 \log^4 d \log^2 q + |\mathcal{S}| \log^4 d \log q\})$ -time algorithm to construct a generating set of size $O(d^2 \log q)$ for the derived subgroup G' of G .*

Remark: Proposition 3.2 will only be applied when $d = 3$ or 4 in case **U**. Hence all occurrences of d in the timing will disappear.

We will also employ a deterministic method for finding normal closures in elementary abelian sections of a matrix group based on the following elementary fact (which follows from [KS], Lemma 2.7).

Lemma 3.3 *Let Y be a $\text{GF}(p^l)$ -space, let $\sigma \in \text{GL}(Y)$ be a $\text{ppd}^\#(p; nl)$ -element ($n < \dim(Y)$), acting irreducibly on the n -space $[Y, \sigma]$ and as the identity on a complementary subspace. For $X \leq Y$, let $X_\sigma = \langle X\sigma^i \mid 0 \leq i < n \rangle \leq Y$. Then the following hold:*

(i) $X_\sigma = \langle X^{\langle \sigma \rangle} \rangle$.

(ii) If X lies in no hyperplane of Y containing $[Y, \sigma]$, then $X_\sigma = Y$.

3.2 The natural module

We next outline the methods which we will use to compute with the natural module V of our given classical group G . We assume that we have the matrix representing a G -invariant classical form relative to a fixed basis of V (note $O(d^2)$ field operations are required for each evaluation).

3.2.1 Quadratic equations

Several methods are discussed in [LN] for finding roots of polynomials over finite fields; we will only need the quadratic case. The timing stated in the next result arises from the $O(l^2)$ field operations involved in various gcd calculations when we specialise to the quadratic case.

Lemma 3.4 *There is a Las Vegas $O(l^3 \log p)$ -time algorithm which, for any given $f(x) = x^2 + \alpha x + \beta \in \mathbb{F}[x]$ having roots in \mathbb{F} , with probability $> 1 - 1/16$ finds $\gamma_1, \gamma_2 \in \mathbb{F}$ such that $f(x) = (x - \gamma_1)(x - \gamma_2)$.*

Remark: The algorithm fails for a reducible input $f(x)$, with probability $< 1/16$, by incorrectly reporting irreducibility. However, if $f(x)$ happens to be irreducible in $\mathbb{F}[x]$, then the algorithm always detects this.

3.2.2 Traces and norms

Suppose that we are in case **U** so that l is even, $q = p^{l/2}$ and $\lambda \mapsto \bar{\lambda} = \lambda^q$ is an involutory automorphism of \mathbb{F} with fixed field $\mathbb{F}_0 = \text{GF}(q)$.

Lemma 3.5 (i) *There is a deterministic $O(l^2 \log q)$ -time algorithm which, for any given $\beta \in \mathbb{F}_0$, finds $\alpha \in \mathbb{F}_0^*$ such that $\alpha + \bar{\alpha} = \beta$.*

(ii) *There is a Las Vegas $O(l^2 \log^2 q \log \log q)$ -time algorithm which, for any given $\beta \in \mathbb{F}_0^*$, with probability $> 1 - 1/2^6$ finds $\alpha \in \mathbb{F}^*$ such that $\alpha \bar{\alpha} = \beta$.*

Proof. (i) This just involves solving a system of $l \times l$ equations over \mathbb{F}_p .

(ii) Use Lemma 3.4 at most twice to find a root $\gamma \in \mathbb{F}$ of $x^2 - \beta$ with probability $> 1 - 1/2^8$. If $\gamma \in \mathbb{F}_0$ or $\gamma \bar{\gamma} = \beta$ then return $\alpha = \gamma$. Otherwise, $\gamma \bar{\gamma} = -\beta$ and we return $\alpha = \gamma \delta$ for $\delta \in \mathbb{F}^*$ such that $\delta \bar{\delta} = -1$ found as follows. If $q \equiv 1 \pmod{4}$ then use Lemma 3.4 to find a root $\delta \in \mathbb{F}_0$ of $x^2 + 1$. If $q \equiv 3 \pmod{4}$ then find the largest integer i such that $2^i | q + 1$. Use i iterations of Lemma 3.4 (at most $2 \lceil \log i \rceil$ times for each iteration) to find a root $\delta \in \mathbb{F}$ of $x^{2^i} + 1$. Since $(q + 1)/2^i$ is odd, it follows that $\delta \bar{\delta} = \delta^{q+1} = \delta^{2^i} = -1$. The stated timing is for the $O(i \log q) = O(\log q \log \log q)$ uses of Lemma 3.4. For a fixed i , all $2 \lceil \log i \rceil$ applications of Lemma 3.4 fail with probability $< 1/i^8$. Hence, the probability that at least one iteration fails is $< 1/i^7 \leq 1/2^7$. Hence, the procedure finds α with probability $> 1 - (1/2^8 + 1/2^7) > 1 - 1/2^6$. \square

3.2.3 Standard bases

Finally we summarise some algorithmic techniques involved in finding a standard basis for the natural module of a classical group.

Lemma 3.6 *Let U be a nonsingular subspace of V of dimension r . Then in $O(rd^3 \log q)$ -time one can find the orthogonal complement U^\perp to U in V .*

Proof. Find the nullspace of the $d \times r$ matrix $[[v_i, u_j]]$ over $\text{GF}(q)$, where v_1, \dots, v_d is a basis for V and u_1, \dots, u_r is a basis for U . The stated timing is that required to obtain $[[v_i, u_j]]$ using the classical form. \square

Lemma 3.7 *There is a Las Vegas $O(d^2 \log q + l^2 \log^2 q \log \log q)$ -time algorithm which, for a given 2-space L of V does one of the following:*

- *if L is hyperbolic, returns a hyperbolic pair $e, f \in V$ of singular vectors such that $(e, f) = 1$ (with probability $> 1 - 1/16$) or incorrectly reports “not hyperbolic” (with probability $< 1/16$);*

- if L is not hyperbolic, detects that this is the case.

Proof. Let $L = \langle v, w \rangle$ be the given line. First consider case **S**. We may assume that $(v, w) \neq 0$ since otherwise L is t.s. (in which case a report of “not hyperbolic” is returned). Output $e := v$ and $f := w/(v, w)$.

Next consider case **U**, and suppose first that v is singular. Again we may assume that $(v, w) \neq 0$ since otherwise L is not hyperbolic. Set $e := v$; replace w with $w/(v, w)$; use Lemma 3.5(i) to find $\alpha \in \mathbb{F}$ such that $\alpha + \bar{\alpha} = -(w, w)$; and set $f := \alpha v + w$. If v is not singular, we replace it with a vector that is as follows. Replace w with any vector in $v^\perp \leq L$ (where v^\perp is found using Lemma 3.6). If w is singular, report that L is not hyperbolic. Otherwise use Lemma 3.5(ii) to find $\alpha \in \mathbb{F}$ such that $\alpha \bar{\alpha} = -(v, v)/(w, w)$ and replace v with $\alpha v + w$.

Finally suppose that we are in case **O**. Unless $\varphi(v) = (v, w) = 0$ or $\varphi(w) = (v, w) = 0$ (whence L is not hyperbolic), use Lemma 3.4 to find the roots $\gamma_1, \gamma_2 \in \mathbb{F}$ of the quadratic $\varphi(v) + (v, w)x + \varphi(w)x^2$. If we do not obtain distinct $\gamma_1, \gamma_2 \in \mathbb{F}$, report that L is not hyperbolic. Otherwise set $e := v + \gamma_1 w$ and $f := (v + \gamma_2 w)/(e, v + \gamma_2 w)$.

It is easy to verify in each case that the pair e, f behaves as stated. The timing is dominated by the use of Lemma 3.5(ii) and 3.6, while the stated reliability is the probability that Lemma 3.5(ii) (case **U**) or Lemma 3.4 (case **O**) succeeds. \square

Lemma 3.8 *There is a Las Vegas $O(d \log d \{d^3 \log q + l^2 \log^2 q \log \log q\})$ -time algorithm which, with probability $> 3/4$, finds a standard basis of the natural module V .*

Proof. The procedure is recursive. If $V = \langle v \rangle$ is a point (hence we are not in case **S**), find a scalar $\alpha \in \mathbb{F}^*$ as follows: in case **O**, replacing φ with $\rho\varphi$ if $\varphi(v)^{(q-1)/2} \neq 1$, use Lemma 3.4 to find a root α of $x^2 - \varphi(v)^{-1}$; and, in case **U**, use Lemma 3.5(ii) to find α such that $\alpha \bar{\alpha} = -1/(v, v)$. In each case return αv (note $(\alpha v, \alpha v) = 1$ in case **U**, and $\varphi(\alpha v) = 1$ in case **O**).

If $L = \langle v, w \rangle$ is a line then use Lemma 3.7 to find a hyperbolic pair, or else find that L is probably not hyperbolic. In the former case, we return the pair. In the latter we may assume that we are in case **O**⁻. Replacing φ with $\rho\varphi$ if necessary, use Lemmas 3.4 and 3.6 to find and return vectors v_1 and v_2 behaving as in Lemma 2.1.

Suppose then that $\dim(V) > 2$. Choose up to $6 \lceil \log(2d) \rceil$ lines of V , using Lemma 3.7 on each, to find a hyperbolic line L and hyperbolic pair e, f in L . Use Lemma 3.6 to find the $d-2$ space L^\perp (the natural module for a classical group of the same type) and use recursion to find a standard basis \mathcal{B}' for L^\perp . Insert the vectors e, f into \mathcal{B}' to obtain a standard basis \mathcal{B} for V and return \mathcal{B} .

The stated timing arises from the $O(d \log d)$ uses of Lemma 3.7. A line in V is hyperbolic with probability $> 1/4$ by Lemma 2.7(ii). Hence, if $d \geq 3$, the probability of failure before the

recursive call is $< 1/(4d^2)$. The entire algorithm therefore fails with probability $\leq \sum_3^d (1/4i^2) < 1/4$. \square

4 The Main Algorithm

Let $G = \langle \mathcal{S} \rangle$ be a classical group in its natural representation as a subgroup of $\text{SL}(V)$ with V of dimension d over the field $\mathbb{F} = \text{GF}(p^l)$. In this section, we present an algorithm to construct a new generating set \mathcal{T} for G using SLPs from \mathcal{S} . In section 5 we will complete the proof of Theorem 1.1 by giving a routine to write an SLP from \mathcal{T} to any given element $g \in G$.

Throughout the entire algorithm, let k, l, q be as defined at the beginning of section 2. Fix a generator ρ of \mathbb{F}^* . Also, in case \mathbf{U} , fix $\zeta = \rho^{q+1}$ (a generator of \mathbb{F}_0^*) and $0 \neq \delta = -\bar{\delta}$ (found using Lemma 3.5(i)). Throughout section 4, we assume that $d \geq 5$ unless stated otherwise.

4.1 Overview of the algorithm

We first give a pseudo-code overview to provide the reader with a reference to the structure of the algorithm. Appropriate references are listed to the right of the principal subroutines.

`ClassicalConstructiveRecognition(G)`

`$M := \text{ClassicalForm}(G);$` [4.2]

`$(\tau , a) := \text{FindGoodElements}(G);$` ^(†) [4.3.1]

`$J := \text{ConstructNaturalSubgroup}(G , a);$` ^(†) [4.3.2]

`$\mathcal{T}_Q := \text{ConstructQ}(J , \tau);$` [4.4]

`$\Delta := \text{ConstructDelta}(\mathcal{T}_Q);$` ^(††) [4.5]

`$\mathcal{T} := \text{ConstructNewGenerators}(\Delta);$` [4.6]

Return \mathcal{T} .

Remarks:

(†) These subroutines are both Las Vegas algorithms; failure may reported at either stage. In fact, failure could be reported for two, very different, reasons: either G is not a classical group on V ; or it is and bad luck occurred with choices of group elements. We note, however, that `ClassicalForm(G)` will already have reported failure if G does not preserve an appropriate nondegenerate form on V . Hence failure for the first reason will only occur when G is a proper subgroup of the corresponding classical group on V .

(††) Failure will be reported by `ConstructDelta` exactly when the input set \mathcal{T}_Q does not generate a certain, desired subgroup of G . This may occur in certain of the cases that we consider, but always with low probability.

Small fields: In order to present an algorithm which has a reasonably uniform appearance, it is necessary, at various stages, to avoid pathologies which occur with small fields. However, beyond a particular stage, our algorithm handles all field sizes equally well. Hence, we will assume throughout 4.3 and 4.4 that $q \geq 16$; replacements for the routines contained therein will be discussed in 4.7 for $q < 16$. Throughout 4.5 and 4.6, we will make no assumption about q .

4.2 A G -invariant form

Several methods exist to determine the matrix representing a nondegenerate alternating, symmetric, or hermitian form left invariant by a given matrix group $G \leq \text{GL}(V)$ on its underlying module V . Perhaps the best known of these is the generalisation of the Parker-Norton ‘Meat-Axe’ algorithm by Holt and Rees [HR]. In case \mathbf{O} , one easily obtains a G -invariant quadratic form with given associated G -invariant symmetric form. Hence, we assume the availability of an efficient algorithm:

`ClassicalForm(G)`

- Given $G = \langle S \rangle \leq \text{GL}(V)$, find a (nondegenerate) G -invariant, classical form on V if such exists.

4.3 The subgroup J

Entirely different methods are required when $d \leq 4$ and we defer further discussion of the various low dimensional groups until section 6. The algorithm `ClassicalConstructiveRecognition` for the general case ($d \geq 5$) requires that we first construct, and then constructively recognise, a suitable low dimensional subgroup J . Hence we will soon make calls to certain of the low dimensional algorithms presented in section 6.

We will construct J in two stages. In the first stage, we will search for a frequently occurring element τ which decomposes V , in a prescribed way, as the sum of τ -invariant subspaces. In the second stage we will construct an element a (using τ) such that $[V, a]$ is a hyperbolic line. With high probability, J is generated by a together with a single random conjugate of a . Recall that $q \geq 16$ for the remainder of 4.3.

4.3.1 The elements τ and a

Let n be the integer defined in Table 7 and let $z = nl/2k$ (since n is odd only in case \mathbf{U} , where $l = 2k$, z is an integer). We now describe a subroutine which constructs, with high probability, elements $\tau, a \in G$ with the following properties: τ has $\text{ppd}^\#(p; nl)$ -order (or $\text{ppd}^\#(p; 2k) \cdot \text{ppd}^\#(p; nk)$ -order in case \mathbf{O}^+), such that τ^{q^2-1} centralises a hyperbolic line of V ; and $a :=$

Table 7: The integer n

Case	n	Case	n
S	$d - 2$	O ⁺	$d - 4$
U ^e	$d - 3$	O ^o	$d - 3$
U ^o	$d - 2$	O ⁻	$d - 2$

$\tau^{(q+1)(q^z+1)}$ has $\text{ppd}^\#(p; k)$ -order (or $\text{ppd}^\#(p; k) \cdot \text{ppd}^\#(p; k/2)$ -order if k is even in cases **U** and **S**) whose support is a hyperbolic line centralised by τ^{q^2-1} .

The procedure is given primarily for case **O**⁺, with the necessary modifications for the other cases given in parentheses. Also, we will prove correctness only in case **O**⁺; the other cases are similar, but easier.

FindGoodElements(G)

For up to $96n$ choices $\tau \in G$

if ($|\tau| = \text{ppd}^\#(p; nk)$ and $|\tau| = \text{ppd}^\#(p; 2k)$) then

[omit the $|\tau| = \text{ppd}^\#(p; 2k)$ test for all other cases]

$a := \tau^{(q+1)(q^{n/2}+1)}$;

if ($|a| = \text{ppd}^\#(p; k)$ and $\dim[V, a] = 2$) then

*[test also $|a| = \text{ppd}^\#(p; k/2)$ if k is even in cases **S** and **U**]*

Return (τ , a).

Correctness: Since $(q^{n/2} + 1, q - 1) = (q + 1, q - 1) \leq 2$, if a pair (τ, a) is returned by the procedure, then τ has $\text{ppd}^\#(p; k) \cdot \text{ppd}^\#(p; 2k) \cdot \text{ppd}^\#(p; nk)$ -order. Furthermore, by [NiP], Lemma 5.1, any element of order $|\tau|$ splits V as a perpendicular direct sum $V = V_4^- \perp V_{d-4}^-$, where V_i^ϵ denotes a nonsingular i -space of type **O** ^{ϵ} . In addition, since a has 2-dimensional support, τ preserves $V = V_2^+ \perp V_2^- \perp V_{d-4}^-$, where a centralises the $d - 2$ -space $V_2^- \perp V_{d-4}^-$ and τ^{q^2-1} centralises V_2 . Hence, any pair (τ, a) returned by the procedure behaves as stated.

Conversely, any element τ having $\text{ppd}^\#(p; k) \cdot \text{ppd}^\#(p; 2k) \cdot \text{ppd}^\#(p; nk)$ -order which preserves a decomposition $V = V_2^+ \perp V_2^- \perp V_{d-4}^-$ will be returned by the algorithm provided that it is divisible also by 16 whenever p is Mersenne and $k = 2$, and whenever p is Fermat and $k = 1$ (cf. Table 6). We claim that there are at least $|G|/32n$ such elements of G . In fact, since our estimates are crude, we need not concern ourselves with the additional divisibility requirement on $|\tau|$ in the Mersenne and Fermat cases. Assuming our claim is true, the procedure will fail to find a pair (τ, a) with probability $\leq \{(1 - 1/32n)^{32n}\}^3 < 1/e^3$.

We first count the number of such elements preserving a fixed decomposition $V = V_2^+ \perp V_2^- \perp$

Table 8: The subgroup J

Case	$J \cong$
\mathbf{S} (q even) and \mathbf{O}	$\Omega^+(4, q)$
\mathbf{S} (q odd)	$\mathrm{Sp}(4, q)$
\mathbf{U}	$\mathrm{SU}(4, q)$

V_{d-4}^- . Recall that $\mathrm{O}^\pm(2, q)$ is dihedral of order $2(q \mp 1)$. There are at least $(q+1)/2$ elements of $D_{2(q+1)}$ of $\mathrm{ppd}^\#(p; 2k)$ -order and at least $(q-1)/2$ elements of $D_{2(q-1)}$ of $\mathrm{ppd}^\#(p; k)$ -order. Hence, by Theorem 2.6, there are at least

$$\frac{q+1}{2} \cdot \frac{q-1}{2} \cdot \frac{|\Omega^-(d-4, q)|}{2(d-4)} = \frac{(q^2-1)|\Omega^-(d-4, q)|}{8(d-4)} \quad (10)$$

suitable elements preserving our fixed decomposition. Next, by Lemma 2.7(i) and (iii), there are exactly

$$\left\{ \frac{q^{2m-4}(q^{m-1}-1)(q^{m-2}-1)}{2(q+1)} \right\} \cdot \left\{ \frac{q^{d-2}(q^m-1)(q^{m-1}+1)}{2(q-1)} \right\} \quad (11)$$

such decompositions of V (where $d = 2m$). Hence, multiplying together (10) and (11), we see that there are at least $|\Omega^+(d, q)|/\{32(d-4)\} = |G|/32n$ suitable elements in G , as claimed.

Timing: Constructing $O(d)$ random elements of G and using Lemma 3.1 up to 3 times on each to test the appropriate ppd property costs $O(d^3 \log d \log^4 q + d\{\xi + \mu d \log q\})$.

Reliability: $1 - 1/e^3$.

4.3.2 Constructing J

In 4.3.1 we constructed an element a of $\mathrm{ppd}^\#(p; k)$ - or $\mathrm{ppd}^\#(p; k/2) \cdot \mathrm{ppd}^\#(p; k)$ -order. We now present an algorithm to construct a naturally embedded 4-dimensional subgroup J of G behaving as in Table 8.

`ConstructNaturalSubgroup(G , a)`

For up $3 \cdot 2^{10}$ conjugates $b = a^g$, proceed as follows:

$J := \langle a, b \rangle$; $V_J := [V, J]$;

$[J \text{ is } 1 \text{ on } V_J^\perp]$

if (V_J is a nonsingular 4-space) then

$K :=$ the 4×4 matrix group induced by J on V_J ;

$[K \cong J]$

$\mathcal{T}_K := \text{ClassicalConstructiveRecognition}(K)^{(*)}$

if (K is the group defined in Table 8) then

Return J , K , \mathcal{T}_K .

Using the appropriate low dimensional routine from section 6.

Correctness: Since J is 1 on $[V, J]^\perp$, it follows that K embeds naturally in G as the subgroup J .

Reliability: Since a is a $\text{ppd}^\#(p; k)$ - or $\text{ppd}^\#(p; k/2) \cdot \text{ppd}^\#(p; k)$ -element having 2-dimensional support, it follows from Theorem 2.8 that, with probability $\geq 1/640$, J induces the desired subgroup on the nonsingular 4-space V_J . For such a subgroup J , the procedure

`ClassicalConstructiveRecognition(K)`

succeeds with probability at least $3/4$ so that a single conjugate b produces a constructively recognised K with probability at least $(3/4) \cdot (1/640) > 1/2^{10}$. Hence, at least one of our choices succeeds with probability $\geq 1 - [(1 - 1/2^{10})^{2^{10}}]^3 > 1 - 1/e^3$.

Timing: $O(\xi \log \log q + \log^2 q + \chi)$, dominated by the cost of recognising K in case \mathbf{U} (cf. 6.4.8).

4.3.3 A standard basis for V

The 4-space V_J is the natural module for J . Observe that when q is even in case \mathbf{S} , V_J is the natural module for both $\text{Sp}(4, q)$ and $\Omega^+(4, q)$. Although J is the latter group, we view V_J as a subspace of the symplectic module V (i.e. we ignore the quadratic form on V_J constructed in 4.3.2).

The element a preserves the decomposition $V = [V, a] \perp [V, a]^\perp$ and fixes exactly two singular points $x = \langle e \rangle$ and $y = \langle f \rangle$ of the hyperbolic line $[V, a]$. The following procedure finds a standard basis \mathcal{B} for V , containing vectors spanning x and y , such that $\mathcal{B} \cap V_J$ is a standard basis for V_J .

`StandardBasis(V , V_J , e , f)`

$e_1 := e$; $f_1 := f / (e, f)$;

[e_1, f_1 is a hyperbolic pair]

Use Lemma 3.6 to find $V_J \cap [V, a]^\perp$;

Use Lemma 3.7 to find a hyperbolic pair e_2, f_2 in this line;

Use Lemma 3.6 again to find $V_J^\perp \leq V$ and use Lemma 3.8 to find a standard basis \mathcal{B}' of V_J^\perp .

Insert e_1, e_2, f_1, f_2 in the appropriate positions of \mathcal{B}' to obtain a standard basis \mathcal{B} of V .

Return \mathcal{B} .

Timing and reliability: Lemma 3.8 succeeds with probability $> 3/4$ in time $O(d \log d \{d^3 \log q + l^2 \log^2 q \log \log q\})$.

4.3.4 Changing basis

For the remainder of the algorithm, we will write matrices of G relative to the basis \mathcal{B} of V constructed in 4.3.3. We now have a convenient embedding of K into J (the groups returned by `ConstructNaturalSubgroup`) sending

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \mapsto \begin{pmatrix} A & 0 & B & 0 \\ 0 & I_{d-m-2} & 0 & 0 \\ C & 0 & D & 0 \\ 0 & 0 & 0 & I_{m-2} \end{pmatrix}, \quad (12)$$

where A, B, C, D are 2×2 matrices satisfying the set of constraints for case \mathbf{S}, \mathbf{U}^e or \mathbf{O}^+ in Table 1, and I_j is the $j \times j$ identity matrix for $j = m - 2, d - m - 2$.

Let \mathcal{T}_J denote the image of \mathcal{T}_K under this embedding.

Timing: $O(\mu(|\mathcal{S}| + \log q))$ to apply the change of basis matrix to each generator of G and to each of our constructed elements.

4.4 The subgroups $Q(x)$ and $Q(y)$

The ability to construct and manipulate the subgroups $Q(x)$ and $Q(y)$ (described in 2.4) is the key to our construction of the new generating set \mathcal{T} for G . Henceforth, it will be understood that (analogues of) all subroutines pertaining to $Q(x)$ are repeated for $Q(y)$ and we make no further mention of the latter. Denoting $Q(x)$ simply by Q , our first goal is to construct a generating set \mathcal{T}_Q for Q .

In cases \mathbf{S} and \mathbf{U} , let T denote the transvection group $T(x)$ and let t_1, \dots, t_k be generators for T defined as follows:

$$t_j := \begin{cases} r_1(0, \rho^{j-1}) & \text{in case } \mathbf{S} \\ r_1(0, \zeta^{j-1}\delta) & \text{in case } \mathbf{U} \end{cases} \quad \text{for } 1 \leq j \leq k. \quad (13)$$

4.4.1 Constructing $Q \cap J$

The following procedure returns a generating set for the subgroup $X = Q \cap J$.

`ConstructSubgroupOfQ(J)`

Initialise $\mathcal{T}_X := \emptyset$.

for $i \in \{1, \dots, l\}$ do

`WriteSLP($r_1(\rho^i e_2, 0)$, J).`

`WriteSLP($r_1(\rho^i f_2, 0)$, J).`

[section 6]

```

 $\mathcal{T}_X := \mathcal{T}_X \cup \{ r_1(\rho^i e_2, 0), r_1(\rho^i f_2, 0) \}.$ 
if ( $X$  is nonabelian) then
  for  $j \in \{1, \dots, k\}$ 
    writeSLP(  $t_j, \mathcal{T}_J$  ). [section 6]
     $\mathcal{T}_X := \mathcal{T}_X \cup \{ t_j \}.$ 
Return  $\mathcal{T}_X.$ 

```

Correctness: Q consists of all elements of G which induce 1 on x and on x^\perp/x . Since J centralises the $(d-4)$ -space V_J^\perp , it follows that $O_p(J_x) \leq Q$. Hence $Q \cap J$ is generated by \mathcal{T}_X .

Timing: $O(\log q(\chi + \log q))$ to write the $\leq 5k$ SLPs.

Remark: When q is even in case **S**, $X = O_p(\Omega^+(V_J)_x)$ of order q^2 , a proper subgroup of the more desirable group $O_p(\text{Sp}(V_J)_x)$ of order q^3 ; we have not yet explicitly constructed the transvection group T at this stage. We will soon be able to construct T even in this case, albeit indirectly.

4.4.2 Constructing Q

We now give the main procedure **ConstructQ** which returns a generating set \mathcal{T}_Q for a subgroup of Q . In cases **S** (q odd), **U**^o and **O**⁻, we will always have $\langle \mathcal{T}_Q \rangle = Q$. In the other cases it could happen (with probability $< 1/8$) that $\langle \mathcal{T}_Q \rangle$ is a proper subgroup of Q . There is, however, no randomised component to the following procedure; the success, or otherwise, in generating Q is determined by previous constructions and will be established later in 4.5.

```

ConstructQ(  $J, \tau$  )

```

```

   $\sigma := \tau^{q^2-1};$ 

```

```

   $\mathcal{T}_X := \text{ConstructSubgroupOfQ}( J );$ 

```

```

  Return

```

$$\mathcal{T}_Q := \bigcup_{i=0}^{d-2} (\mathcal{T}_X)^{\sigma^i}. \quad (14)$$

Correctness: Recall that $\langle e_1, f_1 \rangle = [V, a]$ so $\sigma = \tau^{q^2-1}$ centralises the 1-space $x = \langle e_1 \rangle$. Hence, σ normalises $Q(x)$ by Theorem 2.3(ii) so that $U = \langle \mathcal{T}_Q \rangle \leq Q$ as claimed. Note also that U is the normal closure $\langle X^{\langle \sigma \rangle} \rangle$; this follows from Lemma 3.3(ii).

Reliability: For any subgroup Y of Q , denote by \tilde{Y} the quotient group Y/T if Q is nonabelian or Y if Q is abelian. Since T is the Frattini subgroup of Q in the nonabelian cases, it suffices to show that $\tilde{U} = \tilde{Q}$ with high probability.

Recall the integer n defined in Table 7. In each of the 6 cases, σ acts on the module \tilde{Q} , is irreducible on the nonsingular n -space $[\tilde{Q}, \sigma]$, and is the identity on the orthogonal complement $[\tilde{Q}, \sigma]^\perp$ of $[\tilde{Q}, \sigma]$ in \tilde{Q} . By Lemma 3.3(ii), it suffices to show that, with high probability, the hyperbolic line \tilde{X} is in no hyperplane of \tilde{Q} containing $[\tilde{Q}, \sigma]$. Observe that \tilde{X} was determined by the random conjugate $b = a^g$ (cf. 4.3.2) and is independent of the n -space $[\tilde{Q}, \sigma]$.

Cases **S** (q odd), **U**^o and **O**⁻: Here $n = d - 2$, σ is irreducible on \tilde{Q} , so there are no such hyperplanes of \tilde{Q} . The procedure is deterministic for these cases.

Cases **U**^e and **O**^o: Here we compute the probability that the line \tilde{X} lies inside the nonsingular hyperplane $[\tilde{Q}, \sigma]$. In case **U** the probability is $\{(q^2)^{d-5}(q^{d-4} - 1)(q^{d-3} + 1)\} / \{(q^2)^{d-4}(q^{d-2} - 1)(q^{d-3} + 1)\} < 1/q^4$. In case **O**^o, the probability is $\{q^{d-5}(q^{(d-5)/2} - 1)(q^{(d-3)/2} + 1)\} / \{q^{d-4}(q^{(d-3)/2} - 1)(q^{(d-3)/2} + 1)\} < 1/q^2$.

Case **S** (q even): Here \tilde{X} is a line in the $d - 1$ -dimensional orthogonal space \tilde{Q} . Hence the probability that \tilde{X} lies in the hyperplane $[\tilde{Q}, \sigma]$ is $\{q^{d-4}(q^{(d-4)/2} - 1)(q^{(d-2)/2} + 1)\} / \{q^{d-3}(q^{(d-2)/2} - 1)(q^{(d-2)/2} + 1)\} < 1/q^2$.

Case **O**⁺: Here $n = d - 4$ and there are $q + 1$ hyperplanes containing the n -space $[\tilde{Q}, \sigma]$. By a counting argument similar to the one above, the probability that \tilde{X} is in one of these $q + 1$ hyperplanes is $< (q + 1)/q^2 < 2/q < 1/8$.

Hence, in all cases, $U = Q$ with probability $> 7/8$.

Timing: $O(\mu \log d \log^2 q)$ to construct \mathcal{T}_Q using SLPs from $\mathcal{T}_X \cup \{\sigma\}$.

4.5 The generating sets $\Delta(x)$ and $\Delta(y)$

This is the point at which our differing methods for $q \geq 16$ and $q < 16$ converge; the remaining subroutines handle all field sizes uniformly. At the current stage of the algorithm we have constructed the following (cf. 4.7 when $q < 16$):

- a standard basis \mathcal{B} of V relative to which all matrices are written;
- probable generating sets for the subgroups $Q = Q(x)$ and $Q(y)$; and
- generating sets for $T(x) < Q(x)$ and $T(y)$ whenever $Q(x)$ is nonabelian.

We will now define the elements of standard generating sets $\Delta(x)$ for $Q(x)$ and $\Delta(y)$ for $Q(y)$. As in 4.4, we will only discuss the group $Q = Q(x)$ and therefore denote $\Delta(x)$ simply by Δ . We will give a procedure **ConstructDelta** which constructs Δ using SLPs from the set \mathcal{T}_Q returned by **ConstructQ**. The procedure **ConstructDelta** is deterministic in the sense that it will return the desired set Δ if $\langle \mathcal{T}_Q \rangle = Q$, and fail otherwise. This important property of **ConstructDelta** notwithstanding, the real purpose of the sets $\Delta(x)$ and $\Delta(y)$ is to enable us to construct the generating set \mathcal{T} of Theorem 1.1.

4.5.1 Linear algebra in \tilde{Q}

For computational purposes, we wish to regard the group \tilde{Q} as an \mathbb{F}_p -space relative to a suitable basis (denoted $\tilde{\Delta}$). More precisely, we wish to write

$$\tilde{Q} = \prod_{v \in \tilde{\Delta}} \langle v \rangle$$

such that the decomposition of any given $u \in \tilde{Q}$, as an \mathbb{F}_p -vector relative to $\tilde{\Delta}$, can efficiently be found. We may then store generators for \tilde{Q} as \mathbb{F}_p -vectors and also compute in \tilde{Q} as efficiently as in the row space $\mathbb{F}_p^{l(d-2)}$.

Let $\mathcal{B}' = \mathcal{B} \cap \langle e_1, f_1 \rangle^\perp$ (a standard basis for $\langle e_1, f_1 \rangle^\perp$) and, for our fixed generator ρ of \mathbb{F}^* , let \mathcal{B}'_ρ denote the \mathbb{F}_p -basis obtained from \mathcal{B}' and ρ . Let $\psi: \langle x, y \rangle^\perp \rightarrow Q$ be the function defined, for $w \in \langle x, y \rangle^\perp$, by

$$w\psi = \begin{cases} r_1(w, 0) & \text{in case } \mathbf{S} \\ r_1(w, \gamma(w, w)) & \text{in case } \mathbf{U} \\ r_1(w, \varphi(w)) & \text{in case } \mathbf{O} \end{cases},$$

for fixed $\gamma = 1 - \bar{\gamma}$, found using Lemma 3.5(i). If we are not in case \mathbf{S} (q even), then ψ is linear and, by Theorem 2.4, we have

$$Q = \langle w\psi \mid w \in \langle x, y \rangle^\perp \rangle.$$

In case \mathbf{S} (q even) ψ is not linear, but still Q (of order q^{d-1}) is the direct product

$$Q = T \times \langle w\psi \mid w \in \langle x, y \rangle^\perp \rangle.$$

With this in mind, we define a generating set Δ' for Q as follows:

$$\Delta' := \begin{cases} \{t_1, \dots, t_k\} \cup \{w\psi \mid w \in \mathcal{B}'_\rho\} & \text{in case } \mathbf{S} \text{ (} q \text{ even)} \\ \{w\psi \mid w \in \mathcal{B}'_\rho\} & \text{otherwise} \end{cases} \quad (15)$$

In fact, if $\tilde{\Delta}$ denotes Δ' when Q is abelian and $\Delta'T$ otherwise, then $\tilde{\Delta}$ is an \mathbb{F}_p -basis for \tilde{Q} , inheriting its ordering from \mathcal{B}'_ρ (preceded by t_1, \dots, t_k when q is even in case \mathbf{S}).

The important point here is that, since elements of Q are written relative to our standard basis \mathcal{B} , we can simply “read off” the coordinates of a given $\tilde{u} \in \tilde{Q}$ relative to $\tilde{\Delta}$. For, if $u = r_1(w, \lambda) \in Q$, then $f_1 u = u \pm \lambda e_1 \pm w$ is the row of the matrix u corresponding to the basis vector f_1 . Hence w appears in the matrix u as an \mathbb{F} -linear combination of \mathcal{B}' . Replacing each coordinate with its \mathbb{F}_p -vector relative to the basis $1, \dots, \rho^{l-1}$ of \mathbb{F} gives w as an \mathbb{F}_p -vector relative to \mathcal{B}' so that

$$\text{elements of } \tilde{Q} \text{ are “given” as } \mathbb{F}_p\text{-vectors relative to } \tilde{\Delta}. \quad (16)$$

4.5.2 Constructing Δ' and Δ

Equation (15) defines the generating set Δ' , but we have not yet *constructed* its elements using SLPs from previously constructed elements. In particular, when q is even in case **S**, we have not yet constructed t_1, \dots, t_k . The next routine writes an SLP of length $O(d \log q)$ from \mathcal{T}_Q to each of the $< l(d-1)$ elements of Δ' (let \tilde{T} denote \mathcal{T}_Q or $\mathcal{T}_Q T$ according as Q is abelian or nonabelian, respectively).

`ConstructDeltaPrime(\mathcal{T}_Q)`

```

if ( $\dim\langle\tilde{T}\rangle < \dim\tilde{Q}$ ) then                                     [ $\langle\mathcal{T}_Q\rangle < Q$ ]
    Return fail.
else
    Use linear algebra to prune  $\tilde{T}$  to an  $\mathbb{F}_p$ -basis of  $\tilde{Q}$ .
     $C :=$  matrix whose rows are the “ $\mathbb{F}_p$ -vectors” in  $\tilde{T}$ ; see (16).
     $D := C^{-1}$ .
    for  $i$  in  $\{1, \dots, l(d-2)\}$ 
        Use  $D[i]$  to write an SLP from  $\tilde{T}$  to the  $i$ th element of  $\tilde{\Delta}$ .
        if ( $Q$  is abelian) then                                     [ $\Delta' = \tilde{\Delta}$ ]
            Return list of SLPs to  $\tilde{\Delta}$ .
        else                                                         [ $\Delta' T = \tilde{\Delta}$ ]
            Modify the SLPs to  $\tilde{\Delta}$  to ones from  $\mathcal{T}_Q$  to  $\Delta'$  using  $\{t_1, \dots, t_k\}$ .
            Return the list of SLPs to  $\Delta'$ .

```

Correctness: C is a base change matrix from $\tilde{\Delta}$ to the elements of \tilde{T} so that D is a base change matrix from \tilde{T} to $\tilde{\Delta}$ (the i th row $D[i]$ gives the i th element of $\tilde{\Delta}$ as an \mathbb{F}_p -vector relative to \tilde{T}).

Timing: $O(d^2 \log^2 q)$ to write $O(kd)$ SLPs of length $O(d \log q)$. Observe that we do not evaluate those SLPs; they just serve to record how the elements of Δ' were constructed from \mathcal{T}_Q .

Remarks:

1. The procedure `ConstructDeltaPrime` is the first time that we used methods which depend heavily upon having the natural representation in hand. We were able simply to write down the matrices in Q that we wanted and then construct them from \mathcal{T}_Q using linear algebra. This process is both much harder and much less efficient inside a black box elementary abelian group (see, for example, [KS], 3.4, 4.4, 5.4, 6.4).
2. When q is even in case **S**, we have now constructed the elements $t_i = r_1(0, \rho^i)$ generating T using SLPs from \mathcal{T}_Q . This brings to an end the annoying subdivision of case **S**.

3. If `ConstructDeltaPrime` is successfully executed then we know that the input group G contains the appropriate classical group (if, for some reason, this is still in doubt). For, each classical group is generated by the groups $Q(x)$ and $Q(y)$ which have now been shown to be subgroups of G .

Finally, we define and construct our desired generating set Δ . The timing of the following procedure is dominated by that of `ConstructDeltaPrime`.

`ConstructDelta`(\mathcal{T}_Q , \mathcal{T}_X)

$\Delta' := \text{ConstructDeltaPrime}(\mathcal{T}_Q)$.

if (Q is abelian) then

[$\Delta = \Delta'$]

Return Δ' .

else

Return $\Delta := \{t_1, \dots, t_k\} \cup \Delta'$.

[$\{t_1, \dots, t_k\} \subset \mathcal{T}_X$]

4.6 The generating set \mathcal{T}

We have now constructed, using SLPs from the original generators, the precise sets $\Delta(x)$ and $\Delta(y)$ that we need to construct our target generating set \mathcal{T} for G . We first redefine Δ to be the union of those sets:

$$\Delta := \Delta(x) \cup \Delta(y). \quad (17)$$

Although Δ generates G , we need the larger set \mathcal{T} in order to execute the SLP routine given in section 5.

Recall the subgroups L , $U(E)$ and $U(F)$ defined in Theorem 6; generators for these key subgroups will comprise the lion's share of our generating set \mathcal{T} . We first give a pseudo-code overview of the main procedure which constructs \mathcal{T} from Δ .

`ConstructNewGenerators`(Δ)

$\mathcal{T}_L := \text{ConstructL}(\Delta)$;

[4.6.1]

$\mathcal{T}_E := \text{ConstructUpperU}(\Delta , \mathcal{T}_L)$;

[4.6.2]

$\mathcal{T}_F := \text{ConstructLowerU}(\Delta , \mathcal{T}_L)$;

[4.6.2]

(K , V_K) := the subgroup and support defined in Table 9;

[4.6.3]

if ($K \neq J$) or ($q < 16$) then

$\mathcal{S}_K := K \cap \Delta$;

[identify \mathcal{S}_K with 4×4 matrices induced on V_K]

$\mathcal{T}_K := \text{ClassicalConstructiveRecognition}(\langle \mathcal{S}_K \rangle)^{(\dagger)}$;

[section 6, or section 4 in case \mathbf{O}^-]

else

$\mathcal{T}_K := \mathcal{T}_J$;

[4.6.4]

$\mathcal{T}_U := \text{ConstructUsefulElements}(\mathcal{T}_E);$
Return $\mathcal{T} := \mathcal{T}_L \cup \mathcal{T}_E \cup \mathcal{T}_F \cup \mathcal{T}_K \cup \mathcal{T}_U.$

(†) The routine `ClassicalConstructiveRecognition` applied to the low dimensional subgroup K succeeds with probability $> 1/2$. We repeat the procedure, in the present setting, up to 6 six times to ensure that we successfully obtain \mathcal{T}_K with probability $> 1 - 1/64$.

4.6.1 The set \mathcal{T}_L

Here and in 4.6.2 we will construct elements of G using commutators of elements from Δ . It is clear that a short SLP can easily be written from $\{g, h\}$ to $[g, h]$. For $1 \leq i, j \leq m$ and $0 \leq a < l$, we have $r_1(\rho^a f_j, 0) \in \Delta(x)$ and $r'_1(e_i, 0) \in \Delta(y)$. The following procedure produces the $O(kd^2)$ elements of the set

$$\mathcal{T}_L := \{ r_i(\rho^a f_j, 0) \mid 1 \leq i \neq j \leq m, 0 \leq a < l \}. \quad (18)$$

`ConstructL(Δ)`

```

Initialise  $\mathcal{T}_L := \emptyset;$ 
  for  $i \in \{2, \dots, m\}$ 
    for  $i \neq j \in \{2, \dots, m\}$ 
      for  $a \in \{0, \dots, l-1\}$ 
         $\mathcal{T}_L := \mathcal{T}_L \cup \{ [ r_1(\rho^a e_i, 0), r'_1(f_j, 0) ] \};$ 
Return  $\mathcal{T}_L.$ 

```

Correctness: By Lemma 2.5(iii), $r_i(\rho^a f_j, 0) = [r_1(\rho^a e_i, 0), r'_1(f_j, 0)]$.

Timing: $O(kd^2)$ (again, we do not evaluate the commutators since we already know to which group elements they evaluate).

Proposition 4.1 (Gaussian elimination) *The group $\langle \mathcal{T}_L \rangle$ induces $\text{SL}(E)$ (resp. $\text{SL}(F)$) on the t.s. subspace E (resp. F) of V . If the linear transformation induced by $g \in \langle \mathcal{T}_L \rangle$ on E is represented by the matrix M relative to the basis e_1, \dots, e_m , then the linear transformation induced on F has matrix $\tilde{M}^{-\text{tr}}$ relative to f_1, \dots, f_m , where $\tilde{M} = \overline{M}$ in case \mathbf{U} and $\tilde{M} = M$ otherwise. Furthermore, if $g = \begin{pmatrix} A & * \\ * & * \end{pmatrix}$ where A is $m \times m$, then there are procedures to solve the following problems in $O(d^3 \log q)$ time:*

- (N) *if A is nonsingular, find $g' \in G$, having 'A' entry $\text{diag}(\alpha, 1, \dots, 1)$ for some $\alpha \in \mathbb{F}^*$, and also find SLPs of length $O(d^2 \log q)$ from \mathcal{T}_L to elements $g_1, g_2 \in \langle \mathcal{T}_L \rangle$ such that $g_1 g g_2 = g'$.*
- (S) *if $\text{rank}(A) = r < m$, find a matrix $g' \in G$, having 'A' entry $\begin{pmatrix} A_r & 0 \\ 0 & 0 \end{pmatrix}$, where A_r is an $r \times r$ matrix $\text{diag}(\alpha_1, \dots, \alpha_r)$ for $\alpha_i \in \mathbb{F}^*$, and also find SLPs of length $O(d^2 \log q)$ from \mathcal{T}_L to elements $g_1, g_2 \in \langle \mathcal{T}_L \rangle$ such that $g_1 g g_2 = g'$.*

Remark 4.2 *In the nonsingular and singular cases (N) and (S) above, we are **not** saying that we actually find the matrices g_i (this would require $O(d^5 \log^2 q)$ -time to evaluate the SLPs from \mathcal{T}_L to each g_i); we need only SLPs to the g_i .*

Proof. For $1 \leq i, j \leq m$ and $\lambda \in \mathbb{F}$, let $E_{ij}(\lambda)$ denote the elementary $m \times m$ matrix with λ in position (i, j) and zeros elsewhere, and let $X_{ij}(\lambda)$ be the transvection $I_m + E_{ij}(\lambda)$. A simple calculation shows that the linear transformation induced on F by $r_i(\rho^a f_j, 0) \in \mathcal{T}_L$ has matrix $X_{ij}(\rho^a)$. Hence $\langle \mathcal{T}_L \rangle$ induces $\text{SL}(F)$ on F as claimed. That the action of $g \in \langle \mathcal{T}_L \rangle$ on E and F is as stated follows from Table 4. Finally, the procedures (N) and (S) both use Gaussian elimination in $\text{SL}(E)$ with the transvections induced on E by the elements of \mathcal{T}_L . In particular, the timing and length of SLPs is as stated. \square

4.6.2 The sets \mathcal{T}_E and \mathcal{T}_F

There are both “lower” and “upper” versions of the procedures **ConstructU** to construct generators for $U(E)$ and $U(F)$ respectively. We only give procedures for $U = U(E)$. We begin with a subroutine to construct a useful element of $\langle \mathcal{T}_L \rangle$.

ConjugatingElement(\mathcal{T}_L)

Write down the matrix of any element c of $\langle \mathcal{T}_L \rangle \leq L$ such that $c: e_1 \mapsto e_2 \mapsto \dots \mapsto e_m$.
 Use Proposition 4.1 to write an SLP of length $O(d^2 \log q)$ from \mathcal{T}_L to c .
 Return c together with this SLP.

The structure of U varies depending on the type of G (cf. 2.5.2). The next subroutine constructs generators for the centre of U (recall that $\Delta(x)$ contains the transvection $t_b \in T$ for $1 \leq b \leq k$).

ConstructCentreOfU(Δ , c)

Initialise $\mathcal{T}_E^{(1)} := \Delta \cap Z(U)$.
 for $i \in \{1, \dots, m-1\}$
 for $b \in \{1, \dots, k\}$ *[only in cases **U** and **S**]*
 $\mathcal{T}_E^{(1)} := \mathcal{T}_E^{(1)} \cup \{ (t_b)^{c^i} \}$;
 for $i < j \in \{1, \dots, m\}$
 for $a \in \{0, \dots, l-1\}$
 $\mathcal{T}_E^{(1)} := \mathcal{T}_E^{(1)} \cup \{ [r_1(\rho^a e_j, 0), r'_1(e_i, 0)] \}$;
 Return $\mathcal{T}_E^{(1)}$.

Correctness: By Lemma 2.5(i), $r_i(\rho^a e_j, 0) = [r_1(\rho^a e_j, 0), r'_1(e_i, 0)]$. In cases **U** and **S**, $T(\langle e_{i+1} \rangle) = T(\langle e_1 \rangle)^{c^i} = \langle (t_b)^{c^i} \mid 0 \leq b < k \rangle$ for $1 \leq i \leq m-1$. Note that $r_i(\rho^a e_j, 0)$ has

matrix

$$u(E_{ij}(\rho^a) - E_{ji}(\rho^a)), \quad u(0, E_{ij}(\rho^a) - E_{ji}(\rho^a)) \text{ or } u(0, 0, E_{ij}(\rho^a) - E_{ji}(\rho^a)),$$

while elements of $T(\langle e_i \rangle)$ have matrix $u(E_{ii}(\lambda))$ or $u(0, E_{ii}(\lambda))$. It follows that $\mathcal{T}_E^{(1)}$ is a basis for the \mathbb{F}_p -space $Z(U)$, as required.

Timing: $O(kd^2)$, as in 4.6.1.

We now give the main procedure to construct nice generators for U from $\Delta \cup \mathcal{T}_L$ (recall that Δ contains $u_a = r_1(\rho^a v, \lambda_a)$ in cases \mathbf{O}^o and \mathbf{U} , and $u_{a,s} = r_1(\rho^a v_s, \lambda_{a,s})$ for $s = 1, 2$ in case \mathbf{O}^- , where $\lambda_a = \varphi(\rho^a v)$ in case \mathbf{O}^o , $\lambda_a + \overline{\lambda_a} = -(\rho\bar{\rho})^a$ in case \mathbf{U} , and $\lambda_{a,s} = \varphi(\rho^a v_s)$ in case \mathbf{O}^-).

ConstructU(Δ , \mathcal{T}_L)

```

    c := ConjugatingElement(  $\mathcal{T}_L$  );
     $\mathcal{T}_E^{(1)}$  := ConstructCentreOfU(  $\Delta$  , c );
    Initialise  $\mathcal{T}_E^{(2)}$  :=  $\emptyset$ . [generators for  $U/Z(U)$ ]
    if (U is nonabelian) then
        for i  $\in$  {1, ..., m - 1}
            for a  $\in$  {0, ..., l - 1}
                Add {  $(u_{a,1})^{c^i}$ ,  $(u_{a,2})^{c^i}$  } in case  $\mathbf{O}^-$ , or else  $(u_a)^{c^i}$ , to  $\mathcal{T}_E^{(2)}$ .
    Return  $\mathcal{T}_E := \mathcal{T}_E^{(1)} \cup \mathcal{T}_E^{(2)}$ .

```

Correctness: Assume that U is nonabelian (and hence a class-2 nilpotent group), and consider just the case \mathbf{O}^- . For $\lambda \in \mathbb{F}$ and $1 \leq i < j \leq m$, let

$$\varepsilon_i(\lambda) = \text{the row vector in } \mathbb{F}^m \text{ with } \lambda \text{ in coordinate } i \text{ and } 0\text{'s elsewhere.} \quad (19)$$

Then $\mathcal{T}_E^{(2)}$ is the set of all $r_i(\rho^a v_s, \lambda_{a,s})$, where $1 \leq i \leq m$, $0 \leq a < k$ and $s = 1, 2$, having matrix $u(\varepsilon_i(\rho^a), 0, M_1)$ or $u(0, \varepsilon_i(\rho^a), M_2)$ for $s = 1$, or 2 respectively. It follows that $\mathcal{T}_E^{(2)}$ projects onto an \mathbb{F}_p -basis of the $k(d-2)$ -space $U/Z(U)$.

Timing: $O(d^3 \log q)$, dominated by time required to construct c .

4.6.3 The subgroup K

Our new generating set \mathcal{T} , output by subroutine **ConstructNewGenerators** described on p. 26, contains generators for a certain naturally embedded subgroup K . That subgroup, along with its support V_K , is defined in Table 9 (recall that J and V_J were constructed in 4.3). In the SLP algorithm in section 5, our strategy will be to modify a given matrix g so that it lies inside K .

Timing: $O(\xi \log \log q + \mu \log^2 q + \chi)$, the timing stated in Theorem 1.1 for $d = 5$ in case \mathbf{O}^- .

Remark: In case \mathbf{O}^- , K is a 5-dimensional group rather than the more natural choice $\Omega^-(4, q)$. This is to avoid squaring the size of the field to recognise the latter group (cf. 6.1.4).

Table 9: The subgroup K

case	V_K	K
S (q odd)	V_J	$J = \text{Sp}(V_J)$
S (q even)	V_J	$\text{Sp}(V_J) > J$
U ^{<i>e</i>}	V_J	$J = \text{SU}(V_J)$
U ^{<i>o</i>}	$\langle e_1, v, f_1 \rangle$	$\text{SU}(V_K)$
O ⁺	V_J	$J = \Omega^+(V_J)$
O ^{<i>o</i>}	V_J	$J = \Omega^+(V_J)$
O ⁻	$\langle e_1, e_2, v_1, v_2, f_1 \rangle$	$\Omega(V_K)$

Table 10: The elements x_r

case	S, U	O ^{<i>o</i>} , O ⁻	O ⁺
$x_r =$	$\prod_{i=r+1}^m r_i(0, \alpha)$	$\prod_{i=r+1}^m r_i(v_*, \varphi(v_*))$	$\prod_{i=0}^{t-1} r_{m-i}(e_{m-r+1-i}, 0)$

4.6.4 Some useful elements

Finally, we construct in time $O(\mu d)$ a set \mathcal{T}_U of m (or possibly $\lfloor m/2 \rfloor$) elements to be used in the subroutine `UpperReduce` of `WriteSLP` (cf. 5.2.1). In Table 10, the scalar α is 1 (resp. $0 \neq \delta = -\delta^q$) in case **S** (resp. **U**), and the vector v_* is v (resp. v_1) in case **O**^{*o*} (resp. **O**⁻).

`ConstructUsefulElements`(\mathcal{T}_E)

Initialise $X_U := \emptyset$.

for $\begin{cases} m-r = 2t \in \{2, 4, \dots, 2\lfloor m/2 \rfloor\} & \text{in case } \mathbf{O}^+, \text{ or} \\ r \in \{1, \dots, m\} & \text{otherwise} \end{cases}$

Write an SLP of length $O(d)$ from \mathcal{T}_E to x_r (defined in Table 10).

$\mathcal{T}_U := \mathcal{T}_U \cup \{x_r\}$;

Return \mathcal{T}_U .

Properties of the elements x_r : In each case,

$$x_r = u(M), u(z, M), \text{ or } u(z, 0, M)$$

(cf. 2.5.2) where $z = z(r) \in \mathbb{F}^m$ and $M = M(r)$ is an $m \times m$ matrix

$$M(r) = \begin{pmatrix} 0 & 0 \\ 0 & M_{m-r} \end{pmatrix},$$

(M_{m-r} is an $(m-r) \times (m-r)$ matrix). The key properties of x_r for our purposes are that M_{m-r} is nonsingular and, when applicable (cases \mathbf{O}° and \mathbf{O}^-), we have

$$z(r) = (0, \dots, 0, 1, \dots, 1),$$

containing r 0's and $m-r$ 1's. Multiplying out the defining product in each case using matrices relative to \mathcal{B} , and denoting the $m-r$ identity matrix by I_{m-r} , we find that $M_{m-r} = I_{m-r}$, $-\delta I_{m-r}$ or $-\varphi(v_1)I_{m-r}$ in case \mathbf{S} , \mathbf{U} or \mathbf{O}^- (q even) respectively. In cases \mathbf{O}° and \mathbf{O}^- (q odd), we have

$$M_{m-r} = \lambda \begin{pmatrix} 1 & 2 & \dots & 2 \\ & \ddots & \ddots & \vdots \\ & & \ddots & 2 \\ & & & 1 \end{pmatrix},$$

where λ is $-1/2$ (resp. -1) in case \mathbf{O}° (resp. \mathbf{O}^-). Finally, in case \mathbf{O}^+ , we have

$$M_{m-r} = \begin{pmatrix} 0 & D_{(m-r)/2} \\ -D_{(m-r)/2} & 0 \end{pmatrix},$$

where D_i is the $i \times i$ matrix with 1s on the off-diagonal and 0s elsewhere.

4.6.5 Total timing and reliability

The running time of the routine `ConstructNewGenerators`, summarised at the beginning of 4.6, is dominated by $O(\xi \log \log q + \mu \log^2 q + \chi)$, the time to recognise K and by Proposition 4.1. Hence, we obtain the new generating set

$$\mathcal{T} := \mathcal{T}_L \cup \mathcal{T}_E \cup \mathcal{T}_F \cup \mathcal{T}_K \cup \mathcal{T}_U \tag{20}$$

for G , with probability $> 1 - 1/64$, in time $O(d^3 \log q + \xi \log \log q + \mu \log^2 q + \chi)$.

4.7 Small fields

Our final task in this section is to obtain replacements for the subroutines used at the beginning of the algorithm (4.3 and 4.4) when $q < 16$. It is tempting simply to use the entire algorithm in [KS] for bounded q since the dependence of the timing in [KS] on the field size would then be irrelevant. However, those black box algorithms are recursive, producing an extra factor of d in their timing estimates that does not occur in ours.

We proceed exactly as in [KS] for bounded q , in effect regarding G as a black box group. We will obtain generating sets for subgroups $Q(x)$ and $Q(y)$ for some singular points x and $y \notin x^\perp$. As in 4.4, we will only discuss $Q = Q(x)$. In each case, [KS] constructs an analogue of J ,

together with analogues (called there Q_{8k} , Q_4 or Q_6) of $O_p(J_x)$ that lead to probable generators for Q . However, we do not require all of the constructions used in [KS] so, considering each classical group in turn, we summarise the information we need, together with timing estimates and references to procedures.

- O** Here we assume that $d \geq 9$; otherwise $|G|$ is bounded and is recognised by brute force. Use [KS], 4.2.1 (case 1,6,7,8 or 9), to find a long root element t and an element τ (an analogue of the τ constructed here in 4.3.1). Use [KS], 4.2.2, together with t , to construct $O(\log d)$ subgroups Q_{8k} for $1 \leq k \leq \lceil 2^5 \log(4d) \rceil$, each of which is a 6-dimensional subspace of a group $Q = Q(x)$ for some x . Finally, use [KS], 4.3, together with the groups Q_{8k} and the element τ , to find a generating set \mathcal{T}_Q for a subgroup of Q .
- S** Use [KS], 5.2.1 (for $q < 16$ odd), or [KS], 5.2.2 (for $q < 16$ even), to find: a subgroup Q_4 of order q^3 (q odd), or Q_6 of order q^4 (q even), of $Q = Q(x)$ for some x ; and an element τ (q odd), or two elements τ, τ' (q even), normalising Q . Use [KS], 5.3.1, together with Q_4 and τ (q odd), or Q_6 and τ, τ' (q even), to find a generating set \mathcal{T}_Q for a subgroup of Q .
- U** Assume that $d \geq 7$ if $q = 2$; otherwise $d \geq 5$ as usual. Use [KS], 6.2.1 to find: a subgroup Q_4 of order q^5 (or 2^9 if $q = 2$) of $Q = Q(x)$ for some x ; and an element τ normalising Q . Use [KS], 6.3.1, together with Q_4 and τ , to find a generating set \mathcal{T}_Q for a subgroup of Q .

Timing and reliability: We obtain all of the necessary constructions, with probability $> 3/4$, in time $O(\xi d + \mu d^2)$.

4.8 Total timing and reliability

Adding up the running times of the subroutines in section 4 and also the failure probabilities of the randomised subroutines, the routine

ClassicalConstructiveRecognition(G)

returns a new generating set \mathcal{T} for G , with probability $> 1/2$, in time

$$O(d^3 \log q(d + \log d \log^3 q) + \xi\{d + \log \log q\} + \mu\{|\mathcal{S}| + d^2 \log^2 q\} + \chi \log q).$$

This completes the preprocessing phase of the algorithm.

5 Straightline Programs

In the previous section we gave an algorithm to construct a carefully tailored generating set \mathcal{T} for the given classical group G . In this section we complete the proof of Theorem 1.1, for $d \geq 5$,

by presenting an algorithm to write an SLP from \mathcal{T} to any given element $g \in G$. The algorithm is analogous to that used in [Ce] for $G = \text{Sp}(d, q)$ but is more complicated for some of the other cases. Nevertheless, the same general approach is applied to all classical groups, giving the same running time as the algorithm in [Ce] in each case.

Convention: *If σ is an SLP from a set X , then σ^{-1} will denote an SLP from X to the inverse of the element to which σ evaluates (from X); similarly, $\sigma\sigma'$ will denote an SLP to the product of the elements to which SLPs σ and σ' evaluate.*

5.1 The key subgroups

Recall that the algorithm `ClassicalConstructiveRecognition` returns a new generating set \mathcal{T} for G of the form

$$\mathcal{T} = \mathcal{T}_L \cup \mathcal{T}_E \cup \mathcal{T}_F \cup \mathcal{T}_K \cup \mathcal{T}_U.$$

Our strategy for the main routine `WriteSLP` (which is presented in 5.2) will be to modify a given element $g \in G$ in various ways using elements from the subgroups L , $U(E)$ and $U(F)$. This will require that we are able to use SLPs from \mathcal{T} to construct given elements from each of those groups.

5.1.1 Constructing elements of L

Elements of L have matrix

$$\text{diag}(A, \tilde{A}^{-\text{tr}}), \quad \text{diag}(A, \lambda, \tilde{A}^{-\text{tr}}) \quad \text{or} \quad \text{diag}(A, \Lambda, A^{-\text{tr}}),$$

where $\tilde{A} = \bar{A}$ in case \mathbf{U} and $\tilde{A} = A$ otherwise, and A, λ, Λ behave as in Table 4. The following procedure writes an SLP of length $O(d^2 \log q)$ from $\mathcal{T}_L \cup \mathcal{T}_K$ to any given element $g \in L$ (see 2.5.1 and 4.6.3 for descriptions of the subgroups L and K respectively).

`WriteLSLP`(g , \mathcal{T})

Use Proposition 4.1(N) to find:

$g' \in G$ with ‘ A ’ entry $\text{diag}(\alpha, 1, \dots, 1)$; and

SLPs σ_1, σ_2 from \mathcal{T}_L to elements $g_1, g_2 \in \langle \mathcal{T}_L \rangle$ such that $g_1 g g_2 = g'$.

$\sigma := \text{WriteSLP}(g', \mathcal{T}_K)$.

Return $(\sigma_1)^{-1} \sigma (\sigma_2)^{-1}$.

Correctness: We claim that $g' = g_1 g g_2 \in K$. Indeed, if U denotes the 0, 1 or 2-dimensional subspace of V spanned by the nonsingular vectors in \mathcal{B} , then $[V, g'] \leq \langle x, y, U \rangle \leq [V, K]$ and g' is the identity on $[V, g']^\perp \geq [V, K]^\perp$ so that $g' \in K$.

Timing: We note, in cases \mathbf{U}^e , \mathbf{S} and \mathbf{O}^+ , that g' is a diagonal element of a subgroup of K isomorphic to $\mathrm{SL}(2, q)$. In those cases, we use either 6.1.1 or 6.1.2 to write an SLP to g' . In case \mathbf{O}^0 , $g' \in K \cong \Omega^+(4, q)$, and we use 6.1.3. In case \mathbf{O}^- , $g' \in K \cong \Omega(5, q)$, so we are now in case \mathbf{O}^0 . Finally in case \mathbf{U}^o , we use the SLP routine described in 6.4.7. The cost of the call to `WriteSLP`(g' , \mathcal{T}_K) is dominated by the latter timing so that `WriteLSLP` runs in time $O(d^3 \log q + \log^2 q)$.

5.1.2 Constructing elements of $U(E)$ and $U(F)$

We consider only the group $U = U(E)$. Elements of U have matrix $u(M)$, $u(z, M)$ or $u(z_1, z_2, M)$ as in Table 8. The procedure is very similar for all classical groups so we give details only for the most difficult case \mathbf{O}^- .

Recall that the routine `ConstructUpperU`(Δ) returned $\mathcal{T}_E = \mathcal{T}_E^{(1)} \cup \mathcal{T}_E^{(2)}$, where $\mathcal{T}_E^{(1)}$ generates $Z(U)$ and $\mathcal{T}_E^{(2)}$ generates $U/Z(U)$. Recall also that, for $1 \leq i < j \leq m$ and $0 \leq a < l$, the set $\mathcal{T}_E^{(1)}$ contains the linear transformation $r_i(\rho^a e_j, 0)$ having matrix $u(0, 0, E_{ij}(\rho^a) - E_{ji}(\rho^a)) \in Z(U)$. Hence, elements of $Z(U)$, relative to \mathcal{B} , are also vectors relative to $\mathcal{T}_E^{(1)}$ when viewed as elements of an \mathbb{F}_p -space of dimension $< kd^2$ (cf. (†)).

The following procedure writes an SLP of length $O(d^2 \log q)$ from \mathcal{T}_E to any given element $g \in U$.

`WriteUSLP`(g , \mathcal{T}_E)

[Case \mathbf{O}^-]

if ($g \in Z(U)$) **then**

Express g as an \mathbb{F}_p -linear combination of the basis $\mathcal{T}_E^{(1)(\dagger)}$.

Hence return an SLP of length $O(d^2 \log q)$ from $\mathcal{T}_E^{(1)}$ to g .

else [$g = u(z_1, z_2, M)$ for $z_1, z_2 \in \mathbb{F}^m$ not both zero]

Write $z_s = \sum_{i=1}^m \sum_{a=0}^{k-1} \alpha_{ias} \varepsilon_i(\rho^a)^\ddagger$ for $s = 1, 2$, where $0 \leq \alpha_{ias} < p$.

Hence, write an SLP σ of length $O(d \log q)$ from $\mathcal{T}_E^{(2)}$ to

$$h = \prod_{s=1}^2 \prod_{i=1}^m \prod_{a=0}^{k-1} r_i(\rho^a v_s, \lambda_{a,s})^{\alpha_{ias}}.$$

Recursively set $\sigma' := \text{WriteUSLP}(gh^{-1}, \mathcal{T}_E)$;

[$gh^{-1} \in Z(U)$]

Return $\sigma'\sigma$.

(†) Performing even elementary linear algebra (such as a change of basis) in the $O(kd^2)$ -dimensional vector space $Z(U)$ requires $O(k^3 d^6)$ integer computations, so it is crucial that

we constructed the exact \mathbb{F}_p -basis $\mathcal{T}_E^{(1)}$ for $Z(U)$ directly. This dimensional blow-up was also avoided in [Ce], but using different ideas.

(‡) The row vectors $\varepsilon_i(\lambda)$ are defined in (19).

Correctness: The only part of the procedure which needs justification is that the input to the recursive call, gh^{-1} , is in $Z(U)$. This follows from the construction of h . Indeed, if $g = u(z_1, z_2, M)$, then $h = u(z_1, z_2, M^*)$ for some matrix M^* so that $gh^{-1} = u(0, 0, M') \in Z(U)$ for some matrix M' .

Timing: $\mu = O(d^3 \log q)$ to compute gh^{-1} since the elements of $\mathcal{T}_E^{(2)}$ are sparse.

5.2 The SLP algorithm

We now complete the proof of Theorem 1.1 for $d \geq 5$. Let $g \in \text{GL}(V)$ be given. We begin with some preliminary checks to recognise when $g \notin G$. Compute $\det(g)$ and report that $g \notin G$ if this is not 1. Otherwise, verify that g preserves the G -invariant form obtained in 4.2 and report that $g \notin G$ if this is not the case. We are left with the possibility, in case **O**, that $g \in \text{SO}^\varepsilon(V) \setminus G$. However, there are elementary tests to recognise when this is the case (cf. [KIL] pp.29-30), so we may now assume that $g \in G$.

Our strategy is to use elements from the subgroups $U(E)$, $U(F)$ and L to filter g down the following short chain of subgroups

$$G > G_E > G_{E,F} > \{1\};$$

verifying the correctness of each filtration will usually just involve multiplying matrices. Write g relative to \mathcal{B} , so that

$$g = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \quad \begin{pmatrix} A & \xi^{\text{tr}} & B \\ \eta & \lambda & \omega \\ C & \zeta^{\text{tr}} & D \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} A & \xi_1^{\text{tr}} & \xi_2^{\text{tr}} & B \\ \eta_1 & \lambda_{11} & \lambda_{12} & \omega_1 \\ \eta_2 & \lambda_{21} & \lambda_{22} & \omega_2 \\ C & \zeta_1^{\text{tr}} & \zeta_2^{\text{tr}} & D \end{pmatrix}$$

as in (2), (3) or (4) depending upon the type of G . The following is a pseudo-code overview of the SLP algorithm.

WriteSLP(g , \mathcal{T})

$$(g', \sigma_{u1}, \sigma_{u2}) := \text{UpperReduce}(g, \mathcal{T}), \quad [5.2.1]$$

$$(g'', \sigma_l) := \text{LowerReduce}(g', \mathcal{T}), \text{ and} \quad [5.2.2]$$

$$\sigma := \text{WriteLSLP}(g'', \mathcal{T}). \quad [5.1.1]$$

Return $(\sigma_{u1})^{-1} \sigma (\sigma_{u2} \sigma_l)^{-1}$.

Table 11: Matrix entries z , z_1 , z_2 , M (notation as in 2.5.2)

Case	z	M
$\mathbf{S}, \mathbf{O}^+, \mathbf{U}^e$		$M^{\text{tr}} = -A^{-1}B$
$\mathbf{O}^o, \mathbf{U}^o$	$\tilde{z} = \xi A^{-\text{tr}}$	$M^{\text{tr}} = -A^{-1}B - \tilde{z}^{\text{tr}}z$
\mathbf{O}^-	$z_i = \xi_i A^{-\text{tr}}$	$M^{\text{tr}} = -A^{-1}B - z_1^{\text{tr}}z_{\sigma(1)} - \alpha z_2^{\text{tr}}z_{\sigma(2)}$

5.2.1 From G to G_E

The following algorithm returns a triple (g', σ_1, σ_2) , where $g' \in G_E$ and, for $i = 1, 2$, σ_i is an SLP from of length $O(d^2 \log q)$ from \mathcal{T} to an element h_i such that $g' = h_1 g h_2$.

UpperReduce(g , \mathcal{T})

if (A is nonsingular) then

$u := u(M)$, $u(z, M)$ or $u(z_1, z_2, M)$ in $U(E)$ as in Table 11.

$w := u^{\text{tr}}$.

Return the triple (gw , 1 , WriteUSLP(w , \mathcal{T})).

[5.1.2]

else

[A is singular]

$r := \text{rank}(A)$.

Use Proposition 4.1(S) to find $g' \in G$ with entry ' A ' of the form $\begin{pmatrix} A_r & 0 \\ 0 & 0 \end{pmatrix}$, where $A_r = \text{diag}(\alpha_1, \dots, \alpha_r)$ for $\alpha_i \in \mathbb{F}^*$, together with SLPs σ_i ($i = 1, 2$) of length $O(d^2 \log q)$ from \mathcal{T}_L to elements $g_1, g_2 \in \langle \mathcal{T}_L \rangle$ such that $g_1 g g_2 = g'$.

Recursively set (g'' , 1 , σ) := UpperReduce($g'x_r$, \mathcal{T});

[$x_r \in \mathcal{T}_U$]

Return the triple (g'' , σ_1 , $\sigma_2 x_r \sigma$).

Correctness: Suppose first that A is nonsingular. Then it is an easy matter to verify that the entries of u (Table 11) satisfy the appropriate constraints for an element of U , and then that $gw \in G_E$, as required.

Suppose then that A is singular. We need to show that $g'x_r$ has nonsingular ' A ' entry (so that we can make the recursive call UpperReduce($g'x_r$, \mathcal{T})) and, in case \mathbf{O}^+ , that $m - r$ is even (since otherwise x_r is undefined).

Write the top block entries of g' (corresponding to the first m rows) as follows:

$$\begin{pmatrix} A_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \quad \begin{pmatrix} A_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \xi_1^{\text{tr}} \\ \xi_2^{\text{tr}} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \quad \text{or}$$

$$\begin{pmatrix} A_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \xi_{11}^{\text{tr}} \\ \xi_{12}^{\text{tr}} \end{pmatrix} \begin{pmatrix} \xi_{21}^{\text{tr}} \\ \xi_{22}^{\text{tr}} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix},$$

so that B_{12} is $m \times (m-r)$, B_{22} is $(m-r) \times (m-r)$, $\xi_{12} \in \mathbb{F}^{m-r}$ etc.

Claim 1: $B_{21} = 0$ in all cases; $\xi_2 = 0$ in cases \mathbf{U}^o and \mathbf{O}^o ; and $\xi_{12} = \xi_{22} = 0$ in case \mathbf{O}^- .

This technical fact is an elementary consequence of the matrix constraints imposed upon these m rows of g' (cf. Tables 1 and 2). Again, we give details only for the most difficult case \mathbf{O}^- . We consider three different sets of constraints arising from the three different possible bases for the definite line $\langle v_1, v_2 \rangle$ (cf. Lemma 2.1).

Case $q \equiv 3 \pmod{4}$. Since g' preserves $(,)$, we obtain the constraints

$$\xi_{12}^{\text{tr}} \xi_{12} + \xi_{22}^{\text{tr}} \xi_{22} = 0 \quad \text{and} \quad B_{21} A_r^{\text{tr}} + \xi_{12}^{\text{tr}} \xi_{11} + \xi_{22}^{\text{tr}} \xi_{21} = 0.$$

For nonzero row vectors $v = (v_i), w = (w_i) \in \text{GF}(q)^{m-r}$, $v^{\text{tr}} v + w^{\text{tr}} w = 0$ only if $v_i^2 = -w_i^2$ for $1 \leq i \leq m-r$. This can occur only if $-1 = \square$ and hence only if $q \equiv 1 \pmod{4}$. Hence, $\xi_{12} = \xi_{22} = 0$ and this case follows.

Case $q \equiv 1 \pmod{4}$. This is similar to the preceding case. The constraints here can be satisfied by nonzero ξ_{12}, ξ_{22} only if $-\rho = \square$, which is the case if and only if $q \equiv 3 \pmod{4}$.

Case q is even. Since g' preserves $(,)$, we obtain the constraints

- (i) $\xi_{12}^{\text{tr}} \xi_{22} + \xi_{22}^{\text{tr}} \xi_{12} = 0$; and
- (ii) $B_{21} A_r^{\text{tr}} + \xi_{12}^{\text{tr}} \xi_{21} + \xi_{22}^{\text{tr}} \xi_{11} = 0$.

Since g' also preserves φ , we have the additional constraint

- (iii) each diagonal entry of $\alpha_1 \xi_{12}^{\text{tr}} \xi_{12} + \alpha_2 \xi_{22}^{\text{tr}} \xi_{22} + \xi_{12}^{\text{tr}} \xi_{22}$ is zero,

where α_1, α_2 are as in Lemma 2.1. Equation (i) requires that if both ξ_{12} and ξ_{22} are nonzero, then $\xi_{12} = \lambda \xi_{22}$ for some $\lambda \in \mathbb{F}$. By (iii), the matrix $\xi_{22}^{\text{tr}} \xi_{22} [\alpha_1 \lambda^2 + \lambda + \alpha_2]$ has zeros on its diagonal. Hence, either $\xi_{22} = 0$ or λ is a root of the equation $\alpha_1 x^2 + x + \alpha_2 = 0$. In the latter case $\lambda v_1 + v_2$ is then a singular vector of the definite line $\langle v_1, v_2 \rangle$. Hence $\xi_{22} = 0$ and it follows from (iii) that $\xi_{12} = 0$ and then from (ii) that $B_{21} = 0$.

Claim 2: $g'x_r$ has nonsingular 'A' entry.

Using claim 1 and the matrix x_r (cf. 4.6.4) we see that the 'A' entry of $g'x_r$ is $\begin{pmatrix} A_r & * \\ 0 & A_{m-r} \end{pmatrix}$, where $A_{m-r} = B_{22} M_{m-r}$. A corollary of claim 1 is that B_{22} is nonsingular (since g' is nonsingular) and we showed in 4.6.4 that M_{m-r} is nonsingular. Hence, A_{m-r} is nonsingular and the claim follows.

Table 12: Matrix entries z , z_1 , z_2 , M

Case	z	M
$\mathbf{S}, \mathbf{O}^+, \mathbf{U}^e$		$M = -D^{-1}C$
$\mathbf{O}^o, \mathbf{U}^o$	$\tilde{z} = -\zeta D^{-\text{tr}}$	$M = -D^{-1}C - \tilde{z}^{\text{tr}}z$
\mathbf{O}^-	$z_1 = -\zeta_{\sigma(1)} D^{-\text{tr}}$ $z_2 = -\alpha^{-1} \zeta_{\sigma(1)} D^{-\text{tr}}$	$M = -D^{-1}C - z_{\sigma(1)}^{\text{tr}}z_1 - \alpha z_{\sigma(2)}^{\text{tr}}z_2$

Claim 3: $m - r$ is even in case \mathbf{O}^+ .

Since $g' \in G = \Omega^+(V)$ has ‘ A ’ entry $\begin{pmatrix} A_r & 0 \\ 0 & 0 \end{pmatrix}$, it follows that $\dim(E \cap F) = 0$ and $\dim(Eg' \cap F) = m - r$ have the same parity ([KIL] p. 30, description 4).

Claims 2 and 3 establish the correctness of the procedure.

Timing: $O(d^3 \log q)$, dominated by Proposition 4.1.

5.2.2 From G_E to $G_{E,F}$

The following algorithm takes as input an element $g \in G_E$, and returns a pair (g', σ) , where $g' \in G_{E,F} = L$ and σ is an SLP of length $O(d^2 \log q)$ from \mathcal{T}_E to an element $u \in U(E)$ such that $g' = gu$. Recall the notation for the block entries of g in (2)–(4) and note that: $B = 0$; $\xi = 0$ in cases \mathbf{O}^o and \mathbf{U}^o ; and $\xi_1 = \xi_2 = 0$ in case \mathbf{O}^- .

`LowerReduce`(g , \mathcal{T})

$u := u(M)$, $u(z, M)$ or $u(z_1, z_2, M)$ in $U(E)$ as in Table 12.

Return the pair (gu , `WriteUSLP`(u , \mathcal{T})).

Correctness: Since A is nonsingular, the constraints on the block matrix entries of g mean that $A = D^{-\text{tr}}$ in cases \mathbf{S} and \mathbf{O} and $A = \overline{D}^{-\text{tr}}$ in case \mathbf{U} so that D is also nonsingular (hence Table 12 makes sense). Additionally, in cases \mathbf{O}^o and \mathbf{U}^o , we find that $\omega = 0$ and, in case \mathbf{O}^+ , that $\omega_1 = \omega_2 = 0$. A matrix calculation shows that $gu \in G_{E,F}$.

Timing: $O(d^3 \log q)$.

5.2.3 Timing for `WriteSLP`

Adding together the timing for each of its subroutines, we see that `WriteSLP` returns an SLP of length $O(d^2 \log q)$ from \mathcal{T} to any given element $g \in G$ in time $O(d^3 \log q + \log^2 q)$. This completes the proof of Theorem 1.1 when $d > 4$.

6 Low Dimensions

We now deal with the cases $2 \leq d \leq 4$ and hence complete the proof of Theorem 1.1. Our strategy will be to use the algorithm in [CoLG] for $\mathrm{SL}(2, q)$ subgroups to speed up the more time-consuming calculations. The algorithms for $\mathrm{SU}(3, q)$ and $\mathrm{SU}(4, q)$ (in 6.3 and 6.4 respectively) are the most involved.

We assume that the elements of the generating set \mathcal{S} are written relative to a standard basis of the appropriate type. We also *assume that* $q \geq 16$ throughout section 6, since otherwise $|G|$ is bounded and is handled by brute force.

6.1 Groups involving $\mathrm{SL}(2, q)$

For each of the groups G considered in 6.1, we will first construct a function $\theta: G \rightarrow \mathrm{SL}(2, r)$ or $\mathrm{SL}(2, r) \times \mathrm{SL}(2, r)$, where r is either q or q^2 . The function θ will be a procedure which takes any given $g \in G$ (not necessarily given as an SLP from the generators of G) and computes its image $g\theta$. In each case $g \mapsto g\theta$ (possibly modulo scalars) will be an isomorphism and we will be able to complete the recognition of G by recognising $\mathrm{SL}(2, q)$ in its natural representation. For example, instead of writing an SLP from \mathcal{S} to the given $g \in G$, we will instead write an SLP from $\mathcal{S}\theta$ to $g\theta$, and then pull back to G .

6.1.1 The natural representation

In [CoLG], a fast algorithm is presented which constructively recognises a group $G = \langle \mathcal{S} \rangle \cong \mathrm{SL}(2, q)$, in its natural representation as a 2×2 matrix group with entries in $\mathbb{F} = \mathrm{GF}(q)$, assuming an oracle to compute discrete logarithms in \mathbb{F}^* . The algorithm constructs a generating set $\{h, t_1, t_2\}$ for G using SLPs from \mathcal{S} , where (relative to a suitable basis) h is diagonal of order $q - 1$, and t_1 (resp. t_2) is an upper (resp. lower) unitriangular matrix (transvection).

The unique feature of the algorithm in [CoLG] is the method employed to obtain t_1 and t_2 . A direct search for a transvection would require at least q random choices to guarantee success with high probability, but this is avoided by first finding h , and then using h , together with discrete logarithms, to construct each t_i . The number of random choices required then reduces dramatically to $O(\log \log q)$.

Timing and reliability: $O(\xi \log \log q + \chi + \log q)$ to construct the elements h , t_1 and t_2 . An SLP of length $O(\log q)$ from $\{h, t_1, t_2\}$ to any given element $g \in G$ is found using $O(\log q)$ field operations. We assume (by repetition if necessary) that 6.1.1 succeeds with probability $> 3/4$.

6.1.2 Adapting for $SU(2, q)$

Since we need to work with the defining field $\text{GF}(q^2)$ in case \mathbf{U} it is convenient to have a separate routine for $SU(2, q)$ subgroups (reliability and timing will be as in 6.1.1). Let $G = \langle \mathcal{S} \rangle = SU(V)$, preserving an hermitian form on the 2-space V over $\mathbb{F} = \text{GF}(q^2)$, whose matrices are written relative to a standard basis e, f .

The isomorphism θ ([Ta], Theorem 10.9): Use Lemma 3.5(i) to find $0 \neq \delta = -\bar{\delta}$ and define $\theta: SU(V) \rightarrow SL(V_0)$, where V_0 is a 2-space over \mathbb{F}_0 , sending

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto \begin{pmatrix} a & b\delta \\ c/\delta & d \end{pmatrix}.$$

6.1.3 $\Omega^+(4, q)$

Let $G = \langle \mathcal{S} \rangle = \Omega^+(V)$, preserving a quadratic form $\varphi = \varphi^+$ on the 4-space V . Here G preserves a tensor decomposition $V = U \otimes W$ of V , which gives rise to a homomorphism $\theta: G \rightarrow \text{PSL}(2, q) \times \text{PSL}(2, q)$.

The homomorphism θ : We indicate two alternative methods for constructing θ ; the timing is dominated by 6.1.1.

The first uses the quadratic form φ and the fact that G acts intransitively on the set of $2(q+1)$ t.s. lines of V , with two orbits \mathcal{O}_l and \mathcal{O}_r of equal size ([KIL], p. 30, description 4). Viewing each orbit as a projective line, we compute the action of any given $g \in G$ on each line, and hence obtain $g\theta$.

The second uses the more general algorithm in [LGO1] for computing tensor decompositions. Applied in the present setting, this is equivalent to the SMASH algorithm presented in [H+].

Completion: It is now fairly elementary to complete the recognition of G . Let π_1 and π_2 denote the projections onto the $SL(2, q)$ factors. We construct sets $X_i \subset G$ ($i = 1, 2$) with $\langle X_1 \cup X_2 \rangle = G$ such that $(X_i\theta)\pi_j = 1$ if $i \neq j$ and $\langle (X_i\theta)\pi_i \rangle = SL(2, q)$. This is achieved using 2-element sets X_i with one element of $\text{ppd}^\#(p; k)$ -order and the other of $\text{ppd}^\#(p; 2k)$ -order. Both sets are obtained, with high probability, by sampling at most 40 elements of G .

For example, the probability that a randomly chosen element $g \in G$ is such that $(g\theta)\pi_1$ has $\text{ppd}^\#(p; k)$ -order such that $(|(g\theta)\pi_1|, |(g\theta)\pi_2|) \leq 2$, is at least $1/8$. For such a g , $g^{p(q+1)}$ is selected as the $\text{ppd}^\#(p; k)$ -element of X_1 . Each of the other four types of element occur with similar probability. Hence, the probability that we fail to find at least one of them after 40 tries is $< 4(1 - 1/8)^{40} < 1/8$.

6.1.4 Comments on $\Omega(3, q)$ and $\Omega^-(4, q)$

Here, since $\Omega(3, q) \cong \mathrm{SL}(2, q)$ and $\Omega^-(4, q) \cong \mathrm{PSL}(2, q^2)$, we cite the algorithm in [LGO2] for recognising irreducible representations of $\mathrm{SL}(2, q)$. We remark that, even though $\Omega^-(4, q)$ was the most natural choice for K in 4.6.3, recognising this group using [LGO2] would have required squaring the size of the given field; hence the reason for choosing $\Omega(5, q)$.

6.2 $\mathrm{Sp}(4, q)$

Let $G = \langle \mathcal{S} \rangle = \mathrm{Sp}(V)$, preserving an alternating form on the 4-space V . Here, we will construct a homomorphism $\theta: G \rightarrow \Omega(5, q)$ and then use the main algorithm, applied to $G\theta$, to complete the recognition of G .

The homomorphism θ : Identify the exterior square $\Lambda^2(V)$ with the 6-space of skew-symmetric 4×4 matrices over \mathbb{F} via $u \wedge v := u^{\mathrm{tr}}v - v^{\mathrm{tr}}u \in W$ for $u, v \in V$. For

$$M = \begin{pmatrix} 0 & x_{12} & x_{13} & x_{14} \\ -x_{12} & 0 & x_{23} & x_{24} \\ -x_{13} & -x_{23} & 0 & x_{34} \\ -x_{14} & -x_{24} & -x_{34} & 0 \end{pmatrix} \in \Lambda^2(V),$$

define $\varphi(M) := x_{12}x_{34} - x_{13}x_{24} + x_{14}x_{23}$. Then φ is a quadratic form on $\Lambda^2(V)$. Set

$$M_0 := \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \in \Lambda^2(V).$$

where I is the 2×2 identity matrix. Observe that if matrices of G and row vectors of V are written relative to a fixed standard symplectic basis e_1, e_2, f_1, f_2 of V , then $(u, v) := uM_0v^{\mathrm{tr}}$ defines a G -invariant, alternating form on V . In particular, M_0 is left invariant by G under the linear map $G \rightarrow \mathrm{GL}(\Lambda^2(V))$ sending $(v \mapsto vg) \mapsto (M \mapsto g^{\mathrm{tr}}Mg)$. Hence, that map induces an action of G on the quotient space $W := \Lambda^2(V)/\langle M_0 \rangle$ leaving φ invariant (cf. [KIL], p. 45).

For a vector $M \in \Lambda^2(V)$, let $\overline{M} = M\langle M_0 \rangle$ denote its image in W . Fix a basis $\overline{M}_1, \dots, \overline{M}_5$ of the 5-space W and, for each $s \in \mathcal{S}$, compute the 5×5 matrix $s\theta$ relative to $\overline{M}_1, \dots, \overline{M}_5$ representing the element of $\mathrm{GL}(W)$ induced by s on W under the map $\overline{M} \mapsto s^{\mathrm{tr}}\overline{M}s$. Then $s \mapsto s\theta$ defines the desired homomorphism $\theta: G \rightarrow \Omega(W)$.

Timing: $O(\xi \log \log q + \chi + \log^2 q)$ using Theorem 1.1 with $d = 5$.

6.3 $\mathrm{SU}(3, q)$

Let $G = \langle \mathcal{S} \rangle = \mathrm{SU}(V)$, preserving an hermitian form on the 3-space V . We begin our algorithm by constructing an $\mathrm{SU}(2, q)$ subgroup of G .

6.3.1 Finding τ

Exactly as in [KS], 6.6.1, choose up to 64 elements τ of G in order to find one of $\text{ppd}^\#(p; 2k) \cdot \text{ppd}^\#(p; k)$ - or $\text{ppd}^\#(p; 2k) \cdot \text{ppd}^\#(p; k) \cdot \text{ppd}^\#(p; k/2)$ -order, respectively, depending on whether k is odd or even; in addition we require that $|\tau^{2(q-1)}| > 3$ and, if q is a Mersenne or Fermat prime, that 16 divides $|\tau|$. Such a τ is found, with probability $> 1 - 1/2^4$, in time $O(\xi + k \log^2 q)$.

6.3.2 Constructing L

Set $a := \tau^{2(q-1)}$ so that a is a $\text{ppd}^\#(p; 2k)$ -element having an i -dimensional eigenspace V_i for $i = 1, 2$, where $V_2 = V_1^\perp$. The following procedure uses a to construct a subgroup $L \cong \text{SU}(2, q)$ of G .

Procedure: For up to 10 choices $g \in G$ proceed as follows:

Set $b := a^g$, $A := \langle a, b \rangle$ and use Proposition 3.2 to find $L = \langle \mathcal{S}_L \rangle = A'$.

For each $s \in \mathcal{S}_L$, find a 2×2 matrix \tilde{s} representing the element of $\text{SU}(2, q)$ induced by s on $[V, L]$; set $\tilde{L} := \langle \tilde{s} \mid s \in \mathcal{S}_L \rangle$.

Use 6.1.2 to test whether $\tilde{L} \cong \text{SU}(2, q)$ and, if so, to find a new generating set $\mathcal{T}_{\tilde{L}}$ for \tilde{L} behaving as in Theorem 1.1 for \tilde{L} , and stop.

Return A , L and $\mathcal{T}_{\tilde{L}}$ if the latter has been found, or fail otherwise.

Correctness: If z is a point of V_2 , let V_z denote the 2-space $\langle V_1, z \rangle$. For any given $g \in G$, if $V_1^g \neq V_1$ then V_1^g lies on a unique line V_z for some $z = z(g) \in V_2$. If z is singular, then $\langle V_1, V_1^g \rangle = V_z$ has 1-dimensional radical z , and we will not succeed with that choice of g . If $z = V_1^g$ (i.e. if V_1^g lies on $V_2 = V_1^\perp$) then $A \cong \langle a \rangle \times \langle b \rangle$ and we fail again for such g . Otherwise ($V_1^g \neq z$ is nonsingular) A induces on the hyperbolic line $\langle V_1, V_1^g \rangle$ an irreducible subgroup of the general unitary group $\text{GU}(2, q)$ generated by $\text{ppd}^\#(p; 2k)$ elements of the same order. By Lemma 2.9, with probability $> 1/2$, $A' \cong \text{SU}(2, q)$ fixing the nonsingular point $V_2 \cap V_2^g = \langle V_1, V_1^g \rangle^\perp$. Hence, a fixed choice $b = a^g$ gives rise to a suitable A with probability $> (1/2) \cdot (1 - \{q^2(q+1) + q^2 - q + 1\} / \{q^2(q^2 - q + 1)\}) > 0.46$.

Timing: $O(\xi \log \log q + \log^2 q + \chi)$ dominated by the time for 6.1.2 and Lemma 3.2.

Reliability: For a suitable A , Lemma 3.2 correctly finds A' with probability $> 1/2$ and 6.1.2 correctly confirms that $L \cong \text{SU}(2, q)$ with probability $> 3/4$. Hence, a single conjugate b produces a suitable generating set $\mathcal{T}_{\tilde{L}}$ with probability $> (3/8)(0.46) > 0.17$. It follows that at least one of our choices succeeds with probability $> 1 - (0.83)^{10} > 0.75$.

6.3.3 Some elements of L

Taking as input the subgroup L constructed in 6.3.2, the next procedure returns a standard basis of V relative to which the elements of L are easily recognised, together with some elements of L that we will use later in the algorithm.

Procedure:

Use Lemma 3.7 to find a hyperbolic pair e, f in the line $[V, L]$.

Use Lemma 3.6 to find $\langle v \rangle = [V, L]^\perp$.

Use Lemma 3.5(ii) to find $\alpha \in \mathbb{F}^*$ s.t. $\alpha\bar{\alpha} = 1/(v, v)$; $v := v/\alpha$.

$\mathcal{B} := (e, v, f)$ and write elements of G relative to \mathcal{B} .

Use Lemma 3.5(i) to find $0 \neq \delta = -\bar{\delta}$.

Recalling that ζ generates \mathbb{F}_0^* , use 6.1.2 to construct the $k + 2$ elements

$$h := \begin{pmatrix} \zeta & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/\zeta \end{pmatrix}, \quad r := \begin{pmatrix} 0 & 0 & \bar{\delta} \\ 0 & 1 & 0 \\ 1/\delta & 0 & 0 \end{pmatrix}, \quad t_i := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \zeta^{i-1}\delta & 0 & 1 \end{pmatrix}$$

of L using SLPs of length $O(\log q)$ from $\mathcal{T}_{\bar{L}}$, where $1 \leq i \leq k$.

Return h, r , and $\{t_1, \dots, t_k\}$.

Timing: Dominated by $O(k \log q)$ for the construction of the $k + 2$ elements.

Note that $T(\langle e \rangle) = \langle t_i \mid 1 \leq i \leq k \rangle$; $T(\langle f \rangle) = T(\langle e \rangle)^r$; and $\langle h \rangle = N_L(T(x)) \cap N_L(T(y))$ has order $q - 1$.

6.3.4 The subgroup Q

Let $x = \langle e \rangle$ and $T = T(x)$. The following is a Las Vegas procedure to construct a generating set \mathcal{T}_Q for the group $Q = O_p(G_x)$ of order q^3 .

Procedure:

for $i \in \{1, 2\}$

Choose $g_i \in G$ and set $L_i := \langle T, T^{rg_i} \rangle$.

Use 6.1.2 (twice for each i) to test whether or not $L_i \cong \text{SU}(2, q)$ and, if so, to construct

$h_i \in N_{L_i}(T) \cap N_{L_i}(T^{rg_i})$ of order $q - 1$.

Return $\mathcal{T}_Q := \{ t_j, [h, h_i]^{h^{j-1}} \mid i = 1, 2, 1 \leq j \leq k \}$.

Correctness and reliability: It suffices to show that $\langle \mathcal{T}_Q \rangle T/T = Q/T$ with high probability. For fixed i , $L_i \cong \text{SU}(2, q)$ with probability $1 - 1/(q^3 + 1)$. As 6.1.2 is repeated twice for each choice,

it will fail with probability $< 1/16$ for suitable g_i . Hence, we fail to recognise at least one $\mathrm{SL}(2, q)$ -subgroup L_i with probability $< 2\{1 - 15(1 - 1/q^3)/16\} < 0.15$. For $L_i \cong \mathrm{SL}(2, q)$ ($i = 1, 2$), $[h, h_1]$ and $[h, h_2]$ are in the same 1-space of the 2-space Q/T with probability $1/(q+1) < 1/17$. As h induces on Q/T a scalar generating \mathbb{F}_0^* , it follows that $\langle \mathcal{T}_Q \rangle T/T = Q/T$ with probability $> 1 - \{0.15 + 0.85/17\} = 0.8$.

Timing: $O(\xi \log \log q + \chi + \log q)$ for the calls to 6.1.2.

6.3.5 Algorithmic properties of Q

Since the Witt index m is 1, we have $Q = U(x)$ (cf. 2.5.2) so that elements of $Q(x) = O_p(G_x)$ have the form

$$u(\lambda, \nu) = \begin{pmatrix} 1 & 0 & 0 \\ \lambda & 1 & 0 \\ \nu & -\bar{\lambda} & 1 \end{pmatrix},$$

where $\lambda, \nu \in \mathbb{F}$ are such that $\nu + \bar{\nu} + \lambda\bar{\lambda} = 0$. The next result shows that we can efficiently construct any given element of Q .

Lemma 6.1 *In time $O(\log^2 q)$, one can find an SLP of length $O(\log q)$ from \mathcal{T}_Q to any given $u = u(\lambda, \nu) \in Q$.*

Proof. The $2k$ elements $[h, h_i]^{h^{j-1}}$ ($i = 1, 2$ and $1 \leq j \leq k$) form an \mathbb{F}_p -basis for Q/T . Use linear algebra to find integers $0 \leq \alpha_{ij} < p$, and an element $\tilde{u} := \prod_{i=1}^2 \prod_{j=1}^k ([h, h_i]^{h^{j-1}})^{\alpha_{ij}}$, such that $\tilde{u} = u(-\lambda, \nu')$ for some $\nu' \in \mathbb{F}$. Hence write an SLP σ of length $O(\log q)$ from \mathcal{T}_Q to \tilde{u}^{-1} . Now use 6.1.2 to find integers $0 \leq \beta_j < p$ for $1 \leq j \leq k$ such that $u\tilde{u} = \prod_{j=1}^k t_j^{\beta_j}$ and write an SLP τ of length $O(\log q)$ from $\{t_i \mid 1 \leq i \leq k\}$ to $u\tilde{u}$. Then $\sigma\tau$ is the desired SLP. \square

By Theorem 2.3(vii), the subgroup Q acts regularly on the singular points of V not equal to x . The following is an algorithmic version of this transitivity.

Lemma 6.2 *Using only field operations, one can find the unique $u \in Q$ sending y to any given singular point $z \neq x$.*

Proof. Let $x \neq z = \langle \nu, \lambda, 1 \rangle$ be given, where $\nu + \bar{\nu} + \lambda\bar{\lambda} = 0$. Then $u := u(-\bar{\lambda}, \nu) \in Q$, and one easily checks that $\langle 0, 0, 1 \rangle u(-\bar{\lambda}, \nu) = \langle \nu, \lambda, 1 \rangle$. \square

6.3.6 Straightline programs

Set $\mathcal{T} := \mathcal{T}_Q \cup (\mathcal{T}_Q)^r$ for the element r constructed in 6.3.3. The following is the 3-dimensional version of the SLP algorithm described in 5.2.

WriteSLP(g, \mathcal{T})

if $(\langle e \rangle g \neq \langle e \rangle)$ then

Use Lemma 6.2 to find $w \in Q^r$ such that $\langle e \rangle gw = \langle e \rangle$.

Use Lemma 6.1 to write an SLP σ_1 from $(\mathcal{T}_Q)^r$ to w .

$g := gw$.

[now g fixes $\langle e \rangle$]

if $(\langle f \rangle g \neq \langle f \rangle)$ then

Find $u \in Q$ such that $\langle f \rangle gu = \langle f \rangle$.

Write an SLP σ_2 from \mathcal{T}_Q to u .

$g := gu$.

[now g fixes $\langle f \rangle$]

For $g = \text{diag}(\lambda, \lambda^{q-1}, \lambda^{-q})$ set $\gamma := \lambda\delta$.

Use Lemma 3.5(ii) to find $\eta \in \mathbb{F}^*$ s.t. $\eta\bar{\eta} = -\gamma - \gamma^q$.

Use the equation

$$\text{diag}(\lambda, \lambda^{q-1}, \lambda^{-q}) = u(\eta\gamma^{-1}, \gamma^{-q}) \cdot u(\eta^q, \gamma)^{\text{tr}} \cdot u(\eta\gamma^{-q}, \gamma^{-q}) \cdot r,$$

and Lemma 6.1 to write an SLP σ from $\mathcal{T}_Q \cup \{r\}$ to g .

Return the SLP $\sigma(\sigma_1)^{-1}(\sigma_2)^{-1}$.

Timing: $O(\log^2 q)$, dominated by Lemma 6.1.

6.3.7 Total timing and reliability

Considering the timing estimates of the subroutines in 6.3.1 through 6.3.6, and adding up the failure probabilities of the randomised subroutines, we obtain a suitable \mathcal{T} , with probability $> 1/2$, in time $O(\xi \log \log q + \chi + \log^2 q)$.

6.4 SU(4, q)

Let $G = \langle \mathcal{S} \rangle = \text{SU}(V)$ preserving an hermitian form on the 4-space V . The algorithm we present to recognise G has the same basic structure as that for $\text{SU}(3, q)$ in 6.3. We will only give full details for a subroutine if there is no direct analogue in 6.3.

6.4.1 Finding τ

Proceed exactly as in 6.3.1, this time choosing up to 48 elements of G to find τ of $\text{ppd}^\#(p; 2k) \cdot \text{ppd}^\#(p; 6k)$ -order. The timing and reliability are as in 6.3.1.

6.4.2 Constructing L

Set $a := \tau^{q^2-q+1}$, so that a is a $\text{ppd}^\#(p; 2k)$ -element having an i -dimensional eigenspace for $i = 1, 3$ with $V_3 = V_1^\perp$. Now proceed exactly as in 6.3.2 to find $A := \langle a, b \rangle$, $L = \langle \mathcal{S}_L \rangle := A'$, and new generating set $\mathcal{T}_{\tilde{L}}$ for the $\text{SL}(2, q)$ subgroup \tilde{L} induced by L on $V_L = \langle V_1, V_1^g \rangle$. The timing and reliability are as in 6.3.2.

6.4.3 Some elements of L

As in 6.3.3 we find a standard basis \mathcal{B} , this time of the form e_1, e_2, f_1, f_2 , where $\langle e_1, f_1 \rangle = [V, L]$ and $x = \langle e_1 \rangle$ and $y = \langle f_1 \rangle$ are the 1-dimensional eigenspaces of a in V . This time we use 6.1.2 and $\mathcal{T}_{\tilde{L}}$ just to construct generators t_1, \dots, t_k for $T := T(x)$, and the element r which sends $e_1 \mapsto \bar{\delta}f_1$, $f_1 \mapsto e_1/\delta$, $e_2 \mapsto e_2$ and $f_2 \mapsto f_2$.

6.4.4 The subgroup Q

We use a slightly different approach than in 6.3.4 to construct a generating set \mathcal{T}_Q for the subgroup $Q = Q(x)$ of order q^5 .

Procedure:

```

for  $i \in \{1, 2\}$ 
  repeat (at most 12 times)
    Choose  $g_i \in G$  and set  $J_i := \langle T, T^r, T^{g_i} \rangle$ ;
    if ( $[V, J_i]$  is a nonsingular 3-space) then
      for  $s$  a generator of  $J_i$ 
         $\tilde{s} :=$  element of  $\text{SU}(3, q)$  induced by  $s$  on  $[V, J_i]$ .
         $\tilde{J}_i := \langle \tilde{s} \mid s \text{ a generator of } J_i \rangle$ .
        Use 6.3 to test  $\tilde{J}_i \cong \text{SU}(3, q)$ ; stop when such  $J_i$  is found.

    if ( $J_1 \cong \text{SU}(3, q) \cong J_2$  and  $[V, J_1] \neq [V, J_2]$ ) then
       $\mathcal{T}_{Q_i} :=$  generators for  $O_p((J_i)_x)$ .
      Return  $\mathcal{T}_Q := \mathcal{T}_{Q_1} \cup \mathcal{T}_{Q_2}$ .
  else
    Report failure.

```

Correctness: For distinct $\text{SU}(3, q)$ subgroups J_i of G , the groups $O_p((J_i)_x)/T$ are distinct nonsingular 1-spaces of the 2-dimensional \mathbb{F} -space Q/T so that $Q = \langle O_p((J_i)_x) \mid i = 1, 2 \rangle$.

Timing: $O(\xi \log \log q + \chi + \log q)$ for the ≤ 24 calls to 6.3.

Reliability: Since $q \geq 16$, the probability that two $SU(3, q)$ subgroups of G are equal is small enough to be absorbed by our following (crude) estimates. By Lemma 2.10, for fixed i and fixed choice g_i , $J_i \cong SU(3, q)$ with probability $> 1/2$. For such a J_i , the algorithm 6.3 succeeds with probability $> 3/4$. Thus a single choice g_i produces a suitable J_i with probability $> (1/2)(3/4) > 1/4$. Hence, the procedure fails with probability $< (1 - 1/4)^{12} + (1 - 1/4)^{12} < 2/e^3$.

6.4.5 Algorithmic properties of $Q(x)$

There are versions of Lemmas 6.1 and 6.2 for $d = 4$ (having the same timing) to write an SLP from \mathcal{T}_Q to any given element of Q and to find the unique element of Q sending $\langle f_1 \rangle$ to any given $\langle f' \rangle \notin \langle e_1 \rangle^\perp$.

6.4.6 The generating set \mathcal{T}

We do not follow the approach to SLPs taken in 6.3.6, but rather use the general algorithm `WriteSLP` presented in section 5. Accordingly, we now tailor the generating set \mathcal{T} to be consistent with the input to that algorithm.

Use 6.3.6 to construct an element $\sigma \in J_1$ of order $q^2 - 1$ normalising T and $T(\langle f_1 \rangle)$ and inducing the scalar ρ on $\langle e_1 \rangle$ (hence $\bar{\rho}^{-1}$ on $\langle f_1 \rangle$). Also, construct using SLPs from $\mathcal{T}_Q \cup \{r\}$ each of the $2k$ generators

$$\begin{aligned} \mathcal{T}^{(1)} &:= \{ r_1(\rho^i f_2, 0) \mid 0 \leq i < 2k \} \subset Q \\ \mathcal{T}_L^{(2)} &:= \{ r'_1(\rho^i e_2, 0) \mid 0 \leq i < 2k \} \subset Q^r \end{aligned}$$

of the long root subgroups $R(e_1, f_2)$ and $R(f_1, e_2)$ respectively (cf. (9)). Then $\mathcal{T}^{(1)} \cup \mathcal{T}^{(2)}$ is the generating set for L defined in (18). Use Proposition 4.1 to construct an element $c \in \langle \mathcal{T}_L \rangle$ interchanging $\langle e_1 \rangle$ and $\langle e_2 \rangle$ and set

$$\mathcal{T}_E := \mathcal{T}_Q \cup (\mathcal{T}_Q)^c \quad \text{and} \quad \mathcal{T}_F := (\mathcal{T}_Q)^r \cup (\mathcal{T}_Q)^{rc}.$$

Finally, exactly as in 4.6.4, construct the set \mathcal{T}_U and set

$$\mathcal{T} := (\mathcal{T}^{(1)} \cup \mathcal{T}^{(2)}) \cup \mathcal{T}_E \cup \mathcal{T}_F \cup \mathcal{T}_{\bar{L}} \cup \mathcal{T}_U,$$

exactly as in the main algorithm (but with $\mathcal{T}_{\bar{L}}$ replacing \mathcal{T}_K).

Timing: $O(\xi \log \log q + \log^2 q + \chi)$ to construct the groups $R(e_1, f_2)$ and $R(f_1, e_2)$ and for the call to 6.1.2.

6.4.7 Straightline programs

We use `WriteSLP` (cf. section 5) for case **U** ($d = 4$) to write an SLP of length $O(\log q)$ from \mathcal{T} to any given element $g \in G$. Note that the algorithm `WriteLSLP` (cf. 5.1.1) reduces to the subgroup of K_{e_2, f_2} , which is our group L .

Table 13: Running times for input groups $\Omega^-(d, 2^k)$

		d					
		10	20	30	40	50	60
q	2^4	4	4	6	11	18	26
	2^6	1	2	5	12	31	47
	2^8	1	2	4	11	20	57
	2^{10}	2	5	21	75	159	511
	2^{12}	4	7	34	75	212	430
	2^{14}	17	28	45	105	182	544

Table 14: Running times for input groups $\Omega^-(d, 3^k)$

		d					
		10	20	30	40	50	60
q	3^3	2	2	6	13	29	64
	3^4	2	2	6	15	45	80
	3^5	1	2	6	16	37	101
	3^6	2	4	23	61	245	589
	3^7	2	5	21	74	252	646
	3^8	4	9	35	97	328	583

Table 15: Running times for input groups $\Omega^-(d, p)$

		d					
		10	20	30	40	50	60
p	257	1	3	17	75	131	264
	6563	5	7	22	121	369	723
	8191	9	9	27	92	332	889

Table 16: Performance comparisons for $d = 10$ over fields of size 2^i

Implementation	2^4	2^6	2^8	2^{10}	2^{12}
ClassicalConstructiveRecognition	9	5	5	11	20
Sp(d, q) in “matrix” share package	2	7	21	137	656

6.4.8 Total timing and reliability

A generating set \mathcal{T} for G , behaving as in Theorem 1.1, is obtained with probability $> 1 - (1/4 + 2/e^3) > 1/2$, in time $O(\xi \log \log q + \chi + \log^2 q)$.

7 Performance tests

The author has implemented the algorithm in the computer algebra system GAP4 [GAP4]. Tables 13, 14 and 15 show runtimes (in seconds of CPU time, rounded to the nearest second) for a series of performance tests using the groups $\Omega^-(d, q)$. The stated timings are an average of 20 runs, obtained using an implementation of GAP4 on a 2×PIII, 933 MHz, running RedHat Linux 7.1, 2.4.12 SMP kernel.

To illustrate the performance of our algorithm when handling large fields, we also obtained a comparison with an implementation of Celler’s Sp(d, q) algorithm [Ce] which is available in the GAP3 share package “matrix”. The average termination times of the two implemetations for 20 runs on a Sun UltraSPARC-II, 359 MHz, using a 10 dimensional group are summarised in Table 16.

Acknowledgements: The author would like to thank Bill Kantor for many helpful discussions during the preparation of this manuscript, and Alexander Hulpke and Ákos Seress for their advice during the implementation and testing of the algorithm. The suggestions for improvement provided by both referees were also very much appreciated.

References

- [Ba] L. Babai, *Local expansion of vertex-transitive graphs and random generation in finite groups*, pp. 164–174 in: Proc. ACM Symp. on Theory of Computing, 1991.
- [Ba+] L. Babai, G. Cooperman, L. Finkelstein, E. M. Luks, Á Seress, *Fast Monte Carlo algorithms for permutation groups*, J. Comp. Syst. Sci. 50, 296–308, 1995.
- [Br1] P. A. Brooksbank, *A constructive recognition algorithm for the matrix group $\Omega(d, q)$* , pp. in: Groups and Computation III, Ohio State Univ. Math. Res. Inst. Publ. (W. M. Kantor and Á. Seress eds), Walter de Gruyter, Berlin-New York, 2001.
- [Br2] P. A. Brooksbank, *Constructive recognition of the finite simple classical groups*, Ph. D. thesis, U. Oregon, 2001.
- [Ce] F. Celler, *Matrixgruppenalgorithmen in GAP*, Ph. D. thesis, RWTH Aachen, 1997.
- [CeLG] F. Celler, C. R. Leedham-Green, *A constructive recognition algorithm for the special linear group* pp. in: The Atlas of Finite Groups: Ten Years On (Birmingham 1995), London Math. Soc. Lecture Note Ser. 249, 1998.
- [Ce+] F. Celler, C. R. Leedham-Green, S. H. Murray, A. C. Niemeyer, E. A. O’Brien, *Generating random elements of a finite group*, Comm. in Alg. 23, 4931–4948, 1995.
- [CoLG] M. Conder, C. R. Leedham-Green, *Fast recognition of classical groups over large fields*, pp. in: Groups and Computation III, Ohio State Univ. Math. Res. Inst. Publ. (W. M. Kantor and Á. Seress eds), Walter de Gruyter, Berlin-New York, 2001.
- [GAP4] The GAP Group, *Groups, Algorithms, and Programming, Version 4.2*; Aachen, St Andrews. (<http://www-gap.dcs.st-and.ac.uk/~gap>), 2000.
- [H+] D. F. Holt, C. R. Leedham-Green, E. A. O’Brien, S. Rees, *Computing matrix group decompositions with respect to a normal subgroup*, J. Algebra, 184, 795–817, 1996.
- [HR] D. F. Holt, S. Rees, *Testing modules for irreducibility*, J. Austral. Math. Soc. (Ser. A), 57, 1–16, 1994.

- [KLi] W. M. Kantor, R. A. Liebler, *The rank 3 permutation representations of the finite classical groups*, Trans. AMS, 271, 1–71, 1982.
- [KS] W. M. Kantor, Á Seress, *Black box classical groups*, Mem. AMS, 149, Number 708, 2001.
- [KIL] P. B. Kleidman, M. W. Liebeck, *The subgroup structure of the finite classical groups*, LMS Lecture Note Series 129, Cambridge U. Press, 1990.
- [LG] C. R. Leedham-Green, *The computational matrix group project*, pp. in: Groups and Computation III, Ohio State Univ. Math. Res. Inst. Publ. (W. M. Kantor and Á. Seress eds), Walter de Gruyter, Berlin-New York, 2001.
- [LGO1] C. R. Leedham-Green, E. A. O’Brien, *Recognising tensor products of matrix groups*, Intern. J. Algebra Comp. 7, 541–559, 1997.
- [LGO2] C. R. Leedham-Green, E. A. O’Brien, *Constructive recognition of $SL(2, q)$* (in preparation).
- [LN] R. Lidl, H. Niederreiter, *Finite Fields*, Encyclopedia of Math. and its Applications 20, Addison-Wesley, 1983.
- [NeP] P. M. Neumann, C. E. Praeger, *A recognition algorithm for special linear groups*, Proc. LMS (3), 65, 555–603, 1992.
- [NiP] A. C. Niemeyer, C. E. Praeger, *A recognition algorithm for classical groups over finite fields*, Proc. LMS (1), 77, 117–169, 1998.
- [Se] Á. Seress, *Permutation group algorithms*, Cambridge U. Press (to appear)
- [Ta] D. E. Taylor, *The geometry of the classical groups*, Heldermann, Berlin, 1992.
- [Zs] K. Zsigmondy, *Zur Theorie der Potenzreste*, Monatsh. Math, Phys. 3, 263–284, 1892.

Department of Mathematics
The Ohio State University
231 West 18th Avenue
Columbus, OH 43210
U.S.A