Algorithmic problems in twisted groups of Lie type

Jonas Henrik Ambjörn Bäärnhielm

A thesis submitted for the degree of
Doctor of Philosophy
at Queen Mary, University of London

Queen Mary
University of London

Declaration

I hereby declare that, to the best of my knowledge, the material contained in this thesis is original and my own work, except where otherwise indicated, cited, or commonly-known.

I have not submitted any of this material in partial of complete fulfilment of requirements for another degree at this or any other university.

Abstract

This thesis contains a collection of algorithms for working with the twisted groups of Lie type known as *Suzuki* groups, and small and large *Ree* groups.

The two main problems under consideration are *constructive recognition* and *constructive membership testing*. We also consider problems of generating and conjugating Sylow and maximal subgroups.

The algorithms are motivated by, and form a part of, the Matrix Group Recognition Project. Obtaining both theoretically and practically efficient algorithms has been a central goal. The algorithms have been developed with, and implemented in, the computer algebra system MAGMA.

Contents

List of Figures

Acknowledgements

I would first of all like to thank my supervisor, Charles Leedham-Green, for endless help and encouragement. I would also like to thank the following people, since they have helped me to various extent during the work presented in this thesis: John Bray, Peter Brooksbank, John Cannon, Sergei Haller, Derek Holt, Alexander Hulpke, Bill Kantor, Ross Lawther, Martin Liebeck, Klaus Lux, Frank Lübeck, Scott Murray, Eamonn O'Brien, Geoffrey Robinson, Colva Roney-Dougal, Alexander Ryba, Ákos Seress, Leonard Soicher, Mark Stather, Bill Unger, Maud de Visscher, Robert Wilson.

Notation

$\bar{F}$	Algebraic closure of the field F
g^h	conjugate of g by h, *i.e.* $g^h = h^{-1}gh$
$[g, h]$	commutator of g and h, *i.e.* $[g, h] = g^{-1}h^{-1}gh$
$\mathrm{Tr}(g)$	trace of the matrix g
C_n	cyclic group of order n, *i.e.* $\mathrm{C}_n \cong \mathbb{Z}/n\mathbb{Z}$
$\mathbb{F}_q$	finite field of size q (or its additive group)
$\mathbb{F}_q^\times$	multiplicative group of $\mathbb{F}_q$
$\xi(d)$	The number of field operations required by a random element oracle for $\mathrm{GL}(d, q)$.
ξ	The number of field operations required by a random element oracle for $\mathrm{GL}(k, q)$ with k constant
$\chi_D(q)$	number of field operations required by a discrete logarithm oracle in $\mathbb{F}_q$
$\chi_F(d, q)$	number of field operations required by an integer factorisation oracle (which factorises $q^i - 1$, for $1 \leqslant i \leqslant d$)
$\mathrm{Mat}_n(R)$	matrix algebra of $n \times n$ matrices over ring R
I_n	identity $n \times n$ matrix
$E_{i,j}$	matrix with 1 in position (i, j) and 0 elsewhere
$\lvert g\rvert$	order of a group element g
ϕ	Euler totient function
$\sigma_0(n)$	number of positive divisors of $n \in \mathbb{N}$ (where $\sigma_0(n) < 2^{(1+\varepsilon)\log_{\mathrm{e}}(n)/\log\log_{\mathrm{e}}(n)}$)
ψ	Frobenius automorphism in a field F (*i.e.* $\psi : x \mapsto x^p$ where $\mathrm{char}(F) = p$)
$\Phi(G)$	Frattini subgroup of G (intersection of all maximal subgroups of G)
$\mathrm{O}_p(G)$	largest normal p-subgroup of G
G_P	stabiliser in G of the point P
G'	derived subgroup (commutator subgroup) of G
$\mathrm{N}_G(H)$	normaliser of H in G
$\mathrm{C}_G(g)$	centraliser of g in G
$\mathrm{Z}(G)$	centre of G
$\mathrm{End}_H(M)$	algebra of H-endomorphisms of G-module M, where $H \leqslant G$
$\mathrm{Aut}(M)$	automorphism group of G-module M (if $G \leqslant \mathrm{GL}(d, q) = H$ then $\mathrm{Aut}(M) = \mathrm{C}_H(G)$)
$\mathcal{S}^2(M)$	symmetric square of module M over a field F (where $M \otimes M = \mathcal{S}^2(M) \oplus \wedge^2(M)$ if $\mathrm{char}(F) > 2$)
$\wedge^2(M)$	exterior square of module M

$\mathrm{Sym}(O)$	symmetric group on the set O
$\mathrm{Sym}(n)$	symmetric group on n points
D_n	dihedral group of order n
$\mathbb{P}(V)$	projective space corresponding to vector space V
$\mathbb{P}^n(F)$	n-dimensional projective space over the field F
$[G : H]$	index of H in G
$G.H$	extension of G by H (a group E such that $G \trianglelefteq E$ and $E/G \cong H$)
$G{:}H$	split extension of G by H (as with $G.H$ but E also has a subgroup $H_0 \cong H$ such that $E = GH_0$ and $G \cap H_0 = \langle 1 \rangle$)
$\mathrm{O}(\cdot)$	standard time complexity notation
Ψ	Automorphisms defining $\mathrm{Sz}(q)$ or $\mathrm{Ree}(q)$
φ, θ, ρ	group homomorphisms

CHAPTER 1

Introduction and preliminaries

1.1. Introduction

This thesis contains algorithms for some computational problems involving a few classes of the finite simple groups. The main focus is on providing efficient algorithms for constructive recognition and constructive membership testing, but we also consider the conjugacy problem for Sylow and maximal subgroups.

The work is in the area of *computational group theory* (abbreviated CGT), where one studies algorithmic aspects of groups, or solves group theoretic problems using computers. A group can be represented as a data structure in several ways, and perhaps the most important ones are *permutation groups*, *matrix groups* and *finitely presented groups*.

The permutation group setting has been studied since the early 1970's, and the basic technique which underlies most algorithms is the construction of a *base* and a *strong generating set*. If G is a permutation group of degree n, this involves constructing a descending chain of subgroups of G, where each subgroup in the chain has index at most n in its predecessor. The permutation group algorithm machinery is summarised in [**Ser03**].

For matrix groups, the classical method is also to construct a base and strong generating set. However, in general a matrix group has no faithful permutation representation whose degree is polynomial in the size of the input. Hence the indices in the subgroup chain will be too large and the permutation group algorithms will not be efficient. For example, $\mathrm{SL}(d, q)$ has no proper subgroup of index less than $(q^d - 1)/(q - 1)$, and q is exponential in the size of the input, since a matrix has size $\mathrm{O}\big(d^2 \log(q)\big)$.

Historically there were two schools within CGT. One consists of people with a more computational complexity background, whose primary goal was to find theoretically good (polynomial time) algorithms. The implementation and practical performance of the algorithms were less important, since the computational complexity view, based on many real examples, is that if the "polynomial barrier" is broken, then further research will surely lead also to good practical algorithms. The other school consists of people with a more group theoretic background, whose primary goal was to solve computational problems in group theory (historically often one-time problems with the sporadic groups), and hence to develop algorithms that can be easily implemented and that run fast on the current hardware and on the specific input in question. The asymptotic complexity of the algorithms was less important, and perhaps did not even make sense in the case of sporadic groups.

The distinction between these schools has become less noticeable during the last 15 years, but during that time there has also been much work on algorithms for matrix groups, and there are two main approaches that roughly correspond to these two schools. The first approach is the "black box" approach, that considers the matrix groups as *black box groups* (see [**Ser03**, pp. 17]). This was initiated by [**Luk92**] (but goes back to [**BS84**]) and much of it is summarised in [**BB99**]. The other approach is the "geometric" approach, also known as *The Matrix Group Recognition Project* (abbreviated MGRP), whose underlying idea is to use a famous theorem of Aschbacher[**Asc84**] which roughly says that a matrix group either preserves some geometric structure, or is a simple group.

Although the author has a background perhaps more in the computational complexity school, the work in this thesis forms a part of MGRP. The first specific goal in that approach is to obtain an efficient algorithm that finds a composition series of a matrix group. There exists a recursive algorithm[**LG01**] for this, which relies on a number of other algorithms that determine which kind of geometric structure is preserved, and the base cases in the recursion are the classes of finite simple groups.

Hence this algorithm reduces the problem of finding a composition series to various problems concerning the composition factors of the matrix group, which are simple groups. The work presented here is about computing with some of these simple groups.

For each simple group, a number of problems arise. The simple group is given as $G = \langle X \rangle \leqslant \mathrm{PGL}(d, q)$ for some d, q and we need to consider the following problems:

(1) The problem of *recognition* or *naming* of G, *i.e.* decide the name of G, as in the classification of the finite simple groups.
(2) The *constructive membership* problem. Given $g \in \mathrm{PGL}(d, q)$, decide whether or not $g \in G$, and if so express g as a straight line program in X.
(3) The problem of *constructive recognition*. Construct an isomorphism φ from G to a *standard copy* H of G such that $\varphi(g)$ can be computed efficiently for every $g \in G$. Such an isomorphism is called *effective*. Also of interest is to construct an effective inverse of φ^{-1}, which essentially is constructive membership testing.

To find a composition series using [**LG01**], the problems involving the composition factors that we need to solve are naming and constructive membership. However, the effective isomorphisms of these composition factors to standard copies can also be very useful. Given such isomorphisms, many problems, sometimes including constructive membership, can be reduced to the standard copies. Hence these isomorphisms play a central role in computing with a matrix group once a composition series has been constructed.

In general, the constructive recognition problem is computationally harder than the constructive membership problem, which in turn is harder than the naming problem. Most of our efforts will therefore go towards solving constructive recognition. In fact, the naming problem is not considered in every case here, since the algorithm of [**BKPS02**] solves this problem. However, that algorithm is a Monte Carlo algorithm, and in some cases we can improve on that and provide Las Vegas algorithms (see Section 1.2.2).

The algorithms presented here are not *black box* algorithms, but rely heavily on the fact that the group elements are matrices, and use the representation theory of the groups in question. However, the algorithms will work with all possible representations of the groups, so a user of the algorithms can consider them black box in that sense.

1.2. Preliminaries

We now give preliminary discussions and results that will be necessary later on.

1.2.1. Complexity. We shall be concerned with the time complexity of the algorithms involved, where the basic operations are the field operations, and not the bit operations. All simple arithmetic with matrices can be done using $\mathrm{O}(d^3)$ field operations, and raising a matrix to the $\mathrm{O}(q)$ power can be done using $\mathrm{O}(d^3)$ field operations using [**CLG97**]. These complexity bounds arise when using the naive matrix multiplication algorithm, which uses $\mathrm{O}(d^3)$ field operations to multiply two $d \times d$ matrices. More efficient algorithms for matrix multiplication do exist. Some are also fast in practice, like the famous algorithm of Strassen[**Str69**], which uses $\mathrm{O}(d^{\log_2 7})$ field operations. Currently, the most efficient matrix multiplication algorithm is the Coppersmith-Winograd algorithm of [**CW90, CKSU05**], which uses $\mathrm{O}(d^{2.376})$ field operations, but it is not practical. The improvements made by these algorithms over the naive matrix multiplication algorithm are not noticeable in practice for the matrix dimensions that are currently within range in the MGRP. Therefore we will only use the naive algorithm, which also simplifies the complexity statements.

When we are given a group $G \leqslant \mathrm{GL}(d, q)$ defined by a set X of generators, the size of the input is $\mathrm{O}(|X|\, d^2 \log(q))$. A field element takes up $\mathrm{O}(\log(q))$ space, and a matrix has d^2 entries.

We shall often assume an oracle for the discrete logarithm problem in $\mathbb{F}_q$ (see [**vzGG03**, Section 20.3] and [**Shp99**, Chapter 3]). In the general discrete logarithm problem, we consider a cyclic group G of order n. The input is a generator α of G, and some $x \in G$. The task is to find $1 \leqslant k < n$ such that $\alpha^k = x$. In $\mathbb{F}_q$ the multiplicative group $\mathbb{F}_q^\times$ is cyclic, and the discrete logarithm problem turns up. It is a famous and well-studied problem in theoretical computer science and computational number theory, and it is unknown if it is **NP**-complete or if it is in **P**, although the latter would be very surprising. Currently the most efficient algorithm has sub-exponential complexity. There are also algorithms for special cases, and an important case for us is when $q = 2^n$. Then we can use Coppersmith's algorithm of [**Cop84, GM93**], which is much faster in practice than the general algorithms. It is not polynomial time, but has time complexity $\mathrm{O}\big(\exp(cn^{1/3}\log(n)^{2/3})\big)$, where $c > 0$ is a small constant. We shall assume that the discrete logarithm oracle in $\mathbb{F}_q$ uses $\mathrm{O}\big(\chi_D(q)\big)$ field operations.

Similarly we will sometimes assume an oracle for the integer factorisation problem (see [**vzGG03**, Chapter 19]), whose status in the complexity hierarchy is similar to the discrete logarithm problem. More precisely, we shall assume we have an oracle that, given $d \geqslant 1$ and $\mathbb{F}_q$, factorises all the integers $q^i - 1$ for $1 \leqslant i \leqslant d$, using $\mathrm{O}\big(\chi_F(d, q)\big)$ field operations. By [**BB99**, Theorem 8.2], assuming the Extended Riemann Hypothesis this is equivalent to the standard integer factorisation problem. The reason for having this slightly different factorisation oracle will become clear in Section 1.2.5.

Except for these oracles, our algorithms will be polynomial time, so from a computational complexity perspective our results will imply that the problems we study can be reduced to the discrete logarithm problem or the integer factorisation problem. This is in line with MGRP, whose goal from a complexity point of view is to prove that computations with matrix groups are not harder than (and hence equally hard as) these two well-known problems.

1.2.2. Probabilistic algorithms. The algorithms we consider are probabilistic of the types known as *Monte Carlo* or *Las Vegas* algorithms. These types of algorithms are discussed in [**Ser03**, Section 1.3] and [**HEO05**, Section 3.2.1]. In short, a Monte Carlo algorithm for a language X is a probabilistic algorithm with an input parameter $\varepsilon \in (0, 1)$ such that on input x

- if $x \in X$ then the algorithm returns `true` with probability at least $1 - \varepsilon$ (otherwise it returns `false`),
- if $x \notin X$ then the algorithm returns `false`.

The parameter ε is therefore the maximum error probability. This type of algorithm is also called *one-sided Monte Carlo algorithm* with no *false negatives.* The languages with such algorithms form the complexity class **RP**. In the same way one can define algorithms with no false positives, and the corresponding languages form the class **co-RP**. The class **ZPP** = **RP** $\cap$ **co-RP** consists of the languages that have Las Vegas algorithms, and these are the type of algorithms that we will be most concerned with. A Las Vegas algorithm either returns `failure`, with probability at most ε, or otherwise returns a correct result. Such an algorithm is easily contructed given a Monte Carlo algorithm of each type. The time complexity of a Las Vegas algorithm naturally depends on ε.

Las Vegas algorithms can be presented concisely as probabilistic algorithms that either return a correct result, with probability bounded below by $1/p(n)$ for some polynomial $p(n)$ in the size n of the input, or otherwise return `failure`. By enclosing such an algorithm in a loop that iterates $\lceil \log \varepsilon / \log (1 - 1/p(n)) \rceil$ times, we obtain an algorithm that returns `failure` with probability at most ε, and hence is a Las Vegas algorithm in the above sense. Clearly if the enclosed algorithm is polynomial time, the Las Vegas algorithm is polynomial time.

One can also enclose the algorithm in a loop that iterates until the algorithm returns a correct result, thus obtaining a probabilistic time complexity, and the expected number of iterations is then $\mathrm{O}\big(p(n)\big)$. This is the way we present Las Vegas algorithms since it is the one that is closest to how the algorithm is used in practice.

1.2.3. Straight line programs. For constructive membership testing, we want to express an element of a group $G = \langle X \rangle$ as a word in X. Actually, it should be a *straight line program*, abbreviated to `SLP`. If we express the elements as words, the length of the words might be too large, requiring exponential space complexity.

An `SLP` is a data structure for words, which ensures that during evaluation, subwords occurring multiple times are not computed more often than during construction. Often we want to express an element as an `SLP` in order to obtain its homomorphic image in another group $H = \langle Y \rangle$ where $|X| = |Y|$. The evaluation time for the `SLP` is then bounded by the time to construct it times the ratio of the time required for a group operation in H and in G.

Formally, given a set of generators X, an `SLP` is a sequence $(s_1, s_2, \dots, s_n)$ where each s_i represents one of the following

- an $x \in X$
- a product $s_j s_k$, where $j, k < i$
- a power s_j^n where $j < i$ and $n \in \mathbb{Z}$
- a conjugate $s_j^{s_k}$ where $j, k < i$

so s_i is either a pointer into X, a pair of pointers to earlier elements of the sequence, or a pointer to an earlier element and an integer.

To construct an `SLP` for a word, one starts by listing pointers to the generators of X, and then builds up the word. To evaluate the `SLP`, go through the sequence and perform the specified operations. Since we use pointers to the elements of X, we can immediately evaluate the `SLP` on Y, by just changing the pointers so that they point to elements of Y.

1.2.4. Solving polynomial equations. One of the main themes in this work is that we reduce search problems in computational group theory to the problem of solving polynomial equations over finite fields. The method we use to find the solutions of a system of polynomial equations is the classical resultant technique, described in [**vzGG03**, Section 6.8]. For completeness, we also state the corresponding result for univariate polynomials.

THEOREM 1.1. *Let $f \in \mathbb{F}_q[x]$ have degree d. There exists a Las Vegas algorithm that finds all the roots of f that lie in $\mathbb{F}_q$. The expected time complexity is $\mathrm{O}\big(d(\log d)^2 \log\log(d) \log(dq)\big)$ field operations.*

PROOF. Immediate from [**vzGG03**, Corollary 14.16]. □

THEOREM 1.2. *Let* $f_1, \dots, f_k \in \mathbb{F}_q[x, y] = R$ *be such that the ideal* $I = \langle f_1, \dots, f_k \rangle \trianglelefteqslant R$ *is zero-dimensional. Let* $n_x = \max_i \deg_x f_i \geqslant \max_i \deg_y f_i = n_y$. *There exists a Las Vegas algorithm that finds the corresponding affine variety* $V(I) \subset \mathbb{F}_q^2$. *The expected time complexity is* $\mathrm{O}\big(kn_x^3(\log n_x)^2 \log\log(n_x)\log(n_x q)\big)$ *field operations.*

PROOF. Following [**vzGG03**, Section 6.8], we compute $k-1$ pairwise resultants of the f_i with respect to y to obtain $k-1$ univariate polynomials in x. By [**vzGG03**, Theorem 6.37], the expected time complexity will be $\mathrm{O}\big(k(n_x n_y^2 + n_x^2 n_y)\big)$ field operations and the resultants will be non-zero since the ideal is zero-dimensional.

We can find the set X_1 of roots of the first polynomial, then find the roots X_2 of the second polynomial and simultaneously find $X_1 \cap X_2$. By continuing in the same way we can find the set X of common roots of the resultants. Since their degrees will be $\mathrm{O}\big(n_x^2\big)$, by Theorem 1.1 we can find X using $\mathrm{O}\big(kn_x^2(\log\,(n_x^2))^2 \log\log(n_x^2)\log(n_x^2 q)\big)$ field operations. Clearly $|X| \in \mathrm{O}\big(n_x^2\big)$.

We then substitute each $a \in X$ into the k polynomials and obtain univariate polynomials $f_1(a, y), \dots, f_k(a, y)$. These will have degrees $\mathrm{O}\big(n_y\big)$ and as above we find the set Y_a of their common roots using $\mathrm{O}\big(kn_y(\log\,(n_y))^2 \log\log(n_y)\log(n_y q)\big)$ field operations. Clearly $V(I) = \{(a, b) \mid a \in X, b \in Y_a\}$ and hence we can find $V(I)$ using $\mathrm{O}\big(|X|\, kn_y(\log\,(n_y))^2 \log\log(n_y)\log(n_y q)\big)$ field operations. Thus the time complexity is as stated. □

The following result is a generalisation of the previous result, and we omit the proof, since it is complicated and outside the scope of this thesis.

THEOREM 1.3. *Let* $f_1, \dots, f_k \in \mathbb{F}_q[x_1, \dots, x_k] = F$ *be such that the ideal* $I = \langle f_1, \dots, f_k \rangle \trianglelefteqslant F$ *is zero-dimensional. Let* $n = \max_{i,j} \deg_{x_j} f_i$. *There exists a Las Vegas algorithm that finds the corresponding affine variety* $V(I) \subset \mathbb{F}_q^k$. *The expected time complexity is* $\mathrm{O}\big((kn^3 + n^k k^2(\log n)^2 \log\log(n^k)\log(n^k q))^k\big)$ *field operations.*

As can be seen, if the number of equations, the number of variables and the degree of the equations are constant, then the variety of a zero-dimensional ideal can be found in polynomial time. That is the situation we will be most concerned with. As an alternative to the resultant technique, one can compute a Gröbner basis and then find the variety. By [**LL91**], if the ideal is zero-dimensional the time complexity is $\mathrm{O}\big(n^{\mathrm{O}(k)}\big)$ where n is the maximum degree of the polynomials and k is the number of variables. Hence in the situation above this will be polynomial time.

1.2.5. Orders of invertible matrices. The *order* of a group element g is the smallest $k \in \mathbb{N}$ such that $g^k = 1$. We denote the order of g by $|g|$. For elements g of a matrix group $G \leqslant \mathrm{GL}(d, q)$, an algorithm for finding $|g|$ is presented in [**CLG97**]. In general, to obtain the *precise* order, this algorithm requires a factorisation of

$q^i - 1$ for $1 \leqslant i \leqslant d$, otherwise it might return a multiple of the correct order. Therefore it depends on the presumably difficult problem of integer factorisation, see [**vzGG03**, Chapter 19].

However, in most of the cases we will consider, it will turn out that a multiple of the correct order will be sufficient. For example, at certain points in our algorithms we shall be concerned with finding elements of order *dividing* $q-1$. Hence if we use the above algorithm to find the order of $g \in G$ and it reports that $|g| = q-1$, this is sufficient for us to use g, even though we might have $|g|$ properly dividing $q-1$. Hence integer factorisation is avoided in this case.

In [**BB99**, Section 8] the concept of *pseudo-order* is defined. A pseudo-order of an element g is a product of primes and pseudo-primes. A pseudo-prime is a composite factor of $q^i - 1$ for some $i \leqslant d$ that cannot conveniently be factorised. The order of g is a factor of the pseudo-order in which each pseudo-prime is replaced by a non-identity factor of that pseudo-prime. Hence a pseudo-prime is not a multiple of a "known" prime, and any two pseudo-primes are relatively prime.

The algorithm of [**CLG97**] can also be used to obtain a pseudo-order, and for this it has time complexity $\mathrm{O}\big(d^3 \log(q) \log\log(q^d)\big)$ field operations. In fact, the algorithm computes the order factorised into primes and pseudo-primes. However, even if we have just the pseudo-order, we can still determine if a given prime divides the order, without integer factorisation.

PROPOSITION 1.4. *Let $G \leqslant \mathrm{GL}(d,q)$. There exists a Las Vegas algorithm that, given $g \in G$, a prime $p \in \mathbb{N}$ and $e \in \mathbb{N}$, determines if $p^e \mid |g|$ and if so finds the power of g of order p^e, using $\mathrm{O}\big(d^3 \log(q) \log\log(q^d)\big)$ field operations.*

PROOF. Use [**CLG97**] to find a pseudo-order n of g. Assume that $n = p^k s$ where $p \nmid s$, and $|g| = p^l r$ where $p \nmid r$, $l \leqslant k$ and $r \mid s$. Since p is given, [**CLG97**] will make sure that $k = l$, so we assume that this is the case. Moreover, from [**CLG97**] we see $\gcd(r, s/r) = 1$. If a prime p_1 divides s/r we must have $p_1 \neq p$. So if $p_1 \mid |g|$ we must have $p_1 \mid r$ and hence $p_1 = 1$. Thus $\gcd(s/r, |g|) = 1$.

Hence $\left|g^{s/r}\right| = |g|$, $|g^r| = p^l$ and $|g^s| = \left|(g^{s/r})^r\right| = p^l$. Therefore $g^{sp^{l-e}} \neq 1$ if and only if $p^e \mid |g|$. Then $g^{sp^{l-e}}$ has order p^e. ☐

1.2.6. Random group elements. Our analysis assumes that we can construct uniformly (or nearly uniformly) distributed random elements of a group $G = \langle X \rangle \leqslant \mathrm{GL}(d,q)$. The algorithm of [**Bab91**] produces independent nearly uniformly distributed random elements, but it is not a practical algorithm. It has a preprocessing step with time complexity $\mathrm{O}\big(\log|G|^5\big)$ group operations, and each random element is then found using $\mathrm{O}\big(\log|G|\big)$ group operations.

A more commonly used algorithm is the *product replacement* algorithm of [**CLGM$^+$95**]. It also consists of a preprocessing step, which is polynomial time by [**Pak00**], and each random element is then found using a constant number of group operations (usually 2). This algorithm is practical and included in MAGMA and GAP. Most of the theory about it is summarised in [**Pak01**]. For a discussion of both these algorithms, see [**Ser03**, pp. 26-30].

We shall assume that we have a random element oracle, which produces a uniformly random element of $\langle X \rangle$ using $\mathrm{O}\big(\xi(d)\big)$ field operations, and returns it as an SLP in X.

An important issue is the length of the SLPs that are computed. The length of the SLPs must be polynomial, otherwise it would not be polynomial time to evaluate them. We assume that SLPs of random elements have length $\mathrm{O}\big(n\big)$ where n is the number of random elements that have been selected so far during the execution of the algorithm.

In [**LGM02**], a variant of the product replacement algorithm is presented that finds random elements of the normal closure of a subgroup. This will be used here to find random elements of the derived subgroup of a group $\langle X \rangle$, using the fact that this is precisely the normal closure of $\langle [x, y] : x, y \in X \rangle$.

1.2.7. Constructive recognition overview. If V is an FG-module for some group G and field F, with action $f : V \times FG \to V$, and if Φ is an automorphism of G, denote by V^{Φ} the FG-module which has the same elements as V and where the action is given by $(v, g) \mapsto f(v, \Phi(g))$ for $g \in G$ and $v \in V^{\Phi}$, extended to FG by linearity. We call V^{Φ} a *twisted version* of V, or V *twisted by* Φ. If G is a matrix group and the automorphism Φ is a field automorphism, we call it a *Galois twist*.

From [**HEO05**, Section 7.5.4] we know that G preserves a classical (non-unitary) form if and only if V is isomorphic to its dual. We shall use this fact occasionally.

When we say that an algorithm is "given a group $\langle X \rangle$", then the generating set X is fixed and known to the algorithm. In other words, the algorithm is given the generating set X and will operate in $\langle X \rangle$.

Definition 1.5. Let G, H be matrix groups. An isomorphism $\varphi : G \to H$ is *effective* if there exists a polynomial time Las Vegas algorithm that computes $\varphi(g)$ for any given $g \in G$.

Of course, an effective isomorphism might be deterministic, since $\mathbf{P} \subseteq \mathbf{ZPP}$.

Definition 1.6. The problem of *constructive recognition* is:

Input: A matrix group $G = \langle X \rangle$ with standard copy $H \cong G$.
Ouput: An effective isomorphism $\varphi : G \to H$, such that φ^{-1} is also effective.

Now consider an exceptional group with standard copy $H \leqslant \mathrm{GL}(d, \mathbb{F}_q)$, where $\mathbb{F}_q$ has characteristic p. The standard copies of the exceptional groups under consideration will be defined in Chapter 2. Our algorithms should be able to constructively recognise any input group $G \leqslant \mathrm{GL}(d', q')$ that is isomorphic to H.

The assumptions that we make on the input group G are:

(1) G acts absolutely irreducibly on $\mathbb{F}_{q'}^{d'}$,
(2) G is written over the minimal field modulo scalars,

(3) G is known to be isomorphic to H, and hence d and q are known.

A user of our algorithms can easily first apply the algorithms of [**HR94**] and [**GLGO06**], described in Sections 1.2.10.1 and 1.2.10.2, to make the group satisfy the first two assumptions. The last two assumptions remove much of the need for input verification using non-explicit recognition. They are motivated by the context in which our algorithms are supposed to be used. The idea is that our algorithms will serve as a base case for the algorithm of [**LG01**] or a similar algorithm. In the base case it will be known that the group under consideration is almost simple modulo scalars. We can then assume that the algorithm can decide if it is dealing with a group of Lie type. Then it can use the Monte Carlo algorithm of [**OL05**] to determine the defining characteristic of the group, and next use the Monte Carlo algorithm of [**BKPS02**] to determine the name of the group, as well as the defining field size q. This standard machinery motivates our assumptions. Because the group has only been identified by a Monte Carlo algorithm, there is a small non-zero probability that our algorithms might be executed on the wrong group. This has to be kept in mind when implementing the algorithms.

We do not assume that the input is tensor indecomposable, since the tensor decomposition algorithm described in Section 1.2.10.3 is not polynomial time.

A number of different cases arise:

(1) $G \leqslant \mathrm{GL}(d', \mathbb{F}_{q'})$ where $\mathbb{F}_{q'}$ has characteristic $p' \neq p$. This is called the *cross characteristic* case. Then [**LS74**] and [**SZ93**] tells us that $q \in \mathrm{O}\big(f(d')\big)$ for some polynomial f. This means that q is polynomial in the size of the input, which is not the case in general. In this case we can therefore use algorithms which normally are exponential time. In particular, by [**BB99**, Theorem 8.6] we can use the classical permutation group methods. Therefore we will only consider the case when we are given a group in *defining characteristic*, so that $p = p'$.

(2) $G \leqslant \mathrm{GL}(d', \mathbb{F}_s)$ where $d' > d$ and $\mathbb{F}_s \leqslant \mathbb{F}_q$. Let W be the module of G. If W is isomorphic to a tensor product of two modules which both have dimension less than $\dim W$, then we say that W is *tensor decomposable.* Otherwise W is *tensor indecomposable.*

Every possible W is isomorphic to a tensor product of twisted versions of tensor indecomposable modules of G (and hence of H). By the Steinberg tensor product theorem of [**Ste63**], in our cases the twists are Galois twists and the number of tensor indecomposable modules is independent of the field size, up to twists.

If W is tensor decomposable, we want to construct a tensor indecomposable representation V of G. In general, this is done using the tensor decomposition algorithm described in Section 1.2.10.3 on W, which also provides an effective isomorphism from W to V (*i.e.* between their acting groups). But since the algorithm is not polynomial time, a special version of one its subroutines has to be provided for each exceptional group.

If W is tensor indecomposable, we want to construct a representation V of G of dimension d, and we want do it in a way that also constructs an effective isomorphism. In principle this is always possible by computing tensor products of W and chopping them with the MeatAxe, because a composition factor of dimension d will always turn up. However, this is not always a practical algorithm, and the time complexity is not very good.

Note that if the minimal field $\mathbb{F}_s$ is a proper subfield of $\mathbb{F}_q$, then the tensor decomposition will not succeed. Since we assume that we know q, we can embed W canonically into an $\mathbb{F}_q G$-module. In this case we shall therefore always assume that $s = q$, contrary to our second assumption above.

(3) $G \leqslant \mathrm{GL}(d, \mathbb{F}_q)$, so that, by [**Ste63**], G is conjugate to H in $\mathrm{GL}(d, \mathbb{F}_q)$. This is the most interesting case since there are no standard methods, and we shall devote much effort to this case for the exceptional groups that we consider. A central issue will be to find elements of order a multiple of p. This is a serious obstacle since by [**IKS95, GL01**], the proportion $\rho(G)$ of these elements in G satisfies

$$\frac{2}{5q} < \rho(G) < \frac{5}{q}. \tag{1.1}$$

Hence we cannot find elements of order a multiple of p by random search in polynomial time, so there is no straightforward way to find them.

To be able to deal with these various cases, we need to know all the absolutely irreducible tensor indecomposable representations of H in defining characteristic. We also need to know how they arise from the *natural representation*, which is the representation of dimension d over $\mathbb{F}_q$. In our cases, this information is provided by [**Lüb01**].

1.2.8. Constructive membership testing overview. The other computational problem that we shall consider is the following.

Definition 1.7. The problem of *constructive membership testing* is:

Input: A matrix group $G = \langle X \rangle$, an element $g \in U \geqslant G$.
Output: If $g \in G$, then `true` and an `SLP` for g in X, `false` otherwise.

In our cases, U is always taken to be the general linear group. One can take two slightly different approaches to the problem of expressing an element as an `SLP` in the given generators, depending on whether one wants to find an effective isomorphism or find standard generators.

(1) The approach using an effective isomorphism.
    (a) Given $G = \langle X \rangle$ with standard copy H, first solve constructive recognition and obtain an effective isomorphism $\varphi : G \to H$. Hence obtain a generating set $\varphi(X)$ of H.
    (b) Given $g \in G$, express $\varphi(g)$ as an `SLP` in $\varphi(X)$, hence also expressing g in X.

(2) The approach using standard generators.
  (a) Given $G = \langle X \rangle$ with standard copy $H = \langle y_1, \dots, y_k \rangle$, find $g_1, \dots, g_k \in G$ as SLPs in X, such that the mapping $g_i \mapsto y_i$ is an isomorphism.
  (b) Given $g \in G$, express g as an SLP in $\{g_1, \dots, g_k\}$, hence also expressing it in X.

In the first case, the constructive membership testing takes place in H, which is probably faster than in G, so in this case we use the standard copy in computations. In the second case, the standard copy is only used as a theoretical tool. As it stands, the first approach is stronger, since it provides the effective isomorphism, and the standard generators in G can be obtained in the first approach, if necessary. However, if the representation theory of G is known, so that we can construct a module isomorphic to the module of G from the module of H, then the standard generators can be used, together with the MeatAxe, to solve constructive recognition. Hence the two approaches are not very different. One can also mix them in various ways, for example in the first case by finding standard generators in H expressed in $\varphi(X)$, and then only express each element in the standard generators, which might be easier than to express the elements directly in $\varphi(X)$.

1.2.9. CGT methods. Here we describe some algorithmic methods that we will use. Like many methods in CGT they are not really algorithms (*i.e.* they may not terminate on all inputs), or if they are they have very bad (worst-case) time complexity. Nevertheless, they can be useful for particular groups, as in our cases.

1.2.9.1. *The dihedral trick.* This trick is a method for conjugating involutions (*i.e.* elements of order 2) to each other in a black-box group, defined by a set of generators. The nice feature is that if the involutions are given as straight line programs in the generators, the conjugating element will be found as a straight line program. The dihedral trick is based on the following observation.

PROPOSITION 1.8. *Let G be a group and let $a, b \in G$ be involutions such that $|ba| = 2k + 1$ for some $k \in \mathbb{Z}$. Then $(ba)^k$ conjugates a to b.*

PROOF. Observe that

$$(ba)^{-k} a (ba)^k = (ba)^{k+1} a (ba)^k = (ba)^k b (ba)^k = (ba)^k a (ba)^{k-1} = \cdots = (ba)a = b$$

since a and b are involutions. □

THEOREM 1.9 (The dihedral trick). *Let $G = \langle X \rangle \leqslant \mathrm{GL}(d, q)$. Assume that the probability of the product of two random conjugate involutions in G having odd order is at least $1/c$. There exists a Las Vegas algorithm that, given conjugate involutions $a, b \in G$, finds $g \in G$ such that $a^g = b$. If a, b are given as SLPs of lengths l_a, l_b, then g will be found as an SLP of length $\mathrm{O}\big(c(l_a + l_b)\big)$. The algorithm has expected time complexity $\mathrm{O}\big(c(\xi(d) + d^3 \log(q) \log\log(q^d))\big)$ field operations.*

PROOF. The algorithm proceeds as follows:

(1) Find random $h \in G$ and let $a_1 = a^h$.

(2) Let $b_1 = ba_1$. Use Proposition 1.4 to determine if b_1 has even order, and if so, return to the first step.
(3) Let $n = (|b_1| - 1)/2$ and let $g = hb_1^n = h(a^h b)^n$.

By Proposition 1.8, this is a Las Vegas algorithm. The probability that b_1 has odd order is $1/c$ and hence the expected time complexity is as stated. Note that if a and b are given as `SLP`s in X, then we obtain g as an `SLP` in X. □

1.2.9.2. *Involution centralisers.* In [**HLO**$^+$**06**] an algorithm is described that reduces the constructive membership problem in a group G to the same problem in three involution centralisers in G. The reduction algorithm is known as the *Ryba algorithm* and can be a convenient method to solve the constructive membership problem. However, there are obstacles:

(1) We have to solve the constructive membership problem in the involution centralisers of G. In principle this can be done using the Ryba algorithm recursively, but such a blind descent might not be very satisfactory. For instance, it might not be easy to determine the time complexity of such a procedure. Another approach is to provide a special algorithm for the involution centraliser. This assumes that the structure of G and its involution centralisers are known, which it will be in the cases we consider.
(2) We have to find involutions in G. As described in Section 1.2.7, this is a serious obstacle if the defining field $\mathbb{F}_q$ of G has characteristic 2. In odd characteristic the situation is better, and in [**HLO**$^+$**06**] it is proved that the Ryba algorithm is polynomial time in this case. Another approach is to provide a special algorithm that finds involutions.
(3) We have to find generators Y for $\mathrm{C}_G(j)$ of a given involution $j \in G = \langle X \rangle$. This is possible using the *Bray algorithm* of [**Bra00**]. It works by computing random elements of $\mathrm{C}_G(j)$ until the whole centraliser is generated. This automatically gives the elements of Y as `SLP`s in X, which is a central feature needed by the Ryba algorithm.

    There are two issues involved when using this algorithm. First, the generators that are computed may not be uniformly random in $\mathrm{C}_G(j)$, so that we might have trouble generating the whole centraliser. In [**HLO**$^+$**06**] it is shown that this is not a problem with the exceptional groups. Second, we need to provide an algorithm that determines if the whole centraliser has been generated. In the cases that we will consider, this will be possible. It should be noted that the Bray algorithm works for any black-box group and not just for matrix groups.

Given these obstacles, we will still use the Ryba algorithm for constructive membership testing in some cases. We will also use the Bray algorithm independently, since it is a powerful tool.

1.2.9.3. *The Formula.* Like the dihedral trick, this is a method for conjugating elements to each other. For a group G, denote by $\Phi(G)$ the *Frattini subgroup* of G, which is the intersection of the maximal subgroups of G.

LEMMA 1.10 (The Formula). *Let $G \cong H{:}\mathrm{C}_n$, where H is a 2-group and n is odd. If $a, b \in G$ have order $n = 2k+1$ and $a \equiv b \mod H$, then $b \equiv a^g \mod \Phi(H)$ where $g = (ba)^k$.*

PROOF. The orders of a, b are their orders in C_n. Hence we can replace H with $H/\Phi(H)$ without affecting the rest of the assumptions. We can therefore reduce to the case when $\Phi(H) = \langle 1 \rangle$, in other words when H is elementary abelian.

Then $a = a_1 g$, $b = b_1 g$, with $|g| = n$ and $a_1, b_1 \in H$. We want to prove that $a^h = b$ where $h = (ba)^k$ or equivalently that $(ba)^k b = a(ba)^k$.

Now $(ba)^k = (b_1 g a_1 g)^k = (b_1 a_1^{g^{-1}} g^2)^k$. We can move all occurrences of g to the right, so that

$$(ba)^k = b_1^{1+g^{-2}+g^{-4}+\cdots+g^{-2(k-1)}} a_1^{g^{-1}+g^{-3}+\cdots+g^{-2k-1}} g^{2k}$$

from which we see that

$$\begin{aligned}(ba)^k b_1 g &= b_1^{1+g^{-2}+g^{-4}+\cdots+g^{-2k}} a_1^{g^{-1}+g^{-3}+\cdots+g^{-2k-1}} g^{2k+1} \\ a_1 g (ba)^k &= b_1^{g^{-1}+g^{-3}+\cdots+g^{-2k-1}} a_1^{1+g^{-2}+g^{-4}+\cdots+g^{-2k}} g^{2k+1}\end{aligned}$$

and we want these to be equal. Since $g^{2k+1} = 1$, we see that $(ba)^k b, a(ba)^k \in H$, and because H is elementary abelian, they are equal if and only if their product is the identity. But clearly $(ba)^k ba(ba)^k = b_1^s a_1^s$, where $s = \sum_{i=0}^{k} g^{-i}$, and finally $b_1^s a_1^s = (b_1 g)^{2k+1} (a_1 g)^{2k+1} = 1$. □

COROLLARY 1.11. *Let $H \cong P{:}\mathrm{C}_n$, where P is a 2-group and n is odd, and let $G \cong H{:}S$ for some group S. If $a \in H$ has order $n = 2k+1$ then for $h \in G$, such that $a \equiv a^h \mod P$, we have $a^{g^{-1}h} \equiv a \mod \Phi(P)$, where $g = (a^h a)^k$.*

PROOF. Observe that both a and a^h have order n and lie in $H \trianglelefteq G$. Now apply Lemma 1.10, conclude that $a^g \equiv a^h \mod \Phi(P)$, and the result follows. □

1.2.9.4. *Recognition of* $\mathrm{PSL}(2, q)$*.* In [**CLGO06**], an algorithm for constructive recognition and constructive membership testing of $\mathrm{PSL}(2, q)$ is presented. This algorithm is in several aspects the original which our algorithms are modelled after, and it is in itself an extension of [**CLG01**], which handles the natural representation.

We will use [**CLGO06**] since $\mathrm{PSL}(2, q)$ arise as subgroups of some of the exceptional groups that we consider. Because of this, we state the main results here. Let $\sigma_0(d)$ be the number of divisors of $d \in \mathbb{N}$. From [**HW79**, pp. 64, 359, 262], we know that for every $\varepsilon > 0$, if d is sufficiently large then $\sigma_0(d) < 2^{(1+\varepsilon)\log_{\mathrm{e}}(d)/\log\log_{\mathrm{e}}(d)}$.

Here, $\mathrm{PSL}(2, q)$ is viewed as a quotient of $\mathrm{SL}(2, q)$. Hence the elements are cosets of matrices.

THEOREM 1.12. *Assume an oracle for the discrete logarithm problem in* $\mathbb{F}_q$. *There exists a Las Vegas algorithm that, given* $\langle X \rangle \leqslant \mathrm{GL}(d, q)$ *satisfying the assumptions in Section* 1.2.7, *with* $\langle X \rangle \cong \mathrm{(P)SL}(2, q)$ *and* $q = p^e$, *finds an effective isomorphism* $\varphi : \langle X \rangle \to \mathrm{(P)SL}(2, q)$ *and performs preprocessing for constructive membership testing. The algorithm has expected time complexity*

$$\mathrm{O}\big((\xi(d) + d^3 \log(q) \log\log(q^d)) \log\log(q) + d^5 \sigma_0(d) \left|X\right| + d\chi_D(q) + \xi(d)d\big)$$

field operations.

The inverse of φ *is also effective. Each image of* φ *can be computed using* $\mathrm{O}\big(d^3\big)$ *field operations, and each pre-image using* $\mathrm{O}\big(d^3 \log(q) \log\log(q) + e^3\big)$ *field operations. After the algorithm has run, constructive membership testing of* $g \in \mathrm{GL}(d, q)$ *uses* $\mathrm{O}\big(d^3 \log(q) \log\log(q) + e^3\big)$ *field operations, and the resulting* `SLP` *has length* $\mathrm{O}\big(\log(q) \log\log(q)\big)$.

The existence of this constructive recognition algorithm has led to several other constructive recognition algorithms for the classical groups[**BK01, Bro03a, Bro03b, BK06**], which are polynomial time assuming an oracle for constructive recognition of $\mathrm{PSL}(2, q)$.

In some situations we shall also need a fast non-constructive recognition algorithm of $\mathrm{PSL}(2, q)$. It will be used to test if a given subgroup of $\mathrm{PSL}(2, q)$ is in fact the whole of $\mathrm{PSL}(2, q)$, so it is enough to have a Monte Carlo algorithm with no false positives. We need to use it in any representation, so the correct context is a black-box group.

THEOREM 1.13. *Let* $G = \langle X \rangle \leqslant \mathrm{GL}(d, q')$. *Assume that* G *is isomorphic to a subgroup of* $\mathrm{PSL}(2, q)$ *and that* q *is known. There exists a one-sided Monte Carlo algorithm with no false positives that determines if* $G \cong \mathrm{PSL}(2, q)$. *Given maximum error probability* $\varepsilon > 0$, *the time complexity is*

$$\mathrm{O}\big((\xi(d) + d^3 \log(q) \log\log(q^d)) \left\lceil \log(\varepsilon)/ \log(\delta) \right\rceil\big)$$

field operations, where

$$\delta = (1 - \phi(q - 1)/(q - 1))(1 - \phi(q + 1)/(q + 1)).$$

PROOF. It is well known that $\mathrm{PSL}(2, \mathbb{F}_q)$ is generated by two elements having order dividing $(q \pm 1)/2$, unless one of them lie in $\mathrm{PSL}(2, \mathbb{F}_s)$ for some $\mathbb{F}_s < \mathbb{F}_q$. We are thus led to an algorithm that performs at most n steps. In each step it finds two random elements g and h of G and computes their pseudo-orders. If g has order dividing $(q - 1)/2$, it then determines if g lies in some $\mathrm{PSL}(2, \mathbb{F}_s)$ by testing if $g^{(s-1)/2} = 1$. If $q = p^a$ then the possible values of s are p^b where $b \mid a$, so there are $\sigma_0(a)$ subfields. If g does not lie in any $\mathrm{PSL}(2, \mathbb{F}_s)$ then g is remembered.

Now do similarly for h. Then, if it has found the two elements, it returns `true`. On the other hand, if it completes the n steps without finding these elements, it returns `false`.

The proportion in $\mathrm{PSL}(2,q)$ of elements of order $(q \pm 1)/2$ is $\phi(q \pm 1)/(q \pm 1)$, so the probability that $G \cong \mathrm{PSL}(2,q)$ but we fail to find the elements is at most δ^n. We require $\delta^n \leqslant \varepsilon$ and hence n can be chosen as $\lceil \log(\varepsilon)/\log(\delta) \rceil$. □

1.2.10. Aschbacher classes. The Aschbacher classification of [**Asc84**] classifies matrix groups into a number of classes, and a major part of the MGRP has been to develop algorithms that determine constructively if a given matrix group belongs to a certain class. Some of these algorithms are used here.

1.2.10.1. *The MeatAxe.* Let $G = \langle X \rangle \leqslant \mathrm{GL}(d,q)$ acting on a module $V \cong \mathbb{F}_q^d$. The algorithm known as the *MeatAxe* determines if V is irreducible. If not, it finds a proper non-trivial submodule W of V, and a change of basis matrix $c \in \mathrm{GL}(d,q)$ that exhibits the action of G on W and on V/W. In other words, the first $\dim W$ rows of c form a basis of W, and g^c is block lower triangular for every $g \in G$.

Applying the MeatAxe recursively, one finds a composition series of V, and a change of basis that exhibits the action of G on the composition factors of V. Hence we also obtain effective isomorphisms from G to the groups acting on the composition factors of V.

The MeatAxe was originally developed by Parker in [**Par84**] and later extended and formalised into a Las Vegas algorithm by Holt and Rees in [**HR94**]. They also explain how very similar algorithms can be used to test if V is absolutely irreducible, and if two modules are isomorphic. These are also Las Vegas algorithms with the same time complexity as the MeatAxe. The worst case of the MeatAxe is treated in [**IL00**], where it is proved that the expected time complexity is $\mathrm{O}\big(|X|\, d^4\big)$ field operations. Unless the module is reducible and all composition factors of the module are isomorphic, the expected time complexity is $\mathrm{O}\big(|X|\, d^3\big)$ field operations.

The MeatAxe is also fast in practice and is implemented both in Magma and GAP. This important feature is the reason that it is used rather than the first known polynomial time algorithm for the same problem, which was given in [**Rón90**] (and is, at best, $\mathrm{O}\big(|X|\, d^6\big)$).

Some related problems are the following:

- Determine if G acts absolutely irreducibly on V.
- Given two irreducible G-modules U, W of dimension d, determine if they are isomorphic, and if so find a change of basis matrix $c \in \mathrm{GL}(d,q)$ that conjugates U to W.
- Given that G acts absolutely irreducibly on V, determine if G preserves a classical form and if so find a matrix for the form.
- Find a basis for $\mathrm{End}_G(V)$ as matrices of degree d.

Algorithms for these problems are described in [**HEO05**, Section 7.5] and they all have expected time complexity $\mathrm{O}\big(|X|\, d^3\big)$ field operations. We will refer to the algorithms for all these problems as "the MeatAxe".

1.2.10.2. *Writing matrix groups over subfields.* If $G = \langle X \rangle \leqslant \mathrm{GL}(d, \mathbb{F}_q)$, then G might be conjugate in $\mathrm{GL}(d, \mathbb{F}_q)$ to a subgroup of $\mathrm{GL}(d, \mathbb{F}_s)$ where $\mathbb{F}_s < \mathbb{F}_q$, so that q is a proper power of s. An algorithm for deciding if this is case is given in [**GLGO06**]. It is a Las Vegas algorithm with expected time complexity $\mathrm{O}\big(\sigma_0(\log(q))(|X|\, d^3 + d^2 \log(q))\big)$ field operations. In case G can be written over a subfield, then the algorithm also returns a conjugating matrix c that exhibits this fact, *i.e.* so that G^c can be immediately embedded in $\mathrm{GL}(d, \mathbb{F}_s)$. The algorithm can also write a group over a subfield modulo scalars.

1.2.10.3. *Tensor decomposition.* Now let $G = \langle X \rangle \leqslant \mathrm{GL}(d, q)$ acting on a module $V \cong \mathbb{F}_q^d$. The module might have the structure of a tensor product $V \cong U_1 \otimes U_2$, so that $G \leqslant G_1 \circ G_2$ where $G_1 \leqslant \mathrm{GL}(U_1)$ and $G_2 \leqslant \mathrm{GL}(U_2)$.

The Las Vegas algorithm of [**LGO97a**] determines if V has the structure of a tensor product, and if so it also returns a change of basis $c \in \mathrm{GL}(d, q)$ which exhibits the tensor decomposition. In other words, g^c is an explicit Kronecker product for each $g \in G$. The images of g in G_1 and G_2 can therefore immediately be extracted from g^c, and hence we obtain an effective embedding of G into $G_1 \circ G_2$.

By [**LGO97b**], for tensor decomposition it is sufficient to find a *flat* in a projective geometry corresponding to the decomposition. A flat is a subspace of V of the form $A \otimes U_2$ or $U_1 \otimes B$ where A and B are proper subspaces of U_1 and U_2 respectively. This flat contains a *point*, which is a flat with $\dim A = 1$ or $\dim B = 1$. If we can provide a proposed flat to the algorithm of [**LGO97a**], then it will verify that it is a flat, and if so find a tensor decomposition, using expected $\mathrm{O}\big(|X|\, d^3 \log(q)\big)$ field operations.

However, in general there is no efficient algorithm for finding a flat of V. If we want a polynomial time algorithm for decomposing a specific tensor product, we therefore have to provide an efficient algorithm that finds a flat.

1.2.11. Conjectures. Most of the results presented will depend on a few conjectures. This might be considered awkward and somewhat non-mathematical, but it is a result of how the work in this thesis was produced. In almost every case with the algorithms that are presented, the implementation of the algorithm did exist before the proof of correctness of the algorithm. In fact, the algorithms have been developed using a rather empirical method, an interplay between theory (mathematical thought) and practice (programming). We consider this to be an essential feature of the work, and it has proven to be an effective way to develop algorithms that are good in both theory and practice.

However, it has lead to the fact that there are certain results that have been left unproven, either because they have been too hard to prove or have been from an area of mathematics outside the scope of this thesis (usually both). But because of the way the algorithms have been developed, there should be no doubt that every one of the conjectures are true. The implementations of the algorithms have been tested on a vast number of inputs, and therefore the conjectures have also been tested equally many times. There has been no case of a conjecture failing.

More detailed information about the implementations can be found in Chapter 6.

CHAPTER 2

Twisted exceptional groups

Here we will present the necessary theory about the twisted groups under consideration.

2.1. Suzuki groups

The family of exceptional groups now known as the *Suzuki groups* were first found by Suzuki in [**Suz60, Suz62, Suz64**], and also described in [**HB82**, Chapter 11] which is the exposition that we follow. They should not be confused with the Suzuki 2-groups or the sporadic Suzuki group.

2.1.1. Definition and properties. We begin by defining our standard copy of the Suzuki group. Following [**HB82**, Chapter 11], let $q = 2^{2m+1}$ for some $m > 0$ and let π be the unique automorphism of $\mathbb{F}_q$ such that $\pi^2(x) = x^2$ for every $x \in \mathbb{F}_q$, *i.e.* $\pi(x) = x^t$ where $t = 2^{m+1} = \sqrt{2q}$. For $a, b \in \mathbb{F}_q$ and $c \in \mathbb{F}_q^{\times}$, define the following matrices:

$$S(a,b) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a & 1 & 0 & 0 \\ b & \pi(a) & 1 & 0 \\ a^2\pi(a) + ab + \pi(b) & a\pi(a) + b & a & 1 \end{bmatrix}, \tag{2.1}$$

$$M(c) = \begin{bmatrix} c^{1+2^m} & 0 & 0 & 0 \\ 0 & c^{2^m} & 0 & 0 \\ 0 & 0 & c^{-2^m} & 0 \\ 0 & 0 & 0 & c^{-1-2^m} \end{bmatrix}, \tag{2.2}$$

$$T = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{2.3}$$

By definition,

$$\mathrm{Sz}(q) = \left\langle S(a,b), M(c), T \mid a, b \in \mathbb{F}_q, c \in \mathbb{F}_q^{\times} \right\rangle. \tag{2.4}$$

If we define

$$\mathcal{F} = \{S(a,b) \mid a, b \in \mathbb{F}_q\} \tag{2.5}$$

$$\mathcal{H} = \left\{M(c) \mid c \in \mathbb{F}_q^{\times}\right\} \tag{2.6}$$

then $\mathcal{F} \leqslant \mathrm{Sz}(q)$ with $|\mathcal{F}| = q^2$ and $\mathcal{H} \cong \mathbb{F}_q^{\times}$ so that $\mathcal{H}$ is cyclic of order $q-1$. Moreover, we can write $M(c)$ as

$$M(c) = M'(\lambda) = \begin{bmatrix} \lambda^{t+1} & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda^{-1} & 0 \\ 0 & 0 & 0 & \lambda^{-t-1} \end{bmatrix} \tag{2.7}$$

where $\lambda = c^{2^m}$. Hence $M(\lambda)^t = M'(\lambda)$.

The following result follows from [**HB82**, Chapter 11].

THEOREM 2.1. (1) *The order of the Suzuki group is*

$$|\mathrm{Sz}(q)| = q^2(q^2+1)(q-1) \tag{2.8}$$

and $q^2+1 = (q+t+1)(q-t+1)$.

(2) $\gcd(q-1, q^2+1) = 1$ *and hence the three factors in* (2.8) *are pairwise relatively prime.*

(3) *For all* $a_1, b_1, a_2, b_2 \in \mathbb{F}_q$ *and* $\lambda \in \mathbb{F}_q^{\times}$*:*

$$S(a_1,b_1)S(a_2,b_2) = S(a_1+a_2, b_1+b_2+a_1^t a_2) \tag{2.9}$$

$$S(a,b)^{-1} = S(a, b+a^{t+1}) \tag{2.10}$$

$$S(a_1,b_1)^{S(a_2,b_2)} = S(a_1, b_1 + a_1^t a_2 + a_1 a_2^t) \tag{2.11}$$

$$S(a,b)^{M'(\lambda)} = S(\lambda^t a, \lambda^{t+2} b). \tag{2.12}$$

(4) *There exists* $\mathcal{O} \subseteq \mathbb{P}^3(\mathbb{F}_q)$ *on which* $\mathrm{Sz}(q)$ *acts faithfully and doubly transitively, such that no nontrivial element of* $\mathrm{Sz}(q)$ *fixes more than* 2 *points. This set is*

$$\mathcal{O} = \{(1:0:0:0)\} \cup \left\{(ab+\pi(a)a^2+\pi(b) : b : a : 1) \mid a, b \in \mathbb{F}_q\right\}. \tag{2.13}$$

(5) *The stabiliser of* $P_\infty = (1:0:0:0) \in \mathcal{O}$ *is* $\mathcal{FH}$ *and if* $P_0 = (0:0:0:1)$ *then the stabiliser of* (P_∞, P_0) *is* $\mathcal{H}$.

(6) $\mathrm{Z}(\mathcal{F}) = \{S(0,b) \mid b \in \mathbb{F}_q\}$.

(7) $\mathcal{FH}$ *is a Frobenius group with Frobenius kernel* $\mathcal{F}$.

(8) $\mathrm{Sz}(q)$ *has cyclic Hall subgroups* U_1 *and* U_2 *of orders* $q \pm t + 1$. *These act fixed point freely on* $\mathcal{O}$ *and irreducibly on* $\mathbb{F}_q^4$. *For each non-trivial* $g \in U_i$, *we have* $\mathrm{C}_G(g) = U_i$.

(9) *The conjugates of* $\mathcal{F}$, $\mathcal{H}$, U_1 *and* U_2 *partition* $\mathrm{Sz}(q)$.

(10) *The proportion of elements of order* $q-1$ *in* $\mathcal{FH}$ *is* $\phi(q-1)/(q-1)$, *where* ϕ *is the Euler totient function.*

REMARK 2.2 (Standard generators of $\mathrm{Sz}(q)$). As standard generators for $\mathrm{Sz}(q)$ we will use

$$\{S(1,0), M'(\lambda), T\},$$

where λ is a primitive element of $\mathbb{F}_q$, whose minimal polynomial is the defining polynomial of $\mathbb{F}_q$. Other sets are possible: in [**Bra07**], the standard generators are

$$\left\{S(1,0)^{-1}, M'(\lambda)^{2^m}, T\right\},$$

and MAGMA uses

$$\left\{S(1,0)^{M'(\lambda)^{q/2}}, M'(\lambda)^{1-2^m}, T\right\}.$$

From [**HB82**, Chapter 11, Remark 3.12] we also immediately obtain the following result.

THEOREM 2.3. *A maximal subgroup of $G = \mathrm{Sz}(q)$ is conjugate to one of the following subgroups.*

(1) *The point stabiliser $\mathcal{FH}$.*
(2) *The normaliser $\mathrm{N}_G(\mathcal{H}) \cong \mathrm{D}_{2(q-1)}$.*
(3) *The normalisers $\mathcal{B}_i = \mathrm{N}_G(U_i)$ for $i = 1, 2$. These satisfy $\mathcal{B}_i = \langle U_i, t_i\rangle$ where $u^{t_i} = u^q$ for every $u \in U_i$ and $[\mathcal{B}_i : U_i] = 4$.*
(4) *$\mathrm{Sz}(s)$ where q is a proper power of s.*

PROPOSITION 2.4. *Let $G = \mathrm{Sz}(q)$.*

(1) *Distinct conjugates of $\mathcal{F}$, $\mathcal{H}$, U_1 or U_2 intersect trivially.*
(2) *The subgroups of G of order q^2 are conjugate, and there are q^2+1 distinct conjugates.*
(3) *The cyclic subgroups of G of order $q-1$ are conjugate, and there are $q^2(q^2+1)/2$ distinct conjugates.*
(4) *The cyclic subgroups of G of order $q \pm t + 1$ are conjugate, and there are $q^2(q-1)(q \mp t + 1)/4$ distinct conjugates.*

PROOF. (1) By Theorem 2.1, each conjugate of $\mathcal{F}$ fixes exactly one point of $\mathcal{O}$. If an element g lies in two distinct conjugates it must fix two distinct points and hence lie in a conjugate of $\mathcal{H}$. But, by the partitioning, the conjugates of $\mathcal{H}$ and $\mathcal{F}$ intersect trivially, so $g = 1$.

If $\mathcal{H} \neq \mathcal{H}^x$ for some $x \in G$ and $g \in \mathcal{H} \cap \mathcal{H}^x$, then g fixes more than 2 points of $\mathcal{O}$, so that $g = 1$. If $U_i \neq U_i^x$ for some $x \in G$ and $g \in U_i \cap U_i^x$, then $\mathrm{C}_G(g) = \langle U_i \cup U_i^x\rangle = G$, so that $g = 1$.

(2) This is clear since these subgroups are Sylow 2-subgroups, and hence conjugate to $\mathcal{F}$. Each subgroup fixes a point of $\mathcal{O}$ and hence there are $|\mathcal{O}|$ distinct conjugates.
(3) Because of the partitioning, an element of order $q-1$ must lie in a conjugate of $\mathcal{H}$, which must be the cyclic subgroup that it generates. By Theorem 2.3 there are $[G : \mathrm{N}_G(\mathcal{H})] = q^2(q^2+1)/2$ distinct conjugates.
(4) Analogous to the previous case.

□

PROPOSITION 2.5. *Let $G = \mathrm{Sz}(q)$ and let ϕ be the Euler totient function.*

(1) *The number of elements in G that fix at least one point of $\mathcal{O}$ is* $q^2(q-1)(q^2+q+2)/2$.
(2) *The number of elements in G of order $q-1$ is* $\phi(q-1)q^2(q^2+1)/2$.
(3) *The number of elements in G of order $q \pm t + 1$ is* $\phi(q \pm t+1)(q \mp t+1)q^2(q-1)/4$.

PROOF. (1) By Theorem 2.1, if $g \in G$ fixes exactly one point, then g is in a conjugate of $\mathcal{F}$, and if g fixes two points, then g is in a conjugate of $\mathcal{H}$. Hence by Proposition 2.4, there are $(|\mathcal{F}|-1)(q^2+1)$ elements that fix exactly one point. Similarly, there are $q^2(q^2+1)(|\mathcal{H}|-1)/2$ elements that fix exactly two points.

Thus the number of elements that fix at least one point is

$$1 + (|\mathcal{F}|-1)(q^2+1) + q^2(q^2+1)(|\mathcal{H}|-1)/2 = \frac{q^2(q-1)(q^2+q+2)}{2}. \tag{2.14}$$

(2) By Proposition 2.4, an element of order $q-1$ must lie in a conjugate of $\mathcal{H}$. Since distinct conjugates intersect trivially, the number of such elements is the number of generators of all cyclic subgroups of order $q-1$.
(3) Analogous to the previous case.

□

PROPOSITION 2.6. *If $g \in G = \mathrm{Sz}(q)$ is uniformly random, then*

$$\Pr[|g| = q-1] = \frac{\phi(q-1)}{2(q-1)} > \frac{1}{12\log\log(q)} \tag{2.15}$$

$$\Pr[|g| = q \pm t + 1] = \frac{\phi(q \pm t+1)}{4(q \pm t + 1)} > \frac{1}{24\log\log(q)} \tag{2.16}$$

$$\Pr[g \text{ fixes a point of } \mathcal{O}] = \frac{q^2+q+2}{2(q^2+1)} > \frac{1}{2} \tag{2.17}$$

and hence the expected number of random selections required to obtain an element of order $q-1$ or $q \pm t+1$ is $\mathrm{O}(\log\log q)$, and $\mathrm{O}(1)$ to obtain an element that fixes a point.

PROOF. The first equality follows immediately from Theorem 2.1 and Proposition 2.5. The inequalities follow from [**MSC96**, Section II.8].

Clearly the number of selections required is geometrically distributed, where the success probabilities for each selection are given by the inequalities. Hence the expectations are as stated. □

PROPOSITION 2.7. *Let $G = \mathrm{Sz}(q)$.*

(1) *For every $g \in G$, distinct conjugates of $\mathrm{C}_G(g)$ intersect trivially.*
(2) *If $H \leqslant G$ is cyclic of order $q-1$ and $g \in G \setminus \mathrm{N}_G(H)$ then $|HgH| = (q-1)^2$.*

PROOF. (1) By Theorem 2.1, we consider three cases. If g lies in a conjugate F of $\mathcal{F}$, then $\mathrm{C}_G(g) \leqslant F$. If $g \neq 1$ lies in a conjugate H of $\mathcal{H}$, then $\mathrm{C}_G(g) = H$ and if g lies in a conjugate U of U_1 or U_2, then $\mathrm{C}_G(g) = U$. In each case the result follows from Proposition 2.4.

(2) Since $|H| = q-1$ it is enough to show that $H \cap H^g = \langle 1 \rangle$. This follows immediately from Proposition 2.4.

□

PROPOSITION 2.8. *Elements of odd order in* $\mathrm{Sz}(q)$ *that have the same trace are conjugate.*

PROOF. From [**Suz62**, §17], the number of conjugacy classes of non-identity elements of odd order is $q-1$, and all elements of even order have trace 0. Observe that

$$S(0,b)T = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & b \\ 1 & 0 & b & b^t \end{bmatrix}. \tag{2.18}$$

Since b can be any element of $\mathbb{F}_q$, so can $\operatorname{Tr}(S(0,b)T)$, and this also implies that $S(0,b)T$ has odd order when $b \neq 0$. Therefore there are $q-1$ possible traces for non-identity elements of odd order, and elements with different trace must be non-conjugate, so all conjugacy classes must have different traces. □

PROPOSITION 2.9. *The proportion of elements of order* 4 *among the elements of trace* 0 *is* $1 - 1/q + 1/q^2 - 1/q^3$.

PROOF. The elements of trace 0 are those with orders $1, 2, 4$, and apart from the identity these are the elements that fix precisely one point of $\mathcal{O}$. From the proof of Proposition 2.5, there are q^4 elements of trace 0.

The elements of order 4 lie in a conjugate of $\mathcal{F}$, and there are $q^2 - q$ elements in each conjugate. Hence from Proposition 2.4 there are $q(q-1)(q^2+1)$ elements of order 4. □

PROPOSITION 2.10. *Let* $P = (p_1 : p_2 : p_3 : p_4) \in \mathcal{O}^g$ *be uniformly random, where* $\mathcal{O}^g = \{Rg \mid R \in \mathcal{O}\}$ *for some* $g \in \mathrm{GL}(4,q)$. *Then*

(1)

$$\Pr[p_i \neq 0 \mid i = 1, \ldots, 4] \geqslant \left(1 - \frac{q+2}{q^2+1}\right)^4. \tag{2.19}$$

(2) *If* $Q = (q_1 : q_2 : q_3 : q_4) \in \mathcal{O}^g$ *is fixed, then*

$$\Pr[p_2^t q_3^{t+2} \neq q_2^t p_3^{t+2}] \geqslant 1 - \frac{1 + (t+2)q}{q^2+1}. \tag{2.20}$$

PROOF. (1) By [**HB82**, Chapter 11, Lemma 3.4], $\mathcal{O}$ is an ovoid, so it intersects any (projective) line in $\mathbb{P}^3(\mathbb{F}_q)$ in at most 2 points. The condition $p_i = 0$ defines a projective plane $\mathcal{P} \subseteq \mathbb{P}^3(\mathbb{F}_q)$. If $\mathcal{O}^g \cap \mathcal{P} \neq \emptyset$ then it contains a point A, and there are $q+1$ lines in $\mathcal{P}$ that passes through A. Each one of these lines passes through at most one other point of $\mathcal{O}^g$, but each line contains $q+1$ points of $\mathcal{P}$, and hence at least q of those points are not in $\mathcal{O}^g$. Moreover, each pair of lines has only the point A in common.

Now we have considered $1 + q(q+1)$ distinct points of $\mathcal{P}$, which are all points of $\mathcal{P}$, and we have proved that at most $q+2$ of those lie in $\mathcal{O}^g$.

(2) Clearly, $p_2^t q_3^{t+2} = q_2^t p_3^{t+2}$ if and only if $p_2 q_3^{t+1} = q_2 p_3^{t+1}$. If $P = P'g$, where $P_\infty \neq P' \in \mathcal{O}$ and $g = [g_{i,j}]$, then $p_i = g_{1,i}(a^{t+2}+b^t+ab)+g_{2,i}b+g_{3,i}a+g_{4,i}$ for some $a, b \in \mathbb{F}_q$.

Introducing indeterminates x and y in place a and b, it follows that the expression $p_2 q_3^{t+1} - q_2 p_3^{t+1}$ is a polynomial $f \in \mathbb{F}_q[x, y]$ with $\deg_x(f) \leqslant 3t+4$ and $\deg_y(f) \leqslant t+2$. For each $a \in \mathbb{F}_q$, the number of roots of $f(a, y)$ is therefore at most $t + 2$, so the number of roots of f is at most $q(t + 2)$. □

Proposition 2.11. *If $g_1, g_2 \in \mathcal{FH}$ are uniformly random, then*

$$\Pr[|[g_1, g_2]| = 4] = 1 - \frac{1}{q-1}. \tag{2.21}$$

Proof. Let $A = \mathcal{FH}/\operatorname{Z}(\mathcal{F})$. By Theorem 2.1, $[g_1, g_2] \in \mathcal{F}$ and has order 4 if and only if $[g_1, g_2] \notin \operatorname{Z}(\mathcal{F}) \triangleleft \mathcal{FH}$. It therefore suffices to find the proportion of pairs $k_1, k_2 \in A$ such that $[k_1, k_2] = 1$.

If $k_1 = 1$ then k_2 can be any element of A, which contributes $q(q-1)$ pairs. If $1 \neq k_1 \in \mathcal{F}/\operatorname{Z}(\mathcal{F}) \cong \mathbb{F}_q$ then $\operatorname{C}_A(k_1) = \mathcal{F}/\operatorname{Z}(\mathcal{F})$, so we again obtain $q(q-1)$ pairs. Finally, if $k_1 \notin \mathcal{F}/\operatorname{Z}(\mathcal{F})$ then $|\operatorname{C}_A(k_1)| = q - 1$, so we obtain $q(q-2)(q-1)$ pairs. Thus we obtain $q^2(q-1)$ pairs from a total of $|A \times A| = q^2(q-1)^2$ pairs, and the result follows. □

Proposition 2.12. *Let $G = \operatorname{Sz}(q)$. If $x, y \in G$ are uniformly random, then*

$$\Pr[\langle x, y\rangle = G] = 1 - \operatorname{O}\big(\sigma_0(\log(q))/q^2\big) \tag{2.22}$$

Proof. By (2.8) and Theorem 2.3, the maximal subgroup $M \leqslant G$ with smallest index is $M = \mathcal{FH}$. Then $[G : M] = q^2 + 1$ and since $M = \operatorname{N}_G(M)$, there are $q^2 + 1$ conjugates of M.

$$\Pr[\langle x, y\rangle \leqslant M^g \text{ some } g \in G] \leqslant \sum_{i=1}^{q^2+1} \Pr[\langle x, y\rangle \leqslant M] = \frac{1}{q^2+1} \tag{2.23}$$

The probability that $\langle x, y\rangle$ lies in any maximal not conjugate to M must be less than $1/(q^2+1)$ because the other maximals have larger indices. There are $\operatorname{O}\big(\sigma_0(\log(q))\big)$ number of conjugacy classes of maximal subgroups, and hence the probability that $\langle x, y\rangle$ lies in a maximal subgroup is $\operatorname{O}\big(\sigma_0(\log(q))/q^2\big)$. □

2.1.2. Alternative definition. The way we have defined the Suzuki groups resembles the original definition, but it is not clear that the groups are exceptional groups of Lie type. This was first proved in [**Ono62, Ono63**]. A more common way to define the groups are as the fixed points of a certain automorphism of $\operatorname{Sp}(4, q)$. This approach is followed in [**Wil05**, Chapter 4.10], and it provides a more straightforward method to deal with non-constructive recognition of $\operatorname{Sz}(q)$.

Let $\mathrm{Sp}(4, q)$ denote the standard copy of the symplectic group, preserving the following symplectic form:

$$J = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{2.24}$$

From [**Wil05**, Chapter 4.10], we know that the elements of $\mathrm{Sz}(q)$ are precisely the fixed points of an automorphism Ψ of $\mathrm{Sp}(4, q)$. Computing $\Psi(g)$ for some $g \in \mathrm{Sp}(4, q)$ amounts to taking a submatrix of the exterior square of g and then replacing each matrix entry x by x^{2^m}. Moreover, Ψ is defined on $\mathrm{Sp}(4, F)$ for $F \geqslant \mathbb{F}_q$. A more detailed description of how to compute $\Psi(g)$ can be found in [**Wil05**, Chapter 4.10].

LEMMA 2.13. *Let $G \leqslant \mathrm{Sp}(4, q)$ have natural module V and assume that V is absolutely irreducible. Then $G^h \leqslant \mathrm{Sz}(q)$ for some $h \in \mathrm{GL}(4, q)$ if and only if $V \cong V^{\Psi}$.*

PROOF. Assume $G^h \leqslant \mathrm{Sz}(q)$. Both G and $\mathrm{Sz}(q)$ preserve the form (2.24), and this form is unique up to a scalar multiple, since V is absolutely irreducible. Therefore $hJh^T = \lambda J$ for some $\lambda \in \mathbb{F}_q^{\times}$. But if $\mu = \sqrt{\lambda^{-1}}$ then $(\mu h)J(\mu h)^T = J$, so that $\mu h \in \mathrm{Sp}(4, q)$. Moreover, $G^h = G^{\mu h}$, and hence we may assume that $h \in \mathrm{Sp}(4, q)$. Let $x = h\Psi(h^{-1})$ and observe that for each $g \in G$, $\Psi(g^h) = g^h$. It follows that

$$g^x = \Psi(h) g^h \Psi(h^{-1}) = \Psi(h g^h h^{-1}) = \Psi(g) \tag{2.25}$$

so $V \cong V^{\Psi}$.

Conversely, assume that $V \cong V^{\Psi}$. Then there is some $h \in \mathrm{GL}(4, q)$ such that for each $g \in G$ we have $g^h = \Psi(g)$. As above, since both G and $\Psi(G)$ preserve the form (2.24), we may assume that $h \in \mathrm{Sp}(4, q)$.

Let K be the algebraic closure of $\mathbb{F}_q$. The Steinberg-Lang Theorem (see [**Ste77**]) asserts that there exists $x \in \mathrm{Sp}(4, K)$ such that $h = x^{-1}\Psi(x)$. It follows that

$$\Psi(g^{x^{-1}}) = \Psi(g)^{h^{-1}x^{-1}} = g^{x^{-1}} \tag{2.26}$$

so that $G^{x^{-1}} \leqslant \mathrm{Sz}(q)$. Thus G is conjugate in $\mathrm{GL}(4, K)$ to a subgroup S of $\mathrm{Sz}(q)$, and it follows from [**CR06**, Theorem 29.7], that G is conjugate to S in $\mathrm{GL}(4, q)$. □

2.1.3. Tensor indecomposable representations. It follows from [**Lüb01**] that over an algebraically closed field in defining characteristic, up to Galois twists, there is only one absolutely irreducible tensor indecomposable representation of $\mathrm{Sz}(q)$: the natural representation.

2.2. Small Ree groups

The small Ree groups were first described in [**Ree60, Ree61c**]. Their structure has been investigated in [**War63, War66, LN85, Kle88**]. A short survey is also given in [**HB82**, Chapter 11]. They should not be confused with the Big Ree groups, which are described in Section 2.3.

2.2.1. Definition and properties. We now define our standard copy of the Ree groups. The generators that we use are those described in [**KLM01**]. Let $q = 3^{2m+1}$ for some $m > 0$ and let $t = 3^m$. For $x \in \mathbb{F}_q$ and $\lambda \in \mathbb{F}_q^\times$, define the matrices

$$\alpha(x) = \begin{bmatrix} 1 & x^t & 0 & 0 & -x^{3t+1} & -x^{3t+2} & x^{4t+2} \\ 0 & 1 & x & x^{t+1} & -x^{2t+1} & 0 & -x^{3t+2} \\ 0 & 0 & 1 & x^t & -x^{2t} & 0 & x^{3t+1} \\ 0 & 0 & 0 & 1 & x^t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -x & x^{t+1} \\ 0 & 0 & 0 & 0 & 0 & 1 & -x^t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.27}$$

$$\beta(x) = \begin{bmatrix} 1 & 0 & -x^t & 0 & -x & 0 & -x^{t+1} \\ 0 & 1 & 0 & x^t & 0 & -x^{2t} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & x \\ 0 & 0 & 0 & 1 & 0 & x^t & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & x^t \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.28}$$

$$\gamma(x) = \begin{bmatrix} 1 & 0 & 0 & -x^t & 0 & -x & -x^{2t} \\ 0 & 1 & 0 & 0 & -x^t & 0 & x \\ 0 & 0 & 1 & 0 & 0 & x^t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -x^t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.29}$$

$$h(\lambda) = \begin{bmatrix} \lambda^t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda^{1-t} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda^{2t-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda^{1-2t} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda^{t-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda^{-t} \end{bmatrix} \tag{2.30}$$

$$\Upsilon = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.31}$$

and define the Ree group as

$$\mathrm{Ree}(q) = \left\langle \alpha(x), \beta(x), \gamma(x), h(\lambda), \Upsilon \mid x \in \mathbb{F}_q, \lambda \in \mathbb{F}_q^\times \right\rangle . \tag{2.32}$$

Also, define the subgroups of upper triangular and diagonal matrices:

$$U(q) = \langle \alpha(x), \beta(x), \gamma(x) \mid x \in \mathbb{F}_q \rangle \tag{2.33}$$

$$H(q) = \left\{ h(\lambda) \mid \lambda \in \mathbb{F}_q^\times \right\} \cong \mathbb{F}_q^\times . \tag{2.34}$$

From [**LN85**] we then know that each element of $U(q)$ can be expressed in a unique way as

$$S(a,b,c) = \alpha(a)\beta(b)\gamma(c) \tag{2.35}$$

so that $U(q) = \{S(a,b,c) \mid a,b,c \in \mathbb{F}_q\}$, and it follows that $|U(q)| = q^3$. We also know that $U(q)$ is a Sylow 3-subgroup of $\mathrm{Ree}(q)$, and direct calculations show that

$$\begin{aligned} &S(a_1,b_1,c_1)S(a_2,b_2,c_2) = \\ &= S(a_1+a_2, b_1+b_2-a_1a_2^{3t}, c_1+c_2-a_2b_1+a_1a_2^{3t+1}-a_1^2a_2^{3t}), \end{aligned} \tag{2.36}$$

$$S(a,b,c)^{-1} = S(-a, -(b+a^{3t+1}), -(c+ab-a^{3t+2})), \tag{2.37}$$

$$\begin{aligned} &S(a_1,b_1,c_1)^{S(a_2,b_2,c_2)} = \\ &= S(a_1, b_1-a_1a_2^{3t}+a_2a_1^{3t}, c_1+a_1b_2-a_2b_1+a_1a_2^{3t+1}-a_2a_1^{3t+1}-a_1^2a_2^{3t}+a_2^2a_1^{3t}) \end{aligned} \tag{2.38}$$

and

$$S(a,b,c)^{h(\lambda)} = S(\lambda^{3t-2}a, \lambda^{1-3t}b, \lambda^{-1}c). \tag{2.39}$$

Remark 2.14 (Standard generators of $\mathrm{Ree}(q)$). As standard generators for $\mathrm{Ree}(q)$ we will use

$$\{S(1,0,0), h(\lambda), \Upsilon\},$$

where λ is a primitive element of $\mathbb{F}_q$, whose minimal polynomial is the defining polynomial of $\mathbb{F}_q$.

The Ree groups preserve a symmetric bilinear form on $\mathbb{F}_q^7$, represented by the matrix

$$J = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.40}$$

From [**War66**] and [**HB82**, Chapter 11] we immediately obtain

Proposition 2.15. *Let* $G = \mathrm{Ree}(q)$.

(1) $|G| = q^3(q^3+1)(q-1)$ *where* $\gcd(q^3+1, q-1) = 2$.
(2) *Conjugates of* $U(q)$ *intersect trivially.*
(3) *The centre* $\mathrm{Z}(U(q)) = \{S(0,0,c) \mid c \in F_q\}$.
(4) *The derived group* $U(q)' = \{S(0,b,c) \mid b,c \in F_q\}$, *and its elements have order* 3.

(5) *The elements in* $U(q) \setminus U(q)' = \{S(a,b,c) \mid a \neq 0\}$ *have order* 9 *and their cubes form* $\mathrm{Z}(U(q)) \setminus \langle 1 \rangle$.
(6) $\mathrm{N}_G(U(q)) = U(q)H(q)$ *and* G *acts doubly transitively on the right cosets of* $\mathrm{N}_G(U(q))$, i.e. *on a set of size* $q^3 + 1$.
(7) $U(q)H(q)$ *is a Frobenius group with Frobenius kernel* $U(q)$.
(8) *The proportion of elements of order* $q-1$ *in* $U(q)H(q)$ *is* $\phi(q-1)/(q-1)$, *where* ϕ *is the Euler totient function.*

For our purposes, we want another set to act (equivalently) on.

Proposition 2.16. *There exists* $\mathcal{O} \subseteq \mathbb{P}^6(\mathbb{F}_q)$ *on which* $G = \mathrm{Ree}(q)$ *acts faithfully and doubly transitively. This set is*

$$\begin{aligned}\mathcal{O} = &\{(0:0:0:0:0:0:1)\} \cup \\ &\{(1 : a^t : -b^t : (ab)^t - c^t : -b - a^{3t+1} - (ac)^t : -c - (bc)^t - a^{3t+2} - a^t b^{2t} : \\ &a^t c - b^{t+1} + a^{4t+2} - c^{2t} - a^{3t+1} b^t - (abc)^t)\}\end{aligned} \tag{2.41}$$

Moreover, the stabiliser of $P_\infty = (0:0:0:0:0:0:1)$ *is* $U(q)H(q)$, *the stabiliser of* $P_0 = (1:0:0:0:0:0:0)$ *is* $(U(q)H(q))^\Upsilon$ *and the stabiliser of* (P_∞, P_0) *is* $H(q)$.

Proof. Notice that $\mathcal{O} \setminus \{P_\infty\}$ consists of the first rows of the elements of $U(q)H(q)$. From Proposition 2.15 it follows that G is the disjoint union of $U(q)H(q)$ and $U(q)H(q)\Upsilon U(q)H(q)$. Define a map between the G-sets as $(U(q)H(q))g \mapsto P_\infty g$.

If $g \in U(q)H(q)$ then $P_\infty g = P_\infty$ and hence the stabiliser of P_∞ is $U(q)H(q)$. If $g \notin U(q)H(q)$ then $g = x\Upsilon y$ where $x, y \in U(q)H(q)$. Hence $P_\infty g = P_0 y \in \mathcal{O}$ since $P_0 y$ is the first row of y. It follows that the map defines an equivalence between the G-sets. □

Proposition 2.17. *Let* $G = \mathrm{Ree}(q)$.

(1) *The stabiliser in* G *of any two distinct points of* $\mathcal{O}$ *is conjugate to* $H(q)$, *and the stabiliser of any triple of points has order* 2.
(2) *The number of elements in* G *that fix exactly one point is* $q^6 - 1$.
(3) *All involutions in* G *are conjugate in* G.
(4) *An involution fixes* $q+1$ *points.*

Proof. (1) Immediate from [**HB82**, Chapter 11, Theorem 13.2(d)].
(2) A stabiliser of a point is conjugate to $U(q)H(q)$, and there are $|\mathcal{O}|$ conjugates. The elements fixing exactly one point are the non-trivial elements of $U(q)$. Therefore the number of such elements is $|\mathcal{O}|\,(|U(q)| - 1) = (q^3+1)(q^3-1) = (q^6-1)$.
(3) Immediate from [**HB82**, Chapter 11, Theorem 13.2(e)].
(4) Each involution is conjugate to

$$h(-1) = \mathrm{diag}(-1, 1, -1, 1, -1, 1, -1).$$

Evidently, $h(-1)$ fixes P_∞ since $h(-1) \in H(q)$ and if $P = (p_1 : \cdots : p_7) \in \mathcal{O}$ with $p_1 \neq 0$, then P is fixed by $h(-1)$ if and only if $p_2 = p_4 = p_6 = 0$. But then P is uniquely determined by p_3, so there are q possible choices for P. Thus the number of points fixed by $h(-1)$ is $q + 1$.

□

PROPOSITION 2.18. *Let $G = \mathrm{Ree}(q)$ with natural module V and let $j \in G$ be an involution. Then $V|_{\mathrm{C}_G(j)} \cong S_j \oplus T_j$ where $\dim S_j = 3$ and $\dim T_j = 4$. Moreover, S_j is irreducible and j acts trivially on S_j.*

PROOF. By Proposition 2.17, j is conjugate to $h(-1) = \mathrm{diag}\,(-1, 1, -1, 1, -1, 1, -1)$ so it has two eigenspaces S_j and T_j for 1 and -1 respectively. Clearly $\dim S_j = 3$ and $\dim T_j = 4$, and it is sufficient to show that these are preserved by $\mathrm{PSL}(2, q)$, so that they are in fact submodules of V_j.

Let $v \in S_j$ and $g \in \mathrm{PSL}(2, q)$. Then $(vg)j = (vj)g = vg$ since g centralises j and j fixes v, which shows that $vg \in S_j$, so this subspace is fixed by $\mathrm{PSL}(2, q)$. Similarly, T_j is also fixed.

Let $\gamma : V \times V \to V$ be the bilinear form preserved by G. Observe that if $x \in S_j$, $y \in V_j$ then $\gamma(x, y) = \gamma(xj, yj) = \gamma(x, -y) = -\gamma(x, y) = 0$ and hence $V_j \subseteq S_j^{\perp}$. If $\gamma|_{S_j}$ is degenerate then also $S_j \subseteq S_j^{\perp}$ so that $S_j \subseteq V^{\perp}$ which is impossible since γ is non-degenerate. Hence $\gamma|_{S_j}$ is non-degenerate and S_j is isomorphic to its dual.

Now if S_j is reducible, it must split as a direct sum of two submodules of dimension 1 and 2. Since j acts trivially on S_j, it is in fact a module for $\mathrm{PSL}(2, q)$, but $\mathrm{PSL}(2, q)$ have no irreducible modules of dimension 2. Therefore S_j must be irreducible. □

LEMMA 2.19. *Let $g \in G \leqslant \mathrm{GL}(d, F)$ with d odd and F any finite field, and assume that G preserves a non-degenerate bilinear form and that $\det(g) = 1$. Then g has 1 as an eigenvalue.*

PROOF. Let $V = F^d$ be the natural module of G. Let $f : V \times V \to F$ be a non-degenerate bilinear form preserved by G. Over $\bar{F}$, the multiset of eigenvalues of g is $\lambda_1, \ldots, \lambda_d$. Let $E_{\lambda_1}, \ldots, E_{\lambda_d}$ be the corresponding eigenspaces (some of them might be equal).

If $v \in E_{\lambda_i}$ and $w \in E_{\lambda_j}$ then

$$f(v, w) = f(vg, wg) = f(v\lambda_i, w\lambda_j) = \lambda_i \lambda_j f(v, w)$$

so either $\lambda_i \lambda_j = 1$ or $f(v, w) = 0$. However, for a given i there must be some j such that $\lambda_i \lambda_j = 1$, otherwise $f(v, w) = 0$ for every $v \in E_{\lambda_i}$ and $w \in V$, which is impossible since f is non-degenerate.

Hence the eigenvalues can be arranged into pairs of inverse values. Since d is odd, there must be a k such that λ_k is left over. The above argument then implies that $\lambda_k^2 = 1$, and finally $1 = \det(g) = \lambda_k$. □

From [**LN85**] and [**Kle88**] we obtain

PROPOSITION 2.20. *A maximal subgroup of* $G = \mathrm{Ree}(q)$ *is conjugate to one of the following subgroups*

- $\mathrm{N}_G(U(q)) = U(q)H(q)$*, the point stabiliser .*
- $\mathrm{C}_G(j) \cong \langle j \rangle \times \mathrm{PSL}(2, q)$*, the centraliser of an involution* j*.*
- $\mathrm{N}_G(A_0) \cong (\mathrm{C}_2 \times \mathrm{C}_2 \times A_0){:}\,\mathrm{C}_6$*, where* $A_0 \leqslant \mathrm{Ree}(q)$ *is cyclic of order* $(q + 1)/4$.
- $\mathrm{N}_G(A_1) \cong A_1{:}\,\mathrm{C}_6$*, where* $A_1 \leqslant \mathrm{Ree}(q)$ *is cyclic of order* $q + 1 - 3t$*.*
- $\mathrm{N}_G(A_2) \cong A_2{:}\,\mathrm{C}_6$*, where* $A_2 \leqslant \mathrm{Ree}(q)$ *is cyclic of order* $q + 1 + 3t$*.*
- $\mathrm{Ree}(s)$ *where* q *is a proper power of* s*.*

Moreover, all maximal subgroups except the last are reducible.

PROOF. It is sufficient to prove the final statement.

Clearly the point stabiliser is reducible, and the involution centraliser is reducible by Proposition 2.18.

Let H be a normaliser of a cyclic subgroup and let x be a generator of the cyclic subgroup that is normalised. Since $G \leqslant \mathrm{SO}(7, q)$, by Lemma 2.19, x has an eigenspace E for the eigenvalue 1, where $V \neq E \neq \{0\}$. Given $v \in E$ and $h \in H$, we see that $(vh)x^h = vh$ so that vh is fixed by $\langle x^h \rangle = \langle x \rangle$. This implies that $vh \in E$ and thus E is a proper non-trivial H-invariant subspace, so H is reducible. □

PROPOSITION 2.21. *Let* $G = \mathrm{Ree}(q)$*.*

(1) *All cyclic subgroups of* $G = \mathrm{Ree}(q)$ *of order* $q - 1$ *are conjugate to* $H(q)$ *and hence each one is a stabiliser of two points of* $\mathcal{O}$*.*
(2) *All cyclic subgroups of order* $(q + 1)/2$ *or* $q \pm 3t + 1$ *are conjugate.*
(3) *If* C *is a cyclic subgroup of order* $q \pm 3t + 1$*, then distinct conjugates of* C *intersect trivially.*
(4) *If* C *is a cyclic subgroup of order* $(q + 1)/2$*,* $C > C' \cong A_0$ *and* $x \in G \setminus \mathrm{N}_G(C')$*, then* $C \cap C^x = \langle 1 \rangle$*.*

PROOF. (1) Let $C = \langle g \rangle \leqslant G$ be cyclic of order $q - 1$ and let p be an odd prime such that $p \mid q - 1$. Then there exists $k \in \mathbb{Z}$ such that $\left| g^k \right| = p$. Since $q^3 + 1 \equiv 2 \pmod{p}$, the cycle structure of g^k on $\mathcal{O}$ must be a number of p-cycles and 2 fixed points P and Q. Since G is doubly transitive there exists $x \in G$ such that $Px = P_\infty$ and $Qx = P_0$.

Now either g fixes P and Q or interchanges them, so $g^x \in \mathrm{N}_G(H(q)) = \langle H(q), \Upsilon \rangle \cong \mathrm{D}_{2(q-1)}$. Hence $\langle g^x \rangle = H(q)$ since that is the unique cyclic subgroup of order $q - 1$ in $\langle H(q), \Upsilon \rangle$.

(2) This follows immediately from [**LN85**, Lemma 2].
(3) Let C be such a cyclic subgroup. If $C \neq C^x$ for some $x \in G$ and $g \in C \cap C^x$, then $\mathrm{C}_G(g) = \langle C \cup C^x \rangle$. But $\langle C \cup C^x \rangle = G$, so that $g = 1$.
(4) Since $\langle C \cup C^x \rangle = G$, this is analogous to the previous case.

□

PROPOSITION 2.22. *Let* $G = \mathrm{Ree}(q)$ *and let* ϕ *be the Euler totient function.*

(1) *The centraliser of an involution* $j \in G$ *is isomorphic to* $\langle j \rangle \times \mathrm{PSL}(2, q)$ *and hence has order* $q(q^2 - 1)$.
(2) *The number of involutions in* G *is* $q^2(q^2 - q + 1)$.
(3) *The number of elements in* G *of order* $q - 1$ *is* $\phi(q - 1)q^3(q^3 + 1)/2$.
(4) *The number of elements in* G *of order* $(q + 1)/2$ *is* $\phi((q + 1)/2)q^3(q - 1)(q^2 - q + 1)/6$.
(5) *The number of elements in* G *of order* $(q \pm 3t + 1)$ *is* $\phi(q \pm 3t + 1)q^3(q^2 - 1)(q \mp 3t + 1)/6$.
(6) *The number of elements in* G *of even order is* $q^2(7q^5 - 23q^4 + 8q^3 + 23q^2 - 39q + 24)/24$.
(7) *The number of elements in* G *that fix at least one point is* $q^2(q^5 - q^4 + 3q^2 - 5q + 2)/2$

PROOF. (1) Immediate from [**HB82**, Chapter 11].

(2) All involutions are conjugate, and the index in G of the involution centraliser is
$$\frac{q^3(q^3 + 1)(q - 1)}{q(q^2 - 1)} = q^2(q^2 - q + 1) \tag{2.42}$$
where we have used the fact that $q^3 + 1 = (q + 1)(q^2 - q + 1)$.

(3) By Proposition 2.21, each cyclic subgroup of order $q - 1$ is a stabiliser of two points and is uniquely determined by the pair of points that it fixes. Hence the number of cyclic subgroups of order $q - 1$ is
$$\left| \binom{\mathcal{O}}{2} \right| = \frac{q^3(q^3 + 1)}{2}. \tag{2.43}$$
By Proposition 2.17, the intersection of two distinct subgroups has order 2, so the number of elements of order $q - 1$ is the number of generators of all these subgroups.

(4) By Proposition 2.21, the number of cyclic subgroups of order $(q + 1)/2$ is $[G : \mathrm{N}_G(A_0)] = q^3(q - 1)(q^2 - q + 1)/6$. Since distinct conjugates intersect trivially, the number of elements of order $(q + 1)/2$ is the number of generators of these subgroups.

(5) Analogous to the previous case.

(6) By [**LN85**, Lemma 2], every element of even order lies in a cyclic subgroup of order $q - 1$ or $(q + 1)/2$. In each cyclic subgroup of order $q - 1$ there is a unique involution and hence $(q - 3)/2$ non-involutions of even order, and similarly $(q - 3)/4$ in a cyclic subgroup of order $(q + 1)/2$. By Proposition 2.21 the total number of elements of even order is therefore
$$\begin{aligned}(q - 3)(q^3 + 1)q^3/4 + (q - 3)(q - 1)(q^2 - q + 1)q^3/24 + q^2(q^2 - q + 1) \\ = q^2(7q^5 - 23q^4 + 8q^3 + 23q^2 - 39q + 24)/24\end{aligned} \tag{2.44}$$

(7) The only non-trivial elements of G that fix more than 2 points are involutions. Hence in each cyclic subgroup of order $q - 1$ there are $q - 3$ elements that fix exactly 2 points, so by Proposition 2.17, the number of elements

that fix at least one point is

$$\begin{aligned} q^6 + \frac{(q-3)(q^3+1)q^3}{2} + q^2(q^2-q+1) = \\ = \frac{q^2(q^5 - q^4 + 3q^2 - 5q + 2)}{2} \end{aligned} \tag{2.45}$$

□

LEMMA 2.23. *If $g \in G = \mathrm{Ree}(q)$ is uniformly random, then*

$$\Pr[|g| = q-1] = \frac{\phi(q-1)}{2(q-1)} > \frac{1}{12\log\log(q)} \tag{2.46}$$

$$\Pr[|g| = q \pm 3t + 1] = \frac{\phi(q \pm 3t + 1)}{6(q \pm 3t + 1)} > \frac{1}{36\log\log(q)} \tag{2.47}$$

$$\Pr[|g| = (q+1)/2] = \frac{\phi((q+1)/2)}{6(q+1)} > \frac{1}{36\log\log(q)} \tag{2.48}$$

$$\Pr[|g| \text{ even}] = \frac{7q^2 - 9q - 24}{24q(q+1)} > 1/4 \tag{2.49}$$

$$\Pr[g \text{ fixes a point}] = \frac{-2 + 3q + q^4}{2(q+q^4)} \geqslant 1/2 \tag{2.50}$$

PROOF. In each case, the first equality follows from Proposition 2.22 and Proposition 2.15. In the first case, the inequality follows from [**MSC96**, Section II.8], and in the other cases the inequalities are clear since $m > 0$. □

COROLLARY 2.24. *In $G = \mathrm{Ree}(q)$, the expected number of random selections required to obtain an element of order $q-1$, $q \pm 3t + 1$ or $(q+1)/2$ is $\mathrm{O}(\log\log q)$, and $\mathrm{O}(1)$ to obtain an element that fixes a point, or an element of even order.*

PROOF. Clearly the number of selections is geometrically distributed, where the success probabilities for each selection are given by Lemma 2.23. Hence the expectations are as stated. □

PROPOSITION 2.25. *Elements in $\mathrm{Ree}(q)$ of order prime to 3 with the same trace are conjugate.*

PROOF. From [**War66**], the number of conjugacy classes of non-identity elements of order prime to 3 is $q-1$. Observe that for $\lambda \in \mathbb{F}_q^\times$, $\mathrm{Tr}(S(0,0,1)\Upsilon h(\lambda)) = \lambda^t - 1$ and $|S(0,0,1)\Upsilon h(\lambda)|$ is prime to 3 if also $\lambda \neq -1$.

Moreover, $h(-1)$ has order 2 and trace -1 so there are $q-1$ possible traces for non-identity elements of order prime to 3, and elements with different trace must be non-conjugate. Thus all conjugacy classes must have different traces. □

PROPOSITION 2.26. *If $i, j \in Ree(q)$ are uniformly random involutions, then $\Pr[|ij| \text{ odd}] > c$ for some constant $c > 0$.*

PROOF. Follows immediately from [**WP06**, Theorem 13]. □

PROPOSITION 2.27. *Let $G = \mathrm{PSL}(2,q)$. If $x, y \in G$ are uniformly random, then*

$$\Pr[\langle x, y\rangle = G] = 1 - \mathrm{O}(\sigma_0(\log(q))/q) \tag{2.51}$$

PROOF. The maximal subgroup $M \leqslant G$ consisting of the upper triangular matrices modulo scalars has index $q+1$, and all subgroups isomorphic to M are conjugate. Since $M = \mathrm{N}_G(M)$, there are $q+1$ conjugates of M.

$$\Pr[\langle x, y \rangle \leqslant M^g \text{ some } g \in G] \leqslant \sum_{i=1}^{q+1} \Pr[\langle x, y \rangle \leqslant M] = \frac{1}{q+1} \tag{2.52}$$

The other maximal subgroups have index strictly greater than $q+1$, so the probability that $\langle x, y \rangle$ lies in any maximal not conjugate to M must be less than $1/(q+1)$. The number of conjugacy classes of maximal subgroups is $\mathrm{O}\big(\sigma_0(\log(q))\big)$, and hence the probability that $\langle x, y \rangle$ lies in a maximal subgroup is $\mathrm{O}\big(\sigma_0(\log(q))/q\big)$. □

PROPOSITION 2.28. *Let $P = (p_1 : \cdots : p_7) \in \mathcal{O}^g$ be uniformly random, where $\mathcal{O}^g = \{Rg \mid R \in \mathcal{O}\}$ for some $g \in \mathrm{GL}(7, q)$. Then*

(1)
$$\Pr[p_3 \neq 0] \geqslant 1 - \frac{tq^2 + 1}{q^3 + 1}. \tag{2.53}$$

(2) *If $Q = (q_1 : \cdots : q_7) \in \mathcal{O}^g$ is given, then*
$$\Pr[p_3 q_2^{3t+1} \neq q_3 p_2^{3t+1}] \geqslant 1 - \frac{1 + q^2(3t+1)}{q^3 + 1}. \tag{2.54}$$

PROOF. If $P = P'g$, where $P_\infty \neq P' \in \mathcal{O}$ and $g = [g_{i,j}]$, then $p_i = g_{1,i} + a^t g_{2,i} - b^t g_{3,i} + ((ab)^t - c^t) g_{4,i} + (-b - a^{3t+1} - (ac)^t) g_{5,i} + (-c - (bc)^t - a^{3t+2} - a^t b^{2t}) g_{6,i} + (a^t c - b^{t+1} + a^{4t+2} - c^{2t} - a^{3t+1} b^t - (abc)^t) g_{7,i}$ for some $a, b, c \in \mathbb{F}_q$.

(1) By introducing indeterminates x, y and z in place of a, b and c, it follows that p_3 is a polynomial $f \in \mathbb{F}_q[x, y, z]$ with $\deg_x(f) \leqslant 4t+2$, $\deg_y(f) \leqslant 2t$ and $\deg_z(f) \leqslant t$. For each $(a, b) \in \mathbb{F}_q^2$, the number of roots of $f(a, b, z)$ is therefore at most t, so the number of roots of f is at most $q^2(3t+1)$.
(2) Similarly, by introducing indeterminates x, y and z in place of a, b and c, it follows that the expression $p_3 q_2^{3t+1} - q_3 p_2^{3t+1}$ is a polynomial $f \in \mathbb{F}_q[x, y, z]$ with $\deg_x(f) \leqslant 10t+6$, $\deg_y(f) \leqslant 5t+1$ and $\deg_z(f) \leqslant 3t+1$. For each $(a, b) \in \mathbb{F}_q^2$, the number of roots of $f(a, b, z)$ is therefore at most $3t+1$, so the number of roots of f is at most $q^2(3t+1)$.

□

PROPOSITION 2.29. *If $g_1, g_2 \in U(q)H(q)$ are uniformly random and independent, then*
$$\Pr[|[g_1, g_2]| = 9] = 1 - \frac{1}{q-1} \tag{2.55}$$

PROOF. By Proposition 2.15, $[g_1, g_2] \in U(q)$ and has order 9 if and only if $[g_1, g_2] \notin U(q)' \triangleleft U(q)H(q)$. It is therefore sufficient to find the proportion of (unordered) pairs $k_1, k_2 \in U(q)H(q)/U(q)' = A$ such that $[k_1, k_2] = 1$.

If $k_1 = 1$ then k_2 can be any element of A, which gives $q(q-1)$ pairs. If $1 \neq k_1 \in U(q)/U(q)' \cong \mathbb{F}_q$ then $\mathrm{C}_A(k_1) = U(q)/U(q)'$, so we again obtain $q(q-1)$ pairs. Finally, if $k_1 \notin U(q)$ then $|\mathrm{C}_A(k_1)| = q-1$ so we obtain $q(q-2)(q-1)$ pairs.

Thus we obtain $q^2(q-1)$ pairs from a total of $|A \times A| = q^2(q-1)^2$ pairs, and the result follows. □

2.2.2. Alternative definition. The definition of $\mathrm{Ree}(q)$ that we have given is the one that best suits most our purposes. However, to deal with non-constructive recognition, we need to mention the more common definition of $\mathrm{Ree}(q)$.

Following [**Wil05**, Chapter 4] and [**Wil06**], the exceptional group $G_2(q)$ is constructed by considering the Cayley algebra $\mathbb{O}$ (the octonion algebra), which has dimension 8, and defining $G_2(q)$ to be the automorphism group of $\mathbb{O}$. Thus each element of $G_2(q)$ fixes the identity and preserves the algebra multiplication, and it follows that it is isomorphic to a subgroup of $\mathrm{SO}(7, q)$.

Furthermore, when q is an odd power of 3, the group $G_2(q)$ has a certain automorphism Ψ, sometimes called the *exceptional outer automorphism*, whose set of fixed points form a group, and this is defined to be the Ree group $\mathrm{Ree}(q) = {}^2\mathrm{G}_2(q)$. A more detailed description of how to compute $\Psi(g)$ can be found in [**Wil06**].

2.2.3. Tensor indecomposable representations. It follows from [**Lüb01**] that over an algebraically closed field in defining characteristic, up to Galois twists, there are precisely two absolutely irreducible tensor indecomposable representations of $\mathrm{Ree}(q)$: the natural representation and a representation of dimension 27. Let V be the natural module of $\mathrm{Ree}(q)$, of dimension 7. The symmetric square $\mathcal{S}^2(V)$ has dimension 28, and is a direct sum of two submodules of dimensions 1 and 27. The 1-dimensional submodule arises because $\mathrm{Ree}(q)$ preserves a quadratic form.

2.3. Big Ree groups

The Big Ree groups were first described in [**Ree61a, Ree61b**], and are covered in [**Wil05**, Chapter 4]. The maximal subgroups are given in [**Mal91**], and representatives of the conjugacy classes are given in [**Shi74**] and [**Shi75**]. An elementary construction, suitable for our purposes, is described in [**Wil06**].

2.3.1. Definition and properties. We take the definition of the standard copy of ${}^2\mathrm{F}_4(q)$ from [**Wil06**].

The exceptional group $\mathrm{F}_4(q)$ is constructed by considering the exceptional Jordan algebra (the Albert algebra), which has dimension 27, and defining $\mathrm{F}_4(q)$ to be its automorphism group. Thus each element of $\mathrm{F}_4(q)$ fixes the identity and preserves the algebra multiplication, and one can show that it is a subgroup of $\mathrm{O}^-(26, q)$.

Furthermore, when q is an odd power of 2, the group $\mathrm{F}_4(q)$ has a certain automorphism Ψ whose set of fixed points form a group, and this is defined to be the Big Ree group ${}^2\mathrm{F}_4(q)$. A more detailed description of how to compute $\Psi(g)$ can be found in [**Wil06**].

Let $t = 2^{m+1} = \sqrt{2q}$ and let $V = \mathbb{F}_q^{26}$. From [**Wil06**] we immediately obtain:

PROPOSITION 2.30. *Let $G = {}^2\mathrm{F}_4(q)$ and $g \in G$ with $|g| = q-1$. Then g is conjugate in G to an element of the form*

$$\begin{aligned}\varsigma(a,b) = \mathrm{diag}(&a, b, a^{t-1}b^{t-1}, ab^{1-t}, a^tb^{-1}, ba^{1-t}, b^{t-1}, b^{1-t}a^{2-t}, ab^{-1}, a^{t-1}, 1, a^{-1}b^t,\\ &b^{2-t}a^{1-t}, a^{t-1}b^{t-2}, ab^{-t}, 1, a^{1-t}, ba^{-1}, a^{t-2}b^{t-1}, b^{1-t}, a^{t-1}b^{-1},\\ &ba^{-t}, a^{-1}b^{t-1}, b^{1-t}a^{1-t}, b^{-1}, a^{-1})\end{aligned} \tag{2.56}$$

for some $a, b \in \mathbb{F}_q^\times$.

Let $\{e_1, \dots, e_{26}\}$ be the standard basis of V. Following [**Wil06**], we then define the following matrices as permutations on this basis:

$$\varrho = (15,12)(14,13)(2,5)(7,8)(19,20)(22,25)(3,4)(6,9)(18,21)(24,23) \tag{2.57}$$

$$\kappa = \begin{matrix}(11,16)(1,2)(8,13)(14,19)(25,26)(7,10)\\ (5,12)(15,22)(17,20)(4,6)(9,18)(21,23)\end{matrix} \tag{2.58}$$

We also define linear transformations z and ν, where z fixes e_i for $i = 1, \dots, 15$ and otherwise acts as follows:

$$e_{16} \mapsto e_1 + e_{16} \qquad e_{17} \mapsto e_1 + e_{17} \qquad e_{18} \mapsto e_2 + e_{18} \tag{2.59}$$

$$e_{19} \mapsto e_3 + e_{19} \qquad e_{20} \mapsto e_4 + e_{20} \qquad e_{21} \mapsto e_5 + e_{21} \tag{2.60}$$

$$e_{22} \mapsto e_2 + e_6 + e_{22} \qquad e_{23} \mapsto e_3 + e_7 + e_{23} \qquad e_{24} \mapsto e_4 + e_8 + e_{24} \tag{2.61}$$

$$e_{25} \mapsto e_5 + e_9 + e_{25} \qquad e_{26} \mapsto e_1 + e_{10} + e_{11} + e_{26} \tag{2.62}$$

Furthermore, we define a block-diagonal matrix ζ as follows:

$$\zeta_1 = \begin{bmatrix}1&0&0&0\\1&1&0&0\\1&1&1&0\\1&0&1&1\end{bmatrix} \tag{2.63}$$

$$\zeta_2 = \begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&1&1&0&0&0\\1&1&1&1&0&0\\1&1&0&1&1&0\\0&0&1&0&0&1\end{bmatrix} \tag{2.64}$$

$$\zeta_3 = \begin{bmatrix}1\end{bmatrix} \tag{2.65}$$

and then ζ has diagonal blocks $\zeta_3, \zeta_1, \zeta_1, \zeta_3, \zeta_2, \zeta_3, \zeta_1, \zeta_1, \zeta_3$.

Finally, ν fixes e_i for $i \in \{1, 3, 4, 5, 8, 9, 14, 15, 21, 24, 25\}$ and otherwise acts as follows:

$$e_2 \mapsto e_1 + e_2 \qquad e_6 \mapsto e_4 + e_6 \qquad e_7 \mapsto e_5 + e_7 \tag{2.66}$$

$$e_{10} \mapsto e_5 + e_{10} \qquad e_{11} \mapsto e_9 + e_{11} \qquad e_{12} \mapsto e_5 + e_7 + e_{10} + e_{12} \tag{2.67}$$

$$e_{13} \mapsto e_8 + e_{13} \qquad e_{16} \mapsto e_9 + e_{16} \qquad e_{17} \mapsto e_{15} + e_{17} \tag{2.68}$$

$$e_{18} \mapsto e_9 + e_{11} + e_{16} + e_{18} \qquad e_{19} \mapsto e_{14} + e_{19} \qquad e_{20} \mapsto e_{15} + e_{20} \tag{2.69}$$

$$e_{22} \mapsto e_{15} + e_{17} + e_{20} + e_{22} \qquad e_{23} \mapsto e_{21} + e_{23} \qquad e_{26} \mapsto e_{25} + e_{26} \tag{2.70}$$

From [**Wil06**] we then immediately obtain:

THEOREM 2.31. *Let λ be a primitive element of $\mathbb{F}_q$. The elements ϱ, κ, z, ν and ζ lie in ${}^2\mathrm{F}_4(q)$, and if $G = \langle \varsigma(1,\lambda), \varrho\nu\zeta \rangle$, then $G \cong {}^2\mathrm{F}_4(q)$.*

PROPOSITION 2.32. *Let $G = {}^2\mathrm{F}_4(q)$ and $g \in G$ with $|g| = (q-1)(q+t+1)$. Then g^{q+t+1} is conjugate in G to $\varsigma(1,b)$, which we write as*

$$\begin{split} h(\lambda,\mu) = \operatorname{diag}(1, \lambda, \mu\lambda^{-1}, \lambda\mu^{-1}, \lambda^{-1}, \lambda, \mu\lambda^{-1}, \lambda\mu^{-1}, \lambda^{-1}, 1, 1, \mu, \lambda^2\mu^{-1}, \mu\lambda^{-2}, \\ \mu^{-1}, 1, 1, \lambda, \mu\lambda^{-1}, \lambda\mu^{-1}, \lambda^{-1}, \lambda, \mu\lambda^{-1}, \lambda\mu^{-1}, \lambda^{-1}, 1) \end{split} \tag{2.71}$$

where $\lambda = b \in \mathbb{F}_q^\times$ and $\mu = \lambda^t$.

PROOF. Using the notation of [**Shi75**, Page 10], we see that with respect to a suitable basis, g lies in $T(4) \cong \mathrm{C}_{q-1} \times \mathrm{C}_{q+t+1}$ and that g^{q+t+1} will have the form $(\epsilon, \epsilon^{2\theta-1}, 1, 1)$ for some $\epsilon \in \mathbb{F}_q^\times$. Hence it will lie in one of the factors of $T(1) \cong \mathrm{C}_{q-1}^2$. □

PROPOSITION 2.33. *If $g \in G = {}^2\mathrm{F}_4(q)$ is uniformly random, then*

$$\Pr[|g| = (q-1)(q+t+1)] \geqslant \frac{5349}{54080 \log\log(q)} \approx \frac{1}{10 \log\log(q)} \tag{2.72}$$

and hence the expected number of random selections required to obtain an element of order $(q-1)(q+t+1)$ is $\mathrm{O}(\log\log q)$.

PROOF. Using the notation of [**Shi75**, Page 10], we see that such elements lie in a conjugate of $T(4) \cong \mathrm{C}_{q-1} \times \mathrm{C}_{q+t+1}$. The proportion of the elements in $T(4)$ is therefore $\phi(q-1)\phi(q+t+1)/((q-1)(q+t+1))$.

From [**Shi75**, Table IV] we see that the elements in $T(4)$ are of types t_9 and t_{10} and that the total number of such elements in G is

$$\frac{(q+t)\,|G|}{4q^2(q+t+1)(q-1)(q^2+1)} + \frac{(q-2)(q+t)\,|G|}{8(q-1)(q+t+1)}.$$

The proportion in G of the elements of the required order is therefore at least

$$\frac{\phi(q-1)\phi(q+t+1)}{(q-1)(q+t+1)} \left(\frac{(q+t)}{4q^2(q+t+1)(q-1)(q^2+1)} + \frac{(q-2)(q+t)}{8(q-1)(q+t+1)} \right)$$

and the expression in parentheses is minimised when $m = 1$. The result now follows from [**MSC96**, Section II.8]. □

By definition we have the inclusions ${}^2\mathrm{F}_4(q) < \mathrm{F}_4(q) < \mathrm{O}^-(26,q) < \mathrm{Sp}(26,q) < \mathrm{SL}(26,q) < \mathrm{GL}(26,q)$, so ${}^2\mathrm{F}_4(q)$ preserves a quadratic form Q^* with associated bilinear form β^*. It follows from [**Wil06**] that

$$Q^*(e_i) = \begin{cases} 1 & i \in \{11, 16\} \\ 0 & i \notin \{11, 16\} \end{cases} \tag{2.73}$$

$$\beta^*(e_i, e_j) = \delta_{i,27-j} \tag{2.74}$$

PROPOSITION 2.34. *In $G = {}^2\mathrm{F}_4(q)$, there are two conjugacy classes of involutions. The* rank *of an involution is the number of 2-blocks in the Jordan form.*

Name	*Centraliser*	*Maximal parabolic*	*Rank*	*Representative*
$2A$	$[q^{10}]\colon \mathrm{Sz}(q)$	$[q^{10}]{:}(\mathrm{Sz}(q) \times \mathrm{C}_{q-1})$	10	ϱ
$2B$	$[q^9]\colon \mathrm{PSL}(2,q)$	$[q^{11}]{:}(\mathrm{PSL}(2,q) \times \mathrm{C}_{q-1})$	12	κ

Moreover, in the $2A$ case the centre of the centraliser has order q. The cyclic group C_{q-1} acts fixed-point freely on $[q^{10}]$.

Let $i, j \in G$ be involutions. If i and j are conjugate, then $|ij|$ is odd with probability $1 - \mathrm{O}(1/q)$. If i and j are not conjugate, then $|ij|$ is even.

PROOF. The structure of the centralisers can be found in [**Mal91**]. The statement about i and j follows immediately from [**WP06**, Theorem 13]. □

COROLLARY 2.35 (The dihedral trick). *There exists a Las Vegas algorithm that, given $\langle X\rangle \leqslant \mathrm{GL}(26, q)$ such that $\langle X\rangle = {}^2\mathrm{F}_4(q)$ and given conjugate involutions $a, b \in \langle X\rangle$, finds $c \in \langle X\rangle$ such that $a^c = b$. If a, b are given as* `SLP`*s of length $\mathrm{O}(n)$, then c will be found as an* `SLP` *of length $\mathrm{O}(n)$. The algorithm has expected time complexity $\mathrm{O}(\xi)$ field operations.*

PROOF. Follows from Proposition 2.34 and Theorem 1.9.

□

PROPOSITION 2.36. *Let λ be a primitive element of $\mathbb{F}_q$. A maximal parabolic of type $2A$ is conjugate to $\langle \zeta, z^{\kappa\varrho\kappa}, \varrho, \nu, \varsigma(1,\lambda), \varsigma(\lambda, 1)\rangle$, which consists of lower block-triangular matrices. A maximal parabolic of type $2B$ is conjugate to*

$$\langle \varsigma(1,\lambda), \varsigma(\lambda,1), \kappa, \nu, \zeta, \zeta^{\kappa}, \zeta^{\kappa\varrho}, \zeta^{\kappa\varrho\kappa}, \nu^{\varrho}, \nu^{\varrho\kappa}, \nu^{\varrho\kappa\varrho}\rangle .$$

PROOF. Follows immediately from [**Wil06**]. □

CONJECTURE 2.37. *Let $j \in G = {}^2\mathrm{F}_4(q)$ be an involution of class $2A$ and let $H \leqslant \mathrm{C}_G(j)$ satisfy $H \geqslant \mathrm{Z}(\mathrm{C}_G(j))$ and $H \geqslant S$ where $S \cong \mathrm{Sz}(q)$.*

(1) *Let $g \in H$ be uniformly random such that $|g| = 2l$. Then $g^l \in \mathrm{Z}(\mathrm{C}_G(j))$ with probability $1 - \mathrm{O}(1/q)$.*
(2) *If $H = \mathrm{C}_G(j)$ and $g \in H$ is uniformly random such that $|g| = 4l$, then with high probability $g^l \in \mathrm{O}_2(H)$ and $g^{2l} \in \mathrm{Z}(H)$.*

CONJECTURE 2.38. *Let $j_1, j_2 \in G = {}^2\mathrm{F}_4(q)$ be involutions of class $2A$ such that $j_2 \in \mathrm{Z}(\mathrm{C}_G(j_1))$. Then the proportion of $h \in G$ such that $j_1^h = j_2$, $|h| = q - 1$ and $\langle \mathrm{C}_G(j_1), h\rangle < G$ is high.*

Conjecture 2.39. *Let $j \in G = {}^2\mathrm{F}_4(q)$ be an involution. Let $H \leqslant \mathrm{C}_G(j)$ satisfy $H \geqslant \mathrm{Z}(\mathrm{C}_G(j))$ and $H \geqslant S$ where $S \cong \mathrm{Sz}(q)$. Let M be the natural module of G.*

Class of j	*Constituents and multiplicities of $M\|_H$*
$2A$	$(S_4, 4)$, $(1, 6)$, $(S_4^{\psi^t}, 1)$
$2B$	$(R_2 \otimes R_2^{\psi^i}, 2)$, $(R_2^{\psi^j}, 2)$, $(R_2, 1)$, $(R_2^{\psi^k}, 4)$, $(1, 4)$

Where S_4 is the natural module for $\mathrm{Sz}(q)$, R_2 is the natural module for $\mathrm{SL}(2, q)$ and $0 \leqslant i, j, k \leqslant 2m$.

In the $2A$ case, $M|_H$ has submodules of dimensions

$$26, 25, 21, 20, 17, 16, 15, 11, 10, 9, 6, 5, 1, 0.$$

It has $q + 1$ submodules of dimensions 10 and 16, and the others are unique.

In the $2B$ case, $M|_H$ has submodules of every dimension $0, \ldots, 26$. It has 2 submodules of dimensions $7, 9, 10, 13, 16, 17, 19$, $q + 1$ submodules of dimensions 11 and 15 and $q + 2$ submodules of dimensions 12 and 14. All the others are unique.

Conjecture 2.40. *Let $j \in G = {}^2\mathrm{F}_4(q)$ be an involution of class $2A$, let $P = \mathrm{O}_2(\mathrm{C}_G(j))$ and let H be the corresponding maximal subgroup.*

(1) *P is nilpotent of class 2 and has exponent 4.*
(2) *P has a subnormal series*

$$P = P_0 \triangleright P_1 \triangleright P_2 \triangleright P_3 \triangleright P_4 = 1$$

where $|P_0/P_1| = |P_2/P_3| = q^4$, $|P_1/P_2| = |P_3/P_4| = q$.
(3) *The series induces a filtration of H, so P_i/P_{i+1} are $\mathbb{F}_q H$-modules, for $i = 0, \ldots, 3$.*
(4) *$\Phi(P_0) = P_0' = P_2$, $P_1' = P_3$, P_2 is elementary abelian and $\mathrm{Z}(\mathrm{C}_G(j)) = \mathrm{Z}(P) = P_3$.*
(5) *P has exponent-2 central series*

$$P = P_0 \triangleright P_2 \triangleright P_4 = 1$$

(6) *Pre-images of non-identity elements of P_0/P_1 have order 4 and their squares, which lie in P_2 and have non-trivial images in P_2/P_3, are involutions of class $2B$.*
(7) *Pre-images of non-identity elements of P_1/P_2 have order 4 and their squares, which lie in P_3 and have non-trivial images in P_3/P_4, are involutions of class $2A$.*
(8) *As H-modules, P/P_2 is not isomorphic to P_2.*

By [**Mal91**], G has a maximal subgroup $S \cong \mathrm{Sz}(q) \wr \mathrm{C}_2$, so from Section 2.1 we know that S contains elements of order $4(q - 1)$.

Conjecture 2.41. *Let $H \leqslant G = {}^2\mathrm{F}_4(q)$ be such that $H \cong \mathrm{Sz}(q) \times \mathrm{Sz}(q)$ and let $\mathrm{Sz}(q) \cong S \leqslant H$ be one of its direct factors. Let M be the module of S and let S_4 be the natural module of S.*

(1) $M \cong \left(\oplus_{i=1}^{4} 1 + S_4\right) \oplus (1.S_4^{\psi^t}.1)$.
(2) *M has composition factors with multiplicities:* $(S_4, 4), (1, 6), (S_4^{\psi^t}, 1)$.

Conjecture 2.42. *Let $S \leqslant G = {}^2\mathrm{F}_4(q)$ be such that $S \cong \mathrm{Sz}(q) \times \mathrm{Sz}(q)$ and let M be the module of G.*

(1) *The elements of S of order $4(q-1)$ have 1 as an eigenvalue of multiplicity 6. The proportion of these elements in G (taken over every S) is $1/(2q)$.*
(2) $M|_S \cong M_{16} \oplus M_{10}$ *where* $\dim M_i = i$.
(3) *M_{16} is absolutely irreducible and $M_{16} \cong S_1 \otimes S_2$. M_{10} has shape $1.(S_3 \oplus S_4).1$. The S_i are natural $\mathrm{Sz}(q)$-modules.*
(4) *$M|_S$ has endomorphism ring $\mathrm{End}(M|_S) \cong \mathbb{F}_q^3$ and automorphism group $\mathrm{Aut}(M|_S) \cong \mathrm{C}_2 \times \mathrm{C}_{q-1} \times \mathrm{C}_{q-1}$.*

Proposition 2.43. *Let $S \leqslant G = {}^2\mathrm{F}_4(q)$ such that $S \cong \mathrm{Sz}(q) \times \mathrm{Sz}(q)$ and let M be the module of G. Then $\mathrm{Aut}(M|_S) \cap \mathrm{O}^-(26, q) \cong \mathrm{C}_2$.*

Proof. The subgroup of $\mathrm{O}^-(26, q)$ that preserves the direct sum decomposition of $M|_S$ has shape $\mathrm{O}^+(16, q) \times \mathrm{O}^-(10, q)$ (the 16-dimensional part is of plus type since M_{16} is a tensor product of natural Suzuki modules, and these preserve orthogonal forms of plus type). This implies that

$$\mathrm{Aut}(M|_S) \cap \mathrm{O}^-(26, q) = (\mathrm{Aut}(M_{16}) \cap \mathrm{O}^+(16, q)) \times (\mathrm{Aut}(M_{10}) \cap \mathrm{O}^-(10, q))$$

Since M_{16} is absolutely irreducible, $\mathrm{Aut}(M_{16})$ consists of scalars only, but there are no scalars in $\mathrm{O}^+(16, q)$. Hence it suffices to show that $\mathrm{Aut}(M_{10}) \cap \mathrm{O}^-(10, q) \cong \mathrm{C}_2$.

With respect to a suitable basis, an endomorphism of M_{10} has the form $e(\alpha, \beta) = \alpha I_{10} + \beta E_{10,1}$. It preserves the bilinear form β^* restricted to M_{10} if $\alpha = 1$, and it preserves the quadratic form Q^* restricted to M_{10} if $e(1, \beta)$ is a transvection. This implies that only 2 values of β are possible. ☐

Proposition 2.44. *Let λ be a primitive element of $\mathbb{F}_q$. The subgroup*

$$\langle z, \varrho, \varrho^\kappa, \varsigma(1, \lambda), \varsigma(\lambda, 1)\rangle$$

is isomorphic to $\mathrm{Sz}(q) \wr \mathrm{C}_2$. It contains $S = \langle z^{\kappa\varrho\kappa}, \varrho, \varsigma(1, \lambda)\rangle$ and $h = \kappa\varrho\kappa$, where $S \cong \mathrm{Sz}(q)$ and $[S^h, S] = \langle 1\rangle$. The subgroup $\langle \nu, \varsigma(1, \lambda), \varsigma(\lambda, 1), \kappa, \varrho, \varrho\kappa\varrho\rangle$ is isomorphic to $\mathrm{Sp}(4, q){:}\,\mathrm{C}_2$.

Proof. Follows immediately from [**Wil06**]. ☐

Proposition 2.45. *The proportion of elements $a \in G = {}^2\mathrm{F}_4(q)$, such that $|a| = q - 1$ and a is conjugate to some $h(\lambda, \mu)$ with $\lambda^t = \mu \in \mathbb{F}_q^\times$, is bounded below by a constant $c_1 > 0$.*

Proof. Follows from Proposition 2.33. ☐

Conjecture 2.46. *Let $a \in G = {}^2\mathrm{F}_4(q)$ be such that $|a| = q - 1$ and a is conjugate to some $h(\lambda, \mu)$ with $\lambda^t = \mu \in \mathbb{F}_q^\times$. The proportion of $b \in G$ such that the* *elements in* $\langle a \rangle b$ *with* 1 *as an eigenvalue of multiplicity* 6 *also have even order and power up to an involution of class* $2A$*, is bounded below by a constant* $c_2 > 0$.

CONJECTURE 2.47. *The proportion of elements in* G *that have* 1 *as an eigenvalue of multiplicity* 6 *is* $c_3 \in \mathrm{O}\big(1/q\big)$.

CONJECTURE 2.48. *Let* $a \in G = {}^2\mathrm{F}_4(q)$ *be such that* $|a| = q - 1$ *and* a *is conjugate to some* $h(\lambda, \mu)$ *with* $\lambda^t = \mu \in \mathbb{F}_q^{\times}$. *Then* $\mathrm{N}_G(\langle a \rangle) \cong \mathrm{D}_{2(q-1)} \times \mathrm{Sz}(q)$ *and there exists an absolute constant* c *such that for every* $b \in G \setminus \mathrm{N}_G(\langle a \rangle)$*, the number of* $g \in \langle a \rangle b$ *that have* 1 *as an eigenvalue of multiplicity at least* 6 *is bounded above by* c.

PROPOSITION 2.49. *Let* $G = {}^2\mathrm{F}_4(q)$ *with natural module* M *and let* $H < G$ *be a maximal subgroup. Then either* $M|_H$ *is reducible or* $H \cong {}^2\mathrm{F}_4(s)$ *where* q *is a proper power of* s.

PROOF. Follows from [**Mal91**]. □

2.3.2. Tensor indecomposable representations. It follows from [**Lüb01**] that over an algebraically closed field in defining characteristic, up to Galois twists, there are precisely three absolutely irreducible tensor indecomposable representations of ${}^2\mathrm{F}_4(q)$:

(1) the natural representation V of dimension 26,
(2) a submodule of S of $V \otimes V$ of dimension 246,
(3) a submodule of $V \otimes S$ of dimension 4096

CHAPTER 3

Constructive recognition and membership testing

Here we will present the algorithms for constructive recognition and constructive membership testing. The methods we use are specialised to each family of exceptional groups, so we treat each family separately. When the methods are similar between the families, we present a complete account for each family, in order to make each section self-contained. In the cases where we have a non-constructive recognition algorithm that improves on [**BKPS02**], we will also present it here.

Recall the various cases of constructive recognition of matrix groups, given in Section 1.2.7. For each group, we will deal with some, but not always all, of the cases that arise.

We first give an overview of the various methods. From Chapter 2 we know that both the Suzuki groups and the small Ree groups act doubly transitively on $q^2 + 1$ and $q^3 + 1$ projective points, respectively (the fields of size q have different characteristics), and the idea of how to deal with these groups is to think of them as permutation groups. In fact we proceed similarly as in [**CLGO06**] for $\mathrm{PSL}(2, q)$, which acts doubly transitively on $q + 1$ projective points. The essential problem in all these cases is to find an efficient algorithm that finds an element of the group that maps one projective point to another.

In $\mathrm{PSL}(2, q)$ the algorithm proceeds by finding a random element of order $q - 1$ and considering a random coset of the subgroup generated by this element. Since the coset is exponentially large, we cannot process every element, and the idea is instead to construct the required element by solving equations. We therefore consider a matrix whose entries are indeterminates. In this way we reduce the coset search problem to two other problems from computational algebra: finding roots of quadratic equations over a finite field, and the famous *discrete logarithm problem*.

In the Suzuki groups, the number of points is $q^2 + 1$ instead of $q + 1$, and this requires us to consider double cosets of elements of order $q - 1$, instead of cosets. The problem is again reduced to finding roots of univariate polynomials, in this case of degree at most 60, as well as to the discrete logarithm problem.

The small Ree groups are dealt with slightly differently, since we can easily find involutions by random search and then use the Bray algorithm to find the centraliser of an involution. Then the module of the group restricted to the centraliser decomposes, and this is used to find an element that maps one projective point to another.

The Big Ree groups cannot be considered as permutation groups in the same way, so the essential problem in this case is to find an involution expressed as a product of the given generators. Again the idea is to find a cyclic subgroup of order $q-1$ and search for elements of even order in random cosets of this subgroup. The underlying observation is that the elements of even order are characterised by having 1 as an eigenvalue with a certain multiplicity. Therefore we can again consider matrices whose entries are indeterminates and construct the required element of even order.

3.1. Suzuki groups

Here we will use the notation from Section 2.1. We will refer to Conjectures 3.4, 3.18, 3.19, 3.21 and 3.24 simultaneously as the *Suzuki Conjectures*. We now give an overview of the algorithm for constructive recognition and constructive membership testing. It will be formally proved as Theorem 3.26.

(1) Given a group $G \cong \mathrm{Sz}(q)$, satisfying the assumptions in Section 1.2.7, we know from Section 2.1.3 that the module of G is isomorphic to a tensor product of twisted copies of the natural module of G. Hence we first tensor decompose this module. This is described in Section 3.1.5.
(2) The resulting groups in dimension 4 are conjugates of the standard copy, so we find a conjugating element. This is described in Section 3.1.4.
(3) Finally we are in $\mathrm{Sz}(q)$. Now we can perform preprocessing for constructive membership testing and other problems we want to solve. This is described in Section 3.1.3.

Given a discrete logarithm oracle, the whole process has time complexity slightly worse than $\mathrm{O}\big(d^5 + \log(q)\big)$ field operations, assuming that G is given by a bounded number of generators.

3.1.1. Recognition. We now discuss how to non-constructively recognise $\mathrm{Sz}(q)$. We are given a group $\langle X \rangle \leqslant \mathrm{GL}(4, q)$ and we want to decide whether or not $\langle X \rangle = \mathrm{Sz}(q)$, the group defined in (2.4).

To do this, it suffices to determine if $X \subseteq \mathrm{Sz}(q)$ and if X does not generate a proper subgroup, *i.e.* if X is not contained in a maximal subgroup. To determine if $g \in X$ is in $\mathrm{Sz}(q)$, first determine if g preserves the symplectic form of $\mathrm{Sp}(4, q)$ and then determine if g is a fixed point of the automorphism Ψ of $\mathrm{Sp}(4, q)$, mentioned in Section 2.1.

The recognition algorithm relies on the following result.

LEMMA 3.1. *Let $H = \langle X \rangle \leqslant \mathrm{Sz}(q) = G$, where $X = \{x_1, \dots, x_n\}$, let $C = \{[x_i, x_j] \mid 1 \leqslant i < j \leqslant n\}$ and let M be the natural module of H. Then $H = G$ if and only if the following hold:*

(1) *M is an absolutely irreducible H-module.*
(2) *H cannot be written over a proper subfield.*
(3) *$C \neq \{1\}$ and for every $c \in C \setminus \{1\}$ there exists $x \in X$ such that $[c, c^x] \neq 1$.*

PROOF. By Theorem 2.3, the maximal subgroups of G that do satisfy the first two conditions are $\mathrm{N}_G(\mathcal{H})$, $\mathcal{B}_1$ and $\mathcal{B}_2$. For each, the derived group is contained in the normalised cyclic group, so all these maximal subgroups are metabelian. If H is contained in one of them and H is not abelian, then $C \neq \{1\}$, but $[c, c^x] = 1$ for every $c \in C$ and $x \in X$ since the second derived group of H is trivial. Hence the last condition is not satisfied.

Conversely, assume that $H = G$. Then clearly, the first two conditions are satisfied, and $C \neq \{1\}$. Assume that the last condition is false, so for some $c \in C \setminus \{1\}$ we have that $[c, c^x] = 1$ for every $x \in X$. This implies that $c^x \in \mathrm{C}_G(c) \cap \mathrm{C}_G(c)^{x^{-1}}$, and it follows from Proposition 2.7 that $\mathrm{C}_G(c) = \mathrm{C}_G(c)^{x^{-1}}$. Thus $\mathrm{C}_G(c) = \mathrm{C}_G(c)^g$ for all $g \in G$, so $\mathrm{C}_G(c) \triangleleft G$, but G is simple and we have a contradiction. □

THEOREM 3.2. *There exists a Las Vegas algorithm that, given* $\langle X \rangle \leqslant \mathrm{GL}(4, q)$, *decides whether or not* $\langle X \rangle = \mathrm{Sz}(q)$. *It has expected time complexity* $\mathrm{O}\big(|X|^2 + \sigma_0(\log(q))(|X| + \log(q))\big)$ *field operations.*

PROOF. The algorithm proceeds as follows.

(1) Determine if every $x \in X$ is in $\mathrm{Sz}(q)$, and return `false` if not.
(2) Determine if $\langle X \rangle$ is absolutely irreducible, and return `false` if not.
(3) Determine if $\langle X \rangle$ can be written over a smaller field. If so, return `false`.
(4) Using the notation of Lemma 3.1, try to find $c \in C$ such that $c \neq 1$. Return `false` if it cannot be found.
(5) If such c can be found, and if $[c, c^x] \neq 1$ for some $x \in X$, then return `true`, else return `false`.

From the discussion at the beginning of this section, the first step is easily done using $\mathrm{O}\big(|X|\big)$ field operations. The MeatAxe can be used to determine if the natural module is absolutely irreducible; the algorithm described in Section 1.2.10.2 can be used to determine if $\langle X \rangle$ can be written over a smaller field.

The rest of the algorithm is a straightforward application of the last condition in Lemma 3.1, except that it is sufficient to use the condition for one non-trivial commutator c. By Lemma 3.1, if $[c, c^x] \neq 1$ then $\langle X \rangle = \mathrm{Sz}(q)$; but if $[c, c^x] = 1$, then $\mathrm{C}_{\langle X \rangle}(c) \triangleleft \langle X \rangle$ and we cannot have $\mathrm{Sz}(q)$.

It follows from Section 1.2.10 that the expected time complexity of the algorithm is as stated. Since the MeatAxe is Las Vegas, this algorithm is also Las Vegas. □

We are also interested in determining if a given group is a *conjugate* of $\mathrm{Sz}(q)$, without necessarily finding a conjugating element. We consider the subgroups of $\mathrm{Sp}(4, q)$ and rule out all except those isomorphic to $\mathrm{Sz}(q)$. This relies on the fact that, up to Galois automorphisms, $\mathrm{Sz}(q)$ has only one equivalence class of faithful representations in $\mathrm{GL}(4, q)$, so if we can show that $G \cong \mathrm{Sz}(q)$ then G is a conjugate of $\mathrm{Sz}(q)$.

THEOREM 3.3. *There exists a Las Vegas algorithm that, given $\langle X \rangle \leqslant \mathrm{GL}(4, q)$, decides whether or not there exists $h \in \mathrm{GL}(4, q)$ such that $\langle X \rangle^h = \mathrm{Sz}(q)$. The algorithm has expected time complexity $\mathrm{O}\big(|X|^2 + \sigma_0(\log(q))(|X| + \log(q))\big)$ field operations.*

PROOF. Let $G = \langle X \rangle$. The algorithm proceeds as follows.

(1) Determine if G is absolutely irreducible, and return `false` if not.
(2) Determine if G preserves a non-degenerate symplectic form M. If so we conclude that G is a subgroup of a conjugate of $\mathrm{Sp}(4, q)$, and if not then return `false`. Since G is absolutely irreducible, the form is unique up to a scalar multiple.
(3) Conjugate G so that it preserves the form J. This amounts to finding a symplectic basis, *i.e.* finding an invertible matrix X such that $XJX^T = M$, which is easily done. Then G^X preserves the form J and thus $G^X \leqslant \mathrm{Sp}(4, q)$ so that we can apply Ψ.
(4) Determine if $V \cong V^{\Psi}$, where V is the natural module for G and Ψ is the automorphism from Lemma 2.13. If so we conclude that G is a subgroup of some conjugate of $\mathrm{Sz}(q)$, and if not then return `false`.
(5) Determine if G is a proper subgroup of $\mathrm{Sz}(q)$, *i.e.* if it is contained in a maximal subgroup. This can be done using Lemma 3.1. If so, then return `false`, else return `true`.

From the descriptions in Section 1.2.10.1, the algorithms for finding a preserved form and for module isomorphism testing are Las Vegas, with the same expected time complexity as the MeatAxe. Hence we obtain a Las Vegas algorithm, with the same expected time complexity as the algorithm from Theorem 3.2. □

3.1.2. Finding an element of a stabiliser. In this section the matrix degree is constant, so we set $\xi = \xi(4)$. In constructive membership testing for $\mathrm{Sz}(q)$ the essential problem is to find an element of the stabiliser of a given point $P \in \mathcal{O}$, expressed as an `SLP` in our given generators X of $G = \mathrm{Sz}(q)$. The idea is to map P to $Q \neq P$ by a random $g_1 \in G$, and then compute $g_2 \in G$ such that $Pg_2 = Q$, so that $g_1 g_2^{-1} \in G_P$.

Thus the problem is to find an element that maps P to Q, and the idea is to search for it in double cosets of cyclic subgroups of order $q - 1$. We first give an overview of the method.

Begin by selecting random $a, h \in G$ such that a has pseudo-order $q - 1$, and consider the equation

$$Pa^j h a^i = Q \tag{3.1}$$

in the two indeterminates i, j. If we can solve this equation for i and j, thus obtaining integers k, l such that $1 \leqslant k, l \leqslant q - 1$ and $Pa^l h a^k = Q$, then we have an element that maps P to Q.

Since a has order dividing $q - 1$, by Proposition 2.4, a is conjugate to a matrix $M'(\lambda)$ for some $\lambda \in \mathbb{F}_q^{\times}$. This implies that we can diagonalise a and obtain a matrix $x \in \mathrm{GL}(4, q)$ such that $M'(\lambda)^x = a$. It follows that if we define $P' = Px^{-1}$, $Q' = Qx^{-1}$ and $g = h^{x^{-1}}$ then (3.1) is equivalent to

$$P' M'(\lambda)^j g M'(\lambda)^i = Q'. \tag{3.2}$$

Now change indeterminates to α and β by letting $\alpha = \lambda^j$ and $\beta = \lambda^i$, so that we obtain the following equation:

$$P' M'(\alpha) g M'(\beta) = Q'. \tag{3.3}$$

This determines four equations in α and β, and in Section 3.1.2.1 we will describe how to find solutions for them. A solution $(\gamma, \delta) \in \mathbb{F}_q^{\times} \times \mathbb{F}_q^{\times}$ determines $M'(\gamma), M'(\delta) \in \mathcal{H}$, and hence also $c, d \in H = \mathcal{H}^x$.

If $|a| = q - 1$ then $\langle a \rangle = H$, so that there exist integers k and l as above with $a^l = c$ and $a^k = d$. These integers can be found by computing discrete logarithms, since we also have $\lambda^l = \gamma$ and $\lambda^k = \delta$. Hence we obtain a solution to (3.1) from the solution to (3.3). If $|a|$ is a proper divisor of $q - 1$, then it might happen that $c \notin \langle a \rangle$ or $d \notin \langle a \rangle$, but by Proposition 2.6 we know that this is unlikely.

Thus the overall algorithm is as in Algorithm 3.1. We prove that the algorithm is correct in Section 3.1.2.2.

Algorithm 3.1: FindMappingElement(X, P, Q)

Input: Generating set X for $G = \mathrm{Sz}(q)$ and points $P \neq Q \in \mathcal{O}$.
Output: A random element g of G, as an SLP in X, such that $Pg = Q$.
// Assumes the existence of a function SolveEquation that solves (3.3).
// Also assumes that DiscreteLog returns a positive integer if a
// discrete logarithm exists, and 0 otherwise.
repeat
    // Find random element a of pseudo-order $q - 1$
    repeat
        $a :=$ Random(G)
    until $|a| \mid q - 1$
    $(M'(\lambda), x) :=$ Diagonalise(a)
    // Now $M'(\lambda)^x = a$
    repeat
        $h :=$ Random(G)
        flag $:=$ SolveEquation$(h^{x^{-1}}, Px^{-1}, Qx^{-1})$
    until *flag*
    Let (γ, δ) be a solution to (3.3).
    $l :=$ DiscreteLog(λ, γ)
    $k :=$ DiscreteLog(λ, δ)
until $k > 0$ **and** $l > 0$
return $a^l h a^k$

3.1.2.1. *Solving equation* (3.3). We will now show how to obtain the solutions of (3.3). It might happen that there are no solutions, in which case the method described here will detect this and return with failure.

By letting $P' = (q_1 : q_2 : q_3 : q_4)$, $Q' = (r_1 : r_2 : r_3 : r_4)$ and $g = [g_{i,j}]$, we can write out (3.3) and obtain

$$\begin{aligned} (q_1 g_{1,1}\alpha^{t+1} + q_2 g_{2,1}\alpha + q_3 g_{3,1}\alpha^{-1} + q_4 g_{4,1}\alpha^{-t-1})\beta^{t+1} &= Cr_1 \\ (q_1 g_{1,2}\alpha^{t+1} + q_2 g_{2,2}\alpha + q_3 g_{3,2}\alpha^{-1} + q_4 g_{4,2}\alpha^{-t-1})\beta &= Cr_2 \\ (q_1 g_{1,3}\alpha^{t+1} + q_2 g_{2,3}\alpha + q_3 g_{3,3}\alpha^{-1} + q_4 g_{4,3}\alpha^{-t-1})\beta^{-1} &= Cr_3 \\ (q_1 g_{1,4}\alpha^{t+1} + q_2 g_{2,4}\alpha + q_3 g_{3,4}\alpha^{-1} + q_4 g_{4,4}\alpha^{-t-1})\beta^{-t-1} &= Cr_4 \end{aligned} \tag{3.4}$$

for some constant $C \in \mathbb{F}_q$. Henceforth, we assume that $r_i \neq 0$ for $i = 1, \ldots, 4$, since this is the difficult case, and also very likely when q is large, as can be seen from Proposition 2.10. A method similar to the one described in this section will solve (3.3) when some $r_i = 0$ and Algorithm 3.1 does not assume that all $r_i \neq 0$.

For convenience, we denote the expressions in the parentheses at the left hand sides of (3.4) as K, L, M and N respectively. Since $C = L\beta r_2^{-1}$ we obtain three equations

$$\begin{aligned} K\beta^t &= r_1 r_2^{-1} L \\ M\beta^{-2} &= r_3 r_2^{-1} L \\ N\beta^{-t-2} &= r_4 r_2^{-1} L \end{aligned} \tag{3.5}$$

and in particular β is a function of α, since

$$\beta = \sqrt{L^{-1} M r_3^{-1} r_2}. \tag{3.6}$$

If we eliminate β and β^t by using the first two equations into the third in (3.5), we obtain

$$NKr_2 r_3 = r_1 r_4 ML \tag{3.7}$$

and by raising the first equation to the t-th power and substituting into the second, we obtain

$$r_1 r_3^{t/2} L^{1+t/2} = r_2^{1+t/2} M^{t/2} K. \tag{3.8}$$

Also, $C = M\beta^{-1} r_3^{-1}$ and if we proceed similarly, we obtain two more equations

$$N^t L r_3^{t+1} = M^{t+1} r_2 r_4^t \tag{3.9}$$

$$NL^{t/2} r_3^{1+t/2} = M^{1+t/2} r_4 r_2^{t/2}. \tag{3.10}$$

Now (3.7), (3.8), (3.9) and (3.10) are equations in α only, and by multiplying them by suitable powers of α, they can be turned into polynomial equations such that α only occurs to the powers ti for $i = 1, \ldots, 4$ and to lower powers that are independent of t. The suitable powers of α are $2t + 2$, $t + t/2 + 2$, $2t + 3$ and $2t + t/2 + 2$, respectively.

Thus we obtain the following four equations.

$$\begin{aligned} \alpha^{4t}c_1 + \alpha^{3t}c_2 + \alpha^{2t}c_3 + \alpha^t c_4 &= d_1 \\ \alpha^{4t}c_5 + \alpha^{3t}c_6 + \alpha^{2t}c_7 + \alpha^t c_8 &= d_2 \\ \alpha^{4t}c_9 + \alpha^{3t}c_{10} + \alpha^{2t}c_{11} + \alpha^t c_{12} &= d_3 \\ \alpha^{4t}c_{13} + \alpha^{3t}c_{14} + \alpha^{2t}c_{15} + \alpha^t c_{16} &= d_4 \end{aligned} \tag{3.11}$$

The c_i and d_j are polynomials in α with degree independent of t, for $i = 1, \dots, 16$ and $j = 1, \dots, 4$ respectively, so (3.11) can be considered a linear system in the variables α^{nt} for $n = 1, \dots, 4$, with coefficients c_i and d_j. Now the aim is to obtain a single polynomial in α of bounded degree. For this we need the following conjecture.

CONJECTURE 3.4. *For every $P' = Px^{-1}, Q' = Qx^{-1}, g = h^{x^{-1}}$ where $P, Q \in \mathcal{O}$, $h \in G$ and $x \in \mathrm{GL}(4, q)$, if we regard* (3.11) *as simultaneous linear equations in the variables α^{nt} for $n = 1, \dots, 4$, over the polynomial ring $\mathbb{F}_q[\alpha]$, then it has non-zero determinant.*

In other words, the determinant of the coefficients c_i is not the zero polynomial.

LEMMA 3.5. *Assume Conjecture* 3.4. *Given P', Q' and g as in Conjecture* 3.4, *there exists a univariate polynomial $f(\alpha) \in \mathbb{F}_q[\alpha]$ of degree at most* 60, *such that for every $(\gamma, \delta) \in \mathbb{F}_q^\times \times \mathbb{F}_q^\times$ that is a solution for (α, β) in* (3.3), *we have $f(\gamma) = 0$.*

PROOF. So far in this section we have shown that if we can solve (3.11) we can also solve (3.3). From the four equations of (3.11) we can eliminate α^t. We can solve for α^{4t} from the fourth equation, and substitute into the third, thus obtaining a rational expression with no occurrence of α^{4t}. Continuing this way and substituting into the other equations, we obtain an expression for α^t in terms of the c_i and the d_i only. This can be substituted into any of the equations of (3.11), where α^{nt} for $n = 1, \dots, 4$ is obtained by powering up the expression for α^t. Thus we obtain a rational expression $f_1(\alpha)$ of degree independent of t. We now take $f(\alpha)$ to be the numerator of f_1.

In other words, we think of the α^{nt} as independent variables and of (3.11) as a linear system in these variables, with coefficients in $\mathbb{F}_q[\alpha]$. By Conjecture 3.4 we can solve this linear system.

Two possible problems can occur: f is identically zero or some of the denominators of the expressions for α^{nt}, $n = 1, \dots, 4$ turn out to be 0. However, Conjecture 3.4 rules out these possibilities. By Cramer's rule, the expression for α^t is a rational expression where the numerator is a determinant, so it consists of sums of products of c_i and d_j. Each product consists of three c_i and one d_j. By considering the calculations leading up to (3.11), it is clear that each of the products has degree at most 15. Therefore the expression for α^{4t} and hence also $f(\alpha)$ has degree at most 60.

We have only done elementary algebra to obtain $f(\alpha)$ from (3.11), and it is clear that (3.11) was obtained from (3.4) by elementary means only. Hence all solutions

(γ, δ) to (3.4) must also satisfy $f(\gamma) = 0$, although there may not be any such solutions, and $f(\alpha)$ may also have other zeros. □

COROLLARY 3.6. *Assume Conjecture* 3.4. *There exists a Las Vegas algorithm that, given* P', Q' *and* g *as in Conjecture* 3.4, *finds all* $(\gamma, \delta) \in \mathbb{F}_q^{\times} \times \mathbb{F}_q^{\times}$ *that are solutions of* (3.3). *The algorithm has expected time complexity* $\mathrm{O}\big(\log q\big)$ *field operations.*

PROOF. Let $f(\alpha)$ be the polynomial constructed in Lemma 3.5. To find all solutions to (3.3), we find the zeros γ of $f(\alpha)$, compute the corresponding δ for each zero γ using (3.6), and check which pairs (γ, δ) satisfy (3.4). These pairs must be all solutions of (3.3).

The only work needed is simple matrix arithmetic and finding the roots of a polynomial of bounded degree over $\mathbb{F}_q$. Hence the expected time complexity is $\mathrm{O}\big(\log q\big)$ field operations. The algorithm is Las Vegas, since by Theorem 1.1 the algorithm for finding the roots of $f(\alpha)$ is Las Vegas, with this expected time complexity. □

By following the procedure outlined in Lemma 3.5, it is straightforward to obtain an expression for $f(\alpha)$, where the coefficients are expressions in the entries of g, P' and Q', but we will not display it here, since it would take up too much space.

3.1.2.2. *Correctness and complexity.* There are two issues when considering the correctness of Algorithm 3.1. Using the notation in the algorithm, we have to show that (3.3) has a solution with high probability, and that the integers k and l are positive with high probability.

The algorithm in Corollary 3.6 tries to find an element in the double coset $\mathcal{H}g\mathcal{H}$, where $g = h^{x^{-1}}$, and we will see that this succeeds with high probability when $g \notin \mathrm{N}_G(\mathcal{H})$, which is very likely.

If the element a has order precisely $q-1$, then from the discussion at the beginning of Section 3.1.2, we know that the integers k and l will be positive. By Proposition 2.6 we know that it is likely that a has order precisely $q-1$ rather than just a divisor of $q-1$.

LEMMA 3.7. *Assume Conjecture* 3.4. *Let* $G = \mathrm{Sz}(q)$ *and let* $P \in \mathcal{O}$ *and* $a, h \in G$ *be given, such that* $|a| = q-1$. *Let* $Q \in \mathcal{O}$ *be uniformly random. If* $h \notin \mathrm{N}_G(\langle a \rangle)$, *then*

$$\frac{(q-1)^2}{(q^2+1)\deg f} \leqslant \Pr[Q \in P \langle a \rangle h \langle a \rangle] \leqslant \frac{(q-1)^2}{q^2+1} \tag{3.12}$$

where $f(\alpha)$ *is the polynomial constructed in Lemma* 3.5. *If instead* $h \in \mathrm{N}_G(\langle a \rangle)$ *then*

$$\Pr[Q \in P \langle a \rangle h \langle a \rangle] = \frac{(q-1)(q^2-1)+2}{(q^2+1)^2}. \tag{3.13}$$

PROOF. If $h \notin \mathrm{N}_G(\langle a \rangle)$ then by Lemma 2.7, $|\langle a \rangle h \langle a \rangle| = (q-1)^2$, and hence $|P \langle a \rangle h \langle a \rangle| \leqslant (q-1)^2$.

On the other hand, for every $Q \in \mathcal{O}$ we have

$$|\{(k_1, k_2) \mid k_1, k_2 \in \langle a \rangle, Pk_1hk_2 = Q\}| \leqslant \deg f \tag{3.14}$$

since this is the equation we consider in Section 3.1.2.1, and from Lemma 3.5 we know that all solutions must be roots of f. Thus $|P \langle a \rangle h \langle a \rangle| \geqslant |\langle a \rangle h \langle a \rangle| / \deg f$. Since Q is uniformly random from $\mathcal{O}$, and $|\mathcal{O}| = q^2 + 1$, the result follows.

If $h \in \mathrm{N}_G(\langle a \rangle)$ then $\langle a \rangle h \langle a \rangle = h \langle a \rangle$, and $|Ph \langle a \rangle| = |\langle a \rangle|$ if $\langle a \rangle$ does not fix Ph. By Proposition 2.4, the number of cyclic subgroups of order $q-1$ is $\binom{|\mathcal{O}|}{2}$ and $|\mathcal{O}| - 1$ such subgroups fix Ph. Moreover, if $\langle a \rangle$ fixes Ph then $Ph \langle a \rangle = \{Ph\}$. Thus

$$\begin{aligned} &\Pr[Q \in P \langle a \rangle h \langle a \rangle] = \Pr[Q \in Ph \langle a \rangle] \Pr[Pha \neq Ph] + \\ &+ \Pr[Q = Ph] \Pr[Pha = Ph] = \frac{|Ph \langle a \rangle|}{|\mathcal{O}|} \left(1 - \frac{|\mathcal{O}| - 1}{\binom{|\mathcal{O}|}{2}}\right) + \frac{1}{|\mathcal{O}|} \frac{|\mathcal{O}| - 1}{\binom{|\mathcal{O}|}{2}} \end{aligned} \tag{3.15}$$

and the result follows. □

Theorem 3.8. *Assume Conjecture* 3.4 *and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$. *Algorithm* 3.1 *is a Las Vegas algorithm with expected time complexity* $\mathrm{O}\big((\xi + \chi_D(q) + \log(q) \log\log(q)) \log\log(q)\big)$ *field operations. The length of the returned* `SLP` *is* $\mathrm{O}\big(\log\log(q)\big)$.

Proof. We use the notation from the algorithm. Let $g = h^{x^{-1}}$, $H = \mathcal{H}^x$, $P' = Px^{-1}$ and $Q' = Qx^{-1}$. Corollary 3.6 implies that line 10 will succeed if $Q' \in P' \mathcal{H} g \mathcal{H}$. If $|a| = q - 1$, then $H = \langle a \rangle$, and the previous condition is equivalent to $Q \in P \langle a \rangle h \langle a \rangle$. Moreover, if $|a| = q - 1$ then line 14 will always succeed.

Let s be the probability that the return statement is reached. Then s satisfies the following inequality.

$$\begin{aligned} s \geqslant &\Pr[|a| = q-1](\Pr[h \in \mathrm{N}_G(\langle a \rangle)] \Pr[Q \in P \langle a \rangle h \langle a \rangle \mid h \in \mathrm{N}_G(\langle a \rangle)] + \\ &+ \Pr[h \notin \mathrm{N}_G(\langle a \rangle)] \Pr[Q \in P \langle a \rangle h \langle a \rangle \mid h \notin \mathrm{N}_G(\langle a \rangle)]). \end{aligned} \tag{3.16}$$

Since h is uniformly random, using Theorem 2.3 we obtain

$$\Pr[h \in \mathrm{N}_G(\langle a \rangle)] = \frac{2(q-1)}{|G|} = \frac{2}{q^2(q^2+1)}. \tag{3.17}$$

From Proposition 2.6 and Lemma 3.7 we obtain

$$\begin{aligned} s \geqslant &\frac{\phi(q-1)}{2(q-1)} \\ &\left[\frac{(q-1)^2}{(q^2+1) \deg f} - \frac{2}{q^2(q^2+1)} \frac{(q-1)^2}{(q^2+1)} + \frac{2}{q^2(q^2+1)} \frac{2 + (q-1)(q^2-1)}{(q^2+1)^2}\right] = \\ &= \frac{\phi(q-1)}{2(q-1) \deg f} + \mathrm{O}\big(1/q\big). \end{aligned} \tag{3.18}$$

By Proposition 2.6, the expected number of iterations of the outer repeat statement is $\mathrm{O}\big(\log\log(q)\big)$. The expected number of random selections to find h is $\mathrm{O}\big(1\big)$. By Theorem 1.1, diagonalising a matrix uses expected $\mathrm{O}\big(\log q\big)$ field operations, since it involves finding the eigenvalues, *i.e.* finding the roots of a polynomial of constant degree over $\mathbb{F}_q$. Clearly, the expected time complexity for finding a is $\mathrm{O}\big(\xi + \log(q) \log\log(q)\big)$ field operations.

From Corollary 3.6, it follows that line 9 uses $\mathrm{O}\big(\log q\big)$ field operations. We conclude that Algorithm 3.1 uses expected $\mathrm{O}\big((\xi + \chi_D(q) + \log(q)\log\log(q))\log\log(q)\big)$ field operations.

Each call to Algorithm 3.1 uses independent random elements, so the double cosets under consideration are uniformly random and independent. Therefore the elements returned by Algorithm 3.1 must be uniformly random. The returned `SLP` is composed of `SLP`s of a and h, both of which have length $\mathrm{O}\big(\log\log(q)\big)$ because of the number of iterations of the loops. □

Corollary 3.9. *Assume Conjecture 3.4 and an oracle for the discrete logarithm problem in $\mathbb{F}_q$. There exists a Las Vegas algorithm that, given $\langle X\rangle \leqslant \mathrm{GL}(4, q)$ such that $G = \langle X\rangle = \mathrm{Sz}(q)$ and $P \in \mathcal{O}$, finds a uniformly random $g \in G_P$, expressed as an* `SLP` *in X. The algorithm has expected time complexity $\mathrm{O}\big((\xi + \chi_D(q) + \log(q)\log\log(q))\log\log(q)\big)$ field operations. The length of the returned* `SLP` *is $\mathrm{O}\big(\log\log(q)\big)$.*

Proof. We compute g as follows.

(1) Find random $x \in G$. Let $Q = Px$ and repeat until $P \neq Q$.
(2) Use Algorithm 3.1 to find $y \in G$ such that $Qy = P$.
(3) Now $g = xy \in G_P$.

We see from Algorithm 3.1 that the choice of y does not depend on x. Hence g is uniformly random, since x is uniformly random. Therefore this is a Las Vegas algorithm. The probability that $P = Q$ is $1/|\mathcal{O}|$, so the dominating term in the complexity is the call to Algorithm 3.1, with expected time complexity given by Theorem 3.8.

The element g will be expressed as an `SLP` in X, since x is random and elements from Algorithm 3.1 are expressed as `SLP`s. Clearly the length of the `SLP` is the same as the length of the `SLP`s from Algorithm 3.1. □

Remark 3.10. In fact, the algorithm of Corollary 3.9 works in any conjugate of $\mathrm{Sz}(q)$, since in Algorithm 3.1 the diagonalisation always moves into the standard copy.

3.1.3. Constructive membership testing. We will now give an algorithm for constructive membership testing in $\mathrm{Sz}(q)$. Given a set of generators X, such that $G = \langle X\rangle = \mathrm{Sz}(q)$, and given $g \in G$, we want to express g as an `SLP` in X. The matrix degree is constant here, so we set $\xi = \xi(4)$. Membership testing is straightforward, using the first steps from the algorithm in Theorem 3.2, and will not be considered here.

3.1.3.1. *Preprocessing.* The algorithm for constructive membership testing has a preprocessing step and a main step. The preprocessing step consists of finding "standard generators" for $\mathrm{O}_2(G_{P_\infty}) = \mathcal{F}$ and $\mathrm{O}_2(G_{P_0})$. In the case of $\mathrm{O}_2(G_{P_\infty})$ the standard generators are defined as matrices $\{S(a_i, x_i)\}_{i=1}^{n} \cup \{S(0, b_i)\}_{i=1}^{n}$ for some unspecified $x_i \in \mathbb{F}_q$, such that $\{a_1, \dots, a_n\}$ and $\{b_1, \dots, b_n\}$ form vector space bases of $\mathbb{F}_q$ over $\mathbb{F}_2$ (so $n = \log_2 q = 2m + 1$).

LEMMA 3.11. *There exist algorithms for the following row reductions.*

(1) *Given* $g = M'(\lambda)S(a, b) \in G_{P_\infty}$*, find* $h \in \mathrm{O}_2(G_{P_\infty})$ *expressed in the standard generators, such that* $gh = M'(\lambda)$.
(2) *Given* $g = S(a, b)M'(\lambda) \in G_{P_\infty}$*, find* $h \in \mathrm{O}_2(G_{P_\infty})$ *expressed in the standard generators, such that* $hg = M'(\lambda)$.
(3) *Given* $P_\infty \neq P \in \mathcal{O}$*, find* $g \in \mathrm{O}_2(G_{P_\infty})$ *expressed in the standard generators, such that* $Pg = P_0$.

Analogous algorithms exist for G_{P_0}*. If the standard generators are expressed as* `SLP`*s of length* $\mathrm{O}(n)$*, the elements returned will have length* $\mathrm{O}(n \log(q))$*. The time complexity of the algorithms is* $\mathrm{O}(\log(q)^3)$ *field operations.*

PROOF. The algorithms are as follows.

(1) (a) Solve a linear system of size $\log(q)$ to construct the linear combination $a = g_{2,1}/g_{2,2} = \sum_{i=1}^{2m+1} \alpha_i a_i$ with $\alpha_i \in \mathbb{F}_2$. Let $h' = \prod_{i=1}^{2m+1} S(a_i, x_i)^{\alpha_i}$ and $g' = gh'$, so that $g' = M'(\lambda)S(0, b')$ for some $b' \in \mathbb{F}_q$.
(b) Solve a linear system of size $\log(q)$ to construct the linear combination $b' = g'_{3,1}/g'_{3,3} = \sum_{i=1}^{2m+1} \beta_i b_i$ with $\beta_i \in \mathbb{F}_2$. Let $h'' = \prod_{i=1}^{2m+1} S(0, b_i)^{\beta_i}$ and $g'' = g'h''$, so that $g'' = M'(\lambda)$.
(c) Now $h = h'h''$.

(2) Analogous to the previous case.

(3) (a) Normalise P so that $P = (ab + a^{t+2} + b^t : b : a : 1)$ for some $a, b \in \mathbb{F}_q$. Solve a linear system of size $\log(q)$ to construct the linear combination $a = \sum_{i=1}^{2m+1} \alpha_i a_i$ with $\alpha_i \in \mathbb{F}_2$. Let $h' = \prod_{i=1}^{2m+1} S(a_i, x_i)^{\alpha_i}$.
(b) Solve a linear system of size $\log(q)$ to construct the linear combination $b + h'_{3,1} = \sum_{i=1}^{2m+1} \beta_i b_i$ with $\beta_i \in \mathbb{F}_2$. Let $h'' = \prod_{i=1}^{2m+1} S(0, b_i)^{\beta_i}$
(c) Now $g = h'h''$ maps P to P_0.

Clearly the dominating term in the time complexity is the solving of the linear systems, which requires $\mathrm{O}(\log(q)^3)$ field operations. The elements returned are constructed using $\mathrm{O}(\log(q))$ multiplications, hence the length of the `SLP` follows. □

THEOREM 3.12. *Assume Conjecture* 3.4 *and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$*. The preprocessing step is a Las Vegas algorithm that finds standard generators for* $\mathrm{O}_2(G_{P_\infty})$ *and* $\mathrm{O}_2(G_{P_0})$*, as* `SLP`*s in* X *of length* $\mathrm{O}(\log\log(q)^2)$*. It has expected time complexity* $\mathrm{O}((\xi + \chi_D(q) + \log(q)\log\log(q))(\log\log(q))^2)$ *field operations.*

PROOF. The preprocessing step proceeds as follows.

(1) Find random elements $a_1 \in G_{P_\infty}$ and $b_1 \in G_{P_0}$ using the algorithm from Corollary 3.9. Repeat until a_1 can be diagonalised to $M'(\lambda) \in G$, where $\lambda \in \mathbb{F}_q^\times$ and λ does not lie in a proper subfield of $\mathbb{F}_q$. Do similarly for b_1.

(2) Find random elements $a_2 \in G_{P_\infty}$ and $b_2 \in G_{P_0}$ using the algorithm from Corollary 3.9. Let $c_1 = [a_1, a_2]$, $c_2 = [b_1, b_2]$. Repeat until $|c_1| = |c_2| = 4$.

(3) Let $Y_\infty = \{c_1, a_1\}$ and $Y_0 = \{c_2, b_1\}$. As standard generators for $\mathrm{O}_2(G_{P_\infty})$ we now take

$$L = \bigcup_{i=1}^{2m+1} \left\{ c_1^{d_1^i}, (c_1^2)^{d_1^i} \right\} \tag{3.19}$$

and similarly we obtain U for $\mathrm{O}_2(G_{P_0})$.

It follows from (2.9) and (2.12) that (3.19) provides the standard generators for G_{P_∞}. These are expressed as SLPs in X, since this is true for the elements returned from the algorithm described in Corollary 3.9. Hence the algorithm is Las Vegas.

By Corollary 3.9, the expected time to find a_1 and b_1 is $\mathrm{O}\big((\xi + \chi_D(q) + \log(q)\log\log(q))\log\log(q)\big)$, and these are uniformly distributed independent random elements. The elements of order dividing $q-1$ can be diagonalised as required. By Theorem 2.1, the proportion of elements of order $q-1$ in G_{P_∞} and G_{P_0} is $\phi(q-1)/(q-1)$. Hence the expected time for the first step is $\mathrm{O}\big((\xi + \chi_D(q) + \log(q)\log\log(q))(\log\log(q))^2\big)$ field operations.

Similarly, by Proposition 2.11 the expected time for the second step is $\mathrm{O}\big((\xi + \chi_D(q) + \log(q)\log\log(q))(\log\log(q))^2\big)$ field operations.

By the remark preceding the theorem, L determines two sets of field elements $\{a_1, \ldots, a_{2m+1}\}$ and $\{b_1, \ldots, b_{2m+1}\}$. In this case each $a_i = a\lambda^i$ and $b_i = b\lambda^{i(t+1)}$, for some fixed $a, b \in \mathbb{F}_q^\times$, where λ is as in the algorithm. Since λ does not lie in a proper subfield, these sets form vector space bases of $\mathbb{F}_q$ over $\mathbb{F}_2$.

To determine if a_1 or b_1 diagonalise to some $M'(\lambda)$ it is sufficient to consider the eigenvalues on the diagonal, since both a_1 and b_1 are triangular. To determine if λ lies in a proper subfield, it is sufficient to determine if $|\lambda| \mid 2^n - 1$, for some proper divisor n of $2m+1$. Hence the dominating term in the complexity is the first step. □

3.1.3.2. *Main algorithm.* Now we consider the algorithm that expresses g as an SLP in X. It is given formally as Algorithm 3.2.

THEOREM 3.13. *Algorithm* 3.2 *is a Las Vegas algorithm with expected time complexity* $\mathrm{O}\big(\xi + \log(q)^3\big)$ *field operations. The length of the* SLP *is* $\mathrm{O}\big(\log(q)(\log\log(q))^2\big)$.

PROOF. First observe that since r is randomly chosen we obtain it as an SLP. On line 4 we check if gr fixes a point, and from Proposition 2.6 we see that the probability that the test succeeds is at least 1/2.

The elements found at lines 5 and 7 are constructed using Lemma 3.11, so we can obtain them as SLPs.

The element h found at line 10 clearly has trace x, and it is constructed using Lemma 3.11, so we obtain it as an SLP. From Lemma 2.8 we know that h is conjugate to $M'(\lambda)$ and therefore must fix 2 points of $\mathcal{O}$. Hence lines 13 and 14 make sense, and the elements found are constructed using Lemma 3.11 and therefore we obtain them as SLPs.

Algorithm 3.2: ElementToSLP(L, U, g)

1 **Input:** Standard generators L for G_{P_∞} and U for G_{P_0}. Matrix $g \in \langle X \rangle = G$.
2 **Output:** An `SLP` for g in X.
repeat
3 $r := \mathbf{Random}(G)$
4 **until** gr has an eigenspace $Q \in \mathcal{O}$
5 Find $z_1 \in G_{P_\infty}$ using L such that $Qz_1 = P_0$.
6 // Now $(gr)^{z_1} \in G_{P_0}$.
7 Find $z_2 \in G_{P_0}$ using U such that $(gr)^{z_1} z_2 = M'(\lambda)$ for some $\lambda \in \mathbb{F}_q^\times$.
8 // Express diagonal matrix as `SLP`
9 $x := \mathrm{Tr}(M'(\lambda))$
10 Find $h = [S(0, (x^t)^{1/4}), S(0,1)^T]$ using $L \cup U$.
11 // Now $\mathrm{Tr}(h) = x$.
12 Let $P_1, P_2 \in \mathcal{O}$ be the fixed points of h.
13 Find $a \in G_{P_\infty}$ using L such that $P_1 a = P_0$.
14 Find $b \in G_{P_0}$ using U such that $(P_2 a)b = P_\infty$.
15 // Now $h^{ab} \in G_{P_\infty} \cap G_{P_0} = \mathcal{H}$, so $h^{ab} \in \{M'(\lambda)^{\pm 1}\}$.
16 **if** $h^{ab} = M'(\lambda)$
then
17 Let w be an `SLP` for $(h^{ab} z_2^{-1})^{z_1^{-1}} r^{-1}$.
18 **return** w
else
19 Let w be an `SLP` for $((h^{ab})^{-1} z_2^{-1})^{z_1^{-1}} r^{-1}$.
20 **return** w
end

The only elements in $\mathcal{H}$ that are conjugate to h are $M'(\lambda)^{\pm 1}$, so clearly h^{ab} must be one of them.

Finally, the elements that make up w were found as `SLP`s, and it is clear that if we evaluate w we obtain g. Hence the algorithm is Las Vegas.

From Lemma 3.11 it follows that lines 5, 7, 10, 13 and 14 use $\mathrm{O}(\log(q)^3)$ field operations.

Finding the fixed points of h, and performing the check at line 4 only amounts to considering eigenspaces, which uses $\mathrm{O}(\log q)$ field operations. Thus the expected time complexity of the algorithm is $\mathrm{O}(\xi + \log q^3)$ field operations.

The `SLP`s of the standard generators have length $\mathrm{O}(\log\log(q)^2)$. Because of the row operations, w will have length $\mathrm{O}(\log(q)(\log\log(q))^2)$.

□

3.1.4. Conjugates of the standard copy. Now we assume that we are given $G \leqslant \mathrm{GL}(4, q)$ such that G is a conjugate of $\mathrm{Sz}(q)$, and we turn to the problem of finding some $g \in \mathrm{GL}(4, q)$ such that $G^g = \mathrm{Sz}(q)$, thus obtaining an isomorphism to the standard copy. The matrix degree is constant here, so we set $\xi = \xi(4)$

Lemma 3.14. *There exists a Las Vegas algorithm that, given $\langle X \rangle \leqslant \mathrm{GL}(4, q)$ such that $\langle X \rangle^h = \mathrm{Sz}(q)$ for some $h \in \mathrm{GL}(4, q)$, finds a point $P \in \mathcal{O}^{h^{-1}} =$*

$\{Qh^{-1} \mid Q \in \mathcal{O}\}$. *The algorithm has expected time complexity*

$$\mathrm{O}\big((\xi + \log(q) \log\log(q)) \log\log(q)\big)$$

field operations.

PROOF. Clearly $\mathcal{O}^{h^{-1}}$ is the set on which $\langle X \rangle$ acts doubly transitively. For a matrix $M'(\lambda) \in \mathrm{Sz}(q)$ we see that the eigenspaces corresponding to the eigenvalues $\lambda^{\pm(t+1)}$ will be in $\mathcal{O}$. Moreover, every element of order dividing $q-1$, in every conjugate G of $\mathrm{Sz}(q)$, will have eigenvalues of the form $\mu^{\pm(t+1)}$, $\mu^{\pm 1}$ for some $\mu \in \mathbb{F}_q^\times$, and the eigenspaces corresponding to $\mu^{\pm(t+1)}$ will lie in the set on which G acts doubly transitively.

Hence to find a point $P \in \mathcal{O}^{h^{-1}}$ it is sufficient to find a random $g \in \langle X \rangle$, of order dividing $q-1$. We compute the pseudo-order using expected $\mathrm{O}\big(\log(q)\log\log(q)\big)$ field operations, and by Proposition 2.6, the expected time to find the element is $\mathrm{O}\big((\xi + \log(q)\log\log(q))\log\log(q)\big)$ field operations. We then find the eigenspaces of g.

Clearly this is a Las Vegas algorithm with the stated time complexity. □

LEMMA 3.15. *There exists a Las Vegas algorithm that, given* $\langle X \rangle \leqslant \mathrm{GL}(4,q)$ *such that* $\langle X \rangle^d = \mathrm{Sz}(q)$ *where* $d = \mathrm{diag}(d_1, d_2, d_3, d_4) \in \mathrm{GL}(4,q)$*, finds a diagonal matrix* $e \in \mathrm{GL}(4,q)$ *such that* $\langle X \rangle^e = \mathrm{Sz}(q)$*, using expected*

$$\mathrm{O}\big((\xi + \log(q)\log\log(q))\log\log(q) + |X|\big)$$

field operations.

PROOF. Let $G = \langle X \rangle$. Since $G^d = \mathrm{Sz}(q)$, G must preserve the symplectic form

$$K = dJd = \begin{bmatrix} 0 & 0 & 0 & d_1 d_4 \\ 0 & 0 & d_2 d_3 & 0 \\ 0 & d_2 d_3 & 0 & 0 \\ d_1 d_4 & 0 & 0 & 0 \end{bmatrix} \tag{3.20}$$

where J is given by (2.24). Using the MeatAxe, we can find this form, which is determined up to a scalar multiple. Hence the diagonal matrix $e = \mathrm{diag}(e_1, e_2, e_3, e_4)$, that we want to find, is also determined up to a scalar multiple (and up to multiplication by a diagonal matrix in $\mathrm{Sz}(q)$).

Since e must take J to K, we must have $K_{1,4} = d_1 d_4 = e_1 e_4$ and $K_{2,4} = d_2 d_3 = e_2 e_3$. Because e is determined up to a scalar multiple, we can choose $e_4 = 1$ and $e_1 = K_{1,4}$. Hence it only remains to determine e_2 and e_3.

To conjugate G into $\mathrm{Sz}(q)$, we must have $Pe \in \mathcal{O}$ for every $P \in \mathcal{O}^{d^{-1}}$, which is the set on which G acts doubly transitively. By Lemma 3.14, we can find $P = (p_1 : p_2 : p_3 : 1) \in \mathcal{O}^{d^{-1}}$, and the condition $Pe = (p_1 K_{1,4} : p_2 e_2 : p_3 e_3 : 1) \in \mathcal{O}$ is given by (2.13) and amounts to

$$p_2 p_3 K_{2,3} + (p_2 e_2)^t + (p_3 e_3)^{t+2} - p_1 K_{1,4} = 0 \tag{3.21}$$

which is a polynomial equation in the two variables e_2 and e_3.

Notice that we can consider e_2^t to be the variable, instead of e_2, since if $x = e_2^t$, then $e_2 = \sqrt{x^t}$. Similarly, we can let e_3^{t+2} be the variable instead of e_3, since if $y = e_3^{t+2}$ then $e_3 = y^{1-t/2}$. Thus instead of (3.21) we obtain a linear equation

$$p_2^t x + p_3^{t+2} y = p_1 K_{1,4} - p_2 p_3 K_{2,3} \tag{3.22}$$

in the variables x, y. Thus the complete algorithm for finding e proceeds as follows.

(1) Find the form K that is preserved by G, using the MeatAxe.
(2) Find $P \in \mathcal{O}^{d^{-1}}$ using Lemma 3.14.
(3) Let $P = (p_1 : p_2 : p_3 : p_4)$ and find $Q = (q_1 : q_2 : q_3 : q_4)$ using Lemma 3.14 until the following linear system in the variables x and y is non-singular.

$$\begin{aligned} p_2^t x + p_3^{t+2} y &= p_1 K_{1,4} - p_2 p_3 K_{2,3} \\ q_2^t x + q_3^{t+2} y &= q_1 K_{1,4} - q_2 q_3 K_{2,3} \end{aligned} \tag{3.23}$$

By Proposition 2.10, the probability of finding such a Q is $1 - \mathrm{O}\big(1/\sqrt{q}\big)$.
(4) Let (α, β) be a solution to the linear system. The diagonal matrix $e = \operatorname{diag}(K_{1,4}, \sqrt{\alpha^t}, \beta^{1-t/2}, 1)$ now satisfies $G^e = \mathrm{Sz}(q)$.

By Lemma 3.14 and Section 1.2.10.1, this is a Las Vegas algorithm that uses expected $\mathrm{O}\big((\xi + \log(q)\log\log(q))\log\log(q) + |X|\big)$ field operations. □

LEMMA 3.16. *There exists a Las Vegas algorithm that, given subsets X, Y_P and Y_Q of $\mathrm{GL}(4, q)$ such that $\mathrm{O}_2(G_P) < \langle Y_P \rangle \leqslant G_P$ and $\mathrm{O}_2(G_Q) < \langle Y_Q \rangle \leqslant G_Q$, respectively, where $\langle X \rangle = G$, $G^h = \mathrm{Sz}(q)$ for some $h \in \mathrm{GL}(4, q)$ and $P \neq Q \in \mathcal{O}^{h^{-1}}$, finds $k \in \mathrm{GL}(4, q)$ such that $(G^k)^d = \mathrm{Sz}(q)$ for some diagonal matrix $d \in \mathrm{GL}(4, q)$. The algorithm has expected time complexity $\mathrm{O}\big(|X|\big)$ field operations.*

PROOF. Notice that the natural module $V = \mathbb{F}_q^4$ of $\mathcal{FH}$ is uniserial with four non-zero submodules, namely $V_i = \big\{(v_1, v_2, v_3, v_4) \in \mathbb{F}_q^4 \mid v_j = 0, j > i\big\}$ for $i = 1, \dots, 4$. Hence the same is true for $\langle Y_P \rangle$ and $\langle Y_Q \rangle$ (but the submodules will be different) since they lie in conjugates of $\mathcal{FH}$.

Now the algorithm proceeds as follows.

(1) Let $V = \mathbb{F}_q^4$ be the natural module for $\langle Y_P \rangle$ and $\langle Y_Q \rangle$. Find composition series $V = V_4^P > V_3^P > V_2^P > V_1^P$ and $V = V_4^Q > V_3^Q > V_2^Q > V_1^Q$ using the MeatAxe.
(2) Let $U_1 = V_1^P$, $U_2 = V_2^P \cap V_3^Q$, $U_3 = V_3^P \cap V_2^Q$ and $U_4 = V_1^Q$. For each $i = 1, \dots, 4$, choose $u_i \in U_i$.
(3) Now let k be the matrix such that k^{-1} has u_i as row i, for $i = 1, \dots, 4$.

We now motivate the second step of the algorithm.

One can choose a basis that exhibits the series $\big\{V_i^P\big\}$, in other words, such that the matrices acting on the module are lower triangular with respect to this basis. Similarly one can choose a basis that exhibits the series $\Big\{V_i^Q\Big\}$.

On the other hand, since $P \neq Q$, there exists $g' \in \mathrm{Sz}(q)$ such that $Phg' = P_\infty$ and $Qhg' = P_0$. If we let $z = hg'$, then $\langle Y_P \rangle^z$ and $\langle Y_Q \rangle^z$ consist of lower and upper triangular matrices, respectively. Thus, the rows of z^{-1} form a basis of V that exhibits the series $\left\{V_i^P\right\}$ and the series $\left\{V_i^Q\right\}$ in reversed order.

With respect to this basis, it is clear that $\dim V_2^P \cap V_3^Q = 1$, $\dim V_3^P \cap V_2^Q = 1$ and that all the U_i are distinct. Hence the basis chosen in the algorithm exhibits the series $\left\{V_i^P\right\}$, and it exhibits the series $\left\{V_i^Q\right\}$ in reverse order. Therefore the chosen k satisfies that $\langle Y_P \rangle^k$ is lower triangular and $\langle Y_Q \rangle^k$ is upper triangular. The former implies that kz^{-1} is a lower triangular matrix, and the latter that it is an upper triangular matrix, and hence it must diagonal.

Thus the matrix k found in the algorithm satisfies $z = kd$ for some diagonal matrix $d \in \mathrm{GL}(4, q)$. Since $\mathrm{Sz}(q) = G^h = G^z = (G^k)^d$, the algorithm returns a correct result, and it is Las Vegas because the MeatAxe is Las Vegas. Clearly the expected time complexity is the same as the MeatAxe, so the algorithm uses $\mathrm{O}\big(|X|\big)$ field operations. □

Theorem 3.17. *Assume Conjecture* 3.4. *There exists a Las Vegas algorithm that, given a conjugate* $\langle X \rangle$ *of* $\mathrm{Sz}(q)$*, finds* $g \in \mathrm{GL}(4, q)$ *such that* $\langle X \rangle^g = \mathrm{Sz}(q)$*. The algorithm has expected time complexity* $\mathrm{O}\big((\xi + \log(q) \log\log(q))(\log\log(q))^2 + |X|\big)$ *field operations.*

Proof. Let $G = \langle X \rangle$. By Remark 3.10, we can use Corollary 3.9 in G, and hence we can find generators for a stabiliser of a point in G, using the algorithm described in Theorem 3.12. In this case we do not need the elements as SLPs, so a discrete log oracle is not necessary.

(1) Find points $P, Q \in \mathcal{O}^{h^{-1}}$ using Lemma 3.14. Repeat until $P \neq Q$.
(2) Find generating sets Y_P and Y_Q such that $\mathrm{O}_2(G_P) < \langle Y_P \rangle \leqslant G_P$ and $\mathrm{O}_2(G_Q) < \langle Y_Q \rangle \leqslant G_Q$ using the first three steps of the algorithm from the proof of Theorem 3.12.
(3) Find $k \in \mathrm{GL}(4, q)$ such that $(G^k)^d = \mathrm{Sz}(q)$ for some diagonal matrix $d \in \mathrm{GL}(4, q)$, using Lemma 3.16.
(4) Find a diagonal matrix e using Lemma 3.15.
(5) Now $g = ke$ satisfies that $G^g = \mathrm{Sz}(q)$.

Be Lemma 3.14, 3.16 and 3.15, and the proof of Theorem 3.12, this is a Las Vegas algorithm with expected time complexity as stated. □

3.1.5. Tensor decomposition. Now assume that $G \leqslant \mathrm{GL}(d, q)$ where $G \cong \mathrm{Sz}(q)$, $d > 4$ and $q = 2^{2m+1}$ for some $m > 0$. Then $\mathrm{Aut}\, \mathbb{F}_q = \langle \psi \rangle$, where ψ is the Frobenius automorphism. Let W be the given module of G of dimension d and let V be the natural module of $\mathrm{Sz}(q)$ of dimension 4. From Section 1.2.7 and Section 2.1.3 we know that

$$W \cong V^{\psi^{i_0}} \otimes V^{\psi^{i_1}} \otimes \cdots \otimes V^{\psi^{i_{n-1}}} \tag{3.24}$$

for some integers $0 \leqslant i_0 < i_1 < \cdots < i_{n-1} \leqslant 2m$. In fact, we may assume that $i_0 = 0$ and clearly $d = \dim W = (\dim V)^n = 4^n$. As described in Section 1.2.7, we now want to tensor decompose W to obtain an effective isomorphism from W to V.

3.1.5.1. *The main algorithm.* We now describe our main algorithm that finds a tensor decomposition of W when q is large. It is sufficient to find a flat in W. For $k = 0, \dots, n-1$, let $H_k \leqslant \mathrm{GL}(4, q)$ be the image of the representation corresponding to $V^{\psi^{i_k}}$, and let $\rho_k : G \to H_k$ be an isomorphism. Our goal is then to find ρ_k effectively for some k.

We begin with an overview of the method. Our approach for finding a flat in W is to consider eigenspaces of an element of G of order dividing $q-1$. By Proposition 2.6 we know that such elements are easy to find by random search.

Let $g \in G$ where $|g| = q - 1$, and let $t = 2^{m+1}$. By Proposition 2.4 we know that for $k = 0, \dots, n-1$, $\rho_k(g)$ has four distinct eigenvalues $\lambda_k^{\pm 1}$ and $\lambda_k^{\pm(t+1)}$ for some $\lambda_k \in \mathbb{F}_q^\times$. Also, the eigenspaces of $\rho_k(g)$ have dimension 1. Our method for finding a flat in W is to construct a line as a suitable sum of eigenspaces of g.

Let E be the multiset of eigenvalues of g, so that $|E| = d$ and every element of E has the form

$$\lambda_0^{j_0} \lambda_1^{j_1} \cdots \lambda_{n-1}^{j_{n-1}} \tag{3.25}$$

where each $\lambda_k \in \mathbb{F}_q^\times$ and each $j_k \in \{\pm 1, \pm(t+1)\}$. A set $E' \subseteq \mathbb{F}_q^\times$ that satisfies

- $|E'| = n$
- $\lambda_k \in E'$ or $\lambda_k^{-1} \in E'$ for each $k = 0, \dots, n-1$

is a set of *base values* for E. Clearly E is easily calculated from E'.

Moreover $\lambda_k = \lambda_0^{2^{i_k}}$, and since $|g| = q - 1$ we must have $|\lambda_k| = q - 1$ for $k = 0, \dots, n-1$. For every $0 \leqslant k < l \leqslant n-1$ we have $0 < 2^{i_k} \pm 2^{i_l} < q - 1$ and therefore $\lambda_k \neq \lambda_l^{\mp 1}$.

First we try to find a set of base values for E.

Conjecture 3.18. *Assume $m > n$. Let $S = \left\{ \sqrt{e/f} : e, f \in E \mid e \neq f \right\}$ and*

$$P = \{x \in S \mid \forall\, e \in E\ \exists\, y \in \left\{x, x^{-1}, x^{t+1}, x^{-t-1}\right\} : \\ \left\{eyx, eyx^{-1}, eyx^{t+1}, eyx^{-t-1}\right\} \subseteq E\}. \tag{3.26}$$

Then the following hold:

- *P contains a set of base values for E.*
- *If the twists do not have a subsequence $i_r < i_r + 1 < i_r + 2$, then $|P| = 2n$ and hence P consists of the base values and their inverses.*

If the twists have a subsequence of the form in the Conjecture, or more generally a subsequence of length l, then $\prod_{i=0}^{j} \lambda_{r+i} \in P$ for every $j = 0, \dots, l-1$. Hence we need more conditions to extract the base values.

Conjecture 3.19. *Let P be as in Lemma 3.18. Define $P' \subseteq P$ to be those $x \in P$ for which there exists $0 = \alpha_0 < \alpha_1 < \cdots < \alpha_n \leqslant 2m$ such that $x^{\alpha_i} \in P$ for every $i = 0, \dots, n$. Then $|P'| = 2n$ and hence P' consists of the base values and their inverses.*

Let S_i denote the sum of eigenspaces of g corresponding to the eigenvalues $\lambda_0^{j_0} \cdots \lambda_i \cdots \lambda_{n-1}^{j_{n-1}}$ and $\lambda_0^{j_0} \cdots \lambda_i^{-1} \cdots \lambda_{n-1}^{j_{n-1}}$, where each j_k ranges over $\{\pm 1, \pm(t+1)\}$.

LEMMA 3.20. *If* $\dim S_k = 2 \cdot 4^{n-1}$ *for some* $0 \leqslant k \leqslant n-1$*, then* S_k *is a line in* W*.*

PROOF. For each $i = 0, \dots, n-1$, let e_{j_i} be an eigenvector of $\rho_i(g)$ for the eigenvalue $\lambda_i^{j_i}$, where $j_i \in \{\pm 1, \pm(t+1)\}$. Observe that W contains the following subspace.

$$L = \left\langle e_{j_0} \otimes \cdots \otimes e_{j_{n-1}} \mid j_i \in \{\pm 1, \pm(t+1)\}, i \neq k, j_k \in \{\pm 1\} \right\rangle$$

Clearly, L is of the form $V^{\psi^{i_0}} \otimes \cdots \otimes V^{\psi^{i_{k-1}}} \otimes A \otimes V^{\psi^{i_{k+1}}} \otimes \cdots \otimes V^{\psi^{i_{n-1}}}$ where $\dim A = 2$, so L is a line in W of dimension $2 \cdot 4^{n-1}$.

If $v = e_{j_0} \otimes \cdots \otimes e_{j_{n-1}} \in L$, then

$$\begin{aligned} vg &= e_{j_0}\rho_0(g) \otimes \cdots \otimes e_{j_{n-1}}\rho_{n-1}(g) = \lambda_0^{j_0} e_{j_0} \otimes \cdots \otimes \lambda_{n-1}^{j_{n-1}} e_{j_{n-1}} = \\ &= \lambda_0^{j_0} \cdots \lambda_{n-1}^{j_{n-1}} v \end{aligned}$$

and hence $v \in S_k$. Therefore $L \leqslant S_k$, so if $\dim S_k = 2 \cdot 4^{n-1}$ then $L = S_k$ and thus S_k is a line in W.

□

The success probability of the algorithm for finding a flat relies on the following Conjecture.

CONJECTURE 3.21. *Let* $d = 4^n$ *with* $n > 1$ *be fixed. If* $q = 2^{2m+1}$ *and* $m \geqslant n+1$*, then for every absolutely irreducible* $G \leqslant \mathrm{GL}(d, q)$ *with* $G \cong \mathrm{Sz}(q)$ *and every* $g \in G$ *with* $|g| = q-1$*, we have* $\dim S_i = 2 \cdot 4^{n-1}$ *for some* $0 \leqslant i \leqslant n-1$*.*

The algorithm for finding a flat is shown as Algorithm 3.3.

THEOREM 3.22. *Assume Conjectures* 3.18*,* 3.19 *and* 3.21*. Algorithm* 3.3 *is a Las Vegas algorithm. The algorithm has expected time complexity*

$$\mathrm{O}\big((\xi(d) + d^3 \log(q) \log\log(q^d)) \log\log(q)\big)$$

field operations.

PROOF. The expected number of iterations in the initial loop is $\mathrm{O}(1)$. Hence the expected time for the loop is $\mathrm{O}\big(\xi(d) + d^3 \log(q) \log\log(q^d)\big)$ field operations. If $|g| = q-1$ then line 7 will succeed, and Conjecture 3.21 asserts that line 11 will succeed for some i. If $|g|$ is a proper divisor of $q-1$ then these lines might still succeed, and the probability that $|g| = q-1$ is high. By Proposition 2.6, the expected number of iterations of the outer repeat statement is $\mathrm{O}\big(\log\log(q)\big)$.

If line 12 is reached, then the algorithm returns a correct result and hence it is Las Vegas.

To find the eigenvalues of g, we calculate the characteristic polynomial of g using $\mathrm{O}(d^3)$ field operations, and find its roots using Theorem 1.1. By Conjectures

Algorithm 3.3: TensorDecomposeSz(X)

Input: Generating set X for $G \cong \mathrm{Sz}(q)$ with natural module W, where $\dim W = 4^n$, $n > 1$, $q = 2^{2m+1}$ with $m > n$ and W is absolutely irreducible and over $\mathbb{F}_q$.
Output: A line S in W.
// FindBaseValues is given by Conjectures 3.18 and 3.19.
repeat
    // Find random element g of pseudo-order $q-1$
    repeat
        $g :=$ Random(G)
    until $|g| \mid q-1$
    $E :=$ Eigenvalues(g)
    $N :=$ FindBaseValues(E)
until $|N| = n$
for $i := 0$ **to** $n-1$
  do
    // Let $N = \{\lambda_0, \dots, \lambda_{n-1}\}$
    $E_i := \left\{\lambda_0^{j_0} \cdots \lambda_i \cdots \lambda_{n-1}^{j_{n-1}} \mid j_k \in \{\pm 1, \pm(t+1)\}\right\}$
    $\cup \left\{\lambda_0^{j_0} \cdots \lambda_i^{-1} \cdots \lambda_{n-1}^{j_{n-1}} \mid j_k \in \{\pm 1, \pm(t+1)\}\right\}$
    $S_i := \sum_{e \in E_i}$ Eigenspace(g, e)
    if $\dim S_i = 2 \cdot 4^{n-1}$
      then
        return S_i
    end
end

3.18 and 3.19, the rest of the algorithm uses $\mathrm{O}\big(d^2 n \log(q)\big)$ field operations. Thus the theorem follows. □

3.1.5.2. *Small field approach.* When q is small, the feasibility of Algorithm 3.3 is not guaranteed. In that case the approach is to find standard generators of G using permutation group techniques, then enumerate all tensor products of the form (3.24) and for each one we determine if it is isomorphic to W.

Since q is polynomial in d, this will turn out to be an efficient algorithm which is given as Algorithm 3.4. It finds a permutation representation of $G \cong \mathrm{Sz}(q)$, which is done using the following result.

Lemma 3.23. *There exists a Las Vegas algorithm that, given $\langle X \rangle \leqslant \mathrm{GL}(d, q)$ such that $q = 2^{2m+1}$ with $m > 0$ and $\langle X \rangle \cong \mathrm{Sz}(q)$, finds an effective injective homomorphism $\Pi : \langle X \rangle \to \mathrm{Sym}(O)$ where $|O| = q^2 + 1$. The algorithm has expected time complexity $\mathrm{O}\big(q^2(\xi(d) + |X| d^2 + d^3) + d^4\big)$ field operations.*

Proof. By Theorem 2.1, $\mathrm{Sz}(q)$ acts doubly transitively on a set of size q^2+1. Hence $G = \langle X \rangle$ also acts doubly transitively on O, where $|O| = q^2 + 1$, and we can find the permutation representation of G if we can find a point $P \in O$. The set O is a set of projective points of $\mathbb{F}_q^d$, and the algorithm proceeds as follows.

(1) Choose random $g \in G$. Repeat until $|g| \mid q-1$.

(2) Choose random $x \in G$ and let $h = g^x$. Repeat until $[g,h]^4 = 1$ and $[g,h] \neq 1$.
(3) Find a composition series for the module M of $\langle g,h \rangle$ and let $P \subseteq M$ be the submodule of dimension 1 in the series.
(4) Find the orbit $O = P^G$ and compute the permutation group $S \leqslant \mathrm{Sym}(O)$ of G on O, together with an effective isomorphism $\Pi : G \to S$.

By Proposition 2.4, elements in G of order dividing $q-1$ fix two points of O, and hence $\langle g,h \rangle \leqslant G_P$ for some $P \in O$ if and only if g and h have a common fixed point. All composition factors of M have dimension 1, so a composition series of M must contain a submodule P of dimension 1. This submodule is a fixed point for $\langle g,h \rangle$ and its orbit must have size q^2+1, since $|G| = q^2(q^2+1)(q-1)$ and $|G_P| = q^2(q-1)$. It follows that $P \in O$.

All elements of G of even order lie in the derived group of a stabiliser of some point, which is also a Sylow 2-subgroup of G, and the exponent of this subgroup is 4. Hence $[g,h]^4 = 1$ if and only if $\langle g,h \rangle$ lie in a stabiliser of some point, if and only if g and h have a common fixed point.

To find the orbit $O = P^G$ we can compute a Schreier tree on the generators X, with P as root, using $\mathrm{O}\big(|X|\,|O|\,d^2\big)$ field operations. Then $\Pi(g)$ can be computed for any $g \in \langle X \rangle$ using $\mathrm{O}\big(|O|\,d^2\big)$ field operations, by computing the permutation on O induced by g. Hence Π is effective, and its image S is found by computing the image of each element of X. Therefore the algorithm is correct and it is clearly Las Vegas.

We find g using expected $\mathrm{O}\big((\xi(d) + d^3 \log(q) \log\log(q^d)) \log\log(q)\big)$ field operations and we find h using expected $\mathrm{O}\big((\xi(d) + d^3)q^2\big)$ field operations. Then P is found using the MeatAxe, in expected $\mathrm{O}\big(d^4\big)$ field operations. Thus the result follows. □

PROPOSITION 3.24. *Let $G = \langle X \rangle \leqslant \mathrm{Sym}(\mathcal{O})$ such that $G \cong \mathrm{Sz}(q) = H$. There exists a Las Vegas algorithm that finds $x, h, z \in G$ as* SLP*s in X such that the map*

$$\begin{aligned} x &\mapsto S(1,0) \\ h &\mapsto M'(\lambda) \\ z &\mapsto T \end{aligned} \tag{3.27}$$

is an isomorphism. Its time complexity is $\mathrm{O}\big(q^3 \log(q)^5\big)$. The length of the returned SLP*s are $\mathrm{O}\big(q\big)$.*

PROOF. Follows from [**BB07**, Theorem 1]. □

THEOREM 3.25. *Assume Conjecture* 3.24. *Algorithm* 3.4 *is a Las Vegas algorithm with expected time complexity*

$$\mathrm{O}\big(q^2(\xi(d) + |X|\,d^2 + d^3 \log\log(q) + q^2 \log(q)^3) + d^3(|X| \binom{2m}{n-1} + d)\big)$$

field operations.

Algorithm 3.4: SmallFieldTensorDecompose(X)

Input: Generating set X for $G \cong \mathrm{Sz}(q)$ with natural module W, where $\dim W = 4^n$, $n > 1$, $q = 2^{2m+1}$, $m > 0$ and W is absolutely irreducible and over $\mathbb{F}_q$

Output: A change of basis matrix c which exhibits W as (3.24).

// Find permutation representation, *i.e.* permutation group and
// corresponding isomorphism

$(\pi, P_G) :=$ SuzukiPermRep(G)

$x, h, z :=$ StandardGens(P_G)

Evaluate the SLPs of x, h, z on G to obtain the set Y.

$H := \langle Y \rangle$

$T := \{(i_1, \dots, i_n) \mid 0 \leqslant i_1 < \cdots < i_n \leqslant 2m\}$

// Let V be the natural module of H

for $(i_1, \dots, i_n) \in T$
    do
        $U := V^{\psi^{i_1}} \otimes \cdots \otimes V^{\psi^{i_n}}$
        // Find isomorphism between modules
        $(\mathit{flag}, c) :=$ ModuleIsomorphism(U, W)
        if $\mathit{flag} =$ TRUE
          then
              return c
        end
end

Proof. The permutation representation π can be found using Lemma 3.23, and the elements x, h, z are found using Conjecture 3.24. Testing if modules are isomorphic can be done using the MeatAxe.

If the algorithm returns an element c then the change of basis determined by c exhibits W as a tensor product, so the algorithm is Las Vegas.

The lengths of the SLPs of x, h, z is $\mathrm{O}\big(q^2 \log\log(q)\big)$, so we need $\mathrm{O}\big(d^3 q^2 \log\log(q)\big)$ field operations to obtain Y. The set T has size $\binom{2m}{n-1}$. Module isomorphism testing uses $\mathrm{O}\big(|X|\, d^3\big)$ field operations. Hence by Conjecture 3.24 and Theorem 3.23 the time complexity of the algorithm is as stated. $\square$

3.1.6. Constructive recognition. Finally, we can now state and prove our main theorem.

Theorem 3.26. *Assume the Suzuki Conjectures and an oracle for the discrete logarithm problem in $\mathbb{F}_q$. There exists a Las Vegas algorithm that, given $\langle X \rangle \leqslant \mathrm{GL}(d, q)$ satisfying the assumptions in Section 1.2.7, with $q = 2^{2m+1}$, $m > 0$ and $\langle X \rangle \cong \mathrm{Sz}(q)$, finds an effective isomorphism $\varphi : \langle X \rangle \to \mathrm{Sz}(q)$ and performs preprocessing for constructive membership testing. The algorithm has expected time complexity $\mathrm{O}\big(\xi(d)(d^2 + (\log\log(q))^2) + d^5 \log\log(d) + d^4 |X| + d^3 \log(q) \log\log(q)(\log(d) + \log\log(q)) + \log(q)(\log\log(q))^3 + \chi_D(q)(\log\log(q))^2\big)$ field operations.*

The inverse of φ is also effective. Each image of φ can be computed using $\mathrm{O}\big(d^3\big)$ field operations, and each pre-image using expected

$$\mathrm{O}\big(\xi(d) + \log(q)^3 + d^3 \log(q)(\log\log(q))^2\big)$$

field operations.

PROOF. Let V be the module of $G = \langle X \rangle$. From Section 2.1.3 we know that $d = 4^n$ where $n \geqslant 1$. The algorithm proceeds as follows:

(1) If $d = 4$ then use Theorem 3.17 to obtain $y \in \mathrm{GL}(4, q)$ such that $G^y = \mathrm{Sz}(q)$, and hence an effective isomorphism $\varphi : G \to \mathrm{Sz}(q)$ defined by $g \mapsto g^y$.
(2) If $m > n$, use Algorithm 3.3 to find a flat $L \leqslant V$. Then use the tensor decomposition algorithm described in Section 1.2.10.3 with L, to obtain $x \in \mathrm{GL}(d, q)$ such the change of basis determined by x exhibits V as a tensor product $U \otimes W$, with $\dim U = 4$. Let G_U and G_W be the images of the corresponding representations.
(3) If instead $m \leqslant n$ then use Algorithm 3.4 to find x.
(4) Define $\rho_U : G_{U \otimes W} \to G_U$ as $g_u \otimes g_w \mapsto g_u$ and let $Y = \{\rho_U(g^x) \mid g \in X\}$. Then $\langle Y \rangle$ is conjugate to $\mathrm{Sz}(q)$.
(5) Use Theorem 3.17 to get $y \in \mathrm{GL}(4, q)$ such that $\langle Y \rangle^y = \mathrm{Sz}(q)$.
(6) An effective isomorphism $\varphi : G \to \mathrm{Sz}(q)$ is given by $g \mapsto \rho_U(g^x)^y$.

The map ρ_U is straightforward to compute, since given $g \in \mathrm{GL}(d, q)$ it only involves dividing g into submatrices of degree 4^{n-1}, checking that they are scalar multiples of each other and returning the 4×4 matrix consisting of these scalars. Since x might not lie in G, but only in $\mathrm{N}_{\mathrm{GL}(d,q)}(G) \cong G{:}\mathbb{F}_q$, the result of ρ_U might not have determinant 1. However, since every element of $\mathbb{F}_q$ has a unique 4th root, we can easily scale the matrix to have determinant 1. Hence by Theorems 3.22, 3.25 and 3.17, the algorithm is Las Vegas and any image of φ can be computed using $\mathrm{O}(d^3)$ field operations.

In the case where we use Algorithm 3.4 we have $m \leqslant n$, hence $\binom{2m}{n-1} < d$ and $q \leqslant d$. The expected time complexity to find x in this case is $\mathrm{O}\big(\xi(d)d^2 + d^4(|X| + d \log\log(d))\big)$ field operations.

By Theorem 3.22, finding L uses $\mathrm{O}\big((\xi(d) + d^3 \log(q) \log\log(q^d)) \log\log(q)\big)$ field operations. From Section 1.2.10.3, finding x uses $\mathrm{O}\big(d^3 \log(q)\big)$ field operations when a flat L is given. By Theorem 3.17, finding y uses expected $\mathrm{O}\big((\xi + \chi_D(q) + \log(q)\log\log(q))(\log\log(q))^2 + |Y|\big)$ field operations, given a random element oracle for $\langle Y \rangle$ that finds a random element using $\mathrm{O}\big(\xi\big)$ field operations. In this case we can construct random elements for $\langle Y \rangle$ using the random element oracle for $\langle X \rangle$, and then we will find them in $\mathrm{O}\big(\xi(d)\big)$ field operations.

Hence the expected time complexity is as stated. Finally, $\varphi^{-1}(g)$ is computed by first using Algorithm 3.2 to obtain an `SLP` of g and then evaluating it on X. The necessary precomputations in Theorem 3.12 have already been made during the application of Theorem 3.17, and hence it follows from Theorem 3.13 that the time complexity for computing the pre-image of g is as stated. □

3.2. Small Ree groups

Here we will use the notation from Section 2.2. We will refer to Conjectures 3.44, 3.50, 3.51, 3.52, 3.57 and 3.60 simultaneously as the *small Ree Conjectures*. We now give an overview of the algorithm for constructive recognition and constructive membership testing. It will be formally proved as Theorem 3.62.

(1) Given a group $G \cong \mathrm{Ree}(q)$, satisfying the assumptions in Section 1.2.7, we know from Section 2.2.3 that the module of G is isomorphic to a tensor product of twisted copies of either the natural module of G or its 27-dimensional module. Hence we first tensor decompose this module. This is described in Section 3.2.5.
(2) The resulting group has degree 7 or 27. In the latter case we need to decompose it into degree 7. This is described in Section 3.2.6.
(3) Now we have a group of degree 7, so it is a conjugate of the standard copy. We therefore find a conjugating element. This is described in Section 3.2.4.
(4) Finally we are in $\mathrm{Ree}(q)$. Now we can perform preprocessing for constructive membership testing and other problems we want to solve. This is described in Section 3.2.3.

Given a discrete logarithm oracle, the whole process has time complexity slightly worse than $\mathrm{O}\big(d^6 + \log(q)^3\big)$ field operations, assuming that G is given by a bounded number of generators.

3.2.1. Recognition. We now consider the question of non-constructive recognition of $\mathrm{Ree}(q)$, so we want to find an algorithm that, given $\langle X\rangle \leqslant \mathrm{GL}(d, q)$, decides whether or not $\langle X\rangle \cong \mathrm{Ree}(q)$. We will only consider this problem for the standard copy, *i.e.* we will only answer the question whether or not $\langle X\rangle = \mathrm{Ree}(q)$.

THEOREM 3.27. *There exists a Las Vegas algorithm that, given* $\langle X\rangle \leqslant \mathrm{GL}(7, q)$*, decides whether or not* $\langle X\rangle = \mathrm{Ree}(q)$*. The algorithm has expected time complexity* $\mathrm{O}\big(\sigma_0(\log(q))(|X| + \log(q))\big)$ *field operations.*

PROOF. Let $G = \mathrm{Ree}(q)$, with natural module M. The algorithm proceeds as follows:

(1) Determine if $X \subseteq G$ and return `false` if not. All the following steps must succeed in order to conclude that a given $g \in X$ also lies in G.
   (a) Determine if $g \in \mathrm{SO}(7, q)$, which is true if $\det g = 1$ and if $gJg^T = J$, where J is given by (2.40) and where g^T denotes the transpose of g.
   (b) Determine if $g \in G_2(q)$, which is true if g preserves the algebra multiplication $\cdot$ of M. The multiplication table can easily be precomputed using the fact that if $v, w \in M$ then $v \cdot w = f(v \otimes w)$, where f is a generator of $\mathrm{Hom}_G(M \otimes M, M)$ (which has dimension 1).
   (c) Determine if g is a fixed point of the exceptional outer automorphism of $G_2(q)$, mentioned in Section 2.2.2. Computing the automorphism

amounts to extracting a submatrix of the exterior square of g and then replacing each matrix entry x by x^{3^m}.

(2) If $\langle X \rangle$ is not a proper subgroup of G, or equivalently if $\langle X \rangle$ is not contained in a maximal subgroup, return `true`. Otherwise return `false`. By Proposition 2.20, it is sufficient to determine if $\langle X \rangle$ cannot be written over a smaller field and if $\langle X \rangle$ is irreducible. This can be done using the algorithms described in Sections 1.2.10.1 and 1.2.10.2.

Since the matrix degree is constant, the complexity of the first step of the algorithm is $\mathrm{O}(1)$ field operations. For the same reason, the expected time of the algorithms in Sections 1.2.10.1 and 1.2.10.2 is $\mathrm{O}\big(\sigma_0(\log(q))(|X| + \log(q))\big)$ field operations. Hence our recognition algorithm has expected time as stated, and it is Las Vegas since the MeatAxe is Las Vegas. □

3.2.2. Finding an element of a stabiliser. Let $G = \mathrm{Ree}(q) = \langle X \rangle$. In this section the matrix degree is constant, so we set $\xi = \xi(7)$. The algorithm for the constructive membership problem needs to find independent random elements of G_P for a given point P. This is straightforward if, for any pair of points $P, Q \in \mathcal{O}$, we can find $g \in G$ as an `SLP` in X such that $Pg = Q$.

The general idea is to find an involution $j \in G$ by random search, and then compute $\mathrm{C}_G(j) \cong \langle j \rangle \times \mathrm{PSL}(2, q)$ using the *Bray algorithm* described in Section 1.2.9.2. The given module restricted to the centraliser splits up as in Proposition 2.18, and the points $P, Q \in \mathcal{O}$ restrict to points in the 3-dimensional submodule. If the restrictions satisfy certain conditions, we can then find an element $g \in \mathrm{C}_G(j)$ that maps these restricted points to each other, and we obtain g as an `SLP` in the generators of $\mathrm{C}_G(j)$ using Theorem 1.12. It turns out that with high probability, we can then multiply g by an element that fixes the restriction of P so that g also maps P to Q. A discrete logarithm oracle is needed in that step. Since the Bray algorithm produces generators for the centraliser as `SLP`s in X, we obtain g as an `SLP` in X.

If any of the steps fail, we can try again with another involution j, so using this method we can map P to Q for any pair of points $P, Q \in \mathcal{O}$.

It should be noted that it is easy to find involutions using the method described in Section 1.2.5, since by Corollary 2.24 it is easy to find elements of even order by random search.

3.2.2.1. *The involution centraliser.* To use the Bray algorithm we need to provide an algorithm that determines if the whole centraliser has been generated. Since we know what the structure of the centraliser should be, this poses no problem. If we have the whole centraliser, the derived group should be $\mathrm{PSL}(2, q)$, and by Proposition 2.27, with high probability it is sufficient to compute two random elements of the derived group. Random elements of the derived group can be found as described in Section 1.2.6.

We can therefore find the involution centraliser $\mathrm{C}_G(j) \leqslant G$ and $\mathrm{C}_G(j)' \cong \mathrm{PSL}(2, q)$.

LEMMA 3.28. *There exists a Las Vegas algorithm that, given $\langle Y \rangle \leqslant G$ such that $\langle Y \rangle = \mathrm{C}_G(j)$ for some involution $j \in G$, finds*

- *the submodule $S_j \leqslant V_j$ described in Proposition 2.18,*
- *an effective $\langle Y \rangle$-module homomorphism $\varphi_V : V_j \to S_j$,*
- *the induced map $\varphi_{\mathcal{O}} : \mathbb{P}(V_j) \to \mathbb{P}(S_j)$,*
- *the corresponding map φ_G from the 7-dimensional representation of $\mathrm{C}_G(j)$ to the 3-dimensional representation.*

The maps can be computed using $\mathrm{O}(1)$ field operations. The algorithm has expected time complexity $\mathrm{O}(|Y|)$ field operations.

PROOF. This is a straightforward application of the MeatAxe, so the fact that the algorithm is Las Vegas and has the stated time complexity follows from Section 1.2.10.1. The maps consist of a change of basis followed by a projection to a subspace, and so the Lemma follows. □

LEMMA 3.29. *Use the notation of Lemma 3.28. There exists a Las Vegas algorithm that, given $H = \langle Y \rangle = \varphi_G(\mathrm{C}_G(j)')$ for an involution $j \in G$, finds effective isomorphisms $\rho_G : \langle Y \rangle \to \mathrm{PSL}(2, q)$, $\pi_3 : \mathrm{PSL}(2, q) \to \langle Y \rangle$ and $\pi_7 : \langle Y \rangle \to \mathrm{C}_G(j)'$.*

The map π_3 is the symmetric square map of $\mathrm{PSL}(2, q)$; both $\varphi_G \circ \pi_7$ and $\pi_3 \circ \rho_G$ are the identity on $\langle Y \rangle$. The maps ρ_G and π_3 can be computed using $\mathrm{O}(1)$ field operations and π_7 can be computed using $\mathrm{O}(\log(q)^3)$ field operations. The algorithm has expected time complexity $\mathrm{O}((\xi + \log(q) \log\log(q)) \log\log(q) + |Y| + \chi_D(q))$ field operations.

PROOF. By Proposition 2.18, the group $\langle Y \rangle$ is an irreducible 3-dimensional copy of $\mathrm{PSL}(2, q)$, so it must be a conjugate of the symmetric square of the natural representation. By using a change of basis from the algorithm in Theorem 1.12, we may assume that it is the symmetric square. Moreover, we can use Theorem 1.12 to constructively recognise $\langle Y \rangle$ and obtain the map ρ_G. We can also solve the constructive membership problem in the standard copy, and by evaluating straight line programs we obtain the maps π_3 and π_7.

It follows from Theorem 1.12 that the expected time complexity is as stated. □

3.2.2.2. *Finding a mapping element.* We now consider the algorithm for finding elements that map one point of $\mathcal{O}$ to another. The notation from Lemma 3.28 and 3.29 will be used.

If we let $M = \langle x \rangle \oplus \langle y \rangle$ then we can identify $\mathbb{P}(S_j)$ with the space of quadratic forms in x and y modulo scalars, so that $S_j = \langle x^2 \rangle \oplus \langle xy \rangle \oplus \langle y^2 \rangle$. Then $\varphi_G(\mathrm{C}_G(j)')$ acts projectively on $\mathbb{P}(S_j)$ and $|\mathbb{P}(S_j)| = |\mathbb{P}^2(\mathbb{F}_q)| = (q^3 - 1)/(q - 1) = q^2 + q + 1$.

PROPOSITION 3.30. *Use the notation from Lemma 3.28 and 3.29.*

(1) *The number of points in $\mathcal{O}$ that are contained in $\mathrm{Ker}(\varphi_V)$ is $q + 1$.*

(2) *The map* $\varphi_{\mathcal{O}}$ *restricted to* $\mathcal{O}$ *is not injective, and* $|\varphi_{\mathcal{O}}(\mathcal{O})| \geqslant q^2$.

PROOF. (1) The map φ_V is the projection onto S_j, so the kernel are those vectors that lie in T_j. From the proof of Proposition 2.18, with respect to a suitable basis, T_j is the -1-eigenspace of $h(-1)$. Hence by Proposition 2.17, $|\mathcal{O} \cap \mathbb{P}(T_j)| = q + 1$.

(2) Since $|\mathcal{O}| = q^3 + 1$ and $|\mathbb{P}(S_j)| = q^2 + q + 1$, it is clear that the map is not injective. In the above basis, the map φ_V is defined by $(p_1, \ldots, p_7) \mapsto (p_2, p_4, p_6)$. Hence if $P_\infty \neq P \in \mathcal{O}$ then $\varphi_{\mathcal{O}}(P) = (a^t : (ab)^t - c^t : -c - (bc)^t - a^{3t+2} - a^t b^{2t})$.

If $a = c = 0$ we do not get a point in $\mathbb{P}^2(\mathbb{F}_q)$ and if $a = 0$ and $c \neq 0$ we obtain q points. Now let $a \neq 0$ and let $(x, y) \in \mathbb{F}_q^2$ such that $x^2 + y \neq 0$. Then $-x^2 - y$ is a square in $\mathbb{F}_q$ if and only if $(-x^2 - y)^t$ is a square, so $(-x^2 - y)^{1-t}$ is always a square. Hence, if $c = 0$, $b = x^{3t}$ and $a = (-x^2 - y)^{(1-t)/4}$, the image of P is $(1 : x : y) \in \mathbb{P}^2(\mathbb{F}_q)$. This gives $q^2 - q$ points.

$\square$

PROPOSITION 3.31. *Under the action of* $H = \langle Y \rangle = \varphi_G(\mathrm{C}_G(j)')$, *the set* $\mathbb{P}(S_j)$ *splits up into* 3 *orbits.*

(1) *The orbit containing* xy, i.e. *the non-degenerate quadratic forms that represent* 0, *which has size* $q(q+1)/2$.
(2) *The orbit containing* $x^2 + y^2$, i.e. *the non-degenerate quadratic forms that do not represent* 0, *which has size* $q(q-1)/2$.
(3) *The orbit containing* x^2 *(and* y^2*)*, i.e. *the degenerate quadratic forms, which has size* $q + 1$.

The pre-image in $\mathrm{SL}(2, q)$ *of* $\rho_G(\varphi_G(\mathrm{C}_G(j)')_{xy})$ *is dihedral of order* $2(q-1)$, *generated by the matrices*

$$\begin{bmatrix} \alpha & 0 \\ 0 & \alpha^{-1} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{3.28}$$

where α *is a primitive element of* $\mathbb{F}_q$.

PROOF. Let $g = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ be any element of $\mathrm{PSL}(2, q)$, so that the symmetric square $h = \mathcal{S}^2(g) = \pi_3(g) \in \varphi_G(\mathrm{C}_G(j)')$. Notice that

$$h = \begin{bmatrix} a^2 & -ab & b^2 \\ ac & ad + bc & bd \\ c^2 & -ad & d^2 \end{bmatrix} \tag{3.29}$$

Let $P = (xy)h$, $Q = (x^2)h$ and $R = (x^2 + y^2)h$ be points in $\mathbb{P}(S_j)$. Then $P = (ac)x^2 + (ad + bc)xy + (bd)y^2$, and the equation $P = x^2$ implies that $b = 0$ or $d = 0$. If $b = 0$ then $d = 0$ or $a = 0$ which is impossible since $\det g = 1$. Similarly, we cannot have $d = 0$, and hence xy and x^2 are not in the same orbit.

Algorithm 3.5: FINDMAPPINGELEMENT($X, \mathrm{C}_G(j), P, Q$)

Input: Generating set X for $G = \mathrm{Ree}(q)$.
Points $P \neq Q \in \mathcal{O}$ such that both $\varphi_{\mathcal{O}}(P)$ and $\varphi_{\mathcal{O}}(Q)$ are non-degenerate and represent 0. Involution centraliser $\mathrm{C}_G(j)$ with the maps from Lemma 3.28 and 3.29.
Output: An element $h \in G$, written as an SLP in X, such that $Ph = Q$.
$P_3 := \varphi_{\mathcal{O}}(P)$
$Q_3 := \varphi_{\mathcal{O}}(Q)$
if $\exists$ upper triangular $g \in \mathrm{PSL}(2, q)$ such that $P_3\pi_3(g) = Q_3$ **then**
    $R_3 := \varphi_{\mathcal{O}}(P\pi_7(\pi_3(g)))$
    // Now $R_3 = Q_3$
    Find $c \in \mathrm{GL}(3, q)$ such that $(xy)c = R_3$
    Let D be the image in $\mathrm{PSL}(2, q)$ of the diagonal matrix in (3.28)
    $s := \pi_7(\pi_3(D)^c)$
    // Now $\langle s \rangle \leqslant \varphi_G^{-1}(H_{R_3})$
    $\delta, z :=$ DIAGONALISE(s)
    // Now $\delta = s^z$
    if $\exists\, \lambda \in \mathbb{F}_q^{\times}$ such that $(P\pi_7(\pi_3(g))z)h(\lambda) = Qz$ **then**
        $i :=$ DISCRETELOG($\delta, h(\lambda)$)
        // Now $\delta^i = h(\lambda)$
        return $\pi_7(\pi_3(g))s^i$
    end
end
return FAIL

Similarly, let $Q = a^2x^2 - (ab)xy + b^2y^2$, and then the equation $Q = x^2 + y^2$ implies that $a = 0$ or $b = 0$, which is impossible. Hence x^2 and $x^2 + y^2$ are not in the same orbit.

Finally, let $R = (a^2+c^2)x^2-(ab+cd)xy+(b^2+d^2)y^2$, and the equation $R = xy$ implies that $a^2 + c^2 = 0$. Since -1 is not a square in $\mathbb{F}_q$, we must have $a = c = 0$. But this is impossible since $\det g = 1$. Hence $x^2 + y^2$ and xy are not in the same orbit.

To verify the orbit sizes, consider the stabilisers of the three points. Clearly the equation $Q = x^2$ implies that $b = 0$, so the stabiliser of x^2 consists of the (projections of the) lower triangular matrices. There are $q - 1$ choices for a and q choices for c, so the stabiliser has size $q(q - 1)/2$ modulo scalars, and the index in H is therefore $q + 1$.

Similarly, the equation $P = xy$ implies that $ac = bd = 0$. If $a = d = 0$ we obtain a matrix of order 2 and if $b = c = 0$ we obtain a diagonal matrix. It follows that the stabiliser is dihedral of order $q - 1$, and that the pre-image in $\mathrm{SL}(2, q)$ is as in (3.28).

Finally, in a similar way we obtain that the stabiliser of x^2+y^2 has order $q+1$, and hence the three orbits make up the whole of $\mathbb{P}(S_j)$. □

The algorithm that maps one point to another is given as Algorithm 3.5.

3.2.2.3. *Finding a stabilising element.* The complete algorithm for finding a uniformly random element of G_P is then as follows, given a generating set X for G and $P \in \mathcal{O}$.

(1) Find an involution $j \in G$.
(2) Compute probable generators for $\mathrm{C}_G(j)$ using the Bray algorithm, and probable generators for $\mathrm{C}_G(j)'$ by taking commutators of the generators of $\mathrm{C}_G(j)$.
(3) Use the MeatAxe to verify that the module for $\mathrm{C}_G(j)'$ splits up only as in Proposition 2.18. Use Theorem 1.13 to verify that we have the whole of $\mathrm{C}_G(j)'$. Return to the previous step if not.
(4) Compute the maps $\varphi_{\mathcal{O}}$ and φ_G using Lemma 3.28. Return to the first step if P lies in the kernel of φ_V, if $\varphi_{\mathcal{O}}(P)$ is degenerate, or if it does not represent 0.
(5) Compute the maps from Lemma 3.29.
(6) Take random $g_1 \in \mathrm{C}_G(j)'$ and let $Q = Pg_1$. Then $\varphi_{\mathcal{O}}(Q) = \varphi_{\mathcal{O}}(P)\varphi_G(g_1)$, so Q is not in the kernel of φ_V, and $\varphi_{\mathcal{O}}(Q)$ is non-degenerate and represents 0. Repeat until $P \neq Q$.
(7) Use Algorithm 3.5 to find $g_2 \in \mathrm{C}_G(j)'$ such that $Q = Pg_2$. Return to the previous step if it fails, and otherwise return $g_1 g_2^{-1}$.

3.2.2.4. *Correctness and complexity.*

Lemma 3.32. *If $P \in \mathcal{O}$ is uniformly random, such that $P \not\subseteq \mathrm{Ker}(\varphi_V)$, then $\varphi_{\mathcal{O}}(P)$ is non-degenerate and represents 0 with probability at least $1/2 + \mathrm{O}(1/q)$.*

Proof. Since P is uniformly random and $\varphi_{\mathcal{O}}$ was chosen independently of P, it follows that $\varphi_{\mathcal{O}}(P)$ is uniformly random from $\varphi_{\mathcal{O}}(\mathcal{O})$. From the proof of Proposition 3.30, with probability $1 - \mathrm{O}(1/q)$, $\varphi_{\mathcal{O}}(P) = x^2 + bxy + cy^2$ where $(1 : b : c)$ is uniformly distributed in $\mathbb{P}^2(\mathbb{F}_q)$ such that $b^2 + c \neq 0$.

This represents 0 if the discriminant $b^2 - c$ is a non-zero square in $\mathbb{F}_q$. This is not the case if $b^2 = c$, but since $b^2 + c \neq 0$, this implies $b = c = 0$. If $b^2 - c \neq 0$ then it is a square with probability 1/2, so

$$\Pr[b^2 - c \in (\mathbb{F}_q^{\times})^2] = \frac{1}{2}(1 - \frac{1}{q^2 - q})$$

and the Lemma follows. □

Lemma 3.33. *If $P, Q \in \varphi_{\mathcal{O}}(\mathcal{O})$ are uniformly random, such that $\varphi_{\mathcal{O}}(P)$ and $\varphi_{\mathcal{O}}(Q)$ represent 0, then the probability that there exists an element $g \in \mathrm{PSL}(2, q)$, such that the pre-image of g in $\mathrm{SL}(2, q)$ is upper triangular and $P\pi_3(g) = Q$, is at least $1/2 + \mathrm{O}(1/q)$.*

Proof. Let $P = x^2 + axy + by^2$, $Q = x^2 + lxy + ny^2$ and

$$g = \begin{bmatrix} u & v \\ 0 & 1/u \end{bmatrix} \tag{3.30}$$

where $(1 : a : b)$ and $(1 : l : n)$ are uniformly distributed in $\mathbb{P}^2(\mathbb{F}_q)$, $u, v \in \mathbb{F}_q$ and $u \neq 0$.

We want to determine u, v such that $P\pi_3(g) = Q$. Note that g is the pre-image in $\mathrm{SL}(2, q)$ of an element in $\mathrm{PSL}(2, q)$ and therefore $\pm u$ determine the same element of $\mathrm{PSL}(2, q)$. The map π_3 is the symmetric square map, so

$$\pi_3(g) = \mathcal{S}^2(g) = \begin{bmatrix} u^2 & -uv & v^2 \\ 0 & 1 & v/u \\ 0 & 0 & 1/u^2 \end{bmatrix} \tag{3.31}$$

This leads to the following equations:

$$u^2 = C \tag{3.32}$$

$$-uv + a = Cl \tag{3.33}$$

$$v^2 + avu^{-1} + bu^{-2} = Cn \tag{3.34}$$

for some $C \in \mathbb{F}_q^{\times}$. We can solve for u in (3.32) and for v in (3.33), so that (3.34) becomes

$$C^2(n - m^2) + a^2 - b = 0 \tag{3.35}$$

This quadratic equation has a solution if the discriminant $-(n-m^2)(a^2-b) \in (\mathbb{F}_q^{\times})^2$. This does not happen if $n = m^2$ or $b = a^2$, which each happens with probability $q/(q^2 + q + 1)$. If the discriminant is non-zero then it is a square with probability $1/2$. Therefore, the probability that we can find g is

$$\Pr[-(n - m^2)(a^2 - b) \in (\mathbb{F}_q^{\times})^2] = \frac{1}{2}(1 - \frac{q}{q^2 + q + 1})^2$$

This is $1/2 + \mathrm{O}\big(1/q\big)$ and the Lemma follows. □

THEOREM 3.34. *If Algorithm 3.5 returns an element g, then $Pg = Q$. If P and Q are uniformly random, such that $\varphi_{\mathcal{O}}(P)$ and $\varphi_{\mathcal{O}}(Q)$ represent 0, then the probability that Algorithm 3.5 finds such an element is at least $1/4 + \mathrm{O}\big(1/q\big)$.*

PROOF. By Proposition 3.31, the point R_3 is in the same orbit as xy, so the element c at line 8 can easily be found by diagonalising the form corresponding to R_3. Then $\pi_3(D)^c \in H_{R_3}$ is of order $(q-1)/2$. Hence s also has order $(q-1)/2$, and $s \in \varphi_G^{-1}(H_{R_3})$.

By definition of Q, there exists $h \in \mathrm{C}_G(j)'$ such that $Ph = Q$, and if we let $R = P\pi_7(g)$ then $R\pi_7(g)^{-1}h = Q$ and $\varphi_{\mathcal{O}}(R) = R_3 = Q_3$. Hence $\varphi_G(\pi_7(g)^{-1}h) \in H_{Q_3}$, and therefore $\pi_7(g)^{-1}h \in \varphi_G^{-1}(H_{R_3})$.

By Proposition 3.31, $\varphi_G^{-1}(H_{R_3})$ is dihedral of order $q - 1$, and s generates a subgroup of index 2. Therefore $\Pr[\pi_7(g)^{-1}h \in \langle s \rangle] = 1/2$, which is the success probability of line 14.

It is straightforward to determine if λ exists, since $h(\lambda)$ is diagonal. The success probability of line 5 is given by Lemma 3.33. Hence the success probability of the algorithm is as stated. □

THEOREM 3.35. *Assume an oracle for the discrete logarithm problem in* $\mathbb{F}_q$. *The time complexity of Algorithm* 3.5 *is* $\mathrm{O}\big(\log(q)^3 + \chi_D(q)\big)$ *field operations. The length of the returned* SLP *is* $\mathrm{O}\big(\log(q) \log\log(q)\big)$.

PROOF. By Lemma 3.33, line 5 involves solving a quadratic equation in $\mathbb{F}_q$, and hence uses $\mathrm{O}(1)$ field operations. Evaluating the maps π_3 and π_7 uses $\mathrm{O}\big(\log(q)^3\big)$ field operations, and it is clear that the rest of the algorithm can be done using $\mathrm{O}\big(\chi_D(q)\big)$ field operations.

By Theorem 1.12, the length of the SLP from the constructive membership testing in $\mathrm{PSL}(2, q)$ is $\mathrm{O}\big(\log(q) \log\log(q)\big)$, which is therefore also the length of the returned SLP. ☐

COROLLARY 3.36. *Assume an oracle for the discrete logarithm problem in* $\mathbb{F}_q$. *There exists a Las Vegas algorithm that, given* $\langle X \rangle \leqslant \mathrm{GL}(7, q)$ *such that* $G = \langle X \rangle = \mathrm{Ree}(q)$ *and* $P \in \mathcal{O}$, *computes a random element of* G_P *as an* SLP *in* X. *The expected time complexity of the algorithm is* $\mathrm{O}\big(\xi \log\log(q) + \log(q)^3 + \chi_D(q)\big)$ *field operations. The length of the returned* SLP *is* $\mathrm{O}\big(\log(q) \log\log(q)\big)$.

PROOF. The algorithm is given in Section 3.2.2.3.

An involution is found by finding a random element and then use Proposition 1.4. Hence by Corollary 2.24, the expected time to find an involution is $\mathrm{O}\big(\xi + \log(q) \log\log(q)\big)$ field operations.

As described in Section 1.2.9.2, the Bray algorithm will produce uniformly random elements of the centraliser. Hence as described in Section 1.2.6, we can also obtain uniformly random elements of its derived group. By Proposition 2.27, two random elements will generate $\mathrm{PSL}(2, q)$ with high probability. This implies that the expected time to obtain probable generators for $\mathrm{PSL}(2, q)$ is $\mathrm{O}(1)$ field operations.

By Proposition 3.31, the point Q is equal to P with probability $2/(q(q+1))$ and by Lemma 3.32 the point P do not represent zero with probability $1/2$, so the expected time of the penultimate step is $\mathrm{O}(1)$ field operations.

Since the points P, Q can be considered uniformly random and independent in Algorithm 3.5, the element returned by that algorithm is uniformly random. Hence the element returned by the algorithm in Section 3.2.2.3 is uniformly random.

The expected time complexity of the last step is given by Theorem 3.35 and 3.34. It follows by the above and from Lemma 3.28 and 3.29 and Corollary 2.23 that the expected time complexity of the algorithm in Section 3.2.2.3 is as stated.

The algorithm is clearly Las Vegas, since it is straightforward to check that the element we compute really fixes the point P. ☐

REMARK 3.37. The elements returned by the algorithm in Corollary 3.36 are not uniformly random from the *whole* of G_P, but from $G_P \cap \mathrm{C}_G(j)$. Hence, to obtain generators for the whole stabiliser, it is necessary to execute the algorithm at least twice, with different choices of the involution j.

REMARK 3.38. The algorithm in Corollary 3.36 works in any conjugate of $\operatorname{Ree}(q)$, since it does not assume that the matrices lie in the standard copy.

3.2.3. Constructive membership testing. We now describe the constructive membership algorithm for our standard copy $\operatorname{Ree}(q)$. The matrix degree is constant here, so we set $\xi = \xi(7)$. Given a set of generators X, such that $G = \langle X \rangle = \operatorname{Ree}(q)$, and given an element $g \in G$, we want to express g as an `SLP` in X. Membership testing is straightforward, using the first step from the algorithm in Theorem 3.27, and will not be considered here.

The general structure of the algorithm is the same as the algorithm for the same problem in the Suzuki groups. It consists of a preprocessing step and a main step.

3.2.3.1. *Preprocessing.* The preprocessing step consists of finding "standard generators" for $\mathrm{O}_3(G_{P_\infty}) = U(q)$ and $\mathrm{O}_3(G_{P_0})$. In the case of $\mathrm{O}_3(G_{P_\infty})$ the standard generators are defined as matrices

$$\{S(a_i, x_i, y_i)\}_{i=1}^{n} \cup \{S(0, b_i, z_i)\}_{i=1}^{n} \cup \{S(0, 0, c_i)\}_{i=1}^{n} \qquad (3.36)$$

for some unspecified $x_i, y_i, z_i \in \mathbb{F}_q$, such that $\{a_1, \dots, a_n\}$, $\{b_1, \dots, b_n\}$, $\{c_1, \dots, c_n\}$ form vector space bases of $\mathbb{F}_q$ over $\mathbb{F}_3$ (so $n = \log_3 q = 2m + 1$).

LEMMA 3.39. *There exist algorithms for the following row reductions.*

(1) *Given* $g = h(\lambda)S(a, b, c) \in G_{P_\infty}$*, find* $h \in \mathrm{O}_3(G_{P_\infty})$ *expressed in the standard generators, such that* $gh = h(\lambda)$*.*
(2) *Given* $g = S(a, b, c)h(\lambda) \in G_{P_\infty}$*, find* $h \in \mathrm{O}_3(G_{P_\infty})$ *expressed in the standard generators, such that* $hg = h(\lambda)$*.*
(3) *Given* $P_\infty \neq P \in \mathcal{O}$*, find* $g \in \mathrm{O}_3(G_{P_\infty})$ *expressed in the standard generators, such that* $Pg = P_0$*.*

Analogous algorithms exist for G_{P_0}*. If the standard generators are expressed as* `SLP`*s of length* $\mathrm{O}(n)$*, the elements returned will have length* $\mathrm{O}(n \log(q))$*. The time complexity of the algorithms is* $\mathrm{O}(\log(q)^3)$ *field operations.*

PROOF. The algorithms are as follows.

(1) (a) Solve a linear system of size $\log(q)$ to construct the linear combination $-a = -g_{2,1}/g_{2,2} = \sum_{i=1}^{2m+1} \alpha_i a_i$ with $\alpha_i \in \mathbb{F}_3$. Let $h' = \prod_{i=1}^{2m+1} S(a_i, x_i, y_i)^{\alpha_i}$ and $g' = gh'$, so that $g' = h(\lambda)S(0, b', c')$ for some $b', c' \in \mathbb{F}_q$.
  (b) Solve a linear system of size $\log(q)$ to construct the linear combination $-b' = -g'_{3,1}/g'_{3,3} = \sum_{i=1}^{2m+1} \beta_i b_i$ with $\beta_i \in \mathbb{F}_3$. Let $h'' = \prod_{i=1}^{2m+1} S(0, b_i, y_i)^{\beta_i}$ and $g'' = g'h''$, so that $g'' = h'(\lambda)S(0, 0, c'')$ for some $c'' \in \mathbb{F}_q$.
  (c) Solve a linear system of size $\log(q)$ to construct the linear combination $-c'' = -g''_{4,1}/g''_{4,4} = \sum_{i=1}^{2m+1} \gamma_i c_i$ with $\gamma_i \in \mathbb{F}_3$. Let $h''' = \prod_{i=1}^{2m+1} S(0, 0, z_i)^{\gamma_i}$ and $g''' = g''h'''$, so that $g''' = h'(\lambda)$.
  (d) Now $h = h'h''h'''$.

(2) Analogous to the previous case.
(3) (a) Normalise P so that $P = (1 : p_1 : \cdots : p_6)$.
(b) Let $\alpha = -p_1^{3t}$, $\beta = (p_1\alpha + p_2)^{3t}$ and $\gamma = ((\alpha\beta)^t + p_1(\alpha^{t+1} + \beta^t) + p_2\alpha^t + p_3)^{3t}$. Then $S(\alpha, \beta, \gamma)$ maps P to P_∞.
(c) Use the algorithm above to find $h \in \mathrm{O}_3(G_{P_\infty})$ such that $S(\alpha, \beta, \gamma)h = 1$, and hence express $S(\alpha, \beta, \gamma)$ in the standard generators.

Clearly the dominating term in the time complexity is the solving of the linear systems, which requires $\mathrm{O}\big(\log(q)^3\big)$ field operations. The elements returned are constructed using $\mathrm{O}\big(\log(q)\big)$ multiplications, hence the length of the SLP follows. □

THEOREM 3.40. *Given an oracle for the discrete logarithm problem in $\mathbb{F}_q$, the preprocessing step is a Las Vegas algorithm that finds standard generators for $\mathrm{O}_3(G_{P_\infty})$ and $\mathrm{O}_3(G_{P_0})$ as SLPs in X of length $\mathrm{O}\big(\log(q)(\log\log(q))^2\big)$. It has expected time complexity $\mathrm{O}\big((\xi \log\log(q) + \log(q)^3 + \chi_D(q)) \log\log(q)\big)$ field operations.*

PROOF. The preprocessing step proceeds as follows.

(1) Find random elements $a_1 \in G_{P_\infty}$ and $b_1 \in G_{P_0}$ using the algorithm from Corollary 3.36. Repeat until a_1 can be diagonalised to $h(\lambda) \in G$, where $\lambda \in \mathbb{F}_q^\times$ and λ does not lie in a proper subfield of $\mathbb{F}_q$. Do similarly for b_1.
(2) Find random elements $a_2 \in G_{P_\infty}$ and $b_2 \in G_{P_0}$ using the algorithm from Corollary 3.36. Let $c_1 = [a_1, a_2]$, $c_2 = [b_1, b_2]$. Repeat until $|c_1| = |c_2| = 9$.
(3) Let $Y_\infty = \{c_1, a_1\}$ and $Y_0 = \{c_2, b_1\}$. As standard generators for $\mathrm{O}_3(G_{P_\infty})$ we now take $U = U_1 \cup U_2$ where

$$U_1 = \bigcup_{i=1}^{2m+1} \left\{ c_1^{a_1^i}, (c_1^3)^{a_1^i} \right\} \tag{3.37}$$

and

$$U_2 = \bigcup_{1 \leqslant i < j \leqslant 2m+1} \left\{ [c_1^{a_1^i}, c_1^{a_1^j}] \right\} \tag{3.38}$$

Similarly we obtain L for $\mathrm{O}_3(G_{P_0})$.

It follows from (2.36) and (2.39) that (3.37) and (3.38) provides the standard generators for G_{P_∞}. These are expressed as SLPs in X, since this is true for the elements returned from the algorithm described in Corollary 3.36. Hence the algorithm is Las Vegas.

By Corollary 3.36, the expected time to find a_1 and b_1 is $\mathrm{O}\big(\xi \log\log(q) + \log(q)^3 + \chi_D(q)\big)$, and these are uniformly distributed independent random elements. The elements of order dividing $q-1$ can be diagonalised as required. By Proposition 2.15, the proportion of elements of order $q-1$ in G_{P_∞} and G_{P_0} is $\phi(q-1)/(q-1)$. Hence the expected time for the first step is $\mathrm{O}\big((\xi \log\log(q) + \log(q)^3 + \chi_D(q)) \log\log(q)\big)$ field operations.

Similarly, by Proposition 2.29 the expected time for the second step is

$$\mathrm{O}\big((\xi \log\log(q) + \log(q)^3 + \chi_D(q)) \log\log(q)\big)$$

field operations.

By the remark preceding the Theorem, L determines three sets of field elements $\{a_1, \ldots, a_{2m+1}\}$, $\{b_1, \ldots, b_{2m+1}\}$ and $\{c_1, \ldots, c_{2m+1}\}$. By (2.39), in this case each $a_i = a\lambda^i$, $b_i = b\lambda^{i(t+2)}$ and $c_i = c\lambda^{i(t+3)}$, for some fixed $a, b, c \in \mathbb{F}_q^{\times}$, where λ is as in the algorithm. Since λ does not lie in a proper subfield, these sets form vector space bases of $\mathbb{F}_q$ over $\mathbb{F}_3$.

To determine if a_1 or b_1 diagonalise to some $h(\lambda)$ it is sufficient to consider the eigenvalues on the diagonal, since both a_1 and b_1 are triangular. To determine if λ lies in a proper subfield, it is sufficient to determine if $|\lambda| \mid 3^n - 1$, for some proper divisor n of $2m + 1$. Hence the dominating term in the complexity is the first step. □

3.2.3.2. *Main algorithm.* Given $g \in G$ we now show the procedure for expressing g as an `SLP`. It is given as Algorithm 3.6.

Algorithm 3.6: ElementToSLP(U, L, g)

1 **Input**: Standard generators U for G_{P_∞} and L for G_{P_0}. Matrix $g \in \langle X \rangle = G$.
2 **Output**: `SLP` for g in X
3 **repeat**
4   **repeat**
5         $r :=$ Random(G)
6   **until** gr has an eigenspace $Q \in \mathcal{O}$ and $P \neq Q$
7   Find $z_1 \in G_{P_\infty}$ using U such that $Qz_1 = P_0$.
8   // Now $(gr)^{z_1} \in G_{P_0}$
9   Find $z_2 \in G_{P_0}$ using L such that $(gr)^{z_1} z_2 = h(\lambda)$ for some $\lambda \in \mathbb{F}_q^{\times}$
10     $x := \mathrm{Tr}(h(\lambda))$
11 **until** $x - 1$ is a square in $\mathbb{F}_q^{\times}$
12 // Express diagonal matrix as `SLP`
13 Find $u = S(0, 0, \sqrt{(x-1)^{3t}}) S(0, 1, 0)^{\Upsilon}$ using $U \cup L$
14 // Now $\mathrm{Tr}(u) = x$
15 Let $P_1, P_2 \in \mathcal{O}$ be the fixed points of u
16 Find $a \in G_{P_\infty}$ using U such that $P_1 a = P_0$
17 Find $b \in G_{P_0}$ using L such that $(P_2 a)b = P_\infty$
18 // Now $u^{ab} \in G_{P_\infty} \cap G_{P_0} = H(q)$, so $u^{ab} \in \{h(\lambda)^{\pm 1}\}$
19 **if** $u^{ab} = h(\lambda)$
  then
20   Let w be the `SLP` for $(u^{ab} z_2^{-1})^{z_1^{-1}} r^{-1}$
21   **return** W
  else
22   Let w be the `SLP` for $((u^{ab})^{-1} z_2^{-1})^{z_1^{-1}} r^{-1}$
23   **return** w
end

3.2.3.3. *Correctness and complexity.*

Theorem 3.41. *Algorithm* 3.6 *is correct, and is a Las Vegas algorithm.*

Proof. First observe that since r is randomly chosen, we obtain it as an `SLP`.

The elements found at lines 7 and 9 can be computed using Lemma 3.39, so we can obtain them as `SLP`s.

The element u found at line 13 clearly has trace x. Because u can be computed using Lemma 3.39, we obtain it as an SLP. From Proposition 2.25 we know that u is conjugate to $h(\lambda)^{\pm 1}$ and therefore must fix two points of $\mathcal{O}$. Hence lines 16 and 17 make sense, and the elements found can again be computed using Lemma 3.39, so we obtain them as SLPs.

Finally, the elements that make up w have been found as SLPs, and it is clear that if we evaluate w we obtain g. Hence the algorithm is Las Vegas and the Theorem follows. □

THEOREM 3.42. *Algorithm* 3.6 *has expected time complexity* $\mathrm{O}\big(\xi + \log(q)^3\big)$ *field operations and the length of the returned* SLP *is* $\mathrm{O}\big((\log(q)\log\log(q))^2\big)$.

PROOF. It follows immediately from Lemma 3.39 that the lines 7, 9, 13, 16 and 17 use $\mathrm{O}\big(\log(q)^3\big)$ field operations.

From Corollary 2.24, the expected time to find r is $\mathrm{O}\big(\xi\big)$ field operations. Half of the elements of $\mathbb{F}_q^{\times}$ are squares, and x is uniformly random, hence the expected time of the outer repeat statement is $\mathrm{O}\big(\xi + \log(q)^3\big)$ field operations.

Finding the fixed points of u, and performing the check at line 6 only amounts to considering eigenvectors, which is $\mathrm{O}\big(\log q\big)$ field operations. Thus the expected time complexity of the algorithm is $\mathrm{O}\big(\xi + \log(q)^3\big)$ field operations.

From Theorem 3.40 each standard generator SLP has length $\mathrm{O}\big(\log(q)(\log\log(q))^2\big)$ and hence w will have length $\mathrm{O}\big((\log(q)\log\log(q))^2\big)$. □

3.2.4. Conjugates of the standard copy. Now assume that we are given a conjugate G of $\mathrm{Ree}(q)$, and we turn to the problem of finding some $g \in \mathrm{GL}(7, q)$ such that $G^g = \mathrm{Ree}(q)$, thus obtaining an algorithm that finds effective isomorphisms from any conjugate of $\mathrm{Ree}(q)$ to the standard copy. The matrix degree is constant here, so we set $\xi = \xi(7)$.

LEMMA 3.43. *There exists a Las Vegas algorithm that, given* $\langle X\rangle \leqslant \mathrm{GL}(7, q)$ *such that* $\langle X\rangle^h = \mathrm{Ree}(q)$ *for some* $h \in \mathrm{GL}(7, q)$*, finds a point* $P \in \mathcal{O}^{h^{-1}} = \big\{Qh^{-1} \mid Q \in \mathcal{O}\big\}$*. The algorithm has expected time complexity*

$$\mathrm{O}\big((\xi + \log(q)\log\log(q))\log\log(q)\big)$$

field operations.

PROOF. Clearly $\mathcal{O}^{h^{-1}}$ is the set on which $\langle X\rangle$ acts doubly transitively. For a matrix $h(\lambda) \in \mathrm{Ree}(q)$ we see that the eigenspaces corresponding to the eigenvalues $\lambda^{\pm t}$ will be in $\mathcal{O}$. Moreover, every element of order dividing $q-1$, in every conjugate G of $\mathrm{Ree}(q)$, will have eigenvalues of the form $\mu^{\pm t}, \mu^{\pm(1-t)}, \mu^{\pm(2t-1)}$, for some $\mu \in \mathbb{F}_q^{\times}$, and the eigenspaces corresponding to $\mu^{\pm t}$ will lie in the set on which G acts doubly transitively.

Hence to find a point $P \in \mathcal{O}^{h^{-1}}$ it is sufficient to find a random $g \in \langle X\rangle$ of order dividing $q-1$. We compute the order using expected $\mathrm{O}\big(\log(q)\log\log(q)\big)$ field operations, and by Corollary 2.24, the expected number of iterations to find the element is $\mathrm{O}\big(\log\log(q)\big)$ field operations. We then find the eigenspaces of g.

Clearly this is a Las Vegas algorithm with the stated time complexity. □

CONJECTURE 3.44. *Let $P = (1 : p_2 : \cdots : p_7) \in \mathcal{O}^x$ and $Q = (1 : q_2 : \cdots : q_7) \in \mathcal{O}^x$ for some $x \in \mathrm{GL}(7, q)$. Then for all $a, b, c, s \in \mathbb{F}_q^\times$, the ideal in $\mathbb{F}_q[\alpha, \beta, \gamma, \delta]$ generated by*

$$p_3^{3t}\beta\delta - p_2^{3t+3}\gamma\alpha^3\beta + p_2 s\alpha\beta + p_2^2 p_3\alpha^2\beta - p_5 c\,; \tag{3.39}$$

$$(p_4 s)^{3t} + (p_2p_3)^{3t}\gamma\delta\alpha - p_3p_4s\alpha\beta + p_2p_3^2\alpha^2\beta^2 - p_2^{6t}p_2^3\gamma^2\alpha^4 - p_6 b\,; \tag{3.40}$$

$$\begin{aligned} &-(p_4s)^{3t}p_2\alpha - p_2^{3t+1}p_3^{3t}\gamma\delta\alpha - p_3^{3t+1}\gamma\beta + p_2^{6t+4}\gamma^2\alpha^4 + \\ &(p_2p_3)^2\alpha^2\beta^2 + p_2^{3t+3}p_3\gamma\alpha^3\beta - p_7 a - (p_4 s)^2 \end{aligned} \tag{3.41}$$

as well as the corresponding 3 polynomials from Q, is zero-dimensional.

LEMMA 3.45. *Assume Conjecture 3.44. There exists a Las Vegas algorithm that, given $\langle X\rangle \leqslant \mathrm{GL}(7, q)$ such that $\langle X\rangle^d = \mathrm{Ree}(q)$ where $d = \mathrm{diag}(d_1, d_2, d_3, d_4, d_5, d_6, d_7) \in \mathrm{GL}(7, q)$, finds a diagonal matrix $e \in \mathrm{GL}(7, q)$ such that $\langle X\rangle^e = \mathrm{Ree}(q)$. The expected time complexity is $\mathrm{O}\big(|X| + (\xi + \log(q)\log\log(q))\log\log(q)\big)$ field operations.*

PROOF. Let $G = \langle X\rangle$. Since $G^d = \mathrm{Ree}(q)$, G must preserve the symmetric bilinear form

$$K = dJd = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & d_1d_7 \\ 0 & 0 & 0 & 0 & 0 & d_2d_6 & 0 \\ 0 & 0 & 0 & 0 & d_3d_5 & 0 & 0 \\ 0 & 0 & 0 & -d_4^2 & 0 & 0 & 0 \\ 0 & 0 & d_3d_5 & 0 & 0 & 0 & 0 \\ 0 & d_2d_6 & 0 & 0 & 0 & 0 & 0 \\ d_1d_7 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.42}$$

where J is given by (2.40). Using the MeatAxe, we can find this form, which is determined up to a scalar multiple. In the case where $-K_{4,4}$ turns out to be a non-square in $\mathbb{F}_q$ we can therefore multiply K with a non-square scalar matrix. The diagonal matrix $e = \mathrm{diag}(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ that we want to find is also determined up to a scalar multiple (and up to multiplication by a diagonal matrix in $\mathrm{Ree}(q)$).

Since e must take J to K, we must have $K_{1,7} = d_1d_7 = e_1e_7$, $K_{2,6} = d_2d_6 = e_2e_6$, $K_{3,5} = d_3d_5 = e_3e_5$ and $K_{4,4} = -d_4^2 = -e_4^2$. Because e is determined up to a scalar multiple, we can choose $e_1 = 1$, $e_7 = K_{1,7}$ and $e_4 = s = \sqrt{-K_{4,4}}$. Furthermore, $e_5 = K_{3,5}/e_3$ and $e_6 = K_{2,6}/e_2$ so it only remains to determine e_2 and e_3.

To conjugate G into $\mathrm{Ree}(q)$, we must have $Pe \in \mathcal{O}$ for every $P \in \mathcal{O}^{d^{-1}}$, which is the set on which G acts doubly transitively. By Lemma 3.43, we can find $P = (1 : p_2 : p_3 : p_4 : p_5 : p_6 : p_7) \in \mathcal{O}^{d^{-1}}$, and the condition $Pe = (1 : p_2e_2 : p_3e_3 : p_4s : p_5K_{3,5}/e_3 : p_6K_{2,6}/e_2 : p_7K_{1,7}) \in \mathcal{O}$ is given by (2.41) and amounts to the polynomial equations given in Conjecture 3.44, with $\alpha = e_2$, $\beta = e_3$, $\gamma = e_2^t$, $\delta = e_3^t$, $K_{1,7} = a$, $K_{2,6} = b$, $K_{3,5} = c$.

By finding another random point $Q \in \mathcal{O}^{d^{-1}}$ using Lemma 3.43, we obtain 6 polynomials which we label $1P$, $2P$, $3P$, $1Q$, $2Q$ and $3Q$. Conjecture 3.44 asserts that the resulting ideal is zero-dimensional.

Now it follows from Proposition 2.28 that with high probability we have $p_3 \neq 0$ and $p_3^{3t} q_2^{3t+3} \neq q_3^{3t} p_2^{3t+3}$, so we repeat until P and Q satisfy this. Then we can solve for δ from $1P$ and γ from $1Q$, as

$$\delta = \frac{\alpha^3 \gamma \beta p_2^{3t+3} - \alpha^2 \beta p_2^2 p_3 - \alpha \beta s p_2 p_4 + c p_5}{p_3^{3t} \beta} \tag{3.43}$$

$$\gamma = \frac{\alpha^2 \beta p_3^{3t} q_2^2 q_3 - \alpha^2 \beta p_2^2 p_3 q_3^{3t} - \alpha \beta s p_2 p_4 q_3^{3t} + c p_5 q_3^{3t} + \alpha \beta s p_3^{3t} q_2 q_4 - c p_3^{3t} q_5}{\alpha^3 \beta (p_3^{3t} q_2^{3t+3} - p_2^{3t+3} q_3^{3t})} \tag{3.44}$$

and if we substitute these into the other equations we obtain 4 polynomials in α and β, which generate a zero-dimensional ideal. The variety of this ideal can now be found using Theorem 1.2.

Hence we can find e_2 and e_3. The diagonal matrix

$$e = \operatorname{diag}(1, e_2, e_3, s, K_{3,5} e_3^{-1}, K_{2,6} e_2^{-1}, K_{1,7})$$

now satisfies $G^e = \operatorname{Ree}(q)$.

By Lemma 3.43, Lemma 3.44, Theorem 1.2 and Section 1.2.10.1, this is a Las Vegas algorithm with the stated time complexity. □

LEMMA 3.46. *There exists a Las Vegas algorithm that, given subsets X, Y_P and Y_Q of $\mathrm{GL}(7, q)$ such that $\mathrm{O}_3(G_P) < \langle Y_P \rangle \leqslant G_P$ and $\mathrm{O}_3(G_Q) < \langle Y_Q \rangle \leqslant G_Q$, respectively, where $\langle X \rangle = G$, $G^h = \operatorname{Ree}(q)$ for some $h \in \mathrm{GL}(7, q)$ and $P, Q \in \mathcal{O}^{h^{-1}}$, finds $k \in \mathrm{GL}(7, q)$ such that $(G^k)^d = \operatorname{Ree}(q)$ for some diagonal matrix $d \in \mathrm{GL}(7, q)$. The algorithm has expected time complexity $\mathrm{O}(|X|)$ field operations.*

PROOF. Notice that the natural module $V = \mathbb{F}_q^7$ of $U(q)H(q)$ is uniserial with seven non-zero submodules, namely

$$V_i = \{(v_1, v_2, v_3, v_4, v_5, v_6, v_7) \in \mathbb{F}_q^7 \mid v_j = 0, j > i\}$$

for $i = 1, \ldots, 7$. Hence the same is true for $\langle Y_P \rangle$ and $\langle Y_Q \rangle$ (but the submodules will be different) since they lie in conjugates of $U(q)H(q)$.

Now the algorithm proceeds as follows.

(1) Let $V = \mathbb{F}_q^7$ be the natural module for $\langle Y_P \rangle$ and $\langle Y_Q \rangle$. Find composition series $V = V_7^P > V_6^P > V_5^P > V_4^P > V_3^P > V_2^P > V_1^P$ and $V = V_7^Q > V_6^Q > V_5^Q > V_4^Q > V_3^Q > V_2^Q > V_1^Q$ using the MeatAxe.
(2) Let $U_1 = V_1^P$, $U_2 = V_2^P \cap V_6^Q$, $U_3 = V_3^P \cap V_5^Q$, $U_4 = V_4^P \cap V_4^Q$, $U_5 = V_5^P \cap V_3^Q$, $U_6 = V_6^P \cap V_2^Q$ and $U_7 = V_1^Q$. For each $i = 1, \ldots, 7$, choose $u_i \in U_i$.
(3) Now let k be the matrix such that k^{-1} has u_i as row i, for $i = 1, \ldots, 7$.

The motivation for the second step is analogous to the proof of Theorem 3.16.

Thus the matrix k found in the algorithm satisfies that $z = kd$ for some diagonal matrix $d \in \mathrm{GL}(7, q)$. Since $\mathrm{Ree}(q) = G^h = G^z = (G^k)^d$, the algorithm returns a correct result, and it is Las Vegas because the MeatAxe is Las Vegas. Clearly it has the same time complexity as the MeatAxe. $\square$

THEOREM 3.47. *Assume Conjecture* 3.44 *and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$*. There exists a Las Vegas algorithm that, given a conjugate* $\langle X \rangle$ *of* $\mathrm{Ree}(q)$*, finds* $g \in \mathrm{GL}(7, q)$ *such that* $\langle X \rangle^g = \mathrm{Ree}(q)$*. The algorithm has expected time complexity* $\mathrm{O}\big((\xi \log\log(q) + \log(q)^3 + \chi_D(q)) \log\log(q) + |X|\big)$ *field operations.*

PROOF. Let $G = \langle X \rangle$. By Remark 3.38, we can use Corollary 3.36 in G, so we can find generators for a stabiliser of a point in G, using the algorithm described in Theorem 3.40.

(1) Find points $P, Q \in \mathcal{O}^{h^{-1}}$ using Lemma 3.43. Repeat until $P \neq Q$.
(2) Find generating sets Y_P and Y_Q such that $\mathrm{O}_3(G_P) < \langle Y_P \rangle \leqslant G_P$ and $\mathrm{O}_3(G_Q) < \langle Y_Q \rangle \leqslant G_Q$ using the first two steps of the algorithm from the proof of Theorem 3.40.
(3) Find $k \in \mathrm{GL}(7, q)$ such that $(G^k)^d = \mathrm{Ree}(q)$ for some diagonal matrix $d \in \mathrm{GL}(7, q)$, using Lemma 3.46.
(4) Find a diagonal matrix e using Lemma 3.45.
(5) Now $g = ke$ satisfies that $G^g = \mathrm{Ree}(q)$.

Be Lemma 3.14, 3.46 and 3.45, and the proof of Theorem 3.40, this is a Las Vegas algorithm with time complexity as stated. $\square$

3.2.5. Tensor decomposition. Now assume that $G \leqslant \mathrm{GL}(d, q)$ where $G \cong \mathrm{Ree}(q)$, $d > 7$ and $q = 3^{2m+1}$ for some $m > 0$. Then $\mathrm{Aut}\, \mathbb{F}_q = \langle \psi \rangle$, where ψ is the Frobenius automorphism. Let W be the given module of G and let V be the natural module of $\mathrm{Ree}(q)$, so that $\dim W = d$ and $\dim V = 7$. From Section 1.2.7 and Section 2.2.3 we know that

$$W \cong M^{\psi^{i_0}} \otimes M^{\psi^{i_1}} \otimes \cdots \otimes M^{\psi^{i_{n-1}}} \tag{3.45}$$

for some integers $0 \leqslant i_0 < i_1 < \cdots < i_{n-1} \leqslant 2m$, and where M is either V or the absolutely irreducible 27-dimensional submodule S of the symmetric square $\mathcal{S}^2(V)$. In fact, we may assume that $i_0 = 0$. As described in Section 1.2.7, we now want to tensor decompose W to obtain an effective isomorphism from W to V or to S. In the latter case we also have to decompose S into V to obtain an isomorphism between W and V. We consider this problem in Section 3.2.6.

PROPOSITION 3.48. *Let* $G \leqslant \mathrm{GL}(27, q)$ *such that* $G \cong \mathrm{Ree}(q)$*, let* $j \in G$ *be an involution and let* $H = \mathrm{C}_G(j)' \cong \mathrm{PSL}(2, q)$*. Then* $S|_H \cong V_6 \oplus V_9 \oplus V_{12}$ *as an* H*-module, where* $\dim V_i = i$*. Moreover,* V_9 *is absolutely irreducible,* $V_{12} \cong V_4.V_4^{\psi^l}.V_4$ *and* $V_6 \cong 1.V_4^{\psi^k}.1$*, where* $k \neq l$*.*

PROOF. By Proposition 2.18, $V|_H = V_3 \oplus V_4$ and hence $V_3 = \mathcal{S}^2(V_2)$ and $V_4 = V_2 \otimes V_2^{\psi^n}$ where V_2 is the natural module of $\mathrm{PSL}(2, q)$ and $n > 0$.

$$1 \oplus S|_H = \mathcal{S}^2(V|_H) = \mathcal{S}^2(\mathcal{S}^2(V_2)) \oplus \mathcal{S}^2(V_2 \otimes V_2^{\psi^n}) \oplus (\mathcal{S}^2(V_2) \otimes V_2 \otimes V_2^{\psi^n}) \quad (3.46)$$

Now consider

$$\begin{aligned} &\mathcal{S}^2(V_2 \otimes V_2^{\psi^n}) \oplus \wedge^2(V_2 \otimes V_2^{\psi^n}) = (V_2 \otimes V_2^{\psi^n}) \otimes (V_2 \otimes V_2^{\psi^n}) = \\ &= (V_2 \otimes V_2) \otimes (V_2^{\psi^n} \otimes V_2^{\psi^n}) = (\mathcal{S}^2(V_2) \oplus 1) \otimes (\mathcal{S}^2(V_2^{\psi^n}) \oplus 1) = \\ &= 1 \oplus \mathcal{S}^2(V_2) \oplus \mathcal{S}^2(V_2^{\psi^n}) \oplus (\mathcal{S}^2(V_2) \otimes \mathcal{S}^2(V_2^{\psi^n})) \end{aligned} \quad (3.47)$$

The final summand has dimension 9 and is absolutely irreducible because it is of the form (3.45). Hence $\wedge^2(V_2 \otimes V_2^{\psi^n}) = \mathcal{S}^2(V_2) \oplus \mathcal{S}^2(V_2^{\psi^n})$ and $\mathcal{S}^2(V_2 \otimes V_2^{\psi^n}) = 1 \oplus V_9$ as required.

Furthermore, a direct calculation shows that $\mathcal{S}^2(\mathcal{S}^2(V_2))$ has shape $1.1.1 \oplus \mathcal{S}^2(V_2)$ when restricted to $\mathbb{F}_3$, and over $\mathbb{F}_q$ the $\mathcal{S}^2(V_2)$ must fuse with the middle composition factor, otherwise the module would not be self-dual (S is self-dual since G preserves a bilinear form).

Similarly, $\mathcal{S}^2(V_2) \otimes V_2 \otimes V_2^{\psi^n}$ has shape $1.1.1 \oplus (\mathcal{S}^2(V_2).\mathcal{S}^2(V_2^{\psi^n}).\mathcal{S}^2(V_2))$ over $\mathbb{F}_3$. Over $\mathbb{F}_q$ each 1 fuses with a corresponding $\mathcal{S}^2(V_2)$ and we obtain the structure of V_{12}. This is also proves that the 4-dimensional factor of V_6 is not isomorphic to any of the factors of V_{12}. □

COROLLARY 3.49. *Let* $G \leqslant \mathrm{GL}(27, q)$ *such that* $G \cong \mathrm{Ree}(q)$*, let* $j \in G$ *be an involution and let* $H = \mathrm{C}_G(j)' \cong \mathrm{PSL}(2, q)$*. Then* $\dim \mathrm{Hom}_H(S|_H, S|_H) = 5$.

PROOF. By Proposition 3.48,

$$\mathrm{Hom}_H(S|_H, S|_H) = \mathrm{Hom}_H(V_6, S|_H) \oplus \mathrm{Hom}_H(V_9, S|_H) \oplus \mathrm{Hom}_H(V_{12}, S|_H).$$

The middle summand has dimension 1, by Schur's Lemma.

A homomorphism $V_6 \to S|_H$ must map the 5-dimensional submodule to itself or to 0. The composition factor of dimension 1 at the top can either be mapped to itself, or to the factor at the bottom. Hence $\dim \mathrm{Hom}_H(V_6, S|_H) = 2$.

Similarly, $\dim \mathrm{Hom}_H(V_{12}, S|_H) = 2$ since the top factor can either be mapped to itself, or to the bottom factor. Thus the result follows. □

Given a module W of the form (3.45), we now consider the problem of finding a flat. For $k = 0, \dots, n-1$, let H_k be the image of the representation corresponding to $M^{\psi^{i_k}}$, so $H_k \leqslant \mathrm{GL}(7, q)$ or $H_k \leqslant \mathrm{GL}(27, q)$, and let $\rho_k : G \to H_k$ be an isomorphism. Our goal is then to find ρ_k effectively for some k.

For $\lambda \in \mathbb{F}_q^\times$ denote $E_\lambda = \{1, \lambda^{\pm t}, \lambda^{\pm(t-1)}, \lambda^{\pm(2t-1)}\}$. We need the following conjectures.

CONJECTURE 3.50. *Let* $\mathrm{Ree}(q) \cong G \leqslant \mathrm{GL}(d, q)$ *have module* W *of the form* (3.45)*, with* $\dim W = d = 7^n$ *for some* $n > 1$.

Let $g \in G$ have order $q-1$ and let E be its multiset of eigenvalues. If $2m > n$ then there exists $\lambda \in \mathbb{F}_q^{\times}$ such that $E_\lambda \subset E$, and the sum of the eigenspaces of g corresponding to E_λ has dimension $\dim V$.

CONJECTURE 3.51. *Let $\mathrm{Ree}(q) \cong G \leqslant \mathrm{GL}(d,q)$ have module W of the form (3.45) and $\dim W = d > 7$. Let $j \in G$ be an involution.*

If W has tensor factors both of dimension 7 and 27, then $W|_{\mathrm{C}_G(j)}$ has unique submodules W_3 and W_4 of dimensions 3 and 4, respectively, such that $W_3 + W_4$ is a point of W of dimension 7.

CONJECTURE 3.52. *Let $\mathrm{Ree}(q) \cong G \leqslant \mathrm{GL}(d,q)$ have module W of the form (3.45) and $\dim W = 27^n$ for some $n > 1$. Let $j \in G$ be an involution and let $H = \mathrm{C}_G(j)'$.*

If $2m > n$ then $\dim \mathrm{Hom}_H(S^{\psi^{i_k}}|_H, W|_H) = 5$, for some $0 \leqslant k \leqslant n-1$.

THEOREM 3.53. *Assume Conjecture 3.50. There exists a Las Vegas algorithm that, given $\langle X \rangle \leqslant \mathrm{GL}(d,q)$, where $q = 3^{2m+1}$, $d = 7^n$, $n > 1$, $2m > n$ and $\langle X \rangle \cong \mathrm{Ree}(q)$, with module W of the form (3.45), finds a point of W. The algorithm has expected time complexity*

$$\mathrm{O}\big((\xi(d) + d^3 \log(q) \log\log(q^d)) \log\log(q) + d^4\big)$$

field operations.

PROOF. Let $G = \langle X \rangle$. By Corollary 2.24, we can easily find $g \in G$ such that $|g| \mid q-1$. Our approach is to construct a point as a suitable sum of eigenspaces of g. We know that for $k = 0, \dots, n-1$, $\rho_k(g)$ has 7 eigenvalues $\lambda_k^{\pm t}$, $\lambda_k^{\pm(t-1)}$, $\lambda_k^{\pm(2t-1)}$ and 1 for some $\lambda_k \in \mathbb{F}_q^{\times}$. Let E be the multiset of eigenvalues of g. Each eigenvalue has the form

$$\lambda_0^{j_0} \lambda_1^{j_1} \cdots \lambda_{n-1}^{j_{n-1}} \tag{3.48}$$

where each $\lambda_k \in \mathbb{F}_q^{\times}$ and each $j_k \in \{\pm t, \pm(t-1), \pm(2t-1), 1\}$. We can easily compute E.

Because each $\lambda_k^{j_k}$ may be 1, for each $k = 0, \dots, n-1$ we have $E_{\lambda_k} \subset E$. We can determine which $\lambda \in E$ can be one of the λ_k, since if $\lambda = \lambda_k$ for some k, then $E_{\lambda^{3t}} \subset E$.

Thus we can obtain a list, with length between n and d, of subsets E_λ of E. Now Conjecture 3.50 asserts that there is some $\mu \in \mathbb{F}_q^{\times}$ such that $E_\mu \subset E$, and such that the sum of the eigenspaces corresponding to E_μ has dimension 7, and by its construction it must therefore be a point of W. Since $\mu^t \in E$, the set E_μ will be on our list, and we can easily find the point.

The algorithm is Las Vegas, since we can easily calculate the dimensions of the subspaces. The expected number of random selections for finding g is $\mathrm{O}\big(\log\log(q)\big)$, and we can find its order using expected $\mathrm{O}\big(d^3 \log(q) \log\log(q^d)\big)$ field operations. We find the characteristic polynomial using $\mathrm{O}\big(d^3\big)$ field operations and then find the eigenvalues using expected $\mathrm{O}\big(d(\log d)^2 \log\log(d) \log(dq)\big)$ field operations. Finally we use the Algorithm from Section 1.2.10.3 to verify that we have a point, using $\mathrm{O}\big(d^3 \log(q)\big)$ field operations. The rest of the algorithm is linear algebra, and hence the expected time complexity is as stated. □

THEOREM 3.54. *Assume Conjecture* 3.51. *There exists a Las Vegas algorithm that, given* $\langle X \rangle \leqslant \mathrm{GL}(d, q)$*, where* $q = 3^{2m+1}$, $7 \mid d$, $27 \mid d$ *and* $\langle X \rangle \cong \mathrm{Ree}(q)$*, with module* W *of the form* (3.45)*, finds a point of* W*. The algorithm has expected time complexity*

$$\mathrm{O}\big(\xi(d) + d^3 \log(q)(\log\log(q^d) + \log(d)^2)\big)$$

field operations.

PROOF. Let $G = \langle X \rangle$. Similarly as in Corollary 3.36, we find an involution $j \in G$ and probable generators for $\mathrm{C}_G(j)'$. We can then use Theorem 1.13 to verify that we have got the whole centraliser.

Using the MeatAxe, we find the composition factors of $W|_{\mathrm{C}_G(j)}$. For each pair of factors of dimensions 3 and 4 we compute module homomorphisms into $W|_{\mathrm{C}_G(j)}$ and then find the sum of their images.

Conjecture 3.51 asserts that this will produce a point of W. We can easily calculate the dimensions of the submodules and use the tensor decomposition algorithm to verify that we do obtain a point, so the algorithm is Las Vegas.

The expected time complexity for finding j is $\mathrm{O}\big(\xi(d) + d^3 \log(q) \log\log(q^d)\big)$ field operations. From the proof of Corollary 3.36 we see that we can find probable generators for $\mathrm{C}_G(j)$ and verify that we have the whole centraliser using expected $\mathrm{O}\big(\xi(d) + d^3 \log(q) \log\log(q^d)\big)$ field operations, if we let $\varepsilon = \log\log(q)$ in Theorem 1.13. The MeatAxe uses expected $\mathrm{O}\big(d^3\big)$ field operations in this case, since the number of generators for the centraliser is constant. Then we consider $\mathrm{O}\big(\log(d)^2\big)$ pairs of submodules, and for each one we use the tensor decomposition algorithm to determine if we have a point, using $\mathrm{O}\big(d^3 \log(q)\big)$ field operations. Hence the expected time complexity is as stated. □

THEOREM 3.55. *Assume Conjecture* 3.52 *and an oracle for the discrete logarithm problem. There exists a Las Vegas algorithm that, given* $\langle X \rangle \leqslant \mathrm{GL}(d, q)$*, where* $q = 3^{2m+1}$, $d = 27^n$, $n > 1$, $2m > n$ *and* $\langle X \rangle \cong \mathrm{Ree}(q)$*, with module* W *of the form* (3.45)*, finds a point of* W*. The algorithm has expected time complexity*

$$\mathrm{O}\big((\xi(d) + d^3 \log(q) \log\log(q^d)) \log\log(q) + \log(q)^3 + d^5 \sigma_0(d) \left|X\right| + d\chi_D(q) + \xi(d) d\big)$$

field operations.

PROOF. Let $G = \langle X \rangle$. Similarly as in Corollary 3.36, we find an involution $j \in G$ and probable generators for $H = \mathrm{C}_G(j)' \cong \mathrm{PSL}(2, q)$. We can then use Theorem 1.13 to verify that we have got the whole centraliser.

Using the MeatAxe, we find the composition factors of $W|_H$. Let S_1 be the group corresponding to a non-trivial composition factor. Using Theorem 1.12 we constructively recognise S_1 as $\mathrm{PSL}(2, q)$ and obtain an effective isomorphism $\pi_1 : \mathrm{PSL}(2, q) \to H$.

Now let R be the image of the representation corresponding to S, so $R \leqslant \mathrm{GL}(27, q)$. Again we find an involution $j' \in R$ and probable generators for $K = \mathrm{C}_R(j')' \cong \mathrm{PSL}(2, q)$. As above, we chop the module $S|_K$ with the MeatAxe, constructively recognise one of its non-trivial factors and obtain an effective isomorphism $\pi_2 : \mathrm{PSL}(2, q) \to K$.

Note that both π_1 and π_2 have effective inverses. Hence we can obtain standard generators for H and K. For each $i = 0, \dots, 2m$, do the following:

(1) Find $M = \mathrm{Hom}_{\mathrm{PSL}(2,q)}(S^{\psi^i}|_K, W|_H)$ using the standard generators.
(2) If $\dim M = 5$, then find random $f \in M$ such that $\dim \mathrm{Ker} f = 0$. Use the algorithm in Section 1.2.10.3 to determine if $U = \mathrm{Im}(f)$ is a point.

From Proposition 3.48 we know that $S|_K$ has a submodule of dimension 1. This implies that that $W|_H$ has a submodule $U'_k = \langle v_1 \rangle \otimes \cdots \otimes \langle v_{k-1} \rangle \otimes S^{\psi^{i_k}} \otimes \langle v_{k+1} \rangle \otimes \cdots \otimes \langle v_{n-1} \rangle \cong S^{\psi^{i_k}}$, for any $k = 0, \dots, n-1$, and some $v_1, \dots, v_{n-1}$ (depending on k). Moreover, $\mathrm{Hom}_{\mathrm{PSL}(2,q)}(S^{\psi^{i_k}}|_K, W|_H) \geqslant \mathrm{Hom}_{\mathrm{PSL}(2,q)}(S^{\psi^{i_k}}|_K, U'_k)$, and by Corollary 3.49, the latter has dimension 5.

But by Conjecture 3.52, for some i the former also has dimension 5, and hence these vector spaces are equal. Therefore, for some $i = i_k$, the subspace U found in the algorithm must be equal to U'_k, and hence it is a point.

The expected time complexity for finding the involutions is

$$\mathrm{O}\big(\xi(d) + d^3 \log(q) \log\log(q^d)\big)$$

field operations. From the proof of Corollary 3.36 we see that we can find probable generators for $\mathrm{C}_G(j)$ and verify that we have the whole centraliser using expected $\mathrm{O}\big(\xi(d) + d^3 \log(q) \log\log(q^d)\big)$ field operations, if we let $\varepsilon = \log\log(q)$ in Theorem 1.13. The MeatAxe uses expected $\mathrm{O}\big(d^3\big)$ field operations in this case since the number of generators are constant. In the loop, we require $\mathrm{O}\big(d^3 \log(q)\big)$ field operations to verify that U is a point. Hence the expected time complexity follows from Theorem 1.12. □

Conjectures 3.50 and 3.52 do not apply when $2m \leqslant n$, so in this case we need another algorithm. Then $q \in \mathrm{O}\big(d\big)$ so we are content with an algorithm that has time complexity polynomial in q. The approach is not to use tensor decomposition, since in this case we have no efficient method of finding a flat. Instead we find standard generators of G using permutation group techniques, then enumerate all tensor products of the form (3.45), and for each one we determine if it is isomorphic to W.

Lemma 3.56. *There exists a Las Vegas algorithm that, given $\langle X \rangle \leqslant \mathrm{GL}(d, q)$ such that $q = 3^{2m+1}$ with $m > 0$ and $\langle X \rangle \cong \mathrm{Ree}(q)$, finds an effective injective homomorphism $\Pi : \langle X \rangle \to \mathrm{Sym}(O)$ where $|O| = q^3 + 1$. The algorithm has expected time complexity $\mathrm{O}\big(q^3(\xi(d) + |X|\, d^2 + d^3) + d^4\big)$ field operations.*

Proof. By Proposition 2.16, $\mathrm{Ree}(q)$ acts doubly transitively on a set of size $q^3 + 1$. Hence $G = \langle X \rangle$ also acts doubly transitively on O, where $|O| = q^3 + 1$, and we can find the permutation representation of G if we can find a point $P \in O$. The set O is a set of projective points of $\mathbb{F}_q^d$, and the algorithm proceeds as follows.

(1) Choose random $g \in G$. Repeat until $|g| \mid q-1$.
(2) Choose random $x \in G$ and let $h = g^x$. Repeat until $[g,h]^9 = 1$ and $[g,h] \neq 1$.
(3) Find a composition series for the module M of $\langle g, h\rangle$ and let $P \subseteq M$ be the submodule of dimension 1 in the series.
(4) Find the orbit $O = P^G$ and compute the permutation group $S \leqslant \mathrm{Sym}(O)$ of G on O, together with an effective isomorphism $\Pi : G \to S$.

By Proposition 2.21, elements in G of order dividing $q-1$ fix two points of O, and hence $\langle g, h\rangle \leqslant G_P$ for some $P \in O$ if and only if g and h have a common fixed point. All composition factors of M have dimension 1, so a composition series of M must contain a submodule P of dimension 1. This submodule is a fixed point for $\langle g, h\rangle$, and its orbit must have size q^3+1, since $|G| = q^3(q^3+1)(q-1)$ and $|G_P| = q^3(q-1)$. It follows that $P \in O$.

All elements of G of order a power of 3 lie in the derived group of a stabiliser of some point, which is also a Sylow 3-subgroup of G, and the exponent of this subgroup is 9. Hence $[g,h]^9 = 1$ if and only if $\langle g, h\rangle$ lie in a stabiliser of some point, if and only if g and h have a common fixed point.

To find the orbit $O = P^G$ we can compute a Schreier tree on the generators in X with P as root, using $\mathrm{O}\big(|X|\,|O|\,d^2\big)$ field operations. Then $\Pi(g)$ can be computed for any $g \in \langle X\rangle$ using $\mathrm{O}\big(|O|\,d^2\big)$ field operations, by computing the permutation on O induced by g. Hence Π is effective, and its image S is found by computing the image of each element of X. Therefore the algorithm is correct and it is clearly Las Vegas.

We find g using expected $\mathrm{O}\big((\xi(d) + d^3 \log(q)\log\log(q^d))\log\log(q)\big)$ field operations and we find h using expected $\mathrm{O}\big((\xi(d) + d^3)q^2\big)$ field operations. Then P is found using the MeatAxe, in expected $\mathrm{O}\big(d^4\big)$ field operations. Thus the result follows. $\square$

Conjecture 3.57. *Let $G = \langle X\rangle \leqslant \mathrm{Sym}(\mathcal{O})$ such that $G \cong \mathrm{Ree}(q) = H$. There exists a Las Vegas algorithm that finds $x, h, z \in G$ as* `SLP`*s in X such that the map*

$$x \mapsto S(1,0,0) \tag{3.49}$$

$$h \mapsto h(\lambda) \tag{3.50}$$

$$z \mapsto \Upsilon \tag{3.51}$$

is an isomorphism. Its time complexity is $\mathrm{O}\big(q^5(\log(q))^4\big)$ field operations. The length of the returned `SLP`*s are $\mathrm{O}\big(q^3 \log\log(q)\big)$.*

Remark 3.58. There exists an implementation of the above mentioned algorithm, and the Conjecture is then it always produces a correct result and has the stated complexity.

THEOREM 3.59. *Assume Conjecture* 3.57. *There exists a Las Vegas algorithm that, given* $\langle X \rangle \leqslant \mathrm{GL}(d, q)$, *where* $q = 3^{2m+1}$, $n > 1$ *and* $d = 7^n$ *or* $d = 27^n$ *and* $\langle X \rangle \cong \mathrm{Ree}(q)$, *with module* W *of the form* (3.45), *finds a tensor decomposition of* W. *The algorithm has time complexity* $\mathrm{O}\big(q^3(\xi(d) + |X|\, d^2 + d^3 \log\log(q) + q^2(\log(q))^4) + d^3(|X| \binom{2m}{n-1} + d)\big)$ *field operations.*

PROOF. Let $G = \langle X \rangle$. The algorithm proceeds as follows:

(1) Find permutation representation $\pi : G \to G_S \leqslant \mathrm{Sym}(q^3+1)$ using Lemma 3.56.
(2) Find standard generators $x, h, z \in G$ using Conjecture 3.57. Evaluate them on G to obtain a generating set Y.
(3) Let $H = \langle Y \rangle$ and let V be the module of H. If $3 \mid d$ then replace V with S.
(4) Construct each module of dimension d of the form (3.45) using V as base. For each one test if it is isomorphic to W, using the MeatAxe.
(5) Return the change of basis from the successful isomorphism test.

The returned change of basis exhibits W as a tensor product, so by Lemma 3.56 the algorithm is Las Vegas.

The lengths of the SLPs of x, h, z is $\mathrm{O}\big(q^3 \log\log(q)\big)$, so we need $\mathrm{O}\big(d^3 q^3 \log\log(q)\big)$ field operations to obtain Y. The number of modules of dimension d of the form (3.45) using V as base is $\binom{2m}{n-1}$. Module isomorphism testing uses $\mathrm{O}\big(|X|\, d^3\big)$ field operations. Hence by Conjecture 3.57 and Lemma 3.56 the time complexity of the algorithm is as stated. □

3.2.6. Symmetric square decomposition. The two basic irreducible modules of $\mathrm{Ree}(q)$ are the natural module V of dimension 7, and an irreducible submodule S of the symmetric square $\mathcal{S}^2(V)$. The symmetric square itself is not irreducible, since $\mathrm{Ree}(q)$ preserves a quadratic form, and $\mathcal{S}^2(V)$ therefore has a submodule of dimension 1. The complement of this has dimension 27 and is the irreducible module S.

CONJECTURE 3.60. *The exterior square of* S *has a submodule isomorphic to a twisted version of* V.

THEOREM 3.61. *Assume Conjecture* 3.60. *There exists a Las Vegas algorithm that, given* $\langle X \rangle \leqslant \mathrm{GL}(27, q)$ *with module* W *such that* W *is isomorphic to a twisted version of* S, *finds an effective isomorphism from* $\langle X \rangle$ *to* $\mathrm{Ree}(q)^g$ *for some* $g \in \mathrm{GL}(7, q)$. *The algorithm has expected time complexity* $\mathrm{O}\big(|X|\big)$ *field operations.*

PROOF. Using Conjecture 3.60, this is just an application of the MeatAxe. We construct the exterior square $\wedge^2(W)$ of W, which has dimension 351, and find a composition series of this module using the MeatAxe. By the Conjecture, the natural module of dimension 7 will be one of the composition factors and the MeatAxe will provide an effective isomorphism to this factor, in the form of a change of basis $A \in GL(27, q)$ of W that exhibits the action on the composition factors.

This induces an isomorphism $\varphi : \langle X \rangle \to H$, where H is conjugate to $\mathrm{Ree}(q)$. For $g \in \langle X \rangle$, $\varphi(g)$ is computed by taking a submatrix of g^A of degree 7. Clearly φ can be computed using $\mathrm{O}\big(1\big)$ field operations.

Since the MeatAxe is Las Vegas and has expected time complexity $\mathrm{O}\big(|X|\big)$, the result follows. □

3.2.7. Constructive recognition. Finally, we can now state and prove our main theorem.

THEOREM 3.62. *Assume the small Ree Conjectures, and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$. *There exists a Las Vegas algorithm that, given* $\langle X \rangle \leqslant \mathrm{GL}(d, q)$ *satisfying the assumptions in Section* 1.2.7, *with* $q = 3^{2m+1}$, $m > 0$ *and* $\langle X \rangle \cong \mathrm{Ree}(q)$, *finds an effective isomorphism* $\varphi : \langle X \rangle \to \mathrm{Ree}(q)$ *and performs preprocessing for constructive membership testing. The algorithm has expected time complexity* $\mathrm{O}\big(\xi(d)(d^3 + (\log\log(q))^2) + d^6 \log\log(d) + d^5 \sigma_0(d)\,|X| + d^3 \log(q) \log\log(q) \log\log(q^d) + \log(q)^3 \log\log(q) + \chi_D(q)(d + \log\log(q))\big)$ *field operations.*

Each image of φ *can be computed in* $\mathrm{O}\big(d^3\big)$ *field operations, and each pre-image in expected* $\mathrm{O}\big(\xi(d) + \log(q)^3 + d^3(\log(q)\log\log(q))^2\big)$ *field operations.*

PROOF. Let W be the module of $G = \langle X \rangle$. The algorithm proceeds as follows:

(1) If $d = 7$ then use Theorem 3.47 to obtain $y \in \mathrm{GL}(7, q)$ such that $G^y = \mathrm{Ree}(q)$, and hence an effective isomorphism $\varphi : G \to \mathrm{Ree}(q)$ defined by $g \mapsto g^y$.
(2) If $3 \nmid d$ then $d = 7^n$ for some $n > 1$. If $2m > n$ then use Theorem 3.53 to find a flat $L \leqslant M$. If $3 \mid d$ but d is not a proper power of 27 then use Theorem 3.54 to find such an L. Otherwise $d = 27^n$ for some $n > 1$. If $2m > n$, then use Theorem 3.55 to find a flat $L \leqslant M$.
(3) Use the tensor decomposition algorithm described in Section 1.2.10.3 with L, to obtain $x \in \mathrm{GL}(d, q)$ such the change of basis determined by x exhibits W as a tensor product $A \otimes B$, with $\dim A = 7$ or $\dim A = 27$. If $d = 7^n$ or $d = 27^n$ and $2m \leqslant n$ then use Theorem 3.59 to find x. Let G_A and G_B be the images of the corresponding representations.
(4) Define $\rho_A : G_{A \otimes B} \to G_A$ as $g_a \otimes g_b \mapsto g_a$ and let $Y = \{\rho_A(g^x) \mid g \in X\}$. If $\dim A = 27$ then let θ be the effective isomorphism from Theorem 3.61, otherwise let θ be the identity map.
(5) Let $Z = \{\theta(x) \mid x \in Y\}$. Then $\langle Z \rangle$ is conjugate to $\mathrm{Ree}(q)$. Use Theorem 3.47 to obtain $y \in \mathrm{GL}(7, q)$ such that $\langle Z \rangle^y = \mathrm{Ree}(q)$.
(6) An effective isomorphism $\varphi : G \to \mathrm{Ree}(q)$ is given by $g \mapsto \theta(\rho_A(g^x))^y$.

The map ρ_A is straightforward to compute, since given $g \in \mathrm{GL}(d, q)$ it only involves dividing g into submatrices of degree $d/7$ or $d/27$, checking that they are scalar multiples of each other and returning the 7×7 or 27×27 matrix consisting of these scalars. Since x might not lie in G, but only in $\mathrm{N}_{\mathrm{GL}(d,q)}(G) \cong G{:}\mathbb{F}_q$, the result of ρ_A might not have determinant 1. However, since every element of $\mathbb{F}_q$ has a unique 7th root, we can easily scale the matrix to have determinant 1. Hence by Theorem 3.53, Theorem 3.54, Theorem 3.55, Section 1.2.10.3, Theorem 3.61 and Theorem 3.47, the algorithm is Las Vegas, and φ can be computed using $\mathrm{O}\big(d^3\big)$ field operations.

In the case where we use Theorem 3.59 we have $2m \leqslant n$ and hence $q < 3d$. We see that $\binom{2m}{n-1} \leq n$, and the time complexity of the algorithm to find x, in Theorem 3.59, simplifies to $\mathrm{O}\big(d^3(\xi(d) + |X|\, d^2 + d^3 \log\log(d))\big)$.

In the other cases, finding L uses $\mathrm{O}\big((\xi(d) + d^3 \log(q) \log\log(q^d)) \log\log(q) + d^5\sigma_0(d)\, |X| + d\chi_D(q) + \xi(d)d\big)$ field operations. From Section 1.2.10.3, finding x uses $\mathrm{O}\big(d^3 \log(q)\big)$ field operations when a flat L is given.

From Theorem 3.61 finding θ uses $\mathrm{O}\big(|Y|\big)$ field operations, and from Theorem 3.47, finding y uses $\mathrm{O}\big((\xi(d) \log\log(q) + \log(q)^3 + \chi_D(q)) \log\log(q) + |Z|\big)$ field operations. Hence the expected time complexity is as stated. Finally, $\varphi^{-1}(g)$ is computed by first using Algorithm 3.6 to obtain an `SLP` of g and then evaluating it on X. The necessary precomputations in Theorem 3.40 have already been made during the application of Theorem 3.47, and hence it follows from Theorem 3.42 that the time complexity for computing the pre-image of g is as stated. □

3.3. Big Ree groups

Here we will use the notation from Section 2.3. We will refer to Conjectures 2.37, 2.38, 2.39, 2.40, 2.41, 2.42, 2.46, 2.47 and 3.64 simultaneously as the *Big Ree Conjectures.*

With the Big Ree groups, we will only deal with the natural representation, the last case in Section 1.2.7. We will not attempt any tensor decomposition, or decomposition of the tensor indecomposables listed in Section 2.3.2. This can be partly justified by the fact that at the present time the only representation, other than the one of dimension 26, that it is within practical limits in the MGRP, is the one of dimension 246. We have some evidence that it is feasible to decompose this representation into the natural representation using the technique known as *condensation*, see [**HEO05**, Section 7.4.5] and [**LW98**].

The main constructive recognition theorem is Theorem 3.79.

3.3.1. Recognition. We now consider the question of non-constructive recognition of ${}^2\mathrm{F}_4(q)$, so we want to find an algorithm that, given a set $\langle X\rangle \leqslant \mathrm{GL}(26, q)$, decides whether or not $\langle X\rangle = {}^2\mathrm{F}_4(q)$.

THEOREM 3.63. *There exists a Las Vegas algorithm that, given $\langle X\rangle \leqslant \mathrm{GL}(26, q)$, decides whether or not $\langle X\rangle = {}^2\mathrm{F}_4(q)$. The algorithm has expected time complexity $\mathrm{O}\big(\sigma_0(\log(q))(|X| + \log(q))\big)$ field operations.*

PROOF. Let $G = {}^2\mathrm{F}_4(q)$, with natural module M. The algorithm proceeds as follows:

(1) Determine if $X \subseteq G$ and return `false` if not. All the following steps must succeed in order to conclude that a given $g \in X$ also lies in G.
  (a) Determine if $g \in \mathrm{O}^-(26, q)$, which is true if $\det g = 1$ and if $gQg^T = Q$, where Q is the matrix corresponding to the quadratic form Q^* and where g^T denotes the transpose of g.
  (b) Determine if $g \in \mathrm{F}_4(q)$, which is true if g preserves the exceptional Jordan algebra multiplication. This is easy using the multiplication table given in [**Wil06**].
  (c) Determine if g is a fixed point of the automorphism of $\mathrm{F}_4(q)$ which defines ${}^2\mathrm{F}_4(q)$. By [**Wil06**], computing the automorphism amounts to taking a submatrix of the exterior square of g and then replacing each matrix entry x by x^{2^m}.

(2) If $\langle X \rangle$ is not a proper subgroup of G, or equivalently if $\langle X \rangle$ is not contained in a maximal subgroup, return `true`. Otherwise return `false`. By Proposition 2.49, it is sufficient to determine if $\langle X \rangle$ cannot be written over a smaller field and if $\langle X \rangle$ is irreducible. This can be done using the Las Vegas algorithms from Sections 1.2.10.1 and 1.2.10.2.

Since the matrix degree is constant, the complexity of the first step of the algorithm is $\mathrm{O}(1)$ field operations. For the same reason, the complexity of the algorithms from Sections 1.2.10.1 and 1.2.10.2 is $\mathrm{O}\big(\sigma_0(\log(q))(|X| + \log(q))\big)$ field operations. Hence the expected time complexity is as stated. □

3.3.2. Finding elements of even order. In constructive recognition and membership testing of ${}^2\mathrm{F}_4(q)$, the essential problem is to find elements of even order, as SLPs in the given generators. Let $G = {}^2\mathrm{F}_4(q) = \langle X \rangle$. We begin with an overview of the method. The matrix degree is constant here, so we set $\xi = \xi(26)$.

Choose random $a \in G$ of order $q - 1$, by choosing a random element of order $(q - 1)(q + t + 1)$ and powering up. By Proposition 2.33 it is easy to find such elements, and by Proposition 2.32, we can diagonalise a and obtain $c \in \mathrm{GL}(26, q)$ such that $a^c = \delta = h(\lambda, \mu)$ for some $\lambda, \mu \in \mathbb{F}_q^\times$.

Now choose random $b \in G$. Let $B = b^c$ and let $A(u, v)$ be a diagonal matrix of the same form as $h(\lambda, \mu)$, where λ and μ are replaced by indeterminates u and v, so $A(u, v)$ is a matrix over the function field $\mathbb{F}_q(u, v)$.

For any $r, s \in \mathbb{F}_q^\times$, such that $r^t = s$, the matrix $(A(r, s)B)^{c^{-1}} \in \langle a \rangle\, b$. Hence by Conjecture 2.46 if we can find r, s such that $r^t = s$ and $A(r, s)B$ has the eigenvalue 1 with multiplicity 6, then with high probability $A(r, s)B$ will have even order.

Proposition 3.64. *Assume Conjecture* 2.48. *For every $b \in G \setminus \mathrm{N}_G(\langle a \rangle)$ the* 6 *lowest coefficients $f_1, \dots, f_6 \in \mathbb{F}_q[u, v]$ of the characteristic polynomial of $A(u, v)B - I_{26}$ generate a zero-dimensional ideal.*

Proof. Since b does not normalise $\langle a \rangle$, by Conjecture 2.48, there must be a bounded number of solutions. Hence the system must be zero-dimensional. □

Finally we can solve the discrete logarithm problem and find an integer k such that $\delta^k = A(r, s)$. Then $\delta^k B$ has even order, and therefore also $a^k b$ has even order. Since a and b are random, we obtain an element of even order as an `SLP` in X. The algorithm for finding elements of even order is given formally as Algorithm 3.7.

LEMMA 3.65. *Assume Conjecture* 2.48. *There exists a Las Vegas algorithm that, given the matrices* $A(u, v)$ *and* B*, finds* $r, s \in \mathbb{F}_q^\times$ *such that* $r^t = s$ *and* $A(r, s)B$ *has* 1 *as an eigenvalue of multiplicity at least* 6. *The algorithm has expected time complexity* $\mathrm{O}\big(\log q\big)$ *field operations.*

PROOF. If we can find the characteristic polynomial $f(x) \in \mathbb{F}_q(u, v)[x]$ of $A(u, v)B$, then the condition we want to impose is that 1 should be a root of multiplicity 6, or equivalently that y^5 should divide $g(y) = f(y + 1)$.

Hence we obtain 6 polynomial equations in u and v of bounded degree. By Proposition 3.64, we can use Theorem 1.2 to find the possible values for r and s.

Thus it only remains to find $f(x)$, which has the form

$$f(x) = a_n x^n + \cdots + a_1 x + a_0 \tag{3.52}$$

where $a_i \in \mathbb{F}_q(u, v)$ and $0 \leqslant i \leqslant n \leqslant 26$. Recall that $A(u, v)$ is diagonal of the same form as $h(\lambda, \mu)$. This implies that in an echelon form of $A(u, v)B$, the diagonal has the form $A(u, v)D$ for some diagonal matrix $D \in \mathrm{GL}(26, q)$. We obtain $f(x)$ by multiplying these diagonal elements, and since the sum of the positive powers of u on the diagonal is 10, and the sum of the positive powers of v on the diagonal is 6, each a_i has the form

$$a_i = \sum_j c_{ij} u^{z_{ij}} v^{y_{ij}} \tag{3.53}$$

where each $c_{ij} \in \mathbb{F}_q$, $-10 \leqslant z_{ij} \leqslant 10$ and $-6 \leqslant w_{ij} \leqslant 6$.

Because of these bounds on the exponents z_{ij} and w_{ij}, we can find the coefficients c_{ij}, and hence the coefficients a_i and $f(x)$, using interpolation. Each c_{ij} is uniquely determined by at most $(2 \cdot 10 + 1)(2 \cdot 6 + 1) = 273$ values of u, v and the corresponding value of a_i.

Therefore, choose 273 random pairs $(e_k, f_k) \in \mathbb{F}_q \times \mathbb{F}_q$. For each pair, calculate the characteristic polynomial of $A(e_k, f_k)B$, thus obtaining the corresponding values of the coefficients a_i. Finally perform the interpolation by solving n linear systems with 273 equations and variables.

It is clear that the algorithm is Las Vegas, and the dominating term in the time complexity is the root finding of univariate polynomials of bounded degree over $\mathbb{F}_q$. $\square$

LEMMA 3.66. *Assume Conjectures* 2.47 *and* 2.48. *Let* $a \in G$ *be such that* $|a| = q - 1$ *and* a *is conjugate to some* $h(\lambda, \mu)$ *with* $\lambda^t = \mu \in \mathbb{F}_q^\times$. *The proportion of* $b \in G \setminus \mathrm{N}_G(\langle a \rangle)$*, such that* $\langle a \rangle\, b$ *contains an element with* 1 *as an eigenvalue of multiplicity* 6*, is bounded below by a constant* $c_4 > 0$.

PROOF. Given such a coset $\langle a \rangle b$, from the proof of Lemma 3.65 we see that our algorithm constructs a bounded number d_1 of candidates of elements of the required type in the coset.

Let c be the number of cosets containing an element of the required type. By Conjecture 2.47, the total number of elements of the required type is $c_3 \left|G\right|$. Hence c is minimised if all the c cosets contain d_1 such elements, in which case $cd_1 = c_3 \left|G\right|$. Thus $c \geqslant c_3 \left|G\right| / d_1$ and the proportion of cosets is $c(q-1)/\left|G\right| \geqslant c_3(q-1)/d_1$, which is bounded below by a constant $c_4 > 0$ since $c_3 \in \mathrm{O}\big(1/q\big)$. □

THEOREM 3.67. *Assume Conjectures* 2.48*,* 2.47 *and* 2.46*, and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$*. Algorithm* 3.7 *is a Las Vegas algorithm with expected time complexity* $\mathrm{O}\big((\xi + \log(q)\log\log(q) + \chi_D(q))\log\log(q)\big)$ *field operations. The length of the returned* `SLP` *is* $\mathrm{O}\big(\log\log(q)\big)$*.*

PROOF. By Proposition 2.32, a is conjugate to some $h(\lambda, \mu)$ and by Proposition 2.33, we can find h using expected $\mathrm{O}\big(\xi \log\log(q)\big)$ field operations. The test at line 15 is easy since b either centralises or inverts a. Furthermore, by Lemma 3.66, the test at line 17 will succeed with high probability and by Conjecture 2.46, the test at line 19 will succeed with high probability. The test at line 21 can only fail if $|a|$ is a proper divisor of $q-1$, which happens with low probability.

Hence by Lemma 3.65, the algorithm is Las Vegas and the time complexity is as stated. Clearly, the length of the `SLP` of the returned element is the same as the length of the `SLP` of h. □

Algorithm 3.7: FINDEVENORDERELEMENT(X)

1 **Input**: $\langle X \rangle \leqslant \mathrm{GL}(26, q)$ such that $\langle X \rangle \cong {}^2\mathrm{F}_4(q)$.
2 **Output**: An element of $\langle X \rangle$ of even order, expressed as an `SLP` in X.
3 // FINDELEMENTINCOSET is given by Lemma 3.65
4 **repeat**
5   **repeat**
6         $h :=$ RANDOM($\langle X \rangle$)
7   **until** $|h| \mid (q-1)(q+t+1)$
8     $a := h^{q+t+1}$
9     $\delta, c :=$ DIAGONALISE(a)
10   // Now $a^c = \delta = h(\lambda, \mu)$ where $\lambda, \mu \in \mathbb{F}_q^\times$ and $\mu = \lambda^t$
11   **repeat**
12   **repeat**
13   **repeat**
14                 $b :=$ RANDOM($\langle X \rangle$)
15   **until** $b \notin \mathrm{N}_G(\langle a \rangle)$
16             $(\mathit{flag}, r, s) :=$ FINDELEMENTINCOSET(b^c)
17   **until** *flag*
18   // Now $r^t = s$ and $A(r,s)b^c$ has 1 as a 6-fold eigenvalue
19   **until** $|A(r,s)b^c|$ is even
20     $k :=$ DISCRETELOG($\delta_{2,2}, r$)
21 **until** $k > 0$
22 **return** $a^k b$

REMARK 3.68. If we are given $g \in \langle X \rangle \cong {}^2\mathrm{F}_4(q)$, then a trivial modification of Algorithm 3.7 finds an element of $\langle X \rangle$, of even order, of the form hg for some $h \in \langle X \rangle$. If we also have an SLP of g in X, then we will obtain hg as SLP, otherwise we will only obtain an SLP for h.

PROPOSITION 3.69. *Assume Conjecture* 2.46. *With probability* $1 - \mathrm{O}\big(1/q\big)$*, the element returned by Algorithm* 3.7 *powers up to an involution of class* $2A$.

PROOF. Follows immediately from Conjecture 2.46. □

3.3.3. Constructive membership testing. The overall method we use for constructive membership testing in ${}^2\mathrm{F}_4(q)$ is the *Ryba algorithm* described in Section 1.2.9.2.

Since we know that there are only two conjugacy classes of involutions in ${}^2\mathrm{F}_4(q)$, and since we know the structure of their centralisers, we can improve upon the basic Ryba algorithm. When solving constructive membership testing in the centralisers, instead of applying the Ryba algorithm recursively, we can do it in a more direct way using Theorem 1.12 and the algorithms for constructive membership testing in the Suzuki group, described in Section 3.1. Involutions of class $2A$ can be found using Algorithm 3.7, and Conjecture 2.40 give us a method for finding involutions of class $2B$ using random search. The Ryba algorithm needs to find involutions of both classes, since it needs to find two involutions whose product has even order.

As a preprocessing step to Ryba, we can therefore find an involution of each class and compute their centralisers. In each call to Ryba we then conjugate the involutions that we find to one of these two involutions, which removes the necessity of computing involution centralisers at each call.

3.3.3.1. *The involution centralisers.* We use the Bray algorithm to find generating sets for the involution centralisers. This algorithm is described in Section 1.2.9.2.

The following results show how to precompute generators and how to solve the constructive membership problem for a centraliser of an involution of class $2A$, using our Suzuki group algorithms to constructively recognise $\mathrm{Sz}(q)$. Analogous results hold for the centraliser of an involution of class $2B$, using Theorem 1.12 to constructively recognise $\mathrm{SL}(2, q)$.

LEMMA 3.70. *Assume Conjecture* 2.39*, and use its notation. There exists a Las Vegas algorithm that, given* $\langle Y \rangle \leqslant G \leqslant \mathrm{GL}(26, q)$ *such that* $G \cong {}^2\mathrm{F}_4(q)$*,* $S \leqslant \langle Y \rangle \leqslant \mathrm{C}_G(j)$ *for an involution* $j \in G$ *of class* $2A$ *and* $S \cong \mathrm{Sz}(q)$*, finds a composition series for the natural module* M *of* $\langle Y \rangle$ *such that the composition factors are ordered as* $1, S_4, 1, S_4, 1, S_4^{\psi^t}, 1, S_4, 1, S_4, 1$*, and finds the corresponding filtration of* $\mathrm{O}_2(\langle Y \rangle)$*. The algorithm has time complexity* $\mathrm{O}\big(|Y|\big)$ *field operations.*

PROOF. By Proposition 2.39 the composition factors are as stated, and we just have to order them correctly.

(1) Find a composition factor of F_1 of M, such that $\dim F_1 = 1$, and find $H_1 = \mathrm{Hom}_{\langle Y \rangle}(M, F_1)$. By Conjecture 2.39, M has a unique 1-dimensional submodule, so $\dim H_1 = 1$.
(2) Let $M_{25} = \operatorname{Ker} \alpha_1$ where $\langle \alpha_1 \rangle = H_1$. Then $\dim M_{25} = 25$.
(3) Let $M_1 = M_{25}^{\perp} \cap M_{25}$ be the orthogonal complement under the bilinear form preserved by G, so that $\dim M_1 = 1$.
(4) Find a composition factor of F_4 of M_{25}, such that $\dim F_4 = 4$, and find $H_4 = \mathrm{Hom}_{\langle Y \rangle}(M_{25}, F_4)$. By Conjecture 2.39, M_{25} has a unique 4-dimensional submodule, so $\dim H_4 = 1$ for one of its 5 composition factors of dimension 4.
(5) Let $M_{21} = \operatorname{Ker} \alpha_4$ where $\langle \alpha_4 \rangle = H_4$. Then $\dim M_{21} = 21$.
(6) Let $M_5 = M_{21}^{\perp} \cap M_{21}$ be the orthogonal complement under the bilinear form preserved by G, so that $\dim M_5 = 5$.
(7) Now we have four proper submodules M_1, M_5, M_{21}, M_{25} in the composition series that we want to find, and we can obtain the other submodules by continuing in the same way inside M_{21}.

The filtration is determined by the composition factors, and immediately found. Clearly the time complexity is the same as the MeatAxe, which is $\mathrm{O}\big(|Y|\big)$ field operations. □

Lemma 3.71. *Assume the Suzuki Conjectures, Conjectures* 2.37*,* 2.38*,* 2.39 *and* 2.40*, and an oracle for the discrete logarithm problem in $\mathbb{F}_q$. There exists a Monte Carlo algorithm with no false positives that, given $\langle Y \rangle \leqslant G \leqslant \mathrm{GL}(26, q)$ such that $G \cong {}^2\mathrm{F}_4(q)$, $S \leqslant \langle Y \rangle \leqslant \mathrm{C}_G(j)$ and $\mathrm{Z}(\mathrm{C}_G(j)) \leqslant \langle Y \rangle$, where $j \in G$ is an involution of class $2A$ and $S \cong \mathrm{Sz}(q)$:*

- *decides whether or not $\langle Y \rangle = \mathrm{C}_G(j)$,*
- *finds effective homomorphisms $\varphi : \langle Y \rangle \to \mathrm{Sz}(q)$ and $\pi : \mathrm{Sz}(q) \to \langle Y \rangle$,*
- *finds $u \in G$ such that $|u| \mid q - 1$, $\mathrm{C}_G(j) < \langle Y, u \rangle$ and $\langle Y, u \rangle$ is contained in a maximal parabolic in G,*
- *finds $\langle Z \rangle \leqslant G$ such that $\langle Z \rangle = \mathrm{O}_2(\mathrm{C}_G(j))$, and $|Z| \in \mathrm{O}\big(\log(q)\big)$.*

The algorithm has expected time complexity

$$\mathrm{O}\big(|Y| + \log(q)^3 + (\xi + \chi_D(q))(\log\log(q))^2\big)$$

field operations.

Proof. The algorithm proceeds as follows:

(1) Find a composition series of the natural module of $\langle Y \rangle$, as in Lemma 3.70. By projecting to the middle composition factor we obtain an effective surjective homomorphism $\varphi_1 : \langle Y \rangle \to S_4$ where $S_4 \leqslant \mathrm{GL}(4, q)$ and $S_4 \cong \mathrm{Sz}(q)$. Also obtain an effective surjective homomorphism $\rho : \mathrm{O}_2(\langle Y \rangle) \to N$, where N is the first non-zero block in the filtration of $\mathrm{O}_2(\langle Y \rangle)$ (*i.e.* N is a vector space). By Conjecture 2.40, $\dim N = 4$.

(2) Use Theorem 3.26 to constructively recognise $\varphi_1(\langle Y \rangle)$ and obtain an effective injective homomorphism $\pi : \mathrm{Sz}(q) \to \langle Y \rangle$, and an effective isomorphism $\varphi_2 : S_4 \to \mathrm{Sz}(q)$. Now $\varphi = \varphi_2 \circ \varphi_1$.
(3) Find random $g \in \langle Y \rangle$ such that $|g| = 2l$. By Proposition 2.34, the proportion of such elements is high. Repeat until $g^l = j'$ is of class $2A$ and $j' \neq j$, which by Conjecture 2.37 happens with high probability.
(4) Using the dihedral trick, find $h \in G$ such that $j^h = j'$, $|h| \mid q-1$ and $\langle Y, h \rangle$ is reducible. By Conjecture 2.38, these elements are easy to find.
(5) Let $u = h((\pi \circ \varphi)(h))^{-1}$, so that u commutes with S. Now $\langle \mathrm{C}_G(j), u \rangle$ is contained in a maximal parabolic, and $\mathrm{C}_G(j)$ is a proper subgroup, since $u \notin \mathrm{C}_G(j)$, but $\langle Y, u \rangle$ is reducible and hence a proper subgroup of G.
(6) Diagonalise u to obtain $\varsigma(a, b)$ for some $a, b \in \mathbb{F}_q^{\times}$. Repeat the two previous steps (find another h) if a or b lie in a proper subfield of $\mathbb{F}_q$. The probability that this happens is low, since $|h| = q - 1$ with high probability.
(7) Find random $y_1, \dots, y_4 \in \langle Y \rangle$, and let $x_i = y_i((\pi \circ \varphi)(y_i))^{-1}$ for $i = 1, \dots, 4$. Then $x_1, \dots, x_4$ are random elements of $\mathrm{O}_2(\langle Y \rangle)$. Return `false` if $\rho(x_1), \dots, \rho(x_4)$ are not linearly independent elements of N, since then with high probability $\langle Y \rangle < \mathrm{C}_G(j)$. Clearly, if the elements are linearly independent, then $\langle Y \rangle = \mathrm{C}_G(j)$ so the algorithm has no false positives.
(8) Find random $x \in \langle Y \rangle$ such that $|x| = 4k$. By Conjecture 2.37, with high probability $x_5 = x^k \in \mathrm{O}_2(\langle Y \rangle)$ and $x_5^2 \in \mathrm{Z}(\langle Y \rangle)$. Repeat until this is true.
(9) Finally let
$$Z = \bigcup_{i=0}^{2m+1} \bigcup_{j=1}^{5} \left\{ x_j^{u^i}, (x_j^2)^{u^i} \right\}$$

Clearly, the dominating term in the running time is Theorem 3.26 and the computation of π, so the expected time complexity is as stated. □

LEMMA 3.72. *Assume the Suzuki Conjectures, Conjectures* 2.38, 2.39 *and* 2.40 *and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$. *There exists a Las Vegas algorithm that, given* $G = \langle X \rangle = {}^2\mathrm{F}_4(q)$ *and an involution* $j \in \langle X \rangle$ *of class* $2A$, *as an* SLP *in* X *of length* $\mathrm{O}(n)$,

- *finds* $\langle Y \rangle \leqslant G$ *such that* $\langle Y \rangle = \mathrm{C}_G(j)$,
- *finds effective inverse isomorphisms* $\varphi : \langle Y \rangle \to \mathrm{Sz}(q)$ *and* $\pi : \mathrm{Sz}(q) \to \langle Y \rangle$,
- *finds* $u \in G$ *such that* $|u| \mid q - 1$, $\mathrm{C}_G(j) < \langle Y, u \rangle$ *and* $\langle Y, u \rangle$ *is contained in a maximal parabolic in* G,
- *finds* $\langle Z \rangle \leqslant G$ *such that* $\langle Z \rangle = \mathrm{O}_2(\mathrm{C}_G(j))$, *and* $|Z| \in \mathrm{O}(\log(q))$.

The elements Y, Z, u *are found as* SLP*s in* X *of length* $\mathrm{O}(n)$. *The algorithm has expected time complexity* $\mathrm{O}(|Y| + \log(q)^3 + (\xi + \chi_D(q))(\log\log(q))^2)$ *field operations.*

PROOF. The algorithm proceeds as follows:

(1) Use the Bray algorithm to find probable generators Y for $\mathrm{C}_G(j)$.
(2) Use the MeatAxe to split up the module of $\langle Y \rangle$ and verify that it splits up as in Conjecture 2.39. Use Theorem 3.3 to verify that the groups acting

on the 4-dimensional submodules are Suzuki groups. Return to the first step if not. It then follows from Proposition 2.34 that $\mathrm{Z}(\mathrm{C}_G(j)) \leqslant \langle Y \rangle$.

(3) Use Lemma 3.71 to determine if $\langle Y \rangle = \mathrm{C}_G(j)$. Return to the first step if not. Since the algorithm of Lemma 3.71 has no false positives, this is Las Vegas.

By Proposition 2.34, $\mathrm{O}(1)$ elements is sufficient to generate $\mathrm{C}_G(j)$ with high probability, so the expected time complexity is as stated, and the elements of Y will be found as SLPs of the same length as j.

From Lemma 3.71 we also obtain u, Z, φ and π as needed. We see from its proof that u and Z will be found as SLPs of the same length as j. □

LEMMA 3.73. *There exists a Las Vegas algorithm that, given*

- $\langle Y \rangle, \langle Z \rangle \leqslant G = \langle X \rangle = {}^2\mathrm{F}_4(q)$ *such that* $\langle Y \rangle = \mathrm{C}_G(j)$ *and* $\langle Z \rangle = \mathrm{O}_2(\langle Y \rangle)$ *where* $j \in G$ *is an involution of class* $2A$ *and* Y, Z *are given as* SLP*s in* X *of length* $\mathrm{O}(n)$,
- *an effective surjective homomorphism* $\varphi : \langle Y \rangle \to \mathrm{Sz}(q)$,
- *an effective injective homomorphism* $\pi : \mathrm{Sz}(q) \to \langle Y \rangle$,
- $g \in \mathrm{GL}(26, q)$,

decides whether or not $g \in \langle Y \rangle$ *and if so returns an* SLP *of* g *in* X *of length* $\mathrm{O}\big(n(\log(q)(\log\log(q))^2 + |Z|)\big)$. *The algorithm has expected time complexity* $\mathrm{O}\big(\xi + \log(q)^3 + |Z|\big)$ *field operations.*

PROOF. Note that φ consists of a change of basis followed by a projection to a submatrix, and hence can be applied to any element of $\mathrm{GL}(26, q)$ using $\mathrm{O}(1)$ field operations.

(1) Use Algorithm 3.2 to express $\varphi(g)$ in the generators of $\mathrm{Sz}(q)$, or return `false` if $\varphi(g) \notin \mathrm{Sz}(q)$. Hence we obtain an SLP for $(\pi \circ \varphi)(g)$ in Y, of length $\mathrm{O}\big(\log(q)(\log\log(q))^2\big)$.
(2) Now $h = g((\pi \circ \varphi)(g))^{-1} \in \langle Z \rangle$. Using the elements of Z, we can apply row reduction to h, and hence obtain an SLP for h in Z of length $\mathrm{O}(|Z|)$. Return `false` if h is not reduced to the identity matrix using Z.
(3) Since Y and Z are SLPs in X, in time $\mathrm{O}\big(\log(q)(\log\log(q))^2\big)$ we obtain an SLP for g in X, of the specified length.

The expected time complexity then follows from Theorem 3.13. □

We are now ready to state our modified Ryba algorithm, which assumes that the precomputations given by the above results have been done.

THEOREM 3.74. *Assume Conjectures* 2.40*,* 2.46*,* 2.47 *and* 3.64*, and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$*. Algorithm* 3.8 *is a Las Vegas algorithm with expected time complexity* $\mathrm{O}\big((\xi + \chi_D(q))\log\log(q) + \log(q)^3 + |Z_A| + |Z_B|\big)$ *field operations. The length of the returned* SLP *is* $\mathrm{O}\big(n(\log(q)(\log\log(q))^2 + |Z_A| + |Z_B|)\big)$ *where* n *is the length of the* SLP*s for* X_A, X_B, Z_A, Z_B *in* X*.*

Algorithm 3.8: RYBA(X, g, X_A, X_B)

Input: $\langle X \rangle \leqslant \mathrm{GL}(26, q)$ such that $\langle X \rangle = {}^2\mathrm{F}_4(q)$, $g \in \mathrm{GL}(26, q)$.
Involution centralisers $\langle X_A \rangle, \langle X_B \rangle \leqslant \langle X \rangle$ for involutions $j_A, j_B \in \langle X \rangle$ of class $2A$ and $2B$, respectively. Effective surjective homomorphisms $\varphi_A : \langle X_A \rangle \to \mathrm{Sz}(q)$, $\varphi_B : \langle X_B \rangle \to \mathrm{SL}(2, q)$. Effective injective homomorphisms $\pi_A : \mathrm{Sz}(q) \to \langle X_A \rangle$, $\pi_B : \mathrm{SL}(2, q) \to \langle X_B \rangle$ and $Z_A \subseteq \langle X_A \rangle$, $Z_B \subseteq \langle X_B \rangle$ such that $\langle Z_A \rangle = \mathrm{O}_2(\langle X_A \rangle)$ and $\langle Z_B \rangle = \mathrm{O}_2(\langle X_B \rangle)$.
Output: If $g \in \langle X \rangle$, TRUE and an SLP of g in X, otherwise FALSE.
// FINDEVENORDERELEMENT is given by Remark 3.68
Use Theorem 3.63 to determine if $g \in \langle X \rangle$ and return FALSE if not.
repeat
  $h :=$ FINDEVENORDERELEMENT(X, g)
  Let w_h be the SLP returned for h.
  Let z be an involution obtained from h by powering up.
until z is of class $2A$.
Find random involution $x \in \langle Z_A \rangle$ of class $2B$.
Let y be an involution obtained from xz by powering up.
Find $c \in \langle X \rangle$ as SLP in X, such that $x^c = j_B$.
Let w_y be an SLP for y^c in X
if y is of class $2A$
then
  Find $c \in \langle X \rangle$ as SLP in X, such that $y^c = j_A$. Let $Y := X_A$.
else
  Find $c \in \langle X \rangle$ as SLP in X, such that $y^c = j_B$. Let $Y := X_B$.
end
Let w_z be an SLP for z^c in X.
Find $c \in \langle X \rangle$ as SLP in X, such that $z^c = j_A$.
Let w_{hg} be an SLP for h^c in X.
Let $w_g := w_h^{-1} w_{hg}$ be an SLP for g in X.
return TRUE, w_g

PROOF. By Theorem 3.67, the length of w_h is $\mathrm{O}\big(\log\log(q)\big)$. By Remark 3.68, $h = h_1 g$ and w_h is an SLP for h_1.

Then z is found using Proposition 1.4, and by Proposition 3.69 it is of class $2A$ with high probability. By Proposition 2.34, the class can be determined by computing the Jordan form.

By Conjecture 2.40, x will have class $2B$ with high probability, and then xz has even order by Proposition 2.34. Again we use Proposition 1.4 to find y.

Using the dihedral trick, we find c. Note that c will be found as an SLP of length $\mathrm{O}\big(n\big)$, since we have an SLP for x, and we can assume that the SLP for j_B has length $\mathrm{O}\big(n\big)$. Now $\langle x, z \rangle$ is dihedral with central involution y, so $y^c \in \langle X_B \rangle$. Using the $\mathrm{SL}(2, q)$ version of Lemma 3.73, we find w_y using $\mathrm{O}\big(\xi + \log(q)^3 + |Z_B|\big)$ field operations, and w_y has length $\mathrm{O}\big(n(\log(q)(\log\log(q))^2 + |Z_B|)\big)$.

The next c is again found using the dihedral trick, and comes as an SLP of the same length as w_y. Then $z^c \in \langle Y \rangle$ since y is central in $\langle x, z \rangle$. Hence we again use Lemma 3.73 (or its $\mathrm{SL}(2, q)$ version) to obtain w_z, with the same length as w_y (or with Z_B replaced by Z_A).

Finally, h clearly centralises z, and we now have an `SLP` for z, so we obtain another c as `SLP` in X, and use Lemma 3.73 to obtain an `SLP` for h^c. Hence we obtain w_{gh}, which is an `SLP` for h, and finally an `SLP` w_g for g. Since w_h has length $\mathrm{O}\big(\log\log(q)\big)$, the length of w_g is as specified.

The expected time complexity follows from Theorem 3.67, Lemma 3.73 and Proposition 1.4. □

3.3.4. Conjugates of the standard copy. We now consider the situation where we are given $\langle X \rangle \leqslant \mathrm{GL}(26, q)$, such that $\langle X \rangle \cong {}^2\mathrm{F}_4(q)$, so that $\langle X \rangle$ is a conjugate of ${}^2\mathrm{F}_4(q)$, and the problem is to find $g \in \mathrm{GL}(26, q)$, such that $\langle X \rangle^g = {}^2\mathrm{F}_4(q)$.

LEMMA 3.75. *Assume Conjectures* 2.39, 2.40 *and* 2.41. *There exists a Las Vegas algorithm that, given*

- $G = \langle X \rangle \leqslant \mathrm{GL}(26, q)$ *such that* $\langle X \rangle \cong {}^2\mathrm{F}_4(q)$,
- $\langle Y \rangle \leqslant \langle X \rangle$ *such that* $\langle Y \rangle = \mathrm{C}_G(j)$ *for some involution* $j \in G$ *of class* $2A$,
- $g \in \langle X \rangle$ *such that* $|g| \mid q-1$, $\langle g \rangle \cap \langle Y \rangle = \langle 1 \rangle$ *and* $P = \langle Y, g \rangle$ *is contained in a maximal parabolic in* $\langle X \rangle$,

finds $\langle Z \rangle \leqslant \langle Y, g \rangle$ *such that* $\langle Z \rangle = \mathrm{C}_P(g) \cong \mathrm{Sz}(q) \times \mathrm{C}_{q-1}$ *and* $\langle W \rangle \leqslant \langle Z \rangle$ *such that* $\langle W \rangle \cong \mathrm{Sz}(q)$.

The expected time complexity is $\mathrm{O}\big(\sigma_0(\log(q))\log(q)\big)$ *field operations. If* Y *and* g *are given as* `SLP`*s in* X *of length* $\mathrm{O}\big(n\big)$, *then* Z *and* W *will be returned as* `SLP`*s in* X, *also of length* $\mathrm{O}\big(n\big)$.

PROOF. By Conjecture 2.39 we have $\langle Y \rangle \cong [q^{10}]\colon \mathrm{Sz}(q)$. Since $\langle g \rangle \cap \langle Y \rangle = \langle 1 \rangle$, it follows from Proposition 2.34 that $\langle g \rangle$ lies in the cyclic group C_{q-1} on top of P. Then g acts fixed-point freely on $\mathrm{O}_2(P)$ and hence $\mathrm{C}_P(g) \cong \mathrm{Sz}(q) \times \mathrm{C}_{q-1}$ and $\mathrm{C}_P(g)' \cong \mathrm{Sz}(q)$.

The algorithm proceeds as follows:

(1) Choose random $a_0 \in P$ and use Corollary 1.11 to find b_1, such that $a_1 = b_1^{-1} a_0$ centralises g modulo $\Phi(\mathrm{O}_2(P))$.
(2) Use Corollary 1.11 to find b_2, such that $c_1 = b_2^{-1} a_1$ centralises g modulo $\Phi(\Phi(\mathrm{O}_2(P)))$. By Conjecture 2.40, $\Phi(\Phi(\mathrm{O}_2(P))) = \langle 1 \rangle$, so $c_1 \in \mathrm{C}_P(g)$. Similarly find $c_2 \in \mathrm{C}_P(g)$.
(3) Now $Z = \{g, c_1, c_2\}$ satisfies $\langle Z \rangle \leqslant \mathrm{C}_P(g)$, so find probable generators W for $\langle Z \rangle'$. Clearly, $\langle Z \rangle = \mathrm{C}_P(g)$ if and only if $\langle W \rangle \cong \mathrm{Sz}(q)$. Use the MeatAxe to split up the module for $\langle W \rangle$ and verify that it has the structure given by Conjecture 2.41. Return to the first step if not.
(4) From the 4-dimensional submodules, we obtain an image W_4 of W in $\mathrm{GL}(4, q)$. Use Theorem 3.3 to determine if $\langle W_4 \rangle \cong \mathrm{Sz}(q)$. Return to the first step if not.

By Proposition 2.12, two random elements generate $\mathrm{Sz}(q)$ with high probability, so the probability that $\langle Z \rangle' \cong \mathrm{Sz}(q)$ is also high. Hence by Theorem 3.3, the expected time complexity is as stated. □

LEMMA 3.76. *Assume the Suzuki Conjectures, the Big Ree Conjectures and an oracle for the discrete logarithm problem in $\mathbb{F}_q$. There exists a Las Vegas algorithm that, given $\langle X \rangle \leqslant \mathrm{GL}(26, q)$ such that $\langle X \rangle \cong {}^2\mathrm{F}_4(q)$, finds $\langle Y \rangle \leqslant \langle X \rangle$ and $g \in \langle X \rangle$ such that $\langle Y \rangle \cong \mathrm{Sz}(q)$ and $\langle Y, g \rangle \cong \mathrm{Sz}(q) \wr \mathrm{C}_2$. The elements of Y are expressed as SLPs in X of length $\mathrm{O}(\log\log(q))$; g is expressed as an SLP in X of length $\mathrm{O}((\log\log(q))^2)$. The algorithm has expected time complexity $\mathrm{O}(\lvert X \rvert + \log(q)^3 + (\xi + \chi_D(q))(\log\log(q))^2)$ field operations.*

PROOF. The idea is to first find one copy of $\mathrm{Sz}(q)$ by finding a centraliser of an involution of class $2A$, which has structure $[q^{10}]{:}\,\mathrm{Sz}(q)$, then use the Formula to find the $\mathrm{Sz}(q)$ inside this. Next we find a maximal parabolic inside this $\mathrm{Sz}(q)$ and use the dihedral trick and the Formula to conjugate it back to our involution centraliser. The conjugating element together with the $\mathrm{Sz}(q)$ will generate a copy of $\mathrm{Sz}(q) \wr \mathrm{C}_2$.

The algorithm proceeds as follows:

(1) Use Algorithm 3.7 to find an element of even order and then use Proposition 1.4 to find an involution $j \in \langle X \rangle$. Repeat until j is of class $2A$, which by Proposition 3.69 happens with high probability.
(2) Use Lemma 3.72 to find $\langle C \rangle \leqslant \langle X \rangle$ and $y_1 \in \langle X \rangle$ such that $\langle C \rangle = \mathrm{C}_G(j)$, $\lvert y_1 \rvert \mid q-1$, $y_1 \notin \mathrm{C}_G(j)$ and $\langle C, y_1 \rangle$ is contained in a maximal parabolic in $\langle X \rangle$.
(3) Use Lemma 3.75 to find $\langle Y \rangle \leqslant \langle C, y_1 \rangle$ such that $\langle Y \rangle \cong \mathrm{Sz}(q)$ and $\langle Y \rangle$ commutes with y_1.
(4) Use the MeatAxe to split up the module of $\langle Y \rangle$. By Conjecture 2.41 we obtain 4-dimensional submodules, and hence a homomorphism $\rho : \langle Y \rangle \to \mathrm{GL}(4, q)$.
(5) Use Theorem 3.1 to find an effective isomorphism $\varphi : \rho(\langle Y \rangle) \to \mathrm{Sz}(q)$.
(6) Use the first steps of the proof of Theorem 3.12 to find $a', y_2' \in \rho(\langle Y \rangle)$, as SLPs in the generators of $\rho(\langle Y \rangle)$, such that $\lvert a' \rvert = 4$, $\lvert y_2' \rvert \mid q-1$ and $\langle a', y_2' \rangle$ is contained in a maximal parabolic in $\rho(\langle Y \rangle)$. Evaluate the SLPs on Y to obtain $a, y_2 \in \langle Y \rangle$ with similar properties.
(7) Using the dihedral trick, find $h_1 \in \langle X \rangle$ such that $j^{h_1} = a^2$ and let $y_3 = y_1^{h_1}$. Since $\langle a, y_2 \rangle$ is a proper subgroup of $\langle Y \rangle$, it follows that $\langle \mathrm{C}_G(a^2), y_2 \rangle$ is a proper subgroup of G, and hence it is contained in a maximal parabolic. Clearly $\langle \mathrm{C}_G(a^2), y_3 \rangle$ is also contained in the same maximal parabolic. We know from the structure of $\langle a, y_2 \rangle$ that $y_2 \notin \mathrm{C}_G(a^2)$, and since $y_1 \notin \mathrm{C}_G(j)$, it follows that $y_3 \notin \mathrm{C}_G(a^2)$, so $\langle y_2 \rangle$ and $\langle y_3 \rangle$ both lie in a group of shape $[q^{10}] : (q-1)$ and hence are conjugate modulo $\mathrm{O}_2(\mathrm{C}_G(a^2))$. Now we want to conjugate $\langle y_3 \rangle$ to $\langle y_2 \rangle$ while fixing a^2.

(8) Diagonalise y_2 and y_3 to obtain $\varsigma(a_2, b_2)$ and $\varsigma(a_3, b_3)$. Use the discrete logarithm oracle to find an integer $k \in \mathbb{Z}$ such that $a_2^k = a_3$. If no such k exists, then find another pair of a, y_2, but this can only happen if $|y_2|$ is a proper divisor of $q - 1$.

(9) Notice that y_3 also can diagonalise to $\varsigma(a_3^{-1}, b_3)$, so now

$$y_2^k \equiv y_3 \mod \mathrm{O}_2(\mathrm{C}_G(a^2))$$

or $y_2^k \equiv y_3^{-1} \mod \mathrm{O}_2(\mathrm{C}_G(a^2))$. In the latter case, invert y_3.

(10) Use Lemma 1.10 to find $c_1 \in \langle y_2^k, y_3 \rangle$ such that

$$(y_2^k)^{c_1} \equiv y_3 \mod \Phi(\mathrm{O}_2(\mathrm{C}_G(a^2)))$$

and then $c_2 \in \langle (y_2^k)^{c_1}, y_3 \rangle$ such that

$$(y_2^k)^{c_1 c_2} \equiv y_3 \mod \Phi(\Phi(\mathrm{O}_2(\mathrm{C}_G(a^2)))).$$

By Conjecture 2.40, $\Phi(\Phi(\mathrm{O}_2(\mathrm{C}_G(a^2)))) = \langle 1 \rangle$, so $\langle y_2 \rangle^{c_1 c_2} = \langle y_3 \rangle$.

(11) Now $h_2 = (c_1 c_2)^{-1} \in \mathrm{C}_G(a^2)$ since by Lemma 1.10, both c_1 and c_2 centralise a^2. Clearly, h_2 conjugates $\langle y_3 \rangle$ to $\langle y_2 \rangle$. Hence $g = h_1 h_2$ conjugates $\langle j, y_1 \rangle$ to $\langle a^2, y_2 \rangle$, and by construction $\langle a^2, y_2 \rangle \leqslant \langle Y \rangle$ commutes with both j and y_1. Since $\langle j, y_1 \rangle$ is contained in a maximal parabolic of another copy of $\mathrm{Sz}(q)$, it follows that $\langle Y \rangle$ commutes with this copy. Thus $\langle Y, g \rangle \cong \mathrm{Sz}(q) \wr \mathrm{C}_2$.

By Theorem 3.67, the length of the `SLP` for j is $\mathrm{O}(\log\log(q))$. Then by Lemma 3.72, the `SLP`s of C and y_1 will also have length $\mathrm{O}(\log\log(q))$. By Lemma 3.75, the `SLP`s for Y will have length $\mathrm{O}(\log\log(q))$. By Theorem 3.12, the `SLP`s for a and y_2 will have length $\mathrm{O}(\log\log(q)^2)$, and hence h_1 and h_2 will also have this length.

The expected time complexity follows from Lemma 3.72. □

LEMMA 3.77. *Assume Conjecture* 2.42. *There exists a Las Vegas algorithm that, given* $\langle X \rangle, \langle Y \rangle, \langle Z \rangle \leqslant \mathrm{GL}(26, q)$ *such that* $\langle X \rangle \cong {}^2\mathrm{F}_4(q) = \langle Y \rangle$, $\langle Z \rangle \cong \mathrm{Sz}(q) \times \mathrm{Sz}(q)$ *and* $\langle Z \rangle \leqslant \langle X \rangle \cap \langle Y \rangle$, *finds* $g \in \mathrm{GL}(26, q)$ *such that* $\langle X \rangle^g = \langle Y \rangle$. *The algorithm has expected time complexity* $\mathrm{O}(|X| + |Y| + \log(q))$ *field operations.*

PROOF. Let M be the module of $\langle Z \rangle$. Observe that g must centralise $\langle Z \rangle$, so $g \in \mathrm{C}_{\mathrm{GL}(26,q)}(\langle Z \rangle) = \mathrm{Aut}(M) \subseteq \mathrm{End}_{\langle Z \rangle}(M)$. By Conjecture 2.42, the endomorphism ring of M has dimension 3. The algorithm proceeds as follows:

(1) Use the MeatAxe to find $e_1, e_2, e_3 \in \mathrm{GL}(26, q)$ such that $\mathrm{End}_{\langle Z \rangle}(M) = \oplus_{i=1}^3 \langle e_i \rangle$.

(2) Let x_1, x_2, x_3 be indeterminates and let

$$h(x_1, x_2, x_3) = \sum_{i=1}^{3} x_i e_i \in \mathrm{Mat}_{26}(\mathbb{F}_q[x_1, x_2, x_3]).$$

(3) Use the MeatAxe to find matrices Q_X, Q_Y corresponding to the quadratic forms preserved by $\langle X \rangle$ and $\langle Y \rangle$.

(4) A necessary condition on $h(x_1, x_2, x_3)$ for it to conjugate $\langle X \rangle$ to $\langle Y \rangle$ is the following equation:

$$h(x_1, x_2, x_3) Q_X h(x_1, x_2, x_3) = Q_Y \tag{3.54}$$

which determines 676 quadratic equations in x_1, x_2, x_3.

(5) Hence we obtain $P \subseteq \mathbb{F}_q[x_1, x_2, x_3]$ where each element of P has degree 2 and $|P| \leqslant 676$. Every $f \in P$ has 7 coefficients, so we obtain an additive group homomorphism $\rho : \langle P \rangle \to \mathbb{F}_q^7$.

(6) Now $\dim \rho(P) = 3$, so let b_1, b_2, b_3 be a basis of $\rho(P)$ and let $f_i = \rho^{-1}(b_i) \in P$ for $i = 1, \dots, 3$.

(7) By Proposition 2.43, the variety of the ideal $I = \langle f_1, f_2, f_3 \rangle \trianglelefteq \mathbb{F}_q[x_1, x_2, x_3]$ has size 2. Find this variety using Theorem 1.3.

(8) Let h_1, h_2 be the corresponding elements of $\operatorname{End}_{\langle Z \rangle}(M)$. Clearly, one of them must also lie in $\operatorname{Aut}(M)$ and conjugate $\langle X \rangle$ to $\langle Y \rangle$, since our g exists and satisfies the necessary conditions which led to h_1 and h_2. Use Theorem 3.63 to determine which h_i satisfies $\langle X \rangle^{h_i} = \langle Y \rangle$.

Clearly, this is a Las Vegas algorithm and the expected time complexity follows from Theorem 1.3. □

THEOREM 3.78. *Assume the Suzuki Conjectures, the Big Ree Conjectures, and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$. *There exists a Las Vegas algorithm that, given* $\langle X \rangle, \langle Y \rangle \leqslant \mathrm{GL}(26, q)$ *such that* $\langle X \rangle \cong {}^2\mathrm{F}_4(q) = \langle Y \rangle$, *finds* $g \in \mathrm{GL}(26, q)$ *such that* $\langle X \rangle^g = \langle Y \rangle$. *The algorithm has expected time complexity* $\mathrm{O}\big(|X| + \log(q)^3 + (\xi + \chi_D(q))(\log\log(q))^2\big)$ *field operations.*

PROOF. The algorithm proceeds as follows:

(1) Use Lemma 3.76 to find $S_1 \subseteq \langle X \rangle$ and $c_1 \in \langle X \rangle$ such that $\langle S_1 \rangle \cong \mathrm{Sz}(q)$ and $\langle S_1, c_1 \rangle \cong \mathrm{Sz}(q) \wr \mathrm{C}_2$. Let $\langle S_2 \rangle = \langle S_1 \rangle^{c_1}$ so that $\langle S_1, S_2 \rangle \cong \mathrm{Sz}(q) \times \mathrm{Sz}(q)$.

(2) Similarly find $S_3, S_4 \subseteq \langle Y \rangle$ such that $\langle S_3, S_4 \rangle \cong \mathrm{Sz}(q) \times \mathrm{Sz}(q)$.

(3) Use the MeatAxe to split up the modules of each $\langle S_i \rangle$. By Proposition 2.41, we obtain 4-dimensional submodules, and hence surjective homomorphisms $\rho_i : \langle S_i \rangle \to \mathrm{GL}(4, q)$ for $i = 1, \dots, 4$.

(4) Use Theorem 3.1 to find effective isomorphisms $\varphi_i : \rho_i(\langle S_i \rangle) \to \mathrm{Sz}(q)$ for $i = 1, \dots, 4$. If S is the standard generating set for $\mathrm{Sz}(q)$, we then obtain standard generating sets R_i for $\langle S_i \rangle$ by obtaining `SLP`s for S in the generators of $\rho_i(\langle S_i \rangle)$ and then evaluating these on S_i.

(5) Let M_1 be the module for $\langle R_1, R_2 \rangle$ and let M_2 be the module for $\langle R_3, R_4 \rangle$. Now $M_1 \cong M_2$, and all R_i are equal, so we can use the MeatAxe to find a change of basis matrix $h_1 \in \mathrm{GL}(26, q)$ between M_1 and M_2.

(6) Then $\langle S_1, S_2 \rangle^{h_1} = \langle S_3, S_4 \rangle$ and hence $\langle S_3, S_4 \rangle \leqslant \langle X \rangle^{h_1} \cap \langle Y \rangle$. Use Lemma 3.77 to find $h_2 \in \mathrm{GL}(26, q)$ such that $\langle X \rangle^{h_1 h_2} = \langle Y \rangle$. Hence $g = h_1 h_2$.

Clearly, this is a Las Vegas algorithm and the expected time complexity follows from Lemma 3.76. □

3.3.5. Constructive recognition. Finally, we can now state and prove our main theorem.

THEOREM 3.79. *Assume the Suzuki Conjectures, the Big Ree Conjectues, and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$*. There exists a Las Vegas algorithm that, given* $\langle X \rangle \leqslant \mathrm{GL}(26, q)$ *satisfying the assumptions in Section* 1.2.7*, with* $q = 2^{2m+1}$*,* $m > 0$ *and* $\langle X \rangle \cong {}^2\mathrm{F}_4(q)$*, finds an effective isomorphism* $\varphi : \langle X \rangle \to {}^2\mathrm{F}_4(q)$*. The algorithm has expected time complexity* $\mathrm{O}\big((\xi + \chi_D(q))(\log\log(q))^2 + |X| + \log(q)^3\big)$ *field operations.*

The inverse of φ *is also effective. Each image and pre-image of* φ *can be computed using* $\mathrm{O}\big(1\big)$ *field operations.*

PROOF. Use Theorem 3.78 to obtain $c \in \mathrm{GL}(26, q)$ such that $\langle X \rangle^c = {}^2\mathrm{F}_4(q)$. An an effective isomorphism $\varphi : \langle X \rangle \to {}^2\mathrm{F}_4(q)$ is then defined by $g \mapsto g^c$, which clearly can be computed in $\mathrm{O}\big(1\big)$ field operations. The expected time complexity follows from Theorem 3.78. $\square$

CHAPTER 4

Sylow subgroups

We will now describe algorithms for finding and conjugating Sylow subgroups of the exceptional groups under consideration. Hence we consider the following problems:

(1) Given $\langle X \rangle \leqslant \mathrm{GL}(d,q)$, such that $\langle X \rangle \cong G$ for one of our exceptional groups G, and given a prime number $p \mid |G|$, find $\langle Y \rangle \leqslant \langle X \rangle$ such that $\langle Y \rangle$ is a Sylow p-subgroup of $\langle X \rangle$.

(2) Given $\langle X \rangle \leqslant \mathrm{GL}(d,q)$, such that $\langle X \rangle \cong G$ for some of our exceptional groups G, and given a prime number $p \mid |G|$ and $\langle Y \rangle, \langle Z \rangle \leqslant \langle X \rangle$ such that both $\langle Y \rangle$ and $\langle Z \rangle$ are Sylow p-subgroups of $\langle X \rangle$, find $c \in \langle X \rangle$ such that $\langle Y \rangle^c = \langle Z \rangle$.

The second problem is the difficult one, and often there are some primes that are especially difficult. We will refer to these problems as the "Sylow subgroup problems" for a certain prime p. The first problem is referred to as "Sylow generation" and the second as "Sylow conjugation".

4.1. Suzuki groups

We now consider the Sylow subgroup problems for the Suzuki groups. We will use the notation from Section 2.1, and we will make heavy use of the fact that we can use Theorem 3.26 to constructively recognise the Suzuki groups. Hence we assume that G satisfies the assumptions in Section 1.2.7, so $\mathrm{Sz}(q) \cong G \leqslant \mathrm{GL}(d,q)$.

By Theorem 2.1 and Proposition 2.4, $|G| = q^2(q^2+1)(q-1)$ and all three factors are pairwise relatively prime. Hence we obtain three cases for a Sylow p-subgroup S of G.

(1) p divides q^2, so $p = 2$. Then S is conjugate to $\mathcal{F}$ and hence S fixes a unique point P_S of $\mathcal{O}$, which is easily found using the MeatAxe.

(2) p divides $q-1$. Then S is cyclic and conjugate to a subgroup of $\mathcal{H}$. Hence S fixes two distinct points $P, Q \in \mathcal{O}$, and these points are easily found using the MeatAxe.

(3) p divides q^2+1. Then S is cyclic and conjugate to a subgroup of U_1 or U_2, and S has no fixed points. This is the difficult case.

Theorem 4.1. *Assume the Suzuki Conjectures and an oracle for the discrete logarithm problem in $\mathbb{F}_q$. There exist Las Vegas algorithms that solve the Sylow subgroup problems for $p = 2$ in $\mathrm{Sz}(q) \cong G \leqslant \mathrm{GL}(d,q)$. Once constructive recognition has been performed, the expected time complexity of the Sylow generation is*

$\mathrm{O}\big(d^3 \log(q)(\log\log(q))^2\big)$ *field operations, and* $\mathrm{O}\big(d^3(|Y|+|Z|+\log(q)(\log\log(q))^2)+\log(q)^3\big)$ *field operations for the Sylow conjugation.*

PROOF. Let $H = \mathrm{Sz}(q)$. Using the effective isomorphism, it is sufficient to solve the problems in the standard copy.

The constructive recognition uses Theorem 3.12 to find sets L and U of "standard generators" for H_{P_∞} and H_{P_0}, respectively.

A generating set for a random Sylow 2-subgroup S of H can therefore be computed by taking a random $h \in H$, and as generating set for S take L^h. To obtain a Sylow 2-subgroup R of G, note that we already have the $\mathrm{O}\big(\log(q)\big)$ generators L and h as `SLP`s of length $\mathrm{O}\big((\log\log(q))^2\big)$, so we can evaluate them on X. Hence the expected time complexity is as stated.

Given two Sylow 2-subgroups of G, we use the effective isomorphism to map them to H using $\mathrm{O}\big(d^3(|Y| + |Z|)\big)$ field operations. We can use the MeatAxe to find the points $P_Y, P_Z \in \mathcal{O}$ that are fixed by the subgroups. Then use Lemma 3.11 with U to find $a \in H_{P_0}$ such that $P_Y a = P_\infty$ and $b \in H_{P_0}$ such that $P_Z b = P_\infty$. Then ab^{-1} conjugates one Sylow subgroup to the other, and we already have this element as an `SLP` of length $\mathrm{O}\big(\log(q)(\log\log(q))^2\big)$.

In the case where $P_Y = P_\infty$ and $P_Z = P_0$, we know that T maps P_Y to P_Z. Then use Algorithm 3.2 to obtain an `SLP` for T of length $\mathrm{O}\big(\log(q)(\log\log(q))^2\big)$.

Hence we can evaluate it on X and obtain $c \in G$ that conjugates $\langle Y \rangle$ to $\langle Z \rangle$. Thus the expected time complexity follows from Lemma 3.11 and Theorem 3.13. $\square$

THEOREM 4.2. *Assume the Suzuki Conjectures and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$*. There exist Las Vegas algorithms that solve the Sylow subgroup problems for* $p \mid q-1$ *in* $\mathrm{Sz}(q) \cong G \leqslant \mathrm{GL}(d,q)$*. Once constructive recognition has been performed, the expected time complexity of the Sylow generation is* $\mathrm{O}\big((\xi(d) + \log(q)\log\log(q) + d^3)\log\log(q)\big)$ *field operations, and* $\mathrm{O}\big(d^3(|Y| + |Z| + \log(q)(\log\log(q))^2) + \log(q)^3\big)$ *field operations for the Sylow conjugation.*

PROOF. Let $H = \mathrm{Sz}(q)$. In this case the Sylow generation is easy, since we can determine the highest power e of p such that $p^e \mid q-1$, find a random element of pseudo-order $q-1$, use Proposition 1.4 to obtain an element of order p^e, then evaluate its `SLP` on X. By Proposition 2.6, the expected number of random selections, and hence the length of the `SLP`, is $\mathrm{O}\big(\log\log q\big)$, and we need $\mathrm{O}\big(\log(q)\log\log(q)\big)$ field operations to find the order. Then we evaluate the `SLP` on X using $\mathrm{O}\big(d^3\log\log(q)\big)$ field operations, so the expected time complexity is as stated.

For the Sylow conjugation, recall that the constructive recognition uses Theorem 3.12 to find sets L and U of "standard generators" for H_{P_∞} and H_{P_0}, respectively.

Given two Sylow p-subgroups of G, we use the effective isomorphism to map them to H using $\mathrm{O}\big(d^3(|Y|+|Z|)\big)$ field operations. Let $H_Y, H_Z \leqslant H$ be the resulting subgroups. Using the MeatAxe, we can find $P_Y \neq Q_Y \in \mathcal{O}$ that are fixed by H_Y

and $P_Z \neq Q_Z \in \mathcal{O}$ that are fixed by H_Z. Order the points so that $P_Y \neq P_0$ and $P_Z \neq P_0$.

Use Lemma 3.11 with U to find $a_1 \in H_{P_0}$ such that $P_Y a_1 = P_\infty$. Then use L to find $a_2 \in H_{P_\infty}$ such that $Q_Y a_1 a_2 = P_0$. Similarly we find $b_1, b_2 \in H$ such that $P_Z b_1 b_2 = P_\infty$ and $Q_Z b_1 b_2 = P_0$. Then $a_1 a_2 (b_1 b_2)^{-1}$ conjugates one Sylow subgroup to the other, and we already have this element as an `SLP` of length $\mathrm{O}\big(\log(q)(\log\log(q))^2\big)$. Hence we can evaluate it on X and obtain $c \in G$ that conjugates $\langle Y \rangle$ to $\langle Z \rangle$. Thus the expected time complexity is as stated.

□

Lemma 4.3. *There exists a Las Vegas algorithm that, given $g, h \in \mathrm{Sz}(q)$ with $|g| = |h|$ both dividing $q \pm t + 1$, finds $c \in \langle g \rangle$ such that c is conjugate to h in $\mathrm{Sz}(q)$. The expected time complexity is $\mathrm{O}\big(\log q\big)$ field operations.*

Proof. The algorithm proceeds as follows:

1. Find the minimal polynomial f_1 of g. By Theorem 2.1, g acts irreducibly on $\mathbb{F}_q^4$, so f_1 is irreducible. Let $F = \mathbb{F}_q[x]/\langle f_1 \rangle$, so that F is the splitting field of f_1. Clearly $F \cong \mathbb{F}_{q^4}$ and $F^\times = \langle \alpha \rangle$ where α is a root of f_1. Moreover, $x \mapsto g$ defines an isomorphism $F \to \mathbb{F}_q(\langle g \rangle)$, where the latter is the subfield of $\mathrm{Mat}_4(\mathbb{F}_q)$ generated by g.
2. Find the minimal polynomial f_2 of h. Then F is also the splitting field of f_2, and if $\beta \in F$ is a root of f_2, then β is expressed as a polynomial f_3 in α, with coefficients in $\mathbb{F}_q$. Similarly, $x \mapsto h$ defines an isomorphism $F \to \mathbb{F}_q(\langle h \rangle)$.
3. Now f_3 defines an isomorphism $\mathbb{F}_q(\langle h \rangle) \to \mathbb{F}_q(\langle g \rangle)$ as $h \mapsto f_3(g)$, because h and $f_3(g)$ have the same minimal polynomial. Hence if we let $c = f_3(g)$, then c has the same eigenvalues as h, so c and h are conjugate in $\mathrm{GL}(4, q)$. Then $|c| = |h|$ and $\mathrm{Tr}(c) = \mathrm{Tr}(h)$, so by Proposition 2.8, c is also conjugate to h in $\mathrm{Sz}(q)$. Moreover, both $\langle g \rangle$ and $\langle c \rangle$ are subgroups of $\mathbb{F}_q(\langle g \rangle)^\times$, but since $|c| = |h| = |g|$, they must be the same. Thus $c \in \langle g \rangle$.

By [**Gie95**], the minimal polynomial is found using $\mathrm{O}\big(1\big)$ field operations. Hence by Theorem 1.1, the expected time complexity is as stated. □

The conjugation algorithm described in the following result is essentially due to Mark Stather and Scott Murray.

Theorem 4.4. *Assume the Suzuki Conjectures and an oracle for the discrete logarithm problem in $\mathbb{F}_q$. There exist Las Vegas algorithms that solve the Sylow subgroup problems for $p \mid q^2 + 1$ in $\mathrm{Sz}(q) \cong G \leqslant \mathrm{GL}(d, q)$. Once constructive recognition has been performed, the expected time complexity of the Sylow generation is $\mathrm{O}\big((\xi(d) + \log(q)\log\log(q) + d^3)\log\log(q)\big)$ field operations, and $\mathrm{O}\big(\xi(d) + d^3(|Y| + |Z| + \log(q)(\log\log(q))^2) + \log(q)^3\big)$ field operations for the Sylow conjugation.*

PROOF. Let $H = \mathrm{Sz}(q)$. The Sylow generation is analogous to the case in Theorem 4.2, since by Proposition 2.6, we can easily find elements of pseudo-order $q \pm t + 1$.

Given two Sylow p-subgroups of G, we use the effective isomorphism to map them to H, using $\mathrm{O}\big(d^3(|Y| + |Z|)\big)$ field operations. The resulting generating sets must contain elements $h_y, h_z \in H$ of order p, since the Sylow subgroups are cyclic. Let J be as in (2.24), so that H preserves the symplectic form J, and let Ψ be as in Section 2.1.2: the automorphism of $\mathrm{Sp}(4, q)$ whose set of fixed points is $\mathrm{Sz}(q)$.

(1) Use Lemma 4.3 to replace h_y. Henceforth assume that h_y and h_z are conjugate in H.
(2) Find $g_1 \in \mathrm{GL}(4, q)$ such that $h_y^{g_1} = h_z$. This can be done by a similarity test, or computation of Jordan forms, using [**Ste97**]. The next step is to find a matrix g_2 such that $g_2 g_1 \in \mathrm{Sp}(4, q)$, and $g_2 g_1$ also conjugates h_y to h_z.
(3) Let $A = \mathrm{C}_{\mathrm{GL}(4,q)}(h_y)$ (the automorphism group of the module of $\langle h_y \rangle$). Since $\langle h_y \rangle$ is irreducible, by Schur's Lemma $A \cong \mathbb{F}_{q^4}^{\times}$. Such an isomorphism $\theta : A \to \mathbb{F}_{q^4}$, and its inverse, can be found using the MeatAxe.
(4) Now define an automorphism φ of A as $\varphi(a) = J a^T J^{-1}$. Then φ has order 2 and $A \cong \mathbb{F}_{q^4}^{\times}$. Recall that $\mathbb{F}_{q^4}$ has a unique automorphism of order 2 ($k \mapsto k^{q^2}$), which must be $\theta \circ \varphi \circ \theta^{-1}$.
(5) Let $t_1 = J g_1^{-T} J^{-1} g_1^{-1}$ and observe that $t_1 \in A$. We want to find $g_2 \in A$ such that $g_2 g_1 J (g_2 g_1)^T = J$, which is equivalent to $\varphi(g_2) g_2 = t_1$. Using θ, this is a norm equation in $\mathbb{F}_{q^4}$ over $\mathbb{F}_{q^2}$. In other words, we consider $\theta(g_2)^{q^2+1} = \theta(t_1)$, which is solved for example using [**HRD05**, Lemma 2.2].
(6) Hence $g_2 g_1$ lies in $\mathrm{Sp}(4, q)$, and g_2 fixes h_y, so $g_2 g_1$ conjugates h_y to h_z. The next step is to find a matrix g_3, such that $g_3 g_2 g_1 \in \mathrm{Sz}(q)$, and such that $g_3 g_2 g_1$ also conjugates h_y to h_z. Hence we want $g_3 \in \mathrm{Sp}(4, q)$ and $\Psi(g_3 g_2 g_1) = g_3 g_2 g_1$.
(7) Find $w \in \mathbb{F}_{q^4}^{\times}$ of order $q^2 + 1$, by taking the $q^2 - 1$ power of a primitive element. Then $\varphi(\theta^{-1}(w))\theta^{-1}(w) = 1$, which implies that $\theta^{-1}(w) J \theta^{-1}(w)^T = J$, and hence $\theta^{-1}(w) \in \mathrm{Sp}(4, q)$. Similarly, every element of $\langle w \rangle$ gives rise to matrices in $\mathrm{Sp}(4, q)$. We therefore want to find an integer i, such that $w^i = g_3$.
(8) Moreover, we want

$$\begin{aligned} \Psi(\theta^{-1}(w^i) g_2 g_1) &= \theta^{-1}(w^i) g_2 g_1 \Leftrightarrow \\ \Psi(\theta^{-1}(w))^i \Psi(g_2 g_1) &= \theta^{-1}(w)^i g_2 g_1 \Leftrightarrow \\ \Psi(\theta^{-1}(w))^i \theta^{-1}(w)^{-i} &= g_2 g_1 \Psi(g_2 g_1)^{-1} \Leftrightarrow \\ \theta(\Psi(\theta^{-1}(w)))^i w^{-i} &= \theta(g_2 g_1 \Psi(g_2 g_1)^{-1}) \end{aligned} \tag{4.1}$$

so if we let $t_2 = \theta(g_2 g_1 \Psi(g_2 g_1)^{-1})$, we want to find an integer i such that $\theta(\Psi(\theta^{-1}(w)))^i w^{-i} = t_2$.

(9) Use the discrete log oracle to find k such that $\theta(\Psi(\theta^{-1}(w))) = w^k$. Since $g_2 g_1 \in \mathrm{Sp}(4, q)$ it follows that $t_2 \in \langle w \rangle$. Use the discrete log oracle to find $n \in \mathbb{Z}$ such that $w^n = t_2$. Our equation turns into $(k-1)i \equiv n \pmod{q^2+1}$, which we solve to find i.

By [**HRD05**, Lemma 2.2], this whole process uses expected $\mathrm{O}\big(\log q\big)$ field operations. Finally we use the effective isomorphism to map the conjugating element back to G. Hence the time complexity is as stated. □

4.2. Small Ree groups

We now consider the Sylow subgroup problems for the small Ree groups. We will use the notation from Section 2.2, and we will make heavy use of the fact that we can use Theorem 3.62 to constructively recognise the small Ree groups. Hence we assume that G satisfies the assumptions in Section 1.2.7, so $\mathrm{Ree}(q) \cong G \leqslant \mathrm{GL}(d, q)$.

By Proposition 2.15, we obtain 4 cases for a Sylow p-subgroup S of G.

(1) $p = 2$, so that by [**HB82**, Chapter 11, Theorem 13.2], S is elementary abelian of order 8 and $[\mathrm{N}_G(S) : S] = 21$.
(2) p divides q^3, so $p = 3$. Then S is conjugate to $U(q)$ and hence S fixes a unique point P_S of $\mathcal{O}$, which is easily found using the MeatAxe.
(3) p divides $q - 1$ and $p > 2$. Then S is cyclic and conjugate to a subgroup of $H(q)$. Hence S fixes two distinct points $P, Q \in \mathcal{O}$, and these points are easily found using the MeatAxe.
(4) p divides $q^3 + 1$ and $p > 2$. Then S is cyclic and conjugate to a subgroup of A_0, A_1 or A_2 from Proposition 2.20. In this case, we have only solved the Sylow generation problem.

Theorem 4.5. *Assume the small Ree Conjectures and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$. *There exist Las Vegas algorithms that solve the Sylow subgroup problems for* $p = 3$ *in* $\mathrm{Ree}(q) \cong G \leqslant \mathrm{GL}(d, q)$. *Once constructive recognition has been performed, the expected time complexity of the Sylow generation is* $\mathrm{O}\big(d^3(\log(q)\log\log(q))^2\big)$ *field operations, and* $\mathrm{O}\big(d^3(|Y| + |Z| + (\log(q)\log\log(q))^2) + \log(q)^3\big)$ *field operations for the Sylow conjugation.*

Proof. Let $H = \mathrm{Ree}(q)$. The constructive recognition uses Theorem 3.40 to find sets L and U of "standard generators" for H_{P_∞} and H_{P_0}, respectively.

A generating set for a random Sylow 3-subgroup S of H can therefore be computed by finding a random $h \in H$, and as generating set for S take $\{m^g \mid m \in L\}$. To obtain a Sylow subgroup R of G, note that we already have the $\mathrm{O}\big(\log(q)\big)$ generators of L and h as `SLP`s of length $\mathrm{O}\big(\log(q)(\log\log(q)^2)\big)$, so we can evaluate them on X. Hence the expected time complexity is as stated.

Given two Sylow 3-subgroups of G, we use the effective isomorphism to map them to H using $\mathrm{O}\big(d^3(|Y| + |Z|)\big)$ field operations. We can use the MeatAxe to find the points $P_Y, P_Z \in \mathcal{O}$ that are fixed by the subgroups. Then use Lemma 3.39 with U to find $a \in H$, such that $P_Y a = P_\infty$, and $b \in H$, such that $P_Z b = P_\infty$. Then ab^{-1} conjugates one Sylow subgroup to the other, and we already have this element as an SLP of length $\mathrm{O}\big((\log(q) \log\log(q))^2\big)$.

In the case where $P_Y = P_\infty$ and $P_Z = P_0$, we know that Υ maps P_Y to P_Z. Then use Algorithm 3.6 to obtain an SLP for Υ of length $\mathrm{O}\big((\log(q) \log\log(q))^2\big)$.

Hence we can evaluate it on X and obtain $c \in G$ that conjugates $\langle Y \rangle$ to $\langle Z \rangle$. Thus the expected time complexity follows from Lemma 3.39 and Theorem 3.42. □

THEOREM 4.6. *Assume the small Ree Conjectures and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$. *There exist Las Vegas algorithms that solve the Sylow subgroup problems for* $p \mid q-1$, $p > 2$, *in* $\mathrm{Ree}(q) \cong G \leqslant \mathrm{GL}(d, q)$. *Once constructive recognition has been performed, the expected time complexity of the Sylow generation is* $\mathrm{O}\big((\xi(d) + \log(q) \log\log(q) + d^3) \log\log(q)\big)$ *field operations, and* $\mathrm{O}\big(d^3(|Y| + |Z| + (\log(q) \log\log(q))^2) + \log(q)^3\big)$ *field operations for the Sylow conjugation.*

PROOF. Let $H = \mathrm{Ree}(q)$. In this case the Sylow generation is easy, since we can determine the highest power e of p such that $p^e \mid q-1$, find a random element of pseudo-order $q-1$, use Proposition 1.4 to obtain an element of order p, then evaluate its SLP on X. By Proposition 2.23, the expected number of random selections, and hence the length of the SLP is $\mathrm{O}\big(\log\log q\big)$, and we need $\mathrm{O}\big(\log(q) \log\log(q)\big)$ field operations to find the order. Then we evaluate the SLP on X using $\mathrm{O}\big(d^3 \log\log(q)\big)$ field operations, so the expected time complexity is as stated.

For the Sylow conjugation, recall that the constructive recognition uses Theorem 3.40 to find sets L and U of "standard generators" for H_{P_∞} and H_{P_0}, respectively.

Given two Sylow p-subgroups of G, we use the effective isomorphism to map them to H using $\mathrm{O}\big(d^3(|Y| + |Z|)\big)$ field operations. Let $H_Y, H_Z \leqslant H$ be the resulting subgroups. Using the MeatAxe, we can find $P_Y, Q_Y \in \mathcal{O}$ that are fixed by H_Y and $P_Z, Q_Z \in \mathcal{O}$ that are fixed by H_Z. Order the points so that $P_Y \neq P_0$ and $P_Z \neq P_0$.

Use Lemma 3.39 with U to find $a_1 \in H_{P_0}$, such that $P_Y a_1 = P_\infty$. Then use L to find $a_2 \in H_{P_\infty}$, such that $Q_Y a_1 a_2 = P_0$. Similarly we find $b_1, b_2 \in H$, such that $P_Z b_1 b_2 = P_\infty$ and $Q_Z b_1 b_2 = P_0$. Then $a_1 a_2 (b_1 b_2)^{-1}$ conjugates one Sylow subgroup to the other, and we already have this element as an SLP of length $\mathrm{O}\big((\log(q) \log\log(q))^2\big)$. Hence we can evaluate it on X and obtain $c \in G$ that conjugates $\langle Y \rangle$ to $\langle Z \rangle$. Thus the expected time complexity is as stated. □

THEOREM 4.7. *Assume the small Ree Conjectures and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$. *There exists a Las Vegas algorithm that solves the Sylow generation problem for* $p \mid q^3 + 1$, $p > 2$, *in* $\mathrm{Ree}(q) \cong G \leqslant \mathrm{GL}(d, q)$. *Once constructive recognition has been performed, the expected time complexity of the Sylow generation is* $\mathrm{O}\big((\xi(d) + \log(q) \log\log(q) + d^3) \log\log(q)\big)$ *field operations.*

PROOF. Let $H = \mathrm{Ree}(q)$. The Sylow generation is easy, since we can find an element of pseudo-order $q \pm 3t + 1$ or $(q + 1)/2$, use Proposition 1.4 to obtain an element of order p, then evaluate its SLP on X. By Proposition 2.23, the expected number of random selections, and hence the length of the SLP is $\mathrm{O}\big(\log\log q\big)$, and we need $\mathrm{O}\big(\log(q)\log\log(q)\big)$ field operations to find the order. Then we evaluate the SLP on X using $\mathrm{O}\big(d^3 \log\log(q)\big)$ field operations, so the expected time complexity is as stated. □

This result is due to Mark Stather and is the same as [**Sta06**, Lemma 4.35].

LEMMA 4.8. *Let G be a group and let $k \in \mathbb{Z}$ be such that $|G| = 2^k n$ with n odd. Let $P \leqslant G$ have order 2^{k-1}. Let $\langle P, x\rangle$ and $\langle P, y\rangle$ be Sylow 2-subgroups of G. Then $|xy| = 2^t$ for some $t \in \mathbb{Z}$ if and only if $\langle P, x\rangle = \langle P, y\rangle$. Moreover if $|xy| = 2^t(2s+1)$, then $\langle P, x\rangle^{(yx)^s} = \langle P, y\rangle$*

PROOF. Let $H = \langle P, xy\rangle$. Then H is a subgroup of $\langle P, x, y\rangle$ of index 2, that contains P, but does not contain x or y. Since P has index 2 in both $\langle P, x\rangle$ and $\langle P, y\rangle$ it follows that $xy \in \mathrm{N}_H(P)$. But P is a Sylow 2-subgroup of H so,

$$|xy| = 2^k \Leftrightarrow xy \in P \Leftrightarrow \langle P, x\rangle = \langle P, y\rangle$$

The second statement is an application of Proposition 1.8, modulo H. □

THEOREM 4.9. *Assume the small Ree Conjectures and an oracle for the discrete logarithm problem in $\mathbb{F}_q$. There exists a Las Vegas algorithm that solves the Sylow generation problem for $p = 2$ in $\mathrm{Ree}(q) \cong G \leqslant \mathrm{GL}(d, q)$. Once constructive recognition has been performed, the expected time complexity is $\mathrm{O}\big((\xi(d) + d^3\log(q))\log\log(q) + \log(q)^3 + \chi_D(q)\big)$ field operations.*

PROOF. Let $H = \mathrm{Ree}(q)$. We want to find three commuting involutions in H. Using the first three steps of the algorithm in Section 3.2.2.3, we find an involution $j_1 \in H$ and $\mathrm{C}_H(j_1)' \cong \mathrm{PSL}(2, q)$. Using the notation of that algorithm, we can let the second involution $j_2 \in \mathrm{C}_H(j_1)'$ be $\pi_7(\pi_3(j))$ where j is the second matrix in (3.28).

We then want to find the third involution in the centraliser of j_2 in $\mathrm{C}_H(j_1)'$. In our case this centraliser has structure $(\mathrm{C}_2 \times (\mathrm{C}_2 : A_0)$. Hence its proportion of elements of even order is $3/4$, and $1/2$ of its elements are involutions other than j_2. Using the Bray algorithm we can therefore compute random elements of this centraliser until we find such an involution j_3.

Clearly j_1, j_2, j_3 will all commute. As in the proof of Corollary 3.36, the expected time to find j_1, constructively recognise $\mathrm{C}_H(j_1)'$ and find j_2 is $\mathrm{O}\big(\xi\log\log(q) + {}+\log(q)^3 + \chi_D(q)\big)$ field operations. By the above, the expected time to find j_3 is $\mathrm{O}\big(1\big)$ field operations. The involutions will be found as SLPs, where j_1 and j_3 have length $\mathrm{O}\big(1\big)$, because the generators of $\mathrm{C}_H(j_1)'$ are SLPs of length $\mathrm{O}\big(1\big)$. By Lemma 3.29 and Theorem 1.12, j_2 has length $\mathrm{O}\big(\log(q)\log\log(q)\big)$. Thus we can evaluate them on X using $\mathrm{O}\big(d^3\log(q)\log\log(q)\big)$ field operations, and the expected time complexity is as stated. □

Theorem 4.10. *Assume the small Ree Conjectures and an oracle for the discrete logarithm problem in $\mathbb{F}_q$. There exists a Las Vegas algorithm that solves the Sylow conjugation problem for $p = 2$ in $\mathrm{Ree}(q) \cong G \leqslant \mathrm{GL}(d, q)$. Once constructive recognition has been performed, the expected time complexity is $\mathrm{O}\big(\xi(d) + d^3((\log(q) \log\log(q))^2 + |Y| + |Z|) + \log(q)^3\big)$ field operations.*

Proof. Let $H = \mathrm{Ree}(q)$. Given two Sylow 2-subgroups of G, we use the effective isomorphism to map them to H, using $\mathrm{O}\big(d^3(|Y| + |Z|)\big)$ field operations. The resulting generating sets are $P = \{y_1, y_2, y_3\}$ and $S = \{z_1, z_2, z_3\}$, where both the y_i and z_i are commuting involutions. We may assume that $\langle Y \rangle \neq \langle Z \rangle$ so that $P \neq S$.

The algorithm proceeds as follows:

(1) By Proposition 2.26, we can use the dihedral trick in H and hence find $c_1 \in H$ such that $y_1^{c_1} = z_1$. We then want to conjugate y_2 to z_2 while fixing z_1.
(2) Using the first steps of the algorithm in Section 3.2.2.3, we find $C_1 = \mathrm{C}_H(z_1) \cong \langle z_1 \rangle \times \mathrm{PSL}(2, q)$, and use Theorem 1.13 to determine when we have the whole of C_1. Observe that $y_i^{c_1}, z_i \in C_1$ for all i.
(3) Choose random $g \in C_1'$. If $z_2 g$ has odd order, then $z_2 \in C_1'$. Conversely, if $z_2 \in C_1'$ then $z_2 g$ has odd order with probability $\mathrm{O}(1)$. Similarly, if $z_1 z_2 g$ has odd order, then $z_1 z_2 \in C_1'$, and the probability is the same. Repeat until either z_2 or $z_1 z_2$ has been proved to lie in C_1' and replace z_2 with this element. Do the same procedure with $y_2^{c_1}, y_3^{c_1}, z_3$.
(4) Now $y_2^{c_1}, z_2, y_3^{c_1}, z_3 \in C_1' \cong \mathrm{PSL}(2, q)$, and by [**WP06**, Theorem 13], the dihedral trick works in $\mathrm{PSL}(2, q)$. Hence find $c_2 \in C_1'$ such that $y_2^{c_1 c_2} = z_2$.
(5) Let $|y_3^{c_1 c_2} z_3| = 2^t s$, where s is odd. If $s = 1$ then let $c_3 = 1$ and otherwise let $c_3 = (y_3^{c_1 c_2} z_3)^{(s-1)/2}$. By Lemma 4.8, $P^{c_1 c_2 c_3} = S$.

Finally, we use the effective isomorphism to map $c_1 c_2 c_3$ back to G. As in the proof of Corollary 3.36, C_1 is found using expected $\mathrm{O}\big(\xi(d) + \log(q) \log\log(q)\big)$ field operations, if we let $\varepsilon = \log\log(q)$ in Theorem 1.13. The expected time complexity of the effective isomorphism follows from Theorem 3.62. □

4.3. Big Ree groups

We now consider the Sylow subgroup problems for the Big Ree groups. We will use the notation from Section 2.3, and we will make heavy use of the fact that we can use Theorem 3.79 to constructively recognise the Big Ree groups. However, we can only do this in the natural representation, and hence we will only consider the Sylow subgroup problems in the natural representation. Hence we assume that ${}^2\mathrm{F}_4(q) \cong G \leqslant \mathrm{GL}(26, q)$.

It follows from [**DS99**] that if $g \in G$ then $|g|$ is even, or divides any of the numbers $\big\{q + 1, q - 1, q \pm t + 1, q^2 \pm 1, q^2 - q + 1, q^2 \pm tq + q \pm t + 1\big\}$. Hence we obtain several cases for a Sylow p-subgroup S of G.

(1) $p = 2$. Then S has order q^{12} and lies in $\mathrm{C}_G(j)$ for some involution j of class $2A$. It consists of $\mathrm{O}_2(\mathrm{C}_G(j))$ extended by a Sylow 2-subgroup in a Suzuki group contained in the centraliser.

(2) p divides $o \in \{q-1, q \pm t + 1\}$. Then S has structure $\mathrm{C}_p \times \mathrm{C}_p$. If $G \geqslant H \cong \mathrm{Sz}(q) \times \mathrm{Sz}(q)$, then S is contained in H and consists of Sylow p-subgroups from each Suzuki factor.

(3) p divides $q^2 - q + 1$ or $q^2 \pm tq + q \pm t + 1$. Then S is cyclic of order p, and hence these Sylow subgroups are trivial to find. We do not consider this case.

(4) p divides $q + 1$. We do not consider this case.

THEOREM 4.11. *Assume the Suzuki Conjectures, the Big Ree Conjectures and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$. *There exists a Las Vegas algorithm thats solve the Sylow generation problem for* $p = 2$ *in* ${}^2\mathrm{F}_4(q) \cong \langle X \rangle \leqslant \mathrm{GL}(26, q)$. *Once constructive recognition has been performed, the expected time complexity is* $\mathrm{O}\big(\log(q)(\log\log(q))^2\big)$ *field operations.*

PROOF. Let $G = \langle X \rangle$. We see from the proof of Theorem 3.78 that during the constructive recognition, we find $\mathrm{C}_G(j)$ for some involution $j \in G$ of class $2A$. We also find $\langle Y \rangle, \langle Z \rangle \leqslant \mathrm{C}_G(j)$ such that $\langle Y \rangle = \mathrm{O}_2(\mathrm{C}_G(j))$ and $\langle Z \rangle \cong \mathrm{Sz}(q)$. Moreover, $\langle Z \rangle$ is constructively recognised, and Y, Z are expressed as `SLP`s in X of length $\mathrm{O}\big(\log\log(q)\big)$.

Hence we can apply Theorem 4.1 and obtain a Sylow 2-subgroup $\langle W \rangle$ of $\langle Z \rangle$ using $\mathrm{O}\big(\log(q)(\log\log(q))^2\big)$ field operations. Now $\langle Y, W \rangle$ is a Sylow 2-subgroup of G. □

THEOREM 4.12. *Assume the Suzuki Conjectures, the Big Ree Conjectures and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$. *There exist Las Vegas algorithms that solve the Sylow generation problems for* $p \mid q - 1$ *or* $p \mid q \pm t + 1$ *in* ${}^2\mathrm{F}_4(q) \cong G \leqslant \mathrm{GL}(26, q)$. *Once constructive recognition has been performed, the expected time complexity is* $\mathrm{O}\big((\xi + \log(q)\log\log(q))\log\log(q)\big)$ *field operations.*

PROOF. Let $G = \langle X \rangle$. We see from the proof of Theorem 3.78 that during the constructive recognition, we find $\langle Y_1 \rangle, \langle Y_2 \rangle \cong \mathrm{Sz}(q)$, and they commute, so $\langle Y_1, Y_2 \rangle \cong \mathrm{Sz}(q) \times \mathrm{Sz}(q)$. Moreover, $\langle Y_1 \rangle$ and $\langle Y_2 \rangle$ are constructively recognised, and Y_1, Y_2 are expressed as `SLP`s in X of length $\mathrm{O}\big(\log\log(q)^2\big)$.

Hence we can apply Theorem 4.2 or 4.4 and obtain Sylow p-subgroups of $\langle Y_1 \rangle$ and $\langle Y_2 \rangle$, using $\mathrm{O}\big((\xi + \log(q)\log\log(q))\log\log(q)\big)$ field operations. From the proof of the Theorems, we see that there will be a constant number of generators, which will be expressed as `SLP`s in $\langle Y_i \rangle$ of length $\mathrm{O}\big(\log\log(q)\big)$. Hence we can evaluate them on X using $\mathrm{O}\big((\log\log(q))^3\big)$ field operations. □

CHAPTER 5

Maximal subgroups

We will now describe algorithms for finding and conjugating maximal subgroups of the exceptional groups under consideration. Hence we consider the following problems:

(1) Given $\langle X \rangle \leqslant \mathrm{GL}(d, q)$, such that $\langle X \rangle \cong G$ for some of our exceptional groups G, find representatives $\langle Y_1 \rangle, \ldots, \langle Y_n \rangle$ of the conjugacy classes of (some or all of) the maximal subgroups of G.

(2) Given $\langle X \rangle \leqslant \mathrm{GL}(d, q)$, such that $\langle X \rangle \cong G$ for some of our exceptional groups G, and given $\langle Y \rangle, \langle Z \rangle \leqslant \langle X \rangle$ such that $\langle Y \rangle$ and $\langle Z \rangle$ are conjugate to a specified maximal subgroup of $\langle X \rangle$, find $c \in \langle X \rangle$ such that $\langle Y \rangle^c = \langle Z \rangle$.

It will turn out that because of the results about Sylow subgroup conjugation in Chapter 4, the second problem will most often be easy. The first problem is therefore the difficult one. We will refer to these problems as the "maximal subgroup problems". The first problem is referred to as "maximal subgroup generation" and the second as "maximal subgroup conjugation".

5.1. Suzuki groups

We now consider the maximal subgroup problems for the Suzuki groups. We will use the notation from Section 2.1, and we will make heavy use of the fact that we can use Theorem 3.26 to constructively recognise the Suzuki groups. Hence we assume that G satisfies the assumptions in Section 1.2.7, so $\mathrm{Sz}(q) \cong G \leqslant \mathrm{GL}(d, q)$.

The maximal subgroups are given by Theorem 2.3.

Theorem 5.1. *Assume the Suzuki Conjectures, an oracle for the discrete logarithm problem in $\mathbb{F}_q$ and an oracle for the integer factorisation problem. There exist Las Vegas algorithms that solve the maximal subgroup conjugation in $\mathrm{Sz}(q) \cong G \leqslant \mathrm{GL}(d, q)$. Once constructive recognition has been performed, the expected time complexity is $\mathrm{O}\big(\xi(d)\log(q) + \log(q)^3 + d^3(\log(q)(\log\log(q))^2 + (|Y| + |Z|)\sigma_0(\log(q))) + d^2\log(q)\sigma_0(\log(q)) + \chi_F(4, q)\big)$ field operations.*

Proof. Let $H = \mathrm{Sz}(q)$. In each case, we first use the effective isomorphism to map the subgroups to H using $\mathrm{O}\big(d^3(|Y| + |Z|)\big)$ field operations. Therefore we henceforth assume that $\langle Y \rangle, \langle Z \rangle \leqslant H$.

Observe that all maximal subgroups, except the Suzuki groups over subfields, are the normalisers of corresponding cyclic subgroups or Sylow 2-subgroups. The

Sylow conjugation algorithms can conjugate these cyclic subgroups around, not only the Sylow subgroups that they contain. Moreover, the cyclic subgroups and Sylow 2-subgroups are the derived groups of the corresponding maximal subgroups.

Hence we can obtain probable generators for the cyclic subgroups or Sylow 2-subgroups using $\mathrm{O}\big(\xi(d)\log(q)\big)$ field operations, and we can verify that we have the whole of these subgroups as follows:

(1) For $\mathcal{B}_i$, the generators of the derived group U_i must contain an element of order $q \pm t + 1$.
(2) For $\mathrm{N}_H(\mathcal{H})$, the generators of the derived group $\mathcal{H}$ must contain an element of order $q - 1$.
(3) For $\mathcal{F}\mathcal{H}$, we need not obtain the whole derived group $\mathcal{F}$. It is enough that we obtain a subgroup of the derived group that fixes a unique point of $\mathcal{O}$, but this might require $\mathrm{O}\big(\log(q)\big)$ generators.

Note that in these cases we need the integer factorisation oracle to find the precise order. When we have generators for the cyclic subgroups, we use Theorem 4.1, 4.2 and 4.4 to find conjugating elements for the cyclic subgroups. These elements will also conjugate the maximal subgroups, because they normalise the cyclic subgroups.

Finally, consider the case when $\langle Y \rangle$ and $\langle Z \rangle$ are isomorphic to a Suzuki group over $\mathbb{F}_s < \mathbb{F}_q$. In this case we first use the algorithm in Section 1.2.10.2, to obtain $c_1, c_2 \in \mathrm{GL}(4, q)$ that conjugates $\langle Y \rangle$ and $\langle Z \rangle$ into $\mathrm{GL}(4, s)$. Then we use Theorem 3.17 to find $c_3 \in \mathrm{GL}(4, s)$ that conjugates the Suzuki groups to each other. Hence $c = c_1 c_3 c_2^{-1}$ conjugates $\langle Y \rangle$ to $\langle Z \rangle$, and therefore it normalises H. However, c does not necessarily lie in H, but only in $\mathrm{N}_{\mathrm{GL}(4,q)}(H) \cong H{:}\mathbb{F}_q$, since neither c_1 nor c_2 are guaranteed to lie in H. Therefore $c = (\gamma I_4) g$ where $g \in H$ and $\gamma \in \mathbb{F}_q$. We can find γ by calculating the determinant and taking its (unique) 4th root, so we can divide by the scalar matrix, and we then end up with g, which also conjugates $\langle Y \rangle$ to $\langle Z \rangle$.

Finally we use the effective isomorphism to map g to G. The expected time complexity follows from Theorem 4.1, 4.2, 4.4 and 3.17. □

Lemma 5.2. *If $g, h \in G = \mathrm{Sz}(q)$ satisfy that $|g| = 2$, $|h| = 4$, $|gh| = 4$ and $\left|gh^2\right| = q \pm t + 1$, then $\langle g, h \rangle \cong \mathrm{C}_{q \pm t+1} : \mathrm{C}_4$ and hence is a maximal subgroup of G.*

Proof. Clearly, $\langle g, h \rangle$ is an image in G of the group $H = \left\langle x, y \mid x^2, y^4, (xy)^4 \right\rangle \cong \mathbb{Z}^2 : \mathrm{C}_4$. Since H is soluble and gh^2 has the specified order, $\langle g, h \rangle$ must be one of the $\mathcal{B}_i$ from Theorem 2.3. □

Theorem 5.3. *Assume the Suzuki Conjectures, an oracle for the discrete logarithm problem in $\mathbb{F}_q$ and an oracle for the integer factorisation problem. There exist Las Vegas algorithms that solve the maximal subgroup generation in $\mathrm{Sz}(q) \cong G \leqslant \mathrm{GL}(d, q)$. Once constructive recognition has been performed, the expected time complexity is $\mathrm{O}\big(\xi(d)(\sigma_0(\log(q)) + \log\log(q)) + \chi_F(4, q)\log\log(q) + \sigma_0(\log(q))(\log(q)^3 + d^3 \log(q)(\log\log(q))^2)\big)$ field operations.*

PROOF. Let $H = \mathrm{Sz}(q)$. Using the effective isomorphism, it is sufficient to obtain generators for the maximal subgroups in H.

Let $\alpha \in \mathbb{F}_q$ be a primitive element. Clearly $\mathcal{FH} = \langle M'(\alpha), S(1,0) \rangle$, and $\mathrm{N}_H(\mathcal{H}) = \langle M'(\alpha), T \rangle$. For each $e > 0$ such that $2e+1 \mid 2m+1$, we have $\mathbb{F}_s < \mathbb{F}_q$ where $s = 2^{2e+1}$. Hence $s - 1 \mid q - 1$ and $\mathrm{Sz}(s) = \left\langle T, S(1,0), M'(\alpha)^{(q-1)/(s-1)} \right\rangle$.

The difficult case is therefore $\mathcal{B}_1$ and $\mathcal{B}_2$. We want to use Lemma 5.2, with T and $S(a,b)$ playing the roles of x and y. Hence we proceed as follows:

(1) Choose random $g \in H$ of order $q \pm t + 1$. Note that we need the integer factorisation oracle, since we need the precise order of g. Let $\lambda = \mathrm{Tr}(g)$.
(2) Let a, b be indeterminates and consider the equations $\mathrm{Tr}(TS(a,b)) = 0$ and $\mathrm{Tr}(TS(a,b)^2) = \lambda$. If we can find solutions for a, b, then by Proposition 2.9, $|TS(a,b)| = 4$ with high probability, and Proposition 2.8 implies that $\left|TS(a,b)^2\right| = q \pm t + 1$.
(3) The second equation implies $a = \lambda^{t+2}$, and

$$\begin{aligned} \mathrm{Tr}(TS(a,b)) = a^t + a^{t+2} + ab + b^t = 0 \Leftrightarrow \\ a^2 + a^{2t+2} + a^t b^t + b^2 = 0 \Rightarrow \\ b^2 + a^{t+1} b + a^2 + a^{2t} = 0 \end{aligned} \tag{5.1}$$

where the third equation is a^t times the first added to the second.
(4) The quadratic equation has solutions $b_1 = a^{t+1} \sum_{i=1}^{m+1} a^{-2^i}$ and $b_2 = b_1 + a^{t+1}$, which both give the value $a^{s+2}(1 + \sum_{i=0}^{2m} a^{-2^i})$ of $\mathrm{Tr}(TS(a,b))$. Hence repeat with another g if $\sum_{i=0}^{2m} a^{-2^i} \neq 1$, which happens with probability $1/2$.
(5) Lemma 5.2 now implies that $\langle T, S(a,b) \rangle$ is $\mathcal{B}_1$ or $\mathcal{B}_2$.

Finally, we see that we have $\mathrm{O}\big(\sigma_0(\log(q))\big)$ generators, which we map back to G using the effective isomorphism. Hence the expected time complexity is as stated. □

5.2. Small Ree groups

We now consider the maximal subgroup problems for the small Ree groups. We will use the notation from Section 2.1. We will make heavy use of the fact that we can use Theorem 3.62 to constructively recognise the small Ree groups. Hence we assume that G satisfies the assumptions in Section 1.2.7, so $\mathrm{Ree}(q) \cong G \leqslant \mathrm{GL}(d,q)$.

The maximal subgroups are given by Proposition 2.20.

THEOREM 5.4. *Assume the small Ree Conjectures and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$. *There exist Las Vegas algorithms that solve the maximal subgroup conjugation in* $\mathrm{Ree}(q) \cong G \leqslant \mathrm{GL}(d,q)$ *for the point stabiliser, the involution centraliser and Ree groups over subfields. Once constructive recognition has been performed, the expected time complexity is* $\mathrm{O}\big(\xi(d)\log(q) + \log(q)^3 + \chi_D(q)\log\log(q) + d^3((|Y|+|Z|)\sigma_0(\log(q)) + (\log(q)\log\log(q))^2) + d^2\log(q)\sigma_0(\log(q))\big)$ *field operations.*

PROOF. Let $H = \mathrm{Ree}(q)$. In each case, we first use the effective isomorphism to map the subgroups to H using $\mathrm{O}\big(d^3(|Y| + |Z|)\big)$ field operations. Therefore we henceforth assume that $\langle Y \rangle, \langle Z \rangle \leqslant H$.

Observe that the point stabiliser is the normaliser of a Sylow 3-subgroup, which is the derived group of the point stabiliser. We can therefore obtain probable generators for the Sylow 3-subgroup using $\mathrm{O}\big(\xi(d)\log(q)\big)$ field operations. We only need enough generators so that they generate a subgroup of the derived group that fixes a unique point of $\mathcal{O}$, and this we can easily verify using the MeatAxe. When we have generators for this subgroup, we use Theorem 4.5 to find a conjugating element. This element will also conjugate the maximal subgroup, because it normalises the Sylow subgroup.

For the involution centraliser, we choose random elements of $\langle Y \rangle$. Since $\langle Y \rangle \cong \langle y \rangle \times \mathrm{PSL}(2, q)$ for some involution y, with probability $\mathrm{O}\big(1\big)$ we will obtain an element of even order that powers up to y. We can check that we obtain y since it is the unique involution that is centralised by $\langle Y \rangle$ (and therefore by Y). Hence we can find the involutions y and z that are centralised by $\langle Y \rangle$ and $\langle Z \rangle$. By Proposition 2.26 we can use the dihedral trick to find $c \in H$ that conjugates y to z, using $\mathrm{O}\big(\log(q)\log\log(q)\big)$ field operations. Since $\langle Y \rangle$ and $\langle Z \rangle$ centralise these, it follows that $\langle Y \rangle^c = \langle Z \rangle$.

Finally, consider the case when $\langle Y \rangle$ and $\langle Z \rangle$ are isomorphic to a Ree group over $\mathbb{F}_s < \mathbb{F}_q$. In this case we first use the algorithm in Section 1.2.10.2, to obtain $c_1, c_2 \in \mathrm{GL}(7, q)$ that conjugates $\langle Y \rangle$ and $\langle Z \rangle$ into $\mathrm{GL}(7, s)$. Then we use Theorem 3.47 to find $c \in \mathrm{GL}(7, s)$ that conjugates the resulting Ree groups to each other. Hence $c_1 c c_2^{-1}$ conjugates $\langle Y \rangle$ to $\langle Z \rangle$, and hence normalises H. However, it does not necessarily lie in H, but only in $\mathrm{N}_{\mathrm{GL}(7,q)}(H) \cong H{:}\mathbb{F}_q$, since neither c_1 nor c_2 has to lie in H. Therefore it is of the form $(\gamma I_7)g$, where $g \in H$ and $\gamma \in \mathbb{F}_q$. We can find γ by calculating the determinant and taking the (unique) 7th root, so we can divide by the scalar matrix, and we then end up with g, that also conjugates $\langle Y \rangle$ to $\langle Z \rangle$.

Finally we use the effective isomorphism to map the conjugating element back to G. The expected time complexity follows from Theorem 4.5, 3.47 and 3.62. ☐

LEMMA 5.5. *If $g, h \in G = \mathrm{Ree}(q)$ satisfy that $|g| = 2$, $|h| = 3$, $|gh| = 6$ and $|[g, h]| = q \pm 3t + 1$ or $|[g, h]| = (q + 1)/2$, then $\langle g, h \rangle \cong \mathrm{C}_{q \pm 3t+1} : \mathrm{C}_6$ or $\langle g, h \rangle \cong (\mathrm{C}_2 \times \mathrm{C}_2 \times \mathrm{C}_{(q+1)/4})) : \mathrm{C}_6$ and hence is a maximal subgroup of G.*

PROOF. Clearly, $\langle g, h \rangle$ is an image in G of the group $H = \big\langle x, y \mid x^2, y^3, (xy)^6 \big\rangle \cong \mathbb{Z}^2 : \mathrm{C}_6$. Since H is soluble and $[g, h]$ has the specified order, $\langle g, h \rangle$ must be one of the $\mathrm{N}_G(A_i)$ from Proposition 2.20. ☐

LEMMA 5.6. *Let $G = \mathrm{Ree}(q)$. For each $k \in \{q \pm 3t + 1, (q + 1)/2\}$, there exist $x, y \in G = \mathrm{Ree}(q)$ such that $|x| = 2$, $|y| = 3$, $|xy| = 6$ and $|[x, y]| = k$.*

PROOF. Consider the case $k = q \pm 3t + 1$. There exists $H \leqslant G$ such that $H = \langle a \rangle \langle b \rangle \cong \mathrm{C}_k : \mathrm{C}_6$ with $|a| = k$ and $|b| = 6$. Observe that $H' = \langle a \rangle$. If $x = a^{-i} b^3$

and $y = a^i[b^3, a^i]^{-1}b^{-2}$ then $xy = b$, and $[x, y] = a^j$ for some j, depending on i. Clearly we can choose i such that $\gcd(j, k) = 1$, and hence $|[x, y]| = k$.

The other case is analogous. $\square$

LEMMA 5.7. *Let $G = \operatorname{Ree}(q)$ and $\upsilon = h(-1)\Upsilon$. For each $k \in \{q \pm 3t + 1, (q + 1)/2\}$, there exist $a, b \in \mathbb{F}_q$ such that $|\Upsilon S(0, a, b)| = 6$ and $|[\Upsilon, S(0, a, b)]| = k$ or $|\upsilon S(0, a, b)| = 6$ and $|[\upsilon, S(0, a, b)]| = k$.*

PROOF. Let (x, y) be as in Lemma 5.6. It is sufficient to prove that there exists $a, b \in \mathbb{F}_q$ such that (x, y) is conjugate to $(\Upsilon, S(0, a, b))$ or $(\upsilon, S(0, a, b))$.

Since y has order 3, it fixes a point P. Also, G is doubly transitive so there exists $c_1 \in G$ such that $P_\infty c_1 = P$. Then $y^{c_1} = S(0, a, b)$ for some $a, b \in \mathbb{F}_q$. Observe that x^{c_1} does not fix P_∞, since otherwise $|[x, y]| = 3$.

Now $P_\infty x^{c_1} = R$ for some point R, and G_{P_∞} acts transitively on the points other than P_∞, so there exists $c_2 \in G_{P_\infty}$ such that $Rc_2 = P_0$. Then $x^{c_1c_2}$ interchanges P_0 and P_∞, and so does Υ. Hence $x^{c_1c_2}\Upsilon^{-1} \in \langle h(\lambda)\rangle$, so $x^{c_1c_2} = h(\lambda)^i\Upsilon$, for some $0 \leqslant i < q - 1$.

Let $k \equiv i/2 \pmod{(q-1)/2}$ such that $0 \leqslant k < q-1$, and let $c_3 = h(\lambda)^k$. There are two possible values for k, either $2k = i$ or $2k = i + (q-1)/2$. In the former case $x^{c_1c_2c_3} = \Upsilon$, and in the latter case $x^{c_1c_2c_3} = h(\lambda)^{(q-1)/2}\Upsilon = \upsilon$. $\square$

CONJECTURE 5.8. *Let $q = 3^{2m+1}$ for some $m > 0$ and let $t = 3^m$. For every $a \in \mathbb{F}_q^\times$, the ideals in $\mathbb{F}_q[b_1, c_1, b_2, c_2]$ generated by the following systems are zero-dimensional:*

$$\begin{cases} b_2^2 + b_1b_2 + c_2^2 = 0 \\ b_1^2 + b_2^3b_1 + c_1^2 = 0 \\ 1 - a - b_1^2 + b_2^4 + b_1^2b_2^2 - c_1^2 - b_1b_2c_2^2 + c_2^4 - b_2^2c_2^2 = 0 \\ 1 - a^{3t} - b_2^6 + b_1^4 + b_2^6b_1^2 - c_2^6 - c_1^2b_2^3b_1 + c_1^4 - b_1^2c_1^2 = 0 \end{cases} \tag{5.2}$$

$$\begin{cases} b_2^2 - b_1b_2 - c_2^2 = 0 \\ b_1^2 - b_2^3b_1 - c_1^2 = 0 \\ 1 - a - b_1^2 + b_2^4 + b_1^2b_2^2 + c_1^2 - b_1b_2c_2^2 + c_2^4 + b_2^2c_2^2 = 0 \\ 1 - a^{3t} - b_2^6 + b_1^4 + b_2^6b_1^2 + c_2^6 - c_1^2b_2^3b_1 + c_1^4 + b_1^2c_1^2 = 0 \end{cases} \tag{5.3}$$

THEOREM 5.9. *Assume the small Ree Conjectures, Conjecture 5.8, an oracle for the discrete logarithm problem in $\mathbb{F}_q$ and an oracle for the integer factorisation problem. There exist Las Vegas algorithms that solve the maximal subgroup generation in $\operatorname{Ree}(q) \cong G \leqslant \operatorname{GL}(d, q)$. Once constructive recognition has been performed, the expected time complexity is $\mathrm{O}\big(\xi(d)(\sigma_0(\log(q)) + \log\log(q)) + \chi_F(7, q)\log\log(q) + \sigma_0(\log(q))(d^3(\log(q)\log\log(q))^2 + \log(q)^3)\big)$ field operations.*

PROOF. Let $H = \operatorname{Ree}(q)$. Using the effective isomorphism, it is sufficient to obtain generators for the maximal subgroups in H.

Let $\alpha \in \mathbb{F}_q$ be a primitive element. Clearly $U(q)H(q) = \langle h(\alpha), S(1, 0, 0)\rangle$, and by following a procedure similar to [**Wil06**], we see that $\mathrm{C}_H(h(-1)) = \langle h(\alpha), \Upsilon, S(0, 1, 0)\rangle$.

For each $e > 0$ such that $2e+1 \mid 2m+1$, we have $\mathbb{F}_s < \mathbb{F}_q$ where $s = 3^{2e+1}$. Hence $s-1 \mid q-1$ and $\operatorname{Ree}(s) = \left\langle \Upsilon, S(1,0,0), h(\alpha)^{(q-1)/(s-1)} \right\rangle$.

The difficult cases are therefore the $\mathrm{N}_H(A_i)$. In the light of Lemma 5.7, we can proceed as follows:

(1) Choose random $g \in H$ of order $q \pm 3t + 1$ or $(q+1)/2$, corresponding to the order of A_i. By Corollary 2.24 this is done using expected $\mathrm{O}\big((\xi(d) + \log(q)\log\log(q) + \chi_F(7,q))\log\log(q)\big)$ field operations. Note that we need the integer factorisation oracle since we need the precise order in this case.

(2) Introduce indeterminates x, y and consider the equations $\operatorname{Tr}(\Upsilon S(0,x,y)) = 1$ and $\operatorname{Tr}([\Upsilon, S(0,x,y)]) = \operatorname{Tr}(g)$, or similarly with υ instead of Υ. We want to find solutions a, b for x, y as in the Lemma. By Proposition 2.25, the trace determines the order in these cases, which leads us to consider these equations.

(3) Elements of order 6 have trace 1. Hence we obtain equations in x, y:

$$\begin{aligned} \operatorname{Tr}(\Upsilon S(0,x,y)) &= 1 \\ (\operatorname{Tr}(\Upsilon S(0,x,y)))^{3t} &= 1 \\ \operatorname{Tr}\,[\Upsilon, S(0,x,y)] &= \operatorname{Tr} g \\ (\operatorname{Tr}\,[\Upsilon, S(0,x,y)])^{3t} &= (\operatorname{Tr} g)^{3t} \end{aligned} \tag{5.4}$$

By letting $b_1 = x$, $b_2 = x^t$, $c_1 = y$, $c_2 = y^t$, this is precisely one of the systems in Conjecture 5.8, and thus we can use Theorem 1.3 to find all solutions using $\mathrm{O}\big(\log q\big)$ field operations.

(4) By Lemma 5.7, there will be solutions a, b. By Lemma 5.5, the resulting $S(0,a,b)$ generates $\mathrm{N}_H(A_i)$ together with Υ or υ.

Finally, we see that we have $\mathrm{O}\big(\sigma_0(\log(q))\big)$ generators, which we map back to G using the effective isomorphism. Hence the expected time complexity is as stated. □

5.3. Big Ree groups

We now consider the maximal subgroup problems for the Big Ree groups. We will use the notation from Section 2.3, and we will make heavy use of the fact that we can use Theorem 3.79 to constructively recognise the Big Ree groups. However, we can only do this in the natural representation, and hence we will only consider the maximal subgroup problems in the natural representation.

The maximal subgroups are listed in [**Mal91**], but we will only generate some of them.

THEOREM 5.10. *Assume Conjectures* 3.4 *and* 3.64*, and an oracle for the discrete logarithm problem in* $\mathbb{F}_q$*. There exist Las Vegas algorithms that, given* ${}^2\mathrm{F}_4(q) \cong \langle X \rangle \leqslant \mathrm{GL}(26,q)$ *finds* $\langle Y_1 \rangle, \langle Y_2 \rangle, \langle Y_3 \rangle, \langle Y_4 \rangle \leqslant \langle X \rangle$ *such that* $\langle Y_1 \rangle$ *and* $\langle Y_2 \rangle$ *are the two maximal parabolics,* $\langle Y_3 \rangle \cong \mathrm{Sz}(q) \wr \mathrm{C}_2$ *and* $\langle Y_4 \rangle \cong \mathrm{Sp}(4,q){:}\,\mathrm{C}_2$*. Once constructive* *recognition has been performed, the expected time complexity is* $\mathrm{O}(1)$ *field operations.*

PROOF. Let $H = {}^2\mathrm{F}_4(q)$, and let $\varphi : \langle X \rangle \to H$ be the effective isomorphism. Generators for the subgroups are given in Proposition 2.36 and Proposition 2.44. Since they all have constant size, and pre-images of φ can be computed in $\mathrm{O}(1)$ field operations, we obtain Y_1, Y_2, Y_3, Y_4 in $\mathrm{O}(1)$ field operations. $\square$

CHAPTER 6

Implementation and performance

All the algorithms that have been described have been implemented in the computer algebra system MAGMA. As we remarked in Section 1.2.11, the implementation has been a major part of the work and has heavily influenced the nature of the theoretical results. The algorithms have been developed with the implementation in mind from the start, and hence only algorithms that can be implemented and executed on current hardware have been developed.

This chapter is concerned with the implementation, and we will provide experimental evidence of the fact that the algorithms indeed are efficient in practice. The evidence will be in the form of benchmark results, tables and diagrams. This chapter is therefore not so much about mathematics, but rather about software engineering or computer science.

The implementations were developed during a time span of 2-3 years, using MAGMA versions 2.11-5 and above. The benchmark results have all been produced using version 2.13-12, Intel64 flavour, statically linked.

The hardware used during the benchmark was a PC, with an Intel Xeon CPU, clocked at 2.80 GHz, and with 1 GB of RAM. The operating system was Debian GNU/Linux Sarge, with kernel version 2.6.8-12-em64t-p4-smp.

The implementations used the existing MAGMA implementations of the algorithms described in Chapter 1. These include implementations of the following:

- A discrete log algorithm, in particular Coppersmith's algorithm. The implementation is described in [**Tho01**].
- The product replacement algorithm.
- The algorithm from Theorem 1.1.
- The Order algorithm, for calculating the order (or pseudo-order) of a matrix.
- The black box naming algorithm from [**BKPS02**].
- The algorithms from Theorems 1.12 and 1.13.
- The three algorithms from Section 1.2.10.

We used MATLAB and [**R D05**] to produce the figures. In every case, the benchmark of an algorithm was performed by running the algorithm a number of times for each field with size q lying in some specified range. We then recorded the time t (in seconds) taken by the algorithm. However, to be able to compare the benchmark results with our stated time complexities, we want to display not the time in seconds, but the number of field operations. Moreover, the input size is polynomial in $\log(q)$ and not q. Hence we first recorded the time t_k for k multiplications in $\mathbb{F}_q$ and display t/t_k against $\log(q)$. Of course, $k = 1$ is in principle enough, but we chose $k = 10^6$ to achieve a scaling of the graph.

In Magma, Zech logarithms are used for the finite field arithmetic in $\mathbb{F}_q$ if $q \leqslant q_Z$ (for some q_Z, at present $q_Z \approx 2^{20}$), and for larger fields Magma represents the field elements as polynomials over the largest subfield that is smaller than q_Z, rather than over the prime field. The reason is that the polynomials then have fewer terms, and hence the field arithmetic is faster, than if the polynomials have coefficients in the prime field. Since the subfield is small enough to use Zech logarithms, arithmetic in the subfield is not much slower than in the prime field.

Now consider a field of size p^n. If n is prime, there are no subfields except the prime field, and the field arithmetic will be slow, but if n has a divisor only slightly smaller than q_Z, then the field arithmetic will be fast. Since it might happen that n is prime but $n + 1$ is divisible by q_Z, we will get jumps in our benchmark figures, unless we turn off all these optimisations in Magma. Therefore, this is what we do, and hence any jumps are the result of group theoretical properties, the discrete log and factorisation oracles, and the probabilistic nature of the algorithms.

All the non-constructive recognition algorithms that we have presented, in Sections 3.1.1, 3.2.1 and 3.3.1 are extremely fast, and in practice constant time for the field sizes under consideration. Hence we do not display any benchmarks of them.

6.1. Suzuki groups

In the cases of the Suzuki groups, the field size is always $q = 2^{2m+1}$ for some $m > 0$. Hence we display the time against m.

In Figure 6.1 we show the benchmark of the first two steps of the algorithm in Theorem 3.12, where a stabiliser in $\mathrm{Sz}(q)$ of a point of $\mathcal{O}$ is computed. For each field size, we made 100 runs of the algorithms, using random generating sets and random points.

Notice that the time is very much dominated by the discrete logarithm computations. The oscillations in the discrete log timings have number theoretic reasons. When $m = 52$, the factorisation of $q - 1$ contains no prime with more than 6 decimal digits, hence discrete log is very fast. On the other hand, when $m = 64$, the factorisation of $q - 1$ contains a prime with 26 decimal digits.

In Figure 6.2 we show the benchmark of the algorithm in Theorem 3.17. For each field size, we made 100 runs of the algorithms, using random generating sets of random conjugates of $\mathrm{Sz}(q)$.

The time complexity stated in the theorem suggests that the graph should be slightly worse than linear. Figure 6.2 clearly supports this. The minor oscillations can have at least two reasons. The algorithm is randomised, and the core of the algorithm is to find an element of order $q - 1$ by random search. The proportion of such elements is $\phi(q - 1)/(2(q - 1))$ which oscillates slightly when q increases.

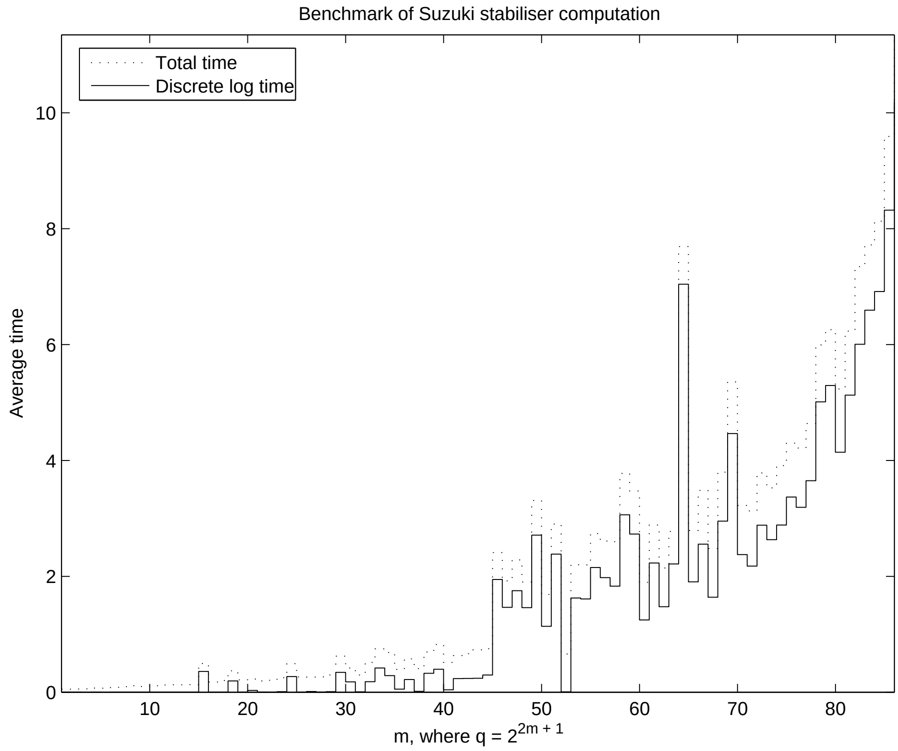

FIGURE 6.1. Benchmark of Suzuki stabiliser computation

We do not include graphs of the tensor decomposition algorithms for the Suzuki groups. The reason is that at the present time, they can only be executed on a small number of inputs (certainly not more than $d \in \{16, 64, 256\}$ and $q \in \{8, 32, 128\}$) before running out of memory. Hence there is not much of a graph to display.

6.2. Small Ree groups

In the cases of the small Ree groups, the field size is always $q = 3^{2m+1}$ for some $m > 0$. Hence we display the time against m.

In Figure 6.3 we show the benchmark of the algorithm in Theorem 3.47. For each field size, we made 100 runs of the algorithms, using random generating sets of random conjugates of $\mathrm{Ree}(q)$. As can be seen from the proof of the Theorem, the algorithm involves many ingredients: discrete logarithms, `SLP` evaluations, $\mathrm{SL}(2, q)$ recognition. To avoid making the graph unreadable, we avoid displaying the timings for these various steps, and only display the total time. The graph still has jumps, for reasons similar as with the Suzuki groups.

We do not include graphs of the tensor decomposition algorithms for the small Ree groups. The reason is that at the present time, they can only be executed on a small number of inputs (certainly not more than $d \in \{49, 189, 729\}$ and $q \in \{27, 243\}$) before running out of memory. Hence there is not much of a graph to display.

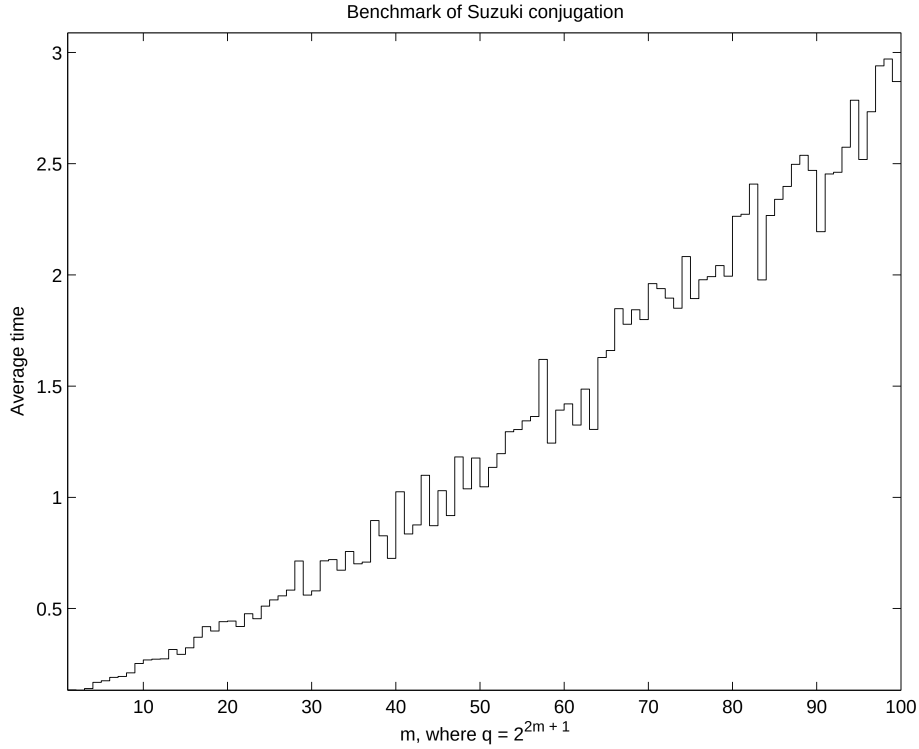

FIGURE 6.2. Benchmark of Suzuki conjugation

6.3. Big Ree groups

In the cases of the Big Ree groups, the field size is always $q = 2^{2m+1}$ for some $m > 0$. Hence we display the time against m.

In Figure 6.4 we show the benchmark of the algorithm in Theorem 3.79. This involves all the results presented for the Big Ree groups.

For each field size, we made 100 runs of the algorithms, using random generating sets of random conjugates of ${}^2\mathrm{F}_4(q)$. As can be seen from the proof of the Theorem, the algorithm involves many ingredients: discrete logarithms, `SLP` evaluations, $\mathrm{Sz}(q)$ recognition, $\mathrm{SL}(2, q)$ recognition. To avoid making the graph unreadable, we avoid displaying the timings for these various steps, and only display the total time.

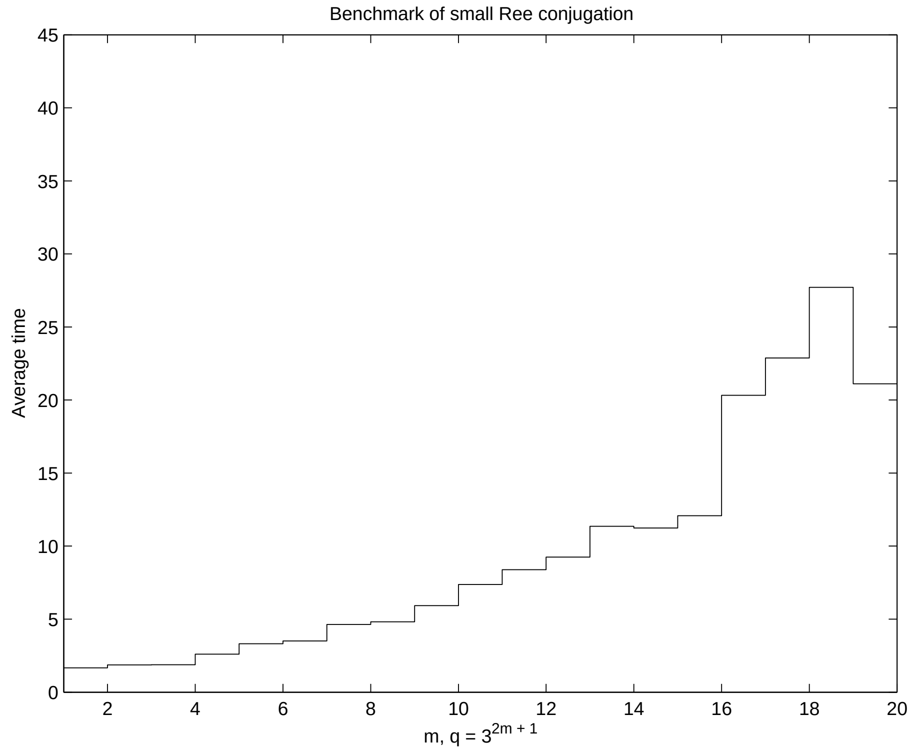

FIGURE 6.3. Benchmark of small Ree conjugation

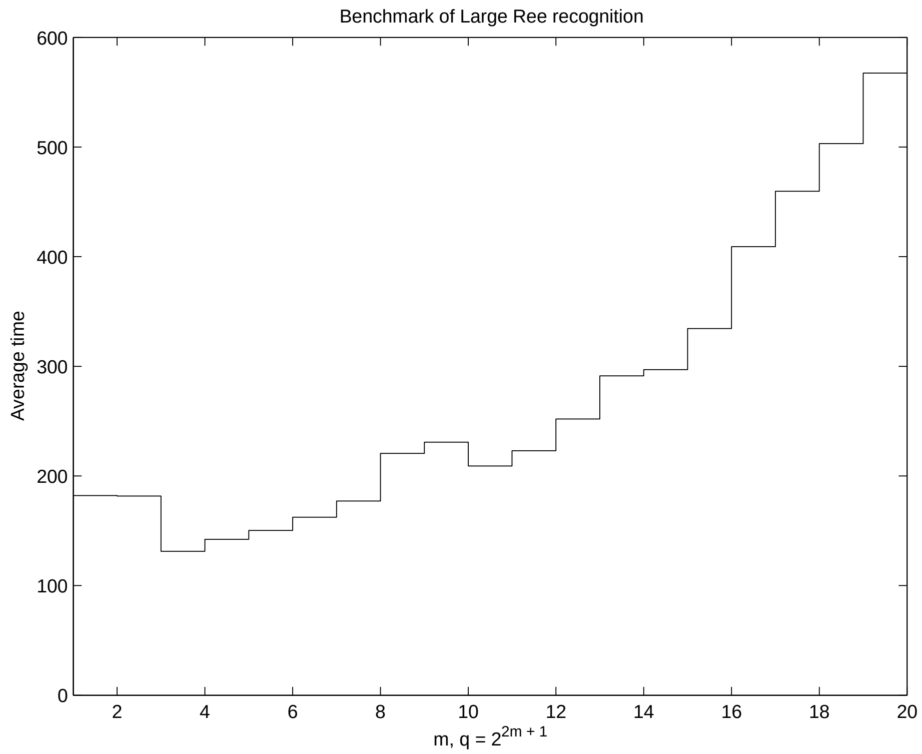

FIGURE 6.4. Benchmark of Large Ree conjugation

Bibliography

[Asc84] M. Aschbacher, *On the maximal subgroups of the finite classical groups*, Invent. Math. **76** (1984), no. 3, 469–514. MR MR746539 (86a:20054)

[Bab91] László Babai, *Local expansion of vertex-transitive graphs and random generation in finite groups*, STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing (New York, NY, USA), ACM Press, 1991, pp. 164–174.

[BB99] László Babai and Robert Beals, *A polynomial-time theory of black box groups. I*, Groups St. Andrews 1997 in Bath, I, London Math. Soc. Lecture Note Ser., vol. 260, Cambridge Univ. Press, Cambridge, 1999, pp. 30–64. MR MR1676609 (2000h:20089)

[BB07] John Bray and Henrik Bäärnhielm, *Standard generators for the Suzuki groups*, preprint, 2007.

[BK01] Peter A. Brooksbank and William M. Kantor, *On constructive recognition of a black box* $\mathrm{PSL}(d, q)$, Groups and computation, III (Columbus, OH, 1999), Ohio State Univ. Math. Res. Inst. Publ., vol. 8, de Gruyter, Berlin, 2001, pp. 95–111. MR MR1829473 (2002m:20078)

[BK06] ———, *Fast constructive recognition of black box orthogonal groups*, J. Algebra **300** (2006), no. 1, 256–288. MR MR2228648

[BKPS02] László Babai, William M. Kantor, Péter P. Pálfy, and Ákos Seress, *Black-box recognition of finite simple groups of Lie type by statistics of element orders*, J. Group Theory **5** (2002), no. 4, 383–401. MR MR1931364 (2003i:20022)

[Bra00] John N. Bray, *An improved method for generating the centralizer of an involution*, Arch. Math. (Basel) **74** (2000), no. 4, 241–245. MR MR1742633 (2001c:20063)

[Bra07] ———, *Presentations of the Suzuki groups*, preprint, 2007.

[Bro03a] Peter A. Brooksbank, *Constructive recognition of classical groups in their natural representation*, J. Symbolic Comput. **35** (2003), no. 2, 195–239. MR MR1958954 (2004c:20082)

[Bro03b] ———, *Fast constructive recognition of black-box unitary groups*, LMS J. Comput. Math. **6** (2003), 162–197 (electronic). MR MR2051584 (2005e:20075)

[BS84] László Babai and Endre Szemerédi, *On the complexity of matrix group problems, I*, Proc. 25th IEEE Sympos. Foundations Comp. Sci., 1984, pp. 229–240.

[CKSU05] Henry Cohn, Robert Kleinberg, Balazs Szegedy, and Christopher Umans, *Group-theoretic algorithms for matrix multiplication*, FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (Washington, DC, USA), IEEE Computer Society, 2005, pp. 379–388.

[CLG97] Frank Celler and C. R. Leedham-Green, *Calculating the order of an invertible matrix*, Groups and computation, II (New Brunswick, NJ, 1995), DIMACS Ser. Discrete Math. Theoret. Comput. Sci., vol. 28, Amer. Math. Soc., Providence, RI, 1997, pp. 55–60. MR MR1444130 (98g:20001)

[CLG01] Marston Conder and Charles R. Leedham-Green, *Fast recognition of classical groups over large fields*, Groups and computation, III (Columbus, OH, 1999), Ohio State Univ. Math. Res. Inst. Publ., vol. 8, de Gruyter, Berlin, 2001, pp. 113–121. MR MR1829474 (2002g:20001)

[CLGM+95] Frank Celler, Charles R. Leedham-Green, Scott H. Murray, Alice C. Niemeyer, and E. A. O'Brien, *Generating random elements of a finite group*, Comm. Algebra **23** (1995), no. 13, 4931–4948. MR MR1356111 (96h:20115)

[CLGO06] M. D. E. Conder, C. R. Leedham-Green, and E. A. O'Brien, *Constructive recognition of* $\mathrm{PSL}(2, q)$, Trans. Amer. Math. Soc. **358** (2006), no. 3, 1203–1221 (electronic). MR MR2187651 (2006j:20017)

[Cop84] Don Coppersmith, *Fast evaluation of logarithms in fields of characteristic two*, IEEE Trans. Inform. Theory **30** (1984), no. 4, 587–594. MR MR755785 (85h:65041)

[CR06] Charles W. Curtis and Irving Reiner, *Representation theory of finite groups and associative algebras*, AMS Chelsea Publishing, Providence, RI, 2006, Reprint of the 1962 original. MR MR2215618 (2006m:16001)

[CW90] Don Coppersmith and Shmuel Winograd, *Matrix multiplication via arithmetic progressions*, J. Symbolic Comput. **9** (1990), no. 3, 251–280. MR MR1056627 (91i:68058)

[DS99] Huiwen Deng and Wujie Shi, *The characterization of Ree groups* $^2F_4(q)$ *by their element orders*, J. Algebra **217** (1999), no. 1, 180–187. MR MR1700483 (2000i:20038)

[Gie95] Mark Giesbrecht, *Nearly optimal algorithms for canonical matrix forms*, SIAM J. Comput. **24** (1995), no. 5, 948–969. MR MR1350753 (96f:65180)

[GL01] Robert M. Guralnick and Frank Lübeck, *On* p*-singular elements in Chevalley groups in characteristic* p, Groups and computation, III (Columbus, OH, 1999), Ohio State Univ. Math. Res. Inst. Publ., vol. 8, de Gruyter, Berlin, 2001, pp. 169–182. MR MR1829478 (2002d:20074)

[GLGO06] S. P. Glasby, C. R. Leedham-Green, and E. A. O'Brien, *Writing projective representations over subfields*, J. Algebra **295** (2006), no. 1, 51–61. MR MR2188850 (2006h:20002)

[GM93] Daniel M. Gordon and Kevin S. McCurley, *Massively parallel computation of discrete logarithms*, CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology (London, UK), Springer-Verlag, 1993, pp. 312–323.

[HB82] Bertram Huppert and Norman Blackburn, *Finite groups. III*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 243, Springer-Verlag, Berlin, 1982. MR MR662826 (84i:20001b)

[HEO05] Derek F. Holt, Bettina Eick, and Eamonn A. O'Brien, *Handbook of computational group theory*, Discrete Mathematics and its Applications (Boca Raton), Chapman & Hall/CRC, Boca Raton, FL, 2005. MR MR2129747 (2006f:20001)

[HLO+06] P. E. Holmes, S.A. Linton, E. A. O'Brien, A. J. E. Ryba, and R. A. Wilson, *Constructive membership testing in black-box groups*, preprint, 2006.

[HR94] Derek F. Holt and Sarah Rees, *Testing modules for irreducibility*, J. Austral. Math. Soc. Ser. A **57** (1994), no. 1, 1–16. MR MR1279282 (95e:20023)

[HRD05] Derek F. Holt and Colva M. Roney-Dougal, *Constructing maximal subgroups of classical groups*, LMS J. Comput. Math. **8** (2005), 46–79 (electronic). MR MR2123130 (2005k:20027)

[HW79] G. H. Hardy and E. M. Wright, *An introduction to the theory of numbers*, fifth ed., The Clarendon Press Oxford University Press, New York, 1979. MR MR568909 (81i:10002)

[IKS95] I. M. Isaacs, W. M. Kantor, and N. Spaltenstein, *On the probability that a group element is* p*-singular*, J. Algebra **176** (1995), no. 1, 139–181. MR MR1345299 (96f:20035)

[IL00] Gábor Ivanyos and Klaus Lux, *Treating the exceptional cases of the MeatAxe*, Experiment. Math. **9** (2000), no. 3, 373–381. MR MR1795309 (2001j:16067)

[Kle88] Peter B. Kleidman, *The maximal subgroups of the Chevalley groups $G_2(q)$ with q odd, the Ree groups ${}^2G_2(q)$, and their automorphism groups*, J. Algebra **117** (1988), no. 1, 30–71. MR MR955589 (89j:20055)

[KLM01] Gregor Kemper, Frank Lübeck, and Kay Magaard, *Matrix generators for the Ree groups ${}^2G_2(q)$*, Comm. Algebra **29** (2001), no. 1, 407–413. MR MR1842506 (2002e:20025)

[LG01] Charles R. Leedham-Green, *The computational matrix group project*, Groups and computation, III (Columbus, OH, 1999), Ohio State Univ. Math. Res. Inst. Publ., vol. 8, de Gruyter, Berlin, 2001, pp. 229–247. MR MR1829483 (2002d:20084)

[LGM02] C. R. Leedham-Green and Scott H. Murray, *Variants of product replacement*, Computational and statistical group theory (Las Vegas, NV/Hoboken, NJ, 2001), Contemp. Math., vol. 298, Amer. Math. Soc., Providence, RI, 2002, pp. 97–104. MR MR1929718 (2003h:20003)

[LGO97a] C. R. Leedham-Green and E. A. O'Brien, *Recognising tensor products of matrix groups*, Internat. J. Algebra Comput. **7** (1997), no. 5, 541–559. MR MR1470352 (98h:20018)

[LGO97b] ———, *Tensor products are projective geometries*, J. Algebra **189** (1997), no. 2, 514–528. MR MR1438187 (98b:20073)

[LL91] Y. N. Lakshman and Daniel Lazard, *On the complexity of zero-dimensional algebraic systems*, Effective methods in algebraic geometry (Castiglioncello, 1990), Progr. Math., vol. 94, Birkhäuser Boston, Boston, MA, 1991, pp. 217–225. MR MR1106424 (92d:13017)

[LN85] V. M. Levchuk and Ya. N. Nuzhin, *The structure of Ree groups*, Algebra i Logika **24** (1985), no. 1, 26–41, 122. MR MR816569 (87h:20085)

[LS74] Vicente Landazuri and Gary M. Seitz, *On the minimal degrees of projective representations of the finite Chevalley groups*, J. Algebra **32** (1974), 418–443. MR MR0360852 (50 #13299)

[Lüb01] Frank Lübeck, *Small degree representations of finite Chevalley groups in defining characteristic*, LMS J. Comput. Math. **4** (2001), 135–169 (electronic). MR MR1901354 (2003e:20013)

[Luk92] Eugene M. Luks, *Computing in solvable matrix groups*, IEEE Symposium on Foundations of Computer Science, 1992, pp. 111–120.

[LW98] Klaus Lux and Markus Wiegelmann, *Condensing tensor product modules*, The atlas of finite groups: ten years on (Birmingham, 1995), London Math. Soc. Lecture Note Ser., vol. 249, Cambridge Univ. Press, Cambridge, 1998, pp. 174–190. MR MR1647421 (99i:20018)

[Mal91] Gunter Malle, *The maximal subgroups of ${}^2F_4(q^2)$*, J. Algebra **139** (1991), no. 1, 52–69. MR MR1106340 (92d:20068)

[MSC96] D. S. Mitrinović, J. Sándor, and B. Crstici, *Handbook of number theory*, Mathematics and its Applications, vol. 351, Kluwer Academic Publishers Group, Dordrecht, 1996. MR MR1374329 (97f:11001)

[OL05] Eamonn A. OB́rien and Martin W. Liebeck, *Finding the characteristic of a group of Lie type*, J. London Math. Soc. (2005), to appear.

[Ono62] Takashi Ono, *An identification of Suzuki groups with groups of generalized Lie type.*, Ann. of Math. (2) **75** (1962), 251–259. MR MR0132780 (24 #A2616)

[Ono63] ———, *Correction to "An identification of Suzuki groups with groups of generalized Lie type"*, Ann. of Math. (2) **77** (1963), 413. MR MR0144967 (26 #2507)

[Pak00] Igor Pak, *The product replacement algorithm is polynomial*, FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science (Washington, DC, USA), IEEE Computer Society, 2000, pp. 476–485.

[Pak01] ______, *What do we know about the product replacement algorithm?*, Groups and computation, III (Columbus, OH, 1999), Ohio State Univ. Math. Res. Inst. Publ., vol. 8, de Gruyter, Berlin, 2001, pp. 301–347. MR MR1829489 (2002d:20107)

[Par84] R. A. Parker, *The computer calculation of modular characters (the meat-axe)*, Computational group theory (Durham, 1982), Academic Press, London, 1984, pp. 267–274. MR MR760660 (85k:20041)

[R D05] R Development Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2005, 3-900051-07-0.

[Ree60] Rimhak Ree, *A family of simple groups associated with the simple Lie algebra of type* (G_2), Bull. Amer. Math. Soc. **66** (1960), 508–510. MR MR0125154 (23 #A2460a)

[Ree61a] ______, *A family of simple groups associated with the simple Lie algebra of type* (F_4), Bull. Amer. Math. Soc. **67** (1961), 115–116. MR MR0125155 (23 #A2460b)

[Ree61b] ______, *A family of simple groups associated with the simple Lie algebra of type* (F_4), Amer. J. Math. **83** (1961), 401–420. MR MR0132781 (24 #A2617)

[Ree61c] ______, *A family of simple groups associated with the simple Lie algebra of type* (G_2), Amer. J. Math. **83** (1961), 432–462. MR MR0138680 (25 #2123)

[Rón90] Lajos Rónyai, *Computing the structure of finite algebras*, J. Symbolic Comput. **9** (1990), no. 3, 355–373. MR MR1056632 (91h:68093)

[Ser03] Ákos Seress, *Permutation group algorithms*, Cambridge Tracts in Mathematics, vol. 152, Cambridge University Press, Cambridge, 2003. MR MR1970241 (2004c:20008)

[Shi74] Ken-ichi Shinoda, *The conjugacy classes of Chevalley groups of type* (F_4) *over finite fields of characteristic* 2, J. Fac. Sci. Univ. Tokyo Sect. I A Math. **21** (1974), 133–159. MR MR0349863 (50 #2356)

[Shi75] ______, *The conjugacy classes of the finite Ree groups of type* (F_4), J. Fac. Sci. Univ. Tokyo Sect. I A Math. **22** (1975), 1–15. MR MR0372064 (51 #8281)

[Shp99] Igor E. Shparlinski, *Finite fields: theory and computation*, Mathematics and its Applications, vol. 477, Kluwer Academic Publishers, Dordrecht, 1999, The meeting point of number theory, computer science, coding theory and cryptography. MR MR1745660 (2001g:11188)

[Sta06] Mark James Stather, *Algorithms for computing with finite matrix groups*, Ph.D. thesis, University of Warwick, August 2006.

[Ste63] Robert Steinberg, *Representations of algebraic groups*, Nagoya Math. J. **22** (1963), 33–56. MR MR0155937 (27 #5870)

[Ste77] ______, *On theorems of Lie-Kolchin, Borel, and Lang*, Contributions to algebra (collection of papers dedicated to Ellis Kolchin), Academic Press, New York, 1977, pp. 349–354. MR MR0466336 (57 #6216)

[Ste97] Allan Steel, *A new algorithm for the computation of canonical forms of matrices over fields*, J. Symbolic Comput. **24** (1997), no. 3-4, 409–432, Computational algebra and number theory (London, 1993). MR MR1484489 (98m:65070)

[Str69] Volker Strassen, *Gaussian elimination is not optimal*, Numer. Math. **13** (1969), 354–356. MR MR0248973 (40 #2223)

[Suz60] Michio Suzuki, *A new type of simple groups of finite order*, Proc. Nat. Acad. Sci. U.S.A. **46** (1960), 868–870. MR MR0120283 (22 #11038)

[Suz62] ______, *On a class of doubly transitive groups*, Ann. of Math. (2) **75** (1962), 105–145. MR MR0136646 (25 #112)

[Suz64] ______, *On a class of doubly transitive groups. II*, Ann. of Math. (2) **79** (1964), 514–589. MR MR0162840 (29 #144)

[SZ93] Gary M. Seitz and Alexander E. Zalesskii, *On the minimal degrees of projective representations of the finite Chevalley groups. II*, J. Algebra **158** (1993), no. 1, 233–243. MR MR1223676 (94h:20017)

[Tho01] Emmanuel Thomé, *Computation of discrete logarithms in* $\mathbb{F}_{2^{607}}$, Advances in cryptology—ASIACRYPT 2001 (Gold Coast), Lecture Notes in Comput. Sci., vol. 2248, Springer, Berlin, 2001, pp. 107–124. MR MR1934518 (2003h:94051)

[vzGG03] Joachim von zur Gathen and Jürgen Gerhard, *Modern computer algebra*, second ed., Cambridge University Press, Cambridge, 2003. MR MR2001757 (2004g:68202)

[War63] Harold N. Ward, *On Ree's series of simple groups*, Bull. Amer. Math. Soc. **69** (1963), 113–114. MR MR0141707 (25 #5104)

[War66] ———, *On Ree's series of simple groups*, Trans. Amer. Math. Soc. **121** (1966), 62–89. MR MR0197587 (33 #5752)

[Wil05] R. A. Wilson, *Finite simple groups*, preprint, 2005.

[Wil06] ———, *Elementary constructions of the Ree groups*, preprint, 2006.

[WP06] R. A. Wilson and C. W. Parker, *Recognising simplicity of black-box groups*, submitted (2006).