

RECOGNISING THE REE GROUPS IN THEIR NATURAL REPRESENTATIONS

HENRIK BÄÄRNHIELM

ABSTRACT. We produce efficient algorithms for constructive recognition and constructive membership testing of the Ree groups $\text{Ree}(q)$, where $q = 3^{2m+1}$ for some $m > 0$, in their natural representations of degree 7. Given an oracle for the discrete logarithm problem in \mathbb{F}_q , the constructive recognition is a Las Vegas algorithm with expected running time $O((\xi + \log(q))(\log \log(q))^2 + \log(q)|X|)$ field operations, where X is the given generating set and ξ is the time for finding a uniformly random element of $\langle X \rangle$. The constructive membership testing is a Las Vegas algorithm with expected running time $O(\xi + \log(q))$, and a preprocessing step with expected running time $O((\xi + \log(q))(\log \log(q))^2)$ field operations that only needs to be executed once for a given X . The discrete logarithm oracle is only needed in the preprocessing step.

We also produce a recognition algorithm for $\text{Ree}(q) = \langle X \rangle$. This is a Las Vegas algorithm with expected running time $O(|X| \log(q))$ field operations.

Implementations of the algorithms are available for the computer system MAGMA.

1. INTRODUCTION

A goal of the *matrix recognition project* is to develop efficient algorithms for the study of subgroups of $\text{GL}(d, q)$. The classification due to Aschbacher (see [1]) provides one framework for this, and the first aim is to develop an algorithm that finds a composition series of a matrix group given by a set of generators. It is possible to do this with a recursive algorithm, and the recursion is described in [20]. However, we still have to deal with the base cases, which are the finite simple groups.

For each base case a number of problems arise. The simple group is given as $G = \langle X \rangle$ where $X \subseteq \text{GL}(d, q)$ 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 $g \in \text{GL}(d, q)$, decide whether or not $g \in G$, and if so express g as a word (or SLP, see Section 2.2) in X .
- (3) The problem of *constructive recognition*. Construct an isomorphism ψ from G to a *standard copy* H of G such that $\psi(g)$ and $\psi^{-1}(h)$ can be computed efficiently for every $g \in G$ and $h \in H$. Such an isomorphism is called *effective*.

To find a composition series using [20], we need only naming and constructive membership, but the effective isomorphisms to a standard copy are also very useful. Given these, many problems, including constructive membership, can be reduced to the standard copy.

This paper will consider the Ree groups $\text{Ree}(q)$, $q = 3^{2m+1}$ for any $m > 0$, which is one of the infinite families of finite simple groups. They are sometimes called the

small Ree groups to distinguish them from the groups ${}^2F_4(q)$ with q an odd power of 2. We will only consider the *natural representations*, which have dimension 7. Other representations are dealt with by the same author in [4]. Our standard copy will be $\text{Ree}(q)$ as it is defined in Section 3. In Section 4 we solve the recognition problem for $\text{Ree}(q)$, *i.e.* given $\langle X \rangle \leq \text{GL}(7, q)$ we give an algorithm that decides whether or not $\langle X \rangle = \text{Ree}(q)$. In Section 5 we solve the constructive membership problem for $\text{Ree}(q)$. In Section 7 we consider conjugates of $\text{Ree}(q)$ and show how to construct effective isomorphisms to $\text{Ree}(q)$.

The main objective of this paper is to prove the following:

Theorem 1.1. *Assume a random element oracle for subgroups of $\text{GL}(d, q)$ with time complexity $O(\xi)$ field operations and an oracle for the discrete logarithm problem in \mathbb{F}_q :*

- *There exists a Las Vegas algorithm that for each $\langle X \rangle \leq \text{GL}(7, q)$, with $q = 3^{2m+1}$ for some $m > 0$, such that $\langle X \rangle \cong \text{Ree}(q)$, finds an effective isomorphism $\Psi : \langle X \rangle \rightarrow \text{Ree}(q)$. The algorithm has time complexity $O((\xi + \log(q))(\log \log(q))^2 + \log(q) |X|)$ field operations.*
- *There exists a Las Vegas algorithm that for each $\langle X \rangle \subseteq \text{GL}(7, q)$, with $q = 3^{2m+1}$ for some $m > 0$, such that $\langle X \rangle \cong \text{Ree}(q)$, solves the constructive membership problem for $\langle X \rangle$. The algorithm has time complexity $O(\xi + \log(q))$ field operations and also has a preprocessing step, which only needs to be executed once for a given X , with time complexity $O((\xi + \log(q))(\log \log(q))^2)$ field operations. The discrete logarithm oracle is only needed in the preprocessing step.*

Proof. The algorithm giving Ψ follows from Theorem 7.4. The constructive membership testing in $\text{Ree}(q)$ follows from 5.2, 5.4 and 5.3, and for constructive membership testing in $\langle X \rangle$ we use Ψ , which can be computed using $O(1)$ field operations. \square

In constructive membership testing of $\text{Ree}(q)$, the essential problem is to find elements of order divisible by 3. In this paper, this is achieved by using the fact that $\text{Ree}(q)$ acts doubly transitively on a certain set $\mathcal{O} \subseteq \mathbb{P}^6(\mathbb{F}_q)$. After finding independent random elements in the stabiliser of a point, which is done by finding elements that map one point to another, it becomes easy to find elements of order divisible by 3.

One can view this as a process of applying permutation group techniques on a set which is exponentially large in the size of the input. Since \mathcal{O} has size q^3+1 , we cannot explicitly write down all its points and still have a polynomial time algorithm, and therefore we cannot write down the elements of $\text{Ree}(q)$ as permutations. However, given two points we can construct in polynomial time an element of $\text{Ree}(q)$ that maps one point to the other, which is a typical permutation group technique.

The ideas that are used here for constructive recognition and membership testing of $\text{Ree}(q)$ are very similar to the ones used in [2], [3] and [11] for $\text{Sz}(q)$ and $\text{SL}(2, q)$, respectively. The results are also similar in the sense that we reduce these problems to discrete log.

Implementations of the algorithms have been done in MAGMA (see [7]).

Acknowledgements should go to John Bray, Alexander Hulpke, Charles Leedham-Green, Maud de Visscher and Robert Wilson for their helpful advice.

2. PRELIMINARIES

We will now briefly discuss some general concepts that are needed later.

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. In most of this paper, the matrix dimension is constant, so all simple arithmetic with matrices can be done using $O(1)$ field operations, and raising a matrix to the $O(q)$ power can be done using $O(1)$ field operations using [9]. We shall also assume an oracle for the discrete logarithm problem for \mathbb{F}_q , so that this can be solved using $O(1)$ field operations.

We will need to find an element of order dividing $q-1$. The order can be computed using the algorithm of [9]. To obtain the *precise* order, this algorithm requires a factorisation of $q-1$, otherwise it might return a multiple of the correct order. For our purposes it is sufficient to know that the order divides $q-1$, so it is sufficient to learn the *pseudo-order* of the element, which is a multiple of its order. Hence we avoid the requirement to factorise $q-1$. The algorithm of [9] can also be used to obtain the pseudo-order, and for this it has time complexity $O(\log(q) \log \log(q))$ field operations.

2.2. 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 subwords occurring multiple times are computed only once. 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.

Thus 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 another set Y of the same size as X , by just changing the pointers so that they point to elements of Y .

2.3. Random elements. Our analysis assumes that we can construct uniformly distributed random elements of a group $\langle X \rangle \leq \text{GL}(7, q)$. The polynomial time algorithm of [5] produces nearly uniformly distributed random elements; an alternative algorithm is the *product replacement* algorithm of [10], which is also polynomial time by [24]. We will assume that we have a random element oracle, which produces a uniformly random element of $\langle X \rangle$ using $O(\xi)$ field operations, and automatically gives 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 $O(n)$ where n is the number of random elements that have been selected so far during the execution of the algorithm.

2.4. Las Vegas algorithms. All the algorithms we consider are probabilistic of the type known as *Las Vegas algorithms*. This type of algorithm is discussed in [27, Section 1.3] and [14, Section 3.2.1]. In short it is a probabilistic algorithm with an input parameter $\varepsilon \in (0, 1)$ that either returns **failure**, with probability at most ε , or otherwise returns a correct result. The time complexity 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 $O(p(n))$. 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.

3. THE SMALL REE GROUPS

The Ree groups were first described in [26], and their structure has been investigated in [19], [22] and [29]. A short survey is also given in [16, Chapter 11].

We now define our standard copy of the Ree groups. The generators that we use are those described in [18]. 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

$$\alpha(x) = \begin{bmatrix} 1 & x^t & 0 & 0 & 0 & -x^{3t+1} & -x^{3t+2} & x^{4t+2} \\ 0 & 1 & x & x^{t+1} & -x^{2t+1} & 0 & 0 & -x^{3t+2} \\ 0 & 0 & 1 & x^t & -x^{2t} & 0 & 0 & x^{3t+1} \\ 0 & 0 & 0 & 1 & x^t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -x & 0 & x^{t+1} \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & -x^t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

$$\beta(x) = \begin{bmatrix} 1 & 0 & -x^t & 0 & -x & 0 & -x^{t+1} \\ 0 & 1 & 0 & x^t & 0 & -x^{2t} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & x \\ 0 & 0 & 0 & 1 & 0 & x^t & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & x^t \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$$\gamma(x) = \begin{bmatrix} 1 & 0 & 0 & -x^t & 0 & -x & -x^{2t} \\ 0 & 1 & 0 & 0 & -x^t & 0 & x \\ 0 & 0 & 1 & 0 & 0 & x^t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -x^t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$$h(\lambda) = \begin{bmatrix} \lambda^t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda^{1-t} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda^{2t-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda^{1-2t} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda^{t-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda^{-t} \end{bmatrix} \quad (3.4)$$

$$\tau = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.5)$$

and define the Ree group as

$$\text{Ree}(q) = \langle \alpha(x), \beta(x), \gamma(x), h(\lambda), \tau \mid x \in \mathbb{F}_q, \lambda \in \mathbb{F}_q^\times \rangle. \quad (3.6)$$

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

$$U(q) = \langle \alpha(x), \beta(x), \gamma(x) \mid x \in \mathbb{F}_q \rangle \quad (3.7)$$

$$H(q) = \{h(\lambda) \mid \lambda \in \mathbb{F}_q^\times\} \cong \mathbb{F}_q^\times. \quad (3.8)$$

From [22] we then know that each element of $U(q)$ can be expressed in a unique way as

$$S(a, b, c) = \alpha(a)\beta(b)\gamma(c) \quad (3.9)$$

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 $\text{Ree}(q)$, and from [16, Chapter 11] we see

that

$$\begin{aligned} S(a_1, b_1, c_1)S(a_2, b_2, c_2) &= \\ &= S(a_1 + a_2, b_2 + b_2 + a_1a_2^{t+1}, c_1 + c_2 - a_1b_2 + b_1a_2 - a_2a_1^{t+2}) \end{aligned} \quad (3.10)$$

and

$$S(a, b, c)^{h(\lambda)} = S(\lambda a, \lambda^{t+2}b, \lambda^{t+3}c). \quad (3.11)$$

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

$$J = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.12)$$

From [29] and [16, Chapter 11] we immediately obtain

Proposition 3.1. *Let $G = \text{Ree}(q)$.*

- (1) $|G| = q^3(q^3 + 1)(q - 1)$.
- (2) *Conjugates of $U(q)$ intersect trivially.*
- (3) *The centre $Z(U(q)) = \{S(0, 0, c) \mid c \in F_q\}$.*
- (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 $Z(U(q)) \setminus \langle 1 \rangle$.*
- (6) $N_G(U(q)) = U(q)H(q)$ and G acts doubly transitively on the right cosets of $N_G(U(q))$, i.e. on a set of size $q^3 + 1$.
- (7) $U(q)H(q)$ is a Frobenius group with Frobenius kernel $U(q)$.

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

Proposition 3.2. *There exists $\mathcal{O} \subseteq \mathbb{P}^6(\mathbb{F}_q)$ on which $G = \text{Ree}(q)$ acts faithfully and doubly transitively. This set is*

$$\begin{aligned} \mathcal{O} &= \{(0 : 0 : 0 : 0 : 0 : 0 : 1)\} \cup \\ &\quad \{(1 : a^t : -b^t : (ab)^t - c^t : -b - a^{3t+1} - (ac)^t : -c - (bc)^t - a^{3t+2} - a^t b^{2t} : \\ &\quad a^t c - b^{t+1} + a^{4t+2} - c^{2t} - a^{3t+1} b^t - (abc)^t)\} \end{aligned} \quad (3.13)$$

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 $(U(q)H(q))^\tau$ and the stabiliser of P_∞ and P_0 is $H(q)$.

Proof. From Proposition 3.1 it follows that G is the disjoint union of $U(q)H(q)$ and $U(q)H(q)\tau U(q)H(q)$. Hence the action of G on \mathcal{O} is equivalent to the action defined in Proposition 3.1 via the map $(U(q)H(q))g \mapsto P_\infty g$. \square

We will use the notation G_P for the stabiliser in G of a point P .

Proposition 3.3. *Let $G = \text{Ree}(q)$.*

- (1) *A stabiliser in G of two distinct points of \mathcal{O} is conjugate to $H(q)$ and a stabiliser of three 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 [16, 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 [16, Chapter 11, Theorem 13.2(e)].
- (4) Each involution is conjugate to $h(-1) = \text{diag}(-1, 1, -1, 1, -1, 1, -1)$. Evidently, $h(-1)$ fixes P_∞ since $h(-1) \in H(q)$ and if $P = (p_1 : \dots : p_7) \in \mathcal{O}$ 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 $q + 1$. \square

Proposition 3.4. *All cyclic subgroups of $G = \text{Ree}(q)$ of order $q - 1$ are conjugate to $H(q)$ and hence each one is a stabiliser of two points of \mathcal{O} .*

Proof. Let $C = \langle g \rangle \leq G$ 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 $|g^k| = p$. 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 $g^x \in N_G(H(q)) = \langle H(q), \tau \rangle$ which is dihedral of order $2(q - 1)$. Hence $\langle g^x \rangle = H(q)$ since that is the unique cyclic subgroup of order $q - 1$ in $\langle H(q), \tau \rangle$. \square

Proposition 3.5. *Let $G = \text{Ree}(q)$ and let ϕ be the Euler totient function.*

- (1) *The centraliser of an involution $j \in G$ is isomorphic to $\langle j \rangle \times \text{PSL}(2, q)$ 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 even order is $q^2(q^5 - 3q^4 + 3q^2 - 5q + 2)/2$.*
- (5) *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 [16, Chapter 11].
- (2) All involutions are conjugate, and the index in G of the involution centraliser is

$$\frac{q^3(q^3 + 1)(q - 1)}{q(q^2 - 1)} = q^2(q^2 - q + 1) \quad (3.14)$$

where we have used the fact that $q^3 + 1 = (q + 1)(q^2 - q + 1)$.

- (3) 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

$$\left| \binom{\mathcal{O}}{2} \right| = \frac{q^3(q^3 + 1)}{2} \quad (3.15)$$

and the number of generators of such a subgroup is $\phi(q - 1)$.

- (4) Every element of even order lies in a cyclic subgroup of order $q - 1$. In each cyclic subgroup of order $q - 1$ there is a unique involution and hence $q - 3$ non-involutions of even order. The total number of elements of even order is therefore

$$\frac{(q - 3)(q^3 + 1)q^3}{2} + q^2(q^2 - q + 1) = \frac{q^2(q^5 - 3q^4 + 3q^2 - 5q + 2)}{2} \quad (3.16)$$

- (5) 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 3.3, the number of elements that fix at least one point is

$$\begin{aligned} & q^6 + \frac{(q-3)(q^3+1)q^3}{2} + q^2(q^2-q+1) = \\ &= \frac{q^2(q^5 - q^4 + 3q^2 - 5q + 2)}{2} \end{aligned} \quad (3.17)$$

□

Lemma 3.6. *If $g \in G = \text{Ree}(q)$ is uniformly random, then*

$$\Pr[|g| = q-1] = \frac{\phi(q-1)}{2(q-1)} > \frac{1}{12 \log \log(q)} \quad (3.18)$$

$$\Pr[|g| \text{ even}] = \frac{2-3q-2q^2+q^3}{2q(q-1)(q+1)} > 2/5 \quad (3.19)$$

$$\Pr[g \text{ fixes a point}] = \frac{-2+3q+q^4}{2(q+q^4)} \geq 1/2 \quad (3.20)$$

Proof. In each case, the first equality follows from Proposition 3.5 and Proposition 3.1. In the first case, the inequality follows from [23, Section II.8], and in the other cases the inequalities are clear since $m > 0$. □

Corollary 3.7. *Given a random element oracle for $G = \text{Ree}(q)$, we expect to obtain an element of order $q-1$ in $O(\log \log q)$ random selections, and an element that has even order or that fixes a point in $O(1)$ random selections.*

Proof. Clearly the number of selections is geometrically distributed, where the success probabilities for each selection are given by Lemma 3.6. Hence the expectations are as stated. □

Proposition 3.8. *Let $G = \text{Ree}(q)$ with natural module V and let $j \in G$ be an involution. Then $V_{C_G(j)} \cong S_j \oplus T_j$ 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 3.3, j is conjugate to $h(-1) = \text{diag}(-1, 1, -1, 1, -1, 1, -1)$ 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 $\text{PSL}(2, q)$, so that they are in fact submodules of V_j .

Let $v \in S_j$ and $g \in \text{PSL}(2, q)$. 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 $\text{PSL}(2, q)$. Similarly, T_j is also fixed.

Let $\gamma : V \times V \rightarrow V$ 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 $\text{PSL}(2, q)$, but $\text{PSL}(2, q)$ have no irreducible modules of dimension 2. Therefore S_j must be irreducible. □

From [22] and [19] we obtain

Proposition 3.9. *A maximal subgroup of $G = \text{Ree}(q)$ is conjugate to one of the following subgroups*

- $N_G(U(q)) = U(q)H(q)$, the point stabiliser

- $C_G(j) \cong \langle j \rangle \times \text{PSL}(2, q)$, the centraliser of an involution j
- $N_G(A_0) \cong (2 \times 2) : A_1 : 6$, where $A_0 \leq \text{Ree}(q)$ is cyclic of order $(q+1)/4$.
- $N_G(A_1) \cong A_1 : 6$, where $A_1 \leq \text{Ree}(q)$ is cyclic of order $q+1-3t$.
- $N_G(A_2) \cong A_2 : 6$, where $A_2 \leq \text{Ree}(q)$ is cyclic of order $q+1+3t$.
- $\text{Ree}(s)$ where q is a prime 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 3.8.

Let H be one of the normalisers and let x be a generator of the cyclic subgroup that is normalised. All elements of orthogonal groups in odd dimension and odd characteristic have 1 as an eigenvalue. Hence x has an eigenspace $V \neq E \neq \{0\}$ for the eigenvalue 1. 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. \square

Proposition 3.10. *Elements in $\text{Ree}(q)$ of order prime to 3 with the same trace are conjugate.*

Proof. From [29], the number of conjugacy classes of non-identity elements of order prime to 3 is $q-1$. Observe that for $\lambda \in \mathbb{F}_q^\times$, $\text{Tr}(\gamma(1)\tau h(\lambda)) = \lambda^t - 1$ and $|\gamma(1)\tau h(\lambda)|$ 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. \square

Proposition 3.11. *Let $G = \text{PSL}(2, q)$. If $x, y \in G$ are uniformly random, then*

$$\Pr[\langle x, y \rangle = G] = 1 - O(1/q) \quad (3.21)$$

Proof. The maximal subgroup $M \leq G$ consisting of the upper triangular matrices modulo scalars has index $q+1$, and all subgroups isomorphic to M are conjugate. Since $M = N_G(M)$, there are $q+1$ conjugates of M .

$$\Pr[\langle x, y \rangle \leq M^g \text{ some } g \in G] \leq \sum_{i=1}^{q+1} \Pr[\langle x, y \rangle \leq M] = \frac{1}{q+1} \quad (3.22)$$

The other maximal subgroups have index strictly less 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 bounded, and hence the probability that $\langle x, y \rangle$ lies in a maximal subgroup is $O(1/q)$. \square

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

Following [30, Chapter 3], 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 $\text{SO}(7, q)$.

Furthermore, when q is an odd power of 3, the group $G_2(q)$ has a certain automorphism, sometimes called the *exceptional outer automorphism*, whose set of fixed points form a group, and this is defined to be the Ree group $\text{Ree}(q) = {}^2G_2(q)$.

4. RECOGNITION

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

Theorem 4.1. *There exists a Las Vegas algorithm that, given $\langle X \rangle \leq \text{GL}(7, q)$, decides whether or not $\langle X \rangle = \text{Ree}(q)$. The algorithm has expected time complexity $O(|X| \log(q))$ field operations.*

Proof. Let $G = \text{Ree}(q)$, 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 $g \in \text{SO}(7, q)$, which is true if $\det g = 1$ and if $gJg^T = J$, where J is given by (3.12) 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. This follows if, for every v, w in a basis for M , $f(vg \otimes wg) = \alpha_{v,w} f(v \otimes w)$, where f is a generator of $\text{Hom}_G(M \otimes M, M)$ (which has dimension 1).
 - (c) Determine if g is a fixed point of the exceptional outer automorphism of $G_2(q)$, mentioned in Section 3.1. Computing the automorphism amounts to taking 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 3.9, it is sufficient to determine if $\langle X \rangle$ cannot be written over a smaller field and if $\langle X \rangle$ is not reducible. This can be done using [12] and the MeatAxe (see [15] and [17]).

Since the matrix degree is constant, the complexity of the first step of the algorithm is $O(1)$ field operations. For the same reason, the expected time of both the MeatAxe and [12] is $O(|X| \log(q))$ field operations. Hence our recognition algorithm has expected time $O(|X| \log(q))$ field operations, and it is Las Vegas since the MeatAxe is Las Vegas. \square

5. CONSTRUCTIVE MEMBERSHIP TESTING

We now describe the constructive membership algorithm for our standard copy $\text{Ree}(q)$. Given a set of generators X , such that $G = \langle X \rangle = \text{Ree}(q)$, and given an element $g \in G$, we want to express g as an SLP in X . We need the following result.

Proposition 5.1. *If $g_1, g_2 \in U(q)H(q)$ are uniformly random and independent, then*

$$\Pr[|[g_1, g_2]| = 9] = 1 - \frac{1}{q-1} \quad (5.1)$$

Proof. By Proposition 3.1, $[g_1, g_2] \in U(q)$ 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 $C_A(k_1) = U(q)/U(q)'$, so we again obtain $q(q-1)$ pairs. Finally, if $k_1 \notin U(q)$ then $|C_A(k_1)| = q-1$ 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. \square

The general structure of the algorithm is the same as the algorithm for the same problem in the Suzuki groups $Sz(q)$, with q an odd power of 2, which is described in [3]. It consists of a preprocessing step and a main step.

5.1. Preprocessing. The preprocessing step consists of finding “standard generators” for $O_3(G_{P_\infty}) = U(q)$ and $O_3(G_{P_0})$. In the case of $O_3(G_{P_\infty})$ the standard generators are defined as matrices

$$\{S(a_i, x_i, y_i)\}_{i=1}^n \cup \{S(0, b_i, z_i)\}_{i=1}^n \cup \{S(0, 0, c_i)\}_{i=1}^n \quad (5.2)$$

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 $n = \log_3 q = 2m + 1$).

For every $a, b, c \in \mathbb{F}_q$, the matrix $S(a, b, c) \in G_{P_\infty}$ can be expressed as a product of the standard generators of $O_3(G_{P_\infty})$, and similarly for G_{P_0} . The standard generators are therefore used in the main step to perform row operations in G_{P_∞} and G_{P_0} .

Theorem 5.2. *Given a random element oracle for G 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 $O_3(G_{P_\infty})$ and $O_3(G_{P_0})$. It has expected time complexity $O((\xi + \log(q))(\log \log(q))^2)$ field operations. The length of the SLPs of the standard generators are $O(\log q)$.*

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 6.9. 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 6.9. 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 $O_3(G_{P_\infty})$ we now take $U = U_1 \cup U_2$ where

$$U_1 = \bigcup_{i=1}^{2m+1} \{c_1^{d_i^i}, (c_1^3)^{d_i^i}\} \quad (5.3)$$

and

$$U_2 = \bigcup_{1 \leq i < j \leq 2m+1} \{[c_1^{d_i^i}, c_1^{d_j^j}]\} \quad (5.4)$$

Similarly we obtain L for $O_3(G_{P_0})$.

It follows from (3.10) and (3.11) that (5.3) and (5.4) 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 6.9. Hence the algorithm is Las Vegas.

By Corollary 6.9, the expected time to find a_1 and b_1 is $O((\xi + \log(q)) \log \log(q))$, and these are uniformly distributed independent random elements. The elements of order dividing $q - 1$ can be diagonalised as required. Since $G_{P_\infty} = U(q)H(q)$, the proportion of elements of order $q - 1$ in G_{P_∞} is $\phi(q - 1)/(q - 1)$, and similarly for G_{P_0} . Hence the expected time for the first step is $O((\xi + \log(q))(\log \log(q))^2)$ field operations.

Similarly, by Proposition 5.1 the expected time for the second step is $O((\xi + \log(q)) \log \log(q))$ field operations.

By the remark preceding the Theorem, L determines three sets of field elements $\{a_1, \dots, a_{2m+1}\}$, $\{b_1, \dots, b_{2m+1}\}$ and $\{c_1, \dots, c_{2m+1}\}$. By (3.11), 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 look at 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 $|\lambda| \mid 3^n - 1$ where n is a proper divisor of $2m + 1$. Hence the dominating term in the complexity is the first step. \square

5.2. Main algorithm. Given $g \in G$ we now show the procedure for expressing g as an SLP. It is given as Algorithm 5.1.

Algorithm 5.1: ELEMENTTOSLP(U, L, g)

```

1 Input: Standard generators  $U$  for  $G_{P_\infty}$  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      $r := \text{RANDOM}(G)$ 
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     $x := \text{Tr}(h(\lambda))$ 
11    until  $x - 1$  is a square in  $\mathbb{F}_q^\times$ 
12    // Express diagonal matrix as SLP
13    Find  $u = S(0, 0, \sqrt{(x-1)^{3t}})S(0, 1, 0)^\tau$  using  $U \cup L$ 
14    // Now  $\text{Tr } u = x$ 
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  $u^{ab} = h(\lambda)$ 
20      then
21        Let  $W$  be the SLP for  $(u^{ab} z_2^{-1})^{z_1^{-1}} r^{-1}$ 
22        return  $W$ 
23      else
24        Let  $W$  be the SLP for  $((u^{ab})^{-1} z_2^{-1})^{z_1^{-1}} r^{-1}$ 
25        return  $W$ 
26    end

```

Theorem 5.3. *Given a random element oracle for G , Algorithm 5.1 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 row operations, so we can obtain them as SLPs.

The element u found at line 13 clearly has trace x . Then u can be computed using row operations and so we obtain it as an SLP. From Proposition 3.10 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 row operations, 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. \square

5.3. Complexity.

Theorem 5.4. *Given a random element oracle for G , Algorithm 5.1 has expected time complexity $O(\xi + \log q)$ field operations and the length of the returned SLP is $O(\log q)$.*

Proof. From (5.3) and (5.4) we see that the number of standard generators is $O(\log q)$. This immediately implies that the row operations performed at lines 7, 9, 13, 16 and 17 use $O(\log q)$ field operations.

From Corollary 3.7, the expected time to find r is $O(\xi)$ 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 $O(\xi + \log q)$ field operations.

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

From Theorem 5.2 each standard generator SLP has length $O(\log q)$ and hence W will have length $O(\log q)$. \square

6. COMPUTING AN ELEMENT OF A STABILISER

Let $G = \text{Ree}(q) = \langle X \rangle$. 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 $Pg = Q$.

In constructive recognition of $\text{PSL}(2, q)$ and $\text{Sz}(q)$, the same problem arise, see [11] and [3]. In each case, the size of the set \mathcal{O} on which the group acts doubly transitively is different. The element mapping one point to another is also found in different parts of the group. Table 1 illustrates this.

TABLE 1. Finding mapping elements

Group	$ \mathcal{O} $	Where the elements are found
$\text{PSL}(2, q)$	$q + 1$	Cosets of $\langle a \rangle$ where $ a = q - 1$
$\text{Sz}(q)$	$q^2 + 1$	Double cosets of $\langle a \rangle$ where $ a = q - 1$
$\text{Ree}(q)$	$q^3 + 1$	The involution centraliser $2 \times \text{PSL}(2, q)$

More specifically, the general idea is to find an involution $j \in G$ by random search, and then compute $C_G(j) \cong \langle j \rangle \times \text{PSL}(2, q)$ using the *Bray algorithm* described in [8]. The given module restricted to the centraliser splits up as in Proposition 3.8, and the points $P, Q \in \mathcal{O}$ restrict to points in the 3-dimensional submodule. We then find an element $g \in C_G(j)$ that maps these 3-dimensional points to each other, and we obtain g as an SLP in the generators of $C_G(j)$ using [11]. It turns out that with high probability, we can then multiply g by an element that fixes one of the 3-dimensional points so that g also maps P and Q to each other. A discrete logarithm oracle is needed in that step. Since [8] produces generators for the centraliser as SLPs in X , we obtain g as an SLP in X .

It should be noted that it is easy to find involutions by random search, because it is sufficient to find an element of even order and then power it up, and this is easy by Corollary 3.7.

6.1. The involution centraliser. The Bray algorithm described in [8] is a black-box algorithm for computing a generating set of an involution centraliser $C_G(j)$. It works by computing random elements of $C_G(j)$ until the whole centraliser is generated, and this automatically gives the generators as SLPs in X .

There are two issues involved when using this algorithm. First, the elements that are computed may not be uniformly random, so that we might have trouble generating the whole centraliser. In [13] it is shown that this is not a problem

in $\text{Ree}(q)$. Second, 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 $\text{PSL}(2, q)$, and by Proposition 3.11 it is sufficient to compute two random elements of the derived group. Random elements of the derived group can be found using [21].

We can therefore find the involution centraliser $C_G(j) \leq G$ and $C_G(j)' \cong \text{PSL}(2, q)$.

Lemma 6.1. *There exists a Las Vegas algorithm that, given $\langle Y \rangle \leq G$ such that $\langle Y \rangle = C_G(j)'$ for some involution $j \in G$, finds*

- the submodule $S_j \leq V_j$ asserted by Proposition 3.8,
- an effective $\langle Y \rangle$ -module homomorphism $\psi_V : V_j \rightarrow S_j$,
- the induced map $\psi_{\mathcal{O}} : \mathbb{P}(V_j) \rightarrow \mathbb{P}(S_j)$,
- the corresponding map ψ_G from the 7-dimensional representation of $C_G(j)'$ to the 3-dimensional representation.

The maps can be computed using $O(1)$ field operations. The algorithm has expected time complexity $O(|Y| \log(q))$ 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 [15] and [17]. The maps consist of a change of basis followed by a projection to a subspace, and so the Lemma follows. \square

Lemma 6.2. *Use the notation of Lemma 6.1. Assume a random element oracle for subgroups of $\text{GL}(d, q)$. There exists a Las Vegas algorithm that, given $\langle Y \rangle = \psi_G(C_G(j)')$ for an involution $j \in G$, finds effective isomorphisms $\rho_G : \langle Y \rangle \rightarrow \text{PSL}(2, q)$, $\xi_3 : \text{PSL}(2, q) \rightarrow \langle Y \rangle$ and $\xi_7 : \langle Y \rangle \rightarrow C_G(j)'$.*

The map ξ_3 is the symmetric square map of $\text{PSL}(2, q)$, both $\psi_G \circ \xi_7$ and $\xi_3 \circ \rho_G$ are the identity on $\langle Y \rangle$ and the maps can be computed using $O(\log q)$ field operations. The algorithm has expected time complexity $O((\xi + \log(q)) \log \log(q))$ field operations.

Proof. By Proposition 3.8, the group $\langle Y \rangle$ is an irreducible 3-dimensional copy of $\text{PSL}(2, q)$, so it must be the symmetric square of the natural representation. Using [11] we can constructively recognise $\langle Y \rangle$ and obtain the maps ϕ_G and $\phi_{\mathcal{O}}$. We can also solve the constructive membership problem in the standard copy, and by evaluating straight line programs we obtain the maps ξ_3 and ξ_7 .

It follows from [11] that the expected time complexity is as stated, and that the length of the straight line programs are $O(\log q)$. Hence the maps can be computed using $O(\log q)$ field operations. \square

6.2. 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 6.1 and 6.2 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 $\psi_G(C_G(j)')$ 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 6.3. *The map $\psi_{\mathcal{O}}$ restricted to \mathcal{O} is surjective but not injective.*

Proof. The map ψ_V is the projection onto S_j , so the kernel are those vectors that lie in T_j . From the proof of Proposition 3.8, with respect to a suitable basis, T_j is the -1 -eigenspace of $h(-1)$. Hence by Proposition 3.13, $|\mathcal{O} \cap \mathbb{P}(T_j)| = q + 1$. Therefore the number of points of \mathcal{O} with a non-zero projection in $\mathbb{P}(S_j)$ must be $q^3 - q$, and the result follows. \square

Proposition 6.4. *Under the action of $\psi_G(C_G(j)')$, the set $\psi_{\mathcal{O}}(\mathcal{O})$ 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 $x^2 + y^2$, which has size $q(q-1)/2$.*
- (3) *The orbit containing x^2 (and y^2), which has size $q+1$.*

The preimage in $\text{SL}(2, q)$ of $\rho_G(\psi_G(C_G(j)')_{xy})$ is dihedral of order $2(q-1)$, generated by the matrices

$$\begin{bmatrix} \alpha & 0 \\ 0 & \alpha^{-1} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (6.1)$$

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

Proof. This is just elementary theory of quadratic forms, except that we work projectively. \square

The algorithm that maps one point to another is given as Algorithm 6.1.

Algorithm 6.1: FINDMAPPINGELEMENT($X, C_G(j), P, Q$)

```

1  Input: Generating set  $X$  for  $G = \text{Ree}(q)$ .
   Points  $P \neq Q \in \mathcal{O}$  such that both  $\psi_{\mathcal{O}}(P)$  and  $\psi_{\mathcal{O}}(Q)$  are non-degenerate and represent 0.
   Involution centraliser  $C_G(j)$  with the maps from Lemma 6.1 and 6.2.
2  Output: An element  $h \in G$ , written as an SLP in  $X$ , such that  $Ph = Q$ .
3   $P_3 := \psi_{\mathcal{O}}(P)$ 
4   $Q_3 := \psi_{\mathcal{O}}(Q)$ 
5  if  $\exists$  upper triangular  $g \in \text{PSL}(2, q)$  such that  $P_3\xi_3(g) = Q_3$ 
   then
6      $R_3 := \psi_{\mathcal{O}}(P\xi_7(g))$ 
7     // Now  $R_3 = Q_3$ 
8     Find  $c \in \text{GL}(3, q)$  such that  $(xy)c = R_3$ 
9     Let  $D$  be the image in  $\text{PSL}(2, q)$  of the diagonal matrix defined in (6.1)
10     $s := \xi_7(\xi_3(D)^c)$ 
11    // Now  $\langle s \rangle \leq \psi_G^{-1}(H_{R_3})$ 
12     $d, z := \text{DIAGONALISE}(s)$ 
13    // Now  $d = s^z$ 
14    if  $\exists \lambda \in \mathbb{F}_q^\times$  such that  $(P\xi_7(g)z)h(\lambda) = Qz$ 
   then
15        $i := \text{DISCRETELOG}(d, h(\lambda))$ 
16       // Now  $d^i = h(\lambda)$ 
17       return  $\xi_7(g)s^i$ 
   end
   end
18 return FAIL
    
```

6.3. Finding a stabilising element. The complete algorithm for finding a non-trivial element of G_P is then as follows, given a generating set X for G and $P \in \mathcal{O}$.

- (1) Find a random involution $j \in G$.
- (2) Compute probable generators for $C_G(j)$ using the Bray algorithm, and probable generators for $C_G(j)'$ by taking commutators of the generators of $C_G(j)$.
- (3) Use the MeatAxe to verify that the module for $C_G(j)'$ splits up only as in Proposition 3.8, and hence verify that we have the whole of $C_G(j)'$. Return to the previous step if not.

- (4) Compute the maps $\psi_{\mathcal{O}}$ and ψ_G using Lemma 6.1. Return to the first step if $\psi_{\mathcal{O}}(P)$ does not represent 0.
- (5) Compute the maps from Lemma 6.2.
- (6) Take random $g_1 \in G$ and let $Q = Pg_1$, so Q is uniformly random from \mathcal{O} . Repeat until $P \neq Q$ and $\psi_{\mathcal{O}}(Q)$ represents 0.
- (7) Use Algorithm 6.1 to find $g_2 \in G$ such that $Q = Pg_2$. Return to the previous step if it fails, and otherwise return $g_1g_2^{-1}$.

6.4. Correctness and complexity.

Lemma 6.5. *If $P \in \mathcal{O}$ is uniformly random, then $\psi_{\mathcal{O}}(P)$ represents 0 with probability $1/2 + O(1/q)$.*

Proof. Since P is uniformly random and $\psi_{\mathcal{O}}$ was chosen independently of P , it follows that $\psi_{\mathcal{O}}(P)$ is uniformly random from $\psi_{\mathcal{O}}(\mathcal{O})$. Hence $\psi_{\mathcal{O}}(P) = x^2 + bxy + cy^2$ where $(1 : b : c)$ is uniformly distributed in $\mathbb{P}^2(\mathbb{F}_q)$.

Now $\psi_{\mathcal{O}}(P)$ represents 0 if the discriminant $b^2 - c$ is a non-zero square in \mathbb{F}_q . This is not the case if $b^2 = c$, in other words for the q points $(1 : b : b^2)$. If $b^2 - c \neq 0$ then it is a square with probability $1/2$, so

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

and the Lemma follows. \square

Lemma 6.6. *If $P, Q \in \psi_{\mathcal{O}}(\mathcal{O})$ are uniformly random, then the probability that there exists an element $g \in \text{PSL}(2, q)$, such that the preimage of g in $\text{SL}(2, q)$ is upper triangular and $P\xi_3(g) = Q$, is at least $1/2 + O(1/q)$.*

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

$$g = \begin{bmatrix} u & v \\ 0 & 1/u \end{bmatrix} \quad (6.2)$$

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 $P\xi_3(g) = Q$. Note that g is the preimage in $\text{SL}(2, q)$ of an element in $\text{PSL}(2, q)$ and therefore $\pm u$ determine the same element of $\text{PSL}(2, q)$. The map ξ_3 is the symmetric square map, so

$$\xi_3(g) = \mathcal{S}^2(g) = \begin{bmatrix} u^2 & -uv & v^2 \\ 0 & 1 & v/u \\ 0 & 0 & 1/u^2 \end{bmatrix} \quad (6.3)$$

This leads to the following equations:

$$u^2 = C \quad (6.4)$$

$$-uv + a = Cl \quad (6.5)$$

$$v^2 + avu^{-1} + bu^{-2} = Cn \quad (6.6)$$

for some $C \in \mathbb{F}_q^\times$. We can solve for u in (6.4) and for v in (6.5), so that (6.6) becomes

$$C^2(n - m^2) + a^2 - b = 0 \quad (6.7)$$

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

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

This is $1/2 + O(1/q)$ and the Lemma follows. \square

Theorem 6.7. *If Algorithm 6.1 returns an element g , then $Pg = Q$. If P and Q are uniformly random, then the probability that Algorithm 6.1 finds such an element is at least $1/4 + O(1/q)$.*

Proof. By Proposition 6.4, 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 R_3 . Then $\xi_3(D)^c \in H_{R_3}$ is of order $(q-1)/2$. Hence s also has order $(q-1)/2$, and $s \in \psi_G^{-1}(H_{R_3})$.

Since G acts doubly transitively on \mathcal{O} , there exists some $h \in G$ such that $Ph = Q$, and if we let $R = P\xi_7(g)$ then $R\xi_7(g)^{-1}h = Q$ and $\psi_{\mathcal{O}}(R) = R_3 = Q_3$. Hence $\psi_G(\xi_7(g)^{-1}h) \in H_{Q_3}$, and therefore $\xi_7(g)^{-1}h \in \psi_G^{-1}(H_{R_3})$.

By Proposition 6.4, $\psi_G^{-1}(H_{R_3})$ is dihedral of order $q-1$, and s generates a subgroup of index 2. Therefore $\Pr[\xi_7(g)^{-1}h \in \langle s \rangle] = 1/2$, which is the success probability of line 14.

The success probability of line 5 is given by Lemma 6.6. Hence the success probability of the algorithm is as stated. \square

Theorem 6.8. *The time complexity of Algorithm 6.1 is $O(\log q)$ field operations, given an oracle for the discrete logarithm problem in \mathbb{F}_q . The length of the returned SLP is $O(\log q)$.*

Proof. By Lemma 6.6, line 5 involves solving a quadratic equation in \mathbb{F}_q , and hence uses $O(1)$ field operations. Computing the maps ξ_3 and ξ_7 uses $O(\log q)$ field operations, and it is clear that the rest of the algorithm can be done using $O(1)$ field operations.

By [11], the length of the SLP from the constructive membership testing in $\text{PSL}(2, q)$ is $O(\log q)$, which is therefore also the length of the returned SLP. \square

Corollary 6.9. *Assume a random element oracle for subgroups of $\text{GL}(d, q)$ and an oracle for the discrete logarithm problem in \mathbb{F}_q . There exists a Las Vegas algorithm that, given $X \subseteq \text{GL}(7, q)$ such that $G = \langle X \rangle = \text{Ree}(q)$ and $P \in \mathcal{O}$, computes a uniformly random element of G_P as an SLP in X . The expected time complexity of the algorithm is $O((\xi + \log(q)) \log \log(q))$ field operations. The length of the returned SLP is $O(\log q)$.*

Proof. The algorithm is given in Section 6.3.

To determine if an element $j \in G$ has even order, [9] can be used, and this uses $O(\log(q) \log \log(q))$ field operations. An involution is then found by powering up. Hence by Corollary 3.7, the expected time to find an involution is $O(\xi + \log(q) \log \log(q))$.

In [13] it is shown that the Bray algorithm will produce uniformly random elements of the centraliser. Hence we can also obtain uniformly random elements of its derived group using [21]. By Proposition 3.11, two random elements will generate $\text{PSL}(2, q)$ with high probability. This implies that the expected time to obtain probable generators for $\text{PSL}(2, q)$ is $O(1)$ field operations.

The point Q is not P with probability $1 - 1/|\mathcal{O}|$ and by Lemma 6.5 the points P, Q do not represent zero with probability $1/4$, so the expected time of the penultimate step is $O(1)$ field operations.

Since the points P, Q can be considered uniformly random and independent in Algorithm 6.1, the element returned by that algorithm is uniformly random. Hence the element returned by the algorithm in Section 6.3 is uniformly random.

The expected time complexity of the last step is given by Theorem 6.8 and 6.7. It follows by the above and from Lemma 6.1 and 6.2 and Corollary 3.6 that the expected time complexity of the algorithm in Section 6.3 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 . \square

7. CONJUGATES OF THE REE GROUP

We now consider representations of $\text{Ree}(q)$ other than the standard copy, so we consider groups $G \cong \text{Ree}(q)$ and we want to find an effective isomorphism from G to $\text{Ree}(q)$. We will only consider the case when $G \leq \text{GL}(7, q)$. Then the problem amounts to finding a conjugating element, since up to Galois automorphisms, $\text{Ree}(q)$ has only one equivalence class of faithful representations in $\text{GL}(7, q)$.

Hence assume that we are given $G \leq \text{GL}(7, q)$ such that $G^h = \text{Ree}(q)$ for some $h \in \text{GL}(7, q)$, and we turn to the problem of finding some $g \in \text{GL}(7, q)$ such that $G^g = \text{Ree}(q)$, thus obtaining an algorithm that finds isomorphisms from any conjugate of $\text{Ree}(q)$ to the standard copy.

Lemma 7.1. *Assume a random element oracle for subgroups of $\text{GL}(d, q)$. There exists a Las Vegas algorithm that, given $\langle X \rangle \leq \text{GL}(7, q)$ such that $\langle X \rangle^h = \text{Ree}(q)$ for some $h \in \text{GL}(7, q)$, finds a point $P \in \mathcal{O}^{h^{-1}} = \{Qh^{-1} \mid Q \in \mathcal{O}\}$. The algorithm has expected time complexity $O((\xi + \log q) \log \log(q))$ field operations.*

Proof. Clearly $\mathcal{O}^{h^{-1}}$ is the set on which $\langle X \rangle$ acts doubly transitively. For a matrix $h(\lambda) \in \text{Ree}(q)$ 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 $\text{Ree}(q)$ 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 [9] using expected $O(\log(q) \log \log(q))$ field operations, and by Corollary 3.7, the expected time to find the element is $O(\xi \log \log(q))$ field operations. We then find the eigenspaces of g .

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

Lemma 7.2. *There exists a Las Vegas algorithm that, given $\langle X \rangle \leq \text{GL}(7, q)$ such that $\langle X \rangle^d = \text{Ree}(q)$ where $d = \text{diag}(d_1, d_2, d_3, d_4, d_5, d_6, d_7) \in \text{GL}(7, q)$, finds a diagonal matrix $e \in \text{GL}(7, q)$ such that $\langle X \rangle^e = \text{Ree}(q)$. The expected time complexity is $O(|X| \log q + (\xi + \log q) \log \log(q))$ field operations.*

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

$$K = dJd = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & d_1 d_7 \\ 0 & 0 & 0 & 0 & 0 & d_2 d_6 & 0 \\ 0 & 0 & 0 & 0 & d_3 d_5 & 0 & 0 \\ 0 & 0 & 0 & -d_4^2 & 0 & 0 & 0 \\ 0 & 0 & d_3 d_5 & 0 & 0 & 0 & 0 \\ 0 & d_2 d_6 & 0 & 0 & 0 & 0 & 0 \\ d_1 d_7 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (7.1)$$

where J is given by (3.12). Using [15] and [17], 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 $e = \text{diag}(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ that we want to find is also determined up to a scalar multiple (and up to multiplication by a diagonal matrix in $\text{Ree}(q)$).

Since e must take J to K , we must have $K_{1,7} = d_1 d_7 = e_1 e_7$, $K_{2,6} = d_2 d_6 = e_2 e_6$, $K_{3,5} = d_3 d_5 = e_3 e_5$ and $K_{4,4} = -d_4^2 = -e_4^2$. The matrix e is determined up to a scalar multiple, so 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 $\text{Ree}(q)$ we must have $Pe \in \mathcal{O}$ for every point $P \in \mathcal{O}^{d-1}$, which is the set on which G acts doubly transitively. By Lemma 7.1, we can find $P = (1 : p_2 : p_2 : p_3 : p_4 : p_5 : p_6 : p_7) \in \mathcal{O}^{d-1}$, and the condition $Pe = (1 : p_2 e_2 : p_3 e_3 : p_4 s : p_5 K_{3,5}/e_3 : p_6 K_{2,6}/e_2 : p_7 K_{1,7}) \in \mathcal{O}$ is given by (3.13) and amounts to a polynomial equation in $e_2 = \alpha$ and $e_3 = \beta$.

The degree in α or β will not be bounded, but will depend on t . We can obtain any number of these equations by finding random points of \mathcal{O}^{d-1} using Lemma 7.1, and by considering α^t and β^t to be independent variables, we can solve for them and thus arrive at polynomial equations in α and β of bounded degree.

Assuming we have two polynomials $f_1(\alpha, \beta)$ and $f_2(\alpha, \beta)$ of bounded degree, we can find the solutions of $f_1(\alpha, \beta) = f_2(\alpha, \beta) = 0$ using resultants. If we compute the resultant with respect to β of f_1 and f_2 , which will be a univariate polynomial $g(\alpha)$, we can then find the roots of g , thus obtaining a number of possible values for α . Substituting these values into f_1 and f_2 , we obtain univariate polynomials in β , and we can then find the possible values for β .

Hence we can find e_2 and e_3 . The diagonal matrix $e = \text{diag}(1, e_2, e_3, s, K_{3,5}e_3^{-1}, K_{2,6}e_2^{-1}, K_{1,7})$ now satisfies $G^e = \text{Ree}(q)$.

By Lemma 7.1, [28, Corollary 14.16], [15] and [17], this is a Las Vegas algorithm with the stated time complexity. \square

Lemma 7.3. *There exists a Las Vegas algorithm that, given subsets X, Y_P and Y_Q of $\text{GL}(7, q)$ such that $\text{O}_3(G_P) < \langle Y_P \rangle \leq G_P$ and $\text{O}_3(G_Q) < \langle Y_Q \rangle \leq G_Q$, respectively, where $\langle X \rangle = G$, $G^h = \text{Ree}(q)$ for some $h \in \text{GL}(7, q)$ and $P, Q \in \mathcal{O}^{h^{-1}}$, finds $k \in \text{GL}(7, q)$ such that $(G^k)^d = \text{Ree}(q)$ for some diagonal matrix $d \in \text{GL}(7, q)$. The algorithm has expected time complexity $\mathcal{O}(|X| \log(q))$ 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 $i = 1, \dots, 7$. 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 $V = V_7^P \supset V_6^P \supset V_5^P \supset V_4^P \supset V_3^P \supset V_2^P \supset V_1^P$ and $V = V_7^Q \supset V_6^Q \supset V_5^Q \supset V_4^Q \supset V_3^Q \supset V_2^Q \supset V_1^Q$ 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 $i = 1, \dots, 7$, choose $u_i \in U_i$.
- (3) Now let k be the matrix such that k^{-1} has u_i as row i , for $i = 1, \dots, 7$.

We now motivate the second step of the algorithm. Let $(M)_i$ denote the i -th row of a matrix M , and let V_i^P and V_i^Q be as in the algorithm.

We may assume that $Y_P = \{x, y\}$, $Y_Q = \{u, v\}$ where $|x| = |u| = 9$ and both $|y|$ and $|v|$ divide $q - 1$ (and y and v are nontrivial).

There exists $g' \in \text{Ree}(q)$ such that $Phg' = P_\infty$ and $Qhg' = P_0$, since $\text{Ree}(q)$ acts doubly transitively on \mathcal{O} . If we let $z = hg'$, then $\langle Y_P \rangle^z$ and $\langle Y_Q \rangle^z$ consist of upper and lower triangular matrices, respectively. Hence there exist $a_1, b_1, c_1 \in \mathbb{F}_q$ such that $x = S(a_1, b_1, c_1)^{z^{-1}}$, and then $V_1^P = \langle (x)_1 \rangle = \langle (S(a_1, b_1, c_1))_1 z^{-1} \rangle = V_1$. But $(S(a_1, b_1, c_1))_1 z^{-1} = (z^{-1})_1$ so by choosing some non-zero vector in V_1^P we obtain a scalar multiple of the first row of z^{-1} . Similarly, there exist $a_2, b_2, c_2 \in \mathbb{F}_q$ such that $u = (S(a_2, b_2, c_2)^\tau)^{z^{-1}}$, and $V_1^Q = \langle (u)_7 \rangle = \langle (S(a_2, b_2, c_2)^\tau)_7 z^{-1} \rangle$. But $(S(a_2, b_2, c_2)^\tau)_7 z^{-1} = (z^{-1})_7$ so by choosing some non-zero vector in V_1^Q we obtain a scalar multiple of the last row of z^{-1} .

In fact $\dim U_i = 1$ for $i = 1, \dots, 7$ and by choosing non-zero vectors from these we obtain scalar multiples of each of the rows of z^{-1} .

Thus the matrix k found in the algorithm satisfies that $z = kd$ for some diagonal matrix $d \in \text{GL}(7, q)$. Since $\text{Ree}(q) = G^h = G^z = (G^k)^d$, the algorithm returns a correct result, and it is Las Vegas because the MeatAxe is Las Vegas (see [15] and [17]). Clearly it has the same time complexity as the MeatAxe. \square

Theorem 7.4. *Assume a random element oracle for subgroups of $\text{GL}(d, q)$ and an oracle for the discrete logarithm problem in \mathbb{F}_q . There exists a Las Vegas algorithm that, given $\langle X \rangle \leq \text{GL}(7, q)$ such that $\langle X \rangle^h = \text{Ree}(q)$ for some $h \in \text{GL}(7, q)$, finds $g \in \text{GL}(7, q)$ such that $\langle X \rangle^g = \text{Ree}(q)$. The algorithm has expected time complexity $O((\xi + \log(q))(\log \log(q))^2 + \log(q) |X|)$ field operations.*

Proof. Let $G = \langle X \rangle$. First note that g is determined up to multiplication by an element of $\text{Ree}(q)$, so we will find g such that $hg' = g$ where $g' \in \text{Ree}(q)$.

The algorithm described in Corollary 6.9 works equally well for a conjugate of $\text{Ree}(q)$, so we can find generators for a stabiliser of a point in G , using the algorithm described in Theorem 5.2. Note that the discrete logarithm oracle is needed by [11].

- (1) Find points $P, Q \in \mathcal{O}^{h^{-1}}$ using Lemma 7.1. Repeat until $P \neq Q$.
- (2) Find generating sets Y_P and Y_Q such that $\text{O}_3(G_P) < \langle Y_P \rangle \leq G_P$ and $\text{O}_3(G_Q) < \langle Y_Q \rangle \leq G_Q$ using the first two steps of the algorithm from the proof of Theorem 5.2.
- (3) Find $k \in \text{GL}(7, q)$ such that $(G^k)^d = \text{Ree}(q)$ for some diagonal matrix $d \in \text{GL}(7, q)$, using Lemma 7.3.
- (4) Find a diagonal matrix e using Lemma 7.2.
- (5) Now $g = ke$ satisfies that $G^g = \text{Ree}(q)$.

Be Lemma 7.1, 7.3 and 7.2, and the proof of Theorem 5.2, this is a Las Vegas algorithm with time complexity as stated. \square

APPENDIX A. IMPLEMENTATION AND PERFORMANCE

Implementations of the algorithms are available in MAGMA. The implementations uses the existing MAGMA implementations of the algorithms described in [6], [8], [9], [10], [11], [12], [15] and [28, Corollary 14.16].

We have benchmarked the computation of generating sets for stabilisers, in other words most of the algorithm from Theorem 5.2. This is shown in Figure A.1. For each field size, $q = 3^{2m+1}$, generating sets for the stabilisers of 100 random points were computed, and the average running time for each call is listed. The amount of this time that was spent in discrete logarithm computations, SLP evaluations and in [11] is also indicated.

Note that when $2m + 1$ has a small enough prime divisor, finite field arithmetic in \mathbb{F}_q in MAGMA is much faster. More specifically, MAGMA uses Zech logarithms for finite fields up to a certain size, and for large fields it tries to find a subfield smaller than this size. If this is possible the arithmetic in the larger field will be much faster. To avoid jumps in the figure, and to measure field operations, we have in each case divided with the time taken to multiply 100000 random pairs of finite field elements.

We used the software package R (see [25]), to produce the figures.

All benchmarks were carried out using MAGMA V2.13-4, on a PC with an Intel Xeon CPU running at 2.8 GHz and with 1 GB of RAM. The highest value of m was 20, since higher values required too much memory.

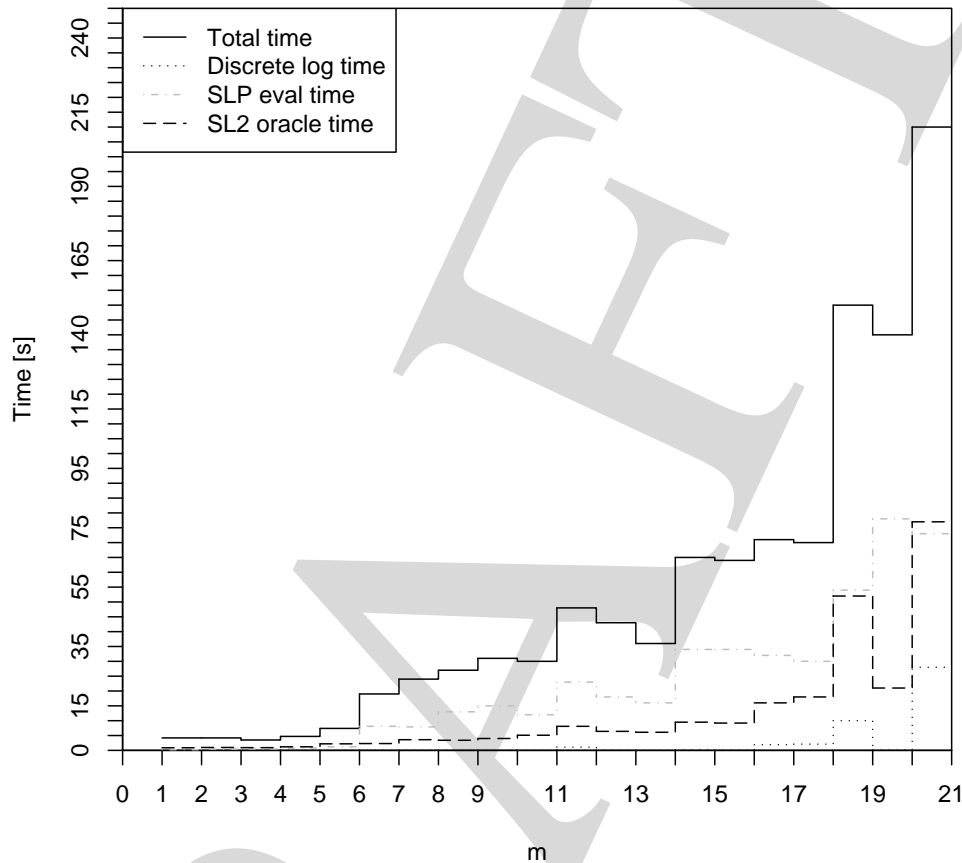


FIGURE A.1. Benchmark of stabiliser computation

REFERENCES

1. M. Aschbacher, *On the maximal subgroups of the finite classical groups*, *Invent. Math.* **76** (1984), 469–514.
2. H. Bäärnhielm, *Tensor decomposition of the Suzuki groups*, (2005), submitted.
3. ———, *Recognising the Suzuki groups in their natural representations*, *J. Algebra* **300** (2006), no. 1, 171–198.
4. ———, *Tensor decomposition of the ree groups*, (2006), preprint.
5. L. Babai, *Local expansion of vertex-transitive graphs and random generation in groups*, *Proc. 23rd ACM Symp. Theory of Computing* (Los Angeles), Association for Computing Machinery, 1991, pp. 164–174.
6. L. Babai, W. M. Kantor, P. P. Pálffy, and Á. Seress, *Black-box recognition of finite simple groups of Lie type by statistics of element orders*, *J. Group Theory* **5** (2002), 383–401.
7. W. Bosma, J. Cannon, and C. Playoust, *The Magma algebra system I: The user language*, *J. Symbolic Comput.* **24** (1997), 235–265.
8. J. N. Bray, *An improved method for generating the centraliser of an involution*, *Arch. Math. (Basel)* **74** (2000), no. 4, 241–245.
9. F. Celler and C. R. Leedham-Green, *Calculating the order of an invertible matrix*, *Groups and Computation II* (Larry Finkelstein and William M. Kantor, eds.), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 28, American Mathematical Society, 1997, pp. 55–60.

10. F. Celler, C. R. Leedham-Green, S. H. Murray, A. C. Niemeyer, and E. A. O'Brien, *Generating random elements of a finite group*, Comm. Algebra (1995), no. 23, 4931–4948.
11. M. D. E. Conder, C. R. Leedham-Green, and E. A. O'Brien, *Constructive recognition of $PSL(2, q)$* , Trans. Amer. Math. Soc. **358** (2006), 1203–1221.
12. S. P. Glasby, C. R. Leedham-Green, and E. A. O'Brien, *Writing projective representations over subfields*, J. Algebra **295** (2006), 51–61.
13. 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.
14. D. F. Holt, B. Eick, and E. A. O'Brien, *Handbook of Computational Group Theory*, Chapman & Hall/CRC, January 2005.
15. D. F. Holt and S. Rees, *Testing modules for irreducibility*, J. Austral. Math. Soc. Series A **57** (1994), 1–16.
16. B. Huppert and N. Blackburn, *Finite groups III*, Grundlehren Math. Wiss., vol. 243, Springer-Verlag, Berlin, Heidelberg, New York, 1982.
17. G. Ivanyos and K. Lux, *Treating the exceptional cases of the MeatAxe*, Experiment. Math. **9** (2000), 373–381.
18. G. Kemper, F. Lübeck, and K. Magaard, *Matrix generators for the Ree groups ${}^2G_2(q)$* , Comm. Algebra **29** (2001), no. 1, 407–413.
19. P. 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), 30–71.
20. C. R. Leedham-Green, *The computational matrix group project*, Groups and Computation III, Ohio State Univ. Math. Res. Inst. Publ., vol. 8, de Gruyter, 2001, pp. 113–121.
21. Charles R. Leedham-Green and Scott H. Murray, *Variants of product replacement*, Contemporary Mathematics **208** (2002).
22. V. M. Levchuck and Ya. N. Nuzhin, *Structure of Ree groups*, Algebra Logika **24** (1985), no. 1, 26–41.
23. D. S. Mitrinovic, J. Sándor, and B. Crstici, *Handbook of number theory, mathematics and its applications*, vol. 351, Kluwer Academic Publishers, 1996.
24. I. Pak, *The product replacement algorithm is polynomial*, Proc. 41st IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Press, 2000, pp. 476–485.
25. R Development Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2005, 3-900051-07-0.
26. R. Ree, *A family of simple groups associated with the simple Lie algebra of type G_2* , Amer. J. Math. **83** (1961), no. 3, 432–462.
27. Á. Seress, *Permutation group algorithms*, Cambridge Tracts in Mathematics, vol. 152, Cambridge University Press, 2003.
28. J. von zur Gathen and J. Gerhard, *Modern computer algebra*, 2nd ed., Cambridge University Press, Cambridge, 2003.
29. H. N. Ward, *On Ree's series of simple groups*, Trans. Amer. Math. Soc. **121** (1966), 62–89.
30. R. A. Wilson, *Finite simple groups*, preprint, 2005.

SCHOOL OF MATHEMATICAL SCIENCES, QUEEN MARY, UNIVERSITY OF LONDON, MILE END ROAD, LONDON E1 4NS, UNITED KINGDOM

URL: <http://www.maths.qmul.ac.uk/~hb/>

E-mail address: h.baarnhielm@qmul.ac.uk