LMS J. Comput. Math. 13 (2010) 307–319 © 2010 Authors
doi:10.1112/S1461157009000060

Normalizers of 2-subgroups in black-box groups

Peter Rowley and Paul Taylor

Abstract

In this paper we refine and extend the applicability of the algorithms in Bates and Rowley (*Arch. Math.* 92 (2009) 7–13) for computing part of the normalizer of a 2-subgroup in a black-box group. Supplementary materials are available with this article.

1. *Introduction*

In [3] algorithms are discussed for obtaining certain elements of $N_G(X)$ when G is a black-box group, X is a p-subgroup of G and p is a prime number. In practise these algorithms, which are Monte Carlo (though there is a potential sting in the tail; see the end of Section 3), exhaust the available memory unless X is 'small'. If X is a 2-group, then 'small' in [3] means that X has a characteristic series of subgroups whose successive factors are elementary abelian of order at most 2^5. The aim of the present paper is to confront this issue for these algorithms when X is a 2-group. As a by-product we produce algorithms that are able to handle 2-subgroups possessing characteristic series whose successive factors are elementary abelian of order at most 2^9 (although 2^{10} is in range if we are prepared to allow some uncertainty in the output).

From now on G stands for a black-box group and X a 2-subgroup of G of order 2^n. This means that the elements of G are encoded (not necessarily uniquely) by 0, 1-strings of uniform length and such that for $g, h \in G$, we can compute (a string representing) gh, g^{-1}, and determine whether $g = h$. Black-box groups were originally introduced in [2] (see [1, 9] for more recent developments). Also we set $N = N_G(X)$, $C = C_G(X)$, $\overline{N} = N/C$ and $K = O^{2'}(N)C$. We recall that $O^{2'}(N)$ is the normal subgroup of N whose quotient is the largest possible quotient of N of odd order, so $O^{2'}(N) = \langle S \mid S \in \mathrm{Syl}_2(N)\rangle$. Note that $\overline{K} = O^{2'}(\overline{N})$. The object of our attentions is, in fact, the subgroup K of N. At the heart of the algorithms in [3] is the case when X is an elementary abelian 2-group. For such an X the set of all maximal chains of subgroups of X plays an essential role. By a maximal chain of subgroups (or maximal flag) of X we mean the chain of subgroups of X

$$1 = X_0 < X_1 < X_2 < \ldots < X_n = X,$$

where $[X_j : X_{j-1}] = 2$, for $j = 1, \ldots, n$. Now using the algorithms of [3] to determine K we must, potentially, perform a certain routine upon each and every maximal chain in X. If, say, $n = 5, 6$ or 10, then the number of maximal chains in X is, respectively, 9765, 615 195 and 10 414 855 105 976 475. This explains the boundary of applicability (being $n = 5$) in [3].

The key, in the present work, to pin-pointing K is Lemma 2.1 together with subsequent refinements of this result given in Section 2. As a consequence, we can achieve our aims using smaller sets of maximal chains. By way of illustration, when $n = 6$ we can determine whether or not $K = C$, using only 1045 maximal chains of X, rather than all 615 195 maximal chains. If $C < K$, then this set of maximal chains will yield a subgroup M of K where $C < M$ and $\overline{M}$ is generated by all the involutions in $\overline{K}$. Regarding, as we may, $\overline{K}$ as a subgroup of $L \cong \mathrm{GL}_n(2)$, we then choose S to be a Sylow 2-subgroup of $\overline{M}$. The next step is to determine all the Sylow

Received 6 March 2009; revised 31 March 2010.

2000 Mathematics Subject Classification 20D06.

2-subgroups of $\overline{K}$ which contain S. This we do utilizing the chains associated with certain Sylow 2-subgroups of L which between them contain all subgroups T of L where $S < T$ and $[T : S] = 2$ (unless S is already a Sylow 2-subgroup of L). By iterating this procedure sufficiently often, we then obtain the set $\{T_1, \ldots, T_l\}$ of Sylow 2-subgroups of $\overline{K}$ containing S. This then quickly gives $\overline{K}$ as a subgroup of L. Translating this information back to G yields K. Further details of this and other aspects of the algorithm are discussed in Sections 2 and 3. Section 4 is devoted to the results of certain calculations carried out using the algorithm. Among these calculations, we consider the case when X is an elementary abelian subgroup in $\mathrm{Sp}_{16}(3)$ of order 2^7, and the case when X is a non-abelian subgroup in $\mathrm{GL}_{20}(2)$ of order 2^{12}. Our final section reflects upon how efficient the sets of chains employed in the algorithm are, as well as tabulating some related data.

Supplementary material is available via the multimedia link on the online article webpage. The aptly named folder `TwoGroupNormalizerAlgorithms` contains an executable implementation of the algorithm and is written to run using MAGMA [5] (though it could be adapted for GAP [7]). The folder `ReducedChainSets` lists the most efficient sets of chains currently available, as introduced in Section 3. Generators for the 2-group examples in Section 4 are given in the folder `ExampleCalculations`, so that the reader may verify these calculations. The `ReadMe` file gives further details.

2. *Notation and crowns*

Let n be a fixed natural number and set $L = \mathrm{GL}_n(2)$, the group of all $n \times n$ invertible matrices over GF(2). The natural n-dimensional GF(2)L-module will be denoted by V. Also, we let $m = n/2$ if n is even and $(n+1)/2$ if n is odd. We use $\mathcal{V}_m$ to denote the set of all m-dimensional subspaces of V. For $U \in \mathcal{V}_m$ we also set

$$Q_U = \{x \in L \mid [V, x] \leqslant U \leqslant C_V(x)\},$$

where $[V, x] = \langle v^x + v \mid v \in V \rangle$. We note that $U = C_V(Q_U)$ and $Q_U = O_2(\mathrm{Stab}_L(U))$. Furthermore, we have

$$|Q_U| = \begin{cases} 2^{m^2} & \text{if } n \text{ is even} \\ 2^{m(m-1)} & \text{if } n \text{ is odd.} \end{cases}$$

Letting $\mathcal{I}(L)$ denote the set of all involutions in L, we come to our first lemma. For each $U \in \mathcal{V}_m$ choose $T_U \in \mathrm{Syl}_2(\mathrm{Stab}_L(U))$, and set $\mathcal{J} = \{T_U \mid U \in \mathcal{V}_m\}$.

LEMMA 2.1. *We have*

$$\mathcal{I}(L) \subseteq \bigcup_{U \in \mathcal{V}_m} Q_U \subseteq \bigcup_{T \in \mathcal{J}} T.$$

Proof. Let $x \in \mathcal{I}(L)$. Since $[V, x] \leqslant C_V(x)$ and $\dim([V, x]) = \dim(V/C_V(x))$, there exists $U \in \mathcal{V}_m$ such that $[V, x] \leqslant U \leqslant C_V(x)$. Hence $x \in Q_U$. Because $Q_U \leqslant T_U$, we then get

$$x \in \bigcup_{U \in \mathcal{V}_m} Q_U \subseteq \bigcup_{T \in \mathcal{J}} T,$$

so proving the lemma. □

For our second lemma we introduce certain subsets of $\mathrm{Syl}_2(L)$. Let $U, W \in \mathcal{V}_m$ be such that $\dim(U \cap W) = m - 1$. So $\dim(\langle U, W \rangle / U \cap W) = 2$. We define

$$\mathcal{J}_U(W) = \{T \in \mathrm{Syl}_2(\mathrm{Stab}_L(W)) \mid T \text{ leaves } \langle U, W \rangle \text{ and } U \cap W \text{ invariant}\}.$$

https://doi.org/10.1112/S1461157009000060 Published online by Cambridge University Press

LEMMA 2.2. *Suppose* $U, W \in \mathcal{V}_m$ *with* $\dim(U \cap W) = m-1$. *If* $T \in \mathcal{J}_U(W)$, *then* $[Q_U : Q_U \cap T] = 2$.

Proof. Since $T \leqslant \mathrm{Stab}_L(W)$ and $T \in \mathcal{J}_U(W)$, T leaves $\langle U, W\rangle$, W and $U \cap W$ invariant. Thus, for suitable subspaces $V_{n-1}, V_{n-2}, \ldots, V_{m+2}, V_{m-2}, \ldots$ of V,

$$V_{n-1} > V_{n-2} > \ldots > V_{m+2} > \langle U, W\rangle > W > U \cap W > V_{m-2} > \ldots$$

is the maximal flag of V whose stabilizer in L is T (where $\dim V_i = i$). Let γ denote this maximal flag. Because $[V, Q_U] \leqslant U$ and $U \cap W \leqslant C_V(Q_U)$ we see that Q_U stabilizes the flag $\eta = \gamma \backslash \{W\}$. Hence $\langle T, Q_U\rangle$ leaves η invariant and so $\langle T, Q_U\rangle \leqslant P$, a minimal parabolic subgroup of L. If $\langle T, Q_U\rangle \neq P$, then $\langle T, Q_U\rangle = T$ and therefore Q_W and Q_U are both contained in T. Since Q_W and Q_U are L-conjugate and Q_W is weakly closed in T with respect to L (see [8, Section 7.5] for the definition of 'weakly closed'), this forces $Q_W = Q_U$. This is impossible as $C_V(Q_W) = W \neq U = C_V(Q_U)$. Thus $\langle T, Q_U\rangle = P$ and therefore, as $P/O_2(P) \cong \mathrm{GL}_2(2)$, we deduce that $T \cap Q_U = O_2(P) \cap Q_U$ has index 2 in Q_U. □

We now describe a configuration of subspaces in V which plays an important role in our algorithm.

DEFINITION 2.3. Suppose $U_1, U_2, U_3, U_4 \in \mathcal{V}_m$, and set $I = \{1, 2, 3, 4\}$. We call $\{U_1, U_2, U_3, U_4\}$ a crown if:

(i) $\bigcap_{i\in I} U_i = U_0$ where $\dim(U_0) = m-2$; and
(ii) for each $i \in I$, $U_i \cap U_j$ $(j \in I\backslash\{i\})$ are the three subspaces of U_i of dimension $m-1$ containing U_0.

REMARK 2.4. Suppose $\{U_1, U_2, U_3, U_4\}$ is a crown, $I = \{1, 2, 3, 4\}$, and set $U_0 = \bigcap_{i\in I} U_i$ and $U^0 = \langle U_1, U_2\rangle$. Then we note the following.

(i) $\dim(U^0) = m+1$ and $\dim(U^0/U_0) = 3$.

(ii) For all $i, j \in I$ with $i \neq j$, $U^0 = \langle U_i, U_j\rangle$ as $U_i = \langle U_i \cap U_1, U_i \cap U_2\rangle$ and $U_j = \langle U_j \cap U_1, U_j \cap U_2\rangle$.

(iii) Assume that $\{U_1, U_2, U_3, U_4'\}$ is a crown, and set $U_0' = U_1 \cap U_2 \cap U_3 \cap U_4'$. Then $U_0 = U_1 \cap U_2 \cap U_3 = U_0'$. Hence $U_1 \cap U_4'$ is one of the three $(m-1)$-dimensional subspaces of U_1 containing U_0 and so is equal to one of $U_1 \cap U_2$, $U_1 \cap U_3$ and $U_1 \cap U_4$. If, say, $U_1 \cap U_4' = U_1 \cap U_2$, then $U_4' = \langle U_1 \cap U_4', U_2 \cap U_4'\rangle = U_2$, whereas $U_2 \neq U_4'$. So $U_1 \cap U_4' \neq U_1 \cap U_2$ and similarly $U_1 \cap U_4' \neq U_1 \cap U_3$. Therefore $U_1 \cap U_4' = U_1 \cap U_4$. A similar argument yields $U_2 \cap U_4' = U_2 \cap U_4$ and therefore $U_4' = \langle U_1 \cap U_4', U_2 \cap U_4'\rangle = \langle U_1 \cap U_4, U_2 \cap U_4\rangle = U_4$. We conclude that two distinct crowns can have at most two subspaces in common.

Figure 1 shows a subspace lattice of the relevant subspaces which gave rise to the name.

LEMMA 2.5. *Suppose that* $\{U_1, U_2, U_3, U_4\}$ *is a crown and that* $T_k \in \mathcal{J}_{U_1}(U_k)$, $k = 2, 3, 4$. *Then for* $2 \leqslant i < j \leqslant 4$,

$$T_i \cap T_j \cap Q_{U_1} = T_2 \cap T_3 \cap T_4 \cap Q_{U_1}.$$

Proof. Set $Q = Q_{U_1}$ and, without loss of generality, we may suppose that $i = 2$ and $j = 3$. So $U_1 = C_V(Q)$. Put $U_0 = U_1 \cap U_2 \cap U_3 \cap U_4$, $U^0 = \langle U_1, U_2\rangle$ and $R = T_2 \cap T_3 \cap Q$. So in order to prove $R = T_2 \cap T_3 \cap T_4 \cap Q$ we must show that $R \leqslant T_4$. Since

$$R \leqslant T_2 \cap T_3 \leqslant \mathrm{Stab}_L(U_2) \cap \mathrm{Stab}_L(U_3),$$

R leaves $U_2 \cap U_3$ invariant. Also, as $U_1 \cap U_2 \leqslant U_1 = C_V(Q)$ and $U_0 \leqslant U_1 = C_V(Q)$, R leaves $U_1 \cap U_2$ and U_0 invariant. Thus the other subspace of U_2 of dimension $m-1$ containing U_0,

https://doi.org/10.1112/S1461157009000060 Published online by Cambridge University Press

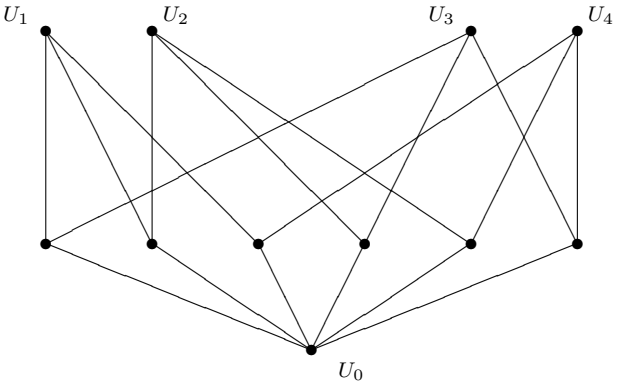

FIGURE 1. *Subspace lattice of a crown.*

namely $U_2 \cap U_4$, must be R-invariant. Because $U_4 = \langle U_2 \cap U_4, U_1 \cap U_4 \rangle$ and $U_1 \cap U_4 \leqslant U_1 = C_V(Q)$ we deduce that U_4 is R-invariant. Thus, as R acts trivially on V/U^0 and $U_1 \cap U_4$, R will stabilize the maximal flag whose stabilizer in L is T_4. As a consequence $R \leqslant T_4$, and Lemma 2.5 holds. □

LEMMA 2.6. *Suppose that $\{U_1, U_2, U_3, U_4\}$ is a crown and that $T_k \in \mathcal{J}_{U_1}(U_k)$, $k = 2, 3, 4$. Then for $2 \leqslant i < j \leqslant 4$,*

$$T_i \cap Q_{U_1} \neq T_i \cap T_j \cap Q_{U_1}.$$

Proof. Again set $Q = Q_{U_1}$ and assume without loss of generality that $i = 2$ and $j = 3$. Put $U_0 = U_1 \cap U_2 \cap U_3 \cap U_4$ and $U^0 = \langle U_1, U_2 \rangle$. Now $\mathrm{Stab}_L(U_1)$ contains subgroups H_1 and H_2 with $H_1 \cong \mathrm{GL}_{n-m}(2)$ and $H_2 \cong \mathrm{GL}_m(2)$. Moreover, V/U_1 is the natural $\mathrm{GF}(2)H_1$-module, while U_1 is the natural $\mathrm{GF}(2)H_2$-module with H_1 acting trivially on U_1 and H_2 acting trivially on U/U_1. Thus we may select $x \in \mathcal{I}(L)$ such that $[V, x] = U_1 \cap U_2$, $C_V(x) \geqslant U_1$ and $C_{V/U_0}(x) \not\geqslant U^0/U_0$. So $x \in Q$ and, as x centralizes $V/U_1 \cap U_2$ and $U_1 \cap U_2$, $x \in T_2$. We claim that $x \notin T_3$. For if $x \in T_3$, then x must leave U_3 invariant. Therefore

$$[U_3, x] \leqslant [V, x] \cap U_3 = U_1 \cap U_2 \cap U_3 = U_0.$$

Hence x centralizes U_3/U_0 and consequently x centralizes $\langle U_1, U_3 \rangle / U_0 = U^0/U_0$ whereas $C_{V/U_0}(x) \not\geqslant U^0/U_0$. Thus $x \in (T_2 \cap Q) \backslash T_3$, so proving the lemma. □

THEOREM 2.7. *Suppose that $\{U_1, U_2, U_3, U_4\}$ is a crown and let $T_k \in \mathcal{J}_{U_1}(U_k)$, $k = 2, 3, 4$. Then $Q_{U_1} \subseteq T_2 \cup T_3 \cup T_4$.*

Proof. By Lemmas 2.2 and 2.6 $[Q_{U_1} : Q_{U_1} \cap T_i] = 2$ for $i = 2, 3, 4$ and $[Q_{U_1} : Q_{U_1} \cap T_i \cap T_j] = 4$ for $2 \leqslant i < j \leqslant 4$. Since $Q_{U_1} \cap T_i \cap T_j = Q_{U_1} \cap T_2 \cap T_3 \cap T_4$ $(2 \leqslant i < j \leqslant 4)$ by Lemma 2.5, we see that

$$Q_{U_1} = (Q_{U_1} \cap T_2) \cup (Q_{U_1} \cap T_3) \cup (Q_{U_1} \cap T_4),$$

whence the theorem holds. □

COROLLARY 2.8. *Let $\{\gamma_1, \gamma_2, \ldots, \gamma_r\}$ be a set of crowns which are pairwise disjoint. Then there exists a set $\mathcal{J}$ of Sylow 2-subgroups of L such that $|\mathcal{J}| = |\mathcal{V}_m| - r$ and $\mathcal{I}(L) \subseteq \bigcup_{T \in \mathcal{J}} T$.*

https://doi.org/10.1112/S1461157009000060 Published online by Cambridge University Press

Proof. For each $\gamma_i = \{U_1, U_2, U_3, U_4\}$, we choose a $T_k \in \mathcal{J}_{U_1}(U_k)$, $k = 2, 3, 4$, and for each $U \in \mathcal{V}_m$ not appearing in a crown in our set we choose an arbitrary $T_U \in \mathrm{Syl}_2(\mathrm{Stab}_L(U))$, and these form our set $\mathcal{J}$. Evidently $\mathcal{J}$ has the required size, and Lemma 2.1 and Theorem 2.7 give that $\mathcal{I}(L) \subseteq \bigcup_{T\in\mathcal{J}} T$. □

A small example of applying Corollary 2.8 is when $n = 4$. Then $|\mathcal{V}_2| = 35$ and it is possible to find a set of 6 crowns which are pairwise disjoint. So there exists a set of 29 Sylow 2-subgroups of $\mathrm{GL}_4(2)$ whose union contains all the involutions of $\mathrm{GL}_4(2)$.

In fact, we can achieve the same result as in Corollary 2.8 with a carefully-chosen set of crowns that are not disjoint, as the following result demonstrates.

THEOREM 2.9. *Let $\{\gamma_1, \gamma_2, \ldots, \gamma_r\}$ be a set of crowns, and in each of these crowns fix a space $U_i \in \gamma_i$. Suppose that for any $U \in \gamma_i \cap \gamma_j$ $(i \neq j)$, the following conditions hold:*

(i) *$U \neq U_i$ and $U \neq U_j$; and*
(ii) *$U \cap U_i = U \cap U_j$ and $\langle U, U_i\rangle = \langle U, U_j\rangle$.*

Then there exists a set $\mathcal{J}$ of Sylow 2-subgroups of L such that $|\mathcal{J}| = |\mathcal{V}_m| - r$ and $\mathcal{I}(L) \subseteq \bigcup_{T\in\mathcal{J}} T$.

Proof. We note that wherever a space U occurs in two crowns γ_i, γ_j, condition (ii) ensures that $\mathcal{J}_{U_i}(U) = \mathcal{J}_{U_j}(U)$. So for each crown γ_i we may, as in Corollary 2.8, take $T_k \in \mathcal{J}_{U_i}(U_k)$ for $k \in \{1, 2, 3, 4\}\backslash\{i\}$. Where a space occurs in two crowns, we are in both cases selecting a crown from the same set, so one choice will suffice. (Note that a space U cannot occur in three crowns $\gamma_i, \gamma_j, \gamma_k$ as this would require four distinct spaces U, U_i, U_j, U_k lying between $U \cap U_i$ and $\langle U, U_i\rangle$, whereas $\dim(\langle U, U_i\rangle/U \cap U_i) = 2$.) Condition (i) ensures that for a crown γ_i the space U_i never occurs in another crown, so we achieve a covering of every involution with $|\mathcal{V}_m| - r$ subgroups, as required. □

3. *The algorithm and its implementation*

We use all the notation introduced earlier and begin this section by recalling some notation from [3]. For $\mathcal{C}$, the maximal chain of subgroups in X

$$1 = X_0 < X_1 < X_2 < \ldots < X_n = X,$$

we set

$$K^{\mathcal{C}} = \{g \in G \mid [X_i, g] \leqslant X_{i-1}, \text{ for all } i = 1, \ldots, n\}.$$

Clearly $K^{\mathcal{C}} \leqslant N$ and, by [8, Theorem 5.3.6], $\overline{K^{\mathcal{C}}}$ is a 2-group. Also, K is generated by $K^{\mathcal{C}}$ as $\mathcal{C}$ ranges over all maximal chains of X. To calculate elements of $K^{\mathcal{C}}$ we adapt the involution centralizer algorithm for black-box groups due to Bray [4] as follows (and refer to its use as Braying a chain).

ALGORITHM 3.1.

Input: The black-box group G and an elementary abelian 2-subgroup X of G;
groups X_j $(j = 0, 1, \ldots, n)$, a maximal chain of subgroups of X;
elements $x_j \in X_j\backslash X_{j-1}$ for $i = 1, \ldots, n$.

(i) Set $M_0 = G$.
(ii) For $j = 1, \ldots, n$, perform steps (iii)–(vi).
(iii) Initialize A to be the empty set.
(iv) Choose a random element h of M_{j-1} and determine the smallest $k \in \mathbb{N}$ for which $[x_j, h]^k \in X_{j-1}$.

https://doi.org/10.1112/S1461157009000060 Published online by Cambridge University Press

(v) If k is even, then define

$$A = A \cup \{[x_j, h]^{k/2}, [x_j, h^{-1}]^{k/2}\},$$

while if k is odd, define

$$A = A \cup \{h[x_j, h]^{(k-1)/2}\}.$$

(vi) After a specified number of random choices for h, set $M_j = \langle A \rangle$.

Output: The group M_n.

We remark that $\{x_i \mid i = 1, \ldots, n\}$ is a generating set for X and so yields an explicit listing of the elements of X.

PROPOSITION 3.2. *The group M_n output by Algorithm 3.1 is contained in $N_G(X)$.*

Proof. For $j = 1$, Algorithm 3.1 simply performs the algorithm in [4], so if the random choices of h yield enough generating elements, then $M_1 = C_G(x_1) = C_G(X_1)$. For $j = 1, \ldots, n$, set $N_j = N_G(X_j)$ and $\overline{N_j} = N_j/X_j$. Now assume that for some $j \in \{1, \ldots, n\}$ we have constructed a subgroup M_{j-1} of G for which $X_{j-1} \trianglelefteq M_{j-1}$. Then $M_{j-1} \leqslant N_{j-1}$, $\overline{x_j}$ is an involution in $\overline{N}_{j-1}$ and $N_{N_{j-1}}(X_j) = (N_{j-1} \cap N_j)$ is the inverse image in N_{j-1} of the centralizer of $\overline{x_j}$ in $\overline{N}_{j-1}$. □

We now direct our attention to the important case when X is an elementary abelian 2-group (for a general 2-group X we apply this special case by working our way up a characteristic series of subgroups of X all of whose successive factors are elementary abelian groups). In particular, suppose we have Y a characteristic subgroup of X with X/Y elementary abelian, and that we have calculated $K_0 \leqslant N_G(Y)$. Then we can easily adapt the algorithm to compute $K \leqslant N_G(X) = N_{N_G(Y)}(X)$ by setting $G = K_0$ and performing all calculations in X modulo Y. See the end of Section 4 for more on the general 2-group case.

We fix once and for all a GF(2) basis for X and an n-dimensional GF(2)L-module V whose basis is in one-to-one correspondence with that of X. Then a maximal chain of subgroups of X will correspond to a maximal flag of V. Moreover, as the maximal flags of V are in one-to-one correspondence with the set of Sylow 2-subgroups of L, we have a one-to-one correspondence between the maximal chains in X and the Sylow 2-subgroups of L. We now describe the major steps in the algorithm.

ALGORITHM 3.3.

Input: The black-box group G and an elementary abelian 2-subgroup X of G;
groups X_j $(j = 0, 1, \ldots, n)$, a maximal chain of subgroups of X;
elements $x_j \in X_j \backslash X_{j-1}$ for $i = 1, \ldots, n$; the group $L = \mathrm{GL}_n(2)$ and the associated vector space V.

(i) Select a set $\mathcal{J}$ of Sylow 2-subgroups of L with the property that

$$\mathcal{I}(L) \subseteq \bigcup_{T \in \mathcal{J}} T.$$

(See Algorithm 3.5.) It is crucial (for large n) that $\mathcal{J}$ is as small as possible and that the listing of $\mathcal{J}$ is well organized for carrying out step (ii). Let $\mathcal{C}(\mathcal{J})$ denote the maximal chains of subgroups of X which correspond to the Sylow 2-subgroups in $\mathcal{J}$.

(ii) Bray every chain $\gamma \in \mathcal{C}(\mathcal{J})$, and let $\overline{M}$ be the group generated by all the involutions produced by this. If $\overline{M} = 1$, then set $\overline{K} = 1$ and terminate.

https://doi.org/10.1112/S1461157009000060 Published online by Cambridge University Press

(iii) Considering $\overline{M}$ as a subgroup of L by the correspondence between X and V, we choose $S \in \mathrm{Syl}_2(\overline{M})$.
(iv) Select a set $\mathcal{S}$ of Sylow 2-subgroups of L such that for every subgroup T with $S < T < L$ and $[T : S] = 2$, $T \leqslant P$ for some $P \in \mathcal{S}$ (if $S \notin \mathrm{Syl}_2(L)$). Then Bray the corresponding chains $\mathcal{C}(\mathcal{S})$.
(v) Apply step (iv) with S replaced with each of the subgroups formed by Braying each chain in $\mathcal{C}(\mathcal{S})$. Repeat with the resulting subgroups, working up index 2 at a time until we arrive at a set of subgroups for which step (iv) finds nothing new. Call this set $\{T_1, \ldots, T_l\}$.
(vi) $\overline{K} = \langle \overline{M}, T_1, \ldots, T_l \rangle$.

Output: the group $\overline{K}$, and elements of N generating K (modulo C).

PROPOSITION 3.4. *The group $\overline{K}$ output by Algorithm 3.3 is indeed $O^{2'}(\overline{N})$.*

Proof. The set of chains formed in step (i) is by construction sufficient that Braying them will find every involution in $\overline{K}$. If $\overline{K}$ has no involutions then it is trivial, so the algorithm terminates. Otherwise, the group $\overline{M}$ formed in step (ii) is $\langle x \in \overline{K} \mid x^2 = 1 \rangle \trianglelefteq \overline{K}$. After selecting an $S \in \mathrm{Syl}_2(\overline{M})$, steps (iv)–(v) then work up index 2 at a time to find $\{T_1, \ldots, T_l\}$, a set of all Sylow 2-subgroups of $\overline{K}$ containing S.

Step (vi) follows from the fact that $\overline{K}$ is generated by its Sylow 2-subgroups. Suppose $T \in \mathrm{Syl}_2(\overline{K})$. Then since $\overline{M} \trianglelefteq \overline{K}$, we have that $T \cap \overline{M} \in \mathrm{Syl}_2 \overline{M}$. Hence $T \cap \overline{M} = S^h \leqslant T$ for some $h \in \overline{M}$. Then $T^{h^{-1}}$ is a Sylow 2-subgroup of $\overline{K}$ containing S, and so $T^{h^{-1}} = T_i$ for some $i \in \{1, \ldots, l\}$. Hence $T = T_i^h$, and so $T \leqslant \langle \overline{M}, T_1, \ldots, T_l \rangle$. Therefore $\overline{K} = \langle \overline{M}, T_1, \ldots, T_l \rangle$. □

After some simple initialization tasks (such as setting up the correspondence between X and V), the algorithm's first step must be to determine the chains for the relevant value of n. In fact, the chains for all smaller values are also needed as will be seen later. Since our set of initial chains (formed from a set of Sylow 2-subgroups of L covering all the involutions of L) is independent of the choice of G and X, we create and store these sets for each choice of n beforehand, and the algorithm loads these pre-formed chains.

It may be tempting to attempt to make these sets of chains which cover all the involutions of L by random searching. However, this approach almost always ends in tears (although for the case $n = 3$, it is possible to search exhaustively and learn that five chains suffice to cover the involutions of L). We generate such sets of chains by the following procedure.

ALGORITHM 3.5.

Input: the group $L = \mathrm{GL}_n(2)$ and the associated vector space V.
(i) A set containing all of the m-dimensional subspaces of V is formed. This is done by generating one such subspace, computing a transversal of its stabilizer across L, and acting on the subspace by each element of the transversal.
(ii) We find a set of crowns $\{\gamma_1, \ldots, \gamma_r\}$ as large as possible satisfying the conditions in Theorem 2.9 by the following heuristic procedure. First, we generate the set of all $(m+1)$-dimensional subspaces of V, then attempt to find as many crowns as possible having each such space as U^0 (in the notation of Section 2). Given a particular space U^0, we generate an arbitrary crown beneath it (choosing, say, U_1, U_2 and U_0 allows us to quickly complete the whole crown), and initialize a set with this crown. We then repeat a process whereby, for each space U_i $(i = 2, 3, 4)$ in each crown in the set (ignoring, of course, any space already used in two crowns), we form a new crown $\{U_1', U_2', U_3', U_4'\}$ with $U_2' = U_i$, $U_1 \cap U_i = U_1' \cap U_2'$ and $U^0 = \langle U_1', U_2' \rangle$. If the crown shares no other spaces with any other crown, it is

https://doi.org/10.1112/S1461157009000060 Published online by Cambridge University Press

added to the set. Once an iteration of this process yields no new crowns, we move to the next space U^0.

(iii) For the spaces U_i $(i = 2, 3, 4)$ in each crown found, we form a chain corresponding to a relevant Sylow 2-subgroup of L by first fixing the spaces $U_i \cap U_1$, U_i, $\langle U_i, U_1\rangle$, and building an arbitrary maximal flag around them. For every m-space in $\mathcal{V}_m$ not used in a crown, a chain is built arbitrarily around it. In order to store these chains more efficiently, and to speed up the process of Braying the chains, they are stored in sets grouped by virtue of their agreeing on sections at the start of the chains. So that if a set of chains share a common first three elements, they are stored as a 'root' of length 3 and a set of 'branches' of length $n - 3$ (each individual chain is stored as a sequence of representative vectors in V).

Output: The set of chains formed.

PROPOSITION 3.6. *Every involution in L is contained in a Sylow 2-subgroups of L corresponding to one of the chains in the set formed by Algorithm 3.5.*

Proof. It suffices to show that our set of crowns meets the conditions imposed in Theorem 2.9. The space U_1 in each crown is by design not used in any other crown, so condition (i) holds. The construction of crowns sharing spaces described ensures condition (ii) is met. □

We note that, while the algorithm is black-box on the input group G, the group $L \cong \mathrm{GL}_n(2)$ is used only internally by the algorithm, and since these groups are sufficiently small (for the values of n we are considering) and well-understood to be computed in efficiently, we assume that we can perform any required computation within this group and its associated module V (as is easily possible in, say, MAGMA). Below we describe how the algorithm carries out some of the steps which must be performed in the large group G, and how the further sets of chains are determined in steps (iv) and (v) of Algorithm 3.3.

We first determine $C = C_G(X)$ by repeated application of Bray's algorithm to find elements in $C_G(x_1)$, then $C_{C_G(x_1)}(x_2) = C_G(x_1) \cap C_G(x_2)$, and after n such steps, arriving at $\bigcap_{i=1}^{n} C_G(x_i) = C_G(X)$.

When Braying chains, the chains are considered in the sets in which they are stored in Algorithm 3.5(iii) above. First, we Bray up to the end of the common 'root'. If all the random elements chosen by this point can be seen to be trivial in $\overline{K}$ (that is, are elements of C), we discard all the chains in the set. Otherwise, the elements found from Braying the root are used as a starting point to Bray the remaining 'branches' of the chains.

In Algorithm 3.1, the process of Braying a chain

$$1 = X_0 < X_1 < \ldots < X_n = X$$

consists at each stage of taking a specified number of random elements from M_{j-1} and applying our modification of Bray's algorithm to find elements normalizing X_j, which we use to generate M_j. However, in the MAGMA implementation of the algorithm provided, we avoid the possibly costly step of generating random group elements at these intermediate stages by using the elements found by Braying at each stage directly as the 'random' input elements for the next stage.

It is necessary in steps (iv) and (v) of Algorithm 3.3 to be able, given a 2-subgroup $\overline{R}$ of $\overline{K}$, to select a set $\mathcal{S}$ of Sylow 2-subgroups of L having the property that every $\overline{T}$ with $\overline{R} < \overline{T} < \overline{K}$ and $[\overline{T} : \overline{R}] = 2$ is contained in some $P \in \mathcal{S}$. Clearly it suffices that for every t such that $\langle \overline{R}, t\rangle = \overline{T}$ for some $\overline{T}$ as above, t is contained in some member of $\mathcal{S}$. The following algorithm performs this task (bypassing, in the case where $|\mathcal{S}| \neq 1$, the creation of the set $\mathcal{S}$ and instead forming the chains $\mathcal{C}(\mathcal{S})$ directly).

https://doi.org/10.1112/S1461157009000060 Published online by Cambridge University Press

ALGORITHM 3.7.

Input: The group $L = \mathrm{GL}_n(2)$ and its associated vector space V;
a 2-subgroup R of L.

(i) Calculate $N_L(\overline{R})$.

(ii) If $N_L(\overline{R})$ is a 2-group, output a set containing one chain corresponding to a Sylow 2-subgroup S with $N_L(\overline{R}) \leqslant S$. This is calculated by repeatedly taking normalizers of $N_L(\overline{R})$ until an $S \in \mathrm{Syl}_2(L)$ is found, and the corresponding chain is taken.

(iii) Otherwise, we must create a new set of several chains. A new basis is formed for V, which begins with a basis for $C_V(\overline{R})$, followed by vectors extending it to a basis for (the inverse image of) $C_{V/C_V(\overline{R})}(\overline{R})$, and so on up to V. Let $d_1, \ldots, d_r$ be the dimensions of the centralizers formed (so $d_1 + \ldots + d_r = n$). We form a new set of chains consisting of all concatenations of chains from our initial sets for $n = d_1, \ldots, d_r$, giving a set of size $d_1 d_2 \ldots d_r$. This set, transformed back to our standard basis, forms our new set of chains.

Output: the set of chains formed.

PROPOSITION 3.8. *The Sylow 2-subgroups corresponding to the set of chains constructed by Algorithm 3.7 contains every t described above.*

Proof. Any such t must lie in $N_L(\overline{R})$. The case where $N_L(\overline{R})$ is a 2-group is trivial. Otherwise we note that an element t of the form described must act as an involution on each of the centralizers formed in step (iii) of the algorithm, so chains formed from our initial sets, designed to cover all involutions, clearly suffice. □

The largest sets created by Algorithm 3.7 occur when V is decomposed into two spaces of dimensions 1, $n-1$, whence the new chain set is three times the size of the set of initial chains for the $n-1$ case. Even here, this set is substantially smaller than the initial set of chains.

Beyond the fundamental risk of error posed by running too few iterations of the Bray algorithm, another risk must be considered. In Bray's original algorithm in [4] to find elements centralizing an involution, consider a group G containing an involution t. Suppose we have that $t \in X \leqslant C_G(t)$ with $X \trianglelefteq G$ and X a 2-group. Then for any $h \in G$, we have that $[t, h] = t^{-1}t^h \in X$. Hence since X is a 2-group, $[t, h]$ has even order $2m$ (in particular two-power order), and the algorithm returns $[t, h]^{m/2}$ or $[t, h^{-1}]^{m/2}$, both of which are in X. Hence we never generate the full centralizer (aside from if we happen to choose a random element $h \in C_G(t) \backslash X$).

In our present algorithm, we can attempt to detect when this situation may have arisen. The value k computed in step (iii) of Algorithm 3.1 corresponds to m in Bray's algorithm. So if for some chain we get exclusively two-power values of k at some level $j \in \{1, \ldots, n\}$ of the algorithm, we may have encountered this undesired scenario. In this case, we may be able to generate a new chain to replace it, and apply the algorithm to this instead, hoping for a better outcome. The new chain must of course satisfy any requirements to include particular spaces that the old chain met. If we are unable to replace the chain, or if several replacements fail in this regard we may have to concede defeat and continue, aware of the possibility that we may fail to generate K fully.

4. *Calculations using the algorithm*

All calculations were carried out on a Unix machine with 8 GB of memory and a 3.2 GHz processor, running MAGMA version 2.11-15.

Given the group $G = \mathrm{Sp}_{12}(3)$, we select an elementary abelian 2-group X of order 2^6 (generators being given in `ExampleCalculations`). Using its standard normalizer routine,

https://doi.org/10.1112/S1461157009000060 Published online by Cambridge University Press

MAGMA will calculate the normalizer $N_G(X)$ in 158.2 seconds. Our algorithm will calculate K (which in this case is the full normalizer) in 138.66 seconds, Braying each chain with 10 random elements of G. Moving to a larger group, the advantages of the present algorithm become more evident. Taking $G = \mathrm{Sp}_{16}(3)$, with our elementary abelian group X (again see `ExampleCalculations` for its generators) having order 2^7, our algorithm computes K, having order $2^{12} \cdot 3^2 \cdot 5 \cdot 7$, in 1 0014.5 seconds, while the standard MAGMA function exhausts the available memory and fails to produce an output.

Taking $G = J_4$, the largest Janko simple group, the smallest available representation in the online ATLAS [10] is as a 112×112 matrix group over GF(2). Using (in ATLAS [6] notation) $X \cong 2^7$ in the maximal subgroup $2^{11} : M_{24}$, our algorithm takes 57 538 seconds Braying 20 random elements in each chain to return a group K having order 2^{15}.

We consider an example in the case where X is a non-elementary abelian 2-group. We take $G = \mathrm{Sym}(20)$ and $X = \Phi(P)$, where $P \in \mathrm{Syl}_2(G)$. Then X may be decomposed into a chain of characteristic subgroups with elementary abelian factors having orders $2,\ 2^5,\ 2^6$. (The Frattini subgroup $\Phi(Y)$ of a 2-group $Y = \langle y_1, \ldots, y_m \rangle$ is given by $\langle [y_i, y_j], y_i^2 \mid i, j = 1, \ldots, m \rangle$; see [8, Theorem 5.1.3]. So $\Phi(Y)$ can be calculated in a black-box group, and gives us our required subgroups.) In this small representation, both the present algorithm and the standard MAGMA normalizer function quickly compute the normalizer N to be P of order 2^{18}. However, if we now consider X represented as a group of permutation matrices over GF(2) and let $G = \mathrm{GL}_{20}(2)$, the standard MAGMA function does not return an output. Our algorithm, using 1000 random elements on each chain, will return a group K having order $2^{31} \cdot 3$ in 760.4 seconds.

TABLE 1. *Crowns: Some statistics.*

n	Number of maximal chains	$\lvert\mathcal{V}_m\rvert$	Crowns found	$\lvert\mathcal{C}(\mathcal{J})\rvert$
3	21	7	1	6
4	315	35	7	28
5	9765	155	41	114
6	615 915	1395	350	1045
7	78 129 765	11 811	3208	8603
8	19 923 090 075	200 787	54 936	145 851
9	10 180 699 028 325	3 309 747	926 280	2 383 467

TABLE 2. $n = 5$.

Chains/inv.	All involutions	t_1^L	t_2^L
1	1323	0	1323
2	2478	0	2478
3	1501	0	1501
4	800	0	800
5	284	0	284
6	88	1	87
7	37	4	33
8	30	27	3
9	46	45	1
10	48	48	0
11	65	65	0
12	90	90	0
13	67	67	0
14	52	52	0
15	34	34	0
16	16	16	0
17	9	9	0
18	1	1	0
19	2	2	0
20	2	2	0
21	2	2	0

https://doi.org/10.1112/S1461157009000060 Published online by Cambridge University Press

TABLE 3. $n = 6$.

Chains/inv.	All involutions	t_1^L	t_2^L	t_3^L
1	60 704	0	0	60 704
2	70 698	0	0	70 698
3	52 608	429	0	52 179
4	32 491	2286	0	30 205
5	18 826	5801	0	13 025
6	15 497	10 531	0	4966
7	16 685	14 891	0	1794
8	18 592	18 023	0	569
9	19 128	18 974	0	154
10	17 537	17 490	0	47
11	15 075	15 061	0	14
12	11 924	11 920	0	4
13	8342	8342	0	0
14	5613	5612	0	1
15	3375	3375	0	0
16	1953	1953	0	0
17	1032	1032	0	0
18	555	555	0	0
19	252	252	0	0
20	94	94	0	0
21	58	58	0	0
22	17	17	0	0
23	8	8	0	0
24	5	5	0	0
25	1	1	0	0
26	0	0	0	0
45	1	0	1	0
46	1	0	1	0
47	0	0	0	0
48	2	0	2	0
49	10	0	10	0
50	5	0	5	0
51	13	0	13	0
52	8	0	8	0
53	15	0	15	0
54	26	0	26	0
55	23	0	23	0
56	43	0	43	0
57	39	0	39	0
58	54	0	54	0
59	51	0	51	0
60	69	0	69	0
61	85	0	85	0
62	97	0	97	0
63	89	0	89	0
64	102	0	102	0
65	107	0	107	0
66	85	0	85	0
67	104	0	104	0
68	117	0	117	0
69	95	0	95	0
70	111	0	111	0
71	84	0	84	0
72	85	0	85	0
73	64	0	64	0
74	52	0	52	0
75	43	0	43	0
76	51	0	51	0
77	46	0	46	0
78	42	0	42	0
79	29	0	29	0
80	30	0	30	0

https://doi.org/10.1112/S1461157009000060 Published online by Cambridge University Press

TABLE 3. *(Continued.)*

Chains/inv.	All involutions	t_1^L	t_2^L	t_3^L
81	18	0	18	0
82	19	0	19	0
83	9	0	9	0
84	6	0	6	0
85	6	0	6	0
86	4	0	4	0
87	6	0	6	0
88	1	0	1	0
89	0	0	0	0
90	0	0	0	0
91	0	0	0	0
92	1	0	1	0
93	0	0	0	0
94	0	0	0	0
95	0	0	0	0
96	0	0	0	0
97	1	0	1	0
98	0	0	0	0
99	0	0	0	0
100	1	0	1	0

The present algorithm fails where the group X has elementary abelian factors of order 2^{10} or above, since we are unable to create the sets of initial chains (specifically, MAGMA fails to compute the transversal required in Algorithm 3.5(i)). However, we may still use the same techniques to find some normalizing elements, albeit with no certainty that we have found the whole normalizer. Suppose X is elementary abelian of order 2^{10} (though larger orders are still in range). Then we may, for example, create a set of chains consisting of a random sampling of chains from the $n = 9$ case above each 1-space of V. Using this approach, we successfully computed the normalizer of $X \cong 2^{10} \leqslant 2^{10} : M_{22}$ in the sporadic simple group Fi_{22}, in a matrix representation of degree 78 over GF(2).

5. *Covering the involutions of L: some statistics*

Recall that $|\mathcal{V}_m|$ is the number of m-spaces of an n-dimensional vector space, and is an upper bound for the number of chains necessary in order to cover the involutions of L. In Table 1, the third column gives the size of the largest set of crowns we have found using the method outlined in Section 3, meeting the constraints of Theorem 2.9 (this set being used to create the initial chains is given in the `ReducedChainSets` folder). Note that in the case $n = 3$ there is, in fact, only one crown. This may be seen by observing that if there were two distinct crowns then five of the 1-spaces that are the intersections of the 2-spaces in the two crowns must be the same. This then yields that the two crowns intersect in at least three 2-spaces, whence the crowns are equal by Remark 2.4(iii).

There remains the issue of how efficient our sets $\mathcal{J}$ are in covering all the involutions of L. Tables 2 and 3, for the cases $n = 5$ and $n = 6$, seem to indicate that there is not too much redundancy.

These tables give, for each value in the first column, the number of involutions that are to be found in exactly that many Sylow 2-subgroups (or equivalently chains) in the smallest sets we have found. We also give the breakdown of this data into conjugacy classes of L. In Table 2, t_1 is an involution with $C_V(t_1)$ having dimension four, and t_2 is an involution with $C_V(t_2)$ having dimension three. In Table 3, t_1 is an involution with $C_V(t_1)$ having dimension four, t_2 is an involution with $C_V(t_2)$ having dimension five, and t_3 is an involution with $C_V(t_3)$ having dimension three.

https://doi.org/10.1112/S1461157009000060 Published online by Cambridge University Press

Acknowledgement

We thank the referee for a number of useful suggestions which have improved this paper.

References

1. L. Babai, W. M. Kantor, P. P. Pálfy and A. 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.
2. L. Babai and E. Szemerdi, 'On the complexity of matrix group problems I', *Proc. 25th IEEE Symp. Found. Comp. Sci.*. Palm Beach, FL, 1984, 229–240.
3. C. J. Bates and P. J. Rowley, 'Normalizers of p-subgroups in finite groups', *Arch. Math.* 92 (2009) 7–13.
4. J. N. Bray, 'An improved method for generating the centralizer of an involution', *Arch. Math.* 74 (2000) 241–245.
5. J. J. Cannon and C. Playoust, 'An introduction to algebraic programming with Magma', *Draft* (1997).
6. J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker and R. A. Wilson, *Atlas of finite groups* (Clarendon, Oxford, 1985).
7. The GAP Group, '*GAP—groups, algorithms, and programming, version 4.3*', 2002, http://www.gap-system.org.
8. D. Gorenstein, *Finite groups* (Harper and Row, New York, 1968).
9. W. M. Kantor and A. Seress, 'Black box classical groups', *Mem. Amer. Math. Soc.* 149 (2001) no. 708,.
10. R. A. Wilson, P. G. Walsh, J. Tripp, I. A. Suleiman, S. Rogers, R. A. Parker, S. P. Norton, S. A. Linton and J. N. Bray, 'Atlas of finite group representations', http://brauer.maths.qmul.ac.uk/Atlas/v3/.

Peter Rowley
School of Mathematics
The University of Manchester
Oxford Road, Manchester M13 9PL
United Kingdom

peter.j.rowley@manchester.ac.uk

Paul Taylor
School of Mathematics
The University of Manchester
Oxford Road, Manchester M13 9PL
United Kingdom

p.taylor@maths.manchester.ac.uk