A Note on Computing Involution Centralizers

Ballantyne, John J. and Rowley, Peter J.

2013

ISSN 1749-9097

A Note on Computing Involution Centralizers

John Ballantyne

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

Peter Rowley

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

Abstract

For a black box group G and t an involution of G we describe a computational procedure which produces elements of $C_G(t)$ by making use of the local fusion graph $\mathcal{F}(G, X)$, where X is the G-conjugacy class of t.

Key words: Group, Graph, Centralizer, Involution.

1. Introduction

One lesson of the celebrated paper of Brauer and Fowler (8) is the importance of centralizers of involutions in understanding finite groups of even order. This was reinforced by subsequent work on finite non-abelian simple groups which resulted in their classification. Recently this central role of involution centralizers has begun percolating into the affairs of computational group theory. See, for example, (1), (15), (16), (20) and (22). One reason for this is the method due to Bray (9) for computing centralizers of involutions. Suppose that G is a finite group and X is a G-conjugacy class of involutions. Let $h \in G$, $x \in X$ and k be the order of $[x, h]$. Bray's method involves the elements $\beta_0(x, h)$ and $\beta_1(x, h)$ where $\beta_0(x, h) = [x, h]^{k/2}$ if k is even and $\beta_1(x, h) = [x, h]^{(k+1)/2}h^{-1}$ if k is odd. Then, as is straightforward to check, $\beta_0(x, h), \beta_0(x, h^{-1}) \in C_G(x)$ (if k is even) and $\beta_1(x, h) \in C_G(x)$ (if k is odd), and the method proceeds by choosing random elements h of G. Sometimes we shall write $\beta(x, h)$ to stand for any one of these three types of Bray element. A precursor to Bray's method, due to R. Parker, just employed the elements

Email addresses: ballantyne@maths.manchester.ac.uk (John Ballantyne), peter.j.rowley@manchester.ac.uk (Peter Rowley).

$\beta_0(x, h)$. One early use of the Parker method (also called the “dihedral group method”) was in a computer construction of the sporadic simple group J_4 - see (19) and (21). The elements $\beta_1(x, h)$ turn out to be very important, giving rise as they do to uniformly distributed elements in $C_G(x)$ - an observation due to R. Parker.

The local fusion graph $\mathcal{F}(G, X)$ is the graph whose vertex set is X with $x, y \in X$ joined by an edge whenever $x \neq y$ and the order of xy is odd. Such graphs have been investigated in (4), (5), (6), and (13) studies graphs of a similar nature. Now suppose that $(x_0, x_1, x_2 \ldots, x_n)$ is a path γ in $\mathcal{F}(G, X)$ of length n, where $x = x_0$ and $y = x_n$ (by a path we just mean that x_{i+1} is adjacent to x_i for $i = 0, \ldots, n-1$ - note that we allow the possibility that $x_i = x_{i+j}$, where $1 \leq j \leq n-i$). Let k_i be the order of $x_i x_{i+1} (0 \leq i \leq n-1)$. Then the element $g_i = (x_i x_{i+1})^{(k_i+1)/2}$ conjugates x_i to x_{i+1} and consequently

$$g(x, \gamma) := g_0 g_1 g_2 \ldots g_{n-1}$$

conjugates x to y. Now let $h \in G$ be such that $x^h = y$. Then

$$g(x, \gamma, h) := g_0 g_1 g_2 \cdots g_{n-1} h^{-1}$$

is an element of $C_G(x)$. Consider the special case when $n = 1$ (so $x = x_0$ and $y = x_1$). Then

$$g(x, \gamma, h) = g_0 h^{-1} = (x_0 x_1)^{(k_0+1)/2} h^{-1} = (xy)^{(k_0+1)/2} h^{-1} = (xx^h)^{(k_0+1)/2} h^{-1} = \beta_1(x, h).$$

So paths of length one yield Bray elements. Furthermore, if we have a path γ' in $\mathcal{F}(G, X)$ based at some vertex x' with corresponding element $g(x', \gamma', h') \in C_G(x')$, then if $x' = x^{h_0}$ we have $g(x', \gamma', h')^{h_0^{-1}} \in C_G(x)$.

Thus foraging in $\mathcal{F}(G, X)$ may produce elements of $C_G(x)$, and we refer to such elements as *path elements*. This note seeks to utilise such elements, as well as elements of a related nature. An algorithm by Kundle and Cooperman (17) for centralizers has some parallels with the work here. Also, we observe that, just as for the Bray algorithm, the above procedure works with black box groups. Finally, we would like to thank the referees, whose comments have led to significant improvements to this paper.

2. Path elements of local fusion graphs

One benefit of the method presented here is that, under favourable circumstances, it requires fewer operations to calculate path elements (using predetermined random elements of G) than to calculate Bray elements (using new random elements). For example, suppose we have two random elements $h_1, h_2 \in G$, for which tt^{h_1} and tt^{h_2} have odd orders k_1 and k_2, respectively. The standard application of Bray's method yields potentially two elements of $C_G(t)$, namely $\beta_1(t, h_1)$ and $\beta_1(t, h_2)$ - these are path elements of length one using our terminology. To obtain further elements of $C_G(t)$, one might simply take a further random element $h_3 \in G$ and calculate $\beta(t, h_3)$. This involves the following:

(i) Producing the random element h_3.

(ii) Calculating t^{h_3}.

(iii) Calculating the order k_3 of the product tt^{h_3}.

(iv) If k_3 is even, calculating $(tt^{h_3})^{k_3/2}$ or $(tt^{h_3^{-1}})^{k_3/2}$, while if k_3 is odd, calculating $(tt^{h_3})^{(k_3+1)/2} h_3^{-1}$.

However, another option is to attempt to calculate additional path elements using the original random elements h_1 and h_2. This involves the following:

(i) Calculating the order k of the product $t^{h_1}t^{h_2}$.

(ii) If k is even:

  (a) Calculating $((t^{h_1}t^{h_2})^{k/2})^{h_1^{-1}}$ or $((t^{h_1}t^{h_2^{-1}})^{k/2})^{h_1^{-1}}$.

  If k is odd:

  (a) Calculating $(t^{h_1}t^{h_2})^{(k+1)/2}$.

  (b) Calculating the product $g(t, \gamma, h_2) = (tt^{h_1})^{(k_1+1)/2}(t^{h_1}t^{h_2})^{(k+1)/2}h_2^{-1}$.

Notice that for k odd, in step (ii)(b) only two multiplications are required, since the elements $(tt^{h_1})^{(k_1+1)/2}$ and h_2^{-1} have already been calculated during the construction of $\beta_1(t, h_1)$ and $\beta_1(t, h_2)$.

We see from the descriptions above that (assuming k is odd) ignoring the cost of computing random group elements, a similar number of operations are required to calculate $g(t, \gamma, h_2)$ as to calculate $\beta(t, h_3)$ (the precise cost of calculating element orders, inverses and conjugates will depend on the specific situation one is working in). However, when we take into account the cost of computing random group elements, the path element method begins to see some advantages. If using the "product replacement algorithm" (11) to produce random elements, then, assuming the pre-processing stage has been previously completed, to generate each new random element requires at least one group operation. Moreover, there may be occasions when one does not wish to employ the product replacement algorithm in its original form. One such situation is described in (3), with another discussed later in this section. Variants and alternatives exist (see (2) and (18), for example) - however, to produce random elements (and subsequently Bray elements) using these procedures usually requires more group operations than in the original product replacement algorithm. This translates into further advantages for the path element method, in terms of computational cost.

While it may be relatively quick to compute, it is not immediately obvious that the element $g(t, \gamma, h_2)$ is independent of $\beta_1(t, h_1)$ and $\beta_1(t, h_2)$ (whereas in general $\beta_1(t, h_3)$ will be). However, the experimental evidence in Section 4 shows that, for a number of groups tested, the subgroup $\langle \beta_1(t, h_1), \beta_1(t, h_2), g(t, \gamma, h_2)\rangle$ is not only usually larger than $\langle \beta_1(t, h_1), \beta_1(t, h_2)\rangle$, but compares favourably with $\langle \beta_1(t, h_1), \beta_1(t, h_2), \beta(t, h_3)\rangle$.

Thus far we have only considered subgraphs of $\mathcal{F}(G, X)$ with three vertices. As we consider progressively larger subgraphs $\mathcal{G}$ of $\mathcal{F}(G, X)$, the number of paths in $\mathcal{G}$ may increase rapidly. However, the following lemma shows that a relatively large number of path elements are redundant when it comes to generating subgroups of $C_G(t)$.

Lemma 1. *Suppose $\mathcal{G}$ is a complete subgraph of a local fusion graph $\mathcal{F}(G, X)$, with vertex set*

$$V = \{t_0, t_1, \ldots, t_n\},$$

where $t_0 = t$ and $t_i = t^{h_i}$ for $1 \leq i \leq n$. Let P be the subgroup of $C_G(t)$ which is generated by all path elements of $C_G(t)$ which arise from paths in $\mathcal{G}$. Then P is generated by the elements of $C_G(t)$ which arise from the paths

$$\mathscr{P} = \{(t_0, t_i), (t_0, t_j, t_k) \mid 1 \leq i \leq n, 1 \leq j < k \leq n\}.$$

Proof. We denote by P_0 the subgroup of P which is generated by the path elements which arise from $\mathscr{P}$. Given a product t_it_j of two involutions from V, we write $g_{i,j} =$

$(t_it_j)^{(o(t_it_j)+1)/2}$, where $o(t_it_j)$ denotes the order of t_it_j. We proceed by induction on n. When $n = 1$ we have

$$P = \langle g_{0,1}h_1^{-1}, (g_{1,0}h_1)^{h_1^{-1}}\rangle.$$

Certainly the first generator lies in P_0, but also

$$(g_{1,0}h_1)^{h_1^{-1}} = h_1g_{1,0} = (g_{0,1}h_1^{-1})^{-1},$$

so the second generator lies in P_0. Hence the result holds when $n = 1$.

Now let γ be a path in $\mathcal{G}$. We claim that, without loss of generality, we may assume that γ is based at t_0. Indeed, suppose γ is based at some vertex $t_r = t^{h_r}$, say $\gamma = (t_r, t_{\alpha_1}, t_{\alpha_2}, \cdots, t_{\alpha_\ell})$. Then the corresponding path element of $C_G(t)$ will look like

$$\begin{aligned}(g_{r,\alpha_1}g_{\alpha_1,\alpha_2}\cdots g_{\alpha_{\ell-1},\alpha_\ell}h_{\alpha_\ell}^{-1}h_r)^{h_r^{-1}} &= h_rg_{r,\alpha_1}g_{\alpha_1,\alpha_2}\cdots g_{\alpha_{\ell-1},\alpha_\ell}h_{\alpha_\ell}^{-1}\\ &= (g_{0,\alpha_r}h_r^{-1})^{-1}g_{0,r}g_{r,\alpha_1}g_{\alpha_1,\alpha_2}\cdots g_{\alpha_{\ell-1},\alpha_\ell}h_{\alpha_\ell}^{-1},\end{aligned}$$

and we see that this element is in fact a product of path elements based at t_0.

Next we claim that we may assume, without loss of generality, that γ passes through any vertex of $\mathcal{G}$ at most once (that is, γ contains no cycles). Indeed, suppose first that γ contains exactly one cycle, based at some vertex t_{α_s}, say. Then, assuming the final vertex in γ is $t^{h_{\alpha_v}}$, say, the path element $g(t_0, \gamma, h_{\alpha_v})$ must look like

$$g(t_0, \gamma, h_{\alpha_v}) = wyz,$$

where $w = g_{0,\alpha_1}\cdots g_{\alpha_{s-1},\alpha_s}$, $y = g_{\alpha_s,\alpha_{s+1}}\cdots g_{\alpha_{s+u},\alpha_s}$ and $z = g_{\alpha_s,\alpha_{s+u+1}}\cdots g_{\alpha_{v-1},\alpha_v}h_{\alpha_v}^{-1}$. We may rewrite this as

$$g(t_0, \gamma, h_{\alpha_v}) = wh_{\alpha_s}^{-1}y^{h_{\alpha_s}^{-1}}(zh_{\alpha_s}^{-1})^{h_{\alpha_s}^{-1}}.$$

Clearly $wh_{\alpha_s}^{-1}$ is a path element based at t_0, and we have shown above that $(zh_{\alpha_s}^{-1})^{h_{\alpha_s}^{-1}}$ is a product of path elements based at t_0. Moreover, we have

$$\begin{aligned}y^{h_{\alpha_s}^{-1}} &= h_{\alpha_s}g_{\alpha_s,\alpha_{s+1}}\cdots g_{\alpha_{s+u},\alpha_s}h_{\alpha_s}^{-1}\\ &= (g_{\alpha_s,\alpha_{s+1}}\cdots g_{\alpha_{s+u-1},\alpha_{s+u}}h_{\alpha_{s+u}}^{-1}h_{\alpha_s})^{h_{\alpha_s}^{-1}}((g_{\alpha_s,\alpha_{s+u}}h_{s+u}^{-1}h_{\alpha_s})^{-1})^{h_{\alpha_s}^{-1}},\end{aligned}$$

which is a product of two path elements which are products of path elements based at t_0. Thus $g(t_0, \gamma, h_{\alpha_v})$ is a product of path elements based at t_0. If we now suppose that γ contains m cycles, then rearranging as above shows that we may write $g(t_0, \gamma, h)$ as a product of elements which contain fewer than m cycles, and the claim follows by induction on m.

We may therefore assume that γ is based at t_0 and passes through each vertex at most once. However, if γ does not pass through some vertex, then it is a path in a subgraph with $n-1$ vertices, and the result follows by induction on n. So we may assume that, after relabelling, the final vertex in γ is t^{h_n} and

$$g(t_0, \gamma, h_n) = g_{0,1}g_{1,2}\cdots g_{n-1,n}h_n^{-1}.$$

But now we may write

$$g(t_0, \gamma, h_n) = (g_{0,1}g_{1,2}\cdots g_{n-2,n-1}h_{n-1}^{-1})(g_{n-1,n}h_n^{-1}h_{n-1})^{h_{n-1}^{-1}},$$

and by induction both parts are generated by elements from P_0, implying that $g(t_0, \gamma, h)$ is also generated by such elements. □

Another benefit of using path elements is that a comparatively large number of elements of $C_G(t)$ may be produced using a small number of random elements. For example, suppose we have a complete subgraph $\mathcal{G}$ of $\mathcal{F}(G, X)$ with n vertices, and denote by P the subgroup of $C_G(t)$ generated by the path elements from $\mathcal{G}$. Then Lemma 1 implies that up to $n(n+1)/2$ generators may be required for P, all of which arise as path elements. This is in comparison to the $n-1$ Bray elements at t which may be produced using the same supply of random elements. A use for this crop of centralizer elements occurs in a recent refinement of the algorithm in (23) whose aim is to calculate part of the normalizer of a 2-subgroup of a black box group. Suppose G is a black box group and Y is an elementary abelian 2-subgroup of G (this being the pivotal case for this algorithm). The aim is to find $O^{2'}(N_G(Y))$, which is generated by 2-subgroups R of G with the property that they stabilize some maximal chain

$$Y > Y_n > Y_{n-1} > \cdots > Y_1 > Y_0 = 1$$

of Y (where each $[Y_i : Y_{i-1}] = 2$). For such a maximal chain one first determines $C = C_G(Y_1)$ (using Bray's method). Then, with an adapted form of Bray's method, $C_C(Y_2/Y_1)$ is calculated, and this working is repeated up the chain eventually obtaining a 2-subgroup such as R. Now, at each stage of this process we are dealing with different groups and so, if using the product replacement algorithm as the supplier of random elements, we must continually pay the cost of the pre-processing that the product replacement algorithm requires. This is a cost we can ill afford as, for example if $|Y| = 2^8$, the total number of chains to be checked is $145,851$ (see Table 1 of (23)). For these reasons the original implementation in (23) does not use the product replacement algorithm. However now we have the option, again with a small adaptation, to use the type of elements discussed in this note.

We also mention that the use of path elements may be of benefit when combined with the algorithm in (7), which endeavours to compute the centralizers of strongly real elements. The latter algorithm requires a very large number of input elements from the centralizer of an involution which inverts the given strongly real element.

Before describing an implementation of the path element method, let us briefly consider groups in which the method may show value. We shall see presently that, given a random element $h \in G$, the path element procedure described in Section 3 produces the Bray element(s) $\beta(t, h)$. Thus, this implementation of the path element method will be applicable in all situations where Bray's method can be applied (in particular in finite simple groups, which was one of the the main motivations for the work in (9)). However, it is evident that $\mathcal{F}(G, X)$ having few (or no) paths will result in little benefit. Indeed, it will become apparent from the description of the procedure in Section 3 that if there are no paths of length at least two in the subgraphs of $\mathcal{F}(G, X)$ we consider, then the path element method simply reduces to an application of Bray's method. A particularly problematic situation therefore is when $X \subseteq O_2(G)$ and $C_G(t) \nleq O_2(G)$, since then $\mathcal{F}(G, X)$ will have no edges, and Bray's method is effectively a random search.

Fortunately, there are many groups in which we can expect the number of edges in our subgraph to be relatively high. For example, if G is a group of Lie-type in odd characteristic, lower bounds on the number of edges in the local fusion graphs of G have been determined by Parker and Wilson in Theorems 1 and 2 of (20). These lower bounds are not sharp, but they conjecture that for an exceptional group of Lie-type in odd characteristic the number of edges in $\mathcal{F}(G, X)$ is at least $|X|/8$. For groups of Lie-type G

in even characteristic q, results of Guralnick and Lübeck (14) show that as q increases the proportion of elements of G with even order tends to zero, suggesting that the valency of $\mathcal{F}(G,X)$ may become particularly high.

If one has the character table of G to hand, then calculation of the class structure constants gives the exact valencies of the local fusion graphs of G. In particular, such calculations have been carried out for the sporadic simple groups. We note that in all but four cases the valency of $\mathcal{F}(G,X)$ is at least $|X|/4$. The exceptions are $(G,X) = (He, 2A)$, $(Co_2, 2C)$, $(M_{24}, 2A)$ and $(B, 2A)$, with the latter having the smallest valency of approximately $|X|/5.72$ (here we have used Atlas notation (12) for the sporadic groups and their conjugacy classes).

3. Calculating Path Elements

We now describe a procedure for applying the path element method to produce elements of $C_G(t)$.

Input: The black box group G and an involution t of G.

First set $V = \{t\}$, and $\mathcal{C} = \emptyset$. We then use an iterative procedure, with the k-th step being as follows:

```
(i) Produce a random element $h \in G$, and calculate $x = t^h$.
(ii) Calculate the order $o(tx)$ of the product $tx$.
(iii) If $o(tx)$ is even:
    (a) Calculate $\beta_0(t,h)$ and $\beta_0(t,h^{-1})$, and add them to $\mathcal{C}$. Return to step (i).
    If $o(tx)$ is odd:
    (a) For each element $y \in V$, calculate and store the order $o(xy)$ of the product $xy$. This gives us a subgraph $\mathcal{G}$ of $\mathcal{F}(G,X)$, with vertex set $V \cup \{x\}$.
    (b) Calculate all path elements $g(t,\gamma,h)$, where $\gamma$ is a path of length at most two in $\mathcal{G}$ from $t$ to $x$, and add them to $\mathcal{C}$.
    (c) For every $y = t^{h_0} \in V$ for which $o(xy)$ is even, calculate $\beta_0(x, h^{-1}h_0)^{h^{-1}}$ and $\beta_0(x, h_0^{-1}h)^{h^{-1}}$, and add them to $\mathcal{C}$. Add $x$ to $V$ and return to step (i).
```

Output: A subset $\mathcal{C} \subseteq C_G(t)$.

Note that included in the set $\mathcal{C}$ which is produced by the above procedure are all the Bray elements $\beta_0(t,h)$, $\beta_0(t,h^{-1})$ and $\beta_1(t,h)$. Indeed, if we encounter only even order products then the procedure simply produces the Bray elements at t.

In circumstances where we might anticipate a paucity of edges in $\mathcal{F}(G,X)$ we may employ the following strategy. If the order of $z_i = tx_i$, k, is even but not a 2-power, then we set $w_i = z_i^{\ell}$ where ℓ is the largest 2-power dividing k. Then we replace x_i by tw_i, in which case we now have that tx_i has odd order. Thus we at least guarantee as many edges in $\mathcal{G}$ at t as possible.

4. Performance

In this final section we give some experimental data to demonstrate how the path element method performs in practice. Our first test records data regarding subgroups of $C_G(t)$ generated by path elements which arise from triangles in $\mathcal{F}(G,X)$. This situation may occur often if applying the procedure described in Section 3. In practice, when applying the procedure from Section 3, when we see even order products we include the corresponding Bray elements - however, for this test we wish to observe the contribution made by path elements from $\mathcal{F}(G, X)$. We therefore take random elements until we find some $h_1, h_2 \in G$, for which $\mathcal{G}$, the subgraph of $\mathcal{F}(G, X)$ with vertices t, t^{h_1} and t^{h_2}, is a complete graph. We then construct the following subgroups of $C_G(t)$:

- B, the subgroup generated by the Bray elements $\beta(t, h_1)$ and $\beta(t, h_2)$;
- P, the subgroup generated by the path elements which arise from the paths in $\mathscr{P}$, as described in Lemma 1;
- B^+, the subgroup generated by B along with a third Bray element $\beta(t, h)$, where h is an arbitrary random element of G;

This test has been carried out for a number of finite simple groups, with some results recorded in Table 1. The first column notes the group being tested, while the second column shows the number of fixed points of involutions in the G-conjugacy class under examination. Columns three, four and five record the number of occasions (out of 1000 tests in each case) where $|P| > |B|$, $|P| > |B^+|$ and $|P| < |B^+|$, respectively, while the final three columns record the number of occasions where these subgroups are in fact the whole of $C_G(t)$.

For the classical groups we have tested, we have used their standard permutation representations as given by Magma (10). For the exceptional groups of Lie-type $G_2(4)$, ${}^3D_4(2)$ and ${}^3D_4(3)$ we have used their permutation representations on 416 points, 819 points and 26572 points respectively, as given by the online Atlas (24). The permutation representations used for the sporadic groups J_2, Suz, Co_2 and HS are those on 100 points, 1782 points, 2300 points and 100 points respectively, again as given by (24). Since this test is concerned with generation of $C_G(t)$, we have use relatively small permutation representations in which it is easy to check whether or not the whole of $C_G(t)$ has been generated. In practice, we expect the path element method to be of greater value when using very large representations.

We observe that for each case tested, we have a non-zero value in the column $|P| > |B|$. Thus, in general it is not the case that $P = B$. Furthermore, in the majority of cases tested (with exceptions being seen for $PSL_8(2)$, ${}^3D_4(3)$, Co_2 and HS) we see that the subgroup P is more often than not larger than the subgroup B^+. This comparison is particularly relevant in view of the observations in Section 2, which noted that less computational effort is required to produce generators for P than for B^+. We note that for the majority of cases tested, the subgroup P is more likely to be the whole of $C_G(t)$ than B^+.

In contrast to the situation in many finite simple groups, there are many groups where a comparatively large number of generators may be required for $C_G(t)$. One way to produce further generators using the path element mathod is to simply perform a similar procedure to that carried out above, using triangles in $\mathcal{F}(G, X)$, but multiple times. Another option, however, is to consider path elements which arise from subgraphs of $\mathcal{F}(G, X)$ with larger vertex sets. This second method has the advantage of producing a potentially larger ratio of centralizer elements to random input elements. To indicate that this approach may be of value, in Table 2 we record some results where the path element method is applied in the (imprimitive) wreath product $\mathrm{Sym}(5) \wr \mathrm{Sym}(5)$. Here we have varied both the size of the vertex set of the subgraph used, and the number of times the procedure has been applied, and noted the number of times (from 1000 tests) we generate the whole of $C_G(t)$. We have used the natural permutation representation of Sym(5) ≀ Sym(5) on 25 points, with representatives for the conjugacy classes we have used being

$$t_1 = (1,6)(2,7)(3,8)(4,9)(5,10)(11,16)(12,17)(13,18)(14,19)(15,20)$$

and

$$t_2 = (1,6)(2,7)(3,8)(4,9)(5,10)(11,12)(13,14)(16,17)(18,19).$$

These involutions have been chosed to have relatively few fixed points, since in conjugacy classes where involutions have a large proportion of fixed points there is a comparatively low chance of two random conjugates having odd order product, and thus the path element method shows little benefit.

We note that for both conjguacy classes tested, we are more likely to produce the whole of $C_G(t)$ using a subgraph with 5 vertices and applying the procedure once (which requires 4 random input elements), than when applying the procedure twice when using subgraphs with 3 vertices (which also requires 4 random input elements). This test has been carried out in other wreath products of a similar structure, with similar results being obtained.

References

[1] C. Altseimer, A.V. Borovik: *Probabilistic recognition of orthogonal and symplectic groups, Groups and computation III* (Columbus, OH, 1999), 120, Ohio State Univ. Math. Res. Inst. Publ., 8, de Gruyter, Berlin, 2001.

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

[3] L. Babai, I. Pak: *Strong bias of group generators: an obstacle to the "product replacement algorithm"*, (English summary) Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms (San Francisco, CA, 2000), 627–635, ACM, New York, 2000.

[4] J. Ballantyne: *On Local Fusion Graphs of Finite Coxeter Groups*, J. Group Theory, to appear, MIMS EPrint 2012.32.

[5] J. Ballantyne, N. Greer, P. Rowley: *Local Fusion Graphs for Symmetric Groups*, J. Group Theory 16 (2013), no. 1, 35–49.

[6] J. Ballantyne, P. Rowley: *Connectivity of Local Fusion Graphs for Finite Simple Groups*, submitted, MIMS EPrint 2012.110.

[7] C. Bates, P. Rowley: *Centralizers of real elements in finite groups*, Arch. Math. (Basel) 85 (2005), no. 6, 485-489.

[8] R. Brauer, K.A. Fowler: *On groups of even order*, Ann. of Math. (2) 62 (1955), 565–583.

[9] J. N. Bray: *An improved method for generating the centralizer of an involution*, Arch. Math. 74 (2000), 241–245.

[10] J. J. Cannon, C. Playoust: *An Introduction to Algebraic Programming with* Magma [draft], Springer-Verlag (1997).

[11] F. Celler, C. R. Leedham-Green, S. H. Murray, A. C. Niemeyer, E. A. O'Brien: *Generating random elements of a finite group*, Comm. Algebra 23 (1995), no. 13, 4931–4948.

[12] J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker, R. A. Wilson: *Atlas of Finite Groups*, Clarendon, Oxford (1985).

[13] A. Devillers, M. Giudici: *Involution graphs where the product of two adjacent vertices has order three*, J. Aust. Math. Soc. 85 (2008), no. 3, 305–322.

[14] R. M. Guralnick, F. Lübeck: *On p-singular elements in Chevalley groups in characteristic p*, (English summary) Groups and computation, III (Columbus, OH, 1999), 169–182, Ohio State Univ. Math. Res. Inst. Publ., 8, de Gruyter, Berlin, 2001.

[15] P. E. Holmes, S. A. Linton, E. A. O'Brien, A. J. E. Ryba, R. A. Wilson: *Constructive membership in black-box groups*, J. Group Theory 11 (2008), no. 6, 747–763.

[16] D. F. Holt, B. Eick, E. A. O'Brien: *Handbook of computational group theory, Discrete Mathematics and its Applications* (Boca Raton), Chapman and Hall/CRC, Boca Raton, FL, 2005. xvi+514 pp.

[17] D. Kunkle, G. Cooperman: *Biased tadpoles: a fast algorithm for centralizers in large matrix groups*, ISSAC 2009 Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation, 223–230, ACM, New York, 2009.

[18] C. R. Leedham-Green, S. H. Murray: *Variants of product replacement*, (English summary) Computational and statistical group theory (Las Vegas, NV/Hoboken, NJ, 2001), 97–104, Contemp. Math., 298, Amer. Math. Soc., Providence, RI, 2002.

[19] S. Norton: *The construction of J_4*, The Santa Cruz Conference on Finite Groups (Univ. California, Santa Cruz, Calif., 1979), pp. 271–277, Proc. Sympos. Pure Math., 37, Amer. Math. Soc., Providence, R.I., 1980.

[20] C. W. Parker, R. A. Wilson: *Recognising simplicity of black-box groups by constructing involutions and their centralisers*, J. Algebra 324 (2010), no. 5, 885–915.

[21] R. Parker: *Notes for lecture* 11,
`http://www.dpmms.cam.ac.uk/study/IV/2010-11/SPOR/spor11.odp`.

[22] C. E. Praeger, A. Seress: *Probabilistic generation of finite classical groups in odd characteristic by involutions*, J. Group Theory 14 (2011), no. 4, 521–545.

[23] P. Rowley, P. Taylor: *Normalizers of* 2*-subgroups in black-box groups*, LMS J. Comput. Math. 13 (2010), 307–319.

[24] R. Wilson, P. Walsh, J. Tripp, I. Suleiman, R. Parker, S. Norton, S. Nickerson, S. Linton, J. Bray, R. Abbott: `http://brauer.maths.qmul.ac.uk/Atlas/v3/`.

Table 1. Comparisons of Path Element Method and Bray's Method

Group	$\lvert\mathrm{Fix}(t)\rvert$	$\lvert P\rvert > \lvert B\rvert$	$\lvert P\rvert > \lvert B^+\rvert$	$\lvert P\rvert < \lvert B^+\rvert$	$B = C_G(t)$	$P = C_G(t)$	$B^+ = C_G(t)$
$PSL_4(16)$	273	175	48	34	778	945	932
$PSL_6(4)$	341	27	9	2	971	996	990
$PSL_6(4)$	85	541	318	52	399	862	630
$PSL_6(4)$	21	615	343	52	303	868	588
$PSL_8(2)$	127	34	10	20	944	970	981
$PSL_8(2)$	63	462	342	71	359	699	498
$PSL_8(2)$	31	539	358	62	319	694	456
$PSL_8(2)$	15	341	126	61	568	870	809
$PSL_8(3)$	368	463	207	145	332	704	685
$PSL_8(3)$	80	351	190	151	346	637	611
$PSL_8(3)$	0	364	219	110	397	670	570
$PSp_8(3)$	368	253	187	26	664	899	748
$PSp_8(3)$	80	142	89	59	730	852	822
$PSp_8(3)$	0	391	227	132	355	647	561
$P\Omega_8^-(4)$	341	349	176	141	377	679	653
$P\Omega_8^-(4)$	277	363	149	44	589	919	822
$G_2(4)$	32	756	351	113	175	790	552
${}^3D_4(2)$	19	151	70	7	839	987	925
${}^3D_4(3)$	116	463	118	190	350	721	795
J_2	20	344	144	62	582	907	820
J_2	0	663	314	113	251	809	592
Suz	54	98	8	2	899	997	991
Suz	42	448	225	115	365	647	604
Co_2	284	89	2	6	908	994	998
Co_2	140	276	38	32	706	962	957
Co_2	52	464	168	158	327	676	661
HS	20	459	91	210	404	719	834
HS	0	423	182	174	318	627	611

Table 2. Variations on the Path Element Method using various sizes of subgraph $\mathcal{G}$

Class rep	No of vertices	No of tests	No of input elements	$P = C_G(t)$
t_1	3	1	2	0
t_1	3	2	4	556
t_1	3	3	6	881
t_1	4	1	3	388
t_1	4	2	6	928
t_1	5	1	4	740
t_1	5	2	8	984
t_2	3	1	2	0
t_2	3	2	4	266
t_2	3	3	6	779
t_2	4	1	3	89
t_2	4	2	6	873
t_2	5	1	4	518
t_2	5	2	8	971