Computing with elation groups

George Havas *C.R. Leedham-Green*
E.A. O'Brien *Michael C. Slattery*

Abstract. We have proved that the elation groups of a certain infinite family of Roman generalized quadrangles are not isomorphic to those of associated flock generalized quadrangles. The proof is theoretical, but is based upon detailed computations. Here we elaborate on the explicit computer calculations which inspired the proof.

1. Introduction

A finite generalized quadrangle with parameters (s, t) is a point-line incidence structure satisfying a symmetric point-line incidence relation. Comprehensive definitions and results about generalized quadrangles, including information about their elation groups, are given in the monograph [8] and some newer results appear in the lecture notes [7].

Payne [6] introduced a new infinite family of generalized quadrangles which he called *Roman*. The bases for his construction were flock generalized quadrangles having parameters (q^2, q), where q is a prime power. He showed geometrically that the Roman generalized quadrangles are distinct from the flock quadrangles for $q = 3^k$ where $k \geq 3$. In his talk at the "Finite Geometries, Groups and Computation" conference [2], Payne asked for a proof that their elation groups are not isomorphic. He also discusses this question in his lecture notes [7, Chapter 7].

In [4] we prove that the elation groups of the Roman generalized quadrangles are distinct from the elation groups of the associated flock quadrangles for $q = 3^k$ where $k \geq 3$. Their nonisomorphism has an important impact on a major problem in geometry: namely which groups admit a 4-gonal family? The answer to this question will contribute to the characterization of the underlying groups of generalized quadrangles. We refer the interested reader to [7] for further details.

Our proof is theoretical, but it was inspired by the insight gained from detailed group computations for the smallest of these groups conducted using the computer algebra systems GAP [3] and MAGMA [1]. In the appendices, we present some code we used in solving the problem, illustrating interesting aspects of our implementation.

In Section 2 we define the elation groups; in Sections 3 and 4 we report on our computer investigations of the smallest interesting case and show that the corresponding elation groups are nonisomorphic. From this observation, we proved the general theoretical result.

This successful investigation raises various questions of direct relevance to isomorphism testing for p-groups. In particular, it may be that the centraliser computations which distinguished the two groups, and related observations made in [4] on the automorphism group, have general applications to a larger class of p-groups.

2. The elation groups

Payne [6, 7] lists multiplication rules for elation groups of flock generalized quadrangles and Roman quadrangles with parameters (q^2, q). We follow his description.

Let $F = \mathrm{GF}(q),\ q = p^k,\ p$ an odd prime. Let $f : F^2 \times F^2 \to F$ be a symmetric, biadditive map. Further, suppose that if $(0,0) \neq \alpha \in F^2$, then $\{\beta \in F^2 : f(\alpha,\beta) = 0\}$ is an additive subgroup of F^2 of order q. For a fixed nonzero $\alpha \in F^2$, this implies that $|\{f(\alpha,\beta) : \beta \in F^2\}| = q$ also. Such an f is called a *nonsingular pairing.*

Let $G = \{(\alpha,\beta,c) : \alpha = (a_1,a_2) \in F^2, \beta = (b_1,b_2) \in F^2, c \in F\}$. Clearly G has q^5 elements. We now impose a group structure on the set G using a nonsingular pairing.

Let $f : F^2 \times F^2 \to F$ be a given nonsingular pairing. Define a binary operation $\otimes$ on G by

$$(\alpha,\beta,c) \otimes (\alpha',\beta',c') \ = \ (\alpha+\alpha', \beta+\beta', c+c'+f(\beta,\alpha')). \tag{1}$$

Now $(G,\otimes)$ is a group that we denote by G_f.

Let $\overline{f} : F^2 \times F^2 \to F$ be a second (not necessarily distinct) nonsingular pairing, so we also have a group $G_{\overline{f}}$ of order q^5.

The elation group of the flock generalized quadrangle with parameters (q^2, q) is G_f where

$$f(\alpha,\beta) = \alpha \cdot \beta^T.$$

Now we specialize to $q = 3^k \geq 27$ and let n be a fixed nonsquare of F. We define $\overline{f}(\alpha,\beta)$ by

$$\overline{f}(\alpha,\beta) = \alpha \begin{pmatrix} -1 & 0 \\ 0 & n \end{pmatrix} \beta^T + \left\{\alpha \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \beta^T\right\}^{\frac{1}{3}} + \left\{\alpha \begin{pmatrix} 0 & 0 \\ 0 & n^{-1} \end{pmatrix} \beta^T\right\}^{\frac{1}{9}}.$$

Then $G_{\overline{f}}$ is the elation group of the Roman quadrangle with parameters (q^2, q). We seek to distinguish G_f from $G_{\overline{f}}$.

3. Computing with the two groups

To gain insight into the structure of these groups, we considered the smallest instance where the corresponding quadrangles are different, namely for $q = 27$. Hence we must distinguish between two groups, G_f and $G_{\overline{f}}$, each of order 3^{15}.

The first step is to convert Payne's description into suitable representations of the group for machine computation. This is perhaps the easiest place where errors could be made, so we did it in two different ways, giving relatively independent confirmation of our results. Given the size and nature of these groups we chose to compute with both permutation representations and power-commutator presentations for each of the groups.

The facility offered by `GAP` to realize concretely a group using only its most basic definition – namely, a group operation defined on an abstract set – was particularly useful. Using Equation (1) to define the group operation, we constructed the two 3-groups. It was then easy to identify core-free subgroups of order 3^6, and hence obtain faithful permutation representations of degree 3^9.

This kind of application is somewhat novel, so we give code in Appendix A which constructs both groups. Perhaps most instructive is the code used to provide the syntactic and semantic foundation needed for working with groups defined this way. Observe that the code which defines the particular groups is simply a translation of the mathematical definitions. By calling `PayneGroup(forig)` we construct G_f and by `PayneGroup(fbar)` we construct $G_{\overline{f}}$, after which we can study the groups using the full power of `GAP`.

We also wrote down explicit power-commutator presentations [9, Chapter 11] for the two groups. This is quite straightforward from Payne's description. MAGMA functions which enable us to do this are given in Appendix B. The relations with nontrivial right hand sides in a power-commutator presentation for G_f on generators $g_1, g_2, \ldots, g_{15}$ are:

$$
\begin{array}{lll}
[g_7, g_1] = g_{13}, & [g_7, g_2] = g_{14}, & [g_7, g_3] = g_{15}, \\
[g_8, g_1] = g_{14}, & [g_8, g_2] = g_{15}, & [g_8, g_3] = g_{13}^2 g_{14}, \\
[g_9, g_1] = g_{15}, & [g_9, g_2] = g_{13}^2 g_{14}, & [g_9, g_3] = g_{14}^2 g_{15}, \\
[g_{10}, g_4] = g_{13}, & [g_{10}, g_5] = g_{14}, & [g_{10}, g_6] = g_{15}, \\
[g_{11}, g_4] = g_{14}, & [g_{11}, g_5] = g_{15}, & [g_{11}, g_6] = g_{13}^2 g_{14}, \\
[g_{12}, g_4] = g_{15}, & [g_{12}, g_5] = g_{13}^2 g_{14}, & [g_{12}, g_6] = g_{14}^2 g_{15}.
\end{array}
$$

The relations with nontrivial right hand sides for $G_{\overline{f}}$ are:

$$
\begin{array}{lll}
[g_7, g_1] = g_{13}^2, & [g_7, g_2] = g_{14}^2, & [g_7, g_3] = g_{15}^2, \\
[g_7, g_4] = g_{13}, & [g_7, g_5] = g_{13} g_{14}, & [g_7, g_6] = g_{13} g_{14}^2 g_{15}, \\
[g_8, g_1] = g_{14}^2, & [g_8, g_2] = g_{15}^2, & [g_8, g_3] = g_{13} g_{14}^2, \\
[g_8, g_4] = g_{13} g_{14}, & [g_8, g_5] = g_{13} g_{14}^2 g_{15}, & [g_8, g_6] = g_{14}, \\
[g_9, g_1] = g_{15}^2, & [g_9, g_2] = g_{13} g_{14}^2, & [g_9, g_3] = g_{14} g_{15}^2,
\end{array}
$$

$$
\begin{array}{lll}
[g_9, g_4] = g_{13}g_{14}^2g_{15}, & [g_9, g_5] = g_{14}, & [g_9, g_6] = g_{14}g_{15}, \\
[g_{10}, g_1] = g_{13}, & [g_{10}, g_2] = g_{13}g_{14}, & [g_{10}, g_3] = g_{13}g_{14}^2g_{15}, \\
[g_{10}, g_4] = g_{15}^2, & [g_{10}, g_5] = g_{13}g_{15}, & [g_{10}, g_6] = g_{13}g_{14}^2, \\
[g_{11}, g_1] = g_{13}g_{14}, & [g_{11}, g_2] = g_{13}g_{14}^2g_{15}, & [g_{11}, g_3] = g_{14}, \\
[g_{11}, g_4] = g_{13}g_{15}, & [g_{11}, g_5] = g_{13}g_{14}^2, & [g_{11}, g_6] = g_{13}g_{15}^2, \\
[g_{12}, g_1] = g_{13}g_{14}^2g_{15}, & [g_{12}, g_2] = g_{14}, & [g_{12}, g_3] = g_{14}g_{15}, \\
[g_{12}, g_4] = g_{13}g_{14}^2, & [g_{12}, g_5] = g_{13}g_{15}^2, & [g_{12}, g_6] = g_{14}^2g_{15}^2.
\end{array}
$$

The presentations look different, but do they define nonisomorphic groups?

4. Distinguishing the two groups

Given that we have concrete representations for the two groups, all that remains is to distinguish them. The obvious approach is to try a conventional method for deciding isomorphism of p-groups.

O'Brien [5] describes an algorithm which decides isomorphism among finite p-groups. He defines a *standard presentation* for a p-group and provides an algorithm which allows its construction from an arbitrary power-commutator presentation. We sought to decide directly using this algorithm whether the two 3-groups were isomorphic. However this approach was unsuccessful and it is instructive to explain why.

The algorithm proceeds by induction down the lower exponent-p central series of a given p-group P; that is, it successively computes the standard presentation for the quotients $P_i = P/\mathcal{P}_i(P)$, where $(\mathcal{P}_i(P))$ is the descending sequence of subgroups defined recursively by $\mathcal{P}_1(P) = P$ and $\mathcal{P}_{i+1}(P) = [\mathcal{P}_i(P), P]\mathcal{P}_i(P)^p$ for $i \geq 1$.

In each case P is a group of order 3^{15} having 3-class 2 and its Frattini factor $P_1 = P/\Phi(P)$ has order 3^{12}. Let $G = F/R$; recall that $G^\star = F/[R, F]R^p$ is the *p-covering group* of G. The class one 3-quotient P_1 of P has order 3^{12} and its 3-covering group $P_1^\star$ has order 3^{78}. The automorphism group of P_1 is $\mathrm{GL}(12, 3)$. The class 2 quotient of P, namely P itself, is now obtained by factoring a subgroup of order 3^{63} from $P_1^\star$. Its standard presentation is determined by the leading element of the orbit of this subgroup under the action of $Aut(P_1) = \mathrm{GL}(12, 3)$. Equivalently, if the two groups of order 3^{15} are isomorphic, then the two subgroups of order 3^{63} are in the same orbit under the action of $Aut(P_1)$. Given its size, it was not feasible to construct this orbit and so we could not directly use the algorithm to decide isomorphism.

Since this direct approach failed, we searched for characteristic subgroups and other invariants which might allow us to either deduce nonisomorphism or reduce the problem to a subgroup of $\mathrm{GL}(12, 3)$ preserving some characteristic structure.

The significantly different power-commutator presentations for the two 3-groups suggested that centralizers of elements might be useful invariants and led us to count both conjugacy classes and numbers of distinct centralizers. While this proved fruitless – the data for both groups being identical – it led us to consider the structure of the centralizers in more detail. A routine investigation showed that all of the noncentral elements of G_f have centralizers whose centers have order 3^6 while the noncentral elements of $G_{\overline{f}}$ have centralizers whose centers have order 3^4.

While the primary computations were performed using the power-commutator presentations, the permutation representations of degree 3^9 were also important, permitting independent confirmation of these observations.

Our proof that a natural generalization of this property suffices to distinguish between the groups G_f and $G_{\overline{f}}$ for arbitrary finite fields F of characteristic 3 appears in [4].

Acknowledgements

We are grateful to Alexander Hulpke who instructed us in the design of the GAP programs to investigate these groups. The first author was partially supported by the Australian Research Council and the third by the Marsden Fund of New Zealand via grant UOA 124.

A. GAP code for elation group computation

```
DeclareCategory("IsMyObj",IsMultiplicativeElementWithInverse);
DeclareCategoryCollections("IsMyObj");
InstallTrueMethod(IsGeneratorsOfMagmaWithInverses,IsMyObjCollection);
DeclareRepresentation("IsMyRep",IsMyObj and IsPositionalObjectRep,[]);

MyObj:=function(type,list)
  list:=ShallowCopy(list);
  return Objectify(type,list);
end;

InstallMethod(PrintObj,"my objects",true,[IsMyRep],0,
function(a)
  Print("MyObj([",a![1],",",a![2],",",a![3],",",a![4],",",a![5],"])");
end);

forig := function(a1,a2,b1,b2)
  return a1*b1+a2*b2;
end;
```

```
n:=PrimitiveElement(GF(27));
fbar := function(a1,a2,b1,b2)
local t1,t2,t3,r;
  t1 := -a1*b1+n*a2*b2; t2 := a2*b1+a1*b2; t3 := (n^-1)*a2*b2;
  r := t1 + t2^9 + t3^3;
  return r;
end;

PayneGroup:=function(f)
local gf, p, q, r, z, MyFamily, MyType, gens;
  gf:=GF(27);
  p:=PrimitiveElement(gf);
  q:=p^2;
  r:=One(gf);
  z:=Zero(gf);
  MyFamily:=NewFamily("PayneFamily",IsMyRep,IsMyRep);
  MyFamily!.field:=gf;
  MyType:=NewType(MyFamily,IsMyRep);
  MyFamily!.func:=f;
  MyFamily!.defaultType:=MyType;
  gens:=List([[p,z,z,z,z],[z,p,z,z,z],[z,z,p,z,z],[z,z,z,p,z],[z,z,z,z,p],
              [q,z,z,z,z],[z,q,z,z,z],[z,z,q,z,z],[z,z,z,q,z],[z,z,z,z,q],
              [r,z,z,z,z],[z,r,z,z,z],[z,z,r,z,z],[z,z,z,r,z],[z,z,z,z,r]],
             i->Objectify(MyType,i));
  return Group(gens);
end;

InstallMethod(OneOp,"payne objects",true,[IsMyRep],0,
function(x)
local fam,z;
  fam:=FamilyObj(x); z:=Zero(fam!.field);
  return Objectify(fam!.defaultType,[z,z,z,z,z]);
end);

InstallMethod(\*,"payne objects",IsIdenticalObj,[IsMyRep,IsMyRep],0,
function(x,y)
local fam, f;
  fam:=FamilyObj(x); f:=fam!.func;
  return Objectify(fam!.defaultType,
  [x![1]+y![1],x![2]+y![2],x![3]+y![3],x![4]+y![4],
    x![5]+y![5]+f(x![3],x![4],y![1],y![2])]);
end);

InstallMethod(\=,"payne objects",IsIdenticalObj,[IsMyRep,IsMyRep],0,
function(x,y)
  return x![1]=y![1] and x![2]=y![2] and x![3]=y![3] and
         x![4]=y![4] and x![5]=y![5];
end);
```

```
InstallMethod(\<,"payne objects",IsIdenticalObj,[IsMyRep,IsMyRep],0,
function(x,y)
  return [x![1],x![2],x![3],x![4],x![5]]<[y![1],y![2],y![3],y![4],y![5]];
end);

InstallMethod(InverseOp,"payne objects",true,[IsMyRep],0,
function(x)
local one, a, la;
  one:=One(x); a:=x; la:=a;
  while a<>one do
    la:=a; a:=a*x;
  od;
  return la;
end);
```

B. Power-commutator presentations in Magma

```
function coord(n, i)
  F27 := GF(27); zero := F27!0;
  Felts := [F27![1,0,0], F27![0,1,0], F27![0,0,1]];
  islot := ((i-1) div 3) + 1; ifld := ((i-1) mod 3) + 1;
  if islot eq n then r := Felts[ifld]; else r := zero; end if;
  return r;
end function;

function PayneGroup(f)
  Fpre := FreeGroup(15); Rels := []; Z := Integers();
  for i in [1..15] do
    Append(~Rels,(Fpre.i)^3=Id(Fpre));
  end for;
  for i in [1..14] do
    a1 := coord(1,i); a2 := coord(2,i); b1 := coord(3,i); b2 := coord(4,i);
    for j in [i+1..15] do
      val := f(coord(3,j),coord(4,j),a1,a2)-f(b1,b2,coord(1,j),coord(2,j));
      vseq := Eltseq(val);
      Append(~Rels, (Fpre.j,Fpre.i) = Fpre.13^(Z!vseq[1])
                                *Fpre.14^(Z!vseq[2])*Fpre.15^(Z!vseq[3]));
    end for;
  end for;
  G := quo<GrpPC: Fpre|Rels>;
  return G;
end function;

function forig(x,y,u,v)
  return x*u+y*v;
end function;
```

```
function fbar(a1,a2,b1,b2)
  n := GF(27).1; t1 := -a1*b1+n*a2*b2; t2 := a2*b1+a1*b2; t3 := n^-1*a2*b2;
  return t1 + t2^9 + t3^3;
end function;
```

References

[1] W. Bosma, J. Cannon and C. Playoust, The Magma algebra system I: the user language, *J. Symbolic Comput.* **24** (1997), 235–265.
See also http://magma.maths.usyd.edu.au/magma/

[2] Finite Geometries, Groups and Computation Conference, Pingree Park, Colorado State University, 4–9 September 2004; http://www.math.colostate.edu/~hulpke/pingree/

[3] The GAP Group, *GAP – Groups, Algorithms, and Programming, Version* 4.4, 2004; http://www.gap-system.org/

[4] George Havas, C.R. Leedham-Green, E.A. O'Brien and Michael C. Slattery, Certain Roman and flock generalized quadrangles have nonisomorphic elation groups, *Advances in Geometry* (to appear).

[5] E.A. O'Brien, Isomorphism testing for p-groups, *J. Symbolic Comput.* **17** (2004), 133–147.

[6] S.E. Payne, An essay on skew translation generalized quadrangles, *Geom. Dedicata* **32** (1989), 93–118.

[7] S.E. Payne, *Ten Lectures in Perth (Elation Generalized Quadrangles, Etc.)*, Research Report 2004/11, School of Mathematics and Statistics, The University of Western Australia, 2004; http://www.maths.uwa.edu.au/research/reports/2004/rr11.pdf

[8] S.E. Payne and J.A. Thas, *Finite Generalized Quadrangles.* Pitman, 1984.

[9] C.C. Sims, *Computation with finitely presented groups.* Cambridge University Press, 1994.

George Havas, ARC Centre for Complex Systems, School of Information Technology and Electrical Engineering, The University of Queensland, Queensland 4072, AUSTRALIA

Email: havas@itee.uq.edu.au

C.R. Leedham-Green, School of Mathematical Sciences, Queen Mary, University of London, E1 4NS, UNITED KINGDOM

Email: c.r.leedham-green@qmul.ac.uk

E.A. O'Brien, Department of Mathematics, University of Auckland, Private Bag 92019, Auckland, NEW ZEALAND

Email: e.obrien@auckland.ac.nz

Michael C. Slattery, Department of Mathematics, Statistics and Computer Science, Marquette University, P.O. Box 1881, Milwaukee WI 53201-1881, USA

Email: mikes@mscs.mu.edu