SOME NEW 5-DESIGNS

R. H. F. DENNISTON

A *t-design* is a collection of k-subsets (*blocks*) of a fixed v-set, with the property that any t-subset is contained in just one block. (The usual definition would say: " any t of the v elements are included in just λ blocks ", but we are not concerned here with values of λ greater than 1.) The present paper specifies a 5-design for which $k = 7$ and $v = 28$, and some for which $k = 6$ and v is 24, 48, or 84.

A well-known method for the construction of t-designs is to set up a one-dimensional projective geometry, consisting of the marks of a Galois field GF(q) together with a point ∞. Various useful groups of permutations act on this $(q+1)$-set: let us choose the group, L say, of projectivities specified by

$$\left\{x \to \frac{ax+b}{cx+d} \;\middle|\; ad-bc \text{ is a square and not zero}\right\}.$$

(Some authors use the letters PSL, and others LF, to designate this group.) The k-subsets of our $(q+1)$-set arrange themselves into transitivity classes under L; and it may happen that a suitably chosen transitivity class turns out to be a t-design.

This method effectively goes back to Mathieu [1; p. 263]. Each of the known 5-designs ($k = 6$, $v = 12$; $k = 8$, $v = 24$) is in fact a single transitivity class under the appropriate L [2; p. 264], and the same applies to certain 3-designs. But why should we not construct a t-design by putting together two (or more) suitable transitivity classes? Such a result might be of equal interest from the combinatorial, though of less from the algebraic, point of view (and it is well known that cyclic groups can be used in this way).

Let us, accordingly, set up GF(3^3), adjoining to GF(3) an element i such that $i^3 = i+2$, and put in the point ∞; and let us consider a 7-subset of the 28-set which is invariant under seven projectivities. For instance,

$$\{\infty,\ 0,\ 1,\ 2i^2+2i,\ 2i^2,\ i^2+2i+2,\ i^2+2i\}$$

is fixed under the subgroup generated by

$$x \to -\frac{i^2+2i}{x-(i^2+2i)}$$

—but in fact all such 7-subsets are in one transitivity class. Let D be the union of this class and the class of the 7-subset

$$\{\infty,\ 0,\ 1,\ 2,\ i,\ i+1,\ i+2\},$$

this latter subset being fixed by the obvious subgroup of three projectivities. Then the cardinality of D is $(\frac{1}{7}+\frac{1}{3})$ times the order of the group L: and this happens to be the right number of blocks for a 5-design.

It is tedious to verify that D is in fact a 5-design, but not impracticable. The

Received 11 September, 1974; revised 4 April, 1975.

[BULL. LONDON MATH. SOC., 8 (1976), 263–267]

group being transitive on unordered 3-subsets, we need only look at the 50 blocks of D that include ∞, 0, 1, and verify that, of the other 25 points, any pair is contained in just one of those blocks. I can send to anyone who is interested a list of the 50 blocks, with group elements that transform the basic blocks into them, and a table that can be used to locate any pair of points.

This question of making it feasible for a reader to check the results, as the referee has emphasized, is the most awkward one that arises over their publication. If v is to go up to 84, even distributing lists will not be very easy. So I have written an ALGOL procedure, which any reader with access to a computer can apply to a short list of numbers; this ought to convince him that 5-designs have been found.

```
procedure design (p, ndata, data, sixth, wrong); value p, ndata;
    integer p, ndata; integer array data, sixth; label wrong;
comment We want to verify the existence of a 5-design on the points of the one-
    dimensional projective geometry of prime order p. We are given, in the array
    data [1:ndata, 1:3], a certain number ndata of triads. Each of these, together
    with the points infinity, 0, 1, forms a block of six points. This is transformed under
    the PSL group into other blocks that include infinity, 0, 1. We expect to find that
    any two variable points x, y will be found just once in a block with the three
    fixed points, the remaining point being denoted by sixth[x, y] and tabulated in
    the array sixth [2:p-1, 2:p-1]. If this expectation is not satisfied, the procedure
    exits to the label wrong;
begin
integer procedure rem (x); value x; integer x;
    rem := x-(entier(x/p))×p;
    comment This reduces x modulo p;
Boolean array square [1:p-1];
integer array recip [1:p-1], negrec [-1:p-1], six [1:6],
    five [1:5], four [1:4], three [1:3];
comment We need to know, for each non-zero element x of the field, whether x is a
    square, and what are 1/x and -1/x;
integer i, ij, j, jk, k, kl, l, n, x, y, z;
for x := 1 step 1 until p-1 do square [x] := false;
for x := 1 step 1 until p/2 do square [rem (x×x)] := true;
negrec [-1] := 0; negrec [0] := -1;
comment We use the integer -1 to represent the point infinity;
for x := 1 step 1 until p-1 do
for y := x step 1 until p-1 do
if rem (x×y) = 1 then
begin recip [x] := y; negrec [x] := p-y;
      recip [y] := x; negrec [y] := p-x
end;
comment If we begin by clearing the array in which we are going to enter sixth[x, y],
    we can afterwards tell whether an entry has already been made at a given place
    in it;
for x := 2 step 1 until p-1 do
for y := 2 step 1 until p-1 do sixth[x, y] := 0;
comment Each triad in turn from the array data goes with the fixed points into the
    array six;
for n := 1 step 1 until ndata do
```

```
begin
  six [1] := −1; six [2] := 0; six [3] := 1; six [4] := data [n, 1];
  six [5] := data [n, 2]; six [6] := data [n, 3];
  comment By subtraction and the operation that takes x to −1/x, the six points
    are so transformed that each in turn goes to infinity, the others being listed
    in the array five;
  for i := 1 step 1 until 6 do
  begin
    if i = 1 then
    begin five [1] := six [2]; five [2] := six [3];
      five [3] := six [4]; five [4] := six [5];
      five [5] := six [6]
    end
    else for j := 1 step 1 until 5 do
    begin ij := if j < i then j else j+1;
      x := if six[ij] = −1 then −1 else rem ( six[ij]−six[i]+p );
      five [j] := negrec [x]
    end;
    comment By subtraction, the five points are so transformed that each in turn
      goes to zero, the others being listed in the array four;
    for j := 1 step 1 until 5 do
    begin
      for k := 1 step 1 until 4 do
      begin jk := if k < j then k else k+1;
        four [k] := rem (five[jk]−five[j]+p )
      end;
      comment Of the four points, we consider those which are squares, and
        send each in turn of these to unity by division, the others being listed
        in the array three;
      for k := 1 step 1 until 4 do if square[four [k]] then
      begin
        for l := 1 step 1 until 3 do
        begin kl := if l < k then l else l+1;
          three [l] := rem (four[kl] × recip[four[k]] )
        end;
        x := three [1]; y := three [2]; z := three [3];
        comment If we have found exactly the same triad before, the inference
          is that some projectivity in the group permutes the six points and
          sends six[i] to some earlier one. So there is no need for six[i] to be
          projected to infinity;
        if sixth [x, y] = z then go to next i;
        comment If some earlier triad has just two points in common with this
          one, it is not true that a 5-design can be built up from the data;
        if sixth [y, z] ≠ 0 then go to wrong;
        if sixth [z, x] ≠ 0 then go to wrong;
        if sixth [x, y] ≠ 0 then go to wrong;
        sixth [y, z] := sixth [z, y] := x;
        sixth [z, x] := sixth [x, z] := y;
        sixth [x, y] := sixth [y, x] := z
```

```
            end
        end;
    next i: end
end;
comment The array sixth ought now to have non-zero entries in all the non-diagonal
    positions;
for x := 2 step 1 until p-2 do
for y := x+1 step 1 until p-1 do
    if sixth [x, y] = 0 then go to wrong
end of procedure design;
```

Here are some sets of data which may be input to this procedure—and, if there is no exit to the label *wrong*, the existence of various 5-designs ($k = 6$) will in this way be verified:

$p = 23$ *ndata* = 3

3	7	10	2	3	12	2	5	14

$p = 23$ *ndata* = 3

2	3	13	2	5	18	3	4	16

$p = 47$ *ndata* = 8

2	3	7	2	5	31	2	12	43	2	10	34
2	14	19	3	4	28	3	13	31	3	17	45

$p = 47$ *ndata* = 14

2	3	25	2	5	10	2	6	42	2	8	14
2	12	18	2	13	26	3	4	28	3	11	43
3	12	26	3	13	42	3	19	22	4	13	15
4	20	33	4	21	35						

$p = 83$ *ndata* = 43

11	16	58	19	23	49	20	30	48	8	57	59
13	38	79	20	25	34	26	50	53	27	39	57
28	55	82	31	34	58	39	52	63	55	60	77
2	11	22	3	19	57	6	17	19	6	39	48
7	15	22	7	20	74	7	32	58	9	19	39
12	39	53	13	19	27	13	31	71	14	19	75
14	22	49	15	29	32	15	53	81	19	22	65
19	56	70	19	58	59	20	23	55	20	27	53
20	78	79	22	60	64	25	43	79	31	72	74
36	55	73	37	55	80	37	74	82	40	55	74
42	75	79	55	57	78	57	64	80			

There is no isomorphism between the two 5-designs for which $v = 24$. It turns out, in fact, that $\{\infty, 0, 1, 3, 7, 10\}$ is fixed by a non-Abelian subgroup of order 6, and $\{\infty, 0, 1, 2, 3, 13\}$ by a cyclic subgroup of the same order; which goes to show

that an isomorphism is very unlikely. We can make sure by calculating invariants of the respective 2-designs we get by omitting ∞, 0, 1 from the blocks to which they all belong. Such a Steiner triple system is just the set of triads $\{x, y, sixth[x, y]\}$, where the array *sixth* has been constructed by the procedure *design*. We may, for instance, define a " quadrilateral " as a set of four triads of a Steiner triple system, all contained in one set of six points. Then the two 2-designs of order 21 have 15 and 24 quadrilaterals respectively. I find that any 5-design for which $k = 6$ and $v = 24$, and which is invariant under the PSL group, will be isomorphic to one or the other of the 5-designs specified.

For the case where $v = 48$, I stopped searching by hand when I got to a hundred solutions of the problem, two extreme cases being specified above. I then used a computer to find the number of " quadrilaterals " to which any point of each Steiner triple system belongs, and so to establish that no two of the 100 5-designs are isomorphic. The 5-design for which $v = 84$ was likewise found by a hand search—I should like, indeed, to make it clear that all the results announced here were found without recourse to computers, even though I suggest that computers may be used to verify them.

References

1. E. Mathieu, " Mémoire sur l'étude des fonctions de plusieurs quantités, sur la manière de les former et sur les substitutions qui les laissent invariables ", *Journal de Math.*, (2) 6 (1861), 241–323.
2. E. Witt, " Die 5-fach transitiven Gruppen von Mathieu ", *Hamburgische Abh.*, 12 (1938), 256–264.

Department of Mathematics,
University of Leicester,
LE1 7RH.