Some applications of Magma in designs and codes: oval designs, hermitian unitals and generalized Reed-Muller codes

J. D. Key*
Department of Mathematical Sciences
Clemson University, Clemson SC 29634

October 19, 1998

Abstract

We describe three applications of Magma to problems in the area of designs and the associated codes:

- Steiner systems, Hadamard designs and symmetric designs arising from a oval in an even order plane, leading in the classical case to bent functions and difference-set designs;
- the hermitian unital as a 2-$(q^3+1, q+1, 1)$ design, and the code over F_p where p divides $q+1$;
- a basis of minimum-weight vectors for the code over F_p of the design of points and hyperplanes of the affine geometry $AG_d(F_p)$, where p is a prime.

1 Introduction

Our principal use of Magma [7] has been to construct examples that are larger than those that can be constructed by hand, using the outcome of these constructions to deduce the possible presence of a theorem. In some cases we may then go ahead and prove the general result by hand. Our examples here demonstrate two such constructions that led to proofs, one that is certainly true and has been verified in some cases but includes some interesting new conjectures, and another that is proved in one case and still in the process of being formulated for others.

Our aim is to demonstrate the effective use of Magma to make these constructions; these applications require no computing background at all, which is of course one of the most useful aspects of Magma.

The three examples we will use for illustration are briefly as follows:

*Support of NSF grant GER-9450080 acknowledged

1. Given a finite projective plane Π of even order n (in practice we take $n = 2^m$) with an oval (i.e. an $(n+2)$-arc, or hyperoval), we define an oval-design, which is a 2-$(\binom{n}{2}, \frac{n}{2}, 1)$ design, by taking the exterior lines as points, and the points not on the oval as blocks. The block graph of such a design gives, on extension, a Hadamard 3-design with parameters 3-$(n^2, \frac{n^2}{2}, \frac{n^2}{4} - 1)$. Any resolution of the oval design — for example one defined by the secants through a point on the oval — may then be used to obtain a constant-sum Hadamard matrix in the same equivalence class, and thereby symmetric $(n^2, \frac{n^2}{2} \pm \frac{n}{2}, \frac{n^2}{4} \pm \frac{n}{2})$ designs. In the case when Π is desarguesian, the oval is regular, and the resolution is defined through the nucleus of the conic, we can obtain designs with the same parameters that have the symmetric difference property, and thus define bent functions, and translate designs.

2. Let Π be the desarguesian projective plane of square order q^2. The set of absolute points and non-absolute lines of a unitary polarity define a 2-$(q^3 + 1, q+1, 1)$ design, a hermitian unital. The unitary group acts 2-transitively on the points of the unital. We are interested in the p-ary codes associated with the hermitian unitals, in the case when p divides $q+1$. Magma helps us conjecture the dimension of this code.

3. Finally we consider the p-ary codes from affine geometry designs of points and t-spaces from $AG_d(F_p)$, where p is a prime. These codes are Reed-Muller (when $p = 2$) or generalized Reed-Muller (when $p > 2$) codes, and their dimensions are well known. We use Magma to help us design a basis of minimum-weight vectors, i.e. of incidence vectors of blocks in this situation.

We arrange this work as follows: in Section 2 we give the necessary general definitions and background to the problems. Sections 3, 4 and 5 will each give a full description of one of the problems and state (but not prove) the relevant theorems, which will be followed by the Magma programs and runs, with a full description of the various steps.

2 Background

The notation used is generally standard and we refer the reader to Assmus and Key [3]. We recall here some of the definitions that we particularly need.

An incidence structure $\mathcal{D} = (\mathcal{P}, \mathcal{B})$ with point set $\mathcal{P}$ and block set $\mathcal{B}$ is a t-(v, k, λ) design if every block is incident with precisely k points and any set of t distinct points are together incident with precisely λ blocks. It follows (see [3, Chapter 1]) that $\mathcal{D}$ is an s-design for any $s < t$; we denote the number of blocks incident with s points by λ_s. The *order* of a t-design, where $t \geq 2$, is $n = \lambda_1 - \lambda_2$. A Steiner design has $\lambda = 1$. A *symmetric design* has $|\mathcal{P}| = |\mathcal{B}|$, and is often denoted simply by the parameters

(v,k,λ). For a symmetric design $\mathcal{D}$, the *complementary structure* is also a symmetric design, and we denote it by $\overline{\mathcal{D}}$. A *parallelism* or *resolution* of a design $\mathcal{D}$ is a partition of the blocks of $\mathcal{D}$ into classes such that each point of $\mathcal{D}$ is on precisely one block from each class. A design with a parallelism is called *resolvable*.

For any field F, $F^{\mathcal{P}}$ is the vector space of functions from $\mathcal{P}$ to F with basis given by the characteristic functions of the singleton subsets of $\mathcal{P}$. If $\mathcal{D} = (\mathcal{P}, \mathcal{B})$ is an incidence structure, the *code* $C_F(\mathcal{D})$ of $\mathcal{D}$ over F is the subspace of $F^{\mathcal{P}}$ spanned by the characteristic functions (incidence vectors) of the blocks of $\mathcal{D}$. If $X \subseteq \mathcal{P}$, denote the characteristic function on X by v^X: thus

$$v^X(x) = \begin{cases} 1 & \text{if } x \in X \\ 0 & \text{if } x \notin X \end{cases},$$

where $v^X(x)$ denotes the value that the function v^X takes at the point x. Then

$$C_p(\mathcal{D}) = \langle v^B | B \in \mathcal{B} \rangle.$$

The dimension of $C_p(\mathcal{D})$ is referred to as the *p-rank* of $\mathcal{D}$.

The *orthogonal* code $C^\perp$ (where the orthogonal is taken with respect to the standard inner product in F^v, i.e. , for $u, w \in F^v$, $(u,w) = \sum_{x \in \mathcal{P}} u(x)w(x)$,) is defined by

$$C^\perp = \{u \,|\, u \in F^v \text{ and } (u,w) = 0 \text{ for all } w \in C\}.$$

Recall that the *weight* of a vector is the number of non-zero entries. Clearly the code from a design will have minimum weight at most the block size k. The vector in F^v, all of whose entries are 1, is called the *all-one vector* and denoted by $\boldsymbol{\jmath}$. Thus $\boldsymbol{\jmath} = v^{\mathcal{P}}$.

The properties of the generalized Reed-Muller codes, and their connection with the codes of the designs from finite geometries may be found in [3, Chapter 5] or [2]. Since the construction in Section 3 was carried out due to a question that arose (see [3, Chapter 7]) concerning the *first order Reed-Muller code* $\mathcal{R}(1,m)$ of length 2^m, we give a brief description here: if V is the vector space of dimension m over $F = F_2$ in the m variables x_i, then $\mathcal{R}(1,m)$ is the subspace of F^V of all polynomial functions in the x_i of degree at most 1. It has dimension $m+1$ and all the vectors other than 0 and $\boldsymbol{\jmath}$ have weight 2^{m-1}, being the incidence vectors of the $(m-1)$-flats of the affine geometry $AG_m(F_2)$. A function $f \in F^V$ is *bent* in the case when $m = 2n$ is even, if the Hamming distance of f from every function in $\mathcal{R}(1,2n)$ is $2^{2n-1} \pm 2^{n-1}$. See [18, Chapter 14] or [3, Chapter 7] for more about bent functions, and further references.

A symmetric 2-design has the *symmetric difference property* (and called an SDP-design) if the symmetric difference of any three blocks is either a block or the complement of a block: see Jungnickel and Tonchev [14], or Kantor [15], for more on this property and designs with these parameters.

3 Oval designs in even-order projective planes

A projective plane of order n is a symmetric Steiner design with parameters 2-$(n^2+n+1, n+1, 1)$. An *oval* in a projective plane of even order n is a set of $n+2$ points that meets each line of the plane in 0 or 2 points; ovals of $n+2$ points are generally called *hyperovals* in the literature. Oval designs form a class of Steiner 2-designs first described by Bose and Shrikhande in [6]. They are defined as follows: let Π be a projective plane of even order $n = 2k$ and let $\mathcal{O}$ be an oval of Π. The *oval design* $W(\Pi, \mathcal{O})$ is the incidence structure having for points the lines of Π exterior to $\mathcal{O}$, and for blocks the points of Π not on the oval $\mathcal{O}$; incidence is given by the incidence in Π. That this is a Steiner system with parameters 2-$(2k^2-k, k, 1)$ and of order $n = 2k$, is easy to show: see [3, Chapter 8].

We use a construction described in [3, Section 7.12], following Goethals and Seidel [12] and Shrikhande and Singh [22]. This initially shows how any 2-$(2k^2-k, k, 1)$ design $\mathcal{D}$, where $k \geq 2$, defines an equivalence class of Hadamard matrices in the following way: take any incidence matrix A for $\mathcal{D}$ and form the $4k^2-1 \times 4k^2-1$ matrix $AA^t - kI$. This is an incidence matrix of a Hadamard 2-$(4k^2-1, 2k^2, k^2)$ design $\mathcal{E}$, whose complementary design $\overline{\mathcal{E}}$ extends (uniquely) to a Hadamard 3-$(4k^2, 2k^2, k^2-1)$ design, $\mathcal{H}$. If the design $\mathcal{D}$ is resolvable then the class of Hadamard matrices that give the design $\mathcal{H}$ contains constant-sum (row or column) matrices (also sometimes called *regular* Hadamard matrices), and hence gives symmetric designs $\mathcal{M}$ and $\overline{\mathcal{M}}$ with parameters 2-$(4k^2, 2k^2 \mp k, k^2 \mp k)$: partition the blocks of $\mathcal{D}$ into parallel classes $\mathcal{P}_i$, for $i = 1, \ldots, 2k+1$, each parallel class containing $2k-1$ lines. Construct an incidence matrix A for $\mathcal{D}$ by labelling the points in any order and the blocks by parallel class, each class $\mathcal{P}_i$ being ordered arbitrarily. Forming $AA^t - kI = M$ gives a (symmetric) incidence matrix M of a 2-$(4k^2-1, 2k^2, k^2)$ design, partitioned through the classes $\mathcal{P}_i$.

To get the constant-sum Hadamard matrix, we partition M into four submatrices

$$M = \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix} \tag{1}$$

where M_1 is $k(2k-1) \times k(2k-1)$, M_2 is $k(2k-1) \times (k+1)(2k-1)$, M_3 is $(k+1)(2k-1) \times k(2k-1)$ and M_4 is $(k+1)(2k-1) \times (k+1)(2k-1)$. We obtain a Hadamard matrix by forming $E = \exp_{-1}(M)$ and bordering with a first row and column of 1's and then show that an equivalent constant row-sum Hadamard matrix can be found. The new symmetric $(4k^2, 2k^2-k, k^2-k)$ design $\mathcal{M}$ has M^* as incidence matrix, where

$$M^* = \begin{bmatrix} 0 & 1\ldots & 0\ldots \\ \underline{1} & M_1 & \overline{M_2} \\ \underline{0} & \overline{M_3} & M_4 \end{bmatrix} \tag{2}$$

and where $\overline{M_i}$ denotes the matrix of the complementary structure to that defined by M_i (i.e. 1 replacing 0, and vice versa), and $\underline{1}$ and $\underline{0}$ denote all-one or all-zero column vectors, respectively.

The only known infinite class of Steiner 2-designs with parameters of the required form are the oval designs, from planes of even order $n = 2^m$ that have ovals. The parameters of these are 2-$(2^{m-1}(2^m-1), 2^{m-1}, 1)$. They are resolvable with resolutions ρ_x defined by each point x on the oval: the 2^m-1 blocks corresponding to the exterior points on a secant through x form a parallel class of blocks, and the full set of 2^m+1 parallel classes forms the resolution ρ_x.

In the case of a desarguesian plane of order 2^m, if the oval $\mathcal{O}$ is regular (a conic together with its nucleus) then a resolution of the oval design using the secants through the *nucleus* was used in Carpenter and Key [9] to show that the code $C_2(\mathcal{H}(\Pi, \mathcal{O}))$ of the Hadamard design contains a copy of the first-order Reed-Muller code $\mathcal{R}(1, 2m)$: see the first paragraph of Result 1 below. In fact the parallel classes corresponding to any two secants through the nucleus produce blocks of the Hadamard design whose incidence vectors in the binary code generate $\mathcal{R}(1, 2m)$. The method of proof of this result leads us to a method of finding designs $\mathcal{S}$ and $\overline{\mathcal{S}}$ with parameters $(2^{2m}, 2^{2m-1} \mp 2^{m-1}, 2^{2m-2} \mp 2^{m-1})$ but for which the binary code has the smallest dimension for these parameters, *viz.* $2m+2$. Using a result of Dillon and Schatz [10], it then follows that the code of the designs has the form $R \cup (f + R)$ where $R \equiv \mathcal{R}(1, 2m)$ and f is a *bent* function. Furthermore, the support of any codeword $w \in R \cup (f + R)$ of weight $2^{2m-1} \pm 2^{m-1}$ is a *difference set* for the elementary abelian subgroup E of $\mathrm{Aut}(R)$, i.e. the translation group in $AGL_{2m}(F_2)$. The difference-set designs defined by these difference sets in E are not in general $\mathcal{S}$ and $\overline{\mathcal{S}}$. The designs $\mathcal{S}$ and $\overline{\mathcal{S}}$ must have the symmetric difference property.

The formal statement of these results, from Carpenter and Key [9, 8], is as follows:

Result 1 *Let Π be the desarguesian projective plane of order 2^m where $m \geq 2$, and let $\mathcal{O}$ be a regular oval (conic plus nucleus) in Π. Let $\mathcal{T}$ be the Hadamard 3-design constructed from the block graph of the oval design $W(\Pi, \mathcal{O})$. Then the binary code $C_2(\mathcal{T})$ contains a copy of the Reed-Muller code $\mathcal{R}(1, 2m)$.*

Further, let $\mathcal{M}$ be a $(2^{2m}, 2^{2m-1}-2^{m-1}, 2^{2m-2}-2^{m-1})$ design obtained as described from a resolution of the oval design $W(\Pi, \mathcal{O})$ obtained by using the secants through the nucleus of $\mathcal{O}$. If M^* is an incidence matrix for $\mathcal{M}$ as in Equation (2), then the incidence vectors of the blocks defined by the first row of M^* together with the blocks defined by two parallel classes $\mathcal{P}_i$ and $\mathcal{P}_j$, where both $i, j \leq 2^{m-1}$, generate a binary code of dimension $2m + 2$ which is the code of a $(2^{2m}, 2^{2m-1} - 2^{m-1}, 2^{2m-2} - 2^{m-1})$ design $\mathcal{S}$ having the symmetric-difference property.

The role of Magma in this work was firstly to be able to construct all the designs and their codes, and to thus be able to examine their various properties. This led to observations that suggested theorems, some of which have now been settled: for example [3, Conjecture 7.12.1] was formulated purely from values for the dimension obtained using Cayley, Magma's predecessor, and has now been proved in [21]. (More updates from [3] are given in [4].) In the case of the result mentioned above, we were able to experiment with the codes using Magma, and this is what led to the suggestion of the result, and then the subsequent proof. What we show below is an illustration of how Magma will create the designs and verify the given properties, as well as examine new questions.

The following program, called 'OvalDesign.m', creates the desarguesian projective plane of order $q = 2^n$ and, in this case, a regular oval. It then forms the oval design.

```
//Special functions required
/* Makes line x_1=0 from sequence pts */
Line := func< pts | { i : i in [1..#pts] | pts[i][1] eq 0 } >;

//Finds the intersection numbers of a set se of points
INos:=func<se,blox|{#(se meet blox[i]): i in [1..#blox]}>;

/*Need to input  n, where q=2^n=order of plane*/
"Input n now, for plane of order 2^n";
readi n;
"n=",n;

 q:=2^n;
 f<w> := GaloisField(q);
 vv := VectorSpace(f, 3);
 gg, pts := ProjectiveGeneralLinearGroup(vv);
 "The projective plane PG_2(",q,")";
 "Number of points=",#pts;
 line := Line(pts);
 "cardinality of line=",#line;
 lines := Setseq(line^gg);
 "Number of lines=",#lines;
 "lines is the sequence of lines";

//Constructs the regular oval x^2=yz in PG(2,2^n)

 fm:={x : x in f} diff {0};
 rov := { Position(pts,  vv![1, y, y^(-1)]  ): y in fm } join
 { Position(pts,  vv![1, 0, 0]  ) }
 join { Position(pts, vv![0,0,1])  }
 join { Position(pts, vv![0,1,0])  };

 #rov,"= |rov|";
 if INos(rov,lines) eq {0,2} then
   "rov IS an oval, and has", #rov,"points";
```

```
     else "NOT an oval";
 end if;

/*Constructs the oval design with blocks called blox from
an oval..default here is the regular oval..in the desarguesian
plane PG_2(2^n). Then makes incidence vectors and tests the
dimension...conjecturally 3^n - 2^n for any oval*/

 ov1:=rov;
/*This forms the seq po of exterior lines to the oval ov1;
then it makes the blocks, bb, of the oval-design*/
 po:=[ lines[i] : i in [1..#lines] | ov1 meet lines[i] eq {}];

 bb1 := SetToIndexedSet( {1..#pts} diff ov1  );
 blox:=[{ j: j in [1..#po]|
     #({bb1[i]} meet po[j]) eq 1}:i in [1..(q^2-1)]];

 ovdes:=Design<2,#po|blox>;
 ovdes, "which is the oval design";

 dc:=LinearCode(ovdes,GF(2));
 "dimension of code is",Dimension(dc);
   if Dimension(dc) eq 3^n-2^n then
    "conjecture true";
      else "conjecture false";
   end if;
 dual := Dual(dc);
 "dimension of perp is",Dimension(dual);
 hull:= dc meet dual;
 "dimension of hull  is",Dimension(hull);
```

The output from a run using this construction, for $n = 4$, now follows:

```
> load ‘‘OvalDesign.m’’;
Loading ‘‘OvalDesign.m’’
Input n now, for plane of order 2^n
4
n= 4
The projective plane PG_2( 16 )
Number of points= 273
cardinality of line= 17
Number of lines= 273
lines is the sequence of lines
18 = |rov|
rov IS an oval, and has 18 points
2-(120, 8, 1) Design with 255 blocks
which is the oval design
```

```
dimension of code is 65
conjecture true
dimension of perp is 55
dimension of hull  is 33
```

Now the longer program, called ‘skew.m’, to get the Hadamard design, and the symmetric designs, as described in Section 2:

```
jvec:=func<v,q| VectorSpace(GF(q),v)![1:i in [1..v]]>;

//Finds the intersection numbers of a set se of points
INos:=func<se,blox|{#(se meet blox[i]): i in [1..#blox]}>;

// to get PG_2(q) where q=2^m1; input m1

p:=2;q:=2^m1;
//First construct the desarguesian projective plane of order q

Line := func< pts | { i : i in [1..#pts] | pts[i][1] eq 0 } >;
f<w> := GaloisField(q);
vv := VectorSpace(f, 3);
gg, pts := ProjectiveGeneralLinearGroup(vv);
 "The projective plane PG(2,",q,")";
 "Number of points=",#pts;
line := Line(pts);
 "Cardinality of line=",#line;
lines := line^gg;
 "Number of lines=",#lines;
lins:=SetToSequence(lines);

/*Constructs the regular oval x^2=yz in PG(2,q) as a set of
points in the sequence called pts */

fm:={x : x in f} diff {0};
rov := {@ Position(pts,  vv![1, y, y^(-1)]  ): y in fm @} join
{@ Position(pts,  vv![1, 0, 0]  ) @}
join {@ Position(pts, vv![0,0,1])  @}
join {@ Position(pts, vv![0,1,0])  @};
/*rov is the oval, i.e. conic plus nucleus*/
 "\nRegular oval=rov =", rov;
if INos(rov,lins) eq {0,2} then
 "rov IS an oval, and has", #rov,"points";
 else "NOT an oval";
end if;

ov1:=rov;
```

```
pt:=Position(pts,vv![1,0,0]);
 "\nNow make the secants to rov through pt=",pt,"=",pts[pt];

/* makes the tangents through a point pt to a conic , ov1*/;
tangents:=[ lins[i] : i in [1..#lins] | pt in (rov meet lins[i])];
 "|tangents through pt|=",#tangents;

/*now construct the hadamard design*/
shblox:=[];
tpts:=[];
for i:=1 to #tangents do
z:=tangents[i];
 for x in (z diff ov1) do
   tpts:=Append(tpts,x);
   bl:=&join[lins[j]   diff ov1:
     j in [1..#lins] | (x in lins[j]) and (#(lins[j] meet ov1) eq 2)];
   shblox:=Append(shblox,bl);
   end for;
end for;

nblox:=[ {Position(tpts,x): x in shblox[i]} : i in [1..#shblox]];
hdes:=Design<2,#tpts|nblox>;
"Hadamard design";
hdes;
mm:=(2^m1-1)*2^(m1-1);mn:=2^(2*m1)-1;
vset1:={1..mm};vset2:={mm+1..mn};

 "\nNow make the regular matrix design";
tblox:=[vset1] cat [ {2^(2*m1)} join
   (vset1 diff nblox[i]) join (vset2 meet nblox[i]):
   i in [1..mm]] cat [(vset2 diff nblox[i]) join (vset1 meet nblox[i]):
i in [mm+1..2^(2*m1)-1]];
tdes:=Design<2,q^2|tblox>;
tdes;
"symmetric, with blocks called tblox";

 "\nNow look at the code of this design";
s:=LinearCode(tdes,GF(2));
 "Dimension of code=",Dimension(s);
jvec:=jvec(q^2,2);
 "Is the all-one vector in the code?";
jvec in s;


 "\nNow construct an SDP design ";
rblox:=[tblox[i]: i in {1..(2*(q-1)+1)}];
```

```
rdes:=Design<0,q^2|rblox>;
sx:=LinearCode(rdes,GF(2));

 "Dimension of SDP code=",Dimension(sx);
"and weight distribution is";
wd:=WeightDistribution(sx);
wd;
sdpwd:=[<0,1>,<2^(2*m1 -1)-2^(m1-1),q^2>,
<2^(2*m1 -1),2*(2^(2*m1) -1)>,<2^(2*m1 -1)+2^(m1-1),q^2>,<q^2,1>];
 if wd eq sdpwd then
   "Checked that weight-", 2^(2*m1 -1)-2^(m1-1),
   "words will give an SDP design";
 else
   "no good";
 end if;
mblox:=[ Support(w): w in MinimumWords(sx)];
mdes:=Design<2,q^2|mblox>;
mdes;
"Blocks of SDP design called mblox";
sm:=LinearCode(mdes,GF(2));
 "checked dimension of code of SDP design=",Dimension(sm);

 "\nNow prepare to find the difference-set design ";
"first form the design of",2*(2^(2*m1)-1),
"supports of the weight-",2^(2*m1-1),"words
in the above code, i.e. a";
n3:=2^(2*(m1)-1);
hblox:=[ Support(w): w in Words(sx,n3)];
hdes:=Design<2,q^2| hblox>;
hdes;
"Find the translation subgroup 'cy' of its automorphism group";
a2:=PointGroup(hdes);
sy:=Sylow(a2,2);
cy:=Core(a2,sy);
 "|cy|=", Order(cy);
 IsElementaryAbelian(cy),"that cy is elementary abelian 2-gp";

/* pick arbitrary block in mblox and translate it by cy*/
"Now translate an arbitrary block of the SDP design to get";
bl:=mblox[15];
nblox:=SetToSequence(bl^cy);
 ndes:=Design<2,q^2| nblox>;
 ndes;
 "This is a difference-set design, blocks called nblox";

 sn:=LinearCode(ndes,GF(2));
```

```
"Dimension of code of difference-set design=",Dimension(sn);

"\nThe automorphism groups of the 2-(",q^2,",",
2^(2*m1 -1)-2^(m1-1),",",2^(2*m1 -2)-2^(m1-1),") designs:";

saut:=PointGroup(tdes);
"tblox design (regular matrix) has aut gp of order",Order(saut);

aut1:=PointGroup(mdes);
"mblox design (SDP or bent function) has aut gp of order",Order(aut1);

but1:=PointGroup(ndes);
"nblox design (difference set) has aut gp of order",Order(but1);
```

A run using the plane of order 8 follows:

```
 m1:=3;
> load ‘‘skew.m’’;
Loading ‘‘skew.m’’
The projective plane PG(2, 8 )
Number of points= 73
Cardinality of line= 9
Number of lines= 73

Regular oval=rov = {@ 33, 56, 1, 2, 3, 48, 72, 7, 52, 63 @}
rov IS an oval, and has 10 points

Now make the secants to rov through pt= 1 = (  1   0   0)
|tangents through pt|= 9
Hadamard design
2-(63, 31, 15) Design with 63 blocks

Now make the regular matrix design
2-(64, 28, 12) Design with 64 blocks
symmetric, with blocks called tblox

Now look at the code of this design
Dimension of code= 14
Is the all-one vector in the code?
true

Now construct an SDP design
Dimension of SDP code= 8
and weight distribution is
[ <0, 1>, <28, 64>, <32, 126>, <36, 64>, <64, 1> ]
Checked that weight- 28 words will give an SDP design
2-(64, 28, 12) Design with 64 blocks
```

```
Blocks of SDP design called mblox
checked dimension of code of SDP design= 8

Now prepare to find the difference-set design
first form the design of 126 supports of the weight- 32 words
in the above code, i.e. a
2-(64, 32, 31) Design with 126 blocks
Find the translation subgroup 'cy' of its automorphism group
|cy|= 64
true that cy is elementary abelian 2-gp
Now translate an arbitrary block of the SDP design to get
2-(64, 28, 12) Design with 64 blocks
This is a difference-set design, blocks called nblox
Dimension of code of difference-set design= 14

The automorphism groups of the 2-( 64 , 28 , 12 ) designs:
tblox design (regular matrix) has aut gp of order 84
mblox design (SDP or bent function) has aut gp of order 43008
nblox design (difference set) has aut gp of order 43008
```

4 Codes from hermitian unitals

A unital, or unitary design, is a Steiner 2-design with parameters 2-$(m^3+1, m+1, 1)$. If Π is the desarguesian plane of square order q^2, then the set of absolute points and non-absolute lines of a unitary polarity form a unital, called the hermitian unital. The codes of these are in general not studied, nor understood; by a result of Mortimer [20] we need only look at the case of p dividing $q+1$. Even the dimensions of the codes are not known except in some small cases. What we did with Magma here was to find the p-rank for hermitian unitals of orders up to $q = 13$, for p each prime divisor of $q+1$. From this a clear formula emerged, and thus we were able to make a conjecture about the p-rank in the general case: see the end of this section.

The following function, stored as a function 'HUnital.fu', constructs the hermitian unital as a design in the plane of order q^2.

```
HUnital := function(p, m);
    /* Given a prime p and an integer m, create the hermitian unital
    on q^3+1 points, where q = p^m. The unital is constructed in the
    plane of order (p^m)^2.*/

    n := 2*m; q1 := p^n; q := p^m;
     "Constructing the hermitian unital in PG_2(",q1,")";
    F<w> := GaloisField(q1);
    V := VectorSpace(F, 3);
    PGL, pts := ProjectiveGeneralLinearGroup(V);
```

```
    line := Line(pts);
    lines := SetToSequence(line^PGL);

    /* Construct the hermitian unital
     x^(q+1) + y^(q+1) + z^(q+1) = 0 in PG(2,q)
    as an indexed set of points, hunital */

    P<t> := PolynomialRing(F);
    hunital := {@ Position(pts, V![1, y, z[1]]) :
       z in Roots(1+y^(q+1)+t^(q+1), F), y in F @}
       join {@ Position(pts, V![0, 1, z]) :  z in F |
        z^(q+1) eq -F!1 @};
    /* The blocks of the design are the intersections of lines of
    the plane with the unital having cardinality q+1.
    The points are renumbered.    */

    blks := [ { Position(hunital, pt) : pt in blk } :
    i in [1..#lines] | #blk eq (q+1) where
                         blk is lines[i] meet hunital ];
     "We have the hermitian unital, a
    2-(",#hunital,",",q+1,",",1,") design";
    BIBD := recformat< v:Z, k:Z, lambda:Z, blocks >
     where Z is Integers();
    return rec< BIBD | v := #hunital, k := q+1,
    lambda := 1, blocks := blks >;

end function;
```

To test the p-rank, for p dividing $q+1$, we ran the following program called 'hermtest.m':

```
  for p in {@2,3,4,5,7,8,9,11@} do
   q:=p;
    hu:=HUnital(p,1);
     blox:=hu`blocks;
      hdes:=Design<0,hu`v|blox>;
       "b/q=",#blox/q;
         for x in Seqset(PrimeDivisors(q+1)) do
           x,"-rank=", Dimension(LinearCode(hdes,GF(x)));
         end for;
   end for;
```

Note that the 'Design' function is used here with $\lambda = 0$ even though we know of course that $\lambda = 2$; this is to save computing time. A run of this went as follows

```
Magma V2.20-2    Sun Sep 27 1998 22:09:49 on mathieu
Type ? for help.  Type <Ctrl>-D to quit.
> load "HUnital.fu";
```

```
Loading "HUnital.fu"
> load "hermtest.m";
Loading "hermtest.m"

Constructing the hermitian unital in PG_2( 4 )
We have the hermitian unital, a 2-( 9 , 3 , 1 ) design
b/q= 6
3 -rank= 6

Constructing the hermitian unital in PG_2( 9 )
We have the hermitian unital, a 2-( 28 , 4 , 1 ) design
b/q= 21
2 -rank= 21

Constructing the hermitian unital in PG_2( 16 )
We have the hermitian unital, a 2-( 65 , 5 , 1 ) design
b/q= 52
5 -rank= 52

Constructing the hermitian unital in PG_2( 25 )
We have the hermitian unital, a 2-( 126 , 6 , 1 ) design
b/q= 105
2 -rank= 105
3 -rank= 105

Constructing the hermitian unital in PG_2( 49 )
We have the hermitian unital, a 2-( 344 , 8 , 1 ) design
b/q= 301
2 -rank= 301

Constructing the hermitian unital in PG_2( 64 )
We have the hermitian unital, a 2-( 513 , 9 , 1 ) design
b/q= 456
3 -rank= 456

Constructing the hermitian unital in PG_2( 81 )
We have the hermitian unital, a 2-( 730 , 10 , 1 ) design
b/q= 657
2 -rank= 657
5 -rank= 657

Constructing the hermitian unital in PG_2( 121 )
We have the hermitian unital, a 2-( 1332 , 12 , 1 ) design
b/q= 1221
2 -rank= 1221
3 -rank= 1221
```

We have computed the p-ranks, for p dividing $q+1$, of the hermitian unitals for all q such that $q \leq 13$, and we have found that all these codes have dimension $b/q = (q^2-q+1)q$, where b is the number of blocks of the unital. This is the formula suggested originally by Andriamanalimanana [1] based on computations up to and including $q=5$. The further computations up to $q=13$ now lead us to formally state this as a conjecture:

Conjecture 1 *Let $\mathcal{H}$ be the hermitian unital on q^3+1 points. If p is any prime dividing $q+1$, then the p-rank of $\mathcal{H}$ is $(q^2-q+1)q$.*

5 Bases of incidence vectors

Here we refer the reader to Assmus and Key [5] or [2], or [3, Chapter 5] for more details of the connection between the codes of the finite geometry designs and the generalized Reed-Muller codes, and the vast bibliography of prior work in this area. The constructions we make here are based on the following two results from Gao and Key [11]. The reader may refer to Key [16] for a discussion of bases of minimum-weight vectors for designs from finite geometries in general. The results we state here were established using the Jennings basis (see Jennings [13] and Lombardo-Radice [17]) in the binary case, and the monomial basis in the general case, using the fact that the dimension of the codes is known. For q any prime power, $AG_{m,r}(F_q)$ and $PG_{m,r}(F_q)$ will denote the affine and projective designs of points and r-dimensional flats and subspaces (respectively), in the affine and projective geometries of dimension m over the finite field F_q.

Result 2 *For any r and any m the binary code of $AG_{m,r}(F_2)$ has a basis of incidence vectors of r-flats consisting of those with equation as follows*

$$X_i = \begin{cases} 0 & \text{for } i \notin \{i_1, \ldots, i_r\} \\ 1 & \text{for } i \in \{i_{r+1}, \ldots, i_{r+t}\} \end{cases}$$

for $0 \leq t \leq m-r$, where

$$1 \leq i_1 < i_2 < \ldots < i_{r+t} \leq m.$$

The dimension of the code is $\sum_{s=0}^{r} \binom{m}{m-s}$.

The following result was formulated after some experiments with Magma, involving testing sets of hyperplanes chosen using geometrical and algebraic guidelines, for linear independence; if the set consisted of the known number for the dimension of the code (which is well known: see, for example, [2]), we could deduce we had a basis. This result has now been proved in [11].

Result 3 *Let $\mathcal{D}$ be the design $AG_{m,m-1}(F_p)$ of points and hyperplanes in the affine space of dimension m over the prime field F_p. For $0 \le t \le \mu = \min(m, p-1)$ define a set $\mathcal{K}_t$ of hyperplanes with equations as follows:*

$$\begin{aligned} \mathcal{K}_0 &= \{X_1 + 1 = 0\} \\ \mathcal{K}_t &= \{\sum_{j=1}^{t} a_j X_{i_j} + b = 0\} \end{aligned}$$

for all choices of $\{i_1, i_2, \ldots, i_t, a_1, a_2, \ldots, a_t, b\}$ such that $1 \le i_1 < i_2 < \ldots < i_t \le m$, $1 = a_1 < a_2 < \ldots < a_t \le p-1$ and $b = 0$ or $1 < b < a_2$. (When $t = 1$ we interpret a_2 as equal to p in the last inequality.)

If $\mathcal{K} = \bigcup_{t=0}^{\mu} \mathcal{K}_t$, then the incidence vectors of the hyperplanes in $\mathcal{K}$ form a basis for the p-ary code $C_p(\mathcal{D})$ of dimension $\binom{p+m-1}{m}$.

When $m = 2$ and we have the desarguesian affine plane over F_p, this basis is of the same form as those found by Moorhouse [19].

A program, called 'basrm.m', that gets the basis described in Result 2 is as follows:

```
/* Numbers m and n, n LESS THAN or equal to  m; gets a sequence of all
ordered subsets (as sequences) of size n of the numbers {1..m}*/
 nmseq := func< n,m | [Sort(Setseq(x)): x in Subsets({1..m}, n)]>;

/* input values of m, r,  to get basis
of vectors of minimum weight (2^r) for R(m-r,m), i.e.
of r-flats in AG_{m,r}(F_2)*/

"q=p=2","m=",m,"r=",r;
f<w> := GaloisField(2);
vv := VectorSpace(f, m);
gg, pts := AffineGeneralLinearGroup(vv);

 "number of points=",#pts;
 "block size=",2^r;
 "design of points and",r,"-flats in AG_",m,"(F_2)";

rseq:=nmseq(r,m);
bas:=[{j: j in [1..#pts]| pts[j] in sub<vv| {Basis(vv)[k]:
   k in rseq[i]}>}: i in [1..#rseq]];

 "standard subspaces give",#bas;
//Now for the translates

for i:=1 to #rseq do
x:=rseq[i][r];
 bt:=Sort(Setseq({y: y in [1..m]|y gt x}));
```

```
  for s:=1 to m-r do
  if s le #bt then
   tvj:=nmseq(s,#bt);
   svj:=[Sort([bt[tvj[l][j]]:j in [1..s]]):l in [1..#tvj]];
    for l:=1 to #svj do
       sp:=sub<vv|{Basis(vv)[k]:
       k in (Seqset(svj[l]) join Seqset(rseq[i]))}>;
       ssp:={ x : x in sp |{x[k] :  k in svj[l] } eq {1}};
       bl:={@ Position(pts,x): x in ssp @};
         bas:=Append(bas,bl);
    end for;
  end if;
 end for;
end for;
 #bas,"=|bas|";
 sdes:=Design<0,#pts|bas>;
 s:=LinearCode(sdes,GF(2));

 Dimension(s),"= dimension";
(#bas eq Dimension(s)) and
(#bas eq Dimension(ReedMullerCode(m-r,m))),
"that bas is a basis";
 " for binary code of AG_{",m,",",r,"}(F_2)";
 "i.e. the Reed-Mullercode R(",m-r,",",m,")";
 "with dimension=",Dimension(ReedMullerCode(m-r,m));
```

The output from a run with $m = 6$ and $r = 3$ and 2:

```
 m:=6;r:=3;
> load ``basrm.m'';
Loading ``basrm.m''
q=p=2 m= 6 r= 3
number of points= 64
block size= 8
design of points and 3 -flats in AG_ 6 (F_2)
standard subspaces give 20
42 =|bas|
42 = dimension
true that bas is a basis
 for binary code of AG_{ 6 , 3 }(F_2)
i.e. the Reed-Mullercode R( 3 , 6 )
with dimension= 42
> r:=2;
> load ``basrm.m'';
Loading ``basrm.m''
q=p=2 m= 6 r= 2
number of points= 64
```

```
block size= 4
design of points and 2 -flats in AG_ 6 (F_2)
standard subspaces give 15
57 =|bas|
57 = dimension
true that bas is a basis
 for binary code of AG_{ 6 , 2 }(F_2)
i.e. the Reed-Mullercode R( 4 , 6 )
with dimension= 57
```

A program, called 'basgrm.m', to demonstrate Result 3 is the following

```
/* Numbers m and n, n LESS THAN or equal to  m; gets a sequence of all
ordered subsets (as sequences) of size n of the numbers {1..m}*/
 nmseq := func< n,m | [Sort(Setseq(x)): x in Subsets({1..m}, n)]>;

/* basis of minimum-weight vectors (hyperplanes)
for p-ary code of design of points and hyperplanes in AG_m(F_p),
p a prime*/
/*input m=dimension,p = prime*/
t:=m-1;

"p=",p,"m=",m,"r=",t;
f<w> := GaloisField(p);
vv := VectorSpace(f, m);
gg, pts := AffineGeneralLinearGroup(vv);
 "number of points=",#pts;
 "block size=",p^t;

mu:=Minimum(m,p-1);
bas:=&cat[[{j: j in [1..#pts]|pts[j][i] eq k}:k in [0..p-2]]:
i in [1..m]] cat [{j: j in [1..#pts]|pts[j][1] eq p-1 }];

 #bas,"gives the standard hyperplanes and translates";

for s:=2 to mu do
sseq:=nmseq(s,m);
pseq:=nmseq(s,p-1);
   p1seq:=[pseq[i]: i in [1..#pseq]| pseq[i][1] eq 1 ];
   for l:=1 to #sseq do
    z:=sseq[l];
     for k:=1 to #p1seq do
       w:=&+[p1seq[k][x]*Basis(vv)[z[x]]:x in [1..s]];
       for b:=0 to p1seq[k][2]-2 do
       bl:={j:j in [1..#pts]|InnerProduct(pts[j],w) eq b};
       bas:=Append(bas,bl);
       end for;
```

```
     end for;
  end for;
end for;
 "|bas|=",#bas;

sdes:=Design<0,#pts|bas>;
s:=LinearCode(sdes,GF(p));

 "p-rank of design of points and hyperplane=",Binomial(m+p-1,m);
 "dimension found=",Dimension(s);
 #bas eq Dimension(s), "that bas is a basis for the",
 p,"-ary code of AG_{",m,",",m-1,"}(F_",p,")";
 "bas is a basis of minimum-weight vectors for the generalized
Reed-Muller code R_",p,"(",p-1,",",m,")";
```

A run with $q = 5$ generates the following output:

```
 p:=5;m:=5;
> load ‘‘basgrm.m’’;
Loading ‘‘basgrm.m’’
p= 5 m= 5 r= 4
number of points= 3125
block size= 625
21 gives the standard hyperplanes and translates
|bas|= 126
p-rank of design of points and hyperplane= 126
dimension found= 126
true that bas is a basis for the 5 -ary code of AG_{ 5 , 4 }(F_ 5 )
bas is a basis of minimum-weight vectors for the generalized
Reed-Muller code R_ 5 ( 4 , 5 )
```

Acknowledgement:

The author would like to thank the Department of Computer Science and Engineering and the Center for Communication and Information Science (CCIS) at the University of Nebraska for their hospitality during the academic year 1994-95. The author also thanks the anonymous referees for their careful reading of the paper, and their suggestions for its improvement.

References

[1] Bruno Ratsimandefitra Andriamanalimanana. *Ovals, Unitals and Codes.* PhD thesis, Lehigh University, 1979.

[2] E. F. Assmus, Jr. and J. D. Key. Polynomial codes and finite geometries. To appear (1998) in Handbook of Coding Theory, edited by V. S. Pless and W. C. Huffman.

[3] E. F. Assmus, Jr. and J. D. Key. *Designs and their Codes*. Cambridge University Press, 1992. Cambridge Tracts in Mathematics, Vol. 103 (Second printing with corrections, 1993).

[4] E. F. Assmus, Jr. and J. D. Key. Designs and codes: an update. *Des. Codes Cryptogr.*, 9:7–27, 1996.

[5] Edward F. Assmus, Jr. and Jennifer D. Key. Codes and finite geometries. Technical report, INRIA, 1993. Report No. 2027.

[6] R. C. Bose and S. S. Shrikhande. On the construction of sets of mutually orthogonal latin squares and the falsity of a conjecture of Euler. *Trans. Amer. Math. Soc.*, 95:191–209, 1960.

[7] John Cannon and Catherine Playoust. *An Introduction to Magma*. School of Mathematics and Statistics, University of Sydney, 1994.

[8] L. L. Carpenter and J. D. Key. On Hadamard matrices from resolvable Steiner designs. *Congr. Numer.*, 108:53–63, 1995.

[9] L. L. Carpenter and J. D. Key. Oval designs and Reed-Muller codes,. *J. Combin. Math. & Combin. Comput.*, 22:79–85, 1996.

[10] J. F. Dillon and J. R. Schatz. Block designs with the symmetric difference property. In Robert L. Ward, editor, *Proceedings of the NSA Mathematical Sciences Meetings*, pages 159–164. The United States Government, 1987.

[11] S. Gao and J. D. Key. Bases of minimum-weight vectors for codes from designs. *Finite Fields Appl.*, 4:1–15, 1998.

[12] J. M. Goethals and J. J. Seidel. Strongly regular graphs derived from combinatorial designs. *Canad. J. Math.*, 22:597–614, 1970.

[13] S. A. Jennings. The structure of the group ring of a p-group over a modular field. *Trans. Amer. Math. Soc.*, 50:175–185, 1941.

[14] Dieter Jungnickel and Vladimir D. Tonchev. On symmetric and quasi-symmetric designs with the symmetric difference property and their codes. *J. Combin. Theory, Ser. A*, 59:40–50, 1992.

[15] William M. Kantor. Plane geometries associated with certain 2-transitive groups. *J. Algebra*, 37:489–521, 1975.

[16] J. D. Key. Bases for codes of designs from finite geometries. *Congr. Numer.*, 102:33–44, 1994.

[17] Lucio Lombardo-Radice. Intorno alle algebre legate ai gruppi di ordine finito. *Rend. Sem. Mat. Fac. Sci. R. Univ. Roma (4)*, 2:312–322, 1938.

[18] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes.* Amsterdam: North-Holland, 1983.

[19] G. Eric Moorhouse. Bruck nets, codes, and characters of loops. *Des. Codes Cryptogr.*, 1:7–29, 1991.

[20] Brian Mortimer. The modular permutation representations of the known doubly transitive groups. *Proc. London Math. Soc. (3)*, 41:1–20, 1980.

[21] Thomas E. Norwood and Qing Xiang. On GMW designs and a conjecture of Assmus and Key. *J. Combin. Theory, Ser. A*, 78:162–168, 1997.

[22] S. S. Shrikhande and N. K. Singh. On a method of constructing incomplete block designs. *Sankhȳa, A*, 24:25–32, 1962.