Result.Math. *Online First*

DOI 10.1007/s00025-007-0268-y

Results in Mathematics

An Algorithm for Constructing Some Maximal Arcs in $\mathrm{PG}(2, q^2)$

Angela Aguglia and Luca Giuzzi

Abstract. In 1974, J. Thas constructed a new class of maximal arcs for the Desarguesian plane of order q^2. The construction relied upon the existence of a regular spread of tangent lines to an ovoid in $\mathrm{PG}(3, q)$ and, in particular, it does apply to the Suzuki–Tits ovoid. In this paper, we describe an algorithm for obtaining a possible representation of such arcs in $\mathrm{PG}(2, q^2)$.

Mathematics Subject Classification (2000). 05B25, 51E21.

Keywords. Maximal arcs, spreads, algebraic curves.

1. Introduction

In a finite projective plane of order q, a maximal (k, n)-arc $\mathcal{K}$, where $k \geq 1$ and $2 \leq n \leq q + 1$, is a non-empty set of k points which is met by every line of the plane in either 0 or n points. The integer n is called the *degree* of the arc $\mathcal{K}$.

Trivial examples of maximal arcs of degree $q + 1$ and q are respectively the point-set of $\mathrm{PG}(2, q)$ and the point-set of an affine subplane $AG(2, q)$ of $\mathrm{PG}(2, q)$.

It has been shown in [3, 4], that non-trivial maximal arcs do not exist in $\mathrm{PG}(2, q)$ for q odd. On the other hand, when q is even, several classes of non-trivial maximal arcs are known. In fact, hyperovals and their duals are maximal arcs. Apart from them, known constructions of degree n maximal arcs in $PG(2, q)$ are: one construction by Denniston [7] based on a linear pencil of conics, two constructions of J. A. Thas [16, 17], constructions by Mathon [15], and by Hamilton and Mathon [10] utilising closed sets of conics. However, most of the known examples of degree n maximal arcs (with the notable exception of a class of maximal arcs arising from the [16] construction) consist of the union of $n - 1$ pairwise disjoint conics, together with their common nucleus N. We shall call these arcs *conical.* Observe that any conical maximal arc is covered by a completely reducible curve of

Research supported by the Italian Ministry MURST, Strutture geometriche, combinatoria e loro applicazioni.

degree $2n-1$, whose components are $n-1$ conics and a line through the point N; in [2] it is shown that such a curve has minimum degree. In the present paper we determine, using the computer algebra package GAP [8], equations for algebraic plane curves of minimum degree passing through all the points of a maximal arc $\mathcal{K}$; we are mostly interested in those cases in which $\mathcal{K}$ is not conical. In particular, we will show that arcs arising from Suzuki–Tits ovoids by the [16] construction cannot be covered by a curve of low degree.

2. Reguli in $\mathrm{PG}(3,q)$

We recall some basic properties of reguli and spreads of $\mathrm{PG}(3,q)$; see [12].

Definition 1. A *regulus* of $\mathrm{PG}(3,q)$ is a collection of $q+1$ mutually disjoint lines such that any line of $PG(3,q)$ meeting three of them necessarily meets them all.

A standard result, see [13], shows that any three pairwise disjoint lines ℓ_1, ℓ_2, ℓ_3 of $\mathrm{PG}(3,q)$ lie together in a unique regulus, say $\mathcal{R}(\ell_1,\ell_2,\ell_3)$.

Definition 2. Let ℓ_1,ℓ_2,ℓ_3 be 3 pairwise disjoint lines of $\mathrm{PG}(3,q)$. The *opposite regulus* to $\mathcal{R}(\ell_1,\ell_2,\ell_3)$ is the set

$$\mathcal{R}^o(\ell_1,\ell_2,\ell_3)$$

of all lines ℓ of $\mathrm{PG}(3,q)$ such that

$$\ell\cap\ell_i\neq\emptyset\,,\quad \text{for}\quad i=1,2,3\,.$$

The set $\mathcal{R}^o(\ell_1,\ell_2,\ell_3)$ is also a regulus. We may compute the regulus containing ℓ_1,ℓ_2 and ℓ_3 as the set

$$\mathcal{R}(\ell_1,\ell_2,\ell_3)=\mathcal{R}^o(m_1,m_2,m_3)\,,$$

where m_1,m_2,m_3 are distinct elements of $\mathcal{R}^o(\ell_1,\ell_2,\ell_3)$.

Definition 3. A k-span of $\mathrm{PG}(3,q)$ is a set of k mutually skew lines. A (q^2+1)-span is called a *spread*.

Observe that a spread is a partition of the points of $\mathrm{PG}(3,q)$ in disjoint lines.

Definition 4. A spread $\mathcal{S}$ is *regular* or *Desarguesian*, if for any three lines $\ell_1,\ell_2,\ell_3\in\mathcal{S}$,

$$\mathcal{R}(\ell_1,\ell_2,\ell_3)\subseteq\mathcal{S}\,.$$

Any two regular spreads of $\mathrm{PG}(3,q)$ are projectively equivalent. To describe a spread of tangent lines to a given ovoid, we shall use the following notion of closure.

Definition 5. The *regular closure* of a set S of lines of $\mathrm{PG}(3,q)$ is the smallest set T of lines of $\mathrm{PG}(3,q)$ containing S such that for any 3 distinct elements $\ell_1,\ell_2,\ell_3\in T$,

$$\mathcal{R}(\ell_1,\ell_2,\ell_3)\subseteq T\,.$$

Examples of sets closed under this operation are regular spreads of $\mathrm{PG}(3, q)$ and reguli. In fact, a regular spread is uniquely determined by four of its lines, supposed they are in suitable position.

Theorem 6. *There exists exactly one regular spread containing any given* 4 *mutually skew lines* $\ell_1, \ell_2, \ell_3, \ell_4$ *of* $\mathrm{PG}(3, q)$*, provided that* $\ell_4 \notin \mathcal{R}(\ell_1, \ell_2, \ell_3)$*.*

Proof. By [14] there is a Desarguesian spread containing any two reguli with 2 lines in common. We now show that this spread is the regular closure of $\ell_1, \ell_2, \ell_3, \ell_4$. Any Desarguesian spread containing $\mathcal{R}(\ell_1, \ell_2, \ell_4)$ and $\mathcal{R}(\ell_1, \ell_3, \ell_4)$ must clearly contain also the $(q^2 - q + 2)$-span of lines given by

$$\bigcup_{\substack{x \in \mathcal{R}(\ell_1, \ell_2, \ell_3) \\ x \neq \ell_1}} \mathcal{R}(\ell_1, x, \ell_4)\,.$$

By [12, Lemma 17.6.2], a spread containing such span is unique. The result follows. □

3. Thas [16] maximal arcs

We shall make extensive use of the representation of $\mathrm{PG}(2, q^2)$ in $\mathrm{PG}(4, q)$ due to André [1] and Bruck and Bose [5,6].

Let $\mathrm{PG}(4, q)$ be a projective 4-space over the finite field $\mathrm{GF}(q)$ and let suppose $\mathcal{S}$ be a regular spread of a fixed hyperplane $\Sigma = \mathrm{PG}(3, q)$ of $\mathrm{PG}(4, q)$. Then $PG(2, q^2)$ can be represented as the incidence structure $(\mathcal{P}, \mathcal{L}, I)$ where the point set $\mathcal{P}$ is given by the points of $\mathrm{PG}(4, q) \setminus \Sigma$ together with the elements of $\mathcal{S}$, the line set $\mathcal{L}$ consists of all the planes of $\mathrm{PG}(4, q) \setminus \Sigma$ which meet Σ in a line of $\mathcal{S}$ together with the spread $\mathcal{S}$, and incidence is inclusion.

In particular, $\mathcal{S}$ represents the "line at infinity" of the affine plane $AG(2, q^2) \subseteq \mathrm{PG}(2, q^2)$. Recall that projectively equivalent spreads of $\mathrm{PG}(3, q)$ induce, via Bruck–Bose construction isomorphic projective planes or order q^2. In particular, any two regular spreads of $\mathrm{PG}(3, q)$ induce a representation of the Desarguesian projective plane $\mathrm{PG}(2, q^2)$.

Using the aforementioned model, Thas obtained maximal arcs in the Desarguesian plane as follows. Let $\mathcal{O}$ be an ovoid in the hyperplane Σ such that every element of the spread $\mathcal{S}$ meets $\mathcal{O}$ in exactly one point. Fix a point V in $PG(4, q) \setminus \Sigma$ and let $\overline{\mathcal{K}}$ be the set of points in $PG(4, q) \setminus \Sigma$ collinear with V and a point on $\mathcal{O}$. Then $\overline{\mathcal{K}}$ corresponds to a maximal $(q^3 - q^2 + q, q)$-arc $\mathcal{K}$ in $PG(2, q^2)$.

In [16], it has been remarked that if $\mathcal{O}$ is an elliptic quadric then the maximal arc thus constructed turns out to be of Denniston type. Using algebraic techniques, it has been shown in [10] that, when $\mathcal{O}$ is a Suzuki–Tits ovoid, $\mathcal{K}$ cannot be obtained from a closed set of conics. In fact, in this case the arc is not conical at all.

In order to provide a direct representation of a Thas [16] maximal arc in $\mathrm{PG}(2, q^2)$, where $q > 4$ is an even prime power, we shall use for $\mathrm{PG}(4, q)$ homogeneous coordinates (z, x_1, x_2, y_1, y_2). The hyperplane at infinity Σ has equation

$z=0$. Let $\mathcal{S}$ be a regular spread of Σ and denote by $\pi=\mathrm{PG}(2,q^2)$ the corresponding Desarguesian plane obtained via Bruck–Bose construction. We shall use homogeneous coordinates (z,x,y) for π, so that the line at infinity has equation $z=0$. It is always possible to assume that, up to a projectivity, the spread $\mathcal{S}$ contains the lines

$$\begin{aligned} \ell_1 &= \big\langle (1,0,0,1),(0,1,1,0) \big\rangle \\ \ell_2 &= \big\langle (1,0,0,0),(0,1,0,0) \big\rangle \\ \ell_3 &= \big\langle (0,0,1,0),(0,0,0,1) \big\rangle . \end{aligned} \tag{1}$$

The application θ which realises the correspondence between the points of $\mathrm{PG}(4,q)$ and those of $\mathrm{PG}(2,q^2)$ should map any line ℓ of the spread $\mathcal{S}$ into a point of $\mathrm{PG}(2,q^2)$. In particular, in order to have

$$\begin{aligned} \theta(\ell_1) &= (0,1,1) \\ \theta(\ell_2) &= (0,1,0) \\ \theta(\ell_3) &= (0,0,1) , \end{aligned}$$

we should choose

$$\theta : \begin{cases} \mathrm{PG}(4,q) \mapsto \mathrm{PG}(2,q^2) \\ (z,x_1,x_2,y_1,y_2) \mapsto (z, x_1+\varepsilon x_2, \varepsilon y_1 + y_2) \end{cases} , \tag{2}$$

where ε is a suitable element of $\mathrm{GF}(q^2)\setminus\mathrm{GF}(q)$.

4. The code

In this section we describe a GAP [8] program to construct a Thas [16] maximal arc $\mathcal{K}$ and determine a minimum degree curve Γ passing through all the points of $\mathcal{K}$. In our code it shall be constantly assumed that $q=2^{2t+1}$, with $t>1$ a global variable.

The simplest way to implement the geometry $\mathrm{PG}(3,q)$ is to consider the point orbit of $\mathrm{GL}(4,q)$ in its action on left-normalised 4-vectors.

```
PG3:=Orbit(GL(4,q),[1,0,0,0]*Z(q)^0,OnLines);
```

Remark 7. It is often convenient to represent the points of $\mathrm{PG}(3,q)$ as integers in the range $1\dots q^3+q^2+q+1$. The number corresponding to any given point is just the position of the corresponding normalised vector in the list `PG3`. This is most interesting when `PG3` is generated as the orbit of a point, say $(1,0,0,0)$, under the action of a Singer group Θ of $\mathrm{PG}(3,q)$.

We now introduce some utility functions.

1. `LineAB` to compute the (projective) line over $\mathrm{GF}(q)$ through two points;
2. `LineAB2` to compute the (projective) line over $\mathrm{GF}(q^2)$ through two points.

3. `Conj` to get the conjugate of a point in PG(n, q^2) under the Frobenius morphism
$$x \mapsto x^q\,.$$

```
#Line (over GF(q))
LineAB:=function(a,b)
 return Set(Union([a],Set(GF(q),x->NormedRowVector(x*a+b))));
end;;

#Line (over GF(q^2))
LineAB2:=function(a,b)
 return Set(Union([a],Set(GF(q^2),x->NormedRowVector(x*a+b))));
end;;

#Conjugate of a point
Conj:=function(x)
 return(
 List(x,t->t^q));
end;;
```

The Suzuki group Sz(q) has two point orbits in PG$(3, q)$, of size respectively $q^3 + q$ and $q^2 + 1$. The latter is a Suzuki–Tits ovoid, say $\mathcal{O} =$ `Ov`.

```
Sg:=SuzukiGroup(IsMatrixGroup,q);
Or:=Orbits(Sg,PG3,OnLines);
Ov:=Filtered(Or,x->Size(x)=q^2+1)[1];
Ovp:=Set(Ov,x->Position(PG3,x));
```

We wrote `Ovp` for the set of all points of `Ov` in the permutation representation.

The following code is used to write the set $\Lambda =$ `AllLines` consisting of all the lines of PG$(3, q)$. Since the full projective general linear group PGL$(4, q)$ is transitive on this set, we may just consider the orbit of
$$\ell_0 = \big\langle (0, 1, 0, 0), (1, 0, 0, 0) \big\rangle$$
under its action.

The group PGL$(4, q)$ has to be written as the action `Pgrp` of GL$(4, q)$ on normalised vectors. The line orbit is obtained considering the action of this group `Pgrp` on the set of points, in the permutation representation, of a given line.

```
  L1:=LineAB([1,0,0,0]*Z(q)^0,[0,1,0,0]*Z(q)^0);
  L1p:=Set(L1,x->Position(PG3,x));
  Pgrp:=Action(GL(4,q),PG3,OnLines);
  AllLines:=Orbit(Pgrp,L1p,OnSets);
```

Remark 8. There might be more efficient ways to obtain the set Λ as union of line-orbits under the action of a Singer cycle Θ of PG$(3, q)$. In fact, see [9], the number to these line-orbits is exactly $q+1$ and a starter set for these (that is a set of representatives for each of them) is given by all the lines passing through a fixed point $P \notin \mathcal{O}^+$ tangent to the elliptic quadric $\mathcal{O}^+$ stabilised by the subgroup of order q^2+1 of Θ.

We are now in position to write the set $T\mathcal{O} =$ `TangentComplex` of all lines tangent to the ovoid `Ov`. This is simply done by enumerating the lines of $\mathrm{PG}(3,q)$ which meet $\mathcal{O}$ in just 1 point.

The function `TCpx` is used to partition the elements of this set according to their tangency point to $\mathcal{O}$.

```
TangentComplex:=
   Set(Filtered(AllLines,
               x->Size(Intersection(Ovp,x))=1),
       x->Set(x));

TCpx:=function(TC,O)
 return List(O,x->Filtered(TC,v->x in v));
end;;
```

As seen in Section 3, given three mutually skew lines ℓ_1, ℓ_2, ℓ_3, it is easy to write the opposite regulus $\mathcal{R}^o$ they induce. The regulus $\mathcal{R}$ containing L is then obtained as $(\mathcal{R}^o)^o$.

```
#Functions to build up a
# regulus
# Here we use a permutation
# representation
OpRegulus:=function(a,b,c)
 return Filtered(AllLines,x->not(
   IsEmpty(Intersection(a,x)) or
   IsEmpty(Intersection(b,x)) or
   IsEmpty(Intersection(c,x))));
end;;

Regulus:=function(a,b,c)
 local l;
 l:=OpRegulus(a,b,c);
 return OpRegulus(l[1],l[2],l[3]);
end;;

#This function uses a normalised # vector representation
RegLines:=function(L)
 local Lp,Rp;
 Lp:=Set(L,x->Set(x,t->Position(PG3,t)));
 Rp:=Regulus(Lp[1],Lp[2],Lp[3]);
```

```
 return Set(Rp,
           x->Set(x,t->PG3[t]));
end;;
```

To construct a regular spread $\mathcal{S}$ we use the following functions:

1. `LookForSpread0` which, given 4 lines $\ell_1, \ell_2, \ell_3, \ell_4$, builds the set R of all lines in reguli of the form $\mathcal{R}(\ell_1, x, \ell_4)$ where $x \in \mathcal{R}(\ell_1, \ell_2, \ell_3) \setminus \{\ell_1\}$;
2. `RClosure` which determines $q^2 + 1$ lines in the *regular closure* of a set of lines R;
3. `LookForSpread1`, `LookForSpread2` and `LookForSpread` which build the requested regular spread of tangent lines to an ovoid `O`.

```
# L = Set of 4 lines
LookForSpread0:=function(L)
 local Reg,RegT,x,Spr;
 Spr:=[];
 Reg:=Regulus(L[1],L[2],L[3]);
  for x in Difference(Reg,[L[1]]) do
   RegT:=Regulus(L[1],x,L[4]);
   Spr:=Union(Spr,RegT);
  od;
 return Spr;
end;;

RClosure0:=function(S)
 local x,X,R,V;
 X:=Combinations(S,3);
 R:=ShallowCopy(S);
 for x in X do
  R:=Union(R,Regulus(x[1],x[2],x[3]));
  if Size(R)=q^2+1 then return R;
  fi;
 od;
 return R;
end;;

RClosure:=function(S)
 local f,T;
 f:=false;
 T:=RClosure0(S);
 if not(T=S) then
  Print(Size(T),"-",Size(S),"\n");
  return RClosure(T);
 else
```

```
  Print("Closed\n");
  return T;
 fi;
end;;

# Hint for regulus
LookForSpread1:=function(TC,x,O)
local Tp,Ct,y,R1,S2,TC2;
  R1:=Regulus(x[1],x[2],x[3]);
  if not(IsSubset(TC,R1)) then return fail; fi;
  TC2:=Filtered(TC,x->IsEmpty(Intersection(x,Union(R1))));
  for y in TC2 do
   Print(".\n");
   S2:=LookForSpread0([x[1],x[2],x[3],y]);
   if IsSubset(TC,S2) then return (S2); fi;
  od;
  return fail;
end;;

LookForSpread2:=function(TC,O)
 local Tp,Ct,x,R;
 Tp:=Set(TCpx(TC,O),x->Set(x));
#First regulus
 Ct:=Filtered(Cartesian(Tp{[1..3]}),
              t->IsEmpty(Intersection(t[1],t[2])) and
                 IsEmpty(Intersection(t[1],t[3])) and
                 IsEmpty(Intersection(t[2],t[3])));
#Look for a second (compatible) regulus
 for x in Ct do
  R:=LookForSpread1(TC,x,O);
  if IsList(R) then return R; fi;
 od;
 return fail;
end;;

LookForSpread:=function(TC,O)
 local T;
 T:=RClosure(LookForSpread2(TC,O));
 if IsSubset(TC,T) then return T; fi;
 return fail;
end;;
```

To check if any given spread is regular, we verify that it contains the regulus spanned by any three of its elements.

```
#Check if a spread is regular
IsRegularS:=function(S)
 local x,X,r;
 X:=Combinations(S,3);
 while Size(X)>2 do
 x:=X[1];
 r:=Regulus(x[1],x[2],x[3]);
  if not(IsSubset(S,r)) then
   Print(Size(Intersection(S,r)),"\n");
   return false;
  else
   X:=Difference(X,Combinations(r,3));
   Print(Size(X),"\n");
  fi;
 od;
 return true;
end;;
```

Our next step in constructing a model of PG$(2, q^2)$ is to embed PG$(3, q)$ in PG$(4, q)$ as hyperplane at infinity, as seen in Section 3. The function `EmbedPG3` does just this; `EmbedSpr` is a utility function to embed sets of points of PG$(3, q)$ in PG$(4, q)$ and it is most useful for spreads.

```
# Embed PG(3,q) in PG(4,q) as
# hyperplane at infinity
EmbedPG3:=function(L)
 return Set(L,x->Concatenation([0*Z(q)],x));
end;;

EmbedSpr:=function(L)
 return Set(L,x->EmbedPG3(x));
end;;
```

Suppose now `Spr` to be a regular spread of tangent lines to `Ov`. We shall determine a linear transformation μ of PG$(3, q)$ such that the spread μ(`Spr`) contains the lines ℓ_1, ℓ_2, ℓ_3 of (1). Recall that, for any spread $\mathcal{S}$ of PG$(3, q)$, there exists a line $L_{\mathcal{S}}$ of PG$(3, q^2) \setminus$ PG$(3, q)$ such that

$$\mathcal{S} = \{PP^q : P \in L_{\mathcal{S}}\}.$$

Clearly, the spread $\mathcal{S}$ is uniquely determined by the line $L_{\mathcal{S}}$, although different lines might be associated to the same spread. The following function, `LookForLine`, computes one of these lines.

```
LookForLine:=function(spr)
 local PSpr,xSpr,LLa,x, y, fl,xq;
```

```
 PSpr:=List(spr,x->LineAB2(PG3[x[1]],PG3[x[2]]));
 xSpr:=List(PSpr,x->Difference(x,PG3));
 LLa:=List(Cartesian(PSpr[1],PSpr[2]),x->LineAB2(x[1],x[2]));;
 for x in LLa do
  Print("x=",x[1],",",x[2],"\n");
#The lines should be disjoint from PG(3,q)
  if not(IsEmpty(Intersection(x,PG3))) then
   Print("!\n");
   continue;
  fi;
#They should also meet any component of the spread
  fl:=true;
  for y in xSpr do
   if IsEmpty(Intersection(y,x)) then
    Print("%");
    fl:=false;
    break;
   fi;
   Print(".");
  od;
  if not(fl) then continue; fi;
# The conjugate line
# should also meet any component of the spread
  xq:=Set(x,t->Conj(t));
  for y in xSpr do
   if IsEmpty(Intersection(y,xq)) then
    fl:=false;
    Print("^");
    break;
   fi;
  Print(",");
  od;
#If this is the case, then we have found
# what we were looking for
  if fl then return x; fi;
 od;
#Bad luck here.
return fail;
end;;
```

Denote now by `LCanon` the line of $\mathrm{PG}(3, q^2)$ associated with a spread, say `SCanon`, containing ℓ_1, ℓ_2, ℓ_3.

```
 GCanon:=[
 LineAB([1,0,0,1]*Z(q)^0,[0,1,1,0]*Z(q)^0),
 LineAB([1,0,0,0]*Z(q)^0,[0,1,0,0]*Z(q)^0),
 LineAB([0,0,1,0]*Z(q)^0,[0,0,0,1]*Z(q)^0)];
 GCanonP:=Set(GCanon,
     x->Set(x,t->Position(PG3,t)));
 RCanon:=Regulus(GCanonP[1],GCanonP[2],GCanonP[3]);
# look for a fourth line to generate the spread
 Get4th:=function(R)
  local j,L4;
  j:=1;
  repeat
   L4:=AllLines[j];
   j:=j+1;
   until IsEmpty(Intersection(L4,Union(R)));
  return L4;
 end;;
 L4:=Get4th(RCanon);
 SCanon:=RClosure(Union(GCanonP,[L4]));
 LCanon:=LookForLine(SCanon);
```

It is now actually possible to write a matrix in GL$(4, q)$ inducing a collineation μ in PG$(3, q)$ which maps the general spread `Spr` into `SCanon`.

```
 SprToCanon:=function(Spr)
  local Lx,M0,N0;
  Lx:=LookForLine(Spr);
  M0:=TransposedMat([Lx[1],Conj(Lx[1]),Lx[2],Conj(Lx[2])]);
  N0:=TransposedMat([LCanon[1],Conj(LCanon[1]),
                     LCanon[2],Conj(LCanon[2])]);
  return N0*M0^(-1);
 end;;
```

Let then `M` = `SprToCanon(Spr)` and suppose `SprT` = μ(`Spr`) and `OvT` = μ(`Ov`).

```
# New spread
SprT:=Set(Spr,x->Set(x,t->NormedRowVector(M*t)));
#Consider also the image of the ovoid under the
# collineation induced by M
OvT:=Set(Ov,x->NormedRowVector(M*x));
```

It is still necessary to determine the parameter ε in the correspondence (2).

```
PG4ToPG2:=function(P,eps)
 return NormedRowVector([P[1],P[2]+eps*P[3],eps*P[4]+P[5]]);
end;;
```

We may proceed as follows.

```
LookForEps:=function(Spr)
 local t,r,sp1,L,R1;
 L:=
 [LineAB([1,0,0,1],[0,1,1,0]),
  LineAB([1,0,0,0],[0,1,0,0]),
  LineAB([0,0,1,0],[0,0,0,1])]*Z(q)^0;
 R1:=RegLines(L);
 sp1:=Difference(Spr,R1);
 t:=sp1[1];
 r:=Filtered(
 Difference(Elements(GF(q^2)),Elements(GF(q))),
   eps->
   (t[1][1]+t[1][2]*eps)/(t[1][3]*eps+t[1][4])=
   (t[2][1]+t[2][2]*eps)/(t[2][3]*eps+t[2][4]));
 return r;
end;;

eps:=LookForEps(SprT)[1];
```

We are now in position to use the construction of [16] in order to obtain a maximal arc. We first embed $\mathrm{PG}(3,q)$ in $\mathrm{PG}(4,q)$ as the hyperplane at infinity; `EOvT` is the image under this embedding of the transformed ovoid (under the collineation given by μ); then, we compute the *affine* cone `FullCone2` with vertex

$$\texttt{Vtx} = (1,0,0,0,0)$$

and basis `EOvT`. The image of this cone under $\theta =$ `PG4ToPG2` is the maximal arc `Arc` of $\mathrm{PG}(2,q^2)$.

```
# Embed OvT\subseteq PG(3,q) in PG(4,q)
EOvT:=EmbedPG3(OvT);
# ... and build the full cone in AG(4,q)
# with vertex
Vtx:=[1,0,0,0,0]*Z(q)^0;
# and basis OvT
FullCone:=Difference(Union(Set(EOvT,x->LineAB(x,Vtx))),EOvT);
# The requested maximal arc is the image of
# the cone under the map PG4ToPG2
Arc:=Set(FullCone,x->PG4ToPG2(x,eps));
```

The following procedure checks whether a set `X` is actually an arc of degree q. In particular, the function `CheckSecants`, verifies that all of the secants of `X` meet `X` in exactly q points. The function `CheckArc` checks also that there is no tangent line at any point of `X`.

```
# Check if a set X is an arc
# step 0:
#  verify if all secants meet X in
#  q points
 CheckSecants0:=function(X)
 local C,l,XX;
 C:=Combinations(X,2);
 XX:=[];
 while(not(IsEmpty(C))) do
  l:=LineAB2(C[1][1],C[1][2]);
  if not(Size(Intersection(l,X))=q) then
   Print(Size(Intersection(l,X)),"\n");
   return [false,[]];
  fi;
  C:=Difference(C,Combinations(Intersection(l,X),2));
  Print("!",Size(C),"!\n");
  Add(XX,l);
 od;
 return [true,XX];
end;;

CheckSecants:=function(X)
 return (CheckSecants0(X)[1]);
end;;

CheckArc:=function(X)
 local C,l,XX,x;
 C:=Combinations(X,2);
#Computes all the secants;
 XX:=CheckSecants0(X);
 if not(XX[1]) then return false; fi;
 for x in X do
  l:=Filtered(XX[2],t->x in t);
  if Size(l)<q^2+1 then return false; fi;
 od;
 return true;
end;;
```

We are now ready to compute a minimum degree curve covering the arc $\mathcal{K} = \texttt{Arc}$. The following is an outline of the procedure.

1. Determine all monic monomials in two variables of degree at most i over $\mathrm{GF}(q^2)$. This is done by the function `AllMon`.
2. A polynomial
$$f(x, y) = \sum_{i,j} c_{ij} x^i y^j$$
corresponds to a curve covering $\mathcal{A}$ if, and only if, the coefficients c_{ij} are a solution of the homogeneous linear system given by
$$\sum_{i,j} c_{ij} p_x^i p_y^j = 0\,, \quad P = (1, p_x, p_y) \in \mathcal{A}\,; \tag{3}$$
3. The function `BuildMat`, for a list of points $\mathcal{K}$ and a maximum degree `i` generates the matrix whose rows are exactly the evaluations of the monomials in `AllMon(i)`, computed on the second and third coordinate of any point in $\mathcal{K}$. In other words, if
$$\texttt{AllMon(i)} = \{f_1(x, y), f_2(x, y), \ldots, f_k(x, y)\}$$
and $P = (1, p_x, p_y) \in \mathcal{K}$, then the row of `BuildMat(K, i)` corresponding to P would be
$$[f_1(p_x, p_y), f_2(p_x, p_y), \ldots, f_k(p_x, p_y)]\,.$$
4. If `BuildMat(K, i)` has full rank, then the only polynomial of degree at most `i` in x, y giving a curve which contains all points of $\mathcal{K}$ is the zero-polynomial.

```
RR:=PolynomialRing(GF(q^2),["x","y"]);
AllMon:=function(i)
 local l;
 l:=Filtered(Cartesian([0..i],[0..i]),t->t[1]+t[2]<i+1);
 return List(l,t->RR.1^t[1]*RR.2^t[2]);
end;;

BuildMat:=function(K,i)
 local m;
 m:=AllMon(i);
 return List(K,x->
        List(m,t->Value(t,[RR.1,RR.2],[x[2],x[3]])));
end;;
```

The minimum index `i` such that `Buildmat(Arc, i)` has not full rank has to be determined. The following function takes as parameters the arc `K` and a maximum degree to test. Observe that

$$\xi(\texttt{i}) = \mathrm{rank}(\texttt{BuildMat(K, i)}) - |\texttt{AllMon(i)}|$$

is non-increasing in `i`. Hence, to look for i, we may use an iterative approach: consider an initial interval to test $[a \ldots b]$, let $c = \lfloor \frac{a+b}{2} \rfloor$ and compute $\xi(c)$. If

$\xi(c) = 0$, then the first value `i` such that $\xi(\mathtt{i}) < 0$ may possibly be found in $[(c+1)\dots b]$; on the other hand, if $\xi(c) < 0$, such `i` is to be found in $[a\dots c]$. We keep bisecting the interval till it contains just one value c'. If $\xi(c') < 0$, then $\mathtt{i} = c'$ is returned; otherwise the algorithm fails.

```
GetIndex:=function(A,mi)
 local tidx,c,d,r;
 tidx:=[1..mi];
 while(Size(tidx)>1) do
  c:=Int((tidx[1]+tidx[Size(tidx)])/2);
  d:=BuildMat(A,c);
  r:=Rank(d);
  Print("c=",c," t=",tidx,"\n");
  Print("r=",r," s=",Size(d[1]),"\n");
  if r=Size(d[1]) then
   tidx:=[(c+1)..tidx[Size(tidx)]];
  else
   tidx:=[tidx[1]..c];
  fi;
 od;
 Print(tidx,"\n");
 c:=tidx[1];
 d:=BuildMat(A,c);
 r:=Rank(d);
 if not(r=Size(d[1])) then
   return c;
 else
   return fail;
 fi;
end;;
```

Remark that the affine curve of equation

$$(x^{q^2} - x) = 0$$

has degree q^2 and passes through all the points of the affine plane $AG(2, q^2)$ (hence, also through all those of $\mathcal{K}$). Thus, this value may be chosen as the maximum degree i to test in `GetIndex`.

```
i:=GetIndex(Arc,q^2);
```

The coefficients of the polynomial giving the curve may now be obtained by solving a linear system of equations.

```
MatOk:=BuildMat(Arc,i);;
SolV:=NullspaceMat(TransposedMat(MatOk))[1];
```

The values in SolV are now used to write the equation of the curve. This is done by the function VecToPoly.

```
VecToPoly:=function(v,i)
 local m;
 m:=AllMon(i);
 return Sum(List([1..Size(v)],x->m[x]*v[x]));
end;;

pp:=VecToPoly(SolV,i);
```

Remark 9. When $q = 8$, the construction of [16] gives a $(456, 8)$-maximal arc $\mathcal{K}$ of $\mathrm{PG}(2, 64)$. If the ovoid $\mathcal{O}$ chosen for this construction is an elliptic quadric, then the minimum degree of a curve Γ containing all the points of $\mathcal{K}$ is 7 and this curve splits into 3 conics and a line. On the other hand, if the Suzuki–Tits ovoid is chosen, then the minimum degree of such a curve Γ is 22 and it splits into an irreducible curve of degree 17, and 5 lines.

References

[1] J. André; *Über nicht-Desarguesche Eben mit transitiviter Translationsgruppe*, Math. Z. **60** (1954), 156–186.

[2] A. Aguglia, L. Giuzzi, G. Korchmàros; *Algebraic curves and maximal arcs*, J. Algebr. Comb., to appear.

[3] S. Ball, A. Blokhuis, F. Mazzocca; *Maximal arcs in Desarguesian planes of odd order do not exist*, Combinatorica **17** (1997), no. 1, 31–41.

[4] S. Ball, A. Blokhuis; *An easier proof of the maximal arcs conjecture*, Proc. Amer. Math. Soc. **126** (1998), 3377–3380.

[5] R. H. Bruck, R. C. Bose; *The construction of translation planes from projective spaces*, J. Algebra **1** (1964), 85–102.

[6] R. H. Bruck, R. C. Bose; *Linear representations of projective planes in projective spaces*, J. Algebra **4** (1966), 117–172.

[7] R. H. .F. Denniston; *Some maximal arcs in finite projective planes*, J. Combin. Theory Ser. A. **6** (1969), 317–319.

[8] The GAP Group; *GAP – Groups, Algorithms, and Programming, Version* 4.4; 2006, (http://www.gap-system.org).

[9] L. Giuzzi, A. Sonnino; *LDPC codes from Singer cycles*, preprint (arxiv 0709.2813).

[10] N. Hamilton, R. Mathon; *More maximal arcs in Desarguesian projective planes and their geometric structure*, Adv. Geometry **3** (2003), 251–261.

[11] N. Hamilton, T. Penttila; *Groups of maximal arcs*, J. Combin. Theory Ser. A **94** (2001), 63–86.

[12] J. W. P. Hirschfeld; *Finite projective spaces of three dimensions*, Oxford University Press (1985).

[13] J. W. P. Hirschfeld, J. A. Thas; *General Galois Geometries*, Oxford Univ. Press, London, (1992).

[14] N. L. Johnson; *Flocks of hyperbolic quadrics and translation planes admitting affine homologies*, J. Geom. **34** (1989), 50–73.

[15] R. Mathon; *New maximal arcs in Desarguesian planes*, J. Combin. Theory Series A **97** (2002), 353–368.

[16] J. Thas; *Construction of maximal arcs and partial geometries*, Geom. Dedicata **3** (1974), 61–64.

[17] J. Thas; *Construction of maximal arcs and dual ovals in translation planes*, Europ. J. Combin **1** (1980), 189–192.

Angela Aguglia and Luca Giuzzi
Dipartimento di Matematica
Politecnico di Bari
Via Orabona 4
I-70125 Bari
Italy
e-mail: `aguglia@poliba.it`
`giuzzi@poliba.it`

Received: January 30, 2007.
Revised: August 6, 2007.