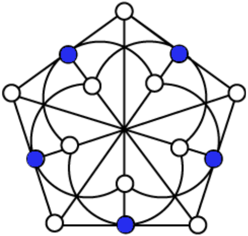
Desargues

… a finite geometry package

John Bamberg

Anton Betten Jan De Beule Maska Law

Max Neunhöffer Michael Pauley Sven Reichard



```
gap> geo := ParabolicQuadric(4, 2);
Q(4, 2)
gap> g := CollineationGroup( geo );
PGammaO(5,2)
gap> points := Points( geo );
<points of Q(4, 2)>
gap> enum := Enumerator( points );
EnumeratorOfVarieties( <points of Q(4, 2)> )
gap> x := enum[ 1];
<a point in Q(4, 2)>
gap> lx := ResidualOfVariety( geo, x, 2 );
<residual lines in Q(4, 2)>
gap> stabx := Stabilizer( g, x, OnLieVarieties
<projective group with Frobenius of size 720>
gap> IsTransitive(stabx, lx, OnLieVarieties );
true
```

Outline

Finite Geometry

- Projective Geometry/Affine Geometry

Finite Geometry

- Projective Geometry/Affine Geometry
- Polar Spaces

Finite Geometry

- Projective Geometry/Affine Geometry
- Polar Spaces
- Generalised Polygons

Finite Geometry

- Projective Geometry/Affine Geometry
- Polar Spaces
- Generalised Polygons

Incidence Geometry

Consists of:

- Objects (points, lines, planes, etc)
- Incidence relation (anti-reflexive and symmetric)
- A maximal flag contains an object of each type.

The *rank* is the number of types of object.

Projective Space

Start with a vector space $V(d, \mathbb{GF}(q))$

- Objects

  Points: 1-dim subspaces
  Lines: 2-dim subspaces
  Planes: 3-dim subspaces, etc...

Projective Space

Start with a vector space $V(d, \mathbb{GF}(q))$

- Objects

  Points: 1-dim subspaces
  Lines: 2-dim subspaces
  Planes: 3-dim subspaces, etc...

- Incidence Relation: $A \subset B$ or $B \subset A$

Projective Space

Start with a vector space $V(d, \mathbb{GF}(q))$

- Objects

  Points: 1-dim subspaces
  Lines: 2-dim subspaces
  Planes: 3-dim subspaces, etc...

- Incidence Relation: $A \subset B$ or $B \subset A$
- Types $1, 2, 3, \ldots, d-1$

Projective Space

Start with a vector space $V(d, \mathbb{GF}(q))$

- Objects

  Points: 1-dim subspaces
  Lines: 2-dim subspaces
  Planes: 3-dim subspaces, etc...

- Incidence Relation: $A \subset B$ or $B \subset A$
- Types $1, 2, 3, \ldots, d-1$

Theorem

A projective space or polar space of rank at least 3 is classical, that is,

it comes from a vector space.

Spreads of W$(5, q)$

W$(5, q)$

Consider $V(6, q)$ equipped with an alternating form:

$$\langle u, v \rangle = u_1 v_2 - u_2 v_1 + u_3 v_4 - u_4 v_3 + u_5 v_6 - u_6 v_5.$$

We get a polar geometry consisting of

Spreads of W(5, q)

W(5, q)

Consider $V(6, q)$ equipped with an alternating form:

$$\langle u, v\rangle = u_1v_2 - u_2v_1 + u_3v_4 - u_4v_3 + u_5v_6 - u_6v_5.$$

We get a polar geometry consisting of

Points: All one-dim subspaces.

Lines: $(q^2+1)\frac{q^6-1}{q-1}$ two-dim subspaces.

Planes: $(q^3+1)\frac{q^4-1}{q-1}$ three-dim subspaces.

Spreads of W(5, q)

W(5, q)

Consider $V(6, q)$ equipped with an alternating form:

$$\langle u, v\rangle = u_1 v_2 - u_2 v_1 + u_3 v_4 - u_4 v_3 + u_5 v_6 - u_6 v_5.$$

We get a polar geometry consisting of

- Points: All one-dim subspaces.
- Lines: $(q^2+1)\frac{q^6-1}{q-1}$ two-dim subspaces.
- Planes: $(q^3+1)\frac{q^4-1}{q-1}$ three-dim subspaces.

Spreads

A spread of W(5, q) is a set of q^3+1 planes which form a partition of the set of points.

Problem

Find all spreads of $W(5, 3)$ which have automorphism group of order divisible by 13.

Problem

Find all spreads of $W(5, 3)$ which have automorphism group of order divisible by 13.

What we need...

Points: Easy, all one-dim subspaces of $\mathbb{GF}(3)^6$.

Problem

Find all spreads of $W(5,3)$ which have automorphism group of order divisible by 13.

What we need...

Points: Easy, all one-dim subspaces of $\mathbb{GF}(3)^6$.

Planes: Take orbit of one plane

Problem

Find all spreads of $W(5,3)$ which have automorphism group of order divisible by 13.

What we need...

Points: Easy, all one-dim subspaces of $\mathbb{GF}(3)^6$.

Planes: Take orbit of one plane

```
gap> sp := Sp(6,3);;
gap> plane := [[1,0,0,1,0,0], [0,1,0,0,1,0],[0,0,1,0,0,1]]*Z(3)^0;;
gap> planes := Orbit(sp, plane, OnSubspacesByCanonicalBasis);;
```

Problem

Find all spreads of $W(5,3)$ which have automorphism group of order divisible by 13.

What we need...

Points: Easy, all one-dim subspaces of $\mathbb{GF}(3)^6$.

Planes: Take orbit of one plane

```
gap> sp := Sp(6,3);;
gap> plane := [[1,0,0,1,0,0], [0,1,0,0,1,0],[0,0,1,0,0,1]]*Z(3)^0;;
gap> planes := Orbit(sp, plane, OnSubspacesByCanonicalBasis);;
```

Group: Sylow 13-subgroup of $Sp(6,3)$

```
gap> syl13 := SylowSubgroup(sp, 13);
<group of 6x6 matrices of size 13 in characteristic 3>
```

Solution: Stitch together orbits on planes

In DESARGUES...

```
gap> w := SymplecticSpace(5, 3);
W(5, 3)
gap> sp := IsometryGroup( w );
PSp(6,3)
```

In Desargues...

```
gap> w := SymplecticSpace(5, 3);
W(5, 3)
gap> sp := IsometryGroup( w );
PSp(6,3)
gap> syl := SylowSubgroup(sp, 13);
<projective group with Frobenius of size 13>
gap> planes := Planes( w );
<planes of W(5, 3)>
gap> planes := AsList( planes );;
```

In DESARGUES...

```
gap> w := SymplecticSpace(5, 3);
W(5, 3)
gap> sp := IsometryGroup( w );
PSp(6,3)
gap> syl := SylowSubgroup(sp, 13);
<projective group with Frobenius of size 13>
gap> planes := Planes( w );
<planes of W(5, 3)>
gap> planes := AsList( planes );;
gap> orbits := Orbits(syl, planes , OnLieVarieties);;
gap> Collected( List( orbits, Size ));
[ [ 1, 2 ], [ 13, 86 ] ]
```

In Desargues...

```
gap> w := SymplecticSpace(5, 3);
W(5, 3)
gap> sp := IsometryGroup( w );
PSp(6,3)
gap> syl := SylowSubgroup(sp, 13);
<projective group with Frobenius of size 13>
gap> planes := Planes( w );
<planes of W(5, 3)>
gap> planes := AsList( planes );;
gap> orbits := Orbits(syl, planes , OnLieVarieties);;
gap> Collected( List( orbits, Size ));
[ [ 1, 2 ], [ 13, 86 ] ]
gap> IsPartialSpread := s ->  ForAll( Combinations(s,2), c ->
            ProjectiveDimension( Meet(c[1], c[2]) ) = -1 );;
gap> partialspreads := Filtered(orbits, IsPartialSpread);;
```

In Desargues...

```
gap> w := SymplecticSpace(5, 3);
W(5, 3)
gap> sp := IsometryGroup( w );
PSp(6,3)
gap> syl := SylowSubgroup(sp, 13);
<projective group with Frobenius of size 13>
gap> planes := Planes( w );
<planes of W(5, 3)>
gap> planes := AsList( planes );;
gap> orbits := Orbits(syl, planes , OnLieVarieties);;
gap> Collected( List( orbits, Size ));
[ [ 1, 2 ], [ 13, 86 ] ]
gap> IsPartialSpread := s ->  ForAll( Combinations(s,2), c ->
            ProjectiveDimension( Meet(c[1], c[2]) ) = -1 );;
gap> partialspreads := Filtered(orbits, IsPartialSpread);;
gap> 13s := Filtered(partialspreads, i -> Size(i) = 13);;
gap> 26s := List(Combinations(13s,2), Union);;
gap> two := Union(Filtered(partialspreads, i -> Size(i) = 1));;
gap> 28s := List(26s, x -> Union(x, two) );;
```

In DESARGUES...

```
gap> w := SymplecticSpace(5, 3);
W(5, 3)
gap> sp := IsometryGroup( w );
PSp(6,3)
gap> syl := SylowSubgroup(sp, 13);
<projective group with Frobenius of size 13>
gap> planes := Planes( w );
<planes of W(5, 3)>
gap> planes := AsList( planes );;
gap> orbits := Orbits(syl, planes , OnLieVarieties);;
gap> Collected( List( orbits, Size ));
[ [ 1, 2 ], [ 13, 86 ] ]
gap> IsPartialSpread := s ->  ForAll( Combinations(s,2), c ->
            ProjectiveDimension( Meet(c[1], c[2]) ) = -1 );;
gap> partialspreads := Filtered(orbits, IsPartialSpread);;
gap> 13s := Filtered(partialspreads, i -> Size(i) = 13);;
gap> 26s := List(Combinations(13s,2), Union);;
gap> two := Union(Filtered(partialspreads, i -> Size(i) = 1));;
gap> 28s := List(26s, x -> Union(x, two) );;
gap> spreads := Filtered( 28s, IsPartialSpread);;
gap> Size(spreads);
5
```

These five spreads of $W(5, 3)$

1. $2\times$ Albert semifield-spread: $3^3 \cdot 13 : 3$
2. $2\times$ Hering spread: $\mathrm{PSL}(2, 13)$
3. $1\times$ Regular spread: $\mathrm{PSL}(2, 27)$

Features of DESARGUES, at the moment...

- Construction of geometries
  - `ProjectiveSpace`(d, q)
  - `EllipticQuadric`(d, q)
  - `HermitianVariety`(d, q^2)
  - `AffineSpace`(d, q)
  - `SplitCayleyHexagon`(q)

Features of DESARGUES, at the moment...

- Construction of geometries
  - `ProjectiveSpace(` d, q)
  - `EllipticQuadric(` d, q)
  - `HermitianVariety(` d, q^2)
  - `AffineSpace(` d, q)
  - `SplitCayleyHexagon(` q)
- Basic functionality
  - `ResidualOfVariety(` geometry, object, type)
  - `ResidualOfFlag(` geometry, flag, type)
  - `Varieties(` geometry, type)
  - `Join(` object, object)
  - `Meet(` object, object)
  - `VectorSpaceToVariety(` geometry, vector or matrix)

Features of DESARGUES, at the moment...

- Construction of geometries
  - `ProjectiveSpace`(d, q)
  - `EllipticQuadric`(d, q)
  - `HermitianVariety`(d, q^2)
  - `AffineSpace`(d, q)
  - `SplitCayleyHexagon`(q)
- Basic functionality
  - `ResidualOfVariety`(geometry, object, type)
  - `ResidualOfFlag`(geometry, flag, type)
  - `Varieties`(geometry, type)
  - `Join`(object, object)
  - `Meet`(object, object)
  - `VectorSpaceToVariety`(geometry, vector or matrix)
- Flexibility with polar spaces: `PolarSpace`(form)

- Generalised polygons
  - `TwistedTrialityHexagon( q )`
  - `EGQByKantorFamily( group, list1, list2 )`
  - `EGQByqClan( q-clan, field )`
  - `BLTSetByqClan( q-clan, field )`

- Generalised polygons
  - `TwistedTrialityHexagon(` q)
  - `EGQByKantorFamily(` group, list1, list2)
  - `EGQByqClan(` q-clan, field)
  - `BLTSetByqClan(` q-clan, field)
- Group actions
  - `OnLieVarieties`
  - `OnAffineVarieties`
  - `OnKantorFamily`

- Generalised polygons
  - `TwistedTrialityHexagon`(q)
  - `EGQByKantorFamily`(group, list1, list2)
  - `EGQByqClan`(q-clan, field)
  - `BLTSetByqClan`(q-clan, field)
- Group actions
  - `OnLieVarieties`
  - `OnAffineVarieties`
  - `OnKantorFamily`
- Enumerators

- Generalised polygons
  - `TwistedTrialityHexagon`(q)
  - `EGQByKantorFamily`(group, list1, list2)
  - `EGQByqClan`(q-clan, field)
  - `BLTSetByqClan`(q-clan, field)
- Group actions
  - `OnLieVarieties`
  - `OnAffineVarieties`
  - `OnKantorFamily`
- Enumerators
- Morphisms
  - `NaturalEmbeddingByVariety`
  - `NaturalProjectionByVariety`
  - `KleinCorrespondence`
  - `ProjectiveCompletion`

Another example

The Patterson ovoid of $Q(6,3)$

- $Q(6,3)$: points of $\mathbb{GF}(3)^7$ which are solutions of

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 + x_7^2 = 0.$$

Another example

The Patterson ovoid of $Q(6,3)$

- $Q(6,3)$: points of $\mathbb{GF}(3)^7$ which are solutions of

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 + x_7^2 = 0.$$

- This polar space also contains lines and planes.

Another example

The Patterson ovoid of $Q(6,3)$

- $Q(6,3)$: points of $\mathbb{GF}(3)^7$ which are solutions of

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 + x_7^2 = 0.$$

- This polar space also contains lines and planes.
- Ovoid: set of 28 points of $Q(6,3)$ which partition the planes of $Q(6,3)$.

Another example

The Patterson ovoid of Q(6, 3)

- Q(6, 3): points of $\mathbb{GF}(3)^7$ which are solutions of

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 + x_7^2 = 0.$$

- This polar space also contains lines and planes.
- Ovoid: set of 28 points of Q(6, 3) which partition the planes of Q(6, 3).
- Patterson: Unique up to projectivity.

Another example

The Patterson ovoid of $Q(6,3)$

- $Q(6,3)$: points of $\mathbb{GF}(3)^7$ which are solutions of

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 + x_7^2 = 0.$$

- This polar space also contains lines and planes.
- Ovoid: set of 28 points of $Q(6,3)$ which partition the planes of $Q(6,3)$.
- Patterson: Unique up to projectivity.
- We will use E. E. Shult's beautiful construction.

Construct specific polar space

```
gap> id := IdentityMat(7, GF(3));;
gap> form := QuadraticFormByMatrix(id, GF(3));
< quadratic form >
gap> ps := PolarSpace( form );
<polar space of dimension 6 over GF(3)>
```

Construct specific polar space

```
gap> id := IdentityMat(7, GF(3));;
gap> form := QuadraticFormByMatrix(id, GF(3));
< quadratic form >
gap> ps := PolarSpace( form );
<polar space of dimension 6 over GF(3)>
```

Construct ovoid

```
gap> psl32 := PSL(3,2);
Group([ (4,6)(5,7), (1,2,4)(3,6,5) ])
gap> reps := [[1,1,1,0,0,0,0], [-1,1,1,0,0,0,0],
              [1,-1,1,0,0,0,0], [1,1,-1,0,0,0,0]]*Z(3)^0;;
gap> ovoid := Union( List(reps, x->
                Orbit(psl32, x, Permuted)) );;
gap> ovoid := List(ovoid, x ->
                VectorSpaceToVariety(ps, x));;
```

Look at it in the canonical polar space

```
gap> pq := ParabolicQuadric(6, 3);
Q(6, 3)
gap> iso := IsomorphismPolarSpaces(ps, pq);
<geometry morphism from <polar space of dimension 6 over GF(3)>
to Q(6, 3)>
gap> ovoid2 := ImagesSet(iso, ovoid);
[ <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)> ]
```

Look at it in the canonical polar space

```
gap> pq := ParabolicQuadric(6, 3);
Q(6, 3)
gap> iso := IsomorphismPolarSpaces(ps, pq);
<geometry morphism from <polar space of dimension 6 over GF(3)>
to Q(6, 3)>
gap> ovoid2 := ImagesSet(iso, ovoid);
[ <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)>, <a point in Q(6, 3)>, <a point in Q(6, 3)>,
  <a point in Q(6, 3)> ]
```

Check that it is an ovoid

```
gap> planes := AsList( Planes(pq) );;
gap> ForAll(planes, p -> Number(ovoid2, x -> x in p) = 1);
true
```

Find the stabiliser

```
gap> g := CollineationGroup( pq );
PGammaO(7,3)
gap> points := AsList( Points(pq) );;
gap> hom := ActionHomomorphism(g, points, OnLieVarieties);
<action homomorphism>
gap> omega := HomeEnumerator( UnderlyingExternalSet(hom) );;
gap> imgs := Filtered([1..Size(omega)], i -> omega[i] in ovoid2);;
gap> stab := Stabilizer(Image(hom), imgs, OnSets);
<permutation group of size 1451520 with 8 generators>
gap> stabovoid := PreImage(hom, stab);;
```

Find the stabiliser

```
gap> g := CollineationGroup( pq );
PGammaO(7,3)
gap> points := AsList( Points(pq) );;
gap> hom := ActionHomomorphism(g, points, OnLieVarieties);
<action homomorphism>
gap> omega := HomeEnumerator( UnderlyingExternalSet(hom) );;
gap> imgs := Filtered([1..Size(omega)], i -> omega[i] in ovoid2);;
gap> stab := Stabilizer(Image(hom), imgs, OnSets);
<permutation group of size 1451520 with 8 generators>
gap> stabovoid := PreImage(hom, stab);;
```

Orbits and composition series

```
gap> OrbitLengths(stabovoid,points,OnLieVarieties);
[ 336, 28 ]
gap> DisplayCompositionSeries(stabovoid);
G (size 1451520)
 | B(3,2) = O(7,2) ~ C(3,2) = S(6,2)
1 (size 1)
```

For more information and updates...

Ghent University (Dept. Pure Math.) website

http://cage.ugent.be/geometry/software.php