# Worksheet on Finite Geometry in GAP

Solutions

Jan De Beule

November 21, 2018

1. A *cap* of a projective space is a set $\mathcal{O}$ of points such that no three points of $\mathcal{O}$ are collinear.
   (a) Initialize $\mathrm{PG}(3, q)$, $q = 7$.

   ```
   gap> pg := PG(3,7);
   ProjectiveSpace(3, 7)
   ```

   (b) Check that an elliptic quadric $\mathrm{Q}^-(3,7)$ is indeed a cap of $\mathrm{PG}(3,7)$.

   ```
   gap> quadric := EllipticQuadric(3,7);
   Q-(3, 7)
   gap> set := List(Points(quadric));
   [ <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)>, <a point in Q-(3, 7)>,
     <a point in Q-(3, 7)>, <a point in Q-(3, 7)> ]
   gap> intersection_numbers :=
   List(Lines(pg),x->Number(set,y->y in x));; #;; to suppress output.
   gap> Collected(intersection_numbers);
   [ [ 0, 1225 ], [ 1, 400 ], [ 2, 1225 ] ]
   ```

   Conclusion: there are 1225 lines not containing a point of $\mathrm{Q}^-(3,7)$, 400 lines meeting $\mathrm{Q}^-(3,7)$ in exactly one point, and 1225 lines meeting $\mathrm{Q}^-(3,7)$ in exactly 2 points, and no line meeting $\mathrm{Q}^-(3,7)$ in more than two points. So no three points of $\mathrm{Q}^-(3,7)$ are collinear.

Useful commands: `PG`, `EllipticQuadric`, `List`, `Points`, `Lines`, `Number`.
solution file: `cappg3q.g`.

2. Consider the projective space $\mathrm{PG}(3,q)$, $q=5$, and its group of projectivities $G=\mathrm{PGL}(4,5)$. Compute a random element $g$ of $G$ of order $q^2+q+1$.

```
gap> pg := PG(3,q);
ProjectiveSpace(3, 5)
gap> group := ProjectivityGroup(pg);
The FinInG projectivity group PGL(4,5)
gap> g := First(group,x->Order(x)=q^2+q+1);
< a collineation: <cmat 4x4 over GF(5,1)>, F^0>
```

(a) Compute the orbits of the points of $\mathrm{PG}(3,q)$ under the group $\langle g\rangle$.

```
gap> cyclic := Group(g);
<projective collineation group with 1 generators>
gap> orbits := FiningOrbits(cyclic,Points(pg));
19%..39%..59%..79%..99%..100%..
[ <closed orbit, 31 points>, <closed orbit, 31 points>,
  <closed orbit, 31 points>, <closed orbit, 31 points>,
  <closed orbit, 31 points>, <closed orbit, 1 points> ]
```

(b) Check that most of these orbits span the complete space $\mathrm{PG}(3,q)$.

```
gap> List(orbits,x->Span(x));
[ ProjectiveSpace(3, 5), ProjectiveSpace(3, 5),
  <a plane in ProjectiveSpace(3, 5)>, ProjectiveSpace(3, 5),
  ProjectiveSpace(3, 5), <a point in ProjectiveSpace(3, 5)> ]
```

(c) Choose one of these orbits that span the full space, and compute the intersection numbers of this orbit with the lines of $\mathrm{PG}(3,q)$.

```
gap> set := First(orbits,x->Span(x)=pg);
<closed orbit, 31 points>
gap> Collected(List(Lines(pg),x->Number(set,y->y in x)));
[ [ 0, 217 ], [ 1, 310 ], [ 2, 186 ], [ 3, 93 ] ]
```

(d) Repeat the same steps for $q=9$ and $q=17$.
Here is the session for $q=9$.

```
gap> q := 9;
9
gap> pg := PG(3,q);
ProjectiveSpace(3, 9)
gap> group := ProjectivityGroup(pg);
The FinInG projectivity group PGL(4,9)
gap> g := First(group,x->Order(x)=q^2+q+1);
< a collineation: <cmat 4x4 over GF(3,2)>, F^0>
gap> cyclic := Group(g);
<projective collineation group with 1 generators>
gap> orbits := FiningOrbits(cyclic,Points(pg));
11%..22%..33%..44%..55%..66%..77%..88%..99%..100%..
[ <closed orbit, 91 points>, <closed orbit, 91 points>,
  <closed orbit, 91 points>, <closed orbit, 91 points>,
  <closed orbit, 91 points>, <closed orbit, 91 points>,
  <closed orbit, 91 points>, <closed orbit, 91 points>,
```

```
  <closed orbit, 91 points>, <closed orbit, 1 points> ]
gap> List(orbits,x->Span(x));
[ ProjectiveSpace(3, 9), <a plane in ProjectiveSpace(3, 9)>,
  ProjectiveSpace(3, 9), ProjectiveSpace(3, 9), ProjectiveSpace(3, 9),
  ProjectiveSpace(3, 9), ProjectiveSpace(3, 9), ProjectiveSpace(3, 9),
  ProjectiveSpace(3, 9), <a point in ProjectiveSpace(3, 9)> ]
  gap> set := First(orbits,x->Span(x)=pg);
<closed orbit, 91 points>
gap> Collected(List(Lines(pg),x->Number(set,y->y in x)));
[ [ 0, 2184 ], [ 1, 3367 ], [ 2, 819 ], [ 3, 1092 ] ]
gap> time;
3481
```

An now for $q = 17$

```
gap> q := 17;
17
gap> pg := PG(3,q);
ProjectiveSpace(3, 17)
gap> group := ProjectivityGroup(pg);
The FinInG projectivity group PGL(4,17)
gap> g := First(group,x->Order(x)=q^2+q+1);
< a collineation: <cmat 4x4 over GF(17,1)>, F^0>
gap> cyclic := Group(g);
<projective collineation group with 1 generators>
gap> orbits := FiningOrbits(cyclic,Points(pg));
5%..11%..17%..23%..29%..35%..41%..47%..52%..58%..64%..
70%..76%..82%..88%..94%..99%..100%..
[ <closed orbit, 307 points>, <closed orbit, 307 points>,
  <closed orbit, 307 points>, <closed orbit, 307 points>,
  <closed orbit, 307 points>, <closed orbit, 307 points>,
  <closed orbit, 307 points>, <closed orbit, 307 points>,
  <closed orbit, 307 points>, <closed orbit, 307 points>,
  <closed orbit, 307 points>, <closed orbit, 307 points>,
  <closed orbit, 307 points>, <closed orbit, 307 points>,
  <closed orbit, 307 points>, <closed orbit, 307 points>,
  <closed orbit, 307 points>, <closed orbit, 1 points> ]
  gap> List(orbits,x->Span(x));
[ ProjectiveSpace(3, 17), ProjectiveSpace(3, 17), ProjectiveSpace(3, 17),
  <a plane in ProjectiveSpace(3, 17)>, ProjectiveSpace(3, 17),
  ProjectiveSpace(3, 17), ProjectiveSpace(3, 17), ProjectiveSpace(3, 17),
  ProjectiveSpace(3, 17), ProjectiveSpace(3, 17), ProjectiveSpace(3, 17),
  ProjectiveSpace(3, 17), ProjectiveSpace(3, 17), ProjectiveSpace(3, 17),
  ProjectiveSpace(3, 17), ProjectiveSpace(3, 17), ProjectiveSpace(3, 17),
  <a point in ProjectiveSpace(3, 17)> ]
gap> set := First(orbits,x->Span(x)=pg);
<closed orbit, 307 points>
gap> Collected(List(Lines(pg),x->Number(set,y->y in x)));
gap> Collected(List(Lines(pg),x->Number(set,y->y in x)));
[ [ 0, 27937 ], [ 1, 41752 ], [ 2, 5526 ], [ 3, 13815 ] ]
gap> time;
```

```
122444
```

(e) Could you make a guess about the set of points of such an orbit?
Any line of $\mathrm{PG}(3, q)$ meets the set in $0, 1, 2$, or $3$ points. Indeed one can show that the set is the set of points of a surface of degree 3.

Useful commands: `ProjectivityGroup`, `Order`, `First`, `FiningOrbits`, `Span`, `Number`.
solution file: `orbits.g`.

3. Consider the classical polar space $\mathrm{H}(5, q^2)$ (the Hermitian variety). This is a rank 3-geometry, it contains points, lines and planes. A *spread* of $\mathrm{H}(5, q^2)$ is a set $\mathcal{S}$ of planes of $\mathrm{H}(5, q^2)$ such that every point of $\mathrm{H}(5, q^2)$ is contained in exactly one plane of $\mathcal{S}$. A *partial spread* is a set $\mathcal{S}$ of planes of $\mathrm{H}(5, q^2)$ such that every point of $\mathrm{H}(5, q^2)$ is contained in at most one plane of $\mathcal{S}$. A partial spread is *maximal* if it cannot be extend to a larger partial spread.

(a) Initialize the classical polar space $\mathrm{H}(5, q^2)$, $q = 2$.

```
gap> q := 2;
2
gap> ps := HermitianPolarSpace(5,q^2);
H(5, 2^2)
```

(b) Initialize a graph (using the package grape) $\Gamma$, with vertex set the set of planes of $\mathrm{H}(5, q^2)$, and two vertices being adjacent if and only if they meet trivially. Note that grape allows to construct graphs using non-trivial symmetry groups.

```
gap> vertices := List(Planes(ps));;
gap> adj := function(x,y)
> return ProjectiveDimension(Meet(x,y))=-1;
> end;
function( x, y ) ... end
gap> group := CollineationGroup(ps);
PGammaU(6,2^2)
gap> act := OnProjSubspaces;
function( var, el ) ... end
gap> gamma := Graph(group,vertices,act,adj,true);;
```

(c) Find all maximal partial spreads of $\mathrm{H}(5, q^2)$. Use the grape command `CompleteSubgraphs`.

```
gap> mpss := CompleteSubgraphs(gamma);;
gap> Length(mpss);
374
```

(d) Refine your search now by checking the sizes of the found subgraphs, and recomputing, up to isomorphism, all subgraphs of a given size.

```
gap> Collected(List(mpss,x->Length(x)));
[ [ 7, 348 ], [ 9, 26 ] ]
gap> mps9 := CompleteSubgraphsOfGivenSize(gamma,9,2,true);
[ [ 1, 23, 45, 65, 87, 106, 165, 184, 206 ],
  [ 1, 23, 45, 65, 87, 106, 167, 189, 199 ],
  [ 1, 23, 45, 65, 169, 204, 275, 326, 651 ] ]
gap> mps7 := CompleteSubgraphsOfGivenSize(gamma,7,2,true);
[ [ 1, 23, 45, 65, 85, 315, 326 ], [ 1, 23, 45, 65, 85, 315, 571 ] ]
```

It turns out that there are only complete subgraphs of size 7 and 9. Up to isomorphism, there are only two examples of size 7 and two of size 9.

Useful commands: `HermtianPolarSpace`, `Planes`, `Meet`, `CollineationGroup`, `OnProjSubspaces`, grape package: `CompleteSubgraphs` `CompleteSubgraphsOfGivenSize`.
solution file: `mpsh5q2.g`.

4. Consider the point line geometry of points and lines of the parabolic quadric $\mathrm{Q}(4,q)$. This is a classical generalized quadrangle. A *partial ovoid* is a set $\mathcal{S}$ of points such that every line of $\mathrm{Q}(4,q)$ meets $\mathcal{S}$ in at most one point. It is an *ovoid* if every line meets $\mathcal{S}$ in exactly one point. By combinatorial arguments, an ovoid has size exactly $q^2+1$. A partial ovoid is *maximal* if it cannot be extended to a larger partial ovoid.

   (a) Initialize the parabolic quadric $\mathrm{Q}(4,q)$, $q=5$.

```
gap> q := 5;
5
gap> ps := ParabolicQuadric(4,q);
Q(4, 5)
```

   (b) Initialize a graph (using the package grape) $\Gamma$, with vertex set the set of points of $\mathrm{Q}(4,q)$, and two vertices being adjacent if and only if they are not collinear as points of $\mathrm{Q}(4,q)$. Note that grape allows to construct graphs using a symmetry groups. Using this option speeds up the construction substantially!

```
gap> vertices := List(Points(ps));;
gap> adj := function(x,y)
> return x<>y and not IsCollinear(AmbientGeometry(x),x,y);
> end;
function( x, y ) ... end
gap> group := CollineationGroup(ps);
PGammaO(5,5)
gap> act := OnProjSubspaces;
function( var, el ) ... end
gap> gamma := Graph(group,vertices,act,adj,true);;
```

   (c) Show that, up to isomorphism, there is only one ovoid of $\mathrm{Q}(4,5)$, and it is an elliptic quadric $\mathrm{Q}^-(3,q)$. Note that this isomorphism check can be done using the grape command(s) in an appropriate way.

```
gap> gamma := Graph(group,vertices,act,adj,true);;
gap> ovoids := CompleteSubgraphsOfGivenSize(gamma,q^2+1,2,true);
[ [ 1, 2, 21, 22, 23, 24, 41, 42, 43, 44, 57, 58, 59, 60, 125, 126, 127,
      128, 141, 142, 143, 144, 153, 154, 155, 156 ] ]
gap> ovoid := List(ovoids[1],x->VertexNames(gamma)[x]);
[ <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
```

```
  <a point in Q(4, 5)>, <a point in Q(4, 5)> ]
gap> Span(ovoid);
<a solid in ProjectiveSpace(4, 5)>
gap> List(Lines(ps),x->Number(ovoid,y->y in x));
[ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1 ]
```

So indeed all lines of $\mathrm{Q}(4,5)$ meet the point set in exactly one point. So it is an ovoid.

(d) Find the largest maximal partial ovoid of $\mathrm{Q}(4,5)$. Check that it is unique up to isomorphism. Show that the lines not containing a point of this maximal partial ovoid, lie in a hyperplane of $\mathrm{PG}(4,q)$, necessarily meeting $\mathrm{Q}(4,q)$ in a hyperbolic quadric $\mathrm{Q}^+(3,q)$.

```
gap> mpos1 := CompleteSubgraphsOfGivenSize(gamma,q^2,2,true);
[  ]
```

So there is no maximal partial ovoid of size $q^2$.

```
gap> mpos2 := CompleteSubgraphsOfGivenSize(gamma,q^2-1,2,true);
[ [ 1, 2, 21, 22, 27, 36, 41, 42, 51, 52, 59, 60, 65, 66, 113, 118, 125,
      127, 132, 139, 144, 149, 151, 154 ] ]
gap> max_partial_ovoid := List(mpos2[1],x->VertexNames(gamma)[x]);
[ <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)>,
  <a point in Q(4, 5)>, <a point in Q(4, 5)>, <a point in Q(4, 5)> ]
gap> skew_lines :=
Filtered(Lines(ps),x->Number(max_partial_ovoid,y->y in x)=0);
[ <a line in Q(4, 5)>, <a line in Q(4, 5)>, <a line in Q(4, 5)>,
  <a line in Q(4, 5)>, <a line in Q(4, 5)>, <a line in Q(4, 5)>,
  <a line in Q(4, 5)>, <a line in Q(4, 5)>, <a line in Q(4, 5)>,
  <a line in Q(4, 5)>, <a line in Q(4, 5)>, <a line in Q(4, 5)> ]
gap> sub := Span(skew_lines);
<a solid in ProjectiveSpace(4, 5)>
gap> TypeOfSubspace(ps,sub);
"hyperbolic"
```

Useful commands: `ParabolicQuadric`, `Points`, `IsCollinear`, `CollineationGroup`, `OnProjSubspaces`, `AmbientGeometry`, grape package: `CompleteSubgraphsOfGivenSize`, `VertexNames`, `Span`.
solution file: `mpoq4q.g`.

5. The *Klein correspondence*, mapping lines of $\mathrm{PG}(3,q)$ to points of $\mathrm{Q}^+(5,q)$ is quite powerful. Here you will combine this geometry morphism with *field reduction* to describe a spread of a symplectic space.

(a) Define $q = 9$. Initialize $\mathrm{PG}(1, q^2)$ and store all its points in a list.

```
gap> pg1 := PG(1,q^2);
ProjectiveSpace(1, 81)
gap> pts := List(Points(pg1));;
```

(b) Initialize the geometry morphism based on field reduction, mapping points of $\mathrm{PG}(1, q^2)$ to lines of $\mathrm{PG}(3, q)$.

```
gap> em := NaturalEmbeddingByFieldReduction(pg1,GF(9));
<geometry morphism from <All elements of ProjectiveSpace(1,
81)> to <All elements of ProjectiveSpace(3, 9)>>
```

(c) Use this geometry morphism to construct a spread of $\mathrm{PG}(3, q)$. Check that the set of lines you get is indeed a spread of $\mathrm{PG}(3, q)$.

```
gap> lines := List(pts,x->x^em);;
gap> pg3 := PG(3,q);
ProjectiveSpace(3, 9)
gap> Union(List(lines,x->List(Points(x))))=Set(Points(pg3));
true
```

Note that the previous command loops over all lines in the list, computes all points of each line, creates the union of all these point set, and compare this set with the set of all points of $\mathrm{PG}(3, q)$.

(d) Now initialize and use the Klein correspondence to convert the line spread of $\mathrm{PG}(3, q)$ into a set of points of the hyperbolic quadric $\mathrm{Q}^+(5, q)$. Note that the representation of $\mathrm{Q}^+(5, q)$ is not important.

```
gap> klein := KleinCorrespondence(GF(9));
<geometry morphism from <lines of ProjectiveSpace(3, 9)> to <points of Q+(5,
9): x_1*x_6+x_2*x_5+x_3*x_4=0>>
gap> hyperbolic := AmbientGeometry(Range(klein));
Q+(5, 9): x_1*x_6+x_2*x_5+x_3*x_4=0
```

Here we just checked what the representation of the quadric of Klein is. Any representation would do.

(e) Show that the point set of $\mathrm{Q}^+(5, q)$ spans a solid $S$ of $\mathrm{PG}(5, q)$, meeting $\mathrm{Q}^+(5, q)$ in an elliptic quadric $\mathrm{Q}^-(3, q)$.

```
gap> pointset := List(lines,x->x^klein);;
gap> S := Span(pointset);
<a solid in ProjectiveSpace(5, 9)>
gap> ProjectiveDimension(S);
3
gap> TypeOfSubspace(hyperbolic,S);
"elliptic"
```

(f) This means that every hyperplane $\pi$ containing $S$ meets $\mathrm{Q}^+(5, q)$ in a parabolic quadric. Choose any hyperplane $\pi$ and embed that standard parabolic quadric $\mathrm{Q}(4, q)$ as hyperplane section in $\mathrm{Q}^+(5, q)$. Then compute the preimage of the point set under this embedding.

```
gap> pi := Random(ElementsIncidentWithElementOfIncidenceStructure(S,5));
<a proj. 4-space in ProjectiveSpace(5, 9)>
```

```
gap> q4q := ParabolicQuadric(4,q);
Q(4, 9)
gap> em2 := NaturalEmbeddingBySubspace(q4q,hyperbolic,pi);
<geometry morphism from <Elements of Q(4, 9)> to <Elements of Q+(5,
9): x_1*x_6+x_2*x_5+x_3*x_4=0>>
gap> dualspread := List(pointset,x->PreImageElm(em2,x));;
```

(g) Now use that natural duality between $\mathrm{Q}(4,q)$ and $\mathrm{W}(3,q)$ to obtain the line spread of $\mathrm{PG}(3,q)$ as a line spread of $\mathrm{W}(3,q)$, the symplectic generalized quadrangle.

```
gap> w3 := SymplecticSpace(3,q);
W(3, 9)
gap> duality := NaturalDuality(q4q,w3);
<geometry morphism from <Elements of Q(4, 9)> to <Elements of W(3, 9)>>
gap> symplectic_spread := List(dualspread,x->x^duality);;
gap> Union(List(symplectic_spread,x->List(Points(x))))=Set(Points(w3));
true
```

Note that the very last comment is comparable with the last command in from (c): we really have a spread of the symplectic space $\mathrm{W}(3,9)$.

Useful commands: `NaturalEmbeddingByFieldReduction`, `KleinCorrespondence`, `Range`, `AmbientGeometry`, `ProjectiveDimension`, `Span`, `ElementsIncidentWithElementOfIncidenceStructure`, `NaturalEmbeddingBySubspace`, `SymplecticSpace`, `NaturalDuality`.
solution file: `symplectic.g`.

6. We will consider in $\mathrm{PG}(3,q)$, $q = 2^{2e+1}$, a point set of which the coordinates are given as follows.

$$\mathcal{O} = \{(1, st + s^{\sigma+2} + t^{\sigma}, s, t) : s, t \in \mathbb{F}_q\} \cup \{(0,1,0,0)\}.$$

(a) Initalize $\mathrm{PG}(3,q)$, with $q = 8$.

```
gap> e := 1;
1
gap> q := 2^(2*e+1);
8
gap> pg := PG(3,q);
ProjectiveSpace(3, 8)
```

(b) Use the above list of coordinates to create the set of points $\mathcal{O}$ in $\mathrm{PG}(3,8)$.

```
gap> field := GF(q);
GF(2^3)
gap> sigma := 2^(e+1);
4
gap> o := One(field);
Z(2)^0
gap> vectors :=
Union(List(field,s->List(field,t->[o,s*t+s^(sigma+2)+t^sigma,s,t])));;
gap> Add(vectors,[0*o,o,0*o,0*o]);
gap> tits := List(vectors,x->VectorSpaceToElement(pg,x));;
```

(c) Check that it is a cap.

```
gap> intersection_numbers_of_lines :=
List(Lines(pg),x->Number(tits,y->y in x));;
gap> Collected(intersection_numbers_of_lines);
[ [ 0, 2080 ], [ 1, 585 ], [ 2, 2080 ] ]
```

(d) Create a list with all tangent planes and check that every point $p \in \mathcal{O}$ defines a unique *tangent plane*, i.e. a plane meeting $\mathcal{O}$ only in $p$.

```
gap> intersection_numbers_of_planes :=
List(Planes(pg),x->Number(tits,y->y in x));;
gap> Collected(intersection_numbers_of_planes);
[ [ 1, 65 ], [ 9, 520 ] ]
gap> tangent_planes :=
Filtered(Planes(pg),x->Number(tits,y->y in x)=1);;
gap> Length(tangent_planes);
65
```

(e) Compute the setwise stabilizer group $S$ of $\mathcal{O}$ in the group of special homographies of $\mathrm{PG}(3, q)$. Compute the order of the group.

```
gap> group := SpecialHomographyGroup(pg);
The FinInG PSL group PSL(4,8)
gap> stab := FiningSetwiseStabiliser(group,tits);
#I  Computing adjusted stabilizer chain...
<projective collineation group with 5 generators>
gap> time;
146697
```

(f) Check which finite simple group you found by computing $S$.

```
gap> IsSimple(stab);
true
gap> StructureDescription(stab);
"Sz(8)"
```

It is one of the Suzuki groups!

Useful commands: `PG`, `VectorSpaceToElement`, `Lines`, `Planes`, `SpecialHomographyGroup`, `FiningSetwiseStabiliser`, `Order`, `StructureDescription`. solution file: `titspg3q.g`. The object described here is the famous ovoid of Jacques Tits.

7. We will consider an ovoid in a classical generalized quadrangle. The classical generalized quadrangle will be the point-line geometry on the parabolic quadric $\mathrm{Q}(4, q)$, $q = p^h$, $p$ an odd prime, $h > 1$.

(a) Let $q = 9$. Initialize the parabolic quadric $\mathrm{Q}(4, q)$ with equation $X_2^2 + X_0X_4 + X_3X_1 = 0$.

```
gap> p := 3;
3
gap> h := 2;
2
gap> q := p^h;
9
gap> field := GF(q);
GF(3^2)
```

```
gap> o := One(field);
Z(3)^0
gap> mat := [[0,0,0,0,1],[0,0,0,1,0],[0,0,1,0,0],[0,0,0,0,0],[0,0,0,0,0]]*o;
[ [ 0*Z(3), 0*Z(3), 0*Z(3), 0*Z(3), Z(3)^0 ],
  [ 0*Z(3), 0*Z(3), 0*Z(3), Z(3)^0, 0*Z(3) ],
  [ 0*Z(3), 0*Z(3), Z(3)^0, 0*Z(3), 0*Z(3) ],
  [ 0*Z(3), 0*Z(3), 0*Z(3), 0*Z(3), 0*Z(3) ],
  [ 0*Z(3), 0*Z(3), 0*Z(3), 0*Z(3), 0*Z(3) ] ]
gap> form := QuadraticFormByMatrix(mat,field);
< quadratic form >
gap> ps := PolarSpace(form);
<polar space in ProjectiveSpace(4,GF(3^2)): x1*x_5+x2*x4+x3^2=0 >
```

(b) Denote $\sigma$ the unique non-trivial automorphism of $\mathbb{F}_9$. Let $-k$ be a non-square in $\mathbb{F}_9$. Define on $\mathrm{Q}(4,q)$ the point set with coordinates given as follows

$$\mathcal{O} = \{(1, s, t, ks^{\sigma}, -t^2 - ks^{\sigma+1}) : s, t \in \mathbb{F}_q\} \cup \{(0, 0, 0, 0, 1)\}$$

```
gap> sigma := p;
3
gap> k := -Z(q);
Z(3^2)^5
gap> vectors :=
Union(List(field,s->List(field,t->[o,s,t,k*s^p,-t^2-k*s^(sigma+1)])));;
gap> Add(vectors,[0*o,0*o,0*o,0*o,o]);
gap> ovoid := List(vectors,x->VectorSpaceToElement(ps,x));;
```

(c) Check that $\mathcal{O}$ is an *ovoid* of $\mathrm{Q}(4,q)$, i.e. a set of points such that every line of $\mathrm{Q}(4,q)$ meets $\mathcal{O}$ in exactly one point.

```
gap> Collected(List(Lines(ps),x->Number(ovoid,y->y in x)));
[ [ 1, 820 ] ]
```

So all 820 lines of $\mathrm{Q}(4,9)$ meet the set in exactly one point, so it is an ovoid.

(d) Check that every elliptic quadric contained in $\mathrm{Q}(4,q)$ meets $\mathcal{O}$ in $1 \bmod p$ points.

```
gap> Collected(List(Hyperplanes(AmbientSpace(ps)),
        x->Number(ovoid,y->y in x)));
[ [ 1, 106 ], [ 4, 810 ], [ 7, 1296 ], [ 10, 4347 ], [ 13, 648 ],
  [ 16, 162 ], [ 28, 12 ] ]
```

After this command, we conclude that every hyperplane of $\mathrm{PG}(4,9)$ meet $\mathcal{O}$ in 1 mod $p$ points. Since every elliptic quadric contained in $\mathrm{Q}(4,9)$ spans a hyperplane, necessarily every elliptic quadric meets $\mathcal{O}$ in 1 mod $p$ points. But this was not necessarily the most efficient way to find the answer to the question.

(e) Compute the stabilizer group of $\mathcal{O}$ in the special isometry group of $\mathrm{Q}(4,q)$.

```
gap> group := SpecialIsometryGroup(ps);
#I  Computing collineation group of canonical polar space...
<projective collineation group of size 3443212800 with 3 generators>
gap> stab := FiningSetwiseStabiliser(group,ovoid);
#I  Computing adjusted stabilizer chain...
<projective collineation group with 8 generators>
gap> time;
61962
```

Useful commands: `QuadraticFormByMatrix` (forms package) `PolarSpace`, `AmbientSpace`, `Hyperplanes`, `SpecialIsometryGroup`, `FiningSetwiseStabiliser`.
solution file: `kantor.g`.
The ovoid is an example of non-classical ovoids of $Q(4,9)$, i.e. an ovoid different from and elliptic quadric (and hence necessarily spanning $PG(4,q)$).

8. Combine exercises 5 and 6. A famous theorem states that any ovoid $\mathcal{O}$ of $PG(3,q)$, $q$ even, defines a symplectic generalized quadrangle $\mathcal{S}$ as follows. The points are the points of $PG(3,q)$, and the lines are the lines of $PG(3,q)$ tangent to $\mathcal{O}$. Starting from a given ovoid $\mathcal{O}$, one can in principle compute the form. In this exercise, we will reconstruct the symplectic space around the ovoid geometrically using the Klein correspondence. Then we will use the natural duality of $W(3,q)$ with itself and the geometry isomorphism between symplectic spaces described by different symplectic forms to investigate the two different spreads we get.

   (a) Construct the ovoid $\mathcal{O}$ of exercise (4) in $PG(3,8)$.

```
gap> e := 1;
1
gap> q := 2^(2*e+1);
8
gap> field := GF(q);
GF(2^3)
gap> sigma := 2^(e+1);
4
gap> o := One(field);
Z(2)^0
gap> $igma,s,t])));;
gap> Add(vectors,[0*o,o,0*o,0*o]);;
gap> pg := PG(3,q);
ProjectiveSpace(3, 8)
gap> tits_ovoid := List(vectors,x->VectorSpaceToElement(pg,x));;
```

   (b) Compute the lines and planes of $PG(3,8)$ tangent to $\mathcal{O}$.

```
gap> tangent_lines := Filtered(Lines(pg),x->Number(tits,y->y in x)=1);;
gap> tangent_planes := Filtered(Planes(pg),x->Number(tits,y->y in x)=1);;
```

   (c) Map these tangent lines, respectively planes to points, respectively planes of $Q^+(5,q)$ using the Klein correspondence.

```
gap> klein := KleinCorrespondenceExtended(q);
<geometry morphism from <All elements of ProjectiveSpace(3,
8)> to <Elements of <polar space in ProjectiveSpace(
5,GF(2^3)): x_1*x_6+x_2*x_5+x_3*x_4=0 >>>
gap> points := List(tangent_lines,x->x^klein);;
gap> prelines := List(tangent_planes,x->x^klein);;
gap> prelines[1];
<a plane in Q+(5, 8): x_1*x_6+x_2*x_5+x_3*x_4=0>
gap> span := Span(points);
<a proj. 4-space in ProjectiveSpace(5, 8)>
gap> prelines2 := List(prelines,x->Meet(x,span));;
gap> prelines2[1];
<a line in ProjectiveSpace(5, 8)>
```

Note that the elements in the list `prelines` are actually planes of $\mathrm{Q}^+(5,q)$. As we have seen, a plane of $\mathrm{PG}(3,q)$ is indeed mapped by the Klein correspondence (in FinInG called the `KleinCorrespondenceExtended`, because the *usual* Klein correspondence is only defined on the lines of $\mathrm{PG}(3,q)$), on a plane of $\mathrm{Q}^+(5,q)$. But a tangent plane to $\mathcal{O}$ determines a unique point of $\mathcal{O}$. So instead of considering the map of the tangent plane under the Klein correspondence, we should consider the map of the $q+1$ tangent lines in the plane under the Klein correspondence. Simply intersecting a plane of $\mathrm{Q}^+(5,q)$ with the hyperplane spanned by the set `points` will produce this set of lines (think about this geometrically). The result is a set of lines of $\mathrm{PG}(5,q)$, that are all lines of the quadric. The command `Embed` converts this lines into lines of $\mathrm{Q}^+(5,q)$.

(d) Check that this set of points (and lines) on $\mathrm{Q}^+(5,q)$ spans a hyperplane of $\mathrm{PG}(5,q)$.

```
gap> kleinquadric := AmbientGeometry(Range(klein));
Q+(5, 8): x_1*x_6+x_2*x_5+x_3*x_4=0
gap> lines := List(prelines2,x->Embed(kleinquadric,x));;
gap> lines[1];
<a line in Q+(5, 8): x_1*x_6+x_2*x_5+x_3*x_4=0>
```

(e) So these points and lines are the points and lines of an embedded parabolic quadric $\mathrm{Q}(4,q)$, embed $\mathrm{Q}(4,q)$ into $\mathrm{Q}^+(5,q)$ as subspace intersection of $\mathrm{Q}^+(5,q)$.

```
gap> em := NaturalEmbeddingBySubspace(q4q,kleinquadric,span);
<geometry morphism from <Elements of Q(4, 8)> to <Elements of Q+(5,
8): x_1*x_6+x_2*x_5+x_3*x_4=0>>
```

(f) No use the natural duality between $\mathrm{Q}(4,q)$ and $\mathrm{W}(3,q)$ to convert this particular set of lines of $\mathrm{Q}(4,q)$ to a set of points of $\mathrm{W}(3,q)$. Now check that the obtained point set of $\mathrm{W}(3,q)$ is indeed an ovoid.

```
gap> q4qlines := List(lines,x->PreImageElm(em,x));;
gap> duality := NaturalDuality(q4q);
<geometry morphism from <Elements of Q(4, 8)> to <Elements of W(3, 8)>>
gap> tits_ovoid_in_w3 := List(q4qlines,x->x^duality);;
gap> w3 := AmbientGeometry(Range(duality));
W(3, 8)
gap> Collected(List(Lines(w3),x->Number(tits_ovoid_in_w3,y->y in x)));
[ [ 1, 585 ] ]
```

Useful commands: `NaturalEmbeddingBySubspace`, `NaturalDuality`, `KleinCorrespondeceExtended`.
solution file: `combination.g`

9. The André-Bruck-Bose representation (ABB) is a representation of *translation planes*. Let $S$ be any spread of $\mathrm{PG}(3,q)$, i.e. a set of lines partitioning the point set. Embed $\mathrm{PG}(3,q)$ as a hyperplane $\pi$ in $\mathrm{PG}(4,q)$. Now define

- Points: points of type (i): the points of $\mathrm{PG}(4,q) \setminus \pi$; points of type (ii): the lines of $S$;
- Lines: lines of type (a): the planes of $\mathrm{PG}(4,q)$ meeting $\pi$ in a line of $S$; line of type (b): the hyperplane $\pi$
- Incidence between Points and Lines is the natural incidence.

(a) Construct (see exercise (5)) a Desarguesian spread of $\mathrm{PG}(3,q)$.

```
gap> pg1 := PG(1,q^2);
ProjectiveSpace(1, 9)
gap> pg3 := PG(3,q);
ProjectiveSpace(3, 3)
gap> em := NaturalEmbeddingByFieldReduction(pg1,pg3);
<geometry morphism from <All elements of ProjectiveSpace(1,
9)> to <All elements of ProjectiveSpace(3, 3)>>
gap> spread := List(Points(pg1),x->x^em);;
```

(b) Turn this spread into a non-Desarguesian spread by swapping a regulus. A regulus is determined by three lines $l_1, l_2, l_3$ as follows. for each point $p \in l_1$, there is a unique line $m_p$ meeting $l_1, l_2, l_3$ in precisely a point. So the three lines $l_1, l_2, l_3$ determine $q+1$ lines $m_i : i = 1 \ldots q+1$. This is the opposite regulus $R'$. The regulus determined by the lines $l_1, l_2, l_3$ is precisely the opposite regulus determined by any three lines of $R'$. Clearly, replacing the lines of $R$ by $R'$ yields again a spread of $\mathrm{PG}(3, q)$. Now define the spread $S := (S \setminus R) \cup R'$.

```
gap> klein := KleinCorrespondence(q);
<geometry morphism from <lines of ProjectiveSpace(3, 3)> to <points of Q+(5,
3): x1*x_6+x2*x_5+x3*x4=0>>
gap> kleinquadric := AmbientGeometry(Range(klein));
Q+(5, 3): x1*x_6+x2*x_5+x3*x4=0
gap> quadric_pts := List(spread,x->x^klein);;
gap> pg5 := PG(5,q);
ProjectiveSpace(5, 3)
gap> $dric,x)="parabolic");
<a plane in ProjectiveSpace(5, 3)>
gap> phi := PolarityOfProjectiveSpace(kleinquadric);
<polarity of PG(5, GF(3)) >
gap> perpplane := plane^phi;
<a plane in ProjectiveSpace(5, 3)>
gap> conic := Filtered(quadric_pts,x->x in plane);;
gap> perpconic := Filtered(Points(kleinquadric),x->x in perpplane);;
gap> ovoid := Union(perpconic,Difference(quadric_pts,conic));;
gap> hall_spread := List(ovoid,x->PreImageElm(klein,x));
[ <a line in ProjectiveSpace(3, 3)>, <a line in ProjectiveSpace(3, 3)>,
  <a line in ProjectiveSpace(3, 3)>, <a line in ProjectiveSpace(3, 3)>,
  <a line in ProjectiveSpace(3, 3)>, <a line in ProjectiveSpace(3, 3)>,
  <a line in ProjectiveSpace(3, 3)>, <a line in ProjectiveSpace(3, 3)>,
  <a line in ProjectiveSpace(3, 3)>, <a line in ProjectiveSpace(3, 3)> ]
```

A nice way to swap a regulus, is to use the Klein correspondence. A regulus corresponds with a conic contained in $\mathrm{Q}^+(5, q)$, and the opposite regulus with the image of this conic under the polarity of the projective space associated with the hyperbolic quadric $\mathrm{Q}^+(5, q)$. This explains the above commands.

(c) Define the points and lines from the ABB representation with relation to the spread $S'$.

```
gap> pg4 := PG(4,q);
ProjectiveSpace(4, 3)
gap> inf := HyperplaneByDualCoordinates(pg4,[1,0,0,0,0]*Z(q)^0);
<a solid in ProjectiveSpace(4, 3)>
gap> em := NaturalEmbeddingBySubspace(pg3,pg4,inf);
```

```
<geometry morphism from <All elements of ProjectiveSpace(3,
3)> to <All elements of ProjectiveSpace(4, 3)>>
gap> hall_spread_embedded := List(hall_spread,x->x^em);;
gap> pointsi := Filtered(Points(pg4),x->not x in inf);;
gap> pointsii := hall_spread_embedded;;
gap> points := Union(pointsi,pointsii);;
gap> Union(List(hall_spread_embedded,
          x->Filtered(Planes(x),y->not y in inf)));;
gap> linesb := [inf];
[ <a solid in ProjectiveSpace(4, 3)> ]
gap> pts := Union(pointsi,pointsii);;
gap> lines := Union(linesa,linesb);;
```

(d) The incidence is the natural incidence.

(e) Look at the command `GeneralisedPolygonByElements`, you need the compute the stabilizer group of the spread $S'$, and the incidence as an argument for this command is `\*`.

```
gap> group_inf := FiningStabiliser(ProjectivityGroup(pg4),inf);
<projective collineation group of size 1965150720 with 3 generators>
gap> hall_group := FiningSetwiseStabiliser(group_inf,hall_spread_embedded);
#I  Computing adjusted stabilizer chain...
<projective collineation group with 14 generators>
gap> GeneralisedPolygonByElements(pts,lines,\*,hall_group,OnProjSubspaces);
<projective plane order 9>
```

Note that the command `GeneralisedPolygonByElements` also requires a group as argument. This group must be a subgroup of the collineation group of the generalised polygon. Clearly, the stabilizer of the spread at infinity will satisfy this condition.

(f) The command `CollineationGroup` is applicable on the generalised polygon you got in (e). However, it will take a lot of time to compute it. If the package `digraph` is available, you can (i) compute the incidence graph using `IncidenceGraph`, (ii) convert this grape object into a digraph object simply by applying `Digraph` on it, (iii) now computing the digraph-automorphism group using `AutomorphismGroup`. Compare the result with the stabilizer group of the non-Desarguesian spread. What is the mathematical conclusion?

```
gap> graph := IncidenceGraph(plane);;
gap> LoadPackage("digraph");
...
true
gap> digraph := Digraph(graph);
<digraph with 182 vertices, 1820 edges>
gap> group := AutomorphismGroup(digraph);
<permutation group with 7 generators>
gap> Order(group);
311040
gap> Order(hall_group);
311040
```

So we conclude that the collineation group of the projective plane is exactly the stabilizer group of the spread at infinity! Also note that the stabilizer group of the spread in this example was computed within the stabilizer group of the hyperplane in the group $\mathrm{P\Gamma L}(5, q)$. A translation plane constructed from a spread that has one regulus swapped compared with a Desarguesian spread, is called the Hall plane (of order $q$).

solution file: `hall.g`.