On doing geometry in CAYLEY

PJC, November 1988

1 Introduction

These notes were produced in response to a question from John Cannon: what issues are involved in adding facilities to CAYLEY for doing finite geometry? More specifically, what facilities would geometers want?

At first sight, the idea of geometry in CAYLEY seems very natural. Groups and geometry have been intimately linked since long before the time of Klein. Many of the groups with which CAYLEY users deal act naturally on geometries. An important point which I'll touch on briefly later is this: if we're given a group which happens to be a classical group, many algorithmic questions (e.g. finding particular kinds of subgroups) become much easier once we've found and coordinatised the geometry on which the group acts.

With further thought, however, we see how opposed in spirit the two areas are. Everyone agrees on what a group is. Moreover, there are only a few ways in which a group is likely to be presented, and CAYLEY has facilities to deal with each of these. In each case, these are basic algorithms of such subtlety as to deter the casual user from implementing them – surely one of the facts that called CAYLEY into being in the first place.

On the other hand, the sheer variety of finite geometries daunts the beginner, who meets projective, affine and polar spaces, buildings, generalised polygons, (semi) partial geometries, t-designs, Steiner systems, Latin squares, nets, codes, matroids, permutation geometries, to name just a few. The range of questions asked about them is even more bewildering. Not surprisingly, there are few algorithms of comparable sophistication to the meataxe, Todd–Coxeter, or Knuth–Bendix. If you want to find something, you probably have to do an exhaustive search, with backtracking, and expect to have to wait a long time for the result.

Fortunately, the system designer doesn't have to know all the different axiom systems. The computer must hold a representation of the object, but doesn't have to do mathematics on it unless the user asks it to by writing a program. (There are some exceptions to this, for example, coordinatisaton procedures; but even there, the computer doesn't have to prove that they work.)

I have tried to do two things in these notes. First, following John Cannon's suggestion, I've described some geometric problems I've worked on recently, to illustrate the kin of situations that geometers face. I've refrained from describing the use I made of computers here; what I did by hand that could be automated is far more relevant. Secondly, I've written a short survey of finite geometry, biased towards this discussion. I cover general classes of geometries, their connections with groups, and operations on them, together with a couple of specific topics (coding theory and coordinatisation). Above all, I've tried to present the issues rather than dictate design principles. I hope these notes can be of some help; but there will be things I've omitted, and other people will surely want to change the emphasis in places. Also, other people's case studies would teach different lessons.

2 Dr Cameron's casebook; or, Diary of a practising geometer

I want to describe here a few comparatively small pieces of work I have done recently, hopefully in enough detail to show the facilities of CAYLEY which I would have used, had I been a regular used and had they been available. In fact, the last of these hasn't been done yet (I did a bit of work on it some years ago, mainly as a learning experience in Z80 assembly language; I intend to return to it soon, but I will start again from scratch), but in the other cases, the results were obtained by hand. In fact, in two of them, I constructed the geometries involved with standalone programs, and used them to check some of the intermediate assertions that had been established by hand; to some extent, it was the inflexibility of the representations I obtained that accounts for my not persisting with the computer.

2.1 A 24-set in $\mathrm{AG}(8,2)$

Flag-transitive 5-designs are very rare. Analysis I did with Cheryl Praeger led me to suspect the existence of such a design on 256 points with block size 24. The analysis showed that, to construct the design, it is necessary and sufficient to find a set S of 24 points in $\mathrm{AG}(8,2)$ with the following properties:

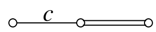
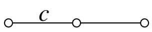
(a) the setwise stabiliser of S in $\mathrm{AGL}(8,2)$ acts transitively on it;

(b) S contains 42 affine planes.

After several unsuccessful attempts to construct such a set, I decided on different tactics. Using a slight variant of a construction I'll describe later (construction B in section 6), the existence of such an S is equivalent to that of a binary code of length 24, dimension 15, with all weights even, having minimum weight 4, and containing exactly 42 words of weight 4.

Such a code exists. The example I found is spanned by the Golay code together with three judiciously chosen words of weight 4. It admits a maximal subgroup $2^6 L_3(2):S_3$ of M_{24}. (The structure of the group I deduced, and its maximality in M_{24} is taken from the ATLAS.) Of course, it would be nice to check the assertions, to have the set S given in a convenient form for study, and to check that the group I have is the full automorphism group. (The last "fact", for which I don't yet have a proof, is needed to establish the value of λ in the flag-transitive 5-design which was the original goal).

2.2 A pattern of ovoids in a generalised quadrangle

With Paul Fisher and Dan Hughes, I was studying extensions of generalised quadrangles to Buekenhout geometries having the diagram . (A natural question, in view of the great importance of extensions of projective planes to geometries with diagram .) In one particular case, we had to examine, the GQ had parameters $s=4$, $t=2$, and so is the operand for the group $U_4(2)$; that is, its points and lines are the totally singluar points and lines of a Hermitian form on $\mathrm{PG}(3,4)$. The question reduced to the following: Does this GQ possess a set of 10 ovoids with the property that any two of them meet in exactly one point? (An ovoid here is a set of 9 pairwise non-collinear points of the GQ.)

The non-existence proof goes roughly as follows.

Step 1 Find 200 ovoids. With each of the 40 non-singular points of $\mathrm{PG}(3,4)$ is associated a set of 5 ovooids partitioning the singular points. The ovoids are of two types: one ovoid labelled by each non-singular point is obvious, the other 4 rather more subtle.

Step 2 Show that there are no more than 200 ovoids. I found it more convenient to show that there are 200 spreads in the dual GQ, working with Schläfli's double-six representation.

Step 3 Show that, if two ovoids meet in one point, then the corresponding non-singular points are distinct and non-perpendicular.

Step 4 Use the amazing fact that the non-singular points of $\mathrm{PG}(3,4)$ can be mapped bijectively to all the points of $\mathrm{PG}(3,3)$ (with a symplectic form) so that perpendicularity is preserved. (This is related to the isomorphism of $U_4(2)$ and $\mathrm{Sp}_4(3)$.) This translates our putative set to an ovoid in the $\mathrm{Sp}_4(3)$ generalised quadrangle! But it is known that this quadrangle has no ovoids.

2.3 A partition into Steiner systems

Cheryl Praeger and I set out to decide whether the set of all 6-subsets of a 13-set can be partitioned into 13 copies of the Steiner system $S(5,6,12)$. (The question was motivated by work of Derrick Breach and Ann Street, who found that the 4-subsets of a 9-set can be partitioned into $S(3,4,8)$s in just two non-isomorphic ways, both admitting 2-transitive groups.)

We found that no such partition exists. The proof is written up (unlike the other things I'm describing). I think it may be worth reading through in this context. In brief, it involves considering successively permutation representations of S_{12} of degree 5040, M_{12} of degree 1584, M_{12} again of degree 144, and $\mathrm{PSL}(2,11)$ of degree 66. Much of the information needed could be obtained quite easily from CAYLEY in its present form. But the significant fact is always that the objects being permuted are not just abstract points, but actual copies of the Steiner system; we need to know things like: given two points, how many blocks do the corresponding Steiner systems have in common?

2.4 Random blocking sets

A blocking set in a projective plane is a set of points which meets every line but contains no line. Blocking sets exist in great profusion, but only a tiny fraction of them have been studied; I want to change the *weltanschauung* of this subject. For very small orders, it is possible to determine all blocking sets and classify them up to equivalence under the collineation group. But for larger orders, I propose to do this: choose random blocking sets to show that there are enormously many different ones (i.e. show that the sample obtained is a very small subset of the total).

To choose a random blocking set, start with the empty set, and add points to it until we obtain a set which meets every line; check that this set contains no line. The search can be tuned by using different strategies for choosing the pointsl but evidence suggests that, even without tuning, the probability of success goes rapidly to 1 as the order increases. (This is itself a question of some interest.)

A simple-minded strategy for checking equivalence is to replace each set found with the lexicographically least set in its orbit, keeping a list of these sets and their frequency of occurrence. Of course, for this we need to have the collineation group available!

3 What is a geometry, then?

It is a truism that no definition of geometry can be valid other than "geometry is what geometers do". The corollary of this is that, no matter how geometry is implemented in CAYLEY, you will have complaints from people that their favourite structures cannot be handled, or can be only with difficulty.

Twice in the past I have written surveys trying to address this question. I've found that the particular point of view of this assignment has concentrated my thoughts beneficially! I will describe one major and one minor class of geometries, together with some mutants and some aliens.

The most important class is that of *incidence geometries*. An incidence geometry consists of a set of *varieties*, divided into a certain number of *types*, together with a relation of *incidence* which holds between certain pairs of varieties of different types. In other words, it is just a multipartite graph (though few geometers would think of it in this way!). The standard example is a projective space: the varieties are the subspaces, type is dimension, and two varieties are incident if one contains the other. Block designs of various kinds are incidence geometries whose varieties are of two types (points and blocks).

Two subclasses of incidence geometries are of particular importance. The first consists of those with just two types (as the above example suggests). The second consists of the point-based geometries. The terminology I use here isn't standard; the class is too important to need a name really.

Suppose that there is a distinguished type whose members are called *points*. The *point-shadow* of a variety is the set of points incident with that variety. (It we adopt the convention that a variety is incident with itself and nothing else of its type, then the point-shadow of a point P is of course $\{P\}$.) I'll say that the geometry is *point-based* if distinct varieties have distinct point-shadows, and *strongly point-based* if two varieties are incident if and only if the point-shadow of one contains that of the other. So a projective geometry is strongly point-based; a block design is (strongly) point-baseed if and only if it has no repeated blocks. The kind of example I have in mind for geometries which are point-based but not strongly so would be the geometry of points, lines and conics in a projcetive plane, where for some reason we have decided to say that a line and a conic are incident if the line is tangent to the conic.

The importance of this is, of course, that in a point-based geometry we can represent all varieties other than points by ther point-shadows; this representation tells everything about incidence between points and other varieties, and if the geometry is strongly point-based then it tells absolutely everything. Even in the non-strongly point-based case, the geometry might be defined by the point-shadow representation with some additional rules (as in my example).

There are some mutants which have evolved out of the class of incidence geometries. Typically there are points and blocks; a block is a set of points with some additional structure. Examples are handcuffed designs and cyclic designs. In the first case, a block carries a betweenness relation (so that its members are handcuffed between the two at the ends); in the second, a block is cyclically ordered.

A very important subclass which fits here, though slightly uneasily, consists of matroids. There is a canonical procedure converting an arbitrary matroid to a geometric matroid (this corresponds to converting a vector space into a projective space by removing zero and identifying scalar multiples). Most geometers deal mainly with geometric matroids, though the more general form is sometimes needed. Now geometric matroids are examples of strongly point-based incidence geometries; the varieties are the flats or closed subsets. They naturally generalise the projective and affine spaces. The problem is that matroids can be equally well presented (and often are so presented) in other ways, i.e. by other families of subsets of the point set: for example, the independent sets, the bases, or the circuits. For handling matroids, one should be ab3e t6 switch between presentations.

My lesser class of geometries I will call chamber systems, though it includes many things which are not usually known by this name. A *chamber system* consists of a set C of *chambers*, with a collection $\{\pi_1, \ldots, \pi_k\}$ of partitions of C (i.e. equivalence relations on C). Two further conditions usually hold:

(a) two distinct chambers are i-equivalent (i.e. equivalent under π_i) for at most one value of i;

(b) for any two chambers, there is a chain connecting them in which successive terms are i-equivalent for some i.

In other words, it's a connected graph whose edges have been coloured with k colours so that each monochromatic subgraph is a disjoint union of complete graphs.

Incidence geometries give rise to chamber systems by the following construction. Consider an incidence geometry with varieties of k types, labelled $1,2,\ldots,k$. A *flag* is a set of pairwise incident varieties. Clearly it contains at most one variety of each type; it is *complete* if it has a variety of each type. Now the chambers of the chamber system are the complete flags; chambers F,F' are i-equivalent if and only if they contain the same varieties of all types except i. (The connectedness doesn't follow; but it is equivalent to a kind of connectivity of the incidence geometry which is often assumed by the practitioners anyway.)

There is a clear moral here. In an interesting incidence geometry, the number of flags greatly exceeds the number of varieties; so this is a terribly inefficient way of representing the geometry. On the other hand, not every chamber system comes from an incidence geometry; and chamber systems are theoretically important.

Buildings are probably the most important examples of chamber systems (though the most important buildings come from incidence geometries). But here's an example to show that there are others. A Latin square can be thought f as a chamber system: the chambers are the n^2 cells of the square, and there are three partitions, defined by the rows, columns, and symbols. This is not a frivolous example, for two reasons:

(i) Thought of as a graph, it's strongly regular; and one of the most prolific and important types of strongly regular graph is obtained.

(ii) This is in fact exactly how statisticians view Latin squares. (To put it simply: cells are agricultural plots, say, arranged in a square grid; symbols are treatments.) Their much more general class of "block structures" are similarly chamber systems. (I can't resist suggesting that, if CAYLEY is taught to handle these things, one useful interface would be to enable it to talk to some of the standard statistics packages!)

Finally, some aliens. There are geometries in which the basic notion is distance rather than incidence. Two examples in finite geometry are:

(i) Connected graphs, especially distance-regular graphs – but these are probably best treated as graphs.

(ii) Permutation geometries: the objects are permutations of a set, distance is Hamming distance. Of course, the facilities for describing these exist already. But there are various transformations between permutation geometries and incidence geometries. For example, sharply transitive sets of permutations correspond to Latin squares; sharply 2-transitive sets to projective planes; and arbitrary permutation geometries give rise to matroids.

4 Groups and geometries

The connection between groups and geometries is so strong that it needs no stressing. But I'll make a few general points before I plunge into details.

First, not all geometries are symmetric. (For example, asymptotically almost all Steiner triple systems have trivial automorphism group.) But it is reasonable to expect that CAYLEY will handle best the geometries with large automorphism grops, and (I claim, but more contentiously) that most CAYLEY users will be mainly interested in these.

Second and related, observe that once a few procedures for constructing geometries from groups have been added to the language, the existing libraries of specific groups will provide a huge database of interesting geometries.

The main thing to observe is that, if a group G acts on a geometry, then it is only necessary to give representatives of the G-orbits on objects and relations in order to define the geometry. (Thus, for an incidence geometry, we specify the permutation representations on varieties and representatives of the orbits on incident pairs; if it is strongly point-based, we need much less, just the representation on points and orbit representatives for the other varieties.) To see just how powerful this is, just look at the tables of block designs in Marshall Hall's "Combinatorial Theory".

In really nice cases, much less information yet is needed. I'll continue with several examples.

4.1 Coset geometries

Let $B, P_1, \ldots, P_k$ be subgroups of a group G, with $B \leq P_i$ for all i and $P_i \cap P_j = B$ for $i \neq j$; suppose that $P_1, \ldots, P_k$ generate G. Then there is a chamber system defined as follows: chambers are the cosets of B; two chambers are i-equivalent if they lie in the same coset of P_i. This construction gives all classical buildings: take for G a group of Lie type, B a Borel subgroup, $P_1, \ldots, P_k$ the minimal parabolics containing B. It also includes many sporadic group geometries (e.g. the 2-local geometries).

The incidence-geometry representation of the building is also easily obtained. Let $Q_1,\ldots,Q_k$ be the maximal parabolics containing B (so $Q_i = \langle P_j : j \neq i\rangle$). Then the varieties are the cosets of $Q_1,\ldots,Q_k$; types are obvious; and $Q_i x$ and $Q_j y$ are incident if and only if their intersection is nonempty (in which case it's a coset of $Q_i \cap Q_j$). Of course, this can be done for arbitrary subgroups $Q_1,\ldots,Q_k$; but flag-transitivity requires some conditions on the Q_i's.

4.2 Gamma and delta spaces

These geometries arise in the following way. G acts transtively on a set Ω of points. B is a self-converse union of G-orbits on Ω^2. For $\alpha \in \Omega$, put $\alpha^\perp = \{\beta : (\alpha,\beta) \in B\}$; for $\Delta \subseteq \Omega$, set $\Delta^\perp = \bigcap\{\alpha^\perp : \alpha \in \Delta\}$. Define the line $\alpha\beta$ to be $\{\alpha,\beta\}^{\perp\perp}$ (so that $\alpha,\beta \in \alpha\beta$). The geometry is interesting if two points lie on just one line and at least some lines have size greater than two. The point-line geometries of the classical polar spaces are all obtained in this way, taking

$$B = \{(\alpha,\beta) : \alpha = \beta \text{ or } \alpha \text{ and } \beta \text{ orthogonal}\}.$$

Other examples include the geometry of hyperbolic lines in symplectic space, Grassmannians, and a sporadic example on 231 points admitting M_{22}.

4.3 Small orbits of stabilisers

The prototype here is that, if G is t-transitive and the stabiliser of t points fixes additional points, then the translates of this fixed-point set are the blocks of a Steiner system. This can be generalised to the fixed-point set of a weakly closed subgroup of the stabiliser; there are related results too, like O'Nan's lemma. But I think that another point of view is likely to be more useful. Let us consider the case $t = 2$.

If there is a Steiner system $S(2,k,n)$ preserved by the 2-transitive group G, then the lilne containing two points α,β can usually be recognised bu the property that its points lie in small orbits of the stabiliser $G_{\alpha\beta}$, whereas the other points lie in large orbits. (Indeed, in projective and affine spaces, the points off the line form a single orbit.) So we can recognise the line through one pair of points very quickly, and find the others by translation.

4.4 Translation planes

A translation plane is defined by a *spread*, or set of n-dimensional subspaces of a $2n$-dimensional vector space V which partition the non-zero vectors; the lines of the plane are all the translates of the spread spaces. For many interesting translation planes, there is quite a large translation complement, or linear group G acting on V preserving the set of spread spaces. Thus the plane can be constructed from the matrix group G and bases for representatives of the orbits on spread spaces.

4.5 Difference sets

The representation of symmetric designs with regular automorphism groups by difference sets is famililar: the difference set is a subset of the group, and the blocks are all its translates. For cyclic groups (e.g. Singer cycles of projective planes) this gives a representation which is particularly convenient, since the cyclic group is so easy to implement. (Of course this is much less relevant in CAYLEY than in stand-alone programs.)

The case where automorphism groups have several orbits (such as the way Marshall Hall represents designs) is a generalisation of this.

Etc., etc...

5 Operations on geometries

These are some of the things that geometers would like to be able to do with their geometries.

5.1 Truncation

For incidence geometries, this just means throwing away some kinds of varieties. So, the point-line geometries of projective and polar spaces are obtained from the buildings by truncation. (Note that Grassmannians, etc. – see 4.2 – are not.)

5.2 Residues

The residue of a flag F consists of all those varieties not in F which are incident with every variety in F. For example, the induced variety on a subspace of a projective space, and the quotient geometry by a subspace, are residues.

Residues are crucial in Buekenhout geometries: apart from some connectivity conditions, a Buekenhout geometry belonging to a particular diagram is defined as one with certain rank 2 residues; the diagram is just a simple way of specifying the residues.

5.3 Geometry induced on a subset

In a point-based incidence geometry, we may wish to generalise 5.2 by defining a geometry on an arbitrary subset S (rather than just a subspace). There are several possibilities. We might take all the varieties contained in S. Alternatively, we might take the "non-trivial" intersections of varieties with S (e.g. lines meeting S in at least two points; planes containing three non-collinear points of S; etc.)

5.4 Matroid restriction and contraction

These are really special cases of 5.2 and 5.3 (the induced subgeometry on all but one point, or the quotient by a point), but deserve special consideration in view of their importance for matroids (and the alternative presentations of matroids).

5.5 Dualities

There are various forms of these.

(i) For incidence structures of rank 2, interchange "point" and "block" as labels for the types. (This is not quite so trivial if the structure is presented as point-based.)

(ii) For point-based structures, replace each variety (other than points) by the complementary set of points.

(iii) For strongly point-based structures of rank 2 with all blocks of fiixed size k. replace the set of blocks by the complementary set of k-sets.

(iv) There is a separate and quite different duality for matroids.

5.6 Special subsets

Find a subset of a certain type, or generate all such subsets. (The things being looked for are often implicitly cliques in some graph. Examples include arcs, ovoids, spreads, packings, etc. Other examples might be subplanes of projective planes, blocking sets, etc.)

5.7 Graphs

Associated with a geometry are various graphs. Especially, if i and j are types, there is a graph whose vertices are the varieties of type i, two vertices adjacent if there is a variety of type j incident with both. This construction yields collinearity graphs, line graphs, etc. We could refine this by declaring two vertices adjacent if there is a unique variety of tpe j incident with both, or possibly some other specified number. (This is used in quasi-symmetric designs, for example.)

Having defined the graph, we want to do things like find its diameter or girth, check distance-regularity, find the spectrum of the adjacency matrix, etc. This would presumably be done with functions from another section of CAYLEY.

5.8 Isomorphisms

Test isomorphism of two geometries, or find the automorphism group of a geometry. Of course, this is a nontrivial problem. One remark is that, if it is known that a group G acts on the geometry. the algorithms should be able to use this information intelligently to reduce the problem.

6 Coding theory

After some thought, I decided to write this as if there were to be a separate module handling coding theory. There are several reasons for this: the techniques are quite different; there are a couple of different interfaces with geometry (see below); and there are links with yet other parts of CAYLEY, e.g. representation theory. I don't want to pre-empt design decisions, but I think that the designers may well find these reasons compelling. I propose to comment on what a coding theory module should do, and what connections it should have with geometry.

We should be able to do the following.

- Find the dual code. (Simple linear algebra.)
- Implement the MacWilliams transform on weight enumerators. (This is a piece of "computer algebra" in the sense of REDUCE, etc.)

The next few things are hard problems, only to be recommended for fairly small codes; but, if we know that a group acts on the code, the algorithms should use it to ease their tasks.

- Produce a word of minimum weight, or generate them all. (Finding the minimum weight is probably no easier.)
- Find the weight enumerator.
- Find the automorphism group.

Finally, if the code admits a group G, then it is a G-module, and indeed it is a submodule of the twisted permutation module. (Remember that G is a monomial group.) It should be possible to use the representation-theoretic resources of CAYLEY here.

There are two main constructions for producing a code from a geometry.

Construction A Given a rank 2 geometry with points and blocks, form its point-block incidence matrix; its row space over $\mathrm{GF}(p)$ (rows = blocks) is the *mod p code* of the geometry.

Any group acting on the geometry acts on the code by permutation matrices.

(This is incidentally, a tool for reducing permutation representations which is sometimes useful.)

The supports of codewords often carry interesting geometric structures. So we should be able to carry these sets back to the geometry.

Construction B Let S be a spanning set of points in the projective space $\mathrm{PG}(d-1,q)$, with $|S| = n$. Choose spanning vectors for the points of S, and write them as the columns of a $d \times n$ matrix A. Then the row space of A is a code with length n, dimension d, and minimum weight at least 3. The construction reverses. It has all kinds of nice properties. For example, two codes are equivalent (up to monomial transformations, as usual) if and only if the corresponding spanning sets lie in the same orbit of $\mathrm{PGL}(d,q)$. Lots of geometric information tranfers back and forth.

Some interesting codes can be defined in this way. For example, taking for S the whole projective space, we get the Hamming code.

7 Coordinatisation

Only a small class of geometries permit any sensible coordinatisation by algebraic objects. These can be divided into two types. First, there are those (essentially projective, affine and polar spaces) which are unique, and are coordinatised in an essentially unique way by a vector space, possibly with extra structure. Here, of course, knowing the coordinatisation gives an enormous amount of information. For example, to find a conic in $\mathrm{PG}(2,q)$, no backtrack search is required; just write down a quadratic form.

For this reason, it is important to be able to coordinatise these geometries, if they are presented in such a way that the coordinatisation is not apparent. This could be, either a presentation of the geometry with no automorphisms or only a small group (perhaps a Singer cycle); or a permutation group which we recognise as the group of the geometry in the correct permutation representation. The latter might, for example, be the output of an algorithm which takes a permutation group identified abstractly as a classical group, and finds its "natural" permutation representation. In this case, coordinatisation would be an important step towards making further use of the permutation representation, e.g. finding subgroups of specific kinds. These comments clearly cuuld stand a lot of further elaboration.

The most powerful method of coordinatisation uses automorphisms. The simplest case is that of an affine space. Suppose that we've recognised that the geometry is $\mathrm{AG}(n,q)$. First,w e construct the group of all translations and dilatations. (These are precisely the automorphisms which fix every parallel class of lines. If we have the automorphism group, this can be found as the kernel of a homomorphism; if not, we have to construct it by some "find automorphism group" algorithm, with the advantage that we know when we have them all!) The translation group then consists of all elements of this group with no fixed points, together with the identity; it is the additive group of the vector space of coordinates. Choose any point to identiry with 0. To identify the field, choose a line through 0, label one of its points 1; the points of the line are identified with the field elements, translations give addition, and dilatations fixing 0 give multiplication. Now the dilatation group fixing 0 gives scalar multiplication on the vector space. Furthermore, the subspace spanned by $v_1,\dots,v_k$ is the orbit of the group generated by

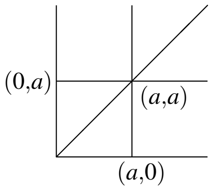
(a) the translations mapping 0 to v_i for $i=1,\dots,k$; and

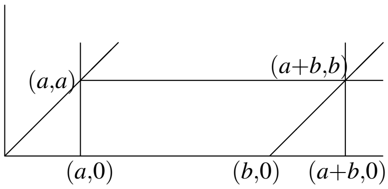
(b) all dilatations fixing 0;

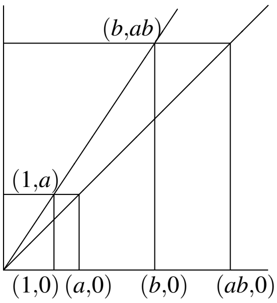
hence we can easily find a basis and set up coordinates.

For projective space, choose a hyperplane; its complement is an affine space and can be coordinatised by the above procedure, and then the original hyperplane consists of the points at infinity. Polar spaces will be a little more difficult, but an algorithm similar to the above can probably be devised. (Perhaps find a disjoint pair of totally isotropic subspaces and coordinatise the first.)

There is, of course, a completely different method of coordinatisation, which is purely combinatorial and doesn't rely on knowledge of any automorphisms; this might be more appropriate in cases where we don't have automorphisms. Take first an affine plane. Select a point to be 0, and three lines through 0 to be the x-axis, the y-axis, and the line $y = x$. The points of the x-axis will be our field elements. The three figures show how to coordinatise the y-axis, how to add, and how to multiply. In an affine space of higher dimension, we do a plane first, then select further axes and coordinatise them as in the first figure. In the mathematics, the hard work is proving consistency and verifying field axioms; but that is not necessary here.







For the other kind of coordinatisable structures, the problem is quite different. This type means chiefly projective planes, though there are some coordinatisations of generalised quadrangles. Here, the coordinatising structure (a planar ternary ring, in the case of projective planes) is just as mysterious as the original geometry. But some geometric properties are reflected in the axioms of the planar ternary ring. There may be some interest in setting up the PTR so that identities can be checked in it.

A closely related construction is that which produces, from an affine plane, a sharply 2-transitive set of permutations, or equivalently a family of mutually orthogonal Latin squares.