For MAGMA conference 8/93. *Comments greatly appreciated.* *Version of November 5, 1994*

Geometry in Computer Algebra Systems

William M. Kantor[1], University of Oregon

Originally this was to be entirely on

Geometry in computer algebra systems—especially CAYLEY/MAGMA *and* GAP

(but perhaps also at least partially in other systems such as MAPLE)—to some extent just a wish list. However, at the request of John Cannon, that will be the second of two parts: I'll start with

PART I. Some geometric algorithms (and applications).

There are a couple of ways to phrase the question I'll discuss. The first is this: suppose you have two permutations of 532,400 things that you know generate a primitive permutation group G, and you also know that $G \cong PSL(3,11)$. You want to know what 3×3 matrices over $GF(11)$ produce these generators. What would you do? Thus, having a group that is simple and familiar is *not* the same thing as having the group in a familiar format.

A better starting point involves Sylow subgroups:

Input: $G \leq S_n$ given in term of a generating set of permutations; a prime p dividing $|G|$.

Output: A Sylow p-subgroup.

(Also: Given two Sylow p-subgroups P_1, P_2 of G, find $g \in G$ with $P_1^g = P_2$.)

A standard type of Sylow algorithm proceeds roughly as follows:

Find $g \in G$ of order p, by randomly choosing elements of G until you get one of order divisible by p.

Find $C_G(g)$ using backtrack.

Recursively find a Sylow p-subgroup P of $C_G(g)$ and test elements in $Z(P) - \{1\}$ to find a nontrivial element h central in a Sylow subgroup of G. (More backtrack.)

Find the set Δ of orbits of $\langle h \rangle$, and recursively find a Sylow p-subgroup of the group induced on Δ by $C_G(h)$.

Pull back to a Sylow subgroup of G.

Related procedures using backtrack to find centralizers are presently employed in cayley and MAGMA (as well as GAP) in order to find Sylow subgroups. They work well when n isn't too big. My interest is in dealing with larger n, and using an algorithm whose efficiency can be *proven* by some means other than trial runs. It should be noted, incidentally, that starting by finding an element of order divisible by p using random choices is now rigorously proved to be efficient in a reasonable sense [IKS]; however, if the group is, say, $G \cong PSL(3,p)$, with p in the hundreds of thousands, then it may be necessary to make almost p choices, and hence take unacceptably long to find such an element in this manner.

There is an alternative approach, originally developed in a purely theoretical (polynomial time) framework [Ka2]. This involves a **reduction** to two problems:

- Find a Sylow p-subgroup of a *simple* group G.
- Given two Sylow p-subgroups of a *simple* group G, find an element of g conjugating one to the other.

[N.B.—CAYLEY, MAGMA and GAP can find elements that conjugate a given subgroup to any known conjugate subgroup, but a lot of backtrack is involved, and this is not very good for moderate sized n.]

It is important to understand that the reduction makes essential use of the ability to conjugate Sylow subgroups of simple groups: this leads to an effective form of the Frattini argument. I won't discuss this reduction. There are a few versions of it [Ka2, Ka3, KLM]. The more times it is reworked the slicker it gets. There is a parallel version of the reduction (for theoretical purposes at this stage [KLM]). There is a nearly linear time[2] version being developed (by Prabhav Morje, a PhD student of Ákos Seress, and Morje will be programming several parts of this). All situations I've been involved in follow the same pattern, reducing to the above special cases for simple groups.

Now the classification of finite simple groups gets invoked (not a big surprise).

---

[1] Supported in part by the NSF.

[2] "Nearly linear time" algorithms (studied by Babai, Luks, Cooperman, Finkelstein, Seress, Beals ....) are those requiring time $O(n \log^c n)$ for small base groups.

Also there is an easy reduction to primitive groups. Now what?

Basic idea: Switch to a better permutation domain, one where it's easy to calculate (e.g., using linear algebra).

Example. Unless you merely want to know the name of a group you're working with, you need to be able to switch to familiar surroundings. For example, suppose $G = \langle g, h \rangle \cong PSL(3, 11)$, acting on $|PSL(3, 11)|/(11^2 + 11 + 1)3 = 532,400$ Sylow 19-subgroups. As suggested before, sometimes it might be necessary to have explicit 3×3 matrices that produce g and h—in particular, in order to switch to a much smaller degree permutation representation (degree 133 in the present situation) or to write down matrices that generate a Sylow 5-subgroup, say.

Example. Suppose you have a (primitive) permutation representation of $G \cong PSL(20, 2)$, and you want a Sylow 5-subgroup of G. Conceivably some algorithm using backtrack (e.g., used in order to find some centralizers) will work adequately. But the smallest degree of a nontrivial permutation representation of G is $2^{20} - 1 = 1,048,575$. Assuming that one can deal with point stabilizers here, nevertheless any kind of backtracking is bound to be slow if it is even presently possible. Therefore, it would be preferable to revert to linear algebra: coordinatize the underlying 1,048,575-set so that each point "becomes" a vector whose 20 coordinates can be found whenever they're needed; and then construct a Sylow 5-subgroup explicitly using matrices.

Example. If $G \cong A_r$ for some r, and if $|G| > n^3$, then the n-set can be identified either with all k-sets of an r-set, or with all partitions into r/k k-sets, for some k [Ka1, (6.1); KP]. Then you can get the standard permutation representation on an r-set. At that stage you can just write down permutations that generate a Sylow p-subgroup (or that conjugate one Sylow subgroup to another).

Similar remarks apply to all the other simple groups: if G is primitive, simple, non-alternating, and $|G| \geq 2n^5$, then G is a classical group and the permutation representation is a familiar one on an orbit of subspaces; you can then get to an orbit of 1-spaces of the vector space [Ka1, (6.2); KP]. (Note that all exceptional simple groups of Lie type have been eliminated by this inequality.) Given some freedom to increase the set a bit you can then get the entire underlying projective space if needed. I'll say a little more about this soon.

There are two steps in this transition from a given permutation representation to an "effective" one:

$$n\text{-set} \longrightarrow \text{nice set} \longrightarrow \text{explicit permutations or matrices.}$$

The algorithms below and in the appendices deal with both of these. However, some basic problems remain:

1. What if your alternating or classical group is SMALL relative to n? This means that the stabilizer of a point is some small maximal subgroup (as in $PSL(3, 11)$ above). It is unlikely that all maximal subgroups of classical groups ever will be determined—and anyway, who wants to program a horrendous table?

2. What if the group is sporadic? A table will take care of writing down Sylow subgroups. But conjugating them is another matter.

3. More important and much more interesting: what if G is an exceptional simple group of Lie type? Once again I suppose you could *find* Sylow subgroups using a table, assuming that $n \leq 10^6$, say; but this would become very unpleasant if the degree had to be allowed to be as high as 10^7. Moreover, once again a database would not deal with conjugating Sylow subgroups.

Note: In polynomial time and many theoretically-directed algorithms, problems 1-3 are inconsequential [Ka1,Ka2,KLM]. This interesting mathematical question arises from practical considerations and in the context of nearly linear time algorithms.

I'll briefly discuss 1, and provide hints concerning 3: there has been significant recent progress concerning these.

Example.

Input: $G \cong PSL(d, q)$, $d \geq 5$ (in an unknown permutation representation).

Output: The permutation representation of G on the set of 1-spaces of $GF(q)^d$.

(N.B.—I've assumed that $d \geq 5$; the other cases are similar but easier.)

Find d and q using $|G|$. Let r be the prime dividing q.

Reduce to the case where G is primitive.

Randomly choose elements of G until one is found of order rt where t is a primitive divisor of $q^{d-2}-1$ (i.e., it does not divide q^i-1 whenever $1 \le i < d-2$). (There are lots of these elements: the probability is high that relatively few choices will be needed to find such an element.)

A power of this element is a transvection g.

Find the set Σ of all conjugates of g. If $|\Sigma| > n$ then the original permutation representation is awfully nice: on the set of all points or hyperplanes, lines or $d-2$-spaces, or 3-spaces when $d=6$. In the latter three situations, switch to the permutation representation on the set of all points or hyperplanes (e.g., using [KP]).

Find and output a minimal block system for G on Σ.

HISTORY: This will look familiar to some of you. I used this for an nonalgorithmic purpose several years ago (generating simple groups in an early version of [Ka4]); but I suspect the idea was known, and perhaps used, earlier than that. It does *not* require anything as hard as the classification of finite simple groups. Then came the wonderful recognition algorithm due to Neumann and Praeger [NP]. I commented to one or both of them that, at the cost of slightly worse probability estimates, they could avoid the classification. But, understandably, they liked their estimates. Then Leedham-Green [L-G] used this same method, more or less, for a version of the Neumann-Praeger algorithm—a highly effective version, I think.

Then Beals and Babai [BB] use the above method, in order to find a composition series of a matrix group. While their result was intended only for theoretical purposes, this and other parts of their methodology turn out to have some very practical implications. The application to the present situation comes from Seress and myself. It works for the other classical groups, with not much change. As above it is necessary to be careful not to get a *larger* permutation domain.

The above method *also* works in the case of the exceptional groups of Lie type, as required in problem 3 above. It produces one of the one or two nicest permutation representations on maximal parabolic subgroups.

In the case of classical groups, once you have all or enough of the projective space of the underlying vector space, you can reconstruct the vector space itself—by methods to be discussed soon—and then write down matrices or linear transformations using geometry and linear algebra. In the case of the exceptional groups of Lie type the geometry is simply not understood to the same level. No effective algorithms are known. **Example.** Given the permutation representation of $E_8(q)$ on the largest maximal parabolic (of type $E_7(q)$), find an element of order a primitive prime divisor of $q^{30}-1$ or $q^{24}-1$. Yes, you can make a number of selections and in a provably short time get such an element. Now conjugate one such cyclic subgroup to another one! This may just be a matter of time and persistence. As suggested above, it is not clear how a database would help with the problem of conjugating Sylow subgroups.

Is it entirely obvious how to write down or conjugate Sylow subgroups efficiently in the case of classical groups? I hope not: I'm glossing over a lot, even in that case (cf. [Ka2, Ka3]).

There's an analogous question for alternating groups. It's much easier: use the disjoint product of a 3-cycle and a cycle as long as possible but of length prime to 6 [BB].

COORDINATIZATION:

Example 1. If you have an "arbitrary" permutation representation of $G \cong PSL(d,q)$ on a set X, it seems natural to get the permutation representation of G on the set of all 1-spaces of $GF(q)^d$; and then to reconstruct the vector space $GF(q)^d$ as well as the linear group $SL(d,q)$ (so you can actually get a *larger new* group). That is, what is needed is a way to assign a vector to each point so that the point "is" the 1-space spanned by the vector.

This is not a big problem: there are very different algorithms for obtaining this identification. One uses classical projective geometry and the fact that X comes equipped with "lines" that can be found quickly [Ka1]. Another uses the group G, by finding and using the pointwise stabilizer of a suitable set of d points (see Appendix 1).

It is important to keep in mind that, even though you *have* a set that can be identified with the set of 1-spaces of a vector space, you do not automatically *have* such an identification.

Example 2. Here's a more interesting example.

Input: A group G isomorphic to a simple symplectic, orthogonal or unitary group defined by a d-dimensional vector space ($d \geq 5$) over a finite field F; and a permutation representation of G on a set X permutation isomorphic to its action on the set of all isotropic or singular points of that vector space.
(Not available: the vector space itself—in fact, that vector space, and the set of all its points, may be too big to be written down.)

Output: An injective "labeling" map $\lambda: X \to F^d$ that preserves "perpendicularity", allows the transfer of the action of G on X to an action on a set of 1-spaces of F^d, and is such that all images of λ have last nonzero coordinate 1.

Perpendicularity? If $x \in X$ let $x^\perp$ consist of x together with the shortest nontrivial orbit $\Delta(x)$ of G_x on X. Then "perpendicularity" in X refers to the relation "$y \in x^\perp$".

A procedure for this is given in Appendix 2. The key to the procedures in both appendices is the use of an r-group (where is r is the characteristic) that can readily be computed using G in order to label enough of X using vectors, after which labeling the rest of X is just a matter of bookkeeping.

How about labelling *all* of the vector space underlying all of this? After all, it's not so hard to find the set of nonsingular points of this space, and then label them.... The snag is that there are about $|F|$ (in the orthogonal case) or $\sqrt{|F|}$ (in the unitary case) times as many nonsingular points as there are isotropic points, and this may be an unacceptable increase in the number of points being permuted (at least undesirable in the practical setting and definitely unacceptable in the nearly linear time setting).

PART II. The following "wish list" is culled from correspondence with a number of mathematicians.

1. TRANSITIONS.

This is what I've just been discussing in a special case. Procedures are needed that allow painless transitions between geometry and group theory: transitions involving permutation groups, coset geometries, building-like geometries; and between graphs and groups and codes and designs. Of course, some of these may be pretty hard to implement. I'm thinking in terms of keeping track of where each set or group came from, and applying procedures to the new objects before translating back into the original setting.

Example. Consider the following transitions:

$$M_{24} \longrightarrow \text{Steiner system} \longrightarrow \text{Golay code } C \longrightarrow \text{semidirect product of } C \text{ and } M_{24}.$$

For group-theoretic applications one needs to be able to keep track of the original 24-set; having the Golay code and matrices generating the Mathieu group may well not be enough. For example, if a presentation of the semidirect product is needed, then one might want to start with a presentation of M_{24} available on the computer, but then to form the semidirect product would require knowing how the generators act on the Golay code or at least on the Steiner system. In other words, a presentation alone doesn't tell you where the generators or relations came from, and that information may be needed.

Similarly, just because you have a presentation for $PSp(10, 2)$ doesn't mean you can easily obtain a presentation for the semidirect product of this group with the 10-dimensional vector space; or of the semidirect product with the 1023-dimensional permutation module.

2. GRAPHS AND GEOMETRIES.

Most finite geometry can be phrased in terms of graphs, especially *multipartite graphs*. While this may not be a very intuitive way to view geometry, it is well-suited for computers—and allows the use of nauty, for example. Of course, I'm really thinking of graphs having algebraic descriptions, which frequently means some sort of description using cosets of some subgroups of a given (automorphism) group. GRAPE seems to be designed to do geometry in this type of framework.

Example. The building for $PSL(5, 3)$ can be viewed as a 4-partite graph, the vertices being the i-spaces of $GF(3)^5$ for $i = 1, 2, 3, 4$. Two vertices are joined if one of the corresponding subspaces contains the other.

In fact, there are quite a few important graphs associated with the geometry of the classical groups. Some should be readily available, at least within a database (see 6 below).

One should be able to check a variety of geometric properties of geometries—such as properties motivated by buildings. This means that the original way a geometry is input using an automorphism group may wind

up being ignored, replaced instead by a geometric description. But of course it may well be necessary to go back and forth between these descriptions (i.e., permutation representations).

It is also important to be able to quickly determine or check various standard properties: girth, diameter, eigenvalues, Krein parameters. Similarly, it should be easy to use a computer system in order to check necessary arithmetic conditions for a graph or geometry to exist. For example, one should be able to quickly check all known feasibility conditions for the parameters of a potential generalized quadrangle before starting to try to construct one, something I've often had to do.

For many purposes one needs not just a multipartite graph but the simplicial complex of all or some cliques of the graph. One of the main questions here—very useful for some aspects of group theory—is whether the complex is simply connected or not. This leads to questions about presentations of fundamental groups, and this may be too difficult to computerize for all but very small instances. But I would hope that something like GRAPE will do reasonably well on questions like this.

One should be able to quickly determine the p-rank of an incidence matrix M: the rank of the Z_p-space (code) spanned by the rows of M mod p; and to find all small weight codewords, and determine their geometric properties (i.e., relationships to the rows of M). A small part of a database would be helpful here: lists of known p-ranks for standard types of geometries (see 6 below).

Once again this requires fast transitions, for example from groups to coset geometries to incidence matrices; and then also to group representations (for the permutation representations in hand and incidence matrices intertwining them).

3. ISOMORPHISM/AUTOMORPHISMS.

Something like nauty seems essential. (The fact that its complexity is exponential does not change this need.)

It would helpful to be able to go further, in order to solve problems such as: Given a subset of a projective, vector or affine space, find its automorphism group. One might merely be given the coordinates of the points in the subset and no further structure.

This also leads to orbit questions for a linear or projective group G on subspaces of suitable dimensions. How many orbits are there? What are representatives? What is the length of the orbit containing a specified subspace? Are two subspaces in the same orbit? These are standard permutation group problems, and are not so easy in that context (in theory, they're probably intractable); but in the vector space setting they are quite a bit harder still—as can be seen by considering just orbits on vectors when the vector space is very large. It might be possible to get at larger subspaces via exterior powers, but that seems inefficient, dubious and even unworkable if large dimensions are involved (imagine trying to look at ordered triples of vectors in order to construct orbits on 3-spaces).

Similarly: For small projective spaces $PG(V)$, be able to classify under the action of $P\Gamma L(V)$ all standard types of structures in $PG(V)$, such as k-arcs, blocking sets, etc.

4. BACKTRACK?

I don't think systems like CAYLEY, MAGMA, GAP, MAPLE ..., are the correct places to *search* for projective planes, translation planes, generalized quadrangles, ovals, etc., by exhaustive backtrack searches. Such questions seem better suited to special purpose programs. However, these types of backtrack questions are appropriate *next to* one of these systems, as is the situation with GRAPE.

Some backtrack searches have, of course, been made using CAYLEY or GAP: it's easier to learn these systems than to figure out how to implement groups, fields, etc., in a standard programming language. But it's very hard to imagine their being useful for BIG backtrack searches.

5. ALGEBRAIC MANIPULATION. This includes the following:

• Formal algebraic manipulations and checks (for small explicit fields) that requirements such as spread or generalized quadrangle or ovoid or cap conditions hold for a *given* set of subspaces or subgroups.

• Likewise for difference sets (especially ones producing symmetric designs with $\lambda > 1$), and transitions to number-theoretic settings for their study and construction.

• Testing whether a given polynomial is a permutation polynomial: there doesn't yet appear to be a provably efficient algorithm for this.

6. DATABASES.

These are needed for interesting types of geometries, and can be especially useful when working with objects outside the area of your own expertise. Several individuals have such databases, but very much tailored to their own interests.

Needed:

- The known projective planes of small order, their collineation groups, statistics on subplanes and ovals, etc.
- Likewise for generalized polygons and related geometries.
- Geometries related to sporadic groups.

  This is especially natural for a system built starting with group theory. This can begin with geometric information in the ATLAS, for example; but that is far from enough. Should *everyone* be asked for details about her own favorite examples of "important" small geometries? I doubt it, this is more appropriate for more specialized add-on databases (e.g., one in Giessen aiming at an "atlas" of all linear spaces with at most 20 or so points). What matters most in the present context seems to be those geometries having some potential group-theoretic aspects. But of course, that is a matter of taste.
- Unusual embeddings of groups in classical groups, together with their geometric properties (involving small orbits of various subspaces).
- Again nicely related to groups: distance-regular graphs.

Another large database can contain information *about* standard structures (instead of the structures themselves): parameters for which existence or nonexistence is known, and references to the literature. **Examples:** For what geometries is it known that no ovoids exist? In what geometries are ovoids known; what parameters do they have, what are their automorphism groups and other invariants.

Literature seems a very important aspect—not encylopedic, but with pointers to the most important sources, especially survey papers where more references can be found. Keeping this up to date, both here and in other databases, might be a nuisance, however.

APPENDIX 1.

Some aspects of this appendix arose in discussions with R. Liebler.

Input: A group $G \cong PSL(d,q)$; a permutation representation of G on a set X, permutation isomorphic with the representation on the set of all points (1-spaces) of a d-dimensional vector space over $GF(q)$.

Output: An injective "labeling" map $\lambda: X \to GF(q)^d$ that preserves collinearity, allows the transfer of the action of G on X to an action on the set of all 1-spaces of $GF(q)^d$, and is such that all images of λ have last nonzero coordinate 1.

Note: Here "collinearity" refers to the fact that there is such a relation implicit in the given description of X. The exact nature of this will appear in the part of the procedure referring to $G_{yy'}$-orbits (namely, the line through any two points y and y' is the union of all $G_{yy'}$-orbits of length $< q$).

Let $x_1 \in X$. Pick $x_2, \ldots, x_{d+1}$ so that x_{i+1} lies in the longest orbit of $G_{x_1 \ldots x_i}$ for each i. Then $B := \{x_1, \ldots, x_{d+1}\}$ is a base for G on X.

Find $Q := O_r(G_{x_1 \ldots x_{d-1}})$, where r is the characteristic. (Here $G_{x_1 \ldots x_{d-1}}/Q$ is abelian: the direct product of $d-1$ copies of F^*.) Find the set Y of fixed points of Q.

Use B to find $t, c \in G$ acting on $\{x_1, \ldots, x_d\}$ as the permutations (x_1, x_d) and $(x_1, \ldots, x_d)$, respectively.

Let $H = \langle Q^t, Q^{tc}, Q^{tc^2}, \ldots, Q^{tc^{d-2}} \rangle$ (which is isomorphic to $SL(d-1,q)$).

Find $J := \langle H, G_{x_1 \ldots x_{d-1}} \rangle$ and $Z(J)$.

Q is an elementary abelian group, and J acts on it as a group of automorphisms (J induces $GL(d-1,q)$ while H induces $SL(d-1,q)$). The set of automorphisms induced by $Z(J)$, together with the 0 endomorphism, is a field $F \cong GF(q)$; identify these two fields. When written additively, Q becomes an F-space. Find a basis of Q in order to identify this group with F^{d-1}. Also, identify $F^{d-1} \times F$ with F^d.

Let $y = x_d$. Label $(y^g)^\lambda = (g, 1)$ for each $g \in Q$.

(Since Q is regular on $X - Y$, it only remains to label Y.)

y is labeled $(0,1)$. Let y' be the point labeled $(e_1, 1)$, where $e_1 = (1, 0, \ldots, 0) \in F^{d-1}$.

Find the unique point z of Y in the union of all $G_{yy'}$-orbits of length $< q$.

Find a complete set R of coset representatives of H/H_z.

For each $h \in R$, label z^h as $(e_1^h, 0)$, normalized so that the last nonzero coordinate is 1 (i.e., $(z^h)^\lambda$ is $(e_1^h, 0)$ normalized by multiplying through by a suitable scalar).

Collinearity is preserved by this definition. Namely, H was defined to fix all vectors in the 1-space y, and hence leaves the label of y unchanged in the above process of labeling all points of Y. In fact, we have merely mimicked within X aspects of the projective space $PG(F^d)$, so the action of G transfers as required.

APPENDIX 2.

Find the unique minimal block system of G_x on $\Delta(x)$, find its pointwise stabilizer K in G_x, then find $Q := O_r(K)$, where r is the characteristic. (There are other ways to find Q, e.g. as $O_r(G_x)$, but the present way is faster: K/L is cyclic, and in fact is isomorphic to a subgroup of F^*. Note that Q acts regularly on $X - x^\perp$; we will ultimately identify Q and $X - x^\perp$ via $g \mapsto y^g, g \in Q$.)

Find the pointwise stabilizer T of $\Delta(x)$ in G_x.

(We now construct an isomorphic copy of Q that is itself coordinatized—i.e., is given in terms of matrices.)

The name of G is known. Equip F^d with the appropriate form (define it on a basis), in such a way that $e_1 := (10\ldots0)$ and $f_1 := (00\ldots1)$ are isotropic or singular and $(e_1, f_1) = 1$. Find the group Q_0 of all isometries of F^d that fix e_1 and induce 1 on $e_1^\perp/\langle e_1\rangle$, and the group T_0 of transvections in Q_0 that fix each vector of $e_1^\perp$. Explicitly, Q_0 consists of the following maps, where $u \in \langle e_1, f_1\rangle^\perp$ and $c \in F$ are arbitrary except that $c + \overline{c} = -(u,u)$ in the unitary case:

$e_1 \mapsto e_1$, and $\forall w \in \langle e_1, f_1\rangle^\perp$,

PSp: $w \mapsto w - (w,u)e_1$, $f_1 \mapsto f_1 + u + ce_1$; T_0 consists of the maps having $u = 0$.

$P\Omega$: $w \mapsto w - (w,u)e_1$, $f_1 \mapsto f_1 + u - \varphi(u)e_1$; $T_0 = 1$.

PSU: $w \mapsto w - (w,u)e_1$, $f_1 \mapsto f_1 + u + ce_1$; T_0 consists of the maps having $u = 0$.

Then $Q_0 \cong Q$. Moreover, Q_0/T_0 naturally inherits the vector space structure of $\langle e_1, f_1\rangle^\perp$, as well as its form. (N.B.—Q is either elementary abelian or, in a sense, "extraspecial": it is extraspecial if $T_0 = Z(Q_0)$ and $|T_0| = r$.)

(We now need to produce an *explicit* identification of Q_0 and Q.)

Find $H := (G_{xy})'$. This is isomorphic to a classical group on $\langle e_1, f_1\rangle^\perp$, and acts on Q/T faithfully as the classical group on $\langle e_1, f_1\rangle^\perp$. The centralizer in $End(Q/T)$ of the group induced by H on Q/T is isomorphic to F. Find this centralizer, identify it with F, and use it to turn Q/T into an F-space. Use this action to find an H-invariant nonsingular form on Q/T of the same type as that on F^d. (See [Ka2; KP]. Note that we really only need the form on a mostly-hyperbolic basis we are about to use.)

Find mostly-hyperbolic bases B of Q/T and B_0 of Q_0/T_0 in order to produce an isometry $\theta: Q/T \to Q_0/T_0$ sending B to B_0. For each b in B or B_0 let $\hat{b}$ be a preimage in Q or Q_0, respectively. Define $\hat{\theta}$ by $\hat{b}^{\hat{\theta}} = \widehat{b^\theta}$ for each $b \in B$. Then $\hat{\theta}$ extends to a unique isomorphism $\pi: Q \to Q_0$; find π (recall that $|Q| < |X|$).

Let $(y^g)^\lambda := f_1^{g^\pi}$ for $g \in Q$.

Thus, using the description of Q_0 given above, each point of $X - x^\perp$ can be viewed as labeled by a vector of the form $(c, u, 1)$ with $c \in F$ and $u \in \langle e_1, f_1\rangle^\perp$; here, $\langle e_1, f_1\rangle^\perp$ is identified with F^{d-2} and $F \times \langle e_1, f_1\rangle^\perp \times F$ with F^d.

Recall that H acts on Q/T, and this action induces an action of H^π on Q_0/T_0 and hence on $\langle e_1, f_1\rangle^\perp$; and the H^π-modules Q/T, Q_0/T_0 and $\langle e_1, f_1\rangle^\perp$ are isomorphic. Moreover, since H was defined so that H^π fixes all vectors in the 1-spaces coresponding to x and y,

• *Any $h \in H$ sends the point of $X - x^\perp$ labeled $(c, u, 1)$ to the point labeled $(c, u^{h^\pi}, 1)$.*

(At this stage we have labeled all points of $X - x^\perp$ in the desired manner. It remains to determine λ on $x^\perp$.)

Recall that $x^\lambda = (1,0,0)$ and $y^\lambda = (0,0,1)$. Let $y' \in X - x^\perp$ be labeled $(0, e_1', 0)$ with $e_1' = (10\ldots0) \in \langle e_1, f_1\rangle^\perp$ (using only $d-2$ coordinates!).

Define z in either of the following two ways—producing the exact same point of $x^\perp$:

• $(y^\perp \cap y'^\perp)^\perp \cap x^\perp$; or

• the unique point of $x^\perp$ in the union of all $G_{yy'}$-orbits of length $< |F|$.

(Which of these definitions to use is a matter of taste or timing; this computation is made just *once*. We are going to label z as $(0, e_1', 0)$, and then extend this to all of $\Delta(x)$.)

Find a complete set R of coset representatives of H/H_z.
Find a complete set S of coset representatives of Q/Q_z (here $|S| = |F|$).
Let $y' = y^k$ with $k \in Q$.
For each $g \in S$ and $h \in R$

y^g is labeled $(y^g)^\lambda = f_1^{g^\pi} = (c, u, 1)$ for some $c \in F$ and $u \in \langle e_1, f_1 \rangle^\perp$ as above

y'^g is labeled $(y'^g)^\lambda = f_1^{k^\pi g^\pi} = f_1^{g^\pi} + (0, e_1', 0)^{g^\pi}$
$= (c + a, u + e_1', 1)$ for some $a \in F$

y^{gh} is labeled $(c, u^{h^\pi}, 1)$

y'^{gh^π} is labeled $(c + a, u^{h^\pi} + e_1'^{h^\pi}, 1)$

so label z^{gh^π} as $(a, e_1'^{h^\pi}, 0)$, normalized so that the last nonzero coordinate is 1 (i.e., $(z^{gh^\pi})^\lambda$ is $(a, e_1'^{h^\pi}, 0)$ normalized by multiplying through by a suitable scalar).

It is easy to see that this labeling preserves perpendicularity, since it is mimicking what happens in F^d.

Remark on H. Note that it might have been better to have defined H as $(G_x)'_y$, thereby producing the same group.

Remark on fields. In both appendices it was assumed that the field F is available at the outset. In case this field is not already available, note that a copy of F was actually constructed in the course of each algorithm.

For example, consider the present appendix. Construct Q, T, H, and the centralizer in $End(Q/T)$ of the group induced by H on Q/T. This centralizer can then be used as the *definition* of F.

References

[BB] R. Beals and L. Babai, Las Vegas algorithms for matrix groups, to appear in Proc. 34th IEEE FOCS.

[IKS] I. M. Isaacs, W. M. Kantor and N. Spaltenstein, On the probability that a group element is p–singular (to appear in J. Algebra).

[Ka1] W. M. Kantor, Polynomial-time algorithms for finding elements of prime order and Sylow subgroups. J. Algorithms 6 (1985) 478–514.

[Ka2] W. M. Kantor, Sylow's theorem in polynomial time. J. Comp. Syst. Sci. 30 (1985) 359–394.

[Ka3] W. M. Kantor, Finding Sylow normalizers in polynomial time. J. Algorithms 11 (1990) 523–563.

[Ka4] W. M. Kantor, $1\&\frac{1}{4}$-generation of classical groups (in preparation).

[KLM] W. M. Kantor, E. M. Luks and P. D. Mark (in preparation).

[KP] W. M. Kantor and T. Penttila, Reconstructing simple group actions (in preparation).

[L-G] C. Leedham-Green, Lecture at Oberwolfach, June 1992.

[NP] P. M. Neumann and C. E. Praeger, A recognition algorithm for special linear groups, Proc. LMS 65 (1992) 555–603.