Acta Cryst. (1997). A**53**, 467–474

Computing Maximal Subgroups and Wyckoff Positions of Space Groups

BETTINA EICK,[a] FRANZ GÄHLER[b]* AND WERNER NICKEL[c]

[a]*Lehrstuhl D für Mathematik, RWTH Aachen, D-52056 Aachen, Germany,* [b]*Centre de Physique Théorique, Ecole Polytechnique, F-91128 Palaiseau, France, and* [c]*School of Mathematical and Computational Sciences, University of St Andrews, St Andrews, Fife KY16 9SS, Scotland. E-mail: gaehler@pth.polytechnique.fr*

(*Received 8 October 1996; accepted 3 March 1997*)

Abstract

This paper describes algorithms for the computation of conjugacy classes of maximal subgroups and the Wyckoff positions of a space group. The algorithms are implemented in the computational group-theory system GAP and use existing standard functions of GAP as well as some simple but useful group-theoretical ideas.

1. Introduction

Traditionally, information about space groups has been made available in the form of lists, for example in *International Tables for Crystallography* (1995) or the list of all four-dimensional space groups (Brown, Bülow, Neubüser, Wondratschek & Zassenhaus, 1978). Space groups in dimensions higher than three have important applications as symmetry groups of quasiperiodic structures, such as incommensurately modulated crystals (Janssen & Janner, 1987; Janssen, Janner, Looijenga-Vos & de Wolff, 1992) and quasicrystals (Janot, 1994; Gratias & Hippert, 1994). However, compiling large lists of information poses many practical problems and is often impossible. A solution to this is to compute the desired information when the need arises. For this, algorithms are needed that can answer specific questions about a given space group. Because the computations involved are often too complex for hand calculations, it is convenient to have those algorithms in the form of computer programs.

Many group-theoretical algorithms are nowadays available as part of group-theory systems. In order to have these algorithms available, we have built our programs on top of GAP (GAP, 1997), a system that is freely availabe from a number of host machines around the world. GAP already contains a variety of algorithms ready to use, so that we can avoid reinventing the wheel. Moreover, the current distribution of GAP also contains the complete list of all three- and four-dimensional space groups, as well as the lists of all irreducible and maximal finite (i.m.f.) integral matrix groups of all dimensions up to 24, as determined by W. Plesken and collaborators (see *e.g.* Nebe & Plesken, 1995, and references therein). In prime dimensions and in all dimensions up to 11, these integral matrix groups are available as $\mathbb{Z}$-class representatives, whereas in the remaining dimensions only $\mathbb{Q}$-class representatives are available. These lists of crystallographic groups provide plenty of material to which our algorithms can be applied.

The algorithms described in this paper compute the Wyckoff positions and the maximal subgroups of any given space group, independently of its dimension. In practice, their efficiency is good enough to compute conjugacy classes of maximal subgroups or the Wyckoff positions of all three- and four-dimensional space groups on an average personal computer in little more than an hour (see §6). They can be seen as a complement to the information contained in the space-group tables of GAP. In addition, an implementation of the Zassenhaus algorithm is also made available, which allows space groups to be determined in dimensions larger than four. The point groups needed as input can be determined from the list of maximal integral matrix groups contained in GAP.

An implementation of these algorithms is distributed as a GAP share package together with GAP Version 3.4.4 (GAP, 1997). This package contains the algorithms as GAP code, manual pages and examples. It also provides a variety of other functions for space groups not mentioned here, so that almost the same functionality that GAP offers for other types of groups, *e.g.* permutation groups, is now available also for space groups.

A different algorithm for computing the Wyckoff positions of a space group has been given by Fuksa & Engel (1994). Their implementation performs well on examples with small point groups. For examples with larger point groups, however, our method seems to be more efficient as the timings given in §6 indicate.

A package for computations with space groups is also being developed at the University of Nijmegen (Thiers, Ephraim, Janssen & Janner, 1993). Results obtained with this package, notably the Wyckoff positions for space groups up to dimension 4, can be obtained in electronic form on the World Wide Web (Thiers, Ephraim & de Hilster, 1996).

The remainder of the paper is organized as follows. §2 describes the mathematical set-up and recalls some basic facts about space groups used throughout the paper. The computational methods used together with short explanations are given in §3. §4 explains the algorithm for the computation of Wyckoff positions and

Acta Crystallographica Section A
ISSN 0108-7673 © 1997

§5 the algorithm for computing conjugacy classes of maximal subgroups. Finally, to give an impression of their use and their efficiency, examples and timings of their implementation in GAP are given in §6.

2. Mathematical set-up

In this section, we give a short overview on space groups to fix the concepts and the notation. We also refer to the introductory chapter of Brown *et al.* (1978) and to the article of Wondratschek (1995) in *International Tables for Crystallography.*

A d-dimensional space group S can be regarded as an affine group acting on d-dimensional Euclidean space $\mathbb{E}$. With respect to a fixed basis of the translation subgroup T of S, the conjugation action of S on T induces a homomorphism from S into $GL(d, \mathbb{Z})$. The kernel of this homomorphism is T itself; its image P is called the point group* of S, which is a finite subgroup of $GL(d, \mathbb{Z})$.

Each element of $s \in S$ can be represented by a matrix of the form

$$\left[\begin{array}{c|c} M_s & 0 \\ \hline \mathbf{t}_s & 1 \end{array}\right],$$

where M_s is an element of $GL(d, \mathbb{Z})$ and $\mathbf{t}_s \in \mathbb{Q}^d$. In this representation, elements of the translation subgroup are matrices of the form

$$\left[\begin{array}{c|c} I_d & 0 \\ \hline \mathbf{t} & 1 \end{array}\right],$$

where I_d is the d-dimensional identity matrix and $\mathbf{t} \in \mathbb{Z}^d$. We call this representation of S a *standard representation* of S. In crystallographer's language, this corresponds to a setting with a primitive cell.

Such a representation of S defines an action (from the right) of S on the row space $\mathbb{Q}^d$ as affine maps *via*

$$\mathbb{Q}^d \times S \to \mathbb{Q}^d$$
$$\mathbf{v}s \mapsto \mathbf{v}M_s + \mathbf{t}_s.$$

The elements of the translation subgroup T of S act as translations on $\mathbb{Q}^d$. It is convenient to write an element s of S as a pair $(M_s, \mathbf{t}_s)$.

Note that our convention of acting from the right differs from the convention in *International Tables for Crystallography* (1995), where the action is from the left on column vectors. We use this convention in order to maintain compatibility with the rest of GAP, where group action is always from the right. Moreover, our notation does not distinguish between points in Euclidean space, row vectors and column vectors.

The structure of a space group is best summarized by the sequence of homomorphisms

$$0 \to T \to S \to P \to I_d.$$

This sequence is exact: the image of each of these homomorphisms is identical to the kernel of the following homomorphism. This structure of a space group is frequently used in many of our algorithms. In particular, the homomorphism $h : S \to P$ with kernel T is of primary importance in the problems dealt with in this paper.

3. Computational methods

This section explains the computational background for §§4 and 5. For definitions and explanations of basic group-theoretical terms, the reader is referred to standard text books on group theory, for example Suzuki (1982). The two algorithms described in this paper take as input a set of generating matrices of a space group in standard representation. From this, the point group in the form of an integral matrix group can easily be extracted. It will prove convenient, however, to use also other representations of the point group. GAP has built-in facilities that allow one to switch easily between different representations. In particular, we can convert the point group P into:

(*a*) a permutation group;

(*b*) a finite presentation of P;

(*c*) a power-commutator presentation if P is solvable.

These facilities allow one to make use of algorithms that are much more efficient than those available for matrix groups. The subgroup lattice of the point group, for instance, is better computed for an isomorphic permutation group, and then the result is translated back into the matrix group. This is, in fact, the primary advantage of building our programs on top of GAP, instead of writing a stand-alone package: there is a large variety of efficient state-of-the-art algorithms available in GAP that are ready to use.

In particular, our programs make use of routines built into GAP to compute:

(*a*) the conjugacy classes of maximal subgroups of a finite group;

(*b*) the conjugacy classes of subgroups of a finite group;

(*c*) maximal submodules of a finite module for a finite group;

(*d*) orbits of points in a set under the action of a group.

In addition, we extended GAP's capabilities by implementing, in the GAP-programming language, functions for the computation of:

(*a*) complements to elementary abelian normal subgroups in a finite group;

* Note that in crystallography the term 'point group' often denotes a whole $\mathbb{Q}$ class of matrix groups; here, however, a point group is a fixed representative of such a $\mathbb{Q}$ class.

(*b*) all solutions of a system of linear equations in $\mathbb{Q}/\mathbb{Z}$.
The complement routine essentially uses the method described by Celler, Neubüser & Wright (1990).

To compute the solutions (mod $\mathbb{Z}$) of a system of linear equations $M\mathbf{x} = \mathbf{b}$, where $M \in \mathbb{Z}^{m\times n}$, $\mathbf{b} \in \mathbb{Q}^m$ and $\mathbf{x}$ is a column vector with n unknowns, we first determine unimodular matrices P and Q such that $PMQ = D$ is a matrix in diagonal form. This amounts to computing the Smith normal form including transformation matrices [see *e.g.* Cohen (1993), ch. 2, or Sims (1994), ch. 8]. The set of all solutions (mod $\mathbb{Z}$) of the system $D\mathbf{x} = P\mathbf{b}$ is then obtained as follows. Let $d_1, \ldots, d_n$ be the diagonal entries of D and $v_1, \ldots, v_m$ the components of $P\mathbf{b}$. The system $D\mathbf{x} = P\mathbf{b}$ has solutions if and only if $v_i = 0$ whenever $d_i = 0$. If solutions exist, the set of solutions for $D\mathbf{x} = P\mathbf{b}$ is described by

$$x_i \in \{0, (1/d_i), \ldots, (d_i - 1)/d_i\} + v_i/d_i \quad \text{if} \quad d_i \neq 0$$

and

$$x_i \in \mathbb{Q} \quad \text{otherwise.}$$

The set of solutions of the system $M\mathbf{x} = \mathbf{b}$ is now obtained from the solutions of $D\mathbf{x}' \equiv PMQ\mathbf{x}' = P\mathbf{b}$, *via* $\mathbf{x} = Q\mathbf{x}'$.

4. Computing Wyckoff positions

In this section, we describe an algorithm for the computation of Wyckoff positions of a space group. In the following, let S be a space group, T its translation subgroup and $\mathbb{E}$ the corresponding d-dimensional Euclidean space.

Definition 4.1. The *stabilizer* (or *site-symmetry group*) of a point $\mathbf{x} \in \mathbb{E}$ is the subgroup of those elements of S that leave $\mathbf{x}$ invariant. A *Wyckoff position* for S is an equivalence class of all points $\mathbf{x} \in \mathbb{E}$, whose stabilizers are conjugate subgroups of S.

Let A_0 be the subset of those points having as stabilizer a fixed representative of a conjugacy class of subgroups. Then A_0 is contained in some affine subspace A of E, such that it forms an open, dense subset in A. The points in $A \setminus A_0$ have stabilizers that are larger than the stabilizer of the points in A_0. The full Wyckoff position is given by the space-group orbit of the set A_0. Clearly, all points in such an orbit have conjugate stabilizers. Conversely, if $\mathbf{y}$ is a point whose stabilizer is conjugate to the common stabilizer of the points in A_0, there exists a space-group element that maps $\mathbf{y}$ into A_0. A Wyckoff position therefore is completely specified by the representative set A_0 of a space-group orbit.

In the following, in line with *International Tables for Crystallography* (1995) [see, in particular, the article of Wondratschek (1995)], we shall not distinguish between the subset A_0 and the full affine subspace A, tacitly admitting that the points in some subset of lower dimension may have a larger stabilizer.

Definition 4.2. An affine subspace of $\mathbb{E}$ is in a *special position* if its point-wise stabilizer in S is non-trivial.

With the exception of the Wyckoff position with trivial stabilizer, a Wyckoff position therefore consists of a space-group orbit of an affine subspace in a special position.

Definition 4.3. Let U be a subgroup of S. An affine subspace $A \subseteq \mathbb{E}$ is *fixed by U modulo T* if for all $u \in U$ there exists an element $t_u \in T$ such that $\mathbf{x}u = \mathbf{x}t_u$ for all $\mathbf{x} \in A$.

An affine subspace $A \subseteq \mathbb{E}$ is fixed by U modulo T if the orbit of A under T is fixed as a set by U.

Lemma 4.4. An affine subspace $A \subset \mathbb{E}$ is in a special position if and only if A is fixed modulo T by some subgroup $T < U \leq S$.

Proof. Let $A \subseteq \mathbb{E}$ be an affine subspace in a special position and let $V \leq S$ be its stabilizer in S. Then $U = VT$ fixes A modulo T.

Let U be a subgroup of S properly containing T, let $\{u_1, \ldots, u_k\}$ be a generating set for U and let $A \subseteq \mathbb{E}$ be fixed by U modulo T. Furthermore, choose $t_1, \ldots, t_k \in T$ such that $\mathbf{v}u_i = \mathbf{v}t_i$ for all $\mathbf{v} \in A$. Then the subgroup V generated by $u_1t_1^{-1}, \ldots, u_kt_k^{-1}$ stabilizes A pointwise and is not trivial. Hence A is in a special position. □

The translation subgroup has no fixed points. Therefore, a subgroup of S that has a fixed point intersects T trivially. Note that V in the second part of the proof is a complement to T in the subgroup VT. The situation can be summarized by the diagram in Fig. 1.

Let U in the previous lemma be given by a generating set $(M_1, \mathbf{t}_1), \ldots, (M_k, \mathbf{t}_k)$ in standard representation. A point $\mathbf{v}_1 \in \mathbb{R}^d$ can be mapped to another point $\mathbf{v}_2 \in \mathbb{R}^d$ by a translation in S if and only if their components differ by an integer or, in other words, if $\mathbf{v}_1 \equiv \mathbf{v}_2$ modulo $\mathbb{Z}$.

Computing the fixed points modulo T amounts to solving the following system of linear equations for $\mathbf{v}$

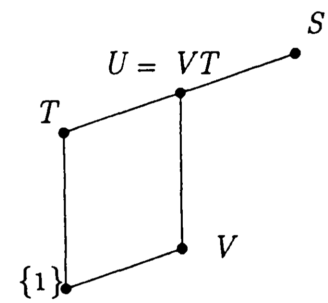

Fig. 1. The stabilizer V of any point x intersects the translation subgroup T trivially.

modulo $\mathbb{Z}$:

$$\mathbf{v}(M_i - I_d) = -\mathbf{t}_i \qquad \text{for } 1 \leq i \leq k.$$

Transposing each equation, we obtain a system

$$M\mathbf{x} = \mathbf{b}$$

with an integer matrix M and a rational vector $\mathbf{b}$ for which we seek solutions modulo $\mathbb{Z}$. The set of solutions of this system is a finite union of affine subspaces of $\mathbb{Q}^d$, containing one representative of each T orbit of affine subspaces left invariant by U modulo T. Each affine subspace is in special position. For each subspace A let $\mathbf{u}_i \in \mathbb{Z}^d$ be such that $\mathbf{v}M_i + \mathbf{t}_i = \mathbf{v} + \mathbf{u}_i$ for all $\mathbf{v} \in A$. Then the elements $(M_1, \mathbf{t}_1 - \mathbf{u}_1), \ldots, (M_k, \mathbf{t}_k - \mathbf{u}_k)$ generate a subgroup $V \leq S$ fixing A pointwise. The subgroup V is the pointwise stabilizer of A in U and a complement to T in U. Note, however, that the full pointwise stabilizer of A in S can be larger than V. This has to be checked afterwards, by looking at the length of the space-group orbit modulo T.

The main idea in computing the Wyckoff positions of S is to run through all subgroups (up to conjugacy) of S containing T and for each such subgroup U to compute the set of affine subspaces of $\mathbb{E}$ fixed modulo T by U. By lemma 4.4, this gives all affine subspaces in special positions, together with their point-wise stabilizers.

Algorithm 4.5. To compute the Wyckoff positions of S:

(i) Compute a set of representatives for the conjugacy classes of subgroups of S/T and determine the set $\mathcal{U}$ of their complete pre-images in S.

(ii) For each $U \in \mathcal{U}$:

(*a*) determine the set of affine subspaces fixed by U (modulo T), one representative for each T orbit of such spaces;

(*b*) eliminate multiple subspaces in the same S orbit (modulo T);

(*c*) retain only those spaces for which the S orbit (modulo T) has size $|S/U|$. Other subspaces must have a larger stabilizer and will show up a second time for a different U;

(*d*) for each of the remaining subspaces, determine its point-wise stabilizer in U.

When $\mathcal{U}$ is exhausted, we have obtained representative affine subspaces of all Wyckoff positions of S, together with their point-wise stabilizers.

The algorithm described above is dimension independent and works, given sufficient resources, for arbitrarily large groups and dimensions. In practice, it is mainly limited by the size of the point group and the complexity of its subgroup lattice. If the point group is too big to compute its subgroup lattice completely, but some subgroups of the point group are known, one can still compute a set $\mathcal{U}$ containing the pre-images in S of these subgroups of the point group, and go with this information directly into step (ii). One then obtains all affine subspaces stabilized by some $U \in \mathcal{U}$ modulo T, and from these the corresponding Wyckoff positions. In this way, one can then obtain at least some partial information on the Wyckoff positions of S.

5. Computing conjugacy classes of maximal subgroups

In this section, an algorithm to compute conjugacy classes of maximal subgroups of a given space group will be introduced. First we have to examine maximal subgroups of space groups from a more theoretical point of view. The following definition will be helpful.

Definition 5.1. Let U be a subgroup of the space group S and let T be the translation subgroup of S.

(*a*) If $T \leq U$ holds, then the subgroup U of S is called *lattice-equal* (or *translationengleich*).

(*b*) If $TU = S$ holds, then the subgroup U of S is called *class-equal* (or *klassengleich*).

Recall that, for any subgroup U of S, the translation subgroup T_U of U is just the intersection of U and T. Therefore, $T_U = T$ holds if U is a lattice-equal subgroup of S. With respect to a fixed basis of $T = T_U$, the point group of a lattice-equal subgroup U of S is a subgroup of the point group of S. If U is a class-equal subgroup of S, then $T_U \leq T$ and the point group of U is $\mathbb{Q}$-equivalent to the point group of S.

Note that every subgroup of a space group may be obtained as a class-equal subgroup of a lattice-equal subgroup of the space group. The space group itself is both a class-equal and a lattice-equal subgroup. There exist subgroups of a space group that are neither lattice-equal nor class-equal. Maximal subgroups, however, are always either class-equal or lattice-equal. This is known as Hermann's theorem (Hermann, 1929).

In the case that M is a lattice-equal maximal subgroup of the space group S, the point group of M is a subgroup of the point group of S which has to be maximal. Thus, we may obtain M/T and therefore M by the methods for finite groups as described in §5.1.

In the case that M is a class-equal maximal subgroup of the space group S, its translation subgroup T_M must be a proper subgroup of T. In the next theorem, we investigate the connection between T and T_M further.

Theorem 5.2. Let M be a class-equal maximal subgroup of S. Let T be the translation subgroup of S and $T_M = T \cap M$ the translation subgroup of M.

(i) T_M is a maximal S-invariant subgroup of T.

(ii) The factor T/T_M is an elementary abelian p group for a prime p. In particular, the index of T_M in T is a p power and thus finite.

(iii) $[S : M] = [T : T_M]$.

Proof.

(i) First we have to prove that T_M is invariant under conjugation action of S. Let $s \in S$. We define $T_M^s := s^{-1}T_M s$. Since $S = TM$, we may write $s = tm$. Since T is abelian, we know that $T_M^t = T_M$ and thus we obtain $T_M^s = T_M^{tm} = (T_M^t)^m = T_M^m$. But T_M is the translation subgroup of M and therefore we have that T_M is normal in M. So this yields $T_M^s = T_M^m = T_M$ and we obtain that T_M is S-invariant.

Now suppose there exists an S-invariant subgroup L with $T_M < L < T$. Since L is normal in S, we have that LM is a subgroup of S. However, this yields $S = TM > LM > T_M M = M$. But this is a contradiction, since M is a maximal subgroup of S.

(ii) By (i) we know that T/T_M does not have any subgroup that is invariant under conjugation action of S/T_M. Since any characteristic subgroup of T/T_M, *i.e.* any subgroup of T/T_M that is invariant under each automorphism of T/T_M, would be invariant under the action of S/T_M, the factor group T/T_M cannot have any characteristic subgroup. However, each finitely generated abelian group without characteristic subgroups is of the form $(\mathbb{Z}/p\mathbb{Z})^n$ for a prime p and an integer n. *I.e.* T/T_M is an elementary abelian p group of order p^n.

(iii) Since S/T_M is a finite group by (ii), this follows directly from the homomorphism theorem. □

By theorem 5.2, a class-equal maximal subgroup of S has p-power index for a prime p. We call such a maximal subgroup *p-maximal*. However, there may exist class-equal p-maximal subgroups for infinitely many primes p. [In fact it is known from group theory that there exists at least one class-equal p-maximal subgroup for each prime p with $p \nmid |S/T|$, see Suzuki (1982), ch. 2, theorem 8.10.] Thus, we cannot compute all class-equal maximal subgroups at once. Here we will restrict the algorithm to compute class-equal p-maximal subgroups of a given p only.

Let M be a class-equal p-maximal subgroup of S. Then theorem 5.2 shows that T_M is a normal subgroup of S. Thus, from the group-theoretical point of view, M/T_M is a complement to T/T_M in S/T_M. The following lemma shows that this is in fact a characterization of class-equal p-maximal subgroups.

Lemma 5.3. Let L be a maximal S-invariant subgroup of T and let M be a subgroup of S such that M/L is a complement of T/L in S/L. Then M is a maximal class-equal subgroup of S.

Proof. Suppose we have a group K with $M < K < S$. Since M is class-equal, the subgroup K is also class-equal. Thus we obtain $L = T_M < T_K < T$ and T_K is S-invariant. But L is maximal S-invariant in T and we have a contradiction. □

This characterization of class-equal p-maximal subgroups of S by complements will be used to compute these subgroups. The following lemma yields that there are only finitely many class-equal p-maximal subgroups of a given space group S and fixed prime p. Futhermore, it will lead to a method to compute the possible translation subgroups T_M for class-equal p-maximal subgroups M.

Lemma 5.4. Let S be a space group with translation subgroup $T \cong \mathbb{Z}^d$. For a fixed prime p, we consider the subgroup T^p consisting of all pth powers of elements of T, *i.e.* $T^p \cong (p\mathbb{Z})^d$.

(i) T/T^p is an elementary abelian group of order p^d.

(ii) T^p is normal in S.

(iii) Let M be a class-equal p-maximal subgroup of S. Then $T^p \leq T_M < T$ holds.

Proof.

(i) Since $T/T^p \cong \mathbb{Z}^d/(p\mathbb{Z})^d \cong (\mathbb{Z}/p\mathbb{Z})^d$, we obtain that T/T^p is an elementary abelian group of order p^d.

(ii) Since T^p is a characteristic subgroup of T and T is normal in S, we obtain that T^p has to be normal in S.

(iii) As explained above, $T_M < T$ holds. Thus it remains to prove $T^p \leq T_M$. By theorem 5.2, the factor T/T_M is an elementary abelian p group. But T^p is the smallest subgroup of T that has an elemenary abelian factor group of p-power order and thus $T^p \leq T_M$ holds. □

So let M be a class-equal p-maximal subgroup of a space group S for a prime p. Then the location of M in S is as illustrated in Fig. 2.

Now we are ready to introduce the algorithms to compute maximal subgroups of a given space group.

5.1. *Lattice-equal maximal subgroups*

To compute the lattice-equal maximal subgroups of a space group S with translation subgroup T, we first obtain a permutation representation of S/T. If S/T is solvable, we then compute a power-commutator presentation of this permutation group, which allows the use of a very efficient method to compute the conjugacy classes of maximal subgroups of S/T (Cannon & Leedham-Green, 1997; Eick, 1993). It then remains to compute

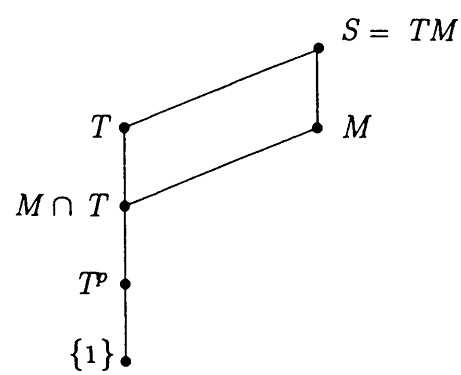

Fig. 2. The location of a class-equal p-maximal subgroup M.

pre-images of the generators of the maximal subgroups in S.

If S/T is not solvable, we use the generic GAP method to compute the conjugacy classes of maximal subgroups of a permutation group. This algorithm will compute all conjugacy classes of subgroups of the given permutation group (which of course would yield all lattice-equal subgroups of S) and then reduce them to the maximal ones. It is evident that this involves a lot of overhead and will not be as efficient as the algorithm for the solvable space groups.

5.2. *Class-equal p-maximal subgroups*

Let S be a space group and p a fixed prime. The computation of the conjugacy classes of class-equal p-maximal subgroups in the general case is done in two parts. We will describe an alternative method for solvable space groups afterwards.

First compute all possible candidates for the translation subgroup of these maximal subgroups. By theorem 5.2, this amounts to the computation of all maximal S-invariant subgroups of T with p-power index. These maximal subgroups will contain T^p, and therefore we may as well compute all maximal S-invariant subgroups of T/T^p. So we compute the linear matrix group in $GL(d,p)$ induced by the conjugation action of S on T/T^p, and then determine the maximal S-invariant subgroups of the elementary abelian group T/T^p. It then remains to determine pre-images of the computed subgroups in T.

Now suppose we have a maximal S-invariant subgroup L of T that has p-power index. We need to compute the conjugacy classes of complements to T/L in S/L. We first determine a finite presentation of S/T, from which we obtain the conjugacy classes of complements to T/L in S/L by the method mentioned in §3. It remains to compute their pre-images in S to obtain the conjugacy classes of class-equal p-maximal subgroups of S with translation subgroup L.

If S is a solvable space group, we can use the following alternative method, which is sometimes more efficient. As outlined above, in this case we can compute a power-commutator presentation of S/T. This may be extended to a power-commutator presentation of S/T^p. By a slight modification of the routine to compute all conjugacy classes of maximal subgroups in a group given by a power-commutator presentation, we may then just compute the conjugacy classes of maximal subgroups that do not contain T/T^p. It remains to compute pre-images of the computed subgroups in S.

6. Examples and timings

In this section, we give some examples and the timings for these examples. All timings have been measured on a PC with a 133 MHz Pentium processor running FreeBSD Unix, Version 2.1.0. The timings are given in seconds.

Table 1. *Timings for the computation of Wyckoff positions*

Space group	Size	Solvable	Symm.	t_{EGN}	t_{FE}
$S_{(3,230)}$	48	Yes	No	1.1	0.6
$S_{(4,22,10,1,2)}$	36	Yes	No	0.9	2.8
$S_{(4,31,7,1,1)}$	240	No	Yes	7.1	430.0
$S_{(4,33,16,1,1)}$	1152	Yes	Yes	56.9	
$S_{(6,1)}$	60	No	No	0.9	35.1
$S_{(6,2)}$	120	No	No	2.8	162.8
$S_{(8)}$	14400	No	Yes	397.2	

The first four examples are extracted from the GAP library. $S_{(3,230)}$ is the three-dimensional non-symmorphic cubic space group $Ia\bar{3}d$, No. 230 in *International Tables for Crystallography* (1995). Its point group has size 48. The next three space groups are four-dimensional. Their labels are those from Brown *et al.* (1978), which are also used in the GAP library. $S_{(4,22,10,1,2)}$ is non-symmorphic as well, and has a solvable point group of size 36. $S_{(4,31,7,1,1)}$ is symmorphic, with a non-solvable point group of size 240, and $S_{(4,33,16,1,1)}$ is symmorphic as well, with a solvable point group of size 1152. The next two examples are six-dimensional space groups relevant for the description of the symmetry of quasicrystals. We take the two non-symmorphic space groups for the primitive icosahedral lattice (Levitov & Rhyner, 1987). $S_{(6,1)}$ has a point group of size 60, isomorphic to A_5, whereas $S_{(6,2)}$ has a point group of size 120, isomorphic to $A_5 \times C_2$. Finally, we consider an eight-dimensional space group $S_{(8)}$ with (non-solvable) point group isomorphic to the Coxeter group H_4, whose order is 14 400. H_4 is the symmetry group of the famous 660-cell polytope in four dimensions. A four-dimensional matrix representation, with matrix entries in $\mathbb{Z}[\tau]$, where $\tau = (1 + 5^{1/2})/2$ is the golden mean, is contained in GAP's share package *Chevie*. This four-dimensional representation can easily be lifted to an eight-dimensional integral representation of H_4, of which we take the semi-direct product with $\mathbb{Z}^8$ as our example space group $S_{(8)}$.

The computations have all been done in a workspace of 15 Mbytes, with two notable exceptions. For the computation of the Wyckoff positions and the lattice-equal subgroups of $S_{(8)}$, a larger workspace was necessary. In those two cases, we have used a workspace of 30 Mbytes. The workspace used is in all cases considerably larger than absolutely necessary. The computations could have been done in a smaller workspace, at the expense of a somewhat longer runtime, owing to the more frequent garbage collections.

Table 1 contains the timings for the computation of the Wyckoff positions for all our examples. For each group, we give the timing for our program (t_{EGN}) and compare it to the runtime for the program of Fuksa & Engel (1994) (t_{FE}). In two cases, such a comparison was not possible, as the latter program could not finish without exceeding the available memory (100 Mbytes).

Table 2. *Timings for the computation of maximal subgroups*

Space group	Lattice Time	Lattice No.	Class 2 Time	Class 2 No.	Class 3 Time	Class 3 No.	Class 5 Time	Class 5 No.	Class 7 Time	Class 7 No.
$S_{(3,230)}$	0.3	5	0.2	0	0.2	1	0.3	1	0.3	1
$S_{(4,22,10,1,2)}$	0.3	5	0.4	2	0.3	0	0.5	2	0.7	2
$S_{(4,31,7,1,1)}$	6.7	6	0.5	1	0.5	1	0.4	1	0.5	1
$S_{(4,33,16,1,1)}$	2.2	6	2.1	1	2.3	1	2.4	1	2.4	1
$S_{(6,1)}$	0.7	3	0.4	1	1.0	1	0.5	0	0.3	1
$S_{(6,2)}$	1.9	4	0.4	2	0.2	1	0.4	1	0.3	1
$S_{(8)}$	408.5	6	4.7	1	4.9	1	4.8	1	5.0	1

The computation of the Wyckoff positions and lattice-equal maximal subgroups of space group $S_{(8)}$ was not fully automatic. For these computations, the subgroup lattice of the point group is needed, for which GAP needs a list of all perfect subgroups of the point group. Perfect subgroups are available in a format accessible by GAP's *Lattice* command only up to size 5000. The solvable residuum of the point group of $S_{(8)}$ has size 7200, however. The solution was to determine the perfect subgroups in a first step and store them in the knowledge of the point group. The rest of the computation is then fully automatic. Since all perfect groups with size a divisor of 7200 are generated by two elements, and for each of these groups the order of these generators is known, we could obtain the list of perfect subgroups by scanning through all subgroups having such a generating system, and checking whether they are perfect. We actually need only one such subgroup per conjugacy class. The time for the determination of the perfect subgroups is included in the timings. This example shows that even space groups with very large point groups can be handled by our programs.

If Wyckoff positions for several space groups in the same $\mathbb{Z}$ class or $\mathbb{Q}$ class are needed, it is possible to compute the subgroup lattice of the point group only for one of these space groups, and use it for the other ones as well. With this trick, we can compute the Wyckoff positions of all 230 three-dimensional space groups in 62 s, and those of all 4783 four-dimensional space groups in 4284 s.

In Table 2, we show for the same seven space groups the timings for the computation of maximal subgroups. In each case, the time required to compute the lattice-equal maximal subgroups is given, as well as the time to compute the class-equal p-maximal subgroups for several prime numbers p. Also, the number of the computed conjugacy classes of maximal subgroups is included in the table. All timings include the computation of the point group (if necessary) as well as the computation of the presentation that is used.

We would like to thank H. Wondratschek for suggesting the problem of computing the maximal subgroups of a space group. We are indebted to J. Neubüser for many fruitful discussions and careful reading of this paper. BE and WN acknowledge financial support from the Graduiertenkolleg 'Analyse und Konstruktion in der Mathematik'. FG was supported by the Swiss Bundesamt für Bildung und Wissenschaft in the framework of the HCM programme of the European Community. This collaboration was in part made possible by financial support from the HCM project 'Computational Group Theory'.

References

Brown, H., Bülow, R., Neubüser J., Wondratschek, H. & Zassenhaus, H. (1978). *Crystallographic Groups of Four-Dimensional Space*. New York: Wiley-Interscience.
Cannon, J. & Leedham-Green, C. R. (1997). In preparation.
Celler, F., Neubüser, J. & Wright, C. (1990). *Acta Appl. Math.* **21**, 57–76.
Cohen, H. (1993). *A Course in Computational Algebraic Number Theory*. Berlin: Springer-Verlag.
Eick, B. (1993). Diploma Thesis, Lehrstuhl D für Mathematik, RWTH Aachen, Aachen, Germany.
Fuksa, J. & Engel, P. (1994). *Acta Cryst.* A**50**, 778–792.
GAP (1997). Groups, Algorithms and Programming, Version 3.4.4, M. Schönert *et al.*, Lehrstuhl D für Mathematik, RWTH Aachen. GAP can be obtained by anonymous ftp from ftp.math.rwth-aachen.de. See also http://www-gap.dcs.st-and.ac.uk/~gap/.
Gratias, D. & Hippert, F. (1994). Editors. *Lectures on Quasicrystals*. Les Ullis: Les Editions de Physique; Berlin: Springer-Verlag.
Hermann, C. (1929). *Z. Kristallogr.* **69**, 533–555.
International Tables for Crystallography (1995). Vol. A, 4th ed., edited by T. Hahn. Dordrecht: Kluwer Academic Publishers.
Janot, C. (1994). *Quasicrystals: a Primer*, 2nd ed. *Monographs on the Physics and Chemistry of Materials*, Vol. 50. Oxford University Press.
Janssen, T. & Janner, A. (1987). *Adv. Phys.* **36**, 519–624.
Janssen, T., Janner, A., Looijenga-Vos, A. & de Wolff, P. M. (1992). *International Tables for Crystallography*, Vol. C, edited by A. J. C. Wilson, pp. 797–844. Dordrecht: Kluwer Academic Publishers.
Levitov, L. S. & Rhyner, J. (1988). *J. Phys. (Paris)*, **49**, 1835–1849.
Nebe, G. & Plesken, W. (1995). *Finite Rational Matrix Groups*. *Mem. Am. Math. Soc.* **116**, No. 556.
Sims, C. (1994). *Computation with Finitely Presented Groups*. Cambridge University Press.
Suzuki, M. (1982). *Group Theory I*. *Grundlehren Mathematische Wissenschaften*, Vol. 247. Berlin: Springer-Verlag.

Thiers, A. H. M., Ephraim, M. J. & de Hilster, H. (1996). http://www.caos.kun.nl/cgi-bin/csecm/csecm/.

Thiers, A. H. M., Ephraim, M. J., Janssen, T. & Janner, A. (1993). *Comput. Phys. Commun.* **77**, 167–189.

Wondratschek, H. (1995). *Introduction to Space-Group Symmetry. International Tables for Crystallography*, Vol. A, 4th ed., edited by T. Hahn, pp. 711–735. Dordrecht: Kluwer Academic Publishers.