71-20,844

LEVINE, Ronald Dorian, 1939-
THE COMPACT EUCLIDEAN SPACE FORMS OF
DIMENSION FOUR.

University of California, Berkeley, Ph.D.,
1970
Mathematics

University Microfilms, A XEROX Company, Ann Arbor, Michigan

The Compact Euclidean Space Forms of Dimension Four

By

Ronald Dorian Levine
A.B. (University of California) 1960
M.A. (University of California) 1964

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Mathematics

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Approved:

Joseph A. Wolf
Morris W. Hirsch
Frank S. Crawford

Committee in Charge

DEGREE CONFERRED DEC. 19, 1970

The Compact Euclidean Space Forms of Dimension Four

Ronald Dorian Levine

PLEASE NOTE:

Some pages have indistinct print. Filmed as received.

UNIVERSITY MICROFILMS.

The Compact Euclidean Space Forms of Dimension Four

Abstract

Ronald Dorian Levine

The main result of this work is a complete affine classification of the compact four-dimensional riemannian manifolds of constant zero curvature. There are 75 classes of which 27 are orientable. These spaces are described by means of tables of their algebraic invariants: fundamental groups, first homology groups, and linear holonomy groups. A second feature of this work is the description of a set of computer programs which were instrumental in obtaining the first result and which promise to be useful in extending the work to higher dimensions and in the more general problem of enumerating the higher dimensional crystallographic groups.

Contents

Chapter 0. Introduction 1

Chapter 1. The Arithmetical Classes of Holonomy 8

Chapter 2. The Cohomology Group of Extensions 15

Chapter 3. The Torsion-free Extensions. 21

Chapter 4. The First Homology Group and the Isomorphism Problem 25

Chapter 5. The Computer Programs. 32

Bibliography 44

Appendix. Tables 46

Chapter 0

Introduction

The main result of this work is a complete affine classification of the 4-dimensional compact riemannian manifolds of constant zero curvature.(*) There are 75 classes, of which 27 are orientable. Tables C and D of the appendix contain their descriptions by means of their algebraic invariants: fundamental groups, first homology groups, and linear holonomy groups. A second feature of this paper is the description of a set of computer programs which were instrumental in obtaining the first result and which promise to be useful in extending the work to higher dimensions and in attacking the more general problem of enumerating higher dimensional crystallographic groups.

The structure theory of riemannian manifolds of zero curvature, also called <u>flat manifolds</u> or <u>euclidean space forms</u>, as well as the history of its development, may be found in the book by Wolf. [24] (**) Here we summarize the principal results for the compact euclidean space forms.

(*) In this paper the world "manifold" is understood to entail "differentiable" and "connected".

(**) Digits in square brackets refer to the bibliography on p. 44.

The most satisfactory equivalence relation for classifying flat manifolds is affine equivalence, i.e. equivalence by diffeomorphisms preserving the affine connection, but in the compact case it turns out that this coincides with classification by homeomorphism. Every n-dimensional compact euclidean space form M can be realized as the quotient of euclidean R^n by the action of a discrete group G of isometries, with each nontrivial element of G having no fixed points in R^n. Then G, as the group of deck transformations of the universal covering of M, is isomorphic to its fundamental group $\pi_1(M)$. Conversely, if G is a discrete group of fixed-point-free isometries of R^n, then the quotient space R^n/G has naturally the structure of a flat riemannian manifold which is compact if and only if G contains a set of n linearly independent translations.

Moreover, two compact euclidean space forms are affinely equivalent if and only if their fundamental groups are isomorphic. For each dimension n, there are only finitely many classes of compact flat manifolds; in dimensions 1, 2, 3, there are 1, 2, 10 (resp.) of them, of which 1, 1, 6, (resp.) are orientable.

Discrete groups of isometries of R^n acting with compact quotient are called n-dimensional crystallographic groups, or space groups. In dimensions 3 and 2 they have famous applications to the description of physical crystal lattices and ornamental designs. It can be seen that, for these groups, acting without fixed points is equivalent to being torsion-free, i.e., to having no non-trivial elements of finite order. Thus, the problem of determining the compact euclidean space forms is reduced to determining all torsion-free crystallographic groups up to isomorphism.

The structure theory of crystallographic groups, due mainly to Bieberbach, can be found in [2, 3, 26]. Again we summarize the well-known main results on which the present computation is based.

Let G be an n-dimensional crystallographic group and let T be the set of translations in G. Then T is a normal subgroup of finite index in G, which is free abelian of rank n and maximal abelian in G. The finite quotient group $F = G/T$ is called the point group or crystal class of G, and in the torsion-free case F is isomorphic to the linear holonomy group of the corresponding flat manifold. Now conjugation of T by coset representatives in G of the elements of F induces a well-defined faithful representation of F in $\mathrm{Aut}(T)$. The choice of an integral basis in T embeds T in Z^n, and consequently F in $GL(n,Z)$. These embeddings can be extended to an embedding of G in $GL(n,Z)\cdot Q^n$ (semi-direct product), and for the remainder of this paper we will work with these groups in this concrete arithmetical form. Specifically, Q^n denotes the additive group of n-component rational column vectors, Z^n denotes the integer lattice in Q^n, and $GL(n,Z)$ denotes the multiplicative group of $n \times n$ unimodular integer matrices. The semi-direct product of $GL(n,Z)\cdot Q^n$ consists of all pairs $(A,s) \in GL(n,Z) \times Q^n$, with the multiplication rule,

$$(A,s)(B,t) = (AB, s + At).$$

We will call A the <u>linear part</u> and s the <u>translation part</u> of (A,s).

According to a theorem of Bieberbach, every isomorphism between two crystallographic groups is induced by an inner automorphism of $GL(n,Z)\cdot Q^n$, and this of course entails that the point groups be conjugate in $GL(n,Z)$. A class of finite subgroups of $GL(n,Z)$ conjugate in $GL(n,Z)$ is called an <u>arithmetical crystal class</u>, while a class of finite subgroups of $GL(n,Z)$ conjugate in $GL(n,Q)$ is called a <u>geometrical crystal class</u>. For any finite subgroup F of $GL(n,Z)$, we will call G a <u>crystallographic extension</u> of F if G is a subgroup of $GL(n,Z)\cdot Q^n$ and there is an exact sequence

$$0 \to Z^n \xrightarrow[i]{} G \xrightarrow[j]{} F \to 1$$

in which i is the canonical inclusion and j is the restriction to G of the canonical projection $GL(n,Z)\cdot Q^n \to GL(n,Z)$.

Our determination of the 4-dimensional compact euclidean space forms follows, in three parts out of four, a general program enunciated by Charlap in his study [6] of the problem of classifying, for each given abstract finite group Φ, the flat manifolds with holonomy isomorphic to Φ. The first step in Charlap's program, "determine all integral representations of Φ", is difficult and has been completed only for the most simple groups Φ. [22] The integral representations of the cyclic groups of prime order have been classified by Reiner [21] in terms of the ideal class groups of the prime cyclotomic fields, and Charlap has used this to classify the flat manifolds of all dimensions whose holonomies are cyclic of prime order.

In the present work, we replace Charlap's first step with

<u>Step</u> <u>I</u>: Find a set of mutually arithmetically inequivalent finite subgroups of $GL(4,Z)$ which contains all possible holonomy groups.

Specifically, as we will see in Chapter 3, any matrix group F which has a torsion-free crystallographic extension necessarily satisfies

<u>Condition</u> <u>A</u>: Every element of F has non-zero fixed space.

In Chapter 1 we show how to determine a complete set of arithmetical crystal classes satisfying Condition A. In Chapter 2 we show how to determine, for each F found in Step I, a complete set of crystallographic extensions of F, which are perhaps not mutually non-isomorphic. Namely, we have

<u>Step</u> <u>II</u>; Compute the second cohomology group $H^2(F,Z^n)$.

In Chapter 3 we treat

<u>Step</u> <u>III</u>: Recognize the torsion-free groups arising in Step II.

In Chapter 4 we carry out

<u>Step</u> <u>IV</u>: Find a mutually non-isomorphic set from the groups selected in Step III.

In each of these chapters the corresponding step is reduced to the level of arithmetical computations which can be completed electro-mechanically and we name the computer codes which have been developed to perform them. Further details on the codes and the mechanics of their use are collected in Chapter 5, and listings of the programs comprise Table E.

Our central computational tool is the Fundamental Theorem of Finitely Generated Abelian Groups, which may be stated in the arithmetical form: If M is any $p \times q$ matrix with integer coefficients, then there exists a $p \times p$ unimodular integer matrix L, and a $q \times q$ unimodular integer matrix K such that

$$LMK = \begin{pmatrix} e_1 & & & & & & 0 \\ & e_2 & & & & & \\ & & \ddots & & & & \\ & & & e_{r} & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ 0 & & & & & & 0 \end{pmatrix}$$

and the integers $r, e_1, \ldots, e_r$ are uniquely determined by

$$0 < e_1 \mid e_2 \mid \ldots \mid e_r \,.$$

The standard constructive proof may be found in [13]. We use this construction at every turn, and will refer to it in the text by the acronym FTAG. One version of our Compass language code for it, called DIAR, is listed in Table E.

In 1957, Calabi [5] announced a determination of the 4-dimensional compact euclidean space forms using a recursive geometric approach which is quite different from our method and which is described in [24]. He gives details neither of the computation, nor of the results, but only the numbers of orientable and non-orientable spaces occuring, and these numbers are contradicted by the present work. Charlap and Sah have privately circulated a report [7] of another determination made in the spirit of Charlap's program, but without the aid of a computer.

They give no details of the computations, but do give a final list of groups, which we find to be incorrect in several respects.

The author wishes to express his thanks to Professor Joseph A. Wolf who introduced him to the problem, and to Mr. D. J. Underwood and the staff of the Humboldt State College Computer Center who arranged his easy access to their facilities.

Chapter 1

The Arithmetical Classes of Holonomy

The crystal classes in dimensions $n < 4$ have been known for some time and are given in Burckhardt's book [3]. In dimension 2 there are 10 geometrical and 13 arithmetical classes, while in dimension 3 there are 32 geometrical and 73 arithmetical classes.

The results in dimension 4 begin in 1889 with Goursat [10], who gave a determination of the (infinitely many) conjugate classes of finite subgroups of $SO(4)$ and of those of $O(4)$ containing the inversion in the origin $-I$. This work was later corrected and extended by Seifert and Threlfall [23]. In these works the principal tool used for pulling up the 3-dimensional groups is the double covering $SO(4) \to SO(3)xSO(3)$. In 1951, Hurley [11] extracted from Goursat's list of classes all those having integral representations, i.e. geometrical crystal classes of R^4, and found the classes not containing $-I$ as subgroups of those containing $-I$. As amended in [12], there are 227 4-dimensional geometrical crystal classes.

The nine maximal 4-dimensional arithmetical crystal classes are worked out in a remarkable paper by Dade [9]. Finally, Neubüser and Bülow [4] have reported a computer search for the subgroups of Dade's groups, finding that there are 720 4-dimensional arithmetical crystal classes in all. However these authors have not yet released their list of groups.

We begin the present computation by extracting from Hurley's list of geometrical classes a list of all those which satisfy Condition A. Since Hurley names a finite matrix group by enumerating its elements by similarity type, this task is quite easy. There are 45 geometrical classes satisfying Condition A, of which 11 are orientation preserving. They are listed in Table A. Now we need to determine for each of these 45 classes the classes into which it splits under arithmetical equivalence, and this task is facilitated by the fact that every one of the 45 classes is (3+1)-reducible.

We proceed according to the theory of [25] and [8] to build the 4-dimensional arithmetical crystal classes from the known 3-dimensional classification. Here we inject some of the language of representation theory, and it is well to note that the usual sense of equivalence for integral representations is not the same as our arithmetical equivalence. Namely, two integral representations $g,h:\Phi \to GL(n,Z)$ of a group Φ are equivalent in the sense of representation theory if there is an inner automorphism i_X of $GL(n,Z)$ such that the diagram

$$\begin{array}{ccc} \Phi & \xrightarrow{g} & GL(n,Z) \\ \| & & \downarrow i_X \\ \Phi & \xrightarrow[h]{} & GL(n,Z) \end{array}$$

commutes, while for the arithmetical equivalence of $g(\Phi)$ with $h(\Phi)$ an arbitrary automorphism of Φ can be substituted for the identity map in the diagram.

Now let F be one of our 45 candidates for holonomy groups.

Concretely we choose F to be a (3+1)-decomposed integral matrix group(*) whose 3x3 block is one of Burckhardt's representatives of a 3-dimensional crystal class. We write

$$A = \begin{pmatrix} g(A) & 0 \\ 0 & h(A) \end{pmatrix} \quad ; \qquad A \in F$$

where g is a 3-dimensional and h a 1-dimensional integral representation of F. Note that g,h are neither unique, nor, in general, faithful. Regarding $g(F)$ as a 3-dimensional geometrical crystal class, we get from Burckhardt a complete set of representatives $g_1, g_2, \ldots, g_r$ of the arithmetical classes into which it splits. Now it follows from the theory of [8] that every arithmetical class belonging to F can be represented in the form

$$\text{(1)} \qquad A \mapsto \begin{pmatrix} g_i(A) & t(A) \\ 0 & h(A) \end{pmatrix} \qquad \text{for some } i = 1,\ldots,r$$

where $t:F \to \{3\text{x}1 \text{ integer matrices}\}$ is a function which satisfies

$$\text{(2)} \qquad \forall A,B \in F \qquad t(AB) = g_i(A)t(B) + t(A)h(B)$$

and is called a <u>binding function</u> for g_i and h.

Moreover, for a given g_i and h, we get a finite, complete, but perhaps not mutually inequivalent set of representations of the form (1) by taking for binding functions a set of coset representatives of the cohomology group,

---

(*) The translation of Hurley's notation given in [18] was useful in constructing these.

(3) $$C(g_i,h) = B(g_i,h)/B'(g_i,h)$$

where $B(g_i,h)$ is the additive group of all binding functions and $B'(g_i,h)$ is the subgroup of all <u>inner binding functions</u>, i.e. all those of the form

(4) $$t(A) = g_i(A)d - dh(A) \qquad \forall A \in F$$

where d is an arbitrary 3x1 integer matrix.

Computing this group for each g_i, $i = 1,\ldots,r$, gives a list $\{F_1,\ldots,F_s\}$ of matrix groups which contains all the arithmetical classes belonging to the geometrical class of F and which now must be tested for equivalences. For each pair (i,j) we must determine whether there exists a matrix $X \in GL(4,Z)$ with $F_iX = XF_j$. Note that equivalences may obtain between F_i arising from distinct g_j.

The computation of the cohomology group is made automatically and the determination of equivalence is made semi-automatically. We describe these computations in the context of binding together an arbitrary p-dimensional integral representation g of a finite group Φ with an arbitrary q-dimensional integral representation h, for the programs are written in this generality.

We identify the additive group of all pxq integer matrices with Z^{pq}, and the set of all functions $\Phi \to Z^{pq}$ with Z^{kpq}, where k is the order of Φ. For each pair $(a,b) \in \Phi \times \Phi$ the binding function condition (2) gives a linear map

$$\begin{aligned} Z^{kpq} &\longrightarrow Z^{pq} \\ t &\longmapsto t(ab)-g(a)t(b)-t(a)h(b) \end{aligned}$$

and taking the direct sum over all pairs $(a,b) \in \Phi \times \Phi$ gives a linear map

$$Z^{kpq} \xrightarrow{\varepsilon} Z^{k^2pq}$$

whose kernel is $B(g,h)$. Now sending an arbitrary $p \times q$ matrix d to the function (4) gives a linear map

(5) $$Z^{pq} \xrightarrow{\delta} Z^{kpq}$$

and we have

$$C(g,h) = B(g,h)/B'(g,h) = \ker \varepsilon/\mathrm{im}\,\delta \; .$$

We claim that all information about $C(g,h)$ is contained in the map δ and the map ε can be ignored. For $B(g,h)$ is the kernel of a map into a free abelian group and therefore is a direct summand of Z^{kpq}; and since the theory guarantees that $C(g,h)$ is finite, $B'(g,h)$ spans $B(g,h)$ over Q (in fact, over $(1/k)Z$). In other words

(6) $$C(g,h) = ((Q\cdot \mathrm{im}\,\delta) \cap Z^{kpq})/\mathrm{im}\,\delta \; .$$

Suppose Φ is generated by $\{a_1,\ldots,a_\mu\}$. Since a binding function is determined by its values on the generators, we can replace Z^{kpq} in (5) and (6) with $Z^{\mu pq}$. Now we represent δ by a $(\mu pq)\times(pq)$ integer matrix Δ, which is easily constructed from the input data $\{g(a_i),\ h(a_i),\ i = 1,\ldots,\mu\}$ according to (4). The columns of Δ span $B'(g,h)$ over Z and $B(g,h)$ over Q. Apply FTAG:

(7) $$L\,\Delta\,K = \mathrm{diag}(e_1,\ldots,e_r,0,\ldots,0).$$

Now it is clear from (6) and (7) that a complete set of representatives of the distinct elements of $C(g,h)$ is obtained by taking the elements (*)

$$L^{-1}[f_1,\ldots,f_r,0,\ldots,0] \qquad 0 \le f_i < e_i,\ i = 1,\ldots,r$$

(*) We reconcile the demands of typographical economy with our preference for writing operators on the left by adopting the convention that a row vector enclosed in square brackets is to be read as a column.

These vectors are then reassembled into sets of $p \times q$ matrices

$$t(a_i), \; i = 1, \ldots, \mu$$

and stuffed into the matrices

$$\begin{pmatrix} g(a_i) & * \\ 0 & h(a_i) \end{pmatrix} \qquad i = 1, \ldots, \mu,$$

to give the list of integral representations of Φ which we seek.

Now let F_1 and F_2 be two integral representations of degree n. To settle the question of their arithmetical equivalence, we must determine whether for any automorphism σ of Φ the equations

$$(8) \qquad F_1(a_i)X - X\,F_2(\sigma(a_i)) = 0 \qquad i = 1, \ldots, \mu$$

have a solution by an nxn unimodular integer matrix X.
For each σ, (8) constitutes a set of μn^2 homogeneous linear equations in the n^2 coefficients of X. The solution set is a subgroup of the additive group of all nxn integer matrices and we find an integral basis for it by applying FTAG to the matrix of coefficients of (8). This allows us to express the general matrix satisfying (8) as a matrix Ξ of homogeneous linear forms in some number $s = n^2 - r$ of variables.(*) Thereby, the problem of the equivalence of F_1 and F_2 is reduced to determining the existence of integer roots of one of the polynomial equations

$$(9) \qquad \det \Xi = \pm 1$$

(*) See the discussion of equation (2a) of Chapter 4 for more details.

The left side of (9) is a homogeneous polynomial of degree n in s variables. No one knows a general mechanical procedure for determining the existence of integer roots of polynomials in several variables. Our computer program, in each case, computes the polynomial det Ξ and then tests it for an obvious necessary condition, namely, that the greatest common divisor of the set of coefficients be 1. If this condition is satisfied, the polynomial is printed out as a problem for the user. In almost every such case in the present computation, the existence of integer roots was apparent by inspection; the few exceptions were not critical due to redundancy in the computations.(**)

Notice that not every abstract automorphism of Φ must be tested in (8). First, if F_1 and F_2 are not equivalent as integral representations, i.e. with σ = id, but are equivalent by some other σ, then σ is necessarily an outer automorphism of ϕ. Second, in many specific cases some automorphisms of ϕ are seen to be forbidden by the numerical invariants of the matrices of F_1 and F_2.

The computations just described are the function of the pair of programs VB/VC described in Chapter 5 and listed in Table E.

The 45 4-dimensional geometrical crystal classes of our initial sample are seen to split into 229 arithmetical classes. Of these, according to the subsequent computations, 49 are holonomy classes of compact euclidean space forms and are listed in Table C.

(**) The same semi-automatic method of deciding arithmetical equivalence was used in [4].

Recently, Janssen [14] has reported determining all the (3+1)-reducible arithmetical crystal classes, using a method rather more geometrical than arithmetical, constructing the Bravais lattices corresponding to these groups from the known 3-dimensional Bravais lattices. Differences in notation, representations, and viewpoint militate against an easy detailed comparison of our results with his; but a cursory comparison shows that we agree in the number of arithmetical classes into which each geometrical class splits.

Chapter 2

The Cohomology Group of Extensions

From now until the end of Chapter 4 we take F to be a fixed finite subgroup of $GL(n,Z)$. We would like to determine all crystallographic extensions G of F up to isomorphism. We begin in Step II by determining all possible G up to a classification which is finer than isomorphism. We call two extensions G and G_1 of F <u>equivalent</u> if there is a commuting diagram

$$\begin{array}{ccccccccc} 0 & \to & Z^n & \to & G & \to & F & \to & 0 \\ & & \| & & \tau\downarrow & & \| & & \\ 0 & \to & Z^n & \to & G_1 & \underset{i}{\to} & F & \to & 0 \end{array}$$

and in this case τ is necessarily an isomorphism.(*)

As is well known [19], the set of all equivalence classes of extensions has naturally the structure of a finite abelian group, the second cohomology group of F in Z^n, denoted $H^2(F,Z^n)$. (**) Moreover, in our crystallographic context, the exact sequence of F-modules

$$0 \to Z^n \subset Q^n \xrightarrow{\pi} (Q/Z)^n \to 0 ,$$

(*) Our nomenclatures "isomorphic" and "equivalent" follow MacLane; Zassenhaus used "gewöhnlich äquivalent" and "stark äquivalent" for these, resp.; Burckhardt used "äquivalent" and "null-äquivalent".

(**) Actually true for extensions of more general type than we are considering.

the corresponding connecting homomorphism in cohomology, and the divisibility of Q^n (as an abelian group), combine to provide an isomorphism

$$H^2(F,Z^n) \approx H^1(F,(Q/Z)^n) ,$$

where the first cohomology group $H^1 = H^1(F,(Q/Z)^n)$ may be realized as a quotient $H^1 = Cr/Pr$, where $Cr = Cr(F,(Q/Z)^n)$ is the group of crossed homomorphisms,

$$Cr = \{F \xrightarrow{t} (Q/Z)^n : \forall A,B \in F, \quad t(AB) = t(A) + At(B)\} ,$$

and $Pr = Pr(F,(Q/Z)^n)$ is the group of all principal crossed homomorphisms,

$$Pr = \{F \xrightarrow{t} (Q/Z)^n : \exists s \in (Q/Z)^n, \forall A \in F, t(A) = S - As\}$$

The crystallographic group determined by a crossed homomorphism t is the set of all elements of $GL(n,Z)\cdot Q^n$ of the form $(A, t(A) + z)$, $z \in Z^n$, $A \in F$. Here we have used the same symbol $t(A)$ for an element of $(Q/Z)^n$ and an arbitrary one of its lifts in Q^n.

Our computation of the cohomology group H^1 is similar to but more complicated than the cohomology calculation of the preceding chapter. The algorithm is due to Zassenhaus. [26] We begin by selecting a presentation for F:

$$\Phi = \langle a_1,\ldots,a_\mu : R_1,\ldots,R_\nu\rangle$$

$$\Phi \to F$$

$$a_i \mapsto A_i$$

where μ and ν are small. Here the A_i comprise a set of matrices generating F and the R_i comprise a complete set of relators. Each R_j is a word in the letters $a_i^{\pm 1}$. For each word W in the letters $a_i^{\pm 1}$ we will use $\overline{W}$ to denote the element of F which it represents. Thus $\overline{a_i} = A_i$.

It is clear that any crossed homomorphism is determined by its values on the generators, so Cr may be isomorphically embedded in $(Q/Z)^{n\mu}$ by $t \mapsto [t(A_1),\ t(A_2),\dots,t(A_\mu)\ ]$. Let Cr', Pr', denote the images of Cr, Pr, under this embedding.

Now we will use the relators to characterize Cr' and Pr' as subgroups of $(Q/Z)^{n\mu}$. We begin by defining a function

$$\{(W,j) : W \text{ a word, } 1 \le j \le \mu\} \to \{n\text{x}n \text{ integer matrices}\}$$
$$(W,j) \longmapsto W^{(j)}$$

with the property: if W is any word, and t is any crossed homomorphism, then $t(\overline{W}) = \sum_{j=1}^{\mu} W^{(j)} t(A_j)$.

One can verify that this property is enjoyed by the function:

if $W = a_{i_1}^{\varepsilon_1}\dots a_{i_p}^{\varepsilon_p}$, $\quad \varepsilon_j = \pm 1,\ p \ge 1,$

$$\text{then}\quad W^{(j)} = \sum_{\alpha=1}^{p} \delta_{i_\alpha j} \quad \varepsilon_\alpha \prod_{\beta=0}^{\alpha-1} A_{i_\beta}^{\varepsilon_\beta} \quad A_j^{(1/2)(\varepsilon_\alpha - 1)}$$

and $(\text{empty})^{(j)} = 0$,

wherein it is understood that A_{i_0} denotes the identity, and the δ is Kronecker's.

Another useful property of this function is:

$$\forall\ W,V,j,\quad (WV)^{(j)} = W^{(j)} + \overline{W}V^{(j)}$$

Now we form the $n\mu$ x $n\nu$ matrix

$$R = \begin{pmatrix} R_1^{(1)} & \dots & R_1^{(\mu)} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ R_\nu^{(1)} & \dots & R_\nu^{(\mu)} \end{pmatrix}$$

R defines a group homomorphism $R : Q^{n\mu} \to Q^{n\nu}$ which maps the integer lattice $Z^{n\mu}$ into $Z^{n\nu}$, thus defining a homomorphism $R' : (Q/Z)^{n\mu} \to (Q/Z)^{n\nu}$ with the commuting diagram

$$\begin{tikzcd} Q^{n\mu} \arrow[r, "R"] \arrow[d, "\pi"'] & Q^{n\nu} \arrow[d, "\pi"] \\ (Q/Z)^{n\mu} \arrow[r, "R'"'] & (Q/Z)^{n\nu} \end{tikzcd}$$

Now, as Zassenhaus shows,

$$Cr' = \ker R'$$

$$Pr' = \pi\ker R.$$

Apply FTAG: $LRK = \mathrm{diag}\,[e_1, e_2, \ldots, e_\rho, 0, \ldots, 0]$

Now, $Cr^* = K^{-1}Cr'$ is isomorphic to Cr' and

$$\begin{array}{ccccccc} Cr^* & = & \ker R'K' & = & \ker L'R'K' & \approx & (Q/Z)^{n\mu-\rho} \oplus \left(\bigoplus_{i=1}^{\rho} Z_{e_i} \right) \\ \cup & & \cup & & \cup & & \| \\ Pr^* & = & \pi\ker R\,K & = & \pi\ker LRK & \approx & (Q/Z)^{n\mu-\rho} \end{array}$$

Therefore, $H^1(F, (Q/Z)^n) \approx \bigoplus_{i=1}^{\rho} Z_{e_i}$

may be represented by the elements

$$[f_1/e_1, f_2/e_2, \ldots, f_\rho/e_\rho, 0, \ldots, 0] \in Cr^*$$

where $0 \le f_i < e_i$, for all i, $0 \le i < \rho$.

Finally, each of these $n\mu$-component column vectors is multiplied on the left by K and recognized as a crossed homomorphism

$$t: \{A_1, \ldots, A_\mu\} \to (Q/Z)^n$$

which lifts to a function

$$s: \{ A_1,\ldots,A_\mu \} \to Q^n$$

which in turn defines a crystallographic group G when the $s(A_i)$ are interpreted as the translation parts of a set of generators of G over Z^n. Clearly, different lifts of a crossed homomorphism give the same (not merely equivalent) group G.

We will call the lift of a crossed homomorphism a Frobenius map, adapting nomenclature used by Burckhardt. Thus, a Frobenius map is any function $s: F \to Q^n$ which satisfies the so-called Frobenius congruences,

$$\forall A,B \in F, \quad s(AB) \equiv s(A) + As(B) \quad (\text{mod } Z^n).$$

We have called the computer code devised to execute this algorithm HILB18, commemorating Hilbert's 18th problem.(*)

The original presentation for F may be used together with the Frobenius map to obtain a presentation for G as follows:

Take for generators	$z_1,\ldots,z_n$,	corresponding to the basis of Z^n,
and	$a_1,\ldots,a_\mu$,	corresponding to the generators of F.
Take for relators	$z_i z_j z_i^{-1} z_j^{-1}$,	$1 \le i < j \le n$,
and	$a_i z_j a_i^{-1} z_1^{-\alpha(ij1)} \ldots z_n^{-\alpha(ijn)}$	; $1 \le i \le \mu$, $1 \le j \le n$,
and	$R_j z_1^{-\beta(j1)} \ldots z_n^{-\beta(jn)}$,	$1 \le j \le \nu$,

(*) A similar program has been written by Janssen and Fast. [15].

where $\alpha(ijk)$ is the (jk) component of A_i, R_j is a relator in the presentation for F, and $\beta(jk)$ is the kth componenet of the translation part of the element of G obtained by substituting $(A_i, s(A_i))$ for a_i in R_j.

We have not given these presentations in our tables of results because we have generally found the matrix representations more useful and suggestive, and the former are easily computed from the latter. We mention these presentations here only because we do use them in our computation of the first homology groups in Chapter 4.

Chapter 3

The Torsion-free Extensions

Our concrete treatment of crystallographic groups as subgroups of $GL(n,Z)\cdot Q^n$ allows a direct arithmetical solution to step III.

Definition: Let G be a crystallographic extension of F. Then we say that $A \varepsilon F$ is critical for G if G has a non-trivial element of finite order with linear part A.

If A is a matrix of finite order k, then we denote $\sum_{i=0}^{k-1} A^i$ by trA. Clearly, $(I-A)\mathrm{tr}A = 0$.

Proposition 1: If $(A,s) \varepsilon GL(n,Z)\cdot Q^n$, then (A,s) has finite order if and only if trAs = 0, and in this case the order of (A,s) is the order of A. Proof: Immediate from the multiplication rule.

Proposition 2: Let G be a crystallographic extension of F with Frobenius map s. Let $A \varepsilon F$ have order $k > 1$. Then A is critical for G if and only if the equation

(1) $\mathrm{tr}A\ s(A) = \mathrm{tr}A\ z$

is satisfied by some $z \varepsilon Z^n$. Proof: Immediate from Proposition 1 and the fact that the elements of G with linear part A are just the $s(A)-z$, $z \varepsilon Z^n$.

Proposition 2 gives the necessity of Condition A for the existence of torsion-free extensions of F, because the image of trA is contained in the fixed space of A and if this is zero then (1) is satisfied trivially independent of s. Now we can discover whether G is torsion-free by testing each $A \varepsilon F$ for solvability of equation (1) by integers. This test is easily made mechanically as seen below. Note that the Frobenius congruences imply that $\mathrm{tr}A\ s(A) \in Z^n$.

The computing may be reduced considerably by noting that it suffices to apply the test to a certain proper subset of F which we proceed to define.

Proposition 3: Let C be conjugate to B in F, $C = ABA^{-1}$. Then C is critical for G if and only if B is critical for G. Proof: Let s be a Frobenius map for G. Then, by the Frobenius congruence, $s(C) = s(A) + As(B) - ABA^{-1}s(A) + z_1$

$$= (I-C)s(A) + As(B) + z_1, \text{ for some } z_1 \in Z^n.$$

Now suppose B is critical. Then $\exists z_2 \in Z^n$ with $\mathrm{tr}Bs(B) = \mathrm{tr}Bz_2$.

Then, since $\mathrm{tr}C = A(\mathrm{tr}B)A^{-1}$, we have

$$\begin{aligned} \mathrm{tr}Cs(C) &= \mathrm{tr}C((I-C)s(A) + As(B) + z_1) \\ &= \mathrm{tr}C(As(B) + z_1) \\ &= A\mathrm{tr}Bs(B) + \mathrm{tr}Cz_1 \\ &= A\mathrm{tr}Bz_2 + \mathrm{tr}Cz_1 \\ &= \mathrm{tr}C(Az_2 + z_1)\ , \end{aligned}$$

showing that equation (1) is satisfied for C by $Az_2 + z_1 \in Z^n$.

Proposition 4: If A, of order k, is critical for G, then so is any of its powers A^p with $1 \leq p \leq k-1$. Proof: Trivial.

Proposition 5: Let $A \in F$ have order k and let s be a Frobenius map for G. If A^p is critical for G then there exists $z \in Z^n$ satisfying

(2) $\qquad \mathrm{ptr}A\ s(A) = \mathrm{tr}A\ z.$

Proof: By Proposition 4, we may assume, without loss of generality, that p divides k. Then $q = k/p$ is the order of A^p and

$$\mathrm{tr}A^p = \sum_{i=0}^{q-1} A^{ip}.$$

By Frobenius, $$s(A^p) = \sum_{j=0}^{p-1} A^j s(A) + z_1 \quad , \quad z_1 \in Z^n$$

Whence $$\mathrm{tr}A^p s(A^p) = \sum_{i=0}^{q-1} \sum_{j=0}^{p-1} A^{ip+j} s(A) + \mathrm{tr}A^p z_1$$

$$= \mathrm{tr}As(A) + \mathrm{tr}A^p z_1.$$

If A^p is critical, then $\mathrm{tr}A^p s(A^p) = \mathrm{tr}A^p z_2$, $z_2 \in Z^n$,

and $\mathrm{tr}As(A) = \mathrm{tr}A^p z_3 \quad , \quad z_3 = z_1 - z_2$.

But the left side of this equation is fixed by A. Thus

$$p\,\mathrm{tr}As(A) = \sum_{j=0}^{p-1} A^j \mathrm{tr}As(A) = \sum_{j=0}^{p-1} A^j \mathrm{tr}A^p z_3$$

$$= \mathrm{tr}Az_3,$$

which is equation (2).

The converse to Proposition 5, even under the added hypothesis $p|k$, is not true. The simplest counterexample occurs in the torsion-free crystallographic extension of C_4 generated over Z^4 by

$$(A,s) = \left(\begin{pmatrix} 0 & -1 & & \\ 1 & 0 & & \\ & & -1 & 1 \\ & & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1/2 \end{pmatrix} \right) .$$

Here

$$\mathrm{tr}As = (1/2)\ \mathrm{tr}A \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

but A^2 is not critical.

Definition: Let Φ be a finite group. Then we call a subset $T \subset \Phi$ a torsion test set if every element of Φ is conjugate to a power of some element of T.

Torsion test sets for all the abstract groups occuring in our initial sample of 4-dimensional crystal classes are given in Table B. Now let G be a particular crystallographic extension of F given by a Frobenius map s, and let T be a torsion test set for F. According to the foregoing propositions, if, for every $A \in T$ and every divisor p of the order of A, equation (2) has no solution by $z \in Z^n$, then G is torsion-free. If for some $A \in T$ equation (1) has a solution, then G is not torsion-free. If neither case obtains, and so for some $A \in T$ equation (2) has a solution with $1 < p|k$, then we must test A^p.

Deciding the solvability of (2) is another application of FTAG. Namely, if $L(trA)K = \mathrm{diag}(f_1,\ldots,f_r,0,\ldots,0)$ and $LtrA \cdot s(A) = [u_1,\ldots,u_n]$, then the smallest positive integer p for which (2) has integral solution z is the least common multiple of the numbers $f_i/(f_i,u_i)$, $1 \leq i \leq r$.

Our computer code for Step III is called FINDFLAT.

Chapter 4

The First Homology Group and the Isomorphism Problem

We have implied in Chapter 0 that all of the affine structure of a compact euclidean space form M is contained in its fundamental (i.e., deck) group G. One quotient group of G, the holonomy G/Z^n, has been instrumental for the construction of G. We now consider another quotient of G, its commutator quotient group $G/[G,G]$. This, of course, is isomorphic to the first integral homology group of M, $H_1(M,Z)$. In the present work it is an important tool in the completion of Step IV, the isomorphism problem. For $H_1(M,Z)$ is a finitely generated abelian group, and the isomorphism of two given finitely generated abelian groups is trivially decideable by FTAG; while to the best of the author's knowledge the isomorphism problem, in the sense of presentation theory, is unsolved for crystallographic groups.

Magnus, Karrass, and Solitar [20] give a mechanical procedure by which the commutator quotient group of any finitely presented group G may be determined from the presentation. We construct the so-called "titled exponent sum matrix" E. Its rows are labelled by the generators and its columns are labelled by the relators. The (ij) element is the sum of the exponents to which generator i appears in relator j. Apply FTAG: $LEK = \text{diag}\ (e_1,\ldots,e_r,0,\ldots,0)$. Now the e_i are the torsion coefficients of $G/[G,G]$ and $\lambda-r$ is its Betti number, where λ is the number of generators in the presentation. That is, $G/[G,G] \approx Z_{e_1} \oplus \ldots \oplus Z_{e_r} \oplus Z^{\lambda-r}$

Our computer program ABELR carries this out for each of the groups determined in Step II according to the presentation given at the end of Chapter 2. The first homology groups of the 4-dimensional compact euclidean space forms are listed in Table D. We note in passing that there are at least two instances ([1], [5]) in the literature on compact euclidean space forms where the spaces with vanishing first Betti number present a special problem to the structure theory. Now $b_1 = 0$ for one of the ten 3-dimensional spaces, and Table D shows that in dimension 4 it happens in only 4 cases out of 73.

We turn now to the isomorphism problem itself. Let G_1 and G_2 be two crystallographic extensions of the same $F \subset GL(n,Z)$ with Frobenius maps s_1 and s_2. Let k be the order of F and e_ρ the exponent of $H^2(F,Z^n)$, so that the $t_i = e_\rho s_i$ are functions $F \to Z^n$. By Bieberbach's theorem, G_1 and G_2 are isomorphic if and only if there is a matrix $X \in GL(n,Z)$ which normalizes F and for which the congruences

$$(1) \quad \forall A \in F \qquad s_2(A) \equiv Xs_1(X^{-1}AX) + (A-I)u \pmod{Z^n}$$

have a solution $u \in Q^n$. Here the element $(X,s) \in GL(n,Z)\cdot Q^n$ conjugates G_1 to G_2. Moreover, one can see from the proof in [26] (involving an "averaging over F") that if a solution u exists then one may always be found in $(1/ke_\rho)Z^n$. Zassenhaus describes an efficient technique for finding all the isomorphisms obtaining among the elements of $H^2(F,Z^n)$ if one has a finite set of generators of the normalizer of F in GL(n,Z), and the author [17] has indicated an approach to the algorithmic construction of such a generating set. However, in the present computation we have taken a more direct, if less elegant, tack, because it proves entirely feasible for our problem and allows the use of several of the subroutines developed for the preceding steps.

The method is in the same spirit as our solution of the arithmetical equivalence problem of Step I, but rather more complicated. Once again, it suffices to consider the congruences (1) for all A in a set of generators $\{A_1,\ldots,A_\mu\}$ of F. We introduce new indeterminates $z_i \in Z^n$, $i = 1,\ldots,\mu$, converting the congruences into equations, which we then multiply by ke_ρ to obtain equations with integer coefficients. Thus, G_1 and G_2 are isomorphic if and only if, for some automorphism σ of F the equations

(a) $$A_iX - X\sigma(A_i) = 0$$

(2)(b) $$kXt_1(\sigma(A_i)) + (A_i - I)u + ke_\rho z_i = kt_2(A_i) \qquad i=1,\ldots,\mu$$

(c) $$\det X = \pm 1$$

have a solution by an nxn unimodular integer matrix X and vectors $u,z_1,\ldots,z_\mu \in Z^n$. For each σ (2ab) presents a system of $\mu(n^2+1)$ inhomogeneous linear equations in $n^2 + \mu n + n$ unknowns, and the left side of (2c) is an n-th degree homogeneous polynomial in the n^2 coefficients of X. Let $(2b)^\circ$ denote the homogeneous form of equation (2b), i.e. replace $kt_2(A_i)$ by 0. The comments of Chapter 1 restricting the automorphisms σ that we must consider apply here as well.

For fixed σ , G_1, G_2, all the matrices X belonging to solutions $\{X,u,z_1,\ldots,z_\mu\}$ of (2ab) are represented by a matrix Ξ whose components are inhomogeneous linear forms in some indeterminates $\xi_1,\ldots,\xi_r$, and the question of the isomorphism of G_1 and G_2 is reduced to the question of the existence of integer solutions to one of the polynomial equations $\det\Xi = \pm 1$ for some admissible σ. Once again, the linear algebra and the computation of the polynomial proceed automatically and the latter is printed out only if it satisfies some obvious necessary conditions. Deciding the existence of integer roots still proceeds smoothly.

We have not applied this method to find all the isomorphism classes of crystallographic extensions of F, but only to decide the isomorphisms for the torsion-free extensions, and these were first divided into classes of common commutator quotient groups. Further, in the program we do not solve (2ab) in one step as implied above, but in three steps, solving first (2a), then (2b)°, then finally (2b). This breakdown is made because (2a) depends only on σ and therefore must be solved only once for each σ admissible for F; and (2b)° depends only on σ and the group G_1, so that if there are m groups in one of our classes, then (2b)° must be solved at most m-1 times for each σ , while (2b) must be solved m(m-1)/2 times; and further, the full matrix of coefficients of (2ab) was sometimes too big to work with in the machine memory available to us.

In order to justify the code listed in Table E, we will show in a little greater detail how this procedure has been reduced to the mechanical manipulation of integer matrices.

We construct the $\mu n^2 x n^2$ matrix of coefficients M_σ of (2a), and regard it as a map $M_\sigma : Z^{n^2} \to Z^{\mu n^2}$ whose kernel is the solution set of (2a) and hence contains all X represented by Ξ. Apply FTAG: $L_\sigma M_\sigma K_\sigma = \mathrm{diag}\ (e_1, \ldots, e_{r_\sigma}, 0, \ldots, 0)$. Let $d_\sigma = n^2 - r_\sigma$ and let K'_σ be the rightmost d_σ columns of K_σ; these columns form an integral basis for ker M_σ. That is, we have expressed the general matrix solution to (2a) as a matrix of homogeneous linear forms in some variables $w_1, \ldots, w_{d_\sigma}$.

Substitution of these forms for the components of X in (2b) makes the latter into a system of μn inhomogeneous linear equations in $d_\sigma + \mu n + n$ unknowns. Let M_b be the matrix of coefficients of this system, and regard it as a map $M_b : Z^{d_\sigma} \oplus Z^{\mu n + n} \to Z^{\mu n}$. We have decomposed the domain in order to emphasize that the leftmost d_σ columns of M_b contain the coefficients of the w_i in (2b) and that we are ultimately not interested in the values, but rather only the existence of, the u, z_i, in the solutions. Now ker M_b is the solution set of (2b)°. Apply FTAG: $L_b M_b K_b = \mathrm{diag}(f_1, \ldots, f_{r_b}, 0, \ldots, 0)$. Let $d_b = d_\sigma + \mu n + n - r_b$ and let K''_b be the upper left $d_\sigma x r_b$ corner and K'_b the upper right $d_\sigma x d_b$ corner of K_b. The rightmost d_b columns of K_b form an integral basis for ker M_b and the columns of K'_b generate $\mathrm{proj}_{d_\sigma}(\ker M_b)$ over Z. But we need a basis for the projection, so apply FTAG. $L_c K'_b K_c = \mathrm{diag}(g_1, \ldots, g_{r_c}, 0 \ldots, 0)$. Let L'_c be the leftmost r_c columns of $K'_b K_c$. Then the columns of

$K'_\sigma L'_c$ form an integral basis for the set of all X in Z which satisfy (2a) and for which there are solutions of (2b)°. All these, and only these, are represented by the matrix Ξ° of homogeneous linear forms given by:

$$\Xi^\circ_{ij} = \sum_{k=1}^{r_c} (K'_\sigma L'_c)_{ck}\, \xi_k ,$$

where $c = i + (j-1)n$ is the index of the column of M_σ containing the coefficients of the (ij) component of X in (2a).

The solution set of (2b) is a coset of ker M_b, so we need to find only one solution. Write the right side of (2b) as a vector $\bar{v} = k[t_2(A_1),\ldots,t_2(A_\mu)] = [v_1,\ldots,v_{\mu n}]$. Now one can check that there is no solution and so no isomorphism $G_1 \approx G_2$ for σ, unless

$$f_i \mid (L_b\bar{v})_i \qquad 1 \le i \le r_b$$

$$(L_b\bar{v})_i = 0 \qquad r_b < i \le \mu n \quad ,$$

in which case a solution is given by

$$\bar{u} = [(L_b\bar{t})_1/f_1,\ldots,(L_b\bar{t})_{r_b}/f_{r_b},0,\ldots,0].$$

Finally,

$$\Xi_{ij} = \Xi^\circ_{ij} + (K'_\sigma K''_b\bar{u})_\alpha \quad , \qquad \alpha = i + (j-1)n,$$

gives an nxn matrix of inhomogeneous linear forms in the variables ξ_k, such that the existence of integer solutions of one of the polynomial equations det $\Xi = \pm 1$ is equivalent to the existence of an isomorphism $G_1 \approx G_2$ inducing σ on F.

The necessary condition:

gcd {coefficients of $\det\Xi$} = 1

and gcd {coefficients of non-constant terms}

divides one of (coefficient of constant term) $\pm$ 1

is tested before printing out the polynomial.

The foregoing constitutes a fairly faithful description of the flow in the core parts of the programs ISG/ISH.

Chapter 5

The Computer Programs

The mechanical parts of this computation were accomplished on the Control Data 3150 computer at Humboldt State College. This machine has a core storage of 2^{14} 24-bit words, of which about 8000 are available to the programmer over the operating system and system I/O routines. Available memory was the principal constraint posed by the machine. Nevertheless, the running versions of the codes have sufficient storage allocated to carry the computations into dimension 5 for the groups which can be generated by 3 elements. This was achieved by using a high degree of segmentation of functions; i.e., we have many small main programs, each doing a small part of the work on one pass over all the data, with intermediate results stored on magnetic tape. On the other hand, these main programs share to a high degree their workhorse subroutines.

The constraint posed by the machine's speed (3.5 microsecond storage access time) was negligible. The 16K operating system does not include run timing, so we have no data on the speed of the programs. Suffice it to say that run time was always small when compared both with turnaround time and machine idle time.

The effect of the hardware constraint of size was intensified by the existence of a countermanding psychological constraint, viz. the desire to keep the programs simple and therefore finitely debuggable and ultimately believable. The coding for the most part is in Fortran [28], with the few exceptional subroutines coded in the Control Data assembly language Compass.[27] In Table E we give the source language listings of the actual versions used to obtain our main result.

It will be noted that there is a fair amount of redundancy in our computations. For example, in some places we will carry out the arithmetic to test whether A is equivalent to C, ignoring the fact that we have already established, say, both that A is equivalent to B and that B and C are not equivalent. These redundancies originated in our deeming too dear the cost in program complexity of their removal. However, they proved to be a definite advantage when noted inconsistencies among redundant results revealed the presence of subtle and long undetected bugs. And the ultimately achieved complete consistency among all the redundancies adds significantly to our confidence in the accuracy of our results.

Almost all of the codes were thoroughly tested on the analogous and previously known 3-dimensional cases before application to our main problem. But we know, by the bugs undetected in this phase and later discovered through inconsistent 4-dimensional results, that the 3-dimensional test did not explore all of the logical loops and branches.

We conclude this paper with a few remarks on each of the listed programs to aid the reader who wishes to study them. We begin with the Compass codes.

Subroutine DIAR.

This is our implementation of the Fundamental Theorem of

Generated Abelian Groups, FTAG. The algorithm is a slight modification of the constructive proof in [13]. We have coded it in assembly language because it is our main workhorse and it is long, whereas our Fortran compiler is particularly space-wasteful.

The main entry-point calling sequence is CALL DIAKL (NL,ML,NK,MK,NM,MM,KK,NR,NC).

The inputs are: MM, an arbitrary NR x NC integer matrix; NM, NL, NK, the calling program row-dimensions of the matrices MM, ML, MK (resp.).

The outputs are: MM, the diagonal form of the input MM as described in Chapter 0; ML and MK, the left and right transforming unimodular matrices; KK, the number of non-zero diagonal elements.

The transforming matrices are computed by beginning with identity matrices and performing at each "pivot step" elementary row and column operations corresponding to those applied to MM.

There are two other entry-points, DIAK, DIALI, each with an appropriately truncated calling sequence. The former omits the computation of the left transforming matrix and the latter omits the computations of the right transforming matrix while computing the inverse of the left transforming matrix.

DIAR is called actually by only one of the main programs, ISG, which is the only one which requires all of the functions in core at one time. The identifiers DIAGK, DIAGL, DIAG, etc., appearing in the remaining main programs, are entry-points to other versions which differ from DIAR only in omitting one or more of its functions with respect to the transforming matrices. We have omitted these shorter versions from Table E.

<u>Subroutine</u> <u>MMPYR</u>.

This is our integer matrix multiplier. The main entry-point calling sequence CALL MMPY1(KA,KB,KC,N1,N2,N3,0,MA,MB,MC) multiplies the N1 × N2 matrix MA by the N2 × N3 matrix MB, returning the product in MC. KA, KB and KC are the calling-program row dimensions of these matrices. The alternate entry points enable one to set in the subroutine some of the values or addresses of the input parameters, and to enter without passing some input values or addresses if unchanged since the previous call.

<u>Function</u> <u>IGCDR</u>.

IGCD(N,M) has the value of the greatest common divisor of the input integers N,M; Euclid's algorithm coded in Compass in order to exploit a certain feature of the hardware; viz., on a fixed-point divide, the quotient is left in one register (A), while the remainder is left in another (Q).

<u>Subroutine</u> <u>POLYR</u>.

This is our co-routine for manipulating integer polynomials in several variables. It is used with Program VC in Step I and with Program ISH in Step IV.

A polynomial is represented by a linked list. Each term of the polynomial is represented by a <u>cell</u> in the list. A cell consists of three consecutive words of core storage; but the cells representing consecutive terms of the polynomial are not necessarily consecutive in core. The first word of a cell contains the absolute core address of the cell representing the succeeding term, zero in the cell representing the last term. The calling Fortran program knows the polynomial by an integer identifier naming a core location which contains the absolute core address of the cell representing the first term.

The second word in each cell gives the variable part of the term. Polynomials in up to 23 variables are allowed; each variable is coded by an integer between 1 and 23 and zero codes the constant term. The product of variables comprising the variable part of the term is coded by writing the variables in a string with the higher variables to the right and repetitions denoting powers, and then taking the value of this string read as a base-24 numeral. Thus, with our 24-bit word, POLYR is limited to polynomials of degree ≤ 5.

The third word in each cell contains the integer coefficient of the term.

A single block of storage, STACK + 1 to STACK + 999, is allocated for all the polynomial lists. At any time, all cells not currently in use by some polynomial are linked together through their first words into one list of free cells which operates as a pushdown stack. Location STACK contains the absolute core address of the first free cell.

We now describe the functions by calling sequence.

CALL CLSTK.

This erases all existing polynomials by linking all the cells of allocated storage into the stack of free cells.

CALL CRPHDI(N,MV,MC,KP)

This creates a list representing the 1st degree polynomial with N terms whose variables (integers between 1 and 23) are given in the array MV and whose coefficients are given in the array MC. On return, KP contains the absolute core address of the first cell of the list.

CALL PDCOD(KP,MV,MC,NT,NTM)

This supplies to the arrays MV and MC of the calling program the variable-part codes and coefficients of the polynomial KP. The output parameter NT gives the number of terms, while the input parameter NTM specifies the maximum number of terms which the calling program can accept.

CALL FREPOL(KP)

erases polynomial KP by attaching its tail to the head of the stack.

CALL PML1(KP,KQ,KR,$\pm$1).

According to the sign of the fourth parameter, this replaces polynomial KR by KR $\pm$ KP·KQ, where KP is assumed to be of degree ≤ 1.

CALL UNHMG(KP,MC)

This assumes that KP is a polynomial without constant term, and adds to it, as leading term, the constant MC

CALL HOMOG(KP)

removes from KP its first term provided that the latter is constant.

Now we go on to the Fortran codes, beginning with the more general subroutines.

Function IDET4(M)

has as its value the (absolute core location of) 4th degree polynomial defined as the determinnant of the 4 x 4 input matrix M whose components are polynomials of degree ≤ 1.

Subroutine NITTY (NA,NB,MU,MA,MB,MDEL).

The first five parameters are input. They specify a set of MU NA×NA matrices stored in the array MA and a set of MU NB×NB matrices contained in the array MB. Write the matrix equations

(1) $MA_iX - XMB_i = 0$ $\qquad$ $i = 1,\ldots,MU$

Where X is an indeterminate NA×NB matrix. Then the output MDEL is the (NANBMU)x(NANB) matrix of coefficients of equations (1) considered as a system of NANBMU equations in the NANB unknown components X_{jk} of X. The coefficients of the (jk) component of the ith equation of (1) are contained in row $j + (k-1)NA + (i-1)NANB$ of MDEL; the coefficients of the variable X_{jk} are contained in column $j + (k-1)NA$. NITTY is called by Programs VB, VC and ISG. Our Fortran system requires agreement between calling program and subroutine in the row dimension of matrix parameters, so several versions of NITTY, differing only in their DIMENSION statements, are used. Only one version is listed in Table E. Similar remarks are applicable to all of our Fortran matrix-manipulation subroutines and will not be repeated.

Subroutine MATP(ND,NP,MA,MB)

returns to MB the NPth power of the ND×ND matrix MA. The alternate entry-point MATT returns to MB instead the sum of the powers 0 through NP of MA. MATT/MATP is called by Subroutine WORDC and by Program FINDFLAT.

Subroutine WORDC(ND,LI,LE,MG,MGA)

The inputs are: ND, the dimension; (LI, LE), a word as explained under Program ACCHK; MG, an array containing a set of

NDxND matrices. The output is MGA, the matrix obtained by substituting the matrices of MG for the corresponding letters of the word. Called by VC.

Subroutine WORDV(LI,LE,MX,MI).

The input is a word W specified by the arrays (LI,LE) as explained under Program ACCHK. The output array MX contains the matrix values $W^{(j)}$, $1 \leq j \leq MU$, of the function defined in Chapter 2. The generating matrices are obtained through COMMON storage. The output MI is the matrix value of W. WORDV is called by Programs ACCHK and ISG.

We proceed now to the main programs. Each of these produces copious printed output displaying the input data as well as the computed results, and we will generally not mention this printout in the I/O comments in each case.

Program VB.

This program carries out the first part of Step I, the computation of the cohomology group of binding functions. The input is a deck of cards consisting of a subdeck for each geometrical crystal class. The first card of a subdeck carries a few parameters identifying and describing the class. Then follows a set of NAUT automorphisms of the class to be used by Program VC in deciding arithmetical equivalence as described in Chapter 1. Each automorphism is specified by giving the images of the generators as words in the generators. See under Program ACCHK for the coding of words. Then follows 2·NDEC·MU cards carrying matrix representives of the generators, thus defining NDEC (NA+NB)-decomposed representations of the class. The components of these are stored in the arrays MA and MB. The cohomology group of binding functions is computed as described in Chapter 1. In addition to the printout, the output consists of

(i) punched cards carrying the representations of degree ND = NA+NB defined by all the elements of all the cohomology groups computed for the geometrical class. These are in the proper format for input to Program ACCHK, but of course contain perhaps several representatives of each arithmetical crystal class. It is at this point that the arithmetical classes acquire the "AC" designations of Table C.

(ii) A scratch tape carrying all the input data and computed results for immediate use by the next program.

<u>Program VC</u>

This reads through the scratch tape just written by VB. For each geometrical crystal class and each pair of representatives of arithmetical classes generated by VB, and each of the given automorphisms, VC computes the polynomial detΞ according to the scheme of Chapter 1, printing it out if it satisfies the necessary condition mentioned. After human verification of the existence of integer roots, the redundant representatives are removed from the deck punched by VB.

<u>Program ACCHK.</u>

This code performs some editorial pre-processing on the data to be used by the programs of Steps II, III, and IV. The input is a deck of cards comprised of a number of subdecks each pertaining to an isomorphism class of point groups. The first card of a subdeck carries identifying information and some parameters of the group, e.g. number of generators, number of relators, etc. Then comes a set of cards giving the relators of a presentation for the group. Each relator is a word in the generators and is coded as a sequence of up to 4 pairs (LRI,LRE) of positive integers; each pair designates a letter in the word, LRI naming the generator and LRE giving the exponent. The relators are followed by cards carrying the elements of a torsion test set, specified also as words in the generators, and also the orders of these elements. The remainder of the isomorphism class subdeck consists of the cards punched out by Program VB giving arithmetical crystal classes of the same isomorphism class.

In each arithmetical class, ACCHK begins by verifying that the input generating matrices indeed satisfy the given relators. During this test the matrix R of Chapter 2, designated MR in the code, is produced. Incidentally, this matrix has the property that, if multiplied on the right by the ($n\mu$)-component column vector formed from the values on the generators of a Frobenius map for an extension of the class, then the components of the product are just the $\beta(jk)$ of the presentation for the extension as defined in Chapter 2.

Next, ACCHK computes, from the given generating matrices and torsion test words, the corresponding matrix representatives of the torsion test elements (array MT), and verifies that these matrices have the given orders.

Finally, the program constructs the matrix MRT with the property that multiplying it by the vector containing the values on the generators of a Frobenius map yields the values on the torsion test elements of a Frobenius map for the same extension.

All of the input and computed result are written on a scratch tape for use by the next program.

Program HILB18.

The computations of $H^2(F,Z^n)$ now consists merely of diagonalizing the matrix MR found on the tape written by ACCHK.

The output tape produced by HILB18 is saved as the main source of data for Steps III and IV. This tape contains several Fortran logical records for each arithmetical crystal class. The first record for each class contains almost all the information on the class so far accumulated; including the parameters of $H^2(F,Z^n)$. Then for each nonzero element of the latter there is a short record carrying certain parameters of the corresponding extension: identifiers; values of a Frobenius map on the generators and torsion test elements, and the $\beta(jk)$ of the presentation. The code provides the ability to update this tape.

Program FINDFLAT:

This determines, according to the scheme of Chapter 3, whether each crystallographic extension represented on the HILB18 output tape is torsion-free. In each arithmetical class and for each torsion test element A, the computation of trA and its diagonalization are not performed until needed, and then are saved should they be needed again. However, for the p-th power of a torsion test element the same computations are carried out anew each time they are needed.

Program ABELR.

computes the commutator quotient group of each extension on the HILB18 output tape, as described in Chapter 4.

Program ISG:

The input is the HILB18 output tape plus a deck of cards specifying sets of extensions to be tested for isomorphism and the admissible automorphisms for the arithmetical crystal classes involved.

Until this point in the computation, no attention was paid to the possible choices (mod Z^n) for the Frobenius maps. But the polynomials produced in Step IV were in some cases rendered more tractable by beginning with Frobenius maps with small components; so, ISG has an (internal) subroutine which chooses the Frobenius map with all of its values on the generators in the cube $0 \leq s_i < 1$, $i = 1,2,3,4$.

The core part of this code was detailed in Chapter 4. In each instance of equations (2ab) the code reports in the printout whether solutions exist; when they do, then the matrix Ξ of inhomogeneous linear forms is saved on a scratch tape for

Program ISH

whose function is analogous to that of Program VC in Step I.

The short editing codes which produced Tables C and D from the data tapes are omitted from the listings in Table E.

Bibliography

1. L. Auslander and M. Kuranishi, On the holonomy group of locally euclidean spaces, Ann. Math, 65, 1957, p.411.

2. L. Bieberbach, Uber die Bewegungsgruppen der Euklidischen Raume I, Math. Ann., 70, 1911, p.297; II, ibid 72, 1912, p.400.

3. J. J.Burckhardt, Die Bewegungsgruppen der Kristallographie, Birkauser Verlag, Basel, second edition, 1966.

4. R. Bulow and J. Neubuser, On some applications of group-theoretical programmes to the derivation of the crystal classes of R_4 in [16], p.131.

5. E. Calabi, Closed, locally euclidean, 4-dimensional manifolds, Bull.A.M.S., 63, 1957, Abstract 295, p.393.

6. L.S. Charlap, Compact flat riemannian manifolds I, Ann. Math., 81, 1965, p.15.

7. L.S. Charlap and C.H. Sah, Compact flat riemannian 4-manifolds, (Pre-preliminary report), unpublished.

8. C. W. Curtis and I. Reiner, Representation theory of finite groups and associative algebras, Wiley, New York, 1962, Chapter IX .

9. E. C. Dade, The maximal finite groups of 4x4 integral matrices, Illinois J. Math., 9, 1965, p.99.

10. E. Goursat, Substitutions orthogonales, Ann. Ec. Norm (3), 6, 1889, p.9.

11. A. C. Hurley, Finite rotation groups and crystal classes in four dimensions, Proc Camb. Phil. Soc., 47, 1951, p.650.

12. A. C. Hurley, J. Neubuser, and H. Wondratschek, Crystal classes of four-dimensional space R4., Acta. Cryst., 22, 1967, p605.

13. N. Jacobson, Lectures in abstract algebra, Vol. II, Van Nostrand, New York, 1953, p.79.

14. T. Janssen, Crystallographic groups in space and time III. Four-dimensional euclidean crystal classes corresponding to generalized magnetic point groups, Physica, 42, 1969, p.71.

15. T. Janssen and G. Fast, Non-equivalent generalized magnetic space-time-groups in four dimensions, Tech.Report 6-68, Katolieke Universiteit Nijmegen, the Netherlands.

16. J. Leech, ed., Computational problems in abstract algebra, Pergamon, London, 1970.

17. R. Levine, On the normalizer of a finite unimodular subgroup, unpublished.

18. A.L. Mackay and G.S. Pawley, Bravais lattices in four-dimensional space, Acta. Cryst., 16, 1963, p.11.

19. S. MacLane, Homology, Academic Press, New York, 1963.

20. W. Magnus, A. Karrass, and D. Solitar, Combinatorial group theory: Presentations of groups in terms of generators and relations, Wiley, New York, 1966.

21. I. Reiner, Integral representations of cyclic groups of prime order, Proc. A.M.S., 8, 1957, p.142.

22. I. Reiner, A survey of integral representation theory, Bull. A.M.S., 76, 1970, p.159.

23. W. Threlfall and H. Seifert, Topologische Untersuchung der Diskontinuitatsbereiche endlicher Bewegungsgruppen des dreidimensionalen spharischen Raumes, Math. Ann., 104, 1930, p. 1.

24. J. A. Wolf, Spaces of constant curvature, McGraw-Hill, New York, 1967.

25. H. Zassenhaus, Neuer Beweis der Endlichkeit der Klassenzahl bei unimodularer Aquivalenz endlicher ganzzahliger Substitutionsgruppen, Hamb. Abh., 12, 1938, p.276.

26. H. Zassenhaus, Uber einen Algorthimus zur Bestimmung der Raumgruppen, Comm. Math. Helv., 21, 1948, p.117.

27. Control Data Corp., Computer Systems Compass Reference Manual, Pub. No. 60236800A.

28. Control Data Corp., Computer Systems Fortran Reference Manual, Pub. No. 60057600, Rev. B.

APPENDIX

Tables

Table A

This table displays the 45 4-dimensional geometrical crystal classes which satisfy Condition A.

Column G: The symbol GCk, $0 \leq k \leq 44$, is an arbitrary name we have given the class for reference. Classes GC0 to GC10 are orientation preserving; GC11 to GC44 are not.

Column I: The abstract group. C_k cylic of order k

D_k dihedral of order 2k

T tetrahedral

Ø octahedral

Column A: The number of arithmetical classes into which the geometrical class splits.

Column H: The number of the latter which occur as holonomy classes of compact euclidean space forms.

Column Hurley: The very useful notation of [11, 12, 18] which gives the distribution of elements in the conjugate classes of GL(4,Q). The latter which contain crystal classes are characterized by the order k and coefficients χ, σ, d, of the characteristic equation: $\lambda^4 - \chi\lambda^3 + \sigma\lambda^2 - \chi\lambda + d = 0$

Here is a short dictionary for those with non-zero fixed space:

Subtable Hurley

	I	E	F	K	N	N'	R	T	T'	Z	
k	1	2	4	3	6	6	4	2	2	6	order
χ	4	0	0	1	1	-1	2	2	-2	3	trace
σ	6	-2	0	0	0	0	2	0	0	4	second invariant
d	1	1	-1	1	-1	-1	1	-1	-1	1	determinant

Table A

G	I	A	H	Hurley
GC0	C_1	1	1	1I
GC1	C_2	3	2	1I 1E
GC2	$C_2 x C_2$	13	6	1I 3E
GC3	C_4	2	2	1I 1E 2R
GC4	D_4	6	2	1I 5E 2R
GC5	C_3	2	2	1I 2K
GC6	C_6	1	1	1I 1E 2K 2Z
GC7	D_3	8	3	1I 3E 2K
GC8	D_6	2	1	1I 7E 2K 2Z
GC9	T	6	2	1I 3E 8K
GC10	∅	5	0	1I 9E 8K 6R
GC11	C_2	2	2	1I 1T
GC12	C_2	2	1	1I 1T'
GC13	$C_2 x C_2$	6	5	1I 1E 2T
GC14	$C_2 x C_2$	7	2	1I 1E 1T 1T'
GC15	$C_2 x C_2$	6	1	1I 1E 2T'
GC16	$C_2 x C_2 x C_2$	12	1	1I 3E 3T 1T'
GC17	$C_2 x C_2 x C_2$	12	1	1I 3E 1T 3T'
GC18	C_4	7	4	1I 1E 2F
GC19	$C_4 x C_2$	6	2	1I 1E 2F 2R 1T 1T'
GC20	D_4	2	0	1I 1E 2R 4T

Table A (Page 2)

G	I	A	H	Hurley
GC21	D_4	2	0	1I 1E 2R 4T'
GC22	D_4	13	1	1I 3E 2F 2T
GC23	D_4	13	1	1I 3E 2F 2T'
GC24	D_4xC_2	6	0	1I 5E 2F 2R 5T 1T'
GC25	D_4xC_2	6	0	1I 5E 2F 2R 1T 5T'
GC26	D_3	3	0	1I 2K 3T
GC27	D_3	3	0	1I 2K 3T'
GC28	C_6	4	1	1I 2K 2N'1T
GC29	C_6	4	2	1I 2K 2N 1T'
GC30	C_6xC_2	2	1	1I 1E 2K 2N 2N'1T 1T'2Z
GC31	D_6	1	0	1I 1E 2K 6T 2Z
GC32	D_6	1	0	1I 1E 2K 6T'2Z
GC33	D_6	6	0	1I 3E 2K 2N'4T
GC34	D_6	6	0	1I 3E 2K 2N 4T'
GC35	D_6	6	0	1I 3E 2K 2N 3T 1T'
GC36	D_6	6	0	1I 3E 2K 2N 1T 3T'
GC37	D_6xC_2	2	0	1I 7E 2K 2N 2N'7T 1T'2Z
GC38	D_6xC_2	2	0	1I 7E 2K 2N 2N'1T 7T'2Z
GC39	TxC_2	5	0	1I 3E 8K 8N'1T 3T'
GC40	TxC_2	5	1	1I 3E 8K 8N 3T 1T'
GC41	∅	6	0	1I 3E 6F 8K 6T'

Table A (Page 3)

G	I	A	H	Hurley
GC42	$\emptyset$	6	0	1I 3E 6F 8K 6T
GC43	$\emptyset xC_2$	5	0	1I 9E 6F 8K 8N'6R 1T 9T'
GC44	$\emptyset xC_2$	5	0	1I 9E 6F 8K 8N 6R 9T 1T'

Table B

This table displays some information input to the programs which depends only on the isomorphism class of the point group; namely, presentations and torsion test sets.

The column headed NG gives the number of generators for the presentation. All the groups in question are generated by three or fewer elements, and these are designated A,B,C.

The column headed Relators gives a complete set of relators for the presentation.

The column headed Torsion Test gives a torsion test set as defined in Chapter 3 and as used in the computations.

Table B

Group	NG	Relators	Torsion Test
C_2	1	A^2	A
C_2xC_2	2	$A^2, B^2, (AB)^2$	A,B,AB
$C_2xC_2xC_2$	3	A^2, B^2, C^2	A,B,C,AB,AC
		$(AB)^2, (AC)^2, (BC)^2$	BC,ABC
C_4	1	A^4	A
D_4	2	$A^4, B^2, (AB)^2$	A,B,AB
C_4xC_2	2	A^4, B^2, A^3BAB	A,B,AB,A^2B
D_4xC_2	3	$A^4, B^2, C^2, (AB)^2$	A,B,C,AB
		$A^3CAC, (BC)^2$	AC,BC,A^2C
C_3	1	A^3	A
D_3	2	$A^3, B^2, (AB)^2$	A,B
C_6	1	A^6	A
D_6	2	$A^6, B^2, (AB)^2$	A,B,AB
C_6xC_2	2	A^6, B^2, A^5BAB	A,AB,A^2B
D_6xC_2	3	$A^6, B^2, C^2, (AB)^2$	A,B,AB,AC
		$A^5CAC, (BC)^2$	A^2C,BC,ABC
T	2	$A^3, B^3, (AB)^2$	A,B,AB
TxC_2	3	$A^3, B^3, C^2, (AB)^2,$	AB,AC,ABC
		A^2CAC, B^2CBC	
$\emptyset$	2	$A^3, B^4, (AB)^2$	A,B,AB

Table C

This table displays the 49 4-dimensional arithmetical classes of holonomy.

The first, unheaded, column contains 3 entries for each class. The first, AC k, is an arbitrary designator used for reference on our data tapes. The second and third give the geometrical class and isomorphism class to which the arithmetical class belongs, in the same notation as Table A.

The 3 columns headed GEN A,B,C give matrix representatives of a set of generators for the class; these correspond to the generators A,B,C of the presentations of Table B.

The last column headed NFLAT gives the number of inequivalent compact euclidean space forms of the class.

TABLE C P. 1

	GEN A	GEN B	GEN C	NFLAT
AC 0	1 0 0 0			1
GC0	0 1 0 0			
C1	0 0 1 0			
	0 0 0 1			
AC 1	-1 0 0 0			1
GC1	0 -1 0 0			
C2	0 0 1 0			
	0 0 0 1			
AC 2	-1 0 0 1			1
GC1	0 -1 0 0			
C2	0 0 1 0			
	0 0 0 1			
AC 7	-1 0 0 0	1 0 0 0		4
GC2	0 -1 0 0	0 -1 0 0		
C2XC2	0 0 1 0	0 0 -1 0		
	0 0 0 1	0 0 0 1		
AC 8	-1 0 0 1	1 0 0 0		1
GC2	0 -1 0 0	0 -1 0 0		
C2XC2	0 0 1 0	0 0 -1 0		
	0 0 0 1	0 0 0 1		
AC 10	-1 0 0 1	1 0 0 0		1
GC2	0 -1 0 1	0 -1 0 1		
C2XC2	0 0 1 0	0 0 -1 0		
	0 0 0 1	0 0 0 1		
AC 14	-1 0 0 1	1 0 0 0		1
GC2	0 -1 0 1	0 -1 0 1		
C2XC2	0 0 1 0	0 0 -1 1		
	0 0 0 1	0 0 0 1		

TABLE C P. 2

	GEN A				GEN B				GEN C	NFLAT
AC 23	-1	0	0	0	0	1	0	0		1
GC2	0	-1	0	0	1	0	0	0		
C2XC2	0	0	1	0	0	0	-1	0		
	0	0	0	1	0	0	0	1		
AC 24	-1	0	0	1	0	1	0	0		1
GC2	0	-1	0	1	1	0	0	0		
C2XC2	0	0	1	0	0	0	-1	0		
	0	0	0	1	0	0	0	1		
AC 27	0	1	0	0						1
GC3	-1	0	0	0						
C4	0	0	1	0						
	0	0	0	1						
AC 28	0	1	0	0						1
GC3	-1	0	0	1						
C4	0	0	1	0						
	0	0	0	1						
AC 30	0	1	0	0	0	1	0	0		3
GC4	-1	0	0	0	1	0	0	0		
D4	0	0	1	0	0	0	-1	0		
	0	0	0	1	0	0	0	1		
AC 31	0	1	0	0	0	1	0	0		1
GC4	-1	0	0	1	1	0	0	0		
D4	0	0	1	0	0	0	-1	0		
	0	0	0	1	0	0	0	1		
AC 36	0	-1	0	0						1
GC5	1	-1	0	0						
C3	0	0	1	0						
	0	0	0	1						

TABLE C P. 3

	GEN A				GEN B				GEN C	NFLAT
AC 37	0	-1	0	0						1
GC5	1	-1	0	1						
C3	0	0	1	0						
	0	0	0	1						
AC 40	0	1	0	0						1
GC6	-1	1	0	0						
C6	0	0	1	0						
	0	0	0	1						
AC 41	0	-1	0	0	0	1	0	0		1
GC7	1	-1	0	0	1	0	0	0		
D3	0	0	1	0	0	0	-1	0		
	0	0	0	1	0	0	0	1		
AC 45	0	-1	0	0	0	-1	0	0		1
GC7	1	-1	0	0	-1	0	0	0		
D3	0	0	1	0	0	0	-1	0		
	0	0	0	1	0	0	0	1		
AC 47	0	-1	0	0	0	-1	0	0		1
GC7	1	-1	0	2	-1	0	0	0		
D3	0	0	1	0	0	0	-1	2		
	0	0	0	1	0	0	0	1		
AC 51	0	1	0	0	0	1	0	0		1
GC8	-1	1	0	0	1	0	0	0		
D6	0	0	1	0	0	0	-1	0		
	0	0	0	1	0	0	0	1		
AC 53	0	0	1	0	0	-1	0	0		1
GC9	1	0	0	0	0	0	-1	0		
T	0	1	0	0	1	0	0	0		
	0	0	0	1	0	0	0	1		

TABLE C P. 4

	GEN A				GEN B				GEN C	NFLAT
AC 54	0	0	1	0	0	-1	0	1		1
GC9	1	0	0	0	0	0	-1	1		
T	0	1	0	0	1	0	0	0		
	0	0	0	1	0	0	0	1		
AC 66	1	0	0	0						1
GC11	0	1	0	0						
C2	0	0	-1	0						
	0	0	0	1						
AC 67	1	0	0	0						1
GC11	0	1	0	0						
C2	0	0	-1	1						
	0	0	0	1						
AC 69	-1	0	0	0						1
GC12	0	-1	0	0						
C2	0	0	-1	0						
	0	0	0	1						
AC 77	-1	0	0	0	-1	0	0	0		6
GC13	0	-1	0	0	0	1	0	0		
C2XC2	0	0	1	0	0	0	1	0		
	0	0	0	1	0	0	0	1		
AC 78	-1	0	0	1	-1	0	0	1		3
GC13	0	-1	0	0	0	1	0	0		
C2XC2	0	0	1	0	0	0	1	0		
	0	0	0	1	0	0	0	1		
AC 80	-1	0	0	1	-1	0	0	1		1
GC13	0	-1	0	1	0	1	0	0		
C2XC2	0	0	1	0	0	0	1	0		
	0	0	0	1	0	0	0	1		

TABLE C P. 5

	GEN A				GEN B				GEN C				NFLAT
AC 81	0	1	0	0	1	1	1	0					1
GC13	1	0	0	0	0	0	-1	0					
C2XC2	-1	-1	-1	0	0	-1	0	0					
	0	0	0	1	0	0	0	1					
AC 84	-1	0	0	0	0	-1	0	0					1
GC13	0	-1	0	0	-1	0	0	0					
C2XC2	0	0	1	0	0	0	1	0					
	0	0	0	1	0	0	0	1					
AC 88	-1	0	0	0	-1	0	0	0					4
GC14	0	-1	0	0	0	-1	0	0					
C2XC2	0	0	1	0	0	0	-1	0					
	0	0	0	1	0	0	0	1					
AC 96	-1	0	0	0	-1	0	0	0					1
GC14	0	0	-1	0	0	-1	0	0					
C2XC2	0	-1	0	0	0	0	-1	0					
	0	0	0	1	0	0	0	1					
AC100	-1	0	0	0	-1	0	0	0					1
GC15	0	-1	0	0	0	-1	0	0					
C2XC2	0	0	1	0	0	0	-1	0					
	0	0	0	-1	0	0	0	1					
AC112	-1	0	0	0	1	0	0	0	-1	0	0	0	10
GC16	0	-1	0	0	0	-1	0	0	0	-1	0	0	
C2XC2XC2	0	0	1	0	0	0	-1	0	0	0	-1	0	
	0	0	0	1	0	0	0	1	0	0	0	1	
AC130	-1	0	0	0	1	0	0	0	-1	0	0	0	2
GC17	0	-1	0	0	0	-1	0	0	0	-1	0	0	
C2XC2XC2	0	0	1	0	0	0	-1	0	0	0	-1	0	
	0	0	0	-1	0	0	0	-1	0	0	0	1	

TABLE C P. 6

	GEN A	GEN B	GEN C	NFLAT
AC148	0 -1 0 0			1
GC18	1 0 0 0			
C4	0 0 -1 0			
	0 0 0 1			
AC150	0 -1 0 0			1
GC18	1 0 0 0			
C4	0 0 -1 1			
	0 0 0 1			
AC151	0 -1 0 0			1
GC18	1 0 0 1			
C4	0 0 -1 1			
	0 0 0 1			
AC152	-1 0 1 0			1
GC18	-1 0 0 0			
C4	-1 1 0 0			
	0 0 0 1			
AC156	0 1 0 0	-1 0 0 0		2
GC19	-1 0 0 0	0 -1 0 0		
C4XC2	0 0 1 0	0 0 -1 0		
	0 0 0 1	0 0 0 1		
AC160	1 0 -1 0	-1 0 0 0		1
GC19	1 0 0 0	0 -1 0 0		
C4XC2	1 -1 0 0	0 0 -1 0		
	0 0 0 1	0 0 0 1		
AC170	0 1 0 0	0 -1 0 0		1
GC22	-1 0 0 0	-1 0 0 0		
D4	0 0 1 1	0 0 1 0		
	0 0 0 -1	0 0 0 1		

TABLE C P. 7

	GEN A				GEN B				GEN C				NFLAT
AC171	0	1	0	0	0	-1	0	0					1
GC22	-1	0	0	1	-1	0	0	0					
D4	0	0	1	1	0	0	1	0					
	0	0	0	-1	0	0	0	1					
AC176	0	1	0	0	0	1	0	0					1
GC23	-1	0	0	0	1	0	0	0					
D4	0	0	1	0	0	0	-1	0					
	0	0	0	-1	0	0	0	1					
AC209	0	-1	0	0									1
GC28	1	-1	0	0									
C6	0	0	-1	0									
	0	0	0	1									
AC215	0	0	-1	0									1
GC29	-1	0	0	0									
C6	0	-1	0	0									
	0	0	0	1									
AC217	0	1	0	0									1
GC29	-1	1	0	0									
C6	0	0	-1	0									
	0	0	0	1									
AC219	0	1	0	0	-1	0	0	0					1
GC30	-1	1	0	0	0	-1	0	0					
C6XC2	0	0	1	0	0	0	-1	0					
	0	0	0	1	0	0	0	1					
AC257	0	0	1	0	0	-1	0	0	-1	0	0	0	1
GC40	1	0	0	0	0	0	-1	0	0	-1	0	0	
TXC2	0	1	0	0	1	0	0	0	0	0	-1	0	
	0	0	0	1	0	0	0	1	0	0	0	1	

Table D

This table displays the 75 4-dimensional compact euclidean space forms.

Under HOLONOMY there are 4 entries for each space. The first 3 name the arithmetical, geometrical, and isomorphism classes of the holonomy group in the same notations as the preceding tables. The 4th entry designates a representative element of $H^2(F,Z^4)$ in the arbitrary notation used in our data tapes.

Under FROBENIUS MAP is given the fundamental group. Columns A,B,C contain the translation parts to be paired with the corresponding generating matrices of Table C to form a set of generators over Z^4 of the deck group. Under DENOM is the common denominator by which all the vectors under A,B,C are understood to be divided.

Under H1(M,Z) is given the first homology group of the space. The sequence of integers k l m ... denotes $(Z/kZ) \oplus (Z/lZ) \oplus (Z/mZ)$ ... In particular, 0 denotes Z.

TABLE D P. 1

HOLONOMY	FROBENIUS MAP A	B	C	DENOM	H1(M,Z)
AC 0	0				0 0 0 0
GC0	0				
C1	0				
1	0			/1	
AC 1	0				2 2 0 0
GC1	0				
C2	1				
1	0			/2	
AC 2	0				2 0 0
GC1	0				
C2	1				
1	0			/2	
AC 7	0	1			4 4 0
GC2	1	0			
C2XC2	1	0			
11	0	0		/2	
AC 7	0	1			2 2 2 0
GC2	0	0			
C2XC2	0	0			
12	1	0		/2	
AC 7	0	1			2 4 0
GC2	1	0			
C2XC2	0	0			
13	1	0		/2	
AC 7	0	1			2 4 0
GC2	1	0			
C2XC2	1	0			
15	1	0		/2	

TABLE D P. 2

HOLONOMY	FROBENIUS MAP A	B	C	DENOM	H1(M,Z)
AC 8	0	1			2 4 0
GC2	0	1			
C2XC2	1	0			
7	0	0		/2	
AC 10	0	1			2 2 0
GC2	0	1			
C2XC2	1	0			
7	0	0		/2	
AC 14	0	1			2 2 0
GC2	0	1			
C2XC2	1	0			
7	0	0		/2	
AC 23	0	0			2 2 0
GC2	0	0			
C2XC2	1	0			
5	0	1		/2	
AC 24	1	0			4 0
GC2	0	0			
C2XC2	1	0			
3	0	1		/2	
AC 27	0				2 0 0
GC3	0				
C4	1				
1	0			/4	
AC 28	0				0 0
GC3	0				
C4	1				
2	0			/4	

TABLE D P. 3

HOLONOMY	FROBENIUS MAP A	B	C	DENOM	HI(M,Z)
AC 30	0	0			2 2 0
GC4	0	0			
D4	1	0			
13	0	2		/4	
AC 30	2	0			4 0
GC4	0	0			
D4	1	0			
14	2	2		/4	
AC 30	2	0			4 0
GC4	0	0			
D4	1	0			
15	0	2		/4	
AC 31	2	0			2 0
GC4	0	0			
D4	1	0			
7	2	2		/4	
AC 36	0				3 0 0
GC5	0				
C3	1				
1	0			/3	
AC 37	0				0 0
GC5	0				
C3	1				
1	0			/3	
AC 40	0				0 0
GC6	0				
C6	1				
1	0			/6	

TABLE D P. 4

HOLONOMY	FROBENIUS MAP A	B	C	DENOM	H1(M,Z)
AC 41	0	0			6 0
GC7	0	0			
D3	2	0			
1	0	3		/6	
AC 45	0	0			2 0
GC7	0	0			
D3	2	0			
1	0	3		/6	
AC 47	0	0			2 0
GC7	0	0			
D3	2	0			
1	0	3		/6	
AC 51	0	0			2 0
GC8	0	0			
D6	1	0			
7	0	3		/6	
AC 53	3	0			0
GC9	0	0			
T	3	0			
1	2	4		/6	
AC 54	3	4			0
GC9	3	0			
T	0	0			
1	4	2		/6	
AC 66	1				2 0 0 0
GC11	0				
C2	0				
1	0			/2	

TABLE D P. 5

HOLONOMY	FROBENIUS MAP A	B	C	DENOM	H1(M,Z)
AC 67	0				0 0 0
GC11	1				
C2	0				
1	0			/2	
AC 69	0				2 2 2 0
GC12	0				
C2	0				
1	1			/2	
AC 77	0	0			2 2 0 0
GC13	0	1			
C2XC2	1	0			
10	0	0		/2	
AC 77	1	0			4 0 0
GC13	0	1			
C2XC2	1	0			
11	0	0		/2	
AC 77	0	0			2 2 0 0
GC13	0	0			
C2XC2	0	1			
20	1	0		/2	
AC 77	1	0			2 0 0
GC13	0	0			
C2XC2	0	1			
21	1	0		/2	
AC 77	1	0			4 0 0
GC13	0	1			
C2XC2	1	1			
27	0	0		/2	

TABLE D P. 6

HOLONOMY	FROBENIUS MAP A	B	C	DENOM	H1(M,Z)
AC 77	1	0			2 0 0
GC13	0	1			
C2XC2	0	1			
29	1	0		/2	
AC 78	0	0			2 0 0
GC13	0	1			
C2XC2	1	0			
5	0	0		/2	
AC 78	1	0			2 2 0 0
GC13	0	0			
C2XC2	1	1			
11	0	0		/2	
AC 78	1	0			2 0 0
GC13	0	1			
C2XC2	1	1			
15	0	0		/2	
AC 80	0	0			2 0 0
GC13	0	1			
C2XC2	1	0			
5	0	0		/2	
AC 81	1	1			0 0
GC13	1	0			
C2XC2	0	0			
3	1	0		/2	
AC 84	0	0			2 0 0
GC13	0	0			
C2XC2	0	1			
6	1	0		/2	

TABLE D P. 7

HOLONOMY	FROBENIUS MAP A	B	C	DENOM	H1(M,Z)
AC 88	0	0			2 2 2 0
GC14	0	0			
C2XC2	1	0			
20	0	1		/2	
AC 88	1	0			2 4 0
GC14	0	0			
C2XC2	1	0			
21	0	1		/2	
AC 88	1	0			2 2 2 0
GC14	0	0			
C2XC2	0	0			
25	1	1		/2	
AC 88	1	0			2 4 0
GC14	0	0			
C2XC2	1	0			
29	1	1		/2	
AC 96	0	1			2 2 0
GC14	0	0			
C2XC2	0	0			
7	1	1		/2	
AC100	1	0			2 4 4
GC15	0	0			
C2XC2	1	0			
13	0	1		/2	
AC112	0	1	0		2 4 0
GC16	1	0	0		
C2XC2XC2	1	0	0		
278	0	0	1	/2	

TABLE D P. 8

HOLONOMY	FROBENIUS MAP A	B	C	DENOM	H1(M,Z)
AC112	0	1	0		2 2 2 0
GC16	1	0	0		
C2XC2XC2	0	0	0		
282	1	0	1	/2	
AC112	0	1	0		2 4 0
GC16	1	0	0		
C2XC2XC2	1	0	0		
286	1	0	1	/2	
AC112	1	1	0		2 2 0
GC16	0	1	0		
C2XC2XC2	1	0	0		
311	0	0	1	/2	
AC112	0	1	0		2 2 2 0
GC16	1	1	0		
C2XC2XC2	0	0	0		
312	1	0	1	/2	
AC112	0	1	0		2 4 0
GC16	1	1	0		
C2XC2XC2	1	0	0		
316	1	0	1	/2	
AC112	1	1	0		2 2 0
GC16	1	1	0		
C2XC2XC2	1	0	0		
317	1	0	1	/2	
AC112	1	1	0		2 2 0
GC16	0	1	0		
C2XC2XC2	1	0	0		
319	1	0	1	/2	

TABLE D P. 9

HOLONOMY	FROBENIUS MAP A	B	C	DENOM	H1(M,Z)
AC112	0	1	0		2 2 0
GC16	1	1	0		
C2XC2XC2	0	1	0		
376	1	0	1	/2	
AC112	1	1	0		2 2 0
GC16	0	1	0		
C2XC2XC2	0	1	0		
379	1	0	1	/2	
AC130	1	1	0		2 2 4
GC17	0	1	0		
C2XC2XC2	1	0	0		
183	0	0	1	/2	
AC130	1	1	0		2 2 4
GC17	1	1	0		
C2XC2XC2	1	0	0		
189	1	0	1	/2	
AC148	0				2 2 0
GC18	0				
C4	0				
1	1			/4	
AC150	0				2 2 0
GC18	0				
C4	0				
1	1			/2	
AC151	0				4 0
GC18	0				
C4	0				
1	1			/2	

TABLE D P.10

HOLONOMY	FROBENIUS MAP A	B	C	DENOM	H1(M,Z)
AC152	0				4 0
GC18	0				
C4	0				
1	1			/4	
AC156	0	0			2 2 0
GC19	2	0			
C4XC2	0	0			
13	1	2		/4	
AC156	0	0			2 2 0
GC19	2	0			
C4XC2	2	0			
15	1	2		/4	
AC160	0	0			2 0
GC19	0	0			
C4XC2	2	0			
7	1	2		/4	
AC170	0	0			4 0
GC22	1	0			
D4	0	0			
5	1	1		/2	
AC171	0	0			2 0
GC22	0	0			
D4	1	0			
5	1	1		/2	
AC176	2	0			4 4
GC23	0	0			
D4	1	0			
7	0	2		/4	

TABLE D P.11

HOLONOMY	FROBENIUS MAP				$H_1(M,Z)$
	A	B	C	DENOM	
AC209	0				6 0
GC28	0				
C6	0				
1	1			/6	
AC215	0				2 0
GC29	0				
C6	0				
1	1			/6	
AC217	0				2 0
GC29	0				
C6	0				
1	1			/6	
AC219	0	0			2 0
GC30	0	0			
C6XC2	3	0			
11	2	3		/6	
AC257	3	3	0		0
GC40	3	3	0		
TXC2	0	0	0		
7	2	4	3	/6	

Table E

This table is comprised of listings of the source language of the actual versions of the programs used for the main result.

```
         IDENT      DIAR
         ENTRY      DIAK,DIAKL,DIALT
GOA      LDA,I      0,1                     NRD
         SWA        F51C
         SWA        F50D
         SWA        FD120B
         SWA        FD150B
         SWA        FI54
         LDA        1,1                     L(MA)
         SWA        LMA1
         SWA        FI50C
         SWA        FI50G
         LDA        2,1                     L(KK)
         SWA        F200B
         LDA,I      3,1                     NRC
         INA,S      -1
         SWA        F70
         SWA        FD86G
         SWA        F109
         SWA        FI30
         SWA        FI49
         SWA        FI50F
         SWA        F50C
         SWA        DLICOP.2
         SWA        DLICS.2
         SWA        DLNRC
         MUA        DIALN
         ANA        -0
         ADA        LML
         SWA        DLROP.1
         SWA        DLRS.1
         LDA,I      4,1                     NCC
         INA,S      -1
         LDQ        F50C
         ANQ        -0
         AQJ,GE     *+2
         SWA        F50C
         SWA        DKNCC
         SWA        DKCOP.4
         SWA        DKC01.4
         SWA        DKCS.4
         MUA        F50D
         ANA        -0
         ADA        LMA1                    L(MA(1,NCC))
         SWA        F70A
         SWA        F120
         SWA        F150
         SWA        FI56
         INI        5,1
         STI        RTRN,1
SW8      BSS        1
SW9      BSS        1
F50A     ENA        0
         STA        KKM
```

```
         SWA     F154A
         UJP     F51
* START NEXT BLOCK
F50      LDA     KKM
F50C     ENQ     **              **=NM-1
         AQJ,GE  F200
         INA     1
F50B     STA     KKM
         SWA     F154A
         LDA     LMA1
F50D     INA     **              **=NRD
         SWA     LMA1
         SWA     F150C
         SWA     F150G
* FIND PIVOT ELEMENT
F51      ENA     FD70
         SWA     F51A
         SWA     F70+1
         ENQ     0
LMA1     ENI     **,2            **=L(MA(1,KK))
         LDI     KKM,1
         UJP     F51A
F51C     INI     **,2            **=NRD
         INI     1,1
F51A     STI     **,2            **=FD70 OR FD70A
F51B     LDI     KKM,3
         UJP     F70+1
FD70     LDA     **,3            **=L(MA(1,J))
         AZJ,EQ  F70
         STI     IP,3
         STI     JP,2
         STI     JP1,1
         ASG,S   1
         XOA,S   -0
         ASG     2
         UJP     FGI+2
         SHAQ    -24
         ENA     FD70A
         SWA     F70+1
         SWA     F51A
         STI     FD70A,2
         UJP     F70
FD70A    LDA     **,3
         AZJ,EQ  F70
         ASG,S   1
         XOA,S   -0
         AQJ,GE  F70
         STI     IP,3
         STI     JP,2
         STI     JP1,1
         ASG     2
         UJP     FGI+2
         SHAQ    -24
F70      ISI     **,3            **=NRC-1
         UJP     **              **=FD70 OR FD70A
```

```
 F70A     ISG       **,2              **=L(MA(1,NCC))
          UJP       F51C
 FGI      SHAQ      24
          AZJ,EQ    F200A
          STA       ABSIV
          LDI       JP,1
 FGI5     STI       FD120A,1          (1)=L(MA(1,JP)
          STI       FD86E,1
          STI       FGI7,1
 FGI6     LDI       IP,1              (1)=IP-1
          STI       F85,1
 FGI7     INI       **,1
          STI       F86,1
          ASG       2
          UJP       FGJ
          ENI       F85B,1
          ENI       FD109A+1,2
          ENI       F1129A,3
          UJP       FGK
 FGJ      LDA,I     F86
          AZJ,GE    FGJ1
          ENI       F86D,1
          UJP       *+2
 FGJ1     ENI       F86C,1
          ENI       F120,2
          ENI       F150B,3
 FGK      STI       F85A,1
          STI       FD109A,2
          STI       F130A,3
 SW1      BSS       1
* IS ROW  0
 FD120    LDI       LMA1,2
          LDI       KKM,1
          UJP       FD120A
 FD120B   INI       **,2              **=NRD
          INI       1,1
 FD120A   ISE       **,2              **=L(MA(1,JP))
          UJP       *+2
          UJP       F120
 F85      LDA       **,2              **=IP-1
          AZJ,EQ    F120
 F85A     UJP       **                **=F85B OR F86C OR F86D
 F85B     SHAQ      -24
 F86      DVA       **                **=L(MA(IP,JP))
 F86C     XOA,S     -0
F86D      STA       KWO
          STI       FD86F,2
          LDI       KKM,3
 FD86E    LDA       **,3              **=L(MA(1,JP)
          MUA       KWO
 FD86F    RAD       **,3              **=L(MA(1,J)
 FD86G    ISI       **,3              **=NRC-1
          UJP       FD86E
 SW2      BSS       1
 FD109A   UJP       **                **=*+1 OR F120
```

```
         LDQ       ABSIV
         STI       FD109,2
         LDI       KKM,3
FD109    LDA       **,3                    **=L(MA(1,J))
         AZJ,EQ    F109
         ASG,S     1
         XOA,S     -0
         AQJ,GE    F109
         STI       IP,3
         SHAQ      -24
F109     ISI       **,3                    **=NRC-1
         UJP       FD109
         LDA       ABSIV
         AQJ,EQ    F120
         STI       JP,2
         STI       JP1,1
         UJP       FGI
F120     ISG       **,2                    **=L(MA(1,NCC))
         UJP       FD120B
* IS COLUMN 0
FD130    LDA       F86
         SWA       F121A
         LDA       JP
         SWA       F121
         LDA       IP
         SWA       **2
FD130A   LDI       KKM,3
 F121B   ISE       **,3
         UJP       F121
         UJP       F130
F121     LDA       **,3
         AZJ,EQ    F130
         SHAQ      -24
F121A    DVA       **                      **=L(MA(IP,JP))
         STQ,I     F121
         STA       KWO
SW3      BSS       1
         LDA,I     F121
         AZJ,EQ    F130
         ASG,S     1
         XOA,S     -0
         STA       ABSIV
         STI       IP,3
         UJP       FGI6
F130     ISI       **,3                    **=NRC-1
         UJP       F121B
F130A    UJP       **                      **=F1129A OR F150B
F1129A   LDA       F121A
         SWA       F1131A
         LDA       JP
         SWA       FD150A
         SWA       F1358
         LDA       IP
         SWA       FD149A
         SWA       F135A
```

```
* DIVIDES REMAINING ELEMENTS
 FD150    LDI          LMA1,2
          LDI          KKM,1
          UJP          FD150A
 FD150B   INI          **,2                    **=NRD
          INI          _,1
 FD150A   ISE          **,2                    **=L(MA(1,JP))
          UJP          *+2
          UJP          F150
          STI          F1131,2
 FD149    LDI          KKM,3
 FD149A   ISE          **,3                    **=IP-1
          UJP          *+2
          UJP          F149
 F1131    LDA          **,3                    **=L(MA(1,J))
          AZJ,EQ       F149
          SHAQ         -24
 F1131A   DVA          **                      **=L(MA(IP,JP))
          SHAQ         24
          AZJ,EQ       F149
          STA,I        F1131
          ASG,S
          XOA,S        -0
          STA          ABSIV
          STQ          KWO
          LDA,I        F1131A
 F135A    STA          **,2                    **=IP-1
          MUA          KWO
          XOA,S        -0
 F135B    STA          **,3                    **=L(MA(1,JP))
          STI          JP,2
          STI          JP1,1
 SW4      BSS          1
          STI          IP,3
 SW5      BSS          1
          LDA          ABSIV
          UJP          FGI+3
 F149     ISI          **,3                    **=NRC-1
          UJP          FD149A
 F150     ISG          **,2                    **=L(MA(1,NCC))
          UJP          FD150B
* MOVE ROWS AND COLUMNS
 F150B    ENQ          0
          LDA          JP
          SWA          F150E
          LDI          KKM,3
 F150C    LDA          **,3                    **=L(MA(1,KK))
 F150D    STQ,I        *-1
 F150E    ST-          **,3                    **=L(MA(1,JP))
 F150F    ISI          **,3                    **=NRC-1
          UJP          F150C
  F150G   ENI          **,2
          LDI          IP,1
          STI          F155,1
          UJP          *+2
```

```
F154     INI      **,2            **=NRD
F154A    LDA      **,2            **=KK-1
         STQ,I    *-1
F155     STA      **,2            **IP-1
F156     ISG      **,2            **=L(MA(1,NCC))
         UJP      F154
         LDA      ABSIV
         LDI      KKM,2
         STA,I    LMAI
SW6      BSS      1
SW7      BSS      1
         UJP      F50
F200     INA      1
         UJP      *+2
F200A    LDA      KKM
F200B    STA      **
RTRN     UJP      **
 KWO     BSS      1
  ABSIV  BSS      1
  IP     OCT      0
  JP     OCT      0
 KKM     OCT      0
DIALI    UJP      **
         LDI      *-1,1
         LDA      DNQP
         STA      SW2
         STA      SW5
         STA      SW6
         STA      SW8
         LDA      DLISW1
         STA      SW1
         LDA      DLISW3
         STA      SW3
         STA      SW4
         LDA      DLISW7
         STA      SW7
         LDA      DLSW9
         STA      SW9
         LDA,I    0,1             NRDL
         SWA      DIALN
         LDA      1,1
         SWA      LML
         INI      2,1
         UJP      GOA
DIAKL    UJP      **
         LDI      *-1,1
         LDA      DLSW3
         STA      SW3
         STA      SW4
         LDA      DLSW7
         STA      SW7
         LDA      DLSW9
         STA      SW9
         LDA,I    0,1             NRDL
         SWA      DIALN
```

```
        SWA     DLROP.2
        SWA     DLRS.2
        LDA     1,1
        SWA     LML
        INI     2,1
        UJP     DIAK.1
DIAK    UJP     **
        LDI     *-1,1
        LDA     DNOP
        STA     SW3
        STA     SW4
        STA     SW7
        STA     SW9
DIAK.1  LDA     DKSW1
        STA     SW1
        LDA     DKSW2
        STA     SW2
        LDA     DKSW5
        STA     SW5
        LDA     DKSW6
        STA     SW6
        LDA     DKSW8
        STA     SW8
        LDA,I   0,1                    NRDK
        SWA     DIAKIN
        LDA     1,1                     L(MK)
        SWA     LMK
        INI     2,1
        UJP     GOA
DIAKIN  ENA     **                     **=NRDK
        SWA     IDM6
DKNCC   ENA     **                     **=NCC-1
        SWA     IDM9
        SWA     IDM15
LMK     ENI     **,1                    **=L(MK)
        RTJ     IDM
        UJP     SW9
DIALN   ENA     **                     **=NRDL
        SWA     IDM6
DLNRC   ENA     **                     **=NRC-1
        SWA     IDM9
        SWA     IDM15
LML     ENI     **,1                   **=L(ML)
        RTJ     IDM
        UJP     F50A
IDM     UJP     **
        ENI     0,3
        ENI     0,2
        ENA     0
        ENQ     1
        UJP     *+2
IDM6    INI     **,1
        STI     *+1,1
        STA     **,3
IDM9    ISI     **,3
```

```
         UJP      *-2
         STI      *+1,1
         STQ      **,2
IDM15    ISI      **,2
         UJP      IDM6
         UJP,I    IDM
JPSET    LDA      JPI
         MUA      DIAKIN
         ANA      -0
         ADA      LMK
         SWA      DKCOP.1              L(MK(1,JP))
         SWA      DKCO1.1
         SWA      DKCS.1
         UJP      SW1+1
IPSET    LDA      IP
         MUA      DIALN
         ANA      -0
         ADA      LML                  L(ML(1,IP))
         SWA      DLICOP.1
         SWA      DLICS.1
         UJP      SW1+1
DKCOP    TIA      1
         MUA      DIAKIN
         ANA      -0
         ADA      LMK
         SWA      *+4
         ENI      0,3
DKCOP.1  LDA      **,3                 **=L(MK(1,JP))
         MUA      KWO
         RAD      **,3
DKCOP.4  ISI      **,3                 **=NCC-1
         UJP      *-4
         UJP      SW2+1
DKCO1    TIA      1
         MUA      DIAKIN
         ANA      -0
         ADA      LMK
         SWA      *+3
         ENI      0,3
DKCO1.1  LDA      **,3                 **=L(MK(1,JP))
         RAD      **,3
DKCO1.4  ISI      **,3                 **=NCC-1
         UJP      *-3
         UJP      SW5+1
DKCS     LDA      KKM
         MUA      DIAKIN
         ANA      -0
         ADA      LMK
         SWA      *+2
         ENI      0,3
         LDA      **,3
DKCS.1   LDQ      **,3                 **=L(MK(1,JP))
         STQ,I    *-2
         STA,I    *-2
DKCS.4   ISI      **,3                 **=NCC=1
```

```
            UJP      *-5
            UJP      SW7
DLROP       UJP      **
            LDA      IP
            SWA      *+5
            STI      *+6,3
            LDI      LML,3
            UJP      *+2
DLROP.2     INI      **,3                    **=NRDL
            LCA      **,3                    **=IP-1
            MUA      KWO
            RAD      **,3                    **=I-1
DLROP.1     ISG      **,3                    **=L(ML(1,NRC))
            UJP      *-5
            LDI      *-3,3
            UJP,I    DLROP
DLRS        LDA      IP
            SWA      *+7
            LDA      KKM
            SWA      *+4
            LDI      LML,3
            UJP      *+2
DLRS.2      INI      **,3                    **=NRDL
            LDA      **,3
            LDQ      **,3
            STQ,I    *-2
            STA,I    *-2
DLRS.1      ISG      **,3                    **=L(ML(1,NRC))
            UJP      *-6
            UJP      SW7+1
DLICOP      UJP      **
            STI      DLICOP.3,3
            TIA      3
            MUA      DIALN
            ANA      -0
            ADA      LML
            SWA      *+2
            ENI      0,3
            LDA      **,3
            MUA      KWO
DLICOP.1    RAD      **,3                    **=L(ML(1,IP))
DLICOP.2    ISI      **,3                    **=NRC-1
            UJP      *-4
DLICOP.3    ENI      **,3
            UJP,I    DLICOP
DLICS       LDA      KKM
            MUA      DIALN
            ANA      -0
            ADA      LML
            SWA      *+2
            ENI      0,3
            LDA      **,3
DLICS.1     LDQ      **,3                    **=L(ML(1,IP))
            STQ,I    *-2
            STA,I    *-2
```

```
DLICS.2  ISI     **,3          **=NRC-1
         UJP     *-5
         UJP     SW7+1
DNOP     IJI     0
DKSW1    UJP     JPSET
DKSW2    UJP     DKCOP
DKSW5    UJP     DKCO1
DKSW6    UJP     DKCS
DKSW8    UJP     DIAKIN
DLSW3    RTJ     DLROP
DLSW7    UJP     DLRS
DLSW9    UJP     DIALN
DLISW1   UJP     IPSET
DLISW3   RTJ     DLICOP
DLISW7   UJP     DLICS
 JP1     OCT     0
         END
```

```
        IDENT     MMPYR
        ENTRY     MMPY,MMPY1,MMPY2,MMPZ,MMPA
MMPY1   UJP       **                      CALL MMPY1(IA,IB,IC,I1,J1,K1,
        LDI       *-1,1                   OR
        LDA,I     0,1                      DITTO ..K1,0,A,B,C)
        SWA       IA
        LDA,I     1,1
        SWA       IB
        LDA,I     2,1
        SWA       IC
MM2     LDA,I     3,1
        INA,S     -1
        SWA       I1
        LDA,I     4,1
        INA,S     -1
        SWA       J1
        LDA,I     5,1
        SWA       K1
        LDA,I     6,1
        AZJ,EQ    *+2
        UJP       7,1
        INI       7,1
        UJP       MM3
MMPY2   UJP       **                      CALL MMPY2(I1,J1,K1,1)
        LDI       *-1,1                   OR
        INI       -3,1
        UJP       MM2                     CALL MMPY2(I1,J1,K1,0,A,B,C)
MMPY    UJP       **
        LDI       *-1,1                   CALL MMPY(A,B,C)
MM3     LDA       0,1
        SWA       A
MM4     LDA       1,1
        SWA       JB
        LDA       2,1
        S...      JC
        INI       3,1
        STI       RTRN,1
        ENA       1
        STA       J
        UJP       A.
        INA       1
        STA       J
     IB ENA       **
        RAD       JB
     IC ENA       **
        RAD       JC
     A. ENI       0,1
      A ENA       **
        SWA       A1
        UJP       AAA
     AA ENA       1
        RAD       A1
    AAA ENA       0
        STA       X1
        ENI       0,3
```

```
      ENI    0,2
      UJP    A1
IA    INI    **,2
A1    LDA    **,2
JB    MUA    **,3
      RAD    X1
J1    ISI    **,3
      UJP    IA
      LDA    X1
JC    STA    **,1
I1    ISI    **,1
      UJP    AA
      LDA    J
K1    ASG    **
      UJP    IB-2
RTRN  UJP    **
J     BSS    1
X1    BSS    1
MMPA  UJP    **                         CALL MMPA(A)
      LDI    *-1,1
      LDA    0,1
      SWA    A
      UJP    1,1
MMPZ  UJP    **                         CALL MMPZ(B,C)
      LDI    *-1,1
      INI    -1,1
      UJP    MM4
      END
```

```
        IDENT     IGCDR
        ENTRY     IGCD
IGCD    UJP       **
        LDI       *-1,1
        LDA,I     0,1
        AZJ,EQ    Z
        ASG,S     0
        XOA,S     -0
        STA       M
        ENA       0
        LDQ,I     1,1
        QSG,S     0
        XOQ,S     -0
     D  DVA       M
        QSE,S     0
        UJP       L
        LDA       M
        UJP       2,1
     L  LDA       M
        STQ       M
        SHAQ      -24
        UJP       D
     Z  L[illegible],I   1,1
        ASG,S     0
        XOA,S     -0
        UJP       2,1
     M  BSS       1
        END
```

```
         IDENT    POLYR
         ENTRY    CLSTK,FREPOL PML1,CKPHD1
         ENTRY    PDCOD
         ENTRY     HOMOG,UN  3
         COMMON
STACK    BSS      1003
         PRG
D24      DEC
ADTMS    LCA      ,1
ADTMA    LDA      2,1
MCTP     UJP      MID+1
MNCTP    IJI      0
PT1      OCT      0
PT2      OCT      0
TEMP     BSS      10
PRTRN    UJP      **
CLSTK    UJP      **
         ENA      STACK+1
         STA      STACK
CLSTK1   TAI      1
         INA      3
         STA      0,1
         ISG      STACK+999,1           END OF STACK
         UJP      CLSTK1
         ENA      0
         STA      0,1
         UJP,I    CLSTK
NXCEL    UJP      **
         LDQ,I    STACK
         LDA      STACK
         ASE      0
         UJP      *+2
         UJP      STKOV
         ANQ      77777B
         STQ      STACK
         UJP,I    NXCEL
FRECEL   UJP      **
         SWA      *+2
         LDA      STACK
         SWA      **
         LDA      *-1
         SWA      STACK
         UJP,I    FRECEL
FREPOL   UJP      **
         LDA,I    *-1
         SWA      *+1
         LDA      **
         ANA      77777B
         AZJ,EQ   FREPOL1
         LDA      STACK,2
         SWA      STACK
         TAI      1
         LDA      0,1
         ASE      0
         UJP      *-3
```

```
         TIA      2
         SWA      0,1
FREPOL1  LDI      FREPOL,1
         UJP      1,1
CRPHD1   UJP      **                    CALL CRPHD1(N,MV,MC,P)
         LDI      *-1,1
         LDA      3,1
         SWA      P
         LDA      2,1
         SWA      MC
         LDA      1,1
         SWA      MV
         LDA,I    0,1
         INA,S    -1
         SWA      N
         INI      4,1
         STI      PRTRN,1
         ENI      0,3
         ENI      0,1
CRP1     RTJ      NXCEL
         TAI      2
MC       LDA      **,1
         STA      2,2
MV       LDA      **,1
         STA      1,2
         STI      *+1,2
         STI      **,3
         TIA      2
         TAI      3
N        ISI      **,1
         UJP      CRP1
P        STI      **,2
         UJP,I    PRTRN
AD12     UJP      **                    (1)=LOC(1 TERM P)
         ENA      0
         STA      TEMP
AD12A    ISE      0,1
         UJP      *+2
         UJP      ADRTRN
         LDQ      1,1
AD12S    ENI      **,3                  **=LOC(EXTL LINK VAR TO Q)
AD12D    LDA      0,3
         TAI      2
         ISE      0,2
         UJP      *+2
         UJP      ADNM
         LDA      1,2
         AQJ,EQ   ADTM
         STI      *+1,2
         ENI      **,3
         UJP      AD12D
ADTM     BSS      1                     LDA OR LCA 2,1
         ADA      2,2
         AZJ,EQ   ADTZ
         STA      2,2
```

```
        UJP     ADNPT
ADTZ    LDA     0,2
        SWA     0,3
        TIA     2
        RTJ     FRECEL
        UJP     ADNPT
ADNM    RTJ     NXCEL
        TAI     2
        LDA     TEMP
        SWA     0,2
        LDA     1,1
        STA     1,2
ADNM1   BSS     1                       LDA OR LCA 2,1
        STA     2,2
        STI     TEMP,2
ADNPT   LDA     0,1
        TAI     1
        UJP     AD12A
ADRTRN  LDQ     TEMP
        QSE     0
        UJP     *+2
        UJP     AD12
        LDI     AD12B,2
        LDA     0,2
        ANA     77777B
        AZJ,EQ  *+3
        TAI     2
        UJP     *-4
        SHAQ    24
        SWA     0,2
        UJP,I   AD12
PML1    UJP     **                      CALL PML1(P,Q,R,ISW)
        LDI     *-1,1
        LDA,I   3,1
        AZJ,GE  *+3
        LDA     DIMS
        UJP     *+2
        LDA     ADTMA
        STA     ADTM
        STA     ADNM1
        LDA     2,1
        SWA     AD12B
        LDA,I   1,1
        SWA     M1
        LDA,I   0,1
        SWA     M.8
        INI     4,1
        STI     PRTRN,1
M.8     ENA     **                      **=LOC(1 TERM P)
M.9     ASE     0
        UJP     *+2
        UJP,I   PRTRN
        STA     PT2                     (PT2)=LOC(CURRENT TERM P)
        INA     1
        SWA     M1AP
```

```
          INA      1
          SWA      M1H
          ENA      0
          STA      PT1
          LDA,I    M1AP
          AZJ,NE   M.99
          LDA      MCTP
          STA      M1APCT
          UJP      M1
 M.99     LDA      MNCTP
          STA      M1APCT
M1        ENI      **,2                **=LOC(I TERM Q)
M1.1      ISE      0,2
          UJP      *+2                 (2)=LOC(CURRENT TERM Q)
          UJP      M2
          LDA      1,2
          STA      TEMP
          ENI      0,3                 SHIFT COUNTER
M1APCT    BSS      1
M1A       ENA      0
          LDQ      TEMP
          QSE,S    0
          UJP      *+2
          UJP      M1F
          DVA      D24
          STA      TEMP
M1AP      LDA      **                  **=LOC(VF TERM P)
          AQJ,GE   M1B
          INI      1,3
          STQ      TEMP,3
          UJP      M1A
M1B       STQ      TEMP+9
          LDA      TEMP
          MUA      D24
          ADA      TEMP+9
          MUA      D24
          ADA,I    M1AP
M1E       ISE      0,3
          UJP      M1C
          UJP      M1D
M1C       MUA      D24
          ADA      TEMP,3
          ISD      1,3
          UJP      M1C
M1D       STA      TEMP
          RTJ      NXCEL
          SWA      *+1
          ENI      **,3
          LDA      PTI
          SWA      0,3
          STI      PT1,3
          LDA      TEMP
          STA      .,3
M1H       LDA      **                  P COEF FIELD
          MUA      2,2
```

```
         STA      2,3
         LDA      0,2
         TAI      2
         UJP      M1,1
MIF      LDA,I    MIAP
         UJP      MIE
M2       LDI      PT1,1
         RTJ      AD12
         RTJ      FREPOL
         77       PT1
         LDA,I    PT2
         UJP      M,9
PDCOD    UJP      **           CALL PDCOD(MP,MV,MC,NT,NTM)
         LDI      *-1,1
         LDA,I    4,1
         SWA      PDNTM
         LDA      3,1
         SWA      PDNT1
         LDA      2,1
         SWA      PDMC
         LDA      1,1
         SWA      PDMV
         ENI      0,2
         LDA,I    0,1
PLP      TAI      3
         ISE      0,3
         UJP      *+2
         UJP      PDNT
         LDA      1,3
PDMV     STA      **,2
         LDA      2,3
PDMC     STA      **,2
         LDA      0,3
         INI      1,2
PDNTM    ISG      **,2
         UJP      PLP
PDNT     TIA      2
PDNT1    STA      **
         UJP      5,1
HOMOG    UJP      **                 CALL HOMOG(MP)
         LDI      *-1,1
         LDA      0,1
         SWA      *+2
         SWA      *+S
         LDI      **,2
         ISG      1,2
         UJP      1,1
         LDA      1,2
         ASE,S    0
         UJP      1,1
         LDA      0,2
         SWA      **
         TIA      2
         RTJ      FRECEL
         UJP      1,1
```

```
UNHMG    UJP     **                    CALL UNHMG(MP,MC)
         LDI     *-1,1
         RTJ     NXCEL
         TAI     2
         LDA,I   1,1
         STA     2,2
         ENA     0
         STA     1,2
         LDA     0,1
         SWA     *+1
UNHMGP   LDA     **
         SWA     0,2
         STI,I   UNHMGP,2
         UJP     2,1
STKOV    RTJ     ABNORMAL
         UJP     **
         EXT     ABNORMAL
         END
```

```
      FUNCTION IDET4(M)
      DIMENSION M(4,4),M1(3,3)
      DO 10 I=1,3
      DO 10 J=1,3
   10 M1(I,J)=M(I+1,J+1)
      ISW=1
      IDET4=0
      DO 100 I=1,4
      IF(M(I).EQ.0)GO TO 15
      IDET=IDET3(M1)
      CALL PML1(M(I),IDET,IDET4,ISW)
      CALL FREPOL(IDET)
   15 IF(ISW.GT.0)GO TO 20
      ISW=1
      GO TO 25
   20 ISW=-1
   25 IF(I.EQ.4)RETURN
      DO 100 J=1,3
  100 M1(I,J)=M(I,J+1)
      END
      FUNCTION IDET3(M)
      DIMENSION M(3,3),M1(2,2)
      DO 10 I=1,2
      DO 10 J=1,2
   10 M1(I,J)=M(I+1,J+1)
      ISW=1
      IDET3=0
      DO 100 I=1,3
      IF(M(I).EQ.0)GO TO 15
      IDET=IDET2(M1)
      CALL PML1(M(I),IDET,IDET3,ISW)
      CALL FREPOL(IDET)
   15 IF(ISW.GT.0)GO TO 20
      ISW=1
      GO TO 25
   20 ISW=-1
   25 IF(I.EQ.3)RETURN
      DO 100 J=1,2
  100 M1(I,J)=M(I,J+1)
      END
      FUNCTION IDET2(M)
      DIMENSION M(2,2)
      IDET2=0
      CALL PML1(M(1,1),M(2,2),IDET2,1)
      CALL PML1(M(1,2),M(2,1),IDET2,-1)
      RETURN
      END
```

```
      SUBROUTINE NITTY(NA,NB,MU,MA,MB,MDEL)
      DIMENSION MA(5,5,3),MB(5,5,3),MDEL(75,25)
      NANB=NA*NB
      NANBMU=NANB*MU
      DO 170 I1=1,NANBMU
      DO 170 J1=1,NANB
  170 MDEL(I1,J1)=0
      DO 200 INU=1,MU
      I0=(INU-1)*NANB
      DO 200 I1=1,NB
      J2=(I1-1)*NA
      I2=J2+I0
      DO 200 I3=1,NA
      I4=I3+I2
      DO 180 J3=1,NA
      J4=J2+J3
  180 MDEL(I4,J4)=MA(I3,J3,INU)
      DO 200 J1=1,NB
      J4=(J1-1)*NA+I3
  200 MDEL(I4,J4)=MDEL(I4,J4)-MB(J1,I1,INU)
      RETURN
      END
```

```
      SUBROUTINE MATP(ND,NP,MA,MB)
      DIMENSION MA(5,5),MB(5,5),M1(5,5)
      JSW=1
      GO TO 10
      ENTRY MATT
      JSW=2
   10 NP1=NP
      IF(NP1)15,15,20
   15 DO 18 I=1,ND
      DO 16 J=1,ND
   16 MB(I,J)=0
   18 MB(I,I)=1
      RETURN
   20 DO 25 I=1,ND
      DO 23 J=1,ND
   23 MB(I,J)=MA(I,J)
      GO TO(25,24),JSW
   24 MB(I,I)=MB(I,I)+1
   25 CONTINUE
      NP1=NP1-1
      IF(NP1)80,80,30
   30 CALL MMPA(MA)
   40 CALL MMPZ(MB,M1)
      ISW=1
      GO TO(60,42),JSW
   42 DO 43 I=1,ND
   43 M1(I,I)=M1(I,I)+1
      GO TO 60
   50 CALL MMPZ(M1,MB)
      ISW=2
      GO TO(60,52),JSW
   52 DO 53 I=1,ND
   53 MB(I,I)=MB(I,I)+1
   60 NP1=NP1-1
      IF(NP1)70,70,65
   65 GO TO(50,40),ISW
   70 GO TO(75,80),ISW
   75 DO 76 I=1,ND
      DO 76 J=1,ND
   76 MB(I,J)=M1(I,J)
   80 RETURN
      END
```

```
      SUBROUTINE WORDC(ND,LI,LE,MG,MGA)
      DIMENSION MG(5,5,3),MGA(5,5),MI(5,5),M2(5,5),LI(4),LE(4)
      L1=LI(1)
      CALL MATP(ND,LE(1),MG(1,1,L1),MGA)
      ISW=1
      DO 10 I=2,4
      L1=LI(I)
      IF(L1.EQ.0)GO TO 20
      CALL MATP(ND,LE(I),MG(1,1,L1),M1)
      GO TO(5,6),ISW
    5 CALL MMPY(MGA,M1,M2)
      ISW=2
      GO TO 10
    6 CALL MMPY(M2,M1,MGA)
      ISW=1
   10 CONTINUE
   20 IF(ISW.EQ.1)RETURN
      CALL MATP(ND,1,M2,MGA)
      RETURN
      END
```

```
      SUBROUTINE WORDV(LI,LE,MX,M1)
      COMMON MA(5,5,5),ND,NDMU
      DIMENSION LI(4),LE(4),MX(5,25),M1(5,5)
      EQUIVALENCE (MA,MCOM(2956)),(ND,MCOM(3911)),(NDMU,MCOM(3912))
      DIMENSION M2(5,5)
      MWL=4
      DO 505 I=1,ND
      DO 505 J=1,NDMU
  505 MX(I,J)=0
      K1=1
      K2=LI(K1)
      K3=LE(K1)
      J1=ND*(K2-1)
      DO 510 J2=1,ND
      J3=J1+J2
      DO 510 I2=1,ND
      IF(I2-J2)509,508,509
  508 MX(I2,J3)=1
      GO TO 510
  509 MX(I2,J3)=0
  510 M1(I2,J2)=MA(I2,J2,K2)
      ISW=1
      GO TO 540
  520 GO TO(521,526),ISW
  521 DO 525 J2=1,ND
      J3=J1+J2
      DO 525 I2=1,ND
  525 MX(I2,J3)=MX(I2,J3)+M1(I2,J2)
      CALL MMPY(M1,MA(1,1,K2),M2)
      ISW=2
      GO TO 540
  526 DO 530 J2=1,ND
      J3=J1+J2
      DO 530 I2=1,ND
  530 MX(I2,J3)=MX(I2,J3)+M2(I2,J2)
      CALL MMPY(M2,MA(1,1,K2),M1)
      ISW=1
  540 K3=K3-1
      IF(K3)545,545,520
  545 IF(K1-MWL)550,600,600
  550 K1=K1+1
      K2=LI(K1)
      IF(K2)600,600,555
  555 J1=ND*(K2-1)
      K3=LE(K1)
      GO TO 520
  600 GO TO(610,605),ISW
  605 DO 606 J=1,ND
      DO 606 I=1,ND
  606 M1(I,J)=M2(I,J)
  610 RETURN
      END
         FINIS
```

```
      PROGRAM V8
      COMMON MA(5,5,3),MB(2,2,3),MDEL(18,6),MLI(18,18),ISOC(2),MCC(6),
     C MCX(6),LAI(4,3,8),LAE(4,3,8)
    1 FORMAT(7I3,2A4)
   20FORMAT(6H1GEOCL,I3,5X,5HISOCL,1X,2A4,5X,4HNDEC,I3/5H GENS,I3,5X,
     1 2HNA,I3,5X,2HNB,I3,5X,4HNAUT,I3)
    3 FORMAT(5H0IDEC,I3,5X,4HCHRK,I3,5X,5HINVAR,6I3)
    4 FORMAT(4H0IAC,I3)
    5 FORMAT(1X/6H*GENER,I3)
    6 FORMAT(12X,8I3)
    7 FORMAT(26I3)
    8 FORMAT(1X/4H*AUT,I3)
      CALL DIAGLI1(18,18,MDEL,KK,MLI)
      MWL=4
      CALL MMPY1(18,6,6,0,0,0,1)
      CALL MMPA(MLI)
      NZERO=0
      NEG1=-1
      N888=888
      READ 7,IAC
C GET GEOM CLASS
   20 READ 1,IFL,NGC,NDEC,MU,NA,NB,NAUT,ISOC
      CALL SSWTCH(1,IP)
      GO TO(21,22),IP
   21 PUNCH1,IFL,NGC [illegible]DEC,MU,NA,NB,NAUT,ISOC
   22 IF(IFL.EQ.999)[illegible] TO 25
      WRITE TAPE 11,(NZERO,I=1,10)
      REWIND 11
      STOP
   25 NANB=NA*NB
      NANBMU=NANB*MU
      NA1=NA+1
      ND=NA+NB
      DO 26 K=1,NAUT
      DO 26 J=1,MU
   26 READ 7,(LAI(I,J,K),LAE(I,J,K),I=1,MWL)
      WRITE TAPE 11,IFL,NGC,ISOC,NDEC,MU,NA,NB,ND,NAUT,
     1 (((LAI(I,J,K),LAE(I,J,K),I=1,MWL),J=1,MU),K=1,NAUT)
      PRINT 2,NGC,ISOC,NDEC,MU,NA,NB,NAUT
      DO 28 K=1,NAUT
      PRINT 8,K
      DO 28 J=1,MU
   28 PRINT 6,(LAI(I,J,K),LAE(I,J,K),I=1,MWL)
      DO 30 K=1,MU
      DO 30 I=NA1,ND
      DO 30 J=1,NA
   30 MA(I,J,K)=0
      DO 300 IDEC=1,NDEC
      DO 60 K=1,MU
      READ 7,((MA(I,J,K),J=1,NA),I=1,NA)
   60 READ 7,[illegible](I,J,K),J=1,NB),I=1,NB)
      CALL [illegible](NA,NB,MU,MA,MB,MDEL)
      CALL DIAGLI(NANBMU,NANB)
      PRINT3,IDEC,KK,(MDEL(I,I),I=1,KK)
      DO 70 K=1,MU
```

```
      DO 70 I=1,NB
      I1=I+NA
      DO 70 J=1,NB
      J1=J+NA
   70 MA(I1,J1,K)=MB(I,J,K)
      CALL MMPY2(NANBMU,KK,1,1)
      DO 105 I=1,KK
  105 MCC(I)=0
  110 CALL MMPZ(MCC,MCX)
      I0=0
      DO 120 K=1,MU
      DO 120 J=NA1,ND
      DO 120 I=1,NA
      I0=I0+1
  120 MA(I,J,K)=MCX(I0)
      IAC=IAC+1
      PRINT 4,IAC
      GO TO(121,122),IP
  121 PUNCH 7,N888,IAC,ND
  122 DO 125 K=1,MU
      GO TO(123,124),IP
  123 PUNCH 7,((MA(I,J,K),J=1,ND),I=1,ND)
  124 PRINT 5,K
      DO 125 I=1,ND
  125 PRINT 6,(MA(I,J,K),J=1,ND)
      WRITE TAPE 11,IDEC,IAC,(((MA(I,J,K),I=1,ND),J=1,ND),K=1,MU)
  210 KINC=1
  215 MC1=MCC(KINC)+1
      IF(MC1.LT.MDEL(KINC,KINC))GO TO 225
      KINC=KINC+1
      IF(KINC.LE.KK)GO TO 215
      GO TO 300
  225 KIN=KINC-1
      DO 235 I=1,KIN
  235 MCC(I)=0
      MCC(KINC)=MC1
      GO TO 110
  300 CONTINUE
      WRITE TAPE 11,NEG1,NEG1,(((NEG1,I=1,ND),J=1,ND),K=1,MU)
      GO TO 20
      END
```

```
      PROGRAM VC
      COMMON MX(75,25),MK(25,25),ISOC(2),MG(5,5,3),MH(5,5,3),MV(100),
     1MC(100),M4(4,4),MP(5),MR,KT,MGA(5,5,24),LAI(4,3,8),LAE(4,3,8)
    1 FORMAT(6HOGEOCL,I3,5X,5HISOCL,1X,2A4,5X,4HGENS,I3,5X,3HDIM,I3)
    2 FORMAT( 9HOPOSS EQU,5I4/10X,4HCOEF,5X,3HVAR)
    3 FORMAT(10X,I6,5X,5I4)
    4 FORMAT(6HOOVFLO,4I4)
      CALL DIAGKI(75,25,MX,KK,MK)
      CALL MMPY1(5,5,5,0,0,0,1)
      MWL=4
   20 READ TAPE 11 ,IFL,NGC,ISOC,NDEC,MU,NA,NB,ND,NAUT,
     1 (((LAI(I,J,K),LAE(I,J,K),I=1,MWL),J=1,MU),K=1,NAUT)
      IF(IFL.EQ.999)GO TO 25
      REWIND 11
      STOP
   25 PRINT 1,NGC,ISOC,MU,ND
      ND2=ND**2
      MXR=MU*ND2
   30 READ TAPE 11,IDEC,IAC,(((MG(I,J,K),I=1,ND),J=1,ND),K=1,MU)
      IF(IAC.LE.0)GO TO 20
      CALL MMPY2(ND,ND,ND,1)
      DO 230 K=1,NAUT
      K1=(K-1)*MU
      DO 230 J=1,MU
      J1=K1+J
  230 CALL WORDC(ND,LAI(1,J,K),LAE(1,J,K),MG,MGA(1,1,J1))
      KBS=0
   35 KBS=KBS+1
      READ TAPE 11,JDEC,JAC,(((MH(I,J,K),I=1,ND),J=1,ND),K=1,MU)
      IF(JAC.GT.0)GO TO 100
      IF(KBS.EQ.1)GO TO 20
      DO 40 I=1,KBS
   40 BACKSPACE 11
      GO TO 30
  100 DO 400 KA=1,NAUT
      KA1=(KA-1)*MU+1
      CALL RITTY(ND,ND,MU,MGA(1,1,KA1),MH,MX)
      CALL DIAGK(MXR,ND2)
      IF(KK.EQ.ND2)GO TO 400
      MC1=KK+1
      CALL CLSTK
      DO 310 J=1,ND
      J0=(J-1)*ND
      DO 310 I=1,ND
      I0=J0+I
      K0=0
      K1=0
      DO 300 K=MC1,ND2
      K0=K0+1
      IF(MK(I0,K).EQ.0)GO TO 300
      K1=K1+1
      MV(K1)=K0
      MC(K1)=MK(I0,K)
  300 CONTINUE
      IF(K1.EL.0)GO TO 305
```

```
      CALL CRPHD1(KI,MV,MC,M4(I,J))
      GO TO 310
  305 M4(I,J)=0
  310 CONTINUE
      MR=IDET4(M4)
      CALL PDCOD(MR,MV,MC,KT,100)
      IF(KT.GE.99)GO TO 380
      I0=0
      DO 350 I=1,KT
  350 I0=IGCD(I0,MC(I))
      IF(I0.NE.1)GO TO 400
      PRINT 2,IDEC,IAC,JDEC,JAC,KA
      DO 360 I=1,KT
      I0=0
      J0=MV(I)
  355 IF(J0.EQ.0)GO TO 360
      K0=J0/24
      L0=J0-24*K0
      J0=K0
      I0=I0+1
      MP(I0)=L0
      GO TO 355
  360 PRINT 3,MC(I),(MP(J),J=1,I0)
      GO TO 35
  380 PRINT 4,IDEC,IAC,JDEC,JAC
      GO TO 35
  400 CONTINUE
      GO TO 35
      END
```

```
      PROGRAM ACCHK
     0DIMENSION LRI(4,8),LRE(4,8),LOTW(7),LTI(4,7),LTE(4,7),MR1(5,25)
     1,M1(5,5),MR(40,25),M2(5,5),MT(5,5,7),MRI(35,25),ISOC(2)
      COMMON MA(5,5,5),ND,NDMU
    1 FORMAT   I3)
    2 FORMAT(5I3,2A4)
    3 FORMAT(5H1SKIP,I4,1X,5HACREC)
    4 FORMAT(5H0NEXT,I3,35H ARITH CRYST CLASSES ISOMORPHIC TO ,2A4
     1/1X,I3,11H GENERATORS,10X,I3,9H RELATORS,10X,I3,13H TOR 1ST ELEM)
    6 FORMAT(4H REL,I3,10X,10I3)
    7 FORMAT(8H ITW ORD,I3,6X,10I3)
    8 FORMAT(18H0ARITH CRYST CLASS,I4,10X,3HDIM,I3,10X,9HISO CLASS,2A4)
    9 FORMAT(1H /4H*GEN,I3)
   10 FORMAT(15X,5I3)
   11 FORMAT(1H /9H*TORT ORD,I3)
   12 FORMAT(14H0TAPE SEQUENCE,I4)
   13 FORMAT(13H ***FOUL DATA, 2I3)
      NLRAC=0
      CALL MMPY1(5,5,5,0,0,0,1)
      MWL=4
C SKIP PREVIOUS RECORDS
      READ 1,NTAC
      PRINT 3,NTAC
      IF(NTAC)30,30,20
   20 DO 25 I=1,NTAC
   25 READ TAPE 1,IFL
C GET ISO CLASS
   30 READ 2,IFL,KAC,MU,NU,NTW,ISOC
      IF(IFL-999)600,31,600
   31 PRINT 4, KAC, ISOC,MU,NU,NTW
      DO 35 I=1,NU
      READ  1,(LRI(J,I),LRE(J,I),J=1,MWL )
   35 PRINT 6,I,(LRI(J,I),LRE(J,I),J=1,MWL )
      DO 40 I=1,NTW
      READ  1,LOTW(I),(LTI(J,I),LTE(J,I),J=1,MWL )
   40 PRINT 7,LOTW(I),(LTI(J,I),LTE(J,I),J=1,MWL )
   50 DO 500 IAC=1,KAC
      NTAC=NTAC+1
      READ  1,IFL,NAC,ND
      IF(IFL-888)560,60,560
   60 CALL MMPY2(ND,ND,ND,1)
      PRINT 8,NAC,ND,ISOC
      DO 65K=1,MU
      PRINT 9,K
      READ  1,((MA(I,J,K),J=1,ND),I=1,ND)
      DO 65 I=1,ND
   65 PRINT 10,(MA(I,J,K),J=1,ND)
      NDMU=ND*MU
      NDNU=ND*NU
      NDNTW=ND*NTW
C CHECK RELATORS
      DO 125 INU=1,NU
      CALL WORDV(LRI(1,INU),LRE(1,INU),MR1,M1)
      DO 120 I=1,ND
      DO 120 J=1,ND
```

```
      IF(I-J)105,110,105
  105 IF(M1(I,J))115,120,115
  110 IF(M1(I,J)-1)115,120,115
  115 IFU=INU
      IGU=1
      GO TO 570
  120 CONTINUE
      I1=ND*(INU-1)
      DO 125 I=1,ND
      I2=I1+I
      DO 125 J=1,NDMU
  125 MR(I2,J)=MR1(I,J)
C COMPUTE TOR TST ELEM AND CHECK ORDER
      DO 280 ITW=1,NTW
      I1=ND*(ITW-1)
      CALL WORDV(LTI(1,ITW),LTE(1,ITW),MR1,MT(1,1,ITW))
      CALL MMPY(MT(1,1,ITW),MT(1,1,ITW),M1)
      ISW=1
      KI=LOTW(ITW)-1
      GO TO 220
  209 GO TO(210,215),ISW
  210 CALL MMPZ(M1,M2)
      ISW=2
      GO TO 220
  215 CALL MMPZ(M2,M1)
      ISW=1
  220 KI=KI-1
  221 IF(KI)225,225,209
  225 GO TO (230,240),ISW
  230 DO 235 I=1,ND
      DO 235 J=1,ND
      IF(I-J)231,232,231
  231 IF(M1(I,J))250,235,250
  232 IF(M1(I,J)-1)250,235,250
  235 CONTINUE
      GO TO 260
  240 DO 245 I=1,ND
      DO 245 J=1,ND
      IF(I-J)241,242,241
  241 IF(M2(I,J))250,245,250
  242 IF(M2(I,J)-1)250,245,250
  245 CONTINUE
      GO TO 260
  250 IFU=ITW
      IGU=2
      GO TO 570
  260 PRINT 11,LOTW(ITW)
      DO 270 I=1,ND
  270 PRINT 10,(MT(I,J,ITW),J=1,ND)
      DO 280 I=1,ND
      I2=I1+I
      DO 280 J=1,NDMU
  280 MT(I2,J)=MR1(I,J)
C OUTPUT
     0WRITE TAPE 1,NLRAC,ND,NAC,ISOC,MU,NU,NTW,NDMU,NDNU,NDNTW,((LRI(I,J
```

```
     1),LRE(I,J),I=1,MWL),J=1,NU),(LOTW(I),(LTI(J,I),LTE(J,I),J=1,MWL),
     2I=1,NTW),(((MA(I,J,K),I=1,ND),J=1,ND),K=1,MU),(((MT(I,J,K),I=1,ND)
     3,J=1,ND),K=1,NTW),((MR(I,J),I=1,NDNU),J=1,NDMU),((MRT(I,J),I=1,
     4NDNTW),J=1,NDMU)
  500 PRINT 12,NTAC
      GO TO 30
  560 IFU=0
      IGU=0
  570 PRINT 13,IFU,IGU
  600 REWIND 1
      STOP
      END
```

```
      PROGRAM HIL818
     0COMMON    MR(40,25),MR1(40,25),MK(25,25),MCV(25),ME(25),MCC(25),
     1MFBS(25),MRFS(25),MFBST(35),MRT(35,25),LRI(4,8),LRE(4,8),LOTW(7),
     2LTI(4,7),LTE(4,7),MA(5,5,5),MT(5,5,7),MCW(25),ISOC(2)
    1 FORMAT(26I3)
   20FORMAT(3H0AC,I4,5X,3HDIM,I4,5X,4HISOC,2A4/
     13H H2,4X,3HORD,I5,4X,4HRANK,I5)
    3 FORMAT(6H INVAR,25I4)
    4 FORMAT(1H /9H*K-MATRIX)
    5 FORMAT(10X,25I4)
      MWL=4
      ICHF=-1
      CALL DIAGK1(40,25,MR1,KK,MK)
C SKIP RECORDS
      READ 1,I1,I2,NPTGP
  101 DO 105 I=1,I1
  105 READ TAPE 1
  115 DO 120 I=1,I2
      READ TAPE 2,NLRAC
  116 DO 119 J=1,NLRAC
  119 READ TAPE 2
  120 CONTINUE
  130 IPTGP=1
  131 IF(IPTGP-NPTGP)200,200,140
  140 REWIND 1
      REWIND 2
      STOP
  2000READ TAPE 1 ,NLRAC,ND,NAC,ISOC,MU,NU,NTW,NDMU,NDNU,NDNTW,((LRI(I,
     1J),LRE(I,J),I=1,MWL),J=1,NU),(LOTW(I),(LTI(J,I),LTE(J,I),J=1,MWL),
     2I=1,NTW),(((MA(I,J,K),I=1,ND),J=1,ND),K=1,MU),(((MT(I,J,K),I=1,ND
     3),J=1,ND),K=1,NTW),((MR(I,J),I=1,NDNU),J=1,NDMU),((MRT(I,J),I=1,
     4NDNTW),J=1,NDMU)
      DO 210 J=1,NDMU
      DO 210 I=1,NDNU
  210 MR1(I,J)=MR(I,J)
      CALL DIAGK(NDNU,NDMU)
C GET H2
      NOH2=1
      KRK=0
  220 MERO=MR1(KK,KK)
      DO 240 IK=1,KK
      K1=MR1(IK,IK)
      NOH2=NOH2*K1
      MCV(IK)=MERO/K1
      ME(IK)=K1
      IF(K1.EQ.1)GO TO 240
      KRK=KRK+1
  240 CONTINUE
      PRINT 2,NAC,ND,ISOC,NOH2,KRK
  245 PRINT 3,(ME(I),I=1,KK)
      PRINT 4
      DO 246 I=1,NDMU
  246 PRINT 5,(MK(I,J),J=1,NDMU)
  260 NLRAC=NOH2-1
     0WRITE TAPE 2,NLRAC,ND,NAC,ISOC,MU,NU,NTW,NDMU,NDNU,NDNTW,((LRI(I,
```

```
      1),LRE(I,J),I=1,MWL),J=1,NU),(LOTW(I),(LTI(J,I),LTE(J,I),J=1,MWL),
      2I=1,NTW),(((MA(I,J,K),I=1,ND),J=1,ND),K=1,MU),(((MT(I,J,K),I=1,ND)
      3,J=1,ND),K=1,NTW),KK ,NOH2,MERO,(ME(I),I=1,KK ),((MK(I,J),I=1,NDMU
      4),J=1,NDMU)
       IF(NLRAC) 400,400,300
300   DO 305 I=1,KK
305   MCC(I)=0
      KEQ=1
310   KINC=1
315   MC1=MCC(KINC)+MCV(KINC)
      IF(MC1-MERO)325,320,320
320   KINC=KINC+1
      IF(KINC-KK )315,315,400
325   KIN=KINC-1
330   DO 335 I=1,KIN
335   MCC(I)=0
340   MCC(KINC)=MC1
      CALL MMPY1(25,25,25,NDMU,KK ,1,0,MK,MCC,MFBS)
      CALL MMPY1(35,25,25,NDNTW,NDMU,1,0,MRT,MFBS,MFBST)
      CALL MMPY1(40,25,25,NDNU,NDMU,1,0,MR,MFBS,MRFS)
      DO 360 I=1,NDNU
360   MRFS(I)=MRFS(I)/MERO
      DO 365 I=1,KK
365   MCW(I)=MCC(I)/MCV(I)
     0WRITE TAPE 2,(ICHF,I=1,5),(MCW(I),I=1,KK ),(MFBS(I),I=1,NDMU),
     1(MRFS(I),I=1,NDNU),(MFBST(I),I=1,NDNTW)
      GO TO 310
400   IPTGP=IPTGP+1
      GO TO 131
      END
```

```
      PROGRAM FINDFLAT
     0COMMON ISOC(2),LOTW(7),MT(5,5,7),LTC(7),MCW(25),MFBST(5,7),M1(5,5)
     1,M2(5,5),M3(5,5),KL(7),MTT(5,5,7),MTTL(5,5,7),MTD(5,7),
     2 MRHS(5),MLHS(5),LHS(5)
    1 FORMAT(26I3)
    2 FORMAT(12H0ARITH CLASS,I4,5X,3HDIM,I2,5X,4HISOC,2A4/
     110H TEST ELEM,I3,10X,7HH2 RANK,I3,5X,5HORDER,I4)
    3 FORMAT(10X,11HCOHOM CLASS,25I3)
    4 FORMAT(10X,14HCRITICAL ELEMT,I3,10X,5HPOWER,I3)
    5 FORMAT(5X,35H************TORSION FREE******ILRAC=,I4)
    6 FORMAT(10X,28H CRITICAL FOR ALL EXTENSIONS)
      CALL MMPY1(5,5,5,0,0,0,1)
      CALL DIAGL1(5,5,M2,KK,M3)
      MWL=4
C SKIP RECORDS
      READ 1,NSK,NDO
      DO 20 I=1,NSK
      READ TAPE 2,NLRAC
      DO 20 J=1,NLRAC
   20 READ TAPE 2
C NEXT ARITH CLASS
  100 IF(NDO)1000,1000,105
  105 READ TAPE 2,NLRAC,ND,NAC,ISOC,MU,NU,NTW,NDMU,NDNU,NDNTW,((KBG,KBG,
     1I=1,MWL),J=1,NU),(LOTW(J),(KBG,KBG,I=1,MWL),J=1,NTW),(((KBG,I=1,ND
     2),J=1,ND),K=1,MU),(((MT(I,J,K),I=1,ND),J=1,ND),K=1,NTW),KRK,NOH2,
     3MERO
      PRINT 2,NAC,ND,ISOC,NTW,KRK,NOH2
      IF(NLRAC)106,106,110
  106 NDO=NDO-1
      GO TO 100
  110 CALL DIAGL2(ND,ND)
      DO 115 I=1,NTW
  115 LTC(I)=0
      DO 500 ICC=1,NLRAC
     0READ TAPE 2,(KBG,I=1,5),(MCW(I),I=1,KRK),(KBG,I=1,NDMU),(KBG,I=1,
     1NDNU),((MFBST(I,J),I=1,ND),J=1,NTW)
      PRINT 3,(MCW(I),I=1,KRK)
      DO 400 ITW=1,NTW
      IF(LTC(ITW))120,120,160
  120 LOR=1
      LO=LOTW(ITW)
      CALL MMPY2(ND,ND,ND,1)
      CALL MATT(ND,LO-1,MT(1,1,ITW),MTT(1,1,ITW))
      CALL MATP(ND,1,MTT(1,1,ITW),M2)
      CALL DIAGL
      IF(KK)390,390,130
  130 KL(ITW)=KK
      CALL MMPY(M3,MTT(1,1,ITW),MTTL(1,1,ITW))
      DO 135 I=1,KK
  135 MTD(I,ITW)=MRHS(I)=M2(I,I)
      LTC(ITW)=1
      GO TO 166
  160 LO=LOTW(ITW)
      KK=KL(ITW)
      DO 165 I=1,KK
```

```
165 MRHS(I)=MTD(I,ITW)
    LOR=1
166 CALL MMPY2(ND,ND,1,0,MTTL(1,1,ITW),MFBST(1,ITW),MLHS)
    DO 170 I=1,KK
170 MLHS(I)=MLHS(I)/MERO
200 L=1
    DO 215 I=1,KK
    K2=MLHS(I)
    IF(K2)205,215,205
205 K2=MRHS(I)/IGCD(MRHS(I),K2)
    K3=LO-(LO/K2)*K2
    IF(K3)400,210,400
210 L=(L*K2)/IGCD(L,K2)
215 CONTINUE
    IF(L-LO)220,400,400
220 IF(L-1)300,380,225
225 IF(LOR-1)230,230,240
230 CALL MATP(ND,1,MT(1,1,ITW),M1)
    CALL MMPY2(ND,ND,1,0,MT(1,1,ITW),MFBST(1,ITW),LHS)
    DO 232 I=1,ND
232 LHS(I)=LHS(I)/MERO
240 LOR=LOR*L
    LO=LO/L
    CALL MMPY2(ND,ND,ND,1)
    CALL MATP(ND,L,M1,M2)
    CALL MATP(ND,1,M2,M1)
    CALL MATT(ND,LO-1,M1,M2)
    CALL DIAGL
    IF(KK)390,390,250
250 DO 255 I=1,KK
255 MRHS(I)=M2(I,I)
    CALL MMPY2(ND,ND,1,0,M3,LHS,MLHS)
    GO TO 200
380 PRINT 4,ITW,LOR
    GO TO 500
390 PRINT 4,ITW,LOR
    PRINT 6
    K2=ICC+1
    DO 395 I=K2,NLRAC
395 READ TAPE 2
    GO TO 106
400 CONTINUE
    PRINT 5 ICC
500 CONTINUE
    GO TO 106
1000 REWIND 2
    STOP
    END
```

```
      PROGRAM ABELR
     0COMMON LRI(4,8),LRE(4,8),MA(5,5,5),MRS(10,33),MR(10,33),ME(25),
     1MCC(25),MRFS(40),ISOC(2)
      INTEGER GBG
    1 FORMAT(26I3)
    2 FORMAT(1H0,10X,3HACC,I4,10X,3HDIM,I4,/10X,2HH2,5X,5HORDER,I4,5X,
     14HRANK,I4/10X,10HINVARIANTS,2X,25I3)
    3 FORMAT(12H0COHOM CLASS,69X,22HEXTENSION MADE ABELIAN)
    4 FORMAT(1H*,25I3)
    5 FORMAT(81X,10I3)
      MWL=4
      CALL DIAG1(10,MR,KK)
C SKIP RECORDS
      READ 1,NSK,NDO
      DO 10 I=1,NSK
      READ TAPE 2,NLRAC
      DO 10 J=1,NLRAC
   10 READ TAPE 2
   15 IF(NDO)16,16,20
   16 REWIND 2
      STOP
   20 READ TAPE 2,NLRAC,ND,NAC,ISOC,MU,NU,NTW,NDMU,NDNU,NDNTW,((LRI(I,J
     1),LRE(I,J),I=1,MWL),J=1,NU), ( GBG,(GBG,GBG,I=1,MWL),J=1,NTW),
     2(((MA(I,J,K),I=1,ND),J=1,ND),K=1,MU),(((GBG,I=1,ND),J=1,ND),K=1,
     3NTW),KRK,NOH2,GBG,(ME(I),I=1,KRK)
      PRINT 2,NAC,ND,NOH2,KRK,(ME(I),I=1,KRK)
      PRINT 3
C SET UP CONSTANT PART OFMR
      NR1=ND+1
      NR2=ND+MU
      NC1=NU+1
      NC2=NU+ND*MU
      J1=NU-ND
      DO 130 J=1,MU
      J1=J1+ND
      DO 130 J2=1,ND
      J3=J1+J2
      DO 129 I=1,ND
  129 MRS(I,J3)=MA(I,J2,J)
  130 MRS(J2,J3)=MRS(J2,J3)-1
      DO 140 I=NR1,NR2
      DO 140 J=1,NC2
  140 MRS(I,J)=0
      DO 165 J=1,NU
      DO 160 J1=1,MWL
      J2=LRI(J1,J)
      IF(J2)165,165,150
  150 J3=ND+J2
  160 MRS(J3,J)=MRS(J3,J)+LRE(J1,J)
  165 CONTINUE
      KEQ=1
C SPLIT EXTENSION
      DO 210 I=1,KRK
  210 MCC(I)=0
      DO 215 J=1,NU
```

```
      DO 215 I=1,ND
  215 MR(I,J)=0
      GO TO 230
C NON TRIVIAL EXTENSIONS
  2250READ TAPE 2,(ICHF,I=1,5),(MCC(I),I=1,KRK),(G8G,I=1,NDMU),
     1(MRFS(I),I=1,NDNU)
      I1=0
      DO 245 J=1,NU
      DO 245 I=1,ND
      I1=I1+1
  245 MR(I,J)=-MRFS(I1)
  230 DO 235 J=NC1,NC2
      DO 235 I=1,NR2
  235 MR(I,J)= MRS(I,J)
      DO 240 I=NR1,NR2
      DO 240 J=1,NU
  240 MR(I,J)= MRS(I,J)
  260 CALL DIAG(NR2,NC2)
      DO 265 I=1,NR2
      IF(MR(I,I)-1)264,265,264
  264 I1=I
      GO TO 266
  265 CONTINUE
  266 PRINT 4,(MCC(I),I=1,KRK)
      PRINT 5,(MR(I,I),I=I1,NR2)
      IF(KEQ-NOH2)270,275,275
  270 KEQ=KEQ+1
      GO TO 225
  275 NDO=NDO-1
      GO TO 15
      END
```

```
      PROGRAM ISG
      0COMMON MCOM(2700),MLB(15,15),IB(10,3)      ,MA(5,5,3),MR1(15,15),
     1 MK(25,25),MFBS(15),NFBS(15),ND,NDMU,MPE(25)
     2,MEH(15),MKH(15,15)
      0DIMENSION MX(75,25),MP(15,45),MKB(25,45),MLC(25,25),MLG(25),
     1 MKB1(45,45),MB(5,5,3),LAI(4),LAE(4)
       DIMENSION MLD(25,25)
      0EQUIVALENCE (MCOM,MX,MP,MLC,LAI),(MCOM(5),LAE),(MCOM(1876),MB),
     1 (MCOM(676),MKB,MKB1), (MCOM(1801),MLD),(MCOM(626),MLG)
    1 FORMAT(25I3)
    2 FORMAT(4H0ACC,4I4)
    3 FORMAT(5X,5HSIMCL,10I4)
    4 FORMAT(1X/1H*,10X,4HAUT ,I4)
    5 FORMAT(25X,8I4)
    6 FORMAT(1X,2I4,1X,12HNO PART SOLN )
    7 FORMAT(6H ERRRR,I5)
      MAXQ=10
      MWL=4
      NZO=0
C GET HOLONOMY
   25 READ 1,NAC,NAUT,NEQ,KORD
      PRINT2,NAC,NAUT,NEQ,KORD
      IF(NAC.GE.1)GO TO 30
   26 REWIND 10
      WRITE TAPE 11,(NZO,I=1,100)
      REWIND 11
      STOP
   30 READ TAPE 10,NLRAC,ND,KAC,I0,I0,MU,I0,II,NDMU,K0,K0,((K0,K0,I=1,
     1 MWL),J=1,I0),(I0,(I0,I0,I=1,MWL),J=1,II),(((MA(I,J,K),I=1,ND),J=1
     2,ND),K=1,MU),(((I0,I=1,ND),J=1,ND),K=1,II),KH,I0,MERO
     3,(MEH(I),I=1,KH),((MKH(I,J),I=1,NDMU),J=1,NDMU)
      IF(KAC-NAC)31,40,35
   31 DO 32 I=1,NLRAC
   32 READ TAPE 10
      GO TO 30
   35 NER =1
      GO TO 950
C GET SIM CLASSES
   40 DO 45 I=1,NEQ
      READ 1, (IB(J,I),J=1,MAXQ)
   45 PRINT 3,(IB(J,I),J=1,MAXQ)
      KORDERO=KORD*MERO
      ND2=ND**2
      ND2MU=ND2*MU
      I0=1
      DO 445 J=1,KH
      JI=MEH(J)
      MEH(J)=I0
        =I0*JI
      II=KORDERO/JI
      DO 445 I=1,NDMU
  445 MKH(I,J)=MKH(I,J)*II
      KH1=KH-1
      KAUT=0
C CENTRALIZER--IDENTITY AUTOMORPHISM
```

```
      CALL NITTY(ND,ND,MU,MA,MA,MX)
      GO TO 60
C NEXT AUTOMORPHISM
   50 KAUT=KAUT+1
      PRINT 4,KAUT
C SET UP MR1
      CALL MMPY1(5,0,5,ND,ND,ND,1)
      K1=1
      DO 55 K=1,MU
      READ 1,(LA1(I),LA2(I),I=1,MWL)
      PRINT 5,(LA1(I),LA2(I),I=1,MWL)
      CALL WORDV(LA1,LA2,MR1(K1),MB(1,1,K))
   55 K1=K1+ND
      CALL NITTY(ND,ND,MU,MA,MB,MX)
   60 CALL DIAK(25,MK,75,MX,KKS,ND2MU,ND2)
      NV1=KKS+1
      NV=ND2-KKS
C NEXT SIMCL
      KEQ=1
   99 IEQ=1
  100 JEQ=IEQ+1
      JLRAC=IB(JEQ,KEQ)
      IF(JLRAC.EQ.0)GO TO 355
      ILRAC=KLRAC-IB(IEQ,KEQ)
      IC=1
      GO TO 450
  120 IF(KAUT.GT.0)GO TO 130
      DO 125 I=1,NDMU
  125 NFBS(I)=MFBS(I)
      GO TO 550
  130 CALL MMPY2(NDMU,NDMU,1,0,MR1,MFBS,NFBS)
      GO TO 550
C SET UP PS
  140 K0=0
      K3=NV+ND
      K4=K3+1
      K5=K3+NDMU
      DO 180 K=1,MU
      DO 175 I=1,ND
      K1=K0+I
      DO 150 L=1,NV
      K2=KKS+L
      ML=0
      J1=I
      DO 145 J=1,ND
      J2=K0+J
      ML=ML+MK(J1,K2)*NFBS(J2)
  145 J1=J1+ND
  150 MP(K1,L)=ML
      DO 160 J=1,ND
      L=J+NV
  160 MP(K1,L)=MA(I,J,K)
      L=I+NV
      MP(K1,L)=MP(K1,L)-1
      DO 165 L=K4,K5
```

```
165 MP(K1,L)=0
    L=K3+K1
    MP(K1,L)=KORDERO
175 CONTINUE
180 K0=K0+ND
    CALL DIAKL(15,MLB,45,MKB1,15,MP,KKB,NDMU,K5)
    DO 185 I=1,KKB
185 MPE(I)=MP(I,I)
    KKB1=KKB+1
    DO 182 J=1,K5
    DO 182 I=1,NV
182 MKB(I,J)=MKB1(I,J)
    CALL DIAL1(25,MLC,25,MKB(1,KKB1),KKC,NV,K5-KKB)
    CALL MMPY1(25,25,25,NV,KKC,KKC,0,MLC,MKB(1,KKB1),MLC)
    CALL MMPY2(ND2,NV,KKC,0,MK(1,NV1),MLC,MLD)
    CALL MMPY2(ND2,NV,KKB,0,MK(1,NV1),MKB,MLC)
205 KLRAC=JLRAC
    I0=2
    GO TO 450
220 CALL MMPY2(NDMU,NDMU,1,0,MLB,MFBS,NFBS)
    GO TO 550
300 DO 310 I=1,KKB
    K1=NFBS(I)
    K2=K1/MPE(I)
    K1=K1-K2*MPE(I)
    IF(K1.NE.0)GO TO 340
310 NFBS(I)=K2
    DO 320 I=KKB1,NDMU
    IF(NFBS(I).NE.0)GO TO 340
320 CONTINUE
    CALL MMPY1(25,25,25,ND2,KKB,1,0,MLC,NFBS,MLG)
   0WRITE TAPE 11,NAC,KAUT,ILRAC,JLRAC,ND,KKC,((MLD(I,J),I=1,ND2),
   1J=1,KKC),(MLG(I),I=1,ND2)
    GO TO 350.
340 PRINT 6,ILRAC,JLRAC
350 IF(JEQ.GE.MAXQ)GO TO 351
    JEQ=JEQ+1
    JLRAC=IB(JEQ,KEQ)
    IF(JLRAC.EQ.0)GO TO 351
    GO TO 205
351 IEQ=IEQ+1
    IF(IEQ.GE.MAXQ)GO TO 355
    GO TO 100
355 KEQ=KEQ+1
    IF(KEQ.LE.NEQ)GO TO 99
    IF(KAUT.LT.NAUT)GO TO 50
    DO 360 I=1,NLRAC
360 READ TAPE 10
    GO TO 25
450 I1=KH1
    DO 455 I=1,KH
    II=I1-I
    JI=KEH(II)
    NFBS(II)=KI=KLRAC/JI
455 KLRAC=KLRAC-KI*JI
```

```
      CALL MMPY1(15,15,15,NDMU,KH,1,0,MKH,NFBS,MFBS)
      GO TO(120,220),IO
  550 DO 556 I=1,NDMU
      I1=NFBS(I)
  552 IF(I1)554,556,556
  554 I1=I1+KORDERO
      GO TO 552
  556 NFBS(I)=I1-KORDERO*(I1/KORDERO)
      GO TO(140,300),IO
  950 PRINT 7,NER
      GO TO 26
      END
```

```
      PROGRAM ISH
      COMMON MSTK(1005),MLD(5,5,25),MLG(5,5),MC(100),MV(100),
     1MX(4,4),MW(25)
      EQUIVALENCE (KAUT,NAUT)
    1 FORMAT(4HONAC,6I4)
    2 FORMAT(9H VAR OVFL,3I4)
    3 FORMAT(9H POSS ISO/I2X,5HCOEFS,5X,9HVARIABLES)
    4 FORMAT(5X,I10,5X,25I3)
    5 FORMAT(8H NON ISO,2I4)
      LAUT=0
      LILR=0
      LAC=0
   10 READ TAPE 11,NAC,KAUT,ILRAC,JLRAC,ND,KKC,(((MLD(I,J,K),I=1,ND),J=
     1,ND),K=1,KKC), ((MLG(I,J),I=1,ND),J=1,ND)
      IF(NAC.GT.0)GO TO 20
      REWIND 11
      STOP
   20 PRINT 1,NAC,NAUT,ILRAC,JLRAC,KKC
      IF(NAC.EQ.LAC.AND.ILRAC.EQ.LILR.AND.NAUT.EQ.LAUT)GO TO 60
      CALL CLSTK
      DO 50 I=1,ND
      DO 50 J=1,ND
      K1=0
      DO 40 K=1,KKC
      K2=MLD(I,J,K)
      IF(K2.EQ.0)GO TO 40
      K1=K1+1
      MV(K1)=K
      MC(K1)=K2
   40 CONTINUE
      IF(K1.LE.23)GO TO 45
      PRINT 2,K1,I,J
      GO TO 10
   45 K2=MLG(I,J)
      IF(K2.EQ.0)GO TO 47
      K1=K1+1
      MV(K1)=0
      MC(K1)=K2
   47 IF(K1.GT.0)GO TO 49
      MX(I,J)=0
      GO TO 50
   49 CALL CRPHO1(K1,MV,MC,MX(I,J))
   50 CONTINUE
      GO TO 100
   60 DO 80 I=1,ND
      DO 80 J=1,ND
      IF(MX(I,J).EQ.0)GO TO 75
      CALL HOMOG(MX(I,J))
   75 K2=MLG(I,J)
      IF(K2.EQ.0)GO TO 80
      CALL UNHMG(MX(I,J),K2)
   80 CONTINUE
  100 KP=IDET4(MX)
      CALL PBCOD(KP,MV,MC,KT,100)
      CALL FREPOL(KP)
```

```
      IF(KT.LT.99)GO TO 105
      PRINT 2,KT
      GO TO 200
  105 I0=0
      K4=0
      DO 110 I=1,KT
      IF(MV(I).EQ.0)GO TO 109
      I0=IGCD(I0,MC(I))
      GO TO 110
  109 K4=MC(I)
  110 CONTINUE
      IF(IGCD(I0,K4).NE.1)GO TO 140
      K2=K4+1
      K1=K4-1
      IF(I0.NE.0)GO TO 120
      IF( K1.EQ.0.OR. K2.EQ.0)GO TO 125
      GO TO 140
  120 IF(K1-(K1/I0)*I0.NE.0.AND.K2-(K2/I0)*I0.NE.0)GO TO 140
  125 PRINT 3
      DO 160 I=1,KT
      I0=0
      J0=MV(I)
  155 IF(J0.EQ.0)GO TO 160
      K0=J0/24
      L0=J0-24*K0
      J0=K0
      I0=I0+1
      MW(I0)=L0
      GO TO 155
  160 PRINT 4,MC(I),(MW(J),J=1,I0)
      GO TO 200
  140 PRINT 5,I0,K4
  200 LAUT=NAUT
      LILR=ILRAC
      LAC=NAC
      GO TO 10
      END
          FINIS
```