N73-25650

NASA TECHNICAL NOTE

NASA TN D-7327

NASA TN D-7327

CASE FILE COPY

AN ALGORITHM FOR GENERATING ALL POSSIBLE 2^{p-q} FRACTIONAL FACTORIAL DESIGNS AND ITS USE IN SCIENTIFIC EXPERIMENTATION

by Steven M. Sidik

Lewis Research Center

Cleveland, Ohio 44135

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JUNE 1973

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TN D-7327		

4. Title and Subtitle	5. Report Date
AN ALGORITHM FOR GENERATING ALL POSSIBLE 2^{p-q} FRACTIONAL FACTORIAL DESIGNS AND ITS USE IN SCIENTIFIC EXPERIMENTATION	June 1973
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Steven M. Sidik	E-7365
	10. Work Unit No.
9. Performing Organization Name and Address	503-35
Lewis Research Center National Aeronautics and Space Administration Cleveland, Ohio 44135	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Note
National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

An algorithm and computer program are presented for generating all the distinct 2^{p-q} fractional factorial designs. Some applications of this algorithm to the construction of tables of designs and of designs for nonstandard situations and its use in Bayesian design are discussed. An appendix includes a discussion of an actual experiment whose design was facilitated by the algorithm.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Factorial designs Fractional factorial designs Optimal design of experiments Analysis of variance	Unclassified - unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	33	\$3.00

* For sale by the National Technical Information Service, Springfield, Virginia 22151

AN ALGORITHM FOR GENERATING ALL POSSIBLE 2^{p-q} FRACTIONAL FACTORIAL DESIGNS AND ITS USE IN SCIENTIFIC EXPERIMENTATION

by Steven M. Sidik

Lewis Research Center

SUMMARY

An algorithm and computer program are presented for generating all the distinct 2^{p-q} fractional factorial designs. Some results using group theory describe the algebraic structure of the groups of defining contrasts for 2^{p-q} fractional factorials.

A number of applications of the algorithm are discussed. Among these are (1) the construction of new tables of designs, (2) extensions of existing tables, (3) construction of designs for experiments subject to special constraints, and (4) the Bayesian design of experiments.

An appendix includes a discussion of an actual experiment subjected to special constraints. Through examination of the output of the program, it was quite easy to construct a suitable experimental design.

INTRODUCTION

The two-level factorial and fractional factorial designs are a class of designs of experiments that yield much information about the factors studied. Fisher (ref. 1), Yates (ref. 2), and Finney (ref. 3) were the principal early investigators of these designs. They were almost exclusively concerned with agricultural experimentation which required large experiments that were performed all at one time. When the industrial applications of these designs became apparent (e.g., Box and Wilson (ref. 4) and Daniel (ref. 5)), much of the emphasis changed to the consideration of experiments that could be performed sequentially in small stages. For example, Webb (ref. 6) (who considers mainly nonorthogonal designs), Addelman (ref. 7), Holms (ref. 8), and Holms and Sidik (ref. 9) (who consider orthogonal designs) have proceeded along these lines. These recent developments have required more information about the possible designs.

In this report an algorithm for the construction of all possible 2^{p-q} fractional factorial designs for any specified values of p and q is presented. The algorithm uses an isomorphism between the group of defining contrasts that define the experiments and a group defined on the set of integers from 0 to $2^p - 1$. All subgroups of order 2^q may then be listed where the generators of the subgroups are in alphabetical order. A recursion relation that yields the number of subgroups for the values of p and q is also developed. Lindenberg and Gerhards (ref. 10) present a similar algorithm for use in much more general situations. They did not discover some of the results described herein, which hold only for the special groups defining 2^{p-q} experiments.

Applications of this algorithm include the generation of tables such as Addelman's (ref. 7), the construction of designs (if they exist) that meet special restrictions, and the use in the optimal design of experiments when certain prior information is available (refs. 11 and 12).

The reader is assumed to have a knowledge of two-level factorial designs such as might be found in Davies (ref. 13), Peng (ref. 14), or John (ref. 15). A more complete presentation of the fundamentals of 2^{p-q} fractional factorials than is given here can be found in reference 11.

FUNDAMENTALS OF 2^{p-q} FRACTIONAL FACTORIALS

In a full factorial experiment with p independent variables (factors) X_A, X_B, . . ., each restricted to assuming only two values, there are 2^p possible distinct combinations of values. It is common practice to say the independent variables can assume either a high level (+1) or a low level (-1). Each of the 2^p distinct combinations of levels is called a treatment combination. From such an experiment it is possible to estimate all the β's in an equation of the form

$$Y = \beta_I + \beta_A X_A + \beta_B X_B + \beta_{BA} X_B X_A + \beta_C X_C + \beta_{CA} X_C X_A + \beta_{CB} X_C X_B + \beta_{CBA} X_C X_B X_A + \ldots + \beta_{\ldots CBA} \cdots X_C X_B X_A + \delta \tag{1}$$

where δ is a random variable with mean zero and finite variance and is uncorrelated from one observation to the next. (Symbols are defined in appendix A.)

As p increases, 2^p increases so rapidly that the number of parameters in equation (1) and the number of treatment combinations required soon become unrealistically large. The most common method of reducing these numbers is to perform a fractional replicate of the experiment. A regular fractional replicate of the full factorial experiment does not allow separate estimation of all the β's. But certain linear combinations (alias sets) of them can be estimated. The usual method of coping with this problem is to simply assume that all the higher order interactions (say those involving three or more factors) are negligible or zero. An alternate method requiring more complete information is given in reference 11.

The particular set of linear combinations that can be estimated depends on the particular treatments composing the fractional replicate or, equivalently, on the choice of the design of the experiment. For example, the one-half replicate of a three-factor experiment defined by the treatment combinations $\{(1), ab, ac, bc\}$ corresponding to the defining contrast $I = -ABC$ would provide estimators for $(\beta_I - \beta_{CBA})$, $(\beta_A - \beta_{CB})$, $(\beta_B - \beta_{CA})$, and $(\beta_{BA} - \beta_C)$.

It will be convenient at this point to introduce an alternate notation for equation (1). Let the p independent variables be denoted as $X_1, X_2, \ldots$, etc. Number the 2^p β's of equation (1) from β_0 to β_{2^p-1} and consider the following equation, which is similar to equation (1):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_2 X_1 + \ldots + \beta_{2^p-1} X_p \ldots X_3 X_2 X_1 + \delta \tag{2}$$

Equations (1) and (2) are both written in what is called the standard order. If the subscripts of the β's are rewritten as p-digit binary numbers, it becomes quite obvious how the terms and coefficients of equation (2) are related. For example, let $p = 3$ and consider the following equation:

$$Y = \beta_0 + \beta_1 X_1 + \beta_{10} X_2 + \beta_{11} X_2 X_1 + \beta_{100} X_3 + \beta_{101} X_3 X_1 + \beta_{110} X_3 X_2 + \beta_{111} X_3 X_2 X_1 + \delta \tag{3}$$

In general, a β, whose subscript in binary notation has ones in the $i_1, i_2, \ldots, i_k$ locations from the right, is the coefficient of the interaction of $X_{i_1}, X_{i_2}, \ldots$, and X_{i_k}.

The set of all 2^p contrasts, which define the estimators from a full factorial, form a group C. These contrasts are denoted here by the combinations of capital letters such as ABC and BDEG. The group operation is $*$ and is defined as the commutative multiplication of the letters with the exponents reduced modulo two. The identity element of the group is I. The defining contrasts that define various fractional replicates are subgroups of C. The aliased sets of parameters that are estimable from these fractions are given by the elements of the subgroup and the elements of the cosets of the subgroup.

SOME PRELIMINARY NOTATION AND RESULTS

So far we have adopted the usual notation involving the use of capital Roman letters to denote the contrasts. In what follows and for purposes of computer programming, it will be more convenient to use a numerical notation. The defining contrast group with p factors is isomorphic to a group on the integers from 0 to $2^p - 1$. The integers are expressed in binary as $(i_p i_{p-1} \ldots i_1)$ where the i_k are zeros or ones. The isomorphism from the group in letter notation to the group of integers in binary representation is defined by the mapping

$$A^{i_1} B^{i_2} C^{i_3} \ldots \rightarrow (i_p i_{p-1} \ldots i_1)$$

The group product is still $*$, but it is defined similarly as

$$(i_p \ldots i_1) * (j_p \ldots j_1) = (k_p \ldots k_1)$$

where $k_n = (i_n + j_n) \pmod 2$.

For example, suppose $p = 4$. Then we have

$$CBA \rightarrow (0\ \ 1\ \ 1\ \ 1)$$

and

$$DCB \rightarrow (1\ \ 1\ \ 1\ \ 0)$$

Then,

$$CBA * DCB = DA$$

and

$$(0\ \ 1\ \ 1\ \ 1) * (1\ \ 1\ \ 1\ \ 0) = (1\ \ 0\ \ 0\ \ 1)$$

where $DA \rightarrow (1\ \ 0\ \ 0\ \ 1)$. The identity element of the group in binary representation is (0).

Let $S(p, q)$ be the set of all subgroups of order 2^q from the full group of order 2^p. Let $K(p, q)$ be the number of such subgroups. Let a group $\mathscr{G}$ be an element of $S(p, q)$. There are 2^q elements in $\mathscr{G}$, which we denote as $\{w_i\colon\ i = 0, \ldots, 2^q - 1\}$. A subset of q elements from $\mathscr{G}$, $\{\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_q\}$, are called <u>independent</u> if, for any set of

$i_1, i_2, \ldots, i_q$, where i_j is either zero or one and $\sum_{j=1}^{q} i_j > 0$,

$$(\widetilde{w}_1)^{i_1} * (\widetilde{w}_2)^{i_2} * \ldots * (\widetilde{w}_q)^{i_q} \neq (0) \tag{4}$$

where (0) is the identity element of the group. It can be shown that, for any q elements of $\mathcal{G}$ which are independent, any other element of the group can be generated by computing the appropriate product of the independent elements. Thus the q independent elements can be called generators of the group. The specification of a group by its generators is not unique, however. That is, if q generators, which yield a group, are given and a different set of generators is given, it does not follow that the resulting groups are different. In developing the algorithm for generating all the groups, it will be convenient to develop a method of uniquely relating a group to a set of generators.

As the first step in deriving a unique relation between a group and its generators, we will assume that for each $\mathcal{G} \in S(p, q)$, the elements are arranged in numerically increasing order so that

$$\mathcal{G} = \left\{w_0, w_1, \ldots, w_{2^q-1}\right\}$$

where

$$0 = w_0 < w_1 < w_2 < \ldots < w_{2^q-1}$$

For any $\mathcal{G}$ whose elements are so ordered, let us define the sets of integers P_n and Q_n as

$$P_n = \left\{w_0, \ldots, w_{2^n-1}\right\} \qquad n = 0, \ldots, q \tag{5}$$

and

$$Q_n = \left\{w_{2^{n-1}}, \ldots, w_{2^n-1}\right\} \qquad n = 1, \ldots, q \tag{6}$$

Let us also define the set of integers from 2^{n-1} through $2^n - 1$ by I_n; that is,

$$I_n = [2^{n-1}, \ldots, 2^n - 1] \tag{7}$$

There are 2^{n-1} elements in I_n. Thus graphically we have P_n and Q_n related as follows:

Elements
of $\mathcal{G}$

$$\left.\begin{array}{l} \left.\begin{array}{l} \left.\begin{array}{ll} w_0 & \} P_0 \\ w_1\} & Q_1 \end{array}\right\} P_1 \\ \left.\begin{array}{l} w_2 \\ w_3 \end{array}\right\} Q_2 \end{array}\right\} P_2 \\ \left.\begin{array}{l} w_4 \\ w_5 \\ w_6 \\ w_7 \end{array}\right\} Q_3 \end{array}\right\} P_3$$

$$\begin{array}{ll} \vdots & \qquad\qquad \vdots \end{array}$$

We may also represent the sets I_n graphically as follows:

Integers (base 10)	Integers (base 2)		Integers (base 10)	Integers (base 2)	
0	0		8	1000	
1	1	$\} I_1$	9	1001	
2	10	I_2	10	1010	
3	11		11	1011	I_4
4	100		12	1100	
5	101	I_3	13	1101	
6	110		14	1110	
7	111		15	1111	
			.	.	
			.	.	
			.	.	

An example of the relations between I_n, P_n, and Q_n is as follows: Let $p = 4$ and $q = 3$ and

$$\mathcal{G} = \{0,\ 1,\ 10,\ 11,\ 1000,\ 1001,\ 1010,\ 1011\}$$

$$= \{I,\ A,\ B,\ BA,\ D,\ DA,\ DB,\ DBA\}$$

Thus

$$\begin{array}{llllll}
w_0 = I = 0 & & \}P_0 & \}P_1 & & \\
w_1 = A = 1 & \}Q_1 & & & & \\
w_2 = B = 10 & \}Q_2 & & & \}P_2 & \\
w_3 = BA = 11 & & & & & \\
w_4 = D = 1000 & & & & & \}P_3 \\
w_5 = DA = 1001 & \}Q_3 & & & & \\
w_6 = DB = 1010 & & & & & \\
w_7 = DBA = 1011 & & & & &
\end{array}$$

We note that w_1, w_2, and w_4 may be used to generate the group $\mathcal{G}$. We also note that $Q_1 \subset I_1$, $Q_2 \subset I_2$, and $Q_3 \subset I_4$, and that

$$P_i = \{w_0\} \cup Q_1 \cup \ldots \cup Q_i$$

We formalize these relations in the following lemmas. The understanding of the lemmas and their proofs will aid in, but are not necessary to, the understanding of the algorithm. Thus the uninterested reader may turn directly to the description of the computer algorithm.

Lemma 1: For any set $\{\tilde{w}_i:\ i = 1, \ldots, j\}$ such that $\tilde{w}_i \in I_{n_i}$ where $1 \le n_1 < \ldots < n_j < p$, the $\tilde{w}_i$ are independent.

Proof: Consider that for any $w \in I_n$ we have the binary representation of w as

$$\begin{array}{ccccccccccc}
 & p & & & n+1 & n & n-1 & & & 1 & \\
w = (& 0 & . & . & 0 & 1 & X & . & . & X &)
\end{array}$$

That is, all digits to the left of the n^{th} position are zeros, the n^{th} digit is a 1, and the digits to the right of the n^{th} position are either zeros or ones. Consider any set of indices $\{i_k:\ k = 1, \ldots, j\}$ where the i_k are either zeros or ones but not all may be zero. Let m be the subscript of the last i_k in the sequence that is not zero. We note

$\tilde{w}_i^0 = (0)$, the identity element. It is clear that, since $\tilde{w}_m$ has a one as the n_mth digit and $\tilde{w}_1, \ldots, \tilde{w}_{m-1}$ do not, the product

$$\tilde{w}_1^{i_1} * \tilde{w}_2^{i_2} * \ldots * \tilde{w}_m^{i_m} * \ldots * \tilde{w}_j^{i_j} = \tilde{w}_1^{i_1} * \tilde{w}_2^{i_2} * \ldots * \tilde{w}_m \neq (0)$$

Hence by definition, the $\tilde{w}_1, \ldots, \tilde{w}_j$ are independent.

Lemma 2: For any $\{\tilde{w}_i: \ i = 1, \ldots, k\}$ such that $\tilde{w}_i \in I_{n_i}$ where $1 \leq n_1 < \ldots < n_k$, consider the sets P_i defined recursively by

$$P_0 = \{w_0 = (0)\}$$

$$P_i = \{\tilde{w}_i * P_{i-1}\} \cup P_{i-1} \qquad i > 0$$

Each of the P_i is a group.

Proof: The notation $\tilde{w}_i * P_{i-1}$ indicates the set of all elements defined by computing the product of $\tilde{w}_i$ and each element of P_{i-1}. From lemma 1 the $\tilde{w}_i$ are independent. Thus they may serve as generators for groups, and the lemma follows immediately.

Lemma 3: Let $\mathcal{G} = \{w_i: \ i = 0, \ldots, 2^{q-1}\}$ where $w_{j+1} > w_j$ for all j and let the sets P_j be defined as previously. Then the following recursion relation holds true

$$P_j = \{w_0\} \bigcup_{i=0}^{j-1} \left\{w_{2^i} * P_i\right\} = P_{j-1} \cup \left\{w_{2^{j-1}} * P_{j-1}\right\} \qquad \text{for } j = 1, \ldots, q$$

and

$$\left\{w_{2^i} * P_i\right\} \subset I_{n_{i+1}} \qquad \text{for } 1 \leq n_1 < \ldots < n_j < p$$

Proof: The proof proceeds by induction. First consider the minimal element of $\mathcal{G} - \{w_0\}$. This is by definition w_1, and it is obvious that $P_1 = \{w_0\} \cup \{w_1\}$ and equally obvious that $w_1 \in I_{n_1}$ for some $1 \leq n_1 < p$ so that the lemma is true for $j = 1$. Now assume that it is true for j. We need to show that this implies the lemma is true for $j + 1$.

Choose the minimal element of $\mathcal{G} - P_j$. This is by definition $w_{2^j} \notin P_j$. Since $w_{2^j} > w_{2^j - 1}$ we must have $w_{2^j} \in I_{n_{j+1}}$ where $n_{j+1} \geq n_j$. If $n_{j+1} = n_j$, then let

$w^* = w_{2^j} * w_{2^j-1} < w_{2^j-1}$. Now w^* must be an element of $w_{2^i} * P_i$ for some $0 \le i \le j - 1$ and hence $w^* \in P_j$ since we have presumably accounted for all elements of $\mathscr{G}$ that are less than w_{2^j-1} with these sets. But in this case, $w^* = w_{2^j} * w_{2^j-1}$, which implies that $w_{2^j} = w_{2^j-1} * w^*$, which implies that $w_{2^j} \in P_j$ since w^* and w_{2^j-1} are elements of P_j. By the induction assumption and lemma 2, P_j is a group and hence P_j is closed under the operation $*$. But this is a contradiction of the fact that $w_{2^j} \notin P_j$. Thus we must have $n_{j+1} > n_j$ and hence $w_{2^j} * P_j \subset I_{n_{j+1}}$. From lemma 1 we have w_{2^j} independent of all the elements of P_j, and hence it may be used as an additional generator to form the larger group $P_{j+1} = P_j \cup \left\{w_{2^j} * P_j\right\}$. Thus, assuming the lemma true for j implies truth for j + 1; hence by induction, the lemma is true.

With the preceding notation and results we can now define a unique set of generators for each element of S(p, q). That is, the ordering convention and the use of w_{2^i} as the generators permit a one-to-one mapping to be made between a group and its generators. We can also order all the elements of S(p, q) in alphabetical order in the sense that the generators of the groups will be in order. As a rather trivial example, suppose p = 2 and q = 1. Then the full group is {00, 01, 10, 11} and all the possible subgroups of order 2 are easily seen to be {00, 01}, {00, 10}, and {00, 11}, which corresponds to the defining contrasts I = A, I = B, I = BA, respectively. The generators of these groups (in terms of letters) are A, B, and BA, respectively, and it is evident that these are in alphabetical order.

We now prove an interesting recursion relation among the numbers K(p, q).

<u>Theorem</u>: Let K(p, q) denote the number of elements of S(p, q). Then

$$K(p, 0) = 1 \qquad \text{for all } p \ge 0 \tag{8}$$

and

$$K(p, q) = \sum_{n=q}^{p} 2^{n-q} K(n - 1, q - 1). \qquad \text{for all } p \ge q \ge 1 \tag{9}$$

<u>Proof</u>: The proof of K(p, 0) = 1 is trivial since for any group there is only the subgroup consisting of the identity which is of order 1.

For the proof of equation (9) it is convenient to introduce the sets of integers J_n defined by

$$J_n = [1, 2, \ldots, 2^n - 1] \tag{10}$$

$$= \bigcup_{j=1}^{n} I_j \quad \text{for } n = 1, \ldots, p$$

Let the generators (uniquely defined for any group by the previously defined conventions) be $w_{2^{i-1}}$ for $i = 1, q$. Suppose we require that $w_{2^{q-1}} \in I_n$ for some fixed n such that $q \leq n \leq p$ ($w_{2^{q-1}}$ cannot be an element of I_n for $n < q$). For any such choice of n and for $w_{2^{i-1}}$ (where $i = 1, \ldots, q - 1$) to be valid choices of generators, it must be true that $w_{2^{i-1}} \in J_{n-1}$ for $i = 1, \ldots, q - 1$. These $q - 1$ generators generate a subgroup of order 2^{q-1} from a group of order 2^{n-1}; hence there are $K(n - 1, q - 1)$ such subgroups. For any choice of $w_{2^{q-1}} \in I_n$ and any choice of the other $q - 1$ generators $w_{2^{i-1}}$ ($i = 1, \ldots, q - 1$), there are then determined $2^{q-1} - 1$ other elements in I_n, which belong to the resulting group; that is, $P_q \subset I_n$. No other elements of I_n belong to this group. This choice of $w_{2^{q-1}}$ thus rules out using the $2^{q-1} - 1$ determined elements of I_n as generators. If we use the same $q - 1$ first generators and use some other choice of $w_{2^{q-1}}$ from I_n that is not ruled out, then again, $2^{q-1} - 1$ other elements of I_n are determined and cannot be used as generators. Thus, proceeding in this manner, we may note that each choice of $w_{2^{i-1}}$ ($i = 1, \ldots, q - 1$) generates a partitioning of I_n into

$$\frac{2^{n-1}}{2^{q-1}} = 2^{n-q}$$

sets, each of which is associated with a unique group. There are, then, 2^{n-q} possible new groups. Thus there are $2^{n-q}K(n - 1, q - 1)$ unique subgroups of order 2^q, which have $w_{2^{q-1}} \in I_n$. No group with $w_{2^{q-1}} \in I_i$ can be identical to any group with $w_{2^{q-1}} \in I_j$ when $j \neq i$. Hence, to find the number of subgroups of order 2^q from the full group of order 2^p, one simply sums, over the permissible range of n, the quantities $2^{n-q}K(n - 1, q - 1)$. This yields

$$K(p, q) = \sum_{n=q}^{p} 2^{n-q}K(n - 1, q - 1)$$

TABLE I. - VALUES OF K(p, q)

p	q								
	0	1	2	3	4	5	6	7	8
0	1	---	---	---	---	---	---	---	---
1	1	1	---	---	---	---	---	---	---
2	1	3	1	---	---	---	---	---	---
3	1	7	7	1	---	---	---	---	---
4	1	15	35	15	1	---	---	---	---
5	1	31	155	155	31	1	---	---	---
6	1	63	651	1 395	651	63	1	---	---
7	1	127	2 667	11 811	11 811	2 667	127	1	---
8	1	255	10 795	97 155	200 787	97 155	10 795	255	1

Table I presents the numbers $K(p, q)$ for the values $p = 0, 8$ and $q = 0, p$.

Corollary: For $p \geq q \geq 1$

$$K(p, q) = K(p - 1,\ q) + 2^{p-q}K(p - 1,\ q - 1) \tag{11}$$

Proof: The proof is trivial on examination of the expansions for $K(p, q)$ and $K(p - 1,\ q)$ given by equation (9) of the theorem.

The corollary provides a simpler recursion relation for practical use. It is also interesting to note the similarity between this relation and the recursion relation that determines the elements of Pascal's triangle.

It should also be noted that Burnside (ref. 16, p. 111) has found a different recursion relation for Abelian groups whose elements are of prime order.

THE ALGORITHM FOR GENERATING SUBGROUPS

The flow diagram of the subroutine that calculates the subgroups by this algorithm is presented in figure 1. To illustrate its operation and to provide motivation for the procedure, the calculations for $p = 4$ and $q = 3$ are given. Then the diagram in figure 1 is discussed. The definitions of the symbols used in figure 1 are included in table II.

The algorithm makes use of a doubly subscripted array, LIST(I, J), where $J = 1, \ldots, q$ and $I = 1, \ldots, 2^{p-(q-J)} - 1$. The array is initially set to 0 and then used to indicate whether particular integers are eligible for use as generators. The use of LIST is indicated for $p = 4$ and $q = 3$ in table III. For these values there are 15 distinct contrast groups as reflected by the 15 stages numbered in the last column. The

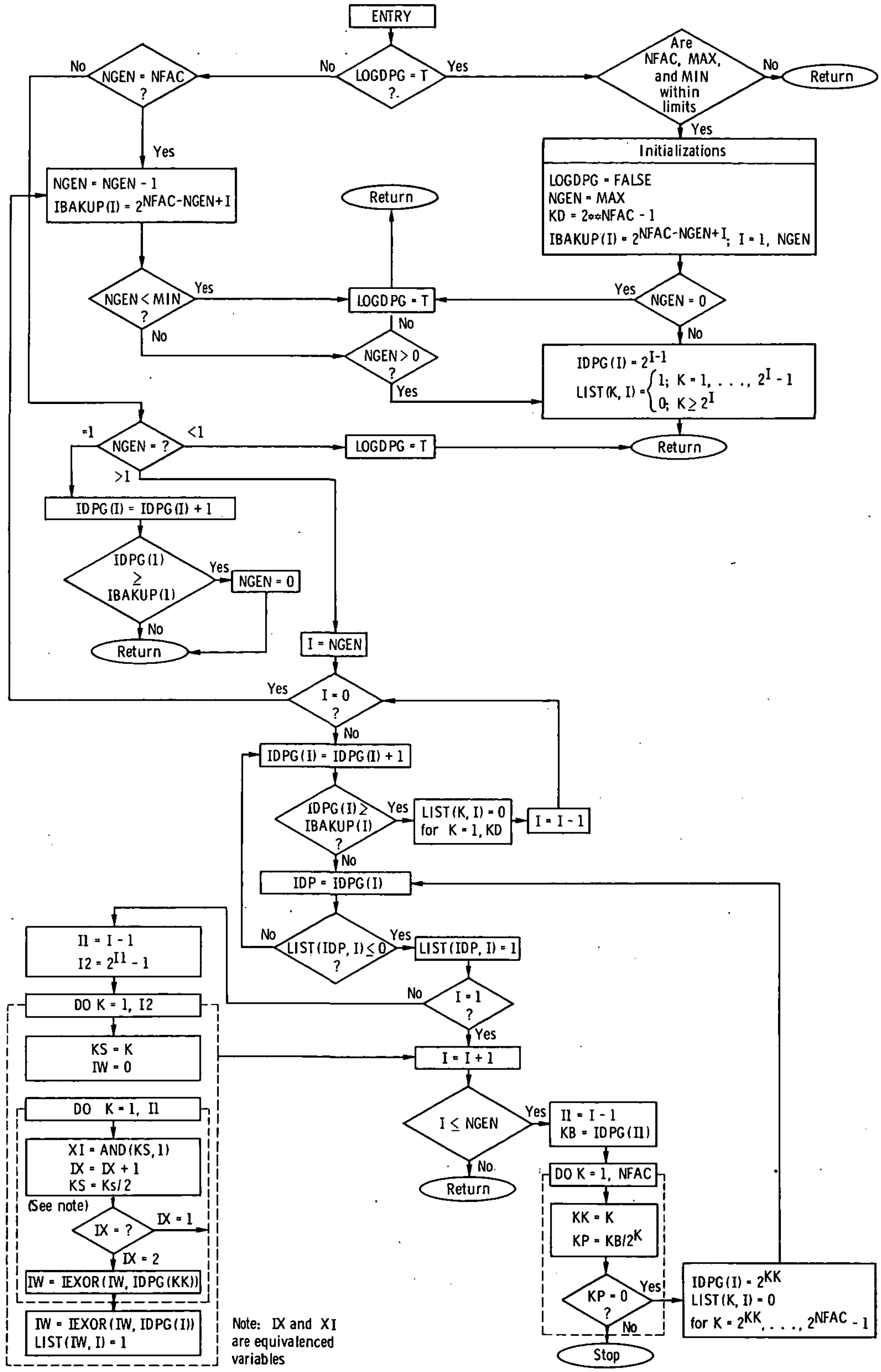

Figure 1. - Flow diagram for subroutine DPGGEN.

TABLE II. - DEFINITIONS OF SYMBOLS USED IN THE

SUBROUTINE DPGGEN

IBAKUP(I)	Upper limit for value the Ith generator may assume. Equals 2^{p-q+I}
IDP	Temporary storage for elements of IDPG(I)
IDPG(I)	The generators of groups
IW	Temporary storage for products of elements in group
IX, XI I1, I2 KB, KK, KS	Temporary storage
KD	Equals $2^p - 1$
LIST(I, J)	Indicator array used to determine if integer may be used as generator
LOGDPG	Logical variable used to indicate when all sets of generators for groups have been found
MASK	Variable set to 1, used in computing elements of group.
MAX	Maximum value for q
MIN	Minimum value for q
NFAC	p
NGEN	Number of generators or current value of q

values tabulated are LIST(I, J) at each of the stages. The J values are given in the second column and correspond to the number of generators. The I values and the contrasts that each of the I values is associated with are given in the headings at the top of the table. The first two columns give the numerical and letter values of the generators at the stages.

The procedure begins with the generators A, B, and C (1, 2, and 4). This is the first set of generators in the alphabetical ordering. Then LIST(1, 1) is set to 1 to indicate that A is the first generator. LIST(2, 1) is set to 1 because A cannot be the second generator. LIST(2, 2) and LIST(2, 3) are set to 1 to indicate that B is the second generator and that BA is in the group generated by A and B. Because BA is in this group, it may not be chosen as a generator. (We find that BA is in this group by computing the product of A and B by use of the IEXOR function.)

The first legitimate choice for the third generator is C. Then LIST(I, 3) for I = 5, 6, 7 is set to 1 because CA, CB, and CBA are also in the group generated by A, B, and C, and hence may not be used as generators. (These elements are found by computing the product of C with the group generated by A and B by use of the IEXOR function.) To show that 1, 2, and 4 are the generators, the 1's at LIST(1, 1), LIST(2, 2), and LIST(4, 3) are underlined.

To progress to stage 2, LIST(I, 3) is sequentially searched beginning at I = 8 until a zero is found. This is LIST(8, 3) from stage 1. The second set of generators corre-

TABLE III. - EXAMPLE OF LIST(I, J) USAGE FOR p = 4 AND q = 3

Stage	J	I															Generators	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Numbers	Letters
		Contrasts																
		A	B	BA	C	CA	CB	CBA	D	DA	DB	DBA	DC	DCA	DCB	DCBA		
1	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	A
	2	1	[a]$\underline{1}$	1	0	0	0	0	0	0	0	0	0	0	0	0	2	B
	3	1	1	1	[a]$\underline{1}$	1	1	1	0	0	0	0	0	0	0	0	4	C
2	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	A
	2	1	[a]$\underline{1}$	1	0	0	0	0	0	0	0	0	0	0	0	0	2	B
	3	1	1	1	1	1	1	1	[a]$\underline{1}$	1	1	1	0	0	0	0	8	D
3	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	A
	2	1	[a]$\underline{1}$	1	0	0	0	0	0	0	0	0	0	0	0	0	2	B
	3	1	1	1	1	1	1	1	1	1	1	1	[a]$\underline{1}$	1	1	1	12	DC
4	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	A
	2	1	1	1	[a]$\underline{1}$	1	0	0	0	0	0	0	0	0	0	0	4	C
	3	0	0	0	0	0	0	0	[a]$\underline{1}$	1	0	0	1	1	0	0	8	D
5	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	A
	2	1	1	1	[a]$\underline{1}$	1	0	0	0	0	0	0	0	0	0	0	4	C
	3	0	0	0	0	0	0	0	1	1	[a]$\underline{1}$	1	1	1	1	1	10	DB
6	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	A
	2	1	1	1	1	1	[a]$\underline{1}$	1	0	0	0	0	0	0	0	0	6	CB
	3	0	0	0	0	0	0	0	[a]$\underline{1}$	1	0	0	0	0	1	1	8	D
7	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	A
	2	1	1	1	1	1	[a]$\underline{1}$	1	0	0	0	0	0	0	0	0	6	CB
	3	0	0	0	0	0	0	0	1	1	[a]$\underline{1}$	1	1	1	1	1	10	DB
8	1	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	0	22	B
	2	0	0	0	[a]$\underline{1}$	0	1	0	0	0	0	0	0	0	0	0	4	C
	3	0	0	0	0	0	0	0	[a]$\underline{1}$	0	1	0	1	0	1	0	8	D
9	1	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	0	2	B
	2	0	0	0	[a]$\underline{1}$	0	1	0	0	0	0	0	0	0	0	0	4	C
	3	0	0	0	0	0	0	0	1	[a]$\underline{1}$	1	1	1	1	1	1	9	DA
10	1	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	0	2	B
	2	0	0	0	1	[a]$\underline{1}$	1	1	0	0	0	0	0	0	0	0	5	CA
	3	0	0	0	0	0	0	0	[a]$\underline{1}$	0	1	0	0	1	0	1	8	D
11	1	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	0	2	B
	2	0	0	0	1	[a]$\underline{1}$	1	1	0	0	0	0	0	0	0	0	5	CA
	3	0	0	0	0	0	0	0	1	[a]$\underline{1}$	1	1	1	1	1	1	9	DA
12	1	1	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	3	BA
	2	0	0	0	[a]$\underline{1}$	0	0	1	0	0	0	0	0	0	0	0	4	C
	3	0	0	0	0	0	0	0	[a]$\underline{1}$	0	0	1	1	0	0	1	8	D
13	1	1	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	3	BA
	2	0	0	0	[a]$\underline{1}$	0	0	1	0	0	0	0	0	0	0	0	4	C
	3	0	0	0	0	0	0	0	1	[a]$\underline{1}$	1	1	1	1	1	1	9	DA
14	1	1	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	3	BA
	2	0	0	0	1	[a]$\underline{1}$	1	1	0	0	0	0	0	0	0	0	5	CA
	3	0	0	0	0	0	0	0	[a]$\underline{1}$	0	0	1	0	1	1	0	8	D
15	1	1	1	[a]$\underline{1}$	0	0	0	0	0	0	0	0	0	0	0	0	3	BA
	2	0	0	0	1	[a]$\underline{1}$	1	1	0	0	0	0	0	0	0	0	5	CA
	3	0	0	0	0	0	0	0	1	[a]$\underline{1}$	1	1	1	1	1	1	9	DA

[a]Underscored 1's indicate contrasts used as the generators at each stage.

sponding to the second group in the alphabetical order is then A, B, D (1, 2, 8). Since DA, DB, and DBA are elements of the resulting group and their use as the third generator would lead to the same group, LIST(I, 3) I = 9, 10, 11 are set to 1. Again, LIST(I, 3) is searched for the next zero beginning at I = 9, which is at I = 12. Thus stage 3 shows the generators to be A, B, DC (1, 2, 12). LIST(I, 3) is set to 1 for I = 12, 13, 14, 15 to indicate that those contrasts may not be used as generators again. At this point, LIST(I, 3) = 1 for all I. Thus LIST(I, 2) is searched beginning at I = 4 for the first zero, which is at I = 4. Hence, the new second generator is C(4). LIST(I, 3) is now set to zeros for I = 8, . . ., 15. Then LIST(I, 3) is searched beginning at I = 8 for the first zero. This is at I = 8. Then LIST(I, 3) is set to 1 for I = 8, 9, 12, and 13 to indicate that DA, DCA, and DC are in the group generated by A, C, D. This procedure is repeated until stage 15. At stage 15 LIST(I, 3) is all ones from I = 8 to I = 15. Thus we examine LIST(I, 2) from I = 4 for the first zero. This occurs at I = 8, which is greater than $2^{4-(3-2)} - 1 = 7$. Hence, this is not a valid choice of generator. Thus, we examine LIST(I, 1) from I = 2 for the first zero. This is at $I = 4 > 2^{4-(3-1)} - 1 = 3$. This is not a valid choice for generator. This indicates that the procedure is completed and all the subgroups have been found.

Figure 1 is the flow diagram of the subroutine DPGGEN, which appears in appendix B. This program is written in FORTRAN IV for the IBM 360/67 under TSS. The subroutine requires the following variables in COMMON:

NFAC p

NGEN q

MIN lower limit of q

MAX upper limit of q

LOGDPG logical variable set TRUE on first entry to DPGGEN. Its return value is FALSE until all subgroups of order 2^q, MIN $\leq q \leq$ MAX have been generated. Then the return value is TRUE

IDPG(10) contains the integer representations of the generators

The input variables are NFAC, MIN, MAX, and LOGDPG. The output variables are NGEN, LOGDPG, and IDPG. NFAC specifies the value of p. MIN and MAX denote the lower and upper limits on q. Then the generators of all subgroups of order 2^q for MIN $\leq q \leq$ MAX are found. The first entry to DPGGEN must have LOGDPG set to TRUE. Each time DPGGEN is called, the next set of generators in the alphabetical ordering is returned in the array IDPG. When all groups have been generated, LOGDPG is returned as TRUE. Otherwise, LOGDPG is returned as FALSE.

The subprograms GROUPS and OUTDPG are sample input and output routines that will compute and print the generators as they are found for various input values of NFAC, MIN, and MAX. Appendix B presents the program listings, sample input, and sample output. The current version is limited to handle values of $p \leq 10$ and MIN and MAX are subject to the relation $0 \leq \text{MIN} \leq \text{MAX} \leq p$.

DISCUSSION OF APPLICATIONS

Use in Sequences of Fractional Factorial Designs

There are certain tabulations of designs which are expansible in the sense that they begin as small fractions and then blocks are added to create larger fractions (Addelman (ref. 7) and Holms (ref. 8)). These tabulations were made with different criteria in mind. Namely, Addelman used the criteria of "number of estimable effects" and/or "smaller average variance of the estimable effects." Holms considers primarily the criteria of resolution level. Holms' tables were apparently constructed by trial and error. Addelman implies that some computer programs were used to evaluate his sequences of designs but does not indicate how these sequences were generated. The algorithm presented in this report would be useful in generating new tables, using new criteria, or in checking or expanding the tables of Addelman and Holms.

USE IN CONSTRUCTING PLANS WITH SPECIAL RESTRICTIONS

Often in practice an experiment will have certain restrictions on the manner in which it may be performed. For example, there may be limitations on the total number of changes of level permitted; some factors may not be permitted to have their level changed as often as others; or perhaps equipment restrictions require certain restraints on the manner of performance of the experiment and must be reflected in the design of the experiment. An example of the case where equipment restrictions limited the choice of design is in Holms and Sidik (ref. 9). Another example is described in appendix C. In these situations, the commonly tabulated designs may be of little or no value. Through generating all the possible defining contrast groups for a particular p and q, it may be possible to determine if a regular fractional factorial design exists that fits the restrictions and that will present all those that do. (Again see appendix C.)

Use in Bayesian Design of Experiments

Sidik and Holms (ref. 11) present a method of constructing the optimal design of two-level fractional factorial experiments when certain prior information is available. Reference 17 presents an algorithm and computer program for performing some of the calculations involved.

This program accepts a specified defining contrast group and certain prior information about the parameters and then maximizes an expected utility value over all the possible permutations of the physical variable-design variable matchings. This is an excessive amount of computing in many instances, however. For instance, if $p = 6$ and $q = 2$, there are $6! = 720$ permutations of the matchings, but there are only 651 distinct designs (see table I). It should also be noted that the 720 permutations are evaluated for each and every input defining contrast group. Since the program also operates in the mode that evaluates only single input matchings and since it is also designed to evaluate multiply telescoping designs, the program is still of value. It might, however, be possible to modify the program to include the algorithm presented in this report.

Another example where the algorithm may be useful is given in reference 12. In that report is presented a sequential adaptive design procedure for discriminating among several linear models. The general linear model is considered, but, of course, two-level factorial designs represent a subclass of the linear models. The criteria used to select an experiment is the expected Kullback-Leibler information function. As presented in reference 12, the method of determining the optimal experiment is exhaustive examination of the space of possible experiments.

For 2^p factorial experiments there are 2^p unique treatment combinations. The experiment space then consists of all the possible combinations of these treatment combinations that do not exceed the maximum allowable number of observations (denoted J_{MAX}). If we include with the treatments the possibility of not taking an observation, then there are $\left(2^p + 1\right)^{J_{MAX}}$ potential experiments. Since this set must be exhaustively searched to determine the optimal experiment, a large number of potential experiments is a drawback to the successful application of the sequential procedure. For example, for $p = 7$ and $J_{MAX} = 8$, there are approximately 10^{17} experiments. We may turn to the suboptimal procedure of considering only those sets of treatment combinations that form regular fractional replicates of 2^p factorials. From the numbers in table I we find that there are approximately 3×10^5 possible regular fractional replicates containing eight or fewer observations. This number of experiments is much more reasonable to evaluate and should be within the capability of many computers.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, April 3, 1973,
503-35.

APPENDIX A

SYMBOLS

A, B, BA, etc.	contrasts
C	group of contrasts
$\mathcal{G}$	group in S(p, q)
I	identity element of C
I_n	set of integers from 2^{n-1} through $2^n - 1$
$(i_p, \ldots, i_1)$	binary representations of integers from 0 through $2^p - 1$
J_{MAX}	maximum number of allowable observations
J_n	set of integers from 1 through $2^n - 1$
K(p, q)	number of elements of S(p, q)
LIST(I, J)	array of indicator values used in algorithm
P_n	set of elements $\left(w_0, \ldots, w_{2^n-1}\right)$
p	number of factors or independent variables
Q_n	set of elements $\left(w_{2^{n-1}}, \ldots, w_{2^n-1}\right)$
q	$(1/2)^q$ is fraction of full replicate under consideration
S(p, q)	set of all distinct subgroups of order 2^q of full group of order 2^p
$w * P$	set of elements defined by product of w with all elements of P
w_i, $\tilde{w}_i$	element of $\mathcal{G}$
X_A, X_B, etc.	independent variables
Y	response variable
$\beta_I, \beta_A, \beta_B, \ldots$	parameters of the model eq. (1)
$\beta_0, \beta_1, \beta_2, \ldots$	parameters of the model eq. (2)
δ	random error variable
$\in$	element of
$\notin$	not an element of
$*$	group product
$\subset$	subset of
$\cup$	union of

APPENDIX B

FORTRAN LISTING OF THE PROGRAM AND SAMPLE OUTPUT

Included in this appendix are the FORTRAN listings of the subroutine DPGGEN and the sample input and output routines. Also given is a sample output.

The two function subprograms AND and IEXOR are available in the FORTRAN IV language at the Lewis Research Center and may not be available at other computer installations. What they do is therefore explained and the user can write function subprograms providing the same capabilities in a language compatible with the available computer system.

(1) AND(A, B). A real function of the real or integer variables A and B. Like bit positions of A and B are compared. A one is placed in those positions of the result where there are ones in both A and B, and a zero is placed in the result otherwise.

(2) IEXOR(A, B). An integer function of the real or integer arguments A and B. Like bit positions of A and B are compared. A one is placed in those positions of the result where exactly one of A or B is a 1, and a zero is placed in the result otherwise.

MAIN INPUT PROGRAM

```
      COMMON NFAC,NGEN,MIN,MAX,LOGDPG,IDPG(10)
      LOGICAL LOGDPG
   10 READ(5,5000) NFAC,MIN,MAX
      WRITE(6,6001) NFAC,MIN,MAX
      IF(NFAC.LT.1.OR.NFAC.GT.10) GO TO 6500
      IF(MIN.LT.0)  GO TO 6600
      IF(MAX.GT.NFAC) GO TO 6700
      IF(MAX.LT.MIN) GO TO 6800
      LOGDPG=.TRUE.
   20 CALL DPGGEN
      IF(LOGDPG) GO TO 10
      IF(NGEN.EQ.0) WRITE(6,7000)
      IF(NGEN.NE.0) CALL OUTDPG
      GO TO 20
 5000 FORMAT(3I6)
 7000 FORMAT(10X,'I')
 6001 FORMAT(' NO. OF FACTORS =  ' I6/' MIN NO. GENERATORS = ' I6/' MAX NO. GENERATORS = 'I6)
 6500 WRITE(6,6501)
 6501 FORMAT(' TERMINATED. NFAC OUT OF RANGE')
      GO TO 10
 6600 WRITE(6,6601)
 6601 FORMAT(' TERMINATED. MIN TOO SMALL')
      GO TO 10
 6700 WRITE(6,6701)
 6701 FORMAT(' TERMINATED. MAX TOO LARGE')
      GO TO 10
 6800 WRITE(6,6801)
 6801 FORMAT(' TERMINATED. MAX IS LESS THAN MIN')
      GO TO 10
      END
```

SAMPLE OUTPUT PROGRAM

```
      SUBROUTINE OUTDPG
      COMMON NFAC,NGEN,MAXBLK,MINBLK,LOGDPG,IDPG(10)
      DATA MASK/1/,BLANK/' '/
      DIMENSION WD(110)
      DIMENSION ALPH(10)
      DATA ALPH/'A','B','C','D','E','F','G','H','J','K'/
      EQUIVALENCE (KS,SK)
      I11=11
      DO 20 I=1,110
20    WD(I)=BLANK
      DO 500 I=1,NGEN
      ISTR=(I-1)*I11
      IS=IDPG(I)
      DO 50 K=1,NFAC
      SK=AND(IS,MASK)
      KS=KS+1
      IS=IS/2
      GO TO (50,40),KS
40    ISTR=ISTR+1
      WD(ISTR)=ALPH(K)
50    CONTINUE
500   CONTINUE
      WRITE(6,600) (WD(I),I=1,110)
600   FORMAT(' ' 110A1)
      RETURN
      END
```

SUBROUTINE DPGGEN

```
      SUBROUTINE DPGGEN
      COMMON NFAC,NGEN,MIN,MAX,LOGDPG,IDPG(10)
      LOGICAL LOGDPG
      DATA MASK/1/
C
C*****************************************************************
C
      DIMENSION IPOW(10)   ,IBAKUP(10)
      DATA IPOW/2,4,8,16,32,64,128,256,512,1024/
      DIMENSION LIST(1024,10)
      EQUIVALENCE (KS,SK),(IX,XI)
C
C*****************************************************************
C
      IF(.NOT.LOGDPG) GO TO 500
      IF(NFAC.GT.10.OR.NFAC.LT.0) RETURN
      IF(MIN.LT.0) RETURN
      IF(MAX.GT.NFAC) RETURN
      IF(MAX.LT.MIN) RETURN
      LOGDPG=.FALSE.
      NGEN=MAX
      IF(NGEN.LE.0) RETURN
      MNGEN=MIN
      KD=2**NFAC-1
      DO 5 J=1,NGEN
      IBAKUP(J)=2**(NFAC-NGEN+J)
    5 CONTINUE
C
C*****************************************************************
C
   10 IF(NGEN) 540,540,15
   15 DO 20 K=1,NGEN
      IDPG(K)= 2**(K-1)
   20 CONTINUE
      IF(NGEN.GE.NFAC) RETURN
      DO 30 J=1,NGEN
      DO 30 K=1,KD
      LIST(K,J) = 0
   30 CONTINUE
      DO 80 J=1,NGEN
      I2=2**J-1
      DO 80 K=1,I2
      LIST(K,J)= 1
   80 CONTINUE
      RETURN
C
C*****************************************************************
C
  100 NGEN = NGEN-1
      DO 105 J=1,NGEN
      IBAKUP(J)=2**(NFAC-NGEN+J)
  105 CONTINUE
      IF(NGEN-MIN) 540,10,10
C
```

```
C*****************************************************************************
C
  500 CONTINUE
      IF(NGEN.EQ.NFAC) GO TO 100
      IF(NGEN-1) 540,520,580
  520 IDPG(1) = IDPG(1)+1
      IF(IDPG(1)-IBAKUP(1)) 2500,530,530
  530 NGEN=0
      RETURN
  540 LOGDPG = .TRUE.
      RETURN
C
C*****************************************************************************
C
  580 I= NGEN
  590 IF(I.EQ.0) GO TO  100
  592 IDPG(I) = IDPG(I)+1
      IF(IDPG(I)-IBAKUP(I)) 690,595,595
  595 CONTINUE
C
C
      DO 620 K=1,KD
      LIST(K,I)= 0
  620 CONTINUE
  625 I=I-1
      GO TO 590
C
C
  690 IDP = IDPG(I)
      IF(LIST(IDP,I)) 695,695,592
  695 LIST(IDP,I)=1
      IF(I.EQ.1) GO TO 835
      I1= I-1
      I2= 2**I1-1
      DO 830  K=1,I2
      KS=K
      IW=0
      DO 730 KK=1,I1
      XI= AND(KS,MASK)
      IX = IX+1
      KS=KS/2
      GO TO (730,725),IX
  725 IW = LEXOR(IW,IDPG(KK))
  730 CONTINUE
      IW = LEXOR(IW,IDPG(I))
      LIST(IW,I)=1
  830 CONTINUE
  835 I=I+1
      IF(I-NGEN) 840,840,2500
  840 I1= I-1
      KB=IDPG(I1)
      DO 845 K=1,NFAC
      KK=K
      KP = KB/IPOW(K)
```

```
      IF(KP) 850,850,845
  845 CONTINUE
      CALL EXIT
      STOP
  850 KL=2**KK
      DO 865 K=KL,KD
      LIST(K,I)=0
  865 CONTINUE
      IDPG(I)= KL
      GO TO 690
C
C*********************************************************************
C
 2500 RETURN
      END
```

SAMPLE OUTPUT FOR NFAC = 4, MAX = 3, AND MIN = 2

A	B	C
A	B	D
A	B	CD
A	C	D
A	C	BD
A	BC	D
A	BC	BD
B	C	D
B	C	AD
B	AC	D
B	AC	AD
AB	C	D
AB	C	AD
AB	AC	D
AB	AC	AD
A	B	
A	C	
A	BC	
A	D	
A	BD	
A	CD	
A	BCD	
B	C	
B	AC	
B	D	
B	AD	
B	CD	
B	ACD	
AB	C	
AB	AC	
AB	D	
AB	AD	
AB	CD	
AB	ACD	
C	D	
C	AD	
C	BD	
C	ABD	
AC	D	
AC	AD	
AC	BD	
AC	ABD	
BC	D	
BC	AD	
BC	BD	
BC	ABD	
ABC	D	
ABC	AD	
ABC	BD	
ABC	ABD	

APPENDIX C

EXAMPLE OF APPLICATION

In this appendix we discuss an actual experiment being planned at Lewis which required a nonstandard design. A knowledge of the technique of doubly telescoping designs is presupposed. This is discussed in Holms (ref. 8), Holms and Sidik (ref. 9), and Holms (ref. 18). A different version of this particular experiment is discussed in Holms (ref. 18).

The purpose of the experiment is to study the amount of corrosion by liquid lithium on sealed metal capsules at high temperatures. The six two-level variables considered are

X_A time (500 and 2000 hr)

X_B temperature (1300 and 1500 K)

X_C amount of oxygen in the capsule alloy (40 and 200 ppm)

X_D amount of nitrogen in the capsule alloy (20 and 200 ppm)

X_E presence of the alloy TZM in the system (absent, present)

X_F amount of nitrogen in the liquid lithium (5 and 500 ppm)

There are thus 64 combinations of the levels. The assembled capsules are to be tested in a near vacuum, at the indicated temperature levels and time levels. The available equipment is such that there are four electrically heated furnaces, each of which can accommodate four capsules, in a single vacuum chamber. Each furnace has one temperature control system and is insulated from the other furnaces within the vacuum chamber. Once the vacuum chamber is loaded and the test begun, it may not be opened again until the tests are completed.

Thus, all four capsules within a furnace must be tested at the same temperature, and all 16 capsules within a chamber loading must be tested for the same time duration. If the experiment is designed as a doubly telescoping experiment, where the two sources of block effects are furnaces and chamber loadings, then the defining contrasts for the experiment consisting of one block of four treatments must contain the main effects A and B.

The program presented in appendix B was used to generate all the designs for $p = 6$ and $q = 4$. From this list all the groups containing A and B, but no other single letters, were found and are given in table IV. Certain of these groups are equivalent to certain of the others if the appropriate permutations of the letters are made. The three unique groups are 1, 3, and 4. The others are equivalent to one of these, as is indi-

TABLE IV. - GENERATORS OF GROUPS CONTAINING

A AND B BUT NO OTHER MAIN EFFECTS

Set	Generators	Equivalence	Permutation
1	A B CD CE	*	--------------------------
2	A B CD CF	1	E → F, F → E
3	A B CD EF	*	--------------------------
4	A B CD CEF	*	--------------------------
5	A B CE CF	1	D → F, F → D
6	A B CE DF	3	D → E, E → D
7	A B CE CDF	4	D → E, E → D
8	A B DE CF	3	D → F, F → D
9	A B DE DF	1	C → F, F → D, D → C
10	A B DE CDF	4	C → E, E → D, D → C
11	A B CDE DF	4	D → F, F → D
12	A B CDE DF	4	C → F, F → D, D → C
13	A B CDE CDF	4	C → E, E → C, D → F, F → D

TABLE V. - THE FULL GROUPS FOR THE

STARRED GENERATORS OF TABLE IV

1	3	4	1	3	4
I	I	I	CE	EF	CEF
A	A	A	ACE	AEF	ACEF
B	B	B	BCE	BEF	BCEF
AB	AB	AB	ABCE	ABEF	ABCEF
CD	CD	CD	DE	CDEF	DEF
ACD	ACD	ACD	ADE	ACDEF	ADEF
BCD	BCD	BCD	BDE	BCDEF	BDEF
ABCD	ABCD	ABCD	ABDE	ABCDEF	ABDEF

cated by the sixth column when the permutations indicated in column seven are made. The full groups corresponding to 1, 3, and 4 are given in table V. Group 3 was chosen to be used since it is the only group that has a six-letter word in it. This has the following advantages. First, the contrast I = ABCDEF may be used to define a resolution six half replicate after two chamber loadings. Second, experimenting may start with the first chamber loading planned for 500 hours. Then the second chamber loading may be planned for 2000 hours. It would be possible, however, in the event of budgetary problems, further information, etc., to only run the second loading 500 hours also. This

TABLE VI. - DETAILS OF THE EXPERIMENT IN APPENDIX C

i	j	Loading	Furnace					i	j	Loading	Furnace				
				1	2	3	4					1	2	3	4
				k								k			
				-1		+1						-1		+1	
				l								l			
				-1	+1	-1	+1					-1	+1	-1	+1
-1	-1	1		(1)	b	b	(1)	1	-1	3		(1)	b	b	(1)
				(1)	e	d	de					ce	c	cde	cd
			(1)	cd	cde	c	ce				a	de	d	e	(1)
				ef	f	def	df					cf	cef	cdf	cdef
				cdef	cdf	cef	cf					df	def	f	ef
	1	2		b	(1)	(1)	b		1	4		b	(1)	(1)	b
				ce	c	cde	cd					(1)	e	d	de
			a	de	d	e	(1)				(1)	cd	cde	c	ce
				cf	cef	cdf	cdef					ef	f	def	df
				df	def	f	ef					cdef	cdf	cef	cf

TABLE VII. - BLOCK CONFOUNDING RESULTING FROM DESIGN

USED IN TABLE VI FOR CORROSION EXPERIMENT

Block variable	Loadings completed		
	4	2	1
m	I	I ABCDEF	I A BCDEF ABCDEF
k	ACD	ACD BEF	CD ACD BEF ABEF
l	AEF	AEF BCD	EF AEF BCD ABCD
kl	CDEF	AB CDEF	B AB CDEF ACDEF

Block variable	Loadings completed		
	4	2	1
j	BCDEF	A BCDEF	-- --
jk	ABEF	CD ABEF	-- --
jl	ABCD	EF ABCD	-- --
jkl	B	B ACDEF	-- --
i	ABCDEF	------	--
ik	BEF	------	--
il	BCD	------	--
ikl	AB	------	--
ij	A	------	--
ijk	CD	------	--
ijl	EF	------	--
ijkl	ACDEF	------	--

would cause time to be suppressed as a variable and would result in a full replicate on the remaining five variables.

Table VI presents the details of the experiment plan. The block variables i and j denote the loading block effects. The block variables k and l denote the furnace block effects.

Table VII presents the defining contrasts and the contrasts that are confounded with block effects on completion of loadings 1, 2, or 4.

An experiment design is not complete until the method to be used for analyzing the results is also described. In this experiment there are two potential methods of analysis. Which method is used depends on the assumptions made about the furnace and loading block effects. Previous work concerning telescoping and multiply telescoping designs has assumed that the block effects can be classified as crossed. In the current experiment this permits the experimenter to identify which contrasts are confounded with which block effects. If the block effects are assumed to be additive, then those contrasts confounded with interactions between i, j and k, l effects are zero. Thus at the half replicate for example, ACD = BEF, AEF = BCD, and AB = CDEF are confounded with furnace block effects; A = BCDEF is confounded with the loading block effect. However, CD = ABEF, EF = ABCD, and B = ACDEF are estimable contracts since the corresponding block effect interactions they are confounded with may be assumed to be zero.

If the block effects are assumed to be random variables, however, the experiment falls within the class of split-plot or hierarchical types. In that event, the estimation of effects is made in the usual manner. Significance tests, however, must be made with the error structure of the experiment in mind. For example, consider the full replicate experiment. The parameter estimates of A, BCDEF, and ABCDEF are all confounded with loading block effects and balanced with respect to all others. Thus, one could use the mean squares due to the effects BCDEF and ABCDEF as the error term for testing the A effect. Likewise, the contrasts, B, AB, CD, EF, ACD, AEF, BEF, BCD, ABCD, ABEF, CDEF, and ACDEF are confounded with furnace block and furnace × loading block effects. Thus, one could use the mean squares due to the three factor and higher order interactions in this list as the error term for testing B, AB, CD, and EF.

The remainder of the contrasts are balanced with respect to all block effects and hence all have the same error structure. These may then be tested in the usual manner.

REFERENCES

1. Fisher, R. A.: The Arrangement of Field Experiments. J. Ministry Agriculture, vol. 23, 1926, pp. 503-513.
2. Yates, F.: The Principles of Orthogonality and Confounding in Replicated Experiments. J. Agricultural Sci., vol. 23, 1933, pp. 108-145.
3. Finney, D. J.: The Fractional Replication of Factorial Arrangements. Annals of Eugenics, vol. 12, 1945, pp. 291-301.
4. Box, G. E. P.; and Wilson, K. B.: On the Statistical Attainment of Optimum Conditions. J. Roy. Stat. Soc., Ser. B, vol. 13, 1951, pp. 1-45.
5. Daniel, Cuthbert: Fractional Replication in Industrial Research. Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability. Vol. 5. Univ. California Press, 1956, pp. 87-98.
6. Webb, Steve R.: Saturated Sequential Factorial Designs. Technometrics, vol. 10, no. 3, Aug. 1968, pp. 535-550.
7. Addelman, Sidney: Sequences of Two-Level Fractional Factorial Plans. Technometrics, vol. 11, no. 3, Aug. 1969, pp. 477-509.
8. Holms, Arthur G.: Designs of Experiments as Telescoping Sequences of Blocks for Optimum Seeking (As Intended for Alloy Development). NASA TN D-4100, 1967.
9. Holms, Arthur G.; and Sidik, Steven M.: Design of Experiments as "Doubly Telescoping" Sequences of Blocks with Application to a Nuclear Reactor Experiment. Technometrics, vol. 13, no. 3, Aug. 1971, pp. 559-574.
10. Lindenberg, W.; and Gerhards, L.: Combinatorial Construction by Computer of the Set of All Subgroups of a Finite Group by Composition of Partial Sets of Its Subgroups. Computational Problems in Abstract Algebra. John Leech, ed., Pergamon Press, 1970, pp. 75-82.
11. Sidik, Steven M.; and Holms, Arthur G.: Optimal Design Procedure for Two-Level Fractional Factorial Experiments Given Prior Information About Parameters. NASA TN D-6527, 1971.
12. Sidik, Steven M.: Kullback-Leibler Information Function and the Sequential Selection of Experiments to Discriminate Among Several Linear Models. Ph.D. Thesis, Case-Western Reserve Univ., 1972.

13. Davies, Owen L., ed.: The Design and Analysis of Industrial Experiments. Second ed., Hafner Publ. Co., 1960.

14. Peng, K. C.: The Design and Analysis of Scientific Experiments. Addison-Wesley Publ. Co., 1967.

15. John, Peter W. M.: Statistical Design and Analysis of Experiments. Macmillan Co., 1971.

16. Burnside, William: Theory of Groups of Finite Order. Second ed., Dover Publications, 1955.

17. Sidik, Steven M.: NAMER - A FORTRAN IV Program for Use in Optimizing Designs of Two-Level Factorial Experiments Given Partial Prior Information. NASA TN D-6545, 1972.

18. Holms, Arthur G.: Design of Experiments as "Multiply Telescoping" Sequences of Blocks with Application to Corrosion by Liquid Metals. NASA TN D-6452, 1971.

NASA-Langley, 1973 —— 19 E-7365

Page Intentionally Left Blank

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546