COMPUTER PROGRAM
FOR DIMENSIONAL ANALYSIS

by A. D. Sloan and W. W. Happ

Electronics Research Center
Cambridge, Mass.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1969
COMPUTER PROGRAM
FOR DIMENSIONAL ANALYSIS

By A. D. Sloan and W. W. Happ
Electronics Research Center
Cambridge, Mass.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

APRIL
1969
In order to scale models of devices (ships, space capsules, integrated circuits, etc.) and phenomena (nuclear detonations, space-charge widening, sloshing in fuel tanks, etc.), it is necessary to determine nondimensional groupings of variables. A computer program establishes sets of invariant or nondimensional sets of variables and searches for an optimum set under specified optimization criteria. This Fortran IV program is based on algorithms of integer programming. Typical running times are 2 minutes for $10^4$ sets of equations.
COMPUTER PROGRAM FOR DIMENSIONAL ANALYSIS

By A. D. Sloan and W. W. Happ
Electronics Research Center

SUMMARY

Documentation, illustrative examples, running time, and failure diagnostics are presented for a program useful in scaling models by dimensional analysis. The program, written in FORTRAN IV, performs exclusively integer operations and calculates a basis for the null space of the matrix of a transformation in terms of a given basis. Application to the calculation of B-numbers is made. The program further optimizes a matrix of integers under specified constraints. "Tearing" (or diakoptics) of large systems, defined as minimal under specified constraints, is performed.

The program is available either from the authors or from the Project COSMIC Library of computer programs at the University of Georgia, at Athens.

INTRODUCTION

A Fortran IV program has been developed for the calculation of Buckingham numbers from physical laws. Buckingham numbers are defined as follows. Given a set of variables \[(A_k) \, k = 1, \ldots, n\], dimensionally expressed in terms of some fixed reference dimensions, a Buckingham number or "B" number is an n-tuple of numbers \[(m_k) \, k = 1, \ldots, n\], such that the product:

\[
\prod_{k=1}^{n} (A_k)^{m_k}
\]

is dimensionless. For example, let the reference dimensions be length, mass, and time \((\ell, m, t)\), and the variables be E-energy, m-mass, and c-speed of light. Then, these variables dimensionally expressed in terms of the reference dimensions by:

\[
E = \frac{m\ell^2}{t^2}
\]

\[
m = m
\]

\[
c = \frac{\ell}{t}
\]
A "B" number for these variables is \((-1,1,2)\) for \((E)^{-1}(m)^{1}(c)^{2}\) is dimensionless. Thus, "B" numbers express formulas and may be used to find dimensional relationships between variables.

In a system with a large number of variables, more than one "B" number may result. By finding all "B" numbers, it is possible to determine all the ways of decomposing the large system into smaller ones. For instance, if along with the variables \(E, m, c\), we included \(\ell\) and \(t\), in the foregoing example, we could have three "B" numbers \((-1,1,2,0,0)\) as before; expressing \(E = mc^2\), \((-1,1,0,2,-2)\), since the units of \(E\) are \(m^1 \ell^2 / t^2\) and \((0,0,-1,1,-1)\) since the units of \(c\) are \(\ell / t\).

**SCOPE**

It is assumed that \((l_k)\) and \([(m_k) \text{ for } k = 1, \ldots, n]\) are two "B" numbers for the variables \((A_k)\) and \(t\) is a real number. Scalar multiplication and addition among "B" numbers are defined by

\[
t \times (l_k) + (m_k) \equiv (t l_k + m_k) \quad k = 1, \ldots, n
\]

Observe that:

\[
\left( \sum_{k=1}^{n} (A_k)^{l_k}_+ m_k \right) = \left( \sum_{k=1}^{n} (A_k)^{l_k} \right)^t \times \left( \sum_{k=1}^{n} (A_k)^{m_k} \right)
\]

is dimensionless since \((l_k)\) and \((m_k)\) are "B" numbers. This shows that the "B" numbers form a vector space, with addition and scalar multiplication defined as before.

The program has the following functions.

1. It determines a basis for the vector space of "B" numbers; that is, it determines a set of "B" numbers such that they are linearly independent, and such that, any other "B" number can be written as a finite combination of the elements of the basis. Such a basis is called a complete set of "B" numbers.

2. According to criteria previously defined (refs. 1,2,3), the program examines other complete sets of "B" numbers and determines a set in which:
(a) The minimal sum of the absolute values of the entries
(b) The maximal number of zero entries relative to all other complete sets examined.

PROBLEM FORMULATION

Coding the Problem

The input of the problem consists of:

(1) NREFDM = the number of reference dimensions
(2) NVARIB = the number of variables
(3) IVAR = an (NREFDM) by (NVARIB) matrix, which acts as a list (indexed by the second variable) of (NREFDM) by 1 matrices. The J'th entry in this list, or equivalently the J'th column of IVAR, defines the J'th variable in terms of the reference dimensions. For example, if the J'th variable was the Boltzmann constant, which has dimensions (length)(length)(mass)/(time)/(time) (temperature), and if the reference dimensions were (length, mass, time, charge, temperature), then:

\[
\begin{align*}
IVAR(1,J) &= 2 \\
IVAR(2,J) &= 1 \\
IVAR(3,J) &= -2 \\
IVAR(4,J) &= 0 \\
IVAR(5,J) &= -1
\end{align*}
\]

(4) M = NREFDM
(5) N = the number of variables in the formula of interest
(6) IFORM = a list of N numbers which indicate which variables are in the formula of interest. The numbers refer to the second subscript of IVAR.

(7) IOPT = 1 or 2. It indicates which optimization procedure will be followed. 1 means that zeros will be maximized first and the sum of the absolute value of entries will be minimized next, while 2 means that the procedure occurs in the reverse order.
Input Format

The first input card contains NREFDM and NVARIB in 2I5 format. The next NVARIB cards contain NREFDM numbers on each in 14I5 format and form the columns of IVAR. If Q formulas are to be analyzed, then there should be Q sequences of cards in the following form:

IOPT in I5 format
M and N in 2I5 format
IForm (=N numbers) in 14I5 format.

Note that all input data must be in integer form.

Example of Problem Statement (Fig. 1)

The reference dimensions are:

(1) Length
(2) Mass
(3) Time
(4) Charge

The variables are:

(1) Surface area
(2) Mass of electrons
(3) Charge of electrons
(4) Current density
(5) Current
(6) Velocity
(7) Energy
(8) Electric potential.

The 12 input data cards for this problem are shown in Figure 1. The card at the bottom of the page is the first, while the one at the top of the page is the last. The resultant printout is shown in Figure 2. The program listing is given in the appendix.

The first card indicates that NREFDM = 4, while NVARIB = 8. Each of the next eight cards forms a column of IVAR. The second card, for example, indicated that the dimensions of variable number 1 (= surface area) are (length)$^2$, while the ninth card indicates that the dimensions of variable No. 8 (= electric potential) are (length)$^2$(mass)(time)$^{-2}$(charge)$^{-1}$. The tenth card indicates that IOPT = 1, and the eleventh card that M = 4, N = 8. The last card is IFORM. This card shows which of the variables are to be considered; presently all eight variables are included.
Figure 1.— Input data cards
THERE ARE FOUR REFERENCE DIMENSIONS AND EIGHT VARIABLES. THE ROWS BELOW ARE THE VARIABLES.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>-2</td>
<td>0</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-2</td>
<td>-1</td>
</tr>
</tbody>
</table>

THE FORMULA NOW BEING CONSIDERED IS DEFINED IN TERMS OF THE FOLLOWING EIGHT VARIABLES.

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

THE COLUMNS OF THE MATRIX BELOW ARE THE VARIABLES IN TERMS OF THE REFERENCE DIMENSIONS.

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
</tbody>
</table>


<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>-3</td>
<td>0</td>
<td>2</td>
<td>-2</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

Figure 2.- Computer Print-out for data given by Figure 1.
COMPUTER APPROACH

Complete Sets of "B" Numbers

The program proceeds as follows. Suppose IVAR is a p-by-k matrix, so that each of the k columns of IVAR expresses a variable in terms of the p reference dimension. A k-tuple \((n_1, \ldots, n_k)\) of numbers is a "B" number if and only if

\[
0 = \text{IVAR} \cdot \begin{bmatrix}
n_1 \\
\vdots \\
n_k
\end{bmatrix}
\]

and this equals the matrix product of IVAR and the column vector

\[
\begin{bmatrix}
n_1 \\
\vdots \\
n_k
\end{bmatrix}
\]

If we consider IVAR as a linear transformation, then the vector space of "B" numbers is exactly equal to the null space of IVAR, and in order to find a complete set of "B" numbers, it suffices to find a basis for this null space. This is accomplished next.

Let \(e_i = (0, \ldots, 0, 1, 0, \ldots, 0)\) be a k-tuple where the 1 occurs in the i-th spot, \(i = 1, \ldots, k\). Form the matrix whose i-th row consists of IVAR \((e_i)\). The matrix is k by p. Row reduce this matrix. As a by-product of the row reduction process one obtains a permutation \(\pi\) of \((1, \ldots, k)\) and number \(a_{i\pi}\) such that \(a_{i\pi} \neq 0\) and such that the rows of the row reduced matrix are given by

\[
\text{IVAR}[e_{\pi(1)}] = a_{21} \text{IVAR}[e_{\pi(1)}] + a_{22} \text{IVAR}[e_{\pi(2)}]
\]

\[
a_{31} \text{IVAR}[e_{\pi(1)}] + a_{32} \text{IVAR}[e_{\pi(2)}] + a_{33} \text{IVAR}[e_{\pi(3)}]
\]

\[
\ldots
\]

\[
a_{k1} \text{IVAR}[e_{\pi(1)}] + \ldots + a_{kk} \text{IVAR}[e_{\pi(k)}]
\]
where the first $s$ rows are non-zero and the last $k-s$ rows are zero. Let

$$w_1 = e_\pi(1)$$

and let

$$w_i = \sum_{y=1}^{i} a_i y e_\pi(y). \quad i = z, \ldots, k.$$ 

Since the $e_i$'s are linearly independent and since $a_{ij} \neq 0$, it follows that the $w_i$'s are linearly independent. Let $W_1$ be the space spanned by $w_1, \ldots, w_s$ and let $W_2$ be the space spanned by $w_{s+1}, \ldots, w_k$. $\text{IVAN}(W_1 \oplus W_2) = \text{IVAR}(W_1)$ so that $W_2$ is the null-space of IVAR. Let $\theta$ be the inverse permutation to $\pi$. Then

$$[a_i \theta(y)] \quad i = 1 + 1, \ldots, k$$

expresses a basis of $W_2$ in terms of the original one. (Note that the $e_i$ could have been chosen to be any $k$ linearly independent $k$-tuples.)

In this way the program determines a complete set of "B" numbers. The input entries of IVAR must be integers, and as a result, the entries of the "B" vectors forming a complete set are integer.

Optimization

Next, it is determined if there are any common factors among the entries of the "B" numbers, and whether there is only one "B" number in the complete set. If there is, there is no need to go through the optimizing procedure. But if there is more than one "B" number, then an attempt to find a more optimal complete set is made. The procedure is as follows.

Suppose $V_1, \ldots, V_k$ is the complete set of "B" numbers which we have. If

$$W_1 = A_1 V_1 + \ldots + A_k V_k$$

where $A_1, \ldots, A_k$ are integers and $A_1 \neq 0$, then $W_1, V_2, \ldots, V_k$ is also a complete set.
In this program, $A_1, \ldots, A_k$ takes on the values $-2, -1, 0, 1, 2$ and the different $W_1$'s so obtained are compared and an optimal one is chosen. Then, $V_1 = V_k, V_2 = W_1, V_3 = V_2, \ldots, V_k = V_{k-1}$ are set. This process is repeated $k$-times. There are at present two choices of the optimizing procedure. The first choice is to maximize the number of zero entries and then minimize the sum of the absolute value, while the second is just the reverse. The two procedures do not appear to be equivalent.

After performing the foregoing operations on one formula, the program goes on and reads another formula.

Performance

The program has been tested on an IBM 7094. The running time for 28 formulas, each with 1, 2, 3, or 4 "B" numbers in its complete set, was about 2 minutes. The time for a formula with 5 "B" numbers in its complete set was slightly more than 4 minutes. This time includes an assembly listing and the punching of a binary deck. Without the listing and deck, a Univac 1108 took 42 seconds to do the same 28 formulas.

Electronics Research Center
National Aeronautics and Space Administration
Cambridge, Massachusetts, September 1968
129-06-02-75
REFERENCES


APPENDIX

PROGRAM LISTING

MAIN - EFN SOURCE STATEMENT - IFN(S) -

THIS PROGRAM PERFORMS THE FOLLOWING FUNCTION. GIVEN A
SET OF REFERENCE DIMENSIONS, A SET OF VARIABLES EXPRESSED
IN TERMS OF THOSE DIMENSIONS AND A FORMULA IN TERMS OF SOME
OF THE VARIABLES, THE PROGRAM FIRST DETERMINES A COMPLETE
(I.E., MAXIMALLY LINEARLY INDEPENDENT) SET OF B-VECTORS. THEN
IT EXAMINES OTHER COMPLETE SETS OF B-VECTORS AND DETERMINES
WHICH HAS THE MINIMAL SUM OF THE ABSOLUTE VALUES OF THE
ENTRIES AND WHICH HAS THE MAXIMAL NUMBER OF ZEROS IN THE
ENTRIES OF THE B-VECTORS.

THE INPUT CONSISTS OF
1. NREFD = THE NUMBER OF REFERENCE DIMENSIONS.
2. NVAR = THE NUMBER OF VARIABLES
3. IVAR = AN (NREFD) BY (NVAR) MATRIX WHICH ACTS AS A
   LIST(INDEXED BY THE SECOND VARIABLE) OF (NREFD) BY 1
   MATRICES. THE J' TH ENTRY IN THIS LIST, OR EQUIVALENTLY
   THE J' TH COLUMN OF IVAR DEFINES THE J' TH VARIABLE IN TERMS
   OF THE REFERENCE DIMENSIONS. FOR EXAMPLE IF THE J' TH
   VARIABLE WAS THE BOLTZMANN CONSTANT WHICH HAS DIMENSIONS
   (LENGTH) (LENGTH) (MASS) / (TIME) (TIME) (TEMPERATURE)
   AND IF THE REFERENCE DIMENSIONS WERE
   (LENGTH, MASS, TIME, CHARGE, TEMPERATURE) THEN
   IVAR(1,J)=2,
   IVAR(2,J)=1,
   IVAR(3,J)=2,
   IVAR(4,J)=0,
   IVAR(5,J)=1.
4. NREFD
5. N = THE NUMBER OF VARIABLES IN THE FORMULA OF INTEREST.
6. IPFORM = A LIST OF N NUMBERS WHICH INDICATE WHICH
   VARIABLES ARE IN THE FORMULA OF INTEREST. THE NUMBERS REFER
   TO THE SECOND SUBSCRIPT OF IVAR.
7. IOPT = 1 OR 2. IT INDICATES WHICH OPTIMIZATION PROCEDURE
   WILL BE FOLLOWED. 1 MEANS THAT ZEROS WILL BE MAXIMIZED
   FIRST AND THE SUM OF THE ABSOLUTE VALUE OF THE ENTRIES WILL
   BE MINIMIZED NEXT, WHILE 2 MEANS THAT THE PROCEDURE OCCURS
   IN THE REVERSE ORDER.

INPUT FORMAT
THE FIRST INPUT CARD CONTAINS NREFD AND NVAR IN
215 FORMAT. THE NEXT NVAR CARDS CONTAIN NREFD NUMBERS
ON EACH IN 1415 FORMAT AND FORM THE COLUMNS OF IVAR.
IF Q FORMULAS ARE TO BE ANALYZED, THEN THERE SHOULD BE Q SEQUENCES OF CARDS IN THE FOLLOWING FORM:

IOPT IN 15 FORMAT
M AND N IN 215 FORMAT
IFORM (= N NUMBERS) IN 1415 FORMAT

DIMENSION MULT(30), NULSAV(30,30), MATSZ(30,30), INMAT(30,30)
DIMENSION IV(15,15), NULL(30,30), IVAR(5,40), IFORM(15), MATSS(30,30)

101 FORMAT(215)
102 FORMAT(1415)
7000 FORMAT(10H THERE ARE, I2, 25H REFERENCE DIMENSIONS AND, I2, 11H VARIABLES.)
7001 FORMAT(34H THE ROWS BELOW ARE THE VARIABLES.)
7002 FORMAT(///70H THE FORMULA NOW BEING CONSIDERED IS DEFINED IN TERMS OF THE FOLLOWING, I2, 11H VARIABLES.///)
7004 FORMAT(///80H THE COLUMNS OF THE MATRIX BELOW ARE THE VARIABLES IN TERMS OF THE REFERENCE DIMENSIONS.)
READ (5,101) NREFDM,NVARIB
WRITE (6,7001)
DO 1900 J=1,NVARIB
    READ (5,102) (IVAR(I,J),I=1,NREFDM)
1900    WRITE (6,102) (IVAR(I,J),I=1,NREFDM)
130 FORMAT(15)
100 READ (5,130) IOPT
READ (5,101) M,N
WRITE (6,7002) N
READ (5,102) (IFORM(J),J=1,N)
WRITE (6,102) (IFORM(J),J=1,N)
DO 111 J=1,N
    IGN=IFORM(J)
111    DO 111 I=1,N
INMAT(I,J)=IVAR(I,IGN)
WRITE (6,7004)
1003 DO 7003 I=1,M
1003    WRITE (6,102) (INMAT(I,J),J=1,N)
7003    DO 1 IV(I,J)=0
1    DO 2 I2=1,N
2    IV(I2,12)=1

FIND A BASIS FOR THE NULL SPACE

3 CALL NULSPA(M,N,INMAT,IV,NULL,K)

THE ROWS OF NULL ARE B VECTORS AND FORM A COMPLETE SET.

K IS THE NUMBER OF B VECTORS IN A COMPLETE SET.

ISUMS IS THE SUM OF THE ABSOLUTE VALUE OF THE ENTRIES IN
MAIN - EFN SOURCE STATEMENT - IFN(S) -

C THE FINAL OPTIMIZED MATRIX, AND IZERS IS THE NUMBER OF ZEROS.

C ISUMS=0
IZERS=0

C ARE THERE ANY NON-ZERO B VECTORS

C IF(K-1).GE.620,601
200 WRITE (6,103)
103 FORMAT(38H THE ONLY B VECTOR IS THE ZERO VECTOR.)
GO TO 100

C IF THERE IS ONLY ONE B VECTOR THERE IS NO NEED TO GO
C THROUGH THE OPTIMIZATION ROUTINE.

C 1. 620 IZERS=0
IZERS=0

C CHECK TO SEE IF ANY OF THE NUMBERS 2, 3, 5, 7, 11 DIVIDES
C EVERY ENTRY OF A B VECTOR.

C CALL DIVCHK(1:N,NULL)
902 DO 622 J=1,N
ISUMS=ISUMS+IABS(NULL(1,J))
MATSZ(1,J)=NULL(1,J)
IF(NULL(1,J))622,621,622
621 IZERS=IZERS+1
622 CONTINUE
GO TO 307

C SAVE ORIGINAL COMPLETE SET OF B VECTORS IN MATRIX NULSAV.

C 801 KK=K-1
DO 9000 IXD=1,K
DO 9000 JXD=1,N
9000 NULSAV(IXD,JXD)=NULL(IXD,JXD)

C NOW GO THROUGH OPTIMIZATION.

C WE HAVE A COMPLETE (I.E., MAXIMAL LINEARLY INDEPENDENT)
C SET OF B VECTORS, SAY V(1), V(2), ..., V(K).

C IF W(1)=A0V(1)+B0V(2)+...+C0V(K), WHERE A0, B0, ..., C0 ARE
C INTEGERS AND A0 IS NOT ZERO, THEN W(1), V(2), ..., V(K) IS
C AGAIN A COMPLETE SET OF B VECTORS. IN THIS PROGRAM
C A0, B0, ..., C0 TAKE ON THE VALUES -2, -1, 0, 1, 2 AND THE DIFFERENT
C W(1)'S ARE OBTAINED IN THIS WAY. THEY ARE COMPARED AND
C AN OPTIMAL ONE IS CHOSEN. THEN WE SET V(1)=V(K), V(2)=W(1),
C V(3)=V(2), ..., V(K)=V(K-1). THIS PROCESS IS REPEATED K
C TIMES.

C MAJROW INDICATES HOW MANY TIMES WE HAVE GONE THROUGH THE
C ABOVE PROCESS.

C DO 3014 MAJROW=1,K
ISUMSA = 9999
IZERSA = -1
IUP = 2**KK

NOW WE ARE DETERMINING THE A*B, ..., C, WHICH WE CALL JZEROX,
MULT(1), ..., MULT(K-1), RESPECTIVELY.

DO 670 JZEROX = 1, 5
   IF (IZEROX = 3) 6650, 670, 650
650 JZEROX = IZEROX - 3
   DO 3006 INUMB1 = 1, IUP
      INUMB1 = INUMB1 - 1
   DO 3006 INUMB2 = 1, IUP
      INUMB2 = INUMB2 - 1
   DO 3006 INUMB3 = 1, IUP
      INUMB3 = INUMB3 - 1
   DO 350 JNUMB = 1, IUP
      ITEST = 2** (JNUMB - 1)
      MULT(JNUMB) = 0
      IF (AND(INUMB1, ITEST) .NE. 0) GO TO 612
560 GO TO 350
612 MULT(JNUMB) = 1
      IF (AND(INUMB2, ITEST) .NE. 0) MULT(JNUMB) = -MULT(JNUMB)
      IF (AND(INUMB3, ITEST) .NE. 0) MULT(JNUMB) = 2*MULT(JNUMB)
   350 CONTINUE
   DO 351 J = 1, N
      NULL(MAJROW, J) = JZEROX*NULSAV(I, J)
540 NULL(MAJROW,.*) IS THE NEW B VECTOR WHICH WE WILL EXAMINE.

DO 354 I = 2, K
   11 = I - 1
   352 DO 353 J = 1, N
      353 NULL(MAJROW, J) = NULL(MAJROW, J) + (MULT(I1)*NULSAV(I, J))
   354 CONTINUE

CHECK TO SEE IF ANY OF THE NUMBERS 2, 3, 5, 7, 11 DIVIDES
EVERY ENTRY OF A B VECTOR.

7 CALL DIVCHK(MAJROW, N, NULL)

CHECK OPTIMIZING CONDITIONS, DEPENDING ON IOPT, EITHER
FIRST DETERMINE THE NUMBER OF ZEROS ENTRIES AND THEN SUM THE
ABSOLUTE VALUES OF THE ENTRIES OR VICE VERSA.

IZERO = 0
   ISUM = 0
   DO 9 JS = 1, N
      ISUM = ISUM + IABS(NULL(MAJROW, JS))
   IF (NULL(MAJROW, JS)) 9, 8, 9
9 IZERO = IZERO + 1
MAIN - EFN SOURCE STATEMENT - IFN(S) -

9 CONTINUE
20 GO TO (723, 724) I0PT
723 IF (IZERO = IZERSA) 3006, 6001, 3003
6001 IF (ISUM = ISUMSA) 3003, 3006, 3006
724 IF (ISUM = ISUMSA) 3003, 6002, 3006
6002 IF (IZERO = IZERSA) 3006, 3006, 3003
3003 IZERSA = IZERO
ISUMSA = ISUM
DO 3004 IK4 = 1, N
3004 MATSZ(MAJROW, IK4) = NULL(MAJROW, IK4)
C C
C MATSZ IS THE MOST OPTIMAL NEW B VECTOR YET EXAMINED.
C C
3006 CONTINUE
670 CONTINUE
20 DO 760 J = 1, N
760 NULSAV(I, J) = NULSAV(K, J)
DO 761 I = 2, K
761 MATSS(I, J) = NULSAV(I, J)
DO 762 I = 2, KK
762 I2 = I + 1
763 NULSAV(I2, J) = MATSS(I, J)
DO 763 J = 1, N
3014 CONTINUE
C C
WE ARE DONE TESTING, PRINT ANSWER
C C
307 WRITE (6, 104) K
104 FORMAT (10H THERE ARE, I3, 32H LINEARLY INDEPENDENT B VECTORS.)
20 GO TO (131, 132) I0PT
131 WRITE (6, 131)
133 FORMAT (44H WE HAVE OPTIMIZED ZEROS FIRST, THEN THE SUM.)
20 GO TO 4006
132 WRITE (6, 132)
134 FORMAT (44H WE HAVE OPTIMIZED THE SUM FIRST, THEN ZEROS.)
4006 WRITE (6, 107)
107 FORMAT (63H THE ROWS OF THE MATRIX BELOW FORM A COMPLETE SET OF B VECTORS.)
WRITE (6, 120) ISUMS, IZERS
120 FORMAT (49H THE SUM OF THE ABSOLUTE VALUES OF THE ENTRIES IS, I3, 34H 1 AND THE NUMBER OF ZERO ENTRIES IS, I3)
20 DO 315 I9 = 1, K
315 WRITE (6, 106) (MATSZ(I9, J9), J9 = 1, N)
106 FORMAT (15I5)
20 GO TO 100
END
SUBROUTINE BASE(M,N,INMAT,MATOUT,K) 
DIMENSION INMAT(30,30),MATOUT(30,30),ICONS2(15),MATSAV(30,30)
C
SUBROUTINE BASE HAS THE FOLLOWING FUNCTION: GIVEM A VECTOR 
SPACE V WITH BASIS(V1,V2,...,VN) AND A SET OF VECTORS IN V (W1,W2, 
...,WN) BASE DETERMINES 1.K=THE DIMENSION OF W WHERE W IS THE 
SUBSPACE GENERATED BY THE WI'S, AND 
2.2K VECTORS IN W IN TERMS OF THE ORIGINAL 
WI'S, WHICH ARE A BASIS FOR W. 
THE INPUT CONSISTS OF- 1.N=THE DIMENSION OF V 
2.M=THE NUMBER OF VECTORS IN THE GIVEN 
SET, AND 
3.INMAT=THE M BY N MATRIX WHICH 
EXpresses THE WI'S IN TERMS OF THE VJ'S. 
THE OUTPUT CONSISTS OF- 1.K=THE DIMENSION OF W AND 
2.MATOUT=THE M BY M MATRIX WHOSE 
FIRST K ROWS DETERMINES THE BASIS OF W IN TERMS OF THE VECTORS IN 
THE GIVEN SET. 
NOTE THE INPUT MATRIX MUST HAVE INTEGRAL ENTRIES. 
C
DO 61 I=1,M
DO 61 J=1,M
MATSAV(I,J)=INMAT(I,J)

IF(I-J).GT.0 .OR. (I-J).LT.1 .OR. MATOUT(I,J).EQ.0 
GO TO 61
60 MATOUT(I,J)=1
61 CONTINUE

IROW=1

C
DO 50 J=1,N
DO 50 I=1,M
IF(INMAT(I,J)).LE.0 .OR. (I-M).LT.0 .OR. (I-M).GT.30 .OR. 
51 JSTART=J

30 ISTART=I

GO TO 31
50 CONTINUE

K=0
RETURN
31 IPLUS=ISTART+1
DO 32 I=IPLUS,M
DO 32 J=1,N
IF(INMAT(I,J)).LT.0 .OR. (I-M).LT.0 .OR. (I-M).GT.30 .OR. 
32 CONTINUE

33 K=1
RETURN
52 DO 15 J=JSTART,N
DO 3 I=IROW,M
IF (INMAT(I,J)).LT.0 .OR. (I-M).LT.0 .OR. (I-M).GT.30 .OR. 
2 ISWOP=1
GO TO 4
3 CONTINUE
GO TO 15
C
INTERCHANGE ROWS OF INMAT.
C
4 DO 5 J1=1,N
SAVE=INMAT(IROW,J1)
INMAT(IROW,J1)=INMAT(ISWOP,J1)
5 INMAT(ISWOP,J1)=SAVE
C
INTERCHANGE CORRESPONDING ROWS OF MATOUT.
C
DO 6 I1=1,M
SAVE=MATOUT(IROW,I1)
MATOUT(IROW,I1)=MATOUT(ISWOP,I1)
6 MATOUT(ISWOP,I1)=SAVE
9 IROW1=IROW+1
ICONS1=INMAT(IROW,J)
C
ROW REDUCE INMAT.
C
CHANGE MATOUT ACCORDINGLY.
C
DO 66 I3=IROW1,M
ICONS2(I3)=INMAT(I3,J)
IF(ICONS2(I3))=65,66,65
65 DO 10 J3=1,N
10 INMAT(I3+J3)=(ICONS1*INMAT(I3+J3))-(ICONS2(I3)*INMAT(IROW+J3))
66 CONTINUE
DO 68 I4=IROW1,M
IF(ICONS2(I4))=67,68,67
67 DO 13 J4=1,N
13 MATOUT(I4+J4)=(ICONS1*MATOUT(I4+J4))-(ICONS2(I4)*MATOUT(IROW+J4))
68 CONTINUE
IF(J-J4)=40,19,19
40 IF(IROW1-M)=41,16,19
41 JONE=J1
DO 42 JTRY=JONE,N
DO 42 ITRY=IROW1,M
IF(INMAT(ITRY+JTRY))=14,42,14
42 CONTINUE
GO TO 19
14 IROW=IROW+1
15 CONTINUE
GO TO 19
16 DO 17 I5=1,N
IF(INMAT(IROW1+I5))=18,17,18
17 CONTINUE
19 K=IROW
GO TO 71
18 K=IROW1
71 DO 70 I6=1,M
DO 70 I7=1,N
70 INMAT(I6+I7)=MATSAV(I6+I7)
RETURN
END
S2 - EFN SOURCE STATEMENT - IFN(S) - 07/

SUBROUTINE NULSPA(M,N,L,IV,NULL,K)

C WE ARE GIVEN TWO VECTOR SPACES V AND W WITH BASIS V(1)*...*
V(N) AND W(1)*...*W(M) RESPECTIVELY AND A LINEAR TRANSFORMA-
C TION L FROM V INTO W. SUBROUTINE NULSPA DETERMINES A BASIS FOR
C THE NULL SPACE OF L IN TERMS OF A GIVEN BASIS V.
C V(1)*...*V(N) IS THE USUAL BASIS, NAMELY V(I) = (0,...,0*
C 1,0,...,0) WHERE THE 1 OCCURS IN THE I'TH LOCATION.

C THE INPUT CONSISTS OF-
C 1. TWO INTEGERS M AND N.
C 2. AN N BY N MATRIX IV WHOSE COLUMNS ARE A BASIS OF V, AND
C 3. AN M BY N MATRIX WHICH CORRESPONDS TO L IN THE GIVEN BASIS.

C THE OUTPUT CONSISTS OF-
C 2. NULL = A K BY N MATRIX WHOSE J'TH ROW (J1*...*JN) REPRESENTS
   THE J'TH VECTOR = J1*V(1) +...+JN*V(N) OF THE BASIS OF THE
   NULL SPACE.

   THE METHOD.
   LET V(I) BE THE I'TH COLUMN VECTOR OF IV. STEP 1
   CONSISTS OF FORMING THE MATRIX IRANGE WHOSE I'TH ROW CONSIS
   TS OF L(V(I)). STEP 2 CONSISTS OF CALLING SUBROUTINE BASE
   WHICH REDUCES IRANGE. IT ALSO DETERMINES THE ROWS OF
   THE ROW REDUCED MATRIX IN TERMS OF THE INITIAL ROWS OF THE
   MATRIX AND EXPRESSES THE RESULT IN THE MATRIX IBSRAN.
   IDMRAN IS THE DIMENSION OF THE RANGE OF L. THE
   FIRST IDMRAN ROWS OF IBSRAN FORM A BASIS OF THE RANGE OF L
   AS FOLLOWS.
   WE HAVE V(1)*...*V(N) AS A BASIS OF V AND WE FORMED
   L(V(I)) FOR I=1,...,N. SUPPOSE IBSRAN=(A(I*J)), THEN THE
   I'TH ELEMENT OF A BASIS FOR THE RANGE OF L IS GIVEN BY
   A(I,*1)*L(V(1)) +...+A(I,*N)*L(V(N)). SINCE L IS LINEAR IT
   FOLLOWS THAT THE VECTORS A(I,*1)*L(V(1)) +...+A(I,*N)*L(V(N))
   FOR I=1,...,IDMRAN FORM A BASIS FOR A SUBSPACE OF V WHICH IS
   ISOMORPHIC, UNDER THE RESTRICTION OF L TO THIS SUBSPACE,
   TO THE RANGE OF L. THE LAST N-IDMRAN ROWS OF THE ROW
   REDUCED IRANGE MATRIX ARE ALL ZERO. SO A(I,*1)*L(V(1)) +...+
   A(I,*N)*L(V(N)) = 0 FOR I=IDMRAN+1,...,N. THE LINEARITY OF L
   SHOWS THAT THE VECTORS A(I,1)*V(1) +...+A(I,N)*V(N) FOR I=
   IDMRAN+1,...,N ARE IN THE NULL SPACE OF L. THE ROW
   REDUCTION PROCESS SHOWS THAT THESE VECTORS ARE LINEARLY
   INDEPENDENT. HENCE THE LAST N-IDMRAN ROWS OF IBSRAN ARE A
   BASIS FOR THE NULL SPACE OF L.

DIMENSION L(30,30),NULL(30,30)
DIMENSION IV(15,15)
S2 - EFN SOURCE STATEMENT - IFN(S) -

DIMENSION IRANGE(30,30), IBSRAN(30,30)
DO 1 I1=1,M
DO 1 J1=1,N
1 IRANGE(J1+I1)=0

C STEP 1
DO 7 I2=1,M
DO 7 J2=1,N
DO 7 J3=1,N
7 IRANGE(J2+I2)=IRANGE(J2+I2)+(I2+J3)*IV(J3+J2)

C STEP 2
CALL BASE(N,M,IRANGE,IBSRAN,IDMRAN)
K=I-IDMRAN
DO 6 I2=1,K
I3=IDMRAN+I2
DO 6 J2=1,N
6 NULL(I2+J2)=IBSRAN(I3,J2)
RETURN
END
SUBROUTINE DIVCHK(L2,N,NULL)
DIMENSION NULL(30,30)

C DIVCHK CHECKS TO SEE IF THE NUMBERS 2,3,5,7,11 DIVIDE EVERY ENTRY
C OF ROW L2 IN THE MATRIX NULL WHICH HAS 'N' COLUMNS. IF SO THEN
C THE DIVISION IS DONE.
C
ICNT=1
732 GO TO (726,727,728,729,730) ICNT
726 L3=2
  60 GO TO 731
727 L3=3
  60 GO TO 731
728 L3=5
  60 GO TO 731
729 L3=7
  60 GO TO 731
730 L3=11
731 DO 720 L4=1,N
  ICHECK=MOD(NULL(L2,L4),L3)
  IF(IKEYCH)722,720,722
720 CONTINUE
  DO 721 L5=1,N
  NULL(L2,L5)=NULL(L2,L5)/L3
  60 GO TO 732
721 IF(ICNT-5)733,802,802
733 ICNT=ICNT+1
  60 GO TO 732
602 RETURN
END
1961


1962


1963


1964


1965


1966


1967


1968


"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.

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 conference proceedings, 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 Notes, and Technology Surveys.

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

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546