NASA CR 108499

Contractor Report

GENHYP — A FORTRAN V
PROGRAM FOR GENERAL LINEAR
HYPOTHESIS TESTING

By Fred C. Delaney and Fred M. Speed

Prepared under Contract No. NAS 9-5384 by
HOUSTON AEROSPACE SYSTEMS DIVISION
LOCKHEED ELECTRONICS COMPANY
Houston, Texas

For

Manned Spacecraft Center
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
May 1970

NASA CR 108499

ABSTRACT

GENHYP is a general, linear, hypothesis-testing program which computes sum-of-squares for linear models with or without missing cells and for the balanced or unbalanced case. It is easy to use and will compute sum-of-squares for linear models of the form

$$Y = Xu + e$$

subject to $\theta^T u = 0$ where the usual assumptions are made about e .

This program is described and numerical examples of various linear models are included in an effort to simplify usage of the program.

CONTENTS

Section Page

GENHYP — A FORTRAN V PROGRAM FOR GENERAL LINEAR HYPOTHESIS TESTING

By Fred C. Delaney and Fred M. Speed*

INTRODUCTION

The program provides a method of analysis that makes use of the simple structure of the observations. It is assumed that each observation comes from a normal population, where each population has a mean and a common variance. Certain restrictions known about the means may be input into the model. Now, while each population may have been sampled a different number of times, only those populations from which at least one sample was taken shall be included in the model. These observations can be expressed as

$$Y_{ijk} = u_{ij} + e_{ijk}$$

where the means satisfy the relations

$$\theta_\ell^T u = \xi_\ell \qquad (\ell = 1,\cdots,r)$$

and Y_{ijk} is the k*th* observation from the (ij)*th* population.

u is the vector of the u_{ij} in some order.

u_{ij} is the mean of the (ij)*th* population.

---

*Associate Professor of Mathematics, Texas A&I College, Kingsville, Texas.

e_{ijk} are normal random variables such that $E(e_{ijk}) = 0$, $V(e_{ijk}) = \sigma^2$.

θ_ℓ^T is the ℓth restriction on the u_{ij}'s.

ξ_ℓ is known.

Naturally, the number of the subscripts is essentially unlimited; however, to simplify the notation, the number of subscripts will be kept to three. In order to better understand the concepts in this section, consider the following example.

Suppose that there are six populations indexed by two subscripts (ij), that is, (11, 12, 21, 22, 31, 32). Also, suppose that the following number of observations came from each population; two from population 1, two from population 2, five from population 3, three from population 4, one from population 5, and five from population 6. Let us further suppose that

$$u_{11} - u_{12} - u_{21} + u_{22} = 0$$

$$u_{11} - u_{12} - u_{31} + u_{32} = 0$$

where u_{ij} is the mean of the (ij)*th* population. The model is

$$Y_{ijk} = u_{ij} + e_{ijk}$$

subject to $u_{ij} - u_{i'j} - u_{ij'} + u_{i'j'} = 0$ for all i, i', j, j' where $i = 1,2,3$, $j = 1,2$, $k = 1,\cdots n_{ij}$.

In order to simplify the presentation, the general model can be expressed in matrix notation. It is

$$Y = Xu + e$$

subject to $\theta^T u = \xi$

where

y is the n×1 vector of observations.

u is the p×1 vector of cell means.

X is the n×p design matrix.

e is a normal random variable such that $E[e] = 0$.

$E[ee^T] = \sigma^2 I$.

θ^T is an r×p matrix of rank r that represents the restrictions known about the means.

ξ is an r×1 known vector.

The example expressed in matrix notation is

$$Y = Xu + e$$

subject to $\theta^T u = 0$

where

Y is an 18×1 vector,

X is an 18×6 matrix,

u is a 6×1 vector,

e is an 18×1 vector,

θ^T is a 2×6 matrix,

ξ is a 2×1 vector,

and where

$$\begin{bmatrix} Y_{111} \\ Y_{112} \\ Y_{121} \\ Y_{122} \\ Y_{211} \\ Y_{212} \\ Y_{213} \\ Y_{214} \\ Y_{215} \\ Y_{221} \\ Y_{222} \\ Y_{223} \\ Y_{311} \\ Y_{321} \\ Y_{322} \\ Y_{323} \\ Y_{324} \\ Y_{325} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{11} \\ u_{12} \\ u_{21} \\ u_{22} \\ u_{31} \\ u_{32} \end{bmatrix} + e$$

subject to $\begin{bmatrix} 1 & -1 & -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 \end{bmatrix} u = 0$.

Because of the formulation of the model, X^TX will always be diagonal with diagonal elements equal to the number of observations from each population. One statistic that will be encountered frequently is $u^* = (X^TX)^{-1}X^TY$. This is nothing more than the vector of cell means. In the example,

$$(X^TX) = \text{Diag}[2,2,5,3,1,5]$$

and

$$(u^*)^T = [Y_{11\cdot}/2, Y_{12\cdot}/2, Y_{21\cdot}/5, Y_{22\cdot}/3, Y_{31\cdot}/1, Y_{32\cdot}/5] \ .$$

This program avoids using special formulas when there are missing cells. There is no need to have one program for the balanced case and another for the unbalanced case. The classical linear models may be considered in the manner that will be described in this paper.

DEFINITIONS AND SYMBOLS

Y	vector of observations
θ	constraint matrix
X	design matrix
λ_i	*ith* hypothesis matrix
SSQ_i	sum-of-squares associated with the *ith* hypothesis matrix

β	transformed constraint matrix
Ω	transformed hypothesis matrix
B,F	orthonormal basis matrices for θ and λ_i , respectively
D,Q	diagonal matrices which contain eigenvalues as entries
E,S	matrices of eigenvectors, constructed columnwise
u^*	vector of cell means
$\hat{u}$	vector of the minimum-variance unbiased estimate of the cell means
C	covariance matrix for $\hat{u}$
R	residual sum of squares
$\hat{\sigma}^2$	minimum-variance unbiased estimate of the population variance
NL	number of observations
N	number of populations (cells)
LX	number of rows in θ^T
H,A,P	matrices used to express computations
u	population mean
e	normal random error variables such that $E[e] = 0$, $V[e] = \sigma^2$

METHOD

The procedure for computing the sum-of-squares is presented in two cases. Case 1 is a linear model that has constraints (i.e., no interactions). Case 2 is a linear model with no constraints.

Case 1

The columns of the constraint matrix θ are considered as a basis for a real vector space. An orthonormal basis B composed of unit vectors which are mutually orthogonal is computed. The matrix H is calculated where

$$H = B^T(X^TX)B \; . \qquad (1)$$

The diagonal matrix D of eigenvalues and the matrix E of eigenvectors are computed from the matrix H . Next, the transformed constraint matrix denoted by β is calculated where

$$\beta = BH^T \; . \qquad (2)$$

The following computations are made:

$$u^* = (X^TX)^{-1}X^TY \qquad (3)$$

$$A = [I - (X^TX)^{-1}\theta D^{-1}\theta^T] \qquad (4)$$

$$C = A(X^TX)^{-1} \qquad (5)$$

$$\hat{u} = Au^* \qquad (6)$$

$$R_1 = [\theta^Tu^*]^T[\theta^Tu^*] \qquad (7)$$

$$R = Y^TY - (X^TX)^T(X^TX)^{-1}(X^TY) + R_1 \qquad (8)$$

$$\hat{\sigma}^2 = R/(NL - N - LX) \; . \qquad (9)$$

For each hypothesis matrix λ_i , an orthonormal basis F composed of unit vectors which are mutually orthogonal is computed. The matrix P is then constructed where

$$P = F^T CF . \tag{10}$$

The diagonal matrix Q of eigenvalues and the matrix S of eigenvectors are computed from P . The transformed hypothesis matrix, denoted by Ω , is calculated where

$$\Omega = FS^T . \tag{11}$$

Finally, the sum-of-squares due to the i*th* hypothesis matrix is computed,

$$SSQ_i = [\Omega^T \hat{u}]^T Q^{-1} [\Omega^T \hat{u}] . \tag{12}$$

Case 2

The minimum-variance unbiased estimate of the cell means is computed,

$$\hat{u} = u^* = (X^T X)^{-T} X^T Y . \tag{13}$$

Since R_1 is equal to zero in this case, equation (8) becomes

$$R = Y^T Y - (X^T X)^T (X^T X)^{-1} (X^T Y) \tag{14}$$

and $\hat{\sigma}^2$ is calculated,

$$\hat{\sigma}^2 = R/(NL - N - LX) . \tag{15}$$

For each hypothesis matrix λ_i , an orthonormal basis F composed of unit vectors which are mutually orthogonal is computed. The matrix P is then constructed, using

$$P = F^T(X^TX)^{-1}F . \tag{16}$$

The diagonal matrix Q of eigenvalues and the matrix S of eigenvectors are computed from P . The transformed hypothesis matrix, denoted by Ω , is calculated where

$$\Omega = FS^T . \tag{17}$$

Finally, the sum-of-squares due to the *ith* hypothesis matrix is computed,

$$SSQ_i = [\Omega^T\hat{u}]^TQ^{-1}[\Omega^T\hat{u}] . \tag{18}$$

For a more theoretical treatment of the method used by this program, see Speed (ref. 1).

PROGRAM USAGE

In order to use this subroutine, the following FORTRAN V calling sequence must be used:

```
CALL GENHYP(Y,TL,KM,M,XL,G,W,E,X,Z,US,UH,SSQ,R,VH,NL,N,NF,
          IND2)
```

where

Y is a one-dimensional single-precision array, provided by the user, that contains the observations. This array should be contructed so that each cell is exhausted before going to the next one. Y must be dimensioned by NL in the calling program.

TL is a three-dimensional single-precision array, provided by the user, that contains the hypothesis matrices and a constraint matrix if the user requires constraints. Consider TL(K,I,J) where K denotes the K*th* hypothesis matrix and I and J denote the I*th* row and J*th* column of the K*th* hypothesis matrix. If a constraint matrix is required, it should be stored (in TL) following the hypothesis matrices. That is, if NH denotes the number of hypothesis matrices, then the TL(NH+1,I,J) would denote the I*th* row and J*th* column of the constraint matrix. TL must be dimensioned NF by N by N in the calling program.

KM	is a one-dimensional integer array provided by the user, that contains the number of rows for corresponding hypothesis matrices in the TL array. KM(K) denotes the number of rows associated with the K*th* hypothesis matrix. Note that KM(K) also denotes the number of degrees-of-freedom associated with the K*th* hypothesis. KM must be dimensioned by NF in the calling program.
M	is a one-dimensional integer array provided by the user, that contains the number of observations per cell (or population). M(I) contains the number of observations in the I*th* cell. M must be dimensioned by N in the calling program.
XL	is a two-dimensional double-precision array used for temporary working storage. XL must be dimensioned N by N in the calling program.
G	is a two-dimensional double-precision array, that will contain the covariance matrix for $\hat{u}$ upon return to the calling program. G requires no input, but must be dimensioned N by N in the calling program.
W	is a two-dimensional double-precision array used for temporary working storage. W must be dimensioned N by N in the calling program.

E	is a two-dimensional double-precision array used for temporary working storage. E must be dimensioned N by N in the calling program.
X	is a one-dimensional double-precision array used for temporary working storage. X must be dimensioned by N in the calling program.
Z	is a one-dimensional double-precision array used as temporary working storage. Z must be dimensioned by N in the calling program.
US	is a one-dimensional single-precision array that will contain the cell (or population) means upon return to the calling program. US(I) contains the mean for the I*th* cell. US requires no input but must be dimensioned by N in the calling program.
UH	is a one-dimensional single-precision array that will contain the minimum-variance unbiased estimate of the call (or population) means upon return to the calling program. UH(I) contains the minimum-variance unbiased estimate of the I*th* cell. UH requires no input but must be dimensioned by N in the calling program.
SSQ	is a one-dimensional single-precision array that will contain the sum-of-squares upon return to the calling program. SSQ(K) will contain the

sum-of-squares associated with the *Kth* hypothesis matrix. SSQ requires no input but must be dimensioned by NF in the calling program.

R is a double-precision number provided by the user. It is used to determine

1. The occurrence of a zero vector during the Gram-Schmidt orthogonalization process
2. The convergence criterion for eigenvalue and eigenvector computations.

The number 10^{-6} was found to be sufficient in all test cases.

VH is a single-precision location that will contain the minimum-variance unbiased estimate of the population variance upon return to the calling program.

NL is an integer provided by the user, that denotes the total number of observations.

N is an integer provided by the user, that denotes the number of cells (populations).

NF is an integer provided by the user, that denotes the number of hypothesis matrices if there is no constraint matrix. If there is a constraint matrix, NF is equal to the number of hypothesis matrices plus one.

$$\text{IND2} \begin{cases} = 1, & \text{if there is no constraint matrix.} \\ \neq 1, & \text{if there is a constraint matrix.} \end{cases}$$

Let n_i and N denote the number of observations in the *ith* population and the number of populations respectively. The arrays Y and M must be constructed in the following manner:

- $Y(1)$ through $Y(n_1)$ contain the observations for the first cell, and $M(1) = n_1$.
- $Y(n_1+1)$ through $Y(n_1+n_2)$ contain the observations for the second cell, and $M(2) = n_2$.

  •
  •
  •
- $Y(n_{N-1})$ through $Y(n_{N-1}+n_N)$ contain the observations for the *Nth* cell, and $M(N) = n_N$.

The first subscript K of the TL array denotes the *Kth* hypothesis. It should be noted that the sum-of-squares and the number of degrees-of-freedom associated with the *Kth* hypothesis are SSQ(K) and KM(K) respectively.

SUBPROGRAMS

The following subprograms are used directly by the main program, GENHYP, or are useful for analysis of linear models:

1. TESTH computes the sum-of-squares.

2. UHAT2 computes u^*, $\hat{u}$, C (covariance matrix for $\hat{u}$), and $\hat{\sigma}^2$.

3. DOURHG (ref. 2) computes an orthonormal basis for a basis of a real vector space of finite dimension. The computed orthonormal basis is composed of unit vectors which are mutually orthogonal.

4. DJACOB (ref. 3) computes eigenvalues and eigenvectors, given a symmetric matrix comprised of real elements.

5. EIGOR2 "transforms" the constraint matrix.

6. FISHIN (ref. 4) is not used by GENHYP, but it is an asset in the analysis of linear models. It computes the inverse F value (Fisher's F distribution) when given the confidence coefficient and number of degrees-of-freedom.

7. FISH (ref. 4) is not used by GENHYP, but it is helpful in the analysis of linear models. It computes an approximation of Fisher's F distribution.

8. PHINV (ref. 4) is not used by GENHYP. It is helpful in the analysis of linear models. Given the probability, it computes the inverse of the normal distribution with a mean of zero and a variance of one.

NUMERICAL EXAMPLES

The examples in this section will be discussed in terms of the calling arguments of GENHYP. The arguments, which are used as temporary working storage, are in the calling sequence for dimensioning purposes only and will not be included in the treatment of the examples.

The One-Way Classification

This example is taken from Ostle (ref. 5), page 287.

Storage Conditions				
1	2	3	4	5
7.3	5.4	8.1	7.9	7.1
8.3	7.4	6.4	9.5	
7.6	7.1		10.0	
8.4				
8.3				

The model is

$$Y_{ij} = u_i + e_{ij} .$$

The hypothesis to be tested is

$$H_0 : u_1 = u_2 = u_3 = u_4 = u_5 .$$

Input:

$$Y = (7.3, 8.3, 7.6, 8.4, 8.3, 5.4, 7.4, 7.1, 8.1, 6.4, 7.9, 9.5, 10.0, 7.1)^T$$

$$TL_1 = \begin{bmatrix} 1. & -1. & 0. & 0. & 0. \\ 1. & 0. & -1. & 0. & 0. \\ 1. & 0. & 0. & -1. & 0. \\ 1. & 0. & 0. & 0. & -1. \end{bmatrix}^T$$

$KM = (4)^T$

$M = (5,3,2,3,1)^T$

$R = 10^{-6}$

NL = 14

N = 5

NF = 1

IND2 = 1

Output:

$UH = US = (8.0,6.6,7.3,9.1,7.1)^T$

$SSQ_1 = 10.66$

VH = .7963

ANALYSIS OF VARIANCE

Source of Variation	Degrees-of-Freedom	Sum-of-Squares	Mean Square	F Statistic	95% Critical Value
Treatments	4	10.662238	2.6655595	3.3475939	3.6330883
Experimental Error	9		.79626129		

The One-Way Classification with Subsampling

Consider the example taken from Ostle (ref. 5), page 292.

Determinations	Micro-Organism No. 1 Sample Number				Micro-Organism No. 2 Sample Number		
	1	2	3	4	1	2	3
1	5.6	5.0	5.4	5.3	7.6	7.4	7.5
2	5.7	5.0	5.4	5.5	7.6	7.0	7.6
3		5.1	5.4		7.8	7.2	7.5
4			5.5				7.4
5			5.4				

It is believed that the hypothesis which will test any significant difference in treatments is

$$H_0: \frac{u_1 + u_2 + u_3 + u_4}{4} = \frac{u_5 + u_6 + u_7}{3} .$$

Input:

Y = $(5.6,5.7,5.0,5.0,5.1,5.4,5.4,5.4,5.5,5.4,5.3,5.5,7.6,7.6,7.8,7.4,7.0,7.2,7.5,7.6,7.5,7.4)^T$

TL_1 = $(3.,3.,3.,3.,-4.,-4.,-4.,)^T$

$$TL_2 = \begin{bmatrix} 1. & -1. & 0. & 0. & 0. & 0. & 0. \\ 1. & 0. & -1. & 0. & 0. & 0. & 0. \\ 1. & 0. & 0. & -1. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. & -1. & 0. \\ 0. & 0. & 0. & 0. & 1. & 0. & -1. \end{bmatrix}^T$$

KM = $(1,5)^T$

M = $(2,3,5,2,3,3,4)^T$

R = 10^{-6}

NL = 22

N = 7

NF = 2

IND2 = 1

It should be noted that the second hypothesis matrix TL_2 is used to determine the sum-of-squares for experimental error.

Output:

$UH = US = (5.65, 5.03, 5.42, 5.4, 7.66, 7.1, 7.5)^T$

$SSQ_1 = 24.09$

$SSQ_2 = .8468$

$VH = .0111$

ANALYSIS OF VARIANCE

Source of Variation	Degrees-of-Freedom	Sum-of-Squares	Mean Square	F Statistic	95% Critical Value
Treatments	1	21.879458	21.879458	1972.9414	4.5430791
Experimental Error	5	.84683336	.16936667	15.272340	2.9012953
Sampling Error	15		.01108976		

The Two-Way Classification Without Interaction (Randomized Complete Block Design)

Consider the example taken from Ostle (ref. 5), page 375.

Day	Machine			
	A	B	C	D
1	293	308	323	333
2	298	353	343	363
3	280	323	350	368
4	288	358	365	345
5	260	343	340	330

The model is

$$Y_{ijk} = u_{ij} + e_{ijk}$$

where $i = 1,\cdots,5$; $j = 1,\cdots,4$; $k = 1$, and the constraints are as follows:

$$u_{11} - u_{21} - u_{12} + u_{22} = 0$$

$$u_{11} - u_{21} - u_{13} + u_{23} = 0$$

$$u_{11} - u_{21} - u_{14} + u_{24} = 0$$

$$u_{11} - u_{31} - u_{12} + u_{32} = 0$$

$$u_{11} - u_{31} - u_{13} + u_{33} = 0$$

$$u_{11} - u_{31} - u_{14} + u_{34} = 0$$

$$u_{11} - u_{41} - u_{12} + u_{42} = 0$$

$$u_{11} - u_{41} - u_{13} + u_{43} = 0$$

$$u_{11} - u_{41} - u_{14} + u_{44} = 0$$

$$u_{11} - u_{51} - u_{12} + u_{52} = 0$$

$$u_{11} - u_{51} - u_{13} + u_{53} = 0$$

$$u_{11} - u_{51} - u_{14} + u_{54} = 0$$

The hypotheses are

$$H_0: \quad u_{1\cdot} = u_{2\cdot} = u_{3\cdot} = u_{4\cdot} = u_{5\cdot}$$

and

$$H_0: \quad u_{\cdot 1} = u_{\cdot 2} = u_{\cdot 3} = u_{\cdot 4} \;.$$

<u>Input</u>:

$$Y = (293.,298.,280.,288.,260.,308.,353.,323.,358.,343., \\ 323.,343.,350.,365.,340.,333.,363.,368.,345.,330.)^T$$

$$TL_1 = \left[\begin{array}{cccccccccccccccccccc}
1. & -1. & 0. & 0. & 0. & 1. & -1. & 0. & 0. & 0. & 1. & -1. & 0. & 0. & 0. & 1. & -1. & 0. & 0. & 0. \\
1. & 0. & -1. & 0. & 0. & 1. & 0. & -1. & 0. & 0. & 1. & 0. & -1. & 0. & 0. & 1. & 0. & -1. & 0. & 0. \\
1. & 0. & 0. & -1. & 0. & 1. & 0. & 0. & -1. & 0. & 1. & 0. & 0. & -1. & 0. & 1. & 0. & 0. & -1. & 0. \\
1. & 0. & 0. & 0. & -1. & 1. & 0. & 0. & 0. & -1. & 1. & 0. & 0. & 0. & -1. & 1. & 0. & 0. & 0. & -1.
\end{array}\right]^T$$

$$TL_2 = \left[\begin{array}{cccccccccccccccccccc}
1. & 1. & 1. & 1. & 1. & -1. & -1. & -1. & -1. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
1. & 1. & 1. & 1. & 1. & 0. & 0. & 0. & 0. & 0. & -1. & -1. & -1. & -1. & -1. & 0. & 0. & 0. & 0. & 0. \\
1. & 1. & 1. & 1. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & -1. & -1. & -1. & -1.
\end{array}\right]^T$$

$$TL_3 = \left[\begin{array}{cccccccccccccccccccc}
1. & -1. & 0. & 0. & 0. & -1. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
1. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
1. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & 1. & 0. & 0. & 0. \\
1. & 0. & -1. & 0. & 0. & -1. & 0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
1. & 0. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & 0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
1. & 0. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & 0. & 1. & 0. & 0. \\
1. & 0. & 0. & -1. & 0. & -1. & 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
1. & 0. & 0. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. \\
1. & 0. & 0. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & 0. & 0. & 1. & 0. \\
1. & 0. & 0. & 0. & -1. & -1. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
1. & 0. & 0. & 0. & -1. & 0. & 0. & 0. & 0. & 0. & -1. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. \\
1. & 0. & 0. & 0. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & 0. & 0. & 0. & 1.
\end{array}\right]^T$$

KM = $(4,3,12)^T$

M = $(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)^T$

R = 10^{-6}

NL = 20

N = 20

NF = 3

IND2 = 0

Output:

SSQ_1 = 2146.2001

SSQ_2 = 13444.804

VH = 218.84999

ANALYSIS OF VARIANCE

Source of Variation	Degrees-of-Freedom	Sum-of-Squares	Mean Square	F Statistic	95% Critical Value
Days	4	2146.2001	536.55003	2.4516794	3.2591666
Machine	3	13444.804	4481.6014	20.477960	3.4902959
Experimental Error	12		218.84999		

The Two-Way Classification Without Interaction

This example was taken from Harvey (ref. 6).

Sire No.	Ration No.					
	1			2		
1		5 6			2 3	
2	2 3	5 6	7	8 8		9
3		3		4 4	6 6	7

The model is

$$Y_{ijk} = u_{ij} + e_{ijk}$$

where $i = 1,\cdots,3$; $j = 1,2$; $k = 1,\cdots n_{ij}$ and the constraints are as follows:

$$u_{11} - u_{21} - u_{12} + u_{22} = 0$$

$$u_{11} - u_{31} - u_{12} + u_{32} = 0 \quad .$$

The hypotheses to be tested are

$$H_0: \quad \bar{u}_{i\cdot} = u_{i'\cdot} \qquad (u_{1\cdot} = u_{2\cdot} = u_{3\cdot})$$

and

$$H_0: \quad \bar{u}_{\cdot j} = \bar{u}_{\cdot j'} \qquad (u_{\cdot 1} = u_{\cdot 2}) \quad .$$

Input:

$Y = (5.,6.,2.,3.,2.,3.,5.,6.,7.,8.,8.,9.,3.,4.,4.,6., 6.,7.)^T$

$TL_1 = (1. \quad -1. \quad 1. \quad -1. \quad 1. \quad -1.)^T$

$$TL_2 = \begin{pmatrix} 1. & 1. & -1. & -1. & 0. & 0. \\ 1. & 1. & 0. & 0. & -1. & -1. \end{pmatrix}^T$$

$$TL_3 = \begin{pmatrix} 1. & -1. & -1. & 1. & 0. & 0. \\ 1. & -1. & 0. & 0. & -1. & 1. \end{pmatrix}$$

$KM = (1,2,2,)^T$

$M = (2,2,5,3,1,5)^T$

$R = 10^{-6}$

$NL = 18$

$N = 6$

NF = 3

IND2 = 0

Output:

US = $(5.5,2.5,4.6,8.3,3.0,5.4)^T$

UH = $(3.19,4.81,5.39,7.01,3.65,5.27)^T$

SSQ_1 = 9.7078648

SSQ_2 = 15.682864

VH = 4.0208672

ANALYSIS OF VARIANCE

Source of Variation	Degrees-of-Freedom	Sum-of-Squares	Mean Square	F Statistic	95% Critical Value
Ration	1	9.7078648	9.7078648	2.4143709	18.512822
Sire	2	15.682864	7.8414319	1.9501843	19.000000
Experimental Error	2		4.0208672		

The Two-Way Classification Without Interaction and with Missing Cells

This example was taken from Blischke (ref. 7).

Fabric	Temperature			
	1	2	3	4
1	No observations	1.8 2.1 2.0 2.1	4.6	7.5 7.9
2	2.2 2.4	4.2 4.0	5.4 5.6	9.2
3	2.8 3.2	No observations	8.7 8.4	13.2
4	No observations	3.2 3.6 3.3	5.7 5.8	10.9 11.1

The model is

$$Y_{ijk} = u_{ij} + e_{ijk}$$

subject to $\theta^T u = 0$ where

$$\theta^T = \begin{array}{c} \begin{array}{ccccccccccccc} u_{12} & u_{13} & u_{14} & u_{21} & u_{22} & u_{23} & u_{24} & u_{31} & u_{33} & u_{34} & u_{42} & u_{43} & u_{44} \end{array} \\ \left[\begin{array}{ccccccccccccc} 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & 0 & 0 \end{array}\right] \end{array}$$

Consider the hypotheses

$$H_0: \quad u_{ij} = u_{ij'}$$

and

$$H_0: \quad u_{ij} = u_{i'j} \quad .$$

These hypotheses test for significant "differences" in temperature and fabric.

<u>Input</u>:

$Y = (1.8,2.0,2.1,2.1,4.6,7.5,7.9,2.2,2.4,4.2,4.0,5.4,5.6,$
$9.2,2.8,3.2,8.7,8.4,13.2,3.2,3.3,3.6,5.7,5.8,10.9,11.1)^T$

$$TL_1 = \begin{bmatrix} 1. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 1. & 0. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 1. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 1. & 0. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 1. & 0. & 0. & -1. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & -1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & 0. & -1. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & -1. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & 0. & -1. \end{bmatrix}^T$$

$$TL_2 = \begin{bmatrix} 0. & 0. & 0. & 1. & 0. & 0. & 0. & -1. & 0. & 0. & 0. & 0. & 0. \\ 1. & 0. & 0. & 0. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & 0. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & 0. \\ 0. & 0. & 1. & 0. & 0. & 0. & -1. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -1. \end{bmatrix}^T$$

$$TL_3 = \begin{bmatrix} 1. & -1. & 0. & 0. & -1. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 1. & 0. & -1. & 0. & -1. & 0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. \\ 1. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & 1. & 0. \\ 1. & 0. & -1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -1. & 0. & 1. \\ 0. & 0. & 0. & 1. & 0. & -1. & 0. & -1. & 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 1. & 0. & 0. & -1. & -1. & 0. & 1. & 0. & 0. & 0. \end{bmatrix}^T$$

$KM = (9,9,6)^T$

$M = (4,1,2,2,2,2,1,2,2,1,3,2,2)^T$

$R = 10^{-6}$

$NL = 26$

$N = 13$

$NF = 3$

$IND2 = 0$

Output:

US = $(2.0,4.6,7.7,2.3,4.1,5.5,9.2,3.0,8.55,13.2,3.37,5.75,11.0)^T$

UH = $(1.84,4.01,8.3,1.55,3.82,5.99,10.3,3.74,8.17,12.47,3.76,5.73,10.22)^T$

SSQ_1 = 215.23209

SSQ_2 = 37.864549

VH = .43067300

ANALYSIS OF VARIANCE

Source of Variation	Degrees-of-Freedom	Sum-of-Squares	Mean Square	F Statistic	95% Critical Value
Temperature	9	215.23209	23.914677	55.528619	4.0990173
Fabric	9	37.864549	4.2071721	9.7688318	4.0990173
Experimental Error	6		.43067300		

REFERENCES

1. Speed, F. M.: A New Approach to the Analysis of Linear Models. NASA TM X-58030, 1969.
2. Delaney, F. C.: OURHOG — Gram-Schmidt Orthogonalization Process. LEC Catalog No. 185, Prog. No. B165, 1968.
3. Oney, J. K.: JACOBI, Eigenvalue/Eigenvector Evaluation Subroutine. LEC Catalog No. 100, Prog. No. B117, 1966.
4. Delaney, F. C., Feiveson, A. H., Hill, F. O.: Stat.-Cat., A Statistical Catalog of Subroutines and Function Subroutines. LEC Catalog No. 162, Prog. No. C057, 1968.
5. Ostle, B.: Statistics in Research. Second ed., Iowa State University Press, 1963.
6. Harvey, W. R.: Least-Squares Analysis of Data with Unequal Subclass Numbers. USDA, ARS-20-8, 1960.
7. Blischke, W. R.: Data Analysis, A Survey of Techniques, Volume II, Example H16. C.E.I.R, Inc. (Subsidiary of Control Data Corp.), 9171 Wilshire Boulevard, Beverly Hills, Calif. 90210, 1969.

APPENDIX

PROGRAM LISTINGS

SAMPLE CALLING PROGRAM FOR BLISCHKE (ref. 7) EXAMPLE

```
 FOR MAIN
      PARAMETER N=13,NY=  26,NF=3
      DOUBLE PRECISION XL(N,N),G(N,N),W(N,N),E(N,N),X(N),Z(N),R
      DIMENSION Y(NY),TL(NF,N,N),KM(NF),M(N),US(N),UH(N),SSQ(NF),F(NF)
      IND2=0
      R=1.0D-06
      ALPHA=0.05
      NL=NY
      KM(1)=9
      KM(2)=9
      KM(3)=6
      READ (5,101) (Y(I),I=1,NY)
      READ (5,102) (M(I),I=1,N)
      WRITE (6,120) (Y(I),I=1,NY)
      WRITE (6,121) (M(I),I=1,N)
      DO 4 K=1,NF
      LM=KM(K)
      DO 2 J=1,LM
    2 READ (5,103) (TL(K,I,J),I=1,N)
      WRITE (6,122) K
      DO 3 I=1,LM
      WRITE (6,123) I
    3 WRITE (6,124) (TL(K,J,I),J=1,N)
    4 CONTINUE
      CALL GENHYP(Y,TL,KM,M,XL,G,W,E,X,Z,US,UH,SSQ,R,VH,NL,N,NF,IND2)
      WRITE (6,127)
      XMS1=SSQ(1)/KM(1)
      XMS2=SSQ(2)/KM(2)
      F(1)=XMS1/VH
      F(2)=XMS2/VH
      FN1=FISHIN(ALPHA,KM(1),KM(3))
      FN2=FISHIN(ALPHA,KM(2),KM(3))
      NV=KM(3)
      WRITE (6,128) KM(1),SSQ(1),XMS1,F(1),FN1,KM(2),SSQ(2),XMS2,F(2),
     1FN2,NV,VH
      WRITE (6,131)
      STOP
  101 FORMAT(26F3.1)
  102 FORMAT(40I2)
  103 FORMAT(40F2.0)
  120 FORMAT('1OBSERVATIONS'//(7(5X,G13.8)))
  121 FORMAT('0POPULATION SIZE'/(44(1X,I2)))
  122 FORMAT('0HYPOTHESIS MATRIX - TRANSPOSED, HYPOTHESIS NUMBER  ',I2/)
  123 FORMAT(5H0ROW ,I3)
  124 FORMAT(8(3X,G13.8))
  127 FORMAT('1',46X,'ANALYSIS OF VARIANCE'/1X,19('-'),'*',20('-'),'*',
     116('-'),'*',15('-'),'*',15('-'),'*',20('-'),'*'/' SOURCE OF VARATI
     2ON * DEGREES OF FREEDOM * SUM OF SQUARES *  MEAN SQUARE  *  F STAT
     3ISTIC  * 95( CRITICAL VALUE *'/1X,19('-'),'*',20('-'),'*',16('-'),
     4'*',15('-'),'*',15('-'),'*',20('-'),'*')
  128 FORMAT(9X,'      TEMP. *',8X,13,9X,'*',2X,G13.8,' * ',G13.8,' * ',G
     113.8,' *   ',G13.8,4X,'*'/'                    FABRIC *',8X,I3,9X,'*  ',G
     213.8,' * ',G13.8,' * ',G13.8,' *   ',G13.8,4X,'*'/' EXPERIMENTAL E
     3RROR *',8X,I3,9X,'*',16X,'* ',G13.8,' *',15X,'*   ',17X,'*')
  131 FORMAT(1X,111('-')//)
      END
```

```
@FOR GENHYP
      SUBROUTINE GENHYP(Y,TL,KM,M,XL,G,W,E,X,Z,US,UH,SSQ,R,VH,NL,N,
     1NF,IND2)
C-----------------------------------------------------------------------
C
C       Y  INPUT  - A SINGLY-DIMENSIONED SINGLE PRECISION ARRAY THAT CONTAINS
C                   THE OBSERVATIONS. THIS ARRAY SHOULD BE CONSTRUCTED SO THAT
C                   EACH CELL IS EXHAUSTED BEFORE GOING TO THE NEXT CELL.
C                   DIMENSIONED BY NL.
C      TL  INPUT  - A THREE-DIMENSIONAL SINGLE PRECISION ARRAY THAT CONTAINS THE
C                   HYPOTHESIS MATRICES. CONSIDER TL(K,I,J). K DENOTES THE
C                   K-TH HYPOTHESIS MATRIX, I AND J DENOTE THE I-TH ROW AND
C                   J-TH COLUMN OF THE K-TH HYPOTHESIS MATRIX. TL IS DIMENSIONED
C                   NF BY N BY N.
C      KM  INPUT  - A SINGLY-DIMENSIONED INTEGER ARRAY THAT CONTAINS THE
C                   CORRESPONDING DEGREES OF FREEDOM FOR THE HYPOTHESIS
C                   MATRICES. THAT IS KM(K) CONTAINS THE DEGREES OF FREEDOM
C                   ASSOCIATED WITH THE K-TH HYPOTHESIS MATRIX. NOTE THAT KM(K)
C                   IS ALSO THE NUMBER OF ROWS IN THE HYPOTHESIS MATRIX
C                   TRANSPOSED. DIMENSIONED NF.
C       M  INPUT  - A SINGLY-DIMENSIONED INTEGER ARRAY CONTAINING THE NUMBER OF
C                   OBSERVATIONS PER CELL (OR POPULATION). THAT IS M(I) CONTAINS
C                   THE NUMBER OF OBSERVATIONS IN THE I-TH POPULATION OR CELL.
C                   M IS DIMENSIONED BY N.
C      XL         - A TWO-DIMENSIONAL DOUBLE PRECISION ARRAY USED FOR TEMPORARY
C                   WORKING STORAGE. DIMENSIONED N BY N.
C       G  OUTPUT - A TWO-DIMENSIONAL DOUBLE PRECISION ARRAY THAT CONTAINS THE
C                   COVARIANCE MATRIX FOR UHAT. DIMENSIONED N BY N.
C       W         - A TWO-DIMENSIONAL DOUBLE PRECISION ARRAY USED FOR TEMPORARY
C                   WORKING STORAGE. DIMENSIONED N BY N.
C       E         - A TWO-DIMENSIONAL DOUBLE PRECISION ARRAY USED FOR TEMPORARY
C                   WORKING STORAGE. DIMENSIONED N BY N.
C       X         - A SINGLY-DIMENSIONED DOUBLE PRECISION ARRAY USED AS TEMPORARY
C                   WORKING STORAGE. DIMENSIONED BY N.
C       Z         - A SINGLY-DIMENSIONED DOUBLE PRECISION ARRAY USED AS TEMPORARY
C                   WORKING STORAGE. DIMENSIONED BY N.
C      US  OUTPUT - A SINGLY-DIMENSIONED SINGLE PRECISION ARRAY THAT CONTAINS
C                   THE CELL (OR POPULATION) MEANS. DIMENSIONED BY N.
C      UH  OUTPUT - A SINGLY-DIMENSIONED SINGLE PRECISION ARRAY THAT CONTAINS
C                   THE MINIMUM VARIANCE UNBIASED ESTIMATE OF CELL (OR
C                   POPULATION) MEANS. DIMENSIONED BY N.
C     SSQ  OUTPUT - A SINGLY-DIMENSIONED SINGLE PRECISION ARRAY THAT CONTAINS THE
C                   SUM OF SQUARES ASSOCIATED WITH THE K-TH HYPOTHESIS MATRIX.
C                   DIMENSIONED BY NF.
C       R  INPUT  - A DOUBLE PRECISION LOCATION THAT CONTAINS CONVERGENCE
C                   CTITERION FOR EIGENVALUE, EIGENVECTOR COMPUTATION.
C                   1.0E-06 HAS BEEN FOUND TO BE SUFFICIENT IN ALL TEST CASES.
C                   THESE TEST CASES WERE RAN ON THE UNIVAC 1108.
C      VH  OUTPUT - A SINGLE PRECISION LOCATION THAT CONTAINS THE MINIMUM
C                   VARIANCE UNBIASED ESTIMATE OF THE POPULATION VARIANCE.
C      NL  INPUT  - TOTAL NO. OF OBSERVATIONS
C       N  INPUT  - NO. OF POPULATIONS OR CELLS
C      NF  INPUT  - THE NUMBER OF HYPOTHESES TO BE TESTED. THE NUMBER OF
C                   HYPOTHESIS MATRICES.
C    IND2  INPUT  - IF IND2.EQ.1, THERE IS NO CONSTRAINT MATRIX.
C                   IF IND2 NE.1, THERE IS A CONSTRAINT MATRIX.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION XL(N,N),W(N,N),G(N,N),Z(1),X(1),E(N,N),R
      DIMENSION Y(1),TL(NF,N,N),KM(1),M(1),US(1),UH(1),SSQ(1)
      IF(IND2.EQ.1) GO TO 2
```

```
      LX=KM(NF)
      DO 1 I=1,N
      DO 1 J=1,LX
    1 XL(I,J)=DBLE(TL(NF,I,J))
      CALL EIGDR2(XL,G,M,W,X,Z,R,N,LX)
    2 CONTINUE
      CALL UHAT2(Y,M,X,XL,US,UH,G,W,Z,VH,NL,N,LX,IND2)
      L=NF
      IF(IND2.NE.1) L=NF-1
      DO 10 K=1,L
      LM=KM(K)
      DO 3 I=1,N
      DO 3 J=1,LM
    3 XL(I,J)=DBLE(TL(K,I,J))
      CALL TESTH(XL,E,M,W,G,X,Z,UH,N,LM,IND2,R,SSQ(K))
   10 CONTINUE
      RETURN
      END
```

```
' FOR TESTH
      SUBROUTINE TESTH(XL,E,M,W,G,X,Z,UH,N,LM,IND2,R,SSL)
C-----------------------------------------------------------------------
C
C      XL   INPUT  - HYPOTHESIS MATRIX, DIMENSIONED N BY N.
C       E   OUTPUT - HYPOTHESIS MATRIX 'ORTHOGONALIZED', DIMENSIONED N BY N.
C       M   INPUT  - VECTOR INPUT CONTAINING THE NO. OF OBSERVATIONS PER
C                      POPULATION, THAT IS M(I) CONTAION THE NO. OF OBS. IN THE
C                      (I)-TH POPULATION OR CELL. M IS DIMENSIONED BY N.
C       W   OUTPUT - USED AS WORKING STORAGE, DIMENSIONED N BY N.
C       G   INPUT  - COVARIANCE MATRIX FOR UHAT, DIMENSIONED N BY N.
C       X   OUTPUT - EIGENVALUES OF TRANSFORMED HYPOTHESIS MATRIX, DIMENSIONED BYN
C       Z   OUTPUT - TEMP. WORKING STORAGE, DIMENSIONED BY N.
C      UH   INPUT  - CELL MEANS, DIMENSIONED BY N.
C       N   INPUT  - NO. OF POPULATIONS OR CELLS
C    IND2   INPUT  - IF IND2.EQ.1, THERE IS NO CONSTRAINT MATRIX.
C                    IF IND2 NE.1, THERE IS A CONSTRAINT MATRIX.
C      LM   INPUT  - NO. OF ROWS IN XL TRANSPOSE OR NO. OF COLUMNS IN XL.
C       R   INPUT  - CONVERGENCE CRITERIOR FOR EIGENVALUE, EIGENVECTOR COMPUTATION
C     SSL   OUTPUT - SUM OF SQUARES DUE TO THE HYPOTHESIS MATRIX.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION XL(N,N),G(N,N),W(N,N),E(N,N),Z(1),X(1),S,R
      DIMENSION UH(1),M(1)
      CALL DOURHG(XL,E,N,LM,N,N,R)
      IF(IND2.NE.1) GO TO 50
      DO 7 I=1,LM
      DO 7 J=1,LM
      S=0.0
      DO 6 K=1,N
    6 S=S+(E(K,I)*E(K,J))/M(K)
    7 W(I,J)=S
      GO TO 51
   50 DO 9 I=1,LM
      DO 9 J=1,LM
      S=0.0
      DO 8 K=1,N
      DO 8 L=1,N
    8 S=S+E(K,I)*G(K,L)*E(L,J)
    9 W(I,J)=S
   51 CONTINUE
      CALL DJACOB(W,XL,X,Z,N,LM,R,1,IER)
      IF(IER.EQ.1) WRITE (6,103) IER
  103 FORMAT('0IER = ',I2,'JACOBI FAILED TO CONVERGE IN 150 ITERATIONS')
      DO 10 I=1,LM
   10 X(I)=W(I,I)
      DO 12 I=1,N
      DO 12 J=1,LM
      S=0.0
      DO 11 K=1,LM
   11 S=S+E(I,K)*XL(K,J)
   12 W(I,J)=S
      SSL=0.0
      DO 14 J=1,LM
      S=0.0
      DO 13 I=1,N
   13 S=S+W(I,J)*UH(I)
C
C  IF D IS KNOWN IN (LAMBDA-TRANSPOSE) X (MU) = D THEN INSERT
C     S=S-D(J)
C

   14 SSL=SSL+(S**2)/X(J)
      RETURN
      END
```

```
* FOR UHAT2
      SUBROUTINE UHAT2(Y,M,X,XL,US,UH,G,W,Z,VH,NL,N,LX,IND2)
C-----------------------------------------------------------------------
C
C       Y  INPUT  - A SINGLE-DIMENSIONED SINGLE PRECISION ARRAY THAT CONTAINS
C                     THE OBSERVATIONS. THIS ARRAY SHOULD BE CONSTRUCTED SO THAT
C                     EACH CELL IS EXHAUSTED BEFORE GOING TO THE NEXT CELL.
C                     DIMENSIONED BY NL.
C       M  VECTOR INPUT CONTAINING THE NO. OF OBSERVATIONS PER POPULATION,
C           THAT IS M(I) CONTAINS THE NO. OF OBS. IN THE I-TH POPULATION
C           OR CELL. M IS DIMENSIONED N.
C       X  INPUT  - EIGENVALUES OF TRANSFORMED THETA MATRIX
C      XL  INPUT  - TRANSFORMED CONSTRAINT MATRIX
C      US  OUTPUT - USTAR. DIMENSIONED N
C      UH  OUTPUT - UHAT.  DIMENSIONED N
C       G  OUTPUT - COVARIANCE MATRIX FOR UHAT
C       W  OUTPUT - I - V X XL X X-INVERSE X XL-TRANSPOSE
C       Z  OUTPUT - (A-TRANSPOSE) X Y, DIMENSIONED N
C      VH  OUTPUT - MINIMUM VARIANCE UNBIASED ESTIMATE OF POPULATION VARIANCE
C      NL  INPUT  - TOTAL NO. OF OBSERVATIONS
C       N  INPUT  - NO. OF POPULATIONS OR CELLS
C      LX  NO. OF ROWS IN XL TRANSPOSE OR NO. OF COLUMNS IN XL.
C    IND2  INPUT  - IF IND2.EQ.1, THERE IS NO CONSTRAINT MATRIX.
C                   IF IND2.NE.1, THETA TRANSPOSE AND XI NE. ZERO
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION XL(N,N),G(N,N),W(N,N),X(1),Z(1),S
      DIMENSION Y(1),M(1),US(1),UH(1)
C
C  COMPUTATION OF USTAR
C
      K=1
      MM=0
      DO 2 J=1,N
      MM=MM+M(J)
      S=0.0
      DO 1 I=K,MM
    1 S=S+Y(I)
      Z(J)=S
      US(J)=S/FLOAT(M(J))
      K=K+M(J)
    2 CONTINUE
      IF(IND2.EQ.1) GO TO 70
C
C  COMPUTATION OF
C   I - (X-TRANSPOSE X X) INVERSE X THETA X D-INVERSE X THETA-TRANSPOSE
C
      DO 3 I=1,N
      DO 3 J=1,LX
    3 G(I,J)=XL(I,J)/(M(I)*X(J))
      DO 5 I=1,N
      DO 5 J=1,N
      S=0.0
      DO 4 K=1,LX
    4 S=S+G(I,K)*XL(J,K)
    5 W(I,J)=S
      DO 6 I=1,N
      DO 6 J=1,N
      W(I,J)=-W(I,J)
      IF(I.EQ.J) W(I,J)=1.0+W(I,J)
    6 CONTINUE
```

```
C
C   COMPUTATION OF COVARIANCE MATRIX FOR UHAT
C
      DO 7 I=1,N
      DO 7 J=1,N
    7 G(I,J)=W(I,J)/M(J)
C
C   COMPUTATION OF UHAT
C
      DO 60 I=1,N
      UH(I)=0.0
      DO 60 J=1,N
   60 UH(I)=UH(I)+W(I,J)*US(J)
C
C   IF C IS KNOWN IN (THETA-TRANSPOSE) X (MU) = C THEN INSERT
C     DO 61 I=1,N
C     S=0.0
C     DO 40 J=1,LX
C  40 S=S+XL(I,J)/(M(I)*X(J))*C(J)
C  61 UH(I)=UH(I)+S
C
C
C   COMPUTATION OF R1
C
      R1=0.0
      DO 63 J=1,LX
      S=0.0
      DO 62 I=1,N
   62 S=S+XL(I,J)*US(I)
C
C   IF C IS KNOWN IN (THETA-TRANSPOSE) X (MU) = C THEN INSERT
C     S=S-C(J)
C
   63 R1=R1+(S**2)/X(J)
C
C   COMPUTATION OF RESIDUAL SUM OF SQUARES
C
      GO TO 71
   70 R1=0.0
      LX=0
      DO 80 J=1,N
   80 UH(J)=US(J)
   71 S=0.0
      DO 64 I=1,NL
   64 S=S+Y(I)**2
      SU=0.0
      DO 65 I=1,N
   65 SU=SU+(Z(I)*Z(I))/M(I)
      RES=S+R1-SU
      VR=RES/FLOAT(NL-N+LX)
      RETURN
      END
```

```
' FOR EIGOR2
      SUBROUTINE EIGOR2(XL,G,M,W,X,Z,R,N,LX)
C-----------------------------------------------------------------------
C
C     XL  INPUT - THE CONSTRANT MATRIX OR HYPOTHESIS MATRIX  (N BY LX)
C         OUTPUT - TRANSFORMED CONSTRANT OR HYPOTHESIS MATRIX
C      G  AFTER OURHOG IS CALLED G IS XL ORTHOGONALIZED      (N BY LX)
C      M  VECTOR INPUT CONTAINING THE NO. OF OBSERVATIONS PER POPULATION,
C          THAT IS M(I) CONTAINS THE NO. OF OBS. IN THE 1-TH POPULATION
C          OR CELL. M IS DIMENSIONED N.
C      W  USED AS WORKING STORAGE                             (N BY N )
C      X  TEMPORARY WORKING STORAGE. DIMENSIONED N
C         OUTPUT - EIGENVALUES OF TRANSFORMED XL MATRIX
C      Z  TEMPORARY WORKING STORAGE. DIMENSIONED N
C      R  DETERMINES OCCURRENCE OF ZERO VECTOR IN OURHOG. SEE OURHOG.
C      N  NO. OF POPULATIONS OR CELLS
C     LX  NO. OF ROWS IN XL TRANSPOSE OR NO. OF COLUMNS IN XL.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION XL(N,N),W(N,N),G(N,N),Z(1),X(1),R,S
      DIMENSION M(1)
      CALL DOURHG(XL,G,N,LX,N,N,R)
      DO 3 I=1,LX
      DO 3 J=1,LX
      S=0.0D0
      DO 2 K=1,N
    2 S=S+(G(K,I)*G(K,J))/M(K)
    3 XL(I,J)=S
      CALL DJACOB(XL,W,X,Z,N,LX,R,1,IER)
      IF(IER.EQ.1) WRITE (6,103) IER
  103 FORMAT(6H0IER =,I2,44H METHOD FAILED TO CONVERGE IN 150 ITERATIONS
     $//)
      DO 5 I=1,LX
    5 X(I)=XL(I,I)
      DO 7 I=1,N
      DO 7 J=1,LX
      S=0.0D0
      DO 6 K=1,LX
    6 S=S+G(I,K)*W(K,J)
    7 XL(I,J)=S
      RETURN
      END
```

```
' FOR DOURHG
      SUBROUTINE DOURHG(A,Z,M,N,MM,NN,EP)
C-----------------------------------------------------------------------
C
C        GIVEN A BASIS (OF N VECTORS CONTAINING M ELEMENTS) FOR A REAL VECTOR
C         SPACE OF FINITE-DIMENSION P (P.LE.N), THIS SUBROUTINE COMPUTES AN
C         ORTHONORMAL BASIS FOR THE SPACE. THE ORTHONORMAL BASIS COMPUTED IS
C         COMPOSED OF UNIT VECTORS WHICH ARE MUTUALLY ORTHOGONAL.
C   A    IS A DOUBLE-DIMENSIONED ARRAY, PROVIDED BY THE USER, THAT CONTAINS THE
C         BASIS OF N VECTORS. THE MATRIX A MUST BE CONSTRUCTED SO THAT EACH
C         COLUMN CONTAINS ONE AND ONLY ONE VECTOR. FOR EXAMPLE, THE I-TH COLUMN
C         MUST CONTAIN THE I-TH VECTOR COMPOSED OF M ELEMENTS. THE MATRIX A IS
C         USED AS WORKING STORAGE AND THEREFORE, UPON RETURN TO THE CALLING
C         PROGRAM A WILL NOT CONTAIN ITS ORIGINAL INPUT.
C   Z    IS A DOUBLE-DIMENSIONED ARRAY, RETURNED TO THE USER, THAT CONTAINS THE
C         ORTHONORMAL BASIS OF UNIT VECTORS WHICH ARE MUTUALLY ORTHOGONAL. THE
C         VECTORS WILL BE STORED COLUMN-WISE.
C   M    IS AN INTEGER, PROVIDED BY THE USER, WHICH DENOTES THE NUMBER OF
C         ELEMENTS OF THE INPUT VECTORS.
C   N    IS AN INTEGER, PROVIDED BY THE USER, WHICH DENOTES THE NUMBER OF
C         INPUT VECTORS.
C   MM   IS AN INTEGER, PROVIDED BY THE USER, WHICH DENOTES THE MAXIMUM ROW
C         DIMENSION OF THE MATRIX A.
C   NN   IS AN INTEGER, PROVIDED BY THE USER, WHICH DENOTES THE MAXIMUM COLUMN
C         DIMENSION OF THE MATRIX A. NOTE THAT A MUST BE DIMENSIONED MM BY NN.
C   EP   IS A REAL NUMBER, PROVIDED BY THE USER, THAT DETERMINES THE OCCURRENCE
C         OF A ZERO VECTOR DURING THE GRAM-SCHMIDT ORTHOGONALIZATION PROCESS.
C         IN OTHER WORDS IF Y IS A VECTOR AND Y . Y .LE. EP, THEN Y IS SET EQUAL
C         TO ZERO.  (SEE METHOD)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION A(MM,NN),Z(MM,NN),XINNER,TEMPN,S,SM,EP
      IF(N.EQ.1) GO TO 9
      DO 5 I=1,M
      Z(I,1)=A(I,1)
      DO 5 J=2,N
      Z(I,J)=0.0D0
    5 CONTINUE
      DO 100 J=2,N
      JMINUS=J-1
C-------------------------------------------------------------
C
C     FIND THE INNER PRODUCT OF THE (J-1) COLUMN OF Z WITH ITSELF
C     DIVIDE THE (J-1) COLUMN OF Z BY ITS INNER PRODUCT
C
C-------------------------------------------------------------
      XINNER=0.0D0
      DO 20 I=1,M
   20 XINNER=XINNER+Z(I,JMINUS)*Z(I,JMINUS)
      IF(XINNER.GT.EP) GO TO 30
      DO 25 I=1,M
      Z(I,JMINUS)=0.0D0
   25 A(I,JMINUS)=0.0D0
      GO TO 41
   30 CONTINUE
      DO 40 I=1,M
   40 A(I,JMINUS)=Z(I,JMINUS)/XINNER
C-------------------------------------------------------------
C
C     FIND INNER PRODUCT OF J-TH COLUMN OF A WITH COLUMNS OF Z LESS THAN J
C
```

```
C-----------------------------------------------------------------------
   41 CONTINUE
      DO 50 I=1,JMINUS
      TEMPN=0.0D0
      DO 45 K=1,M
   45 TEMPN=TEMPN+A(K,J)*Z(K,I)
      DO 46 K=1,M
   46 Z(K,J)=Z(K,J)+A(K,I)*TEMPN
   50 CONTINUE
      DO 75 I=1,M
   75 Z(I,J)=A(I,J)-Z(I,J)
  100 CONTINUE
      XINNER=0.0D0
      DO 80 I=1,M
   80 XINNER=XINNER+Z(I,N)*Z(I,N)
      IF(XINNER.GT.EP) GO TO 85
      DO 81 I=1,M
      A(I,N)=0.0D0
   81 Z(I,N)=0.0D0
   85 CONTINUE
      S=0.0D0
      DO 7 J=2,N
      SM=S
      S=0.0D0
      DO 6 I=1,M
    6 S=S+Z(I,1)*Z(I,J)
      S=DABS(S)
      IF(SM.GT.S) S=SM
    7 CONTINUE
      DO 8 I=1,M
      DO 8 J=1,N
      IF(DABS(Z(I,J)).LE.S) Z(I,J)=0.0D0
    8 CONTINUE
      GO TO 11
    9 CONTINUE
      DO 10 I=1,M
   10 Z(I,1)=A(I,1)
   11 CONTINUE
      DO 3 J=1,N
      S=0.0D0
      DO 1 I=1,M
    1 S=S+Z(I,J)*Z(I,J)
      S=SQRT(S)
      DO 2 I=1,M
    2 Z(I,J)=Z(I,J)/S
    3 CONTINUE
      RETURN
      END
```

```
' FOR DJACOB
      SUBROUTINE DJACOB(A,UMAT,X,Y,MMM,N,RHO,OPT,ERR)
      DOUBLE PRECISION A(MMM,MMM),UMAT(MMM,MMM),X(1),Y(1),S,SUM,TAN22,
     $ERR,RHO,FLOATN,T,TF,AMAX,TANO,TEMP,SINO,COSO,CCO,SSO,SCO
      INTEGER OPT,ERR
       IF(N.LE.0) GO TO 150
      IF(N.EQ.1) GO TO 149
       NM1 = N-1
      S=0.0D0
      SUM=0.0D0
      TAN22=0.414213562373095D0
       NCOUNT = 0
      ERR=0.0D0
       DO 10 I=1,NM1
       L = I+1
       DO 10 J=L,N
      S=S+A(I,J)*A(I,J)
   10  SUM = SUM + A(I,J)*A(I,J)
       SUM = 2.0*SUM
      IF(OPT.EQ.0) GO TO 40
      S=SQRT(S)
       DO 30 I=1,N
       DO 20 J=1,N
   20 UMAT(I,J)=0.0D0
   30 UMAT(I,I)=1.0D0
      IF(S.LT.RHO) RETURN
   40  FLOATN = FLOAT(N)
       T = SQRT(SUM)/FLOATN
       TF = RHO*T
       IF(TF.GT.T) GO TO 150
   50  MCOUNT = 0
      AMAX=DABS(A(1,2))
      I = 1
   11 J = I+1
   12 IF(DABS(A(I,J)).GT.AMAX) AMAX=DABS(A(I,J))
      IF(DABS(A(I,J)).LT.T) GO TO 141
       MCOUNT = 1
      IF(DABS(A(I,I)-A(J,J)).LT.RHO) GO TO 70
       TANO = A(I,J)/(2.0*(A(I,I)-A(J,J)))
      IF(DABS(TANO).GT.TAN22) GO TO 70
       TEMP = 1.0 + TANO*TANO
       SINO = (2.0*TANO)/TEMP
       COSO = (2.0-TEMP)/TEMP
       GO TO 80
   70 W=1.0D0
      IF(A(I,J).LT.0.0D0) W=-1.0D0
      IF((A(I,I)-A(J,J)).LT.0.0D0) W=-1.0D0*W
      SINO=0.707106781186548D0*W
      COSO=0.707106781186548D0
   80  CCO = COSO*COSO
       SSO = SINO*SINO
       SCO = SINO*COSO
       DO 90 L=1,N
       IF(L.EQ.I.OR.L.EQ.J) GO TO 90
       Y(L) = A(J,L)*COSO - A(I,L)*SINO
       X(L) = A(I,L)*COSO + A(J,L)*SINO
   90  CONTINUE
       X(I) = A(J,J)*SSO + A(I,I)*CCO + 2.0*A(I,J)*SCO
       Y(I) = (A(J,J)-A(I,I))*SCO + A(I,J)*(CCO-SSO)
       X(J) = Y(I)
       Y(J) = A(J,J)*CCO + A(I,I)*SSO - 2.0*A(I,J)*SCO
```

```
       DO 100 L=1,N
       A(J,L) = Y(L)
       A(L,J) = A(J,L)
       A(I,L) = X(L)
       A(L,I) = A(I,L)
 100   CONTINUE
      IF(OPT.EQ.0) GO TO 130
       DO 110 L=1,N
       X(L) = UMAT(L,I)*COSO + UMAT(L,J)*SINO
       Y(L) = UMAT(L,J)*COSO - UMAT(L,I)*SINO
 110   CONTINUE
       DO 120 L=1,N
       UMAT(L,I) = X(L)
      UMAT(L,J)=Y(L)
 120   CONTINUE
 130   CONTINUE
 141  J = J+1
      IF(J.LE.N)GO TO 12
      I = I+1
      IF(I.LE.NM1)GO TO 11
      NCOUNT = NCOUNT +1
      IF(NCOUNT .GT. 150)GO TO 160
       IF(MCOUNT.EQ.1) GO TO 50
      IF(TF.GT.AMAX) GO TO 150
      IF(TF.GT.T) GO TO 150
       T = T/FLOATN
      GO TO 50
 160  ERR=1.0D0
      RETURN
 149  UMAT(1,1)=1.0D0
 150   RETURN
      END
```

```
' FOR FISHIN,FISHIN
C-----------------------------------------------------------------------
C CALCULATES THE INVERSE 'F' VALUE GIVEN THE CONFIDENCE COEFFICIENT
C    ALPHA AND THE DEGREES OF FREEDOM(N).
C-----------------------------------------------------------------------
      FUNCTION FISHIN(ALPHA,N1,N2)
      Y1=N1
      Y2=N2
C-----------------------------------------------------------------------
C  ADJUST FOR DEGREES OF FREEDOM EQUAL TO 1
C-----------------------------------------------------------------------
      IF(N1.EQ.1)  Y1=2
      IF(N2.EQ.1)  Y2=2
C-----------------------------------------------------------------------
C  CALL TINORM TO GET INVERSE NORMAL VALUE OF 1.-ALPHA
C-----------------------------------------------------------------------
      X=PHINV(1.0-ALPHA)
C-----------------------------------------------------------------------
C  COMPUTE LAMDA VALUE
C-----------------------------------------------------------------------
      Y=(X**2-3.)/6.
      IC=0
C-----------------------------------------------------------------------
C  COMPUTE THE INITIAL APPROXIMATION TO THE INVERSE 'F' FUNCTION
C-----------------------------------------------------------------------
      Y1=1./(Y1-1.)
      Y2=1./(Y2-1.)
      H=2./(Y1+Y2)
      X=X*SQRT(H+Y)/H-(Y1-Y2)*(Y+5./6.-2./(3.*H))
      X=EXP(2.*X)
C-----------------------------------------------------------------------
C  COMPUTE THE CONSTANT TO THE 'F' DISTRIBUTION,TESTING FOR N1 AND/OR N2
C   ODD OR EVEN.
C-----------------------------------------------------------------------
      G=1.
      IB1=2
      IF(MOD(N1,2).EQ.0) GO TO 1
      G=1.7724539
      IB1=1
    1 IB2=2
      IF(MOD(N2,2).EQ.0) GO TO 2
      G=G*1.7724539
      IB2=1
    2 IB3=2
      IF(MOD(N1+N2,2).EQ.0) GO TO 3
      G=G/1.7724539
      IB3=1
    3 IF((IB1+IB2).NE.2)  G=2.*G
      IF((N1+N2).LE.3)  GO TO 5
      ND=N1+N2-2-IB3
      DO 4 I=0,ND,2
      IF((IB1+I).LE.(N1-2))  G=G*(IB1+I)
      IF((IB2+I).LE.(N2-2))  G=G*(IB2+I)
    4 G=G/(IB3+I)
C-----------------------------------------------------------------------
C  COMPUTE THE VALUE OF FISHIN
C-----------------------------------------------------------------------
    5 Y2=N2/(N2+N1*X)
      Y1=1.-Y2
      Y=1.+(G*(1.-ALPHA-FISH(X,N1,N2)))/SQRT(Y1**N1*Y2**N2)
      FISHIN=X*Y
```

```
C-----------------------------------------------------------------------
C  IF FISHIN IS NEGATIVE, RESET FISHIN TO .5*LAST APPROXIMATION(X).
C-----------------------------------------------------------------------
      IF(Y.LT.0.)  FISHIN=.5*X
C-----------------------------------------------------------------------
C  IF THE ABSOLUTE VALUE OF THE DIFFERENCE IS LESS THAN .5E-6, RETURN
C-----------------------------------------------------------------------
      IF(ABS(X/FISHIN-1.).LT.(.5E-6))  GO TO 7
C-----------------------------------------------------------------------
C  IF THE RELATIVE VALUE OF THE DIFFERENCE IS LESS THAN .5E-6, RETURN
C-----------------------------------------------------------------------
      IF(ABS(X-FISHIN).LT.(.5E-6)) GO TO 7
      IC=IC+1
      IF(IC.GT.100)  RETURN
C-----------------------------------------------------------------------
C  SET THE APPROXIMATION EQUAL TO FISHIN AND CONTINUE TO ITERATE
C-----------------------------------------------------------------------
      X=FISHIN
      GO TO 5
C-----------------------------------------------------------------------
C  IF OVERFLOW OCCURRED DURING COMPUTATION OF TINORM, RETURN TO
C   STATEMENT K IN THE CALLING PROGRAM
C-----------------------------------------------------------------------
    7 RETURN
      END
```

```
' FOR FISH,FISH
      FUNCTION FISH(F,N1,N2)
      LOGICAL E1,E2,E3                                                  FISH
       IF(N1.GE.100.AND.N2.GE.100) GOTO 9
C-----------------------------------------------------------------------
C     INITIALIZATION AND SETTING OF LOGICAL SWITCHES  TO .TRUE. IF
C     THE DEGREES OF FREEDOM ARE EVEN
C-----------------------------------------------------------------------
      E1=.FALSE.                                                        FISH
      E2=.FALSE.                                                        FISH
      E3=.FALSE.                                                        FISH
      IF(MOD(N1,2).EQ.0) E1=.TRUE.                                      FISH
      IF(MOD(N2,2).EQ.0) E2=.TRUE.                                      FISH
      X=N2/(N2+N1*F)                                                    FISH
      IF(.NOT.(E1.OR.E2)) GO TO 5                                       FISH
      IF(E1.AND..NOT.E2) GO TO 1                                        FISH
      IF(.NOT.E1.AND.E2) GO TO 2                                        FISH
      IF(N1.LE.N2) GO TO 1                                              FISH
C-----------------------------------------------------------------------
C   INITIALIZATION FOR SECOND DEGREE OF FREEDOM EVEN  AND LESS THAN
C      FIRST DEGREE OF FREEDOM IF IT TOO IS EVEN
C-----------------------------------------------------------------------
    2 I=N1                                                              FISH
      N1=N2                                                             FISH
      N2=I                                                              FISH
      X=1.0-X                                                           FISH
      E3=.TRUE.                                                         FISH
C-----------------------------------------------------------------------
C     INITIALIZATION FOR FIRST DEGREE OF FREEDOM EVEN AND LESS THAN
C   SECOND DEGREE OF FREEDOM IF IT IS EVEN
C-----------------------------------------------------------------------
    1 Y=1.0-X                                                           FISH
C-----------------------------------------------------------------------
C     CALCULATION OF PROBABILITY FOR AT LEAST ONE DEGREE OF FREEDOM EVEN
C-----------------------------------------------------------------------
      FISH=0.0                                                          FISH
      H=SQRT(X**N2)                                                     FISH
      M=N1/2-1                                                          FISH
      DO  3 I=0,M                                                       FISH
      FISH=FISH+H                                                       FISH
    3 H=(H*Y*(N2+2.*I))/(2.*(I+1.))
      IF(E3) GO TO 4                                                    FISH
C-----------------------------------------------------------------------
C        ADJUST CALCULATED PROBABILITY IF ITS ONES COMPLEMENT WAS
C    CALCULATED ORIGINALLY
C-----------------------------------------------------------------------
      FISH=1.0-FISH                                                     FISH
      RETURN                                                            FISH
    4 I=N1                                                              FISH
      N1=N2                                                             FISH
      N2=I                                                              FISH
      RETURN                                                            FISH
C-----------------------------------------------------------------------
C     CALCULATION OF THE PROBABILITY FOR BOTH DEGREES OF FREEDOM ODD
C-----------------------------------------------------------------------
    5 Y=1.0-X                                                           FISH
      H=.63661977*SQRT(X*Y)                                             FISH
      FISH=.63661977*ACOS(SQRT(X))                                      FISH
      IF(N2.EQ.1) GO TO 8                                               FISH
      M=N2-2                                                            FISH
      DO  6  I=1,M,2                                                    FISH
```

```
      FISH=FISH+H                                                   FISH
    6 H=H*X*(I+1)/(I+2)                                             FISH
    8 IF(N1.EQ.1) RETURN                                            FISH
      H=H*N2                                                        FISH
      M=N1-2                                                        FISH
      DO   7   I=1,M,2                                              FISH
      FISH=FISH-H                                                   FISH
    7 H=H*Y*(N2+I)/(I+2)                                            FISH
      RETURN                                                        FISH
    9  D1=N1
       D2=N2
       DT=(D1/D2)*F
       DN=SQRT((2.*D2-1.)*DT)-SQRT(2.*D1-1.)
       X=DN/SQRT(1.+DT)
       FISH=PHI(X)
       RETURN
      END                                                           FISH
```

```
' FOR PHINV
      FUNCTION PHINV( P )
      IF(P .EQ. 1.0)GO TO 98
      IF(P .EQ. 0.0)GO TO 97
      IF(P .GT. 1.0)GO TO 88
      IF(P .LT. 0.0)GO TO 88
      K = 1
      IF(P .GT. 0.5)GO TO 47
    8 T3=SQRT(-2.0*ALOG(P))
      T4P=2.515517+.802853*T3+.010328*T3*T3
      T5P=1.0+1.432788*T3+.189269*T3*T3+.001308*T3*T3*T3
      XT=T3-T4P/T5P
      XT=-XT
   13 DO 53 I=1,100
      PHP = EXP(-0.5*XT*XT)
      PT = PHI (XT)
      IF(ABS(P-PT) .LT. P*4.0E-8)GO TO 99
      Z = (P-PT)*2.50662827 / PHP
      XT = XT + Z
   53 CONTINUE
      GO TO 99
   47  P = 1.0 - P
      K = 2
      GO TO 8
   99 GO TO (26,27),K
   26 PHINV = XT
      RETURN
   27 PHINV = -XT
      P = 1.0 - P
      RETURN
   98 PHINV = 1.0E+38
      RETURN
   97 PHINV = -1.0E+38
      RETURN
   88 WRITE(6,10) P
   10 FORMAT(1H0,5X,29HARGUMENT NOT A PROBABILITY = ,5X,E14.7   )
      RETURN
      END
```

DATA FOR BLISCHKE (ref. 7) EXAMPLE

```
18 20 21 21 46 75 79 22 24 42 40 54 56 92 28 32 87 84132 32 33 36 57 58109111
4 1 2 2 2 2 1 2 2 1 3 2 2
1-1
1 0-1
0 0 0 1-1
0 0 0 1 0-1
0 0 0 1 0 0-1
0 0 0 0 0 0 0 1-1
0 0 0 0 0 0 0 1 0-1
0 0 0 0 0 0 0 0 0 0 1-1
0 0 0 0 0 0 0 0 0 0 1 0-1
0 0 0 1 0 0 0-1
1 0 0 0-1
1 0 0 0 0 0 0 0 0 0-1
0 1 0 0 0-1
0 1 0 0 0 0 0 0-1
0 1 0 0 0 0 0 0 0 0 0-1
0 0 1 0 0 0-1
0 0 1 0 0 0 0 0 0-1
0 0 1 0 0 0 0 0 0 0 0 0-1
1-1 0 0-1 1
1 0-1 0-1 0 1
1-1 0 0 0 0 0 0 0 0-1 1
1 0-1 0 0 0 0 0 0 0-1 0 1
0 0 0 1 0-1 0-1 1
0 0 0 1 0 0-1-1 0 1
```