GRAPHICAL REPRESENTATION OF NON-ORTHOGONAL DATA

Samuel G. Lindle[1] and David M. Allen[1]

BU-555-M[2] April, 1975

Abstract

Graphical representation is used to show the incompleteness of information obtained from the pair of distinct AOV tables in which each of two non-orthogonal factors appear sequentially one before the other. More information is contained in the four projection vectors considered. A three-step algorithm using the singular value decomposition is produced that projects these four vectors onto three-dimensional space while keeping distortion small. The interpretation of results obtained from non-orthogonal data is discussed. Four tables and eleven figures are given.

---

[1] Department of Statistics, University of Kentucky, Lexington, Ky. 40506. This paper was prepared while the authors were on leave at Cornell University.

[2] The Biometrics Unit Mimeo Series, Cornell University, Ithaca, N.Y. 14853.

GRAPHICAL REPRESENTATION OF NON-ORTHOGONAL DATA

Samuel G. Lindle[1] and David M. Allen[1]

BU-555-M[2] April, 1975

1. Introduction. In the context of orthogonal designs, Scheffé [1] has used graphical methods to show that the sum of squares of an AOV table are squared norms of a partitioning of y into orthogonal parts. Searle [2, Chapter 7] demonstrates the considerable difficulty involved in problems of interpretation and analysis in the case of non-orthogonal data. He uses a pair of distinct AOV tables in which each of two non-orthogonal factors appear sequentially one before the other as the principle analytic method. It is shown in this paper that there is information about the data not contained in both of the two AOV tables. Graphical methods are used to illustrate this information. The technique involves the displaying of relationships among the four n dimensional vectors P_1y, P_2y, Py and y. In order to obtain perspective viewing these four vectors are projected onto three-dimensional space in a manner that keeps distortion small. An algorithm for achieving this is developed in the following sections.

2. The Model. Let y be an $n \times 1$ vector of observations, X be an $n \times k$ matrix of known constants, β be an $n \times 1$ vector of unknown parameters and ϵ be the $n \times 1$ vector of error terms. Partition X as

$$X = [\underbrace{X_0}_{k_0} \vdots \underbrace{X_1}_{k_1} \vdots \underbrace{X_2}_{k_2}]$$

---

[1] Department of Statistics, University of Kentucky, Lexington, Ky. 40506. This paper was prepared while the authors were on leave at Cornell University.

[2] The Biometrics Unit Mimeo Series, Cornell University, Ithaca, N.Y. 14853.

where $k = k_0 + k_1 + k_2$ and correspondingly partition β as

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} .$$

Let X_0 be the matrix associated with blocking factors and covariables. X_1 and X_2 are associated with factors of interest. The model is the usual one

$$y = X\beta + \epsilon$$

where

$$\epsilon \sim N(\underline{0}, \sigma^2 I) .$$

Statistics for inference about β_1 and β_2 depend only on

$$[\underbrace{X^*}_{k_1+k_2} \vdots \underbrace{y^*}_{1}] = [\underbrace{X_1^*}_{k_1} \vdots \underbrace{X_2^*}_{k_2} \vdots \underbrace{y^*}_{1}] = I - X_0(X_0'X\ )^- X_0')[X_1 \vdots X_2 \vdots y] .$$

Let

$$\begin{aligned} P_1 &= X_1^*(X_1^{*\prime} X_1^*)^- X_1^{*\prime} \\ P_2 &= X_2^*(X_2^{*\prime} X_2^*)^- X_2^{*\prime} \\ P &= X^*(X^{*\prime} X^*)^- X^{*\prime} \end{aligned}$$

be the projection matrices formed from X^* .

3. <u>Illustration of Information Gained</u>. First consider figures 1a) and 1b) below. Note that y has been omitted from the figures as it is presently unimportant. Both figures illustrate the case of two non-orthogonal factors. In both figures $P_1 y^*$, $(P-P_1)y^*$ and Py^* are the same. Further, the lengths of $P_2 y$ and $P_2^* y^*$ and of $(P-P_2)y^*$ and $(P-P_2^*)y^*$ are the same. Thus both 1a) and

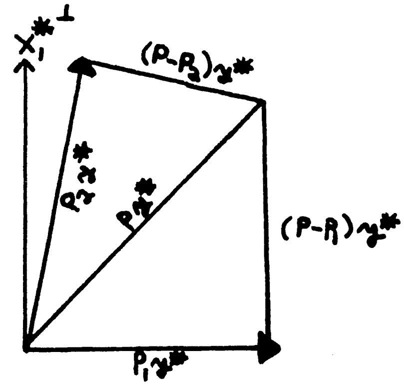


Figure 1 a)

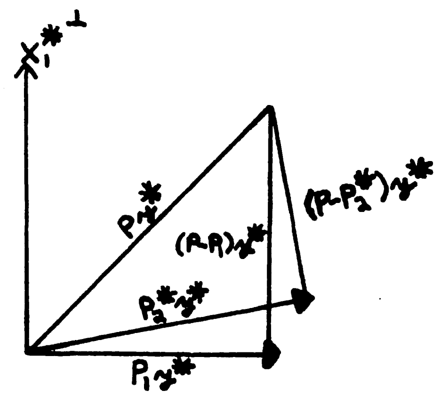


Figure 1 b)

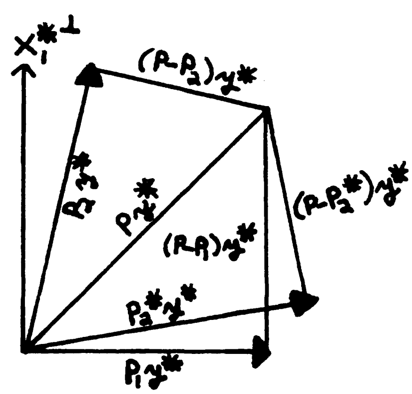


Figure 1 c)

1b) have identical pairs of distinct AOV tables. However, clearly the two figures represent two distinctly different situations since the two factors are nearly orthogonal in 1a) whereas they are obviously highly non-orthogonal in 1b). Where non-orthogonality exists it is important that it is detected and that the interpretation of the data be appropriately adjusted for it. A graphical method is developed in the following sections to aid in achieving this purpose. Figure 1c) is the unification of 1a) and 1b) into one figure so that the concepts of this section are more concisely represented by it.

4. A Three-Step Procedure for Graphical Representation of Non-Orthogonal Data Which Keeps Distortion Small.

Since all the information about β_1 and β_2 in the data is contained in the four n dimensional vectors P_1y^*, P_2y^*, Py^* and y^*, hereafter let

$$Z = [Z_1 \vdots y^*] = [P_1y^* \vdots P_2y^* \vdots Py^* \vdots y^*] \quad .$$

One needs to project Z onto three-dimensional space in order to view it in perspective as a three-dimensional plot on two-dimensional paper. It is desirable that the visually appealing constraint that P_1y^*, P_2y^* and Py^* all lie in the same plane be built into the procedure. The following three-step procedure describes a way of doing this while keeping distortion small.

First Step. In order to obtain the appealing property that the three vectors P_1y^*, P_2y^* and Py^* all lie in the same plane, an $n \times 2$ matrix $A_1 = [\underset{-}{a}_1 \;\; \underset{-}{a}_2]$ with orthonormal columns is needed such that $A_1A_1'Z_1$ is a projection of Z which for some criterion produces minimum distortion. A natural criterion is to choose the A_1 which minimizes

(1) $$\|Z_1 - A_1A_1'Z_1\|_F$$

where $\|\cdot\|_F$ is the Frobenius norm, or equivalently to choose the A_1 which maximizes

$$(2) \qquad \mathrm{tr}(A_1'Z_1Z_1'A_1) \quad .$$

Choosing this as the criterion the A_1 which maximizes (2) (or minimizes (1)) is

$$A_1 = A_1^* = [\underset{\sim}{a}_1^* \vdots \underset{\sim}{a}_2^*]$$

where $\underset{\sim}{a}_1^*$ is the eigenvector associated with the largest eigenvalue (λ_1) of Z_1Z_1' and $\underset{\sim}{a}_2^*$ is the eigenvector associated with the second largest eigenvalue (λ_2) of Z_1Z_1' . Henceforth, the "$*$" in A_1^* will be dropped and it will be referred to simply as A_1 .

Second Step. Let $A = [A_1 \;\; \underset{\sim}{a}_3]$ where $\underset{\sim}{a}_3$ is a vector of unit length such that $\underset{\sim}{a}_3'Z_1 = 0$ and $AA' = A_1A_1' + \underset{\sim}{a}_3\underset{\sim}{a}_3'$. The matrix A is sought that maximizes

$$\mathrm{tr}(A'ZZ'A) \quad .$$

But

$$\max_{\substack{\underset{\sim}{a}_3'Z_1 = 0 \\ AA' = A_1A_1' + \underset{\sim}{a}_3\underset{\sim}{a}_3'}} \mathrm{tr}(A'ZZ'A)$$

$$= \max_{\underset{\sim}{a}_3} \; \mathrm{tr}\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & (\underset{\sim}{a}_3'\underset{\sim}{y}^*)^2 \end{bmatrix}$$

$$= \lambda_1+\lambda_2 + \max_{\underset{\sim}{a}_3} (\underset{\sim}{a}_3'\underset{\sim}{y}^*)^2 \quad .$$

The problem thus reduces to finding the vector $\underset{\sim}{a}_3$ that maximizes

$$(\underline{a}_3'\underline{y}^*)^2 \quad .$$

Let $\bar{Z}_1$ be an $n \times n - 2$ matrix such that $\bar{Z}_1'Z_1 = 0$ and $\bar{Z}_1'\bar{Z}_1 = I$. Then, there exists an $n \times 1$ vector $\underline{c}$ such that

$$\underline{a}_3 = \frac{1}{\|\underline{c}\|} \bar{Z}_1\underline{c} \quad .$$

Consequently it is needed to find the $\underline{c}$ that maximizes

$$\frac{(\underline{c}'\bar{Z}_1'\underline{y}^*)^2}{\underline{c}'\underline{c}} \quad .$$

But it is well known that this is maximized for

$$\underline{c}^* \propto \bar{Z}_1'\underline{y}^*$$

and thus

$$\underline{a}_3^* \propto \bar{Z}_1\bar{Z}_1'\underline{y}^* \quad .$$

But

$$\bar{Z}_1\bar{Z}_1'\underline{y}^* = (I-Z_1(Z_1'Z_1)^-Z_1')\underline{y}^* = (I-P)\underline{y}^* \quad .$$

Also $\underline{a}_3^*$ is of unit length so

$$\underline{a}_3^* = \frac{(I-P)\underline{y}^*}{\|(I-P)\underline{y}^*\|} = \frac{(I-P)\underline{y}^*}{\sqrt{SSE}}$$

where SSE is the sum of squares error for the model. Consequently $A^* = [\underline{a}_1^*\underline{a}_2^*\underline{a}_3^*]$ is the required matrix and $A^*A^{*\prime}Z$ is a projection of Z such that the first three columns of $A^*A^{*\prime}Z$ all line in the same plane in n dimensional space. The distortion incurred in $A^*A^{*\prime}Z$ as a projection of Z is small. Henceforth, the "$*$" in A^* will be dropped and it will be referred to simply as A. Also, the vector $\underset{\sim}{a}_3^*$ will be referred to simply as $\underset{\sim}{a}_3$.

Third Step. The column vectors of $AA'Z$ are still in n dimensional space. Premultiplying $AA'Z$ by A' does not change the lengths of or the angles between the columns of $AA'Z$. Thus, no distortion occurs upon premultiplying $AA'Z$ by A' and one obtains

$$A'Z \quad .$$

Note that $A'Z$ is a 3×4 matrix so that its columns are vectors in three-dimensional space as desired. Also,

$$\begin{aligned}
A'Z = A'(AA'Z) &= A'[A_1A_1'+\underset{\sim}{a}_3\underset{\sim}{a}_3'][Z_1 \vdots \underset{\sim}{y}^*] \\
&= \begin{bmatrix} A_1' \\ \underset{\sim}{a}_3' \end{bmatrix} [A_1A_1'Z_1 | A_1A_1'\underset{\sim}{y}^* + \underset{\sim}{a}_3(\underset{\sim}{a}_3'\underset{\sim}{y}^*)] \\
&= \begin{bmatrix} A_1'Z_1 & A_1'\underset{\sim}{y}^* \\ \underset{\sim}{0} & \underset{\sim}{a}_3'\underset{\sim}{y}^* \end{bmatrix} = \begin{bmatrix} A_1'Z_1 & A_1'\underset{\sim}{y}^* \\ \underset{\sim}{0} & \sqrt{SSE} \end{bmatrix}
\end{aligned}$$

because $\underset{\sim}{a}_3'\underset{\sim}{y}^* = \dfrac{\underset{\sim}{y}^{*\prime}(I-P)\underset{\sim}{y}^*}{\sqrt{SSE}} = \sqrt{SSE}$. The third row of $A'Z$ is deleted in order to project four-dimensional space onto three-dimensional space. (The third row was deleted since it corresponds to the smallest eigenvalue of $Z_1'Z_1$.) To triangularize the resulting 3×4 matrix, it is premultiplied by the appropriate elementary reflector.

5. <u>Actual Calculating Algorithm</u>. A FORTRAN program NORTH (RANK 2) has been written to do the actual calculations for the three-step procedure discussed in the last section.

Basically NORTH (RANK 2) uses the singular value decomposition to decompose Z_1 . That is, it computes the matrices U, S and V such that

$$Z_1 = USV'$$

where

U is an n x 3 matrix such that $U'U = I$
V is a 3 x 3 matrix such that $V'V = I$
S is a 3 x 3 diagonal matrix .

Consequently one has

$$Z = [USV'|y^*] \quad . \tag{3}$$

Choose $\bar{U}$ such that $\bar{U}'\bar{U} = I$ and $U\bar{U} = 0$ and premultiply (3) by $[U \quad \bar{U}]'$ to obtain

$$\begin{bmatrix} U' \\ \bar{U}' \end{bmatrix} Z = \begin{bmatrix} SU' & U'y^* \\ 0 & \bar{U}'y^* \end{bmatrix} \quad .$$

Deleting the third row, triangularizing the resulting matrix by premultiplying it by the appropriate elementary reflector and noting that $\bar{U}'y^*$ is the square root of the sum of squares for error, the matrix described at the end of the previous section is obtained.

The program does three other things to make the plots appealing to the eye. First, in order to keep the resultant vector corresponding to the original y^* vector in the first octant, if either of the two remaining elements of $U'y^*$ is negative the corresponding row is multiplied by -1 . Second, since it is difficult to reference the position of points outside the limits of the grid drawn by the program, if either the (1,2) element or the (2,3) element or both are negative then they are subtracted from every element in their row and from the corresponding element of the origin. Third, the program finds the element of the resulting matrix which has the largest absolute value (E) and scales all the elements of the matrix by multiplying by 10/E .

The columns of the resulting matrix will be the terminal points of the vectors P_1y, P_2y, Py and y . The initial point of these vectors is the origin if neither the (1,2) element nor the (2,3) element above was negative. Otherwise, the initial point is the appropriately shifted origin. For the vectors $(P-P_1)y$, $(P-P_2)y$, $(I-P_1)y$, $(I-P_2)y$ and $(I-P)y$, appropriate initial and terminal points are chosen from the four vectors: P_1y, P_2y, Py and y . X_1 and X_2 are scalar multiples of P_1y and P_2y (respectively).

One can choose which of the above mentioned vectors he desires to draw. Various options are available concerning vectorheads and vector labels if these are desired. NORTH (RANK 2) is flexible and can be used for other purposes than applying the techniques of this paper. The comment cards of the program describe in detail the applications of and the options available for NORTH (RANK 2).

Figure 2a) in Appendix A gives the printed output of a run of NORTH (RANK 2) for the data given at the end of Appendix B. This run and the labeled output follow the calculating scheme developed in this section for the three-step algorithm of the preceding section. Figure 2b) gives the resulting plot. Appendix B also gives a complete listing of NORTH (RANK 2). The subroutine DSVD was written by P. Businger at Bell Telephone Laboratories with some changes and editing done by R. Underwood at Stanford University.

6. Discussion. Let $F(\beta_1,\beta_2|\beta_0)$, $F(\beta_1|\beta_0)$, $F(\beta_2|\beta_0,\beta_1)$, $F(\beta_2|\beta_0)$ and $F(\beta_1|\beta_0,\beta_2)$ be the usual F statistics for the pair of distinct AOV tables in which each of two non-orthogonal factors appear sequentially one before the other. If $F(\beta_1,\beta_2|\beta_0)$ is significant then it indicates that joint fitting of β_1 and β_2 has explanatory value for variations in y^*. Each of the last four F statistics may be either significant or non-significant. This creates sixteen different situations. One good way to illustrate these different situations is to make a table with four different situations for $F(\beta_1|\beta_0)$ and $F(\beta_2|\beta_0,\beta_1)$ determining the rows and the four different situations for $F(\beta_2|\beta_0)$ and $F(\beta_1|\beta_0,\beta_2)$ determining the columns of a four by four array. Instead of putting numbers in the array, the effects which should be included in the model are put in the array. Searle [2] has used this type table in his Table 7.4. A reproduction of this table is given below in Table 1.

Since symmetry is a natural property of such tables, one needs only consider the ten different situations in the upper (or lower) triangular portion. If one uses the upper triangular portion and numbers the elements by rows, one obtains Table 2. It should be noted that situation 4 is not possible unless the number of degrees of freedom are different for β_1 and β_2. It is clear that one should fit both β_1 and β_2 in situation 1 and neither β_1 nor β_2 in situation 10. How the rest of the table is filled in is personal preference.

In this section, three of the many valid schemes for fitting in the remaining elements of the table will be described. It is not claimed that any one is superior to any of the others. The last scheme described will be equivalent to Searle's Table 7.4. Nine figures representing each of the nine possible situations are given in Appendix A.

	Fitting β_2 and then β_1 after β_2				
Fitting β_1 and then β_2 after β_1	$F(\beta_2 \mid \beta_0)$ $F(\beta_1 \mid \beta_0, \beta_2)$	Sig Sig	N S Sig	Sig N S	N S N S
		Effects to be included in model			
$F(\beta_1 \mid \beta_0)$: $F(\beta_2 \mid \beta_0, \beta_1)$:	Sig Sig	β_1 and β_2	β_1 and β_2	β_2	β_1 and β_2
$F(\beta_1 \mid \beta_0)$: $F(\beta_2 \mid \beta_0, \beta_1)$:	N S Sig	β_1 and β_2	β_1 and β_2	β_2	β_1 and β_2
$F(\beta_1 \mid \beta_0)$: $F(\beta_2 \mid \beta_0, \beta_1)$:	Sig N S	β_1	β_1	β_1 and β_2	β_1
$F(\beta_1 \mid \beta_0)$: $F(\beta_2 \mid \beta_0, \beta_1)$:	N S N S	β_1 and β_2	β_1 and β_2	β_2	neither β_1 nor β_2

Sig = Significant; N S = Not Significant

Table 1. A Reproduction of Searle's Table 7.4 [2, page 278] With β_1 and β_2 used in Place of α and β

	Fitting β_2 and then β_1 after β_2				
Fitting β_1 and then β_2 after β_1	$F(\beta_2 \mid \beta_0)$ $F(\beta_1 \mid \beta_0, \beta_2)$	Sig Sig	N S Sig	Sig N S	N S N S
$F(\beta_1 \mid \beta_0)$ $F(\beta_2 \mid \beta_0, \beta_1)$:	Sig Sig	1	2	3	4
$F(\beta_1 \mid \beta_0)$: $F(\beta_2 \mid \beta_0, \beta_1)$:	N S Sig		5	6	7
$F(\beta_1 \mid \beta_0)$: $F(\beta_2 \mid \beta_0, \beta_1)$:	Sig N S			8	9
$F(\beta_1 \mid \beta_0)$: $F(\beta_2 \mid \beta_0, \beta_1)$:	N S N S				10

Table 2. Numbering Scheme

The first scheme will be called the full set and null set avoiding scheme for reasons which will become apparent later. It can be described by the following three steps:

<u>First Step</u>. Include β_1 in the model if both $F(\beta_1|\beta_0)$ and $F(\beta_1|\beta_0,\beta_2)$ are significant.

<u>Second Step</u>. Include β_2 in the model if both $F(\beta_2|\beta_0)$ and $F(\beta_2|\beta_0,\beta_1)$ are significant.

<u>Third Step</u>. If neither β_1 nor β_2 is in the model at this point, then put $\beta_1(\beta_2)$ in the model if the angle between P_1y and Py is smaller (larger) in absolute value than the angle between P_2y and Py . If the angles are equal, randomly choose one (unless one is more economical than the other). In the case where one of β_1 and β_2 has already been included during the first two steps, do not add the other to the model.

Note that this scheme, as its name suggests, discourages any model in which neither β_1 nor β_2 is included or in which both β_1 and β_2 are included. See Table 3 for the appropriate table for this scheme. In situations 4, 5, 7 and 8 a computer plot can be drawn by using NORTH (RANK 2) and the "smaller angle" determined by sight if the angles are sufficiently different. Otherwise, the cosine of the angle can be determined from the output accompanying the plot.

If $\underline{\beta}^{*\prime} = (\beta_1,\beta_2)'$ and $\underline{\mu} = X^*\underline{\beta}^*$, then

$$E(P_1y^*) = P_1\underline{\mu}$$
$$E(P_2y^*) = P_2\underline{\mu}$$
$$E(Py^*) = P\underline{\mu} \quad .$$

Let θ_1 be the angle between $E(P_1\underline{y}^*)$ and $E(P\underline{y}^*)$ and let θ_2 be the angle between

Fitting β_1 and then β_2 after β_1		Fitting β_2 and then β_1 after β_2			
	$F(\beta_2 \mid \beta_0)$ $F(\beta_1 \mid \beta_0, \beta_2)$	Sig Sig	N S Sig	Sig N S	N S NN S
		Effects to be included in model			
$F(\beta_1 \mid \beta_0)$: $F(\beta_2 \mid \beta_0, \beta_1)$:	Sig Sig	β_1 and β_2	β_1	β_2	smaller angle
$F(\beta_1 \mid \beta_0)$: $F(\beta_2 \mid \beta_0, \beta_1)$:	N S Sig		smaller angle	β_2	smaller angle
$F(\beta_1 \mid \beta_0)$: $F(\beta_2 \mid \beta_0, \beta_1)$:	Sig N S			smaller angle	β_1
$F(\beta_1 \mid \beta_0)$: $F(\beta_2 \mid \beta_0, \beta_1)$:	N S N S				neither β_1 nor β_2

P

Table 3. Full Set and Null Set Avoiding Scheme

Fitting β_1 and then β_2 after β_1		Fitting β_2 and then β_1 after β_2			
	$F(\beta_2 \mid \beta_0)$ $F(\beta_1 \mid \beta_0, \beta_2)$	Sig Sig	N S Sig	Sig N S	N S N S
		Effects to be included in model			
$F(\beta_1 \mid \beta_0)$ $F(\beta_2 \mid \beta_0, \beta_1)$		β_1 and β_2	β_1	β_2	β_1
$F(\beta_1 \mid \beta_0)$ $F(\beta_2 \mid \beta_0, \beta_1)$			β_1 and β_2	β_2	β_2
$F(\beta_1 \mid \beta_0)$ $F(\beta_2 \mid \beta_0, \beta_1)$				β_1 and β_2	β_1
$F(\beta_1 \mid \beta_0)$ $F(\beta_2 \mid \beta_0, \beta_1)$					neither β_1 nor β_2

Table 4. Null Set Avoiding Scheme

$E(P_2y^*)$ and $E(Py^*)$, then

$$\cos\theta_1 = \underline{\mu}'P_1'P\underline{\mu}$$

$$\cos\theta_2 = \underline{\mu}'P_2'P\underline{\mu} \quad .$$

Consequently, one could also estimate $\cos\theta_1$ and $\cos\theta_2$ by

$$\cos\theta_1 = \underline{y}^{*\prime}P_1'P\underline{y}^*$$

and

$$\cos\theta_2 = \underline{y}^{*\prime}P_2'P\underline{y}^* \quad .$$

The second scheme shall be called the null set avoiding scheme. The first two steps are the same as the first two steps for the preceding scheme. The last step is:

Third Step. If neither β_1 nor β_2 has been put in the model in the first two steps, then add β_1 to the model if $F(\beta_1|\beta_0)$ is significant and add β_2 to the model if $F(\beta_2|\beta_0)$ is significant. If neither of these is significant then add β_1 to the model if $F(\beta_1|\beta_0,\beta_2)$ is significant and add β_2 if $F(\beta_2|\beta_0,\beta_1)$ is significant.

This scheme is equivalent to Table 4.

If in addition to the three steps of the previous scheme, the following step is added, Searle's Table 7.4 results:

Fourth Step. If one of β_1 and β_2 has been added in the three previous steps and if one calls the other factor γ, then if $F(\gamma|\beta_0,\gamma^c)$ is significant (where γ^c is the factor different from γ) add γ . If $F(\gamma|\beta_0,\gamma^c)$ is significant but nothing else is significant, add γ^c .

It should also be noted that if λ_1, λ_2 and λ_3 are the eigenvalues of $Z_1'Z_1$ in descending order than either

$$\frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3}$$

or

$$\frac{\lambda_3}{\lambda_2}$$

could be used as feasible measures of distortion incurred in using NORTH (RANK 2) to plot the four vectors P_1y^*, P_2y^*, Py^* and y^* in three dimensions. These measures have range between 0 and 1 except that the first cannot attain 1 .

Appendix A: Figures related to section 6.

```
ZZ=|¯ P1Y P2Y PY Y ¯|
    _              _
```

5.30330	2.12132	5.30330	5.30330
-5.30330	-2.12132	-5.30330	-5.30330
0.0	4.63721	3.18198	3.18198
0.0	-4.63721	-3.18198	-3.18198
0.0	0.0	0.0	9.00000

```
    |¯ S*VT       UT*Y       ¯|
PZ= |                         |
    |  0          SQRT(SSE)   |
     _                       _
```

6.44871	5.92789	8.74634	8.74634
-3.82937	4.10701	0.03986	0.03986
0.00000	0.00000	-0.00000	-0.00000
0.0	0.0	0.0	9.00000

DELETING THE THIRD ROW OF PZ

6.44871	5.92789	8.74634	8.74634
-3.82937	4.10701	0.03986	0.03986
0.0	0.0	0.0	9.00000

PZ PREMULTIPLIED BY APPROPRIATE ELEMENTARY REFLECTOR IN ORDER TO TRIANGULARIZE IT

-7.50000	-3.00000	-7.50000	-7.50000
0.0	6.55801	4.50000	4.50000
0.0	0.0	0.0	9.00000

IF THE LAST ELEMENT OF A ROW IS NEGATIVE, IT IS MULTIPLIED BY -1

7.50000	3.00000	7.50000	7.50000
0.0	6.55801	4.50000	4.50000
0.0	0.0	0.0	9.00000

SCALING TO MAKE ALL ELEMENTS HAVE ABSOLUTE VALUE LESS THAN 10

8.33333	3.33333	8.33333	8.33333
0.0	7.28667	5.00000	5.00000
0.0	0.0	0.0	10.00000

THE ORIGIN, P1Y, X1, P2Y, X2, PY AND Y (RESPECTIVELY) ARE:

0.0	0.0	0.0
8.33333	0.0	0.0
9.16666	0.0	0.0
3.33333	7.28667	0.0
3.66666	8.01533	0.0
8.33333	5.00000	0.0
8.33333	5.00000	10.00000

Figure 2 a) Printed Output of a Run of NORTH (RANK 2)

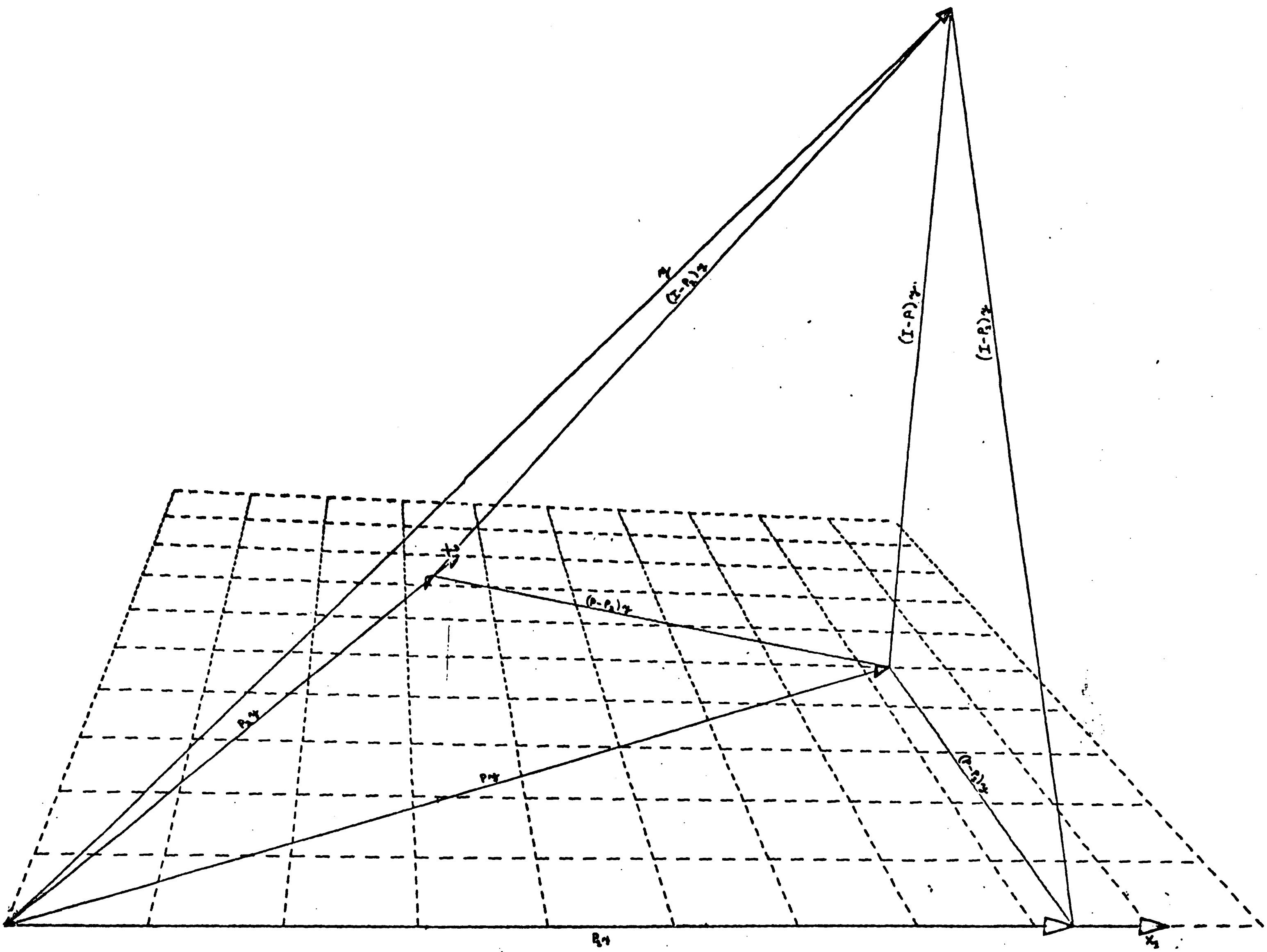
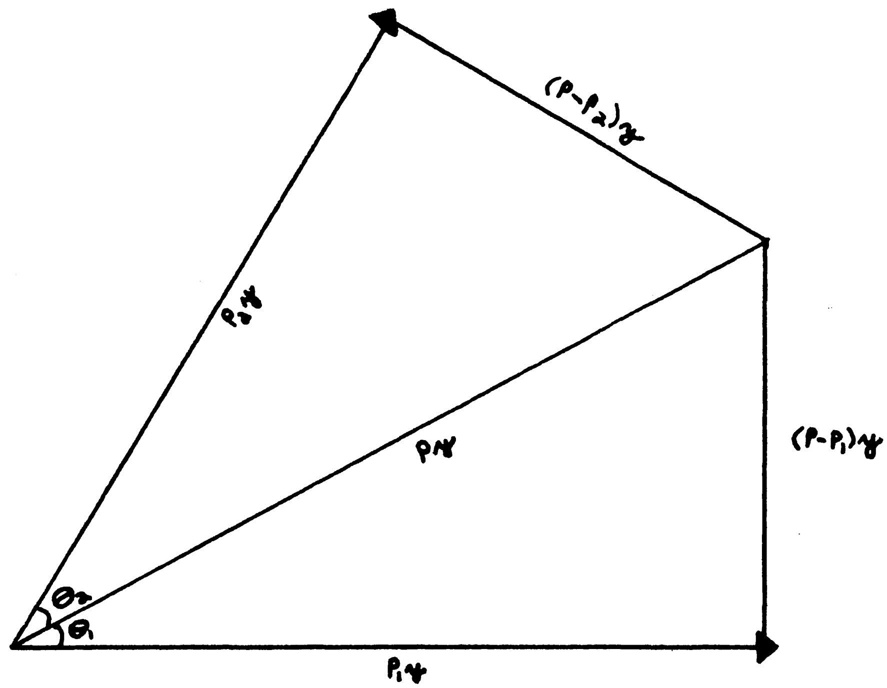


Figure 2 b) Plot Produced by a Run of NORTH (RANK 2)



$\|P_1y\|^2$	100.00	Sig	$\|P_2y\|^2$	94.44	Sig
$\|(P-P_1)y\|^2$	28.44	Sig	$\|(P-P_2)y\|^2$	34.00	Sig
$\|Py\|^2$	128.44		$\|Py\|^2$	128.44	
$\|(I-P)y\|^2$	9.00		$\|(I-P)y\|^2$	9.00	
$\|y\|^2$	137.44		$\|y\|^2$	137.44	

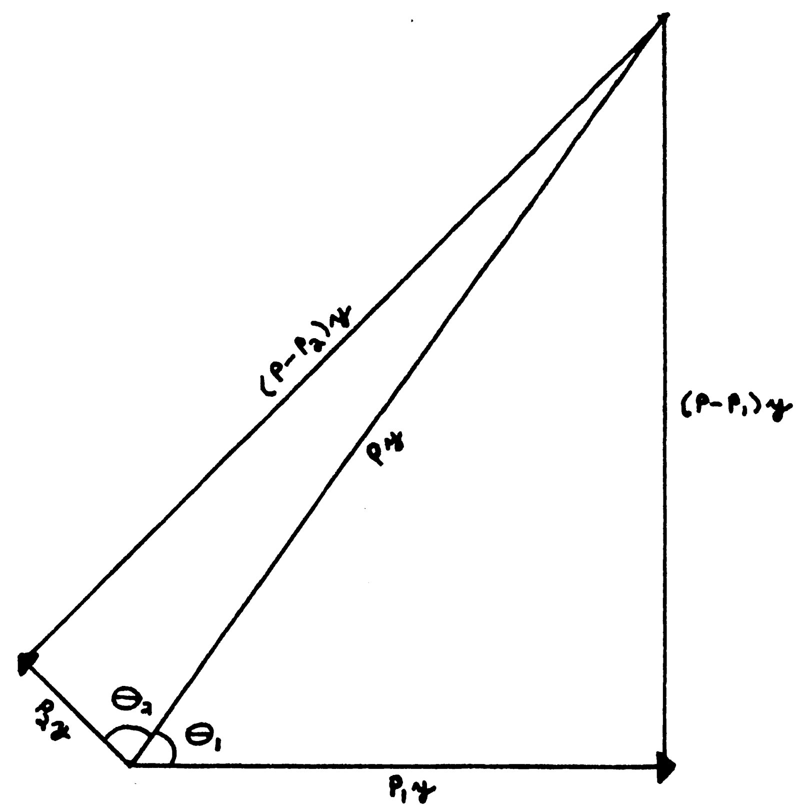
$$\cos\theta_1 = \frac{\|P_1y\|}{\|Py\|} = \frac{10}{\sqrt{128,44}} = .88237$$

$$\theta_1 \approx 28^\circ$$

$$\cos\theta_2 = \frac{\|P_2y\|}{\|Py\|} = \frac{\sqrt{94.44}}{\sqrt{128.44}} = .25749$$

$$\theta_2 \approx 31^\circ$$

Figure 3. Situation Number 1

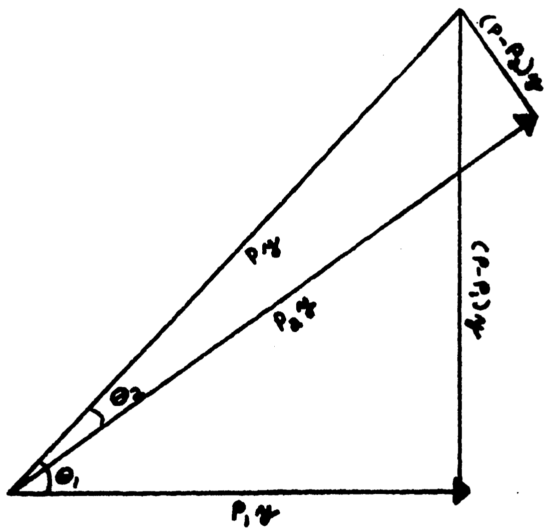


$\|\|P_1y\|\|^2$	50.00	Sig	$\|\|P_2y\|\|^2$	4.00	N S
$\|\|(P-P_1)y\|\|^2$	98.00	Sig	$\|\|(P-P_2)y\|\|^2$	144.00	Sig
$\|\|Py\|\|^2$	148.00		$\|\|Py\|\|^2$	148.00	
$\|\|(I-P)y\|\|^2$	9.00		$\|\|(I-P)y\|\|^2$	9.00	
$\|\|y\|\|^2$	157.00		$\|\|y\|\|^2$	157.00	

$$\text{Cos } \theta_1 = \frac{5\sqrt{2}}{\sqrt{148}} = .58124 \qquad \cos\theta_2 = \frac{2}{\sqrt{148}} = .16440$$

$$\theta_1 \approx 54^\circ 30' \qquad \theta_2 \approx 80^\circ 30'$$

Figure 4. Situation Number 2



$\|\|P_1y\|\|^2$	36.00	Sig	$\|\|P_2y\|\|^2$	74.00	Sig
$\|\|(P-P_1)y\|\|^2$	<u>40.96</u>	Sig	$\|\|(P-P_2)y\|\|^2$	<u>2.96</u>	N S
$\|\|Py\|\|^2$	76.96		$\|\|Py\|\|^2$	76.96	
$\|\|(I-P)y\|\|^2$	<u>9.00</u>		$\|\|(I-P)y\|\|^2$	<u>9.00</u>	
$\|\|y\|\|^2$	85.96		$\|\|y\|\|^2$	85.96	

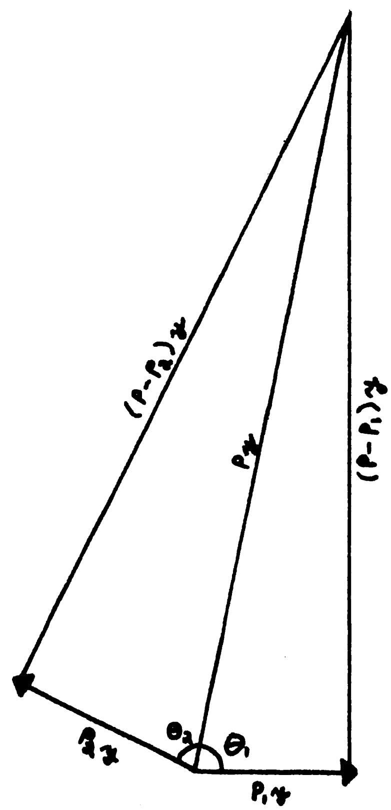
$$\cos\theta_1 = \frac{6}{\sqrt{76.96}} = .68394$$

$$\theta_1 \approx 46^\circ 50'$$

$$\cos\theta_2 = \frac{\sqrt{74.00}}{\sqrt{76.96}} = .98058$$

$$\theta_2 \approx 11^\circ 20'$$

Figure 5. Situation Number 3



$\|\|P_1y\|\|^2$	4.99	N S	$\|\|P_2y\|\|^2$	7.20	N S
$\|\|(P-P_1)y\|\|^2$	100.00	Sig	$\|\|(P-P_2)y\|\|^2$	96.80	Sig
$\|\|Py\|\|^2$	104.00		$\|\|Py\|\|^2$	104.00	
$\|\|(I-P)y\|\|^2$	9.00		$\|\|(I-P)y\|\|^2$	9.00	
$\|\|y\|\|^2$	113.00		$\|\|y\|\|^2$	113.00	

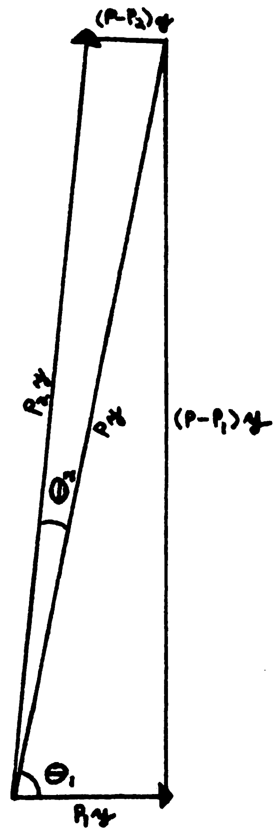
$$\cos \theta_1 = \frac{2}{\sqrt{104}} = .19612$$

$$\theta_1 \approx 78^\circ 40'$$

$$\cos \theta_2 = \frac{\sqrt{7.20}}{\sqrt{104.00}} = .26312$$

$$\theta_2 \approx 74^\circ 40'$$

Figure 6. Situation Number 5

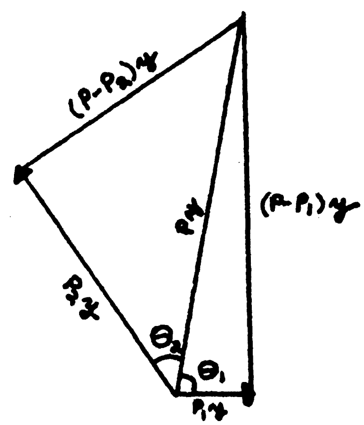


$\|P_1y\|^2$	4.00	N S	$\|P_2y\|^2$	101.00	Sig
$\|(P-P_1)y\|^2$	98.01	Sig	$\|(P-P_2)y\|^2$	1.01	N S
$\|Py\|^2$	102.01		$\|Py\|^2$	102.01	
$\|(I-P)y\|^2$	9.00		$\|(I-P)y\|^2$	9.00	
$\|y\|^2$	111.01		$\|y\|^2$	111.01	

$$\cos \theta_1 = \frac{2}{\sqrt{102.01}} = .19802 \qquad \cos \theta_2 = \frac{\sqrt{101}}{\sqrt{102.01}} = .99504$$

$$\theta_1 \approx 78^\circ 30' \qquad \theta_2 \approx 5^\circ 40'$$

Figure 7. Situation Number 6

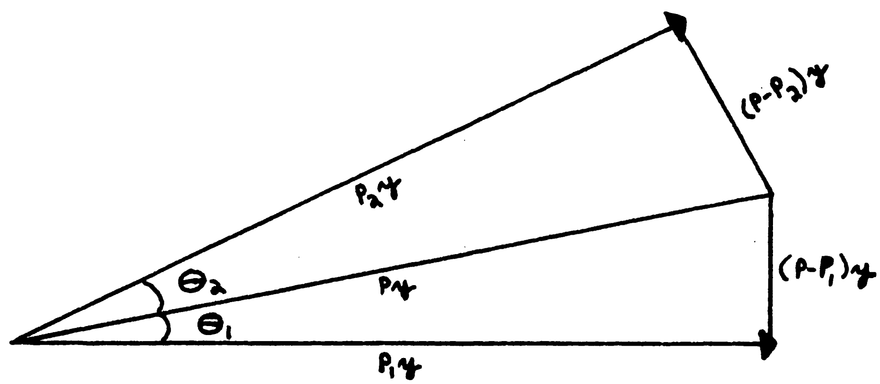


$\|\|P_1y\|\|^2$	1.00	N S	$\|\|P_2y\|\|^2$	14.00	N S
$\|\|(P-P_1)y\|\|^2$	25.60	Sig	$\|\|(P-P_2)y\|\|^2$	12.60	N S
$\|\|Py\|\|^2$	26.60		$\|\|Py\|\|^2$	26.60	
$\|\|(I-P)y\|\|^2$	9.00		$\|\|(I-P)y\|\|^2$	9.00	
$\|\|y\|\|^2$	35.60		$\|\|y\|\|^2$	35.60	

$$\cos\theta_1 = \frac{1}{\sqrt{26.60}} = .19389 \qquad \cos\theta_2 = \frac{\sqrt{14.00}}{\sqrt{26.60}} = .72548$$

$$\theta_1 \approx 78^\circ 50' \qquad \theta_2 \approx 43^\circ 30'$$

Figure 8. Situation Number 7

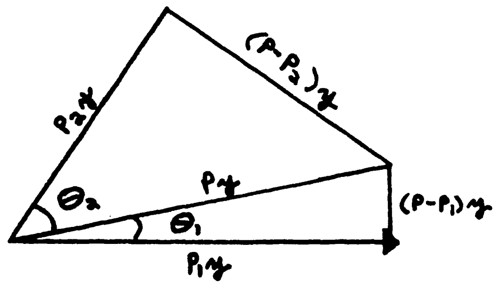


$\|\|P_1y\|\|^2$	100.00	Sig	$\|\|P_2y\|\|^2$	96.80	Sig
$\|\|(P-P_1)y\|\|^2$	4.00	N S	$\|\|(P-P_2)y\|\|^2$	7.20	N S
$\|\|Py\|\|^2$	104.00		$\|\|Py\|\|^2$	104.00	
$\|\|(I-P)y\|\|^2$	9.00		$\|\|(I-P)y\|\|^2$	9.00	
$\|\|y\|\|^2$	113.00		$\|\|y\|\|^2$	113.00	

$$\cos\theta_1 = \frac{10}{\sqrt{104}} = .98058 \qquad \cos\theta_2 = \frac{\sqrt{96.80}}{\sqrt{104.00}} = .96476$$

$$\theta_1 \approx 11^\circ 20' \qquad \theta_2 \approx 15^\circ 20'$$

Figure 9. Situation Number 8

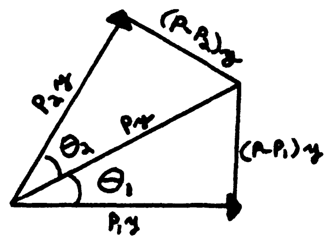


$\|\|P_1y\|\|^2$	25.00	Sig	$\|\|P_2y\|\|^2$	13.00	N S
$\|\|(P-P_1)y\|\|^2$	1.00	N S	$\|\|(P-P_2)y\|\|^2$	13.00	N S
$\|\|Py\|\|^2$	26.00		$\|\|Py\|\|^2$	26.00	
$\|\|(I-P)y\|\|^2$	9.00		$\|\|(I-P)y\|\|^2$	9.00	
$\|\|y\|\|^2$	35.00		$\|\|y\|\|^2$	35.00	

$$\cos \theta_1 = \frac{5}{\sqrt{26}} = .98058 \qquad \cos \theta_2 = \frac{\sqrt{13}}{\sqrt{26}} = \frac{\sqrt{2}}{2} = .70711$$

$$\theta_1 \approx 11^\circ 20' \qquad \theta_2 \approx 45^\circ$$

Figure 10. Situation Number 9



$\|\|P_1y\|\|^2$	9.00	N S	$\|\|P_2y\|\|^2$	8.50	N S
$\|\|(P-P_1)y\|\|^2$	2.56	N S	$\|\|(P-P_2)y\|\|^2$	3.06	N S
$\|\|Py\|\|^2$	11.56		$\|\|Py\|\|^2$	11.56	
$\|\|(I-P)y\|\|^2$	9.00		$\|\|(I-P)y\|\|^2$	9.00	
$\|\|y\|\|^2$	20.56		$\|\|y\|\|^2$	20.56	

$$\cos\theta_1 = \frac{3}{\sqrt{11.56}} = .88235 \qquad \cos\theta_2 = \frac{\sqrt{8.50}}{\sqrt{11.56}} = .85749$$

$$\theta_1 \approx 28^\circ \qquad \theta_2 \approx 31^\circ$$

Figure 11. Situation Number 10

Appendix B. Complete listing of NORTH (RANK 2).

```
C
C
C     MAIN PROGRAM OF NORTH (RANK 2)
C
C
C     COMMENT CARDS FOR MAIN PROGRAM
C
C
C          NORTH (RANK 2) WAS PRIMARILY WRITTEN TO  APPLY THE TECHNIQUES OF
C     THE PAPER "GRAPHICAL METHODS FOR NON-ORTHOGONAL DATA" (BY SAMUEL G.
C     LINDLE AND DAVID M. ALLEN, BU-555-M OF THE BIOMETRICS UNIT MIMEO SERIES,
C     CORNELL UNIVERSITY, ITHACA, N.Y. 14853). HOWEVER, A CERTAIN AMOUNT OF
C     FLEXIBILITY HAS BEEN ADDED TO MAKE IT USEFUL FOR OTHER PURPOSES.
C     ITS SPECIFIC APPLICATIONS TO THE PAPER WILL BE DISCUSSED FIRST.
C
C     SPECIFIC APPLICATION OF NORTH (RANK 2):
C
C          FOR A COMPLETE DESCRIPTION OF THE STATISTICAL MOTIVATION AND
C     THEORY SEE BU-555-M.
C          NORTH (RANK 2) USES THE SINGULAR VALUE DECOMPOSITION TO DECOMPOSE
C     THE N BY 3 MATRIX
C
C                                   -           -
C                              Z1=| P1Y P2Y PY |
C                                   -           -
C
C     DESCRIBED IN BU-555-M. (THE "1" AND "2" SHOULD BE SUBSCRIPTS, BUT
C     SUBSCRIPTS ARE NOT AVAILABLE ON IBM KEYPUNCHES.) THAT IS, IT COMPUTES
C     THE MATRICES U, S AND V SUCH THAT
C
C                                   Z1=U*S*VT
C
C     WHERE
C              U IS AN N BY 3 NATRIX SUCH THAT UT*U=I (UT=TRANSPOSE OF U)
C              V IS A 3 BY 3 MATRIX SUCH THAT VT*V=I (VT=TRANSPOSE OF V)
C         AND S IS A 3 BY 3 DIAGONAL MATRIX.
C          THE PROGRAM THEN CALCULATES THE NONZERO SUBMATRICES OF
C
C                          -                     -
C                         |  S*VT      UT*Y       |
C                      PZ=|                       |
C                         |  0         SQRT(SSE)  |
C                          -                     -
C
C     WHERE SSE IS THE SUM OF SQUARES FOR ERROR. THE THIRD ROW OF PZ IS DELETED
C     IN ORDER TO PROJECT 4 SPACE ONTO 3 SPACE. THE RESULTING MATRIX IS THEN
C     PREMULTIPLIED BY THE APPROPRIATE ELEMENTARY REFLECTOR IN ORDER TO
C     TRIANGULARIZE IT. IN ORDER TO KEEP THE RESULTANT VECTOR CORRESPONDING
C     TO THE ORIGINAL Y VECTOR IN THE FIRST OCTANT, IF EITHER OF THE TWO REMAIN-
C     ING ELEMENTS OF UT*Y IS NEGATIVE THE CORRESPONDING ROW IS MULTIPLIED BY
C     -1. SINCE IT IS DIFFICULT TO REFERENCE THE POSITION OF POINTS OUTSIDE THE
C     LIMITS OF THE GRID, IN ORDER TO KEEP ANY VECTORS FROM PROTRUDING
C     OUTSIDE THE GRID, IF EITHER THE (1,2) ELEMENT OR THE (2,3) ELEMENT
C     OR BOTH ARE NEGATIVE THEN THEIR VALUE IS SUBTRACTED FROM EVERY ELEMENT
C     IN THEIR ROW AND FROM THE CORRESPONDING ELEMENT OF THE ORIGIN. THIS
```

```
C      SHIFTS THE VECTORS APPROPRIATELY.
C           NEXT THE PROGRAM FINDS THE ELEMENT OF THE RESULTING MATRIX WITH
C      THE LARGEST ABSOLUTE VALUE (E) AND MULTIPLIES ALL THE ELEMENTS OF THE
C      MATRIX BY 10./E IN ORDER THAT ALL ELEMENTS WILL BE SCALED SO AS TO
C      HAVE ABSOLUTE MAGNITUDE LESS THAN OR EQUAL TO 10. THE COLUMNS OF THE
C      RESULTING MATRIX ARE THE TERMINAL POINTS OF THE VECTORS P1Y, P2Y,
C      PY AND Y. (IF NEITHER THE (1,2) ELEMENT NOR THE (2,3) ELEMENT ABOVE WAS
C      NEGATIVE, THE INITIAL POINT OF THESE VECTORS IS THE ORIGINAL ORIGIN.
C      OTHERWISE, THE INITIAL POINT IS THE APPROPRIATELY SHIFTED ORIGIN.)
C      FOR THE VECTORS (P-P1)Y, (P-P2)Y, (I-P1)Y, (I-P2)Y AND (I-P)Y
C      APPROPRIATE INITIAL AND TERMINAL POINTS ARE CHOSEN FROM THESE FOUR
C      VECTORS (P1Y, P2Y, PY AND Y). X1 AND X2 HAVE ARBITRARILY BEEN SET TO
C      TO BE EQUAL TO 1.1 TIMES P1Y AND P2Y (RESPECTIVELY). THIS CAN BE
C      EASILY MODIFIED IF DESIRED.
C           THE PROGRAM THEN USES SUBROUTINES SETUP, GRID, VECT01 AND P3D TO
C      DRAW THE GRID AND THE VECTORS AND/OR LINES DESIRED. DETAILED INFORMATION
C      CONCERNING THESE SUBROUTINES ARE CONTAINED IN THE COMMENT CARDS FOR
C      EACH. IN PARTICULAR, OPTIONS DESCRIBING THE PRESENCE OR ABSENCE OF
C      VECTORHEADS, THE TYPE OF VECTORHEADS, THE TYPE OF DARKENING IN OF
C      THE VECTORHEADS, THE MEASUREMENTS OF THE VECTORHEADS, THE PRESENCE
C      OR ABSENCE OF VECTOR LABELS, THE DISTANCE OF THE VECTOR LABELS ABOVE
C      (BELOW) THE LINE  OR VECTOR, AND MEASUREMENTS OF THE VECTOR LABEL
C      CAN BE OBTAINED FROM THE COMMENT CARDS OF VECR01. THE VARIABLES IN THE
C      MAIN PROGRAM CORRESPONDING TO THE PARAMETERS OF THE SUBROUTINES OR IN
C      COMMON WITH CERTAIN SUBROUTINES WILL BE DESCRIBED BELOW ALONG WITH
C      INFORMATION ON THE FORMAT OF INPUT READ IN BY THE MAIN PROGRAM OF
C      NORTH (RANK 2).
C
C      FLEXIBLE APPLICATION OF NORTH (RANK 2):
C
C           NORTH (RANK 2) CAN BE USED WITHOUT SUBROUTINE DSVD AND THE POINTS
C      FROM AND TO WHICH VECTORS OR LINES MAY BE DRAWN CAN BE DIRECTLY READ
C      IN. THIS ALLOWS A GREAT DEAL OF FLEXIBILITY SINCE THE VECTORS NEED
C      NOT HAVE ANY RELATIONSHIP TO ONE ANOTHER NOR DO THEY HAVE TO BE
C      RESTRICTED TO THE LIMITS OF THE GRID. LIMITS REQUIRING, FOR INSTANCE,
C      ALL THE VECTORS TO BE CONTAINED WITHIN A 10 BY 10 BY 10 CUBE CAN BE
C      IMPOSED IF DESIRED (SEE THE COMMENT CARDS FOR SETUP). ALL THE OPTIONS
C      PROVIDED PREVIOUSLY CONCERNING VECTORHEADS AND VECTOR LABELS ARE
C      ADDITIONALLY AVAILABLE FOR THE FLEXIBLE APPLICATION. THE LABELS MAY,
C      HOWEVER, NOT BE APPROPRIATE ONES SINCE THE LABELS WERE DEVELOPED FOR
C      THE SPECIFIC APPLICATION. IF SUCH IS THE CASE THE COMMENT CARDS FOR
C      VECT01 EXPLAIN HOW TO PRODUCE THE APPROPRIATE LABELS IF THEY ARE
C      DESIRED.
C           THE SUBROUTINES CAN ALSO BE SEPARATED FROM THE MAIN PROGRAM AND
C      USED FOR A VARIETY OF PURPOSES.
C
C
C      INPUT INTO NORTH (RANK 2) AND DESCRIPTION OF VARIABLES USED:
C
C      INPVEC DETERMINES WHICH TYPE APPLICATION OF NORTH (RANK 2) IS TO BE
C             USED FOR THE RUN. IF INPVEC=1, THE FLEXIBLE APPLICATION OF
C             THE PROGRAM IS TO BE USED. IF INPVEC=2, THE SPECIFIC APPLICATION
C             OF THE PROGRAM IS TO BE USED.
C
C      ISTERS DETERMINES WHETHER OR NOT THE PROGRAM WILL BE USED TO PRODUCE
```

```
C           PLOTS WHICH CAN BE PHOTOGRAPHICALLY REDUCED AND PLACED UNDER
C           A STEREOSCOPE IN ORDER TO ACCENTUATE DEPTH PERCEPTION. IF
C           ISTERS=1, THE PLOTS WILL NOT BE USED FOR THIS PURPOSE AND
C           REGULAR PLOTS WILL BE PRODUCED. IF ISTERS=2, THE SPECIAL PLOTS
C           NEEDED FOR STEREOSCOPIC VIEWING WILL BE PRODUCED. FOR FUTHER
C           INFORMATION SEE THE DISCUSSION OF ISTERS IN THE COMMENT CARDS
C           FOR SUBROUTINE SETUP.
C
C     F IS THE PARAMETER OF THE SYSTEM SUPPLIED SUBROUTINE FACTOR WHICH IS
C       USED IN ORDER TO ENLARGE OR REDUCE THE FIGURE(S) DRAWN ACCORDING
C       TO THE VALUE OF F. EVERY LINE DRAWN WILL BE ENLARGED (IF F IS GREATER
C       THAN OR EQUAL TO 1) OR REDUCED (IF F IS LESS THAN OR EQUAL TO 1)
C       IN LENGTH TO F TIMES THE LENGTH ORIGINALLY SPECIFIED.
C
C     NNN IS A SEQUENTIAL INDEX. NNN+1 IS THE NUMBER OF THE PLOT TO BE
C         DRAWN NEXT.
C
C     IREDUC IS ALWAYS ZERO UNLESS TWO SIZES OF PLOTS ARE TO BE DRAWN. (NO
C           MORE THAN TWO SIZES CAN BE DRAWN ON A RUN.) IT HAS NO REAL
C           EFFECT UNLESS IT BECOMES 1. IN WHICH CASE, IT ALLOWS ONE TO
C           AVOID DUPLICATING THE DATA CARD CONTAINING MM AND FROM EXCESSIVE
C           RESETTING OF THE PEN WHEN A SECOND SIZE IS REQUESTED.
C
C     MMAX, NMAX, M, N, WITHU, WITHV, S, U AND V ARE VARIABLES (ARRAYS)
C           USED IN THE MAIN PROGRAM THAT CORRESPOND TO THE PARAMETERS OF
C           SUBROUTINE DSVD OF THE SAME NAMES. THE ARRAY
C
C
C                         ZZ=|- P1Y P2Y PY Y -|
C                              -             -
C
C           OF THE MAIN PROGRAM CORRESPONDS TO THE ARRAY A OF DSVD.
C           LIKEWISE, L CORRESPONDS TO P. SEE COMMENT CARDS FOR DSVD FOR
C           THEIR DESCRIPTION.
C
C     VT IS THE TRANSPOSE OF V.
C
C     D EVENTUALLY BECOMES THE SQUARE ROOT OF THE SUM OF SQUARES FOR ERROR
C       AFTER TAKING ON SOME INTERMEDIATE VALUES DURING THE CALCULATIONS
C       USED TO OBTAIN THIS.
C
C     PZ IS CEFINED ABOVE.
C
C     T IS THE RESULT OF TRIANGULARIZING PZ BY PREMULTIPLTING BY THE
C       APPRCPRIATE ELEMENTARY REFLECTOR AND ALSO OF MULTIPLYING EACH
C       ROW WHOSE LAST ELEMENT IS NEGATIVE BY -1.
C
C     P IS THE VECTOR DEFINING THE ELEMENTARY REFLECTOR I-2*P*PT USED IN
C       TRIANGULARIZING PZ TO HELP OBTAIN T.
C
C     AS IS T SHIFTED (IF NECESSARY) AND SCALED (IF NECESSARY) AS DESCRIBED
C        IN THE COMMENT CARDS ABOVE.
C
C     A1 IS THE AMOUNT OF SHIFT (IF NECESSARY) FOR THE FIRST ROW OF T.
C
```

```
C      A2 IS THE AMOUNT OF SHIFT (IF NECESSARY) FOR THE SECOND ROW OF T.
C
C      ZZ, MMAX, NMAX, M, N, WITHU, WITHV, S, U, V, L, VT, D, PZ, T, P, AS,
C      A1 AND A2 ARE ONLY USED IF INPVEC=2.
C
C      X, Y AND Z ARE SINGLY DIMENSIONED ARRAYS. THEY SPECIFY THE COORDINATES
C                OF POINTS TO OR FROM WHICH VECTORS OR LINES MAY BE DRAWN.
C                LET PT(I)=(X(I),Y(I),Z(I)) BE THE I-TH SUCH POINT. THEN
C                LINES AND/OR VECTORS MAY BE DRAWN FROM PT(I) TO PT(J)
C                PROVIDED I IS NOT EQUAL TO J.
C
C      C IS A SINGLY DIMENSIONED ARRAY WHICH IS DESCRIBED IN THE COMMENT
C        CARDS OF SUBROUTINES SETUP AND P3D.
C
C      N CORRESPONDS TO THE PARAMETER OF THE SAME NAME IN SUBROUTINE GRID
C        AND IS DESCRIBED IN THE COMMENT CARDS FOR THAT SUBROUTINE.
C
C      ND IS THE NUMBER OF POINTS TO OR FROM WHICH VECTORS OR LINES MAY BE
C         DRAWN.
C
C      NV IS THE NUMBER OF VECTORS AND LINES TO BE DRAWN.
C
C      THE FOLLOWING VARIABLES IN THE MAIN PROGRAM CORRESPOND TO THE
C      PARAMETERS OF VECT01 ENCLOSED WITHIN PARANTHESIS AND ARE DISCUSSED
C      IN THE COMMENT CARDS OF THAT SUBROUTINE: S1(VHL), S2(VHHW), IAH(IVHT),
C      ICA(ITOK), DS1(DOS), D1(VLDFV), D2(VLH), N1(IIP), N2(ITP), N3(IVHP)
C      AND M3(IVLP).
C
C
C      THE FOLLOWING FIVE STATEMENTS DESCRIBE THE TYPE AND DIMENSION OF
C      VARIOUS VARIABLES AND ARRAYS USED IN THE MAIN PROGRAM OF NORTH (RANK 2):
       REAL*8 ZZ(10,4),U(10,4),S(4),V(4,4),VT(3,3),PZ(3,4),T(3,4),
      1PZ4(4,4),P(2),E,D,R
C      IF SPACE IS AT A PREMIUM OR IF MORE THAN 10 PARAMETERS ARE OF INTEREST
C      FOR ONE OR MORE PLOTS ON A RUN, THE TWO 10'S IN THE ABOVE REAL*8
C      STATEMENT SHOULD BE CHANGED TO EQUAL THE NUMBER OF PARAMETERS OF
C      INTEREST FOR THE PLOT THAT INVOLVES THE LARGEST NUMBER OF PARAMETERS
C      OF INTEREST IN THE CALCULATIONS.
       REAL*4 AS(3,4)
       LOGICAL WITHU,WITHV
       DIMENSION C(22),X(25),Y(25),Z(25)
C      IF SPACE IS AT A PREMIUM OR IF MORE THAN 25 OF THE POINTS PT(I)=
C      (X(I),Y(I),Z(I)) ARE NEEDED FOR ONE OR MORE PLOTS ON A RUN, THE
C      THREE 25'S IN THE ABOVE DIMENSION STATEMENT SHOULD BE CHANGED TO
C      EQUAL THE NUMBER OF POINTS USED FOR THE PLOT THAT REQUIRES THE
C      MOST POINTS.
       COMMON C,XPV1,YPV3
C      BOTH SPECIFIC AND FLEXIBLE APPLICATION
       CALL PLOTS(DABA,CABA)
       READ(5,191)INPVEC,ISTERS
 191   FORMAT(1X,I1,1X,I1)
       IREDUC=0
C      ENLARGEMENT OR REDUCTION OF PLOTS PRODUCED
       F=2.
 880   CALL FACTOR(F)
```

```
C
      NNN=0
      IF(INPVEC.EQ.1) GO TO 192
C     SPECIFIC APPLICATICN
C     SINGULAR VALUE DECOMPOSITICN
338   READ(5,100)MMAX,NMAX,M,N,L,WITHU,WITHV
100   FCRMAT(5I5,2L5)
      READ(5,29)((ZZ(I,J),J=1,4),I=1,M)
29    FCRMAT(4F10.5)
      WRITE(6,28)
28    FORMAT('0         _                 _ '/
     1        ' ZZ=|¯  P1Y P2Y PY Y ¯|'/
     2        '     -                 - '//)
      WRITE(6,10)((ZZ(I,J),J=1,4),I=1,M)
      CALL DSVD(ZZ,MMAX,NMAX,M,N,L,WITHU,WITHV,S,U,V)
C     SQUARED NORM OF Y
      E=0.
      DC 730 I=1,M
730   E=E+ZZ(I,4)**2
      D=E
C     TRANSPCSING V
      DO 733 I=1,3
      DC 733 J=1,3
733   VT(I,J)=V(J,I)
C     S*VT WITH THIRD ROW DELETED
      DO 734 I=1,2
      DO 734 J=1,3
734   PZ(I,J)=S(I)  *VT(I,J)
C     FIRST THREE ELEMENTS OF THE LAST ROW OF PZ ARE ZERO
      DO 735 J=1,3
735   PZ(3,J)=0.
C     UT*Y WITH THE ELEMENT IN THE THIRD ROW DELETED
      DO 736 I=1,2
736   PZ(I,4)=ZZ(I,4)
C     SQUARE ROOT OF SUM OF SQUARES FOR ERROR
      E=0.
      DO 737 I=1,2
737   E=E+ZZ(I,4)**2
      IF(S(3).LT..00001.AND.S(3).GT.-.00001) GO TO 52
      E=E+ZZ(3,4)**2
52    D=DSQRT(D-E)
      PZ(3,4)=D
      DO 41 I=1,2
      DO 41 J=1,4
41    PZ4(I,J)=PZ(I,J)
      DO 42 J=1,4
42    PZ4(4,J)=PZ(3,J)
      DO 43 J=1,3
43    PZ4(3,J)=S(3)*VT(3,J)
      PZ4(3,4)=ZZ(3,4)
      WRITE(6,31)
31    FCRMAT('0         _                           _ '/
     1        '      |¯ S*VT       UT*Y          ¯|'/
     2        ' PZ=|                              |'/
     3        '      |  0          SQRT(SSE)      |'/
```

```
     4        '        -                          - '//)
      WRITE(6,10)((PZ4(I,J),J=1,4),I=1,4)
      WRITE(6,38)
38    FCRMAT('0 DELETING THE THIRD ROW OF PZ'//)
      WRITE(6,10)((PZ(I,J),J=1,4),I=1,3)
      DO 738 I=1,3
      DO 738 J=1,4
738   T(I,J)=PZ(I,J)
C     PREMULTIPLICATICN BY AN ELEMENTARY REFLECTOR
      E=0.
      DC 90 I=1,2
90    E=E+T(I,1)**2
      E=E**.5
      IF(T(1,1).GE.0.)P(1)=T(1,1)+E
      IF(T(1,1).LT.0.)P(1)=T(1,1)-E
      P(2)=T(2,1)
      E=0.
      DO 4 J=1,2
4     E=E+P(J)**2
      DO 1 M=1,4
      R=0.
      DC 5 L=1,2
5     R=R+P(L)*T(L,M)
      R=2.*R/E
      DO 1 L=1,2
1     T(L,M)=T(L,M)-R*P(L)
      WRITE(6,32)
32    FORMAT('0 PZ PREMULTIPLIED BY APPROPRIATE ELEMENTARY REFLECTOR',
     1       ' IN ORDER TO TRIANGULARIZE IT'//)
      WRITE(6,10)((T(I,J),J=1,4),I=1,3)
C     FCRCES Y TO BE IN FIRST OCTANT
      IF(T(1,4).GE.0.) GO TO 739
      DC 740 J=1,4
740   T(1,J)=-T(1,J)
739   CONTINUE
      IF(T(2,4).GE.0.) GO TO 741
      DO 742 J=1,4
742   T(2,J)=-T(2,J)
741   CCNTINUE
      WRITE(6,33)
33    FORMAT('0 IF THE LAST ELEMENT OF A ROW IS NEGATIVE, IT IS',
     1       ' MULTIPLIED BY -1'//)
      WRITE(6,10)((T(I,J),J=1,4),I=1,3)
      DC 101 I=1,3
      DO 101 J=1,4
101   AS(I,J)=T(I,J)
C     SHIFTING
      IF(AS(1,2).GE.0.) GO TO 882
      A1=AS(1,2)
      DC 883 I=1,4
883   AS(1,I)=AS(1,I)-A1
882   IF(AS(2,3).GE.0.) GO TO 884
      A2=AS(2,3)
      DO 885 I=1,4
885   AS(2,I)=AS(2,I)-A2
```

```
      WRITE(6,34)
34    FORMAT('0 SHIFTING IN ORDER TO KEEP ALL VECTORS WITHIN THE',
     1       ' LIMITS OF THE GRID'//)
      WRITE(6,10)((AS(I,J),J=1,4),I=1,3)
C     FINDS THE ABSOLUTE VALUE OF THE ELEMENT WITH LARGEST ABSOLUTE VALUE
884   E=0.
      DO 301 I=1,3
      DO 301 J=1,4
      IF(ABS(AS(I,J)).GT.E)E=ABS(AS(I,J))
301      CONTINUE
C     SCALING
      DO 111 I=1,3
      DO 111 J=1,4
111   AS(I,J)=10./E*AS(I,J)
      WRITE(6,35)
35    FORMAT('0 SCALING TO MAKE ALL ELEMENTS HAVE ABSOLUTE VALUE LESS',
     1       ' THAN 10'//)
      DO 112 I=1,3
112   WRITE(6,10)(AS(I,J),J=1,4)
10    FORMAT(' ',4F15.5)
C     CREATES APPROPRIATE (X,Y,Z) COORDINATES OF THE INITIAL AND TERMINAL
C     POINTS OF THE LINES AND/OR VECTORS TO BE DRAWN
      X(1)=0.
      IF(A1.LT.0.) X(1)=-A1*10./E
      Y(1)=0.
      IF(A2.LT.0.) Y(1)=-A2*10./E
      Z(1)=0.
      X(2)=AS(1,1)
      Y(2)=AS(2,1)
      Z(2)=AS(3,1)
      X(3)=AS(1,1)*1.1
      Y(3)=AS(2,1)*1.1
      Z(3)=AS(3,1)*1.1
      X(4)=AS(1,2)
      Y(4)=AS(2,2)
      Z(4)=AS(3,2)
      X(5)=AS(1,2)*1.1
      Y(5)=AS(2,2)*1.1
      Z(5)=AS(3,2)*1.1
      X(6)=AS(1,3)
      Y(6)=AS(2,3)
      Z(6)=AS(3,3)
      X(7)=AS(1,4)
      Y(7)=AS(2,4)
      Z(7)=AS(3,4)
      WRITE(6,36)
36    FORMAT('0 THE ORIGIN, P1Y, X1, P2Y, X2, PY AND Y (RESPECTIVELY)',
     1       ' ARE:'//)
      WRITE(6,113)(X(I),Y(I),Z(I),I=1,7)
113   FORMAT(' ',3F15.5)
      IF(NNN.NE.0) GO TO 339
192   IF(IREDUC.EQ.1) GO TO 339
      IF(ISTERS.EQ.2) C(16)=0.
C     SETTING THE PEN AT ONE INCH FROM THE BOTTOM OF THE PLOTTING PAPER
      CALL PLOT(0.,-11.,-3)
```

```
      CALL PLOT(1.,1.,-3)
C     THE NUMBER OF PLOTS TO BE DRAWN ON THIS RUN
      READ(5,13)MM
13    FORMAT(I2)
339   IF(INPVEC.EQ.2) GO TO 340
C     FLEXIBLE APPLICATION
C     BEGINNING OF LOOP FOR FLEXIBLE APPLICATION
      DO 14 J=1,MM
      READ(5,20)(C(I),I=1,7),N,ND,NV,S1,S2,IAH,IDA,DS1,D1,D2
20    FCRMAT(7F7.4,3I3,2F4.2,I1,I1,F4.4,2F3.2)
      CALL SETUP
      CALL GRID(N)
      DO 40 I=1,ND
40    READ(5,30) X(I),Y(I),Z(I)
30    FORMAT(3F5.3)
      WRITE(6,37)
37    FCRMAT('0 POINTS THAT CAN BE USED AS INITIAL OR TERMINAL POINTS',
     1       ' OF LINES OR VECTORS'//)
      WRITE(6,113)(X(I),Y(I),Z(I),I=1,ND)
      DO 50 I=1,NV
      READ(5,60) N1,N2,N3,M3
60    FCRMAT(4I3)
50    CALL VECT01(N1,N2,X(N1),Y(N1),Z(N1),X(N2),Y(N2),Z(N2),N3,IAH,S1,
     1S2,IDA,DS1,M3,D1,D2)
      IF(ISTERS.EQ.1) CALL PLOT(11.0,0.0,-3)
      IF(ISTERS.EQ.2) CALL PLOT(11.4,0.0,-3)
14    CCNTINUE
C     END CF LOOP FOR FLEXIBLE APPLICATION
      GO TO 341
C     SPECIFIC APPLICATION
340   READ(5,20)(C(I),I=1,7),N,ND,NV,S1,S2,IAH,IDA,DS1,D1,D2
      CALL SETUP
      CALL GRID(N)
      DO 51 I=1,NV
      READ(5,60) N1,N2,N3,M3
51    CALL VECT01(N1,N2,X(N1),Y(N1),Z(N1),X(N2),Y(N2),Z(N2),N3,IAH,S1,
     1S2,IDA,DS1,M3,D1,D2)
      IF(ISTERS.EQ.1) CALL PLOT(11.0,0.0,-3)
      IF(ISTERS.EQ.2) CALL PLOT(11.4,0.0,-3)
      NNN=NNN+1
      IF(NNN.LT.MM)GO TO 338
341   CONTINUE
C     BOTH SPECIFIC AND FLEXIBLE APPLICATION
      IF(F.EQ.2.) GO TO 881
C     IF ONLY ONE SIZE OF PLOT IS WANTED, THEN MAKE THE VALUE AFTER .EQ.
C     EQUAL TO THE SIZE WANTED AND EQUAL TO THE VALUE PUNCHED AFTER F=
C     AT THE BEGINNING OF THE MAIN PROGRAM. TWO SIZES CAN BE PLOTED FOR
C     ALL PLOTS IF THE FIRST SIZE IS SPECIFIED AFTER THE F= STATEMENT
C     AT THE BEGINNING CF THE PROGRAM AND THE SECOND SIZE IS SPECIFIED
C     AFTER THE .EQ. ABOVE AND AFTER THE F= BELOW. THE NUMBER MM OF DIF-
C     FERENT PLOTS NEED NOT BE CHANGED BECAUSE THE PROGRAM CONSIDERS PLOTS
C     WHICH ARE DUPLICATE EXCEPT FOR SIZE TO BE THE SAME AND THUS MM NEED
C     NOT BE INCREMENTED FOR THEM. HOWEVER, THE APPROPRIATE DATA CARDS
C     MUST BE DUPLICATED. ALL THE PLOTS OF THE SECOND SIZE WILL FOLLOW
C     THE PLCTS OF THE FIRST SIZE.
```

```
      IREDUC=1
      F=1.
      GC TO 880
881   CALL PLTEND
      STOP
      END
C
C
      SUBROUTINE DSVD(A,MMAX,NMAX,M,N,P,WITHU,WITHV,S,U,V)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(MMAX,NMAX),U(MMAX,NMAX),V(NMAX,NMAX)
      DIMENSION S(N), B(100), C(100), T(100)
      INTEGER P
      LOGICAL WITHU,WITHV
C
C     THIS SUBROUTINE CCMPUTES THE SINGUALR VALUE DECOMPOSITION OF
C     A REAL M*N MATRIX A, I.E. IT CCMPUTES MATRICES U,S, AND V
C     SUCH THAT
C
C                  A = U * S * VT,
C     WHERE
C               U IS AN M*N MATRIX AND UT*U = I, (UT=TRANSPOSE
C                                                     OF U),
C               V IS AN N*N MATRIX AND VT*V = I, (VT=TRANSPOSE
C                                                     OF V),
C         AND   S IS AN N*N DIAGONAL MATRIX.
C
C     DESCRIPTION OF PARAMETERS:
C
C     A = REAL*8 ARRAY. A CONTAINS THE MATRIX TO BE DECOMPOSED.
C
C     MMAX = INTEGER*4 VARIABLE. THE NUMBER OF ROWS IN THE
C            ARRAYS A AND U.
C
C     NMAX = INTEGER*4 VARIABLE. THE NUMBER OF ROWS IN THE
C            ARRAY V.
C
C     M,N = INTEGER*4 VARIABLES. THE NUMBER OF ROWS AND COLUMNS
C           IN THE MATRIX STORED IN A.  (N<=M<=100.  IF IT IS
C           NECESSARY TO SOLVE A LARGER PROBLEM, THEN THE
C           AMOUNT OF STORAGE ALLOCATED TO THE ARRAYS B, C, AND
C           T MUST BE INCREASED ACCORDINGLY.)
C
C      P = INTEGER*4 VARIABLE.  IF P>0, THEN COLUMNS N+1, . . . ,
C           N+P OF A ARE ASSUMED TO CONTAIN THE COLUMNS OF AN M*P
C           MATRIX B.  THIS MATRIX IS MULTIPLIED BY UT, AND UPON
C           EXIT, A CONTAINS IN THESE SAME COLUMNS THE N*P MATRIX
C           UT*B. (P>0)
C
C     WITHU, WITHV = LOGICAL*4 VARIABLES.  IF WITHU=.TRUE., THEN
C         THE MATRIX U IS COMPUTED AND STORED IN THE ARRAY U.
C         SIMILARLY FOR V.
C
C     S = REAL*8 ARRAY.  S(1), . . . , S(N) CONTAIN THE DIAGONAL
C         ELEMENTS OF THE MATRIX S ORDERED SO THAT S(I)>=S(I+1),
```

```
C            I=1, . . . , N-1.
C
C      U,V = REAL*8 ARRAYS.  U,V CONTAIN THE MATRICES U AND V.
C           IF WITHU=.TRUE. AND WITHV=.FALSE., THEN THE ACTUAL
C          PARAMETER CORRESPONDING TO A AND U MAY BE THE SAME.
C          SIMILARLY FOR V IF WITHV=.TRUE. AND WITHU=.FALSE..
C
C      THIS SUBROUTINE IS A TRANSLATICN OF AN ALGOL 60 PROCEDURE
C      DESCRIBED IN THE ARTICLE "SINGUALR VALUE DECOMPOSITION AND
C      LEAST SQUARES SOLUTIONS", NUM. MATH. 14 (1970), PP. 403-420.
C      THE TRANSLATION WAS DONE BY P. BUSINGER AT BELL TELEPHONE
C          LABORATORIES WITH SOME CHANGES AND EDITING DONE BY R.
C      UNDERWOOD AT STANFORD UNIVERSITY.
C
C
       DATA ETA /Z3410000000000000/
       DATA TCL / Z0D10000000000000/
C
C      ETA AND TOL ARE MACHINE DEPENDENT CONSTANTS WHOSE
C      VALUES ARE 16**(-13) AND 16**(-52), RESPECTIVELY,
C      ON IBM SYSTEM/360 COMPUTERS.
C
C
       NP=N+P
       N1=N+1
C
C      HCUSE HOLDER REDUCTICN TO BIDIAGONAL FORM
       C(1)=0.0D0
       K=1
    10 K1=K+1
C
C      ELIMINATION OF A(I,K), I=K+1, . . . , M
       Z=0.0D0
       DO 20 I=K,M
    20    Z=Z+A(I,K)**2
       B(K)=0.0D0
       IF (Z.LE.TOL) GO TO 70
       Z=DSQRT(Z)
       B(K)=Z
       W=DABS(A(K,K))
       Q=1.0D0
       IF (W.NE.0.0D0) Q=A(K,K)/W
       A(K,K)=Q*(Z+W)
       IF (K.EQ.NP) GO TO 70
       DO 50 J=K1,NP
          Q=0.0D0
          DO 30 I=K,M
    30       Q=Q+A(I,K)*A(I,J)
          Q=Q/(Z*(Z+W))
          DO 40 I=K,M
    40       A(I,J)=A(I,J)-Q*A(I,K)
    50 CONTINUE
C
C      PHASE TRANSFORMATICN
       Q=-A(K,K)/DABS(A(K,K))
```

```
      DO 60 J=K1,NP
   60   A(K,J)=Q*A(K,J)
C
C     ELIMINATION OF A(K,J), J=K+2, . . . , N
   70 IF (K.EQ.N) GO TO 140
      Z=0.0D0
      DO 80 J=K1,N
   80   Z=Z+A(K,J)**2
      C(K1)=0.0D0
      IF (Z.LE.TOL) GO TO 130
      Z=DSQRT(Z)
      C(K1)=Z
      W=DABS(A(K,K1))
      Q=1.0D0
      IF (W.NE.0.0D0) Q=A(K,K1)/W
      A(K,K1)=Q*(Z+W)
      DO 110 I=K1,M
         Q=0.0D0
        DO 90 J=K1,N
   90       Q=Q+A(K,J)*A(I,J)
        Q=Q/(Z*(Z+W))
         DO 100 J=K1,N
  100       A(I,J)=A(I,J)-Q*A(K,J)
  110 CONTINUE
C
C     PHASE TRANSFORMATION
      Q=-A(K,K1)/DABS(A(K,K1))
      DO 120 I=K1,M
  120   A(I,K1)=A(I,K1)*Q
C
  130 K=K1
      GO TO 10
C
C     TOLERANCE FOR NEGLIGIBLE ELEMENTS
  140 EPS=0.0D0
      DO 150 K=1,N
        S(K)=B(K)
        T(K)=C(K)
  150   EPS=DMAX1(EPS,S(K)+T(K))
      EPS=EPS*ETA
C
C     INITIALIZATION OF U AND V
      IF (.NOT.WITHU) GO TO 180
      DO 170 J=1,N
        DO 160 I=1,M
  160       U(I,J)=0.0D0
  170   U(J,J)=1.0D0
C
  180 IF (.NOT.WITHV) GO TO 210
      DO 200 J=1,N
        DO 190 I=1,N
  190       V(I,J)=0.0D0
  200   V(J,J)=1.0D0
C
C     QR DIAGONALIZATION
```

```
  210 DO 380 KK=1,N
        K=N1-KK
C
C     TEST FOR SPLIT
  220   DO 230 LL=1,K
            L=K+1-LL
            IF (DABS(T(L)).LE.EPS) GO TO 290
            IF (DABS(S(L-1)).LE.EPS) GO TO 240
  230       CONTINUE
C
C     CANCELLATION
  240   CS=0.0D0
        SN=1.0D0
        L1=L-1
        DO 280 I=L,K
            F=SN*T(I)
            T(I)=CS*T(I)
            IF (DABS(F).LE.EPS) GO TO 290
            H=S(I)
            W=DSQRT(F*F+H*H)
            S(I)=W
            CS=H/W
            SN=-F/W
            IF (.NOT.WITHU) GO TO 260
            DO 250 J=1,N
               X=U(J,L1)
               Y=U(J,I)
               U(J,L1)=X*CS+Y*SN
  250          U(J,I)=Y*CS-X*SN
  260          IF (NP.EQ.N) GO TO 280
            DO 270 J=N1,NP
               Q=A(L1,J)
               R=A(I,J)
               A(L1,J)=Q*CS+R*SN
  270          A(I,J)=R*CS-Q*SN
  280       CONTINUE
C
C     TEST FOR CONVERGENCE
  290   W=S(K)
        IF (L.EQ.K) GO TO 360
C
C
C     ORIGIN SHIFT
        X=S(L)
        Y=S(K-1)
        G=T(K-1)
        H=T(K)
        F=((Y-W)*(Y+W)+(G-H)*(G+H))/(2.0D0*H*Y)
        G=DSQRT(F*F+1.0D0)
        IF (F.LT.0.0D0) G=-G
        F=((X-W)*(X+W)+(Y/(F+G)-H)*H)/X
C
C     QR STEP
        CS=1.0D0
        SN=1.0D0
```

```
      L1=L+1
      DO 350 I=L1,K
          G=T(I)
          Y=S(I)
          H=SN*G
          G=CS*G
          W=DSQRT(H*H+F*F)
          T(I-1)=W
          CS=F/W
          SN=H/W
          F=X*CS+G*SN
          G=G*CS-X*SN
          H=Y*SN
          Y=Y*CS
          IF (.NOT.WITHV) GO TO 310
          DO 300 J=1,N
             X=V(J,I-1)
             W=V(J,I)
             V(J,I-1)=X*CS+W*SN
  300        V(J,I)=W*CS-X*SN
  310     W=DSQRT(H*H+F*F)
          S(I-1)=W
          CS=F/W
          SN=H/W
          F=CS*G+SN*Y
          X=CS*Y-SN*G
          IF (.NOT.WITHU) GO TO 330
          DO 320 J=1,N
                Y=U(J,I-1)
          W=U(J,I)
         U(J,I-1)=Y*CS+W*SN
  320     U(J,I)=W*CS-Y*SN
  330     IF (N.EQ.NP) GO TO 350
          DO 340 J=N1,NP
             Q=A(I-1,J)
             R=A(I,J)
             A(I-1,J)=Q*CS+R*SN
  340        A(I,J)=R*CS-Q*SN
  350 CCNTINUE
C
       T(L)=0.0D0
      T(K)=F
      S(K)=X
      GO TO 220
C
C
C     CONVERGENCE
  360  IF (W.GE.0.0D0) GO TO 380
       S(K)=-W
       IF (.NOT.WITHV) GO TO 380
      DC 370 J=1,N
  370     V(J,K)=-V(J,K)
  380  CONTINUE
C
C      SORT SINGULAR VALUES
```

```
      DO 450 K=1,N
         G=-1.0D0
         J=K
         DO 390 I=K,N
            IF (S(I).LE.G) GO TO 390
            G=S(I)
            J=I
  390       CONTINUE
         IF (J.EQ.K) GO TO 450
         S(J)=S(K)
         S(K)=G
         IF (.NOT.WITHV) GO TO 410
         DO 400 I=1,N
            Q=V(I,J)
         V(I,J)=V(I,K)
  400       V(I,K)=Q
  410 IF (.NOT.WITHU) GO TO 430
         DO 420 I=1,N
            Q=U(I,J)
            U(I,J)=U(I,K)
  420       U(I,K)=Q
  430    IF (N.EQ.NP) GO TO 450
         DO 440 I=N1,NP
            Q=A(J,I)
            A(J,I)=A(K,I)
  440          A(K,I)=Q
  450    CONTINUE
C
C     BACK TRANSFORMATION
      IF (.NOT.WITHU) GO TO 510
      DO 500 KK=1,N
         K=N1-KK
         IF (B(K).EQ.0.0D0) GO TO 500
         Q=-A(K,K)/DABS(A(K,K))
         DO 460 J=1,N
  460       U(K,J)=Q*U(K,J)
         DO 490 J=1,N
            Q=0.0D0
            DO 470 I=K,M
  470          Q=Q+A(I,K)*U(I,J)
            Q=Q/(DABS(A(K,K))*B(K))
            DO 480 I=K,M
  480          U(I,J)=U(I,J)-Q*A(I,K)
  490        CONTINUE
  500 CONTINUE
C
  510 IF (.NOT.WITHV) GO TO 570
      IF (N.LT.2) GO TO 570
      DO 560 KK=2,N
         K=N1-KK
         K1=K+1
         IF (C(K1).EQ.0.0D0) GO TO 560
         Q=-A(K,K1)/DABS(A(K,K1))
         DO 520 J=1,N
  520          V(K1,J)=Q*V(K1,J)
```

```
        DO 550 J=1,N
            Q=0.0D0
            DO 530 I=K1,N
  530                Q=Q+A(K,I)*V(I,J)
            C=Q/(DABS(A(K,K1))*C(K1))
            DO 540 I=K1,N
  540          V(I,J)=V(I,J)-Q*A(K,I)
  550       CONTINUE
  560   CONTINUE
C
  570 RETURN
        END
C
C
      SUBRCUTINE SETUP
      COMMON C(22),XPV1,YPV3
C
C
C     INPUT INTO THIS SUBRCUTINE VIA THE COMMON STATEMENT CONNECTING IT
C     WITH THE MAIN PROGRAM OF NORTH (RANK 2):
C
C          THE FOCAL PCINT: FP=(C(1),C(2),C(3))
C
C          THE CBSERVATION DIRECTION VECTOR: ODV=(C(4),C(5),C(6))
C
C          THE DISTANCE BETWEEN THE OBSERVATICN POINT AND THE FOCAL
C          PCINT: C(7)
C          C(7) SHOULD ALWAYS BE GREATER THAN ZERO.
C
C          THE CBSERVATION POINT: FP+C*ODV WHERE C=C(7)/NORM OF ODV
C
C     THIS SUBROUTINE DEFINES C(8) THRU C(20).
C
C        C(8) THRU C(15) ARE DISPLAYED ROWWISE AS THE NON-ZERO ELEMENTS
C     OF THE MATRIX P:
C
C                    _                                            _
C                   | -C(5)/R              C(4)/R             0     |
C                   |                                                |
C                   | -C(4)/S              -C(5)/S            -C(6)/S |
C                   |                                                |
C                   | -C(4)*C(5)/(R*S)     -C(5)*C(6)/(R*S)   R/S    |
C                    -                                            -
C
C     WHERE R=SQRT(C(4)**2+C(5)**2) AND S IS THE NORM OF THE
C     CBSERVATION DIRECTION VECTOR.
C
C        P IS THE PRODUCT OF TWO ASYMMETRIC PLANE ROTATIONS:
C
C            _                            _        _                     _
C           | 1        0         0         |      | -C(5)/R   C(4)/R    0 |
C           |                              |      |                       |
C           | 0        R/S       -C(6)/S   |   *  | -C(4)/R   -C(5)/R   0 |
C           |                              |      |                       |
C           | 0        C(6)/S    R/S       |      | 0         0         1 |
```

```
C        -                         -        -                  -
C
C      PREMULTIPLICATION OF THE OBSERVATION DIRECTION VECTOR BY THE
C      SECOND ASYMMETRIC PLANE ROTATION WOULD PRODUCE THE VECTOR:
C
C                                 (0,-R,C(6))
C      AS DESIRED. SUBSEQUENTLY PREMULTIPLYING BY THE FIRST ASYMMETRIC
C      PLANE ROTATION WOULD PRODUCE THE VECTOR:
C
C                                 (0,-S,0)
C
C      AS DESIRED. EQUIVALENTLY PREMULTIPLICATION OF THE OBSERVATION POINT
C      (OP) BY P WILL GIVE THE VECTOR:
C
C                                 (0,-C*S,0)
C
C      PROVIDED THE FOCAL POINT IS USED AS THE ORIGIN. (THE FOCAL POINT
C      IS MADE THE NEW ORIGIN IN THE TRANSLATE (FIRST) STEP OF P3D BEFORE
C      THE P MATRIX IS USED IN THE ROTATION (SECOND) STEP. ALSO NOTE THAT
C      THE OBSERVATION DIRECTION VECTOR HAS FIRST BEEN ROTATED CLOCKWISE
C      IN THE XY PLANE D=ARCCOS(-C(5)/R)=ARCSIN(C(4)/R) DEGREES AND THEN
C      ROTATED CLOCKWISE IN THE YZ PLANE M1=ARCCOS(R/S)=ARCSIN(-C(6)/S)
C      CEGREES. ANY VECTOR PREMULTIPLIED IN THIS WAY WOULD BE
C      SIMILIARLY ROTATEC.) THE MATRIX P SIMULTANEOUSLY PREFORMS THE
C      TASK OF THE TWO ASYMMETRIC PLANE ROTATIONS.
C
C           WHEN C(4) AND C(5) ARE BOTH ZERO AND C(6) IS NONZERO THE
C      ELEMENTS OF P ARE CREATED IN A DIFFERENT PORTION OF THE SUBROUTINE
C      TO CORRESPOND TO:
C
C               -                                      -
C              | 1                0                    0 |
C              |                                         |
C              | 0                0                   -1 |
C              |                                         |
C              | 0                1                    0 |
C               -                                      -
C
C           THE MATRIX GIVEN ABOVE ROTATES THE Y AND Z COORDINATES FOR
C      EVERY POINT 270 DEGREES CLOCKWISE OR 90 DEGREES COUNTER-CLOCKWISE.
C      (NOTE THAT IN THIS SPECIAL CASE THE SECOND ASYMMETRIC PLANE
C      RCTATICN ABOVE IS THE IDENTITY MATRIX AND THE FIRST ASYMMETRIC
C      PLANE ROTATION IS JUST THE MATRIX ABOVE.) ALSO, ODV=(0,0,C(6))
C      SC THAT PREMULTIPLYING ODV BY P GIVES (0,-C(6),0). SINCE S IS THE
C      ABSOLUTE VALUE OF C(6), IF C(6) IS GREATER THAN 0, THIS POINT BECOMES
C      (0,-S,0) AS ABOVE. IF C(6) IS LESS THAN ZERO, THE POINT BECOMES
C      (C,S,0) WHICH IS NOT DESIRABLE. (THE CASE WHERE C(6) IS EQUAL
C      TO ZERO IS COVERED BELOW.) THUS, IF C(4) AND C(5) ARE BOTH EQUAL TO
C      ZERC, THEN C(6) SHOULD BE A POSITIVE NUMBER.(IN SUCH A CASE, IT
C      IS ALSO WISE TO MAKE C(7) LARGE BECAUSE THIS CAUSES THE PLOT
C      TO FILL THE PAGE. THEORETICALLY, WHEN C(7) IS EQUAL TO INFINITY
C      MAXIMUM FILLING UP OF THE PAGE OCCURS. PRACTICALLY, ANY MODERATELY LARGE
C      NUMBER WILL DO.) HAVING C(4) AND C(5) BOTH ZERO ALLOWS ONE TO LOOK
C      STRAIGHT DCWN ONTO THE GRID AND TO PRODUCE TWO DIMENSIONAL PLOTS.
C           WHEN C(4), C(5) AND C(6) ARE ALL ZERO (THE ODV IS THE ZERO
```

```
C       VECTOR), THE MESSAGE "THE OBSERVATION POINT IS UNDEFINED SINCE
C       IT IS IMPOSSIBLE TO MOVE FROM THE FOCAL POINT A POSITIVE DISTANCE
C       IN THE DIRECTION (0,0,0)." IS PRINTED OUT. THE PLOTTAPE IS ENDED
C       AND THE PROGRAM IS TERMINATED.
C            EXCEPT IN THE CASE COVERED IMMEDIATELY ABOVE WHEN THE ODV IS
C       THE ZERO VECTOR OR IN THE CASE MENTIONED IN THE COMMENT CARDS BELOW
C       WHEN ISTERS=2 AND C(16) IS NONZERO, THE LIMITING POINTS READ IN BY
C       SETUP ABOVE (WHICH, FOR EXAMPLE, MIGHT BE THE VERTICES OF A 10 BY 10
C       BY 10 CUBE BUT IN GENERAL ARE VERTICES OF SOME FIGURE) ARE PROJECTED
C       ONTO TWO SPACE BY P3D WHICH MUST THUS ACCOMPANY SETUP IN THE SOURCE
C       DECK. C(16) AND C(18) BECOME THE MINIMUM X AND Y COORDINATES
C       (RESPECTIVELY) OF THE PROJECTIONS. C(17) AND C(19) BECOME THE
C       RANGE OF THE X AND Y COORDINATES OF THE PROJECTIONS.
C
C            C(20) IS ORIGINALLY ZERO IN ORDER THAT NEITHER PLOTTING
C       NOR SCALING OCCURS IN SUBROUTINE P3D WHEN THE BOUNDARY LIMITATIONS
C       ARE BEING CREATED (OR SETUP). AFTER THIS IS COMPLETED AT THE
C       END OF SETUP, C(20) IS SET TO ONE SO THAT HEREAFTER P3D WILL
C       SCALE AND PLOT. C(21) AND C(22) ARE CREATED IN SUBROUTINE P3D
C       EVERY TIME IT IS CALLED.
C
C
        C(20)=0.
        R=C(4)**2+C(5)**2
        S=R+C(6)**2
        IF(R.EQ.0) GO TO 6
        R=SQRT(R)
        S=SQRT(S)
        C(8 )=-C(5)/R
        C(9 )= C(4)/R
        C(10)=-C(4)/S
        C(11)=-C(5)/S
        C(12)=-C(6)/S
        C(13)=-C(4)*C(6)/(R*S)
        C(14)=-C(5)*C(6)/(R*S)
        C(15)= R/S
        GO TO 8
6       IF(S.EQ.0.)GO TO 7
        WRITE(6,112)
112     FORMAT(' R IS ZERO.')
        C(8)= 1.
        C(9)= 0.
        C(10)= 0.
        C(11)= 0.
        C(12)=-1.
        C(13)= 0.
        C(14)= 1.
        C(15)= 0.
C
C
C       IF ISTERS=1 IN THE MAIN PROGRAM OF NORTH (RANK 2) USE THE CARD:
C
C 8     C(16)=10.**(70)
C
C       ISTERS=1 IS THE USUAL NONSTEREOSCOPIC USE OF NORTH (RANK 2)
```

```
C      WHICH REQUIRES THAT C(16) THRU C(19) BE RECALCULATED FOR EACH
C      PLOT IF OPTIMALLY SIZED PLOTS ARE TO BE OBTAINED. THE ABOVE
C      FORTRAN STATEMENT (IF THE 'C' FOR MAKING IT A COMMENT CARD IS
C      DELETED) ALONG WITH ANOTHER STATEMENT DESCRIBED IN THE COMMENT
C      CARDS NEAR THE END OF SETUP CAN BE USED TO ACCOMPLISH THIS.
C
C      IF ISTERS=2 USE THE CARDS:
C
C 8    IF(C(16).NE.0.) GO TO 10
C      C(16)=10.**(70)
C
C      FOR ISTERS=2, ANY APPROPRIATE (SEE COMMENT CARDS IN MAIN PROGRAM
C      OF NORTH (RANK 2)) PAIR OF THE PLOTS PRODUCED BY NORTH (RANK 2)
C      CAN BE PHOTOGRAPHICALLY REDUCED AND PLACED UNDER A STEREOSCOPE TO
C      ACCENTUATE DEPTH PERCEPTION. TO PRODUCE THE APPROPRIATE PLOTS
C      C(16) THRU C(19) MUST NOT BE REDEFINED ON EACH PLOT. RATHER,
C      THE VALUES CALCULATED FOR THEM ON THE FIRST PLOT MUST BE RETAINED
C      THROUGHOUT.THIS IS ACCOMPLISHED BY THE ABOVE TWO FORTRAN STATE-
C      MENTS ALONG WITH ANOTHER STATEMENT DESCRIBED IN THE COMMENT CARDS
C      NEAR THE END OF SETUP. THE PURPOSE OF NOT REDEFINING C(16) THRU C(19)
C      IS THAT THIS GUARANTIES THAT THE PICTURE WILL BE OF A DIFFERENT
C      SIZE WHEN VIEWED FROM DIFFERENT OBSERVATION POINTS. RETAINING C(16)
C      THRU C(19) DOES NOT ALLOW THE PICTURES TO BE BLOWN UP DIFFERENTLY.
C      DEPTH PERCEPTION IS ACCENTUATED WHEN THE DISTANCE BETWEEN THE TWO
C      OBSERVATION POINTS IS SOMEWHAT LARGER THAN THE INTEROCULAR DISTANCE.
C      THE INTEROCULAR DISTANCE FOR THE AVERAGE ADULT IS ABOUT 2.5 INCHES
C      (63 MILLIMETERS).
C
C
8      C(16)=10.**(70)
       C(17)=-C(16)
       C(18)=C(16)
       C(19)=C(17)
C      LIMITING POINTS
2      READ(5,3) X,Y,Z,TRIP
3      FORMAT(4F5.3)
C
       IF(TRIP.NE.0.) GO TO 1
       CALL P3D(X,Y,Z,3)
       IF(C(21).LT.C(16)) C(16)=C(21)
       IF(C(21).GT.C(17)) C(17)=C(21)
       IF(C(22).LT.C(18)) C(18)=C(22)
       IF(C(22).GT.C(19)) C(19)=C(22)
       GC TC 2
1      CCNTINUE
       C(17)=C(17)-C(16)
       IF(C(17).EQ.0.) GO TO 6
       C(19)=C(19)-C(18)
       IF(C(19).EQ.0.) GO TO 6
C
C
C      IF ISTERS=1 IN THE MAIN PROGRAM OF NORTH (RANK 2) USE THE CARD:
C
C      C(20)=1.0
C
```

```
C     IF ISTERS=2 USE THE CARD:
C
C 10  C(20)=1.0
C
C
      C(20)=1.0
      RETURN
7     WRITE(6,9)
9     FORMAT(' THE OBSERVATION POINT IS UNDEFINED SINCE IF IS IMPOSSIBL'
     1,'E TO MOVE FROM THE FOCAL POINT A POSITIVE DISTANCE IN THE DIREC'
     2,'TION (0,0,0).')
      CALL PLTEND
      STOP
      END
C
C
      SUBROUTINE GRID(N)
C
C
C     THIS SUBROUTINE DRAWS A PERSPECTIVE GRID TO HELP IDENTIFY THE
C     POSITIONS OF POINTS ABOVE, BELOW OR IN THE PLANE OF THE GRID.
C     SUBROUTINE P3D MUST ALWAYS ACCOMPANY SUBROUTINE GRID IN THE
C     FORTRAN SOURCE DECK.
C
C     DESCRIPTION OF THE SYMBOLS USED IN SUBROUTINE GRID:
C
C     N = INTEGER*4 VARIABLE. IT IS THE ONLY PARAMETER OF SUBROUTINE
C         GRID AND THE ONLY INPUT FROM THE CALLING PROGRAM SINCE THERE
C         IS NO COMMON STATEMENT. IT IS THE TOTAL NUMBER OF DASHES AND
C         SPACES BETWEEN DASHES ON EACH LINE OF THE GRID TO BE DRAWN.
C
C     K = INTEGER*4 VARIABLE. IT SPECIFIES WETHER THE PEN WILL BE UP
C         OR DOWN WHILE PLOTTING. K IS ALWAYS EITHER 2 OR 3. IF K=2 THE
C         PEN IS DOWN WHILE PLOTTING AND IF K=3 IT IS UP WHILE PLOTTING.
C
C     L = INTEGER*4 VARIABLE. IT INDEXES THE LOOP THAT PROVIDES THE TWO
C         STEPS FOR THE GRID. IN STEP 1 (L=1) ELEVEN "VERTICAL" DASHED
C         LINES ARE PRODUCED. IN STEP 2 (L=2) ELEVEN "HORIZONTAL" DASHED
C         LINES ARE PRODUCED.
C
C     X1,Y1,X2,Y2 ARE REAL*4 VARIABLES WHICH ARE THE COORDINATES OF THE
C                 POINTS (X1,Y1) AND (X2,Y2) WHICH ARE USED IN
C                 INTERMEDIATE STEPS TO CALCULATE X AN Y.
C
C     X AND Y ARE REAL*4 VARIABLES WHICH ARE THE COORDINATES OF THE
C             POINT (X,Y) WHICH DETERMINES VIA SUBROUTINE P3D THE
C             DIRECTION AND LENTH OF EACH DASH AND SPACE BETWEEN DASH
C             WHICH IS TO BE PLOTTED.
C
C          TWO STEPS ARE INVOLVED IN DRAWING THE GRID. THE FIRST STEP
C     CAN BE DESCRIBED AS FOLLOWS: BEGINNING WITH THE PEN IN THE DOWN
C     POSITION AND BY ALTERNATING THE UP AND DOWN POSITION OF THE PEN
C     SUBROUTINE GRID, AIDED BY SUBROUTINE P3D, DRAWS A DASHED LINE
C     FROM THE POINT ON THE PLOTTER PAPER CORRESPONDING TO THE ORIGIN
C     TO THE POINT CORRESPONDING TO (0,10,0). THIS DASHED LINE CONTAINS
```

```
C       THE GREATEST INTEGRAL VALUE OF N/2 PLUS 1 DASHES AND N MINUS GREATEST
C       INTEGRAL VALUE OF N/2 PLUS 1 SPACES BETWEEN THE DASHES. THE LENGTHS AND
C       DIRECTICNS OF THE CASHES AND THE SPACES BETWEEN THE DASHES DEPEND
C       ON THEIR RESPECTIVE DISTANCE AND DIRECTION FROM THE OBSERVATION
C       PCINT AS WELL AS ON SCALING FACTORS FOR THE X AND Y AXES. THEY
C       ARE DETERMINED BY P3D. THEN, WITH THE PEN UP THE PLOTTER PEN SKIPS
C       OVER TO THE POINT CORRESPONDING TO (1,10,0). AGAIN BEGINNING WITH
C       PEN DOWN AND BY ALTERNATING THE UP AND DOWN POSITION OF THE PEN
C       A CASHED LINE SIMILAR TO THE FIRST ONE IS DRAWN FROM THE POINT
C       CORRESPONDING TO (1,10,0) TO THE POINT CORRESPONDING TO (1,0,0).
C       WITH THE PEN UP ,THE PLOTTER PEN SKIPS OVER TO THE POINT
C       CORRESPCNDING TO THE POINT (2,0,0). THEN A DASHED LINE
C       SIMILAR TO THE FIRST TWO IS DRAWN FROM THE POINT CORRESPONDING
C       TO (2,0,0) TO THE POINT CORRESPONDING TO (2,10,0).
C       THE PROCESS CONTINUES IN THIS WAY UNTIL ELEVEN "VERTICAL"
C       CASHED LINES EACH WITH THE GREATEST INTEGRAL VALUE OF N/2 PLUS 1 DASHES
C       ARE DRAWN. THESE LINES CORRESPCND TO PERSPECTIVE VIEWING OF LINES
C       PARALLEL TO THE Y AXIS THAT ARE EQUALLY SPACED.
C             THE SECOND STEP (L=2) IS IDENTICAL TO THE FIRST EXCEPT THAT
C       THE PROGRAM DRAWS "HORIZONTAL" LINES INSTEAD OF "VERTICAL" LINES.
C       THE PEN STARTS IN THE DOWN POSITION CORRESPONDING TO THE ORIGIN AND
C       DRAWS A DASHED LINE TO THE POINT CORRESPONDING TO (10,0,0). THEN
C       IT SIMILARLY DRAWS DASHED LINES AS FOLLOWS: FROM THE POINT COR-
C       RESPONCING TO (10,1,0) TO THE POINT CORRESPONDING TO (0,1,0); FROM
C       THE POINT CORRESPONDING TO (0,2,0) TO THE POINT CORRESPONDING TO
C       (10,2,0) AND SO ON UNTIL ELEVEN "HORIZONTAL" DASHED LINES ARE
C       DRAWN EACH WITH THE GREATEST INTEGRAL VALUE OF N/2 PLUS 1 DASHES.
C
C
        K=2
        DO 1 L=1,2
        K=5-K
        X1=0.
        Y1=0.
        CALL P3D(0.,0.,0.,3)
        DC 1 I=1,11
        X2=I-1
        Y2=10.-Y1
        DC 2 J=1,N
        K=5-K
        X=((N-J)*X1+J*X2)/N
        Y=((N-J)*Y1+J*Y2)/N
        IF(L.EQ.1) CALL P3D(X,Y,0.,K)
        IF(L.EQ.2) CALL P3D(Y,X,0.,K)
      2 CCNTINUE
        K=5-K
        X1=X1+1
        Y1=Y2
        IF(L.EQ.1) CALL P3D(X1,Y1,0.,3)
        IF(L.EC.2) CALL P3D(Y1,X1,0.,3)
      1 CONTINUE
        RETURN
        END
C
C
```

```
      SUBROUTINE VECT01(IIP,ITP,XIP,YIP,ZIP,XTP,YTP,ZTP,IVHP,IVHT,VHL,  NRT30001
     1VHHW,ITDK,DOS,IVLP,VLDFV,VLH)                                     NRT30002
      DIMENSION XSI(11,17),YSI(11,17)                                   NRT30003
      COMMON C(22),XPV1,YPV3                                            NRT30004
      REAL*8 NDV,NPDV                                                   NRT30005
      INTEGER SN                                                        NRT30006
C
C
C
C     SUBROUTINE VECT01 IS USED TO DRAW LINES AND/OR VECTORS ON OR ABOVE THE
C     GRID PRODUCED BY SUBROUTINE GRID. IN CERTAIN CASES APPROPRIATE LABELS
C     ARE OPTIONALLY AVAILABLE FOR THE LINES AND/OR VECTORS DRAWN. ADD-
C     ITIONAL DATA CARDS CAN BE PROVIDED IF OTHER LABELS THAN THOSE
C     SPECIFIED ARE DESIRED. SUBROUTINES SETUP AND P3D MUST ACCOMPANY
C     VECT01 IN THE FORTRAN SOURCE DECK. VECT01 CAN BE USED WITH OR
C     WITHOUT SUBROUTINE GRID. A PERSPECTIVE GRID IS, HOWEVER, OFTEN
C     HELPFUL IN IDENTIFYING THE POSITIONS OF THE POINTS OF INTEREST.
C
C
C     DESCRIPTION OF THE PARAMETERS OF SUBROUTINE VECT01:
C
C          IIP,ITP,IVHP,IVHT,ITDK AND IVLP ARE INTEGER*4 VARIABLES. XIP,
C          YIP,ZIP,XTP,YTP,ZTP,VHL,VHHW,DOS,VLDFV AND VLH ARE REAL*4
C          VARIABLES.
C
C          INDEX FOR INITIAL POINT OF LINE OR VECTOR: IIP
C
C          INDEX FOR TERMINAL POINT OF LINE OR VECTOR: ITP
C
C          INITIAL POINT OF LINE OR VECTOR: IP=(XIP,YIP,ZIP)
C
C          TERMINAL POINT OF LINE OR VECTOR: TP=(XTP,YTP,ZTP)
C
C          INDICATOR OF THE PRESENCE (OR ABSENCE) OF A VECTORHEAD: IVHP
C
C          INDICATOR OF THE TYPE OF VECTORHEAD TO BE USED: IVHT
C
C          LENGTH OF THE VECTORHEAD IN INCHES: VHL*F
C          (F IS THE PARAMETER OF THE SYSTEM SUPPLIED SUBROUTINE FACTOR
C          WHICH IS CALLED IN THE MAIN PROGRAM OF NORTH (RANK 2) IN ORDER
C          TO ENLARGE OR REDUCE THE FIGURE(S) DRAWN ACCORDING TO THE
C          VALUE OF F. F IS DESCRIBED IN THE COMMENT CARDS OF THE MAIN
C          PROGRAM OF NORTH (RANK 2).)
C
C          ONE HALF THE WIDTH OF THE VECTORHEAD IN INCHES: VHHW*F
C
C          INDICATOR OF THE TYPE OF DARKENING IN OF THE VECTORHEAD TO
C          BE USED: ITDK
C
C          DISTANCE OF SHRINKING (IN INCHES) TO BE USED IN THE DARKENING
C          IN OF THE VECTORHEAD: DOS*F
C          (TDOS=2*DOS)
C
C          INDICATOR OF THE PRESENCE (OR ABSENCE) OF A LABEL FOR THE LINE
C          OR VECTOR DRAWN AND, IF PRESENT, THE POSITIONING OF THE LABEL
```

```
C          WITH RESPECT TO THE LINE OR VECTOR: IVLP
C
C          DISTANCE IN INCHES OF THE LABEL FROM (ABOVE OR BELOW) THE
C          VECTOR: VLDFV*F
C
C          HEIGHT IN INCHES OF THE LABEL: VLH*F
C          (THE LENGTH OF THE LABEL VARIES ACCORDING TO THE NUMBER OF
C          SYMBOLS AND THE LENGTH OF THE SYMBOLS USED TO MAKE THE LABEL.)
C
C
C      OTHER SYMBOLS USED IN SUBROUTINE VECTO1:
C
C          NPS AND KS ARE INTEGER*4 ARRAYS. XS,YS,XSI AND YSI ARE REAL*4
C          ARRAYS. K,LN,SN AND N ARE INTEGER*4 VARIABLES. XDV,YDV,ZDV,
C          T,XPDV,YPDV,ZPDV,XMP,YMP,ZMP,SVLH,SXIP,SYIP,SZIP,SXPDV,SYPDV,
C          SZPDV AND SNPDV ARE REAL*4 VARIABLES. NDV AND NPDV ARE REAL*8
C          VARIABLES.
C
C          DIRECTION VECTOR FROM INITIAL POINT TO TERMINAL POINT: DV=
C          (XDV,YDV,ZDV)=TP-IP
C
C          SQUARE OF THE X AND Y COORDINATES OF THE DIRECTION VECTOR: T
C
C          A VECTOR PERPENDICULAR TO THE DIRECTION VECTOR: PDV=(XPDV,YPDV,
C          ZPDV)
C
C          NORM OF THE DIRECTION VECTOR: NDV
C
C          NORM OF VECTOR PERPENDICULAR TO THE DIRECTION VECTOR: NPDV
C
C          MIDPOINT OF THE LINE OR VECTOR TO BE DRAWN: MP=(XMP,YMP,ZMP)
C          =(IP+TP)/2
C
C          NUMBER OF POINTS USED TO DRAW THE J-TH ELEMENTARY SYMBOL:
C          NPS(J),J=1,2,...,17
C      THE FOLLOWING DATA INITIALIZATION STATEMENT DECLARES THE TYPE AND
C      DEFINES THE INITIAL VALUES OF NPS(J):
C
       INTEGER NPS(17)/10,6,4,6,5,9,11,11,2,8*0/
C
C          THE X AND Y COORDINATES FOR EACH OF THE NPS(J) POINTS USED TO
C          DRAW THE J-TH ELEMENTARY SYMBOL: (XS(I,J),YS(I,J)),I=1,...,NPS(J)
C      THE FOLLOWING TWO DATA INITIALIZATION STATEMENTS DECLARE THE TYPE AND
C      DEFINE THE INITIAL VALUES OF XS(I,J) AND YS(I,J):
C
       REAL XS(11,17)/
     1       0.,  0.,  5.,  8.,  9., 10.,  9.,  8.,  5.,  0.,  0.,
     2       0.,  5.,  5.,  0.,  5., 10.,  0.,  0.,  0.,  0.,  0.,
     3       0., 10.,  0., 10.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
     4       8., 18., 13., 13.,  8., 18.,  0.,  0.,  0.,  0.,  0.,
     5       4.,  6.,  5.,  5.,  4.,  0.,  0.,  0.,  0.,  0.,  0.,
     6       3.,  4.,  7.,  8.,  8.,  7.,  5.,  3.,  8.,  0.,  0.,
     7       6.,  4., 2.5, 1.5,  .8,  0.,  .8, 1.5, 2.5,  4.,  6.,
     8      40., 42.,43.5,44.5,45.2, 46.,45.2,44.5,43.5, 42., 40.,
     9      20., 25.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
```

```
     +      88*0./
C
      REAL YS(11,17)/
     1         0., 20., 20., 19., 18., 15., 12., 11., 10., 10.,  0.,
     2         0.,  0., 10., 20., 10., 20.,  0.,  0.,  0.,  0.,  0.,
     3         0., 20., 20.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
     4         0.,  0.,  0., 20., 20., 20.,  0.,  0.,  0.,  0.,  0.,
     5        -6., -6., -6.,  5.,  4.,  0.,  0.,  0.,  0.,  0.,  0.,
     6         2.,  4.,  4.,  2.,  0., -2., -4., -6., -6.,  0.,  0.,
     7         0.,  2.,  4.,  6.,  8., 10., 12., 14., 16., 18., 20.,
     8         0.,  2.,  4.,  6.,  8., 10., 12., 14., 16., 18., 20.,
     9        10., 10.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
     +      88*0./
C
C           THE POSITION OF THE PEN (EITHER UP OR DOWN) WHILE IT IS MOVING
C           FROM ONE POINT TO THE NEXT IN ORDER TO DRAW THE J-TH ELEMENTARY
C           SYMBOL: KS(I,J),I=1,...NPS(J)
C     THE FOLLOWING DATA INITIALIZATION STATEMENT DECLARES THE TYPE AND
C     DEFINES THE INITIAL VALUES OF KS(I,J):
C
      INTEGER KS(11,17)/
     1      3,2,2,2,2,2,2,2,2,2,0,
     2      3,3,2,2,3,2,0,0,0,0,0,
     3      3,2,3,2,0,0,0,0,0,0,0,
     4      3,2,3,2,3,2,0,0,0,0,0,
     5      3,2,3,2,2,0,0,0,0,0,0,
     6      3,2,2,2,2,2,2,2,2,0,0,
     7      3,2,2,2,2,2,2,2,2,2,2,
     8      3,2,2,2,2,2,2,2,2,2,2,
     9      3,2,0,0,0,0,0,0,0,0,0,
     +      88*0/
C
C           INTERMEDIATE SCALING OF XS(I,J) AND YS(I,J): XSI(I,J) AND
C           YSI(I,J)
C
C           FINAL THREE DIMENSIONAL POINT CORRESPONDING TO THE ORIGINAL
C           POINT (XS(I,J),YS(I,J)): (XSF,YSF,ZSF)
C           (AFTER P3D IS CALLED, EACH OF THESE THREE DIMENSIONAL POINTS
C           ARE PLOTTED IN PERSPECTIVE IN TWO DIMENSIONAL SPACE. THE UP OR
C           DOWN POSITION OF THE PEN IS DETERMINED BY THE VALUE OF K=KS(I,J)
C           FOR EACH POINT.)
C
C           SYMBOL NUMBER: SN
C           THE SEVENTEEN ELEMENTARY SYMBOLS PRESENTLY SUPPLIED CAN BE
C     IDENTIFIED BY SYMBOL NUMBER AS FOLLOWS:
C
C              SN   ELEMENTARY SYMBOL
C              --   ---------- ------
C              1    P
C              2    Y
C              3    X
C              4     I
C              5     1 (SUBSCRIPT)
C              6     2 (SUBSCRIPT)
C              7    (
```

```
C             8         )
C             9       -
C             10    P
C             11       P
C             12     Y
C             13        Y
C             14     1    (SUBSCRIPT)
C             15        1 (SUBSCRIPT)
C             16     2    (SUBSCRIPT)
C             17        2 (SUBSCRIPT)
C
C      (THE SYMBOLS CORRESPONDING TO SYMBOL NUNBERS 10 THRU 17 ARE SIMPLY
C      TRANSLATIONS OF THE SYMBOLS CORRESPONDING TO 1,2,5 AND 6. ALSO THE
C      "1" FOR SYMBOL NUMBERS 5 AND 14 AND THE "2" FOR SYMBOL NUMBERS 6 AND 16
C      ARE IN SLIGHTLY DIFFERENT POSITIONS EVEN THOUGH THEY APPEAR TO BE IN
C      THE SAME POSITION BECAUSE THE KEYPUNCH CAN NOT DISCRIMINATE THIS.
C      THE "1" IN SYMBOL NUMBER 5 CORRESPONDS TO THE "1" IN THE LABEL "P1Y"
C      (THE "1" IS REALLY A SUBSCRIPT BUT SUBSCRIPTS ARE NOT AVAILABLE
C      ON IBM KEYPUNCHES) . THE "1" IN "X1" (THE "1" IS A SUBSCRIPT)
C      IS LOCATED IN A SLIGHTLY DIFFERENT POSITICN THAN THE "1" IN "P1Y".
C      LIKEWISE THE "2" IN SYMBOL NUMBER 6 CORRESPONDS TO THE "2" IN "P2Y"
C      AND THEREFORE IS IN A SLIGHTLY DIFFERENT POSITION THAN THE "2" IN "X2"
C
C           LABEL NUMBER: LN
C           THE ELEVEN DIFFERENT LABELS PRESENTLY SUPPLIED CAN BE IDENTIFIED
C      BY LABEL NUMBER AND BY SEQUENCE OF SYMBOL NUMBERS (FOR THE ELEMENTARY
C      SYMBOLS THAT ARE USED TO PRODUCE THEM) AS FOLLOWS:
C
C             LN    LABEL      SYMBOL NUMBER SEQUENCE
C             --    -----      ------ ------ --------
C             1     P1Y        1,5,12
C             2     X1         3,14
C             3     P2Y        1,6,12
C             4     X2         3,16
C             5     PY         1,12
C             6     Y          2
C             7     (P-P1)Y    7,10,9,11,15,8,13
C             8     (I-P1)Y    7,4,9,11,15,8,13
C             9     (I-P)Y     7,4,9,11,8,13
C             10    (P-P2)Y    7,10,9,11,17,8,13
C             11    (I-P2)Y    7,4,9,11,17,8,13
C
C           THE VARIABLE USED TO SAVE THE ORIGINAL VALUE OF VHL: SVHL
C           (SIMILARLY SVHHW SAVES THE ORIGINAL VALUE OF VHHW AS DO EACH
C           OF THE FOLLOWING VARIABLES SAVE THE ORIGINAL VALUE OF THE
C           VARIABLE IN PARENTHESIS NEXT TO THEM: SVLH(VLH),SXIP(XIP),
C           SYIP(YIP),SZIP(ZIP),SXPDV(XPDV),SYPDV(YPDV),SZPDV(ZPDV),
C           SNPDV(NPDV).)
C
C
C           IIP AND ITP ARE POSITIVE INTEGERS BETWEEN 1 AND ND. ND IS THE
C      NUMBER OF POINTS FROM OR TO WHICH LINES/OR VECTORS CAN BE DRAWN.
C      IT IS READ IN IN THE MAIN PROGRAM OF NORTH (RANK 2) ALONG WITH NV
C      WHICH IS THE NUMBER OF LINES AND VECTORS DRAWN. THE VALUES OF IIP
C      AND ITP UNIQUELY SPECIFY A VALUE OF LN.
```

```
C
C          IF IVHP IS EQUAL TO 999 NO VECTORHEAD IS DRAWN. OTHERWISE A
C     VECTORHEAD IS DRAWN.
C
C          IF IVHT=1 AND T IS NONZERO, PDV IS SET EQUAL TO (XDV*ZDV,
C     YDV*ZDV,-T). BESIDES BEING PERPENDICULAR TO THE DIRECTION VECTOR,
C     THIS VECTOR IS IN THE PLANE OF THE GRID SINCE IT EQUALS ZDV*(XDV,
C     YDV,-T/ZDV). CONSEQUENTLY, THE VECTORHEAD DRAWN WILL BE IN THE  PLANE
C     OF THE GRID. IF IVHT=2 AND T IS NONZERO, PDV IS SET EQUAL TO (-YDV,XDV,0)
C     THIS VECTOR IS PERPENDICULAR TO THE DIRECTION VECTOR. IT IS ALSO
C     PERPENDICULAR TO THE PLANE OF THE GRID . CONSEQUENTLY, THE VECTOR-
C     HEAD WILL BE PERPENDICULAR TO THE PLANE OF THE GRID. IF T IS ZERO,
C     PDV IS SET EQUAL TO (1,0,0) SO THAT THE VECTORHEAD WILL BE DRAWN
C     IN THE XZ PLANE. IF WISHED, THE PROGRAM CAN BE EASILY MODIFIED SO
C     THAT THE VECTORHEAD WILL BE IN THE YZ PLANE INSTEAD OF THE XZ PLANE.
C     TC ACCCMPLISH THIS, SIMPLY CHANGE THE TWO CARDS
C
C     XPDV=1.
C     YPDV=0.
C
C     WHICH ARE FOUND AFTER THE CARD
C
C     IF(T.GT.0) GO TO 110
C
C     TC THE FOLLOWING TWC CARDS
C
C     XPDV=0.
C     YPDV=1.
C
C
C          THERE ARE THREE WAYS TO DARKEN IN A VECTORHEAD. ALL BEGIN BY
C     DRAWING A TRIANGLE WITH LENGTH (ALTITUDE) EQUAL TO SVHL AND WITH
C     WIDTH OF THE BASE EQUAL TO 2*SVHHW. THE APEX OF THE TRIANGLE IS AT
C     THE TERMINAL POINT (TP) OF THE VECTOR. A LARGE NUMBER OF SUCCESSIVELY
C     SMALLER TRIANGLES ARE THEN DRAWN FILLING IN THE SPACE INTERIOR TO THE
C     CRIGINAL TRIANGLE. THE METHCDS OF DARKENING IN OF THE VECTORHEAD
C     DIFFER IN WHAT TYPE OF SMALLER TRIANGLES ARE USED. IF ITDK=1, THE
C     SMALLER TRIANGLES ARE SIMILAR TO THE ORIGINAL TRIANGLE AND THEIR
C     APEXIAL POINT IS THE TERMINAL POINT OF THE VECTOR TO BE DRAWN.
C     IF ITDK=2, THE SMALLER TRIANGLES ARE AGAIN SIMILAR TO THE ORIGINAL
C     TRIANGLE BUT THEIR APEXAL POINTS CHANGE IN SUCH A WAY THAT THEIR
C     CENTROIDS ARE ALL IDENTICAL TO THE CENTROID OF THE ORIGINAL TRIANGLE.
C     IF ITDK=3, THE SMALLER TRIANGLES ALL HAVE THE TERMINAL POINT AS THEIR
C     APEXIAL POINT BUT THE SMALLER TRIANGLES ARE NOT SIMILIAR TO THE
C     ORIGINAL ONE. RATHER THEIR LENGTH (ALTITUDE) IS MAINTAINED CONSTANT
C     AND EQUAL TO SVHL. THEIR WIDTH IS THE CHANGING FACTOR.
C
C          IF ITDK=1 AND DOS IS GREATER THAN OR EQUAL TO SVHL OR IF ITDK=2
C     AND COS IS GREATER THAN SVHL/2 OR IF ITDK=3 AND DOS IS GREATER THAN
C     OR EQUAL TO VHHW, THEN THE VECTORHEAD WILL BE DRAWN BUT NOT DARKENED
C     IN. (THAT IS, THE ORIGINAL TRIANGLE DESCRIBED ABOVE WILL BE DRAWN
C     BUT NOT THE SMALLER TRIANGLES.)
C
C          IF IVLP=999, NO VECTOR LABEL WILL BE DRAWN. IF IVLP IS NOT
C     EQUAL TO 999 A VECTOR LABEL WILL BE DRAWN. IF THE LABEL IS BELOW
```

```
C     (ABOVE) THE VECTOR AND IT IS DESIRED TO PLACE IT ABOVE (BELOW), MAKE
C     IVLP ZERO IF IT WAS NONZERO (BUT NOT EQUAL TO 999) PREVIOUSLY OR MAKE
C     IVLP NONZERO (BUT NOT EQUAL TO 999) IF IT WAS ZERO PREVIOUSLY.
C     (THIS PROCEDURE MAY FAIL TO ACCOMPLISH THE DESIRED TASK DEPENDING
C     ON THE ANGLE OF VIEWING USED. IF THIS IS THE CASE, EITHER THE DATA
C     FED INTO THE ARRAYS XS AND YS CAN BE APPROPRIATELY MODIFIED OR THE
C     APPROPRIATE FORTRAN STATEMENTS CAN BE MODIFIED.)
C
C
C
      CALL P3D(XIP,YIP,ZIP,3)                                           NRT30007
      CALL P3D(XTP,YTP,ZTP,2)                                           NRT30008
      IF(IVHP.EQ.999.AND.IVLP.EQ.999) RETURN                            NRT30009
      XDV=XTP-XIP                                                       NRT30010
      YDV=YTP-YIP                                                       NRT30011
      ZDV=ZTP-ZIP                                                       NRT30012
      T=XDV**2+YDV**2                                                   NRT30013
      IF (T.GT.0.) GO TO 110                                            NRT30014
      XPDV=1.                                                           NRT30015
      YPDV=0.                                                           NRT30016
      ZPDV=0.                                                           NRT30017
      GO TO 220                                                         NRT30018
110   IF(IVHT.EQ.2) GO TO 33                                            NRT30019
      XPDV=XDV*ZDV                                                      NRT30020
      YPDV=YDV*ZDV                                                      NRT30021
       ZPDV=-T                                                          NRT30022
      GO TO 220                                                         NRT30023
33    XPDV=-YDV                                                         NRT30024
      YPDV=XDV                                                          NRT30025
      ZPDV=0.0                                                          NRT30026
220   CONTINUE                                                          NRT30027
      NDV=XDV**2+YDV**2+ZDV**2                                          NRT30028
      NDV=DSQRT(NDV)                                                    NRT30029
      NPDV=XPDV**2+YPDV**2+ZPDV**2                                      NRT30030
      NPDV=DSQRT(NPDV)                                                  NRT30031
      IF(IVHP.EQ.999.AND.IVLP.NE.999)GO TO 991                          NRT30032
      SVHL=VHL                                                          NRT30033
      SVHHW=VHHW                                                        NRT30034
      IOS=1                                                             NRT30035
55    C1=-VHL/NDV                                                       NRT30036
      C2=VHHW/NPDV                                                      NRT30037
      XPPDV=XTP+C1*XDV+C2*XPDV                                          NRT30038
      YPPDV=YTP+C1*YDV+C2*YPDV                                          NRT30039
      ZPPDV=ZTP+C1*ZDV+C2*ZPDV                                          NRT30040
      XMPDV=XTP+C1*XDV-C2*XPDV                                          NRT30041
      YMPDV=YTP+C1*YDV-C2*YPDV                                          NRT30042
      ZMPDV=ZTP+C1*ZDV-C2*ZPDV                                          NRT30043
      CALL P3D(XPPDV,YPPDV,ZPPDV,2)                                     NRT30044
      CALL P3D(XMPDV,YMPDV,ZMPDV,2)                                     NRT30045
      IF(ITDK.NE.2.OR.IOS.EQ.1) CALL P3D(XTP,YTP,ZTP,2)                 NRT30046
      IF(ITDK.EQ.2.AND.IOS.NE.1) CALL P3D( XTIP,YTIP,ZTIP,2)            NRT30047
      IF(ITDK.NE.1) GO TO 1                                             NRT30048
      VHL=VHL-DOS                                                       NRT30049
      VHHW=VHL*SVHHW/SVHL                                               NRT30050
      IF(VHL.LE.DOS) GO TO 77                                           NRT30051
```

```
      GO TO 55                                                          NRT30052
1     CONTINUE                                                          NRT30053
      IF(ITDK.NE.2) GO TO 2                                             NRT30054
      VHL=VHL-DOS                                                       NRT30055
      VHHW=VHL*SVHHW/SVHL                                               NRT30056
      IOS=IOS+1                                                         NRT30057
      CC1=-(SVHL-VHL)/NDV                                               NRT30058
      XTIP=XTP+CC1*XDV                                                  NRT30059
      YTIP=YTP+CC1*YDV                                                  NRT30060
      ZTIP=ZTP+CC1*ZDV                                                  NRT30061
      CALL P3D(XTIP,YTIP,ZTIP,3)                                        NRT30062
      TDOS=2*DOS                                                        NRT30063
      IF(VHL.LT.(SVHL+TDOS)/2.) GO TO 77                                NRT30064
      GO TO 55                                                          NRT30065
2     CONTINUE                                                          NRT30066
      IF(ITDK.NE.3) GO TO 3                                             NRT30067
      VHHW=VHHW-DOS                                                     NRT30068
      IF(VHHW.LE.DOS) GO TO 77                                          NRT30069
      GO TO 55                                                          NRT30070
3     CONTINUE                                                          NRT30071
77    CONTINUE                                                          NRT30072
      VHL=SVHL                                                          NRT30073
      VHHW=SVHHW                                                        NRT30074
991   IF(IVLP.EQ.999)RETURN                                             NRT30075
      IF(ITP.LE.IIP) GO TO 989                                          NRT30076
      IF(IIP.EQ.3.OR.IIP.EQ.5.OR.IIP.GE.7) GO TO 989                    NRT30077
      IF(IIP.EQ.2.AND.(ITP.EQ.3.OR.ITP.EQ.4.OR.ITP.EQ.5)) GO TO 989     NRT30078
      IF(IIP.EQ.4.AND.ITP.EQ.5) GO TO 989                               NRT30079
      NPS(10)=NPS(1)                                                    NRT30080
      NPS(11)=NPS(1)                                                    NRT30081
      NPS(12)=NPS(2)                                                    NRT30082
      NPS(13)=NPS(2)                                                    NRT30083
      NPS(14)=NPS(5)                                                    NRT30084
      NPS(15)=NPS(5)                                                    NRT30085
      NPS(16)=NPS(6)                                                    NRT30086
      NPS(17)=NPS(6)                                                    NRT30087
      DO 401 I=1,11                                                     NRT30088
      KS(I,10)=KS(I,1)                                                  NRT30089
      KS(I,11)=KS(I,1)                                                  NRT30090
      KS(I,12)=KS(I,2)                                                  NRT30091
      KS(I,13)=KS(I,2)                                                  NRT30092
      KS(I,14)=KS(I,5)                                                  NRT30093
      KS(I,15)=KS(I,5)                                                  NRT30094
      KS(I,16)=KS(I,6)                                                  NRT30095
      KS(I,17)=KS(I,6)                                                  NRT30096
      YS(I,10)=YS(I,1)                                                  NRT30097
      YS(I,11)=YS(I,1)                                                  NRT30098
      YS(I,12)=YS(I,2)                                                  NRT30099
      YS(I,13)=YS(I,2)                                                  NRT30100
      YS(I,14)=YS(I,5)                                                  NRT30101
      YS(I,15)=YS(I,5)                                                  NRT30102
      YS(I,16)=YS(I,6)                                                  NRT30103
      YS(I,17)=YS(I,6)                                                  NRT30104
      XS(I,10)=XS(I,1)+8                                                NRT30105
      XS(I,11)=XS(I,1)+28                                               NRT30106
```

```
      XS(I,12)=XS(I,2)+12                                               NRT30107
      XS(I,13)=XS(I,2)+48                                               NRT30108
      XS(I,14)=XS(I,5)+10                                               NRT30109
      XS(I,15)=XS(I,5)+28                                               NRT30110
      XS(I,16)=XS(I,6)+10                                               NRT30111
401   XS(I,17)=XS(I,6)+28                                               NRT30112
      IF(IIP.EQ.1) LN=ITP-1                                             NRT30113
      IF(IIP.EQ.2) LN=ITP+1                                             NRT30114
      IF(IIP.EQ.4) LN=ITP+4                                             NRT30115
      IF(IIP.EQ.6) LN=ITP+2                                             NRT30116
      SVLH=VLH                                                          NRT30117
      IF(LN.EQ.1.OR.LN.EQ.3.OR.LN.EQ.5) VLH=VLH*2.2                     NRT30118
      IF(LN.EQ.2.OR.LN.EQ.4) VLH=1.7*VLH                                NRT30119
      IF(LN.GE.7) VLH=VLH*4.6                                           NRT30120
      SXIP=XIP                                                          NRT30121
      SYIP=YIP                                                          NRT30122
      SZIP=ZIP                                                          NRT30123
      IF(LN.EQ.2.OR.LN.EQ.4) XIP=XTP                                    NRT30124
      IF(LN.EQ.2.OR.LN.EQ.4) YIP=YTP                                    NRT30125
      IF(LN.EQ.2.OR.LN.EQ.4) ZIP=ZTP                                    NRT30126
      XMP=(XIP+XTP)/2                                                   NRT30127
      YMP=(YIP+YTP)/2                                                   NRT30128
      ZMP=(ZIP+ZTP)/2                                                   NRT30129
      XIP=SXIP                                                          NRT30130
      YIP=SYIP                                                          NRT30131
      ZIP=SZIP                                                          NRT30132
      DO 111 J=1,17                                                     NRT30133
      DO 111 I=1,11                                                     NRT30134
      IF(IVLP.NE.0) YSI(I,J)=YS(I,J)                                    NRT30135
      IF(IVLP.EQ.0) YSI(I,J)=YS(I,J)-40                                 NRT30136
      IF(LN.EQ.10)YSI(I,J)=YS(I,J)-40                                   NRT30137
      IF(LN.EQ.11) YSI(I,J)=-YS(I,J)+20                                 NRT30138
      XSI(I,J)=XS(I,J)-VLH/SVLH*10.                                     NRT30139
      YSI(I,J)=YSI(I,J)+10.                                             NRT30140
      XSI(I,J)=XSI(I,J)/20*2*SVLH                                       NRT30141
      YSI(I,J)=YSI(I,J)/20*2*SVLH                                       NRT30142
111   CONTINUE                                                          NRT30143
      IF(LN.NE.10.AND.LN.NE.11) GO TO 987                               NRT30144
      SXPDV=XPDV                                                        NRT30145
      SYPDV=YPDV                                                        NRT30146
      SZPDV=ZPDV                                                        NRT30147
      SNPDV=NPDV                                                        NRT30148
      XPDV=-YDV                                                         NRT30149
      YPDV= XDV                                                         NRT30150
      ZPDV=0.0                                                          NRT30151
      NPDV=XPDV**2+YPDV**2+ZPDV**2                                      NRT30152
      NPDV=DSQRT(NPDV)                                                  NRT30153
987   CONTINUE                                                          NRT30154
      N=7                                                               NRT30155
      IF(LN.NE.1.AND.LN.NE.3.AND.LN.NE.5) GO TO 11                      NRT30156
      SN=1                                                              NRT30157
      N=8                                                               NRT30158
5     NN=NPS(SN)                                                        NRT30159
      DO 202 I=1,NN                                                     NRT30160
      XSF=XMP+(XDV/NDV)*XSI(I,SN)-(XPDV/NPDV)*YSI(I,SN)                 NRT30161
```

```
      YSF=YMP+(YDV/NDV)*XSI(I,SN)-(YPDV/NPDV)*YSI(I,SN)                 NRT30162
      ZSF=ZMP+(ZDV/NDV)*XSI(I,SN)-(ZPDV /NPDV)*YSI(I,SN)                NRT30163
      K=KS(I,SN)                                                        NRT30164
202   CALL P3D(XSF,YSF,ZSF,K)                                           NRT30165
      GO TO (101,101,101,101,101,101,101,8,9,10,11,12,13,14,15,16,17,   NRT30166
     118,19,20,21,22,23,24,101),N                                       NRT30167
8     IF(LN.EQ.1) SN=5                                                  NRT30168
      N=N+1                                                             NRT30169
      IF(LN.EQ.1) GO TO 5                                               NRT30170
9     IF(LN.EQ.3) SN=6                                                  NRT30171
      N=N+1                                                             NRT30172
      IF(LN.EQ.3) GO TO 5                                               NRT30173
10    IF(LN.EQ.1.OR.LN.EQ.3.OR.LN.EQ.5) SN=12                           NRT30174
      N=N+1                                                             NRT30175
      IF(LN.EQ.1.OR.LN.EQ.3.OR.LN.EQ.5) GO TO 5                         NRT30176
11    IF(LN.NE.2.AND.LN.NE.4) GO TO 14                                  NRT30177
      SN=3                                                              NRT30178
      N=12                                                              NRT30179
      GO TO 5                                                           NRT30180
12    IF(LN.EQ.2) SN=14                                                 NRT30181
      N=N+1                                                             NRT30182
      IF(LN.EQ.2) GO TO 5                                               NRT30183
13    IF(LN.EQ.4) SN=16                                                 NRT30184
      N=N+1                                                             NRT30185
      IF(LN.EQ.4) GO TO 5                                               NRT30186
14    IF(LN.NE.6) GO TO 15                                              NRT30187
      SN=2                                                              NRT30188
      N=15                                                              NRT30189
      GO TO 5                                                           NRT30190
15    IF(LN.LT.7) GO TO 24                                              NRT30191
      SN=7                                                              NRT30192
      N=16                                                              NRT30193
      GO TO 5                                                           NRT30194
16    IF(LN.EQ.7.OR.LN.EQ.10) SN=10                                     NRT30195
      N=N+1                                                             NRT30196
      IF(LN.EQ.7.OR.LN.EQ.10) GO TO 5                                   NRT30197
17    IF(LN.EQ.8.OR.LN.EQ.9.OR.LN.EQ.11) SN=4                           NRT30198
      N=N+1                                                             NRT30199
      IF(LN.EQ.8.OR.LN.EQ.9.OR.LN.EQ.11) GO TO 5                        NRT30200
18    IF(LN.GE.7) SN=9                                                  NRT30201
      N=N+1                                                             NRT30202
      IF(LN.GE.7) GO TO 5                                               NRT30203
19    IF(LN.GE.7) SN=11                                                 NRT30204
      N=N+1                                                             NRT30205
      IF(LN.GE.7) GO TO 5                                               NRT30206
20    IF(LN.EQ.7.OR.LN.EQ.8) SN=15                                      NRT30207
      N=N+1                                                             NRT30208
      IF(LN.EQ.7.OR.LN.EQ.8) GO TO 5                                    NRT30209
21    IF(LN.EQ.10.OR.LN.EQ.11) SN=17                                    NRT30210
      N=N+1                                                             NRT30211
      IF(LN.EQ.10.OR.LN.EQ.11) GO TO 5                                  NRT30212
22    IF(LN.GE.7) SN=8                                                  NRT30213
      N=N+1                                                             NRT30214
      IF(LN.GE.7) GO TO 5                                               NRT30215
23    IF(LN.GE.7) SN=13                                                 NRT30216
```

```
      N=N+1                                                             NRT30217
      IF(LN.GE.7) GO TO 5                                               NRT30218
101   WRITE(6,997)                                                      NRT30219
997   FCRMAT(' NOT  WITHIN 8-24')                                       NRT30220
      GO TO 24                                                          NRT30221
989   WRITE(6,988)                                                      NRT30222
988   FCRMAT(' LABELS HAVE NOT BEEN SUPPLIED FOR THIS COMBINATION'      NRT30223
     1,' OF IIP AND ITP.')                                              NRT30224
24    CCNTINUE                                                          NRT30225
      IF(LN.NE.10.AND.LN.NE.11) GO TO 981                               NRT30226
      XPDV=SXPDV                                                        NRT30227
      YPDV=SYPDV                                                        NRT30228
      ZPDV=SZPDV                                                        NRT30229
      NPDV=SNPDV                                                        NRT30230
981   CONTINUE                                                          NRT30231
      VLH=SVLH                                                          NRT30232
      RETURN                                                            NRT30233
      END                                                               NRT30234
C
C
      SUBROUTINE P3D(X,Y,Z,IC)
C
C
C          THIS SUBROUTINE PLOTS THREE DIMENSIONAL POINTS IN PERSPECTIVE
C     CN TWO DIMENSIONAL PAPER. IT MUST BE ACCOMPANIED BY SUBROUTINE
C     SETUP IN THE FORTRAN SOURCE DECK.
C          THE COMMENT CARDS OF SETUP DESCRIBE THE VARIABLES C(1) THRU
C     C(20) CCNNECTED TO P3D BY THE COMMON STATEMENT. C(21) AND C(22)
C     ARE CESCRIBED BELOW. THE PARAMETERS X,Y AND Z OF SUBROUTINE P3D
C     ARE REAL*4 VARIABLES. THEY ARE THE COORDINATES OF THE POINT (X,Y,Z).
C     THE PARAMETER IC IS AN INTEGER*4 VARIABLE. IT INDICATES WHETHER THE
C     PEN IS IN AN UP OR A DOWN POSITION. IF IC=3 THE PEN IS UP AND IF
C     IC=2 THE PEN IS DOWN.
C          P3D IS COMPOSED CF FIVE STEPS. IN THE FIRST STEP EACH POINT
C     (X,Y,Z) IS TRANSLATED BY -FP=(-C(1),-C(2),-C(3)) IN ORDER TO MAKE
C     THE FOCAL POINT THE NEW ORIGIN. IN THE SECOND STEP, EACH POINT
C     PRCDUCED IN THE FIRST STEP IS ROTATED BY THE APPROPRIATE P MATRIX
C     (CREATED AND DESCRIBED IN THE COMMENT CARDS FOR SETUP) SUCH THAT THE
C     CBSERVATICN DIRECTION VECTOR BECOMES (0,-S,0). IN THE THIRD STEP AN
C     IMAGINARY LINE IS DRAWN FRCM EACH POINT PRODUCED IN THE SECOND STEP
C     TC THE POINT (0,-S,0). THE POINT OF INTERSECTION OF THIS LINE WITH
C     THE XZ PLANE IS OBTAINED. THIS INTERSECTION IS THE PERSPECTIVE
C     PROJECTION OF THE POINT PRODUCED IN STEP TWO ONTO THE XZ PLANE. IF
C     C(20) IS ZERO, CCNTROL IS GIVEN BACK TO SETUP SINCE THE BOUNDARY
C     LIMITATIONS (MINIMUM AND RANGE OF EACH OF THE X AND Y COORDINATES)
C     CF THE PROJECTED 10 BY 10 BY 10 CUBE NEED TO BE SPECIFIED. EQUI-
C     VALENTLY, C(16) THRU C(19) NEED TO BE CREATED OR REDEFINED.
C          IF C(20) IS NONZERC, THE PROGRAM PROCEEDS TO STEP FOUR OF P3D
C     WHERE THE VALUES GIVEN TO C(16) THRU C(19) DURING THE LAST TIME THAT
C     C(20) WAS ZERO, ARE USED. EACH POINT PRODUCED IN STEP THREE IS SCALED
C     SO THAT THE DIMENSIONS OF THE RESULTING FIGURE ARE 10*F BY 8*F. (F
C     IS THE PARAMETER OF THE SYSTEM SUPPLIED SUBROUTINE FACTOR WHICH IS
C     CALLED IN THE MAIN PROGRAM OF NCRTH (RANK 2) IN ORDER TO ENLARGE OR
C     RECUCE THE FIGURE(S) DRAWN ACCORDING TO THE VALUE OF F. F IS DESCRIBED
C     IN THE COMMENT CARDS CF THE MAIN PROGRAM OF NORTH (RANK 2).)
```

```
C            IN THE LAST STEP, FOR EVERY POINT (C(21),C(22)) PRODUCED IN
C       STEP FOUR, THE PEN MOVES SEQUENTIALLY FROM CNE POINT TO THE NEXT
C       DEPENDING CN THE ORDER IN WHICH THE POINT WAS PRODUCED. IF IC=2
C       (THE PEN IS DOWN), THE PEN WILL DRAW A LINE FROM CNE POINT TO THE
C       NEXT. IF IC=3 (THE PEN IS UP), IT WILL MOVE FROM CNE POINT TO THE
C       NEXT WITHOUT DRAWING A LINE.
C
C
        COMMON C(22),XPV1,YPV3
C       TRANSLATE
        XO=X-C(1)
        YO=Y-C(2)
        ZC=Z-C(3)
C       ROTATE
        C(21)=C(8)*XO+C(9)*YO
        C(22)=C(13)*XO+C(14)*YO+C(15)*ZO
        YC=C(10)*XO+C(11)*YO+C(12)*ZO
C       PROJECTICN
        YC=YC+C(7)
        IF(YO.GT.0.) GO TO 1
        WRITE(6,2)
    2   FCRMAT(' VIEWING IMPOSSIBLE')
1       YC=C(7)/YO
        C(21)=YO*C(21)
        C(22)=YO*C(22)
        IF (C(20).EQ.0.) RETURN
C       SCALE
        C(21)=(C(21)-C(16))/C(17)*10.
        C(22)=(C(22)-C(18))/C(19)*8.
C       PLOT
        CALL PLOT(C(21),C(22),IC)
        XPV1=C(21)
        YPV3=C(22)
        RETURN
        END

*********DATA***********DATA***********DATA**********DATA***********DATA************
 2 1
  10     4    5    3    1    T    T
5.3033     2.12132   5.3033    5.3033
-5.3033    -2.12132  -5.3033   -5.3033
0.0        4.63721   3.18198   3.18198
0.0        -4.63721  -3.18198  -3.18198
0.0        0.0       0.0       9.
 1
5.     5.     10.    -2.    -12.    0.     18.    100  7 11 .25  .111.010.10.10
0.   0.   0.
10.  0.   0.
0.   10.  0.
0.   0.   10.
10.  10.  C.
10.  0.   10.
0.   10.  10.
10.  1C.  10.
```

```
               9.999
1  2    999
1  3    999
1  4    999
1  5    999
1  6    999
1  7    999
2  6999999
2  7999999
4  6999999
4  7999999
6  7999999
```

Bibliography

[1] H. Scheffé. The Analysis of Variance, John Wiley, New York, 1959.

[2] S. R. Searle. Linear Models, John Wiley, New York, 1971.