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.
1. **Introduction.** In the context of orthogonal designs, Scheffe [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 \( P_1y, P_2y, Py, y \). 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, \( \beta \) be an \( n \times 1 \) vector of unknown parameters and \( \varepsilon \) be the \( n \times 1 \) vector of error terms. Partition \( X \) as

\[
X = [X_0; X_1; X_2]
\]

\[
\begin{array}{c}
\uparrow \\
\Rightarrow \\
k_0 \ k_1 \ k_2
\end{array}
\]

---

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 \( \beta \) 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

\[
\epsilon \sim N(0, \sigma^2 I).
\]

Statistics for inference about \( \beta_1 \) and \( \beta_2 \) depend only on

\[
\begin{bmatrix}
X^* : y^* \\
X_1^* : X_2^* : y^*
\end{bmatrix} = [X_1^* : X_2^* : y^*] = I - X_0 (X_0'X_0)^{-1} X_0' [X_1^* : X_2^* : y^*].
\]

Let

\[
P_1 = X_1^* (X_1^* X_1^*)^{-1} X_1^*
\]
\[
P_2 = X_2^* (X_2^* X_2^*)^{-1} X_2^*
\]
\[
P = X^* (X^* X^*)^{-1} X^*
\]

be the projection matrices formed from \( X^* \).

3. **Illustration of Information Gained.** First consider figures la) and lb) 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 la) and
Figure 1 a)

Figure 1 b)

Figure 1 c)
lb) 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 la) whereas they are obviously highly non-orthogonal in lb). 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 la) and lb) into one figure so that the concepts of this section are more concisely represented by it.


Since all the information about $\beta_1$ and $\beta_2$ in the data is contained in the four n dimensional vectors $P_1y^*$, $P_2y^*$, $Py^*$ and $y^*$, hereafter let

$$Z = [Z_1:y^*] = [P_1y^*:P_2y^*:Py^*:y^*] .$$

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 $A_1 = [a_1 \ a_2]$ with orthonormal columns is needed such that $A_1A_1^TZ_1$ is a projection of $Z$ which for some criterion produces minimum distortion. A natural criterion is to choose the $A_1$ which minimizes

$$\|Z_1-A_1A_1^TZ_1\|_F$$
where \(|\cdot|_F^p\) is the Frobenius norm, or equivalently to choose the \(A_1\) which maximizes

\[
(2) \quad \text{tr}(A_1^T Z_1 Z_1 A_1).
\]

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

\[
A_1 = A_1^* = [a_1^*; a_2^*]
\]

where \(a_1^*\) is the eigenvector associated with the largest eigenvalue \((\lambda_1)\) of \(Z_1 Z_1^T\) and \(a_2^*\) is the eigenvector associated with the second largest eigenvalue \((\lambda_2)\) of \(Z_1 Z_1^T\). Henceforth, the "*" in \(A_1^*\) will be dropped and it will be referred to simply as \(A_1\).

Second Step. Let \(A = [A_1 \ a_3]\) where \(a_3\) is a vector of unit length such that \(a_3^T Z_1 = 0\) and \(AA' = A_1 A_1^T + a_3 a_3^T\). The matrix \(A\) is sought that maximizes

\[
\text{tr}(A'ZZ'A).
\]

But

\[
\max \quad a_3^T Z_1 = 0 \quad \text{tr}(A'ZZ'A) \quad \text{subject to} \quad AA' = A_1 A_1^T + a_3 a_3^T
\]

\[
= \max \quad \text{tr} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & (a_3 y^*)^2 \end{bmatrix}
\]

\[
= \lambda_1 + \lambda_2 + \max_{a_3} (a_3 y^*)^2.
\]

The problem thus reduces to finding the vector \(a_3\) that maximizes
\[(a'y*)^2.\]

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

\[
a_3 = \frac{1}{||c||}\tilde{Z}_1c.
\]

Consequently it is needed to find the \(c\) that maximizes

\[
\frac{(c'\tilde{Z}_1y*)^2}{c'c}.
\]

But it is well known that this is maximized for

\[
c^* \propto \tilde{Z}_1'y^*
\]

and thus

\[
a_3^* \propto \tilde{Z}_1'\tilde{Z}_1'y^*.
\]

But

\[
\tilde{Z}_1'\tilde{Z}_1'y^* = (I-Z_1(Z_1'Z_1)'Z_1'y^*) = (I-P)y^*
\]

Also \(a_3^*\) is of unit length so

\[
a_3^* = \frac{(I-P)y^*}{||(I-P)y^*||} = \frac{(I-P)y^*}{\sqrt{SSE}}
\]

where SSE is the sum of squares error for the model. Consequently \(A^* = [a_1^*a_2^*a_3^*]\) is the required matrix and \(A^*A'^*Z\) is a projection of \(Z\) such that the first three columns of \(A^*A'^*Z\) all line in the same plane in \(n\) dimensional space. The
distortion incurred in $A^*A^*Z$ 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 $a^*_3$ will be referred to simply as $a_3$.

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

$$A'Z.$$  

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

$$A'Z = A'(AA'Z) = A'[A_1A'_1+a_3a'_3][Z_1^\prime y^\prime]$$

$$= \begin{bmatrix} A'_1 \\ a'_3 \end{bmatrix} [A_1A'_1Z_1A_1A'_1y^\prime+a_3(a'_3y^\prime)]$$

$$= \begin{bmatrix} A'_1Z_1 & A'_1y^\prime \\ 0 & a'_3y^\prime \end{bmatrix} = \begin{bmatrix} A'_1Z_1 & A'_1y^\prime \\ 0 & \sqrt{\text{SSE}} \end{bmatrix}$$

because $a'_3y^\prime = \frac{y^\prime(I-P)y^\prime}{\sqrt{\text{SSE}}}$, 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 $Z_1^\prime Z_1$.) To triangularize the resulting $3 \times 4$ matrix, it is premultiplied by the appropriate elementary reflector.
5. **Actual Calculating Algorithm.** 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

\[
\begin{align*}
  z_1 &= USV' \\
  U &= \text{an } n \times 3 \text{ matrix such that } U'U = I \\
  V &= \text{a } 3 \times 3 \text{ matrix such that } V'V = I \\
  S &= \text{a } 3 \times 3 \text{ diagonal matrix }.
\end{align*}
\]

Consequently one has

\[
Z = \begin{bmatrix} USV' \mid y^* \end{bmatrix}.
\]

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

\[
\begin{bmatrix} U' \\ \tilde{U}' \end{bmatrix}Z = \begin{bmatrix} SU' & U'y^* \\ 0 & \tilde{U}'y^* \end{bmatrix}.
\]

Deleting the third row, triangularizing the resulting matrix by premultiplying it by the appropriate elementary reflector and noting that \( \tilde{U}'y^* \) 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 diffi-
cult 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 \( F(\beta_1, \beta_2 | \beta_0) \), \( F(\beta_1 | \beta_0, \beta_1) \), \( F(\beta_2 | \beta_0, \beta_1) \) 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 \( \beta_1 \) and \( \beta_2 \) 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 \( \beta_1 \) and \( \beta_2 \). It is clear that one should fit both \( \beta_1 \) and \( \beta_2 \) in situation 1 and neither \( \beta_1 \) nor \( \beta_2 \) 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.
### Table 1. A Reproduction of Searle's Table 7.4 [2, page 278]

With $\beta_1$ and $\beta_2$ used in Place of $\alpha$ and $\beta$

| Effects to be included in model | $F(\beta_1 | \beta_0)$ | $F(\beta_1 | \beta_0, \beta_2)$ | Sig | N S | Sig | N S |
|---------------------------------|------------------------|---------------------------------|-----|-----|-----|-----|
| $F(\beta_2 | \beta_0)$ | Sig | $\beta_1$ and $\beta_2$ | $\beta_1$ and $\beta_2$ | $\beta_2$ | $\beta_1$ and $\beta_2$ |
| $F(\beta_2 | \beta_0, \beta_1)$ | Sig | $\beta_1$ and $\beta_2$ | $\beta_1$ and $\beta_2$ | $\beta_2$ | $\beta_1$ and $\beta_2$ |
| $F(\beta_2 | \beta_0)$ | NS | $\beta_1$ | $\beta_1$ | $\beta_1$ and $\beta_2$ | $\beta_1$ |
| $F(\beta_2 | \beta_0, \beta_1)$ | NS | $\beta_1$ and $\beta_2$ | $\beta_1$ and $\beta_2$ | $\beta_2$ | neither $\beta_1$ nor $\beta_2$ |

Sig = Significant; NS = Not Significant

### Table 2. Numbering Scheme

| Fitting $\beta_1$ and then $\beta_2$ after $\beta_1$ | $F(\beta_2 | \beta_0)$ | $F(\beta_2 | \beta_0, \beta_1)$ | Sig | NS | Sig | NS |
|--------------------------------------------------|------------------------|---------------------------------|-----|-----|-----|-----|
| $F(\beta_1 | \beta_0)$ | Sig | 1 | 2 | 3 | 4 |
| $F(\beta_2 | \beta_0, \beta_1)$ | Sig | | | | |
| $F(\beta_1 | \beta_0)$ | NS | 5 | 6 | 7 |
| $F(\beta_2 | \beta_0, \beta_1)$ | Sig | | | |
| $F(\beta_1 | \beta_0)$ | NS | 8 | 9 |
| $F(\beta_2 | \beta_0, \beta_1)$ | NS | | 10 |
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:

**First Step.** Include $\beta_1$ in the model if both $F(\beta_1 | \beta_0)$ and $F(\beta_1 | \beta_0, \beta_2)$ are significant.

**Second Step.** Include $\beta_2$ in the model if both $F(\beta_2 | \beta_0)$ and $F(\beta_2 | \beta_0, \beta_1)$ are significant.

**Third Step.** If neither $\beta_1$ nor $\beta_2$ 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 $\beta_1$ and $\beta_2$ 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 $\beta_1$ nor $\beta_2$ is included or in which both $\beta_1$ and $\beta_2$ 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 $\beta^* = (\beta_1, \beta_2)$' and $\mu = X^* \beta^*$, then

$$E(P_1y^*) = P_1\mu$$
$$E(P_2y^*) = P_2\mu$$
$$E(Py^*) = P\mu$$

Let $\theta_1$ be the angle between $E(P_1y^*)$ and $E(Py^*)$ and let $\theta_2$ be the angle between
### Table 3. Full Set and Null Set Avoiding Scheme

| Fitting $\beta_1$ and then $\beta_2$ after $\beta_1$ | $F(\beta_2 | \beta_0)$ | Sig | N S | Sig | N S |
|---|---|---|---|---|---|
| $F(\beta_1 | \beta_0)$ | Sig | $\beta_1$ and $\beta_2$ | $\beta_1$ | $\beta_2$ | smaller angle |
| $F(\beta_2 | \beta_0, \beta_1)$ | Sig | $\beta_1$ and $\beta_2$ | $\beta_1$ | $\beta_2$ | smaller angle |
| $F(\beta_1 | \beta_0)$ | N S | smaller angle | $\beta_2$ | smaller angle |
| $F(\beta_2 | \beta_0, \beta_1)$ | Sig | smaller angle | $\beta_1$ | neither $\beta_1$ nor $\beta_2$ |
| $F(\beta_1 | \beta_0)$ | N S | smaller angle | $\beta_2$ | neither $\beta_1$ nor $\beta_2$ |
| $F(\beta_2 | \beta_0, \beta_1)$ | N S | smaller angle | $\beta_2$ | neither $\beta_1$ nor $\beta_2$ |

### Table 4. Null Set Avoiding Scheme

| Fitting $\beta_2$ and then $\beta_1$ after $\beta_2$ | $F(\beta_2 | \beta_0)$ | Sig | N S | Sig | N S |
|---|---|---|---|---|---|
| $F(\beta_1 | \beta_0)$ | Sig | $\beta_1$ and $\beta_2$ | $\beta_1$ | $\beta_2$ | $\beta_1$ |
| $F(\beta_2 | \beta_0, \beta_1)$ | Sig | $\beta_1$ and $\beta_2$ | $\beta_2$ | $\beta_2$ | $\beta_1$ |
| $F(\beta_1 | \beta_0)$ | Sig | $\beta_1$ and $\beta_2$ | $\beta_1$ | $\beta_2$ | $\beta_1$ |
| $F(\beta_2 | \beta_0, \beta_1)$ | Sig | $\beta_1$ and $\beta_2$ | $\beta_1$ | neither $\beta_1$ nor $\beta_2$ | $\beta_1$ |

**Effects to be included in model**

- $\beta_1$ and $\beta_2$
- $\beta_1$
- $\beta_2$
- smaller angle
- $\beta_1$
- neither $\beta_1$ nor $\beta_2$
\( E(P_2 y^*) \) and \( E(P_y^*) \), then

\[
\begin{align*}
\cos \theta_1 &= \mu P_1' P_1 \\
\cos \theta_2 &= \mu P_2' P_2
\end{align*}
\]

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

\[
\begin{align*}
\cos \theta_1 &= y^* P_1' P_1 y^* \\
\cos \theta_2 &= y^* P_2' P_2 y^*
\end{align*}
\]

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 \( \beta_1 \) nor \( \beta_2 \) has been put in the model in the first two steps, then add \( \beta_1 \) to the model if \( F(\beta_1 | \beta_0) \) is significant and add \( \beta_2 \) to the model if \( F(\beta_2 | \beta_0) \) is significant. If neither of these is significant then add \( \beta_1 \) to the model if \( F(\beta_1 | \beta_0, \beta_2) \) is significant and add \( \beta_2 \) 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 \( \beta_1 \) and \( \beta_2 \) has been added in the three previous steps and if one calls the other factor \( \gamma \), then if \( F(\gamma | \beta_0, \gamma^c) \) is significant (where \( \gamma^c \) is the factor different from \( \gamma \)) add \( \gamma \). If \( F(\gamma | \beta_0, \gamma^c) \) is significant but nothing else is significant, add \( \gamma^c \).
It should also be noted that if $\lambda_1$, $\lambda_2$ and $\lambda_3$ are the eigenvalues of $Z_1^T Z_1$ 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.

\[ ZZ = [1 \quad P1Y \quad P2Y \quad PY \quad Y] \]

<table>
<thead>
<tr>
<th></th>
<th>5.30330</th>
<th>2.12132</th>
<th>5.30330</th>
<th>5.30330</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-5.30330</td>
<td>-2.12132</td>
<td>-5.30330</td>
<td>-5.30330</td>
</tr>
<tr>
<td>2</td>
<td>0.0</td>
<td>4.63721</td>
<td>3.18198</td>
<td>3.18198</td>
</tr>
<tr>
<td>3</td>
<td>0.0</td>
<td>-4.63721</td>
<td>-3.18198</td>
<td>-3.18198</td>
</tr>
<tr>
<td>4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9.00000</td>
</tr>
</tbody>
</table>

\[ Z*VT = VT*Y \]

|   | \sqrt{SSE} \]
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
</tr>
</tbody>
</table>

DELETING THE THIRD ROW OF \( PZ \)

<table>
<thead>
<tr>
<th></th>
<th>6.44871</th>
<th>5.92789</th>
<th>8.74634</th>
<th>8.74634</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-3.82937</td>
<td>4.10701</td>
<td>0.03986</td>
<td>0.03986</td>
</tr>
<tr>
<td>2</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.00000</td>
<td>-0.00000</td>
</tr>
<tr>
<td>3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9.00000</td>
</tr>
</tbody>
</table>

\( PZ \) PREMULTIPLIED BY APPROPRIATE ELEMENTARY REFLECTOR IN ORDER TO TRIANGULARIZE IT

<table>
<thead>
<tr>
<th></th>
<th>-7.50000</th>
<th>-3.00000</th>
<th>-7.50000</th>
<th>-7.50000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>6.55861</td>
<td>4.50000</td>
<td>4.50000</td>
</tr>
<tr>
<td>2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9.00000</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>7.50000</th>
<th>3.00000</th>
<th>7.50000</th>
<th>7.50000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>6.55861</td>
<td>4.50000</td>
<td>4.50000</td>
</tr>
<tr>
<td>2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9.00000</td>
</tr>
</tbody>
</table>

SCALING TO MAKE ALL ELEMENTS HAVE ABSOLUTE VALUE LESS THAN 10

<table>
<thead>
<tr>
<th></th>
<th>8.33333</th>
<th>3.33333</th>
<th>8.33333</th>
<th>8.33333</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>7.28667</td>
<td>5.00000</td>
<td>5.00000</td>
</tr>
<tr>
<td>2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>10.0000</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.33333</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>9.16666</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>3.33333</td>
<td>7.28667</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>3.66666</td>
<td>8.01533</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>8.33333</td>
<td>5.00000</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>8.33333</td>
<td>5.00000</td>
<td>10.0000</td>
</tr>
</tbody>
</table>

Figure 2 a) Printed Output of a Run of NORTH (RANK 2)
Figure 2b) Plot Produced by a Run of NORTH (RANK 2)
\[ \|P_1y\|^2 = 100.00 \quad \text{Sig} \]
\[ \|(P-P_1)y\|^2 = 28.44 \quad \text{Sig} \]
\[ \|Py\|^2 = 128.44 \]
\[ \|(I-P)y\|^2 = 9.00 \]
\[ \|y\|^2 = 137.44 \]

\[ \cos \theta_1 = \frac{\|P_1y\|}{\|Py\|} = \frac{10}{\sqrt{128.44}} = .88237 \]
\[ \theta_1 \approx 26^\circ \]

\[ \cos \theta_2 = \frac{\|P_2y\|}{\|Py\|} = \frac{\sqrt{94.44}}{\sqrt{128.44}} = .25749 \]
\[ \theta_2 \approx 31^\circ \]

Figure 3. Situation Number 1
\[
\begin{align*}
\| P_1 y \|^2 &= 50.00 \quad \text{Sig} \\
\| (P-P_1) y \|^2 &= 98.00 \quad \text{Sig} \\
\| P y \|^2 &= 148.00 \\
\| (I-P) y \|^2 &= 9.00 \\
\| y \|^2 &= 157.00
\end{align*}
\]

\[
\cos \theta_1 = \frac{5 \sqrt{2}}{\sqrt{148}} = 0.58124
\]

\[
\theta_1 \approx 54.30°
\]

\[
\cos \theta_2 = \frac{2}{\sqrt{148}} = 0.1644
\]

\[
\theta_2 \approx 80° 30´
\]

Figure 4. Situation Number 2
\[ \|P_1 y\|^2 = 36.00 \quad \text{Sig} \]
\[ \|(P-P_1)y\|^2 = 40.96 \quad \text{Sig} \]
\[ \|P y\|^2 = 76.96 \]
\[ \|(I-P)y\|^2 = 9.00 \]
\[ \|y\|^2 = 85.96 \]

\[ \cos \theta_1 = \frac{6}{\sqrt{76.96}} = .68394 \quad \theta_1 \approx 46^\circ 50' \]

\[ \cos \theta_2 = \frac{\sqrt{74.00}}{\sqrt{76.96}} = .98058 \quad \theta_2 \approx 11^\circ 20' \]

Figure 5. Situation Number 3
\[
\begin{array}{ccc}
\|P_1y\|^2 & 4.99 & N S \\
\|(P-P_1)y\|^2 & 100.00 & \text{Sig} \\
\|Py\|^2 & 104.00 & \\
\|(I-P)y\|^2 & 9.00 & \\
\|y\|^2 & 113.00 &
\end{array}
\]

\[
\cos \theta_1 = \frac{2}{\sqrt{104}} = .19612 \\
\theta_1 \approx 78^\circ 40'
\]

\[
\cos \theta_2 = \frac{\sqrt{7.20}}{\sqrt{104.00}} = .26312 \\
\theta_2 \approx 74^\circ 40'
\]

Figure 6. Situation Number 5
\[
\begin{align*}
\|P_1 y\|^2 & = 4.00 \quad \text{N S} \\
\|(P-P_1)y\|^2 & = 98.01 \quad \text{Sig} \\
\|P_2 y\|^2 & = 101.00 \quad \text{Sig} \\
\|(P-P_2)y\|^2 & = 1.01 \quad \text{N S} \\
\|P y\|^2 & = 102.01 \\
\|(I-P)y\|^2 & = 9.00 \\
\|y\|^2 & = 111.01 \\
\end{align*}
\]

\[
\begin{align*}
\cos \theta_1 = \frac{2}{\sqrt{102.01}} & = .19802 \\
\theta_1 & \approx 78^\circ 30' \\
\cos \theta_2 = \frac{\sqrt{101}}{\sqrt{102.01}} & = .99504 \\
\theta_2 & \approx 5^\circ 40'
\end{align*}
\]

Figure 7. Situation Number 6
<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Direction</th>
<th>Expression</th>
<th>Value</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>$|P_1y|^2$</td>
<td>1.00</td>
<td>N S</td>
<td>$|P_2y|^2$</td>
<td>14.00</td>
<td>N S</td>
</tr>
<tr>
<td>$|P-P_1y|^2$</td>
<td>25.60</td>
<td>Sig</td>
<td>$|P-P_2y|^2$</td>
<td>12.60</td>
<td>N S</td>
</tr>
<tr>
<td>$|P_y|^2$</td>
<td>26.60</td>
<td></td>
<td>$|P_y|^2$</td>
<td>26.60</td>
<td></td>
</tr>
<tr>
<td>$|(I-P)y|^2$</td>
<td>9.00</td>
<td></td>
<td>$|(I-P)y|^2$</td>
<td>9.00</td>
<td></td>
</tr>
<tr>
<td>$|y|^2$</td>
<td>35.60</td>
<td></td>
<td>$|y|^2$</td>
<td>35.60</td>
<td></td>
</tr>
</tbody>
</table>

\[
\cos \theta_1 = \frac{1}{\sqrt{26.60}} = 0.19389 \\
\theta_1 \approx 78^\circ 50' \\
\cos \theta_2 = \frac{14.00}{\sqrt{26.60}} = 0.72548 \\
\theta_2 \approx 43^\circ 30'
\]

Figure 8. Situation Number 7
\[ \|P_1y\|^2 = 100.00 \text{ Sig} \]

\[ \|(P-P_1)y\|^2 = 4.00 \text{ N/S} \]

\[ \|Py\|^2 = 104.00 \]

\[ \|(I-P)y\|^2 = 9.00 \]

\[ \|y\|^2 = 113.00 \]

\[ \cos \theta_1 = \frac{10}{\sqrt{104}} = 0.98058 \]

\[ \theta_1 \approx 11^\circ 20' \]

\[ \cos \theta_2 = \frac{\sqrt{96.80}}{\sqrt{104.00}} = 0.96476 \]

\[ \theta_2 \approx 15^\circ 20' \]

Figure 9. Situation Number 8
\[
\begin{align*}
&\|P_1y\|^2 = 25.00 & \text{Sig} & \|P_2y\|^2 = 13.00 & \text{NS} \\
&\|(P-P_1)y\|^2 = 1.00 & \text{NS} & \|(P-P_2)y\|^2 = 13.00 & \text{NS} \\
&\|Py\|^2 = 26.00 & \|Py\|^2 = 26.00 \\
&\|(I-P)y\|^2 = 9.00 & \|Py\|^2 = 35.00 & \|Py\|^2 = 35.00 \\
&\|y\|^2 = 35.00 & \|y\|^2 = 35.00
\end{align*}
\]

\[
\cos \theta_1 = \frac{5}{\sqrt{26}} = 0.98058 \\
\theta_1 \approx 11^\circ 20'
\]

\[
\cos \theta_2 = \frac{\sqrt{13}}{\sqrt{26}} = \frac{\sqrt{2}}{2} = 0.70711 \\
\theta_2 \approx 45^\circ
\]

Figure 10. Situation Number 9
\[
\begin{align*}
\|P_1 y\|^2 & = 9.00 \quad \text{NS} \\
\|(P-P_1)y\|^2 & = 2.56 \quad \text{NS} \\
\|P_2 y\|^2 & = 8.50 \quad \text{NS} \\
\|(P-P_2)y\|^2 & = 3.06 \quad \text{NS} \\
\|P_1 y\|^2 & = 11.56 \\
\|(I-P)y\|^2 & = 9.00 \\
\|y\|^2 & = 20.56 \\
\|P_2 y\|^2 & = 11.56 \\
\|(I-P)y\|^2 & = 9.00 \\
\|y\|^2 & = 20.56
\end{align*}
\]

\[
\cos \theta_1 = \frac{3}{\sqrt{11.56}} = 0.88235 \\
\theta_1 \approx 28^\circ
\]

\[
\cos \theta_2 = \frac{\sqrt{8.50}}{\sqrt{11.56}} = 0.85749 \\
\theta_2 \approx 31^\circ
\]

Figure 11. Situation Number 10
Appendix B. Complete listing of NORTH (RANK 2).

MAIN PROGRAM OF NORTH (RANK 2)

COMMENT CARDS FOR MAIN PROGRAM

NORTH (RANK 2) WAS PRIMARILY WRITTEN TO APPLY THE TECHNIQUES OF THE PAPER "GRAPHICAL METHODS FOR NON-ORTHOGONAL DATA" (BY SAMUEL G. LINDLE AND DAVID M. ALLEN, BU-555-M OF THE BIOMETRICS UNIT MIXED SERIES, CORNELL UNIVERSITY, ITHACA, N.Y. 14853). HOWEVER, A CERTAIN AMOUNT OF FLEXIBILITY HAS BEEN ADDED TO MAKE IT USEFUL FOR OTHER PURPOSES. ITS SPECIFIC APPLICATIONS TO THE PAPER WILL BE DISCUSSED FIRST.

SPECIFIC APPLICATION OF NORTH (RANK 2):

FOR A COMPLETE DESCRIPTION OF THE STATISTICAL MOTIVATION AND THEORY SEE BU-555-M.

NORTH (RANK 2) USES THE SINGULAR VALUE DECOMPOSITION TO DECOMPOSE THE N BY 3 MATRIX

\[ Z_1 = [\begin{pmatrix} \mathbf{p}_1 \mathbf{p}_2 \mathbf{p}_3 \end{pmatrix} \]  

where \( \mathbf{p}_1 \), \( \mathbf{p}_2 \), and \( \mathbf{p}_3 \) are the first three columns of \( 
\begin{pmatrix} 
U \end{pmatrix} \) and \( \begin{pmatrix} \mathbf{v}_1 \mathbf{v}_2 \mathbf{v}_3 \end{pmatrix} \) are the first three columns of \( 
\begin{pmatrix} 
V \end{pmatrix} \) such that

\[ U^*U = I \quad \text{and} \quad V^*V = I \]

and \( S \) is a 3 by 3 diagonal matrix.

THE PROGRAM THEN CALCULATES THE NONZERO SUBMATRICES OF

\[ P_2 = \begin{pmatrix} 
S^*V & U^*Y \\
0 & \text{SQRT(SSE)} 
\end{pmatrix} \]

where SSE IS THE SUM OF SQUARES FOR ERROR. THE THIRD ROW OF \( P_2 \) IS DELETED IN ORDER TO PROJECT 4 SPACE ONTO 3 SPACE. THE RESULTING MATRIX IS THEN PREMULTIPLIED BY THE APPROPRIATE ELEMENTARY REFLECTOR IN ORDER TO TRIANGULARIZE IT. IN ORDER TO KEEP THE RESULTANT VECTOR CORRESPONDING TO THE ORIGINAL \( \mathbf{y} \) VECTOR IN THE FIRST OCTANT, IF EITHER OF THE TWO REMAINING ELEMENTS OF \( U^*Y \) IS NEGATIVE THE CORRESPONDING ROW IS MULTIPLIED BY -1. SINCE IT IS DIFFICULT TO REFERENCE THE POSITION OF POINTS OUTSIDE THE LIMITS OF THE GRID, IN ORDER TO KEEP ANY VECTORS FROM PROTRUDING OUTSIDE THE GRID, IF EITHER THE (1,2) ELEMENT OR THE (2,3) ELEMENT OR BOTH ARE NEGATIVE THEN THEIR VALUE IS SUBTRACTED FROM EVERY ELEMENT IN THEIR ROW AND FROM THE CORRESPONDING ELEMENT OF THE ORIGIN. THIS
SHIFTS THE VECTORS APPROPRIATELY.

NEXT THE PROGRAM FINDS THE ELEMENT OF THE RESULTING MATRIX WITH
THE LARGEST ABSOLUTE VALUE (E) AND MULTIPLIES ALL THE ELEMENTS OF THE
MATRIX BY $10^{-E}$ IN ORDER THAT ALL ELEMENTS WILL BE SCALED SO AS TO
HAVE ABSOLUTE MAGNITUDE LESS THAN OR EQUAL TO 10. THE COLUMNS OF THE
RESULTING MATRIX ARE THE TERMINAL POINTS OF THE VECTORS $p_1y$, $p_2y$,
$p_y$ AND $y$. (IF NEITHER THE $(1, 2)$ ELEMENT NOR THE $(2, 3)$ ELEMENT ABOVE WAS
NEGATIVE, THE INITIAL POINT OF THESE VECTORS IS THE ORIGINAL ORIGIN.
OTHERWISE, THE INITIAL POINT IS THE APPROPRIATELY SHIFTED ORIGIN.)

FOR THE VECTORS $(p-p_1)y$, $(p-p_2)y$, $(1-p_1)y$, $(1-p_2)y$ AND $(1-p)y$
APPROPRIATE INITIAL AND TERMINAL POINTS ARE CHOSEN FROM THESE FOUR
VECTORS ($p_1y$, $p_2y$, $p_y$ AND $y$). $x_1$ AND $x_2$ HAVE ARBITRARILY BEEN SET TO
TO BE EQUAL TO 1.1 TIMES $p_1y$ AND $p_2y$ (RESPECTIVELY). THIS CAN BE
EASILY MODIFIED IF DESIRED.

THE PROGRAM THEN USES SUBROUTINES SETUP, GRID, VECT01 AND P3D TO
DRAW THE GRID AND THE VECTORS AND/OR LINES DESIRED. DETAILED INFORMATION
CONCERNING THESE SUBROUTINES ARE CONTAINED IN THE COMMENT CARDS FOR
EACH. IN PARTICULAR, OPTIONS DESCRIBING THE PRESENCE OR ABSENCE OF
VECTORHEADS, THE TYPE OF VECTORHEADS, THE TYPE OF DARKENING IN OF
OR ABSENCE OF VECTOR LABELS, THE DISTANCE OF THE VECTOR LABELS ABOVE
(BELOW) THE LINE OR VECTOR, AND MEASUREMENTS OF THE VECTOR LABEL
CAN BE OBTAINED FROM THE COMMENT CARDS OF VECT01. THE VARIABLES IN THE
MAIN PROGRAM CORRESPONDING TO THE PARAMETERS OF THE SUBROUTINES OR IN
COMMON WITH CERTAIN SUBROUTINES WILL BE DESCRIBED BELOW ALONG WITH
INFORMATION ON THE FORMAT OF INPUT READ IN BY THE MAIN PROGRAM OF
NORTH (RANK 2).

FLEXIBLE APPLICATION OF NORTH (RANK 2):

NORTH (RANK 2) CAN BE USED WITHOUT SUBROUTINE DSVD AND THE POINTS
FROM AND TO WHICH VECTORS OR LINES MAY BE DRAWN CAN BE DIRECTLY READ
IN. THIS ALLOWS A GREAT DEAL OF FLEXIBILITY SINCE THE VECTORS NEED
NOT HAVE ANY RELATIONSHIP TO ONE ANOTHER NOR DO THEY HAVE TO BE
RESTRICTED TO THE LIMITS OF THE GRID. LIMITS REQUIRING, FOR INSTANCE,
ALL THE VECTORS TO BE CONTAINED WITHIN A 10 BY 10 BY 10 CUBE CAN BE
IMPOSED IF DESIRED (SEE THE COMMENT CARDS FOR SETUP). ALL THE OPTIONS
PROVIDED PREVIOUSLY CONCERNING VECTORHEADS AND VECTOR LABELS ARE
ADDITIONALLY AVAILABLE FOR THE FLEXIBLE APPLICATION. THE LABELS MAY,
HOWEVER, NOT BE APPROPRIATE ONES SINCE THE LABELS WERE DEVELOPED FOR
THE SPECIFIC APPLICATION. IF SUCH IS THE CASE THE COMMENT CARDS FOR
VECT01 EXPLAIN HOW TO PRODUCE THE APPROPRIATE LABELS IF THEY ARE
DESIRED.

THE SUBROUTINES CAN ALSO BE SEPARATED FROM THE MAIN PROGRAM AND
USED FOR A VARIETY OF PURPOSES.

INPUT INTO NORTH (RANK 2) AND DESCRIPTION OF VARIABLES USED:

INPVEC DETERMINES WHICH TYPE APPLICATION OF NORTH (RANK 2) IS TO BE
USED FOR THE RUN. IF INPVEC=1, THE FLEXIBLE APPLICATION OF
THE PROGRAM IS TO BE USED. IF INPVEC=2, THE SPECIFIC APPLICATION
OF THE PROGRAM IS TO BE USED.

ISTERS DETERMINES WHETHER OR NOT THE PROGRAM WILL BE USED TO PRODUCE
C PLOTS WHICH CAN BE PHOTOGRAPHICALLY REDUCED AND PLACED UNDER A STEREOSCOPE IN ORDER TO ACCENTUATE DEPTH PERCEPTION. IF ISTERS=1, THE PLOTS WILL NOT BE USED FOR THIS PURPOSE AND REGULAR PLOTS WILL BE PRODUCED. IF ISTERS=2, THE SPECIAL PLOTS NEEDED FOR STEREOSCOPIC VIEWING WILL BE PRODUCED. FOR FURTHER INFORMATION SEE THE DISCUSSION OF ISTERS IN THE COMMENT CARDS FOR SUBROUTINE SETUP.

F IS THE PARAMETER OF THE SYSTEM SUPPLIED SUBROUTINE FACTOR WHICH IS USED IN ORDER TO ENLARGE OR REDUCE THE FIGURE(S) DRAWN ACCORDING TO THE VALUE OF F. EVERY LINE DRAWN WILL BE ENLARGED (IF F IS GREATER THAN OR EQUAL TO 1) OR REDUCED (IF F IS LESS THAN OR EQUAL TO 1) IN LENGTH TO F TIMES THE LENGTH ORIGINALLY SPECIFIED.

NNN IS A SEQUENTIAL INDEX. NNN+1 IS THE NUMBER OF THE PLOT TO BE DRAWN NEXT.

IREDUC IS ALWAYS ZERO UNLESS TWO SIZES OF PLOTS ARE TO BE DRAWN. (NO MORE THAN TWO SIZES CAN BE DRAWN ON A RUN.) IT HAS NO REAL EFFECT UNLESS IT BECOMES 1. IN WHICH CASE, IT ALLOWS ONE TO AVOID DUPLICATING THE DATA CARD CONTAINING MM AND FROM EXCESSIVE RESETTING OF THE PEN WHEN A SECOND SIZE IS REQUESTED.

MMAX, NMAX, M, N, WIDTHU, WIDTHV, S, U AND V ARE VARIABLES (ARRAYS) USED IN THE MAIN PROGRAM THAT CORRESPOND TO THE PARAMETERS OF SUBROUTINE DSVD OF THE SAME NAMES. THE ARRAY

\[ ZZ = [ - P1Y \ P2Y \ PY \ Y ] \]

OF THE MAIN PROGRAM CORRESPONDS TO THE ARRAY A OF DSVD. LIKEWISE, L CORRESPONDS TO P. SEE COMMENT CARDS FOR DSVD FOR THEIR DESCRIPTION.

VT IS THE TRANSPOSE OF V.

D EVENTUALLY BECOMES THE SQUARE ROOT OF THE SUM OF SQUARES FOR ERROR AFTER TAKING ON SOME INTERMEDIATE VALUES DURING THE CALCULATIONS USED TO OBTAIN THIS.

PZ IS DEFINED ABOVE.

T IS THE RESULT OF TRIANGULARIZING PZ BY PREMULTIPLING BY THE APPROPRIATE ELEMENTARY REFLECTOR AND ALSO OF MULTIPLYING EACH ROW WHOSE LAST ELEMENT IS NEGATIVE BY -1.

P IS THE VECTOR DEFINING THE ELEMENTARY REFLECTOR I-2*P*PT USED IN TRIANGULARIZING PZ TO HELP OBTAIN T.

AS IS T SHIFTED (IF NECESSARY) AND SCALED (IF NECESSARY) AS DESCRIBED IN THE COMMENT CARDS ABOVE.

A1 IS THE AMOUNT OF SHIFT (IF NECESSARY) FOR THE FIRST ROW OF T.
A2 is the amount of shift (if necessary) for the second row of T.

ZZ, MMAX, NMAX, M, N, WITHU, WITHV, S, U, V, L, VT, D, PZ, T, P, AS,
A1 and A2 are only used if INPVEC = 2.

X, Y, and Z are singly dimensioned arrays. They specify the coordinates of points to or from which vectors or lines may be drawn.
Let PT(i) = (X(i), Y(i), Z(i)) be the i-th such point. Then lines and/or vectors may be drawn from PT(i) to PT(j) provided i is not equal to j.

C is a singly dimensioned array which is described in the comment cards of subroutines SETUP and P3D.

A corresponds to the parameter of the same name in subroutine GRID and is described in the comment cards for that subroutine.

ND is the number of points to or from which vectors or lines may be drawn.

NV is the number of vectors and lines to be drawn.

The following variables in the main program correspond to the parameters of vector enclosed within paranthesis and are discussed in the comment cards of that subroutine: S1(VHL), S2(VHM), IAH(IVHT), ICA(ITOR), OSI(IDOS), DI(VLDVF), D2(VLH), NI(SP), N2(TP), N3(VHP) and M3(VLP).

The following five statements describe the type and dimension of 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),
1PZ(4,4), P(2), E, D, R

IF space is at a premium or if more than 10 parameters are of interest for one or more plots on a run, the two 10's in the above REAL*8 statement should be changed to equal the number of parameters of interest for the plot that involves the largest number of parameters of interest in the calculations.

REAL*4 AS(3,4)
LOGICAL WITHU, WITHV
DIMENSION C(22), X(25), Y(25), Z(25)

IF space is at a premium or if more than 25 of the points PT(i) = (X(i), Y(i), Z(i)) are needed for one or more plots on a run, the three 25's in the above dimension statement should be changed to equal the number of points used for the plot that requires the most points.

COMMON C, XPV1, YPV3

BGH specific and flexible application
CALL PLOTS(DABA, CABA)
READ(I5, 191) INPVEC, ITERS
191 FORMAT(1X, I11, IX, I11)
IREUC = 0

ENLARGEMENT OR REDUCTION OF PLOTS PRODUCED F = 2.

880 CALL FACTOR(F)
C

NNN=0
IF(INPVEC.EQ.1) GO TO 192
C
SPECIFIC APPLICATION
C
SINGULAR VALUE DECOMPOSITION
338 REAC(5,100)MMAX,NMAX,M,N,U,L,WITHU,WITHV
100 FMAT(515,2L5)
READ(S,29)((ZZ(I,J),J=1,4),I=1,M)
29 FMAT(4F10.5)
WRITE(6,28)
28 FORMAT(*0

1  ' ZZ= ' PZ Y ** ' 
2  ' - ' ' 
WRITE(6,10)((ZZ(I,J),J=1,4),I=1,M)
CALL DSVDC((ZZ,MMAX,NMAX,M,N,U,L,WITHU,WITHV)
C
SQUARED NORM OF Y
E=O.
DC 730 I=1,M
730 E=E+ZZ(I,4)**2
D=E
C
TRANSPCING V
DO 733 I=1,3
DC 733 J=1,3
733 VT(I,J)=V(I,J)
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
C
UT*Y WITH THE ELEMENT IN THE THIRD ROW DELETED
DO 736 I=1,2
PZ(I,4)=ZZ(I,4)
C
SQUARE ROOT OF SUM OF SQUARES FOR ERROR
E=O.
DO 737 I=1,2
737 E=E+ZZ(I,4)**2
IF(S(3,J).LT.-0.0001.AND.S(3,I).GT.+0.0001) GO TO 52
E=E+ZZ(3,4)**2
52 D=DSQRT(D-E)
PZ(3,4)=0
DO 41 I=1,2
DO 41 J=1,4
41 PZ(4,I,J)=PZ(4,I,J)
DO 42 J=1,4
PZ(4,4,J)=PZ(4,4,J)
DO 43 J=1,3
PZ(3,J)=S(3)*VT(3,J)
PZ(4,3,4)=ZZ(3,4)
WRITE(6,31)
31 FMAT(*0

1  ' S*VT 
2  ' ' 
3  ' 0 SQRT(SSE) 

WRITE(6,10)(((PZ4(I,J)),J=1,4),I=1,3)
WRITE(6,30)

38 FORMAT(*'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 PREMULTIPLICATION BY AN ELEMENTARY REFLECTOR
E=0.
DC 90 I=1,2
90 E=E+T(I,1)**2
E=E**.5
IF(T(I,1),GE.0.)P(1)=T(I,1)+E
IF(T(I,1),LT.0.)P(1)=T(I,1)-E
P(2)=T(2,1)
E=0.
DO 4 J=1,2
4 E=E+P(J)**2
DC 1 M=1,4
R=0.
DC 5 L=1,2
5 R=R+P(L)*T(L,M)
R=2.*R/E
DC 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 FORCES Y TO BE IN FIRST OCTANT
IF(T(1,4),GE.0.) GO TO 739
DC 740 J=1,4
739 CONTINUE
IF(T(2,4),GE.0.) GO TO 741
DO 742 J=1,4
742 T(2,J)=T(2,J)
741 CONTINUE
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
DC 101 J=1,4
101 AS(1,J)=T(1,J)
C SHIFTING
IF(AS(1,2),GE.0.) GO TO 882
A1=AS(1,2)
DC 883 I=1,4
883 AS(1,1)=AS(1,1)-A1
882 IF(AS(2,3),GE.0.) GO TO 884
A2=AS(2,3)
DC 885 I=1,4
885 AS(2,1)=AS(2,1)-A2
WRITE(6,34)
34 FORMAT('O SHIFTING IN ORDER TO KEEP ALL VECTORS WITHIN THE*',
' 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
E=0.
DC 301 I=1,3
DC 301 J=1,4
IF(ABS(AS(I,J)).GT.E)E=ABS(AS(I,J))
CONTINUE
C SCALING
DC 111 I=1,3
DC 111 J=1,4
AS(I,J)=10./E*AS(I,J)
WRITE(6,35)
35 FORMAT('O SCALING TO MAKE ALL ELEMENTS HAVE ABSOLUTE VALUE LESS*',
' THAN 10''//)
DO 112 I=1,3
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('O THE ORIGIN, P1Y, X1, P2Y, X2, PY AND Y (RESPECTIVELY)*',
' ARE:*//)
WRITE(6,113)(X(I),Y(I),Z(I),I=1,7)
113 FORMAT('4F15.5')
IF(NN.NE.0) GO TO 339
192 IF(REDUC.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.,1.-3)
CALL PLOT(1.,1.,-3)
 C THE NUMBER OF PLOTS TO BE DRAWN ON THIS RUN
 READ(5,13)MM
 13 FORMAT(12)
 339 IF(INPVEG.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 FORMAT(7F7.4,3I3,2F4.2,11,11,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 FORMAT(40 POINTS THAT CAN BE USED AS INITIAL OR TERMINAL POINTS)
      OF LINES OR VECTORS//)
  WRITE(6,113)(X(I),Y(I),Z(I),I=1,ND)
  DO 50 I=1,NV
  50 FORMAT(4I3)
  CALL VEC01(N1,N2,X(N1),Y(N1),Z(N1),X(N2),Y(N2),Z(N2),N3,IAH,S1,
      S2,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 CONTINUE
  C END OF LOOP FOR FLEXIBLE APPLICATION
  GO TO 341
  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,ND
  51 FORMAT(4I3)
  CALL VEC01(N1,N2,X(N1),Y(N1),Z(N1),X(N2),Y(N2),Z(N2),N3,IAH,S1,
      S2,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 OF 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.
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

THIS SUBROUTINE COMPUTES THE SINGULAR VALUE DECOMPOSITION OF
A REAL M*N MATRIX A, I.E. IT COMPUTES MATRICES U,S, AND V
SUCH THAT

A = U * S * VT,

WHERE
U IS AN M*N MATRIX AND UT*U = I, (UT=TRANSPOSE
OF U),
V IS AN N*N MATRIX AND VT*V = I, (VT=TRANSPOSE
OF V),
AND S IS AN N*N DIAGONAL MATRIX.

DESCRIPTION OF PARAMETERS:
A = REAL*8 ARRAY. A CONTAINS THE MATRIX TO BE DECOMPOSED.

MMAX = INTEGER*4 VARIABLE. THE NUMBER OF ROWS IN THE
ARRAYS A AND U.

NMAX = INTEGER*4 VARIABLE. THE NUMBER OF ROWS IN THE
ARRAY V.

M,N = INTEGER*4 VARIABLES. THE NUMBER OF ROWS AND COLUMNS
IN THE MATRIX STORED IN A. (N<=M<=100. IF IT IS
NECESSARY TO SOLVE A LARGER PROBLEM, THEN THE
AMOUNT OF STORAGE ALLOCATED TO THE ARRAYS B, C, AND
T MUST BE INCREASED ACCORDINGLY.)

P = INTEGER*4 VARIABLE. IF P>0, THEN COLUMNS N+1, ..., N+P
OF A ARE ASSUMED TO CONTAIN THE COLUMNS OF AN M*P
MATRIX B. THIS MATRIX IS MULTIPLIED BY UT, AND UPON
EXIT, A CONTAINS IN THESE SAME COLUMNS THE N*P MATRIX
UT*B. (P>0)

WITHU, WITHV = LOGICAL*4 VARIABLES. IF WITHU=.TRUE., THEN
THE MATRIX U IS COMPUTED AND STORED IN THE ARRAY U.
SIMILARLY FOR V.

S = REAL*8 ARRAY. S(1), ..., S(N) CONTAIN THE DIAGONAL
ELEMENTS OF THE MATRIX S ORDERED SO THAT S(I)>S(I+1),
I=1, \ldots, N-1.

U, V = REAL*8 ARRAYS. U, V CONTAIN THE MATRICES U AND V.
IF WITHU=.TRUE. AND WITHV=.FALSE., THEN THE ACTUAL PARAMETER CORRESPONDING TO A AND U MAY BE THE SAME.
SIMILARLY FOR V IF WITHV=.TRUE. AND WITHU=.FALSE..

THIS SUBROUTINE IS A TRANSLATION OF AN ALGOL 60 PROCEDURE DESCRIBED IN THE ARTICLE "SINGULAR VALUE DECOMPOSITION AND LEAST SQUARES SOLUTIONS", NUM. MATH. 14 (1970), PP. 403-420.
THE TRANSLATION WAS DONE BY P. BUSINGER AT BELL TELEPHONE LABORATORIES WITH SOME CHANGES AND EDITING DONE BY R. UNDERWOOD AT STANFORD UNIVERSITY.

DATA ETA /Z3410000000000000/
DATA TOL /ZOD10000000000000/
ETA AND TOL ARE MACHINE DEPENDENT CONSTANTS WHOSE VALUES ARE 16\times(-13) AND 16\times(-52), RESPECTIVELY, ON IBM SYSTEM/360 COMPUTERS.

NP=N+P
N1=N+1
HCUSE HOLDER REDUCTION TO BIDIAGONAL FORM
C(1)=0.000
K=1
10 K1=K+1
ELIMINATION OF A(I,K), I=K+1, \ldots, M
Z=0.000
DO 20 I=K,M
20 Z=Z+A(I,K)**2
B(K)=0.000
IF (Z.LE.TOL) GO TO 70
Z=DSQRT(Z)
B(K)=Z
W=DABS(A(K,K))
C=1.000
IF (W.NE.0.000) Q=A(K,K)/W
A(K,K)=Q*(Z+W)
IF (K.EQ.NP) GO TO 70
DO 50 J=K1,NP
Q=0.000
DO 30 I=K,M
Q=Q+A(I,K)*A(I,J)
Q=Q/(Z*(Z+W))
DO 40 I=K,M
A(I,J)=A(I,J)-Q*A(I,K)
40 CONTINUE
PHASE TRANSFORMATION
C=-A(K,K)/DABS(A(K,K))
DC 60 J=K to 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=0.000
80   Z=Z+A(K,J)*J
     C(K1)=0.000
     IF (Z.LE.TOL) GO TO 130
     Z=DSQRT(Z)
     C(K1)=Z
     W=DAABS(A(K,K1))
     Q=1.000
     IF (W.NE.0.000) Q=A(K,K1)/W
     A(K,K1)=Q*(Z+W)
     DO 110 I=K1,M
          C=0.000
       DO 9C J=K1,N
          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
     C=-A(K,K1)/DAABS(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=0.000
     DC 150 K=1,N
     S(K)=B(K)
     T(K)=C(K)
150   EPS=DMIN1(EPS,S(K)+T(K))
     EPS=EPS*ETA
C
C   INITIALIZATION OF U AND V
     IF (.NOT.WITHU) GO TO 180
     DC 170 J=1,N
     DO 160 I=1,M
       U(I,J)=0.000
      DO 190 J=1,N
160       U(J,J)=1.000
190     V(I,J)=0.000
200   V(J,J)=1.000
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=0.000
   SN=1.000
   L1=L-1
   DC 280 I=L,K
      F=SN*T(I)
      T(I)=CS*T(I)
      IF (DABS(F) .LE. EPS) GO TO 290
      H=SI(I)
      W=DQRT(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 CONTINUE
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 CONTINUE
280 CONTINUE
C
C TEST FOR CONVERGENCE
290 W=S(K)
   IF (L .EQ. K) GO TO 360
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=DQRT(F+F+1.0D0)
   IF (F .LT. 0.000) G=-G
   F=((X-W)*(X+W)+(Y/(F+G)-H)*H)/X
C
C QR STEP
   CS=1.000
   SN=1.000
L1=L+1
DO 350 I=L1,K
G=T(I)
Y=S(I)
H=SN*G
G=CS*G
W=DSQRT(H**2+F**2)
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**2+F**2)
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=1,N
Q=A(I-1,J)
R=A(I,J)
A(I-1,J)=Q*CS+R*SN
A(I,J)=R*CS-Q*SN
340 CONTINUE
C
T(L)=0.000
T(K)=F
S(K)=X
GO TO 220
C
C CONVERGENCE
360 IF (W.GE.0.000) GO TO 380
S(K)=W
IF (.NOT.WITHV) GO TO 380
DO 370 J=1,N
370 V(J,K)=-V(J,K)
380 CONTINUE
C
C SORT SINGULAR VALUES
DO 450 K=1,N
  G=-1.000
  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 CONTINUE
  IF (J.EQ.K) GO TO 450
  S(J)=S(K)
  S(K)=G
  IF (.NOT.WITHU) GO TO 430
  DO 420 I=1,N
    Q=U(I,J)
    U(I,J)=U(I,K)
  420 CONTINUE
  IF (N.EQ.NP) GO TO 450
  Q=V(I,J)
  A(J,I)=A(K,I)
  440 A(K,I)=Q
  450 CONTINUE.

BACK TRANSFORMATION
IF (.NOT.WITHU) GO TO 510
DO 500 KK=1,N
  K=N1-KK
  IF (B(K).EQ.0.000) GO TO 500
  Q=-A(K,K)/DABS(A(K,K))
  DO 460 J=1,N
    U(K,J)=Q*U(K,J)
  460 CONTINUE
  DO 490 J=1,N
    Q=0.000
    DO 470 I=K,N
      Q=Q+A(I,K)*U(I,J)
    470 Q=Q/(DABS(A(K,K))*B(K))
    DO 480 I=K,N
      U(I,J)=U(I,J)-Q*A(I,K)
  490 CONTINUE
  500 CONTINUE

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.0.000) GO TO 560
    Q=-A(K,K1)/DABS(A(K,K1))
    DO 520 J=1,N
      V(K1,J)=Q*V(K1,J)
  520 CONTINUE
DO 550 J=1,N
   Q=0.000
   DO 530 I=1,K,N
  530   Q=Q+A(I,J)*V(I,J)
   C=Q/(DABS(A(I,K))*C(K))
   DO 540 I=1,K,N
  540   V(I,J)=V(I,J)-Q*A(I,J)
   CONTINUE
550  CONTINUE
560  RETURN
END

SUBROUTINE SETUP
COMMON C(22),XPV1,YPV3

INPUT INTO THIS SUBROUTINE VIA THE COMMON STATEMENT CONNECTING IT WITH THE MAIN PROGRAM OF NORTH (RANK 2):

THE FOCAL POINT: FP=(C(1),C(2),C(3))

THE OBSERVATION DIRECTION VECTOR: ODV=(C(4),C(5),C(6))

THE DISTANCE BETWEEN THE OBSERVATION POINT AND THE FOCAL PCINT: C(7)

C(7) SHOULD ALWAYS BE GREATER THAN ZERO.

THE OBSERVATION POINT: FP+C*ODV WHERE C=C(7)/NORM OF ODV

THIS SUBROUTINE DEFINES C(8) THRU C(20).

C(8) THRU C(15) ARE DISPLAYED ROWWISE AS THE NON-ZERO ELEMENTS OF THE MATRIX P:

\[
\begin{pmatrix}
-C(5)/R & C(4)/R & 0 & 0 \\
-C(4)/S & -C(5)/S & -C(6)/S & 0 \\
-C(4)*C(5)/(R*S) & -C(5)*C(6)/(R*S) & R/S & 0 \\
\end{pmatrix}
\]

WHERE R=SQRT(C(4)**2+C(5)**2) AND S IS THE NORM OF THE OBSERVATION DIRECTION VECTOR.

P IS THE PRODUCT OF TWO ASYMMETRIC PLANE ROTATIONS:

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & R/S & -C(6)/S & * \\
0 & C(6)/S & R/S & 0 \\
\end{pmatrix}
\]

-1

-1

0

0
PREMULTIPLICATION OF THE OBSERVATION DIRECTION VECTOR BY THE
SECOND ASYMMETRIC PLANE ROTATION WOULD PRODUCE THE VECTOR:

\[(0, -R, C(6))\]

AS DESIRED. SUBSEQUENTLY PREMULTIPLYING BY THE FIRST ASYMMETRIC
PLANE ROTATION WOULD PRODUCE THE VECTOR:

\[(0, -S, 0)\]

AS DESIRED. EQUIVALENTLY PREMULTIPLICATION OF THE OBSERVATION POINT
(OP) BY P WILL GIVE THE VECTOR:

\[(0, -C*S, 0)\]

PROVIDED THE FOCAL POINT IS USED AS THE ORIGIN. (THE FOCAL POINT
IS MADE THE NEW ORIGIN IN THE TRANSLATE (FIRST) STEP OF P3D BEFORE
THE P MATRIX IS USED IN THE ROTATION (SECOND) STEP. ALSO NOTE THAT
THE OBSERVATION DIRECTION VECTOR HAS FIRST BEEN ROTATED CLOCKWISE
IN THE XY PLANE \(D = \arccos(-C(5)/R) = \arcsin(C(4)/R)\) DEGREES AND THEN
ROTATED CLOCKWISE IN THE YZ PLANE \(M\) = \(\arccos(R/S) = \arcsin(-C(6)/S)\)
DEGREES. ANY VECTOR PREMULTIPLIED IN THIS WAY WOULD BE
SIMILARLY ROTATED.) THE MATRIX P SIMULTANEOUSLY PREFORMS THE
TASK OF THE TWO ASYMMETRIC PLANE ROTATIONS.

WHEN C(4) AND C(5) ARE BOTH ZERO AND C(6) IS NONZERO THE
ELEMENTS OF P ARE CREATED IN A DIFFERENT PORTION OF THE SUBROUTINE
to correspond to:

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 1 \\
0 & -1 & 0
\end{bmatrix}
\]

THE MATRIX GIVEN ABOVE ROTATES THE Y AND Z COORDINATES FOR
EVERY POINT 270 DEGREES CLOCKWISE OR 90 DEGREES COUNTER-CLOCKWISE.
(NOTE THAT IN THIS SPECIAL CASE THE SECOND ASYMMETRIC PLANE
ROTATION ABOVE IS THE IDENTITY MATRIX AND THE FIRST ASYMMETRIC
PLANE ROTATION IS JUST THE MATRIX ABOVE.) ALSO, ODV = (0, 0, C(6))
SC THAT PREMULTIPLYING ODV BY P GIVES (0, -C(6), 0). SINCE S IS THE
ABSOLUTE VALUE OF C(6), IF C(6) IS GREATER THAN 0, THIS POINT BECOMES
(0, -S, 0) AS ABOVE. IF C(6) IS LESS THAN ZERO, THE POINT BECOMES
(C, S, 0) WHICH IS NOT DESIRABLE. (THE CASE WHERE C(6) IS EQUAL
tO ZERO IS COVERED BELOW.) THUS, IF C(4) AND C(5) ARE BOTH EQUAL TO
ZERO, THEN C(6) SHOULD BE A POSITIVE NUMBER. (IN SUCH A CASE, IT
IS ALSO WISE TO MAKE C(7) LARGE BECAUSE THIS CAUSES THE PLOT
tO FILL THE PAGE. THEORETICALLY, WHEN C(7) IS EQUAL TO INFINITY
MAXIMUM FILLING UP OF THE PAGE OCCURS. PRACTICALLY, ANY MODERATELY LARGE
NUMBER WILL DO.) HAVING C(4) AND C(5) BOTH ZERO ALLOWS ONE TO LOOK
STRAIGHT DOWN ONTO THE GRID AND TO PRODUCE TWO DIMENSIONAL PLOTS.
WHEN C(4), C(5) AND C(6) ARE ALL ZERO (THE ODV IS THE ZERO
VECTOR, THE MESSAGE "THE OBSERVATION POINT IS UNDEFINED SINCE
IT IS IMPOSSIBLE TO MOVE FROM THE FOCAL POINT A POSITIVE DISTANCE
IN THE DIRECTION (0,0,0)." IS PRINTED OUT. THE PLOTTAPE IS ENDED
AND THE PROGRAM IS TERMINATED.
EXCEPT IN THE CASE COVERED IMMEDIATELY ABOVE WHEN THE ODV IS
THE ZERO VECTOR OR IN THE CASE MENTIONED IN THE COMMENT CARDS BELOW
WHEN ITERS=2 AND C(16) IS NONZERO, THE LIMITING POINTS READ IN BY
SETUP ABOVE (WHICH, FOR EXAMPLE, MIGHT BE THE VERTICES OF A 10 BY 10
BY 10 CUBE BUT IN GENERAL ARE VERTICES OF SOME FIGURE) ARE PROJECTED
ONTO TWO SPACE BY P3D WHICH MUST THUS ACCOMPANY SETUP IN THE SOURCE
DECK. C(16) AND C(18) BECOME THE MINIMUM X AND Y COORDINATES
(REPECTIVELY) OF THE PROJECTIONS. C(17) AND C(19) BECOME THE
RANGE OF THE X AND Y COORDINATES OF THE PROJECTIONS.
C(20) IS ORIGINALLY ZERO IN ORDER THAT NEITHER PLOTTING
NOR SCALING OCCURS IN SUBROUTINE P3D WHEN THE BOUNDARY LIMITATIONS
ARE BEING CREATED (OR SETUP). AFTER THIS IS COMPLETED AT THE
END OF SETUP, C(20) IS SET TO ONE SO THAT HEREAFTER P3D WILL
SCALE AND PLOT. C(21) AND C(22) ARE CREATED IN SUBROUTINE P3D
EVERY TIME IT IS CALLED.

C(20)=0.
R=C(4)**2+C(5)**2
S=R+C(6)**2
IF(R.EQ.0) GO TO 6
R=SQR(R)
S=SQR(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.

IF ITERS=1 IN THE MAIN PROGRAM OF NORTH (RANK 2) USE THE CARD:
C(16)=10.**(70)

ISTERS=1 IS THE USUAL NONSTEREOSCOPIC USE OF NORTH (RANK 2)
WHICH REQUIRES THAT C(16) THRU C(19) BE RECALCULATED FOR EACH
PLOT IF OPTIMALLY SIZED PLOTS ARE TO BE OBTAINED. THE ABOVE
FORTRAN STATEMENT (IF THE 'C' FOR MAKING IT A COMMENT CARD IS
DELETED) ALONG WITH ANOTHER STATEMENT DESCRIBED IN THE COMMENT
CARDS NEAR THE END OF SETUP CAN BE USED TO ACCOMPLISH THIS.

IF ISTERS=2 USE THE CARDS:

IF(C(16).NE.0.) GO TO 10
C(16)=10.**(70)

FOR ISTERS=2, ANY APPROPRIATE (SEE COMMENT CARDS IN MAIN PROGRAM
OF NORTH (RANK 2)) PAIR OF THE PLOTS PRODUCED BY NORTH (RANK 2)
CAN BE PHOTOGRAPHICALLY REDUCED AND PLACED UNDER A STEREOSCOPE TO
ACCENTUATE DEPTH PERCEPTION. TO PRODUCE THE APPROPRIATE PLOTS
C(16) THRU C(19) MUST NOT BE REDEFINED ON EACH PLOT. RATHER,
THE VALUES CALCULATED FOR THEM ON THE FIRST PLOT MUST BE RETAINED
THROUGHOUT. THIS IS ACCOMPLISHED BY THE ABOVE TWO FORTRAN STATE-
MENTS ALONG WITH ANOTHER STATEMENT DESCRIBED IN THE COMMENT CARDS
NEAR THE END OF SETUP. THE PURPOSE OF NOT REDEFINING C(16) THRU C(19)
IS THAT THIS GUARANTEES THAT THE PICTURE WILL BE OF A DIFFERENT
SIZE WHEN VIEWED FROM DIFFERENT OBSERVATION POINTS. RETAINING C(16)
THRU C(19) DOES NOT ALLOW THE PICTURES TO BE BLOWN UP DIFFERENTLY.
DEPTH PERCEPTION IS ACCENTUATED WHEN THE DISTANCE BETWEEN THE TWO
OBSERVATION POINTS IS SOMEWHAT LARGER THAN THE INTEROCULAR DISTANCE.
The INTEROCULAR DISTANCE FOR THE AVERAGE ADULT IS ABOUT 2.5 INCHES
(63 MILLIMETERS).

C(16)=10.**(70)
C(17)=-C(16)
C(18)=C(16)
C(19)=C(17)

LIMITING POINTS
READ(5,3) X,Y,Z,TRIP
3 FORMAT(4F5.3)

IF(TRIP.NE.0.) GO TO 1
CALL P3D(X,Y,Z,TRIP)
1 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
CONTINUE
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

IF ISTERS=1 IN THE MAIN PROGRAM OF NORTH (RANK 2) USE THE CARD:
C(20)=1.0
IF ITERS=2 USE THE CARD:

10 C(20)=1.0

RETURN

WRITE(6,9)

FORMAT('THE OBSERVATION POINT IS UNDEFINED SINCE IT IS IMPOSSIBLE TO MOVE FROM THE FOCAL POINT A POSITIVE DISTANCE IN THE DIRECTION (0,0,0).')

CALL PLTEND

STOP

END

SUBROUTINE GRID(N)

THIS SUBROUTINE DRAWS A PERSPECTIVE GRID TO HELP IDENTIFY THE POSITIONS OF POINTS ABOVE, BELOW OR IN THE PLANE OF THE GRID. SUBROUTINE P3D MUST ALWAYS ACCOMPANY SUBROUTINE GRID IN THE FCRTAN SOURCE DECK.

DESCRIPTION OF THE SYMBOLS USED IN SUBROUTINE GRID:

N = INTEGER*4 VARIABLE. IT IS THE ONLY PARAMETER OF SUBROUTINE GRID AND THE ONLY INPUT FROM THE CALLING PROGRAM SINCE THERE IS NO COMMON STATEMENT. IT IS THE TOTAL NUMBER OF DASHES AND SPACES BETWEEN DASHES ON EACH LINE OF THE GRID TO BE DRAWN.

K = INTEGER*4 VARIABLE. IT SPECIFIES WHETHER THE PEN WILL BE UP OR DOWN WHILE PLOTTING. K IS ALWAYS EITHER 2 OR 3. IF K=2 THE PEN IS DOWN WHILE PLOTTING AND IF K=3 IT IS UP WHILE PLOTTING.

L = INTEGER*4 VARIABLE. IT INDEXES THE LOOP THAT PROVIDES THE TWO STEPS FOR THE GRID. IN STEP 1 (L=1) ELEVEN "VERTICAL" DASHED LINES ARE PRODUCED. IN STEP 2 (L=2) ELEVEN "HORIZONTAL" DASHED LINES ARE PRODUCED.

X1,Y1,X2,Y2 ARE REAL*4 VARIABLES WHICH ARE THE COORDINATES OF THE POINTS (X1,Y1) AND (X2,Y2) WHICH ARE USED IN INTERMEDIATE STEPS TO CALCULATE X AND Y.

X AND Y ARE REAL*4 VARIABLES WHICH ARE THE COORDINATES OF THE POINT (X,Y) WHICH DETERMINES VIA SUBROUTINE P3D THE DIRECTION AND LENGTH OF EACH DASH AND SPACE BETWEEN DASH WHICH IS TO BE PLOTTED.

TWO STEPS ARE INVOLVED IN DRAWING THE GRID. THE FIRST STEP CAN BE DESCRIBED AS FOLLOWS: BEGINNING WITH THE PEN IN THE DOWN POSITION AND BY ALTERNATING THE UP AND DOWN POSITION OF THE PEN SUBROUTINE GRID, AIDED BY SUBROUTINE P3D, DRAWS A DASHED LINE FROM THE POINT ON THE PLOTTER PAPER CORRESPONDING TO THE ORIGIN 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 DIRECTIONS OF THE DASHES AND THE SPACES BETWEEN THE DASHES DEPEND
C ON THEIR RESPECTIVE DISTANCE AND DIRECTION FROM THE OBSERVATION
C POINT 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 CORRESPONDING 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 CORRESPOND 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 RESPONDING TO \((10,1,0)\) TO THE POINT CORRESPONDING TO \((1,1,0)\); FROM
C THE POINT CORRESPONDING TO \((0,1,0)\) TO THE POINT CORRESPONDING TO
C \((0,2,0)\) AND SO ON UNTIL ELEVEN "HORIZONTAL" DASHED LINES ARE
C DRAWN EACH WITH THE GREATEST INTEGRAL VALUE OF \(N/2\) PLUS 1 DASHES.
C
K=2
DO 1 L=1,2
   K=5-K
   X1=0.
   Y1=0.
   CALL P3D(0.,0.,0.,0.,3)
   DO 1 J=1,N
      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.,0.,K)
   IF(L.EQ.2) CALL P3D(Y,X,0.,0.,K)
2 CONTINUE
   K=5-K
   X1=X1+1
   Y1=Y2
   IF(L.EQ.1) CALL P3D(X1,Y1,0.,0.,3)
   IF(L.EQ.2) CALL P3D(Y1,X1,0.,0.,3)
1 CONTINUE
RETURN
END
SUBROUTINE VECTOl(IIP, ITP, XIP, YIP, ZIP, XTP, YTP, ZTP, IVHP, IVHT, VHL, NRT30001
IVHHW, ITDK, DOS, IVLP, VLDFV, VLH) NRT30002
DIMENSION XSI(11,17), YSI(11,17) NRT30003
CMN (122), XPV1, YPV3 NRT30004
REAL*8 NDV, NPDV NRT30005
INTEGER SN NRT30006

SUBROUTINE VECTOl IS USED TO DRAW LINES AND/OR VECTORS ON OR ABOVE THE
GRID PRODUCED BY SUBROUTINE GRID. IN CERTAIN CASES APPROPRIATE LABELS
ARE OPTIONALLY AVAILABLE FOR THE LINES AND/OR VECTORS DRAWN. ADDI-
TIONAL DATA CARDS CAN BE PROVIDED IF OTHER LABELS THAN THOSE
SPECIFIED ARE DESIRED. SUBROUTINES SETUP AND P3D MUST ACCOMPANY
VECTOl IN THE FORTRAN SOURCE DECK. VECTOl CAN BE USED WITH OR
WITHOUT SUBROUTINE GRID. A PERSPECTIVE GRID IS, HOWEVER, OFTEN
HELPFUL IN IDENTIFYING THE POSITIONS OF THE POINTS OF INTEREST.

DESCRIPTION OF THE PARAMETERS OF SUBROUTINE VECTOl:

IIP, ITP, IVHP, IVHT, ITDK AND IVLP ARE INTEGER*4 VARIABLES. XIP,
YIP, ZIP, XTP, YTP, ZTP, VHL, VLHHW, DOS, VLDFV AND VLH ARE REAL*4
VARIABLES.

INDEX FOR INITIAL POINT OF LINE OR VECTOR: IIP
INDEX FOR TERMINAL POINT OF LINE OR VECTOR: ITP
INITIAL POINT OF LINE OR VECTOR: IP=(XIP, YIP, ZIP)
TERMINAL POINT OF LINE OR VECTOR: TP=(XTP, YTP, ZTP)
INDICATOR OF THE PRESENCE (OR ABSENCE) OF A VECTORHEAD: IVHP
INDICATOR OF THE TYPE OF VECTORHEAD TO BE USED: IVHT
LENGTH OF THE VECTORHEAD IN INCHES: VHL*F
(F IS THE PARAMETER OF THE SYSTEM SUPPLIED SUBROUTINE FACTOR
WHICH IS CALLED IN THE MAIN PROGRAM OF NORTH (RANK 2) IN ORDER
to ENLARGE OR REDUCE THE FIGURE(S) DRAWN ACCORDING TO THE
VALUE OF F. F IS DESCRIBED IN THE COMMENT CARDS OF THE MAIN
PROGRAM OF NORTH (RANK 2).)
ONE HALF THE WIDTH OF THE VECTORHEAD IN INCHES: VHHW*F

INDICATOR OF THE TYPE OF DARKENING IN OF THE VECTORHEAD TO
BE USED: ITDK

DISTANCE OF SHRINKING (IN INCHES) TO BE USED IN THE DARKENING
IN OF THE VECTORHEAD: DOS*F

(TDOS=2*DOS)

INDICATOR OF THE PRESENCE (OR ABSENCE) OF A LABEL FOR THE LINE
OR VECTOR DRAWN AND, IF PRESENT, THE POSITIONING OF THE LABEL
WITH RESPECT TO THE LINE OR VECTOR: IVLP

DISTANCE IN INCHES OF THE LABEL FROM (ABOVE OR BELOW) THE
VECTOR: VLDFVF

HEIGHT IN INCHES OF THE LABEL: VLH*F
(THE LENGTH OF THE LABEL VARIES ACCORDING TO THE NUMBER OF
SYMBOLS AND THE LENGTH OF THE SYMBOLS USED TO MAKE THE LABEL.)

OTHER SYMBOLS USED IN SUBROUTINE VECT01:

NPS AND KS ARE INTEGER*4 ARRAYS. XS,YS,XSI AND YSI ARE REAL*4
ARRAYS. K1,K2,N AND N ARE INTEGER*4 VARIABLES. XDV,YDV,ZDV,
T,XPDV,YPDV,ZPDV,XMP,YMP,ZMP,SVLH,SXIP,SYIP,SZIP,SXPDV,SYPDV,
Z2PDV AND SNPDV ARE REAL*4 VARIABLES. NDV AND NPDV ARE REAL*8
VARIABLES.

DIRECTION VECTOR FROM INITIAL POINT TO TERMINAL POINT: DV=
(XDV,YDV,ZDV)=TP-IP

SQUARE OF THE X AND Y COORDINATES OF THE DIRECTION VECTOR: T

A VECTOR PERPENDICULAR TO THE DIRECTION VECTOR: PDV=(XPDV,YPDV,
ZPDV)

NORM OF THE DIRECTION VECTOR: NOV

NORM OF VECTOR PERPENDICULAR TO THE DIRECTION VECTOR: NPDV

MIDPOINT OF THE LINE OR VECTOR TO BE DRAWN: MP=(XMP,YMP,ZMP)
=(IP+TP)/2

NUMBER OF POINTS USED TO DRAW THE J-TH ELEMENTARY SYMBOL:
NPS(J),J=1,2,...,17

THE FOLLOWING DATA INITIALIZATION STATEMENT DECLARES THE TYPE AND
DEFINES THE INITIAL VALUES OF NPS(J):

INTEGER NPS(17),10,6,4,6,5,9,11,2,8*0/

THE X AND Y COORDINATES FOR EACH OF THE NPS(J) POINTS USED TO
DRAW THE J-TH ELEMENTARY SYMBOL: (XS(I,J),YS(I,J)),I=1,...,NPS(J)

THE FOLLOWING TWO DATA INITIALIZATION STATEMENTS DECLARE THE TYPE AND
DEFINE THE INITIAL VALUES OF XS(I,J) AND YS(I,J):

REAL XS(11,17)/
1 0., 0., 5., 8., 9., 10., 9., 8., 5., 0., 0., 0.,
2 0., 5., 5., 0., 5., 10., 0., 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., 4., 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.,
THE POSITION OF THE PEN (EITHER UP OR DOWN) WHILE IT IS MOVING FROM ONE POINT TO THE NEXT IN ORDER TO DRAW THE J-TH ELEMENTARY SYMBOL: KS(i,j), i = 1, ..., NPS(J)

THE FOLLOWING DATA INITIALIZATION STATEMENT DECLARES THE TYPE AND DEFINES THE INITIAL VALUES OF KS(i,j):

INTEGER KS(i,j) /

Interimmediate scaling of XSI(I,J) and YSI(I,J): XSI(I,J) AND YSI(I,J)

Final three dimensional point corresponding to the original point [XSI(I,J), YSI(I,J)]: [XSF, YSF, ZSF] (AFTER P3D IS CALLED, EACH OF THESE THREE DIMENSIONAL POINTS ARE PLOTTED IN PERSPECTIVE IN TWO DIMENSIONAL SPACE. THE UP OR DOWN POSITION OF THE PEN IS DETERMINED BY THE VALUE OF K=KS(I,J) FOR EACH POINT.)

Symbol number: SN

The seventeen elementary symbols presently supplied can be identified by symbol number as follows:

<table>
<thead>
<tr>
<th>SN</th>
<th>ELEMENTARY SYMBOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>p</td>
</tr>
<tr>
<td>2</td>
<td>y</td>
</tr>
<tr>
<td>3</td>
<td>x</td>
</tr>
<tr>
<td>4</td>
<td>i</td>
</tr>
<tr>
<td>5</td>
<td>1 (SUBSCRIPT)</td>
</tr>
<tr>
<td>6</td>
<td>2 (SUBSCRIPT)</td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>
(THE SYMBOLS CORRESPONDING TO SYMBOL NUMBERS 10 THRU 17 ARE SIMPLY
TRANSLATIONS OF THE SYMBOLS CORRESPONDING TO 1, 2, 5 AND 6. ALSO THE
"1" FOR SYMBOL NUMBERS 5 AND 14 AND THE "2" FOR SYMBOL NUMBERS 6 AND 16
ARE IN SLIGHTLY DIFFERENT POSITIONS EVEN THOUGH THEY APPEAR TO BE IN
THE SAME POSITION BECAUSE THE KEYPUNCH CAN NOT DISCRIMINATE THIS.
THE "1" IN SYMBOL NUMBER 5 CORRESPONDS TO THE "1" IN THE LABEL "PIY"
(THE "1" IS REALLY A SUBSCRIPT BUT SUBSCRIPTS ARE NOT AVAILABLE
ON IBM KEYPUNCHES). THE "1" IN "X1" (THE "1" IS A SUBSCRIPT)
IS LOCATED IN A SLIGHTLY DIFFERENT POSITION THAN THE "1" IN "PIY".
LIKEWISE THE "2" IN SYMBOL NUMBER 6 CORRESPONDS TO THE "2" IN "P2Y"
AND THEREFORE IS IN A SLIGHTLY DIFFERENT POSITION THAN THE "2" IN "X2"

LABEL NUMBER: LN

THE ELEVEN DIFFERENT LABELS PRESENTLY SUPPLIED CAN BE IDENTIFIED
BY LABEL NUMBER AND BY SEQUENCE OF SYMBOL NUMBERS (FOR THE ELEMENTARY
SYMBOLS THAT ARE USED TO PRODUCE THEM) AS FOLLOWS:

<table>
<thead>
<tr>
<th>LN</th>
<th>LABEL</th>
<th>SYMBOL NUMBER SEQUENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P1Y</td>
<td>1, 5, 12</td>
</tr>
<tr>
<td>2</td>
<td>X1</td>
<td>3, 14</td>
</tr>
<tr>
<td>3</td>
<td>P2Y</td>
<td>1, 6, 12</td>
</tr>
<tr>
<td>4</td>
<td>X2</td>
<td>3, 16</td>
</tr>
<tr>
<td>5</td>
<td>Y</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>(P-P1)Y</td>
<td>7, 10, 9, 11, 15, 8, 13</td>
</tr>
<tr>
<td>7</td>
<td>(I-P1)Y</td>
<td>7, 4, 9, 11, 15, 8, 13</td>
</tr>
<tr>
<td>8</td>
<td>(I-P2)Y</td>
<td>7, 4, 9, 11, 17, 8, 13</td>
</tr>
<tr>
<td>9</td>
<td>(P-P2)Y</td>
<td>7, 10, 9, 11, 17, 8, 13</td>
</tr>
<tr>
<td>10</td>
<td>(I-P2)Y</td>
<td>7, 4, 9, 11, 17, 8, 13</td>
</tr>
</tbody>
</table>

THE VARIABLE USED TO SAVE THE ORIGINAL VALUE OF VHL: SVHL
(SIMILARLY SVHHW SAVES THE ORIGINAL VALUE OF VHHW AS DO EACH
OF THE FOLLOWING VARIABLES SAVE THE ORIGINAL VALUE OF THE
VARIABLE IN PARENTHESIS NEXT TO THEM: SVLH(VLH), SXIP(XIP),
SYIP(YIP), SZIP(ZIP), SXPDV(XPDV), SYPDV(YPDV), SZPDV(2PDV),
SAPDV(NPDV)).

I1P AND ITP ARE POSITIVE INTEGERS BETWEEN 1 AND NO. NO IS THE
NUMBER OF POINTS FROM OR TO WHICH LINES OR VECTORS CAN BE DRAWN.
IT IS READ IN THE MAIN PROGRAM OF NORTH (RANK 2) ALONG WITH NV
WHICH IS THE NUMBER OF LINES AND VECTORS DRAWN. THE VALUES OF I1P
AND ITP UNIQUELY SPECIFY A VALUE OF LN.
IF IVHP IS EQUAL TO 999 NO VECTORHEAD IS DRAWN. OTHERWISE A
VECTORHEAD IS DRAWN.

IF IVHT=1 AND T IS NONZERO, PDV IS SET EQUAL TO (XDV*ZDV,
YDV*ZDV,-T). BESIDES BEING PERPENDICULAR TO THE DIRECTION VECTOR,
THIS VECTOR IS IN THE PLANE OF THE GRID SINCE IT EQUALS ZDV*(XDV,
YDV,-T/ZDV). CONSEQUENTLY, THE VECTORHEAD DRAWN WILL BE IN THE PLANE
OF THE GRID. IF IVHT=2 AND T IS NONZERO, PDV IS SET EQUAL TO (-YDV,XDV,0)
THIS VECTOR IS PERPENDICULAR TO THE DIRECTION VECTOR. IT IS ALSO
PERPENDICULAR TO THE PLANE OF THE GRID. CONSEQUENTLY, THE VECTOR
HEAD WILL BE PERPENDICULAR TO THE PLANE OF THE GRID. IF T IS ZERO,
PDV IS SET EQUAL TO (1,0,0) SO THAT THE VECTORHEAD WILL BE DRAWN
IN THE XZ PLANE. IF WISHED, THE PROGRAM CAN BE EASILY MODIFIED SO
THAT THE VECTORHEAD WILL BE IN THE YZ PLANE INSTEAD OF THE XZ PLANE.
TC ACCOMPLISH THIS, SIMPLY CHANGE THE TWO CARDS

XPDV=1.
YPDV=0.

WHICH ARE FOUND AFTER THE CARD
IF(T.GT.0) GO TO 110
TC THE FOLLOWING TWO CARDS

XPDV=0.
YPDV=1.

THERE ARE THREE WAYS TO DARKEN IN A VECTORHEAD. ALL BEGIN BY
DRAWING A TRIANGLE WITH LENGTH (ALTITUDE) EQUAL TO SVHL AND WITH
WIDTH OF THE BASE EQUAL TO 2*SVHHW. THE APEX OF THE TRIANGLE IS AT
THE TERMINAL POINT (TP) OF THE VECTOR. A LARGE NUMBER OF SUCCESSIVELY
SMALLER TRIANLES ARE THEN DRAWN FILLING IN THE SPACE INTERIOR TO THE
ORIGINAL TRIANGLE. THE METHODS OF DARKENING IN OF THE VECTORHEAD
DIFFER IN WHAT TYPE OF SMALLER TRIANGLES ARE USED. IF ITDK=1, THE
SMALLER TRIANGLES ARE SIMILAR TO THE ORIGINAL TRIANGLE AND THEIR
APEXIAL POINT IS THE TERMINAL POINT OF THE VECTOR TO BE DRAWN.
IF ITDK=2, THE SMALLER TRIANGLES ARE AGAIN SIMILAR TO THE ORIGINAL
TRIANGLE BUT THEIR APEXIAL POINTS CHANGE IN SUCH A WAY THAT THEIR
CENTROIDS ARE ALL IDENTICAL TO THE CENTROID OF THE ORIGINAL TRIANGLE.
IF ITDK=3, THE SMALLER TRIANGLES ALL HAVE THE TERMINAL POINT AS THEIR
APEXIAL POINT BUT THE SMALLER TRIANGLES ARE NOT SIMILAR TO THE
ORIGINAL ONE. RATHER THEIR LENGTH (ALTITUDE) IS MAINTAINED CONSTANT
AND EQUAL TO SVHL. THEIR WIDTH IS THE CHANGING FACTOR.

IF ITDK=1 AND DOS IS GREATER THAN OR EQUAL TO SVHL OR IF ITDK=2
AND DOS IS GREATER THAN SVHL/2 OR IF ITDK=3 AND DOS IS GREATER THAN
OR EQUAL TO VHHW, THEN THE VECTORHEAD WILL BE DRAWN BUT NOT DARKENED
IN. (THAT IS, THE ORIGINAL TRIANGLE DESCRIBED ABOVE WILL BE DRAWN
BUT NOT THE SMALLER TRIANGLES.)

IF IVLP=999, NO VECTOR LABEL WILL BE DRAWN. IF IVLP IS NOT
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 CALL P3D(XIP,YIP,ZIP,3)
CALL P3D(XTP,YTP,ZTP,2)
IF(IHVP.EQ.999.AND.IVLP.EQ.999)RETURN
XDV=XTP-XIP
YDV=YTP-YIP
ZDV=ZTP-ZIP
T=XDV**2+YDV**2
IF(T.GT.0.)GO TO 110
XPDV=1.
YPDV=0.
ZPDV=0.
GO TO 220
110 IF(IHVT.EQ.2)GO TO 33
XPDV=XOV*ZDV
YPDV=YOV*ZDV
ZPDV=-T
GO TO 220
33 XPDV=-YDV
YPDV=XDV
ZPDV=0.0
GO TO 220
220 CONTINUE
NDV=XDV**2+YDV**2+ZDV**2
NDV=DSQRT(NDV)
NPDV=XPDV**2+YPDV**2+ZPDV**2
NPDV=DSQRT(NPDV)
IF(IHVP.EQ.999.AND.IVLP.NE.999)GO TO 991
SVHL=VHL
SVHHW=VHHW
IDS=1
Cl=-VHL/NDV
C2=VHHW/NPDV
XPPDV=XTP+C1*XDVC2*XPDV
YPPDV=YTP+C1*YDV+C2*YPDV
ZPPDV=ZTP+C1*ZDV+C2*ZPDV
XMPDV=XTP+C1*XDVC2*XPDV
YMPDV=YTP+C1*YDV+C2*YPDV
ZMPDV=ZTP+C1*ZDV+C2*ZPDV
CALL P3D(XPPDV,YPPDV,ZPPDV,2)
CALL P3D(XMPDV,YMPDV,ZMPDV,2)
IF(ITOK.NE.2.OR.IOS.EQ.1)CALL P3D(XTP,YTP,ZTP,2)
IF(ITOK.EQ.2.AND.IOS.EQ.1)CALL P3D(XTP,YTP,ZTP,2)
IF(ITOK.NE.1)GO TO 1
VHL=VHL-DOS
VHHW=VHL-SVHHW/WSL
IF(VHL.LE.DOS)GO TO 77
GO TO 55
CCONTINUE
IF(ITOK.NE.2) GO TO 2
VHL=VHL-DOS
VHHW=VHL*SVHHW/SVHL
I0S=IOS+1
CC1=-(SVHL-VHL)/NDV
XTIP=XTP+CC1*XDV
YTIP=YP+CC1*YDV
ZTIP=ZTP+CC1*ZDV
CALL P3D(XTIP, YTIP, ZTIP, 3)
TG0S=2*DOS
IF(VHL.LT.(SVHL+TG0S)/2.) GO TO 77
GO TO 55
CCONTINUE
IF(ITOK.NE.3) GO TO 3
VHHW=VHHW-DOS
IF(VHHW.LE.0.) GO TO 77
GO TO 55
CCONTINUE
VHL=SVHL
VHHW=SVHHW
991 IF(ILVPL.EQ.999) RETURN
IF(ITP.LE.IIP) GO TO 989
IF(ITP.EQ.4.OR.IIP.EQ.5.OR.IIP.GE.7) GO TO 989
IF(ITP.EQ.2.AND.3) GO TO 989
IF(ITP.EQ.4.AND.5) GO TO 989
NPS(10)=NPS(1)
NPS(11)=NPS(1)
NPS(12)=NPS(2)
NPS(13)=NPS(2)
NPS(14)=NPS(5)
NPS(15)=NPS(5)
NPS(16)=NPS(6)
NPS(17)=NPS(6)
DC 401 I=1,11
KS(I,10)=KS(I,1)
KS(I,11)=KS(I,1)
KS(I,12)=KS(I,2)
KS(I,13)=KS(I,2)
KS(I,14)=KS(I,5)
KS(I,15)=KS(I,5)
KS(I,16)=KS(I,6)
KS(I,17)=KS(I,6)
YS(I,10)=YS(I,1)
YS(I,11)=YS(I,1)
YS(I,12)=YS(I,2)
YS(I,13)=YS(I,2)
YS(I,14)=YS(I,5)
YS(I,15)=YS(I,5)
YS(I,16)=YS(I,6)
YS(I,17)=YS(I,6)
XS(I,10)=XS(I,1)+8
XS(I,11)=XS(I,1)+28
XS(I,12) = XS(I,1,2) + 12
XS(I,13) = XS(I,1,2) + 48
XS(I,14) = XS(I,1,5) + 10
XS(I,15) = XS(I,1,5) + 28
XS(I,16) = XS(I,1,6) + 10

401
XS(I,17) = XS(I,1,6) + 28
IF (IIP.EQ.1) LN = ITP - 1
IF (IIP.EQ.2) LN = ITP + 1
IF (IIP.EQ.4) LN = ITP + 4
IF (IIP.EQ.6) LN = ITP + 2
SVLH = VHL
IF (LN.EQ.1.OR.LN.EQ.3.OR.LN.EQ.5) VHL = VHL * 2.2
IF (LN.EQ.2.OR.LN.EQ.4) VHL = 1.7 * VHL
IF (LN.GE.7) VHL = VHL * 4.6
SVLH = VHL
SXIP = XIP
SYIP = YIP
SZIP = ZIP
IF (LN.EQ.2.OR.LN.EQ.4) XIP = XTP
IF (LN.EQ.2.OR.LN.EQ.4) YIP = YTP
IF (LN.EQ.2.OR.LN.EQ.4) ZIP = ZTP
XMP = (ZIP + XTP) / 2
YMP = (ZIP + YTP) / 2
ZMP = (ZIP + ZTP) / 2
XIP = SXIP
YIP = SYIP
ZIP = SZIP
DO 111 J = 1, 17
DO 111 I = 1, 11
IF (IIP.EQ.0) YSI(I, J) = YS(I, J)
IF (IIP.EQ.0) YSI(I, J) = YS(I, J) - 40
IF (LN.EQ.10) YSI(I, J) = YS(I, J) - 40
IF (LN.EQ.11) YSI(I, J) = -YS(I, J) + 20
XSI(I, J) = XS(I, J) - VHL / SVLH * 10.
YSI(I, J) = YSI(I, J) + 10.
XSI(I, J) = XSI(I, J) / 20 * 2 * SVLH
YSI(I, J) = YSI(I, J) / 20 * 2 * SVLH
CONTINUE
GO TO 987
SXPDV = XPDV
SYPDV = YPDV
SZPDV = ZPDV
SNPDV = NPDV
XPDV = YPDV
YPDV = XDV
ZPDV = 0.0
NPDV = XPDV ** 2 + YPDV ** 2 + ZPDV ** 2
NPDV = SQRT(NPDV)
GO TO 987
CONTINUE
N = 7
GO TO 11
SN = 1
N = 8
GO TO 11
N = NN = NPS(SN)
DO 2C2 I = 1, NN
XS = XMP + (XDV / NDV) * XSI(I, SN) - (XPDV / NPDV) * YSI(I, SN)
YSF = YMP + (YCV/NPV)*XSI(1,SN) - (ZPDV/NPDV)*YSI(1,SN)
ZSF = ZMF + (ZOV/NDV)*XSI(1,SN) - (ZPDV/NPDV)*YSI(1,SN)
K = KS(1,SN)

202 CALL PSO(XSF,YSF,ZSF,K)
GO TO (101,101,101,101,101,101,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,101),N
8 IF(LN.EQ.1) SN=5
N=N+1
IF(LN.EQ.1) GO TO 5
9 IF(LN.EQ.3) SN=6
N=N+1
IF(LN.EQ.3) GO TO 5
10 IF(LN.EQ.1 .OR. LN.EQ.3 .OR. LN.EQ.5) SN=12
N=N+1
IF(LN.EQ.1 .OR. LN.EQ.3 .OR. LN.EQ.5) GO TO 5
11 IF(LN.NE.2 .AND. LN.NE.4) GO TO 14
SN=3
N=N+1
GO TO 5
12 IF(LN.EQ.2) SN=14
N=N+1
IF(LN.EQ.2) GO TO 5
13 IF(LN.EQ.4) SN=16
N=N+1
IF(LN.EQ.4) GO TO 5
14 IF(LN.NE.6) GO TO 15
SN=2
N=N+1
GO TO 5
15 IF(LN.LT.7) GO TO 24
SN=7
N=N+1
GO TO 5
16 IF(LN.EQ.7 .OR. LN.EQ.10) SN=10
N=N+1
IF(LN.EQ.7 .OR. LN.EQ.10) GO TO 5
17 IF(LN.EQ.8 .OR. LN.EQ.9 .OR. LN.EQ.11) SN=4
N=N+1
IF(LN.EQ.8 .OR. LN.EQ.9 .OR. LN.EQ.11) GO TO 5
18 IF(LN.GE.7) SN=9
N=N+1
IF(LN.GE.7) GO TO 5
19 IF(LN.GE.7) SN=11
N=N+1
IF(LN.GE.7) GO TO 5
20 IF(LN.EQ.7 .OR. LN.EQ.8) SN=15
N=N+1
IF(LN.EQ.7 .OR. LN.EQ.8) GO TO 5
21 IF(LN.EQ.10 .OR. LN.EQ.11) SN=17
N=N+1
IF(LN.EQ.10 .OR. LN.EQ.11) GO TO 5
22 IF(LN.GE.7) SN=8
N=N+1
IF(LN.GE.7) GO TO 5
23 IF(LN.GE.7) SN=13
N+1
101 IF (LN.GE.7) GO TO 5
107 WRITE(6,997)
997 FORMAT(* NOT WITHIN 8-24*)
GO TO 24
989 WRITE(6,988)
988 FORMAT(* LABELS HAVE NOT BEEN SUPPLIED FOR THIS COMBINATION*
'1', 'OF IIP AND ITP,*)
24 CONTINUE
IF (LN.NE.10.AND.LN.NE.11) GO TO 981
100 XPDV=SXPDV
101 YPDV=SYPDV
102 ZPDV=SZPDV
103 NPDV=SNPDV
981 CONTINUE
VLH=SVLH
RETURN
END

SUBROUTINE P3D(X,Y,Z,IC)

THIS SUBROUTINE PLOTS THREE DIMENSIONAL POINTS IN PERSPECTIVE
ON TWO DIMENSIONAL PAPER. IT MUST BE ACCOMPANIED BY SUBROUTINE
SETUP IN THE FORTRAN SOURCE DECK.

THE COMMENT CARDS OF SETUP DESCRIBE THE VARIABLES C(1) THRU
C(20) CONNECTED TO P3D BY THE COMMON STATEMENT. C(21) AND C(22)
ARE DESCRIBED BELOW. THE PARAMETERS X,Y AND Z OF SUBROUTINE P3D
ARE REAL*4 VARIABLES. THEY ARE THE COORDINATES OF THE POINT (X,Y,Z).
THE PARAMETER IC IS AN INTEGER*4 VARIABLE. IT INDICATES WHETHER THE
PEN IS IN AN UP OR A DOWN POSITION. IF IC=3 THE PEN IS UP AND IF
IC=2 THE PEN IS DOWN.

P3D IS COMPOSED OF FIVE STEPS. IN THE FIRST STEP EACH POINT
(X,Y,Z) IS TRANSLATED BY -FP=(-C(1),-C(2),-C(3)) IN ORDER TO MAKE
THE FOCAL POINT THE NEW ORIGIN. IN THE SECOND STEP, EACH POINT
PRODUCED IN THE FIRST STEP IS ROTATED BY THE APPROPRIATE P MATRIX
(CREATED AND DESCRIBED IN THE COMMENT CARDS FOR SETUP) SUCH THAT THE
OBSERVATION DIRECTION VECTOR BECOMES (0,-S,0). IN THE THIRD STEP AN
IMAGINARY LINE IS DRAWN FROM EACH POINT PRODUCED IN THE SECOND STEP
TO THE POINT (0,-S,0). THE POINT OF INTERSECTION OF THIS LINE WITH
THE XZ PLANE IS OBTAINED. THIS INTERSECTION IS THE PERSPECTIVE
PROJECTION OF THE POINT PRODUCED IN STEP TWO ONTO THE XZ PLANE. IF
C(20) IS ZERO, CONTROL IS GIVEN BACK TO SETUP SINCE THE BOUNDARY
LIMITATIONS (MINIMUM AND RANGE OF EACH OF THE X AND Y COORDINATES)
OF THE PROJECTED 10 BY 10 BY 10 CUBE NEED TO BE SPECIFIED. EQUI-
VALENTLY, C(16) THRU C(19) NEED TO BE CREATED OR REDEFINED.

IF C(20) IS NONZERO, THE PROGRAM PROCEEDS TO STEP FOUR OF P3D
WHERE THE VALUES GIVEN TO C(16) THRU C(19) DURING THE LAST TIME THAT
C(20) WAS ZERO, ARE USED. EACH POINT PRODUCED IN STEP THREE IS SCALED
SO THAT THE DIMENSIONS OF THE RESULTING FIGURE ARE 10*F BY 8*F. (F
IS THE PARAMETER OF THE SYSTEM SUPPLIED SUBROUTINE FACTOR WHICH IS
CALLED IN THE MAIN PROGRAM OF NCRTH (RANK 2) IN ORDER TO ENLARGE OR
REDUCE THE FIGURE(S) DRAWN ACCORDING TO THE VALUE OF F. F IS DESCRIBED
IN THE COMMENT CARDS OF THE MAIN PROGRAM OF NORTH (RANK 2).)
IN THE LAST STEP, FOR EVERY POINT \((C(21),C(22))\) PRODUCED IN
STEP FOUR, THE PEN MOVES SEQUENTIALLY FROM ONE POINT TO THE NEXT
DEPENDING ON THE ORDER IN WHICH THE POINT WAS PRODUCED. IF \(IC=2\)
(THE PEN IS DOWN), THE PEN WILL DRAW A LINE FROM ONE POINT TO THE
NEXT. IF \(IC=3\) (THE PEN IS UP), IT WILL MOVE FROM ONE POINT TO THE
NEXT WITHOUT DRAWING A LINE.

COMMON \((C(22),XPV1,YPV3)\)

TRANSLATE
\[XO=X-C(1)\]
\[YO=Y-C(2)\]
\[ZC=Z-C(3)\]

ROTATE
\[C(21)=C(8)*XO+C(9)*YO\]
\[C(22)=C(13)*XO+C(14)*YO+C(15)*ZO\]
\[YC=C(10)*XO+C(11)*YO+C(12)*ZO\]

PROJECT
\[YC=Y+8(7)\]
IF \((YO>0.0)\) GO TO 2
WRITE(6,2)

2 FORMAT('VIEWING IMPOSSIBLE')

1 \[YC=C(7)/YO\]
\[C(21)=YO*C(21)\]
\[C(22)=YO*C(22)\]
IF \((C(20).EQ.0.)\) RETURN

SCALE
\[C(21)=(C(21)-C(21))/C(17)*10.\]
\[C(22)=(C(22)-C(18))/C(19)*8.\]

PLOT
CALL PLOT\((C(21),C(22),IC)\)
\[XPV1=C(21)\]
\[YPV3=C(22)\]
RETURN

**********DATA**********DATA**********DATA**********DATA**********DATA**********

<p>| | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.3033</td>
<td>2.12132</td>
<td>5.3033</td>
<td>5.3033</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-5.3033</td>
<td>-2.12132</td>
<td>-5.3033</td>
<td>-5.3033</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>4.63721</td>
<td>3.18198</td>
<td>3.18198</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>-4.63721</td>
<td>-3.18198</td>
<td>-3.18198</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5. | 5. | 10. | -2. | -12. | 0. | 18. | 100 | 7 | 11 | 25 | .1111010.10.10 |
| 0. | 0. | 0. |   |   |   |   |   |   |   |   |   |   |   |   |
| 10. | 0. | 0. |   |   |   |   |   |   |   |   |   |   |   |   |
| 0. | 10. | 0. |   |   |   |   |   |   |   |   |   |   |   |   |
| 0. | 0. | 10. |   |   |   |   |   |   |   |   |   |   |   |   |
| 10. | 10. | 0. |   |   |   |   |   |   |   |   |   |   |   |   |
| 10. | 10. | 10. |   |   |   |   |   |   |   |   |   |   |   |   |
Bibliography
