Reports of the Department of Geodetic Science
Report No. 184

COORDINATE TRANSFORMATION BY MINIMIZING CORRELATIONS BETWEEN PARAMETERS

by
Muneendra Kumar

Prepared for
National Aeronautics and Space Administration
Washington, D.C.

Contract No. NGR 36-008-093
OSURF Project No. 2514

The Ohio State University
Research Foundation
Columbus, Ohio 43212

July, 1972

Reports of the Department of Geodetic Science

Report No. 184

COORDINATE TRANSFORMATION BY MINIMIZING CORRELATIONS BETWEEN PARAMETERS

By

Muneendra Kumar

Prepared for

National Aeronautics and Space Administration
Washington, D. C.

Contract No. NGR 36-008-093
OSURF Project No. 2514

The Ohio State University
Research Foundation
Columbus, Ohio 43212

July, 1972

PREFACE

This project is under the supervision of Ivan I. Mueller, Professor of the Department of Geodetic Science at The Ohio State University, and is under the technical direction of James P. Murphy, Special Programs, Code ES, NASA Headquarters, Washington, D. C. The contract is administered by the Office of University Affairs, NASA, Washington, D. C. 20546

A revised version of this report has been submitted to the Graduate School of The Ohio State University in partial fulfillment of the requirements for the Master of Science degree.

ABSTRACT

The subject of this investigation is to determine the transformation parameters (three rotations, three translations and a scale factor) between two Cartesian coordinate systems from sets of coordinates given in both systems. The objective is the determination of well separated transformation parameters with reduced correlations between each other, a problem especially relevant when the sets of coordinates are not well distributed. The above objective is achieved by preliminarily determining the three rotational parameters and the scale factor from the respective direction cosines and chord distances (these being independent of the translation parameters) between the common points, and then computing all the seven parameters from a solution in which the rotations and the scale factor are entered as weighted constraints according to their variances and covariances obtained in the preliminary solutions.

Numerical tests involving two geodetic reference systems were performed to evaluate the effectiveness of this approach as follows:

(a) A non-constrained solution for general transformation for the seven parameters (including the three translations and scale factor).

(b) A constrained solution for general transformation for the seven parameters utilizing the three rotations with their statistics as constraints.

(c) A constrained solution for general transformation for the seven parameters using the three rotations and scale factor with their statistics as constraints.

The above schemes were then separately repeated for each of the following three cases:

(i) Using the full variance-covariance matrix between coordinates of the geodetic reference systems.

(ii) Using only a (3 X 3) banded diagonal variance-covariance matrix, thus assuming no correlation between coordinates of any two points within the system.

(iii) Using only variances for the coordinates, thereby further omitting the correlation between the three coordinates of any one point in the system.

In the case of seven parameter general transformation, the best estimates were obtained using full variance-covariance matrix and constraining three rotations and the scale factor, case (c) and (iii) above. The improvement in correlation between translations and rotations was more significant compared to between translation and scale factor.

ACKNOWLEDGMENTS

The writer of this report expresses his sincere gratitude to Dr. Ivan I. Mueller for his constant help, cooperation and most valuable guidance in the execution of this study.

Special indebtedness and thanks are owed by the author to Dr. R. H. Rapp, Dr. N. K. Saxena and Mr. J. P. Reilly for the valuable suggestions and comments.

The writer is also thankful to Mrs. Evelyn Rist and Ms. Michelle Neff for all the help and cooperation extended and for the excellent typing of this report.

TABLE OF CONTENTS

LIST OF TABLES

1. INTRODUCTION

During the last twenty-five years with the availability of computer technology and its phenomenal growth in basic hardware and core storage capacity and the exceptional increase in a computer's ability of solving problems in lesser and lesser time, a trend has set in to analyze the problems in geodesy and photogrammetry more and more in three dimensional space rather than to follow traditional concepts.

Further, the advent of artificial satellites and their subsequent use in geodesy made it possible to obtain Cartesian coordinates of points on earth surface.

Several projects involving satellite-networks of continental or global extent were begun and at present they are in varying stages of completion. Many new solutions have recently come out, each delineating its own reference system. These systems in reality should differ from each other only in having different origins, sets of axes or scale.

Thus, the relationship between any two such reference systems (e.g., UVW and XYZ) would generally consist of seven parameters—three translations ($\Delta X, \Delta Y, \Delta Z$) between the two origins, three rotations (ω, ψ, ϵ) of the Euler's angle type between the two sets of axes and the scale factor (Δs), if any (Figure 1).

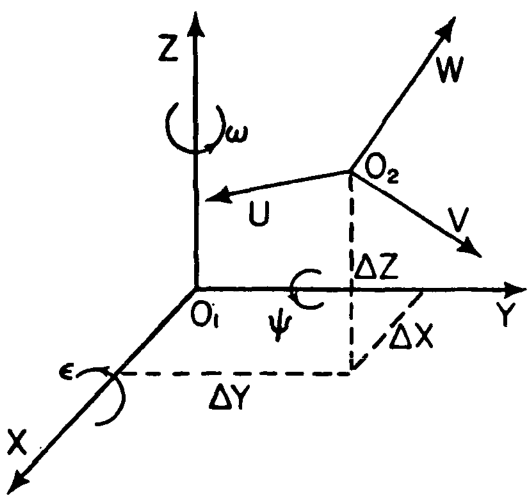

Figure 1.

The mathematical model to be used in the computations of the above seven parameters from a least squares solution may be written in the following form [Badekas, 1969; Bursa, 1965; Wolf, 1963]:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \equiv \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_i - \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix}_i - \begin{bmatrix} 1 & \omega & -\psi \\ -\omega & 1 & \epsilon \\ \psi & -\epsilon & 1 \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix}_i - \Delta s \begin{bmatrix} U \\ V \\ W \end{bmatrix}_i = 0, \tag{1}$$

where "i" denotes any point common to both the systems. The three angles ω, ψ, and ϵ of the Euler type correspond to small rotations about the Z, Y and X axes respectively— the positive direction of rotations taken in counter clockwise mode, when viewed from the end of the repsective axes towards the origin. It may be worth while to mention here that the station coordinates in both the systems (U_i, V_i, W_i and X_i, Y_i, Z_i) are treated as observations in the above model.

The above equation written in matrix notation can then be modified into the observation equation below [Uotila, 1967]:

$$BV + AX + W = 0, \tag{2}$$

where

$$B \equiv \begin{bmatrix} \frac{\partial f_1}{\partial X} & \frac{\partial f_1}{\partial Y} & \frac{\partial f_1}{\partial Z} & \frac{\partial f_1}{\partial U} & \frac{\partial f_1}{\partial V} & \frac{\partial f_1}{\partial W} \\ \frac{\partial f_2}{\partial X} & \frac{\partial f_2}{\partial Y} & \frac{\partial f_2}{\partial Z} & \frac{\partial f_2}{\partial U} & \frac{\partial f_2}{\partial V} & \frac{\partial f_2}{\partial W} \\ \frac{\partial f_3}{\partial X} & \frac{\partial f_3}{\partial Y} & \frac{\partial f_3}{\partial Z} & \frac{\partial f_3}{\partial U} & \frac{\partial f_3}{\partial V} & \frac{\partial f_3}{\partial W} \end{bmatrix}_i$$

$$= \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix},$$

$$A \equiv \begin{bmatrix} \frac{\partial f_1}{\partial \Delta X} & \frac{\partial f_1}{\partial \Delta Y} & \frac{\partial f_1}{\partial \Delta Z} & \frac{\partial f_1}{\partial \Delta S} & \frac{\partial f_1}{\partial \omega} & \frac{\partial f_1}{\partial \psi} & \frac{\partial f_1}{\partial \epsilon} \\ \frac{\partial f_2}{\partial \Delta X} & \frac{\partial f_2}{\partial \Delta Y} & \frac{\partial f_2}{\partial \Delta Z} & \frac{\partial f_2}{\partial \Delta S} & \frac{\partial f_2}{\partial \omega} & \frac{\partial f_2}{\partial \psi} & \frac{\partial f_2}{\partial \epsilon} \\ \frac{\partial f_3}{\partial \Delta X} & \frac{\partial f_3}{\partial \Delta Y} & \frac{\partial f_3}{\partial \Delta Z} & \frac{\partial f_3}{\partial \Delta S} & \frac{\partial f_3}{\partial \omega} & \frac{\partial f_3}{\partial \psi} & \frac{\partial f_3}{\partial \epsilon} \end{bmatrix}_i$$

$$
= \begin{bmatrix} -1 & 0 & 0 & -U & -V & W & 0 \\ 0 & -1 & 0 & -V & U & 0 & -W \\ 0 & 0 & -1 & -W & 0 & -U & V \end{bmatrix}_i ,
$$

$$
W = \begin{bmatrix} X - U \\ Y - V \\ Z - W \end{bmatrix}_i ,
$$

while V and X represent the residuals to the observations and corrections to the parameter estimates, respectively. Hence, collecting all the matrices as above, pointwise in the systems, the observation equation becomes:

$$
\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} V_X \\ V_Y \\ V_Z \\ V_U \\ V_V \\ V_W \end{bmatrix}_i + \begin{bmatrix} -1 & 0 & 0 & -U & -V & W & 0 \\ 0 & -1 & 0 & -V & U & 0 & -W \\ 0 & 0 & -1 & -W & 0 & -U & V \end{bmatrix}_i \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ \Delta s \\ \omega \\ \psi \\ \epsilon \end{bmatrix} + \begin{bmatrix} X - U \\ Y - V \\ Z - W \end{bmatrix}_i = 0 \qquad (3)
$$

Defining the geodetic reference systems on the assumption that the Laplace-condition has been enforced throughout the network (which implies that the axes of the reference ellipsoid are parallel to the conventional earth-fixed axes), many experiments have been made in recent times to determine the seven transformation parameters in relating the different geodetic systems to each other using an observation equation of type (3) [Lambeck, 1971; Marsh et. al., 1971].

However, in the above general transformation, if the geodetic reference systems are properly oriented through the Laplace-condition, the three rotations arising due to the improper relative orientation of the systems are generally never more than a few seconds of arc, while translations may amount up to 200 to 300 meters. Also, due to the presence of high correlations between the rotations, the scale factor and the translations, satisfactory independent estimates for these parameters are difficult to obtain from a combined general solution using equation (3).

This investigation separates the determinations of the rotations and the scale factor (from that of the translations) for subsequent use as constraints in a combined general solution.

2. THE INDEPENDENT DETERMINATIONS OF ROTATIONAL AND SCALAR PARAMETERS

2.1 Determination of Rotations

2.1.1 Mathematical Model

The mathematical model used in this study is as follows [Bursa, 1966]:

$$\begin{aligned} T_{ik}^{(1)} - T_{ik}^{(2)} + \omega + \psi \sin T_{ik}^{(1)} \tan \delta_{ik}^{(1)} - \epsilon \cos T_{ik}^{(1)} \tan \delta_{ik}^{(1)} &= 0 \\ \delta_{ik}^{(1)} - \delta_{ik}^{(2)} \qquad + \psi \cos T_{ik}^{(1)} \qquad + \epsilon \sin T_{ik}^{(1)} &= 0 \end{aligned} \tag{4}$$

where T_{ik} and δ_{ik} are defined as the geodetic hour angle and declination of the $(i-k)^{th}$ direction of the observed point at k^{th} station and the observer at i^{th} station. The indexes (1) and (2) denote the two systems with the transformation proceeding from system #1 to system #2.

If A_{ik}, B_{ik}, C_{ik} are taken to denote the direction cosines of the $(i-k)^{th}$ line of length R_{ik}, then for the first (UVW) system one gets:

$$\left.\begin{aligned} A_{ik} &= \frac{U_k - U_i}{R_{ik}} = \frac{\Delta U_{ik}}{R_{ik}} , \\ B_{ik} &= \frac{V_k - V_i}{R_{ik}} = \frac{\Delta V_{ik}}{R_{ik}} , \\ C_{ik} &= \frac{W_k - W_i}{R_{ik}} = \frac{\Delta W_{ik}}{R_{ik}} , \end{aligned}\right\} \qquad (5)$$

and

$$\left.\begin{aligned} T_{ik} &= -\arctan \frac{B_{ik}}{A_{ik}} , \\ \delta_{ik} &= \arctan \frac{C_{ik}}{(A_{ik}^2 + B_{ik}^2)^{\frac{1}{2}}} . \end{aligned}\right\} \qquad (6)$$

In the above relations (4) through (6) the elements of translation do not enter the picture. A similar set of relations as per (5) and (6) can be established for the second (XYZ) system.

2.1.2 Observation Equations

The mathematical model (4) then, for each $(i-k)^{th}$ line, yields the following generalized form of observation equations [Uotila, 1967]:

$$\begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_T \\ v_\delta \end{bmatrix}_{ik} + \begin{bmatrix} 1 & \sin T_{ik}^{(1)} \tan \delta_{ik}^{(1)} & -\cos T_{ik}^{(1)} \tan \delta_{ik}^{(1)} \\ 0 & \cos T_{ik}^{(1)} & \sin T_{ik}^{(1)} \end{bmatrix}_{ik} \begin{bmatrix} \omega \\ \psi \\ \epsilon \end{bmatrix}$$

$$+ \begin{bmatrix} \left(T_{ik}^{(1)} - T_{ik}^{(2)}\right) \\ \left(\delta_{ik}^{(1)} - \delta_{ik}^{(2)}\right) \end{bmatrix}_{ik} = 0 \qquad (7)$$

Using the conventional weight matrix P for the coordinates of points included in the transformation (see section 2.1.3), and the principle of least squares by making $V'PV$ as minimum, the equation (7) is then solved for correction vector (ω, ψ, ϵ) and for the variance-covariance matrix $(\Sigma\omega\psi\epsilon)$ of the three parameters.

2.1.3 Weights

Using the variance-covariance matrices ΣX and ΣU in respect of i^{th} and k^{th} points for the XYZ and UVW systems, the variance-covariance matrices $\Sigma_{T\delta}$ for the two systems of coordinates can be computed through propagation of errors [Uotila, 1967].

Two distinct cases would arise here. Firstly, when in addition to correlation between X, Y, Z-coordinates of any point, the correlation between the coordinates of one point to others is also considered. In such a case, the necessary relation will be

$$\left[\Sigma_{T\delta}^{(1)}\right]_{2,2} = G \begin{bmatrix} \Sigma U_i & \Sigma U_{ik} \\ \Sigma U_{ik} & \Sigma U_k \end{bmatrix} G' \tag{8}$$

where

$$G = \begin{bmatrix} \frac{\partial T_{ik}^{(1)}}{\partial U_i} & \frac{\partial T_{ik}^{(1)}}{\partial V_i} & \frac{\partial T_{ik}^{(1)}}{\partial W_i} & \frac{\partial T_{ik}^{(1)}}{\partial U_k} & \frac{\partial T_{ik}^{(1)}}{\partial V_k} & \frac{\partial T_{ik}^{(1)}}{\partial W_k} \\ \frac{\partial \delta_{ik}^{(1)}}{\partial U_i} & \frac{\partial \delta_{ik}^{(1)}}{\partial V_i} & \frac{\partial \delta_{ik}^{(1)}}{\partial W_i} & \frac{\partial \delta_{ik}^{(1)}}{\partial U_k} & \frac{\partial \delta_{ik}^{(1)}}{\partial V_k} & \frac{\partial \delta_{ik}^{(1)}}{\partial W_k} \end{bmatrix},$$

and

$$\frac{\partial T_{ik}}{\partial U_i} = -\frac{\partial T_{ik}}{\partial U_k} = -\frac{\Delta V_{ik}}{\Delta U_{ik}^2 + \Delta V_{ik}^2},$$

$$\frac{\partial T_{ik}}{\partial V_i} = -\frac{\partial T_{ik}}{\partial V_k} = -\frac{\Delta U_{ik}}{\Delta U_{ik}^2 + \Delta V_{ik}^2},$$

$$\frac{\partial T_{ik}}{\partial W_i} = -\frac{\partial T_{ik}}{\partial W_k} = 0,$$

$$\frac{\partial \delta_{ik}}{\partial U_i} = -\frac{\partial \delta_{ik}}{\partial U_k} = \frac{\Delta U_{ik}\,\Delta W_{ik}}{R_{ik}^{2(1)}\sqrt{\Delta U_{ik}^2 + \Delta V_{ik}^2}},$$

$$\frac{\partial \delta_{ik}}{\partial V_i} = -\frac{\partial \delta_{ik}}{\partial V_k} = \frac{\Delta V_{ik}\,\Delta W_{ik}}{R_{ik}^{2(1)}\sqrt{\Delta U_{ik}^2 + \Delta V_{ik}^2}},$$

$$\frac{\partial \delta_{ik}}{\partial W_i} = -\frac{\partial \delta_{ik}}{\partial W_k} = -\frac{\sqrt{\Delta U_{ik}^2 + \Delta V_{ik}^2}}{R_{ik}^{2(1)}} ,$$

$$R_{ik}^{2(1)} = \Delta U_{ik}^2 + \Delta V_{ik}^2 + \Delta W_{ik}^2 .$$

Secondly, ignoring the correlations between the coordinates of different points within a system, equation (8) can be modified as under:

$$\left[\Sigma_{T\delta}^{(1)}\right]_{2,2} = G \begin{bmatrix} \Sigma U_i & 0 \\ 0 & \Sigma U_k \end{bmatrix} G' \tag{9}$$

In the equations (8) and (9), ΣU_i and ΣU_k correspond to i^{th} and k^{th} point of the first system and can be either full (3 × 3) matrices with covariances between the three coordinates of a point, or may contain variances for U, V and W in a diagonal form only. However, in the case of covariances (ΣU_{ik}) between the points being included, the matrix in equation (8) would be a full (6 × 6).

Obtaining similarly $\Sigma_{T\delta}^{(2)}$, the combined variance-covariance matrix, to be used with equation (7), is given by:

$$P_{4,4} = \left[\begin{array}{c|c} \Sigma_{T\delta}^{(2)} & 0 \\ \hline 0 & \Sigma_{T\delta}^{(1)} \end{array}\right] \tag{10}$$

It may be noted here that the matrix P is always in 2 × 2 banded diagonal form.

2.2 Determination of Scale Factor

2.2.1 Mathematical Model

The scale factor between the systems #1 and #2 would be given as follows:

$$\Delta s_{ik} = \frac{R_{ik}^{(2)}}{R_{ik}^{(1)}} - 1 \tag{11}$$

where
$$R_{ik}^{(2)} = (\Delta X_{ik}^2 + \Delta R_{ik}^2 + \Delta Z_{ik}^2)^{\frac{1}{2}}$$

$$R_{ik}^{(1)} = (\Delta U_{ik}^2 + \Delta V_{ik}^2 + \Delta W_{ik}^2)^{\frac{1}{2}} \quad .$$

2.2.2. Weights

Using the variance-covariances matrices ΣX and ΣU for the coordinates of i_t^{th} and k^{th} points in the two systems included in the transformation (section 2.1.3), a variance $\sigma_{\Delta s}^2$ is established for the scale factor through error propagation. Two cases similar to equations (8) and (9) would arise according to the case when full variance-covariance matrix between different points within the system is considered or not.

The matrix G for the scale factor determination is

$$G = \left[\frac{\partial \Delta s}{\partial U_i} \; \frac{\partial \Delta s}{\partial V_i} \; \frac{\partial \Delta s}{\partial W_i} \; \frac{\partial \Delta s}{\partial U_k} \; \frac{\partial \Delta s}{\partial V_k} \; \frac{\partial \Delta s}{\partial W_k} \; \frac{\partial \Delta s}{\partial X_i} \; \frac{\partial \Delta s}{\partial Y_i} \; \frac{\partial \Delta s}{\partial Z_i} \; \frac{\partial \Delta s}{\partial X_k} \; \frac{\partial \Delta s}{\partial Y_k} \; \frac{\partial \Delta s}{\partial Z_k}\right] ,$$

where
$$\frac{\partial \Delta s}{\partial U_i} = -\frac{\partial \Delta s}{\partial U_k} = \frac{\Delta U_{ik} \cdot R_{ik}^{(2)}}{[R_{ik}^{(1)}]^{3/2}} \quad ,$$

$$\frac{\partial \Delta s}{\partial V_i} = -\frac{\partial \Delta s}{\partial V_k} = \frac{\Delta V_{ik} \cdot R_{ik}^{(2)}}{[R_{ik}^{(1)}]^{3/2}} \quad ,$$

$$\frac{\partial \Delta s}{\partial W_i} = -\frac{\partial \Delta s}{\partial W_k} = \frac{\Delta W_{ik} \cdot R_{ik}^{(2)}}{[R_{ik}^{(1)}]^{3/2}} \quad ,$$

$$\frac{\partial \Delta s}{\partial X_i} = -\frac{\partial \Delta s}{\partial X_k} = -\frac{\Delta X_{ik}}{R_{ik}^{(1)} \cdot R_{ik}^{(2)}} \quad ,$$

$$\frac{\partial \Delta s}{\partial Y_i} = -\frac{\partial \Delta s}{\partial Y_k} = -\frac{\Delta Y_{ik}}{R_{ik}^{(1)} \cdot R_{ik}^{(2)}} \quad ,$$

$$\frac{\partial \Delta s}{\partial Z_i} = -\frac{\partial \Delta s}{\partial Z_k} = -\frac{\Delta Z_{ik}}{R_{ik}^{(1)} \cdot R_{ik}^{(2)}} \quad .$$

Hence,

$$\sigma^2_{\Delta s_{ik}} = G \left[\begin{array}{cc|cc} \Sigma U_i & \Sigma U_{ik} & & 0 \\ \Sigma U_{ik} & \Sigma U_k & & \\ \hline & & \Sigma X_i & \Sigma X_{ik} \\ & 0 & \Sigma X_{ik} & \Sigma X_k \end{array} \right]_{12} G' \qquad (12)$$

,

where the full (12 × 12) matrix would become a (3 × 3) banded diagonal matrix in case ΣU_{ik} and ΣX_{ik} are zero, i.e., covariances are not considered. The complete (12 × 12) matrix would assume a diagonal pattern when only variances are used for station coordinates.

Using the value of Δs_{ik} and $\sigma^2_{\Delta s_{ik}}$ from equations (11) and (12), the value for weighted mean and its variance for the transformation under investigation is established as given below [Hirvonen, 1971]:

$$\Delta s_m = \frac{[w_{ik} \cdot \Delta s_{ik}]}{[w_{ik}]} \qquad (13)$$

$$\sigma^2_{\Delta s_m} = \frac{[w_{ik} \cdot (\Delta s_{ik} - \Delta s_m)^2]}{[w_{ik}](n-1)} \qquad (14)$$

where

w_{ik} = $1/\sigma_{\Delta s_{ik}}$ and $[w_{ik}]$ denotes the sum of all such weights.

n = Total number of scale factor values used in the sample.

3. BRIEF DISCUSSION ON THE FORTRAN PROGRAM

Appendix I gives the complete computer program for obtaining the constrained or non-constrained solution for seven parameters. With appropriate coding non-constrained solutions for three parameters (ΔX, ΔY and ΔZ) and scale factor Δs can also be obtained.

The input coordinates can either be Cartesian or geodetic (ellipsoidal) with 35 as the maximum number of points in each system. However, the matrices can easily be re-dimensioned to accomodate more points when required. The program is self-explanatory with regard to definition of various option codes for input, type of solution and inclusion of correlation data, etc.

The broad basic divisions of the program are as under:

(a) Main Program: This section takes as input the various options in input/solutions, coordinates of points, rectangular or ellipsoidal, and semimajor axis and flattening of the ellipsoid used, if required. It then prints out the two sets of coordinates used for checking purposes.

The various options of input/solutions have been designated in the program as KCODE e.g., KCODE (1) refers to number of common points involved in the transformation. A complete list with necessary explanatory remarks has been included in the beginning of the program.

(b) Subroutine "EULERS": This subroutine first reads the variance-covariance matrices of the station coordinates, with or without correlation, and then sets up matrices A, W and P to be used for the solutions of three rotations through direction cosines (equation (7)).

The subroutine writes up the variance-covariance matrices for the coordinates on the disk and stores the estimates for ω, ψ and ϵ, and their variance-covariance matrix $[\Sigma\omega\psi\epsilon]$ in the common block for subsequent use.

(c) Subroutine "SCALE": This subroutine computes the weighted mean value for scale factor Δs and its variance by direct chord comparison independent of other transformation parameters (equations (13) and (14)).

(d) Subroutine "TFORM": This subroutine solves for a general transformation (equation (3)), utilizing the common block core memory for coordinates of points and variance-covariance matrices from the disk.

The matrix M^{-1} to be utilized for generating normal equations is computed by calling another subroutine "SETUP".

NOTE: In case the solution is required ONLY for three translation or three translations and scale factor, KCODE (3) is coded as "0" and then subroutine "EULERS" is skipped by the program.

(e) Subroutine "CSTRNT": This subroutine uses the results of subroutines SCALE and EULERS as constraints with their appropriate statistics and computes for a constrained solution of seven parameters. The results are returned to subroutine TFORM for printout. KCODE (11) refers to the option whether 3 or 4 parameters are to be constrained.

(f) Subroutine "RESIDU": This subroutine computes the residuals vector V for observations i.e., the station coordinates used in the program. The residuals are printed station wise for both systems #1 and #2.

In the computer program, the storage mode used for major computation is in vector form for increased flexibility and saving of core storage.

Appendix II gives a typical set of Job Control Cards (JCL).

4. NUMERICAL EXAMPLE

The above transformation models were used to study the relationship between the transformation parameters and obtaining their best estimates by minimizing correlation for the following two reference systems:

(i) System MPS-7, [Mueller and Whiting, 1972].

(ii) System NA-9, [Mueller et. al., 1972].

Using the same set of thirty common stations of the above two systems, the following solutions were obtained during the investigation:

Serial Number	Type of Variance-Covariance Matrix Used	7-Parameter General Transformation		
		Unconstrained Solution	Constrained Solution @	
			Constraints: 3 Rotation	Constraints: 3 Rotations and Scale Factor
		(a)	(b)	(c)
(i)	Only Variances	√	√	√
(ii)	(3 X 3) Banded Diagonal Variance-Covariance Matrix	√	√	√
(iii)	Full Variance-Covariance Matrix	√	√	√

@Note: The constraints for these solutions (rotations and/or scale factor) with their statistics were computed independently of the translation parameters (subroutine EULERS and SCALE of the Fortran IV program).

Two solutions in full have been appended in the report as specimens in Tables 1 and 2 as under:

Table 1: Sample printout of the solution for three rotations (ω, ψ, ϵ) and scale factor (Δs), using full variance-covariance matrix.

Table 2: Sample printout of the constrained seven parameter general solution between NA-9 and MPS-7 with three rotations and

TABLE 1

Sample Printout of the solutions for three rotations as parameters and the scale factor, using full variance-covariance matrix.

TABLE 1

SOLUTION FOR '3' ROTATION PARAMETERS

(FROM DIRECTION COSINES -- UNITS SECONDS OF ARC)

(USING FULL VARIANCE-COVARIANCE MATRIX)

OMEGA	PSI	EPSILON
0.1693791D+00	-0.3520145D-01	-0.2173630D+00

VARIANCE - COVARIANCE MATRIX

MO2= 1.36

0.16753861D-02	0.40623287D-03	-0.93767764D-03
0.40623287D-03	0.12317991D-02	-0.48803740D-03
-0.93767764D-03	-0.48803740D-03	0.27191935D-02

COEFFICIENT OF CORRELATION

0.10000000D+01	0.28277933D+00	-0.43931501D+00
0.28277933D+00	0.10000000D+01	-0.26666321D+00
-0.43931501D+00	-0.26666321D+00	0.10000000D+01

SOLUTION FOR SCALE FACTOR

(FROM CHORD COMPARISON)

SCALE FACTOR (10.D+5)	VARIANCE (10.D+11)
5.16	0.06

TABLE 2

Sample printout of the constrained seven parameters general solution, using full variance-covariance matrix (case (c)/(iii)).

TABLE 2

SCALE FACTOR AND ROTATION PARAMETERS CONSTRAINED

SOLUTION FOR 3 TRANSLATION, 1 SCALE AND 3 ROTATION PARAMETERS

(USING FULL VARIANCE-COVARIANCE MATRIX)

DX METERS	DY METERS	DZ METERS	DL (10.D+5)	OMEGA SECONDS	PSI SECONDS	EPSILON SECONDS
-45.38	171.94	187.44	5.14	0.17	-0.04	-0.22

VARIANCE - COVARIANCE MATRIX

MO2= 0.84

0.176D+01	0.250D+00	0.453D+00	-0.310D-07	0.126D-06	0.778D-07	-0.852D-07
0.250D+00	0.228D+01	-0.322D-01	0.243D-06	0.551D-07	0.238D-07	-0.124D-06
0.453D+00	-0.322D-01	0.206D+01	-0.149D-06	0.615D-07	0.222D-07	-0.177D-06
-0.310D-07	0.243D-06	-0.149D-06	0.441D-13	-0.325D-17	-0.298D-16	-0.127D-16
0.126D-06	0.551D-07	0.615D-07	-0.325D-17	0.225D-13	0.525D-14	-0.125D-13
0.778D-07	0.238D-07	0.222D-07	-0.298D-16	0.525D-14	0.167D-13	-0.654D-14
-0.852D-07	-0.124D-06	-0.177D-06	-0.127D-16	-0.125D-13	-0.654D-14	0.364D-13

COEFFICIENTS OF CORRELATION

0.100D+01	0.125D+00	0.238D+00	-0.111D+00	0.635D+00	0.454D+00	-0.337D+00
0.125D+00	0.100D+01	-0.149D-01	0.765D+00	0.244D+00	0.122D+00	-0.429D+00
0.238D+00	-0.149D-01	0.100D+01	-0.493D+00	0.286D+00	0.120D+00	-0.648D+00
-0.111D+00	0.765D+00	-0.493D+00	0.100D+01	-0.103D-03	-0.110D-02	-0.317D-03
0.635D+00	0.244D+00	0.286D+00	-0.103D-03	0.100D+01	0.271D+00	-0.436D+00
0.454D+00	0.122D+00	0.120D+00	-0.110D-02	0.271D+00	0.100D+01	-0.265D+00
-0.337D+00	-0.429D+00	-0.648D+00	-0.317D-03	-0.436D+00	-0.265D+00	0.100D+01

scale factor as constraints, using full variance-covariance matrix (case (c)/(iii)).

A summary of the results for cases (a) through (c) and (i) through (iii) are presented in the following tables:

TABLE 3 gives the results for three rotations, as obtained independently of translations and scale factor from direction cosines, for cases (i) through (iii).

TABLE 4 gives the results for the scale factor, as obtained by direct chord comparisons independent of other transformation parameters, for cases (i) through (iii).

TABLE 5 gives the results for the constrained and non-constrained seven parameters general transformation solutions (cases (a) through (c) and (i) through (iii)).

TABLE 6 gives the comparative study of the results for seven parameters general transformation solutions as regards correlation between translations and rotations/scale factor, using different variance-covariance matrices (cases (i) through (iii)).

TABLE 7 gives the comparative study of the results for seven parameters general transformation solutions as regards correlation between translations and rotations/scale factor, using different constraints (cases (a) through (c)).

TABLE 3

Three Rotation Parameters from Direction Cosines

NA-9~MPS-7

	Using Variances Only	Using (3×3) Banded Diagonal Variance-Covariance Matrix	Using full Variance-Covariance Matrix
Case	(i)	(ii)	(iii)
$\omega\,('')$	0.17 ± 0.05	0.17 ± 0.04	0.17 ± 0.04
$\psi\,('')$	0.04 ± 0.04	−0.02 ± 0.04	−0.04 ± 0.04
$\epsilon\,('')$	−0.20 ± 0.06	−0.24 ± 0.05	−0.22 ± 0.05
σ_0^2	1.15	1.30	1.36

TABLE 4

Scale Factor From Chord Comparison

NA-9~MPS-7

	Using Variances Only	Using (3×3) Banded Diagonal Variance-Covariance Matrix	Using full Variance-Covariance Matrix
Case	(i)	(ii)	(iii)
$\Delta s(\times 10^6)$	5.46 ± 0.24	5.37 ± 0.24	5.18 ± 0.24

TABLE 5

Seven Parameters General Transformation Solutions

NA-9~MPS-7

	Non-Constrained Solutions			Constrained Solutions					
				Constraints: 3 Rotations			Constraints: 3 Rotations & Scale Factor		
	Using Variances Only	Using (3×3) Banded Diagonal Variance-Covariance Matrix	Using Full Variance-Covariance Matrix	Using Variances Only	Using (3×3) Banded Diagonal Variance-Covariance Matrix	Using Full Variance-Covariance Matrix	Using Variances Only	Using (3×3) Banded Diagonal Variance-Covariance Matrix	Using Full Variance-Covariance Matrix
Case	(a)/(i)	(a)/(ii)	(a)/(iii)	(b)/(i)	(b)/(ii)	(b)/(iii)	(c)/(i)	(c)/(ii)	(c)/(iii)
ΔX (m)	-44.5 ± 5.2	-44.9 ± 3.6	-44.9 ± 3.6	-44.0 ± 1.7	-44.8 ± 1.4	-45.0 ± 1.4	-44.4 ± 1.7	-45.2 ± 1.4	-45.4 ± 1.3
ΔY (m)	171.5 ± 5.1	170.3 ± 4.7	170.3 ± 4.7	170.1 ± 3.8	169.6 ± 4.0	169.4 ± 4.0	173.0 ± 1.7	173.2 ± 1.5	171.9 ± 1.5
ΔZ (m)	190.4 ± 5.5	190.4 ± 4.3	190.4 ± 4.3	188.1 ± 2.8	189.4 ± 2.7	189.0 ± 2.7	186.3 ± 1.8	187.2 ± 1.5	187.4 ± 1.4
ω ($''$)	0.15 ± 0.16	0.17 ± 0.12	0.17 ± 0.12	0.16 ± 0.04	0.17 ± 0.03	0.17 ± 0.03	0.16 ± 0.04	0.17 ± 0.03	0.17 ± 0.03
ψ ($''$)	0.04 ± 0.14	-0.03 ± 0.11	-0.03 ± 0.11	0.04 ± 0.04	-0.02 ± 0.03	-0.04 ± 0.03	0.04 ± 0.04	-0.02 ± 0.03	-0.04 ± 0.03
ϵ ($''$)	-0.30 ± 0.20	-0.28 ± 0.15	-0.28 ± 0.15	-0.21 ± 0.05	-0.24 ± 0.04	-0.22 ± 0.04	-0.21 ± 0.05	-0.24 ± 0.04	-0.22 ± 0.04
$\Delta s (\times 10^6)$	4.9 ± 0.7	4.7 ± 0.7	4.7 ± 0.7	4.9 ± 0.7	4.7 ± 0.7	4.7 ± 0.7	5.4 ± 0.23	5.3 ± 0.2	5.1 ± 0.2
σ_0^2	0.95	0.83	0.83	0.91	0.79	0.79	0.97	0.85	0.84

TABLE 6

Comparative Study of Correlation Coefficients

Between Transformation Parameters

(Using Different Variance-Covariance Matrices)

Case (i): USING VARIANCES ONLY

	Non-Constrained Solution			Constrained Solutions					
				3 Rotations			3 Rotations and Scale Factor		
Case	(a)			(b)			(c)		
Translations / Rotations and Scale Factor	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ
ω	0.88	0.40	0.43	0.68	0.14	0.22	0.71	0.32	0.35
ψ	0.63	0.19	0.13	0.49	0.07	0.08	0.51	0.14	0.13
ϵ	-0.47	-0.67	-0.88	-0.38	-0.23	-0.45	-0.40	-0.51	-0.73
Δs	-0.10	0.74	-0.40	-0.29	0.95	-0.83	-0.10	0.72	-0.44

Case (ii): USING (3 × 3) BANDED DIAGONAL

VARIANCE-COVARIANCE MATRIX

	Non-Constrained Solution			Constrained Solutions					
				3 Rotations			3 Rotations and Scale Factor		
Case	(a)			(b)			(c)		
Translations / Rotations and Scale Factor	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ
ω	0.83	0.27	0.33	0.58	0.09	0.14	0.62	0.24	0.27
ψ	0.54	0.11	0.13	0.38	0.04	0.08	0.40	0.12	0.13
ϵ	-0.45	-0.51	0.80	-0.32	-0.16	-0.34	-0.34	-0.44	-0.66
Δs	-0.15	0.84	-0.56	-0.36	0.97	-0.89	-0.11	0.76	-0.49

TABLE 6 (Continued)

Case (iii): USING FULL VARIANCE-COVARIANCE MATRIX

	Non-Constrained Solution			Constrained Solutions					
				3 Rotations			3 Rotations and Scale Factor		
Case	(a)			(b)			(c)		
Translations / Rotations and Scale Factor	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ
ω	0.83	0.27	0.33	0.60	0.09	0.15	0.64	0.24	0.29
ψ	0.54	0.11	0.13	0.43	0.04	0.07	0.45	0.12	0.12
ϵ	-0.45	-0.51	-0.80	-0.32	-0.16	-0.34	-0.34	-0.43	-0.65
Δs	-0.15	0.84	-0.56	-0.36	0.97	-0.89	-0.11	0.76	-0.49

TABLE 7

Comparative Study of Correlation Coefficients

Between Transformation Parameters

(Using Different Constraints)

Case (a): NON-CONSTRAINED SOLUTION

	Using Variances Only			Using (3X3) Banded Diagonal Variance-Covariance Matrix			Using Full Variance-Covariance Matrix		
Case	(i)			(ii)			(iii)		
Translations / Rotations and Scale Factor	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ
ω	0.88	0.40	0.43	0.83	0.27	0.33	0.83	0.27	0.33
ψ	0.63	0.19	0.13	0.54	0.11	0.13	0.54	0.11	0.13
ϵ	-0.47	-0.67	-0.88	-0.45	-0.51	0.80	-0.45	-0.51	0.80
Δs	-0.10	0.74	-0.40	-0.15	0.84	-0.56	-0.15	0.84	-0.56

Case (b): CONSTRAINED SOLUTIONS

(CONSTRAINTS: 3 ROTATIONS)

	Using Variances Only			Using (3X3) Banded Diagonal Variance-Covariance Matrix			Using Full Variance-Covariance Matrix		
Case	(i)			(ii)			(iii)		
Translations / Rotations and Scale Factor	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ
ω	0.68	0.14	0.22	0.58	0.09	0.14	0.60	0.09	0.15
ψ	0.49	0.07	0.08	0.38	0.04	0.08	0.43	0.04	0.07
ϵ	-0.38	-0.23	-0.45	-0.32	-0.16	-0.34	-0.32	-0.16	-0.34
Δs	-0.29	0.95	-0.83	-0.36	0.97	-0.89	-0.36	0.97	-0.89

TABLE 7 (Continued)

Case (c): CONSTRAINED SOLUTIONS

(CONSTRAINTS: 3 ROTATIONS AND SCALE FACTOR)

	Using Variances Only			Using (3×3) Banded Diagonal Variance-Covariance Matrix			Using Full Variance-Covariance Matrix		
Case	(i)			(ii)			(iii)		
Translations / Rotations and Scale Factor	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ
ω	0.71	0.32	0.35	0.62	0.24	0.27	0.64	0.24	0.29
ψ	0.51	0.14	0.13	0.40	0.12	0.13	0.45	0.12	0.12
ϵ	-0.40	-0.51	-0.73	-0.34	-0.44	-0.66	-0.34	-0.43	-0.65
Δs	-0.10	0.72	-0.44	-0.11	0.76	-0.49	-0.11	0.76	-0.49

5. CONCLUSIONS

The comparison between different columns of Table 3 shows that the estimates for three rotation parameters remain more or less the same, but that their standard deviations show some improvement as we proceed from column 1 (variances only) to column 3 (full variance-covariance matrix). However, in the case of scale factor (Table 4) the estimates for Δs indicate a definite trend while standard deviation remains constant.

In the case of seven parameters general transformation (Table 5) the comparisons among different columns indicate a definite overall improvement in all parameter estimates. The best estimates were obtained in the solution using full variance-covariance matrix and three rotations (ω, ψ, ϵ) and scale factor (Δs) as constraints (column 10). In this case the standard deviations for all the parameters are smaller (or at the most, equal) compared to those in any other column of Table 5.

Further, it is also noticeable that the improvement from a non-constrained solution to a constrained solution, both with three or four constraints, is more significant compared to the improvement from a constrained solution using variances only to a constrained solution using (3 $\times$ 3) banded diagonal or full variance-covariance matrix. The improvement from the solution using (3 $\times$ 3) banded diagonal to the solution using full variance-covariance matrix is, however, marginal.

A study of Table 6 indicates in all the three cases an overall improvement in correlation from a non-constrained to a constrained solution with four constraints (three rotations and one scale factor). The improvement in correlation between translations and rotations is quite significant while the same in not reflected between translations and scale factor. However, the improvement pattern from Table 7 is not straightforward. The correlations between translations and rotations show a downward trend from the solutions using variances only to the solutions using full variance-covariance matrix in all the three cases while the correlations between translations and Δs show an upward trend.

REFERENCES

Badekas, John (1969). "Investigations Related to the Establishment of a World Geodetic System," Reports of the Department of Geodetic Science, No. 124, The Ohio State University, Columbus.

Bursa, M. (1966). "Fundamentals of the Theory of Geometric Satellite Geodesy," Travaux de L'Institut Geophysique de L'Academie Teheco-Slovaque des Sciences, No. 241.

Hirvonen, R. A. (1971). "Adjustment by Least Squares in Geodesy and Photogrammetry," Frederick Ungar Publishing Co.

Lambeck, K. (1971). "The Relation of Some Geodetic Datums to a Global Geocentric Reference System," Bulletin Géodésique, No. 99, March, 1971.

Marsh, J. G., B. C. Douglas and S. M. Klosko (1971). "A Unified Set of Tracking Stations Coordinates Derived from Geodetic Satellite Tracking Data," Report No. X-553-71320, Goddard Space Flight Center, Greenbelt, Maryland.

Mueller, Ivan I., James P. Reilly and Tomas Soler (1972). "Geodetic Satellite Observation in North America (Solution NA-9)," Reports of the Department of Geodetic Science, No. 187, The Ohio State University, Columbus.

Mueller, Ivan I. and Marvin C. Whiting (1972). "Free Adjustment of a Geometric Global Satellite Network (Solution MPS-7)," Reports of the Department of Geodetic Science, No. 188, The Ohio State University, Columbus.

Uotila, Urho A. (1967). "Introduction to Adjustment Computation with Matrices," Department of Geodetic Science, The Ohio State University, Columbus.

Wolf, H. (1963). "Geometric Connection and Re-orientation of Three-dimensional Triangulation Nets," Bulletin Géodésique, No. 68, June.

Page Intentionally Left Blank

APPENDIX I

Fortran IV Program with Subroutines

```
C ********                                                            ******** *
C ********                                                            ******** *
C ********                                                            ******** *
C ********<<<<<<<<<*******************************************<<<<<<<<<******** *
C ********                                                            ******** *
C ********                 TRANSFORMATION  OF  AXES                   ******** *
C ********                                                            ******** *
C ********<<<<<<<<<*******************************************<<<<<<<<<******** *
C ********<<<<<<<<<*******************************************<<<<<<<<<******** *
C ********                                                            ******** *
C ********                                                            ******** *
C ********                                                                **** *
C ********   PROGRAM WORKS ON TWO SETS OF COORDINATES  --  EITHER         **** *
C ********   SET CAN BE INPUT AS ELLIPSOIDAL COORDINATES, BOTH IN         **** *
C ********   DEGREES AND METERS OR IN GEOS FORMAT.  IN SUCH CASE          **** *
C ********   SEMI-MAJOR AXIS 'A' AND ECCENTRICITY 'E' ARE NEEDED.         **** *
C ********                                                                **** *
C ********   UVW MATRIX TAKES COORDINATES IN THE FIRST SYSTEM             **** *
C ********   ---( IN FORMAT  I5,3F15.5  )                                 **** *
C ********                                                                **** *
C ********   XYZ MATRIX TAKES COORDINATES IN THE SECOND SYSTEM            **** *
C ********   ---( IN FORMAT  I4,5X,OF16.5  )                              **** *
C ********                                                                **** *
C ********   MAXIMUM NUMBER OF INPUT POINTS FOR EACH SYSTEM ---35         **** *
C ********                                                                **** *
C ********                                                            ******** *
C ********<<<<<<<<<*******************************************<<<<<<<<<******** *
C ********<<<<<<<<<*******************************************<<<<<<<<<******** *
C ********<<<<<<<<<*******************************************<<<<<<<<<******** *
C ********                                                            ******** *
C ********                                                            ******** *
C ********                 SUBROUTINE -- 'CSTRNT'                         **** *
C ********                                                                **** *
C ********                                                                **** *
C ********   SOLVES FOR TRANSFORMATION CASE WHEN CONSTRAINTS ARE          **** *
C ********   TO BE APPLIED FOR THREE ROTATIONS.  NECESSARY COUNTER        **** *
C ********   -- KCODE(11) -- IS TO BE CODED AS  ' 4 ' .                   **** *
C ********                                                                **** *
C ********                                                                **** *
C ********   TWO SOLUTIONS ARE OBTAINED WITH THE SAME DATA  ---           **** *
C ********   FIRST WITHOUT CONSTRAINTS AND SECOND WITH CONSTRAINTS.       **** *
C ********                                                                **** *
C ********   INPUT CONSTRAINTS ARE OBTAINED FROM SUBROUTINE 'EULERS'      **** *
C ********    AND SUBROUTINE 'SCALE'.                                     **** *
C ********                                                            ******** *
C ********                                                            ******** *
C ********<<<<<<<<<*******************************************<<<<<<<<<******** *
C ********<<<<<<<<<*******************************************<<<<<<<<<******** *
C ********                                                            ******** *
C ********                 SUBROUTINE -- 'TFORM'                          **** *
C ********                                                                **** *
C ********                                                                **** *
C ********   TRANSFORMATION PARAMETERS SOLVED UNDER THREE CASES.          **** *
C ********   REFER  KCODE(3) ALSO.                                        **** *
C ********                                                            ******** *
C ********                                                            ******** *
C ********<<<<<<<<<*******************************************<<<<<<<<<******** *
C ********<<<<<<<<<*******************************************<<<<<<<<<******** *
```

```
C ********<<<<<<<<<<***********************************************<<<<<<<<<<******** *
C ********<<<<<<<<<<***********************************************<<<<<<<<<<******** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********                  SUBROUTINE -- 'SCALE'                                 **** *
C ********                                                                        **** *
C ********                                                                        **** *
C ********                                                                        **** *
C ********     SCALE FACTOR BETWEEN SYSTEM #1 AND SYSTEM #2 IS                    **** *
C ********     COMPUTED BY COMPARISON OF (I-K)TH CHORDS IN THE                    **** *
C ********     TWO SYSTEMS.  THE WEIGHT FOR EACH ESTIMATE OF                      **** *
C ********     SCALE FACTOR CORRESPONDING TO (I-K)TH CHORD IS                     **** *
C ********     COMPUTED USING VARIANCE - COVARIANCE MATRICES                      **** *
C ********     OF 'I' TH AND 'K' TH POINTS OF SYSTEM #1 AND                       **** *
C ********     SYSTEM #2 THROUGH ERROR PROPOGATION.                               **** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********<<<<<<<<<<***********************************************<<<<<<<<<<******** *
C ********<<<<<<<<<<***********************************************<<<<<<<<<<******** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********                  SUBROUTINE -- 'EULERS'                                **** *
C ********                                                                        **** *
C ********                                                                        **** *
C ********    EULERS ANGLES ARE COMPUTED FROM DIRECTION COSINES.                  **** *
C ********    WEIGHT MATRIX 'P' FOR  'TIK AND DIK'  IS COMPUTED                   **** *
C ********    USING VARIANCE - COVARIANCE MATRICES OF THE POINT                   **** *
C ********    COORDINATES OF THE SYSTEMS BY ERROR PROPOGATION .                   **** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********<<<<<<<<<<***********************************************<<<<<<<<<<******** *
C ********<<<<<<<<<<***********************************************<<<<<<<<<<******** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********                  SUBROUTINE -- 'SETUP'                                 **** *
C ********                                                                        **** *
C ********                                                                        **** *
C ********    SETS UP AND STORES WEIGHT MATRIX 'P' IN VECTOR FORM.                **** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********<<<<<<<<<<***********************************************<<<<<<<<<<******** *
C ********<<<<<<<<<<***********************************************<<<<<<<<<<******** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********                    INPUT --- KCODES                                ******** *
C ********              TO BE CODED WITH EACH DATASET                         ******** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********                                                                    ******** *
C ********<<<<<<<<<<***********************************************<<<<<<<<<<******** *
C ********<<<<<<<<<<***********************************************<<<<<<<<<<******** *
```

```
C ********<<<<<<<<<<*********************************************<<<<<<<<<<******** *
C ********<<<<<<<<<<*********************************************<<<<<<<<<<******** *
C ********                                                                  ******** *
C ********                                                                  ******** *
C ********                                                                  ******** *
C ********                                                                  ******** *
C ********    KCODE( 1) = 'TOTAL NUMBER OF POINTS -- IN (I2) FIELD.             **** *
C ********                                                                      **** *
C ********    KCODE( 2) = 'PARAMETERS REQUIRED IN THE SOLUTION'                 **** *
C ********                  3     DENOTES ONLY TRANSLATIONS OR                  **** *
C ********                        ROTATIONS  --  SEE KCODE(14)                  **** *
C ********                        TO SEPARATE THESE SOLUTIONS.                  **** *
C ********                  4     DENOTES TRANSLATIONS AND SCALE.               **** *
C ********                  7     DENOTES TRANSLATION, SCALE AND                **** *
C ********                        THREE ROTATIONS.                              **** *
C ********                                                                      **** *
C ********    KCODE( 3) = 'WHETHER CONSTRAINED SOLUTION IS NEEDED'              **** *
C ********                  0     DENOTES NO SUCH SOLUTION                      **** *
C ********                  1     DENOTES CONSTRAINED SOLUTION.                 **** *
C ********                                                                      **** *
C ********    KCODE( 4) = 'FIRST SYSTEM IN ELLIPSOIDAL COORDINATES              **** *
C ********                IN DEGREES FOR PHI AND LEMDA --                       **** *
C ********                AND METERS FOR HEIGHTS'                               **** *
C ********                  0     DENOTES NO SUCH CASE                          **** *
C ********                  1     DENOTES SUCH INPUT                            **** *
C ********                                                                      **** *
C ********    KCODE( 5) = 'INPUT FOR FIRST SYSTEM IN GEOS FORMAT                **** *
C ********                  0     DENOTES NO SUCH CASE                          **** *
C ********                  1     DENOTES SUCH INPUT                            **** *
C ********                                                                      **** *
C ********    KCODE( 6) = 'SECOND SYSTEM IN ELLIPSOIDAL COORDINATES             **** *
C ********                IN DEGREES FOR PHI AND LEMDA --                       **** *
C ********                AND HEIGHTS IN METERS'                                **** *
C ********                  0     DENOTES NO SUCH CASE                          **** *
C ********                  1     DENOTES SUCH INPUT                            **** *
C ********                                                                      **** *
C ********    KCODE( 7) = 'INPUT FOR SECOND SYSTEM IN GEOS FORMAT'              **** *
C ********                  0     DENOTES NO SUCH CASE                          **** *
C ********                  1     DENOTES SUCH INPUT                            **** *
C ********                                                                      **** *
C ********    KCODE( 8) = 'VARIANCE - COVARIANCE MATRIX AS DIAGONAL'            **** *
C ********                IN FORMAT  '3F10.2 -- 1 CARDS PER STN.                **** *
C ********                  0     DENOTES NO SUCH CASE                          **** *
C ********                  1     DENOTES SUCH INPUT                            **** *
C ********                                                                      **** *
C ********    KCODE( 9) = 'VARIANCE - COVARIANCE MATRIX IN 3X3 FORM'            **** *
C ********                  0     DENOTES NO SUCH CASE                          **** *
C ********                IN FORMAT  '3F5.2' -- 3 CARDS PER STN.                **** *
C ********                  1     DENOTES SUCH INPUT                            **** *
C ********                                                                      **** *
C ********    KCODE(10) = 'VARIANCE - COVARIANCE MATRIX IN FULL AS              **** *
C ********                UPPER TRIANGLE (ROW-WISE) IN VECTOR FORM'             **** *
C ********                IN FORMAT  '8F10.4' -- EACH NEW ROW TO                **** *
C ********                BEGIN ON A NEW CARD FROM COLUMN 1.                    **** *
C ********                SEE KCODE(12) ALSO.                                   **** *
C ********                  0     DENOTES NO SUCH CASE                          **** *
C ********                  1     DENOTES SUCH INPUTSE                          **** *
C ********                                                                      **** *
```

```
C ********     KCODE(11) = °FOR CONSTRAINED SOLUTION TO BE CODED AS°         **** *
C ********                   °3°  ROTATIONS ARE CONSTRAINED .                  **** *
C ********                   °4°  ROTATIONS AND SCALE ARE CONSTRAINED .        **** *
C ********                                                                     **** *
C ********     KCODE(12) = °TO BE CODED °1° TO OMIT CORRELATION                **** *
C ********                 WITH KCODE(10) AS AN ALTERNATE SOLUTION.            **** *
C ********                                                                     **** *
C ********     KCODE(13) = ° TOTAL NUMBER OF TRANSFORMATIONS                   **** *
C ********                  TO BE PERFORMED ° --- TO BE CODED                  **** *
C ********                  WITH THE LAST DATA SET IN (I2) FIELD.              **** *
C ********                                                                     **** *
C ********     KCODE(14) = °3 PARAMETER SOLUTION ONLY°                         **** *
C ********                   0     SOLUTION FOR TRANSLATIONS                   **** *
C ********                   1     SOLUTION FOR ROTATIONS                      **** *
C ********                                                                     **** *
C ********                                                                     **** *
C ********                                                                 ******** *
C ********                                                                 ******** *
C ********                                                                 ******** *
C ********<<<<<<<<<<****************************************<<<<<<<<<******** *
C ********<<<<<<<<<<****************************************<<<<<<<<<******** *
C ********<<<<<<<<<<****************************************<<<<<<<<<******** *
C ********<<<<<<<<<<****************************************<<<<<<<<<******** *
C ********                                                                 ******** *
C ********                                                                 ******** *
C ********                                                                 ******** *
C ********                 ° HOW TO SETUP INPUT DATA °                     ******** *
C ********                                                                 ******** *
C ********                                                                 ******** *
C ********<<<<<<<<<<****************************************<<<<<<<<<******** *
C ********<<<<<<<<<<****************************************<<<<<<<<<******** *
C ********                                                                 ******** *
C ********                                                                 ******** *
C ********                                                                 ******** *
C ********                                                                     **** *
C ********      FIRST CARD ---      TO CONTAIN ALL KCODES                      **** *
C ********                                                                     **** *
C ********                                                                 ******** *
C ********      CARDS CONTAINING COORDINATES FOR THE FIRST SYSTEM              **** *
C ********      CARDS CONTAINING COORDINATES FOR THE SECOND SYSTEM             **** *
C ********                                                                 ******** *
C ********                                                                 ******** *
C ********      CARDS CONTAINING VARIANCE - COVARIANCE MATRIX                  **** *
C ********      FOR THE FIRST SYSTEM.                                          **** *
C ********      CARDS CONTAINING VARIANCE - COVARIANCE MATRIX                  **** *
C ********      FOR THE SECOND SYSTEM.                                         **** *
C ********                                                                     **** *
C ********                                                                 ******** *
C ********                                                                 ******** *
C ********                                                                 ******** *
C ********<<<<<<<<<<****************************************<<<<<<<<<******** *
C ********<<<<<<<<<<****************************************<<<<<<<<<******** *
C ********<<<<<<<<<<****************************************<<<<<<<<<******** *
C ********                                                                 ******** *
C ********                                                                 ******** *
C ********                                                                 ******** *
C ********                                                                 ******** *
C ********                                                                 ******** *
```

```
      IMPLICIT    REAL * 8(A-H , O-Z)
      REAL        *8 LEMDA,NI,MO2
      DIMENSION   XYZ(35,3),RANGLE(4),VROT(4,4),NAME1(3),
     2            A(3600),W(1200),P(2400),UVW(35,3),NAME2(3),
     3            AA(3,105),BB(3,105),NSTA(35),KSTA(35),KCODE(15)
      COMMON      /WEIGHT/ P
      COMMON      /CODE/  KCODE
      COMMON      /INAME/  NAME1,NAME2
      COMMON      NSTA,KSTA,NN,NM,UVW,XYZ,A,W,KPR,KPARM
      COMMON      /ANGLE/ RANGLE,VROT
      DATA        MINUS/1H-/
      PII     =   3.14159265358979300
      RHO     =   180.D0/PII
      RHOS    =   RHO*3600.D0
      KOUNT    =    1
C
C
C ********     READ IN VARIOUS CODES INVOLVED
C
C
1000  READ        (5,  1) (KCODE(I), I = 1,15),(NAME1(I),I=1,3),
     2            (NAME2(I),I=1,3)
    1 FORMAT      (I2,11I1,I2,2I1,3X,3A4,3X,3A4)
      WRITE       (6, 2) (KCODE(I), I = 1,15)
    2 FORMAT      ('1',//////,25X,'KCODE  INPUT',//,20X,15I2,//)
      NO      =   KCODE(1)
      IF            (KCODE(4).EQ.0.AND.KCODE(5).EQ.0)  GO  TO  12
C
C ********     READ IN DATA FOR THE FIRST SYSTEM
C
      READ        (5,  3) AE1,F
    3 FORMAT      (2F15.10)
      F        =    1.D0/F
      E2      =   2.D0*F - F*F
      IF            ( KCODE(5) .EQ. 1)  GO  TO  6
C
C ********     READ IN ELLIPSOIDAL COORDINATES IN DEGREES AND HEIGHT
C
      DO  5 I =   1 , NO
      READ        (5,  4) NSTA(I),PHI,LEMDA,HT
    4 FORMAT      (I4,5X,3F16.9)
      PHI     =   PHI / RHO
      LEMDA   =   LEMDA / RHO
      WW      =   (1.D0-E2 *DSIN(PHI)*DSIN(PHI))**0.5D0
      UVW(I,1)=   (AE1/WW+HT)*DCOS(PHI)*DCOS(LEMDA)
      UVW(I,2)=   (AE1/WW+HT)*DCOS(PHI)*DSIN(LEMDA)
      UVW(I,3)=   (((AE1*(1.D0-E2 ))/WW)+HT)*DSIN(PHI)
    5 CONTINUE
      GO  TO  15
C
C
C ********     READ IN ELLIPSOIDAL COORDINATES IN GEOS FORMAT
C
C
    6 DO 11 I   =    1 , NO
      READ        (5 , 7) NSTA(I),ISN,IPH,MPH,SPH,ILM,MLM,SLM,HT
    7 FORMAT      (I4,20X,A1,2I3,F8.3,2I3,F8.3,F10.2)
      LEMDA   =    (ILM+((MLM+(SLM/60.D0))/60.D0))/RHO
```

```
      IF          (ISN .EQ. MINUS)  GO  TO  8
      PHI       =   (IPH+((MPH+(SPH/60.D0))/60.D0))/RHO
      GO  TO  10
  8   PHI       =  -(IPH+((MPH+(SPH/60.D0))/60.D0))/RHO
 10   WW        =   (1.D0-E2*DSIN(PHI)*DSIN(PHI))**0.5D0
      UVW(I,1) =  (AE1/WW+HT)*DCOS(PHI)*DCOS(LEMDA)
      UVW(I,2) =  (AE1/WW+HT)*DCOS(PHI)*DSIN(LEMDA)
      UVW(I,3) =  (((AE1*(1.D0-E2))/WW)+HT)*DSIN(PHI)
 11   CONTINUE
      GO  TO  15
C
C
C ********    READ IN RECTANGULAR COORDINATES ( U, V, W ) IN METERS
C
C
 12   DO 14 I =  1 , NO
      READ        (5, 13) NSTA(I),(UVW(I,J),J=1,3)
 13   FORMAT(I4,5X,3F16.5)
 14   CONTINUE
C
C
C **** READ IN COORDINATES OF THE SECOND SYSTEM
C
C
 15   IF             (KCODE(6).EQ.1.OR .KCODE(7).EQ.1)  GO  TO  20
C
C
C ********    READ IN RECTANGULAR COORDINATES ( X, Y, Z ) IN METERS
C
C
      DO 18 I =  1 , NO
      READ        (5, 16) KSTA(I),(XYZ(I,J), J=1,3)
 16   FORMAT      (I4,5X,3F16.9)
 18   CONTINUE
      GO  TO  40
 20   READ        (5, 22) AE2,F
 22   FORMAT      (2F15.10)
      F         =   1.D0/F
      E2        =  2.D0*F - F*F
      IF          ( KCODE(7) .EQ. 1)  GO  TO  25
C
C
C ********    READ IN ELLIPSOIDAL COORDINATES IN DEGREES AND HEIGHT
C
C
      DO 24 I =  1 , NO
      READ        (5, 23) KSTA(I),PHI,LEMDA,HT
 23   FORMAT      (I4,5X,3F16.9)
      PHI       =  PHI / RHO
      LEMDA     =  LEMDA / RHO
      WW        =  (1.D0-E2 *DSIN(PHI)*DSIN(PHI))**0.5D0
      XYZ(I,1)=  (AE2/WW+HT)*DCOS(PHI)*DCOS(LEMDA)
      XYZ(I,2)=  (AE2/WW+HT)*DCOS(PHI)*DSIN(LEMDA)
      XYZ(I,3)=  (((AE2*(1.D0-E2 ))/WW)+HT)*DSIN(PHI)
 24   CONTINUE
      GO  TO  40
C
C
```

```
C ********    READ IN ELLIPSOIDAL COORDINATES IN GEOS FORMAT
C
C
   25  DO 31 I =   1 , NO
       READ        (5, 26) KSTA(I),ISN,IPH,MPH,SPH,ILM,MLM,SLM,HT
   26  FORMAT      (I4,20X,A1,2I3,F8.3,2I3,F8.3,F10.2)
       LEMDA     =   (ILM+((MLM+(SLM/60.D0))/60.D0))/RHO
       IF          (ISN .EQ. MINUS)  GO  TO  28
       PHI       =   (IPH+((MPH+(SPH/60.D0))/60.D0))/RHO
       GO  TO  30
   28  PHI       =  -(IPH+((MPH+(SPH/60.D0))/60.D0))/RHO
   30  WW        =   (1.D0-E2*DSIN(PHI)*DSIN(PHI))**0.5D0
       XYZ(I,1) =  (AE2/WW+HT)*DCOS(PHI)*DCOS(LEMDA)
       XYZ(I,2) =  (AE2/WW+HT)*DCOS(PHI)*DSIN(LEMDA)
       XYZ(I,3) =  (((AE2*(1.D0-E2))/WW)+HT)*DSIN(PHI)
   31  CONTINUE
C
C
C
C **** WRITING OF READ IN DATA FOR THE TWO SYSTEM IN RECTANGULAR COORDINATES
C
C
C
   40  WRITE(6,  42)
   42  FORMAT('1',///,25X,'RECTANGULAR COORDINATES FOR FIRST SYSTEM',///)
       WRITE(6, 43)
   43  FORMAT(' ',13X,'STN.NO.',12X,'U',13X,'V',16X,'W',/)
       DO 46 I =  1 , NO
       WRITE(6,  44) NSTA(I), (UVW(I,J), J=1,3)
   44  FORMAT(' ',13X,I5,F20.4,2F16.4,(14X,I5,F20.4,2F16.4))
   46  CONTINUE
       WRITE(6,50)
   50  FORMAT('1',///,25X,'RECTANGULAR COORDINATES FOR SECOND SYSTEM',/)
       WRITE(6,52)
   52  FORMAT('-',13X,'STN.NO.',12X,'X',13X,'Y',16X,'Z',/)
       DO 60 I =  1 , NO
       WRITE(6,  58) KSTA(I), (XYZ(I,J), J=1,3)
   58  FORMAT(' ',13X,I5,F20.4,2F16.4,(14X,I5,F20.4,2F16.4))
   60  CONTINUE
C
C
C **** SEPARATING THE TYPE OF SOLUTION REQUIRED
C
C
       KPARM     =   KCODE(11)
       IF          (KCODE(8) .NE. 1)  GO  TO  62
       KPR       =   1
       GO  TO  75
   62  IF          (KCODE(9) .NE. 1)  GO  TO  64
       KPR       =   2
       GO  TO  75
   64  KPR       =   3
       IF          (KCODE(10).EQ.1.AND.KCODE(12).EQ.1)  KPR = 2
   75  NM        =   NO - 1
       NN       =  NO * NM
       NNN      =  3*NO
       IF          (KCODE(14) .EQ. 0)  GO  TO  85
       CALL EULERS  (NO,NNN,AA,BB)
```

```
      IF          (KCODE(14).EQ.1.AND.KCODE(2).EQ.3)  GO  TO  95
 85   CALL TFORM (NO,NNN)
      IF (KOUNT .EQ. KCODE(13))  GO  TO  95
      KOUNT    =   KOUNT + 1
      GO  TO  1000
 95   STOP
      END
```

```
C
C
C ******************************************************************<<******
C                                                                          *
C ****                          'EULER'S   ANGLES'                    *** *
C                                                                          *
C                                                                          *
C ******************************************************************<<******
C                                                                          *
C                                                                          *
C **** BETWEEN TWO COORDINATES SYSTEMS COMPUTED FROM DIRECTION COSINES   **** *
C                                                                          *
C                                                                          *
C                                                                          *
C ******************************************************************<<******
C
C
C
C
C
C
      SUBROUTINE EULERS (NO,NNN,AA,BB)
      IMPLICIT REAL *8 (A-H,O-Z)
      REAL        *8 NI,NZ,MO2
      DIMENSION UVW(35,3),XYZ(35,3),A(3600),W(1200),NAME1(3),
     2P1(6,6),G(2,6),GP(2,6),GT(6,2),PP(2,2),KX(2),KY(2),NAME2(3),
     3B(2,4),BT(4,2),P2(6,6),INDEX(40),INV(40),QXYZ(4500),NZ(4,4),
     4P(2400),BS(2,4),KSTA(35),NSTA(35),QUVW(4500),AA(3,NNN),BB(3,NNN),
     5PQ(2,2),PR(4,4),NI(3,3),DX(3),U(3),VAR(3,3),KQ(3),KCODE(15),LQ(3)
      COMMON      /WEIGHT/ P
      COMMON      /CODE/  KCODE
      COMMON      /ANGLE/ S,DX,NZ
      COMMON      /INAME/  NAME1,NAME2
      COMMON        /SFAC/ DW,DS,DA1,DB1,DC1,
     2              DA2,DB2,DC2,RIK1,RIK2,P1,P2
      COMMON      NSTA,KSTA,NN,NM,UVW,XYZ,A,W,KPR,KPARM
      PII      =  3.141592653589793D0
      RHO      =  180.D0/PII
      RHOS     =  RHO*3600.D0
      DW        =   0.D0
      DS        =   0.D0
      S         =   0.D0
      VSF       =   0.D0
      WT        =   0.D0
      LL       =  1
C
C
C
C **** SETTING UP OF MATRIX 'B' -- COMMON TO ALL SOLUTION
C
C
C
      B(1,1)   =  -1.D0
      B(1,2)   =  0.D0
      B(1,3)   =  1.D0
      B(1,4)   =  0.D0
      B(2,1)   =  0.D0
      B(2,2)   =  -1.D0
```

```
      B(2,3)  =  0.D0
      NS         =   NN/2
      B(2,4)  =  1.D0
      DO 1 I  =  1 , 2
      DO 1 J  =  1 , 4
      BT(J,I) =  B(I,J)
    1 CONTINUE
      DO 2 I  =  1 , 4
      DO  2 J =  1 , 4
      PR(I,J) =  0.D0
    2 CONTINUE
      IF          (KCODE(8).EQ.1.OR.KCODE(9).EQ.1)  GO  TO  10
C
C
C
C ********************************************************************<<**** *
C                                                                             *
C                                                                             *
C ****               FULL  VARIANCE-COVARIANCE  CASE                     **** *
C                                                                             *
C                                                                             *
C ********************************************************************<<**** *
C
C
C
C **** READING IN VARIANCE-COVARIANCES FOR 'FIRST  SYSTEM'
C
C
C
      JK         =    1
      DO  6 I   =    1 , NNN
      JL         =    JK + NNN - I
      READ            (5, 3)  (QUVW(J), J = JK,JL)
    3 FORMAT          (8F10.4)
      DO  4 L =  LL , 3
      P1(LL,L)=  QUVW(JK+L-LL)
    4 P1(L,LL)=  P1(LL,L)
      WRITE         (1) (P1(LL,M), M = 1 , 3)
      LL         =  LL + 1
      IF            (LL .EQ. 4)  LL = 1
    6 JK         =  JL + 1
      REWIND        1
C
C
C **** READING IN VARIANCE-COVARIANCES FOR 'SECOND  SYSTEM'
C
C
C
      LL         =  1
      JK          =    1
      DO   9 I   =    1 , NNN
      JL          =    JK + NNN - I
      READ             (5, 7)  (QXYZ(J), J = JK,JL)
    7 FORMAT          (8F10.4)
      DO  8 L =  LL , 3
      P2(LL,L)=  QXYZ(JK+L-LL)
    8 P2(L,LL)=  P2(LL,L)
      WRITE         (2) (P2(LL,M), M = 1 , 3)
```

```
      LL        =  LL + 1
      IF           (LL .EQ. 4)  LL = 1
    9 JK        =    JL + 1
      REWIND       2
      GO  TO  24
C
C
C ************************************************************************<<**** *
C                                                                                 *
C                                                                                 *
C ****               DIAGONAL  OR  3X3  BANDED  CASE                        **** *
C                                                                                 *
C                                                                                 *
C ************************************************************************<<**** *
C
C
C
C ****  READING IN VARIANCE-COVARIANCE FOR FIRST SYSTEM
C
C
   10 DO 17 I =  1 , NO
      KK        =  (I-1)*3 + 1
      KM        =  KK + 2
      IF           (KCODE(8) .EQ. 1)  GO  TO  13
C
C
C
C ****  VARIANCE - COVARIANCE MATRIX IN 3X3 BANDED FORM
C
C
C
C
      DO 12 J =  1 , 3
      READ         (5,11)  (AA(J,K), K = KK,KM)
   11 FORMAT(3F5.2)
   12 WRITE(1)  (AA(J,K), K=KK,KM)
      GO  TO  17
C
C
C ****  VARIANCE - COVARIANCE MATRIX IN DIAGONAL FORM (ONLY VARIANCES)
C
C
   13 DO 14 J  = 1 , 3
      DO 14 K  = KK , KM
   14 AA(J,K)  = 0.D0
      READ         (5,15)  (AA(K,(K+KK-1)), K = 1,3)
   15 FORMAT (3F10.2)
      DO 16 J  = 1 , 3
   16 WRITE(1)     (AA(J,K), K=KK,KM)
   17 CONTINUE
      REWIND  1
C
C
C ****  READING IN VARIANCE-COVARIANCE FOR SECOND SYSTEM
C
C
      DO 23 I =  1 , NO
      KK        =  (I-1)*3 + 1
```

```
      KM         =  KK + 2
      IF            (KCODE(8) .EQ. 1)  GO  TO  20
C
C
C ****  VARIANCE - COVARIANCE MATRIX IN 3X3 BANDED FORM
C
C
      DO 19 J =  1 , 3
      READ          (5,18)  (BB(J,K), K=KK,KM)
  18  FORMAT (3F5.2)
  19  WRITE(2)  (BB(J,K), K=KK,KM)
      GO  TO  23
C
C
C
C
C ****  VARIANCE - COVARIANCE MATRIX IN DIAGONAL FORM (ONLY VARIANCES)
C
C
  20  DO 21 J  = 1 , 3
      DO 21 K  = KK,KM
  21  BB(J,K)  = 0.DO
      READ          (5,15)  (BB(K,(K+KK-1)), K = 1,3)
      DO 22 J  = 1 , 3
  22  WRITE(2)    (BB(J,K), K=KK,KM)
  23  CONTINUE
      REWIND    2
C
C
C ******************************************************************<<******
C                                                                          *
C **** FORMING MATRICES 'A', 'W', AND 'P' FOR THE ENTIRE SYSTEM      **** *
C **** BY COMPUTING DIRECTION COSINES FOR EACH LINE BETWEEN          **** *
C **** ANY ONE SET OF TWO GIVEN POINTS.                              **** *
C                                                                          *
C                                                                          *
C ******************************************************************<<**** *
C
C
  24  MKR       =  1
      KMT       =  1
      MK         =   1
      INDEX(1) =    1
      MM1        =    NNN + 1
      DO 25 I   =    1 , NO
  25  INV(I)     =    3*I - 1
      DO 50 I =  1 , NM
      DO 26 J =  1 , 6
      DO 26 K =  1 , 6
      P1(J,K) =  0.DO
      P2(J,K) =  0.DO
  26  CONTINUE
      IF           (KCODE(10).EQ. 1)  GO  TO  28
      DO 27 J =  1 , 3
      DO 27 L =  1 , 3
      LL      =  (I-1) * 3 + L
      P1(J,L) =  AA(J,LL)
      P2(J,L) =  BB(J,LL)
```

```
   27   CONTINUE
        GO   TO   32
   28   LL          =    INDEX(I)
        DO 30 J     =    1 , 3
        DO 29 L     =    J , 3
        LLL         =    LL + L - J
        P1(J,L)     =    QUVW(LLL)
   29   P2(J,L)     =    QXYZ(LLL)
        MM1         =    MM1 - 1
   30   LL          =    LL + MM1
   32   JJ        =  I + 1
        INDEX(JJ)=    LL
        MM2         =    MM1
        DO 50 K =   JJ , NO
        IF          (KCODE(8).EQ.1.OR.KCODE(9).EQ.1)  GO  TO  43
        LL          =    INDEX(K)
        DO 34 J     =    4 , 6
        DO 33 L     =    J , 6
        LLL         =    LL + L - J
        P1(J,L)     =    QUVW(LLL)
   33   P2(J,L)     =    QXYZ(LLL)
        MM2         =    MM2 - 1
   34   LL          =    LL + MM2
        KP          =    K + 1
        INDEX(KP)=    LL
        III         =    INDEX(I) + INV(K-I)
        IF          (KCODE(12) .EQ. 1)  GO  TO  41
        DO 38 J     =    1 , 3
        DO 36 L     =    4 , 6
        LLL         =    III + L - 3
        P1(J,L)     =    QUVW(LLL)
   36   P2(J,L)     =    QXYZ(LLL)
   38   III         =    III + (NNN -(3*(I-1))-J)
   41   DO 42 J     =    1 , 6
        DO 42 L     =    1 , 6
        P1(L,J)     =    P1(J,L)
   42   P2(L,J)     =    P2(J,L)
        GO   TO   45
   43   DO 44 L =   4 , 6
        JKL       =   L - 3
        DO 44 M =   4 , 6
        KLM       =   (K-2)*3 + M
        P1(L,M) =   AA(JKL,KLM)
        P2(L,M) =   BB(JKL,KLM)
   44   CONTINUE
   45   KSM       =   MKR + NN
        KMS       =   MKR + (2*NN)
C
C
C **** COMPUTING DIRECTION COSINES FOR  FIRST  SYSTEM
C
C
        DA1       =   UVW(K,1) - UVW(I,1)
        DB1       =   UVW(K,2) - UVW(I,2)
        DC1       =   UVW(K,3) - UVW(I,3)
        RIK1      =   DSQRT(DA1*DA1+DB1*DB1+DC1*DC1)
        AIK1      =   DA1/RIK1
        BIK1      =   DB1/RIK1
```

```
      CIK1     =  DC1/RIK1
      TIK1     =  -DATAN2(BIK1,AIK1)
      IF          (TIK1.LT.0.)  TIK1 =(360.D0+TIK1*RHO)/RHO
      AB1      =  DSQRT(AIK1*AIK1+BIK1*BIK1)
      DIK1     =  DATAN2(CIK1,AB1)
C
C
C **** COMPUTING DIRECTION COSINES FOR  SECOND  SYSTEM
C
C
      DA2      =  XYZ(K,1) - XYZ(I,1)
      DB2      =  XYZ(K,2) - XYZ(I,2)
      DC2      =  XYZ(K,3) - XYZ(I,3)
      RIK2     =  DSQRT(DA2*DA2+DB2*DB2+DC2*DC2)
      AIK2     =  DA2/RIK2
      BIK2     =  DB2/RIK2
      CIK2     =  DC2/RIK2
      TIK2     =  -DATAN2(BIK2,AIK2)
      IF          (TIK2.LT.0.)  TIK2 =(360.D0+TIK2*RHO)/RHO
      AB2      =  DSQRT(AIK2*AIK2+BIK2*BIK2)
      DIK2     =  DATAN2(CIK2,AB2)
C
C
C **** SETTING UP MATRICES 'A' AND 'W'  --  COMMON TO ALL SOLUTION
C
C
      A(MKR)   =  1.D0
      A(MKR+1)=   0.D0
      A(KSM)   =  DSIN(TIK2)*DTAN(DIK2)
      A(KSM+1)=   DCOS(TIK2)
      A(KMS)   =  -DCOS(TIK2)*DTAN(DIK2)
      A(KMS+1)=   DSIN(TIK2)
      W(MKR)   =  TIK1 - TIK2
      W(MKR+1)=   DIK1 - DIK2
C
C ***********************************************************************<<**** *
C                                                                               *
C                                                                               *
C **** FORMING VAR-COVARIANCE MATRIX FOR 'TIK' AND 'DIK'                   **** *
C **** THROUGH PROPOGATION OF ERRORS  --  WHERE 'TIK' AND                  **** *
C **** ARE GEODETIC HOUR ANGLE AND DECLINATION.                            **** *
C                                                                               *
C                                                                               *
C ***********************************************************************<<**** *
C
C
C
C ****                            FIRST   SYSTEM                           **** *
C
C
C
C
      DAB1     =  DA1*DA1+DB1*DB1
      DBA      =  DSQRT(DAB1)
      G(1,1)   =  -DB1/DAB1
      G(1,2)   =  DA1/DAB1
      G(1,3)   =  0.D0
      G(1,4)   =  -G(1,1)
```

```
      G(1,5)   =  -G(1,2)
      G(1,6)   =  0.D0
      G(2,1)   =  DA1*DC1/(DBA*RIK1*RIK1)
      G(2,2)   =  DB1*DC1/(DBA*RIK1*RIK1)
      G(2,3)   =  -DBA/(RIK1*RIK1)
      G(2,4)   =  -G(2,1)
      G(2,5)   =  -G(2,2)
      G(2,6)   =  -G(2,3)
      DO 46 L  =   1 , 2
      DO 46 M  =   1 , 6
      GT(M,L) =  G(L,M)
   46 CONTINUE
      CALL        DGMPRD(G,P1,GP,2,6,6)
      CALL        DGMPRD(GP,GT,PP,2,6,2)
C
C
C
C ****                       SECOND   SYSTEM                          **** *
C
C
C
      DAB2     =  DA2*DA2+DB2*DB2
      DAB      =  DSQRT(DAB2)
      G(1,1)   =  -DB2/DAB2
      G(1,2)   =  DA2/DAB2
      G(1,3)   =  0.D0
      G(1,4)   =  -G(1,1)
      G(1,5)   =  -G(1,2)
      G(1,6)   =  0.D0
      G(2,1)   =  DA2*DC2 / (DAB*RIK2*RIK2)
      G(2,2)   =  DB2*DC2 / (DAB*RIK2*RIK2)
      G(2,3)   =  -DAB/(RIK2*RIK2)
      G(2,4)   =  -G(2,1)
      G(2,5)   =  -G(2,2)
      G(2,6)   =  -G(2,3)
      DO 47 L  =   1 , 2
      DO 47 M  =   1 , 6
      GT(M,L) =  G(L,M)
   47 CONTINUE
      CALL        DGMPRD(G,P2,GP,2,6,6)
      CALL        DGMPRD(GP,GT,PQ,2,6,2)
C
C
C
C **** FORMING MATRIX 'MI' FOR THE COMBINED SYSTEM
C
C
C
      DO 48 L =  1 , 2
      J       =  L + 2
      DO 48 M =  1 , 2
      N       =  M + 2
      PR(L,M) =  PQ(L,M)
      PR(J,N) =  PP(L,M)
   48 CONTINUE
      CALL        DGMPRD(B,PR,BS,2,4,4)
      CALL        DGMPRD(BS,BT,PP,2,4,2)
      CALL        DMINV(PP,2,DT,KX,KY)
```

```
      P(KMT)   =  PP(1,1)
      P(KMT+1)=  PP(2,1)
      P(KMT+2)=  PP(1,2)
      P(KMT+3)=  PP(2,2)
      MKR       =  MKR + 2
      KMT       =  KMT + 4
C ********                                                              ******** *
      IF            (KCODE(11) .EQ. 3)  GO  TO  50
      CALL          SCALE (NS,MK,S,VSF,WT)
      MK       =    MK + 1
C ********                                                              ******** *
C ********<<<<<<<<<<*****************************************<<<<<<<<<<******** *
C ********                                                              ******** *
C ********                                                              ******** *
C ********          FINDING WEIGHTED MEAN AND VARIANCE FOR              ******** *
C ********          'SCALE FACTOR' BY COMPARISON OF CHORDS IN           ******** *
C ********          THE TWO SYSTEMS BY CALLING SUBROUTINE 'SCALE'.      ******** *
C ********                                                              ******** *
C ********                                                              ******** *
C ********                                                              ******** *
C ********<<<<<<<<<<*****************************************<<<<<<<<<<******** *
C ********                                                              ******** *
C ********                                                              ******** *
 50   CONTINUE
      VSF       =   VSF * 10.D11
      DO 75  J =  1 , 3
      IJK       =  (J-1)*NN + 1
      JKL       =  IJK + NN - 1
 75   WRITE(3)    (A(I), I = IJK , JKL)
      REWIND       3
      WRITE(4)    (W(K), K=1,NN)
      REWIND       4
C
C
C
C **** FORMING MATRIX 'N' AND INVERTING THE SAME
C
C
C
      DO 80 I =  1 , 3
      READ (3)    (W(J),J=1,NN)
      K1        =  (I-1)*NN + 1
      K2        =  K1 + NN - 1
      MMM       =  0
      DO 78 K =  K1 , K2
      A(K)      =  0.D0
      L1        =  (((K-K1)/2)*2) + 1
      L2        =  L1 + 1
      DO 78 L =  L1 , L2
      MMM       =  MMM + 1
  78  A(K)      =  A(K) + W(L)*P(MMM)
  80  CONTINUE
      DO 84 I  =   1 , 4
      DO 84 J  =   1 , 4
  84  NZ(I,J)  =   0.D0
      REWIND       3
      DO 88 I =  1 , 3
      READ(3)     (W(L), L=1 , NN)
```

```
      DO 85 J =  1 , 3
      NI(J,I) =  0.D0
      DO 85 K =  1 , NN
      III     =  (J-1)*NN + K
   85 NI(J,I) =  NI(J,I) + A(III)*W(K)
   88 CONTINUE
      REWIND     3
      IF           (KCODE(11) .EQ. 3)  GO  TO  89
      NZ(1,1)  =   WT
   89 DO 91 I  =   2 , 4
       DO 91 J  =   2 , 4
   91 NZ(I,J)  =  NI(I-1,J-1)
      CALL  DMINV(NI,3,DET,KQ,LQ)
C
C
C ******************************************************************<<******
C                                                                          *
C **** COMPUTING SOLUTION VECTOR ' DX ' FOR 3 ROTATION PARAMETERS     **** *
C                                                                          *
C ******************************************************************<<******
C
C
      READ(4)     (W(I), I=1 , NN)
      REWIND      4
      DO 92 J =   1 , 3
      U(J)    =   0.D0
      DO 92 I =   1 , NN
      KKK     =   (J-1)*NN + I
      U(J)    =   U(J) - A(KKK)*W(I)
   92 CONTINUE
      CALL  DGMPRD(NI,U,DX,3,3,1)
      DO 95 I =   1 , 3
      JK      =   (I-1)*NN + 1
      JM      =   JK + NN - 1
   95 READ(3)     (A(J), J= JK , JM)
      REWIND      3
C
C
C
C
C **** COMPUTING VARIANCE OF UNIT WEIGHT ' MO2 '
C
C
C
C
      DO 96 I =   1 , NN
      W(I)    =   0.D0
      DO 96 J =   1 , 3
      K       =   (J-1)*NN + I
      W(I)    =   W(I) - A(K)*DX(J)
   96 CONTINUE
      READ(4)     (A(I), I= 1 , NN)
      REWIND      4
      DO 97 K = 1 , NN
      W(K)    =   W(K) - A(K)
   97 CONTINUE
      MMM     =   0
      DO 98 K =   1 , NN
```

```
      A(K)      =  0.D0
      L1        =  ((K-1)/2)*2 + K
      L2        =  L1 + 2
      DO 98 L =  L1 , L2 , 2
      MMM       =  ((L-1)/2) + 1
   98 A(K)      =  A(K) + P(L)*W(MMM)
      READ(4)      (W(I), I = 1 , NN)
      REWIND       4
      VPV       =  0.D0
      DO 99 K =  1 , NN
   99 VPV       =  VPV - A(K)*W(K)
      MO2       =  VPV/(NN - 3)
C
C
C
C **** COMPUTING VARIANCE- COVARIANCE MATRIX ' VAR '
C
C
C
      DO 100 I=  1 , 3
      DO 100 J=  1 , 3
      VAR(I,J)=  MO2*RHOS*RHOS*NI(I,J)
  100 CONTINUE
      DO 105 I=  1 , 3
  105 DX(I)     =  DX(I)*RHOS
C
C
C **** COMPUTING COEFFICIENTS OF CO-RELATIONS FOR PARAMETERS
C
C
      DO 110 I=  1 , 3
      IF          (I.EQ.3)  GO  TO  107
      JJ        =  I + 1
      DO 106 J=  JJ , 3
      NI(I,J) =  VAR(I,J)/(DSQRT(VAR(I,I))*DSQRT(VAR(J,J)))
  106 NI(J,I) =  NI(I,J)
  107 NI(I,I) =  1.D0
  110 CONTINUE
C
C ********************************************************************<<******
C                                                                            *
C                                                                            *
C **** WRITING OF FINAL SOLUTION VECTOR AND VARIANCE-COVARIANCE MATRIX       *
C                                                                            *
C                                                                            *
C ********************************************************************<<******
C
      WRITE(6,6025)
 6025 FORMAT('1',///)
      WRITE(6,6028)   (NAME1(I),I=1,3),(NAME2(I),I=1,3)
 6028 FORMAT(' ',5X,3A4,'-TO-',3A4,/,
     26X,'*****************************'////)
      WRITE(6,6030)
 6030 FORMAT(' ',30X,'SOLUTION FOR ''3'' ROTATION PARAMETERS',/,
     231X,'-------------------------------------',/,
     325X,'(FROM DIRECTION COSINES -- UNITS SECONDS OF ARC)',/)
      GO  TO  (112,114,116), KPR
  112 WRITE(6,6031)
```

```
6031 FORMAT(' ',37X,'(USING VARIANCES ONLY)',//)
     GO  TO  120
114  WRITE(6,6032)
6032 FORMAT(' ',21X,
    2'(USING 3X3 BANDED DIAGONAL VARIANCE-COVARIANCE MATRIX)',//)
     GO  TO  120
116  WRITE(6,6033)
6033 FORMAT(' ',29X,'(USING FULL VARIANCE-COVARIANCE MATRIX)',//)
120  WRITE(6,6035)
6035 FORMAT(' ',20X,'OMEGA',19X,'PSI',20X,'EPSILON',//)
     WRITE(6,6040)(DX(I), I=1,3)
6040 FORMAT(' ', 5X,3D24.7,//)
     WRITE(6,6045)
6045 FORMAT(' ',32X,'VARIANCE - COVARIANCE MATRIX',/,
    233X,'----------------------------',/)
     WRITE(6,6048) MO2
6048 FORMAT(' ',17X,'MO2=',F6.2,//)
     WRITE(6,6050) ((VAR(I,J), J=1,3), I=1,3)
6050 FORMAT(' ', 3X,3D25.8,//( 4X,3D25.8,/))
     WRITE(6,6075)
6075 FORMAT(' ',33X,'COEFFICIENT OF CORRELATION',/,
    234X,'--------------------------',//)
     WRITE(6,6085) ((NI(I,J),J=1,3),I=1,3)
6085 FORMAT(' ', 3X,3D25.8,//( 4X,3D25.8,/))
     IF            (KCODE(11) .EQ. 3)  GO  TO  150
     WRITE (6,7000)
7000 FORMAT('-',//,34X,'SOLUTION FOR SCALE FACTOR',/,
    234X,'-------------------------',/,
    335X,'(FROM CHORD COMPARISON)',//)
     WRITE (6,7004)
7004 FORMAT(' ',20X,'SCALE FACTOR',27X,'VARIANCE',/,
    223X,'(10.D+5)',29X,'(10.D+11)',//)
     WRITE (6,7010) S , VSF
7010 FORMAT(' ',20X,F8.2,30X,F7.2,//)
150  KCODE(11) =  4
     RETURN
     END
```

```
C
C
C *******************************************************************<<**** *
C                                                                           *
C                                   'TFORM'                                 *
C                                                                           *
C                                                                           *
C *******************************************************************<<**** *
C                                                                           *
C                                                                           *
C                                                                           *
C **** PROGRAM TO TRANSFORM  ONE RECTANGULAR COORDINATES SYSTEM             *
C **** TO SECOND RECTANGULAR COORDINATES SYSTEM AND VICE-VERSA              *
C                                                                           *
C                                                                           *
C *******************************************************************<<**** *
C                                                                           *
C *******************************************************************<<**** *
C
C
C
      SUBROUTINE TFORM  (NO,NQ)
      IMPLICIT REAL * 8 (A-H, O-Z)
      REAL * 8        MI,KK,KL,NI,MO2
      DIMENSION XYZ(35,3),UVW(35,3),SIGMAX(7,7),NAME1(3),
     2A(3600),W(1200),VAR(7,7),DX(7),NI(49),NSTA(35),NAME2(3),
     3ANG(4),U(7),LT(7),KCODE(15),CNT(7,4),TT(7,4),CN(4,7),ZP(4,4),
     4MT(7),KSTA(35),VR(7,7),XD(7),KL(150),KK(150),MI(2400),ROT(4,4)
      COMMON      /WEIGHT/ MI
      COMMON      /CODE/  KCODE
      COMMON      /ANGLE/ ANG,ROT
      COMMON      /INAME/  NAME1,NAME2
      COMMON        /CRNT/ VPV,DX,SO2,XD,SIGMAX
      COMMON      NSTA,KSTA,NN,NM,UVW,XYZ,A,W,KPR,KPARM
      PII= 3.14159265358979300
      RHO = 180.D0/PII
      RHOS = RHO*3600.D0
      IPARA   =  KCODE(2)
      IC      =  KCODE(11)
      KOUNT  = 1
      DO  5 I   =  1 , 4
      DO  5 J   =  1 , 7
      CN(I,J)   =  0.D0
      TT(J,I)   =  0.D0
    5 CNT(J,I) =  0.D0
      DO 10 I   =  1 , 4
      DO 10 J   =  1 , 4
   10 ZP(I,J)   =  0.D0
C
C
C
C **** SETTING UP MATRIX 'A' -- COMMON TO ALL SOLUTION
C
C
C
      NNN = 6*NO
      NNZ     = NQ*IPARA
      DO 13 I = 1 , NNZ
```

```
      A(I)     = 0.D0
 13   CONTINUE
      DO 15 I = 1, NO
      KKK      = (3*I-2)
      LLL      = KKK+NQ+1
      MMM      = LLL+NQ+1
      A(KKK)    =    1.D0
      A(LLL)    =    1.D0
      A(MMM)    =    1.D0
C
C
C **** SETTING UP MATRIX 'W' WHICH IS COMMON TO ALL SOLUTION
C
C
C
      W(KKK)   = (UVW(I,1)-XYZ(I,1))
      W(KKK+1)= (UVW(I,2)-XYZ(I,2))
      W(KKK+2)= (UVW(I,3)-XYZ(I,3))
 15   CONTINUE
      IF (KCODE(2) .NE. 3)  GO  TO  50
C
C
C **** SOLUTION FOR 3 TRANSLATION PARAMETERS
C
C
C
      N = 3
      ICASE    = 1
      GO  TO  81
C
C
C
C
C **** SOLUTION FOR 3 TRANSLATION AND 1 SCALE PARAMETERS
C
C
C
 50   N = 4
      DO 60 I = 1, NO
      KKK      = 3*(NQ+1)-2
      A(KKK)   = UVW(I,1)
      A(KKK+1)= UVW(I,2)
      A(KKK+2)= UVW(I,3)
 60   CONTINUE
      IF (KCODE(2) .NE. 4)  GO  TO  70
      ICASE    = 2
      GO  TO  81
C
C
C
C **** SOLUTION FOR 3 TRANSLATION, 1 SCALE AND 3 ROTATION PARAMETERS
C
C
C
 70   N = 7
      ICASE    = 3
      DO 80 I = 1, NO
      KKK      = 4*NQ+(3*I-2)
```

```
      LLL      = KKK + NQ
      MMM      = LLL + NQ + 1
      A(KKK)      = UVW(I,2)
      A(KKK+1)    =-UVW(I,1)
      A(LLL)      =-UVW(I,3)
      A(LLL+2) = UVW(I,1)
      A(MMM)   =   UVW(I,3)
      A(MMM+1) =-UVW(I,2)
 80   CONTINUE
 81   DO  85 I = 1 , N
      KKK      = (I-1)*NQ+1
      LLL      = KKK+NQ-1
      WRITE(3) (A(J), J=KKK,LLL)
 85   CONTINUE
      REWIND       3
      WRITE(4) (W(I), I=1,NQ)
      REWIND    4
C
C ******************************************************************<<**** *
C                                                                          *
C **** FORMING NORMAL EQUATIONS -- MATRICES 'N' AND 'U'                    *
C                                                                          *
C                                                                          *
C ******************************************************************<<**** *
C
 100  CALL   SETUP (NO,NQ,IPARA)
      DO 118 I = 1 , N
      READ(3) (W(J), J=1,NQ)
      K1       =  (I-1)*NQ+1
      K2       = K1+NQ-1
      MMM      =  0
      DO 116 K= K1, K2
      A(K)     =  0.D0
      L1       = (((K-K1)/3)*3)+1
      L2       = L1 + 2
      DO 116 L = L1 , L2
      MMM      = MMM + 1
 116  A(K)     = A(K) + W(L)*MI(MMM)
 118  CONTINUE
      REWIND       3
      DO 120 I = 1 , N
      READ(3) (W(L), L= 1,NQ)
      JK       = (I-1)*N+1
      JL       = JK+N-1
      DO 119 J = JK,JL
      NI(J)    = 0.D0
      DO 119 K  = 1 , NQ
      II       =  (J-JK)*NQ + K
 119  NI(J)    = NI(J) + A(II)*W(K)
 120  CONTINUE
      REWIND       3
      DO 121 I = 1 , N
      DO 121 J = 1 , N
      K    =  (I-1)*N +J
 121  SIGMAX(I,J) = NI(K)
      READ(4)  (W(I), I= 1,NQ)
      REWIND    4
      DO 122 J=1 , N
```

```
      U(J)    =  0.D0
      DO 122 I=1  , NQ
      KKK       =  (J-1)*NQ+I
      U(J)    = U(J) - A(KKK)*W(I)
 122  CONTINUE
C
C ***********************************************************************<<******
C                                                                               *
C **** COMPUTING SOLUTION VECTOR 'DX'  FOR TRANSFORMATION PARAMETERS            *
C                                                                               *
C                                                                               *
C ***********************************************************************<<******
C
      CALL  DMINV(NI,N,DT,LT,MT)
      CALL   DARRAY(1,N,N,7,7,NI,VR)
      CALL  DGMPRD(NI,U,DX,N,N,1)
      DO 123 I   = 1 , N
      JK          =  (I-1)*NQ + 1
      JM          =  JK + NQ -1
 123  READ(3)  (A(J),J=JK,JM)
      REWIND        3
C
C
C **** COMPUTING VARIANCE OF UNIT WEIGHT 'MO2'
C
C
      DO 125 I = 1 , NQ
      W(I)    = 0.D0
      DO 125 J  = 1 , N
      KZX       = (J-1)*NQ+I
      W(I)    = W(I) -A(KZX)*DX(J)
 125  CONTINUE
      READ(4)  (A(I), I= 1, NQ)
      REWIND   4
      DO 126 K = 1, NQ
      W(K)    =  W(K)-A(K)
 126  CONTINUE
      MMM     = 0
      DO 128 K = 1 , NQ
      A(K)    = 0.D0
      L1    = ((K-1)/3)*6 + K
      L2     =  L1 + 6
      DO 128 L = L1,L2,3
      MMM       =  ((L-1) /3) +1
 128  A(K)      = A(K) +MI(L)*W(MMM)
      CALL         RESIDU  (NQ,NNN)
      READ(4)  (W(I),I= 1,NQ)
      REWIND   4
      VPV     = 0.D0
      DO 130 K = 1,NQ
 130  VPV     = VPV - A(K)*W(K)
      MO2     =  VPV/(NQ-N)
C
C
C **** COMPUTING VARIANCE-COVARIANCE MATRIX 'VAR'
C
C
      DO 132 I = 1, N
```

```
      DO 132 J = 1, N
      VAR(I,J) = MO2*VR(I,J)
 132  CONTINUE
      IF (KCODE(2) .EQ. 3)  GO  TO  140
      DX(4) = DX(4) * 10.D5
      IF (KCODE(2) .EQ. 4)  GO  TO  140
      DO 135 I = 5 , 7
      DX(I) = DX(I) * RHOS
 135  CONTINUE
C
C
C
C **** COMPUTING COEFFICIENTS OF CORRELATIONS FOR PARAMETERS
C
C
C
 140  DO 145 I = 1,N
      IF(I.EQ.N)  GO  TO  144
      JJ   = I + 1
      DO 142 J = JJ , N
      VR(I,J)  =  VAR(I,J)/(DSQRT(VAR(I,I))*DSQRT(VAR(J,J)))
 142  VR(J,I)  = VR(I,J)
 144  VR(I,I)  = 1.D0
 145  CONTINUE
 200  WRITE(6,250)
 250  FORMAT('1',//)
      WRITE(6, 300)  (NAME1(I),I=1,3),(NAME2(I),I=1,3)
 300  FORMAT(' ',5X,3A4,'-TO-',3A4,/,
     26X,'****************************'///)
      GO  TO (500,600,700) , ICASE
C
C
C
C ******************************************************************<<******
C                                                                          *
C                                                                          *
C **** WRITING OF FINAL SOLUTION VECTOR AND VARIANCE-COVARIANCE MATRIX     *
C                                                                          *
C                                                                          *
C                                                                          *
C ******************************************************************<<******
C
C
C
 500  WRITE(6,6025)
 6025 FORMAT(' ',/////)
      WRITE(6,6030)
 6030 FORMAT('-',21X,'SOLUTION FOR 3 TRANSLATION PARAMETERS',/,
     232X,'(UNITS - METERS)',///)
      GO  TO  (512,514,516), KPR
 512  WRITE(6,6032)
 6032 FORMAT(' ',29X,'(USING VARIANCES ONLY)',//)
      GO  TO  520
 514  WRITE(6,6034)
 6034 FORMAT(' ',15X,
     2'(USING 3X3 BANDED DIAGONAL VARIANCE-COVARIANCE MATRIX)',//)
      GO  TO  520
 516  WRITE(6,6036)
```

```
6036 FORMAT(' ',22X,'(USING FULL VARIANCE-COVARIANCE MATRIX)',//)
520  WRITE(6,6038)
6038 FORMAT('-',16X,'DX',20X,'DY',22X,'DZ',//)
     WRITE(6,6040)(DX(I), I=1,3)
6040 FORMAT(' ', 1X,3D23.8,//////)
     WRITE(6,6045)
6045 FORMAT(' ',26X,'VARIANCE - COVARIANCE MATRIX',//)
     WRITE(6,6048) MO2
6048 FORMAT('-',14X,'MO2=',F6.2,//)
     WRITE(6,6050) ((VAR(I,J), J=1,3), I=1,3)
6050 FORMAT(' ', 1X,3D23.8,//( 2X,3D23.8,/))
     WRITE(6,6075)
6075 FORMAT('-',//,27X,'COEFFICIENTS OF CORRELATION',////)
     WRITE(6,6085)((VR(I,J), J=1,N), I=1,N)
6085 FORMAT(' ', 1X,3D23.8,//( 2X,3D23.8,/))
     GO  TO  1000
600  WRITE(6,6500)
6500 FORMAT(' ',/////)
     WRITE(6,6510)
6510 FORMAT('-',17X,'SOLUTION FOR 3 TRANSLATION AND 1 SCALE PARAMETERS'
    2,/,34X,'(UNITS - METERS)',///)
     GO  TO  (612,614,616), KPR
612  WRITE(6,6512)
6512 FORMAT(' ',29X,'(USING VARIANCES ONLY)',//)
     GO  TO  620
614  WRITE(6,6514)
6514 FORMAT(' ',19X,
    2'(USING 3X3 BANDED DIAGONAL VARIANCE-COVARIANCE MATRIX)',//)
     GO  TO  620
616  WRITE(6,6516)
6516 FORMAT(' ',22X,'(USING FULL VARIANCE-COVARIANCE MATRIX)',//)
620  WRITE(6,6520)
6520 FORMAT(' ', 6X,'DX',22X,'DY',23X,'DZ',22X,'DL',//)
     WRITE(6,6550)(DX(I), I=1,4)
6550  FORMAT(' ',D15.8,3D24.8//////)
     WRITE(6,6600)
6600 FORMAT(' ',26X,'VARIANCE - COVARIANCE MATRIX',//)
     WRITE(6,6625) MO2
6625 FORMAT(' ', 8X,'MO2=',F6.2,//)
     WRITE(6,6650) ((VAR(I,J), J=1,4), I=1,4)
6650 FORMAT(' ', 1X,4D20.8,//( 2X,4D20.8,/))
     WRITE(6,6675)
6675 FORMAT('-',//,27X,'COEFFICIENTS OF CORRELATION',////)
     WRITE(6,6685)((VR(I,J), J=1,N), I=1,N)
6685 FORMAT(' ', 1X,4D20.8,//( 2X,4D20.8,/))
     GO  TO  1000
700  GO TO ( 710,705) , KOUNT
705  IF (KPARM  .EQ.   4 )  GO  TO  708
     WRITE (6,7002)
7002  FORMAT(' ',28X,'ROTATION PARAMETERS CONSTRAINED',/,
    229X,'-------------------------------',//)
     GO  TO  710
708  WRITE(6,7005)
7005 FORMAT(' ',20X,'SCALE FACTOR AND ROTATION PARAMETERS CONSTRAINED',
    2/,21X,'------------------------------------------------',///)
710  WRITE(6,7010)
7010 FORMAT(' ',13X,'SOLUTION FOR 3 TRANSLATION, 1 SCALE AND 3 ROTATION
    2 PARAMETERS',/,14X,'---------------------------------------------'
```

```
    3,T58,'------------------   ',/)
      GO  TO  (712,714,716), KPR
 712  WRITE(6,7012)
 7012 FORMAT(' ',34X,'(USING VARIANCES ONLY)',//)
      GO  TO  720
 714  WRITE(6,7014)
 7014 FORMAT(' ',16X,
     2'(USING 3X3 BANDED DIAGONAL VARIANCE-COVARIANCE MATRIX)',//)
      GO  TO  720
 716  WRITE(6,7016)
 7016 FORMAT(' ',24X,'(USING FULL VARIANCE-COVARIANCE MATRIX)',//)
 720  WRITE(6,7020)
 7020 FORMAT(' ',16X,'DX', 6X,'DY', 6X,'DZ', 7X,'DL', 5X,'OMEGA',
     2T 59,'PSI', 4X,'EPSILON',/,
     315X,'METERS', 2X,'METERS', 2X,'METERS',1X,'(10.D+5)',1X,'SECONDS',
     4T57,'SECONDS', 2X,'SECONDS',/)
      WRITE(6,7030) DX
 7030 FORMAT(' ',12X,F7.2,2F8.2,F8.2,F9.2,T55,F8.2,F9.2,//)
      WRITE(6,7040)
 7040 FORMAT('0',28X,'VARIANCE - COVARIANCE MATRIX',//)
      WRITE(6,7045) MO2
 7045 FORMAT(' ',10X,'MO2=',F6.2,/)
      WRITE(6,7050)  ((VAR(I,J), J=1,7), I = 1,7)
 7050 FORMAT(' ',2X,7D11.3,//(3X,7D11.3,/))
      WRITE(6,7075)
 7075 FORMAT(' ',/, 29X,'COEFFICIENTS OF CORRELATION',//)
      WRITE(6,7085)((VR(I,J), J=1,N), I=1,N)
 7085 FORMAT(' ', 2X,7D11.3,//( 3X,7D11.3,/))
      IF(IC.EQ.0)  GO  TO  1000
      WRITE (6,7090)
7090  FORMAT ('1',////,36X,'RESIDUALS V',/,36X,'-----------',///,
     212X,'FIRST SYSTEM',33X,'SECOND SYSTEM',/)
      KSM  =  NNN + 1
      KMR  =  NNN - 1 +KSM
      WRITE (6,8000)  (A(I), I = KSM,KMR)
8000  FORMAT(' ',4X,3F8.3,22X,3F8.3,/(5X,3F8.3,22X,3F8.3))
      IF          (KCODE(3) .EQ. 0)  GO  TO  1000
C
C **********************************************************************<<******
C                                                                              *
C                                                                              *
C                                                                              *
C **** OBTAINING CONSTRAINED SOLUTION FOR ROTATION PARAMETERS                  *
C                                                                              *
C                                                                              *
C                                                                              *
C **********************************************************************<<******
C
C
      CALL CSTRNT(N,NQ,IC,U,CN,CNT,TT,ZP)
      KCODE(3)  =  0
      DO 725 I = 1 , 7
      DX(I)    = XD(I)
      DO 725 J = 1 , 7
      VAR(I,J) = SIGMAX(I,J)
 725  CONTINUE
      DO 750 I = 1,N
      IF(I.EQ.N)  GO  TO  740
```

```
      JJ    = I + 1
      DO 735 J = JJ , N
      VR(I,J)  =  VAR(I,J)/(DSQRT(VAR(I,I))*DSQRT(VAR(J,J)))
735   VR(J,I)  = VR(I,J)
740   VR(I,I)  = 1.D0
750   CONTINUE
      KOUNT = 2
      MO2      = SO2
      GO  TO  200
1000  RETURN
      END
```

```
C ********                                                           ******** *
C ********                                                           ******** *
C ********                                                           ******** *
C ********<<<<<<<<<*******************************************<<<<<<<<<******** *
C ********                                                           ******** *
C ********                                                           ******** *
C ********                     SUBROUTINE  SCALE                     ******** *
C ********                                                           ******** *
C ********                                                           ******** *
C ********                                                           ******** *
C ********<<<<<<<<<*******************************************<<<<<<<<<******** *
C ********                                                           ******** *
C ********                                                           ******** *
C ********         FINDING WEIGHTED MEAN AND VARIANCE FOR            ******** *
C ********         'SCALE FACTOR' BY COMPARISON OF CHORDS IN         ******** *
C ********         THE TWO SYSTEMS BY CALLING SUBROUTINE 'SCALE'.    ******** *
C ********                                                           ******** *
C ********                                                           ******** *
C ********                                                           ******** *
C ********<<<<<<<<<*******************************************<<<<<<<<<******** *
C ********                                                           ******** *
C ********                                                           ******** *
C ********                                                           ******** *
C ********                                                           ******** *
      SUBROUTINE    SCALE (NO,N,S,VSF,WT)
      IMPLICIT      REAL * 8 (A-H , O-Z)
      DIMENSION     P(12,12),H(12),PF(6,6),PS(6,6),
     2              HI(12),DL(600),VI(600),WI(600)
      COMMON        /SFAC/ SW,SF,DU,DV,DW,DX,DY,DZ,R1,R2,PF,PS
      RR      =     R1 * R2
      RT      =     R2 /(R1**3)
C ********
C ********
C ********
C ********  —  SETTING UP OF VARIANCES FOR EACH CHORD THROUGH ERROR PROPOGATION
C ********
C ********
C ********
      H(1)    =     DU * RT
      H(2)    =     DV * RT
      H(3)    =     DW * RT
      H(7)    =     -DX/RR
      H(8)    =     -DY/RR
      H(9)    =     -DZ/RR
      DO 10 I =     1 , 3
      H(I+3)  =     -H(I)
      H(I+9)  =     -H(I + 6)
   10 CONTINUE
      DO 15 I =     1 , 12
      DO 15 J =     1 , 12
   15 P(I,J)  =     0.D0
      DO 20 I =     1 , 6
      L       =     I + 6
      DO 20 J =     1 , 6
      M       =     J + 6
      P(I,J)  =     PF(I,J)
   20 P(L,M)  =     PS(I,J)
      CALL          DGMPRD (H,P,HI,1,12,12)
```

```
      CALL          DGMPRD (HI,H,WS,1,12,1)
C ********
C ********
C ********      FINDING WEIGHTED MEAN FOR SCALE FACTOR OF THE GIVEN SAMPLE
C ********
C ********
      WS       =    1.DO/WS
      WI(N)    =    WS
      SFI      =    R2/R1 - 1
      DL(N)    =    SFI
      SF       =    SF + SFI *WS
      SW       =    SW + WS
      S        =    SF/SW
C ********
C ********
C ********
C ********      FINDING VARIANCE FOR THE WEIGHTED MEAN OF THE SCALE FACTOR
C ********
C ********
C ********
      IF            (N .NE. NO)  GO  TO  500
      PVV      =    0.DO
      DO 50 K  =    1 , NO
      VI(K)    =    ((S-DL(K))**2)*WI(K)
   50 PVV      =    PVV + VI(K)
      VSF      =    PVV/(SW*(NO-1))
      S        =    S * 10.D5
      WT       =    1.DO / VSF
  500 RETURN
      END
```

```
C
C
C
C
C *******************************************************************<<******
C                                                                            *
C                                    ' CSTRNT '                              *
C                                                                            *
C *******************************************************************<<******
C                                                                            *
C                                                                            *
C **** SUBROUTINE SOLVES FOR CONSTRAINED CASE IN RESPECT OF '3'              *
C **** ROTATION PARAMETERS.  CONSTRAINTS ARE CODED FOR 'ALL' THE             *
C **** PARAMETERS -- BLANKS CARDS ARE NEEDED FOR NON-CONSTRAINTS.            *
C                                                                            *
C **** INPUT CONSTRAINTS FOR ROTATION PARAMETERS ARE IN SECONDS OF ARC.      *
C                                                                            *
C                                                                            *
C *******************************************************************<<******
C ********                                                          ******** *
C ********                                                          ******** *
      SUBROUTINE CSTRNT(N,NN,IC,WS,CN,CNT,TT,ZP)
      IMPLICIT REAL * 8 (A-H , O-Z)
      REAL * 8 MO2,KC
      DIMENSION  XD(7),WS(7),WX(7),KC(4),
     2WC(4),LM(7),MM(7),PZ(4,4),CNT(7,IC),GG(7,7),
     3SIGMAX(7,7),DX(7),TK(7),TT(7,IC),CN(IC,7),ZP(IC,IC)
      COMMON      /CRNT/ VPV,DX,SO2,XD,SIGMAX
      COMMON      /ANGLE/ WC,PZ
      PII       =   3.141592653589793D0
      RHO           = 180.D0/PII
      RHOS          = RHO * 3600.D0
      WC(1)     =   WC(1) / 10.D5
      DO 23 I   =   2 , IC
      WC(I)     =  WC(I)/ RHOS
 23   CONTINUE
C
C
C **** SETUP CONSTRAINTS MATRIX 'CN' REQUIRED FOR SOLUTION
C
C
      DO 25 I = 1 , IC
      DO 25 J = 1 , N
      CN(I,J) = 0.D0
      TT(J,I) = 0.D0
  25  CONTINUE
      IF (IC .EQ. 4)  GO  TO  100
      DO 50 I   =   1 , IC
      DO 50 J   =   1 , IC
  50  ZP(I,J)   =   PZ(I+1,J+1)
      GO  TO  200
 100  DO 150 I =   1 , IC
      DO 150 J =   1 , IC
 150  ZP(I,J)   =   PZ(I,J)
 200  DO 300 I= 1 , 4
      J         = I + 3
      CN(I,J) =-1.0
 300  CONTINUE
```

```
C
C ********************************************************************<<******
C                                                                              *
C **** SOLVE FOR EFFECTS OF CONSTRAINTS ON THE SOLUTION VECTOR 'DX'            *
C **** OBTAINED FROM NON-CONSTRAINT SOLUTION                                   *
C                                                                              *
C ********************************************************************<<******
C
      DO 520 I= 1 , IC
      DO 520 J= 1,N
      CNT(J,I)= CN(I,J)
 520  CONTINUE
      CALL MTPY(CNT,ZP,N,IC,IC,TT)
      CALL  MTPY(TT,CN,N,IC,N,GG)
      DO 522 I = 1 , N
      DO 522 J = 1 , N
 522  GG(I,J) =  SIGMAX(I,J) + GG(I,J)
      CALL DMINV(GG,N,DTT,LM,MM)
      CALL MTPY(TT,WC,N,IC,1,WX)
      DO 525 I= 1 , N
      WS(I)    =  (WS(I) - WX(I))
 525  CONTINUE
      CALL MTPY(GG,WS,N,N,1,XD)
C
C **** COMPUTE NEW VARIANCE OF UNIT WEIGHT AND
C **** NEW VARIANCE - COVARIANCE MATRIX
C
      CALL  MTPY(CN,XD,IC,N,1,KC)
      DO 535 I = 1 , IC
 535  KC(I)    = -KC(I)-WC(I)
      CALL  MTPY(PZ,KC,IC,IC,1,DX)
      SUM      = 0.0
      DO 540 I= 1, IC
      SUM      = SUM + DX(I) * WC(I)
 540  CONTINUE
      PVV      = VPV - SUM
      SO2      = PVV/(NN-N+IC)
      DO 550 I= 1 , N
      DO 550 J= 1 , N
      SIGMAX(I,J) =  SO2*GG(I,J)
 550  CONTINUE
      XD(4)          = XD(4) * 10.D5
      DO 560      I = 5 , 7
      XD(I)          = XD(I) * RHOS
 560  CONTINUE
1000  RETURN
      END
```

```
C
C
C ******************************************************************<<****
C                                                                         *
C                                  SETUP                                  *
C                                                                         *
C                                                                         *
C **** SETUP MATRIX B'TP -- MI -- AFTER READING VARIANCE                  *
C **** COVARIANCE MATRIX FOR EACH POINT SEPARATELY AND                    *
C **** THEN STORING THE ELEMENTS SO FORMED IN THE PROPER PLACE IN 'MI'.   *
C                                                                         *
C ******************************************************************<<****
C
C
C
C
      SUBROUTINE  SETUP (NO,NN,IPARA)
      IMPLICIT REAL *8(A-H,O-Z)
      REAL * 8 MINK
      DIMENSION B(3,6),BT(6,3),PI(6,6),PK(3,3),
     2XM(3,3),XK(3,6),MINK(2400),LM(3),MM(3),KCODE(15)
      COMMON      /RES/  BT
      COMMON      /WEIGHT/ MINK
      COMMON      /CODE/  KCODE
C
C
C **** SETTING UP MATRIX 'B' WHICH WILL BE SAME FOR ALL SOLUTION
C
C
      NV        =  NN*3
      DO 8 I    = 1 , NV
      MINK(I) = 0.D0
    8 CONTINUE
      DO 10 I = 1 , 3
      DO 10 J = 1 , 6
      B(I,J) = 0.D0
      XK(I,J)    = 0.D0
   10 CONTINUE
      B(1,1)    = -1.D0
      B(2,2)    = -1.D0
      B(3,3)    = -1.D0
      B(1,4)    =  1.D0
      B(2,5)    =  1.D0
      B(3,6)    =  1.D0
      DO 12 I = 1 , 3
      DO 12 J = 1 , 6
      BT(J,I) = B(I,J)
   12 CONTINUE
      DO 15 I = 1 , 6
      DO 15 J = 1 , 6
      PI(I,J) = 0.D0
   15 CONTINUE
      DO 20 I = 1 , 3
      DO 20 J = 1 , 3
      XM(I,J)  = 0.D0
   20 CONTINUE
      KMS      = 0
      IF      (KCODE(14) .EQ. 1)  GO  TO  65
```

```
      IF          (KCODE(8) .EQ. 1)  GO  TO  54
      DO 40 L  =    1 , NO
      DO 39 J  =    1 , 3
      READ(5,38)  (PI(J,K), K=1,3)
   38 FORMAT(3F5.2)
   39 WRITE (2)     (PI(J,K) , K = 1 , 3)
   40 CONTINUE
      DO 52 M  =    1 , NO
      DO 44 J = 4 , 6
      READ(5,42)  (PI(J,K), K= 4, 6)
   42 FORMAT(3F5.2)
   44 CONTINUE
      DO 45 I  =  1 , 3
      DO 45 J  =  1 , 3
   45 PK(I,J)  =    PI(I+3,J+3)
      DO 50 I  =  1 , 3
   50 WRITE(1)  (PK(I,J), J=1,3)
   52 CONTINUE
      REWIND   1
      REWIND   2
      GO  TO  65
   54 DO 58 M  =    1 , NO
      READ (5,55)  (PI(I,I) , I= 1,3)
   55 FORMAT (3F10.2)
      DO 56 I  =  1 , 3
   56 WRITE (2) (PI(I,J), J=1,3)
   58 CONTINUE
      DO 64 M  =    1 , NO
      READ (5,55)  (PI(I,I) , I= 4,6)
      DO 60 J  =  1 , 3
      DO 60 K  =  1 , 3
   60 PK(J,K)  =  PI(J+3,K+3)
      DO 62 I  =  1 , 3
   62 WRITE(1) (PK(I,J), J = 1,3)
   64 CONTINUE
      REWIND   1
      REWIND   2
   65 DO 100 I =    1 , NO
      KMS      =  (I-1)*9 + 1
C
C
C **** READ IN VARIANCE - COVARIANCE MATRIX AS BLOCK DIAGONALS
C **** OF (6,6) MATRICES FOR EACH POINT USED IN TRANSFORMATION.
C **** MATRIX 'PI' IS BUILT UP POINTWISE - FIRST (3,3) BLOCK
C **** REFERS TO SECOND COORDINATE SYSTEM AND SECOND (3,3) BLOCK
C **** THEN CORRESPONDS TO FIRST  COORDINATE SYSTEM.
C
C
      DO 70 J = 1 , 3
      READ(2)    (PI(J,K), K= 1,3)
   70 CONTINUE
      DO 74 L = 4 , 6
      READ(1)    (PI(L,M), M=4,6)
   74 CONTINUE
      CALL   MTPY(B,PI,3,6,6,XK)
      CALL MTPY(XK,BT,3,6,3,XM)
      CALL    DMINV(XM,3,DET,LM,MM)
      MINK(KMS  ) = XM(1,1)
```

```
      MINK(KMS+1) = XM(2,1)
      MINK(KMS+2) = XM(3,1)
      MINK(KMS+3) = XM(1,2)
      MINK(KMS+4) = XM(2,2)
      MINK(KMS+5) = XM(3,2)
      MINK(KMS+6) = XM(1,3)
      MINK(KMS+7) = XM(2,3)
      MINK(KMS+8) = XM(3,3)
  100 CONTINUE
      REWIND 1
      REWIND 2
      RETURN
      END
```

```
C ********                                                                ******** *
C ********                                                                ******** *
C ********                                                                ******** *
C ********<<<<<<<<<<*******************************************<<<<<<<<<<******** *
C ********                                                                ******** *
C ********                                                                ******** *
C ********                    SUBROUTINE   'RESIDUE'                      ******** *
C ********                                                                ******** *
C ********                                                                ******** *
C ********<<<<<<<<<<*******************************************<<<<<<<<<<******** *
C ********                                                                ******** *
C ********                                                                ******** *
C ********            THIS SUBROUTINE COMPUTES RESIDUALS FOR EACH         ******** *
C ********        SYSTEM COORDINATES (USED AS OBSERVATIONS) .             ******** *
C ********                                                                ******** *
C ********                                                                ******** *
C ********<<<<<<<<<<*******************************************<<<<<<<<<<******** *
C ********<<<<<<<<<<*******************************************<<<<<<<<<<******** *
C ********                                                                ******** *
C ********                                                                ******** *
C ********                                                                ******** *
C ********                                                                ******** *
C ********                                                                ******** *
C ********                                                                ******** *
      SUBROUTINE   RESIDU  (NO , NNN)
      IMPLICIT     REAL * 8 (A-H , O-Z)
      DIMENSION    NSTA(35),KSTA(35),UVW(35,3),MM(6),BT(6,3),LM(6),
     2             XYZ(35,3),A(3600),W(1200),PI(6,6),KCODE(15),BS(6,3)
      COMMON       /RES/  BT
      COMMON       /CODE/ KCODE
      COMMON       NSTA,KSTA,NN,NM,UVW,XYZ,A,W
      DO   5 I  =  1 , 6
      DO   5 J  =  1 , 6
    5 PI(I,J)   =  0.D0
      DO  25 I  =  1 , NO
      JJ        =  NNN + (I-1)*6
      DO  10 J  =  1 , 3
      L         =  J + 3
C
C
C ********
C ********            READING VARIANCE - COVARIANCE MATRICES FOR
C ********        THE FIRST SYSTEM -- POINTWISE -- AS (3X3) .
C ********
C
C
      READ         (1)  (PI(L,M), M = 4,6)
C
C
C ********
C ********            READING VARIANCE - COVARIANCE MATRICES FOR
C ********        THE SECOND SYSTEM -- POINTWISE -- AS (3X3) .
C ********
C
C
      READ         (2)  (PI(J,K), K = 1,3)
   10 CONTINUE
```

```
      CALL          DGMPRD (PI,BT,BS,6,6,3)
C ********                                                               ******** *
C ********<<<<<<<<<<*********************************************<<<<<<<<<<******** *
C ********                                                               ******** *
C ********                  COMPUTING   RESIDUALS                        ******** *
C ********                                                               ******** *
C ********<<<<<<<<<<*********************************************<<<<<<<<<<******** *
C ********                                                               ******** *
      DO 15 K  =    1 , 6
      KK       =    JJ + K
      A(KK)    =    0.D0
      KM       =    (I-1) * 3
      DO 15 L  =    1 , 3
      KM       =    KM + 1
   15 A(KK)    =    A(KK) + BS(K,L) * A(KM)
      DO 20 L  =    1 , 3
      LL       =    JJ + L
      KM       =    LL + 3
      W(L)     =    A(LL)
      A(LL)    =    A(KM)
   20 A(KM)    =    W(L)
   25 CONTINUE
      REWIND        1
      REWIND        2
      RETURN
      END
```

```
C
C
C
C ***********************************************************************<<****
C                                                                             *
C                                   DARRAY                                    *
C                                                                             *
C ***********************************************************************<<****
C                                                                             *
C **** SUBROUTINE SETS UP A VARIABLE DIMENSIONED MATRIX IN                    *
C **** PROPER STORAGE MODE AS REQUIRED BY ' SSP LIBRARY'                      *
C                                                                             *
C ***********************************************************************<<****
C
C                                                                             *
C
      SUBROUTINE DARRAY(MODE,I,J,N,M,S,D)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION S(1),D(1)
      NI=N-I
      IF(MODE-1) 100,100,120
  100 IJ=I*J+1
      NM=N*J+1
      DO 110 K=1,J
      NM=NM-NI
      DO 110 L=1,I
      IJ=IJ-1
      NM=NM-1
  110 D(NM)=S(IJ)
      GO TO 140
  120 IJ=0
      NM=0
      DO 130 K=1,J
      DO 125 L=1,I
      IJ=IJ+1
      NM=NM+1
  125 S(IJ)=D(NM)
  130 NM=NM+NI
  140 RETURN
      END
```

```
C
C
C ***************************************************************************<<*****
C                                                                                  *
C                                                                                  *
C                                    MTPY                                          *
C                                                                                  *
C                                                                                  *
C ***************************************************************************<<*****
C                                                                                  *
C                                                                                  *
C **** MULTIPLY TWO MATRICES -- FINAL OUTPUT IS A MATRIX.                          *
C                                                                                  *
C                                                                                  *
C ***************************************************************************<<*****
C
C
C
      SUBROUTINE  MTPY(AMT,BMT,M1,M2,M3,CMT)
      IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION  AMT(M1,M2),BMT(M2,M3),CMT(M1,M3)
      DO 10  I = 1 , M1
      DO 10  J = 1 , M3
      CMT(I,J) = 0.0
      DO 10 L = 1 , M2
   10 CMT(I,J) = CMT(I,J) + AMT(I,L) * BMT(L,J)
      RETURN
      END
```

APPENDIX II

Job Control Cards

APPENDIX II

```
//    (2500,100),CLASS=C
//STEP1 EXEC PROC=FORTRANG,PARM='MAP,ID',TIME.CMP=(0,30)
//CMP.SYSIN DD *
```

FORTRAN PROGRAM DECK

```
/*
//STEP2 EXEC PROC=RUNFORT,PARM.LKED='OVLY,LIST,MAP',TIME.LKED=(0,20),
//    TIME.GO=(3,10),REGION.GO=252K
//LKED.SYSLIB DD DSNAME=SYS1.FORTLIB,DISP=SHR
//             DD DSNAME=SYS2.FORTSSP,DISP=SHR
//LKED.SYSLIN DD DSNAME=*.STEP1.CMP.SYSLIN,DISP=(OLD,DELETE)
//   DD   *
    OVERLAY    ALPHA
  INSERT       EULERS,SCALE
  OVERLAY      BETA
  INSERT       TFORM,RESIDU,MTPY,SETUP,CSTRNT,DARRAY
/*
//GO.FT01F001 DD   UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
//             DCB=(RECFM=VBS,LRECL=600,BLKSIZE=604)
//GO.FT02F001 DD   UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
//             DCB=(RECFM=VBS,LRECL=600,BLKSIZE=604)
//GO.FT03F001 DD   UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
//             DCB=(RECFM=VBS,LRECL=600,BLKSIZE=604)
//GO.FT04F001 DD   UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
//             DCB=(RECFM=VBS,LRECL=600,BLKSIZE=604)
//GO.FT07F001 DD SYSOUT=B
//GO.SYSIN DD *
```

DATA DECK

```
/*
//
```